

OCF Core Specification

VERSION 2.0.2 | April 2019



OPEN CONNECTIVITY
FOUNDATION™

CONTACT admin@openconnectivity.org
Copyright Open Connectivity Foundation, Inc. © 2019
All Rights Reserved.



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

Legal Disclaimer

NOTHING CONTAINED IN THIS DOCUMENT SHALL BE DEEMED AS GRANTING YOU ANY KIND OF LICENSE IN ITS CONTENT, EITHER EXPRESSLY OR IMPLIEDLY, OR TO ANY INTELLECTUAL PROPERTY OWNED OR CONTROLLED BY ANY OF THE AUTHORS OR DEVELOPERS OF THIS DOCUMENT. THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN "AS IS" BASIS, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE AUTHORS AND DEVELOPERS OF THIS SPECIFICATION HEREBY DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OPEN CONNECTIVITY FOUNDATION, INC. FURTHER DISCLAIMS ANY AND ALL WARRANTIES OF NON-INFRINGEMENT, ACCURACY OR LACK OF VIRUSES.

The OCF logo is a trademark of Open Connectivity Foundation, Inc. in the United States or other countries. *Other names and brands may be claimed as the property of others.

Copyright © 2016-2019 Open Connectivity Foundation, Inc. All rights reserved.

Copying or other form of reproduction and/or distribution of these works are strictly prohibited.

CONTENTS

20			
21	1	Scope.....	1
22	2	Normative references	1
23	3	Terms, definitions, and abbreviated terms	3
24	3.1	Terms and definitions	3
25	3.2	Abbreviated terms	7
26	4	Document conventions and organization.....	8
27	4.1	Conventions	8
28	4.2	Notation	9
29	4.3	Data types	9
30	5	Architecture.....	11
31	5.1	Overview	11
32	5.2	Principle.....	11
33	5.3	Functional block diagram	12
34	5.4	Framework	14
35	5.5	Example Scenario with roles	14
36	5.6	Example Scenario: Bridging to Non- OCF ecosystem	15
37	5.7	OCF Cloud architecture.....	16
38	6	Identification and addressing	18
39	6.1	Introduction	18
40	6.2	Identification.....	18
41	6.2.1	Overview.....	18
42	6.2.2	Resource identification and addressing.....	18
43	6.3	Namespace:	19
44	6.4	Network addressing.....	20
45	7	Resource model.....	20
46	7.1	Introduction	20
47	7.2	Resource.....	21
48	7.3	Property	21
49	7.3.1	Introduction.....	21
50	7.3.2	Common Properties.....	22
51	7.4	Resource Type	24
52	7.4.1	Introduction.....	24
53	7.4.2	Resource Type Property.....	24
54	7.4.3	Resource Type definition.....	24
55	7.4.4	Multi-value "rt" Resource.....	26
56	7.5	Device Type	26
57	7.6	OCF Interface	27
58	7.6.1	Introduction.....	27
59	7.6.2	OCF Interface Property.....	27
60	7.6.3	OCF Interface methods.....	28
61	7.7	Resource representation.....	42
62	7.8	Structure	42

63	7.8.1	Introduction.....	42
64	7.8.2	Resource Relationships	42
65	7.8.3	Collections	48
66	7.8.4	Atomic Measurement	50
67	7.9	Third (3 rd) party specified extensions	52
68	7.10	Query Parameters	53
69	7.10.1	Introduction.....	53
70	7.10.2	Use of multiple parameters within a query.....	53
71	7.10.3	Application to multi-value "rt" Resources	54
72	7.10.4	OCF Interface specific considerations for queries.....	54
73	8	CRUDN.....	54
74	8.1	Overview	54
75	8.2	CREATE.....	55
76	8.2.1	Overview.....	55
77	8.2.2	CREATE request	55
78	8.2.3	Processing by the Server	56
79	8.2.4	CREATE response	56
80	8.3	RETRIEVE	56
81	8.3.1	Overview.....	56
82	8.3.2	RETRIEVE request.....	56
83	8.3.3	Processing by the Server	56
84	8.3.4	RETRIEVE response.....	57
85	8.4	UPDATE.....	57
86	8.4.1	Overview.....	57
87	8.4.2	UPDATE request	57
88	8.4.3	Processing by the Server	57
89	8.4.4	UPDATE response	58
90	8.5	DELETE	58
91	8.5.1	Overview.....	58
92	8.5.2	DELETE request.....	58
93	8.5.3	Processing by the Server	59
94	8.5.4	DELETE response.....	59
95	8.6	NOTIFY.....	59
96	8.6.1	Overview.....	59
97	8.6.2	NOTIFICATION response	59
98	9	Network and connectivity	59
99	9.1	Introduction	59
100	9.2	Architecture.....	60
101	9.3	IPv6 network layer requirements	61
102	9.3.1	Introduction.....	61
103	9.3.2	IPv6 node requirements	61
104	10	OCF Endpoint	61
105	10.1	OCF Endpoint definition.....	61
106	10.2	OCF Endpoint information.....	62

107	10.2.1	Introduction.....	62
108	10.2.2	"ep".....	62
109	10.2.3	"pri".....	63
110	10.2.4	OCF Endpoint information in "eps" Parameter.....	63
111	10.3	OCF Endpoint discovery.....	64
112	10.3.1	Introduction.....	64
113	10.3.2	Implicit discovery.....	64
114	10.3.3	Explicit discovery with "/oic/res" response.....	64
115	10.4	CoAP based OCF Endpoint discovery.....	69
116	11	Functional interactions.....	70
117	11.1	Introduction.....	70
118	11.2	Onboarding, Provisioning and Configuration.....	70
119	11.3	Resource discovery.....	72
120	11.3.1	Introduction.....	72
121	11.3.2	Resource based discovery: mechanisms.....	72
122	11.3.3	Resource based discovery: Information publication process.....	74
123	11.3.4	Resource based discovery: Finding information.....	75
124	11.3.5	Resource discovery using "/oic/res".....	80
125	11.3.6	Resource Directory (RD) based discovery.....	83
126	11.4	Notification.....	94
127	11.4.1	Overview.....	94
128	11.4.2	Observe.....	94
129	11.5	Device management.....	95
130	11.5.1	Overview.....	95
131	11.5.2	Diagnostics and maintenance.....	95
132	11.5.3	Network monitoring.....	96
133	11.6	Scenes.....	100
134	11.6.1	Introduction.....	100
135	11.6.2	Scenes.....	100
136	11.6.3	Security considerations.....	105
137	11.7	Icons.....	106
138	11.7.1	Overview.....	106
139	11.7.2	Resource.....	106
140	11.8	Introspection.....	106
141	11.8.1	Overview.....	106
142	11.8.2	Usage of Introspection.....	109
143	11.9	Alerts.....	110
144	11.9.1	Overview.....	110
145	11.9.2	Resource Types.....	111
146	11.9.3	Example of Use.....	112
147	12	Messaging.....	112
148	12.1	Introduction.....	112
149	12.2	Mapping of CRUDN to CoAP.....	113
150	12.2.1	Overview.....	113

151	12.2.2	URIs.....	113
152	12.2.3	CoAP method with request and response.....	113
153	12.2.4	Content-Format negotiation.....	115
154	12.2.5	OCF-Content-Format-Version information.....	116
155	12.2.6	Content-Format policy.....	116
156	12.2.7	CRUDN to CoAP response codes.....	119
157	12.2.8	CoAP block transfer.....	119
158	12.3	Mapping of CRUDN to CoAP serialization over TCP.....	119
159	12.3.1	Overview.....	119
160	12.3.2	URIs.....	119
161	12.3.3	CoAP method with request and response.....	120
162	12.3.4	Content-Format negotiation.....	120
163	12.3.5	OCF-Content-Format-Version information.....	120
164	12.3.6	Content-Format policy.....	120
165	12.3.7	CRUDN to CoAP response codes.....	120
166	12.3.8	CoAP block transfer.....	120
167	12.3.9	Keep alive (connection health).....	120
168	12.4	Payload Encoding in CBOR.....	120
169	13	Security.....	121
170	Annex A	(informative) Operation Examples.....	122
171	A.1	Introduction.....	122
172	A.2	When at home: From smartphone turn on a single light.....	122
173	A.3	GroupAction execution.....	123
174	A.4	When garage door opens, turn on lights in hall; also notify smartphone.....	123
175	A.5	Device management.....	123
176	Annex B	(informative) OCF interaction scenarios and deployment models.....	125
177	B.1	OCF interaction scenarios.....	125
178	B.2	Deployment model.....	126
179	Annex C	(informative) Other Resource models and OCF mapping.....	128
180	C.1	Multiple Resource models.....	128
181	C.2	OCF approach for support of multiple Resource models.....	128
182	C.3	Resource model indication.....	129
183	C.4	An Example Profile (OMA SpecWorks profile).....	129
184	C.4.1	Overview.....	129
185	C.5	Conceptual equivalence.....	129
186	C.5.1	Resource Type: Light Control.....	130
187	Annex D	(normative) Resource Type definitions.....	132
188	D.1	List of Resource Type definitions.....	132
189	D.2	Atomic Measurement links list representation.....	132
190	D.2.1	Introduction.....	132
191	D.2.2	Example URI.....	132
192	D.2.3	Resource type.....	132
193	D.2.4	OpenAPI 2.0 definition.....	133
194	D.2.5	Property definition.....	139

195	D.2.6	CRUDN behaviour.....	140
196	D.3	Collection	140
197	D.3.1	Introduction.....	140
198	D.3.2	Example URI.....	140
199	D.3.3	Resource type	140
200	D.3.4	OpenAPI 2.0 definition.....	140
201	D.3.5	Property definition.....	148
202	D.3.6	CRUDN behaviour.....	149
203	D.4	Device Configuration	149
204	D.4.1	Introduction.....	149
205	D.4.2	Example URI.....	149
206	D.4.3	Resource type	149
207	D.4.4	OpenAPI 2.0 definition.....	149
208	D.4.5	Property definition.....	154
209	D.4.6	CRUDN behaviour.....	154
210	D.5	Platform Configuration	155
211	D.5.1	Introduction.....	155
212	D.5.2	Example URI.....	155
213	D.5.3	Resource type	155
214	D.5.4	OpenAPI 2.0 definition.....	155
215	D.5.5	Property definition.....	158
216	D.5.6	CRUDN behaviour.....	158
217	D.6	Device	158
218	D.6.1	Introduction.....	158
219	D.6.2	Well-known URI	158
220	D.6.3	Resource type	158
221	D.6.4	OpenAPI 2.0 definition.....	159
222	D.6.5	Property definition.....	162
223	D.6.6	CRUDN behaviour.....	162
224	D.7	Icon	162
225	D.7.1	Introduction.....	162
226	D.7.2	Example URI.....	163
227	D.7.3	Resource type	163
228	D.7.4	OpenAPI 2.0 definition.....	163
229	D.7.5	Property definition.....	164
230	D.7.6	CRUDN behaviour.....	165
231	D.8	Introspection Resource	165
232	D.8.1	Introduction.....	165
233	D.8.2	Well-known URI	165
234	D.8.3	Resource type	165
235	D.8.4	OpenAPI 2.0 definition.....	165
236	D.8.5	Property definition.....	168
237	D.8.6	CRUDN behaviour.....	168
238	D.9	Maintenance.....	168

239	D.9.1	Introduction.....	168
240	D.9.2	Well-known URI	168
241	D.9.3	Resource type	168
242	D.9.4	OpenAPI 2.0 definition.....	169
243	D.9.5	Property definition.....	171
244	D.9.6	CRUDN behaviour.....	172
245	D.10	Network Monitoring.....	172
246	D.10.1	Introduction.....	172
247	D.10.2	Example URI.....	172
248	D.10.3	Resource type	172
249	D.10.4	OpenAPI 2.0 definition.....	172
250	D.10.5	Property definition.....	175
251	D.10.6	CRUDN behaviour.....	176
252	D.11	Platform	176
253	D.11.1	Introduction.....	176
254	D.11.2	Well-known URI	176
255	D.11.3	Resource type	176
256	D.11.4	OpenAPI 2.0 definition.....	176
257	D.11.5	Property definition.....	179
258	D.11.6	CRUDN behaviour.....	180
259	D.12	Resource directory resource	180
260	D.12.1	Introduction.....	180
261	D.12.2	Well-known URI	180
262	D.12.3	Resource type	180
263	D.12.4	OpenAPI 2.0 definition.....	180
264	D.12.5	Property definition.....	184
265	D.12.6	CRUDN behaviour.....	185
266	D.13	Discoverable Resources	185
267	D.13.1	Introduction.....	185
268	D.13.2	Well-known URI	185
269	D.13.3	Resource type	185
270	D.13.4	OpenAPI 2.0 definition.....	185
271	D.13.5	Property definition.....	189
272	D.13.6	CRUDN behaviour.....	190
273	D.14	Scene List.....	190
274	D.14.1	Introduction.....	190
275	D.14.2	Example URI.....	190
276	D.14.3	Resource type	191
277	D.14.4	OpenAPI 2.0 definition.....	191
278	D.14.5	Property definition.....	194
279	D.14.6	CRUDN behaviour.....	195
280	D.15	Scene Collection.....	195
281	D.15.1	Introduction.....	195
282	D.15.2	Example URI.....	195

283	D. 15.3	Resource type	195
284	D. 15.4	OpenAPI 2.0 definition.....	196
285	D. 15.5	Property definition.....	200
286	D. 15.6	CRUDN behaviour.....	201
287	D.16	Scene Member	201
288	D. 16.1	Introduction.....	201
289	D. 16.2	Example URI.....	201
290	D. 16.3	Resource type	201
291	D. 16.4	OpenAPI 2.0 definition.....	201
292	D. 16.5	Property definition.....	205
293	D. 16.6	CRUDN behaviour.....	206
294	D.17	Alert.....	206
295	D. 17.1	Introduction.....	206
296	D. 17.2	Example URI.....	206
297	D. 17.3	Resource type	206
298	D. 17.4	OpenAPI 2.0 definition.....	206
299	D. 17.5	Property definition.....	209
300	D. 17.6	CRUDN behaviour.....	209
301	D.18	Alert Collection.....	210
302	D. 18.1	Introduction.....	210
303	D. 18.2	Example URI.....	210
304	D. 18.3	Resource type	210
305	D. 18.4	OpenAPI 2.0 definition.....	210
306	D. 18.5	Property definition.....	214
307	D. 18.6	CRUDN behaviour.....	215
308	Annex E (informative)	OIC 1.1 Resource Type definitions.....	216
309	E. 1	List of Resource Type Definitions	216
310	E. 2	OCF Collection.....	216
311	E. 2.1	Introduction.....	216
312	E. 2.2	Wellknown URI	216
313	E. 2.3	Resource type	216
314	E. 2.4	OpenAPI 2.0 definition.....	216
315	E. 2.5	Property definition.....	226
316	E. 2.6	CRUDN behaviour.....	228
317	E. 3	Discoverable Resources	228
318	E. 3.1	Introduction.....	228
319	E. 3.2	Wellknown URI	228
320	E. 3.3	Resource type	228
321	E. 3.4	OpenAPI 2.0 definition.....	228
322	E. 3.5	Property definition.....	233
323	E. 3.6	CRUDN behaviour.....	235
324	Annex F (informative)	OpenAPI 2.0 Schema Extension.....	236
325	F. 1	OpenAPI 2.0 Schema Reference	236
326	F. 2	OpenAPI 2.0 Introspection empty file.....	236

327

328

329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369

Figures

Figure 1 – Architecture - concepts.....	12
Figure 2 – Functional block diagram.....	13
Figure 3 – Communication layering model.....	14
Figure 4 – Example illustrating the roles.....	15
Figure 5 – Framework - Architecture Detail.....	16
Figure 6 – Server bridging to Non- OCF device.....	16
Figure 7 – OCF Cloud deployment architecture.....	17
Figure 8 – OCF Endpoint routing.....	17
Figure 9 – Example Resource.....	21
Figure 10 – CREATE operation.....	55
Figure 11 – RETRIEVE operation.....	56
Figure 12 – UPDATE operation.....	57
Figure 13 – DELETE operation.....	58
Figure 14 – High Level Network & Connectivity Architecture.....	60
Figure 15 – Resource based discovery: Information publication process.....	74
Figure 16 – Resource based discovery: Finding information.....	75
Figure 17 – Indirect discovery of Resources by via an RD.....	83
Figure 18 – RD discovery and RD supported query of Resources support.....	85
Figure 19 – Resource Direction Deployment Scenarios.....	86
Figure 20 – Observe Mechanism.....	94
Figure 21 – Interactions with the network monitoring Resource.....	99
Figure 22 – State transition diagram of collecting network information.....	100
Figure 23 – Generic Scene Resource structure.....	101
Figure 24 – Interactions to check Scene support and setup of specific Scenes.....	102
Figure 25 – Client interactions on a specific Scene.....	103
Figure 26 – Interaction overview due to a Scene change.....	105
Figure 27 – Example usage of oneOf JSON schema.....	109
Figure 28 – Interactions to check Introspection support and download the Introspection Device Data.....	110
Figure 29 – Content-Format Policy for OCF Servers supporting error responses and backward compatibility responses.....	118
Figure 30 – Content-Format Policy for OCF Clients supporting error responses and backward compatibility responses.....	118
Figure 31 – Content-Format Policy for backward compatible OCF Clients negotiating lower OCF Content-Format-Version.....	119
Figure A.1 – When at home: from smartphone turn on a single light.....	123
Figure A.2 – Device management (maintenance).....	124
Figure B.1 – Direct interaction between Server and Client.....	125

370 Figure B.2– Interaction between Client and Server using another Server 125
371 Figure B.3 – Interaction between Client and Server using Intermediary 125
372 Figure B.4 – Interaction between Client and Server using support from multiple Servers
373 and Intermediary 126
374 Figure B.5 – Example of Devices 126

375

Tables

376
377

378 Table 1 – Additional OCF Types 10
379 Table 2 – Name Property Definition 23
380 Table 3 – Resource Identity Property Definition 23
381 Table 4 – Resource Type Common Property definition 24
382 Table 5 – Example foobar Resource Type 25
383 Table 6 – Example foobar Properties 25
384 Table 7 – Resource Interface Property definition 27
385 Table 8 – OCF standard OCF Interfaces 28
386 Table 9 – Batch OCF Interface Example 33
387 Table 10 – "bm" Property definition 44
388 Table 11 – Resource Types Property definition 47
389 Table 12 – Mandatory Resource Types Property definition 47
390 Table 13 – Common Properties for Collections (in addition to Common Properties defined
391 in 7.3.2) 49
392 Table 14 – Common Properties for Atomic Measurement (in addition to Common
393 Properties defined in 7.3.2) 50
394 Table 15 – Atomic Measurement Resource Type 52
395 Table 16 – Properties for Atomic Measurement (in addition to Common Properties defined
396 in 7.3.2) 52
397 Table 17 – 3rd party defined Resource elements 52
398 Table 18 – Parameters of CRUDN messages 55
399 Table 19 – "ep" value for Transport Protocol Suite 63
400 Table 20 – List of Core Resources 70
401 Table 21 – Configuration Resource 70
402 Table 22 – "oic.wk.con" Resource Type definition 71
403 Table 23 – "oic.wk.con.p" Resource Type definition 72
404 Table 24 – Mandatory discovery Core Resources 76
405 Table 25 – "oic.wk.res" Resource Type definition 76
406 Table 26 – Protocol scheme registry 77
407 Table 27 – "oic.wk.d" Resource Type definition 78
408 Table 28 – "oic.wk.p" Resource Type definition 80
409 Table 29 – "oic.wk.rd" Resource Type definition 84

410	Table 30 – "oic.wk.rd" Properties	84
411	Table 31 – Optional diagnostics and maintenance Device management Core Resources.....	96
412	Table 32 – "oic.wk.mnt" Resource Type definition.....	96
413	Table 33 – Optional monitoring Device management Core Resources.....	97
414	Table 34 – "oic.wk.nmon" Resource Type definition	97
415	Table 35 – list of Resource Types for Scenes.....	105
416	Table 36 – Optional Icon Core Resource	106
417	Table 37 – "oic.r.icon" Resource Type definition	106
418	Table 38 – Introspection Resource	109
419	Table 39 – "oic.wk.introspection" Resource Type definition.....	109
420	Table 40 – Optional Alert Core Resources	111
421	Table 41 – "oic.r.alert" Resource Type definition.....	111
422	Table 42 – "oic.r.alertcollection" Resource Type definition.....	112
423	Table 43 – CoAP request and response	113
424	Table 44 – OCF Content-Formats.....	115
425	Table 45 – OCF-Content-Format-Version and OCF-Accept-Content-Format-Version Option	
426	Numbers	116
427	Table 46 – OCF-Accept-Content-Format-Version and OCF-Content-Format-Version	
428	Representation	116
429	Table 47 – Examples of OCF-Content-Format-Version and OCF-Accept-Content-Format-	
430	Version Representation	116
431	Table A-1 – "oic.example.light" Resource Type definition.....	122
432	Table A-2 – oic.example.garagedoor Resource Type definition	122
433	Table C-1 – Light control Resource Type definition.....	130
434	Table C-2 – Light control Resource Type definition.....	130
435	Table D-1 – Alphabetized list of Core Resources.....	132
436	Table D-2 – The Property definitions of the Resource with type "rt" =	
437	"oic.wk.atomicmeasurement".	139
438	Table D-3 – The CRUDN operations of the Resource with type "rt" =	
439	"oic.wk.atomicmeasurement".	140
440	Table D-4 – The Property definitions of the Resource with type "rt" = "oic.wk.col".	148
441	Table D-5 – The CRUDN operations of the Resource with type "rt" = "oic.wk.col".....	149
442	Table D-6 – The Property definitions of the Resource with type "rt" = "oic.wk.con".	154
443	Table D-7 – The CRUDN operations of the Resource with type "rt" = "oic.wk.con".	154
444	Table D-8 – The Property definitions of the Resource with type "rt" = "oic.wk.con.p".	158
445	Table D-9 – The CRUDN operations of the Resource with type "rt" = "oic.wk.con.p".....	158
446	Table D-10 – The Property definitions of the Resource with type "rt" = "oic.wk.d".....	162
447	Table D-11 – The CRUDN operations of the Resource with type "rt" = "oic.wk.d".	162
448	Table D-12 – The Property definitions of the Resource with type "rt" = "oic.r.icon".....	165
449	Table D-13 – The CRUDN operations of the Resource with type "rt" = "oic.r.icon".	165

450	Table D-14 – The Property definitions of the Resource with type "rt" =	
451	"oic.wk.introspection".	168
452	Table D-15 – The CRUDN operations of the Resource with type "rt" =	
453	"oic.wk.introspection".	168
454	Table D-16 – The Property definitions of the Resource with type "rt" = "oic.wk.mnt".	171
455	Table D-17 – The CRUDN operations of the Resource with type "rt" = "oic.wk.mnt".	172
456	Table D-18 – The Property definitions of the Resource with type "rt" = "oic.wk.nmon".	175
457	Table D-19 – The CRUDN operations of the Resource with type "rt" = "oic.wk.nmon".	176
458	Table D-20 – The Property definitions of the Resource with type "rt" = "oic.wk.p".	179
459	Table D-21 – The CRUDN operations of the Resource with type "rt" = "oic.wk.p".	180
460	Table D-22 – The Property definitions of the Resource with type "rt" = "oic.wk.rd".	185
461	Table D-23 – The CRUDN operations of the Resource with type "rt" = "oic.wk.rd".	185
462	Table D-24 – The Property definitions of the Resource with type "rt" = "None".	189
463	Table D-25 – The CRUDN operations of the Resource with type "rt" = "None".	190
464	Table D-26 – The Property definitions of the Resource with type "rt" = "oic.wk.scenelist".	194
465	Table D-27 – The CRUDN operations of the Resource with type "rt" = "oic.wk.scenelist".	195
466	Table D-28 – The Property definitions of the Resource with type "rt" =	
467	"oic.wk.scenecollection".	200
468	Table D-29 – The CRUDN operations of the Resource with type "rt" =	
469	"oic.wk.scenecollection".	201
470	Table D-30 – The Property definitions of the Resource with type "rt" =	
471	"oic.wk.scenemember".	205
472	Table D-31 – The CRUDN operations of the Resource with type "rt" =	
473	"oic.wk.scenemember".	206
474	Table D-32 – The Property definitions of the Resource with type "rt" = "oic.r.alert".	209
475	Table D-33 – The CRUDN operations of the Resource with type "rt" = "oic.r.alert".	209
476	Table D-34 – The Property definitions of the Resource with type "rt" =	
477	"oic.r.alertcollection".	214
478	Table D-35 – The CRUDN operations of the Resource with type "rt" =	
479	"oic.r.alertcollection".	215
480	Table E.1 – Alphabetized list of referenced OIC 1.1 Core Resources	216
481	Table E.2 – The Property definitions of the Resource with type 'rt' = ['oic.wk.col']	226
482	Table E.3 – The CRUDN operations of the Resource with type 'rt' = ['oic.wk.col']	228
483	Table E.4 – The Property definitions of the Resource with type 'rt' = ['oic.wk.res']	233
484	Table E.5 – The CRUDN operations of the Resource with type 'rt' = ['oic.wk.res']	235
485		
486		

487

488 1 Scope

489 The OCF specifications are divided into two sets of documents:

- 490 – Core Specification documents: The Core Specification documents specify the Framework, i.e.,
491 the OCF core architecture, interfaces, protocols and services to enable OCF profiles
492 implementation for Internet of Things (IoT) usages and ecosystems.
- 493 – Vertical Domain Specification documents: The Vertical Domain Specification documents specify
494 OCF Device profiles to enable IoT usages for different vertical market segments such as smart
495 home, industrial, healthcare, and automotive. They also specify Resource definitions to enable
496 vertical services and use case. Such specifications include ISO/IEC 30118-5:2018 which is built
497 upon the interfaces and network security of the OCF core architecture defined in the Core
498 Specification.

499 This document is the OCF Core specification which specifies the Framework and core architecture.

500 2 Normative references

501 The following documents, in whole or in part, are normatively referenced in this document and are
502 indispensable for its application. For dated references, only the edition cited applies. For undated
503 references, the latest edition of the referenced document (including any amendments) applies.

504 ISO 8601, *Data elements and interchange formats – Information interchange –Representation of*
505 *dates and times*, International Standards Organization, December 3, 2004

506 ISO/IEC DIS 20924, *Information Technology – Internet of Things – Vocabulary*, June 2018
507 <https://www.iso.org/standard/69470.html>

508 ISO/IEC 30118-2:2018, Information technology – Open Connectivity Foundation (OCF)
509 Specification – Part 2: Security specification
510 <https://www.iso.org/standard/74239.html>
511 Latest version available at: https://openconnectivity.org/specs/OCF_Security_Specification.pdf

512 ISO/IEC 30118-5:2018, Information technology – Open Connectivity Foundation (OCF)
513 Specification – Part 5: Smart home device specification
514 <https://www.iso.org/standard/74242.html>
515 Latest version available at: https://openconnectivity.org/specs/OCF_Device_Specification.pdf

516 OCF Easy Wi-Fi Setup, Information technology – Open Connectivity Foundation (OCF)
517 Specification – Part 7: Wi-Fi Easy Setup specification
518 Latest version available at: [https://openconnectivity.org/specs/OCF_Wi-](https://openconnectivity.org/specs/OCF_Wi-Fi_Easy_Setup_Specification_v2.0.1.pdf)
519 [Fi_Easy_Setup_Specification_v2.0.1.pdf](https://openconnectivity.org/specs/OCF_Wi-Fi_Easy_Setup_Specification_v2.0.1.pdf)

520 IETF RFC 768, *User Datagram Protocol*, August 1980
521 <https://www.rfc-editor.org/info/rfc768>

522 IETF RFC 3339, *Date and Time on the Internet: Timestamps*, July 2002
523 <https://www.rfc-editor.org/info/rfc3339>

524 IETF RFC 3986, *Uniform Resource Identifier (URI): General Syntax, January 2005*.
525 <https://www.rfc-editor.org/info/rfc3986>

526 IETF RFC 4122, *A Universally Unique Identifier (UUID) URN Namespace*, July 2005
527 <https://www.rfc-editor.org/info/rfc4122>

528 IETF RFC 4287, *The Atom Syndication Format*, December 2005,
529 <https://www.rfc-editor.org/info/rfc4287>

530 IETF RFC 4941, *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, September
531 2007
532 <https://www.rfc-editor.org/info/rfc4941>

533 IETF RFC 5424, *The Syslog Protocol*, March 2009
534 <https://tools.ietf.org/html/rfc5424>IETF RFC 5646, *Tags for Identifying Languages*, September
535 2009
536 <https://www.rfc-editor.org/info/rfc5646>

537 IETF RFC 5988, *Web Linking: General Syntax*, October 2010
538 <https://www.rfc-editor.org/info/rfc5988>

539 IETF RFC 6347, *Datagram Transport Layer Security Version 1.2*, January 2012
540 <https://www.rfc-editor.org/info/rfc6347>

541 IETF RFC 6434, *IPv6 Node Requirements*, December 2011
542 <https://www.rfc-editor.org/info/rfc6434>

543 IETF RFC 6573, *The Item and Collection Link Relations*, April 2012
544 <https://www.rfc-editor.org/info/rfc6573>

545 IETF RFC 6690, *Constrained RESTful Environments (CoRE) Link Format*, August 2012
546 <https://www.rfc-editor.org/info/rfc6690>

547 IETF RFC 7049, *Concise Binary Object Representation (CBOR)*, October 2013
548 <https://www.rfc-editor.org/info/rfc7049>

549 IETF RFC 7084, *Basic Requirements for IPv6 Customer Edge Routers*, November 2013
550 <https://www.rfc-editor.org/info/rfc7084>

551 IETF RFC 7159, *The JavaScript Object Notation (JSON) Data Interchange Format*, March 2014
552 <https://www.rfc-editor.org/info/rfc7159>

553 IETF RFC 7252, *The Constrained Application Protocol (CoAP)*, June 2014
554 <https://www.rfc-editor.org/info/rfc7252>

555 IETF RFC 7301, *Transport Layer Security (TLS) Application-Layer Protocol Negotiation
556 Extension*, July 2014
557 <https://www.rfc-editor.org/info/rfc7301>

558 IETF RFC 7595, *Guidelines and Registration Procedures for URI Schemes*, June 2015
559 <https://www.rfc-editor.org/info/rfc7595>

560 IETF RFC 7641, *Observing Resources in the Constrained Application Protocol
561 (CoAP)*, September 2015
562 <https://www.rfc-editor.org/info/rfc7641>

563 IETF RFC 7721, *Security and Privacy Considerations for IPv6 Address Generation Mechanisms*,
564 March 2016
565 <https://www.rfc-editor.org/info/rfc7721>

566 IETF RFC 7959, *Block-Wise Transfers in the Constrained Application Protocol (CoAP)*, August
567 2016
568 <https://www.rfc-editor.org/info/rfc7959>

569 IETF RFC 8075, *Guidelines for Mapping Implementations: HTTP to the Constrained Application
570 Protocol (CoAP)*, February 2017
571 <https://www.rfc-editor.org/info/rfc8075>

572 IETF RFC 8323, *CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets*,
573 February 2018
574 <https://www.rfc-editor.org/info/rfc8323>

575 IANA IPv6 Multicast Address Space Registry
576 <http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xhtml>

577 IANA Media Types Assignment, March 2017
578 <http://www.iana.org/assignments/media-types/media-types.xhtml>

579 IANA Link Relations, October 2017
580 <http://www.iana.org/assignments/link-relations/link-relations.xhtml>

581 JSON Schema Validation, *JSON Schema: interactive and non-interactive validation*, January 2013
582 <http://json-schema.org/draft-04/json-schema-validation.html>

583 OpenAPI specification, *fka Swagger RESTful API Documentation Specification*, Version 2.0
584 <https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md>

585 **3 Terms, definitions, and abbreviated terms**

586 **3.1 Terms and definitions**

587 For the purposes of this document, the terms and definitions given in the following apply.

588 ISO and IEC maintain terminological databases for use in standardization at the following
589 addresses:

- 590 – ISO Online browsing platform: available at <https://www.iso.org/obp>.
- 591 – IEC Electropedia: available at <http://www.electropedia.org/>.

592 **3.1.1**

593 **Alert**

594 information provided by the Device (3.1.10) by means of a specialised Resource Type (3.1.32) that
595 provides details with regard to potential problems, issues, or Device (3.1.10) status of interest on
596 which action can be taken

597 **3.1.2**

598 **Atomic Measurement**

599 a design pattern that ensures that the Client (3.1.3) can only access the Properties (3.1.31) of
600 linked Resources (3.1.28) atomically, that is as a single group

601 **3.1.3**

602 **Client**

603 a logical entity that accesses a Resource (3.1.28) on a Server (3.1.38)

604 **3.1.4**

605 **Collection**

606 a Resource (3.1.28) that contains zero or more Links (3.1.18)

607 **3.1.5**

608 **Common Properties**

609 Properties (3.1.31) specified for all Resources (3.1.28)

610 **3.1.6**

611 **Composite Device**

612 a Device (3.1.10) that is modelled as multiple Device Types (3.1.11); with each component Device
613 Type (3.1.11) being exposed as a Collection (3.1.4)

614 **3.1.7**
615 **Configuration Source**
616 a cloud or service network or a local read-only file which contains and provides configuration
617 related information to the Devices (3.1.10)

618 **3.1.8**
619 **Core Resources**
620 those Resources (3.1.28) that are defined in this document

621 **3.1.9**
622 **Default OCF Interface**
623 an OCF Interface (3.1.15) used to generate the response when an OCF Interface (3.1.15) is omitted
624 in a request

625 **3.1.10**
626 **Device**
627 a logical entity that assumes one or more roles, e.g., Client (3.1.3), Server (3.1.38)

628 Note 1 to entry: More than one Device (3.1.10) can exist on a Platform (3.1.27).

629 **3.1.11**
630 **Device Type**
631 a uniquely named definition indicating a minimum set of Resource Types (3.1.32) that a Device
632 (3.1.10) supports

633 Note 1 to entry: A Device Type (3.1.11) provides a hint about what the Device (3.1.10) is, such as a light or a fan, for
634 use during Resource (3.1.28) discovery.

635 **3.1.12**
636 **Discoverable Resource**
637 a Resource (3.1.28) that is listed in "/oic/res"

638 **3.1.13**
639 **OCF Endpoint**
640 entity participating in the OCF protocol, further identified as the source or destination of a request
641 and response messages for a given Transport Protocol Suite

642 Note 1 to entry: Example of a Transport Protocol Suite would be CoAP over UDP over IPv6.

643 **3.1.14**
644 **Framework**
645 a set of related functionalities and interactions defined in this document, which enable
646 interoperability across a wide range of networked devices, including IoT

647 **3.1.15**
648 **OCF Interface**
649 interface description in accordance with IETF RFC 6690 and as defined by OCF that provides a
650 view to and permissible responses from a Resource (3.1.28)

651 **3.1.16**
652 **Introspection**
653 mechanism to determine the capabilities of the hosted Resources (3.1.28) of a Device (3.1.10)

654 **3.1.17**
655 **Introspection Device Data (IDD)**
656 data that describes the payloads per implemented method of the Resources (3.1.28) that make up
657 the Device (3.1.10)

658 Note 1 to entry: See 11.8 for all requirements and exceptions.

659 **3.1.18**
660 **Links**
661 extends typed web links according to IETF RFC 5988

662 **3.1.19**
663 **Non-Discoverable Resource**
664 a Resource (3.1.28) that is not listed in "/oic/res"

665 Note 1 to entry: The Resource (3.1.28) can be reached by a Link (3.1.18) which is conveyed by another Resource
666 (3.1.28). For example a Resource (3.1.28) linked in a Collection (3.1.4) does not have to be listed in "/oic/res", since
667 traversing the Collection (3.1.4) would discover the Resource (3.1.28) implemented on the Device (3.1.10).

668 **3.1.20**
669 **Non-OCF Device**
670 a Device (3.1.10) which does not comply with the OCF Device (3.1.10) requirements

671 **3.1.21**
672 **Notification**
673 the mechanism to make a Client (3.1.3) aware of state changes in a Resource (3.1.28)

674 **3.1.22**
675 **Observe**
676 the act of monitoring a Resource (3.1.28) by sending a RETRIEVE operation which is cached by
677 the Server (3.1.38) hosting the Resource (3.1.28) and reprocessed on every change to that
678 Resource (3.1.28)

679 **3.1.23**
680 **OpenAPI 2.0**
681 Resource (3.1.28) and Introspection Device Data (3.1.17) definitions used in this document as
682 defined in the OpenAPI specification

683 **3.1.24**
684 **Parameter**
685 an element that provides metadata about a Resource (3.1.28) referenced by the target URI of a
686 Link (3.1.18)

687 **3.1.25**
688 **Partial UPDATE**
689 an UPDATE operation to a Resource (3.1.28) that includes a subset of the Properties (3.1.31) that
690 are visible via the OCF Interface (3.1.15) being applied for the Resource Type (3.1.32)

691 **3.1.26**
692 **Physical Device**
693 the physical thing on which a Device(s) (3.1.10) is exposed

694 **3.1.27**
695 **Platform**
696 a Physical Device (3.1.26) containing one or more Devices (3.1.10)

697 **3.1.28**
698 **Resource**
699 represents an entity modelled and exposed by the Framework (3.1.14)

700 **3.1.29**
701 **Resource Directory**
702 a set of descriptions of Resources (3.1.28) where the actual Resources (3.1.28) are held on Servers
703 (3.1.38) external to the Device (3.1.10) hosting the Resource Directory (3.1.29), allowing lookups
704 to be performed for those Resources (3.1.28)

705 Note 1 to entry: This functionality can be used by sleeping Servers (3.1.38) or Servers (3.1.38) that choose not to
706 listen/respond to multicast requests directly.

707 **3.1.30**
708 **Resource Interface**
709 a qualification of the permitted requests on a Resource (3.1.28)

710 **3.1.31**
711 **Property**
712 a significant aspect or Parameter (3.1.24) of a Resource (3.1.28), including metadata, that is
713 exposed through the Resource (3.1.28)

714 **3.1.32**
715 **Resource Type**
716 a uniquely named definition of a class of Properties (3.1.31) and the interactions that are supported
717 by that class

718 Note 1 to entry: Each Resource (3.1.28) has a Property (3.1.31) "rt" whose value is the unique name of the Resource
719 Type (3.1.32).

720 **3.1.33**
721 **Scene**
722 a static entity that stores a set of defined Property (3.1.31) values for a Collection (3.1.4) of
723 Resources (3.1.28)

724 Note 1 to entry: A Scene (3.1.33) is a prescribed setting of a set of Resources (3.1.28) with each having a predetermined
725 value for the Property (3.1.31) that has to change.

726 **3.1.34**
727 **Scene Collection**
728 a Collection (3.1.4) that contains an enumeration of possible Scene Values (3.1.36) and the current
729 Scene Value (3.1.36)

730 Note 1 to entry: The member values of the Scene Collection (3.1.34) are Scene Members (3.1.35).

731 **3.1.35**
732 **Scene Member**
733 a Resource (3.1.28) that contains mappings of Scene Values (3.1.36) to values of a Property
734 (3.1.31) in the Resource (3.1.28)

735 **3.1.36**
736 **Scene Value**
737 a Scene (3.1.33) enumerator representing the state in which a Resource (3.1.28) can be

738 **3.1.37**
739 **Secure OCF Endpoint**
740 an OCF Endpoint (3.1.13) with a secure connection (e.g., CoAPS)

741 **3.1.38**
742 **Server**
743 a Device (3.1.10) with the role of providing Resource (3.1.28) state information and facilitating
744 remote interaction with its Resources (3.1.28)

745 Note 1 to entry: A Server (3.1.38) can be implemented to expose Non-OCF Device (3.1.20) resources to Clients (3.1.3)
746 (see 5.6).

747 **3.1.39**
748 **Unsecure OCF Endpoint**
749 an OCF Endpoint () with an unsecure connection (e.g., CoAP)

750 **3.1.40**
751 **Vertical Resource Type**
752 a Resource Type (3.1.32) in a vertical domain specification

753 Note 1 to entry: An example of a Vertical Resource Type (3.1.40) would be "oic.r.switch.binary".

754 **3.2 Abbreviated terms**

755 **3.2.1**
756 **ACL**
757 Access Control List

758 Note 1 to entry: The details are defined in ISO/IEC 30118-2:2018.

759 **3.2.2**
760 **BLE**
761 Bluetooth Low Energy

762 **3.2.3**
763 **CBOR**
764 Concise Binary Object Representation

765 **3.2.4**
766 **CoAP**
767 Constrained Application Protocol

768 **3.2.5**
769 **CoAPS**
770 Secure Constrained Application Protocol

771 **3.2.6**
772 **DTLS**
773 Datagram Transport Layer Security

774 Note 1 to entry: The details are defined in IETF RFC 6347.

775 **3.2.7**
776 **EXI**
777 Efficient XML Interchange

778 **3.2.8**
779 **IP**
780 Internet Protocol

781 **3.2.9**
782 **IRI**
783 Internationalized Resource Identifiers

784 **3.2.10**
785 **ISP**
786 Internet Service Provider

787 **3.2.11**
788 **JSON**
789 JavaScript Object Notation

790 **3.2.12**
791 **mDNS**
792 Multicast Domain Name Service

793 **3.2.13**
794 **MTU**
795 Maximum Transmission Unit

796 **3.2.14**
797 **NAT**
798 Network Address Translation

799 **3.2.15**
800 **OCF**
801 Open Connectivity Foundation

802 the organization that created this document

803 **3.2.16**
804 **REST**
805 Representational State Transfer

806 **3.2.17**
807 **RESTful**
808 REST-compliant Web services

809 **3.2.18**
810 **UDP**
811 User Datagram Protocol

812 Note 1 to entry: The details are defined in IETF RFC 768.

813 **3.2.19**
814 **URI**
815 Uniform Resource Identifier

816 **3.2.20**
817 **URN**
818 Uniform Resource Name

819 **3.2.21**
820 **UTC**
821 Coordinated Universal Time

822 **3.2.22**
823 **UUID**
824 Universal Unique Identifier

825 **3.2.23**
826 **XML**
827 Extensible Markup Language

828 **4 Document conventions and organization**

829 **4.1 Conventions**

830 In this document a number of terms, conditions, mechanisms, sequences, parameters, events,
831 states, or similar terms are printed with the first letter of each word in uppercase and the rest
832 lowercase (e.g., Network Architecture). Any lowercase uses of these words have the normal
833 technical English meaning.

834 **4.2 Notation**

835 In this document, features are described as required, recommended, allowed or DEPRECATED as
836 follows:

837 Required (or shall or mandatory)(M).

- 838 – These basic features shall be implemented to comply with Core Architecture. The phrases "shall
839 not", and "PROHIBITED" indicate behaviour that is prohibited, i.e. that if performed means the
840 implementation is not in compliance.

841 Recommended (or should)(S).

- 842 – These features add functionality supported by Core Architecture and should be implemented.
843 Recommended features take advantage of the capabilities Core Architecture, usually without
844 imposing major increase of complexity. Notice that for compliance testing, if a recommended
845 feature is implemented, it shall meet the specified requirements to be in compliance with these
846 guidelines. Some recommended features could become requirements in the future. The phrase
847 "should not" indicates behaviour that is permitted but not recommended.

848 Allowed (may or allowed)(O).

- 849 – These features are neither required nor recommended by Core Architecture, but if the feature
850 is implemented, it shall meet the specified requirements to be in compliance with these
851 guidelines.

852 DEPRECATED.

- 853 – Although these features are still described in this document, they should not be implemented
854 except for backward compatibility. The occurrence of a deprecated feature during operation of
855 an implementation compliant with the current document has no effect on the implementation's
856 operation and does not produce any error conditions. Backward compatibility may require that
857 a feature is implemented and functions as specified but it shall never be used by
858 implementations compliant with this document.

859 Conditionally allowed (CA).

- 860 – The definition or behaviour depends on a condition. If the specified condition is met, then the
861 definition or behaviour is allowed, otherwise it is not allowed.

862 Conditionally required (CR).

- 863 – The definition or behaviour depends on a condition. If the specified condition is met, then the
864 definition or behaviour is required. Otherwise the definition or behaviour is allowed as default
865 unless specifically defined as not allowed.

866 Strings that are to be taken literally are enclosed in "double quotes".

867 Words that are emphasized are printed in italic.

868 In all of the Property and Resource definition tables that are included throughout this document the
869 "Mandatory" column indicates that the item detailed is mandatory to implement; the mandating of
870 inclusion of the item in a Resource Payload associated with a CRUDN action is dependent on the
871 applicable schema for that action.

872 **4.3 Data types**

873 Resources are defined using data types derived from JSON values as defined in IETF RFC 7159.
874 However, a Resource can overload a JSON defined value to specify a particular subset of the
875 JSON value, using validation keywords defined in JSON Schema Validation.

876 Among other validation keywords, clause 7 in JSON Schema Validation defines a "format" keyword
 877 with a number of format attributes such as "uri" and "date-time", and a "pattern" keyword with a
 878 regular expression that can be used to validate a string. This clause defines patterns that are
 879 available for use in describing OCF Resources. The pattern names can be used in documenttext
 880 where JSON format names can occur. The actual JSON schemas shall use the JSON type and
 881 pattern instead.

882 For all rows defined in Table 1, the JSON type is string.

883 **Table 1 – Additional OCF Types**

Pattern Name	Pattern	Description
"csv"	<none>	A comma separated list of values encoded within a string. The value type in the csv is described by the Property where the csv is used. For example a csv of integers. NOTE csv is considered deprecated and an array of strings should be used instead for new Resources.
"date"	^([0-9]{4})-(1[0-2] 0[1-9])-(3[0-1] 2[0-9] 1[0-9] 0[1-9])\$	The full-date format pattern according to IETF RFC 3339
"duration"	^(P(?:\$)([0-9]+Y)?([0-9]+M)?([0-9]+W)?([0-9]+D)?(T(?:=[0-9]+[HMS])([0-9]+H)?([0-9]+M)?([0-9]+S)?)?))\$ ^([0-9]+W)\$ ^([0-9]{4})-(1[0-2] 0[1-9])-(3[0-1] 2[0-9] 1[0-9] 0[1-9])T(2[0-3] 1[0-9] 0[1-9]):([0-5][0-9]):([0-5][0-9])\$ ^([0-9]{4})(1[0-2] 0[1-9])(3[0-1] 2[0-9] 1[0-9] 0[1-9])T(2[0-3] 1[0-9] 0[1-9])([0-5][0-9])([0-5][0-9])\$	A string representing duration formatted as defined in ISO 8601. Allowable formats are: P[n]Y[n]M[n]DT[n]H[n]M[n]S, P[n]W, P[n]Y[n]-M[n]-DT[0-23]H[0-59]:M[0-59]:S, and P[n]W, P[n]Y[n]M[n]DT[0-23]H[0-59]M[0-59]S. P is mandatory, all other elements are optional, time elements must follow a T.
"int64"	^0 (-?[1-9][0-9]{0,18})\$	A string instance is valid against this attribute if it contains an integer in the range $[-(2^{63}), (2^{63}-1)]$ NOTE IETF RFC 7159 clause 6 explains that JSON integers outside the range $[-(2^{53}+1), (2^{53}-1)]$ are not interoperable and so JSON numbers cannot be used for 64-bit numbers.
"language-tag"	^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*\$	An IETF language tag formatted according to IETF RFC 5646 clause 2.1.
"uint64"	^0 ([1-9][0-9]{0,19})\$	A string instance is valid against this attribute if it contains an integer in the range $[0, (2^{64}-1)]$ Also see note for "int64"
"uuid"	^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}\$	A UUID string representation formatted according to IETF RFC 4122 clause 3.

884

885 Strings shall be encoded as UTF-8 unless otherwise specified.

886 In a JSON schema, "maxLength" for a string indicates the maximum number of characters not
887 octets. However, "maxLength" shall also indicate the maximum number of octets. If no "maxLength"
888 is defined for a string, then the maximum length shall be 64 octets.

889 **5 Architecture**

890 **5.1 Overview**

891 The architecture enables resource based interactions among IoT artefacts, i.e. physical devices or
892 applications. The architecture leverages existing industry standards and technologies and provides
893 solutions for establishing connections (either wireless or wired) and managing the flow of
894 information among Devices, regardless of their form factors, operating systems or service providers.

895 Specifically, the architecture provides:

- 896 – A communication and interoperability framework for multiple market segments (Consumer,
897 Enterprise, Industrial, Automotive, Health, etc.), OSs, platforms, modes of communication,
898 transports and use cases.
- 899 – A common and consistent model for describing the environment and enabling information and
900 semantic interoperability.
- 901 – Common communication protocols for discovery and connectivity.
- 902 – Common security and identification mechanisms.
- 903 – Opportunity for innovation and product differentiation.
- 904 – A scalable solution addressing different Device capabilities, applicable to smart devices as well
905 as the smallest connected things and wearable devices.

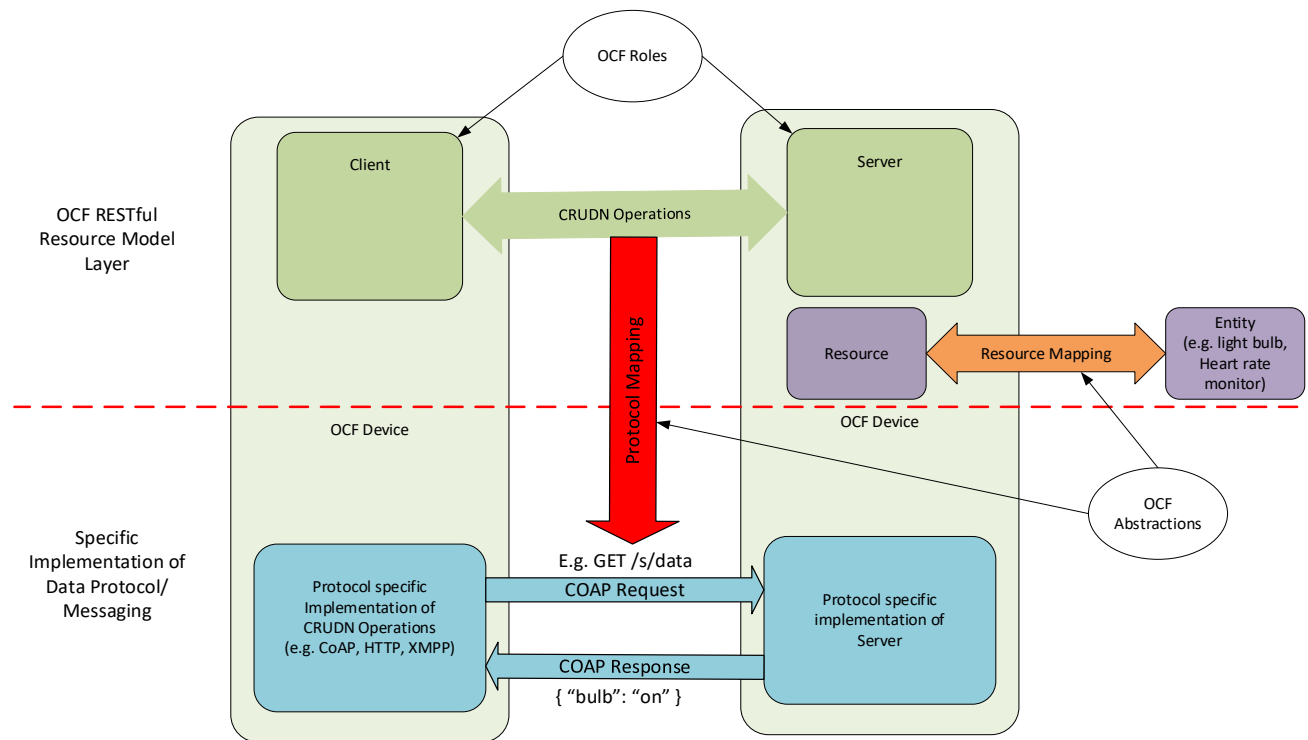
906 The architecture is based on the Resource Oriented Architecture design principles and described
907 in the 5.2 through 5.6 respectively. 5.2 presents the guiding principles for OCF operations. 5.3
908 defines the functional block diagram and Framework. 5.5 provides an example scenario with roles.
909 5.6 provides an example scenario of bridging to non- OCF ecosystem.

910 **5.2 Principle**

911 In the architecture, Entities in the physical world (e.g., temperature sensor, an electric light or a
912 home appliance) are represented as Resources. Interactions with an entity are achieved through
913 its Resource representations (see 7.7) using operations that adhere to Representational State
914 Transfer (REST) architectural style, i.e., RESTful interactions.

915 The architecture defines the overall structure of the Framework as an information system and the
916 interrelationships of the Entities that make up OCF. Entities are exposed as Resources, with their
917 unique identifiers (URIs) and support interfaces that enable RESTful operations on the Resources.
918 Every RESTful operation has an initiator of the operation (the Client) and a responder to the
919 operation (the Server). In the Framework, the notion of the Client and Server is realized through
920 roles (see 5.5). Any Device can act as a Client and initiate a RESTful operation on any Device
921 acting as a Server. Likewise, any Device that exposes Entities as Resources acts as a Server.
922 Conformant to the REST architectural style, each RESTful operation contains all the information
923 necessary to understand the context of the interaction and is driven using a small set of generic
924 operations, i.e., CREATE, RETRIEVE, UPDATE, DELETE and NOTIFY (CRUDN) defined in clause
925 8, which include representations of Resources.

926 Figure 1 depicts the architecture.



927
928

929

Figure 1 – Architecture - concepts

930 The architecture is organized conceptually into three major aspects that provide overall separation
931 of concern: Resource model, RESTful operations and abstractions.

932 – Resource model: The Resource model provides the abstractions and concepts required to
933 logically model, and logically operate on the application and its environment. The Core
934 Resource model is common and agnostic to any specific application domain such as smart
935 home, industrial or automotive. For example, the Resource model defines a Resource which
936 abstracts an entity and the representation of a Resource maps the entity’s state. Other
937 Resource model concepts can be used to model other aspects, for example behaviour.

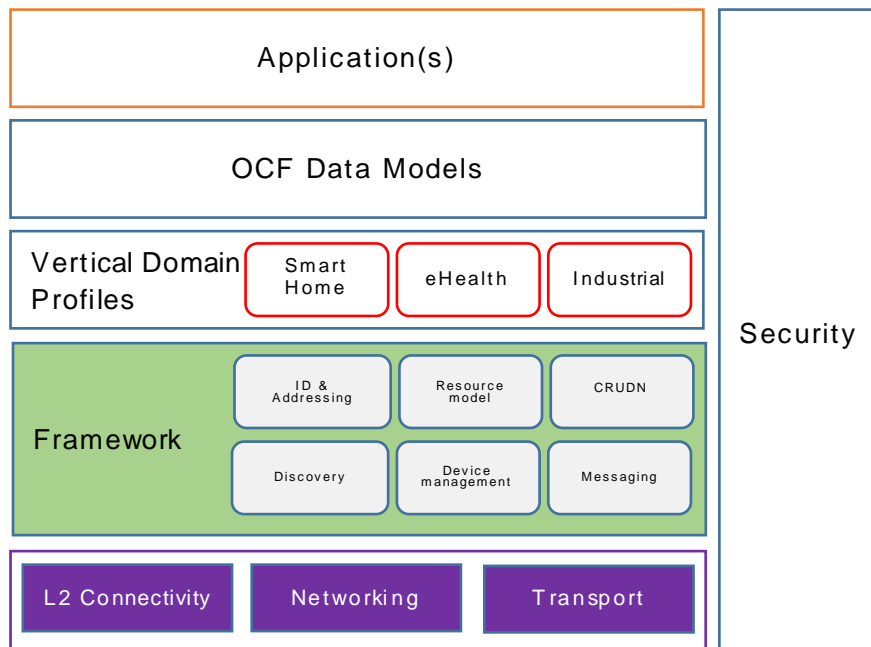
938 – RESTful operations: The generic CRUDN operations are defined using the RESTful paradigm
939 to model the interactions with a Resource in a protocol and technology agnostic way. The
940 specific communication or messaging protocols are part of the protocol abstraction and
941 mapping of Resources to specific protocols is provided in 11.8.

942 – Abstraction: The abstractions in the Resource model and the RESTful operations are mapped
943 to concrete elements using abstraction primitives. An entity handler is used to map an entity to
944 a Resource and connectivity abstraction primitives are used to map logical RESTful operations
945 to data connectivity protocols or technologies. entity handlers may also be used to map
946 Resources to Entities that are reached over protocols that are not natively supported by OCF.

947 **5.3 Functional block diagram**

948 The functional block diagram encompasses all the functionalities required for operation. These
949 functionalities are categorized as L2 connectivity, networking, transport, Framework, and
950 application profiles. The functional blocks are depicted in Figure 2.

951

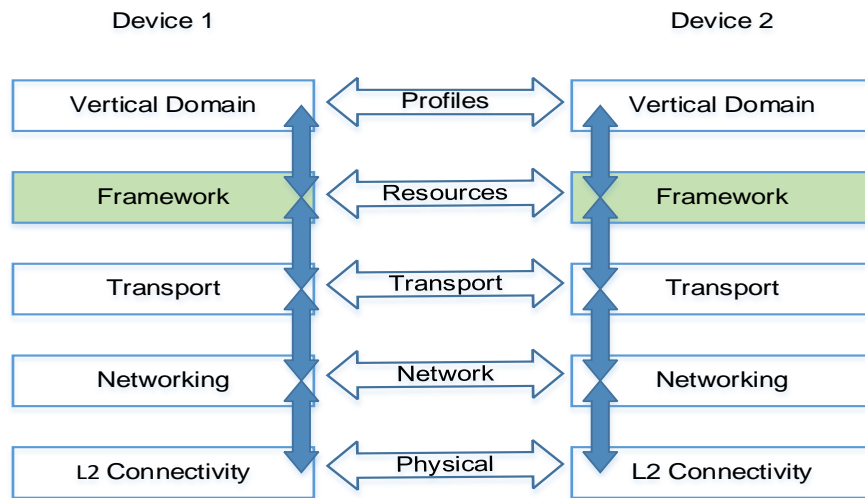


952

Figure 2 – Functional block diagram

- 953 – *L2 connectivity*: Provides the functionalities required for establishing physical and data link
954 layer connections (e.g., Wi-Fi™ or Bluetooth® connection) to the network.
- 955 – *Networking*: Provides functionalities required for Devices to exchange data among themselves
956 over the network (e.g., Internet).
- 957 – *Transport*: Provides end-to-end flow transport with specific QoS constraints. Examples of a
958 transport protocol include TCP and UDP or new Transport protocols under development in the
959 IETF, e.g., Delay Tolerant Networking (DTN).
- 960 – *Framework*: Provides the core functionalities as defined in this document. The functional block
961 is the source of requests and responses that are the content of the communication between
962 two Devices.
- 963 – *Vertical Domain profile*: Provides market segment specific functionalities, e.g., functions for the
964 smart home market segment.

965 When two Devices communicate with each other, each functional block in a Device interacts with
966 its counterpart in the peer Device as shown in Figure 3.



967
968 **Figure 3 – Communication layering model**

969 **5.4 Framework**

970 Framework consists of functions which provide core functionalities for operation.

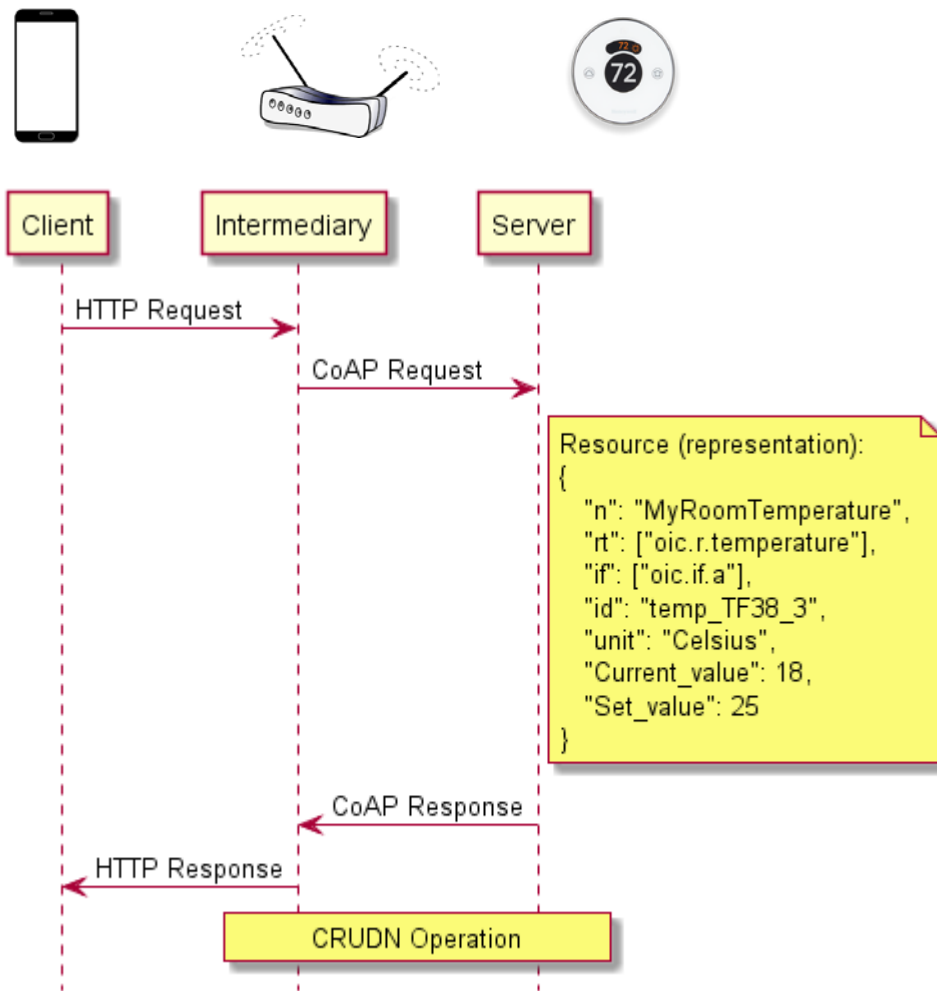
- 971 – *Identification and addressing.* Defines the identifier and addressing capability. The Identification
972 and addressing function is defined in clause 6.
- 973 – *Discovery.* Defines the process for discovering available.
- 974 – Devices (OCF Endpoint Discovery in clause 10) and
- 975 – Resources (Resource discovery in 11.3).
- 976 – *Resource model.* Specifies the capability for representation of entities in terms of Resources
977 and defines mechanisms for manipulating the Resources. The Resource model function is
978 defined in clause 7.
- 979 – *CRUDN.* Provides a generic scheme for the interactions between a Client and Server as defined
980 in clause 8.
- 981 – *Messaging.* Provides specific message protocols for RESTful operation, i.e. CRUDN. For
982 example, CoAP is a primary messaging protocol. The messaging function is defined in 12.
- 983 – *Device management.* Specifies the discipline of managing the capabilities of a Device, and
984 includes Device provisioning and initial setup as well as Device monitoring and diagnostics.
985 The Device management function is defined in 11.5.
- 986 – *Security.* Includes authentication, authorization, and access control mechanisms required for
987 secure access to Entities. The security function is defined in clause 13.

988 **5.5 Example Scenario with roles**

989 Interactions are defined between logical entities known as roles. Three roles are defined: Client,
990 Server and Intermediary.

991 Figure 4 illustrates an example of the roles in a scenario where a smart phone sends a request
992 message to a thermostat; the original request is sent over HTTP, but is translated into a CoAP
993 request message by a gateway in between, and then delivered to the thermostat. In this example,
994 the smart phone takes the role of a Client, the gateway takes the role of an Intermediary and the
995 thermostat takes the role of a Server.

996



997

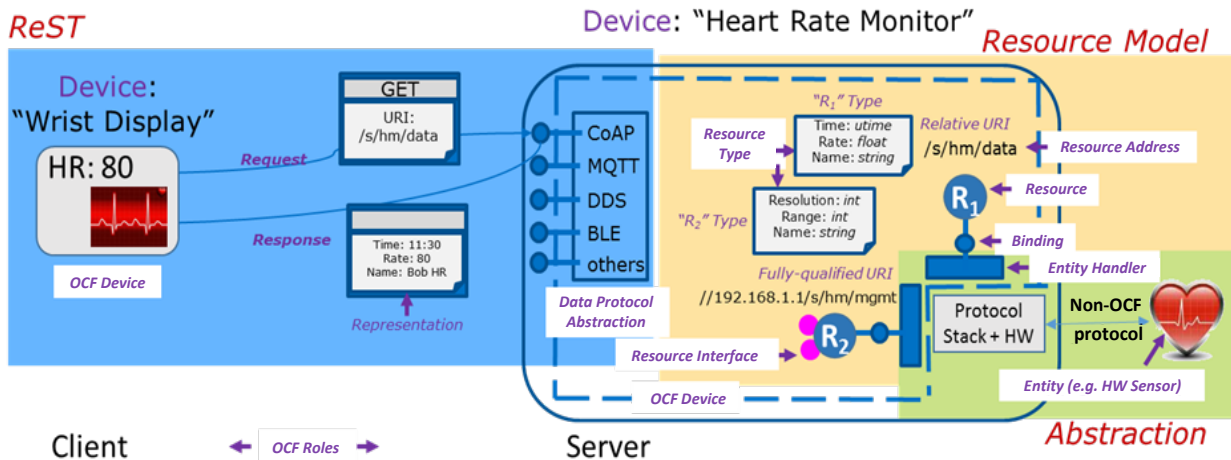
998

Figure 4 – Example illustrating the roles

999 **5.6 Example Scenario: Bridging to Non- OCF ecosystem**

1000 The use case for this scenario is a display (like a wrist watch) that is used to monitor a heart rate
1001 sensor that implements a protocol that is not OCF supported.

1002 Figure 5 provides a detailed logical view of the concepts described in Figure 1.



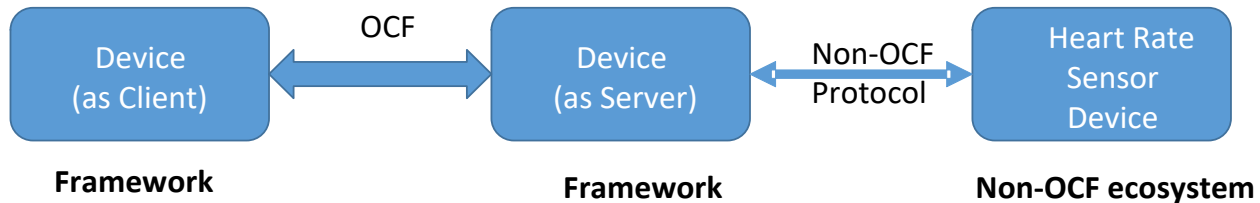
1003

1004

Figure 5 – Framework - Architecture Detail

1005
1006

The details may be implemented in many ways, for example, by using a Server with an entity handler to interface directly to a Non- OCF device as shown in Figure 6.



1007
1008

1009

Figure 6 – Server bridging to Non- OCF device

1010
1011
1012
1013

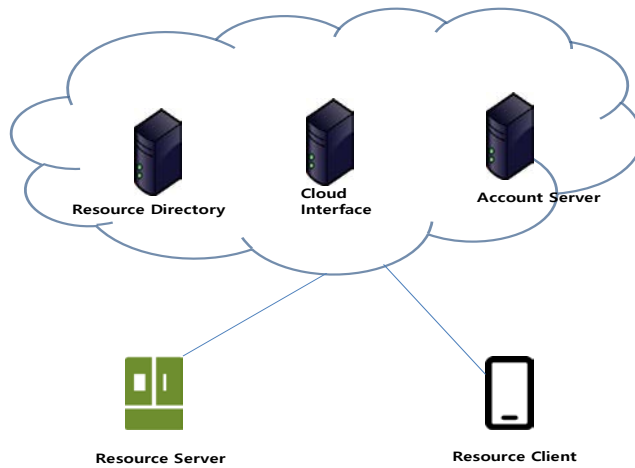
On start-up the Server runs the entity handlers which discover the non- OCF systems (e.g., Heart Rate Sensor Device) and create Resources for each Device or functionality discovered. The entity handler creates a Resource for each discovered Device or functionality and binds itself to that Resource. These Resources are made discoverable by the Server.

1014
1015
1016
1017
1018
1019

Once the Resources are created and made discoverable, then the Display Device can discover these Resources and operate on them using the mechanisms described in this document. The requests to a Resource on the Server are then interpreted by the entity handler and forwarded to the Non- OCF device using the protocol supported by the Non-OCF device. The returned information from the Non- OCF device is then mapped to the appropriate response for that Resource.

1020 5.7 OCF Cloud architecture

1021 This clause describes the architecture of OCF Cloud in Figure 7: and Figure 8



1022

1023

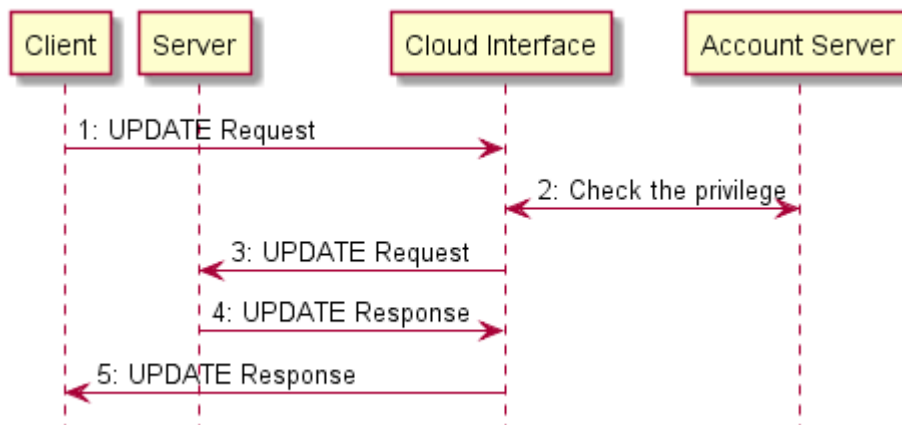
Figure 7 – OCF Cloud deployment architecture

1024 The Cloud architecture comprises of following three network entities:

- 1025 – *Cloud Interface Server* – A logical entity to which an OCF Device primarily. It encapsulates
1026 Account Server and Resource Directory features. The Cloud Interface routes the packet
1027 between OCF Devices based on the request URI in the packet header. The Client needs to
1028 keep the persistent connection alive to the Server.
- 1029 – *Account Server* – A logical entity that handles Device registration, Auth Token validation and
1030 handles sign-in and token-refresh requests from the Device.
- 1031 – *Resource Directory* – A logical entity holding Resource information published by Servers. A
1032 Client when looking for a Resource receives a response from the Resource Directory on behalf
1033 of the Server. Then with information included in the response form the Resource Directory, the
1034 Client directly connects to the Server.

1035 When a Client try to access a Server, the Client connects to Cloud Interface Server then Cloud
1036 Interface routes the received message to the indicated Server after checking the privilege.

1037



1038

1039

Figure 8 – OCF Endpoint routing

1040 **6 Identification and addressing**

1041 **6.1 Introduction**

1042 Facilitating proper and efficient interactions between elements in the Framework, requires a means
1043 to identify, name and address these elements.

1044 The *identifier* unambiguously identifies an element in a context or domain. The context or domain
1045 may be determined by the use or the application. The identifier is expected to be immutable over
1046 the lifecycle of that element and is unambiguous within a context or domain.

1047 The *address* is used to define a place, way or means of reaching or accessing the element in order
1048 to interact with it. An address may be mutable based on the context.

1049 The *name* is a handle that distinguishes the element from other elements in the Framework. The
1050 name may be changed over the lifecycle of that element.

1051 There may be methods or resolution schemes that allow determining any of these based on the
1052 knowledge of one or more of others (e.g., determine name from address or address from name).

1053 Each of these aspects may be defined separately for multiple contexts (e.g., a context could be a
1054 layer in a stack). So an address may be a URL for addressing Resource and an IP address for
1055 addressing at the connectivity layer. In some situations, both these addresses would be required.
1056 For example, to do RETRIEVE (see 8.3) operation on a particular Resource representation, the
1057 Client needs to know the address of the target Resource and the address of the Server through
1058 which the Resource is exposed.

1059 In a context or domain of use, a name or address could be used as identifier or vice versa. For
1060 example, a URL could be used as an identifier for a Resource and designated as a URI.

1061 The remainder of this clause discusses the identifier, address and naming from the point of view
1062 of the Resource model and the interactions to be supported by the Resource model. Examples of
1063 interactions are the RESTful interactions, i.e. CRUDN operation (clause 8) on a Resource. Also
1064 the mapping of these to transport protocols, e.g., CoAP is described.

1065 **6.2 Identification**

1066 **6.2.1 Overview**

1067 An identifier is unambiguous within the context or domain of use. There are many schemes that
1068 may be used to generate an identifier that has the required properties. The identifier may be
1069 context-specific in that the identifier is expected to be and guaranteed to be unambiguous only
1070 within that context or domain. Identifier may also be context-independent where these identifiers
1071 are guaranteed to be unambiguous across all contexts and domains both spatially and temporally.
1072 The context-specific identifiers could be defined by simple schemes like monotonic enumeration or
1073 may be defined by overloading an address or name, for example an IP address may be an identifier
1074 within the private domain behind a gateway in a smart home. On the other hand, context-
1075 independent identifiers require a stronger scheme that derives universally unique identities, for
1076 example any one of the versions of Universally Unique Identifiers (UUIDs). Context independent
1077 identifier may also be generated using hierarchy of domains where the root of the hierarchy is
1078 identified with a UUID and sub-domains may generate context independent identifier by
1079 concatenating context-specific identifiers for that domain to the context-independent identifier of
1080 their parent.

1081 **6.2.2 Resource identification and addressing**

1082 A Resource may be identified using a URI and addressed by the same URI if the URI is a URL. In
1083 some cases a Resource may need an identifier that is different from a URI; in this case, the

1084 Resource may have a Property whose value is the identifier. When the URI is in the form of a URL,
1085 then the URI may be used to address the Resource.

1086 An OCF URI is based on the general form of a URI as defined in IETF RFC 3986 as follows:

1087 `<scheme>://<authority>/<path>?<query>`

1088 Specifically the OCF URI is specified in the following form:

1089 `ocf://<authority>/<path>?<query>`

1090 The following is a description of values that each component takes.

1091 The *scheme* for the URI is "ocf". The "ocf" scheme represents the semantics, definitions and use
1092 as defined in this document. If a URI has the portion preceding the "://" (double slash) omitted, then
1093 the "ocf" scheme shall be assumed.

1094 Each transport binding is responsible for specifying how an OCF URI is converted to a transport
1095 protocol URI before sending over the network by the requestor. Similarly on the receiver side, each
1096 transport binding is responsible for specifying how an OCF URI is converted from a transport
1097 protocol URI before handing over to the Resource model layer on the receiver.

1098 The authority of an OCF URI shall be the Device ID ("di") value, as defined in [OCF Security], of
1099 the Server.

1100 The *path* is a string that unambiguously identifies or references a Resource within the context of
1101 the Server. In this version of the document, a path shall not include pct-encoded non-ASCII
1102 characters or NUL characters. A *path* shall be preceded by a "/" (slash). The *path* may have "/"
1103 (slash) separated segments for human readability reasons. In the OCF context, the "/" (slash)
1104 separated segments are treated as a single string that directly references the Resources (i.e. a flat
1105 structure) and not parsed as a hierarchy. On the Server, the path or some substring in the path
1106 may be shortened by using hashing or some other scheme provided the resulting reference is
1107 unique within the context of the host.

1108 Once a path is generated, a Client accessing the Resource or recipient of the URI should use that
1109 path as an opaque string and should not parse to infer a structure, organization or semantic.

1110 A query string shall contain a list of "<name>=<value>" segments (aka name-value pair) each
1111 separated by a "&" (ampersand). The query string will be mapped to the appropriate syntax of the
1112 protocol used for messaging. (e.g., CoAP).

1113 A URI may be either fully qualified or relative generation of URI.

1114 A URI may be defined by the Client which is the creator of that Resource. Such a URI may be
1115 relative or absolute (fully qualified). A relative URI shall be relative to the Device on which it is
1116 hosted. Alternatively, a URI may be generated by the Server of that Resource automatically based
1117 on a pre-defined convention or organization of the Resources, based on an OCF Interface, based
1118 on some rules or with respect to different roots or bases.

1119 The absolute path reference of a URI is to be treated as an opaque string and a Client should not
1120 infer any explicit or implied structure in the URI – the URI is simply an address. It is also
1121 recommended that Devices hosting a Resource treat the URI of each Resource as an opaque string
1122 that addresses only that Resource. (e.g., URI's "/a" and "/a/b" are considered as distinct addresses
1123 and Resource b cannot be construed as a child of Resource a).

1124 **6.3 Namespace:**

1125 The relative URI prefix "/oic/" is reserved as a namespace for URIs defined in OCF specifications
1126 and shall not be used for URIs that are not defined in OCF specifications.

1127 **6.4 Network addressing**

1128 The following are the addresses used in this document:

1129 IP address

- 1130 – An IP address is used when the Device is using an IP configured interface.
- 1131 – When a Device only has the identity information of its peer, a resolution mechanism is needed
- 1132 to map the identifier to the corresponding address.

1133 **7 Resource model**

1134 **7.1 Introduction**

1135 The Resource model defines concepts and mechanisms that provide consistency and core
1136 interoperability between Devices in the OCF ecosystems. The Resource model concepts and
1137 mechanisms are then mapped to the transport protocols to enable communication between the
1138 Devices – each transport provides the communication protocol interoperability. The Resource
1139 model, therefore, allows for interoperability to be defined independent of the transports.

1140 In addition, the concepts in the Resource model support modelling of the primary artefacts and
1141 their relationships to one and another and capture the semantic information required for
1142 interoperability in a context. In this way, OCF goes beyond simple protocol interoperability to
1143 capture the rich semantics required for true interoperability in Wearable and Internet of Things
1144 ecosystems.

1145 The primary concepts in the Resource model are: entity, Resources, Uniform Resource Identifiers
1146 (URI), Resource Types, Properties, Representations, OCF Interfaces, Collections and Links. In
1147 addition, the general mechanisms are CREATE, RETRIEVE, UPDATE, DELETE and NOTIFY.
1148 These concepts and mechanisms may be composed in various ways to define the rich semantics
1149 and interoperability needed for a diverse set of use cases that the Framework is applied to.

1150 In the OCF Resource model Framework, an entity needs to be visible, interacted with or
1151 manipulated, it is represented by an abstraction called a Resource. A Resource encapsulates and
1152 represents the state of an entity. A Resource is identified, addressed and named using URIs.

1153 Properties are "key=value" pairs and represent state of the Resource. A snapshot of these
1154 Properties is the Representation of the Resource. A specific view of the Representation and the
1155 mechanisms applicable in that view are specified as OCF Interfaces. Interactions with a Resource
1156 are done as Requests and Responses containing Representations.

1157 A Resource instance is derived from a Resource Type. The uni-directional relationship between
1158 one Resource and another Resource is defined as a Link. A Resource that has Properties and
1159 Links is a Collection.

1160 A set of Properties can be used to define a state of a Resource. This state may be retrieved or
1161 updated using appropriate Representations respectively in the response from and request to that
1162 Resource.

1163 A Resource (and Resource Type) could represent and be used to expose a capability. Interactions
1164 with that Resource can be used to exercise or use that capability. Such capabilities can be used to
1165 define processes like discovery, management, advertisement etc. For example: *discovery of*
1166 *Resources on a Device* can be defined as the retrieval of a representation of a specific Resource
1167 where a Property or Properties have values that describe or reference the Resources on the Device.

1168 The information for Request or Response with the Representation may be communicated on the
1169 wire by serializing using a transfer protocol or encapsulated in the payload of the transport protocol

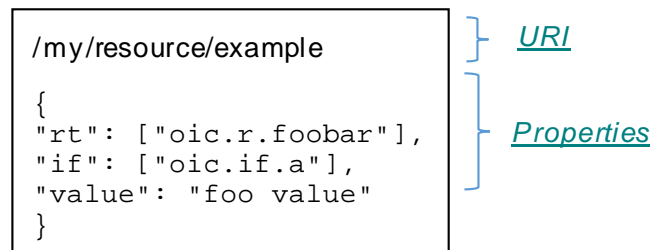
1170 – the specific method is determined by the normative mapping of the Request or Response to the
1171 transport protocol. See 11.8 for transport protocols supported.

1172 The OpenAPI 2.0 definitions (Annex D) used in this document are normative. This includes that all
1173 defined JSON payloads shall comply with the indicated OpenAPI 2.0 definitions. Annex D contains
1174 all of the OpenAPI 2.0 definitions for Resource Types defined in this document.

1175 7.2 Resource

1176 A Resource shall be defined by one or more Resource Type(s) – see Annex D for Resource Type.
1177 A request to CREATE a Resource shall specify one or more Resource Types that define that
1178 Resource.

1179 A Resource is hosted in a Device. A Resource shall have a URI as defined in clause 6. The URI
1180 may be assigned by the Authority at the creation of the Resource or may be pre-defined by the
1181 specification of the Resource Type. An example Resource representation is depicted in Figure 9.



1182

1183

Figure 9 – Example Resource

1184 Core Resources are the Resources defined in this document to enable functional interactions as
1185 defined in clause 10 (e.g., Discovery, Device management, etc). Among the Core Resources,
1186 "/oic/res", "/oic/p", and "/oic/d" shall be supported on all Devices. Devices may support other Core
1187 Resources depending on the functional interactions they support.

1188 7.3 Property

1189 7.3.1 Introduction

1190 A Property describes an aspect that is exposed through a Resource including meta-information
1191 related to that Resource.

1192 A Property shall have a name i.e. Property Name and a value i.e. Property Value. The Property is
1193 expressed as a key-value pair where key is the Property Name and value the Property Value like
1194 <Property Name> = <Property Value>. For example if the "temperature" Property has a Property
1195 Name "temp" and a Property Value "30F", then the Property is expressed as "temp=30F". The
1196 specific format of the Property depends on the encoding scheme. For example, in JSON, Property
1197 is represented as "key": value (e.g., "temp": 30).

1198 In addition, the Property definition shall have a

1199 – *Value Type* – the Value Type defines the values that a Property Value may take. The Value
1200 Type may be a simple data type (e.g. string, Boolean) as defined in 4.3 or may be a complex
1201 data type defined with a schema. The Value Type may define

1202 – Value Rules define the rules for the set of values that the Property Value may take. Such
1203 rules may define the range of values, the min-max, formulas, the set of enumerated values,
1204 patterns, conditional values, and even dependencies on values of other Properties. The
1205 rules may be used to validate the specific values in a Property Value and flag errors.

1206 – *Mandatory* – specifies if the Property is mandatory or not for a given Resource Type.

1207 – *Access modes* – specifies whether the Property may be read, written or both. Updates are
1208 equivalent to a write. "r" is used for read and "w" is used for write – both may be specified.
1209 Write does not automatically imply read.

1210 The definition of a Property may include the following additional information – these items are
1211 informative:

1212 – *Property Title* - a human-friendly name to designate the Property; usually not sent over the wire.

1213 – *Description* – descriptive text defining the purpose and expected use of this Property.

1214 In general, a Property is meaningful only within the Resource to which it is associated. However a
1215 base set of Properties that may be supported by all Resources, known as Common Properties,
1216 keep their semantics intact across Resources i.e. their "key=value" pair means the same in any
1217 Resource. Detailed tables for all Common Properties are defined in 7.3.2.

1218 **7.3.2 Common Properties**

1219 **7.3.2.1 Introduction**

1220 The Common Properties defined in this clause may be specified for all Resources. The following
1221 Properties are defined as Common Properties: Resource Type, Resource Interface, Name, and
1222 Resource Identity.

1223 The name of a Common Property shall be unique and shall not be used by other Properties. When
1224 defining a new Resource Type, its non-common Properties shall not use the name of existing
1225 Common Properties (e.g., "rt", "if", "n", "id"). When defining a new "Common Property", it should
1226 be ensured that its name has not been used by any other Properties. The uniqueness of a new
1227 Common Property name can be verified by checking all the Properties of all the existing OCF
1228 defined Resource Types. However, this may become cumbersome as the number of Resource
1229 Types grow. To prevent such name conflicts in the future, OCF may reserve a certain name space
1230 for Common Property. Potential approaches are (1) a specific prefix (e.g. "oic") may be designated
1231 and the name preceded by the prefix (e.g. "oic.psize") is only for Common Property; (2) the names
1232 consisting of one or two letters are reserved for Common Property and all other Properties shall
1233 have the name with the length larger than the 2 letters; (3) Common Properties may be nested
1234 under specific object to distinguish themselves.

1235 The ability to UPDATE a Common Property (that supports write as an access mode) is restricted
1236 to the "oic.if.rw" (read-write) OCF Interface; thus a Common Property shall be updatable using the
1237 read-write OCF Interface if and only if the Property supports write access as defined by the Property
1238 definition and the associated schema for the read-write OCF Interface.

1239 The following Common Properties for all Resources are specified in 7.3.2.2 through 7.3.2.6 and
1240 summarized as follows:

1241 – *Resource Type* ("rt") – this Property is used to declare the Resource Type of that Resource.
1242 Since a Resource could be define by more than one Resource Type the Property Value of the
1243 Resource Type Property can be used to declare more than one Resource type. For example:
1244 "rt": ["oic.wk.d", "oic.d.airconditioner"] declares that the Resource containing this Property is
1245 defined by either the "oic.wk.d" Resource Type or the "oic.d.airconditioner" Resource Type.
1246 See 7.3.2.3 for details.

1247 – *OCF Interface* ("if") – this Property declares the OCF Interfaces supported by the Resource.
1248 The Property Value of the OCF Interface Property can be multi-valued and lists all the OCF
1249 Interfaces supported. See 7.3.2.4 for details.

1250 – *Name* ("n") – the Property declares human-readable name assigned to the Resource. See
1251 7.3.2.5.

1252 – *Resource Identity* ("id"): its Property Value shall be a unique (across the scope of the host
 1253 Server) instance identifier for a specific instance of the Resource. The encoding of this identifier
 1254 is Device and implementation dependent. See 7.3.2.6 for details.

1255 **7.3.2.2 Property Name and Property Value definitions**

1256 The Property Name and Property Value as used in this document:

1257 – *Property Name*– the key in "key=value" pair. Property Name is case sensitive and its data type
 1258 is "string". Property names shall contain only letters A to Z, a to z, digits 0 to 9, hyphen, and
 1259 dot, and shall not begin with a digit.

1260 – *Property Value*– the value in "key=value" pair. Property Value is case sensitive when its data
 1261 type is "string".

1262 **7.3.2.3 Resource Type**

1263 Resource Type Property is specified in 7.4.

1264 **7.3.2.4 OCF Interface**

1265 OCF Interface Property is specified in 7.6.

1266 **7.3.2.5 Name**

1267 A human friendly name for the Resource, i.e. a specific resource instance name (e.g.,
 1268 MyLivingRoomLight), The Name Property is as defined in Table 2

1269 **Table 2 – Name Property Definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Name	"n"	"string"	N/A	N/A	R, W	No	Human understandable name for the Resource.

1270
 1271 The Name Property is read-write unless otherwise restricted by the Resource Type (i.e. the
 1272 Resource Type does not support UPDATE or does not support UPDATE using read-write).

1273 **7.3.2.6 Resource Identity**

1274 The Resource Identity Property shall be a unique (across the scope of the host Server) instance
 1275 identifier for a specific instance of the Resource. The encoding of this identifier is Device and
 1276 implementation dependent as long as the uniqueness constraint is met, noting that an
 1277 implementation may use a uuid as defined in 4.3. The Resource Identity Property is as defined in
 1278 Table 3.

1279 **Table 3 – Resource Identity Property Definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Resource Identity	"id"	"string" or uuid	Implementation Dependent	N/A	R	No	Unique identifier of the Resource (over all Resources in the Device)

1280

1281 **7.4 Resource Type**

1282 **7.4.1 Introduction**

1283 Resource Type is a class or category of Resources and a Resource is an instance of one or more
1284 Resource Types.

1285 The Resource Types of a Resource is declared using the Resource Type Common Property as
1286 described in 7.3.2.3 or in a Link using the Resource Type Parameter.

1287 A Resource Type may either be pre-defined by OCF or in custom definitions by manufacturers, end
1288 users, or developers of Devices (vendor-defined Resource Types). Resource Types and their
1289 definition details may be communicated out of band (i.e. in documentation) or be defined explicitly
1290 using a meta-language which may be downloaded and used by APIs or applications. OCF has
1291 adopted OpenAPI 2.0 as the specification method for OCF's RESTful interfaces and Resource
1292 definitions.

1293 Every Resource Type shall be identified with a Resource Type ID which shall be represented using
1294 the requirements and ABNF governing the Resource Type attribute in IETF RFC 6690 (clause 2 for
1295 ABNF and clause 3.1 for requirements) with the caveat that segments are separated by a "."
1296 (period). The entire string represents the Resource Type ID. When defining the ID each segment
1297 may represent any semantics that are appropriate to the Resource Type. For example, each
1298 segment could represent a namespace. Once the ID has been defined, the ID should be used
1299 opaquely and implementations should not infer any information from the individual segments. The
1300 string "oic", when used as the first segment in the definition of the Resource Type ID, is reserved
1301 for OCF-defined Resource Types. All OCF defined Resource Types are to be registered with the
1302 IANA Core Parameters registry as described also in IETF RFC 6690.

1303 **7.4.2 Resource Type Property**

1304 A Resource when instantiated or created shall have one or more Resource Types that are the
1305 template for that Resource. The Resource Types that the Resource conforms to shall be declared
1306 using the "rt" Common Property for the Resource as defined in Table 4. The Property Value for the
1307 "rt" Common Property shall be the list of Resource Type IDs for the Resource Types used as
1308 templates (i.e., "rt"=<list of Resource Type IDs>).

1309 **Table 4 – Resource Type Common Property definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Resource Type	"rt"	"array"	Array of strings, conveying Resource Type IDs	N/A	R	Yes	The Property name rt is as described in IETF RFC 6690

1310
1311 Resource Types may be explicitly discovered or implicitly shared between the user (i.e. Client) and
1312 the host (i.e. Server) of the Resource.

1313 **7.4.3 Resource Type definition**

1314 Resource Type is specified as follows:

- 1315 – *Pre-defined URI* (optional) – a pre-defined URI may be specified for a specific Resource Type
1316 in an OCF specification. When a Resource Type has a pre-defined URI, all instances of that
1317 Resource Type shall use only the pre-defined URI. An instance of a different Resource Type
1318 shall not use the pre-defined URI.
- 1319 – *Resource Type Title* (optional) – a human friendly name to designate the Resource Type.

- 1320 – *Resource Type ID* – the value of "rt" Property which identifies the Resource Type, (e.g.,
1321 "oic.wk.p").
- 1322 – *Resource Interfaces* – list of the OCF Interfaces that may be supported by the Resource Type.
- 1323 – *Properties* – definition of all the Properties that apply to the Resource Type. The Resource Type
1324 definition shall define whether a property is mandatory, conditional mandatory, or optional.
- 1325 – *Related Resource Types* (optional) – the specification of other Resource Types that may be
1326 referenced as part of the Resource Type, applicable to Collections.
- 1327 – *Mime Types* (optional) – mime types supported by the Resource including serializations (e.g.,
1328 application/cbor, application/json, application/xml).

1329 Table 5 and Table 6 provides an example description of an illustrative foobar Resource Type and
1330 its associated Properties.

1331 **Table 5 – Example foobar Resource Type**

Pre-defined URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction	M/CR/O
none	"foobar"	"oic.r.foobar"	"oic.if.a"	Example "foobar" Resource	Actuation	O

1332

1333 **Table 6 – Example foobar Properties**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Resource Type	"rt"	"array"	N/A	N/A	R	Yes	Resource Type
OCF Interface	"if"	"array"	N/A	N/A	R	Yes	OCF Interface
Foo value	value	"string"	N/A	N/A	R	Yes	Foo value

1334

1335 For example, an instance of the foobar Resource Type.

```
1336 {
1337   "rt": ["oic.r.foobar"],
1338   "if": ["oic.if.a"],
1339   "value": "foo value"
1340 }
```

1341

1342 For example, a schema representation for the foobar Resource Type.

```
1343 {
1344   "$schema": "http://json-schema.org/draft-04/schema",
1345   "type": "object",
1346   "properties": {
1347     "rt": {
1348       "type": "array",
1349       "items": {
1350         "type": "string",
1351         "maxLength": 64
1352       },

```

```

1353     "minItems" : 1,
1354     "readOnly": true,
1355     "description": "Resource Type of the Resource"
1356   },
1357   "if": {
1358     "type": "array",
1359     "items": {
1360       "type" : "string",
1361       "enum" : ["oic.if.baseline", "oic.if.ll", "oic.if.b", "oic.if.lb", "oic.if.rw",
1362 "oic.if.r", "oic.if.a", "oic.if.s"]
1363     },
1364     "value": {"type": "string"}
1365   },
1366   "required": ["rt", "if", "value"]
1367 }

```

1368 **7.4.4 Multi-value "rt" Resource**

1369 Multi-value "rt" Resource means a Resource with multiple Resource Types where none of the
1370 included Resource Types denote a well-known Resource Type (i.e. "oic.wk.<thing>"). Such a
1371 Resource is associated with multiple Resource Types and so has an "rt" Property Value of multiple
1372 Resource Type IDs (e.g. "rt": ["oic.r.switch.binary", "oic.r.light.brightness"]). The order of the
1373 Resource Type IDs in the "rt" Property Value is meaningless. For example, "rt":
1374 ["oic.r.switch.binary", "oic.r.light.brightness"] and "rt": ["oic.r.light.brightness", "oic.r.switch.binary"]
1375 have the same meaning.

1376 Resource Types for multi-value "rt" Resources shall satisfy the following conditions:

- 1377 – Property Name – Property Names for each Resource Type shall be unique (within the scope of
1378 the multi-value "rt" Resource) with the exception of Common Properties, otherwise there will be
1379 conflicting Property semantics. If two Resource Types have a Property with the same Property
1380 Name, a multi-value "rt" Resource shall not be composed of these Resource Types.

1381 A multi-value "rt" Resource satisfies all the requirements for each Resource Type and conforms to
1382 the OpenAPI 2.0 definitions for each component Resource Type. Thus the mandatory Properties
1383 of a multi-value "rt" Resource shall be the union of all the mandatory Properties of each Resource
1384 Type. For example, mandatory Properties of a Resource with "rt": ["oic.r.switch.binary",
1385 "oic.r.light.brightness"] are "value" and "brightness", where the former is mandatory for
1386 "oic.r.switch.binary" and the latter for "oic.r.light.brightness".

1387 The multi-value "rt" Resource Interface set shall be the union of the sets of OCF Interfaces from
1388 the component Resource Types. The Resource Representation in response to a CRUDN action on
1389 an OCF Interface shall be the union of the schemas that are defined for that OCF Interface. The
1390 Default OCF Interface for a multi-value "rt" Resource shall be the baseline OCF Interface
1391 ("oic.if.baseline") as that is the only guaranteed common OCF Interface between the Resource
1392 Types.

1393 For clarity if each Resource Type supports the same set of OCF Interfaces, then the resultant multi-
1394 value "rt" Resource has that same set of OCF Interfaces with a Default OCF Interface of baseline
1395 ("oic.if.baseline").

1396 See 7.10.3 for the handling of query parameters as applied to a multi-value "rt" Resource.

1397 **7.5 Device Type**

1398 A Device Type is a class of Device. Each Device Type defined will include a list of minimum
1399 Resource Types that a Device shall implement for that Device Type. A Device may expose
1400 additional standard and vendor defined Resource Types beyond the minimum list. The Device Type
1401 is used in Resource discovery as specified in 11.3.4.

1402 Like a Resource Type, a Device Type can be used in the Resource Type Common Property or in a
1403 Link using the Resource Type Parameter.

1404 A Device Type may either be pre-defined (ISO/IEC 30118-5:2018) or in custom definitions by
1405 manufacturers, end users, or developers of Devices (vendor-defined Device Types). Device Types
1406 and their definition details may be communicated out of band (like in documentation).

1407 Every Device Type shall be identified with a Resource Type ID using the same syntax constraints
1408 as a Resource Type.

1409 7.6 OCF Interface

1410 7.6.1 Introduction

1411 An OCF Interface provides first a view into the Resource and then defines the requests and
1412 responses permissible on that view of the Resource. So this view provided by an OCF Interface
1413 defines the context for requests and responses on a Resource. Therefore, the same request to a
1414 Resource when targeted to different OCF Interfaces may result in different responses.

1415 An OCF Interface may be defined by either this document (a Core OCF Interface), ISO/IEC 30118-
1416 5:2018 (a vertical OCF Interface) or manufacturers, end users or developers of Devices (a vendor-
1417 defined OCF Interface).

1418 The OCF Interface Property lists all the OCF Interfaces the Resource support. All Resources shall
1419 have at least one OCF Interface. The Default OCF Interface shall be defined by an OCF
1420 specification and inherited from the Resource Type definition. The Default OCF Interface
1421 associated with all Resource Types defined in this documentshall be the supported OCF Interface
1422 listed first within the applicable enumeration in the definition of the Resource Type (see Annex D).
1423 All Default OCF Interfaces specified in an OCF specification shall be mandatory.

1424 In addition to any OCF specification defined OCF Interface, all Resources shall support the
1425 baseline OCF Interface ("oic.if.baseline") as defined in 7.6.3.2.

1426 See 7.10.4 for the use of queries to enable selection of a specific OCF Interface in a request.

1427 An OCF Interface may accept more than one media type. An OCF Interface may respond with more
1428 than one media type. The accepted media types may be different from the response media types.
1429 The media types are specified with the appropriate header parameters in the transfer protocol.
1430 (NOTE: This feature has to be used judiciously and is allowed to optimize representations on the
1431 wire) Each OCF Interface shall have at least one media type.

1432

1433 7.6.2 OCF Interface Property

1434 **Table 7 – Resource Interface Property definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
OCF Interface	"if"	"array"	Array of strings, conveying OCF Interfaces	N/A	R	Yes	Property to declare the OCF Interfaces supported by a Resource.

1435

1436 The OCF Interfaces supported by a Resource shall be declared using the OCF Interface Common
1437 Property (Table 7), e.g., "if": ["oic.if.ll", "oic.if.baseline"]. The Property Value of an OCF Interface
1438 Property shall be a lower case string with segments separated by a "." (dot). The string "oic", when
1439 used as the first segment in the OCF Interface Property Value, is reserved for OCF-defined OCF

1440 Interfaces. The OCF Interface Property Value may also be a reference to an authority similar to
 1441 IANA that may be used to find the definition of an OCF Interface. A Resource Type shall support
 1442 one or more of the OCF Interfaces defined in 7.6.3.

1443 **7.6.3 OCF Interface methods**

1444 **7.6.3.1 Overview**

1445 OCF Interface methods shall not violate the defined OpenAPI 2.0 definitions for the Resources as
 1446 defined in Annex D.

1447 The defined OCF Interfaces are listed in Table 8:

1448 **Table 8 – OCF standard OCF Interfaces**

OCF Interface	Name	Applicable Operations	Description
baseline	"oic.if.baseline"	RETRIEVE, NOTIFY, UPDATE	The baseline OCF Interface defines a view into all Properties of a Resource including the Meta Properties. This OCF Interface is used to operate on the full Representation of a Resource.
links list	"oic.if.ll"	RETRIEVE, NOTIFY	The links list OCF Interface provides a view into Links in a Collection (Resource). Since Links represent relationships to other Resources, the links list OCF Interfaces may be used to discover Resources with respect to a context. The discovery is done by retrieving Links to these Resources. For example: the Core Resource "/oic/res" uses this OCF Interface to allow discovery of Resource hosted on a Device.
batch	"oic.if.b"	RETRIEVE, NOTIFY, UPDATE	The batch OCF Interface is used to interact with a Collection of Resources at the same time. This also removes the need for the Client to first discover the Resources it is manipulating – the Server forwards the requests and aggregates the responses
read-only	"oic.if.r"	RETRIEVE NOTIFY	The read-only OCF Interface exposes the Properties of a Resource that may be read. This OCF Interface does not provide methods to update Properties, so can only be used to read Property Values.
read-write	"oic.if.rw"	RETRIEVE, NOTIFY, UPDATE	The read-write OCF Interface exposes only those Properties that may be read from a Resource during a RETRIEVE operation and only those Properties that may be written to a Resource during and UPDATE operation.
actuator	"oic.if.a"	RETRIEVE, NOTIFY, UPDATE	The actuator OCF Interface is used to read or write the Properties of an actuator Resource.
sensor	"oic.if.s"	RETRIEVE, NOTIFY	The sensor OCF Interface is used to read the Properties of a sensor Resource.

1449

1450 **7.6.3.2 Baseline OCF Interface**

1451 **7.6.3.2.1 Overview**

1452 The Representation that is visible using the baseline OCF Interface includes all the Properties of
 1453 the Resource including the Common Properties. The baseline OCF Interface shall be defined for
 1454 all Resource Types. All Resources shall support the baseline OCF Interface.

1455 **7.6.3.2.2 Use of RETRIEVE**

1456 The baseline OCF Interface is used when a Client wants to retrieve all Properties of a Resource;
 1457 that is the Server shall respond with a Resource representation that includes all of the implemented

1458 Properties of the Resource. When the Server is unable to send back the whole Resource
1459 representation, it shall reply with an error message. The Server shall not return a partial Resource
1460 representation.

1461 An example response to a RETRIEVE request using the baseline OCF Interface:

```
1462 {  
1463   "rt": ["oic.r.temperature"],  
1464   "if": ["oic.if.a", "oic.if.baseline"],  
1465   "temperature": 20,  
1466   "units": "C",  
1467   "range": [0,100]  
1468 }
```

1469 **7.6.3.2.3 Use of UPDATE**

1470 Using the baseline OCF Interface, all Properties of a Resource with the exception of Common
1471 Properties may be modified using an UPDATE request with a list of Properties and their desired
1472 values if a Resource Type has an associated schema for UPDATE using baseline. If the OCF
1473 Interfaces exposed by a Resource in addition to the baseline OCF Interface do not support the
1474 UPDATE semantic then UPDATE using the baseline OCF Interface is also not supported.

1475 **7.6.3.3 Links List OCF Interface**

1476 **7.6.3.3.1 Overview**

1477 The links list OCF Interface provides a view into the list of Links in a Resource. The Representation
1478 visible through this OCF Interface has only the Links exposed as Property(-ies) that is(are) an array
1479 (or arrays) of Links by the Resource – so this OCF Interface is used to manipulate or interact with
1480 the list of Links. The Links list may be RETRIEVED using this OCF Interface.

1481 The links list OCF Interface is defined as follows:

- 1482 – The links list OCF Interface name is "oic.if.ll".
- 1483 – If there are no Links present in a Resource, then an empty list shall be returned in response to
1484 a RETRIEVE request using the links list OCF Interface.
- 1485 – The Representation determined by this OCF Interface depends on the requesting Client. For a
1486 Client that includes an OCF-Accept-Content-Format-Version option as defined in 12.2.5 in the
1487 request the response only includes the Property value(s) of the Property(-ies) that are arrays
1488 of Links, hence a Collection or "/oic/res" response with oic.if.ll is an array of Links. For a Client
1489 that does not include an OCF-Accept-Content-Format-Version option the response is as defined
1490 in Annex E.
- 1491 – The array of Links may be observed by a Client using the links list OCF Interface (i.e. by
1492 following the procedures in clause 11.4.2 with the addition of a query parameter of "?if=oic.if.ll").
- 1493 – Any CREATE, UPDATE, or DELETE operation on any Link in the array of Links shall result in
1494 the complete Resource representation for the links list OCF Interface as defined for the target
1495 Resource (i.e. the full array of Links) subject to any applied filtering being provided in the
1496 notification that is sent to the Client that initiated the Observe request.
- 1497 – If the act of deleting a Link results in no Links being present then an empty list shall be sent in
1498 a notification.

1499 **7.6.3.3.2 Example: links list OCF Interface**

1500 A request to a Collection, where the request is to RETRIEVE the Links in room (the Links could be
1501 referencing lights, fans, electric sockets etc).

```
1502 GET ocf://<devID>/a/room/1?if=oic.if.ll  
1503 The response would be the array of OCF Links  
1504
```

```

1505 [
1506   {
1507     "href": "/the/light/1",
1508     "rt": ["oic.r.switch.binary"],
1509     "if": ["oic.if.a", "oic.if.baseline"],
1510     "eps": [
1511       {"ep": "coaps://[2001:db8:a::b1d4]:55555"}]
1512     },
1513     {
1514       "href": "/the/light/2",
1515       "rt": ["oic.r.switch.binary"],
1516       "if": ["oic.if.a", "oic.if.baseline"],
1517       "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:55555"}]
1518     },
1519     {
1520       "href": "/my/fan/1",
1521       "rt": ["oic.r.switch.binary"],
1522       "if": ["oic.if.a", "oic.if.baseline"],
1523       "eps": [
1524         {"ep": "coaps://[2001:db8:a::b1d4]:55555"}]
1525         {
1526           "href": "/his/fan/2",
1527           "rt": ["oic.r.switch.binary"],
1528           "if": ["oic.if.a", "oic.if.baseline"],
1529           "eps": [
1530             {"ep": "coaps://[2001:db8:a::b1d4]:55555"}]
1531           }
1532       ]

```

1533 7.6.3.4 Batch OCF Interface

1534 7.6.3.4.1 Overview

1535 The batch OCF Interface is used to interact with a Collection of Resources using a single/same
 1536 Request. The batch OCF Interface can be used to RETRIEVE or UPDATE the Properties of the
 1537 linked Resources with a single request.

1538 The batch OCF Interface is defined as follows:

- 1539 – The batch OCF Interface name is "oic.if.b"
- 1540 – A Collection Resource has linked Resources that are represented as URIs. In the "href"
 1541 Property of the batch payload the URI shall be fully qualified for remote Resources and a
 1542 relative reference for local Resources.
- 1543 – The original request is modified to create new requests targeting each of the linked Resources
 1544 in the Collection by substituting the URI in the original request with the URI of the linked
 1545 Resource. The payload in the original request is replicated in the payload of the new requests.
- 1546 – The requests shall be forwarded assuming use of the Default OCF Interface of the linked
 1547 Resources.
- 1548 – Requests shall only be forwarded to linked Resources that are identified by relation types "item"
 1549 or "hosts" ("hosts" is the default relation type value should the "rel" Link Parameter not be
 1550 present). Requests shall not be forwarded to linked Resources that do not contain the "item" or
 1551 "hosts" relation type values.
- 1552 – Properties of the Collection Resource itself may be included in payloads using "oic.if.b" OCF
 1553 Interface by exposing a single Link with the link relation "self" along with "item" within the
 1554 Collection, and ensuring that Link resolution cannot become an infinite loop due to recursive
 1555 references. For example, if the Default OCF Interface of the Collection is "oic.if.b", then the
 1556 Server might recursively include its batch representation within its batch representation, in an
 1557 endless loop. See 7.6.3.4.2 for an example of use of a Link containing "rel": ["self", "item"] to

- 1558 include Properties of the Collection Resource, along with linked Resources, in "oic.if.b"
1559 payloads.
- 1560 – If the Default OCF Interface of a Collection Resource is exposed using the Link relation "self",
1561 and the Default OCF Interface contains Properties that expose any Links, those Properties shall
1562 not be included in a batch representation which includes the "self" Link.
 - 1563 – Any request forwarded to a linked Resource that is a Collection (including a "self" Link reference)
1564 shall have the Default OCF Interface of the linked Collection Resource applied.
 - 1565 – All the responses from the linked Resources shall be aggregated into a single Response to the
1566 Client. The Server may timeout the response to a time window, the Server may choose any
1567 appropriate window based on conditions.
 - 1568 – If a linked Resource cannot process the request, an empty response, i.e. a JSON object with
1569 no content ("{}") as the representation for the "rep" Property, or error response should the linked
1570 Resource Type provide an error schema or diagnostic payload, shall be returned by the linked
1571 Resource. These empty or error responses for all linked Resources that exhibit an error shall
1572 be included in the aggregated response to the original Client request. See the example in
1573 7.6.3.4.2.
 - 1574 – If any of the linked Resources returns an error response, the aggregated response sent to the
1575 Client shall also indicate an error (e.g. 4.xx in CoAP). If all of the linked Resources return
1576 successful responses, the aggregated response shall include the success response code.
 - 1577 – The aggregated response shall be an array of objects representing the responses from each
1578 linked Resource. Each object in the response shall include at least two items: (1) the URI of
1579 the linked Resource (fully qualified for remote Resources, or a relative reference for local
1580 Resources) as "href": <URI> and (2) the individual response object or array of objects if the
1581 linked Resource is itself a Collection using "rep" as the key, e.g. "rep": { <representation of
1582 individual response> }.
 - 1583 – If the Collection Resource is marked as Observable, linked Resources referenced in the
1584 Collection may be Observed using the batch OCF Interface. If the Collection Resource is not
1585 marked as Observable then the Collection cannot be Observed and Observe requests to the
1586 Collection shall be handled as defined for the case where request validation fails in clause
1587 11.4.2.4. The Observe mechanism shall work as defined in 11.4.2 with the Observe request
1588 forwarded to each of the linked Resources. All responses to the request shall be aggregated
1589 into a single response to the Client using the same representations and status codes as for
1590 RETRIEVE operations using the batch OCF Interface.
 - 1591 – Should any one of the Observable linked Resources fail to honour the Observe request the
1592 response to the batch Observe request shall also indicate that the entire request was not
1593 honoured using the mechanism described in 11.4.2.4.
 - 1594 – If any of the Observable Resources in a request to a Collection using the batch OCF Interface
1595 replies with an error or Observe Cancel, the Observations of all other linked Resources shall
1596 be cancelled and the error or Observe Cancel status shall be returned to the Observing Client.
- 1597 NOTE Behavior may be different for Links that do network requests vs. local Resources.
- 1598 – All notifications to the Client that initiated an Observe request using the batch OCF Interface
1599 shall use the batch representation for the Collection. This is the aggregation of any individual
1600 Observe notifications received by the Device hosting the Collection from the individual Observe
1601 requests that were forwarded to the linked Resources.
 - 1602 – Linked Resources which are not marked Observable in the Links of a Collection shall not trigger
1603 Notifications, but may be included in the response to, and subsequent Notifications resulting
1604 from, an Observe request to the batch OCF Interface of a Collection.
 - 1605 – Each notification shall contain the most current values for all of the Linked Resources that would
1606 be included if the original Observe request were processed again. The Server hosting the

- 1607 Collection may choose to RETRIEVE all of the linked Resources each time, or may choose to
1608 employ caching to avoid retrieving linked Resources on each Notification.
- 1609 – If a Linked Resource is Observable and has responded with a successful Observe response,
1610 the most recently reported value of that Resource is considered to be the most current value
1611 and may be reported in all subsequent Notifications.
 - 1612 – Links in the Collection should be Observed by using the "oic.if.ll" OCF Interface. A notification
1613 shall be sent any time the contents of the "oic.if.ll" OCF Interface representation are changed;
1614 that is, if a Link is added, if a Link is removed, or if a Link is updated. Notifications on the
1615 "oic.if.ll" OCF Interface shall contain all of the Links in the "oic.if.ll" OCF Interface representation.
 - 1616 – Other Properties of the Collection Resource, if present, may be Observed by using the OCF
1617 Interfaces defined in the definition for the Resource Type, including using the "oic.if.baseline"
1618 OCF Interface.
 - 1619 – The Client may choose to restrict the linked Resources to which the request is forwarded by
1620 including additional query parameters in the request. The Server should process any additional
1621 query parameters in a request that includes "oic.if.b" as selectors for linked Resources that are
1622 to be processed by the request.
 - 1623 – A Client shall perform UPDATE operations using the batch OCF Interface by creating a payload
1624 that is similar to a RETRIEVE response payload from a batch OCF Interface request. The Server
1625 shall send a separate UPDATE request to each of the linked Resources according to each "href"
1626 Property and the corresponding value of the "rep" Property.
 - 1627 – If the "href" value is empty, denoted by a zero length string or "" in JSON, the "rep" Property
1628 shall be applied to linked Resources in the Collection.
 - 1629 – Items with the empty "href" and link-specific "href" shall not be mixed in the same UPDATE
1630 request.
 - 1631 – All of the Properties in the UPDATE request may not be supported by the linked Resource. In
1632 such cases, writable Properties in the UPDATE request that are supported by the linked
1633 Resource shall be modified and Properties that are not supported shall be silently ignored.
 - 1634 – The UPDATE response shall contain the updated values using the same payload schema as
1635 RETRIEVE operations if provided by the linked Resource, along with the appropriate status
1636 code. The aggregated response payload shall reflect the known state of the updated Properties
1637 after the batch update was completed. If no payload is provided by the updated Resource then
1638 an empty response (i.e. "rep": {}) shall be provided for that Resource.
 - 1639 – A Collection shall not support the use of the UPDATE operation to add, modify or remove Links
1640 in an existing Collection using the "oic.if.baseline", "ic.if.rw" or "oic.if.a" OCF Interfaces.

1641 **7.6.3.4.2 Examples: Batch OCF Interface**

1642 Note that the examples provided in Table 9 are illustrative and do not include all mandatory schema
1643 elements in all cases. It is assumed that the Default OCF Interface for the Resource Type
1644 "x.org.example.rt.room" is specified in its Resource Type definition file as "oic.if.rw", which exposes
1645 the Properties "x.org.example.colour" and "x.org.example.size".

Table 9 – Batch OCF Interface Example

Resources	
	<pre> /a/room/1 { "rt": "x.org.example.rt.room", "if": ["oic.if.rw","oic.if.baseline","oic.if.b","oic.if.ll"], "x.org.example.colour": "blue", "x.org.example.dimension": "15bx15wx10h", "links": [{ "href": "/a/room/1", "rel": ["self", "item"], "rt": ["x.org.example.rt.room"], "if": ["oic.if.rw","oic.if.baseline","oic.if.b","oic.if.ll"],"p": {"bm": 2} }, { "href": "/the/light/1", "rel": ["item"], "rt": ["oic.r.switch.binary"], "if": ["oic.if.a","oic.if.baseline"], "ins": "11111", "p": {"bm": 2} }, { "href": "/the/light/2", "rel": ["item"], "rt": ["oic.r.switch.binary"], "if": ["oic.if.a", "oic.if.baseline"], "ins": "22222", "p": {"bm": 2} }, { "href": "/my/fan/1", "rel": ["item"], "rt": ["oic.r.switch.binary"], "if": ["oic.if.a", "oic.if.baseline"], "ins": "33333", "p": {"bm": 2} }, { "href": "/his/fan/2", "rel": ["item"], "rt": ["oic.r.switch.binary"], "if": ["oic.if.a", "oic.if.baseline"], "ins": "44444", "p": {"bm": 2} }, { "href": "/the/switches/1", "rel": ["item"], "rt": ["oic.wk.col"], "if": ["oic.if.ll", "oic.if.b", "oic.if.baseline"], "ins": "55555", "p": {"bm": 2} }] } /the/light/1 { "rt": ["oic.r.switch.binary"], "if": ["oic.if.a", "oic.if.baseline"], "value": false } /the/light/2 { "rt": ["oic.r.switch.binary"], "if": ["oic.if.a", "oic.if.baseline"], "value": true } /my/fan/1 { "rt": ["oic.r.switch.binary"], "if": ["oic.if.a", "oic.if.baseline"], "value": true } /his/fan/2 { "rt": ["oic.r.switch.binary"], "if": ["oic.if.a", "oic.if.baseline"], "value": false } /the/switches/1 { "rt": ["oic.wk.col"], "if": ["oic.if.ll", "oic.if.b", "oic.if.baseline"], "links": [{ "href": "/switch-1a", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a","oic.if.baseline"], "p": {"bm": 2} }] } </pre>

	<pre>{ "href": "/switch-lb", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a","oic.if.baseline"], "p": {"bm": 2 } }</pre>
--	---

<p>Use of batch, successful response</p>	<pre> Request: GET /a/room/1?if=oic.if.b Becomes the following individual request messages issued by the Device in the Client role GET /a/room/1 (NOTE: uses the Default OCF Interface as specified for the Collection Resource, in this example oic.if.rw) GET /the/light/1 (NOTE: Uses the Default OCF Interface as specified for this Resource) GET /the/light/2 (NOTE: Uses the Default OCF Interface as specified for this Resource) GET /my/fan/1 (NOTE: Uses the Default OCF Interface as specified for this Resource) GET /his/fan/2 (NOTE: Uses the Default OCF Interface as specified for this Resource) GET /the/switches/1 (NOTE: Uses the Default OCF Interface for the Collection that is within the Collection) Response: [{ "href": "/a/room/1", "rep": {"x.org.example.colour": "blue", "x.org.example.dimension": "15bx15wx10h"} }, { "href": "/the/light/1", "rep": {"value": false} }, { "href": "/the/light/2", "rep": {"value": true} }, { "href": "/my/fan/1", "rep": {"value": true} }, { "href": "/his/fan/2", "rep": {"value": false} }, { "href": "/the/switches/1", "rep": [{ "href": "/switch-1a", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a", "oic.if.baseline"], "p": {"bm": 2}, "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:5555"}] }, { "href": "/switch-1b", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a", "oic.if.baseline"], "p": {"bm": 2}, "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:5555"}] }] }] </pre>
---	---

<p>Use of batch, error response</p>	<p>Should any of the RETRIEVE requests in the previous example fail then the response includes an empty payload for that Resource instance and an error code is sent. The following example assumes errors from "/my/fan/1" and "/the/switches/1"</p> <p>Error Response:</p> <pre>[{ "href": "/a/room/1", "rep": {"x.org.example.colour": "blue", "x.org.example.dimension": "15bx15wx10h"} }, { "href": "/the/light/1", "rep": {"value": false} }, { "href": "/the/light/2", "rep": {"value": true} }, { "href": "/my/fan/1", "rep": {} }, { "href": "/his/fan/2", "rep": {"value": false} }, { "href": "/the/switches/1", "rep": {} }]</pre>
--	--

<p>Use of batch</p> <p>(UPDATE has POST semantics)</p>	<pre>UPDATE /a/room/1?if=oic.if.b [{ "href": "", "rep": { "value": false } }]</pre> <p>Since the "href" value in the UPDATE request is empty, the request is forwarded to all Resources in the Collection and becomes:</p> <pre>UPDATE /a/room/1 { "value": false } UPDATE /the/light/1 { "value": false } UPDATE /the/light/2 { "value": false } UPDATE /my/fan/1 { "value": false } UPDATE /his/fan/2 { "value": false } UPDATE /the/switches/1 { "value": false }</pre> <p>Response:</p> <pre>[{ "href": "/the/light/1", "rep": {"value": false} }, { "href": "/the/light/2", "rep": {"value": false} }, { "href": "/my/fan/1", "rep": {"value": false} }, { "href": "/his/fan/2", "rep": {"value": false} }, { "href": "/the/switches/1", "rep": { } }]</pre> <p>Since /a/room/1 does not have a "value" Property exposed by its Default OCF Interface, the UPDATE request will be silently ignored and it will not be included in the UPDATE response.</p> <p>Since the UPDATE request with the links list OCF Interface is not allowed, an empty payload for the "/the/switches/1" is included in the UPDATE response and an error code is sent.</p>
--	--

**Use of batch
(UPDATE has
POST
semantics)**

```
UPDATE /a/room/1?if=oic.if.b
[
  {
    "href": "/the/light/1",
    "rep": {
      "value": false
    }
  },
  {
    "href": "/the/light/2",
    "rep": {
      "value": true
    }
  },
  {
    "href": "/a/room/1",
    "rep": {
      "x.org.example.colour": "red"
    }
  }
]
```

This turns /the/light/1 off, turns /the/light/2 on, and sets the colour of /a/room/1 to "red".

The response will be same as response for GET /a/room/1?if=oic.if.b with the updated Property values as shown.

```
[
  {
    "href": "/a/room/1",
    "rep": {"x.org.example.colour": "red",
           "x.org.example.dimension": "15bx15wx10h"}
  },
  {
    "href": "/the/light/1",
    "rep": {"value": false}
  },
  {
    "href": "/the/light/2",
    "rep": {"value": true}
  }
]
```

Example use of additional query parameters to select items by matching Link Parameters.

Turn on light 1 based on the "ins" Link Parameters value of "11111"

```
UPDATE /a/room/1?if=oic.if.b&ins=11111
[
  {
    "href": "",
    "rep": {
      "value": false
    }
  }
]
```

Similar to the earlier example, "href": "" applies the UPDATE request to all of the Resources in the Collection. Since the additional query parameter ins=11111 selects only links that have a matching "ins" value, only one link is selected. The payload is applied to the target Resource of that link, /the/light/1.

	<p>Retrieving the item using the same query parameter:</p> <pre>RETRIEVE /a/room/1?if=oic.if.b&ins=11111</pre> <p>Response payload:</p> <pre>[{ "href": "/the/light/1", "rep": { "value": false } }]</pre>
--	--

1647

1648 **7.6.3.5 Actuator OCF Interface**

1649 The actuator OCF Interface is the OCF Interface for viewing Resources that may be actuated i.e.
 1650 changes some value within or the state of the entity abstracted by the Resource:

- 1651 – The actuator OCF Interface name shall be "oic.if.a"
- 1652 – The actuator OCF Interface shall expose in the Resource Representation all mandatory
 1653 Properties as defined by the applicable OpenAPI 2.0 schema; the actuator OCF Interface may
 1654 also expose in the Resource Representation optional Properties as defined by the applicable
 1655 OpenAPI 2.0 schema that are implemented by the target Device.

1656 For example, a "Heater" Resource (for illustration only):

```
1657 /a/act/heater
1658 {
1659   "rt": ["acme.gas"],
1660   "if": ["oic.if.baseline", "oic.if.r", "oic.if.a", "oic.if.s"],
1661   "settemp": 10,
1662   "currenttemp" : 7
1663 }
```

1664 The actuator OCF Interface with respect to "Heater" Resource (for illustration only):

1665

1666 a) Retrieving values of an actuator.

1667 Request: GET /a/act/heater?if="oic.if.a"

1668

1669 Response:

```
1670 {
1671   "settemp": 10,
1672   "currenttemp" : 7
1673 }
```

1674 b) Correct use of actuator OCF Interface.

1675

1676 Request: POST /a/act/heater?if="oic.if.a"

1677

```
{
  "settemp": 20
}
```

1679

1680 Response:

```
1681 {
1682   Ok
1683 }
```

1684 c) Incorrect use of actuator OCF Interface.

1685

1686 Request: POST /a/act/heater?if="oic.if.a"
1687 {
1688 "if": ["oic.if.s"] ← this is visible through baseline OCF Interface
1689 }
1690 Response:
1691 {
1692 Error
1693 }

1694 – A RETRIEVE request using this OCF Interface shall return the Representation for this Resource
1695 subject to any query and filter parameters that may also exist.

1696 – An UPDATE request using this OCF Interface shall provide a payload or body that contains the
1697 Properties that will be updated on the target Resource.

1698 **7.6.3.6 Sensor OCF Interface**

1699 The sensor OCF Interface is the OCF Interface for retrieving measured, sensed or capability
1700 specific information from a Resource that senses:

1701 – The sensor OCF Interface name shall be "oic.if.s".

1702 – The sensor OCF Interface shall expose in the Resource Representation all mandatory
1703 Properties as defined by the applicable OpenAPI 2.0 schema; the sensor OCF Interface may
1704 also expose in the Resource Representation optional Properties as defined by the applicable
1705 OpenAPI 2.0 schema that are implemented by the target Device.

1706 – A RETRIEVE request using this OCF Interface shall return this representation for the Resource
1707 subject to any query and filter parameters that may also exist.

1708 NOTE: The example here is with respect to retrieving values of a sensor

1709
1710 Request: GET /a/act/heater?if="oic.if.s"
1711
1712 Response:
1713 {
1714 "currenttemp": 7
1715 }
1716

1717 **Incorrect use of the sensor.**

1718 Request: PUT /a/act/heater?if="oic.if.s" ← PUT is not allowed
1719 {
1720 "settemp": 20 ← this is possible through actuator OCF Interface
1721 }
1722 Response:
1723 {
1724 Error
1725 }
1726

1727 **Another incorrect use of the sensor.**

1728 Request: POST /a/act/heater?if="oic.if.s" ← POST is not allowed
1729 {
1730 "currenttemp": 15 ← this is possible through actuator OCF Interface
1731 }
1732 Response:
1733 {
1734 Error
1735 }

1736 **7.6.3.7 Read-only OCF Interface**

1737 The read-only OCF Interface exposes only the Properties that may be read. This includes
1738 Properties that may be read-only, read-write but not Properties that are write-only or set-only. The
1739 applicable operations that can be applied to a Resource are only RETRIEVE and NOTIFY. An
1740 attempt by a Client to apply a method other than RETRIEVE or NOTIFY to a Resource shall be
1741 rejected with an error response code.

1742 **7.6.3.8 Read-write OCF Interface**

1743 The read-write OCF Interface is a generic OCF Interface to support reading and setting Properties
1744 in a Resource. The applicable methods that can be applied to a Resource are only RETRIEVE,
1745 NOTIFY, and UPDATE. For the RETRIEVE and NOTIFY operations, the behaviour is the same as
1746 for the "oic.if.r" OCF Interface defined in 7.6.3.7. For the UPDATE operation, read-only Properties
1747 (i.e. Properties tagged with "readOnly=True" in the OpenAPI 2.0 definition) shall not be in the
1748 UPDATE payload. An attempt by a Client to apply a method other than RETRIEVE, NOTIFY, or
1749 UPDATE to a Resource shall be rejected with an error response code.

1750 **7.7 Resource representation**

1751 Resource representation captures the state of a Resource at a particular time. The Resource
1752 representation is exchanged in the request and response interactions with a Resource. A Resource
1753 representation may be used to retrieve or update the state of a Resource.

1754 The Resource representation shall not be manipulated by the data connectivity protocols and
1755 technologies (e.g., CoAP, UDP/IP or BLE).

1756 **7.8 Structure**

1757 **7.8.1 Introduction**

1758 In many scenarios and contexts, the Resources may have either an implicit or explicit structure
1759 between them. A structure can, for example, be a tree, a mesh, a fan-out or a fan-in. The
1760 Framework provides the means to model and map these structures and the relationships among
1761 Resources. The primary building block for Resource structures in Framework is the Collection. A
1762 Collection represents a container, which is extensible to model complex structures.

1763 **7.8.2 Resource Relationships**

1764 **7.8.2.1 Introduction**

1765 Resource relationships are expressed as Links. A Link embraces and extends typed web links
1766 concept as a means of expressing relationships between Resources. A Link consists of a set of
1767 Parameters that define:

- 1768 – a context URI,
- 1769 – a target URI,
- 1770 – a relation from the context URI to the target URI, and
- 1771 – elements that provide metadata about the target URI, the relationship or the context of the Link.

1772 The target URI is mandatory and the other items in a Link are optional. Additional items in the Link
1773 may be made mandatory based on the use of the links in different contexts (e.g. in Collections, in
1774 discovery, in bridging etc.). OpenAPI 2.0 schema for the Link payload is provided in Annex D.

1775 An example of a Link is:

```
1776 {"href": "/switch", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a",  
1777 "oic.if.baseline"], "p": {"bm": 3}, "rel": "item"}
```

1778 Two Links are distinct from each other when at least one Parameter is different. For example the
1779 two Links show here are distinct and can appear in the same list of Links.


```

1780 {"href": "/switch", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a",
1781 "oic.if.baseline"], "p": {"bm": 2}, "rel": "item"}
1782 {"href": "/switch", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a",
1783 "oic.if.baseline"], "p": {"bm": 2}}

```

1784 The document may mandate Parameters and Parameter values as required for certain capabilities.
1785 For all Links returned in a response to a RETRIEVE on "/oic/res", if a Link does not explicitly include
1786 the "rel" Parameter, a value of "rel"="hosts" shall be assumed. The relation value of "hosts" is
1787 defined by IETF RFC 6690, the value of "item" by IETF RFC 6573, and the value of "self" by
1788 IETF RFC 4287 and all are registered in the IANA Registry for Link Relations defined in
1789 IANA Link Relations.

1790 As shown in Annex D the relation between the context URI and target URI in a Link is specified
1791 using the "rel" JSON element and the value of this element specifies the particular relation.

1792 The context URI of the Link shall implicitly be the URI of the Resource (or specifically a Collection)
1793 that contains the Link unless the Link specifies the "anchor" Parameter. The "anchor" Parameter
1794 is used to change the context URI of a Link – the relationship with the target URI is based off the
1795 anchor URI when the "anchor" is specified. "Anchor" Parameter uses transfer protocol URI for OIC
1796 1.1 Link (e.g. "anchor": "coaps://[fe80::b1d6]:44444") and OCF URI defined in Sec 6 for OCF 1.0
1797 Links (e.g. "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989"). For optional backward
1798 compatibility with OIC 1.1, "anchor" Parameter uses transfer protocol URI for OIC 1.1 Link (e.g.
1799 "anchor": "coaps://[fe80::b1d6]:44444").

1800 An example of using "anchors" in the context of Collections – a floor has rooms and rooms have
1801 lights – the lights may be defined in floor as Links but the Links will have the "anchor" set to the
1802 URI of the rooms that contain the lights (the relation is contains). This allows all lights in a floor to
1803 be turned on or off together while still having the lights defined with respect to the rooms that
1804 contain them (lights may also be turned on by using the room URI too). For example, here is the
1805 use of "anchor" in Link:

```

1806 /a/floor {
1807   "links": [
1808     {
1809       "href": "/x/light1",
1810       "anchor": "/a/room1",   ** Note: /a/room1 has the item relationship with /x/light1; not /a/floor **
1811       "rel": "item"
1812     }
1813   ]
1814 }
1815
1816 /a/room1 {
1817   "links": [
1818     {
1819       ** Note: /a/room1 "contains" the /x/light since /a/room1 is the implicit context URI **
1820       "href": "/x/light1",
1821       "rel": "item"
1822     }
1823   ]
1824 }

```

1825 7.8.2.2 Parameters

1826 7.8.2.2.1 "ins" or Link Instance Parameter

1827 The "ins" Parameter identifies a particular Link instance in a list of Links. The "ins" Parameter may
1828 be used to modify or delete a specific Link in a list of Links. The value of the "ins" Parameter is set
1829 at instantiation of the Link by the OCF Device (Server) that is hosting the list of Links – once it has
1830 been set, the "ins" Parameter shall not be modified for as long as the Link is a member of that list.

1831 **7.8.2.2.2 "p" or Policy Parameter**

1832 The Policy Parameter defines various rules for correctly accessing a Resource referenced by a
 1833 target URI. The Policy rules are configured by a set of key-value pairs.

1834 The policy Parameter "p" is defined by:

- 1835 – "bm" key: The "bm" key corresponds to an integer value that is interpreted as an 8-bit bitmask.
 1836 Each bit in the bitmask corresponds to a specific Policy rule. The rules are specified for "bm"
 1837 in Table 10:

1838 **Table 10 – "bm" Property definition**

Bit Position	Policy rule	Comment
Bit 0 (the LSB)	discoverable	The discoverable rule defines whether the Link is to be included in the Resource discovery message via "/oic/res". If the Link is to be included in the Resource discovery message, then "p" shall include the "bm" key and set the discoverable bit to value 1. If the Link is NOT to be included in the Resource discovery message, then "p" shall either include the "bm" key and set the discoverable bit to value 0 or omit the "bm" key entirely.
Bit 1 (2 nd LSB)	observable	The Observable rule defines whether the Resource referenced by the target URI supports the NOTIFY operation. With the self-link, i.e. the Link with "rel" value of "self", "/oic/res" can have a Link with the target URI of "/oic/res" and indicate itself Observable. The "self" is defined by IETF RFC 4287 and registered in the IANA Registry for "rel" value defined at IANA Link Relations. If the Resource supports the NOTIFY operation, then "p" shall include the "bm" key and set the Observable bit to value 1. If the Resource does NOT support the NOTIFY operation, then "p" shall either include the "bm" key and set the Observable bit to value 0 or omit the "bm" key entirely.
Bits 2-7	--	Reserved for future use. All reserved bits in "bm" shall be set to value 0.

1839
 1840 NOTE If all the bits in "bm" are defined to value 0, then the "bm" key may be omitted entirely from "p" as an efficiency
 1841 measure. However, if any bit is set to value 1, then "bm" shall be included in "p" and all the bits shall be defined
 1842 appropriately.

- 1843 – "sec" and "port" in the remaining bullets shall be used only in a response payload when the
 1844 request does not include an OCF-Accept-Content-Format-Version option as defined in 12.2.5.
 1845 In a payload sent in response to a request that includes an OCF-Accept-Content-Format-
 1846 Version option "sec" and "port" shall not be used and instead the "eps" Parameter shall provide
 1847 the information for an encrypted connection. See **Annex E** for the schema for the "p" Parameter
 1848 that includes "sec" and "port".

- 1849 – "sec" key: The "sec" key corresponds to a Boolean value that indicates whether the Resource
 1850 referenced by the target URI is accessed via an encrypted connection. If "sec" is true, the
 1851 Resource is accessed via an encrypted connection, using the "port" specified. If "sec" is false,
 1852 the Resource is accessed via an unencrypted connection, or via an encrypted connection (if
 1853 such a connection is made using the "port" settings for another Resource, for which "sec" is
 1854 true).

- 1855 – "port" key: The "port" key corresponds to an integer value that is used to indicate the port
 1856 number where the Resource referenced by the target URI may be accessed via an encrypted
 1857 connection.

- 1858 – If the Resource is only available via an encrypted connection (i.e. DTLS over IP), then

- 1859 – "p" shall include the "sec" key and its value shall be true.
- 1860 – "p" shall include the "port" key and its value shall be the port number where the encrypted
- 1861 connection may be established.
- 1862 – If the Resource is only available via an unencrypted connection, then
 - 1863 – "p" shall include the "sec" key and its value shall be false or "p" shall omit the "sec" key; the
 - 1864 default value of "sec" is false.
 - 1865 – "p" shall omit the "port" key.
- 1866 – If the Resource is available via both an encrypted and unencrypted connection, then
 - 1867 – "p" shall include the "sec" key and its value shall be false or "p" shall omit the "sec" key; the
 - 1868 default value of "sec" is false.
 - 1869 – "p" may omit the "port" key. If the "port" key is omitted, the Resource shall be available
 - 1870 using the same "port" information as another Resource on the Device for which "sec" is true.
- 1871 – Access to the Resource on the port specified by the "port" key shall be made by an encrypted
- 1872 connection (e.g. "coaps://"). (Note that unencrypted connection to the Resource may be
- 1873 possible on a separate port discovered thru multicast discovery).
- 1874 – Note that access to the Resource is controlled by the ACL for the Resource. A successful
- 1875 encrypted connection does not ensure that the requested action will succeed. See
- 1876 ISO/IEC 30118-2:2018 clause 12 for more information.

1877 Example 1: This shows the Policy Parameter for a Resource that is discoverable but not Observable,
1878 and for which authenticated accesses shall be done via CoAPS port 33275.

```
1879 "p": {"bm": 1}
```

1880 Example 2: This shows a self-link, i.e. the "/oic/res" Link in itself that is discoverable and
1881 Observable.

```
1882 {
1883   "href": "/oic/res",
1884   "rel": "self",
1885   "rt": ["oic.wk.res"],
1886   "if": ["oic.if.ll", "oic.if.baseline"],
1887   "p": {"bm": 3}
1888 }
```

1889 **7.8.2.2.3 "type" or Media Type Parameter**

1890 The "type" Parameter may be used to specify the various media types that are supported by a
1891 specific target Resource. The default type of "application/vnd.ocf+cbor" shall be used when the
1892 "type" element is omitted. Once a Client discovers this information for each Resource, it may use
1893 one of the available representations in the appropriate header field of the Request or Response.

1894 **7.8.2.2.4 "di" or Device ID Parameter**

1895 The "di" Parameter specifies the Device ID of the Device that hosts the target Resource defined in
1896 the in the "href" Parameter.

1897 The Device ID may be used to qualify a relative reference used in the "href" or to lookup OCF
1898 Endpoint information for the relative reference.

1899 **7.8.2.2.5 "eps" Parameter**

1900 The "eps" Parameter indicates the OCF Endpoint information of the target Resource.

1901 "eps" shall have as its value an array of items and each item represents OCF Endpoint information
1902 with "ep" and "pri" as specified in 10.2. "ep" is mandatory but "pri" is optional.

1903 This is an example of "eps" with multiple OCF Endpoints.

```
1904 "eps": [  
1905   { "ep": "coap://[fe80::b1d6]:1111", "pri": 2},  
1906   { "ep": "coaps://[fe80::b1d6]:1122"},  
1907   { "ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}  
1908 ]
```

1909 When "eps" is present in a link, the OCF Endpoint information in "eps" can be used to access the
1910 target Resource referred by the "href" Parameter.

1911 Note that the type of OCF Endpoint – Secure or Unsecure – that a Resource exposes merely
1912 determines the connection type(s) guaranteed to be available for sending requests to the Resource.
1913 For example, if a Resource only exposes a single CoAP "ep", it does not guarantee that the
1914 Resource cannot also be accessed via a Secure OCF Endpoint (e.g. via a CoAPS "ep" from another
1915 Resource's "eps information). Nor does exposing a given type of OCF Endpoint ensure that access
1916 to the Resource will be granted using the "ep" information. Whether requests to the Resource are
1917 granted or denied by the Access Control layer is separate from the "eps" information, and is
1918 determined by the configuration of the /acl2 Resource (see ISO/IEC 30118-2:2018 clause 13.5.3
1919 for details).

1920 When present, max-age information (e.g. Max-Age option for CoAP defined in IETF RFC 7252)
1921 determines the maximum time "eps" values may be cached before they are considered stale.

1922 **7.8.2.3 Formatting**

1923 When formatting in JSON, the list of Links shall be an array.

1924 **7.8.2.4 List of Links in a Collection**

1925 A Resource that exposes one or more Properties that are defined to be an array of Links where
1926 each Link can be discretely accessed is a Collection. The Property Name "links" is recommended
1927 for such an array of Links.

1928 This is an example of a Resource with a list of Links.

```
1929 /Room1  
1930 {  
1931   "rt": ["my.room"],  
1932   "if": ["oic.if.ll", "oic.if.baseline" ],  
1933   "color": "blue",  
1934   "links":  
1935   [  
1936     {  
1937       "href": "/oic/d",  
1938       "rt": ["oic.d.light", "oic.wk.d"],  
1939       "if": [ "oic.if.r", "oic.if.baseline" ],  
1940       "p": {"bm": 1}  
1941     },  
1942     {  
1943       "href": "/oic/p",  
1944       "rt": ["oic.wk.p"],  
1945       "if": [ "oic.if.r", "oic.if.baseline" ],  
1946       "p": {"bm": 1}  
1947     },  
1948     {  
1949       "href": "/switch",  
1950       "rt": ["oic.r.switch.binary"],  
1951       "if": [ "oic.if.a", "oic.if.baseline" ],  
1952       "p": {"bm": 3},  
1953       "mt": [ "application/vnd.ocf+cbor", "application/exi+xml" ]  
1954     },  
1955   ]  
1956 }
```

```

1955     {
1956         "href": "/brightness",
1957         "rt": [ "oic.r.light.brightness" ],
1958         "if": [ "oic.if.a", "oic.if.baseline" ],
1959         "p": { "bm": 3 }
1960     }
1961 ]
1962 }

```

1963 7.8.2.5 Properties describing an array of Links

1964 If a Resource Type that defines an array of Links (e.g. Collections, Atomic Measurements) has
1965 restrictions on the "rt" values that can be within the array of Links, the Resource Type will define
1966 the "rts" Property. The "rts" Property as defined in Table 11 will include all "rt" values allowed for
1967 all Links in the array. If the Resource Type does not define the "rts" Property or the "rts" Property
1968 is an empty array, then any "rt" value is permitted in the array of Links.

1969 For all instances of a Resource Type that defines the "rts" Property, the "rt" Link Parameter in
1970 every Link in the array of Links shall be one of the "rt" values that is included in the "rts"
1971 Property.

1972 **Table 11 – Resource Types Property definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Resource Types	"rts"	"array"	Array of strings, conveying Resource Type IDs	N/A	R	No	An array of Resource Types that are supported within an array of Links exposed by a Resource.

1973

1974 If a Resource Type that defines an array of Links has "rt" values which are required to be in the
1975 array, the Resource Type will define the "rts-m" Property, as defined in Table 12, which will contain
1976 all of the "rt" values that are required to be in the array of Links. If "rts-m" is defined, and "rts" is
1977 defined and is not an empty array, then the "rt" values present in "rts-m" will be part of the values
1978 present in "rts". Moreover, if the "rts-m" Property is defined, it shall be mandated (i.e. included in
1979 the "required" field of a JSON definition) in the Resource definition and Introspection Device Data
1980 (see 11.8).

1981 For all instances of a Resource Type that defines the "rts-m" Property, there shall be at least one
1982 Link in the array of Links corresponding to each one of the "rt" values in the "rts-m" Property; for
1983 all such Links the "rt" Link Parameter shall contain at least one of the "rt" values in the "rts-m"
1984 Property.

1985 **Table 12 – Mandatory Resource Types Property definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Mandatory Resource Types	"rts-m"	"array"	Array of strings, conveying Resource Type IDs	N/A	R	No	An array of Resource Types that are mandatory to be exposed within an array of Links exposed by a Resource.

1986

1987 **7.8.3 Collections**

1988 **7.8.3.1 Overview**

1989 A Resource that contains one or more references (specified as Links) to other Resources is a
1990 Collection. These references may be related to each other or just be a list; the Collection provides
1991 a means to refer to this set of references with a single handle (i.e. the URI). A simple Resource is
1992 kept distinct from a Collection. Any Resource may be turned into a Collection by binding Resource
1993 references as Links. Collections may be used for creating, defining or specifying hierarchies,
1994 indexes, groups, and so on.

1995 A Collection shall have at least one Resource Type and at least one OCF Interface bound at all
1996 times during its lifetime. During creation time of a Collection the Resource Type and OCF Interfaces
1997 are specified. The initial defined Resource Types and OCF Interfaces may be updated during its
1998 life time. These initial values may be overridden using mechanism used for overriding in the case
1999 of a Resource. Additional Resource Types and OCF Interfaces may be bound to the Collection at
2000 creation or later during the lifecycle of the Collection.

2001 A Collection shall define a Property that is an array with zero or more Links. The target URIs in the
2002 Links may reference another Collection or another Resource. The referenced Collection or
2003 Resource may reside on the same Device as the Collection that includes that Link (called a local
2004 reference) or may reside on another Device (called a remote reference). The context URI of the
2005 Links in the array shall (implicitly) be the Collection that contains that Property. The (implicit)
2006 context URI may be overridden with explicit specification of the "anchor" Parameter in the Link
2007 where the value of "anchor" is the new base of the Link.

2008 A Resource may be referenced in more than one Collection, therefore, a unique parent-child
2009 relationship is not guaranteed. There is no pre-defined relationship between a Collection and the
2010 Resource referenced in the Collection, i.e., the application may use Collections to represent a
2011 relationship but none is automatically implied or defined. The lifecycles of the Collection and the
2012 referenced Resource are also independent of one another.

2013 If the "drel" Property is defined for the Collection then all Links that don't explicitly specify a
2014 relationship shall inherit this default relationship in the context of that Collection. The default
2015 relationship defines the implicit relationship between the Collection and the target URI in the Link.

2016 In the following example a Property "links" represents the list of Links in a Collection. The "links"
2017 Property has, as its value, an array of items and each item is a Link.

```
2018 /my/house ← This is IRI/URI of the Resosource
2019 {
2020   "rt": ["my.r.house"], ← This and the next 3 lines are the Properies of the Resource
2021   "color": "blue",
2022   "n": "myhouse",
2023   "links": [
2024     { ← This and the next 4 lines are the Parameters of a Liink
2025       "href": "/door",
2026       "rt": ["oic.r.door"],
2027       "if": ["oic.if.b", "oic.if.ll", "oic.if.baseline"]
2028     },
2029
2030     {
2031       "href": "/door/lock",
2032       "rt": ["oic.r.lock"],
2033       "if": ["oic.if.b", "oic.if.baseline"],
2034       "type": ["application/cbor", "application/exi+xml"]
2035     },
2036
2037     {
2038       "href": "/light",
```

```

2039     "rt": ["oic.r.light"],
2040     "if": ["oic.if.s", "oic.if.baseline"]
2041   },
2042
2043   {
2044     "href": "/binarySwitch",
2045     "rt": ["oic.r.switch.binary"],
2046     "if": ["oic.if.a", "oic.if.baseline"],
2047     "type": ["application/cbor"]
2048   }
2049 ]
2050 }
2051 }

```

2052 A Collection may be:

- 2053 – A pre-defined Collection where the Collection has been defined a priori and the Collection is
2054 static over its lifetime. Such Collections may be used to model, for example, an appliance that
2055 is composed of other Devices or fixed set of Resources representing fixed functions.
- 2056 – A Device local Collection where the Collection is used only on the Device that hosts the
2057 Collection. Such Collections may be used as a short-hand on a Client for referring to many
2058 Servers as one.
- 2059 – A centralized Collection where the Collection is hosted on a Device but other Devices may
2060 access or update the Collection.
- 2061 – A hosted Collection where the Collection is centralized but is managed by an authorized agent
2062 or party.

2063 7.8.3.2 Collection Properties

2064 A Collection shall define a Property that is an array of Links (the Property Name "links" is
2065 recommended). In addition, other Properties may be defined for the Collection by the Resource
2066 Type. The mandatory and recommended Common Properties for a Collection are shown in Table 13.
2067 This list of Common Properties is in addition to those defined for Resources in 7.3.2.

2068 **Table 13 – Common Properties for Collections (in addition to Common Properties defined**
2069 **in 7.3.2)**

Property	Description	Property Name	Value Type	Mandatory
Links	The array of Links in the Collection	Per Resource Type definition	json Array of Links	Yes
Resource Types	The list of allowed Resource Types for Links in the Collection. If this Property is not defined or is null string then any Resource Type is permitted	As defined in Table 11	As defined in Table 11	No
Mandatory Resource Types	The list of Resource Types for Links that are mandatory in the Collection.	As defined in Table 12	As defined in Table 12	No

2070

2071 **7.8.3.3 Default Resource Type**

2072 A default Resource Type, "oic.wk.col", is available for Collections. This Resource Type shall be
 2073 used only when another type has not been defined on the Collection or when no Resource Type
 2074 has been specified at the creation of the Collection.

2075 The default Resource Type provides support for the Common Properties including an array of Links
 2076 with the Property Name "links".

2077 **7.8.3.4 Default OCF Interface**

2078 All instances of a Collection shall support the links list ("oic.if.ll") OCF Interface in addition to the
 2079 baseline ("oic.if.baseline") OCF Interface. An instance of a Collection may optionally support
 2080 additional OCF Interfaces that are defined within this document. The Default OCF Interface for a
 2081 Collection shall be links list ("oic.if.ll") unless otherwise specified by the Resource Type definition.

2082 **7.8.4 Atomic Measurement**

2083 **7.8.4.1 Overview**

2084 Certain use cases require that the Properties of multiple Resources are only accessible as a group
 2085 and individual access to those Properties of each Resource by a Client is prohibited. The Atomic
 2086 Measurement Resource Type is defined to meet this requirement. This is accomplished through
 2087 the use of the Batch OCF Interface.

2088 **7.8.4.2 Atomic Measurement Properties**

2089 An Atomic Measurement shall define a Property that is an array of Links (the Property Name "links"
 2090 is recommended). In addition, other Properties may be defined for the Atomic Measurement by the
 2091 Resource Type. The mandatory and recommended Common Properties for an Atomic
 2092 Measurement are shown in Table 14. This list of Common Properties is in addition to those defined
 2093 for Resources in 7.3.2.

2094 **Table 14 – Common Properties for Atomic Measurement (in addition to Common Properties**
 2095 **defined in 7.3.2)**

Property	Description	Property Name	Value Type	Mandatory
Links	The array of Links in the Atomic Measurement	Per Resource Type definition	json Array of Links	Yes
Resource Types	The list of allowed Resource Types for Links in the Atomic Measurement. If this Property is not defined or is null string then any Resource Type is permitted	As defined in Table 11	As defined in Table 11	No
Mandatory Resource Types	The list of Resource Types for Links that are mandatory in the Atomic Measurement.	As defined in Table 12	As defined in Table 12	No

2096

2097 **7.8.4.3 Normative behaviour**

2098 The normative behaviour of an Atomic Measurement is as follows:

- 2099 – The behaviour of the Batch OCF Interface ("oic.if.b") on the Atomic Measurement is defined as
2100 follows:
- 2101 – Only RETRIEVE and NOTIFY operations are supported, for Batch OCF Interface, on Atomic
2102 Measurement; the behavior of the RETRIEVE and NOTIFY operations shall be the same as
2103 specified in 7.6.3.4, with exceptions as provided for in 7.8.4.3.
- 2104 – The UPDATE operation is not allowed, for Batch OCF Interface, on Atomic Measurement; if
2105 an UPDATE operation is received, it shall result in a method not allowed error code.
- 2106 – An error response shall not include any representation of a linked Resource (i.e. empty
2107 response for all linked Resources).
- 2108 – Any linked Resource within an Atomic Measurement (i.e. the target Resource of a Link in an
2109 Atomic Measurement) is subject to the following conditions:
- 2110 – Linked Resources within an Atomic Measurement and the Atomic Measurement itself shall
2111 exist on a single Server.
- 2112 – CRUDN operations shall not be allowed on linked Resources and shall result in a forbidden
2113 error code.
- 2114 – Linked Resources shall not expose the "oic.if.ll" OCF Interface. Since CRUDN operations
2115 are not allowed on linked Resources, the "oic.if.ll" OCF Interface would never be accessible.
- 2116 – Links to linked Resources in an Atomic Measurement shall only be accessible through the
2117 "oic.if.ll" or the "oic.if.baseline" OCF Interfaces of an Atomic Measurement.
- 2118 – The linked Resources shall not be listed in "/oic/res".
- 2119 – A linked Resource in an Atomic Measurement shall have defined one of "oic.if.a", "oic.if.s",
2120 "oic.if.r", or "oic.if.rw" as its Default OCF Interface.
- 2121 – Not all linked Resources in an Atomic Measurement are required to be Observable. If an Atomic
2122 Measurement is being Observed using the "oic.if.b" OCF Interface, notification responses shall
2123 not be generated when the linked Resources which are not marked Observable are updated or
2124 change state.
- 2125 – All linked Resources in an Atomic Measurement shall be included in every RETRIEVE and
2126 Observe response when using the "oic.if.b" OCF Interface.
- 2127 – An Atomic Measurement shall support the "oic.if.b" and the "oic.if.ll" OCF Interfaces.
- 2128 – Filtering of linked Resources in an Atomic Measurement is not allowed. Query parameters that
2129 select one or more individual linked Resources in a request to an Atomic Measurement shall
2130 result in a "forbidden" error code.
- 2131 – If the "rel" Link Parameter is included in a Link contained in an Atomic Measurement, it shall
2132 have either the "hosts" or the "item" value.
- 2133 – The Default OCF Interface of an Atomic Measurement is "oic.if.b".

2134 **7.8.4.4 Security considerations**

2135 Access rights to an Atomic Measurement Resource Type is as specified in clause 12.2.7.2 (ACL
2136 considerations for batch request to the Atomic Measurement Resource Type) of ISO/IEC 30118-
2137 2:2018).

2138 **7.8.4.5 Default Resource Type**

2139 The Resource Type is defined as "oic.wk.atomicmeasurement" as defined in Table 15.

2140

Table 15 – Atomic Measurement Resource Type

Pre-defined URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction	M/CR/O
none	Atomic Measurement	"oic.wk.atomicmeasurement"	"oic.if.ll" "oic.if.baseline" "oic.if.b"	A specialisation of the Collection pattern to ensure atomic RETRIEVAL of its referred Resources	RETRIEVE, NOTIFY	O

2141

2142 The Properties for Atomic Measurement are as defined in Table 16.

2143 **Table 16 – Properties for Atomic Measurement (in addition to Common Properties defined**
2144 **in 7.3.2)**

Property	Description	Property name	Value Type	Mandatory
Links	The set of links that point to the linked Resources	Per Resource Type definition	json Array of Links	Yes

2145

2146 **7.9 Third (3rd) party specified extensions**

2147 This clause describes how a 3rd party may add Device Types, Resource Types, 3rd party defined
2148 Properties to an existing or 3rd party defined Resource Type, 3rd party defined enumeration values
2149 to an existing enumeration and 3rd party defined Parameters to an existing defined Property.

2150 A 3rd party may specify additional (non-OCF) Resources within an OCF Device. A 3rd party may
2151 also specify additional Properties within an existing OCF defined Resource Type. Further a 3rd
2152 party may extend an OCF defined enumeration with 3rd party defined values.

2153 A 3rd party defined Device Type may expose both 3rd party and OCF defined Resource Types. A
2154 3rd party defined Device Type must expose the mandatory Resources for all OCF Devices defined
2155 within this document.

2156 A 3rd party defined Resource Type shall include any mandatory Properties defined in this
2157 document and also any vertical specified mandatory Properties. All Properties defined within a 3rd
2158 party defined Resource Type that are part of the OCF namespace that are not Common Properties
2159 as defined in this document shall follow the 3rd party defined Property rules in Table 17.

2160 The following table defines the syntax rules for 3rd party defined Resource Type elements. Within
2161 the table the term "Domain_Name" refers to a domain name that is owned by the 3rd party that is
2162 defining the new element.

2163

Table 17 – 3rd party defined Resource elements

	Resource Element	Vendor Definition Rules
New 3 rd party defined Device Type	"rt" Property Value of "/oic/d"	"x.<Domain_Name>.<Resource identification>"
New 3 rd party defined Resource Type	"rt" Property Value	"x.<Domain_Name>.<Resource identification>"
New 3 rd party defined Property within the OCF namespace	Property Name	"x.<Domain_Name>.<Property>"

Additional 3 rd party defined values in an OCF specified enumeration	Enumeration Property Value	"x.<Domain_Name>.<enum value>"
Additional 3 rd party defined Parameter in an OCF specified Property	Parameter key word	x.<Domain_Name>.<parameter keyword>

2164

2165 With respect to the use of the Domain_Name in this scheme the labels are reversed from how they
 2166 appear in DNS or other resolution mechanisms. The 3rd party defined Device Type and Resource
 2167 Type otherwise follow the rules defined in clause 7.4.2. 3rd party defined Resource Types should
 2168 be registered in the IANA Constrained RESTful Environments (CoRE) Parameters registry.

2169 For example:

2170 x.com.samsung.galaxyphone.accelerator
 2171 x.com.cisco.ciscorouterport
 2172 x.com.hp.printerhead
 2173 x.org.allseen.newinterface.newproperty

2174 7.10 Query Parameters

2175 7.10.1 Introduction

2176 Properties and Parameters (including those that are part of a Link) may be used in the query part
 2177 of a URI (see 6.2.2) as one criterion for selection of a particular Resource. This is done by declaring
 2178 the Property (i.e. <Property Name> = <desired Property Value>) as one of the segments of the
 2179 query. Only ASCII strings are permitted in query filters, and NULL characters are disallowed in
 2180 query filters. This means that only Property Values with ASCII characters may be matched in a
 2181 query filter.

2182 The Resource is selected when all the declared Properties or Link Parameters in the query match
 2183 the corresponding Properties or Link Parameters in the target.

2184 7.10.2 Use of multiple parameters within a query

2185 When a query contains multiple separate query parameters these are delimited by an "&" as
 2186 described in 6.2.2.

2187 A Client may apply multiple separate query parameters, for
 2188 example "?ins=11111&rt=oc.r.switch.binary". If such queries are supported by the Server this shall
 2189 be accomplished by matching "all of" the different query parameter types ("rt", "ins", "if", etc)
 2190 against the target of the query. In the example, this resolves to an instance of oc.r.switch.binary
 2191 that also has an "ins" populated as "11111". There is no significance applied to the order of the
 2192 query parameters.

2193 A Client may select more than one Resource Type using repeated query parameters, for example
 2194 "?rt=oc.r.switch.binary&rt=oc.r.ramptime". If such queries are supported by the Server this shall
 2195 be accomplished by matching "any of" the repeated query parameters against the target of the
 2196 query. In the example, any instances of "oc.r.switch.binary" and/or "oc.r.ramptime" that may exist
 2197 are selected.

2198 A Client may combine both multiple repeated parameters and multiple separate parameters in a
 2199 single query, for example "?if=oc.if.b&ins=11111&rt=oc.r.switch.binary&rt=oc.r.ramptime". If
 2200 such queries are supported by the Server this shall be accomplished by matching "any of" the
 2201 repeated query parameters and then matching "all of" the different query parameter types. In the
 2202 example any instances of "oc.r.switch.binary" and/or "oc.r.ramptime" that also have an "ins" of
 2203 "11111" that may exist are selected in a batch response.

2204 NOTE The parameters within a query string are represented within the actual messaging protocol as defined in clause
 2205 11.9.

2206 **7.10.3 Application to multi-value "rt" Resources**

2207 An "rt" query for a multi-value "rt" Resource with the Default OCF Interface of "oic.if.a", "oic.if.s",
2208 "oic.if.r", "oic.if.rw" or "oic.if.baseline" is an extension of a generic "rt" query. When a Server
2209 receives a RETRIEVE request for a multi-value "rt" Resource with an "rt" query, (i.e. GET
2210 /ResExample?rt=oic.r.foo), the Server should respond only when the query value is an item of the
2211 "rt" Property Value of the target Resource and should send back only the Properties associated
2212 with the query value(s). For example, upon receiving GET /ResExample?rt=oic.r.switch.binary
2213 targeting a Resource with "rt": ["oic.r.switch.binary", "oic.r.light.brightness"], the Server responds
2214 with only the Properties of oic.r.switch.binary.

2215 **7.10.4 OCF Interface specific considerations for queries**

2216 **7.10.4.1 OCF Interface selection**

2217 When an OCF Interface is to be selected for a request, it shall be specified as a query parameter
2218 in the URI of the Resource in the request message. If no query parameter is specified, then the
2219 Default OCF Interface shall be used. If the selected OCF Interface is not one of the permitted OCF
2220 Interfaces on the Resource then selecting that OCF Interface is an error and the Server shall
2221 respond with an error response code.

2222 For example, the baseline OCF Interface may be selected by adding "if=oic.if.baseline" to the list
2223 of query parameters in the URI of the target Resource. For example: "GET
2224 /oic/res?if=oic.if.baseline".

2225 **7.10.4.2 Batch OCF Interface**

2226 See 7.6.3.4 for details on the batch OCF Interface itself. Query parameters may be used with the
2227 batch OCF Interface in order to select particular Resources in a Collection for retrieval or update;
2228 these parameters are used to select items in the Collection by matching Link Parameter Values.

2229 When Link selection query parameters are used with RETRIEVE operations applied using the batch
2230 OCF Interface, only the Resources in the Collection with matching Link Parameters should be
2231 returned.

2232 When Link selection query parameters are used with UPDATE operations applied using the batch
2233 OCF Interface, only the Resources having matching Link Parameters should be updated.

2234 See 7.6.3.4.2 for examples of RETRIEVE and UPDATE operations that use Link selection query
2235 parameters.

2236 **8 CRUDN**

2237 **8.1 Overview**

2238 CREATE, RETRIEVE, UPDATE, DELETE, and NOTIFY (CRUDN) are operations defined for
2239 manipulating Resources. These operations are performed by a Client on the Resources contained
2240 in n Server.

2241 On reception of a valid CRUDN operation a Server hosting the Resource that is the target of the
2242 request shall generate a response depending on the OCF Interface included in the request; or
2243 based on the Default OCF Interface for the Resource Type if no OCF Interface is included.

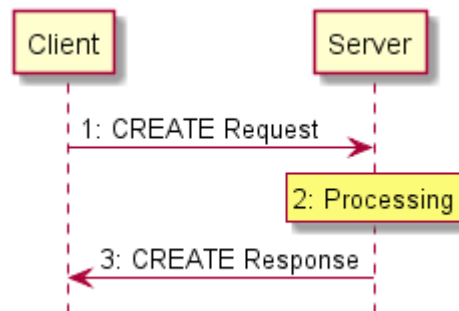
2244 CRUDN operations utilize a set of parameters that are carried in the messages and are defined in
2245 Table 18. A Device shall use CBOR as the default payload (content) encoding scheme for Resource
2246 representations included in CRUDN operations and operation responses; a Device may negotiate
2247 a different payload encoding scheme (e.g, see in 12.2.4 for CoAP messaging). Clauses 8.2 through
2248 8.6 respectively specify the CRUDN operations and use of the parameters. The type definitions for
2249 these terms will be mapped in the clause 12 for each protocol.

Table 18 – Parameters of CRUDN messages

Applicability	Name	Denotation	Definition
All messages	<i>fr</i>	From	The URI of the message originator.
	<i>to</i>	To	The URI of the recipient of the message.
	<i>ri</i>	Request Identifier	The identifier that uniquely identifies the message in the originator and the recipient.
	<i>cn</i>	Content	Information specific to the operation.
Requests	<i>op</i>	Operation	Specific operation requested to be performed by the Server.
	<i>obs</i>	Observe	Indicator for an Observe request.
Responses	<i>rs</i>	Response Code	Indicator of the result of the request; whether it was accepted and what the conclusion of the operation was. The values of the response code for CRUDN operations shall conform to those as defined in clause 5.9 and 12.1.2 in IETF RFC 7252.
	<i>obs</i>	Observe	Indicator for an Observe response.

2251 **8.2 CREATE**2252 **8.2.1 Overview**

2253 The CREATE operation is used to request the creation of new Resources on the Server. The
 2254 CREATE operation is initiated by the Client and consists of three steps, as depicted in Figure 10.



2255

2256

Figure 10 – CREATE operation2257 **8.2.2 CREATE request**

2258 The CREATE request message is transmitted by the Client to the Server to create a new Resource
 2259 by the Server. The CREATE request message will carry the following parameters:

- 2260 – *fr*: Unique identifier of the Client
- 2261 – *to*: URI of the target Resource responsible for creation of the new Resource.
- 2262 – *ri*: Identifier of the CREATE request.
- 2263 – *cn*: Information of the Resource to be created by the Server.
- 2264 – *cn* will include the URI and Resource Type Property of the Resource to be created.
- 2265 – *cn* may include additional Properties of the Resource to be created.
- 2266 – *op*: CREATE

2267 **8.2.3 Processing by the Server**

2268 Following the receipt of a CREATE request, the Server may validate if the Client has the
2269 appropriate rights for creating the requested Resource. If the validation is successful, the Server
2270 creates the requested Resource. The Server caches the value of *ri* parameter in the CREATE
2271 request for inclusion in the CREATE response message.

2272 **8.2.4 CREATE response**

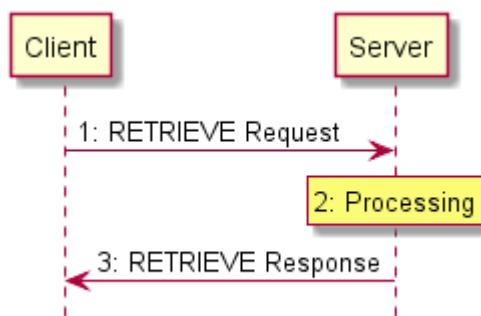
2273 The Server shall transmit a CREATE response message in response to a CREATE request
2274 message from a Client. The CREATE response message will include the following parameters.

- 2275 – *fr*: Unique identifier of the Server
- 2276 – *to*: Unique identifier of the Client
- 2277 – *ri*: Identifier included in the CREATE request
- 2278 – *cn*: Information of the Resource as created by the Server.
 - 2279 – *cn* will include the URI of the created Resource.
 - 2280 – *cn* will include the Resource representation of the created Resource.
- 2281 – *rs*: The result of the CREATE operation.

2282 **8.3 RETRIEVE**

2283 **8.3.1 Overview**

2284 The RETRIEVE operation is used to request the current state or representation of a Resource. The
2285 RETRIEVE operation is initiated by the Client and consists of three steps, as depicted in Figure 11.



2286

2287

Figure 11 – RETRIEVE operation

2288 **8.3.2 RETRIEVE request**

2289 RETRIEVE request message is transmitted by the Client to the Server to request the representation
2290 of a Resource from a Server. The RETRIEVE request message will carry the following parameters.

- 2291 – *fr*: Unique identifier of the Client.
- 2292 – *to*: URI of the Resource the Client is targeting.
- 2293 – *ri*: Identifier of the RETRIEVE request.
- 2294 – *op*: RETRIEVE.

2295 **8.3.3 Processing by the Server**

2296 Following the receipt of a RETRIEVE request, the Server may validate if the Client has the
2297 appropriate rights for retrieving the requested data and the Properties are readable. The Server
2298 caches the value of *ri* parameter in the RETRIEVE request for use in the response

2299 **8.3.4 RETRIEVE response**

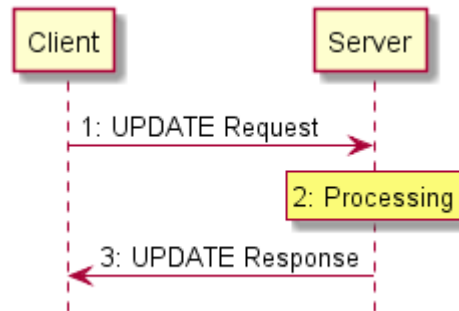
2300 The Server shall transmit a RETRIEVE response message in response to a RETRIEVE request
2301 message from a Client. The RETRIEVE response message will include the following parameters.

- 2302 – *fr*: Unique identifier of the Server.
- 2303 – *to*: Unique identifier of the Client.
- 2304 – *ri*: Identifier included in the RETRIEVE request.
- 2305 – *cn*: Information of the Resource as requested by the Client.
- 2306 – *cn* should include the URI of the Resource targeted in the RETRIEVE request.
- 2307 – *rs*: The result of the RETRIEVE operation.

2308 **8.4 UPDATE**

2309 **8.4.1 Overview**

2310 The UPDATE operation is either a Partial UPDATE or a complete replacement of the information
2311 in a Resource in conjunction with the OCF Interface that is also applied to the operation. The
2312 UPDATE operation is initiated by the Client and consists of three steps, as depicted in Figure 12.



2313

2314

Figure 12 – UPDATE operation

2315 **8.4.2 UPDATE request**

2316 The UPDATE request message is transmitted by the Client to the Server to request the update of
2317 information of a Resource on the Server. The UPDATE request message will carry the following
2318 parameters.

- 2319 – *fr*: Unique identifier of the Client.
- 2320 – *to*: URI of the Resource targeted for the information update.
- 2321 – *ri*: Identifier of the UPDATE request.
- 2322 – *op*: UPDATE.
- 2323 – *cn*: Information, including Properties, of the Resource to be updated at the target Resource.

2324 **8.4.3 Processing by the Server**

2325 **8.4.3.1 Overview**

2326 Following the receipt of an UPDATE request, the Server may validate if the Client has the
2327 appropriate rights for updating the requested data. If the validation is successful the Server updates
2328 the target Resource information according to the information carried in *cn* parameter of the
2329 UPDATE request message. The Server caches the value of *ri* parameter in the UPDATE request
2330 for use in the response.

2331 An UPDATE request that includes Properties that are read-only shall be rejected by the Server with
2332 an *rs* indicating a bad request.

2333 An UPDATE request shall be applied only to the Properties in the target Resource visible via the
2334 applied OCF Interface that support the operation. An UPDATE of non-existent Properties is ignored.

2335 An UPDATE request shall be applied to the Properties in the target Resource even if those Property
2336 Values are the same as the values currently exposed by the target Resource.

2337 **8.4.3.2 Resource monitoring by the Server**

2338 The Server shall monitor the state the Resource identified in the Observe request from the Client.
2339 Anytime there is a change in the state of the Observed Resource or an UPDATE operation applied
2340 to the Resource, the Server sends another RETRIEVE response with the Observe indication. The
2341 mechanism does not allow the Client to specify any bounds or limits which trigger a notification,
2342 the decision is left entirely to the Server.

2343 **8.4.3.3 Additional RETRIEVE responses with Observe indication**

2344 The Server shall transmit updated RETRIEVE response messages following Observed changes in
2345 the state of the Resources requested by the Client. The RETRIEVE response message shall include
2346 the parameters listed in 11.4.2.4.

2347 **8.4.4 UPDATE response**

2348 The UPDATE response message will include the following parameters:

- 2349 – *fr*: Unique identifier of the Server.
- 2350 – *to*: Unique identifier of the Client.
- 2351 – *ri*: Identifier included in the UPDATE request.
- 2352 – *rs*: The result of the UPDATE request.

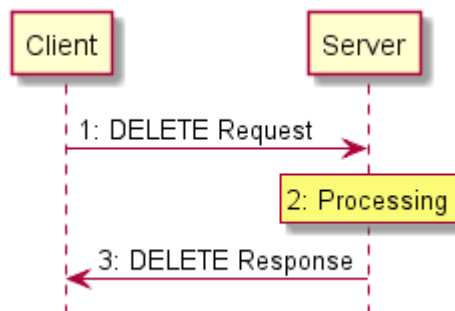
2353 The UPDATE response message may also include the following parameters:

- 2354 – *cn*: The Resource representation following processing of the UPDATE request.

2355 **8.5 DELETE**

2356 **8.5.1 Overview**

2357 The DELETE operation is used to request the removal of a Resource. The DELETE operation is
2358 initiated by the Client and consists of three steps, as depicted in Figure 13.



2359

2360

Figure 13 – DELETE operation

2361 **8.5.2 DELETE request**

2362 DELETE request message is transmitted by the Client to the Server to delete a Resource on the
2363 Server. The DELETE request message will carry the following parameters:

- 2364 – *fr*: Unique identifier of the Client.
- 2365 – *to*: URI of the target Resource which is the target of deletion.
- 2366 – *ri*: Identifier of the DELETE request.
- 2367 – *op*: DELETE.

2368 **8.5.3 Processing by the Server**

2369 Following the receipt of a DELETE request, the Server may validate if the Client has the appropriate
2370 rights for deleting the identified Resource, and whether the identified Resource exists. If the
2371 validation is successful, the Server removes the requested Resource and deletes all the associated
2372 information. The Server caches the value of *ri* parameter in the DELETE request for use in the
2373 response.

2374 **8.5.4 DELETE response**

2375 The Server shall transmit a DELETE response message in response to a DELETE request message
2376 from a Client. The DELETE response message will include the following parameters.

- 2377 – *fr*: Unique identifier of the Server.
- 2378 – *to*: Unique identifier of the Client.
- 2379 – *ri*: Identifier included in the DELETE request.
- 2380 – *rs*: The result of the DELETE operation.

2381 **8.6 NOTIFY**

2382 **8.6.1 Overview**

2383 The NOTIFY operation is used to request asynchronous notification of state changes. Complete
2384 description of the NOTIFY operation is provided in 11.4. The NOTIFY operation uses the
2385 NOTIFICATION response message which is defined here.

2386 **8.6.2 NOTIFICATION response**

2387 The NOTIFICATION response message is sent by a Server to notify the URLs identified by the
2388 Client of a state change. The NOTIFICATION response message carries the following parameters.

- 2389 – *fr*: Unique identifier of the Server.
- 2390 – *to*: URI of the Resource target of the NOTIFICATION message.
- 2391 – *ri*: Identifier included in the CREATE request.
- 2392 – *op*: NOTIFY.
- 2393 – *cn*: The updated state of the Resource.

2394 **9 Network and connectivity**

2395 **9.1 Introduction**

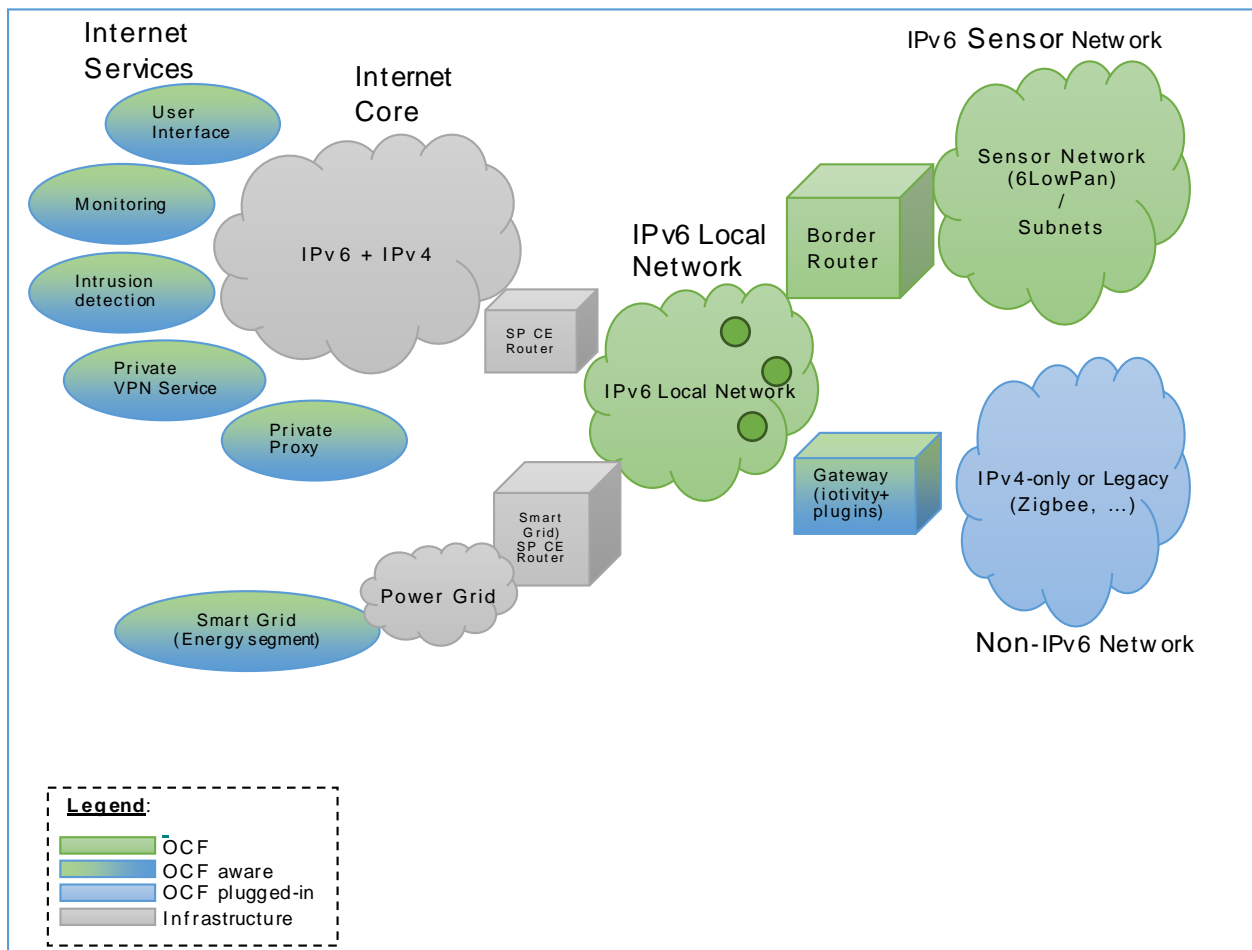
2396 The Internet of Things is comprised of a wide range of applications which sense and actuate the
2397 physical world with a broad spectrum of device and network capabilities: from battery powered
2398 nodes transmitting 100 bytes per day and able to last 10 years on a coin cell battery, to mains
2399 powered nodes able to maintain Megabit video streams. It is estimated that many 10s of billions of
2400 IoT devices will be deployed over the coming years.

2401 It is desirable that the connectivity options be adapted to the IP layer. To that end, IETF has
2402 completed considerable work to adapt Bluetooth®, Wi-Fi, 802.15.4, LPWAN, etc. to IPv6. These
2403 adaptations, plus the larger address space and improved address management capabilities, make
2404 IPv6 the clear choice for the OCF network layer technology.

2405 **9.2 Architecture**

2406 While the aging IPv4 centric network has evolved to support complex topologies, its deployment
 2407 was primarily provisioned by a single Internet Service Provider (ISP) as a single network. More
 2408 complex network topologies, often seen in residential home, are mostly introduced through the
 2409 acquisition of additional home network devices, which rely on technologies like private Network
 2410 Address Translation (NAT). These technologies require expert assistance to set up correctly and
 2411 should be avoided in a home network as they most often result in breakage of constructs like
 2412 routing, naming and discovery services.

2413 The multi-segment ecosystem OCF addresses will not only cause a proliferation of new devices
 2414 and associated routers, but also new services introducing additional edge routers. All these new
 2415 requirements require advance architectural constructs to address complex network topologies like
 2416 the one shown in Figure 14.



2417

2418 **Figure 14 – High Level Network & Connectivity Architecture**

2419 In terms of IETF RFC 6434, IPv6 nodes assume either a router or host role. Nodes may further
 2420 implement various specializations of those roles:

- 2421 – A Router may implement Customer Edge Router capabilities as defined in IETF RFC 7084.
- 2422 – Nodes limited in processing power, memory, non-volatile storage or transmission capacity
 2423 requires special IP adaptation layers (6LoWPAN) and/or dedicated routing protocols (RPL).
 2424 Examples include devices transmitting over low power physical layer like IEEE 802.14.5, ITU
 2425 G9959, Bluetooth Low Energy, DECT Ultra Low Energy, and Near Field Communication (NFC).

2426 – A node may translate and route messaging between IPv6 and non-IPv6 networks.

2427 **9.3 IPv6 network layer requirements**

2428 **9.3.1 Introduction**

2429 Projections indicate that many 10s of billions of new IoT endpoints and related services will be
2430 brought online in the next few years. These endpoint's capabilities will span from battery powered
2431 nodes with limited compute, storage, and bandwidth to more richly resourced devices operating
2432 over Ethernet and WiFi links.

2433 Internet Protocol version 4 (IPv4), deployed some 30 years ago, has matured to support a wide
2434 variety of applications such as Web browsing, email, voice, video, and critical system monitoring
2435 and control. However, the capabilities of IPv4 are at the point of exhaustion, not the least of which
2436 is that available address space has been consumed.

2437 The IETF long ago saw the need for a successor to IPv4, thus the development of IPv6. OCF
2438 recommends IPv6 at the network layer. Amongst the reasons for IPv6 recommendations are:

- 2439 – Larger address space. Side-effect: greatly reduce the need for NATs.
- 2440 – More flexible addressing architecture. Multiple addresses and types per interface: Link-local,
2441 ULA, GUA, variously scoped Multicast addresses, etc. Better ability to support multi-homed
2442 networks, better re-numbering capability, etc.
- 2443 – More capable auto configuration capabilities: DHCPv6, SLAAC, Router Discovery, etc.
- 2444 – Technologies enabling IP connectivity on constrained nodes are based upon IPv6.
- 2445 – All major consumer operating systems (iOS, Android, Windows, Linux) are already IPv6 enabled.
- 2446 – Major Service Providers around the globe are deploying IPv6.

2447 **9.3.2 IPv6 node requirements**

2448 **9.3.2.1 Introduction**

2449 In order to ensure network layer services interoperability from node to node, mandating a common
2450 network layer across all nodes is vital. The protocol should enable the network to be: secure,
2451 manageable, and scalable and to include constrained and self-organizing meshed nodes. OCF
2452 mandates IPv6 as the common network layer protocol to ensure interoperability across all Devices.
2453 More capable Devices may also include additional protocols creating multiple-stack Devices. The
2454 remainder of this clause will focus on interoperability requirements for IPv6 hosts, IPv6 constrained
2455 hosts and IPv6 routers. The various protocol translation permutations included in multi-stack
2456 gateway devices may be addresses in subsequent addendums of this document.

2457 **9.3.2.2 IP Layer**

2458 An IPv6 node shall support IPv6 and it shall conform to the requirements as specified in
2459 IETF RFC 6434.

2460 **10 OCF Endpoint**

2461 **10.1 OCF Endpoint definition**

2462 The specific definition of an OCF Endpoint depends on the Transport Protocol Suite being used.
2463 For the example of CoAP over UDP over IPv6, the OCF Endpoint is identified by an IPv6 address
2464 and UDP port number.

2465 Each Device shall associate with at least one OCF Endpoint with which it can exchange request
2466 and response messages. When a message is sent to an OCF Endpoint, it shall be delivered to the
2467 Device which is associated with the OCF Endpoint. When a request message is delivered to an
2468 OCF Endpoint, path component is enough to locate the target Resource.

2469 A Device can be associated with multiple OCF Endpoints. For example, an Device can have several
2470 IP addresses or port numbers or support both CoAP and HTTP transfer protocol. Different
2471 Resources in an Device may be accessed with the same OCF Endpoint or need different ones.
2472 Some Resources may use one OCF Endpoint and others a different one. It depends on an
2473 implementation.

2474 On the other hand, an OCF Endpoint can be shared among multiple Devices, only when there is a
2475 way to clearly designate the target Resource with request URI. For example, when multiple CoAP
2476 servers use uniquely different URI paths for all their hosted Resources, and the CoAP
2477 implementation demultiplexes by path, they can share the same CoAP OCF Endpoint. However,
2478 this is not possible in this version of the document, because a pre-determined URI (e.g. "/oic/d") is
2479 mandatory for some mandatory Resources (e.g. "oic.wk.d").

2480 **10.2 OCF Endpoint information**

2481 **10.2.1 Introduction**

2482 OCF Endpoint is represented by OCF Endpoint information which consists of two items of key-
2483 value pair, "ep" and "pri".

2484 **10.2.2 "ep"**

2485 "ep" represents Transport Protocol Suite and OCF Endpoint Locator specified as follows:

- 2486 – *Transport Protocol Suite* - a combination of protocols (e.g. CoAP + UDP + IPv6) with which
2487 request and response messages can be exchanged for RESTful transaction (i.e. CRUDN). A
2488 Transport Protocol Suite shall be indicated by a URI scheme name. All scheme names
2489 supported by this document are IANA registered, these are listed in Table 19. A vendor may
2490 also make use of a non-IANA registered scheme name for their own use (e.g.
2491 "com.example.foo"), this shall follow the syntax for such scheme names defined by
2492 IETF RFC 7595. The behaviour of a vendor-defined scheme name is undefined by this
2493 document. All OCF defined Resource Types when exposing OCF Endpoint Information in an
2494 "eps" (see 10.2.4) shall include at least one "ep" with a Transport Protocol Suite as defined in
2495 Table 19.
- 2496 – *OCF Endpoint Locator* – an address (e.g. IPv6 address + Port number) or an indirect identifier
2497 (e.g., DNS name) resolvable to an IP address, through which a message can be sent to the
2498 OCF Endpoint and in turn associated Device. The OCF Endpoint Locator for "coap" and "coaps"
2499 shall be specified as "IP address: port number". The OCF Endpoint Locator for "coap+tcp" or
2500 "coaps+tcp" shall be specified as "IP address: port number" or "DNS name: port number" or
2501 "DNS name" such that the DNS name shall be resolved to a valid IP address for the target
2502 Resource with a name resolution service (i.e., DNS). For the 3rd case, when the port number
2503 is omitted, the default port "5683" (and "5684") shall be assumed for "coap+tcp" (and for
2504 "coaps+tcp") scheme respectively as defined in IETF RFC 8323. Temporary addresses should
2505 not be used because OCF Endpoint Locators are for the purpose of accepting incoming
2506 sessions, whereas temporary addresses are for initiating outgoing sessions (IETF RFC 4941).
2507 Moreover, its inclusion in "/oic/res" can cause a privacy concern (IETF RFC 7721).

2508 "ep" shall have as its value a URI (as specified in IETF RFC 3986) with the scheme component
2509 indicating Transport Protocol Suite and the authority component indicating the OCF Endpoint
2510 Locator.

2511 An "ep" example for "coap" and "coaps" is as illustrated:

```
"ep": "coap://[fe80::b1d6]:1111"
```

2512 An "ep" example for "coap+tcp" and "coaps+tcp" is as illustrated:

```
"ep": "coap+tcp://[2001:db8:a::123]:2222"
"ep": "coap+tcp://foo.bar.com:2222"
"ep": "coap+tcp://foo.bar.com"
```

2513 The current list of "ep" with corresponding Transport Protocol Suite is shown in Table 19:

2514 **Table 19 – "ep" value for Transport Protocol Suite**

Transport Protocol Suite	scheme	OCF Endpoint Locator	"ep" Value example
coap+udp+ip	"coap"	IP address + port number	"coap://[fe80::b1d6]:1111"
coaps + udp + ip	"coaps"	IP address + port number	"coaps://[fe80::b1d6]:1122"
coap + tcp + ip	"coap+tcp"	IP address + port number DNS name: port number DNS name	"coap+tcp://[2001:db8:a::123]:2222" "coap+tcp://foo.bar.com:2222" "coap+tcp://foo.bar.com"
coaps + tcp + ip	"coaps+tcp"	IP address + port number DNS name: port number DNS name	"coaps+tcp://[2001:db8:a::123]:2233" "coaps+tcp://[2001:db8:a::123]:2233" "coaps+tcp://foo.bar.com:2233"

2515

2516 **10.2.3 "pri"**

2517 When there are multiple OCF Endpoints, "pri" indicates the priority among them.

2518 "pri" shall be represented as a positive integer (e.g. "pri": 1) and the lower the value, the higher the
2519 priority.

2520 The default "pri" value is 1, i.e. when "pri" is not present, it shall be equivalent to "pri": 1.

2521 **10.2.4 OCF Endpoint information in "eps" Parameter**

2522 To carry OCF Endpoint information, a new Link Parameter "eps" is defined in 7.8.2.2.5. "eps" has
2523 an array of items as its value and each item represents OCF Endpoint information with two key-
2524 value pairs, "ep" and "pri", of which "ep" is mandatory and "pri" is optional.

2525 OCF Endpoint Information in an "eps" Parameter is valid for the target Resource of the Link, i.e.,
2526 the Resource referred by "href" Parameter. OCF Endpoint information in an "eps" Parameter may
2527 be used to access other Resources on the Device, but such access is not guaranteed.

2528 A Client may resolve the "ep" value to an IP address for the target Resource, i.e., the address to
2529 access the Device which hosts the target Resource. A valid (transfer protocol) URI for the target
2530 Resource can be constructed with the scheme, host and port components from the "ep" value and
2531 the "path" component from the "href" value.

2532 Links with an "eps":

```
2533 {
2534   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9 ",
2535   "href": "/myLightSwitch",
2536   "rt": ["oic.r.switch.binary"],
2537   "if": ["oic.if.a", "oic.if.baseline"],
```

```

2538     "p": { "bm": 3 },
2539     "eps": [
2540       { "ep": "coap://[fe80::b1d6]:1111", "pri": 2 },
2541       { "ep": "coaps://[fe80::b1d6]:1122" }
2542     ]
2543   }
2544
2545   {
2546     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2547     "href": "/myTemperature",
2548     "rt": ["oic.r.temperature"],
2549     "if": ["oic.if.a", "oic.if.baseline"],
2550     "p": { "bm": 3 },
2551     "eps": [
2552       { "ep": "coap+tcp://foo.bar.com", "pri": 2 },
2553       { "ep": "coaps+tcp://foo.bar.com:1122" }
2554     ]
2555   }

```

2556 In the previous example, "anchor" represents the hosting Device, "href", target Resource and "eps"
 2557 the two OCF Endpoints for the target Resource. The (fully-qualified) URIs for the target Resource
 2558 are as illustrated:

```

2559 coap://[fe80::b1d6]:1111/myLightSwitch
2560 coaps://[fe80::b1d6]:1122/myLightSwitch
2561 coap+tcp://foo.bar.com:5683/myTemperature

```

2562 coaps+tcp://foo.bar.com:1122/myTemperature If the target Resource of a Link requires a secure
 2563 connection (e.g. CoAPS), "eps" Parameter shall be used to indicate the necessary information (e.g.
 2564 port number) in OCF 1.0 payload. For optional backward compatibility with OIC 1.1, the "sec" and
 2565 "port" shall only be used in OIC 1.1 payload.

2566 10.3 OCF Endpoint discovery

2567 10.3.1 Introduction

2568 OCF Endpoint discovery is defined as the process for a Client to acquire the OCF Endpoint
 2569 information for Device or Resource.

2570 10.3.2 Implicit discovery

2571 If a Device is the source of a CoAP message (e.g. "/oic/res" response), the source IP address and
 2572 port number may be combined to form the OCF Endpoint Locator for the Device. Along with a
 2573 "coap" scheme and default "pri" value, OCF Endpoint information for the Device may be constructed.

2574 In other words, a "/oic/res" response message with CoAP may implicitly carry the OCF Endpoint
 2575 information of the responding Device and in turn all the hosted Resources, which may be accessed
 2576 with the same transfer protocol of CoAP. In the absence of an "eps" Parameter, a Client shall be
 2577 able to utilize implicit discovery to access the target Resource.

2578 10.3.3 Explicit discovery with "/oic/res" response

2579 OCF Endpoint information may be explicitly indicated with the "eps" Parameter of the Links in
 2580 "/oic/res".

2581 As in 10.3.2, an "/oic/res" response may implicitly indicate the OCF Endpoint information for some
 2582 Resources hosted by the responding Device. However implicit discovery, i.e., inference of OCF
 2583 Endpoint information from CoAP response message, may not work for some Resources on the
 2584 same Device. For example, some Resources may allow only secure access via CoAPS which
 2585 requires the "eps" Parameter to indicate the port number. Moreover "/oic/res" may expose a target
 2586 Resource which belongs to another Device.

2587 When the OCF Endpoint for a target Resource of a Link cannot be implicitly inferred, the "eps"
2588 Parameter shall be included to provide explicit OCF Endpoint information with which a Client can
2589 access the target Resource. In the presence of the "eps" Parameter, a Client shall be able to utilize
2590 it to access the target Resource. For "coap" and "coaps", a Client may use the IP address in the
2591 "ep" value in the "eps" Parameter to access the target Resource. For "coap+tcp" and "coaps+tcp",
2592 a Client may use the IP address in the "eps" Parameter or resolve the DNS name in the "eps"
2593 Parameter to acquire a valid IP address for the target Resource. If "eps" Parameter omits the port
2594 number, then the default port "5683" (and "5684") shall be assumed for "coap+tcp" (and
2595 "coaps+tcp") scheme as defined in IETF RFC 8323. To access the target Resource of a Link, a
2596 Client may use the "eps" Parameter in the Link, if it is present and fall back on implicit discovery if
2597 not.

2598 This applies to the case of "/oic/res" for a Resource Directory or Bridge Device which usually carries
2599 the Links for Resources which another Device hosts.

2600 This is an example of an "/oic/res" response from a Bridge Device with two Bridged Devices, having
2601 the "eps" Parameter in Links.

```
2602 [
2603 {
2604   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2605   "href": "/oic/res",
2606   "rel": "self",
2607   "rt": ["oic.wk.res"],
2608   "if": ["oic.if.ll", "oic.if.baseline"],
2609   "p": {"bm": 3},
2610   "eps": [
2611     {"ep": "coap://[2001:db8:a::b1d4]:55555"},
2612     {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2613   ]
2614 },
2615 {
2616   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2617   "href": "/oic/d",
2618   "rt": ["oic.wk.d", "oic.d.bridge"],
2619   "if": ["oic.if.r", "oic.if.baseline"],
2620   "p": {"bm": 3},
2621   "eps": [
2622     {"ep": "coap://[2001:db8:a::b1d4]:55555"},
2623     {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2624   ]
2625 },
2626 {
2627   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2628   "href": "/oic/p",
2629   "rt": ["oic.wk.p"],
2630   "if": ["oic.if.r", "oic.if.baseline"],
2631   "p": {"bm": 3},
2632   "eps": [
2633     {"ep": "coap://[2001:db8:a::b1d4]:55555"},
2634     {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2635   ]
2636 },
2637 {
2638   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2639   "href": "/mySecureMode",
2640   "rt": ["oic.r.securemode"],
2641   "if": ["oic.if.rw", "oic.if.baseline"],
2642   "p": {"bm": 3},
2643   "eps": [
2644
```

```

2645     {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2646   ]
2647 },
2648 {
2649   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2650   "href": "/oic/sec/doxm",
2651   "rt": ["oic.r.doxm"],
2652   "if": ["oic.if.baseline"],
2653   "p": {"bm": 1},
2654   "eps": [
2655     {"ep": "coap://[2001:db8:a::b1d4]:55555"},
2656     {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2657   ]
2658 },
2659 {
2660   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2661   "href": "/oic/sec/pstat",
2662   "rt": ["oic.r.pstat"],
2663   "if": ["oic.if.baseline"],
2664   "p": {"bm": 1},
2665   "eps": [
2666     {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2667   ]
2668 },
2669 {
2670   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2671   "href": "/oic/sec/cred",
2672   "rt": ["oic.r.cred"],
2673   "if": ["oic.if.baseline"],
2674   "p": {"bm": 1},
2675   "eps": [
2676     {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2677   ]
2678 },
2679 {
2680   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2681   "href": "/oic/sec/acl2",
2682   "rt": ["oic.r.acl2"],
2683   "if": ["oic.if.baseline"],
2684   "p": {"bm": 1},
2685   "eps": [
2686     {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2687   ]
2688 },
2689 {
2690   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2691   "href": "/myIntrospection",
2692   "rt": ["oic.wk.introspection"],
2693   "if": ["oic.if.r", "oic.if.baseline"],
2694   "p": {"bm": 3},
2695   "eps": [
2696     {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2697   ]
2698 },
2699 {
2700   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2701   "href": "/oic/res",
2702   "rt": ["oic.wk.res"],
2703   "if": ["oic.if.ll", "oic.if.baseline"],
2704   "p": {"bm": 3},
2705   "eps": [
2706     {"ep": "coap://[2001:db8:a::b1d4]:66666"},
2707     {"ep": "coaps://[2001:db8:a::b1d4]:22222"}

```



```

2708     ]
2709   },
2710   {
2711     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2712     "href": "/oic/d",
2713     "rt": ["oic.wk.d", "oic.d.light", "oic.d.virtual"],
2714     "if": ["oic.if.r", "oic.if.baseline"],
2715     "p": {"bm": 3},
2716     "eps": [
2717       {"ep": "coap://[2001:db8:a::b1d4]:66666"},
2718       {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
2719     ]
2720   },
2721   {
2722     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2723     "href": "/oic/p",
2724     "rt": ["oic.wk.p"],
2725     "if": ["oic.if.r", "oic.if.baseline"],
2726     "p": {"bm": 3},
2727     "eps": [
2728       {"ep": "coap://[2001:db8:a::b1d4]:66666"},
2729       {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
2730     ]
2731   },
2732   {
2733     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2734     "href": "/myLight",
2735     "rt": ["oic.r.switch.binary"],
2736     "if": ["oic.if.a", "oic.if.baseline"],
2737     "p": {"bm": 3},
2738     "eps": [
2739       {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
2740     ]
2741   },
2742   {
2743     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2744     "href": "/oic/sec/doxm",
2745     "rt": ["oic.r.doxm"],
2746     "if": ["oic.if.baseline"],
2747     "p": {"bm": 1},
2748     "eps": [
2749       {"ep": "coap://[2001:db8:a::b1d4]:66666"},
2750       {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
2751     ]
2752   },
2753   {
2754     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2755     "href": "/oic/sec/pstat",
2756     "rt": ["oic.r.pstat"],
2757     "if": ["oic.if.baseline"],
2758     "p": {"bm": 1},
2759     "eps": [
2760       {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
2761     ]
2762   },
2763   {
2764     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2765     "href": "/oic/sec/cred",
2766     "rt": ["oic.r.cred"],
2767     "if": ["oic.if.baseline"],
2768     "p": {"bm": 1},
2769     "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:22222"}]
2770   },

```

```

2771 {
2772   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2773   "href": "/oic/sec/acl2",
2774   "rt": ["oic.r.acl2"],
2775   "if": ["oic.if.baseline"],
2776   "p": {"bm": 1},
2777   "eps": [
2778     {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
2779   ],
2780 },
2781 {
2782   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2783   "href": "/myLightIntrospection",
2784   "rt": ["oic.wk.introspection"],
2785   "if": ["oic.if.r", "oic.if.baseline"],
2786   "p": {"bm": 3},
2787   "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:22222"}]
2788 },
2789 {
2790   "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
2791   "href": "/oic/res",
2792   "rt": ["oic.wk.res"],
2793   "if": ["oic.if.ll", "oic.if.baseline"],
2794   "p": {"bm": 3},
2795   "eps": [
2796     {"ep": "coap://[2001:db8:a::b1d4]:77777"},
2797     {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2798   ],
2799 },
2800 {
2801   "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
2802   "href": "/oic/d",
2803   "rt": ["oic.wk.d", "oic.d.fan", "oic.d.virtual"],
2804   "if": ["oic.if.r", "oic.if.baseline"],
2805   "p": {"bm": 3},
2806   "eps": [
2807     {"ep": "coap://[2001:db8:a::b1d4]:77777"},
2808     {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2809   ],
2810 },
2811 {
2812   "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
2813   "href": "/oic/p",
2814   "rt": ["oic.wk.p"],
2815   "if": ["oic.if.r", "oic.if.baseline"],
2816   "p": {"bm": 3},
2817   "eps": [
2818     {"ep": "coap://[2001:db8:a::b1d4]:77777"},
2819     {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2820   ],
2821 },
2822 {
2823   "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
2824   "href": "/myFan",
2825   "rt": ["oic.r.switch.binary"],
2826   "if": ["oic.if.a", "oic.if.baseline"],
2827   "p": {"bm": 3},
2828   "eps": [
2829     {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2830   ],
2831 },
2832 {
2833   "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",

```

```

2834     "href": "/oic/sec/doxm",
2835     "rt": ["oic.r.doxm"],
2836     "if": ["oic.if.baseline"],
2837     "p": {"bm": 1},
2838     "eps": [
2839         {"ep": "coap://[2001:db8:a::b1d4]:77777"},
2840         {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2841     ],
2842 },
2843 {
2844     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
2845     "href": "/oic/sec/pstat",
2846     "rt": ["oic.r.pstat"],
2847     "if": ["oic.if.baseline"],
2848     "p": {"bm": 1},
2849     "eps": [
2850         {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2851     ]
2852 },
2853 {
2854     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
2855     "href": "/oic/sec/cred",
2856     "rt": ["oic.r.cred"],
2857     "if": ["oic.if.baseline"],
2858     "p": {"bm": 1},
2859     "eps": [
2860         {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2861     ]
2862 },
2863 {
2864     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
2865     "href": "/oic/sec/acl2",
2866     "rt": ["oic.r.acl2"],
2867     "if": ["oic.if.baseline"],
2868     "p": {"bm": 1},
2869     "eps": [
2870         {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2871     ]
2872 },
2873 {
2874     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
2875     "href": "/myFanIntrospection",
2876     "rt": ["oic.wk.introspection"],
2877     "if": ["oic.if.r", "oic.if.baseline"],
2878     "p": {"bm": 3},
2879     "eps": [
2880         {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2881     ]
2882 }
2883 ]
2884

```

2885 The exact format of the "/oic/res" response and a way for a Client to acquire a "/oic/res" response
2886 message is specified in Annex D and 11.3.5 respectively.

2887 **10.4 CoAP based OCF Endpoint discovery**

2888 The following describes CoAP based OCF Endpoint discovery:

- 2889 – Devices shall join the *All OCF Nodes* multicast groups (as defined in [IANA IPv6 Multicast
2890 Address Space Registry]) with scopes 2, 3, and 5 (i.e., ff02::158, ff03::158 and ff05::158) and
2891 shall listen on the port 5683. For compliance to IETF RFC 7252 a Device may additionally join
2892 the *All CoAP Nodes* multicast groups.

- 2893 – Clients intending to discover Resources shall join the multicast groups as defined in a).
- 2894 – Devices shall expose "/oic/res" via an unsecured OCF Endpoint.
- 2895 – Clients shall send discovery requests (GET request) to the *All OCF Nodes* multicast group
- 2896 address with scope 2 ("ff02::158") at port "5683". The requested URI shall be "/oic/res". For
- 2897 compliance to IETF RFC 7252 a Client may additionally send to the *All CoAP Nodes* multicast
- 2898 groups.
- 2899 – If the discovery request is intended for a specific Resource Type, the query parameter "rt" shall
- 2900 be included in the request (see 6.2.2) with its value set to the desired Resource Type. Only
- 2901 Devices hosting the Resource Type shall respond to the discovery request.
- 2902 – When the "rt" query parameter is omitted, all Devices shall respond to the discovery request.
- 2903 – Handling of multicast requests shall be as described in clause 8 of IETF RFC 7252 and clause
- 2904 4.1 in IETF RFC 6690.
- 2905 – Devices which receive the request shall respond using CBOR payload encoding. A Device shall
- 2906 indicate support for CBOR payload encoding for multicast discovery as described in 12.4.

2907 **11 Functional interactions**

2908 **11.1 Introduction**

2909 The functional interactions between a Client and a Server are described in 11.2 through 11.9
 2910 respectively. The functional interactions use CRUDN messages (clause 8) and include Discovery,
 2911 Notification, and Device management. These functions require support of core defined Resources
 2912 as defined in Table 20.

2913 **Table 20 – List of Core Resources**

Pre-defined URI	Resource Name	Resource Type	Related Functional Interaction	Mandatory
"/oic/res"	Default	"oic.wk.res"	Discovery	Yes
"/oic/p"	Platform	"oic.wk.p"	Discovery	Yes
"/oic/d"	Device	"oic.wk.d"	Discovery	Yes
(none)	Configuration	"oic.wk.con"	Device management	No
"/oic/mnt"	Maintenance	"oic.wk.mnt"	Device management	No

2914

2915 **11.2 Onboarding, Provisioning and Configuration**

2916 Onboarding and Provisioning are fully defined by the ISO/IEC 30118-2:2018.

2917 Should a Device support Client update of configurable information it shall do so via exposing an
 2918 oic.wk.con Core Resource (Table 21) in "/oic/res".

2919

Table 21 – Configuration Resource

Example URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
"/example/oic/con"	Device Configuration	"oic.wk.con"	"oic.if.rw"	The Resource Type through which configurable information specific to the Device is exposed. The Resource Properties exposed in "oic.wk.con" are listed in Table 22.	Configuration

"/example/oic/con"	Platform Configuration	"oic.wk.con.p"	"oic.if.rw"	The optional Resource Type through which configurable information specific to the Platform is exposed. The Properties exposed in "oic.wk.con.p" are listed in Table 23.	Configuration
--------------------	------------------------	----------------	-------------	---	---------------

2920

2921 Table 22 defines the "oic.wk.con" Resource Type.

2922

Table 22 – "oic.wk.con" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
(Device) Name	"n" (Common Property of "/example/oic/con")	"string"	N/A	N/A	R, W	Yes	Human friendly name configurable by the end user (e.g. Bob's thermostat). The "n" Common Property of the oic.wk.con Core Resource and the "n" Common Property of the "/oic/d" Core Resource shall have the same Value. When the "n" Common Property Value of the oic.wk.con Core Resource is modified, it shall be reflected to the "n" Common Property of "/oic/d" Core Resource.
Location	"loc"	array of float (has two elements, the first is latitude, the second is longitude)	N/A	Degrees	R, W	No	Provides location information where available.
Location Name	"locn"	"string"	N/A	N/A	R, W	no	Human friendly name for location For example, "Living Room".
Currency	"c"	"string"	N/A	N/A	R,W	no	Indicates the currency that is used for any monetary transactions
Region	"r"	"string"	N/A	N/A	R,W	no	Free form text Indicating the current region in which the Device is located geographically.
Localized Names	"ln"	"array"	N/A	N/A	R,W	no	Human-friendly name of the Device, in one or more languages. This Property is an array of objects where each object has a "language" field (containing an IETF RFC 5646 language tag) and a "value" field containing the Device name in the indicated language. If this Property and the Device Name (n) Property are both supported, the Device Name (n) value shall be included in this array.
Default Language	"dl"	"language-tag"	N/A	N/A	R,W	no	The default language supported by the Device, specified as an IETF RFC 5646 language tag. By default, clients can treat any string Property as being in this

							language unless the Property specifies otherwise.
--	--	--	--	--	--	--	---

2923

2924 Table 23 defines the "oic.wk.con.p" Resource Type.

2925

Table 23 – "oic.wk.con.p" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Platform Names	"mnpn"	"array"	N/A	N/A	R,W	No	<p>Friendly name of the Platform. This Property is an array of objects where each object has a "language" field (containing an IETF RFC 5646 language tag) and a "value" field containing the platform friendly name in the indicated language.</p> <p>For example, <pre>[{"language": "en", "value": "Dave's Laptop"}]</pre></p>

2926

2927 11.3 Resource discovery

2928 11.3.1 Introduction

2929 Discovery is a function which enables OCF Endpoint discovery as well as Resource based
 2930 discovery. OCF Endpoint discovery is described in detail in clause 10. This clause mainly describes
 2931 the Resource based discovery.

2932 11.3.2 Resource based discovery: mechanisms

2933 11.3.2.1 Overview

2934 As part of discovery, a Client may find appropriate information about other OCF peers. This
 2935 information could be instances of Resources, Resource Types or any other information represented
 2936 in the Resource model that an OCF peer would want another OCF peer to discover.

2937 At the minimum, Resource based discovery uses the following:

- 2938 – A Resource to enable discovery shall be defined. The representation of that Resource shall
 2939 contain the information that can be discovered.
- 2940 – The Resource to enable discovery shall be specified and commonly known a-priori. A Device
 2941 for hosting the Resource to enable discovery shall be identified.
- 2942 – A mechanism and process to publish the information that needs to be discovered with the
 2943 Resource to enable discovery.
- 2944 – A mechanism and process to access and obtain the information from the Resource to enable
 2945 discovery. A query may be used in the request to limit the returned information.
- 2946 – A scope for the publication.
- 2947 – A scope for the access.
- 2948 – A policy for visibility of the information.

2949 Depending on the choice of the base aspects, the Framework defines three Resource based
2950 discovery mechanisms:

- 2951 – Direct discovery, where the Resources are published locally at the Device hosting the
2952 Resources and are discovered through peer inquiry.
 - 2953 – Indirect discovery, where Resources are published at a third party assisting with the discovery
2954 and peers publish and perform discovery against the Resource to enable discovery on the
2955 assisting 3rd party.
 - 2956 – Advertisement discovery, where the Resource to enable discovery is hosted local to the initiator
2957 of the discovery inquiry but remote to the Devices that are publishing discovery information.
- 2958 A Device shall support direct discovery.

2959 **11.3.2.2 Direct discovery**

2960 In direct discovery,

- 2961 – The Device that is providing the information shall host the Resource to enable discovery.
- 2962 – The Device publishes the information available for discovery with the local Resource to enable
2963 discovery (i.e. local scope).
- 2964 – Clients interested in discovering information about this Device shall issue RETRIEVE requests
2965 directly to the Resource. The request may be made as a unicast or multicast. The request may
2966 be generic or may be qualified or limited by using appropriate queries in the request.
- 2967 – The Server Device that receives the request shall send a response with the discovered
2968 information directly back to the requesting Client Device.
- 2969 – The information that is included in the request is determined by the policies set for the Resource
2970 to be discovered locally on the responding Device.

2971 **11.3.2.3 Indirect discovery of Resources (Resource Directory based discovery)**

2972 In indirect discovery the information about the resource to be discovered is hosted on a Server that
2973 is not hosting the Resource. See 11.3.6 for details on Resource Directory based discovery.

2974 In indirect discovery:

- 2975 – The Resource to be discovered is hosted on a Device that is neither the Client initiating the
2976 discovery nor the Device that is providing or publishing the information to be discovered. This
2977 Device may use the same Resource to provide discovery for multiple agents looking to discover
2978 and for multiple agents with information to be discovered.
- 2979 – The Device to be discovered or with information to discover, publishes that information with
2980 Resource to be discovered on a different Device. The policies on the information shared
2981 including the lifetime/validity are specified by the publishing Device. The publishing Device may
2982 modify these policies as required.
- 2983 – The Client doing the discovery may send a unicast discovery request to the Device hosting the
2984 discovery information or send a multicast request that shall be monitored and responded to by
2985 the Device. In both cases, the Device hosting the discovery information is acting on behalf of
2986 the publishing Device.
- 2987 – The discovery policies may be set by the Device hosting the discovery information or by the
2988 party that is publishing the information to be discovered. The discovery information that is
2989 returned in the discovery response shall adhere to the policies that are in effect at the time of
2990 the request.

2991 **11.3.2.4 Advertisement Discovery**

2992 In advertisement discovery:

- 2993 – The Resource to enable discovery is hosted local to the Device that is initiating the discovery
- 2994 request (Client). The Resource to enable discovery may be a Core Resource or discovered as
- 2995 part of a bootstrap.
- 2996 – The request could be an implementation dependent lookup or be a local RETRIEVE request
- 2997 against the Resource that enables discovery.
- 2998 – The Device with information to be discovered shall publish the appropriate information to the
- 2999 Resource that enables discovery.
- 3000 – The publishing Device is responsible for the published information. The publishing Device may
- 3001 UPDATE the information at the resource to enable discovery based on its needs by sending
- 3002 additional publication requests. The policies on the information that is discovered including
- 3003 lifetime is determined by the publishing Device.

3004 **11.3.3 Resource based discovery: Information publication process**

3005 The mechanism to publish information with the Resource to enable discovery can be done either

3006 locally or remotely. The publication process is depicted in Figure 15. The Device which has

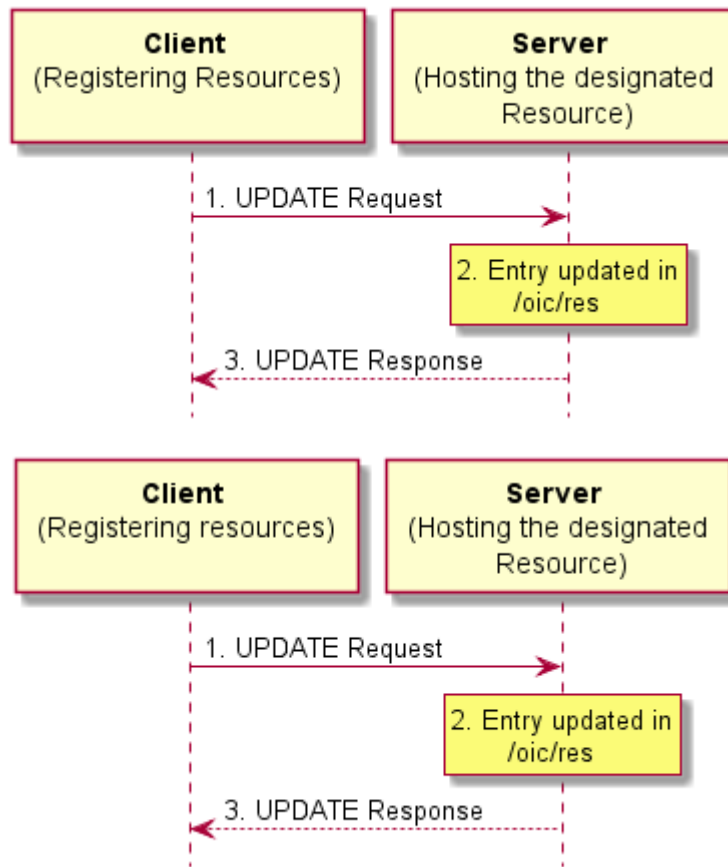
3007 discovery information to publish shall a) either update the Resource that enables discovery if

3008 hosted locally or b) issue an UPDATE request with the information to the Device which hosts the

3009 Resource that enables discovery. The Device hosting the Resource to enable discovery

3010 adds/updates the Resource to enable discovery with the provided information and then responds

3011 to the Device which has requested the publication of the Resource with an UPDATE response.



3012

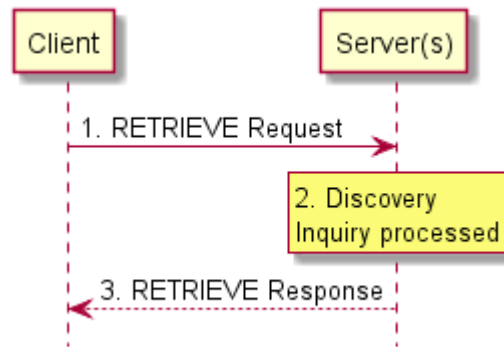
3013

3014

Figure 15 – Resource based discovery: Information publication process

3015 **11.3.4 Resource based discovery: Finding information**

3016 The discovery process (Figure 16) is initiated as a RETRIEVE request to the Resource to enable
3017 discovery. The request may be sent to a single Device (as in a Unicast) or to multiple Devices (as
3018 in Multicast). The specific mechanisms used to do Unicast or Multicast are determined by the
3019 support in the data connectivity layer. The response to the request has the information to be
3020 discovered based on the policies for that information. The policies can determine which information
3021 is shared, when and to which requesting agent. The information that can be discovered can be
3022 Resources, types, configuration and many other standards or custom aspects depending on the
3023 request to appropriate Resource and the form of request. Optionally the requester may narrow the
3024 information to be returned in the request using query parameters in the URI query.



3025

3026 **Figure 16 – Resource based discovery: Finding information**

3027

3028 *Discovery Resources*

3029 The following Core Resources shall be implemented on all Devices to support discovery:

- 3030 – "/oic/res" for discovery of Resources.
- 3031 – "/oic/p" for discovery of Platform.
- 3032 – "/oic/d" for discovery of Device information.

3033 Devices shall expose each of "/oic/res", "/oic/d", and "/oic/p" via an unsecured OCF Endpoint.
3034 Further details for these mandatory Core Resources are described in Table 24.

3035 *Platform Resource*

3036 The OCF recognizes that more than one instance of Device may be hosted on a single Platform.
3037 Clients need a way to discover and access the information on the Platform. The Core Resource,
3038 "/oic/p" exposes Platform specific Properties. All instances of Device on the same Platform shall
3039 have the same values of any Properties exposed (i.e. a Device may choose to expose optional
3040 Properties within "/oic/p" but when exposed the value of that Property should be the same as the
3041 value of that Property on all other Devices on that Platform).

3042 *Device Resource*

3043 The Device Resource shall have the pre-defined URI "/oic/d". The Resource "/oic/d" exposes the
3044 Properties pertaining to a Device as defined in Table 24. The Properties exposed are determined
3045 by the specific instance of Device and defined by the Resource Type(s) of "/oic/d" on that Device.
3046 Since all the Resource Types of "/oic/d" are not known a priori, the Resource Type(s) of "/oic/d"
3047 shall be determined by discovery through the Core Resource "/oic/res". The Device Resource
3048 "/oic/d" shall have a default Resource Type that helps in bootstrapping the interactions with this
3049 Device (the default type is described in Table 24).

3050 *Protocol indication*

3051 A Device may need to support different messaging protocols depending on requirements for
 3052 different vertical domain profiles. For example, a Smart Home profile may use CoAP and an
 3053 Industrial profile may use DDS. To enable interoperability, a Device uses the protocol indication to
 3054 indicate the transport protocols they support and can communicate over.

3055 **Table 24 – Mandatory discovery Core Resources**

Pre-defined URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
"/oic/res"	Default	"oic.wk.res"	"oic.if.ll"	The Resource through which the corresponding Server is discovered and introspected for available Resources. "/oic/res" shall expose the Resources that are discoverable on a Device. When a Server receives a RETRIEVE request targeting "/oic/res" (e.g., "GET /oic/res"), it shall respond with the links list of all the Discoverable Resources of itself. The "/oic/d" and "/oic/p" are Discoverable Resources, hence their links are included in "/oic/res" response. The Properties exposed by "/oic/res" are listed in Table 25.	Discovery
"/oic/p"	Platform	"oic.wk.p"	"oic.if.r"	The Discoverable Resource through which Platform specific information is discovered. The Properties exposed by "/oic/p" are listed in Table 28	Discovery
"/oic/d"	Device	"oic.wk.d" and/or one or more Device Specific Resource Type ID(s)	"oic.if.r"	The discoverable via "/oic/res" Resource which exposes Properties specific to the Device instance. The Properties exposed by "/oic/d" are listed in Table 27 "/oic/d" may have one or more Resource Type(s) that are specific to the Device in addition to the default Resource Type or if present overriding the default Resource Type. The base type "oic.wk.d" defines the Properties that shall be exposed by all Devices. The Device specific Resource Type(s) exposed are dependent on the class of Device (e.g. air conditioner, smoke alarm, and combined light/fan); applicable values are defined by ISO/IEC 30118-5:2018.	Discovery

3056 Table 25 defines "oic.wk.res" Resource Type.

3057 **Table 25 – "oic.wk.res" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Name	"n"	string	N/A	N/A	R	No	Human-friendly name defined by the vendor

Links	"links"	array	See 7.8.2	N/A	R	Yes	The array of Links describes the URI, supported Resource Types and OCF Interfaces, and access policy.
-------	---------	-------	-----------	-----	---	-----	---

3058

3059 A Device shall support CoAP based discovery as the baseline discovery mechanism (see 10.4).

3060 The "/oic/res" shall list all Resources that are indicated as discoverable (see 11.3). Also the
3061 following architecture Resource Types shall be listed:

- 3062 – Introspection Resource indicated with an "rt" value of "oic.wk.introspection".
- 3063 – "/oic/p" indicated with an "rt" value of "oic.wk.p".
- 3064 – "/oic/d" indicated with an "rt" value of "oic.wk.d"
- 3065 – "/oic/sec/doxm" indicated with an "rt" value of "oic.r.doxm" as defined in ISO/IEC 30118-2:2018.
- 3066 – "/oic/sec/pstat" indicated with an "rt" value of "oic.r.pstat" as defined in ISO/IEC 30118-2:2018.
- 3067 – "/oic/sec/acl2" indicated with an "rt" value of "oic.r.acl2" as defined in ISO/IEC 30118-2:2018.
- 3068 – "/oic/sec/cred" indicated with an "rt" value of "oic.r.cred" as defined in ISO/IEC 30118-2:2018.

3069 Conditionally required:

- 3070 – "/oic/res" with an "rt" value of "oic.wk.res" as self-reference, on the condition that "oic/res" has
3071 to signal that it is Observable by a Client.

3072 The Introspection Resource is only applicable for Devices that host Vertical Resource Types (e.g.
3073 "oic.r.switch.binary") or vendor-defined Resource Types. Devices that only host Resources
3074 required to onboard the Device as a Client do not have to implement the Introspection Resource.

3075 Table 26 provides an OCF registry for protocol schemes.

3076 **Table 26 – Protocol scheme registry**

SI Number	Protocol
1	"coap"
2	"coaps"
3	"http"
4	"https"
5	"coap+tcp"
6	"coaps+tcp"

3077

3078 NOTE The discovery of an OCF Endpoint used by a specific protocol is out of scope. The mechanism used by a Client
3079 to form requests in a different messaging protocol other than discovery is out of scope.

3080 The following applies to the use of "/oic/d":

- 3081 – A Device may choose to expose its Device Type(s) (e.g., refrigerator or A/C or composite of
3082 multiple Device Types) by adding the Device Type to the list of Resource Types associated with
3083 "/oic/d".

- 3084 – For example; "rt" of "/oic/d" becomes ["oic.wk.d", "oic.d.<thing1>", "oic.d.<thing2>"]; where
 3085 "oic.d.<thing1>" and "oic.d.<thing2>" are defined in another specification such as
 3086 ISO/IEC 30118-5:2018.
- 3087 – This implies that the Properties exposed by "/oic/d" are by default the mandatory Properties
 3088 in Table 27.
- 3089 – A vertical may choose to extend the list of Properties defined by the Resource Type "oic.wk.d".
 3090 In that case, the vertical shall assign a new Device Type specific Resource Type ID. The
 3091 mandatory Properties defined in Table 27 shall always be present.
- 3092 – A Device may choose to expose a separate, Discoverable Resource with its Resource Type ID
 3093 set to an OCF defined Device Type. In this case the Resource is equivalent to an instance of
 3094 "oic.wk.d" and adheres to the definition thereof. As such the Resource shall at a minimum
 3095 expose the mandatory Properties of "oic.wk.d". In the case where the Resource tagged in this
 3096 manner is defined to be an instance of a Collection in accordance with 7.8.3 then the Resources
 3097 that are part of that Collection shall at a minimum include the Resource Types mandated for
 3098 the Device Type. For example, if a Collection Resource has an "rt" value of ["oic.d.light"], the
 3099 Collection includes an instance of "oic.r.switch.binary" which is mandatory for an "oic.d.light"
 3100 as per ISO/IEC 30118-5:2018.

3101 Table 27 "oic.wk.d" Resource Type definition defines the base Resource Type for the "/oic/d"
 3102 Resource.

3103

Table 27 – "oic.wk.d" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
(Device) Name	"n"	"string"	N/A	N/A	R	Yes	Human friendly name defined by the vendor. In the presence of "n" Property of "/oic/con", both have the same Property Value. When "n" Property Value of "/oic/con" is modified, it shall be reflected to "n" Property Value of "/oic/d".
Spec Version	"icv"	"string"	N/A	N/A	R	Yes	Spec version of this document this Device is implemented to, The syntax is "ocf.<major>.<minor>.<sub-version>" where <major>, <minor>, and <sub-version> are the major, minor and sub-version numbers of the document respectively. For this version of the document, the string value shall be "ocf.2.0.2".
Device ID	"di"	"uuid"	N/A	N/A	R	Yes	Unique identifier for Device. This value shall be the same value (i.e. mirror) as the doxm.deviceuuid Property as defined in ISO/IEC 30118-2:2018. Handling privacy-sensitivity for the "di" Property, refer to clause 13.16 in ISO/IEC 30118-2:2018.
Data Model Version	"dmv"	"csv"	N/A	N/A	R	Yes	Spec version of the Resource specification to which this Device data model is implemented; if implemented against a Vertical specific Device specification(s), then the Spec version of the vertical specification this Device model is implemented to. The syntax is a comma separated list of <res>.<major>.<minor>.<sub-

							version> or <vertical>. <major>. <minor>. <sub-version>. <res> is the string "ocf.res" and <vertical> is the name of the vertical defined in the Vertical specific Resource specification. The <major>, <minor>, and <sub-version> are the major, minor and sub-version numbers of the specification respectively. One entry in the csv string shall be the applicable version of the Resource Type Specification for the Device (e.g "ocf.res.1.0.0"). If applicable, additional entry(-ies) in the csv shall be the vertical(s) being realized (e.g. "ocf.sh.1.0.0"). This value may be extended by the vendor. The syntax for extending this value, as a comma separated entry, by the vendor shall be by adding x.<Domain_Name>.<vendor_string>. For example "ocf.res.1.0.0, ocf.sh.1.0.0, x.com.example.string". The order of the values in the comma separated string can be in any order (i.e. no prescribed order). This Property shall not exceed 256 octets.
Protocol Independent ID	"piid"	"uuid"	N/A	N/A	R	Yes	A unique and immutable Device identifier. A Client can detect that a single Device supports multiple communication protocols if it discovers that the Device uses a single Protocol Independent ID value for all the protocols it supports. Handling privacy-sensitivity for the "piid" Property, refer to clause 13.16 in ISO/IEC 30118-2:2018.
Localized Descriptions	"ld"	"array"	N/A	N/A	R	No	Detailed description of the Device, in one or more languages. This Property is an array of objects where each object has a "language" field (containing an IETF RFC 5646 language tag) and a "value" field containing the Device description in the indicated language.
Software Version	"sv"	"string"	N/A	N/A	R	No	Version of the Device software.
Manufacturer Name	"dmn"	"array"	N/A	N/A	R	No	Name of manufacturer of the Device, in one or more languages. This Property is an array of objects where each object has a "language" field (containing an IETF RFC 5646 language tag) and a "value" field containing the manufacturer name in the indicated language.
Model Number	"dmno"	"string"	N/A	N/A	R	No	Model number as designated by manufacturer.

3104 The additional Resource Type(s) of the "/oic/d" Resource are defined by ISO/IEC 30118-5:2018.

3105 Table 28 defines "oic.wk.p" Resource Type.

Table 28 – "oic.wk.p" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Platform ID	"pi"	"string"	N/A	N/A	R	Yes	Unique identifier for the physical Platform (UIUID); this shall be a UUID in accordance with IETF RFC 4122. It is recommended that the UUID be created using the random generation scheme (version 4 UUID) specific in the RFC. Handling privacy-sensitivity for the "pi" Property, refer to clause 13.16 in ISO/IEC 30118-2:2018.
Manufacturer Name	"mnmn"	"string"	N/A	N/A	R	Yes	Name of manufacturer.
Manufacturer Details Link	"mnml"	"uri"	N/A	N/A	R	No	Reference to manufacturer, represented as a URI.
Model Number	"mnmo"	"string"	N/A	N/A	R	No	Model number as designated by manufacturer.
Date of Manufacture	"mndt"	"date"	N/A	Time	R	No	Manufacturing date of Platform.
Serial number	"mnsel"	"string"	N/A	s	R	No	Serial number of the Platform, may be unique for each Platform of the same model number.
Platform Version	"mnpv"	"string"	N/A	N/A	R	No	Version of Platform – string (defined by manufacturer).
OS Version	"mnos"	"string"	N/A	N/A	R	No	Version of Platform resident OS – string (defined by manufacturer).
Hardware Version	"mnhw"	"string"	N/A	N/A	R	No	Version of Platform hardware.
Firmware version	"mnfv"	"string"	N/A	N/A	R	No	Version of Platform firmware.
Support link	"mnsli"	"uri"	N/A	N/A	R	No	URI that points to support information from manufacturer.
SystemTime	"st"	"date-time"	N/A	N/A	R	No	Reference time for the Platform.
Vendor ID	"vid"	"string"	N/A	N/A	R	No	Vendor defined string for the Platform. The string is freeform and up to the vendor on what text to populate it.

3107 **11.3.5 Resource discovery using "/oic/res"**3108 Discovery using "/oic/res" is the default discovery mechanism that shall be supported by all Devices
3109 as follows:

- 3110 – Every Device updates its local "/oic/res" with the Resources that are discoverable (see 7.3.2.2).
3111 Every time a new Resource is instantiated on the Device and if that Resource is discoverable

3112 by a remote Device then that Resource is published with the "/oic/res" Resource that is local to
3113 the Device (as the instantiated Resource).

3114 – A Device wanting to discover Resources or Resource Types on one or more remote Devices
3115 makes a RETRIEVE request to the "/oic/res" on the remote Devices. This request may be sent
3116 multicast (default) or unicast if only a specific host is to be probed. The RETRIEVE request may
3117 optionally be restricted using appropriate clauses in the query portion of the request. Queries
3118 may select based on Resource Types, OCF Interfaces, or Properties.

3119 – The query applies to the representation of the Resources. "/oic/res" is the only Resource whose
3120 representation has "rt". So "/oic/res" is the only Resource that can be used for Multicast
3121 discovery at the transport protocol layer.

3122 – The Device receiving the RETRIEVE request responds with a list of Resources, the Resource
3123 Type of each of the Resources and the OCF Interfaces that each Resource supports.
3124 Additionally, information on the policies active on the Resource can also be sent. The policy
3125 supported includes Observability and discoverability.

3126 – The receiving Device may do a deeper discovery based on the Resources returned in the
3127 request to "/oic/res".

3128 The information that is returned on discovery against "/oic/res" is at the minimum:

3129 – The URI (relative or fully qualified URL) of the Resource.

3130 – The Resource Type(s) of each Resource. More than one Resource Type may be returned if the
3131 Resource enables more than one type. To access Resources of multiple types, the specific
3132 Resource Type that is targeted shall be specified in the request.

3133 – The OCF Interfaces supported by that Resource. Multiple OCF Interfaces may be returned. To
3134 access a specific OCF Interface that OCF Interface shall be specified in the request. If the OCF
3135 Interface is not specified, then the Default OCF Interface is assumed.

3136 Different "/oic/res" responses are returned according to requesting Clients, which indicate their
3137 preference via inclusion or otherwise of an OCF-Accept-Content-Format-Version option.

3138 For Clients that do not include the OCF-Accept-Content-Format-Version option, an "/oic/res"
3139 response shall use "sec" and "port" to provide the information for an encrypted connection. See
3140 Annex E for the schema for the Link.

3141 For Clients that do include the OCF-Accept-Content-Format-Version option, an "/oic/res" response
3142 includes an array of Links to conform to IETF RFC 6690. Each Link shall use an "eps" Parameter
3143 to provide the information for an encrypted connection and carry "anchor" of the value OCF URI
3144 where the authority component of <deviceId> indicates the Device hosting the target Resource.

3145 The OpenAPI 2.0 file for discovery using "/oic/res" is described in Annex D the schema that is
3146 applicable to requesting Clients that do not include an OCF-Accept-Content-Format-Version option
3147 is described in Annex E. Also refer to clause 10 (OCF Endpoint discovery) for details of Multicast
3148 discovery using "/oic/res" on a CoAP transport.

3149 For example, a Light Device might return the following to OIC 1.1 Clients:

```
3150 [
3151   {
3152     "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
3153     "links": [
3154       {
3155         "href": "coaps://[fe80::b1d6]:44444/oic/res",
3156         "rel": "self",
3157         "rt": ["oic.wk.res"],
3158         "if": ["oic.if.ll", "oic.if.baseline"],
3159         "p": {"bm": 3}
3160       }
3161     ]
3162   }
3163 ]
```

```

3160     },
3161     {
3162         "href": "/oic/p",
3163         "rt": ["oic.wk.p"],
3164         "if": ["oic.if.r", "oic.if.baseline"],
3165         "p": {"bm": 3, "sec": true, "port": 11111}
3166     },
3167     {
3168         "href": "/oic/d",
3169         "rt": ["oic.wk.d", "oic.d.light"],
3170         "if": ["oic.if.r", "oic.if.baseline"],
3171         "p": {"bm": 3, "sec": true, "port": 11111}
3172     },
3173     {
3174         "href": "/myLight",
3175         "rt": ["oic.r.switch.binary"],
3176         "if": ["oic.if.a", "oic.if.baseline"],
3177         "p": {"bm": 3, "sec": true, "port": 11111}
3178     }
3179 ]
3180 }
3181 ]

```

3182 The light Device might return the following to Clients that request with the Content Format of
3183 "application/vnd.ocf+cbor" in Accept Option:

```

3184 [
3185     {
3186         "href": "/oic/res",
3187         "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989/oic/res",
3188         "rel": "self",
3189         "rt": ["oic.wk.res"],
3190         "if": ["oic.if.ll", "oic.if.baseline"],
3191         "p": {"bm": 3},
3192         "eps": [{"ep": "coap://[fe80::b1d6]:44444"}]
3193     },
3194     {
3195         "href": "/oic/p",
3196         "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
3197         "rt": ["oic.wk.p"],
3198         "if": ["oic.if.r", "oic.if.baseline"],
3199         "p": {"bm": 3},
3200         "eps": [{"ep": "coap://[fe80::b1d6]:44444"},
3201                 {"ep": "coaps://[fe80::b1d6]:11111"}]
3202     },
3203     {
3204         "href": "/oic/d",
3205         "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
3206         "rt": ["oic.wk.d", "oic.d.light"],
3207         "if": ["oic.if.r", "oic.if.baseline"],
3208         "p": {"bm": 3},
3209         "eps": [{"ep": "coap://[fe80::b1d6]:44444"},
3210                 {"ep": "coaps://[fe80::b1d6]:11111"}]
3211     },
3212     {
3213         "href": "/myLight",
3214         "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
3215         "rt": ["oic.r.switch.binary"],
3216         "if": ["oic.if.a", "oic.if.baseline"],
3217         "p": {"bm": 3},

```



```

3220     "eps": [ { "ep": "coap://[fe80::b1d6]:44444" },
3221               { "ep": "coaps://[fe80::b1d6]:11111" }
3222           ]
3223     }
3224 ]

```

3225 After performing discovery using "/oic/res", Clients may discover additional details about Server by
3226 performing discovery using "/oic/p", "/oic/rts" etc. If a Client already knows about Server it may
3227 discover using other Resources without going through the discovery of "/oic/res".

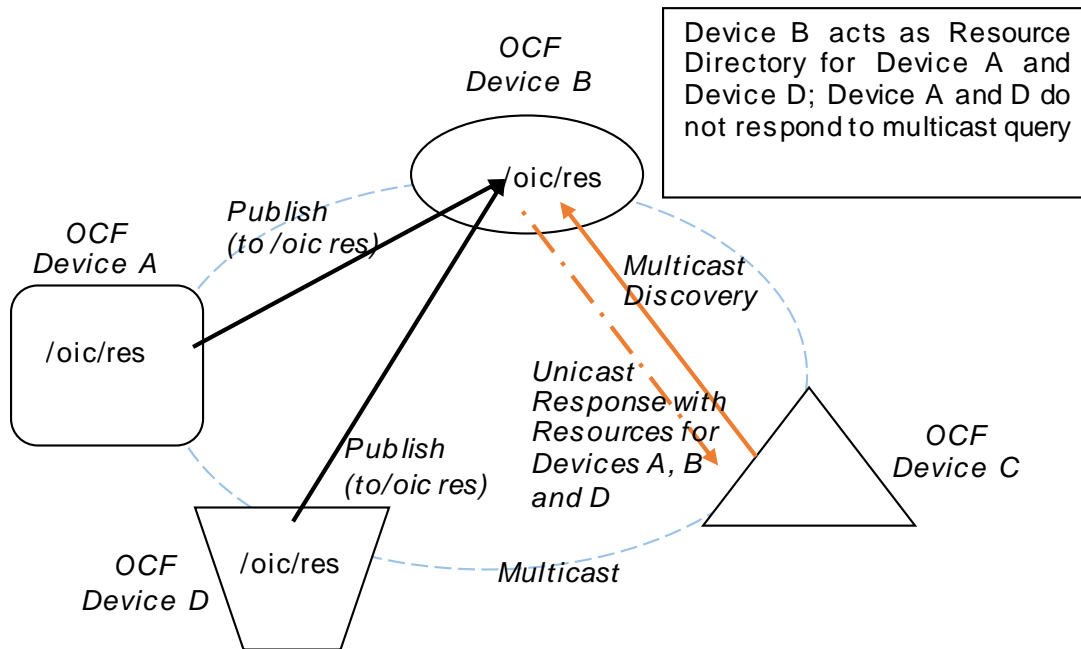
3228 11.3.6 Resource Directory (RD) based discovery

3229 11.3.6.1 Introduction

3230 11.3.6.1.1 Indirect discovery for lookup of the Resources

3231 Direct discovery is the mechanism used currently to find Resources in the network. When needed,
3232 Resources are queried at a particular Device directly or a multicast packet is sent to all Devices.
3233 Each queried Device responds directly with its Resources to the discovering Device. Resources
3234 available locally are registered on the same Device.

3235 In some situations, one of the other mechanisms described in 11.3.2.3, called indirect discovery,
3236 may be required. Indirect discovery is when a 3rd party Device, other than the discovering Device
3237 and the discovered Device, assists with the discovery process. The 3rd party Device, called
3238 Resource Directory (RD), only provides information on Resources on behalf of another Device but
3239 does not host Resources on part of that Device.



3240

3241 **Figure 17 – Indirect discovery of Resources by via an RD**

3242 In Figure 17, Device B acts as Resource Directory for Device A and Device D. Device A and Device
3243 D publish their Resource information to Device B. Device C may query Deice B to acquire the
3244 Resource information of Device A and Device D. Device A and Device D may not respond to a
3245 multicast query when Device B, as a Resource Directory, responds to the query on their behalf.

3246 Indirect discovery is useful for a constrained Device that needs to sleep to manage power and
3247 cannot process every discovery request, or when Devices may not be on the same network and
3248 requires optimization for discovery. Once Resources are discovered using indirect discovery, i.e.,

3249 RD query, then the access to the Resource is done by a request sent directly to the Device that
 3250 hosts that Resource.

3251 **11.3.6.1.2 Resource Directory**

3252 A Resource Directory (RD) is a Device that assists with indirect discovery. A Device which acts as
 3253 an RD will be involved in the following operations.

- 3254 – *RD discovery* – the procedure with which publishing Devices discover an RD and acquire the
 3255 criteria to select from among multiple detected RDs.
- 3256 – *Resource publish* – the procedures with which Devices publish their Resource information, i.e.
 3257 Links. Future revision of this document will allow modifying RD entries with UPDATE and
 3258 DELETE operations. Any UPDATE or DELETE operations performed on an RD in this
 3259 documentshould be either silently ignored or generate an error.
- 3260 – *Resource exposure* – the feature with which RDs expose the Links hosted by the 3rd party
 3261 Devices via their own "/oic/res".

3262 The RDs make use of Resource Type "oic.wk.rd" defined in Table 29 and Table 30. A Device that
 3263 supports the capability to host indirect discovery shall expose an instance of "oic.wk.rd" in its
 3264 "/oic/res" to announce that it serves as an RD. The discoverable instance of "oic.wk.rd" shall allow
 3265 only secure connections (e.g. OCF Endpoint with a scheme of "coaps" or "coaps+tcp"). A publishing
 3266 Device may send a RETRIEVE request to "/oic/rd" to acquire the selection criteria among multiple
 3267 RDs. Then it may send an UPDATE request to "/oic/rd" with its Links in the payload to publish the
 3268 Links in "/oic/res" of the RD. A publishing Device is responsible to insure an RD has the correct
 3269 published Links to expose via its "/oic/res".

3270 **Table 29 – "oic.wk.rd" Resource Type definition**

Pre-defined URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
"/oic/rd"	Resource Directory	"oic.wk.rd"	"oic.if.baseline"	The Discoverable Resource Type through with which an RD 1) facilitates its discovery and provides the criteria to select an RD and 2) allows Devices to publish their Links in "/oic/res" of the RD.	Discovery

3271

3272

3273

Table 30 – "oic.wk.rd" Properties

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Selector	"sel"	"integer"	N/A	N/A	R	Yes	Provides the criteria for RD selection. An integer representing a value calculated by the RD. The value is in the range of 0 to 100. The lower the value, the more preferable the RD is.

3274

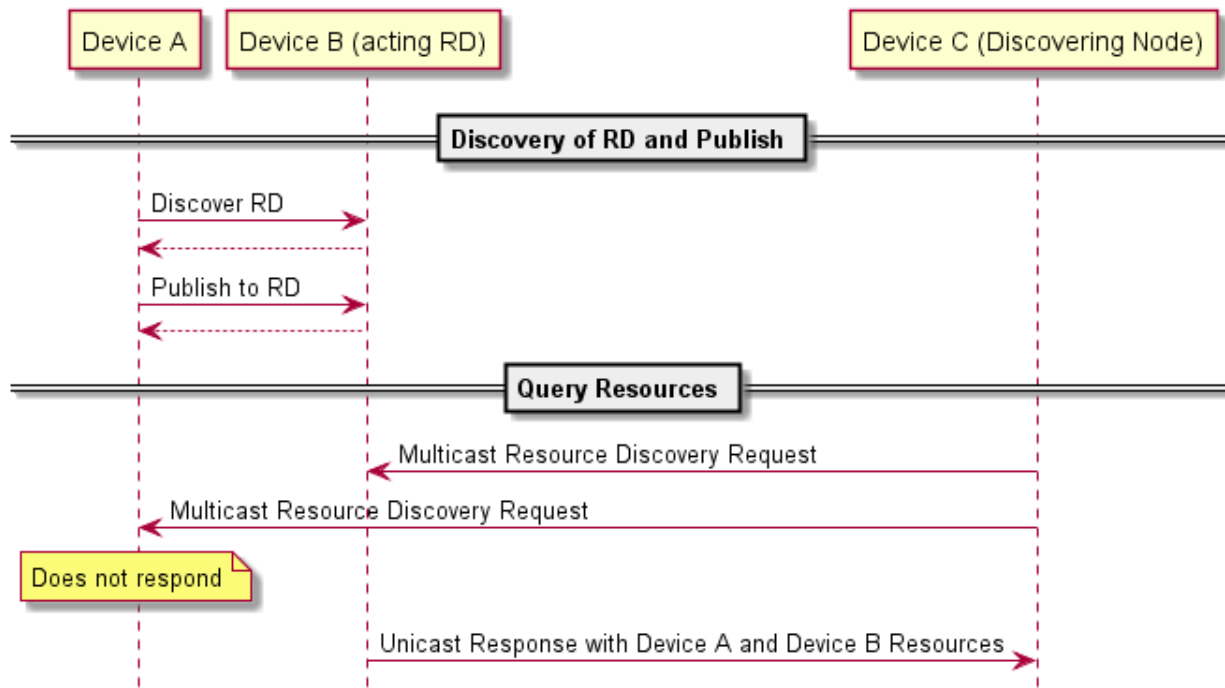
3275 An RD may be queried at its "/oic/res" Resource to find Resources hosted on other Devices. These
 3276 Devices can be sleepy nodes or any other Device that cannot or may not respond to discovery
 3277 requests. A publishing Device may publish all or a partial list of Resources they host to an RD. The
 3278 RD then responds to queries for Resource discovery on behalf of the publishing Device (for
 3279 example: when a Device may go to sleep). For general Resource discovery, the RD behaves like
 3280 any other Server in responding to requests to "/oic/res".

3281 The remainder of 11.3.6 is divided into three parts. The first part covers "RD Discovery" (see
 3282 11.3.6.2), i.e., discovering and selecting of an RD. The second part covers "Resource publish" (see
 3283 11.3.6.3), i.e., publishing of Resources. The third part covers "Resource exposure" (see 11.3.6.4)
 3284 where the RD replies to queries from Devices looking to discover Resources.

3285 **11.3.6.2 RD discovery**

3286 **11.3.6.2.1 Discovering an RD**

3287 An RD shall support RD discovery.



3288

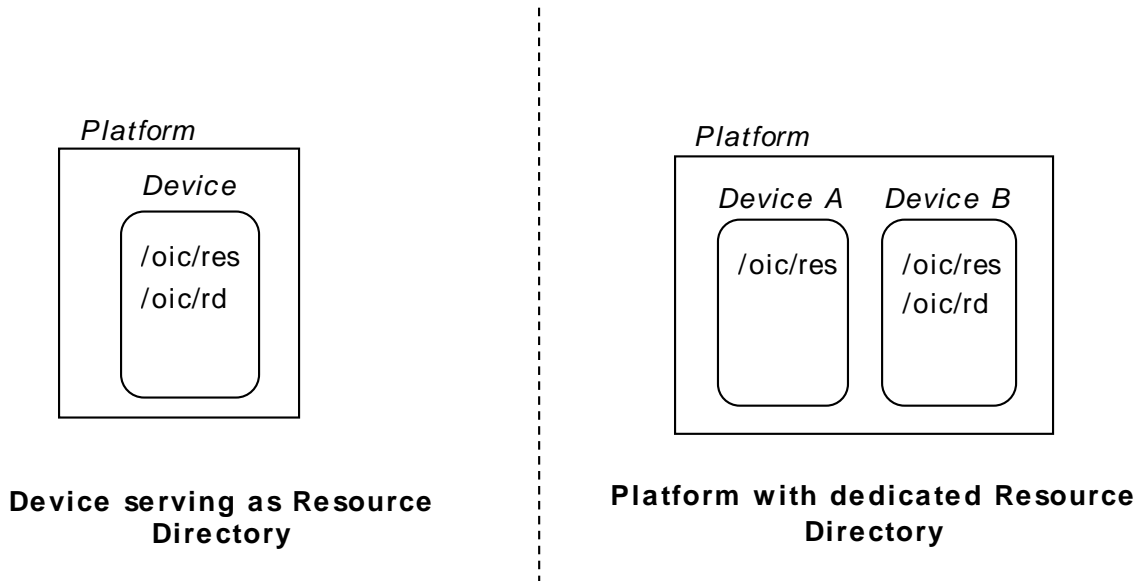
3289 **Figure 18 – RD discovery and RD supported query of Resources support**

3290 In Figure 18 and Figure 19, a Device that wishes to publish its Resources first discovers an RD
 3291 and then publishes the desired Resource information. Once a set of Resources have been
 3292 published to an RD then the publishing Device should not respond to multicast Resource discovery
 3293 queries for those published Resources when the RD is on the same multicast domain. In that case,
 3294 only the RD should respond to multicast Resource discovery requests on the Resource published
 3295 to it.

3296 It is allowed for more than one Device to act as an RD. The reason to have multiple RD support is
 3297 to make networks scalable, handle network failures and prevent centralized Device failure
 3298 bottlenecks. This does not preclude a scenario where a use case or deployment environment may
 3299 require a single Device in the environment to be deployed as the only RD (e.g. gateway model).

3300 Discovering an RD may result in responses from more than one RD. If more than one RD responds,
 3301 the discovering Device may select one of them based on the weighting parameter(s) provided in
 3302 the response from the RD.

3303 A Client that performs Resource discovery uses an RD just like it uses any other Server for
 3304 discovery. It may send a unicast request to the RD when it needs only the Resources published on
 3305 the RD or do a multicast query when it does not require or have explicit knowledge of an RD.



3306
 3307 **Figure 19 – Resource Direction Deployment Scenarios**

3308 RDs may also be discovered in the following ways:

- 3309 – Pre-configuration: Devices wishing to publish Resource information may be configured a priori
 3310 with the information (e.g. IP address, port, transport etc.) of a specific RD. This pre-
 3311 configuration may be done at onboarding or may be updated on the Device using an out-of-
 3312 band method. This pre-configuration may be done by the manufacturer.
- 3313 – Query-oriented: A publishing Device wanting to discover Resource Directories using query-
 3314 oriented discovery may issue a multicast Resource discovery request for "/oic/res?rt=oic.wk.rd".
 3315 Only and all Devices that can be an RD shall respond to this query. The "/oic/rd" response shall
 3316 include information about the RD i.e., the presence of "oic.wk.rd" Link (as defined by the
 3317 Resource Type) and a subsequent query to "/oic/rd" would produce weighting parameters to
 3318 allow the discovering Device to select between RDs (see details in RD selection 11.3.6.2.2).
 3319 The "oic.wk.rd" Resource shall be instantiated on the Devices acting as RDs. The "oic.wk.rd"
 3320 schema is as defined in Annex D.

3321 **11.3.6.2.2 RD selection process**

3322 The Device that wants to use an RD will find zero or more RDs on the network. There may not be
 3323 an RD within the network. When discovering RDs, the Device needs to select an RD of all RDs
 3324 found on the network. The Device may send a RETRIEVE request to "/oic/rd/" of a specific RD, the
 3325 RD shall respond with the representation of "/oic/rd/" containing selection criteria as defined by the
 3326 "sel" Property. The lower the "sel" Property value is, the more preferable the responding RD is.
 3327 The creation of the "sel" value is vendor defined.

3328 For example an "/oic/rd" response may return the following.

```
3329 {
3330   "rt": ["oic.wk.rd"],
```

```
3331     "if": ["oic.if.baseline"],
3332     "sel": 50
3333 }
```

3334 The selection based on the "sel" Property value will ensure that a Device can judge if the found RD
3335 is suitable for its needs.

3336 The following situations may occur during the selection of an RD:

- 3337 – A single or multiple RDs are present in the network.
- 3338 – No RD is present in the network.
- 3339 – An additional RD arrives on the network.

3340 In the first scenario, the RDs are already present. If a single RD is detected then that RD may be
3341 used. When multiple RDs are detected the Device may use the "sel" Property value to select the
3342 RD.

3343 In the second scenario, the publishing Device may continue looking for an RD until one is found or
3344 give up using an RD altogether.

3345 In the third scenario, the Device has already published its Resources to an existing RD, then
3346 discovers a new RD on the network. After judging the "sel" Property value, the Device may choose
3347 to move to the new RD. The Device should delete its Resource information from the currently used
3348 RD and publish the information to the new RD.

3349 **11.3.6.3 Resource publish**

3350 **11.3.6.3.1 Overview**

3351 An RD shall provide the facility to allow Devices to publish their Resource information to an RD.

3352 **11.3.6.3.2 Publish Resources**

3353 **11.3.6.3.2.1 Overview**

3354 After the selection process of an RD, a Device may push its Resource information to the selected
3355 RD, i.e., publish the Links in its "/oic/res" to the "/oic/res" of the RD.

3356 The publishing Device may decide to publish all Resources or just a few of the Resources on the
3357 RD. The publishing Device should only publish Resources that are otherwise published to its own
3358 "/oic/res"; a publishing Device should not publish non-Discoverable Resources or Resources
3359 hosted by some other Device. A publishing Device shall respond to discovery requests on its
3360 "/oic/res" Resource unless all its Discoverable Resources have been published in an RD.

3361 **11.3.6.3.2.2 Publish: Push Resource information**

3362 Resource information may be published using an UPDATE request sent to "/oic/rd".

3363 A Device which hosts a Resource may publish the Resource information, i.e. the Link targeting the
3364 Resource, to an RD by sending an UPDATE request with the Link in the payload. The published
3365 Link shall be exposed through the "/oic/res" of the RD.

3366 When a Device first publishes a Link or Links, it shall send an UPDATE request to the "/oic/rd"
3367 Resource of the RD including the following key-value pairs in the payload:

- 3368 – "di" –its value shall be the Device ID of the publishing Device, i.e. the "di" value of "/oic/d".
- 3369 – "links" –its value shall be the array of Links to be published. Links may omit the "ins" Parameter
3370 in which case the RD will assign a value for each Link. The supplied "ins" Parameter by the
3371 Client is allowed to be overruled by the RD, e.g. an RD can ignore the supplied "ins" value.

3372 – "ttl" –its value indicates how long (in seconds) the publishing Device requests the RD to keep
3373 this published Link.

3374 Notice that the payload shall carry the appropriate Content-Format of "application/vnd.ocf+cbor".

```
3375 {  
3376   "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",  
3377   "links": [  
3378     {  
3379       "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9"  
3380       "href": "/myLightSwitch",  
3381       "rt": ["oic.r.switch.binary"],  
3382       "if": ["oic.if.a", "oic.if.baseline"],  
3383       "p": {"bm": 3},  
3384       "eps": [  
3385         {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},  
3386         {"ep": "coaps://[fe80::b1d6]:1122"},  
3387         {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}  
3388       ]  
3389     },  
3390     {  
3391       "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",  
3392       "href": "/myLightBrightness",  
3393       "rt": ["oic.r.brightness"],  
3394       "if": ["oic.if.a", "oic.if.baseline"],  
3395       "p": {"bm": 3},  
3396       "eps": [  
3397         {"ep": "coaps://[[2001:db8:a::123]:2222"}  
3398       ]  
3399     }  
3400   ],  
3401   "ttl": 600  
3402 }
```

3403 When an RD receives this initial UPDATE request, it determines whether to grant the request or
3404 not. Upon granting the request, the RD shall send back an UPDATE response to the publishing
3405 Device. The response shall include a payload with the same information as the original UPDATE
3406 request with the following possible differences:

- 3407 – For each Link, an "ins" Parameter shall be included in the response. The RD shall assign a
3408 unique "ins" value identifying the Link among all the Links it advertises. If the publishing Device
3409 included an "ins" value in the UPDATE request, the RD may use it as long as it doesn't match
3410 any existing "ins" value in the published Links.
- 3411 – The "ttl" Property Value shall be assigned by the RD and it shall be included in the response.
3412 The RD should use the value included in the UPDATE request but may assign a value that is
3413 lower if it is not able to honour the requested "ttl" value. After this time elapses, the RD shall
3414 remove the Links. To keep a Link alive the publishing Device may update the "ttl" using the
3415 UPDATE schema.

3416 The RD shall add the new Links to its "/oic/res" and expose them to a valid discovery query, i.e.
3417 RETRIEVE request:

```
3418 {  
3419   "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",  
3420   "links": [  
3421     {  
3422       "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",  
3423       "href": "/myLightSwitch",  
3424       "rt": ["oic.r.switch.binary"],  
3425       "if": ["oic.if.a", "oic.if.baseline"],  
3426       "p": {"bm": 3},  
3427       "eps": [  

```

```

3428         {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
3429         {"ep": "coaps://[fe80::b1d6]:1122"},
3430         {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
3431     ],
3432     "ins": 11235
3433 },
3434 {
3435     "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
3436     "href": "/myLightBrightness",
3437     "rt": ["oic.r.brightness"],
3438     "if": ["oic.if.a", "oic.if.baseline"],
3439     "p": {"bm": 3},
3440     "eps": [
3441         {"ep": "coaps://[[2001:db8:a::123]:2222"}
3442     ],
3443     "ins": 112358
3444 }
3445 ].
3446 "ttl": 600
3447 }

```

3448 Once a publishing Device has published Resources to an RD, it may choose not respond to the
3449 multicast discovery queries for the same Resources against its own `"/oic/res"`, especially when on
3450 the same multicast domain as the RD. After publishing Resources, primarily it is the RDs
3451 responsibility to reply to the queries for the published Resources.

3452 There is another possibility that the RD and the publishing Device both respond to the multicast
3453 query from the discovering Device. This will create a duplication of the information but is an
3454 alternative that may be used for non-robust networks. It is not a recommended option but for
3455 industrial scenarios, this is one of the possibilities. Either way, discovering Clients need to always
3456 be prepared to process duplicate information in responses to multicast discovery request. The
3457 `"/oic/rd"` schema is as defined in Annex D to specify publishing to the `"/oic/rd"` Resource.

3458 11.3.6.4 Resource exposure

3459 11.3.6.4.1 `"/oic/res"` and retrieving of the Resources

3460 The `"/oic/res"` based discovery process remains the same as that in the absence of an RD.
3461 Resources may be discovered by retrieving the `"/oic/res"` Resource by sending a multicast or
3462 unicast request. In the case of a multicast discovery request, an RD shall include in its response
3463 any published Resources on behalf of the Device that hosts the Resources. Clients should be
3464 prepared to process duplicate Resource information from more than one RD responding with the
3465 same information or from an RD and the hosting Device (publishing the Resource information) both
3466 responding to the request. Interaction with Resources discovered using the RD is done using the
3467 same mechanism and methods as with Resources discovered by retrieving the `"/oic/res"` Resource
3468 of the Device hosting the Resources (e.g., connect to the hosting Device and perform CRUDN
3469 operations on the Resource).

3470 Resource Directories provide different `"/oic/res"` responses according to the requesting Clients,
3471 which indicate their preference with content format. OCF 1.0 Clients request with a Content Format
3472 of `"application/vnd.ocf+cbor"` in the Accept Option, whereas the Content-Format `"application/cbor"`
3473 in the Accept Option indicates OIC 1.1 Clients. For OIC 1.1 Clients, the `"/oic/res"` response includes
3474 Links conforming to OIC 1.1 specification, which OIC 1.1 Clients can understand. In this case the
3475 Resources hosted by the same Device shall be grouped together within a single JSON Object with
3476 `"di"` indicating the hosting Device. For a 3rd party Resource, i.e., a Resource which doesn't belong
3477 to the responding RD, its `"href"` value shall be a fully qualified transfer protocol URI with an IP
3478 address and port number as its authority component (e.g.,
3479 `coaps://[2001:db8:b::c2e5]:22222/myLightSwitch`).

3480 For example, an RD might return the following to an OIC 1.1 Clients:

```

3481 [
3482 {
3483   "di": "88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3484   "links": [
3485     {
3486       "href": "/oic/res",
3487       "rel": "self",
3488       "rt": ["oic.wk.res"],
3489       "if": ["oic.if.ll", "oic.if.baseline"],
3490       "p": {"bm": 3, "sec": false}
3491     },
3492     {
3493       "href": "/oic/d",
3494       "rt": ["oic.wk.d", "oic.d.fan"],
3495       "if": ["oic.if.r", "oic.if.baseline"],
3496       "p": {"bm": 3, "sec": false}
3497     },
3498     {
3499       "href": "/oic/p",
3500       "rt": ["oic.wk.p"],
3501       "if": ["oic.if.r", "oic.if.baseline"],
3502       "p": {"bm": 3, "sec": true, "port": 33333}
3503     },
3504     {
3505       "href": "/myFanIntrospection",
3506       "rt": ["oic.wk.introspection"],
3507       "if": ["oic.if.r", "oic.if.baseline"],
3508       "p": {"bm": 3, "sec": true, "port": 33333}
3509     },
3510     {
3511       "href": "/oic/rd",
3512       "rt": ["oic.wk.rd"],
3513       "if": ["oic.if.baseline"],
3514       "p": {"bm": 3, "sec": true, "port": 33333}
3515     },
3516     {
3517       "href": "/myFanSwitch",
3518       "rt": ["oic.r.switch.binary"],
3519       "if": ["oic.if.a", "oic.if.baseline"],
3520       "p": {"bm": 3, "sec": true, "port": 33333}
3521     },
3522     {
3523       "href": "/oic/sec/doxm",
3524       "rt": ["oic.r.doxm"],
3525       "if": ["oic.if.baseline"],
3526       "p": {"bm": 1, "sec": false}
3527     },
3528     {
3529       "href": "/oic/sec/pstat",
3530       "rt": ["oic.r.pstat"],
3531       "if": ["oic.if.baseline"],
3532       "p": {"bm": 1, "sec": true, "port": 33333}
3533     },
3534     {
3535       "href": "/oic/sec/cred",
3536       "rt": ["oic.r.cred"],
3537       "if": ["oic.if.baseline"],
3538       "p": {"bm": 1, "sec": true, "port": 33333}
3539     },
3540     {
3541       "href": "/oic/sec/acl2",
3542       "rt": ["oic.r.acl2"],
3543       "if": ["oic.if.baseline"],

```



```

3544         "p": {"bm": 1, "sec": true, "port": 33333}
3545     }
3546 ]
3547 },
3548 {
3549     "di": "dc70373c-1e8d-4fb3-962e-017eaa863989",
3550     "links": [
3551         {
3552             "href": "coap://[2001:db8:b::c2e5]:66666/oic/d",
3553             "rt": ["oic.wk.d", "oic.d.light", "oic.d.virtual"],
3554             "if": ["oic.if.r", "oic.if.baseline"],
3555             "p": {"bm": 3, "sec": false}
3556         },
3557         {
3558             "href": "coaps://[2001:db8:b::c2e5]:22222/oic/p",
3559             "rt": ["oic.wk.p"],
3560             "if": ["oic.if.r", "oic.if.baseline"],
3561             "p": {"bm": 3, "sec": true, "port": 22222}
3562         },
3563         {
3564             "href": "coaps://[2001:db8:b::c2e5]:22222/myLightSwitch",
3565             "rt": ["oic.r.switch.binary"],
3566             "if": ["oic.if.a", "oic.if.baseline"],
3567             "p": {"bm": 3, "sec": true, "port": 22222}
3568         },
3569         {
3570             "href": "coaps://[2001:db8:b::c2e5]:22222/myLightBrightness",
3571             "rt": ["oic.r.brightness"],
3572             "if": ["oic.if.a", "oic.if.baseline"],
3573             "p": {"bm": 3, "sec": true, "port": 22222}
3574         }
3575     ]
3576 }
3577 ]
3578

```

3579 For OCF 1.0 Clients, the "/oic/res" response includes the OCF 1.0 Links with the "anchor"
3580 Parameter containing an OCF URI. The "/oic/res" response has a single array of Links to conform
3581 to IETF RFC 6690. Each Link shall contain the "anchor" Parameter of the value OCF URI where
3582 the authority component of <deviceID> indicates the Device hosting the target Resource.

3583 For example, an RD may return the following to an OCF 1.0 Client.

```

3584 [
3585     {
3586         "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3587         "href": "/oic/res",
3588         "rel": "self",
3589         "rt": ["oic.wk.res"],
3590         "if": ["oic.if.ll", "oic.if.baseline"],
3591         "p": {"bm": 3},
3592         "eps": [
3593             {"ep": "coap://[2001:db8:a::b1d4]:77777"},
3594             {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3595         ]
3596     },
3597     {
3598         "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3599         "href": "/oic/d",
3600         "rt": ["oic.wk.d", "oic.d.fan"],
3601         "if": ["oic.if.r", "oic.if.baseline"],
3602         "p": {"bm": 3},
3603         "eps": [

```

```

3604         {"ep": "coap://[2001:db8:a::b1d4]:77777"},
3605         {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3606     ]
3607 },
3608 {
3609     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3610     "href": "/oic/p",
3611     "rt": ["oic.wk.p"],
3612     "if": ["oic.if.r", "oic.if.baseline"],
3613     "p": {"bm": 3},
3614     "eps": [
3615         {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3616     ]
3617 },
3618 {
3619     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3620     "href": "/myFanIntrospection",
3621     "rt": ["oic.wk.introspection"],
3622     "if": ["oic.if.r", "oic.if.baseline"],
3623     "p": {"bm": 3},
3624     "eps": [
3625         {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3626     ]
3627 },
3628 {
3629     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3630     "href": "/oic/rd",
3631     "rt": ["oic.wk.rd"],
3632     "if": ["oic.if.baseline"],
3633     "p": {"bm": 3},
3634     "eps": [
3635         {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3636     ]
3637 },
3638 {
3639     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3640     "href": "/myFanSwitch",
3641     "rt": ["oic.r.switch.binary"],
3642     "if": ["oic.if.a", "oic.if.baseline"],
3643     "p": {"bm": 3},
3644     "eps": [
3645         {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3646     ]
3647 },
3648 {
3649     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3650     "href": "/oic/sec/doxm",
3651     "rt": ["oic.r.doxm"],
3652     "if": ["oic.if.baseline"],
3653     "p": {"bm": 1},
3654     "eps": [
3655         {"ep": "coap://[2001:db8:a::b1d4]:77777"},
3656         {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3657     ]
3658 },
3659 {
3660     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3661     "href": "/oic/sec/pstat",
3662     "rt": ["oic.r.pstat"],
3663     "if": ["oic.if.baseline"],
3664     "p": {"bm": 1},
3665     "eps": [
3666         {"ep": "coaps://[2001:db8:a::b1d4]:33333"}

```

```

3667     ]
3668   },
3669   {
3670     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3671     "href": "/oic/sec/cred",
3672     "rt": ["oic.r.cred"],
3673     "if": ["oic.if.baseline"],
3674     "p": {"bm": 1},
3675     "eps": [
3676       {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3677     ]
3678   },
3679   {
3680     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3681     "href": "/oic/sec/acl2",
3682     "rt": ["oic.r.acl2"],
3683     "if": ["oic.if.baseline"],
3684     "p": {"bm": 1},
3685     "eps": [
3686       {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3687     ]
3688   },
3689   {
3690     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
3691     "href": "/oic/d",
3692     "rt": ["oic.wk.d", "oic.d.light"],
3693     "if": ["oic.if.r", "oic.if.baseline"],
3694     "p": {"bm": 3},
3695     "eps": [
3696       {"ep": "coap://[2001:db8:b::c2e5]:66666"},
3697       {"ep": "coaps://[2001:db8:b::c2e5]:22222"}
3698     ]
3699   },
3700   {
3701     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
3702     "href": "/oic/p",
3703     "rt": ["oic.wk.p"],
3704     "if": ["oic.if.r", "oic.if.baseline"],
3705     "p": {"bm": 3},
3706     "eps": [
3707       {"ep": "coaps://[2001:db8:b::c2e5]:22222"}
3708     ]
3709   },
3710   {
3711     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
3712     "href": "/myLightSwitch",
3713     "rt": ["oic.r.switch.binary"],
3714     "if": ["oic.if.a", "oic.if.baseline"],
3715     "p": {"bm": 3},
3716     "eps": [
3717       {"ep": "coaps://[2001:db8:b::c2e5]:22222"}
3718     ]
3719   },
3720   {
3721     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
3722     "href": "/myLightBrightness",
3723     "rt": ["oic.r.brightness"],
3724     "if": ["oic.if.a", "oic.if.baseline"],
3725     "p": {"bm": 3},
3726     "eps": [
3727       {"ep": "coaps://[2001:db8:b::c2e5]:22222"}
3728     ]
3729   }

```

3730]

3731 11.4 Notification

3732 11.4.1 Overview

3733 A Server shall support NOTIFY operation to enable a Client to request and be notified of desired
3734 states of one or more Resources in an asynchronous manner. 11.4.2 specifies the Observe
3735 mechanism in which updates are delivered to the requester.

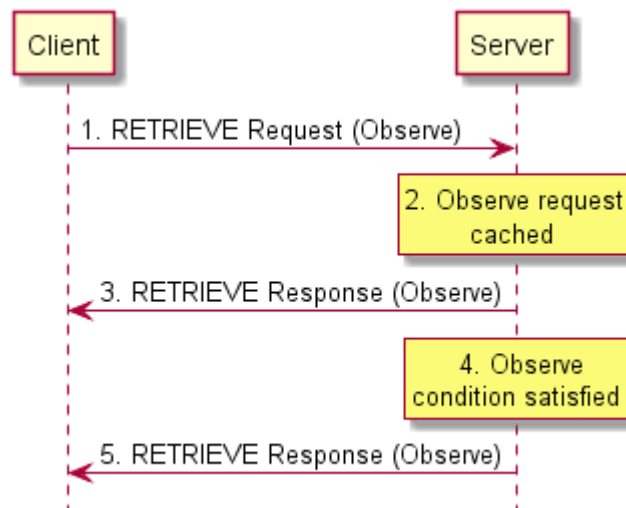
3736 11.4.2 Observe

3737 11.4.2.1 Overview

3738 In the Observe mechanism the Client utilizes the RETRIEVE operation to require the Server for
3739 updates in case of Resource state changes. The Observe mechanism consists of five steps which
3740 are depicted in Figure 20.

3741 NOTE the Observe mechanism can only be used for a resource with a Property of Observable (see 7.3.2.2).

3742



3743

3744

3745

Figure 20 – Observe Mechanism

3746 11.4.2.2 RETRIEVE request with Observe indication

3747 The Client transmits a RETRIEVE request message to the Server to request updates for the
3748 Resource on the Server if there is a state change. The RETRIEVE request message carries the
3749 following parameters:

- 3750 – *fr*: Unique identifier of the Client.
- 3751 – *to*: Resource that the Client is requesting to Observe.
- 3752 – *ri*: Identifier of the RETRIEVE operation.
- 3753 – *op*: RETRIEVE.
- 3754 – *obs*: Indication for Observe operation.

3755 11.4.2.3 Processing by the Server

3756 Following the receipt of the RETRIEVE request, the Server may validate if the Client has the
3757 appropriate rights for the requested operation and the Properties are readable and Observable. If

3758 the validation is successful, the Server caches the information related to the Observe request. The
3759 Server caches the value of the *ri* parameter from the RETRIEVE request for use in the initial
3760 response and future responses in case of a change of state.

3761 **11.4.2.4 RETRIEVE response with Observe indication**

3762 The Server shall transmit a RETRIEVE response message in response to a RETRIEVE request
3763 message from a Client. The RETRIEVE response message shall include the following parameters.
3764 If validation succeeded, the response includes an Observe indication. If not, the Observe indication
3765 is omitted from the response which signals to the requesting Client that registration for notification
3766 was not allowed.

3767 The RETRIEVE response message shall include the following parameters:

- 3768 – *fr*: Unique identifier of the Server.
- 3769 – *to*: Unique identifier of the Client.
- 3770 – *ri*: Identifier included in the RETRIEVE operation.
- 3771 – *cn*: Information Resource representation as requested by the Client.
- 3772 – *rs*: The result of the RETRIEVE operation.
- 3773 – *obs*: Indication that the response is made to an Observe operation.

3774 **11.4.2.5 Resource monitoring by the Server**

3775 The Server shall monitor the state the Resource identified in the Observe request from the Client.
3776 Anytime there is a change in the state of the Observed Resource, the Server sends another
3777 RETRIEVE response with the Observe indication. The mechanism does not allow the client to
3778 specify any bounds or limits which trigger a notification, the decision is left entirely to the server.

3779 **11.4.2.6 Additional RETRIEVE responses with Observe indication**

3780 The Server shall transmit updated RETRIEVE response messages following Observed changes in
3781 the state of the Resources indicated by the Client. The RETRIEVE response message shall include
3782 the parameters listed in 11.4.2.4.

3783 **11.4.2.7 Cancelling Observe**

3784 The Client can explicitly cancel Observe by sending a RETRIEVE request without the Observe
3785 indication field to the same Resource on the Server which it was Observing. For certain protocol
3786 mappings, the Client may also be able to cancel an Observe by ceasing to respond to the
3787 RETRIEVE responses.

3788 **11.5 Device management**

3789 **11.5.1 Overview**

3790 The Device management includes the following functions:

- 3791 – Diagnostics and maintenance.

3792 The Device management functionalities specified in this version of document are intended to
3793 address the basic Device management features. Addition of new Device management features in
3794 the future versions of the document is expected.

3795 **11.5.2 Diagnostics and maintenance**

3796 The Diagnostics and Maintenance function is intended for use by administrators to resolve issues
3797 encountered with the Devices while operating in the field. If diagnostics and maintenance is
3798 supported by a Device, the Core Resource "/oic/mnt" shall be supported as described in Table 31.

3799 **Table 31 – Optional diagnostics and maintenance Device management Core Resources**

Pre-defined URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
"/oic/mnt"	Maintenance	"oic.wk.mnt"	"oic.if.rw"	The Resource through which the Device is maintained and can be used for diagnostic purposes. The Properties exposed by "/oic/mnt" are listed in Table 32.	Device management

3800 Table 32 defines the "oic.wk.mnt" Resource Type. At least one of the Factory Reset, Reboot or last
3801 error Properties shall be implemented.

3802 **Table 32 – "oic.wk.mnt" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Factory_Reset	"fr"	"boolean"	N/A	N/A	R, W	No	When writing to this Property: false – No action (Default*) true – Start Factory Reset After factory reset all configuration and state data will be lost. When reading this Property, a value of true indicates a pending factory reset. Once the factory reset has been completed, the Device shall set the value back to false.
Reboot	"rb"	"boolean"	N/A	N/A	R, W	No	When writing to this Property: false – No action (Default) true – Start Reboot After Reboot, this value shall be changed back to the default value (i.e., false)
Last error	"err"	"integer"	HTTP error code	N/A	R	No	Last occurred error code, shall be cleared to 503 (service unavailable), when doing a Factory Reset or Reboot. All HTTP errors outside the 100, 200 or 300 range shall be stored.

3803
3804 NOTE Default indicates the value of this Property as soon as the Device is rebooted or factory reset.

3805 **11.5.3 Network monitoring**

3806 Network monitoring is used for monitoring the current network state of the Device.

3807 The network monitoring Resource Type is "oic.wk.nmon" and is described in Table 33. The
3808 Resource Type may occur multiple times if more than 1 network interface is implemented. The
3809 Common Property "n" may be used to distinguish the different network interfaces, like distinguishing
3810 the 2.4 and 5G Wi-Fi network interfaces.

3811

Table 33 – Optional monitoring Device management Core Resources

Example URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
"/example/oic/nmon"	Network Monitoring	"oic.wk.nmon"	"oic.if.rw oic.if.baseline"	The Resource through which the Device is monitored. The Resource exposes Properties relevant to aspects that may be monitored. The Resource Properties exposed by Resource Type "oic.wk.nmon" are listed in Table 34	Device management

3812

3813 Table 34 defines "oic.wk.nmon" Resource Type.

3814

Table 34 – "oic.wk.nmon" Resource Type definition

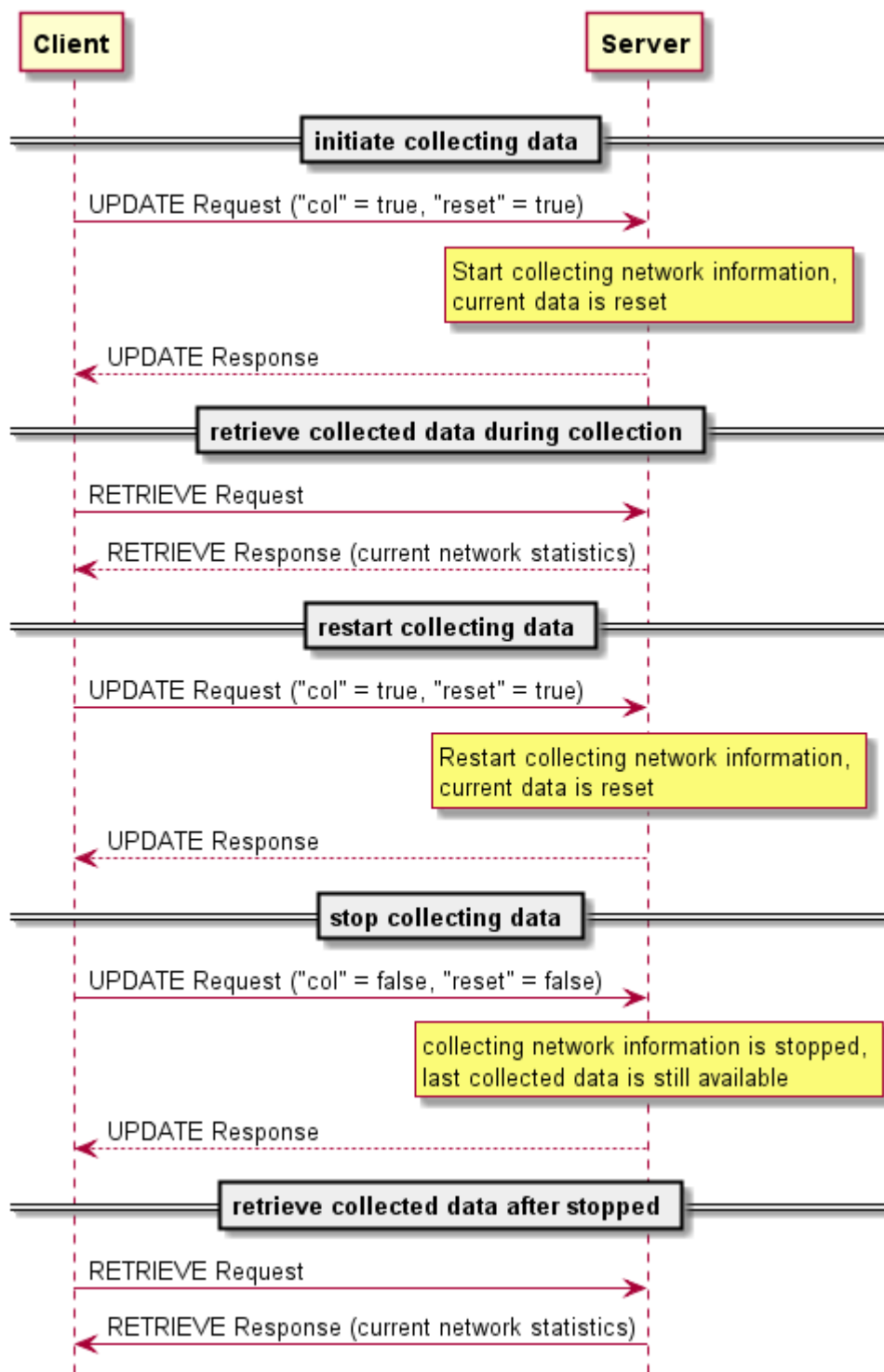
Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Network indicator	"ianaifType"	"integer"	The integer value of the ianaifType	N/A	R	Yes	The network type this Resource is collecting information from as defined by: https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib
reset	"reset"	"boolean"	True, all collected values should be reset. The server should reset the value automatically to false after the reset occurred.	N/A	RW	Yes	Reset of the collected values
Collecting status indication	"col"	"boolean"	True: collecting data. False: not collecting data	N/A	RW	Yes	Boolean to start/stop collecting data.
Transmission bytes	"tx"	"integer"	N/A	kilo bytes	R	No	Amount of transmitted kilo bytes from the collection
Reception bytes	"rx"	"integer"	N/A	kilo bytes	R	No	Amount of received kilo bytes from the collection.
Maximum message size tx	"mmstx"	"integer"	bytes	bytes	R	No	Maximum transmitted message, e.g. Max(tx) in the collection period

Maximum message size -rx	"mmsrx"	"integer"	bytes	bytes	R	No	Maximum received message, e.g. Max(rx) in the collection period
Average message size -tx	"amstx"	"integer"	bytes	bytes	R	No	Average transmitted message size, e.g. AVG(tx) in the collection period.
Average message size -rx	"amsrx"	"integer"	bytes	bytes	R	No	Average received message size e.g. AVT(rx) in the collection period.

3815

3816 Examples of typical used values for ianaifType are 71 (ieee80211) for Wi-Fi and 6 (ethernetCsmacd)
 3817 for Ethernet.

3818 A Device should start collecting network monitoring data when receiving an UPDATE operation
 3819 with the parameter "col" = true. A Device should stop collecting network data when receiving an
 3820 UPDATE operation with parameter "col" = false. The collected network data should be reset when
 3821 an UPDATE operation with parameter "reset" = true is received, if the parameter "reset" is false
 3822 then the values should not be reset. Figure 21 illustrates the interactions with the network
 3823 monitoring Resource.

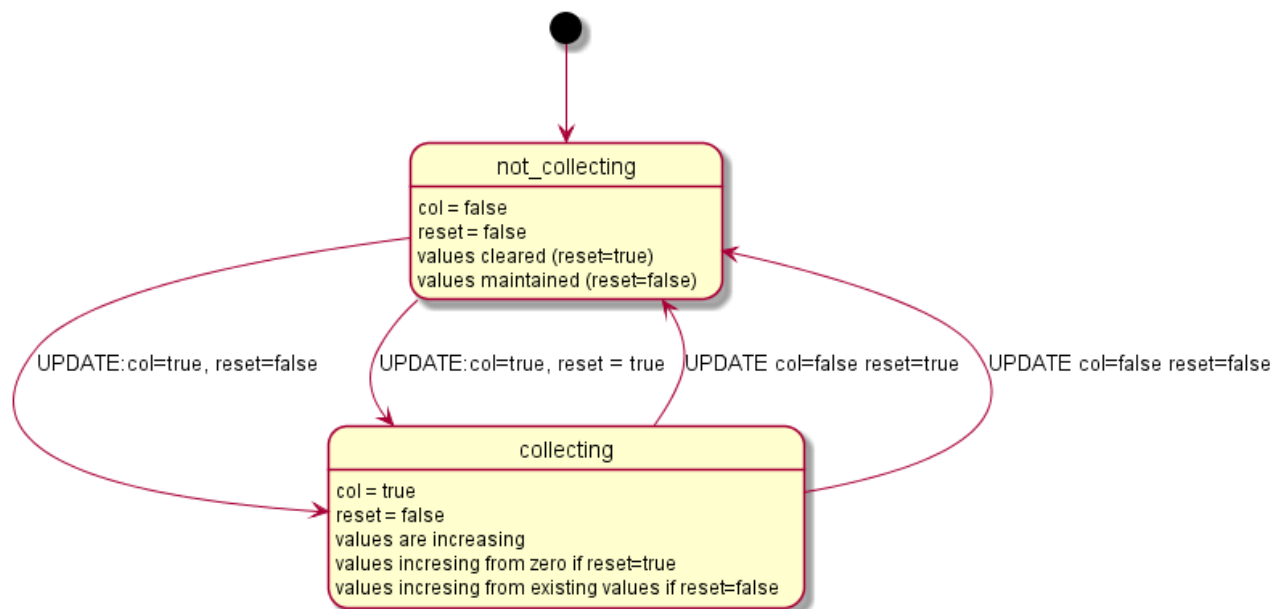


3824

3825

Figure 21 – Interactions with the network monitoring Resource

3826 The state transition diagram for collecting or not collecting network information is described by
 3827 Figure 22.



3828

3829 **Figure 22 – State transition diagram of collecting network information**

3830 **11.6 Scenes**

3831 **11.6.1 Introduction**

3832 Scenes are a mechanism for automating certain operations.

3833 A Scene is a static entity that stores a set of defined Property values for a Collection of Resources.
 3834 Scenes provide a mechanism to store a setting over multiple Resources that may be hosted by
 3835 multiple separate Servers. Scenes, once set up, can be used by multiple Clients to recall a setup.

3836 Scenes can be grouped and reused, a group of Scenes is also a Scene.

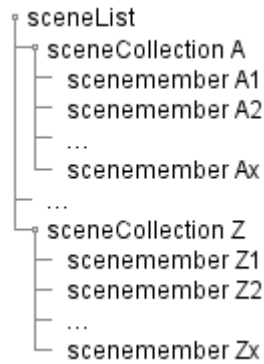
3837 In short, Scenes are bundled user settings.

3838 **11.6.2 Scenes**

3839 **11.6.2.1 Introduction**

3840 Scenes are described by means of Resources. The Scene Resources are hosted by a Server and
 3841 the top level Resource is listed in "/oic/res". This means that a Client can determine if the Scene
 3842 functionality is hosted on a Server via Resource discovery as defined in 11.3. The setup of Scenes
 3843 is driven by Client interactions. This includes creating new Scenes, and mappings of Server
 3844 Properties that are part of a Scene.

3845 The Scene functionality is created by multiple Resources and has the structure depicted in
 3846 Figure 23. The sceneList and sceneCollection Resources are overloaded Collection Resources.
 3847 The sceneCollection Resource contains a list of Scenes. This list contains zero or more Scenes.
 3848 The sceneMember Resource contains the mapping between a Scene and what needs to happen
 3849 according to that Scene on an indicated Resource.



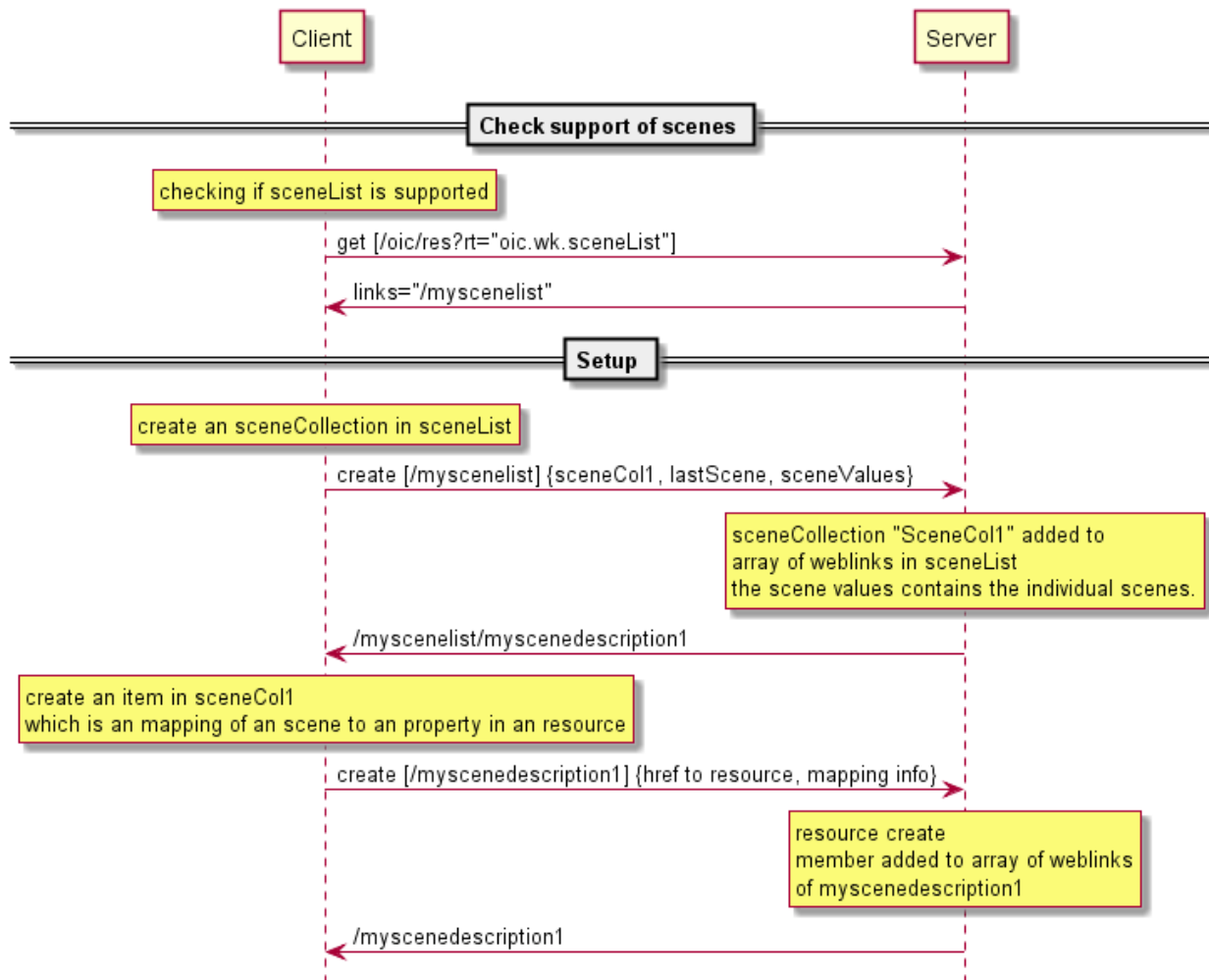
3850

3851

Figure 23 – Generic Scene Resource structure

3852 **11.6.2.2 Scene creation**

3853 A Client desiring to interact with Scenes needs to first determine if the Server supports the Scene
 3854 feature; the sceneMembers of a Scene that are Resources of end Device being updated by the
 3855 Scene change do not have to be co-located on the Server supporting the Scene feature. This can
 3856 be done by checking if "/oic/res" contains the "rt" of the sceneList Resource. This is depicted in
 3857 first steps of Figure 24. The sceneCollection Resource is created by the Server using some out of
 3858 bound mechanism, Client creation of Scenes is not supported at this time. This will entail defining
 3859 the Scene with an applicable list of Scene Values and the mappings for each Resource being part
 3860 of the Scene. The mapping for each Resource being part of the sceneCollection Resource is
 3861 described by a Resource called sceneMember. The sceneMember Resource contains the link to a
 3862 Resource and the mapping between the Scene listed in the "sceneValues" Property and the actual
 3863 Property value of the Resource indicated by the Link.



3864

3865

Figure 24 – Interactions to check Scene support and setup of specific Scenes

3866

11.6.2.3 Interacting with Scenes

3867

3868

3869

3870

3871

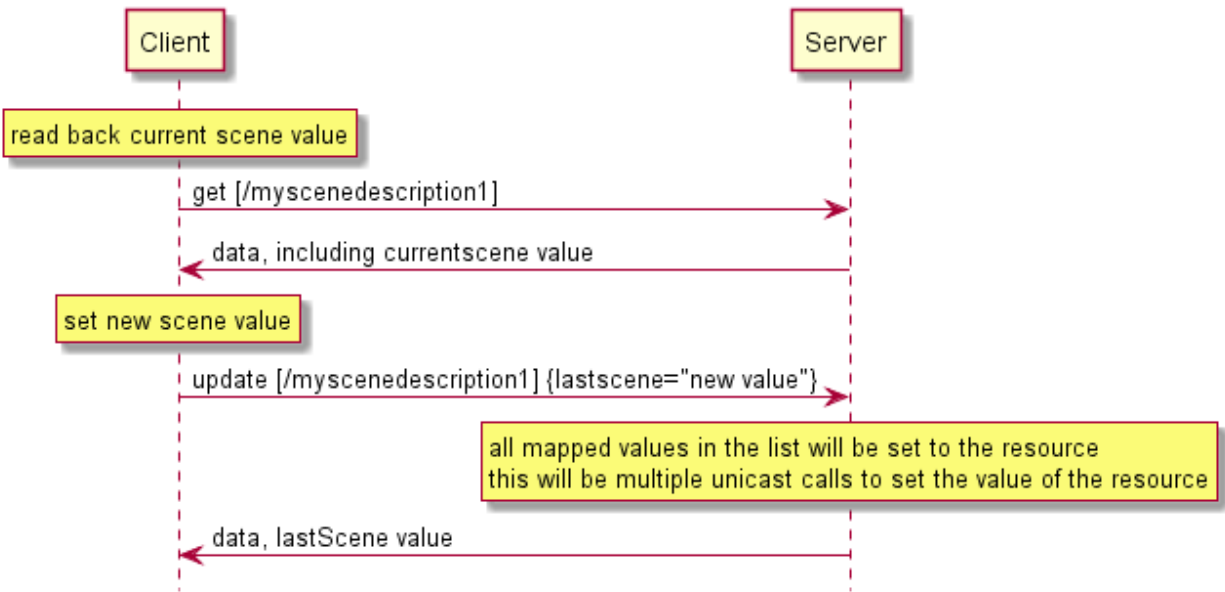
3872

3873

3874

All capable Clients can interact with Scenes. The allowed Scene Values and the last applied Scene Value can be retrieved from the Server hosting the Scene. The Scene Value shall be changed by issuing an UPDATE operation with a payload that sets the "lastScene" Property to one of the listed allowed Scene Values. These steps are depicted in Figure 25. Note that the "lastScene" Property value does not imply that the current state of all Resources that are part of the Scene will be at the mapped value. This is due to that the setting the Scene Values are not modelled as actual states of the system. This means that another Client can change just one Resource being part of the Scene without having feedback that the state of the Scene is changed.

3875

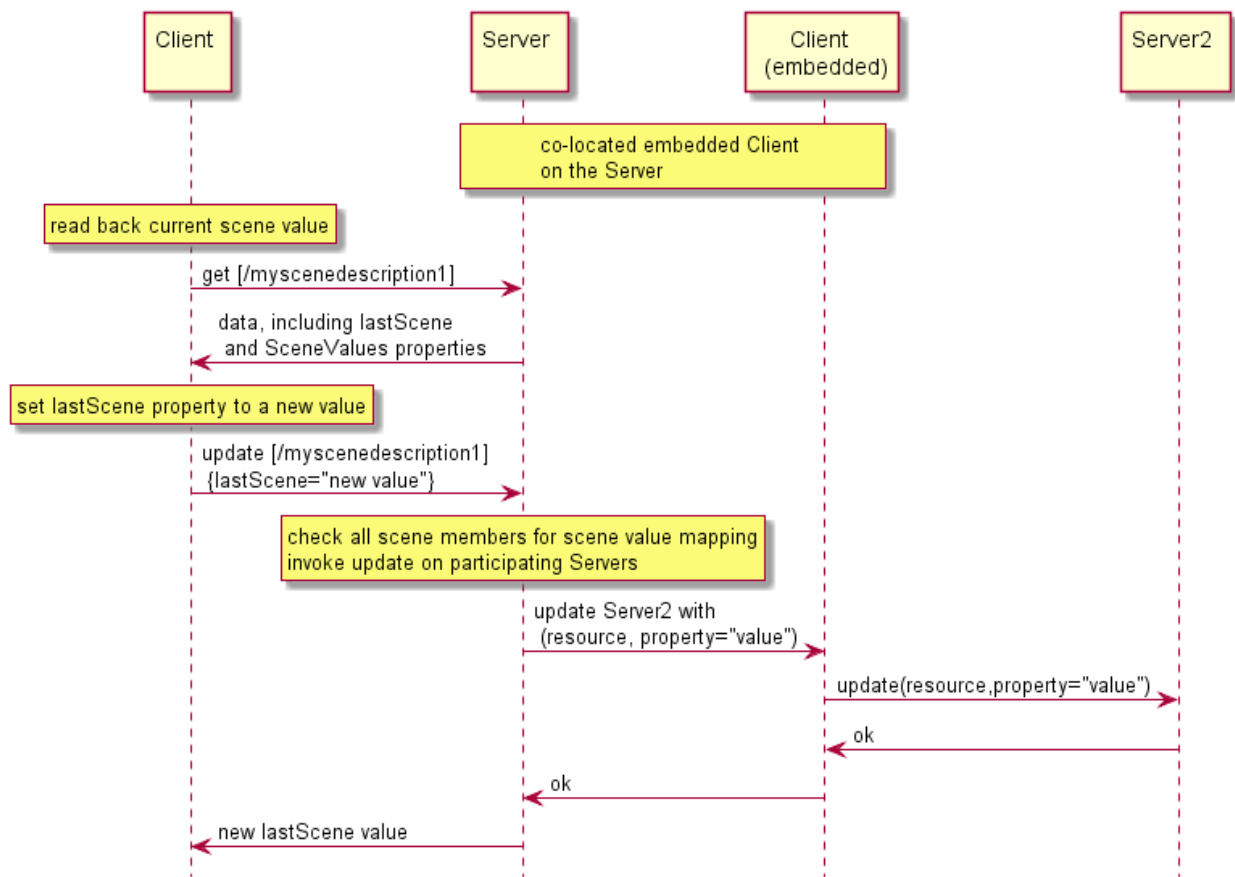


3876

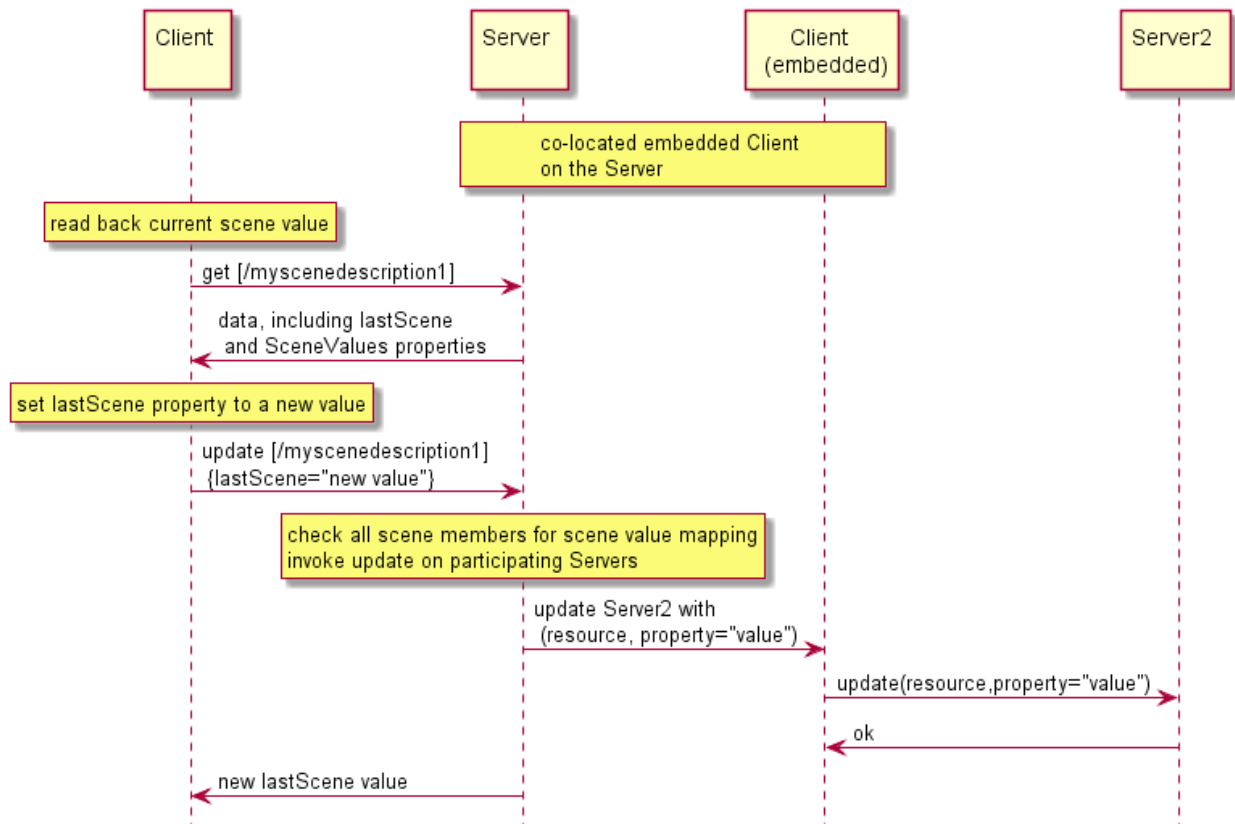
3877

Figure 25 – Client interactions on a specific Scene

3878 As described previously, a Scene can reference one or more Resources (i.e., sceneMembers) that
 3879 are present on one or more Servers. The Scene Members are re-evaluated each time a Scene
 3880 change takes place. This evaluation is triggered by a Client that is either embedded as part of the
 3881 Server hosting the Scene, or separate to the Server having knowledge of the Scene via a
 3882 RETRIEVE operation, Observing the referenced Resources using the mechanism described in
 3883 11.4.2. The embedded Client located in the same Device with the Server is a general Client but
 3884 interacts only with Scene functionalities. During the evaluation the mappings for the new Scene
 3885 Value will be applied to the Servers which contain sceneMembers from the Scene that is being
 3886 updated. This behaviour is depicted in Figure 26.



3887



3888

3889

Figure 26 – Interaction overview due to a Scene change

3890

11.6.2.4 Summary of Resource Types defined for Scene functionality

3891

Table 35 summarizes the list of Resource Types that are part of Scenes.

3892

Table 35 – list of Resource Types for Scenes

Friendly Name (informative)	Resource Type (rt)	Short Description	Clause
sceneList	"oic.wk.scenelist"	Top Level Collection containing sceneCollections	N/A
sceneCollection	"oic.wk.scenecollection"	Description of zero or more scenes	N/A
sceneMember	"oic.wk.scenemember"	Description of mappings for each specific Resource part of the sceneCollection	N/A

3893

3894

11.6.3 Security considerations

3895

Creation of Scenes on a Server that is capable of this functionality is dependent on the ACLs applied to the Resources and the Client having the appropriate permissions. Interaction between a Client (embedded or separate) and a Server that hosts the Resource that is referenced as a Scene Member is contingent on the Client having appropriate permissions to access the Resource on the host Server.

3896

3897

3898

3899

3900 See ISO/IEC 30118-2:2018 for details on the use of ACLs and also the mechanisms around Device
 3901 Authentication that are necessary to ensure that the correct permissions exist for the Client to
 3902 access the Scene Member Resource(s) on the Server.

3903 **11.7 Icons**

3904 **11.7.1 Overview**

3905 Icons are a primitive that are needed by various OCF subsystems, such as bridging. An optional
 3906 Resource Type of "oic.r.icon" has been defined to provide a common representation of an icon
 3907 Resource that can be used by Devices.

3908 **11.7.2 Resource**

3909 The icon Resource is as defined in Table 36.

3910 **Table 36 – Optional Icon Core Resource**

Example URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
"/example/oic/icon"	Icon	"oic.r.icon"	"oic.if.r"	The Resource through which the Device can obtain icon images. The Properties exposed by "/example/oic/mnt" are listed in Table 37.	Icon

3911

3912 Table 37 defines the details for the "oic.r.icon" Resource Type.

3913 **Table 37 – "oic.r.icon" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Mime Type	"mimetype"	"string"	N/A	N/A	R	Yes	Specifies the format (media type) of the icon. It should be a template string as specified in IANA Media Types Assignment
Width	"width"	"integer"	>= 1	pixels	R	Yes	Width of the icon in pixels greater than or equal to 1.
Height	"height"	"integer"	>= 1	pixels	R	Yes	Height of the icon in pixels greater than or equal to 1.
Icon	"media"	"uri"	N/A	N/A	R	Yes	URI to the location of the icon image.

3914

3915 **11.8 Introspection**

3916 **11.8.1 Overview**

3917 Introspection is a mechanism to announce the capabilities of Resources hosted on the Device.

3918 The intended usage of the Introspection Device Data (IDD) is to enable dynamic Clients e.g. Clients
 3919 that can use the IDD) to generate dynamically a UI or dynamically create translations of the hosted
 3920 Resources to another eco-system. Other usages of Introspection is that the information can be
 3921 used to generate Client code. The IDD is designed to augment the existing data already on the

3922 wire. This means that existing mechanisms need to be used to get a full overview of what is
3923 implemented in the Device. For example, the IDD does not convey information about Observability,
3924 since that is already conveyed with the "p" Property on the Links in "/oic/res" (see 7.8.2.2.2).

3925 The IDD is recommended to be conveyed as static data. Meaning that the data does not change
3926 during the uptime of a Device. However, when the IDD is not static, the Introspection Resource
3927 shall be Observable and the url Property Value of "oic.wk.introspection" Resource shall change to
3928 indicate that the IDD is changed.

3929 The IDD describes the Resources that make up the Device. For the complete list of included
3930 Resources Table 20. The IDD is described as a OpenAPI 2.0 in JSON format file. The text in the
3931 following bulleted list contains OpenAPI 2.0 terms, such as paths, methods etc. The OpenAPI 2.0
3932 file shall contain the description of the Resources:

- 3933 – The IDD will use the HTTP syntax, e.g., define the CRUDN operation as HTTP methods and
3934 use the HTTP status codes.
- 3935 – The IDD does not have to define all the status codes that indicate an error situation.
- 3936 – The IDD does not have to define a schema when the status code indicates that there is no
3937 payload (see HTTP status code 204 as an example).
- 3938 – The paths (URLs) of the Resources in the IDD shall be without the OCF Endpoint description,
3939 e.g. it shall not be a fully-qualified URL but only the relative path from the OCF Endpoint, aka
3940 the "href". The relative path may include a query parameter (e.g. "?if=oic.if.ll"), in such cases
3941 the text following (and including) the "?" delimiter shall be removed before equating to the "href"
3942 that is conveyed by "/oic/res".
- 3943 – The following Resources shall be excluded in the IDD:
 - 3944 – Resource with Resource Type: "oic.wk.res" unless 3rd party defined or optional Properties
3945 are implemented.
 - 3946 – Resource with Resource Type: "oic.wk.introspection".
 - 3947 – Resources that handle Wi-Fi Easy Setup, see OCF Easy Wi-Fi Setup.
- 3948 – The following Resources shall be included in the IDD when optional or 3rd party defined
3949 Properties are implemented:
 - 3950 – Resources with type: "oic.wk.p" and "oic.wk.d" (e.g. discovery related Resources).
 - 3951 – Security Virtual Resources from ISO/IEC 30118-2:2018.
- 3952 – When the Device does not expose instances of Vertical Resource Types, and does not have
3953 any 3rd party defined Resources (see 7.8.4.4), and does not need to include Resources in the
3954 IDD due to other clauses in this clause, then the IDD shall be an empty OpenAPI 2.0 file. An
3955 example of an empty OpenAPI 2.0 file can be found in found in Annex F.2.
- 3956 – All other Resources that are individually addressable by a Client (i.e. the "href" can be resolved
3957 and at least one operation is supported with a success path response) shall be listed in the IDD.
- 3958 – Per Resource the IDD shall include:
 - 3959 – All implemented methods
 - 3960 – For an OCF defined Resource Type, only the methods that are listed in the OpenAPI 2.0
3961 definition are allowed to exist in the IDD. For an OCF defined Resource Type, methods
3962 not listed in the OpenAPI 2.0 definition shall not exist in the IDD. The supported methods
3963 contained in the IDD shall comply with the listed OCF Interfaces. For example, if the
3964 POST method is listed in the IDD, then an OCF Interface that allows UPDATE will be
3965 listed in the IDD.
 - 3966 – Per supported method:
 - 3967 – Implemented query parameters per method.

- 3968 – This includes the supported OCF Interfaces ("if") as enum values.
- 3969 – Schemas of the payload for the request and response bodies of the method.
- 3970 – Where the schema provides the representation of a batch request or response ("oic.if.b")
- 3971 the schema shall contain the representations for all Resource Types that may be
- 3972 included within the batch representation. The representations shall be provided within
- 3973 the IDD itself.
- 3974 – The schema data shall be conveyed by the OpenAPI 2.0 schema.
- 3975 – The OpenAPI 2.0 schema object shall comply with:
 - 3976 – The schemas shall be fully resolved, e.g. no references shall exist outside the
 - 3977 OpenAPI 2.0 file.
 - 3978 – The schemas shall list which OCF Interfaces are supported on the method.
 - 3979 – The schemas shall list if a Property is optional or required.
 - 3980 – The schemas shall include all Property validation keywords. Where an enum is
 - 3981 defined the enum shall contain the values supported by the Device. When vendor
 - 3982 defined extensions exist to the enum (defined in accordance to 7.8.4.4) these shall
 - 3983 be included in the enum.
 - 3984 – The schemas shall indicate if an Property is read only or read-write.
 - 3985 – By means of the readOnly schema tag belonging to the Property.
 - 3986 – Default value of readOnly is false as defined by OpenAPI 2.0.
 - 3987 – The default value of the "rt" Property shall be used to indicate the supported
 - 3988 Resource Types.
 - 3989 – oneOf and anyOf constructs are allowed to be used as part of a OpenAPI 2.0 schema
 - 3990 object. The OpenAPI 2.0 schema with oneOf and anyOf constructs can be found in
 - 3991 Annex F.1.
- 3992 – For Atomic Measurements (see clause 7.8.4), the following apply:
 - 3993 – The "rts" Property Value in the IDD shall include only the Resource Types the instance
 - 3994 contains and not the theoretical maximal set allowed by the schema definition.
 - 3995 – The Resources that are part of an Atomic Measurement, excluding the Atomic Measurement
 - 3996 Resource itself, shall not be added to their own individual path in the IDD, as they are not
 - 3997 individually addressable; however, the schemas for the composed Resource Types shall be
 - 3998 provided in the IDD as part of the batch response definition along with the "href" for the
 - 3999 Resource.

4000 Dynamic Resources (e.g. Resources that can be created on a request by a Client) shall have a
 4001 URL definition which contains a URL identifier (e.g. using the {} syntax). A URL with {} identifies
 4002 that the Resource definition applies to the whole group of Resources that may be created. The
 4003 actual path may contain the Collection node that links to the Resource.

4004 Example of a URL with identifiers:

4005 /SceneListResURI/{SceneCollectionResURI}/{SceneMemberResURI}:

4006 When different Resource Types are allowed to be created in a Collection, then the different
 4007 schemas for the CREATE method shall define all possible Resource Types that may be created.
 4008 The schema construct oneOf allows the definition of a schema with selectable Resources. The
 4009 oneOf construct allows the integration of all schemas and that only one existing sub schema shall
 4010 be used to indicate the definition of the Resource that may be created.

4011 Example usage of oneOf JSON schema construct is shown in Figure 27:

4012 {

```

4013     "oneOf": [
4014         { <<subschema 1 definition>> },
4015         { << sub schema 2 definition >> }
4016     ...
4017     ]
4018 }

```

4019 **Figure 27 – Example usage of oneOf JSON schema**

4020 A Client using the IDD of a Device should check the version of the supported IDD of the Device.
4021 The OpenAPI 2.0 version is indicated in each file with the tag "swagger". Example of the 2.0
4022 supported version of the tag is: "swagger": "2.0". Later versions of the specification may reference
4023 newer versions of the OpenAPI specification, for example 3.0.

4024 A Device shall support one Resource with a Resource Type of "oic.wk.introspection" as defined in
4025 Table 38. The Resource with a Resource Type of "oic.wk.introspection" shall be included in the
4026 Resource "/oic/res".

4027 An empty IDD file, e.g. no URLs are exposed, shall still have the mandatory OpenAPI 2.0 fields.
4028 See OpenAPI specification. An example of an empty OpenAPI 2.0 file can be found in found in
4029 Annex F.2.

4030 **Table 38 – Introspection Resource**

Pre-defined URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
none	Introspection	"oic.wk.introspection"	"oic.if.r"	The Resource that announces the URL of the Introspection file.	Introspection

4031 Table 39 defines "oic.wk.introspection" Resource Type.

4032 **Table 39 – "oic.wk.introspection" Resource Type definition**

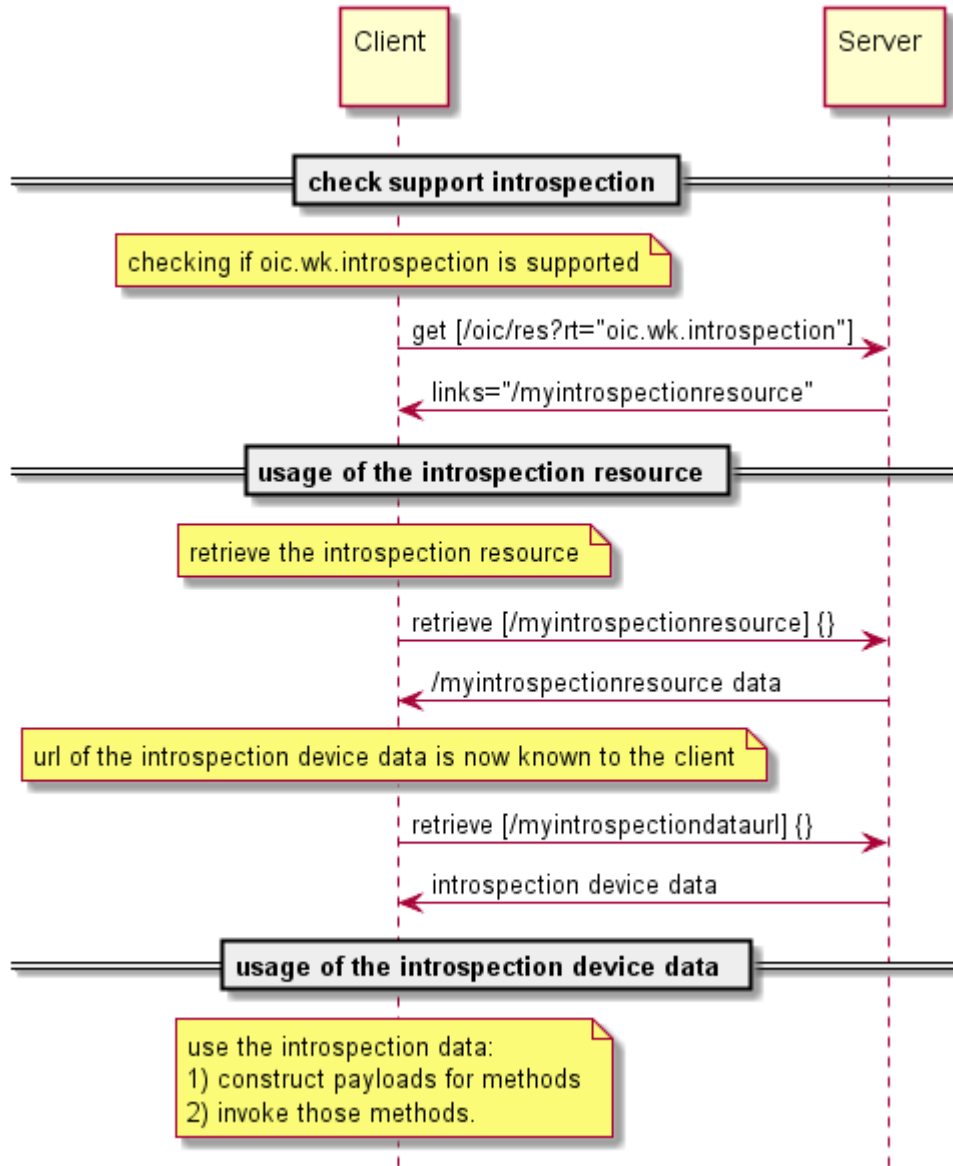
Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
urlInfo	"urlInfo"	"array"	N/A	N/A	R	Yes	array of objects
url	"url"	"string"	"uri"	N/A	R	Yes	URL to the hosted payload
protocol	"protocol"	"string"	"enum"	N/A	R	Yes	Protocol definition to retrieve the Introspection Device Data from the url.
content-type	"content-type"	"string"	"enum"	N/A	R	No	content type of the url.
version	"version"	"integer"	"enum"	N/A	R	No	Version of the Introspection protocol, indicates which rules are applied on the Introspection Device Data regarding the content of the OpenAPI 2.0 file. Current value is 1.

4034 **11.8.2 Usage of Introspection**

4035 The Introspection Device Data is retrieved in the following steps and as depicted in Figure 28:

- 4036 – Check if the Introspection Resource is supported and retrieve the URL of the Resource.

- 4038 – Retrieve the contents of the Introspection Resource
- 4039 – Download the Introspection Device Data from the URL specified the Introspection Resource.
- 4040 – Usage of the Introspection Device Data by the Client
- 4041



4042
 4043 **Figure 28 – Interactions to check Introspection support and download the Introspection**
 4044 **Device Data.**

4045 **11.9 Alerts**

4046 **11.9.1 Overview**

4047 Alerts provide a means by which a Device provides information to an interested party with regard
 4048 to error or other conditions that the Device is experiencing. An Alert contains human readable text
 4049 that is dependent on the Device itself and the condition being reported. A Device may expose
 4050 discrete instances of an Alert Resource Type ("oic.r.alert") or may also expose Alerts within an

4051 Alert Collection ("oic.r.alertcollection"). If the instance of "oic.r.alertcollection" is Observable (see
 4052 clause 7.8.2.2.2) then a Client may Observe the Collection using the mechanisms defined in clause
 4053 11.4. As the Device adds and removes Alerts from the Collection notifications may be generated
 4054 for any registered Observers, the format of which is dependent upon the OCF Interface used for
 4055 the initial Observe, see clause 7.6.3.

4056 11.9.2 Resource Types

4057 The Alert and Alert Collection Resource Types are as defined in Table 40.

4058 **Table 40 – Optional Alert Core Resources**

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
"/example/alertURI"	Alert	"oic.r.alert"	"oic.if.r", "oic.if.baseline"	The Resource through which the Device exposes Alerts. The Properties exposed by "oic.r.alert" are listed in Table 41.	Alerts
"/example/alertcollectionURI"	Alert Collection	"oic.r.alertcollection"	"oic.if.ll", "oic.if.b", "oic.if.baseline"	A specialisation of a Collection that contains only instances of "oic.r.alert" that may be Observed by a Client in order to consume Alerts as they are created by the Device.	Alerts

4059
 4060 Table 41 defines the details for the "oic.r.alert" Resource Type.

4061 **Table 41 – "oic.r.alert" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Category	"category"	"string"			R	Yes	Device defined category for the Alert (e.g. "System", "I/O")
Generated Time	"generatedtime"	"date-time"			R	Yes	IETF RFC 3339 formatted time at which the Alert was generated.
Originator ID	"originatorid"	"string"			R	Yes	Identity of the originator of the Alert. May be the Device ID of the Device or some other Device defined identity.
Severity	"severity"	"integer"	[0,7]		R	Yes	IETF RFC 5424 defined severity value
Subject	"subject"	"array"			R	No	Human-friendly subject of the Alert in one or more languages. This Property is an array of objects where each object has a "language" field (containing an IETF RFC 5646 language tag) and a "value" field containing the subject of the Alert name in the indicated language.
Account ID	"accounted"	"string"			R	No	Identity of the account with which the Device

							generating this Alert is associated.
--	--	--	--	--	--	--	--------------------------------------

4062

4063 The Alert Collection ("oic.r.alertcollection") Resource Type defines no Properties additional to
 4064 those defined for all instances of a Collection in Table 9. However the Alert Collection does impose
 4065 restrictions of the values that shall be populated in the "rt" and "rts" Properties. These are described
 4066 in Table 42.

4067 **Table 42 – "oic.r.alertcollection" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Links	"links"	"array"	See Table 11		R	Yes	See Table 11
Resource Type	"rt"	"array"	["oic.r.alertcollection"]		R	Yes	See Table 4.
Resource Types	"rts"	"array"	["oic.r.alert"] or ["oic.r.alert","oic.r.value.conditional"]		R	Yes	See Table 11

4068

4069 **11.9.3 Example of Use**

4070 Consider a Device that is capable of generating Alerts; it exposes an empty instance of an Alert
 4071 Collection ("oic.r.alertcollection"); that is the array of Links (the "links" Property) contains no items.

4072 As the Device under whatever conditions generates Alerts, the Device adds a Link to the Alert
 4073 Resource in the instance of an Alert Collection. A Client that has discovered the Device and is
 4074 Observing the Alert Collection using the links list OCF Interface ("oic.if.ll") will receive a notification
 4075 containing the complete Alert Collection (not just any Links that were added). It is the responsibility
 4076 of the Client to determine which Links were added (or removed if the Alert was removed); noting
 4077 that the "generatedtime" Property may be used to determine the generated order. The Client then
 4078 retrieves the Alert itself via a RETRIEVE to the "href" Link Parameter in the newly added Link to
 4079 the Collection.

4080 See D. 17 for an example of an Alert Resource and the applicable schema.

4081 **12 Messaging**

4082 **12.1 Introduction**

4083 This clause specifies the protocol messaging mapping to the CRUDN messaging operations (clause
 4084 8) for each messaging protocol specified (e.g., CoAP.). Mapping to additional protocols is expected
 4085 in later version of this document. All the Property information from the Resource model shall be
 4086 carried within the message payload. This payload shall be generated in the Resource model layer
 4087 and shall be encapsulated in the data connectivity layer. The message header shall only be used
 4088 to describe the message payload (e.g., verb, mime-type, message payload format), in addition to
 4089 the mandatory header fields defined in messaging protocol (e.g., CoAP) specification. If the
 4090 message header does not support this, then this information shall also be carried in the message
 4091 payload. Resource model information shall not be included in the message header structure unless
 4092 the message header field is mandatory in the messaging protocol specification.

4093 When a Resource is specified with a RESTful description language like OpenAPI 2.0 then the HTTP
 4094 syntax definitions are used in the description (e.g., HTTP syntax for the CRUDN operations, status
 4095 codes, etc). The HTTP syntax will be mapped to the actual used web transfer protocol (e.g., CoAP).

4096 **12.2 Mapping of CRUDN to CoAP**

4097 **12.2.1 Overview**

4098 A Device implementing CoAP shall conform to IETF RFC 7252 for the methods specified in clause
 4099 12.2.3. A Device implementing CoAP shall conform to IETF RFC 7641 to implement the CoAP
 4100 Observe option. Support for CoAP block transfer when the payload is larger than the MTU is defined
 4101 in 12.2.8.

4102 **12.2.2 URIs**

4103 An OCF: URI is mapped to a coap: URI by replacing the scheme name "ocf" with "coap" if unsecure
 4104 or "coaps" if secure before sending over the network by the requestor. Similarly on the receiver
 4105 side, the scheme name is replaced with "ocf".

4106 Any query string that is present within the URI is encoded as one or more URI-Query Options as
 4107 defined in IETF RFC 7252 clause 6.4.

4108 **12.2.3 CoAP method with request and response**

4109 **12.2.3.1 Overview**

4110 Every request has a CoAP method that realizes the request. The primary methods and their
 4111 meanings are shown in Table 43, which provides the mapping of GET/PUT/POST/DELETE
 4112 methods to CREATE, RETRIEVE, UPDATE, and DELETE operations. The associated text provides
 4113 the generic behaviours when using these methods, however Resource OCF Interfaces may modify
 4114 these generic semantics. The HTTP codes in the RESTful descriptions will be translated as
 4115 described in IETF RFC 8075 clause 7 Response Code Mapping.

4116 **Table 43 – CoAP request and response**

Method for CRUDN	(mandatory) Request data	(mandatory) Response data
GET for RETRIEVE	- Method code: GET (0.01). - Request URI: an existing URI for the Resource to be retrieved	- Response code: success (2.xx) or error (4.xx or 5.xx). - Payload: Resource representation of the target Resource (when successful).
POST for CREATE	- Method code: POST (0.02). - Request URI: an existing URI for the Resource responsible for the creation. - Payload: Resource presentation of the Resource to be created.	- Response code: success (2.xx) or error (4.xx or 5.xx). - Payload: the URI of the newly created Resource (when successful).
PUT for CREATE	- Method code: PUT (0.03). - Request URI: a new URI for the Resource to be created. - Payload: Resource presentation of the Resource to be created.	- Response code: success (2.xx) or error (4.xx or 5.xx).
POST for UPDATE	- Method code: POST (0.02). - Request URI: an existing URI for the Resource to be updated. - Payload: representation of the Resource to be updated.	- Response Code: success (2.xx) or error (4.xx or 5.xx).
DELETE for DELETE	- Method code: DELETE (0.04). - Request URI: an existing URI for the Resource to be deleted.	- Response code: success (2.xx) or error (4.xx or 5.xx).

4117

4118

4119 **12.2.3.2 CREATE with POST or PUT**

4120 **12.2.3.2.1 With POST**

4121 POST shall be used only in situations where the request URI is valid, that is it is the URI of an
4122 existing Resource on the Server that is processing the request. If no such Resource is present, the
4123 Server shall respond with an error response code of 4.xx. The use of POST for CREATE shall use
4124 an existing request URI which identifies the Resource on the Server responsible for creation. The
4125 URI of the created Resource is determined by the Server and provided to the Client in the response.

4126 A Client shall include the representation of the new Resource in the request payload. The new
4127 resource representation in the payload shall have all the necessary Properties to create a valid
4128 Resource instance, i.e. the created Resource should be able to properly respond to the valid
4129 Request with mandatory OCF Interface (e.g., "GET with ?if=oic.if.baseline").

4130 Upon receiving the POST request, the Server shall either:

- 4131 – Create the new Resource with a new URI, respond with the new URI for the newly created
4132 Resource and a success response code (2.xx); or
- 4133 – respond with an error response code (4.xx or 5.xx).

4134 POST is unsafe and is the supported method when idempotent behaviour cannot be expected or
4135 guaranteed.

4136 **12.2.3.2.2 With PUT**

4137 PUT shall be used to create a new Resource or completely replace the entire representation of an
4138 existing Resource. The Resource representation in the payload of the PUT request shall be the
4139 complete representation. PUT for CREATE shall use a new request URI identifying the new
4140 Resource to be created.

4141 The new Resource representation in the payload shall have all the necessary Properties to create
4142 a valid Resource instance, i.e. the created Resource should be able to properly respond to the
4143 valid Request with mandatory OCF Interface (e.g. "GET with ?if=oic.if.baseline").

4144 Upon receiving the PUT request, the Server shall either:

- 4145 – Create the new Resource with the request URI provided in the PUT request and send back a
4146 response with a success response code (2.xx); or
- 4147 – respond with an error response code (4.xx or 5.xx).

4148 PUT is an unsafe method but it is idempotent, thus when a PUT request is repeated the outcome
4149 is the same each time.

4150 **12.2.3.3 RETRIEVE with GET**

4151 GET shall be used for the RETRIEVE operation. The GET method retrieves the representation of
4152 the target Resource identified by the request URI.

4153 Upon receiving the GET request, the Server shall either:

- 4154 – Send back the response with the representation of the target Resource with a success response
4155 code (2.xx); or
- 4156 – respond with an error response code (4.xx or 5.xx) or ignore it (e.g. non-applicable multicast
4157 GET).

4158 GET is a safe method and is idempotent.

4159 **12.2.3.4 UPDATE with POST**

4160 POST shall be used only in situations where the request URI is valid, that is it is the URI of an
4161 existing Resource on the Server that is processing the request. If no such Resource is present, the
4162 Server shall respond with an error response code of 4.xx. A client shall use POST to UPDATE
4163 Property values of an existing Resource.

4164 Upon receiving the request, the Server shall either:

- 4165 – Apply the request to the Resource identified by the request URI in accordance with the applied
4166 OCF Interface (i.e. POST for non-existent Properties is ignored) and send back a response with
4167 a success response code (2.xx); or
- 4168 – respond with an error response code (4.xx or 5.xx). Note that if the representation in the payload
4169 is incompatible with the target Resource for POST using the applied OCF Interface (i.e. the
4170 overwrite semantic cannot be honored because of read-only Property in the payload), then the
4171 error response code 4.xx shall be returned.

4172 POST is unsafe and is the supported method when idempotent behaviour cannot be expected or
4173 guaranteed.

4174 **12.2.3.5 DELETE with DELETE**

4175 DELETE shall be used for DELETE operation. The DELETE method requests that the Resource
4176 identified by the request URI be deleted.

4177 Upon receiving the DELETE request, the Server shall either:

- 4178 – Delete the target Resource and send back a response with a success response code (2.xx); or
- 4179 – respond with an error response code (4.xx or 5.xx).

4180 DELETE is unsafe but idempotent (unless URIs are recycled for new instances).

4181 **12.2.4 Content-Format negotiation**

4182 The Framework mandates support of CBOR, however it allows for negotiation of the payload body
4183 if more than one Content-Format (e.g. CBOR and JSON) is supported by an implementation. In this
4184 case the Accept Option defined in clause 5.10.4 of IETF RFC 7252 shall be used to indicate which
4185 Content-Format (e.g. JSON) is requested by the Client.

4186 The Content-Formats supported are shown in Table 44.

4187 **Table 44 – OCF Content-Formats**

Media Type	ID
"application/cbor"	60
"application/vnd.ocf+cbor"	10000

4188
4189 Clients shall include a Content-Format Option in every message that contains a payload. Servers
4190 shall include a Content-Format Option for all success (2.xx) responses with a payload body. Per
4191 IETF RFC 7252 clause 5.5.1, Servers shall include a Content-Format Option for all error (4.xx or
4192 5.xx) responses with a payload body unless they include a Diagnostic Payload; error responses
4193 with a Diagnostic Payload do not include a Content-Format Option. The Content-Format Option
4194 shall use the ID column numeric value from Table 44. An OCF vertical may mandate a specific
4195 Content-Format Option.

4196 Clients shall also include an Accept Option in every request message. The Accept Option shall
4197 indicate the required Content-Format as defined in Table 44 for response messages. The Server

4198 shall return the required Content-Format if available. If the required Content-Format cannot be
 4199 returned, then the Server shall respond with an appropriate error message.

4200 **12.2.5 OCF-Content-Format-Version information**

4201 Servers and Clients shall include the OCF-Content-Format-Version Option in both request and
 4202 response messages with a payload. Clients shall include the OCF-Accept-Content-Format-Version
 4203 Option in request messages. The OCF-Content-Format-Version Option and OCF-Accept-Content-
 4204 Format-Version Option are specified as Option Numbers in the CoAP header as shown in Table 45.

4205 **Table 45 – OCF-Content-Format-Version and OCF-Accept-Content-Format-Version Option**
 4206 **Numbers**

CoAP Option Number	Name	Format	Length (bytes)
2049	OCF-Accept-Content-Format-Version	uint	2
2053	OCF-Content-Format-Version	uint	2

4207
 4208 The value of both the OCF-Accept-Content-Format-Version Option and the OCF-Content-Format-
 4209 Version Option is a two-byte unsigned integer that is used to define the major, minor and sub
 4210 versions. The major and minor versions are represented by 5 bits and the sub version is
 4211 represented by 6 bits as shown in Table 46.

4212 **Table 46 – OCF-Accept-Content-Format-Version and OCF-Content-Format-Version**
 4213 **Representation**

	Major Version					Minor Version					Sub Version					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

4214
 4215 Table 47 illustrates several examples:

4216 **Table 47 – Examples of OCF-Content-Format-Version and OCF-Accept-Content-Format-**
 4217 **Version Representation**

OCF version	Binary representation	Integer value
"1.0.0"	"0000 1000 0000 0000"	2048
"1.1.0"	"0000 1000 0100 0000"	2112

4218
 4219 The OCF-Accept-Content-Format-Version Option and OCF-Content-Format-Version Option for this
 4220 version of the document shall be "1.0.0" (i.e. "0b0000 1000 0000 0000").

4221 **12.2.6 Content-Format policy**

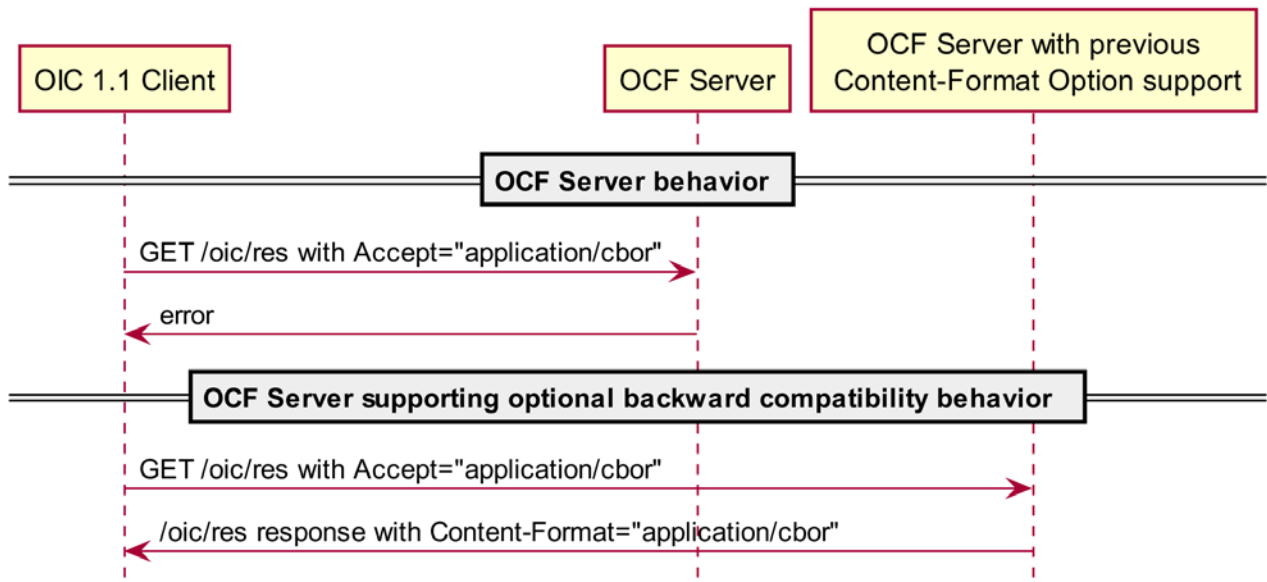
4222 All Devices shall support the current Content-Format Option, "application/vnd.ocf+cbor", and OCF-
 4223 Content-Format-Version "1.0.0".

4224 For backward compatibility with previous OCF-Content-Format-Version Options:

- 4225 – All Client Devices shall support OCF-Content-Format-Version Option set to "1.0.0" and higher.

- 4226 – All Client Devices shall support OCF-Accept-Content-Format-Version Option set to "1.0.0" and
4227 higher.
- 4228 – A Client shall send a discovery request message with its Accept Option set to
4229 "application/vnd.ocf+cbor", and its OCF-Accept-Content-Format-Version Option matching its
4230 highest supported version.
- 4231 – A Server shall respond to a Client's discovery request that is higher than its OCF-Content-
4232 Format-Version by responding with its Content-Format Option set to "application/vnd.ocf+cbor",
4233 and OCF-Content-Format-Version matching its highest supported version. The response
4234 representation shall be encoded with the OCF-Content-Format-Version matching the Server's
4235 highest supported version.
- 4236 – A Server may support previous Content-Formats and OCF-Content-Format-Versions to support
4237 backward compatibility with previous versions.
- 4238 – For a Server that supports multiple OCF-Content-Format-Version Options, the Server should
4239 attempt to respond with an OCF-Content-Format-Version that matches the OCF-Accept-
4240 Content-Format-Version of the request.
- 4241 For optional backward compatibility with OIC 1.1:
- 4242 – All Devices that claim backward compatibility to the OIC 1.1 specification shall support the
4243 "application/cbor" media type.
- 4244 – For a Client supporting backward compatibility with OIC 1.1, the Client shall send a discovery
4245 request with its Accept Option set to "application/cbor" in response to an error from an OIC 1.1
4246 Server.
- 4247 – A Server supporting OIC 1.1 compatibility shall handle a Client request containing the Accept
4248 Option = "application/cbor" by responding with its Content-Format Option set to
4249 "application/cbor" and no OCF-Content-Format-Version Option.
- 4250 – For more OIC 1.1 information see Annex E.
- 4251 To maintain compatibility between Devices implemented to different versions of this document,
4252 Devices should follow the policy as described in Figure 29, Figure 30 and Figure 31.
- 4253 The OIC 1.1 Clients and Servers represented in Figure 29 and Figure 30 support sending Content-
4254 Format Option set to "application/cbor" and Accept Option set to "application/cbor". The OIC 1.1
4255 Clients and Servers do not support OCF-Content-Format-Version Option, nor the OCF-Accept-
4256 Content-Format-Version Option. The OCF Clients in Figure 30 and Figure 31 support sending
4257 Content-Format Option set to "application/vnd.ocf+cbor", Accept Option set to
4258 "application/vnd.ocf+cbor", OCF-Content-Format-Version Option set to "1.0.0", and OCF-Accept-
4259 Content-Format-Version Option set to "1.0.0" (representing OCF 1.0 and later Clients). The OCF
4260 Servers in Figure 29 and Figure 31 support sending Content-Format Option set to
4261 "application/vnd.ocf+cbor" and OCF-Content-Format-Version Option set to "1.0.0" (representing
4262 OCF 1.0 and later Servers).

4263

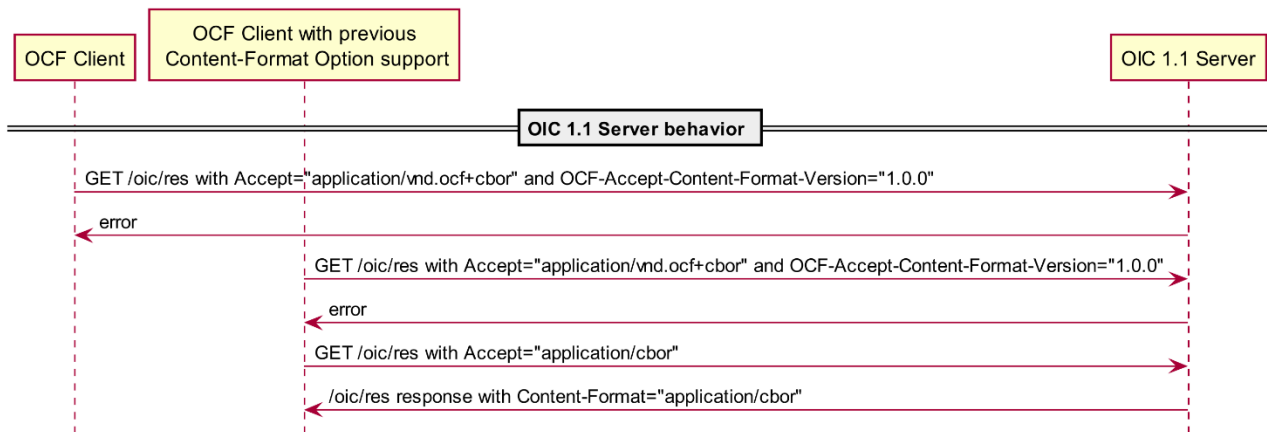


4264
4265

Figure 29 – Content-Format Policy for OCF Servers supporting error responses and backward compatibility responses

4268

4269

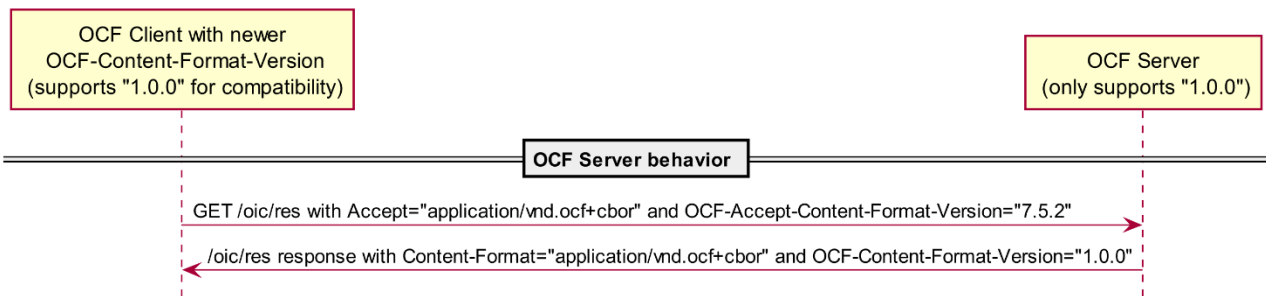


4270
4271

Figure 30 – Content-Format Policy for OCF Clients supporting error responses and backward compatibility responses

4273

4274



4275
4276 **Figure 31 – Content-Format Policy for backward compatible OCF Clients negotiating lower**
4277 **OCF Content-Format-Version**

4278 **12.2.7 CRUDN to CoAP response codes**

4279 The mapping of CRUDN operations response codes to CoAP response codes are identical to the
4280 response codes defined in IETF RFC 7252.

4281 **12.2.8 CoAP block transfer**

4282 Basic CoAP messages work well for the small payloads typical of light-weight, constrained IoT
4283 devices. However scenarios can be envisioned in which an application needs to transfer larger
4284 payloads.

4285 CoAP block-wise transfer as defined in IETF RFC 7959 shall be used by all Servers which generate
4286 a content payload that would exceed the size of a CoAP datagram as the result of handling any
4287 defined CRUDN operation.

4288 Similarly, CoAP block-wise transfer as defined in IETF RFC 7959 shall be supported by all Clients.
4289 The use of block-wise transfer is applied to both the reception of payloads as well as transmission
4290 of payloads that would exceed the size of a CoAP datagram.

4291 All blocks that are sent using this mechanism for a single instance of a transfer shall all have the
4292 same reliability setting (i.e. all confirmable or all non-confirmable).

4293 A Client may support both the block1 (as descriptive) and block2 (as control) options as described
4294 by IETF RFC 7959. A Server may support both the block1 (as control) and block2 (as descriptive)
4295 options as described by IETF RFC 7959.

4296 **12.3 Mapping of CRUDN to CoAP serialization over TCP**

4297 **12.3.1 Overview**

4298 In environments where TCP is already available, CoAP can take advantage of it to provide reliability.
4299 Also in some environments UDP traffic is blocked, so deployments may use TCP. For example,
4300 consider a cloud application acting as a Client and the Server is located at the user's home. A
4301 Server which already support CoAP as a messaging protocol could easily support CoAP
4302 serialization over TCP rather than utilizing another messaging protocol. A Device implementing
4303 CoAP Serialization over TCP shall conform to IETF RFC 8323.

4304 **12.3.2 URIs**

4305 When UDP is blocked, Clients are dependent on pre-configured details of the Device to determine
4306 if the Device supports CoAP serialization over TCP. When UDP is not-blocked, a Device which
4307 supports CoAP serialization over TCP shall populate the "eps" Parameter in the "/oic/res" response,
4308 as defined in 10.2, with the URI scheme(s) as defined in clause 8.1 or 8.2 of IETF RFC 8323. For
4309 the "coaps+tcp" URI scheme, as defined in clause 8.2 of IETF RFC 8323, IETF RFC 7301 shall be
4310 used. In addition, the URIs used for CoAP serialization over TCP shall conform to 12.2.2 by

4311 substituting the scheme names with the scheme names defined in clauses 8.1 and 8.2 of
4312 IETF RFC 8323 respectively.

4313 **12.3.3 CoAP method with request and response**

4314 The CoAP methods used for CoAP serialization over TCP shall conform to 12.2.3.

4315 **12.3.4 Content-Format negotiation**

4316 The Content Format negotiation used for CoAP serialization over TCP shall conform to 12.2.4.

4317 **12.3.5 OCF-Content-Format-Version information**

4318 The OCF Content Format Version information used for CoAP serialization over TCP shall conform
4319 to 12.2.5.

4320 **12.3.6 Content-Format policy**

4321 The Content Format policy used for CoAP serialization over TCP shall conform to 12.2.6.

4322 **12.3.7 CRUDN to CoAP response codes**

4323 The CRUDN to CoAP response codes for CoAP serialization over TCP shall conform to 12.2.7.

4324 **12.3.8 CoAP block transfer**

4325 The CoAP block transfer for CoAP serialization over TCP shall conform to clause 6 of
4326 IETF RFC 8323.

4327 **12.3.9 Keep alive (connection health)**

4328 The Device that initiated the CoAP over TCP connection shall send a Ping message as described
4329 in clause 5.4 in IETF RFC 8323. The Device to which the connection was made may send a Ping
4330 message. The recipient of any Ping message shall send a Pong message as described in clause
4331 5.4 in IETF RFC 8323.

4332 Both sides of an established CoAP over TCP connection may send subsequent Ping (and
4333 corresponding Pong) messages.

4334 **12.4 Payload Encoding in CBOR**

4335 OCF implementations shall perform the conversion to CBOR from JSON defined schemas and to
4336 JSON from CBOR in accordance with IETF RFC 7049 clause 4 unless otherwise specified in this
4337 clause.

4338 Properties defined as a JSON integer shall be encoded in CBOR as an integer (CBOR major types
4339 0 and 1). Properties defined as a JSON number shall be encoded as an integer, single- or double-
4340 precision floating point (CBOR major type 7, sub-types 26 and 27); the choice is implementation
4341 dependent. Half-precision floating point (CBOR major 7, sub-type 25) shall not be used. Integer
4342 numbers shall be within the closed interval $[-2^{53}, 2^{53}]$. Properties defined as a JSON number
4343 should be encoded as integers whenever possible; if this is not possible Properties defined as a
4344 JSON number should use single-precision if the loss of precision does not affect the quality of
4345 service, otherwise the Property shall use double-precision.

4346 On receipt of a CBOR payload, an implementation shall be able to interpret CBOR integer values
4347 in any position. If a Property defined as a JSON integer is received encoded other than as an
4348 integer, the implementation may reject this encoding using a final response as appropriate for the
4349 underlying transport (e.g. 4.00 for CoAP) and thus optimise for the integer case. If a Property is
4350 defined as a JSON number an implementation shall accept integers, single- and double-precision
4351 floating point.

4352 **13 Security**

4353 The details for handling security and privacy are specified in ISO/IEC 30118-2:2018.

Annex A
(informative)

Operation Examples

A.1 Introduction

This clause describes some example scenarios using sequence of operations between the entities involved. In all the examples illustrated in Figure A.1 *Light* is a Server and *Smartphone* is a Client. In one of the scenario *Garage* additionally acts as a Server. All the examples are based on the following example Resource definitions:

"rt=oc.example.light" with Resource Type definition as illustration in Figure A.1.

Table A-1 – "oc.example.light" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Name	"n"	"string"	N/A	N/A	R, W	No	N/A
on-off	"of"	"boolean"	N/A	N/A	R, W	Yes	On/Off Control: "0" = Off "1" = On
dim	"dm"	"integer"	0-255	N/A	R, W	Yes	Resource which can take a range of values minimum being "0" and maximum being "255"

rt=oc.example.garagedoor with Resource Type definition as illustration in Table A-2.

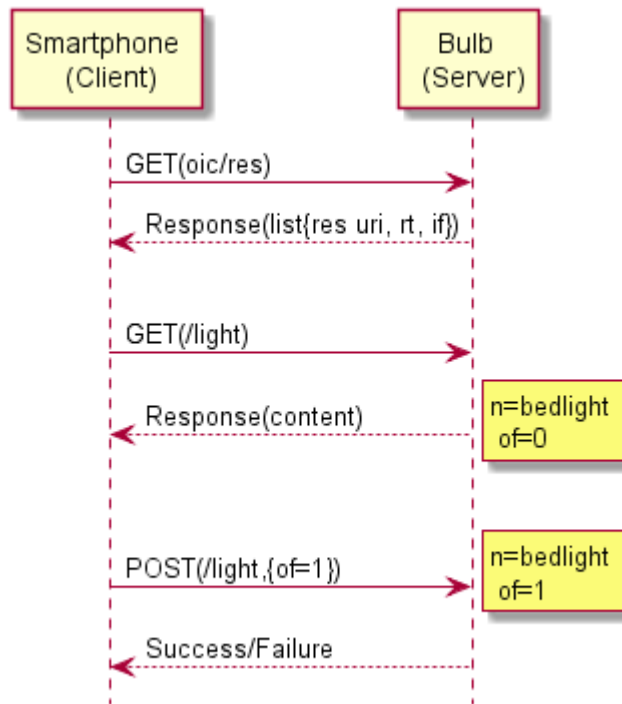
Table A-2 – oc.example.garagedoor Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Name	"n"	"string"	N/A	N/A	R, W	No	N/A
open-close	"oc"	"boolean"	N/A	N/A	R, W	Yes	Open/Close Control: 0 = Open 1 = Close

"/oc/mnt" ("rt=oc.wk.mnt") used in the examples in Figure A.2 is defined in 11.5.2.

A.2 When at home: From smartphone turn on a single light

This sequence highlights (Figure A.1) the discovery and control of an OCF light Resource from an OCF smartphone.



4374

4375 **Figure A.1 – When at home: from smartphone turn on a single light**

4376 Discovery request can be sent to *All OCF Nodes* Multicast address FF0X::158 or can be sent
 4377 directly to the IP address of Device hosting the light Resource.

- 4378 – Smartphone sends a GET request to "/oic/res" Resource to discover all Resources hosted on
 4379 targeted end point.
- 4380 – The end point (bulb) responds with the list of Resource URI, Resource Type and OCF Interfaces
 4381 supported on the end point (one of the Resource is "/light" whose "rt=oic.example.light").
- 4382 – Smartphone sends a GET request to "/light" Resource to know its current state.
- 4383 – The end point responds with representation of light Resource ({n=bedlight;of=0}).
- 4384 – Smartphone changes the "of" Property of the light Resource by sending a POST request to
 4385 "/light" Resource ("of=1").
- 4386 – On Successful execution of the request, the end point responds with the changed Resource
 4387 representation. Else, error code is returned. Details of the error codes are defined in 12.2.7.

4388 **A.3 GroupAction execution**

4389 This example will be added when groups feature is added in later version of the document.

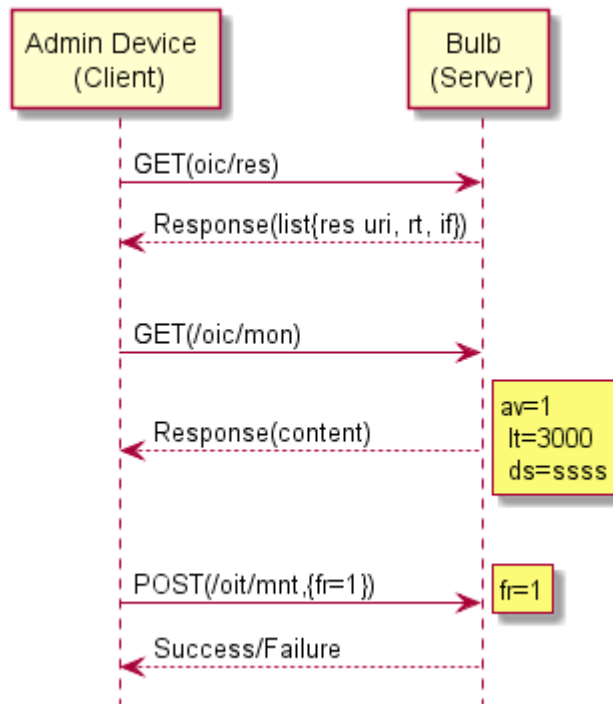
4390 **A.4 When garage door opens, turn on lights in hall; also notify smartphone**

4391 This example will be added when scripts feature is added in later version of the document.

4392 **A.5 Device management**

4393 This sequence highlights (Figure A.2) the Device management function of maintenance.

4394



4395

4396

Figure A.2 – Device management (maintenance)

4397 **Pre-Condition:** Admin Device has different security permissions and hence can perform Device
 4398 management operations on the Device.

- 4399 – Admin Device sends a GET request to "/oic/res" Resource to discover all Resources hosted on
 4400 a targeted end point (in this case Bulb).
- 4401 – The end point (bulb) responds with the list of Resource URI, Resource Type and OCF Interfaces
 4402 supported on the end point (one of the Resources is "/oic/mnt" whose "rt=oic.wk.mnt").
- 4403 – Admin Device changes the "fr" Property of the maintenance Resource by sending a POST
 4404 request to "/oic/mnt" Resource ("{fr=1}"). This triggers a factory reset of the end point (bulb).
- 4405 – On successful execution of the request, the end point responds with the changed Resource
 4406 representation. Else, error code is returned. Details of the error codes are defined in 12.2.7.

Annex B
(informative)

OCF interaction scenarios and deployment models

B.1 OCF interaction scenarios

A Client connects to one or multiple Servers in order to access the Resources provided by those Servers. The following are scenarios representing possible interactions among roles:

- Direct interaction between Client and Server (Figure B.1). In this scenario the Client and the Server directly communicate without involvement of any other Device. A smartphone which controls an actuator directly uses this scenario.

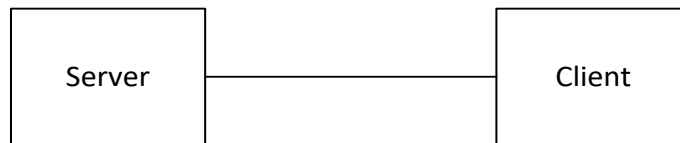


Figure B.1 – Direct interaction between Server and Client

- Interaction between Client and Server using another server (Figure B.). In this scenario, another Server provides the support needed for the Client to directly access the desired Resource on a specific Server. This scenario is used for example, when a smartphone first accesses a discovery server to find the addressing information of a specific appliance, and then directly accesses the appliance to control it.

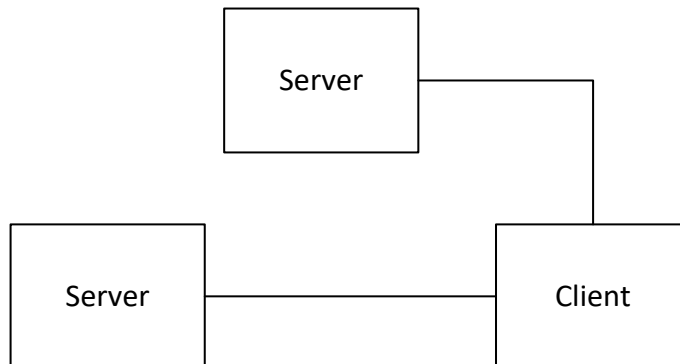


Figure B.2– Interaction between Client and Server using another Server

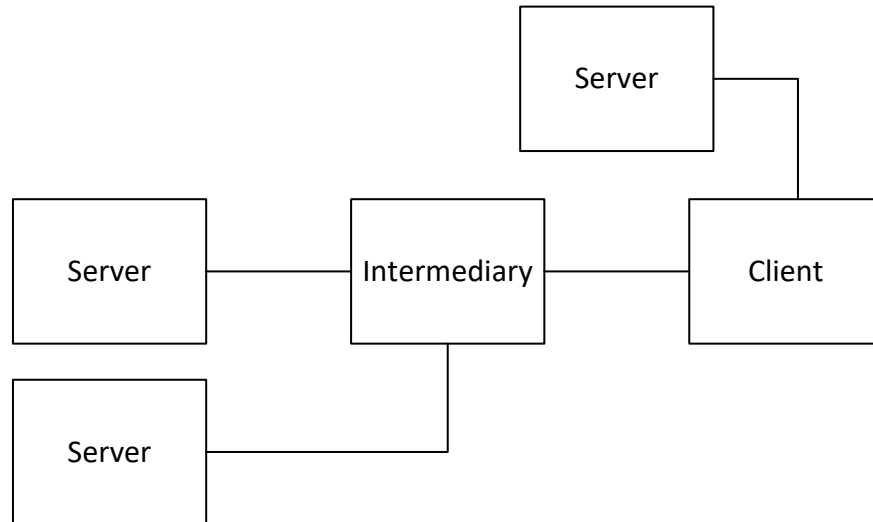
- Interaction between Client and Server using Intermediary (Figure B.3). In this scenario an Intermediary facilitates the interaction between the Client and the Server. A smartphone which controls appliances in a smart home via MQTT broker uses this scenario.



Figure B.3 – Interaction between Client and Server using Intermediary

- Interaction between Client and Server using support from multiple Servers and intermediary (Figure B.4). In this scenario, both Server and Intermediary roles are present to facilitate the transaction between the Client and a specific Server. An example scenario is when a

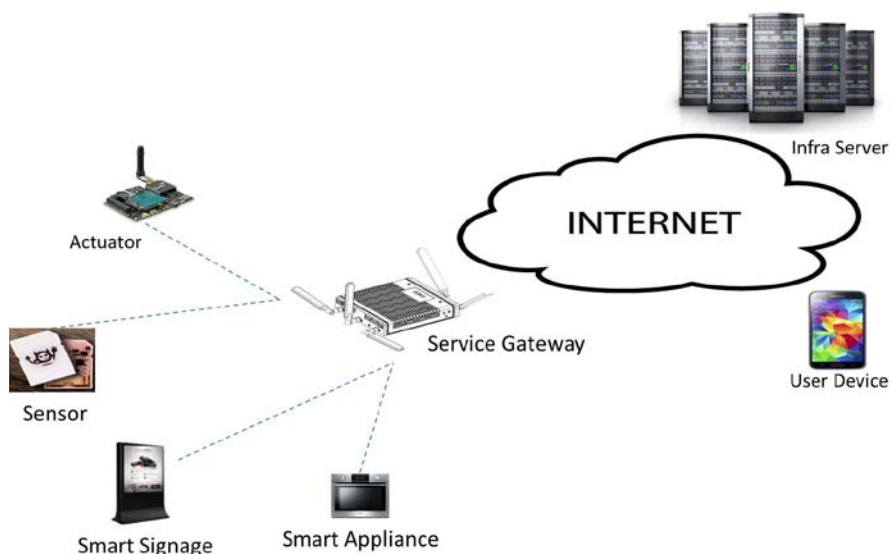
4434 smartphone first accesses a Resource Directory (RD) server to find the address to a specific
 4435 appliance, then utilizes MQTT broker to deliver a command message to the appliance. The
 4436 smartphone can utilize the mechanisms defined in CoRE Resource Directory such as default
 4437 location, anycast address or DHCP to discover the Resource Directory information.



4438
 4439 **Figure B.4 – Interaction between Client and Server using support from multiple Servers**
 4440 **and Intermediary**

4441 **B.2 Deployment model**

4442 In deployment, Devices are deployed and interact via either wired or wireless connections. Devices
 4443 are the physical entities that may host Resources and play one or more roles. There is no constraint
 4444 on the structure of a deployment or number of Devices in it. Architecture is flexible and scalable
 4445 and capable of addressing large number of Devices with different Device capabilities, including
 4446 constrained Devices which have limited memory and capabilities. Constrained Devices are defined
 4447 and categorized in [TCNN].



4448
 4449 **Figure B.5 – Example of Devices**

4450 Figure B.5 depicts a typical deployment and set of Devices, which may be divided in the following
4451 categories:

- 4452 – *Things*: Networked Devices which are able to interface with physical environments. Things are
4453 the Devices which are primarily controlled and monitored. Examples include smart appliances,
4454 sensors, and actuators. Things mostly take the role of Server but they may also take the role of
4455 Client, for example in machine-to-machine communications.
- 4456 – *User Devices*: Devices employed by the users enabling the users to access Resources and
4457 services. Examples include smart phones, tablets, and wearable devices. User Devices mainly
4458 take the role of Client, but may also take the role of Server or Intermediary.
- 4459 – *Service Gateways*: Network equipment which take the role of Intermediary. Examples are home
4460 gateways.
- 4461 – *Infra Servers*: Data centers residing in cloud infrastructure, which facilitate the interaction
4462 among Devices by providing network services such as AAA, NAT traversal or discovery. It can
4463 also play the role of Client or Intermediary.

4464 Annex C
4465 (informative)

4466
4467 **Other Resource models and OCF mapping**

4468 **C.1 Multiple Resource models**

4469 RESTful interactions are defined dependent on the Resource model; hence, Devices require a
4470 common understanding of the Resource model for interoperability.

4471 There are multiple Resource models defined by different organizations including OCF, OMA
4472 SpecWorks and oneM2M used in the industry, which may restrict interoperability among respective
4473 ecosystems. The main differences from Resource model are as follows:

- 4474 – *Resource structure*: Resources may be defined to have Properties (e.g., oneM2M defined
4475 Resources), or may be defined as an atomic entity and not be decomposable into Properties
4476 (e.g., OMA SpecWorks defined Resources). For example, a smart light may be represented as
4477 a Resource with an on-off Property or a Resource Collection containing an on-off Resource. In
4478 the former, on-off Property doesn't have a URI of its own and can only be accessed indirectly
4479 via the Resource. In the latter, being a Resource itself, on-off Resource is assigned its own
4480 URI and can be directly manipulated.
- 4481 – *Resource name & type*: Resources may be allowed to be named freely and have their
4482 characteristics indicated using a Resource Type Property (e.g., as defined in oneM2M).
4483 Alternatively, the name of Resources may be defined a priori in a way that the name by itself is
4484 indicative of its characteristic (e.g., as defined by OMA SpecWorks). For example, in oneM2M
4485 Resource model, a smart light can be named with no restrictions, such as "LivingRoomLight_1"
4486 but in OMA SpecWorks Resource model it is required to have the fixed Object name with
4487 numerical Object ID of OMA SpecWorks Light Control ("3311"). Consequently, it's likely that in
4488 the former case the data path in URI is freely defined and in the latter case it is predetermined.
- 4489 – *Resource hierarchy*: Resources may be allowed to be organized in hierarchy where a Resource
4490 contains another Resource with a parent-child relationship (e.g., in oneM2M definition of
4491 Resource model). Resources may also be required to have a flat structure and associate with
4492 other Resources only by referencing their links.

4493 In addition, different organizations use different syntax and define different features (e.g., Resource
4494 OCF Interface), which preclude interoperability.

4495 **C.2 OCF approach for support of multiple Resource models**

4496 In order to expand the IoT ecosystem the Framework takes an inclusive approach for interworking
4497 with existing Resource models. Specifically, the Framework defines a Resource model while
4498 providing a mechanism to easily map to other models. By embracing existing Resource models
4499 OCF is inclusive of existing ecosystems while allowing for the transition toward definition of a
4500 comprehensive Resource model integrating all ecosystems.

4501 The following OCF characteristics enable support of other Resource models:

- 4502 – *Resource model is the superset of multiple models*: the Resource model is defined as the
4503 superset of existing Resource models. In other words, any existing Resource model can be
4504 mapped to a subset of Resource model concepts.
- 4505 – *Framework may allow for Resource model negotiation*: the Client and Server exchange the
4506 information about what Resource model(s) each supports. Based on the exchanged information,
4507 the Client and Server choose a Resource model to perform RESTful interactions or to perform
4508 translation. This feature is out of scope of the current version of this document, however, the
4509 following is a high level description for Resource model negotiation.

4510 **C.3 Resource model indication**

4511 The Client and server exchange the information about what Resource model(s) each supports.
4512 Based on the exchanged information, the Client and Server choose a Resource model to perform
4513 RESTful interactions or to perform translation. The exchange could be part of discovery and
4514 negotiation. Based on the exchange, the Client and Server follow a procedure to ensure
4515 interoperability among them. They may choose a common Resource model or execute translation
4516 between Resource models.

4517 – *Resource model schema exchange*: The Client and Server may share the Resource model
4518 information when they initiate a RESTful interaction. They may exchange the information about
4519 which Resource model they support as part of session establishment procedures. Alternatively,
4520 each request or response message may carry the indication of which Resource model it is using.
4521 For example, [COAP] defines Content-Format option to indicate the representation format such
4522 as "application/json". It's possible to extend the Content-Format Option to indicate the Resource
4523 model used with the representation format such as "application/ips0-json".

4524 – *Ensuing procedures*: After the Client and Server exchange the Resource model information,
4525 they perform a suitable procedure to ensure interoperability among them. The simplest way is
4526 to choose a Resource model supported by both the Client and Server. In case there is no
4527 common Resource model, the Client and Server may interact through a 3rd party.

4528 In addition to translation which can be resource intensive, a method based on profiles can be used
4529 in which an OCF implementation can accommodate multiple profiles and hence multiple
4530 ecosystems.

4531 – *Resource Model Profile*: the Framework defines Resource model profiles and implementers or
4532 users choose the active profile. The chosen profile constraints the Device to strict rules in how
4533 Resources are defined, instantiated and interacted with. This would allow for interoperation with
4534 devices from the ecosystem identified by the profile (e.g., OMA SpecWorks, OneM2M etc.).
4535 Although this enables a Device to participate in and be part of any given ecosystem, this scheme
4536 does not allow for generic interoperability at runtime. While this approach may be suitable for
4537 resource constrained devices, more resource capable devices are expected to support more
4538 than one profile.

4539 **C.4 An Example Profile (OMA SpecWorks profile)**

4540 **C.4.1 Overview**

4541 OMA SpecWorks defines smart objects that have specific Resources and they take values
4542 determined by the data type of that Resource. The smart object specification defines a category of
4543 such objects. Each Resource represents a characteristic of the smart object being modelled.

4544 While the terms may be different, there are equivalent concepts in OCF to represent these terms.
4545 This clause provides the equivalent OCF terms and then frames the OMA SpecWorks smart object
4546 in OCF terms.

4547 The OMA SpecWorks object Light Control defined in clause 16 of the OMA SpecWorks Smart
4548 Objects 1.0 is used as the reference example.

4549 **C.5 Conceptual equivalence**

4550 The OMA SpecWorks smart object definition is equivalent to a Resource Type definition which
4551 defines the relevant characteristics of an entity being modelled. The specific OMA SpecWorks
4552 Resource is equivalent to a Property that like an OMA SpecWorks Resource has a defined data
4553 type, enumeration of acceptable values, units, a general description and access modes (based on
4554 the OCF Interface).

4555 The general method for developing the equivalent Resource Type from an OMA SpecWorks Smart
 4556 Object definition is to ignore the Object ID and replace the Object URN with and OCF "." (dot)
 4557 separated name that incorporates the OMA SpecWorks object. Alternatively the Object URN can
 4558 be used as the Resource Type ID as is (as long as the URN does not contain any "." (dots)) – using
 4559 the same Object URN as the Resource Type ID allows for compatibility when interacting with an
 4560 OMA SpecWorks compliant device. The object URN based naming does not have any bearing for
 4561 OCF to OCF interoperability and so the OCF format is preferred – for OCF to OCF interoperability
 4562 only the data model consistency is required.

4563 Two models are available to render OMA SpecWorks objects into OCF.

- 4564 – One is where the OMA SpecWorks Smart Object represents a Resource. In this case, the IP
 4565 Smart Object is regarded as a Resource with the Resource Type matching the description of
 4566 the Smart Object. Furthermore, each Resource in the OMA SpecWorks definition is represented
 4567 as a Property in the Resource Type (the OMA SpecWorks Resource ID is replaced with a string
 4568 representing the Property). This is the preferred approach when the OMA SpecWorks Data
 4569 Model is expressed in the Resource Model.
- 4570 – The other approach is to model an OMA SpecWorks Smart Object as a Collection. Each OMA
 4571 SpecWorks Resource is then modelled as a Resource with a Resource Type that matches the
 4572 definition of the OMA SpecWorks Resource. Each of these Resource instances are then bound
 4573 to the Collection that represents this OMA SpecWorks Smart Object.

4574 C.5.1 is an example showing how an OMA SpecWorks LightControl Object is modelled as a
 4575 Resource.

4576 **C.5.1 Resource Type: Light Control**

4577 Description: This Object is used to control a light source, such as a LED or other light. It allows a
 4578 light to be turned on or off and its dimmer setting to be controlled as a percentage value between
 4579 0 and 100. An optional colour setting enables a string to be used to indicate the desired colour.
 4580 Table C-1 and Table C-2 define the Resource Type and its Properties, respectively.

4581 **Table C-1 – Light control Resource Type definition**

Resource Type	Resource Type ID	Multiple Instances	Description
Light Control	"oic.light.control" or "urn:oma:lwm2m:ext:3311"	Yes	Light control object with on/off and optional dimming and energy monitor

4582

4583 **Table C-2 – Light control Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
On/Off	"on-off"	"boolean"	N/A	N/A	R, W	Yes	On/Of Control: 0 = Off 1 = On
Dimmer	"dim"	"integer"	N/A	%	R, W	No	Proportional Control, integer value between 0 and 100 as percentage
Color	"color"	"string"	0 – 100	Defined by "units" Property	R, W	No	String representing some value in color space
Units	"units"	"string"	N/A	N/A	R	No	Measurement Units Definition e.g., "Cel"

							for Temperature in Celsius.
On Time	"ontime"	"integer"	N/A	s	R, W	No	The time in seconds that the light has been on. Writing a value of "0" resets the counter
Cumulative active power	"cumap"	"float"	N/A	Wh	R	No	The cumulative active power since the last cumulative energy reset or device start
Power Factor	"powfact"	"float"	N/A	N/A	R	No	The power factor of the load

4584

4585

4586 Annex D
4587 (normative)

4588 **Resource Type definitions**
4589

4590 **D.1 List of Resource Type definitions**

4591 All the clauses in Annex D and Annex E describe the Resource Types with a RESTful API definition
4592 language. The Resource Type definitions presented in Annex D and Annex E are formatted for
4593 readability, and so may appear to have extra line breaks. Table D-1 contains the list of defined
4594 Core Common Resources in this document.

4595 **Table D-1 – Alphabetized list of Core Resources**

Friendly Name (informative)	Resource Type (rt)	Clause
Alerts	"oic.r.alert"	D.17
Alerts Collection	"oic.r.alertcollection"	D.18
Atomic Measurement	"oic.wk.atomicmeasurement"	D.2
Collections	"oic.wk.col"	D.3
Device Configuration	"oic.wk.con"	D.4
Platform Configuration	"oic.wk.con.p"	D.5
Device	"oic.wk.d"	D.6
Discoverable Resource	"oic.wk.res"	D.13
Icon	"oic.r.icon"	D.7
Introspection	"oic.wk.introspection"	D.8
Maintenance	"oic.wk.mnt"	D.9
Network Monitoring	"oic.wk.nmon"	D.10
Platform	"oic.wk.p"	D.11
Resource Directory	"oic.wk.rd"	D.12
Scenes (Top Level)	"oic.wk.scenelist"	D.14
Scenes Collections	"oic.wk.scenecollection"	D.15
Scene Member	"oic.wk.scenemember"	D.16

4596
4597
4598 **D.2 Atomic Measurement links list representation**

4599 **D.2.1 Introduction**

4600 The oic.if.baseline OCF Interface exposes a representation of the links and
4601 the Common Properties of the Atomic Measurement Resource.
4602

4603 **D.2.2 Example URI**

4604 /AtomicMeasurementResURI

4605 **D.2.3 Resource type**

4606 The Resource Type is defined as: "oic.wk.atomicmeasurement".

```

4607 D.2.4 OpenAPI 2.0 definition
4608 {
4609     "swagger": "2.0",
4610     "info": {
4611         "title": "Atomic Measurement links list representation",
4612         "version": "2019-03-04",
4613         "license": {
4614             "name": "OCF Data Model License",
4615             "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
4616             "x-copyright": "Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights reserved."
4617         },
4618         "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
4619     },
4620     "schemes": ["http"],
4621     "consumes": ["application/json"],
4622     "produces": ["application/json"],
4623     "paths": {
4624         "/AtomicMeasurementResURI?if=oic.if.ll": {
4625             "get": {
4626                 "description": "The oic.if.ll OCF Interface exposes a representation
of the Links",
4627                 "parameters": [
4628                     {
4629                         "$ref": "#/parameters/interface-all"
4630                     }
4631                 ],
4632                 "responses": {
4633                     "200": {
4634                         "description": "",
4635                         "x-example": [{
4636                             "href": "/temperature",
4637                             "rt": ["oic.r.temperature"],
4638                             "if": ["oic.if.s", "oic.if.baseline"]
4639                         }],
4640                         {
4641                             "href": "/bodylocation",
4642                             "rt": ["oic.r.body.location.temperature"],
4643                             "if": ["oic.if.s", "oic.if.baseline"]
4644                         },
4645                         {
4646                             "href": "/timestamp",
4647                             "rt": ["oic.r.time.stamp"],
4648                             "if": ["oic.if.s", "oic.if.baseline"]
4649                         }
4650                     ]],
4651                     "schema": {
4652                         "$ref": "#/definitions/links"
4653                     }
4654                 }
4655             }
4656         },
4657         "/AtomicMeasurementResURI?if=oic.if.b": {
4658             "get": {
4659                 "description": "The oic.if.b OCF Interface returns data items
retrieved from Resources pointed to by the Links.\n",
4660                 "parameters": [
4661                     {
4662                         "$ref": "#/parameters/interface-all"
4663                     }
4664                 ],
4665                 "responses": {
4666                     "200": {
4667                         "description": "Normal response, no errors, all
Properties are returned correctly\n",
4668                         "x-example": [{
4669                             "href": "/temperature",
4670                             "rep": {
4671                                 "temperature": 38,
4672                                 "units": "C",
4673                                 "range": [25, 45]

```

```

4677     }
4678     },
4679     {
4680         "href": "/bodylocation",
4681         "rep": {
4682             "bloc": "ear"
4683         }
4684     },
4685     {
4686         "href": "/timestamp",
4687         "rep": {
4688             "timestamp": "2007-04-05T14:30+09:00"
4689         }
4690     }
4691     ]],
4692     "schema": {
4693         "$ref": "#/definitions/batch-retrieve"
4694     }
4695     }
4696     },
4697     },
4698     "/AtomicMeasurementResURI?if=oic.if.baseline": {
4699         "get": {
4700             "description": "The oic.if.baseline OCF Interface exposes a
4701 representation of the links and\nthe Common Properties of the Atomic Measurement Resource.\n",
4702             "parameters": [
4703                 {
4704                     "$ref": "#/parameters/interface-all"
4705                 }
4706             ],
4707             "responses": {
4708                 "200": {
4709                     "description": "",
4710                     "x-example": {
4711                         "rt": ["oic.wk.atomicmeasurement"],
4712                         "if": ["oic.if.b", "oic.if.ll",
4713                             "oic.if.baseline"],
4714                         "rts": ["oic.r.temperature",
4715                             "oic.r.body.location.temperature", "oic.r.time.stamp"],
4716                         "rts-m": ["oic.r.temperature",
4717                             "oic.r.body.location.temperature", "oic.r.time.stamp"],
4718                         "links": [{
4719                             "href": "/temperature",
4720                             "rt": ["oic.r.temperature"],
4721                             "if": ["oic.if.s", "oic.if.baseline"]
4722                         },
4723                         {
4724                             "href": "/bodylocation",
4725                             "rt":
4726 ["oic.r.body.location.temperature"],
4727                             "if": ["oic.if.s", "oic.if.baseline"]
4728                         },
4729                         {
4730                             "href": "/timestamp",
4731                             "rt": ["oic.r.time.stamp"],
4732                             "if": ["oic.if.s", "oic.if.baseline"]
4733                         }
4734                     ]
4735                 },
4736                 "schema": {
4737                     "$ref": "#/definitions/baseline"
4738                 }
4739             }
4740         }
4741     }
4742     },
4743     "parameters": {
4744         "interface-all": {
4745             "in": "query",
4746             "name": "if",
4747             "type": "string",

```

```

4748         "enum": ["oic.if.b", "oic.if.ll", "oic.if.baseline"]
4749     },
4750 },
4751 "definitions": {
4752     "links": {
4753         "type": "array",
4754         "items": {
4755             "$ref": "#/definitions/oic.oic-link"
4756         }
4757     },
4758     "batch-retrieve": {
4759         "title": "Collection Batch Retrieve Format (auto merged)",
4760         "minItems": 1,
4761         "items": {
4762             "additionalProperties": true,
4763             "properties": {
4764                 "href": {
4765                     "$ref":
4766 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4767 schema.json#/definitions/href"
4768                 },
4769                 "rep": {
4770                     "oneOf": [{
4771                         "description": "The response payload from a
4772 single Resource",
4773                         "type": "object"
4774                     },
4775                     {
4776                         "description": " The response payload from a
4777 Collection (batch) Resource",
4778                         "items": {
4779                             "properties": {
4780                                 "anchor": {
4781                                     "$ref":
4782 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4783 schema.json#/definitions/anchor"
4784                                 },
4785                                 "di": {
4786                                     "$ref":
4787 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4788 schema.json#/definitions/di"
4789                                 },
4790                                 "eps": {
4791                                     "$ref":
4792 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4793 schema.json#/definitions/eps"
4794                                 },
4795                                 "href": {
4796                                     "$ref":
4797 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4798 schema.json#/definitions/href"
4799                                 },
4800                                 "if": {
4801                                     "description": "The OCF
4802 Interface set supported by this Resource",
4803                                     "items": {
4804                                         "enum": [
4805                                             "oic.if.baseline",
4806                                             "oic.if.ll",
4807                                             "oic.if.b",
4808                                             "oic.if.rw",
4809                                             "oic.if.r",
4810                                             "oic.if.a",
4811                                             "oic.if.s"],
4812                                         "type":
4813 "string"
4814                                     },
4815                                     "minItems": 1,
4816                                     "uniqueItems": true,
4817                                     "type": "array"
4818                                 }
4819                             }
4820                         }
4821                     }
4822                 }
4823             }
4824         }
4825     }
4826 }

```

```

4819                                     },
4820                                     "ins": {
4821                                         "$ref":
4822 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4823 schema.json#/definitions/ins"
4824                                     },
4825                                     "p": {
4826                                         "$ref":
4827 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4828 schema.json#/definitions/p"
4829                                     },
4830                                     "rel": {
4831                                         "description": "The relation of the target URI
4832 referenced by the Link to the context URI",
4833                                         "oneOf": [
4834                                             {
4835                                                 "$ref":
4836 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4837 schema.json#/definitions/rel_array"
4838                                             },
4839                                             {
4840                                                 "$ref":
4841 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4842 schema.json#/definitions/rel_string"
4843                                             }
4844                                         ]
4845                                     },
4846                                     "rt": {
4847                                         "description":
4848 "Resource Type of the Resource",
4849                                         "items": {
4850                                             "maxLength":
4851 64,
4852                                             "type":
4853 "string"
4854                                         },
4855                                         "minItems": 1,
4856                                         "uniqueItems": true,
4857                                         "type": "array"
4858                                     },
4859                                     "title": {
4860                                         "$ref":
4861 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4862 schema.json#/definitions/title"
4863                                     },
4864                                     "type": {
4865                                         "$ref":
4866 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4867 schema.json#/definitions/type"
4868                                     }
4869                                     },
4870                                     "required": [
4871                                         "href",
4872                                         "rt",
4873                                         "if"
4874                                     ],
4875                                     "type": "object"
4876                                 },
4877                                 "type": "array"
4878                            }
4879                        }
4880                    },
4881                    "required": [
4882                        "href",
4883                        "rep"
4884                    ],
4885                    "type": "object"
4886                },
4887                "type": "array"
4888            },
4889            "baseline": {

```

```

4890         "properties": {
4891             "links": {
4892                 "description": "A set of simple or individual Links.",
4893                 "items": {
4894                     "$ref": "#/definitions/oic.oic-link"
4895                 },
4896                 "type": "array"
4897             },
4898             "n": { "$ref" :
4899 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
4900 schema.json#/definitions/n"},
4901             "id": { "$ref" :
4902 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
4903 schema.json#/definitions/id"},
4904             "rt": {
4905                 "description": "Resource Type of this Resource",
4906                 "items": {
4907                     "enum": ["oic.wk.atomicmeasurement"],
4908                     "type": "string",
4909                     "maxLength": 64
4910                 },
4911                 "minItems": 1,
4912                 "readOnly": true,
4913                 "uniqueItems": true,
4914                 "type": "array"
4915             },
4916             "rts": {
4917                 "description": "An array of Resource Types that are supported
4918 within an array of Links exposed by the Resource",
4919                 "items": {
4920                     "maxLength": 64,
4921                     "type": "string"
4922                 },
4923                 "minItems": 1,
4924                 "readOnly": true,
4925                 "uniqueItems": true,
4926                 "type": "array"
4927             },
4928             "rts-m": {
4929                 "description": "An array of Resource Types that are mandatory
4930 to be exposed within an array of Links exposed by the Resource",
4931                 "items": {
4932                     "maxLength": 64,
4933                     "type": "string"
4934                 },
4935                 "minItems": 1,
4936                 "readOnly": true,
4937                 "uniqueItems": true,
4938                 "type": "array"
4939             },
4940             "if": {
4941                 "description": "The OCF Interface set supported by this
4942 Resource",
4943                 "items": {
4944                     "enum": ["oic.if.b", "oic.if.ll", "oic.if.baseline"],
4945                     "type": "string"
4946                 },
4947                 "minItems": 3,
4948                 "readOnly": true,
4949                 "uniqueItems": true,
4950                 "type": "array"
4951             }
4952         },
4953         "type": "object",
4954         "required": [
4955             "rt",
4956             "if",
4957             "links"
4958         ]
4959     },
4960     "oic.oic-link": {

```

```

4961         "properties": {
4962             "anchor": {
4963                 "$ref":
4964                 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4965                 schema.json#/definitions/anchor"
4966             },
4967             "di": {
4968                 "$ref":
4969                 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4970                 schema.json#/definitions/di"
4971             },
4972             "eps": {
4973                 "$ref":
4974                 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4975                 schema.json#/definitions/eps"
4976             },
4977             "href": {
4978                 "$ref":
4979                 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4980                 schema.json#/definitions/href"
4981             },
4982             "if": {
4983                 "description": "The OCF Interface set supported by this
4984 Resource",
4985                 "items": {
4986                     "enum": [
4987                         "oic.if.baseline",
4988                         "oic.if.ll",
4989                         "oic.if.b",
4990                         "oic.if.rw",
4991                         "oic.if.r",
4992                         "oic.if.a",
4993                         "oic.if.s"],
4994                     "type": "string"
4995                 },
4996                 "minItems": 1,
4997                 "uniqueItems": true,
4998                 "type": "array"
4999             },
5000             "ins": {
5001                 "$ref":
5002                 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5003                 schema.json#/definitions/ins"
5004             },
5005             "p": {
5006                 "$ref":
5007                 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5008                 schema.json#/definitions/p"
5009             },
5010             "rel": {
5011                 "description": "The relation of the target URI referenced by the Link to the context URI",
5012                 "oneOf": [
5013                     {
5014                         "$ref":
5015                         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5016                         schema.json#/definitions/rel_array"
5017                     },
5018                     {
5019                         "$ref":
5020                         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5021                         schema.json#/definitions/rel_string"
5022                     }
5023                 ]
5024             },
5025             "rt": {
5026                 "description": "Resource Type of the Resource",
5027                 "items": {
5028                     "maxLength": 64,
5029                     "type": "string"
5030                 },
5031                 "minItems": 1,

```



```

5032         "uniqueItems": true,
5033         "type": "array"
5034     },
5035     "title": {
5036         "$ref":
5037         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5038         schema.json#/definitions/title"
5039     },
5040     "type": {
5041         "$ref":
5042         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5043         schema.json#/definitions/type"
5044     }
5045     },
5046     "required": [
5047         "href",
5048         "rt",
5049         "if"
5050     ],
5051     "type": "object"
5052 }
5053 }
5054 }
5055

```

5056 D.2.5 Property definition

5057 Table D-2 defines the Properties that are part of the "oic.wk.atomicmeasurement" Resource Type.

5058 **Table D-2 – The Property definitions of the Resource with type "rt" =**
5059 **"oic.wk.atomicmeasurement".**

Property name	Value type	Mandatory	Access mode	Description
href	multiple types: see schema	Yes	Read Write	
rep	multiple types: see schema	Yes	Read Write	
links	array: see schema	Yes	Read Write	A set of simple or individual Links.
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
rt	array: see schema	Yes	Read Only	Resource Type of this Resource.
rts	array: see schema	No	Read Only	An array of Resource Types that are supported within an array of Links exposed by the Resource.
rts-m	array: see schema	No	Read Only	An array of Resource Types that are mandatory to be exposed within an array of Links exposed by the Resource.
if	array: see schema	Yes	Read Only	The OCF Interface set supported by this Resource.

anchor	multiple types: see schema	No	Read Write	
di	multiple types: see schema	No	Read Write	
eps	multiple types: see schema	No	Read Write	
href	multiple types: see schema	Yes	Read Write	
if	array: see schema	Yes	Read Write	The OCF Interface set supported by this Resource.
ins	multiple types: see schema	No	Read Write	
p	multiple types: see schema	No	Read Write	
rel	multiple types: see schema	No	Read Write	The relation of the target URI referenced by the Link to the context URI.
rt	array: see schema	Yes	Read Write	Resource Type of the Resource.
title	multiple types: see schema	No	Read Write	
type	multiple types: see schema	No	Read Write	

5060 **D.2.6 CRUDN behaviour**

5061 Table D-3 defines the CRUDN operations that are supported on the "oic.wk.atomicmeasurement"
5062 Resource Type.

5063 **Table D-3 – The CRUDN operations of the Resource with type "rt" =**
5064 **"oic.wk.atomicmeasurement".**

Create	Read	Update	Delete	Notify
	get			observe

5065 **D.3 Collection**

5066 **D.3.1 Introduction**

5067 Collection Resource Type contains Properties and Links.
5068 The oic.if.baseline OCF Interface exposes a representation of
5069 the Links and the Properties of the Collection Resource itself
5070

5071 **D.3.2 Example URI**

5072 /CollectionResURI

5073 **D.3.3 Resource type**

5074 The Resource Type is defined as: "oic.wk.col".

5075 **D.3.4 OpenAPI 2.0 definition**

5076 {
5077 "swagger": "2.0",

```

5078 "info": {
5079   "title": "Collection",
5080   "version": "2019-03-04",
5081   "license": {
5082     "name": "OCF Data Model License",
5083     "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
5084     "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
5085   },
5086   "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
5087 },
5088 "schemes": [
5089   "http"
5090 ],
5091 "consumes": [
5092   "application/json"
5093 ],
5094 "produces": [
5095   "application/json"
5096 ],
5097 "paths": {
5098   "/CollectionResURI?if=oic.if.ll" : {
5099     "get": {
5100       "description": "Collection Resource Type contains Properties and Links.\n\nThe oic.if.ll OCF
Interface exposes a representation of the Links\n",
5101       "parameters": [
5102         {
5103           "$ref": "#/parameters/interface-all"
5104         }
5105       ],
5106       "responses": {
5107         "200": {
5108           "description": "",
5109           "x-example": [
5110             {
5111               "href": "/switch",
5112               "rt": ["oic.r.switch.binary"],
5113               "if": ["oic.if.a", "oic.if.baseline"],
5114               "eps": [
5115                 {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
5116                 {"ep": "coaps://[fe80::b1d6]:1122"},
5117                 {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
5118               ]
5119             }
5120           ],
5121           {
5122             "href": "/airFlow",
5123             "rt": ["oic.r.airflow"],
5124             "if": ["oic.if.a", "oic.if.baseline"],
5125             "eps": [
5126               {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
5127               {"ep": "coaps://[fe80::b1d6]:1122"},
5128               {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
5129             ]
5130           }
5131         ]
5132       },
5133       "schema": {
5134         "$ref": "#/definitions/slinks"
5135       }
5136     }
5137   }
5138 },
5139 "/CollectionResURI?if=oic.if.baseline" : {
5140   "get": {
5141     "description": "Collection Resource Type contains Properties and Links.\n\nThe oic.if.baseline
OCF Interface exposes a representation of\nthe Links and the Properties of the Collection Resource
itself\n",
5142     "parameters": [
5143       {
5144         "$ref": "#/parameters/interface-all"
5145       }
5146     ]
5147   }
5148 },

```

```

5149     "responses": {
5150       "200": {
5151         "description" : "",
5152         "x-example": {
5153           "rt": ["oic.wk.col"],
5154           "if": ["oic.if.ll", "oic.if.b", "oic.if.baseline"],
5155           "rts": [ "oic.r.switch.binary", "oic.r.airflow" ],
5156           "rts-m": [ "oic.r.switch.binary" ],
5157           "links": [
5158             {
5159               "href": "/switch",
5160               "rt": [ "oic.r.switch.binary" ],
5161               "if": [ "oic.if.a", "oic.if.baseline" ],
5162               "eps": [
5163                 { "ep": "coap://[fe80::b1d6]:1111", "pri": 2 },
5164                 { "ep": "coaps://[fe80::b1d6]:1122" },
5165                 { "ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3 }
5166               ]
5167             },
5168             {
5169               "href": "/airFlow",
5170               "rt": [ "oic.r.airflow" ],
5171               "if": [ "oic.if.a", "oic.if.baseline" ],
5172               "eps": [
5173                 { "ep": "coap://[fe80::b1d6]:1111", "pri": 2 },
5174                 { "ep": "coaps://[fe80::b1d6]:1122" },
5175                 { "ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3 }
5176               ]
5177             }
5178           ]
5179         },
5180         "schema": {
5181           "$ref": "#/definitions/sbaseline"
5182         }
5183       }
5184     },
5185     "post": {
5186       "description": "Update on Baseline OCF Interface\n",
5187       "parameters": [
5188         {
5189           "$ref": "#/parameters/interface-update"
5190         },
5191         {
5192           "name": "body",
5193           "in": "body",
5194           "required": true,
5195           "schema": {
5196             "$ref": "#/definitions/sbaseline-update"
5197           }
5198         }
5199       ]
5200     },
5201     "responses": {
5202       "200": {
5203         "description" : "",
5204         "schema": {
5205           "$ref": "#/definitions/sbaseline"
5206         }
5207       }
5208     }
5209   },
5210 ],
5211 "/CollectionResURI?if=oic.if.b" : {
5212   "get": {
5213     "description": "Collection Resource Type contains Properties and Links.\n\nThe oic.if.b OCF
5214 Interface exposes a composite representation of the\nResources pointed to by the Links\n",
5215     "parameters": [
5216       {
5217         "$ref": "#/parameters/interface-all"
5218       }
5219     ]

```

```

5220     "responses": {
5221       "200": {
5222         "description": "All targets returned OK status",
5223         "x-example": [
5224           {
5225             "href": "/switch",
5226             "rep": {
5227               "value": true
5228             }
5229           },
5230           {
5231             "href": "/airFlow",
5232             "rep": {
5233               "direction": "floor",
5234               "speed": 3
5235             }
5236           }
5237         ],
5238         "schema": {
5239           "$ref": "#/definitions/sbatch-retrieve"
5240         }
5241       },
5242       "404": {
5243         "description": "One or more targets did not return an OK status, return a
5244 representation containing returned Properties from the targets that returned OK",
5245         "x-example": [
5246           {
5247             "href": "/switch",
5248             "rep": {
5249               "value": true
5250             }
5251           }
5252         ],
5253         "schema": {
5254           "$ref": "#/definitions/sbatch-retrieve"
5255         }
5256       }
5257     },
5258     "post": {
5259       "description": "Update on Batch OCF Interface\n",
5260       "parameters": [
5261         {
5262           "$ref": "#/parameters/interface-update"
5263         },
5264         {
5265           "name": "body",
5266           "in": "body",
5267           "required": true,
5268           "schema": {
5269             "$ref": "#/definitions/sbatch-update"
5270           }
5271         },
5272         "x-example": [
5273           {
5274             "href": "/switch",
5275             "rep": {
5276               "value": true
5277             }
5278           },
5279           {
5280             "href": "/airFlow",
5281             "rep": {
5282               "direction": "floor",
5283               "speed": 3
5284             }
5285           }
5286         ]
5287       }
5288     ],
5289     "responses": {
5290       "200": {

```

```

5291         "description" : "All targets returned OK status, return a representation of the current
5292 state of all targets",
5293         "x-example": [
5294             {
5295                 "href": "/switch",
5296                 "rep": {
5297                     "value": true
5298                 }
5299             },
5300             {
5301                 "href": "/airFlow",
5302                 "rep": {
5303                     "direction": "demist",
5304                     "speed": 5
5305                 }
5306             }
5307         ],
5308         "schema": {
5309             "$ref": "#/definitions/sbatch-retrieve"
5310         }
5311     },
5312     "403": {
5313         "description" : "One or more targets did not return OK status; return a retrieve
5314 representation of the current state of all targets in the batch",
5315         "x-example": [
5316             {
5317                 "href": "/switch",
5318                 "rep": {
5319                     "value": true
5320                 }
5321             },
5322             {
5323                 "href": "/airFlow",
5324                 "rep": {
5325                     "direction": "floor",
5326                     "speed": 3
5327                 }
5328             }
5329         ],
5330         "schema": {
5331             "$ref": "#/definitions/sbatch-retrieve"
5332         }
5333     }
5334 }
5335 }
5336 },
5337 },
5338 "parameters": {
5339     "interface-all" : {
5340         "in" : "query",
5341         "name" : "if",
5342         "type" : "string",
5343         "enum" : ["oic.if.ll", "oic.if.b", "oic.if.baseline"]
5344     },
5345     "interface-update" : {
5346         "in" : "query",
5347         "name" : "if",
5348         "type" : "string",
5349         "enum" : ["oic.if.b", "oic.if.baseline"]
5350     }
5351 },
5352 "definitions": {
5353     "sbaseline" : {
5354         "properties": {
5355             "links" : {
5356                 "description": "A set of simple or individual Links.",
5357                 "items": {
5358                     "$ref": "#/definitions/oic.oic-link"
5359                 },
5360                 "type": "array"
5361             },

```

```

5362         "n": {
5363             "$ref" :
5364 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
5365 schema.json#/definitions/n"
5366         },
5367         "id": {
5368             "$ref" :
5369 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
5370 schema.json#/definitions/id"
5371         },
5372         "rt": {
5373             "$ref": "#/definitions/oic.core.rt-col"
5374         },
5375         "rts": {
5376             "$ref": "#/definitions/oic.core.rt"
5377         },
5378         "rts-m": {
5379             "$ref": "#/definitions/oic.core.rt"
5380         },
5381         "if": {
5382             "description": "The OCF Interfaces supported by this Resource",
5383             "items": {
5384                 "enum": [
5385                     "oic.if.ll",
5386                     "oic.if.baseline",
5387                     "oic.if.b"
5388                 ],
5389                 "type": "string",
5390                 "maxLength": 64
5391             },
5392             "minItems": 2,
5393             "uniqueItems": true,
5394             "readOnly": true,
5395             "type": "array"
5396         },
5397     },
5398     "additionalProperties": true,
5399     "type" : "object",
5400     "required": [
5401         "rt",
5402         "if",
5403         "links"
5404     ]
5405 },
5406 "sbaseline-update": {
5407     "additionalProperties": true
5408 },
5409     "oic.core.rt-col": {
5410         "description": "Resource Type of the Resource",
5411         "items": {
5412             "enum": ["oic.wk.col"],
5413             "type": "string",
5414             "maxLength": 64
5415         },
5416         "minItems": 1,
5417         "uniqueItems": true,
5418         "readOnly": true,
5419         "type": "array"
5420     },
5421     "oic.core.rt": {
5422         "description": "Resource Type or set of Resource Types",
5423         "items": {
5424             "type": "string",
5425             "maxLength": 64
5426         },
5427         "minItems": 1,
5428         "uniqueItems": true,
5429         "readOnly": true,
5430         "type": "array"
5431     },
5432     "sbatch-retrieve" : {

```

```

5433     "minItems" : 1,
5434     "items" : {
5435       "additionalProperties": true,
5436       "properties": {
5437         "href": {
5438           "$ref":
5439 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5440 schema.json#/definitions/href"
5441         },
5442         "rep": {
5443           "oneOf": [
5444             {
5445               "description": "The response payload from a single Resource",
5446               "type": "object"
5447             },
5448             {
5449               "description": " The response payload from a Collection (batch) Resource",
5450               "items": {
5451                 "$ref": "#/definitions/oic.oic-link"
5452               },
5453               "type": "array"
5454             }
5455           ]
5456         },
5457       },
5458       "required": [
5459         "href",
5460         "rep"
5461       ],
5462       "type": "object"
5463     },
5464     "type" : "array"
5465   },
5466   "sbatch-update" : {
5467     "title" : "Collection Batch Update Format",
5468     "minItems" : 1,
5469     "items" : {
5470       "$ref": "#/definitions/sbatch-update.item"
5471     },
5472     "type" : "array"
5473   },
5474   "sbatch-update.item" : {
5475     "additionalProperties": true,
5476     "description": "Array of Resource representations to apply to the batch Collection, using href
5477 to indicate which Resource(s) in the batch to update. If the href Property is empty, effectively
5478 making the URI reference to the Collection itself, the representation is to be applied to all
5479 Resources in the batch",
5480     "properties": {
5481       "href": {
5482         "$ref":
5483 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5484 schema.json#/definitions/href"
5485       },
5486       "rep": {
5487         "oneOf": [
5488           {
5489             "description": "The payload for a single Resource",
5490             "type": "object"
5491           },
5492           {
5493             "description": " The payload for a Collection (batch) Resource",
5494             "items": {
5495               "$ref": "#/definitions/oic.oic-link"
5496             },
5497             "type": "array"
5498           }
5499         ]
5500       },
5501     },
5502     "required": [
5503       "href",

```



```

5504     "rep"
5505     ],
5506     "type": "object"
5507   },
5508   "slinks" : {
5509     "type" : "array",
5510     "items" : {
5511       "$ref": "#/definitions/oic.oic-link"
5512     }
5513   },
5514   "oic.oic-link": {
5515     "properties": {
5516       "if": {
5517         "description": "The OCF Interfaces supported by the Linked target",
5518         "items": {
5519           "enum": [
5520             "oic.if.baseline",
5521             "oic.if.ll",
5522             "oic.if.b",
5523             "oic.if.rw",
5524             "oic.if.r",
5525             "oic.if.a",
5526             "oic.if.s"
5527           ],
5528           "type": "string",
5529           "maxLength": 64
5530         },
5531         "minItems": 1,
5532         "uniqueItems": true,
5533         "readOnly": true,
5534         "type": "array"
5535       },
5536       "rt": {
5537         "$ref": "#/definitions/oic.core.rt"
5538       },
5539       "anchor": {
5540         "$ref":
5541 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5542 schema.json#/definitions/anchor"
5543       },
5544       "di": {
5545         "$ref":
5546 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5547 schema.json#/definitions/di"
5548       },
5549       "eps": {
5550         "$ref":
5551 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5552 schema.json#/definitions/eps"
5553       },
5554       "href": {
5555         "$ref":
5556 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5557 schema.json#/definitions/href"
5558       },
5559       "ins": {
5560         "$ref":
5561 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5562 schema.json#/definitions/ins"
5563       },
5564       "p": {
5565         "$ref":
5566 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5567 schema.json#/definitions/p"
5568       },
5569       "rel": {
5570         "$ref":
5571 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5572 schema.json#/definitions/rel_array"
5573       },
5574       "title": {

```

```

5575         "$ref":
5576         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5577         schema.json#/definitions/title"
5578         },
5579         "type": {
5580         "$ref":
5581         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5582         schema.json#/definitions/type"
5583         },
5584         },
5585         "required": [
5586         "href",
5587         "rt",
5588         "if"
5589         ],
5590         "type": "object"
5591     }
5592 }
5593 }
5594

```

D.3.5 Property definition

Table D-4 defines the Properties that are part of the "oic.wk.col" Resource Type.

Table D-4 – The Property definitions of the Resource with type "rt" = "oic.wk.col".

Property name	Value type	Mandatory	Access mode	Description
links	array: see schema	Yes	Read Write	A set of simple or individual Links.
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
rt	multiple types: see schema	Yes	Read Write	
rts	multiple types: see schema	No	Read Write	
rts-m	multiple types: see schema	No	Read Write	
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by this Resource.
href	multiple types: see schema	Yes	Read Write	
rep	multiple types: see schema	Yes	Read Write	
href	multiple types: see schema	Yes	Read Write	
rep	multiple types: see schema	Yes	Read Write	
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by the Linked target.
rt	multiple types: see schema	Yes	Read Write	
anchor	multiple types: see schema	No	Read Write	

di	multiple types: see schema	No	Read Write	
eps	multiple types: see schema	No	Read Write	
href	multiple types: see schema	Yes	Read Write	
ins	multiple types: see schema	No	Read Write	
p	multiple types: see schema	No	Read Write	
rel	multiple types: see schema	No	Read Write	
title	multiple types: see schema	No	Read Write	
type	multiple types: see schema	No	Read Write	

5598 **D.3.6 CRUDN behaviour**

5599 Table D-5 defines the CRUDN operations that are supported on the "oic.wk.col" Resource Type.

5600 **Table D-5 – The CRUDN operations of the Resource with type "rt" = "oic.wk.col".**

Create	Read	Update	Delete	Notify
	get	post		observe

5601 **D.4 Device Configuration**

5602 **D.4.1 Introduction**

5603 Resource that allows for Device specific information to be configured.
5604

5605 **D.4.2 Example URI**

5606 /exampleDeviceConfigurationResURI

5607 **D.4.3 Resource type**

5608 The Resource Type is defined as: "oic.wk.con".

5609 **D.4.4 OpenAPI 2.0 definition**

```
5610 {
5611   "swagger": "2.0",
5612   "info": {
5613     "title": "Device Configuration",
5614     "version": "2019-02-28",
5615     "license": {
5616       "name": "OCF Data Model License",
5617       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
5618       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
5619     },
5620     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
5621   },
5622   "schemes": [
5623     "http"
5624   ],
5625   "consumes": [
5626     "application/json"
5627   ],
5628   "produces": [
```

```

5629     "application/json"
5630 ],
5631 "paths": {
5632     "/exampleDeviceConfigurationResURI" : {
5633         "get": {
5634             "description": "Resource that allows for Device specific information to be configured.\n",
5635             "parameters": [
5636                 {
5637                     "$ref": "#/parameters/interface-all"
5638                 }
5639             ],
5640             "responses": {
5641                 "200": {
5642                     "description": "",
5643                     "x-example": {
5644                         "n": "My Friendly Device Name",
5645                         "rt": ["oic.wk.con"],
5646                         "loc": [32.777,-96.797],
5647                         "locn": "My Location Name",
5648                         "c": "USD",
5649                         "r": "MyRegion",
5650                         "dl": "en"
5651                     },
5652                     "schema": {
5653                         "$ref": "#/definitions/Configuration"
5654                     }
5655                 }
5656             }
5657         },
5658         "post": {
5659             "description": "Update the information about the Device\n",
5660             "parameters": [
5661                 {
5662                     "$ref": "#/parameters/interface-rw"
5663                 },
5664                 {
5665                     "name": "body",
5666                     "in": "body",
5667                     "required": true,
5668                     "schema": {
5669                         "$ref": "#/definitions/Update"
5670                     },
5671                     "x-example": {
5672                         "n": "Nuevo Nombre Amistoso",
5673                         "r": "MyNewRegion",
5674                         "ln": [ { "language": "es", "value": "Nuevo Nombre Amistoso" } ],
5675                         "dl": "es"
5676                     }
5677                 }
5678             ],
5679             "responses": {
5680                 "200": {
5681                     "description": "",
5682                     "x-example": {
5683                         "n": "Nuevo Nombre Amistoso",
5684                         "r": "MyNewRegion",
5685                         "ln": [ { "language": "es", "value": "Nuevo Nombre Amistoso" } ],
5686                         "dl": "es"
5687                     },
5688                     "schema": {
5689                         "$ref": "#/definitions/Update"
5690                     }
5691                 }
5692             }
5693         }
5694     }
5695 },
5696 "parameters": {
5697     "interface-rw" : {
5698         "in": "query",
5699         "name": "if",

```

```

5700     "type" : "string",
5701     "enum" : ["oic.if.rw"]
5702   },
5703   "interface-all" : {
5704     "in" : "query",
5705     "name" : "if",
5706     "type" : "string",
5707     "enum" : ["oic.if.rw", "oic.if.baseline"]
5708   }
5709 },
5710 "definitions": {
5711   "Configuration": {
5712     "properties": {
5713       "rt": {
5714         "description": "Resource Type of the Resource",
5715         "items": {
5716           "enum": ["oic.wk.con"],
5717           "type": "string",
5718           "maxLength": 64
5719         },
5720         "minItems": 1,
5721         "uniqueItems": true,
5722         "readOnly": true,
5723         "type": "array"
5724       },
5725       "loc": {
5726         "description": "Location information (lat, long)",
5727         "items": {
5728           "type": "number"
5729         },
5730         "maxItems": 2,
5731         "minItems": 2,
5732         "type": "array"
5733       },
5734       "c": {
5735         "description": "Currency",
5736         "maxLength": 64,
5737         "type": "string"
5738       },
5739       "ln": {
5740         "description": "Localized names",
5741         "items": {
5742           "properties": {
5743             "language": {
5744               "allOf": [
5745                 {
5746                   "description": "Format pattern according to IETF RFC 5646 (language tag).",
5747                   "pattern": "^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$",
5748                   "type": "string"
5749                 },
5750                 {
5751                   "description": "An RFC 5646 language tag."
5752                 }
5753               ]
5754             },
5755             "value": {
5756               "description": "The Device name in the indicated language.",
5757               "maxLength": 64,
5758               "type": "string"
5759             }
5760           },
5761           "type": "object"
5762         },
5763         "minItems": 1,
5764         "type": "array"
5765       },
5766       "locn": {
5767         "description": "Human Friendly Name for location",
5768         "maxLength": 64,
5769         "type": "string"
5770       },

```

```

5771     "dl": {
5772       "allOf": [
5773         {
5774           "description": "Format pattern according to IETF RFC 5646 (language tag).",
5775           "pattern": "^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$",
5776           "type": "string"
5777         },
5778         {
5779           "description": "Default Language as an RFC 5646 language tag."
5780         }
5781       ]
5782     },
5783     "n": {
5784       "$ref" :
5785       "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
5786       schema.json#/definitions/n"
5787     },
5788     "id": {
5789       "$ref" :
5790       "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
5791       schema.json#/definitions/id"
5792     },
5793     "r": {
5794       "description": "Region",
5795       "maxLength": 64,
5796       "type": "string"
5797     },
5798     "if" : {
5799       "description": "The OCF Interfaces supported by this Resource",
5800       "items": {
5801         "enum": [
5802           "oic.if.baseline",
5803           "oic.if.rw"
5804         ],
5805         "type": "string",
5806         "maxLength": 64
5807       },
5808       "minItems": 1,
5809       "uniqueItems": true,
5810       "readOnly": true,
5811       "type": "array"
5812     }
5813   },
5814   "type" : "object",
5815   "required": ["n"]
5816 },
5817 "Update" : {
5818   "properties": {
5819     "loc": {
5820       "description": "Location information (lat, long)",
5821       "items": {
5822         "type": "number"
5823       },
5824       "maxItems": 2,
5825       "minItems": 2,
5826       "type": "array"
5827     },
5828     "c": {
5829       "description": "Currency",
5830       "maxLength": 64,
5831       "type": "string"
5832     },
5833     "ln": {
5834       "description": "Localized names",
5835       "items": {
5836         "properties": {
5837           "language": {
5838             "allOf": [
5839               {
5840                 "description": "Format pattern according to IETF RFC 5646 (language tag).",
5841                 "pattern": "^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$",

```

```

5842         "type": "string"
5843     },
5844     {
5845         "description": "An RFC 5646 language tag."
5846     }
5847 ]
5848 },
5849 "value": {
5850     "description": "The Device name in the indicated language.",
5851     "maxLength": 64,
5852     "type": "string"
5853 },
5854 },
5855 "type": "object"
5856 },
5857 "minItems": 1,
5858 "type": "array"
5859 },
5860 "locn": {
5861     "description": "Human Friendly Name for location",
5862     "maxLength": 64,
5863     "type": "string"
5864 },
5865 "dl": {
5866     "allOf": [
5867         {
5868             "description": "Format pattern according to IETF RFC 5646 (language tag).",
5869             "pattern": "^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$",
5870             "type": "string"
5871         },
5872         {
5873             "description": "Default Language as an RFC 5646 language tag."
5874         }
5875     ]
5876 },
5877 "n": {
5878     "description": "The human friendly name to be set on the Resource, this is also reflected
5879 in the same Property in oic.wk.d",
5880     "maxLength": 64,
5881     "type": "string"
5882 },
5883 "r": {
5884     "description": "Region",
5885     "maxLength": 64,
5886     "type": "string"
5887 },
5888 },
5889 "anyOf": [
5890     {
5891         "required": ["loc"]
5892     },
5893     {
5894         "required": ["locn"]
5895     },
5896     {
5897         "required": ["c"]
5898     },
5899     {
5900         "required": ["r"]
5901     },
5902     {
5903         "required": ["ln"]
5904     },
5905     {
5906         "required": ["dl"]
5907     },
5908     {
5909         "required": ["n"]
5910     }
5911 ],
5912 "type" : "object"

```

5913 }
 5914 }
 5915 }
 5916 }

5917 **D.4.5 Property definition**

5918 Table D-6 defines the Properties that are part of the "oic.wk.con" Resource Type.

5919 **Table D-6 – The Property definitions of the Resource with type "rt" = "oic.wk.con".**

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	Resource Type of the Resource.
loc	array: see schema	No	Read Write	Location information (lat, long).
c	string	No	Read Write	Currency.
ln	array: see schema	No	Read Write	Localized names.
locn	string	No	Read Write	Human Friendly Name for location.
dl	multiple types: see schema	No	Read Write	
n	multiple types: see schema	Yes	Read Write	
id	multiple types: see schema	No	Read Write	
r	string	No	Read Write	Region.
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.
loc	array: see schema	No	Read Write	Location information (lat, long).
c	string	No	Read Write	Currency.
ln	array: see schema	No	Read Write	Localized names.
locn	string	No	Read Write	Human Friendly Name for location.
dl	multiple types: see schema	No	Read Write	
n	string	Yes	Read Write	The human friendly name to be set on the Resource, this is also reflected in the same Property in oic.wk.d.
r	string	No	Read Write	Region.

5920 **D.4.6 CRUDN behaviour**

5921 Table D-7 defines the CRUDN operations that are supported on the "oic.wk.con" Resource Type.

5922 **Table D-7 – The CRUDN operations of the Resource with type "rt" = "oic.wk.con".**

Create	Read	Update	Delete	Notify
	get	post		observe

5923 D.5 Platform Configuration

5924 D.5.1 Introduction

5925 Resource that allows for Platform specific information to be configured.
5926

5927 D.5.2 Example URI

5928 /examplePlatformConfigurationResURI

5929 D.5.3 Resource type

5930 The Resource Type is defined as: "oic.wk.con.p".

5931 D.5.4 OpenAPI 2.0 definition

```
5932 {  
5933   "swagger": "2.0",  
5934   "info": {  
5935     "title": "Platform Configuration",  
5936     "version": "2019-03-04",  
5937     "license": {  
5938       "name": "OCF Data Model License",  
5939       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",  
5940       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."  
5941     },  
5942     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"  
5943   },  
5944   "schemes": [  
5945     "http"  
5946   ],  
5947   "consumes": [  
5948     "application/json"  
5949   ],  
5950   "produces": [  
5951     "application/json"  
5952   ],  
5953   "paths": {  
5954     "/examplePlatformConfigurationResURI": {  
5955       "get": {  
5956         "description": "Resource that allows for Platform specific information to be configured.\n",  
5957         "parameters": [  
5958           {  
5959             "$ref": "#/parameters/interface-all"  
5960           }  
5961         ],  
5962         "responses": {  
5963           "200": {  
5964             "description": "",  
5965             "x-example": {  
5966               "rt": ["oic.wk.con.p"],  
5967               "mnpn": [ { "language": "en", "value": "My Friendly Device Name" } ]  
5968             },  
5969             "schema": { "$ref": "#/definitions/Conf_Platform" }  
5970           }  
5971         }  
5972       },  
5973       "post": {  
5974         "description": "Update the information about the Platform\n",  
5975         "parameters": [  
5976           {  
5977             "$ref": "#/parameters/interface-rw"  
5978           },  
5979           {  
5980             "name": "body",  
5981             "in": "body",  
5982             "required": true,  
5983             "schema": { "$ref": "#/definitions/Update_Platform" },  
5984             "x-example": {
```

```

5985         "n": "Nuevo nombre",
5986         "mnpn": [ { "language": "es", "value": "Nuevo nombre de Plataforma Amigable" } ]
5987     }
5988 }
5989 ],
5990 "responses": {
5991     "200": {
5992         "description": "",
5993         "x-example": {
5994             "n": "Nuevo nombre",
5995             "mnpn": [ { "language": "es", "value": "Nuevo nombre de Plataforma Amigable" } ]
5996         },
5997         "schema": { "$ref": "#/definitions/Update_Platform" }
5998     }
5999 }
6000 }
6001 }
6002 },
6003 "parameters": {
6004     "interface-rw": {
6005         "in": "query",
6006         "name": "if",
6007         "type": "string",
6008         "enum": ["oic.if.rw"]
6009     },
6010     "interface-all": {
6011         "in": "query",
6012         "name": "if",
6013         "type": "string",
6014         "enum": ["oic.if.rw", "oic.if.baseline"]
6015     }
6016 },
6017 "definitions": {
6018     "Conf_Platform": {
6019         "properties": {
6020             "rt": {
6021                 "description": "Resource Type of the Resource",
6022                 "items": {
6023                     "enum": ["oic.wk.con.p"],
6024                     "type": "string",
6025                     "maxLength": 64
6026                 },
6027                 "minItems": 1,
6028                 "uniqueItems": true,
6029                 "readOnly": true,
6030                 "type": "array"
6031             },
6032             "n": {
6033                 "$ref":
6034 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6035 schema.json#/definitions/n"
6036             },
6037             "mnpn": {
6038                 "description": "Platform names",
6039                 "items": {
6040                     "properties": {
6041                         "language": {
6042                             "allOf": [
6043                                 {
6044                                     "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
6045 schema.json#/definitions/language-tag"
6046                                 },
6047                                 {
6048                                     "description": "An RFC 5646 language tag."
6049                                 }
6050                             ]
6051                         },
6052                         "value": {
6053                             "description": "The Platform description in the indicated language.",
6054                             "maxLength": 64,
6055                             "type": "string"

```

```

6056     }
6057     },
6058     "type": "object"
6059   },
6060   "minItems": 1,
6061   "type": "array"
6062 },
6063 "id": {
6064   "$ref":
6065 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6066 schema.json#/definitions/id"
6067 },
6068 "if": {
6069   "description": "The OCF Interfaces supported by this Resource",
6070   "items": {
6071     "enum": [
6072       "oic.if.rw",
6073       "oic.if.baseline"
6074     ],
6075     "type": "string",
6076     "maxLength": 64
6077   },
6078   "minItems": 1,
6079   "readOnly": true,
6080   "uniqueItems": true,
6081   "type": "array"
6082 }
6083 },
6084 "type" : "object"
6085 },
6086 "Update_Platform": {
6087   "properties": {
6088     "n": {
6089       "description": "The human friendly name to be set on the Resource, this is also reflected
6090 in the same Property in oic.wk.p",
6091       "maxLength": 64,
6092       "type": "string"
6093     },
6094     "mnpn" : {
6095       "description": "Platform names",
6096       "items": {
6097         "properties": {
6098           "language": {
6099             "allOf": [
6100               {
6101                 "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
6102 schema.json#/definitions/language-tag"
6103               },
6104               {
6105                 "description": "An RFC 5646 language tag."
6106               }
6107             ]
6108           },
6109           "value": {
6110             "description": "The Platform description in the indicated language.",
6111             "maxLength": 64,
6112             "type": "string"
6113           }
6114         },
6115         "type": "object"
6116       },
6117       "minItems": 1,
6118       "type": "array"
6119     }
6120   },
6121   "type": "object",
6122   "anyOf": [
6123     {
6124       "required": ["mnpn"]
6125     },
6126     {

```

```

6127         "required": ["n"]
6128     }
6129 ]
6130 }
6131 }
6132 }
6133

```

6134 D.5.5 Property definition

6135 Table D-8 defines the Properties that are part of the "oic.wk.con.p" Resource Type.

6136 **Table D-8 – The Property definitions of the Resource with type "rt" = "oic.wk.con.p".**

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema		Read Only	Resource Type of the Resource.
n	multiple types: see schema		Read Write	
mnpn	array: see schema		Read Write	Platform names.
id	multiple types: see schema		Read Write	
if	array: see schema		Read Only	The OCF Interfaces supported by this Resource.
n	string	Yes	Read Write	The human friendly name to be set on the Resource, this is also reflected in the same Property in oic.wk.p.
mnpn	array: see schema	No	Read Write	Platform names.

6137 D.5.6 CRUDN behaviour

6138 Table D-9 defines the CRUDN operations that are supported on the "oic.wk.con.p" Resource Type.

6139 **Table D-9 – The CRUDN operations of the Resource with type "rt" = "oic.wk.con.p".**

Create	Read	Update	Delete	Notify
	get	post		observe

6140 D.6 Device

6141 D.6.1 Introduction

6142 Known Resource that is hosted by every Server.
 6143 Allows for logical Device specific information to be discovered.
 6144

6145 D.6.2 Well-known URI

6146 /oic/d

6147 D.6.3 Resource type

6148 The Resource Type is defined as: "oic.wk.d".

6149 D.6.4 OpenAPI 2.0 definition

```
6150 {
6151   "swagger": "2.0",
6152   "info": {
6153     "title": "Device",
6154     "version": "2019-03-13",
6155     "license": {
6156       "name": "OCF Data Model License",
6157       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
6158       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
6159     },
6160     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
6161   },
6162   "schemes": [
6163     "http"
6164   ],
6165   "consumes": [
6166     "application/json"
6167   ],
6168   "produces": [
6169     "application/json"
6170   ],
6171   "paths": {
6172     "/oic/d" : {
6173       "get": {
6174         "description": "Known Resource that is hosted by every Server.\nAllows for logical Device
6175 specific information to be discovered.\n",
6176         "parameters": [
6177           {
6178             "$ref": "#/parameters/interface"
6179           }
6180         ],
6181         "responses": {
6182           "200": {
6183             "description": "",
6184             "x-example":
6185               {
6186                 "n": "Device 1",
6187                 "rt": ["oic.wk.d"],
6188                 "di": "54919CA5-4101-4AE4-595B-353C51AA983C",
6189                 "icv": "ocf.2.0.2",
6190                 "dmv": "ocf.res.1.0.0, ocf.sh.1.0.0",
6191                 "piid": "6F0AAC04-2EB0-468D-B57C-16570A26AE48"
6192               },
6193             "schema": {
6194               "$ref": "#/definitions/Device"
6195             }
6196           }
6197         }
6198       }
6199     }
6200   },
6201   "parameters": {
6202     "interface" : {
6203       "in": "query",
6204       "name": "if",
6205       "type": "string",
6206       "enum": ["oic.if.r", "oic.if.baseline"]
6207     }
6208   },
6209   "definitions": {
6210     "Device": {
6211       "properties": {
6212         "rt": {
6213           "description": "Resource Type of the Resource",
6214           "items": {
6215             "type": "string",
6216             "maxLength": 64
6217           },
6218           "minItems": 1,

```

```

6219         "readOnly": true,
6220         "uniqueItems": true,
6221         "type": "array"
6222     },
6223     "ld": {
6224         "description": "Localized Descriptions.",
6225         "items": {
6226             "properties": {
6227                 "language": {
6228                     "allOf": [
6229                         {
6230                             "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
6231 schema.json#/definitions/language-tag"
6232                         },
6233                         {
6234                             "description": "An RFC 5646 language tag.",
6235                             "readOnly": true
6236                         }
6237                     ]
6238                 },
6239                 "value": {
6240                     "description": "Device description in the indicated language.",
6241                     "maxLength": 64,
6242                     "readOnly": true,
6243                     "type": "string"
6244                 }
6245             },
6246             "type": "object"
6247         },
6248         "minItems": 1,
6249         "readOnly": true,
6250         "type": "array"
6251     },
6252     "piid": {
6253         "allOf": [
6254             {
6255                 "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
6256 schema.json#/definitions/uuid"
6257             },
6258             {
6259                 "description": "Protocol independent unique identifier for the Device that is
6260 immutable.",
6261                 "readOnly": true
6262             }
6263         ]
6264     },
6265     "di": {
6266         "allOf": [
6267             {
6268                 "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
6269 schema.json#/definitions/uuid"
6270             },
6271             {
6272                 "description": "Unique identifier for the Device",
6273                 "readOnly": true
6274             }
6275         ]
6276     },
6277     "dmno": {
6278         "description": "Model number as designated by manufacturer.",
6279         "maxLength": 64,
6280         "readOnly": true,
6281         "type": "string"
6282     },
6283     "sv": {
6284         "description": "Software version.",
6285         "maxLength": 64,
6286         "readOnly": true,
6287         "type": "string"
6288     },
6289     "dmn": {

```

```

6290         "description": "Manufacturer Name.",
6291         "items": {
6292             "properties": {
6293                 "language": {
6294                     "allOf": [
6295                         {
6296                             "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
6297 schema.json#/definitions/language-tag"
6298                         },
6299                         {
6300                             "description": "An RFC 5646 language tag.",
6301                             "readOnly": true
6302                         }
6303                     ]
6304                 },
6305                 "value": {
6306                     "description": "Manufacturer name in the indicated language.",
6307                     "maxLength": 64,
6308                     "readOnly": true,
6309                     "type": "string"
6310                 }
6311             },
6312             "type": "object"
6313         },
6314         "minItems": 1,
6315         "readOnly": true,
6316         "type": "array"
6317     },
6318     "icv": {
6319         "description": "The version of the Device",
6320         "maxLength": 64,
6321         "readOnly": true,
6322         "type": "string"
6323     },
6324     "dmv": {
6325         "description": "Specification versions of the Resource and Device Specifications to which
6326 this device data model is implemented",
6327         "maxLength": 256,
6328         "readOnly": true,
6329         "type": "string"
6330     },
6331     "n": {
6332         "$ref" :
6333 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6334 schema.json#/definitions/n"
6335     },
6336     "id": {
6337         "$ref" :
6338 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6339 schema.json#/definitions/id"
6340     },
6341     "if": {
6342         "description": "The OCF Interfaces supported by this Resource",
6343         "items": {
6344             "enum": [
6345                 "oic.if.r",
6346                 "oic.if.baseline"
6347             ],
6348             "type": "string",
6349             "maxLength": 64
6350         },
6351         "minItems": 2,
6352         "uniqueItems": true,
6353         "readOnly": true,
6354         "type": "array"
6355     }
6356 },
6357 "type": "object",
6358 "required": ["n", "di", "icv", "dmv", "piid"]
6359 }
6360 }

```

6361 }

6362

6363 D.6.5 Property definition

6364 Table D-10 defines the Properties that are part of the "oic.wk.d" Resource Type.

6365 **Table D-10 – The Property definitions of the Resource with type "rt" = "oic.wk.d".**

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	Resource Type of the Resource.
ld	array: see schema	No	Read Only	Localized Descriptions.
piid	multiple types: see schema	Yes	Read Write	
di	multiple types: see schema	Yes	Read Write	
dmno	string	No	Read Only	Model number as designated by manufacturer.
sv	string	No	Read Only	Software version.
dmn	array: see schema	No	Read Only	Manufacturer Name.
icv	string	Yes	Read Only	The version of the Device
dmv	string	Yes	Read Only	Specification versions of the Resource and Device Specifications to which this device data model is implemented.
n	multiple types: see schema	Yes	Read Write	
id	multiple types: see schema	No	Read Write	
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.

6366 D.6.6 CRUDN behaviour

6367 Table D-11 defines the CRUDN operations that are supported on the "oic.wk.d" Resource Type.

6368 **Table D-11 – The CRUDN operations of the Resource with type "rt" = "oic.wk.d".**

Create	Read	Update	Delete	Notify
	get			observe

6369 D.7 Icon

6370 D.7.1 Introduction

6371 This Resource describes the attributes associated with an Icon.

6372

6373 **D.7.2 Example URI**

6374 /IconResURI

6375 **D.7.3 Resource type**

6376 The Resource Type is defined as: "oic.r.icon".

6377 **D.7.4 OpenAPI 2.0 definition**

```
6378 {
6379   "swagger": "2.0",
6380   "info": {
6381     "title": "Icon",
6382     "version": "2019-02-26",
6383     "license": {
6384       "name": "OCF Data Model License",
6385       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
6386       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
6387     },
6388     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
6389   },
6390   "schemes": [
6391     "http"
6392   ],
6393   "consumes": [
6394     "application/json"
6395   ],
6396   "produces": [
6397     "application/json"
6398   ],
6399   "paths": {
6400     "/IconResURI" : {
6401       "get": {
6402         "description": "This Resource describes the attributes associated with an Icon.\n",
6403         "parameters": [
6404           {
6405             "$ref": "#/parameters/interface"
6406           }
6407         ],
6408         "responses": {
6409           "200": {
6410             "description": "",
6411             "x-example": {
6412               "rt": ["oic.r.icon"],
6413               "mimetype": "image/png",
6414               "width": 256,
6415               "height": 256,
6416               "media": "http://findbetter.ru/public/uploads/1481662800/2043.png"
6417             },
6418             "schema": {
6419               "$ref": "#/definitions/Icon"
6420             }
6421           }
6422         }
6423       }
6424     }
6425   },
6426   "parameters": {
6427     "interface" : {
6428       "in" : "query",
6429       "name" : "if",
6430       "type" : "string",
6431       "enum" : ["oic.if.r", "oic.if.baseline"]
6432     }
6433   },
6434   "definitions": {
6435     "Icon" : {
6436       "properties": {
6437         "mimetype": {
6438           "description": "The Media Type of the icon",
```

```

6439         "maxLength": 64,
6440         "readOnly": true,
6441         "type": "string"
6442     },
6443     "rt": {
6444         "description": "Resource Type of the Resource",
6445         "items": {
6446             "enum": ["oic.r.icon"],
6447             "type": "string",
6448             "maxLength": 64
6449         },
6450         "minItems": 1,
6451         "uniqueItems": true,
6452         "readOnly": true,
6453         "type": "array"
6454     },
6455     "media": {
6456         "description": "Specifies the URI to the icon",
6457         "format": "uri",
6458         "maxLength": 256,
6459         "readOnly": true,
6460         "type": "string"
6461     },
6462     "n": {
6463         "$ref" :
6464         "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6465         schema.json#/definitions/n"
6466     },
6467     "id": {
6468         "$ref" :
6469         "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6470         schema.json#/definitions/id"
6471     },
6472     "width": {
6473         "description": "The width in pixels",
6474         "minimum": 1,
6475         "readOnly": true,
6476         "type": "integer"
6477     },
6478     "height": {
6479         "description": "The height in pixels",
6480         "minimum": 1,
6481         "readOnly": true,
6482         "type": "integer"
6483     },
6484     "if": {
6485         "description": "The OCF Interfaces supported by this Resource",
6486         "items": {
6487             "enum": [
6488                 "oic.if.r",
6489                 "oic.if.baseline"
6490             ],
6491             "maxLength": 64,
6492             "type": "string"
6493         },
6494         "minItems": 2,
6495         "uniqueItems": true,
6496         "readOnly": true,
6497         "type": "array"
6498     }
6499 },
6500 "type" : "object",
6501 "required": ["mimetype", "width", "height", "media"]
6502 }
6503 }
6504 }
6505

```

6506 D.7.5 Property definition

6507 Table D-12 defines the Properties that are part of the "oic.r.icon" Resource Type.

6508 **Table D-12 – The Property definitions of the Resource with type "rt" = "oic.r.icon".**

Property name	Value type	Mandatory	Access mode	Description
mimetype	string	Yes	Read Only	The Media Type of the icon.
rt	array: see schema	No	Read Only	Resource Type of the Resource.
media	string	Yes	Read Only	Specifies the URI to the icon.
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
width	integer	Yes	Read Only	The width in pixels.
height	integer	Yes	Read Only	The height in pixels.
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.

6509 **D.7.6 CRUDN behaviour**

6510 Table D-13 defines the CRUDN operations that are supported on the "oic.r.icon" Resource Type.

6511 **Table D-13 – The CRUDN operations of the Resource with type "rt" = "oic.r.icon".**

Create	Read	Update	Delete	Notify
	get			observe

6512 **D.8 Introspection Resource**

6513 **D.8.1 Introduction**

6514 This Resource provides the means to get the Introspection Device Data (IDD) specifying all the
 6515 OCF Endpoints of the Device.
 6516 The url hosted by this Resource is either a local or an external url.
 6517

6518 **D.8.2 Well-known URI**

6519 /IntrospectionResURI

6520 **D.8.3 Resource type**

6521 The Resource Type is defined as: "oic.wk.introspection".

6522 **D.8.4 OpenAPI 2.0 definition**

```

6523 {
6524   "swagger": "2.0",
6525   "info": {
6526     "title": "Introspection Resource",
6527     "version": "2019-03-04",
6528     "license": {
6529       "name": "OCF Data Model License",
6530       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
6531       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
6532     },
6533     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
6534   },
6535   "schemes": [

```

```

6536     "http"
6537   ],
6538   "consumes": [
6539     "application/json"
6540   ],
6541   "produces": [
6542     "application/json"
6543   ],
6544   "paths": {
6545     "/IntrospectionResURI": {
6546       "get": {
6547         "description": "This Resource provides the means to get the Introspection Device Data (IDD)
6548 specifying all the OCF Endpoints of the Device.\n\nThe url hosted by this Resource is either a local
6549 or an external url.\n",
6550         "parameters": [
6551           {
6552             "$ref": "#/parameters/interface"
6553           }
6554         ],
6555         "responses": {
6556           "200": {
6557             "description": "",
6558             "x-example": {
6559               "rt": ["oic.wk.introspection"],
6560               "urlInfo": [
6561                 {
6562                   "content-type": "application/cbor",
6563                   "protocol": "coap",
6564                   "url": "coap://[fe80::1]:1234/IntrospectionExampleURI"
6565                 }
6566               ]
6567             },
6568             "schema": {
6569               "$ref": "#/definitions/oic.wk.introspectionInfo"
6570             }
6571           }
6572         }
6573       }
6574     }
6575   ],
6576   "parameters": {
6577     "interface": {
6578       "in": "query",
6579       "name": "if",
6580       "type": "string",
6581       "enum": ["oic.if.r", "oic.if.baseline"]
6582     }
6583   },
6584   "definitions": {
6585     "oic.wk.introspectionInfo": {
6586       "properties": {
6587         "rt": {
6588           "description": "Resource Type of the Resource",
6589           "items": {
6590             "enum": ["oic.wk.introspection"],
6591             "type": "string",
6592             "maxLength": 64
6593           },
6594           "minItems": 1,
6595           "readOnly": true,
6596           "uniqueItems": true,
6597           "type": "array"
6598         },
6599         "n": {
6600           "$ref":
6601 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6602 schema.json#/definitions/n"
6603         },
6604         "urlInfo": {
6605           "description": "Information on the location of the Introspection Device Data (IDD).",
6606           "items": {

```

```

6607     "properties": {
6608         "content-type": {
6609             "default": "application/cbor",
6610             "description": "content-type of the Introspection Device Data",
6611             "enum": [
6612                 "application/json",
6613                 "application/cbor"
6614             ],
6615             "type": "string"
6616         },
6617         "protocol": {
6618             "description": "Identifier for the protocol to be used to obtain the Introspection
6619 Device Data",
6620             "enum": [
6621                 "coap",
6622                 "coaps",
6623                 "http",
6624                 "https",
6625                 "coap+tcp",
6626                 "coaps+tcp"
6627             ],
6628             "type": "string"
6629         },
6630         "url": {
6631             "description": "The URL of the Introspection Device Data.",
6632             "format": "uri",
6633             "type": "string"
6634         },
6635         "version": {
6636             "default": 1,
6637             "description": "The version of the Introspection Device Data that can be
6638 downloaded",
6639             "enum": [
6640                 1
6641             ],
6642             "type": "integer"
6643         }
6644     },
6645     "required": [
6646         "url",
6647         "protocol"
6648     ],
6649     "type": "object"
6650 },
6651 "minItems": 1,
6652 "readOnly": true,
6653 "type": "array"
6654 },
6655 "id": {
6656     "$ref":
6657 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6658 schema.json#/definitions/id"
6659 },
6660 "if": {
6661     "description": "The OCF Interfaces supported by this Resource",
6662     "items": {
6663         "enum": [
6664             "oic.if.r",
6665             "oic.if.baseline"
6666         ],
6667         "type": "string",
6668         "maxLength": 64
6669     },
6670     "minItems": 2,
6671     "readOnly": true,
6672     "uniqueItems": true,
6673     "type": "array"
6674 }
6675 },
6676 "type": "object",
6677 "required": ["urlInfo"]

```

6678 }
 6679 }
 6680 }
 6681 }

6682 **D.8.5 Property definition**

6683 Table D-14 defines the Properties that are part of the "oic.wk.introspection" Resource Type.

6684 **Table D-14 – The Property definitions of the Resource with type "rt" =**
 6685 **"oic.wk.introspection".**

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	Resource Type of the Resource.
n	multiple types: see schema	No	Read Write	
urlInfo	array: see schema	Yes	Read Only	Information on the location of the Introspection Device Data (IDD).
id	multiple types: see schema	No	Read Write	
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.

6686 **D.8.6 CRUDN behaviour**

6687 Table D-15 defines the CRUDN operations that are supported on the "oic.wk.introspection"
 6688 Resource Type.

6689 **Table D-15 – The CRUDN operations of the Resource with type "rt" =**
 6690 **"oic.wk.introspection".**

Create	Read	Update	Delete	Notify
	get			observe

6691 **D.9 Maintenance**

6692 **D.9.1 Introduction**

6693 The Resource through which a Device is maintained and can be used for diagnostic purposes.
 6694 fr (Factory Reset) is a boolean.
 6695 The value 0 means No action (Default), the value 1 means Start Factory Reset
 6696 After factory reset, this value shall be changed back to the default value
 6697 rb (Reboot) is a boolean.
 6698 The value 0 means No action (Default), the value 1 means Start Reboot
 6699 After Reboot, this value shall be changed back to the default value
 6700

6701 **D.9.2 Well-known URI**

6702 /oic/mnt

6703 **D.9.3 Resource type**

6704 The Resource Type is defined as: "oic.wk.mnt".

6705 D.9.4 OpenAPI 2.0 definition

```
6706 {
6707   "swagger": "2.0",
6708   "info": {
6709     "title": "Maintenance",
6710     "version": "2019-03-04",
6711     "license": {
6712       "name": "OCF Data Model License",
6713       "url":
6714         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
6715 CENSE.md",
6716       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
6717     },
6718     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
6719   },
6720   "schemes": ["http"],
6721   "consumes": ["application/json"],
6722   "produces": ["application/json"],
6723   "paths": {
6724     "/oic/mnt" : {
6725       "get": {
6726         "description": "The Resource through which a Device is maintained and can be used for
6727 diagnostic purposes.\nfr (Factory Reset) is a boolean.\n The value 0 means No action (Default), the
6728 value 1 means Start Factory Reset\nAfter factory reset, this value shall be changed back to the
6729 default value\nrb (Reboot) is a boolean.\n The value 0 means No action (Default), the value 1 means
6730 Start Reboot\nAfter Reboot, this value shall be changed back to the default value\n",
6731         "parameters": [
6732           {"$ref": "#/parameters/interface-all"}
6733         ],
6734         "responses": {
6735           "200": {
6736             "description": "",
6737             "x-example": {
6738               "rt": ["oic.wk.mnt"],
6739               "fr": false,
6740               "rb": false,
6741               "err" : 503
6742             },
6743             "schema": { "$ref": "#/definitions/mnt" }
6744           }
6745         }
6746       },
6747       "post": {
6748         "description": "Set the maintenance action(s)\n",
6749         "parameters": [
6750           {"$ref": "#/parameters/interface-rw"},
6751           {
6752             "name": "body",
6753             "in": "body",
6754             "required": true,
6755             "schema": { "$ref": "#/definitions/mnt-update" },
6756             "x-example": {
6757               "fr": false,
6758               "rb": false
6759             }
6760           }
6761         ],
6762         "responses": {
6763           "200": {
6764             "description": "",
6765             "x-example": {
6766               "fr": false,
6767               "rb": false
6768             },
6769             "schema": { "$ref": "#/definitions/mnt" }
6770           }
6771         }
6772       }
6773     }
6774   },
}
```

```

6775 "parameters": {
6776   "interface-all" : {
6777     "in" : "query",
6778     "name" : "if",
6779     "type" : "string",
6780     "enum" : ["oic.if.rw", "oic.if.baseline"]
6781   },
6782   "interface-rw" : {
6783     "in" : "query",
6784     "name" : "if",
6785     "type" : "string",
6786     "enum" : ["oic.if.rw"]
6787   }
6788 },
6789 "definitions": {
6790   "mnt" : {
6791     "properties": {
6792       "rt" : {
6793         "description": "Resource Type of the Resource",
6794         "items": {
6795           "enum": ["oic.wk.mnt"],
6796           "type": "string",
6797           "maxLength": 64
6798         },
6799         "minItems": 1,
6800         "uniqueItems": true,
6801         "readOnly": true,
6802         "type": "array"
6803       },
6804       "fr" : {
6805         "description": "Factory Reset",
6806         "type": "boolean"
6807       },
6808       "err" : {
6809         "description": "Last HTTP occurred error",
6810         "maximum": 599,
6811         "minimum": 399,
6812         "readOnly": true,
6813         "type": "integer"
6814       },
6815       "n" : {
6816         "$ref":
6817 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6818 schema.json#/definitions/n"
6819       },
6820       "rb" : {
6821         "description": "Reboot Action",
6822         "type": "boolean"
6823       },
6824       "id" : {
6825         "$ref":
6826 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6827 schema.json#/definitions/id"
6828       },
6829       "if" : {
6830         "description": "The OCF Interfaces supported by this Resource",
6831         "items": {
6832           "enum": [
6833             "oic.if.rw",
6834             "oic.if.baseline"
6835           ],
6836           "type": "string",
6837           "maxLength": 64
6838         },
6839         "minItems": 1,
6840         "readOnly": true,
6841         "uniqueItems": true,
6842         "type": "array"
6843       }
6844     },
6845     "anyOf" : [

```



```

6846     {
6847         "required": [ "fr" ]
6848     },
6849     {
6850         "required": [ "rb" ]
6851     },
6852     {
6853         "required": [ "err" ]
6854     }
6855 ],
6856 "type" : "object"
6857 },
6858 "mnt-update" : {
6859     "properties": {
6860         "fr" : {
6861             "description": "Factory Reset",
6862             "type": "boolean"
6863         },
6864         "n" : {
6865             "$ref":
6866 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6867 schema.json#/definitions/n"
6868         },
6869         "rb" : {
6870             "description": "Reboot Action",
6871             "type": "boolean"
6872         }
6873     },
6874     "anyOf" : [
6875         {
6876             "required": [
6877                 "fr"
6878             ]
6879         },
6880         {
6881             "required": [
6882                 "rb"
6883             ]
6884         }
6885     ],
6886     "type" : "object"
6887 }
6888 }
6889 }
6890

```

6891 D.9.5 Property definition

6892 Table D-16 defines the Properties that are part of the "oic.wk.mnt" Resource Type.

6893 **Table D-16 – The Property definitions of the Resource with type "rt" = "oic.wk.mnt".**

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	Resource Type of the Resource.
fr	boolean	No	Read Write	Factory Reset.
err	integer	Yes	Read Only	Last HTTP occurred error.
n	multiple types: see schema	No	Read Write	
rb	boolean	No	Read Write	Reboot Action.
id	multiple types: see schema	No	Read Write	

if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.
fr	boolean	No	Read Write	Factory Reset.
n	multiple types: see schema	No	Read Write	
rb	boolean	Yes	Read Write	Reboot Action.

6894 **D.9.6 CRUDN behaviour**

6895 Table D-17 defines the CRUDN operations that are supported on the "oic.wk.mnt" Resource Type.

6896 **Table D-17 – The CRUDN operations of the Resource with type "rt" = "oic.wk.mnt".**

Create	Read	Update	Delete	Notify
	get	post		observe

6897 **D.10 Network Monitoring**

6898 **D.10.1 Introduction**

6899 The Resource through which a Device can monitor network traffic.
6900

6901 **D.10.2 Example URI**

6902 /nmonResURI

6903 **D.10.3 Resource type**

6904 The Resource Type is defined as: "oic.wk.nmon".

6905 **D.10.4 OpenAPI 2.0 definition**

```
6906 {
6907   "swagger": "2.0",
6908   "info": {
6909     "title": "Network Monitoring",
6910     "version": "2019-03-27",
6911     "license": {
6912       "name": "OCF Data Model License",
6913       "url":
6914         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
6915         CENSE.md",
6916       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
6917     },
6918     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
6919   },
6920   "schemes": ["http"],
6921   "consumes": ["application/json"],
6922   "produces": ["application/json"],
6923   "paths": {
6924     "/nmonResURI" : {
6925       "get": {
6926         "description": "The Resource through which a Device can monitor network traffic.\n",
6927         "parameters": [
6928           {"$ref": "#/parameters/interface-all"}
6929         ],
6930         "responses": {
6931           "200": {
6932             "description": "",
6933             "x-example": {
6934               "rt": ["oic.wk.nmon"],
6935               "ianaifType": 71,
6936               "reset": false,
```

```

6937         "col" : false,
6938         "tx" : 10,
6939         "rx" : 15,
6940         "mmstx" : 50,
6941         "amstx" : 35,
6942         "mmsrx" : 35,
6943         "amsrx" : 20
6944     },
6945     "schema": { "$ref": "#/definitions/nmon" }
6946 }
6947 }
6948 },
6949 "post": {
6950     "description": "Start/Stop collecting and reset the networking monitor Resource\n",
6951     "parameters": [
6952         {"$ref": "#/parameters/interface-rw"},
6953         {
6954             "name": "body",
6955             "in": "body",
6956             "required": true,
6957             "schema": { "$ref": "#/definitions/nmon-update" },
6958             "x-example": {
6959                 "col": true,
6960                 "reset": true
6961             }
6962         }
6963     ],
6964     "responses": {
6965         "200": {
6966             "description": "",
6967             "x-example": {
6968                 "rt": ["oic.wk.nmon"],
6969                 "ianaifType": 71,
6970                 "reset": false,
6971                 "col" : true,
6972                 "tx" : 0,
6973                 "rx" : 0,
6974                 "mmstx" : 0,
6975                 "amstx" : 0,
6976                 "mmsrx" : 0,
6977                 "amsrx" : 0
6978             },
6979             "schema": { "$ref": "#/definitions/nmon" }
6980         }
6981     }
6982 }
6983 },
6984 },
6985 "parameters": {
6986     "interface-rw" : {
6987         "in" : "query",
6988         "name" : "if",
6989         "type" : "string",
6990         "enum" : ["oic.if.rw"]
6991     },
6992     "interface-all" : {
6993         "in" : "query",
6994         "name" : "if",
6995         "type" : "string",
6996         "enum" : ["oic.if.rw", "oic.if.baseline"]
6997     }
6998 },
6999 "definitions": {
7000     "nmon" : {
7001         "properties": {
7002             "amstx" : {
7003                 "description": "Average transmitted message size in bytes (tx) in the collection period",
7004                 "readOnly": true,
7005                 "type": "integer"
7006             },
7007             "reset" : {

```

```

7008         "description": "True: reset the collected values",
7009         "readOnly": false,
7010         "type": "boolean"
7011     },
7012     "mmsrx" : {
7013         "description": "Maximum received message size in bytes (rx) in the collection period",
7014         "readOnly": true,
7015         "type": "integer"
7016     },
7017     "mmstx" : {
7018         "description": "Maximum transmitted message size in bytes (tx) in the collection period",
7019         "readOnly": true,
7020         "type": "integer"
7021     },
7022     "tx" : {
7023         "description": "Amount of transmitted kilo bytes from the collection",
7024         "readOnly": true,
7025         "type": "integer"
7026     },
7027     "rt" : {
7028         "description": "Resource Type of the Resource",
7029         "items": {
7030             "enum": [ "oic.wk.nmon" ],
7031             "type": "string",
7032             "maxLength": 64
7033         },
7034         "minItems": 1,
7035         "uniqueItems": true,
7036         "readOnly": true,
7037         "type": "array"
7038     },
7039     "ianaifType" : {
7040         "description": "The type of the network connection, as defined by iana
7041 https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib",
7042         "readOnly": true,
7043         "type": "integer"
7044     },
7045     "rx" : {
7046         "description": "Amount of received kilobytes from the collection",
7047         "readOnly": true,
7048         "type": "integer"
7049     },
7050     "id" : {
7051         "$ref":
7052 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7053 schema.json#/definitions/id"
7054     },
7055     "amsrx" : {
7056         "description": "Average received message size in bytes (rx) in the collection period",
7057         "readOnly": true,
7058         "type": "integer"
7059     },
7060     "n" : {
7061         "$ref":
7062 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7063 schema.json#/definitions/n"
7064     },
7065     "col" : {
7066         "description": "True: Device is collecting values",
7067         "readOnly": false,
7068         "type": "boolean"
7069     },
7070     "if" : {
7071         "description": "The OCF Interfaces supported by this Resource",
7072         "items": {
7073             "enum": [
7074                 "oic.if.rw",
7075                 "oic.if.baseline"
7076             ],
7077             "type": "string",
7078             "maxLength": 64

```

```

7079     },
7080     "minItems": 1,
7081     "readOnly": true,
7082     "uniqueItems": true,
7083     "type": "array"
7084   },
7085 },
7086 "type" : "object",
7087 "required": ["reset", "col", "ianaifType"]
7088 },
7089 "nmon-update" : {
7090   "properties": {
7091     "reset" : {
7092       "description": "True: reset the collected values",
7093       "readOnly": false,
7094       "type": "boolean"
7095     },
7096     "n" : {
7097       "$ref":
7098 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7099 schema.json#/definitions/n"
7100     },
7101     "col" : {
7102       "description": "True: Device is collecting values",
7103       "readOnly": false,
7104       "type": "boolean"
7105     }
7106   },
7107   "type" : "object",
7108   "required": ["reset", "col"]
7109 }
7110 }
7111 }
7112

```

7113 D.10.5 Property definition

7114 Table D-18 defines the Properties that are part of the "oic.wk.nmon" Resource Type.

7115 **Table D-18 – The Property definitions of the Resource with type "rt" = "oic.wk.nmon".**

Property name	Value type	Mandatory	Access mode	Description
amstx	integer	No	Read Only	Average transmitted message size in bytes (tx) in the collection period
reset	boolean	Yes	Read Write	True: reset the collected values
mmsrx	integer	No	Read Only	Maximum received message size in bytes (rx) in the collection period
mmstx	integer	No	Read Only	Maximum transmitted message size in bytes (tx) in the collection period
tx	integer	No	Read Only	Amount of transmitted kilo bytes from the collection
rt	array: see schema	No	Read Only	Resource Type of the Resource
ianaifType	integer	Yes	Read Only	The type of the network connection, as defined by iana https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib
rx	integer	No	Read Only	Amount of received kilobytes from the collection
id	multiple types: see schema	No	Read Write	
amsrx	integer	No	Read Only	Average received message size in bytes (rx) in the collection period

n	multiple types: see schema	No	Read Write	
col	boolean	Yes	Read Write	True: Device is collecting values
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource
reset	boolean	Yes	Read Write	True: reset the collected values
n	multiple types: see schema	No	Read Write	
col	boolean	Yes	Read Write	True: Device is collecting values

7116 **D.10.6 CRUDN behaviour**

7117 Table D-19 defines the CRUDN operations that are supported on the "oic.wk.nmon" Resource Type.

7118 **Table D-19 – The CRUDN operations of the Resource with type "rt" = "oic.wk.nmon".**

Create	Read	Update	Delete	Notify
	get	post		observe

7119 **D.11 Platform**

7120 **D.11.1 Introduction**

7121 Known Resource that is defines the Platform on which an Server is hosted.
7122 Allows for Platform specific information to be discovered.
7123

7124 **D.11.2 Well-known URI**

7125 /oic/p

7126 **D.11.3 Resource type**

7127 The Resource Type is defined as: "oic.wk.p".

7128 **D.11.4 OpenAPI 2.0 definition**

```

7129 {
7130   "swagger": "2.0",
7131   "info": {
7132     "title": "Platform",
7133     "version": "2019-03-04",
7134     "license": {
7135       "name": "OCF Data Model License",
7136       "url":
7137         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
7138         CENSE.md",
7139       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
7140     },
7141     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
7142   },
7143   "schemes": ["http"],
7144   "consumes": ["application/json"],
7145   "produces": ["application/json"],
7146   "paths": {
7147     "/oic/p" : {
7148       "get": {
7149         "description": "Known Resource that is defines the Platform on which an Server is
7150         hosted.\nAllows for Platform specific information to be discovered.\n",
7151         "parameters": [
7152           {"$ref": "#/parameters/interface"}
7153         ],
7154         "responses": {

```

```

7155         "200": {
7156             "description" : "",
7157             "x-example": {
7158                 "pi": "54919CA5-4101-4AE4-595B-353C51AA983C",
7159                 "rt": ["oic.wk.p"],
7160                 "mnmn": "Acme, Inc"
7161             },
7162             "schema": { "$ref": "#/definitions/Platform" }
7163         }
7164     }
7165 }
7166 },
7167 },
7168 "parameters": {
7169     "interface" : {
7170         "in" : "query",
7171         "name" : "if",
7172         "type" : "string",
7173         "enum" : ["oic.if.r", "oic.if.baseline"]
7174     }
7175 },
7176 "definitions": {
7177     "Platform" : {
7178         "properties": {
7179             "rt" : {
7180                 "description": "Resource Type of the Resource",
7181                 "items": {
7182                     "enum": ["oic.wk.p"],
7183                     "type": "string",
7184                     "maxLength": 64
7185                 },
7186                 "minItems": 1,
7187                 "uniqueItems": true,
7188                 "readOnly": true,
7189                 "type": "array"
7190             },
7191             "pi" : {
7192                 "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
7193 9]{12}$",
7194                 "type": "string",
7195                 "description": "Platform Identifier",
7196                 "readOnly": true
7197             },
7198             "mnfv" : {
7199                 "description": "Manufacturer's firmware version",
7200                 "maxLength": 64,
7201                 "readOnly": true,
7202                 "type": "string"
7203             },
7204             "vid" : {
7205                 "description": "Manufacturer's defined information for the Platform. The content is
7206 freeform, with population rules up to the manufacturer",
7207                 "maxLength": 64,
7208                 "readOnly": true,
7209                 "type": "string"
7210             },
7211             "mnmn" : {
7212                 "description": "Manufacturer name",
7213                 "maxLength": 64,
7214                 "readOnly": true,
7215                 "type": "string"
7216             },
7217             "mnmo" : {
7218                 "description": "Model number as designated by the manufacturer",
7219                 "maxLength": 64,
7220                 "readOnly": true,
7221                 "type": "string"
7222             },
7223             "mnhw" : {
7224                 "description": "Platform Hardware Version",
7225                 "maxLength": 64,

```

```

7226         "readOnly": true,
7227         "type": "string"
7228     },
7229     "mnos" : {
7230         "description": "Platform Resident OS Version",
7231         "maxLength": 64,
7232         "readOnly": true,
7233         "type": "string"
7234     },
7235     "mndt" : {
7236         "pattern": "^[0-9]{4}-(1[0-2]|0[1-9])-(3[0-1]|2[0-9]|1[0-9]|0[1-9])$",
7237         "type": "string",
7238         "description": "Manufacturing Date.",
7239         "readOnly": true
7240     },
7241     "id" : {
7242         "$ref":
7243         "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7244         schema.json#/definitions/id"
7245     },
7246     "mnsi" : {
7247         "description": "Manufacturer's Support Information URL",
7248         "format": "uri",
7249         "maxLength": 256,
7250         "readOnly": true,
7251         "type": "string"
7252     },
7253     "mpv" : {
7254         "description": "Platform Version",
7255         "maxLength": 64,
7256         "readOnly": true,
7257         "type": "string"
7258     },
7259     "st" : {
7260         "description": "The date-time format pattern according to IETF RFC 3339.",
7261         "format": "date-time",
7262         "readOnly": true,
7263         "type": "string"
7264     },
7265     "n" : {
7266         "$ref":
7267         "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7268         schema.json#/definitions/n"
7269     },
7270     "mnm" : {
7271         "description": "Manufacturer's URL",
7272         "format": "uri",
7273         "maxLength": 256,
7274         "readOnly": true,
7275         "type": "string"
7276     },
7277     "mnsn" : {
7278         "description": "Serial number as designated by the manufacturer",
7279         "maxLength": 64,
7280         "readOnly": true,
7281         "type": "string"
7282     },
7283     "if" : {
7284         "description": "The OCF Interfaces supported by this Resource",
7285         "items": {
7286             "enum": [
7287                 "oic.if.r",
7288                 "oic.if.baseline"
7289             ],
7290             "type": "string",
7291             "maxLength": 64
7292         },
7293         "minItems": 2,
7294         "readOnly": true,
7295         "uniqueItems": true,
7296         "type": "array"

```



```

7297     }
7298     },
7299     "type" : "object",
7300     "required": ["pi", "mmn"]
7301   }
7302 }
7303 }
7304 }
7305 }

```

7306 **D.11.5 Property definition**

7307 Table D-20 defines the Properties that are part of the "oic.wk.p" Resource Type.

7308 **Table D-20 – The Property definitions of the Resource with type "rt" = "oic.wk.p".**

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	Resource Type of the Resource.
pi	string	Yes	Read Only	Platform Identifier.
mnfv	string	No	Read Only	Manufacturer's firmware version.
vid	string	No	Read Only	Manufacturer's defined information for the Platform. The content is freeform, with population rules up to the manufacturer.
mmn	string	Yes	Read Only	Manufacturer name.
mnmo	string	No	Read Only	Model number as designated by the manufacturer.
mnhw	string	No	Read Only	Platform Hardware Version.
mnos	string	No	Read Only	Platform Resident OS Version.
mndt	string	No	Read Only	Manufacturing Date.
id	multiple types: see schema	No	Read Write	
mnsi	string	No	Read Only	Manufacturer's Support Information URL.
mnpv	string	No	Read Only	Platform Version
st	string	No	Read Only	The date-time format pattern according to IETF RFC 3339.
n	multiple types: see schema	No	Read Write	
mnml	string	No	Read Only	Manufacturer's URL.
mnsel	string	No	Read Only	Serial number as designated by the manufacturer.
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.

7309 **D.11.6 CRUDN behaviour**

7310 Table D-21 defines the CRUDN operations that are supported on the "oic.wk.p" Resource Type.

7311 **Table D-21 – The CRUDN operations of the Resource with type "rt" = "oic.wk.p".**

Create	Read	Update	Delete	Notify
	get			observe

7312 **D.12 Resource directory resource**

7313 **D.12.1 Introduction**

7314 Resource to be exposed by any Device that can act as a Resource Directory.
 7315 1) Provides selector criteria (e.g., integer) with GET request
 7316 2) Publish a Link in /oic/res with POST request
 7317

7318 **D.12.2 Well-known URI**

7319 /oic/rd

7320 **D.12.3 Resource type**

7321 The Resource Type is defined as: "oic.wk.rd".

7322 **D.12.4 OpenAPI 2.0 definition**

```

7323 {
7324   "swagger": "2.0",
7325   "info": {
7326     "title": "Resource directory resource",
7327     "version": "2019-02-22",
7328     "license": {
7329       "name": "OCF Data Model License",
7330       "url":
7331         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
7332         CENSE.md",
7333       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
7334     },
7335     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
7336   },
7337   "schemes": ["http"],
7338   "consumes": ["application/json"],
7339   "produces": ["application/json"],
7340   "paths": {
7341     "/oic/rd" : {
7342       "get": {
7343         "description": "Resource to be exposed by any Device that can act as a Resource
7344         Directory.\n1) Provides selector criteria (e.g., integer) with GET request\n2) Publish a Link in
7345         /oic/res with POST request\n",
7346         "parameters": [
7347           {"$ref": "#/parameters/rdgetinterface"}
7348         ],
7349         "responses": {
7350           "200": {
7351             "description": "Respond with the selector criteria - either the set of attributes or
7352             the bias factor\n",
7353             "x-example": {
7354               "rt": ["oic.wk.rd"],
7355               "if": ["oic.if.baseline"],
7356               "sel": 50
7357             },
7358             "schema": { "$ref": "#/definitions/rdSelection" }
7359           }
7360         }
7361       }
7362     }
  
```

```

7362     "post": {
7363         "description": "Publish the Resource information for the first time in /oic/res. Updates to
7364 existing entries are not allowed.\nAppropriate parts of the information, i.e., Links of the
7365 published Resources will be discovered through /oic/res.\n1) When a Device first publishes a Link,
7366 the request payload to RD may include the Links without an \"ins\" Parameter.\n2) Upon granting the
7367 request, the RD assigns a unique instance value identifying the Link among all the Links it
7368 advertises\n and sends back the instance value in the \"ins\" Parameter in the Link to the
7369 publishing Device.\n",
7370         "parameters": [
7371             { "$ref": "#/parameters/rdpostinterface",
7372               {
7373                 "name": "body",
7374                 "in": "body",
7375                 "required": true,
7376                 "schema": { "$ref": "#/definitions/rdPublish" },
7377                 "x-example": {
7378                     "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
7379                     "links": [
7380                         {
7381                             "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
7382                             "href": "/myLightSwitch",
7383                             "rt": [ "oic.r.switch.binary" ],
7384                             "if": [ "oic.if.a", "oic.if.baseline" ],
7385                             "p": { "bm": 3 },
7386                             "eps": [
7387                                 { "ep": "coaps://[2001:db8:a::b1d6]:1111", "pri": 2 },
7388                                 { "ep": "coaps://[2001:db8:a::b1d6]:1122" },
7389                                 { "ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3 }
7390                             ]
7391                         },
7392                         {
7393                             "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
7394                             "href": "/myLightBrightness",
7395                             "rt": [ "oic.r.brightness" ],
7396                             "if": [ "oic.if.a", "oic.if.baseline" ],
7397                             "p": { "bm": 3 },
7398                             "eps": [
7399                                 { "ep": "coaps://[2001:db8:a::123]:2222" }
7400                             ]
7401                         }
7402                     ],
7403                     "ttl": 600
7404                 }
7405             }
7406         ],
7407         "responses": {
7408             "200": {
7409                 "description": "Respond with the same schema as publish with the additional \"ins\"
7410 Parameter in the Link.\n",
7411                 "x-example": {
7412                     "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
7413                     "links": [
7414                         {
7415                             "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
7416                             "href": "/myLightSwitch",
7417                             "rt": [ "oic.r.switch.binary" ],
7418                             "if": [ "oic.if.a", "oic.if.baseline" ],
7419                             "p": { "bm": 3 },
7420                             "eps": [
7421                                 { "ep": "coaps://[2001:db8:a::b1d6]:1111", "pri": 2 },
7422                                 { "ep": "coaps://[2001:db8:a::b1d6]:1122" },
7423                                 { "ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3 }
7424                             ],
7425                             "ins": 11235
7426                         },
7427                         {
7428                             "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
7429                             "href": "/myLightBrightness",
7430                             "rt": [ "oic.r.brightness" ],
7431                             "if": [ "oic.if.a", "oic.if.baseline" ],
7432                             "p": { "bm": 3 },

```

```

7433         "eps": [
7434             { "ep": "coaps://[2001:db8:a::123]:2222" }
7435         ],
7436         "ins": 112358
7437     }
7438 ],
7439     "ttl": 600
7440 },
7441     "schema": { "$ref": "#/definitions/rdPublish" }
7442 }
7443 }
7444 }
7445 }
7446 },
7447 "parameters": {
7448     "rdgetinterface" : {
7449         "in" : "query",
7450         "name" : "if",
7451         "type" : "string",
7452         "enum" : ["oic.if.baseline"]
7453     },
7454     "rdpostinterface" : {
7455         "in" : "query",
7456         "name" : "if",
7457         "type" : "string",
7458         "enum" : ["oic.if.baseline"]
7459     }
7460 },
7461 "definitions": {
7462     "rdSelection" : {
7463         "properties": {
7464             "rt" : {
7465                 "description": "Resource Type of the Resource",
7466                 "items": {
7467                     "enum": ["oic.wk.rd"],
7468                     "type": "string",
7469                     "maxLength": 64
7470                 },
7471                 "minItems": 1,
7472                 "uniqueItems": true,
7473                 "readOnly": true,
7474                 "type": "array"
7475             },
7476             "n" : {
7477                 "$ref":
7478 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7479 schema.json#/definitions/n"
7480             },
7481             "sel" : {
7482                 "description": "A bias factor calculated by the Resource Directory",
7483                 "maximum": 100,
7484                 "minimum": 0,
7485                 "readOnly": true,
7486                 "type": "integer"
7487             },
7488             "id" : {
7489                 "$ref":
7490 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7491 schema.json#/definitions/id"
7492             },
7493             "if" : {
7494                 "description": "The OCF Interfaces supported by this Resource",
7495                 "items": {
7496                     "enum": [
7497                         "oic.if.baseline"
7498                     ],
7499                     "type": "string",
7500                     "maxLength": 64
7501                 },
7502                 "minItems": 1,
7503                 "readOnly": true,

```

```

7504         "uniqueItems": true,
7505         "type": "array"
7506     }
7507 },
7508 "type" : "object",
7509 "required": ["sel"]
7510 },
7511 "rdPublish" : {
7512     "properties": {
7513         "di" : {
7514             "$ref":
7515 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7516 schema.json#/definitions/di"
7517         },
7518         "ttl" : {
7519             "description": "Time to indicate a RD, i.e. how long to keep this published item.",
7520             "type": "integer"
7521         },
7522         "links" : {
7523             "description": "A set of simple or individual OCF Links.",
7524             "items": {
7525                 "properties": {
7526                     "anchor": {
7527                         "$ref":
7528 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7529 schema.json#/definitions/anchor"
7530                     },
7531                     "di": {
7532                         "$ref":
7533 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7534 schema.json#/definitions/di"
7535                     },
7536                     "eps": {
7537                         "$ref":
7538 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7539 schema.json#/definitions/eps"
7540                     },
7541                     "href": {
7542                         "$ref":
7543 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7544 schema.json#/definitions/href"
7545                     },
7546                     "if": {
7547                         "description": "The interface set supported by the published resource",
7548                         "items": {
7549                             "enum": [
7550                                 "oic.if.baseline",
7551                                 "oic.if.ll",
7552                                 "oic.if.b",
7553                                 "oic.if.rw",
7554                                 "oic.if.r",
7555                                 "oic.if.a",
7556                                 "oic.if.s"
7557                             ],
7558                             "type": "string",
7559                             "maxLength": 64
7560                         },
7561                         "minItems": 1,
7562                         "uniqueItems": true,
7563                         "type": "array"
7564                     },
7565                     "ins": {
7566                         "$ref":
7567 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7568 schema.json#/definitions/ins"
7569                     },
7570                     "p": {
7571                         "$ref":
7572 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7573 schema.json#/definitions/p"
7574                     },

```

```

7575         "rel": {
7576             "description": "The relation of the target URI referenced by the Link to the context
7577 URI",
7578             "oneOf": [
7579                 {
7580                     "default": [
7581                         "hosts"
7582                     ],
7583                     "items": {
7584                         "maxLength": 64,
7585                         "type": "string"
7586                     },
7587                     "minItems": 1,
7588                     "type": "array"
7589                 },
7590                 {
7591                     "default": "hosts",
7592                     "maxLength": 64,
7593                     "type": "string"
7594                 }
7595             ]
7596         },
7597         "rt": {
7598             "description": "Resource Type of the published Resource",
7599             "items": {
7600                 "maxLength": 64,
7601                 "type": "string"
7602             },
7603             "minItems": 1,
7604             "maxItems": 1,
7605             "uniqueItems": true,
7606             "type": "array"
7607         },
7608         "title": {
7609             "$ref":
7610 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7611 schema.json#/definitions/title"
7612         },
7613         "type": {
7614             "$ref":
7615 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7616 schema.json#/definitions/type"
7617         }
7618     },
7619     "required": [
7620         "href",
7621         "rt",
7622         "if"
7623     ],
7624     "type": "object"
7625 },
7626 "type": "array"
7627 }
7628 },
7629 "type" : "object",
7630 "required": ["di", "links", "ttl"]
7631 }
7632 }
7633 }
7634

```

7635 D.12.5 Property definition

7636 Table D-22 defines the Properties that are part of the "oic.wk.rd" Resource Type.

7637 **Table D-22 – The Property definitions of the Resource with type "rt" = "oic.wk.rd".**

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	Resource Type of the Resource.
n	multiple types: see schema	No	Read Write	
sel	integer	Yes	Read Only	A bias factor calculated by the Resource Directory.
id	multiple types: see schema	No	Read Write	
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.
di	multiple types: see schema	Yes	Read Write	
ttl	integer	Yes	Read Write	Time to indicate a RD, i.e. how long to keep this published item.
links	array: see schema	Yes	Read Write	A set of simple or individual OCF Links.

7638 **D.12.6 CRUDN behaviour**

7639 Table D-23 defines the CRUDN operations that are supported on the "oic.wk.rd" Resource Type.

7640 **Table D-23 – The CRUDN operations of the Resource with type "rt" = "oic.wk.rd".**

Create	Read	Update	Delete	Notify
	get	post		observe

7641 **D.13 Discoverable Resources**

7642 **D.13.1 Introduction**

7643 Baseline representation of /oic/res; list of discoverable Resources
7644

7645 **D.13.2 Well-known URI**

7646 /oic/res

7647 **D.13.3 Resource type**

7648 The Resource Type is defined as: "oic.wk.res".

7649 **D.13.4 OpenAPI 2.0 definition**

```
7650 {
7651   "swagger": "2.0",
7652   "info": {
7653     "title": "Discoverable Resources",
7654     "version": "2019-03-13",
7655     "license": {
7656       "name": "OCF Data Model License",
7657       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
7658       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
7659     }
7660   }
7661 }
```

```

7660     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
7661   },
7662   "schemes": [
7663     "http"
7664   ],
7665   "consumes": [
7666     "application/json"
7667   ],
7668   "produces": [
7669     "application/json"
7670   ],
7671   "paths": {
7672     "/oic/res?if=oic.if.ll": {
7673       "get": {
7674         "description": "Links list representation of /oic/res; list of discoverable Resources\n",
7675         "parameters": [
7676           {
7677             "$ref": "#/parameters/interface-all"
7678           }
7679         ],
7680         "responses": {
7681           "200": {
7682             "description": "",
7683             "x-example": [
7684               {
7685                 "href": "/humidity",
7686                 "rt": ["oic.r.humidity"],
7687                 "if": ["oic.if.s", "oic.if.baseline"],
7688                 "p": {"bm": 3},
7689                 "eps": [
7690                   {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
7691                   {"ep": "coaps://[fe80::b1d6]:1122"},
7692                   {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
7693                 ]
7694               },
7695               {
7696                 "href": "/temperature",
7697                 "rt": ["oic.r.temperature"],
7698                 "if": ["oic.if.s", "oic.if.baseline"],
7699                 "p": {"bm": 3},
7700                 "eps": [
7701                   {"ep": "coaps://[[2001:db8:a::123]:2222"}
7702                 ]
7703               }
7704             ],
7705             "schema": {
7706               "$ref": "#/definitions/slinklist"
7707             }
7708           }
7709         }
7710       }
7711     },
7712     "/oic/res?if=oic.if.baseline": {
7713       "get": {
7714         "description": "Baseline representation of /oic/res; list of discoverable Resources\n",
7715         "parameters": [
7716           {
7717             "$ref": "#/parameters/interface-all"
7718           }
7719         ],
7720         "responses": {
7721           "200": {
7722             "description": "",
7723             "x-example": [
7724               {
7725                 "rt": ["oic.wk.res"],
7726                 "if": ["oic.if.ll", "oic.if.baseline"],
7727                 "links": [
7728                   {
7729                     "href": "/humidity",
7730                     "rt": ["oic.r.humidity"],

```



```

7731         "if": ["oic.if.s", "oic.if.baseline"],
7732         "p": {"bm": 3},
7733         "eps": [
7734             {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
7735             {"ep": "coaps://[fe80::b1d6]:1122"},
7736             {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
7737         ]
7738     },
7739     {
7740         "href": "/temperature",
7741         "rt": ["oic.r.temperature"],
7742         "if": ["oic.if.s", "oic.if.baseline"],
7743         "p": {"bm": 3},
7744         "eps": [
7745             {"ep": "coaps://[[2001:db8:a::123]:2222"}
7746         ]
7747     }
7748 ]
7749 },
7750 ],
7751 "schema": {
7752     "$ref": "#/definitions/sbaseline"
7753 }
7754 }
7755 }
7756 }
7757 },
7758 },
7759 "parameters": {
7760     "interface-all": {
7761         "in": "query",
7762         "name": "if",
7763         "type": "string",
7764         "enum": ["oic.if.ll", "oic.if.baseline"]
7765     }
7766 },
7767 "definitions": {
7768     "oic.oic-link": {
7769         "type": "object",
7770         "properties": {
7771             "anchor": {
7772                 "$ref":
7773 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7774 schema.json#/definitions/anchor"
7775             },
7776             "di": {
7777                 "$ref":
7778 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7779 schema.json#/definitions/di"
7780             },
7781             "eps": {
7782                 "$ref":
7783 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7784 schema.json#/definitions/eps"
7785             },
7786             "href": {
7787                 "$ref":
7788 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7789 schema.json#/definitions/href"
7790             },
7791             "if": {
7792                 "description": "The OCF Interfaces supported by the Linked Resource",
7793                 "items": {
7794                     "enum": [
7795                         "oic.if.baseline",
7796                         "oic.if.ll",
7797                         "oic.if.b",
7798                         "oic.if.rw",
7799                         "oic.if.r",
7800                         "oic.if.a",
7801                         "oic.if.s"

```

```

7802         ],
7803         "type": "string",
7804         "maxLength": 64
7805     },
7806     "minItems": 1,
7807     "uniqueItems": true,
7808     "type": "array"
7809 },
7810 "ins": {
7811     "$ref":
7812 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7813 schema.json#/definitions/ins"
7814 },
7815 "p": {
7816     "$ref":
7817 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7818 schema.json#/definitions/p"
7819 },
7820 "rel": {
7821     "description": "The relation of the target URI referenced by the Link to the context URI",
7822     "oneOf": [
7823         {
7824             "$ref":
7825 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7826 schema.json#/definitions/rel_array"
7827         },
7828         {
7829             "$ref":
7830 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7831 schema.json#/definitions/rel_string"
7832         }
7833     ]
7834 },
7835 "rt": {
7836     "description": "Resource Type of the Linked Resource",
7837     "items": {
7838         "maxLength": 64,
7839         "type": "string"
7840     },
7841     "minItems": 1,
7842     "uniqueItems": true,
7843     "type": "array"
7844 },
7845 "title": {
7846     "$ref":
7847 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7848 schema.json#/definitions/title"
7849 },
7850 "type": {
7851     "$ref":
7852 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7853 schema.json#/definitions/type"
7854 },
7855 },
7856 "required": [
7857     "href",
7858     "rt",
7859     "if"
7860 ]
7861 },
7862 "slinklist": {
7863     "type": "array",
7864     "readOnly": true,
7865     "items": {
7866         "$ref": "#/definitions/oic.oic-link"
7867     }
7868 },
7869 "sbaseline": {
7870     "type": "array",
7871     "minItems": 1,
7872     "maxItems": 1,

```

```

7873     "items": {
7874         "type": "object",
7875         "properties": {
7876             "n": {
7877                 "$ref":
7878                 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7879                 schema.json#/definitions/n"
7880             },
7881             "id": {
7882                 "$ref":
7883                 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7884                 schema.json#/definitions/id"
7885             },
7886             "rt": {
7887                 "description": "Resource Type of this Resource",
7888                 "items": {
7889                     "enum": ["oic.wk.res"],
7890                     "type": "string",
7891                     "maxLength": 64
7892                 },
7893                 "minItems": 1,
7894                 "readOnly": true,
7895                 "uniqueItems": true,
7896                 "type": "array"
7897             },
7898             "if": {
7899                 "description": "The OCF Interfaces supported by this Resource",
7900                 "items": {
7901                     "enum": [
7902                         "oic.if.ll",
7903                         "oic.if.baseline"
7904                     ],
7905                     "type": "string",
7906                     "maxLength": 64
7907                 },
7908                 "minItems": 2,
7909                 "readOnly": true,
7910                 "uniqueItems": true,
7911                 "type": "array"
7912             },
7913             "links": {
7914                 "type": "array",
7915                 "items": {
7916                     "$ref": "#/definitions/oic.oic-link"
7917                 }
7918             }
7919         },
7920     },
7921     "required": [
7922         "rt",
7923         "if",
7924         "links"
7925     ]
7926 }
7927 }
7928 }
7929

```

7930 D.13.5 Property definition

7931 Table D-24 defines the Properties that are part of the "oic.wk.res" Resource Type.

7932 **Table D-24 – The Property definitions of the Resource with type "rt" = "None".**

Property name	Value type	Mandatory	Access mode	Description
anchor	multiple types: see schema	No	Read Write	
di	multiple types: see schema	No	Read Write	

eps	multiple types: see schema	No	Read Write	
href	multiple types: see schema	Yes	Read Write	
if	array: see schema	Yes	Read Write	The OCF Interfaces supported by the Linked Resource.
ins	multiple types: see schema	No	Read Write	
p	multiple types: see schema	No	Read Write	
rel	multiple types: see schema	No	Read Write	The relation of the target URI referenced by the Link to the context URI.
rt	array: see schema	Yes	Read Write	Resource Type of the Linked Resource.
title	multiple types: see schema	No	Read Write	
type	multiple types: see schema	No	Read Write	
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
rt	array: see schema	Yes	Read Only	Resource Type of this Resource.
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by this Resource.
links	array: see schema	Yes	Read Write	

7933 **D.13.6 CRUDN behaviour**

7934 Table D-25 defines the CRUDN operations that are supported on the "None" Resource Type.

7935 **Table D-25 – The CRUDN operations of the Resource with type "rt" = "None".**

Create	Read	Update	Delete	Notify
	get			observe

7936 **D.14 Scene List**

7937 **D.14.1 Introduction**

7938 Toplevel Scene Resource.
7939 This Resource is a generic Collection Resource.
7940 The rts value contains oic.wk.scenecollection Resource Types.
7941

7942 **D.14.2 Example URI**

7943 /SceneListResURI

7944 **D.14.3 Resource type**

7945 The Resource Type is defined as: "oic.wk.scenelist".

7946 **D.14.4 OpenAPI 2.0 definition**

```
7947 {
7948   "swagger": "2.0",
7949   "info": {
7950     "title": "Scene List",
7951     "version": "2019-03-04",
7952     "license": {
7953       "name": "OCF Data Model License",
7954       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
7955       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
7956     },
7957     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
7958   },
7959   "schemes": [
7960     "http"
7961   ],
7962   "consumes": [
7963     "application/json"
7964   ],
7965   "produces": [
7966     "application/json"
7967   ],
7968   "paths": {
7969     "/SceneListResURI?if=oic.if.ll": {
7970       "get": {
7971         "description": "Toplevel Scene Resource.\nThis Resource is a generic Collection
7972 Resource.\nThe rts value contains oic.wk.scenecollection Resource Types.\n",
7973         "parameters": [
7974           {
7975             "$ref": "#/parameters/interface-all"
7976           }
7977         ],
7978         "responses": {
7979           "200": {
7980             "description": "",
7981             "x-example": [
7982               { "href": "/scenecollection1", "rt": ["oic.wk.scenecollection"], "if": ["oic.if.ll",
7983 "oic.if.baseline"] },
7984               { "href": "/scenecollection2", "rt": ["oic.wk.scenecollection"], "if": ["oic.if.ll",
7985 "oic.if.baseline"] }
7986             ],
7987             "schema": {
7988               "$ref": "#/definitions/slinks"
7989             }
7990           }
7991         }
7992       }
7993     },
7994     "/SceneListResURI?if=oic.if.baseline": {
7995       "get": {
7996         "description": "Toplevel Scene Resource.\nThis Resource is a generic Collection
7997 Resource.\nThe rts value contains oic.wk.scenecollection Resource Types.\n",
7998         "parameters": [
7999           {
8000             "$ref": "#/parameters/interface-all"
8001           }
8002         ],
8003         "responses": {
8004           "200": {
8005             "description": "",
8006             "x-example": {
8007               "rt": ["oic.wk.scenelist"],
8008               "if": ["oic.if.ll", "oic.if.baseline"],
8009               "n": "list of scene collections",
8010               "rts": ["oic.wk.scenecollection"],
8011               "links": [
```

```

8012         {"href": "/scenecollection1","rt": ["oic.wk.scenecollection"],"if":["oic.if.ll",
8013 "oic.if.baseline"]},
8014         {"href": "/scenecollection2","rt": ["oic.wk.scenecollection"],"if":["oic.if.ll",
8015 "oic.if.baseline"]}
8016     ]
8017 },
8018     "schema": { "$ref": "#/definitions/Collection" }
8019 }
8020 }
8021 }
8022 },
8023 },
8024 "parameters": {
8025     "interface-all" : {
8026         "in" : "query",
8027         "name" : "if",
8028         "type" : "string",
8029         "enum" : ["oic.if.ll", "oic.if.baseline"]
8030     }
8031 },
8032 "definitions": {
8033     "Collection": {
8034         "properties": {
8035             "links": {
8036                 "description": "A set of simple or individual OCF Links.",
8037                 "items": {
8038                     "$ref": "#/definitions/oic.oic-link"
8039                 },
8040                 "type": "array"
8041             },
8042             "n": {
8043                 "$ref" :
8044 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
8045 schema.json#/definitions/n"
8046             },
8047             "id": {
8048                 "$ref" :
8049 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
8050 schema.json#/definitions/id"
8051             },
8052             "if": {
8053                 "type": "array",
8054                 "description": "The OCF Interfaces supported by this Resource",
8055                 "items": {
8056                     "enum": [
8057                         "oic.if.ll",
8058                         "oic.if.baseline"
8059                     ],
8060                     "type": "string",
8061                     "maxLength": 64
8062                 },
8063                 "minItems": 2,
8064                 "uniqueItems": true,
8065                 "readOnly": true
8066             },
8067             "rts": {
8068                 "description": "The list of allowable Resource Types in Links included in the Collection",
8069                 "items": {
8070                     "enum": ["oic.wk.scenecollection"],
8071                     "type": "string",
8072                     "maxLength": 64
8073                 },
8074                 "minItems": 1,
8075                 "uniqueItems": true,
8076                 "readOnly": true,
8077                 "type": "array"
8078             },
8079             "rt": {
8080                 "description": "Resource Type of the Resource",
8081                 "items": {
8082                     "enum": ["oic.wk.scenelist"],

```

```

8083         "type": "string",
8084         "maxLength": 64
8085     },
8086     "minItems": 1,
8087     "readOnly": true,
8088     "uniqueItems": true,
8089     "type": "array"
8090 }
8091 },
8092 "type": "object",
8093 "required": [
8094     "rt",
8095     "if",
8096     "links"
8097 ]
8098 },
8099 "slinks" : {
8100     "type" : "array",
8101     "items" : {
8102         "$ref": "#/definitions/oic.oic-link"
8103     }
8104 },
8105 "oic.oic-link": {
8106     "properties": {
8107         "if": {
8108             "description": "The OCF Interfaces supported by the Linked Resource",
8109             "items": {
8110                 "enum": [
8111                     "oic.if.ll",
8112                     "oic.if.baseline"
8113                 ],
8114                 "type": "string",
8115                 "maxLength": 64
8116             },
8117             "minItems": 1,
8118             "uniqueItems": true,
8119             "readOnly": true,
8120             "type": "array"
8121         },
8122         "rt": {
8123             "description": "The Resource Type of the Linked Resource",
8124             "items": {
8125                 "enum": ["oic.wk.scenecollection"],
8126                 "type": "string",
8127                 "maxLength": 64
8128             },
8129             "minItems": 1,
8130             "uniqueItems": true,
8131             "readOnly": true,
8132             "type": "array"
8133         },
8134         "anchor": {
8135             "$ref":
8136 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8137 schema.json#/definitions/anchor"
8138         },
8139         "di": {
8140             "$ref":
8141 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8142 schema.json#/definitions/di"
8143         },
8144         "eps": {
8145             "$ref":
8146 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8147 schema.json#/definitions/eps"
8148         },
8149         "href": {
8150             "$ref":
8151 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8152 schema.json#/definitions/href"
8153         },

```

```

8154     "ins": {
8155         "$ref":
8156         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8157         schema.json#/definitions/ins"
8158     },
8159     "p": {
8160         "$ref":
8161         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8162         schema.json#/definitions/p"
8163     },
8164     "rel": {
8165         "$ref":
8166         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8167         schema.json#/definitions/rel_array"
8168     },
8169     "title": {
8170         "$ref":
8171         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8172         schema.json#/definitions/title"
8173     },
8174     "type": {
8175         "$ref":
8176         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8177         schema.json#/definitions/type"
8178     }
8179 },
8180 "required": [
8181     "href",
8182     "rt",
8183     "if"
8184 ],
8185 "type": "object"
8186 }
8187 }
8188 }
8189

```

D.14.5 Property definition

Table D-26 defines the Properties that are part of the "oic.wk.scenelist" Resource Type.

Table D-26 – The Property definitions of the Resource with type "rt" = "oic.wk.scenelist".

Property name	Value type	Mandatory	Access mode	Description
links	array: see schema	Yes	Read Write	A set of simple or individual OCF Links.
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by this Resource.
rts	array: see schema	No	Read Only	The list of allowable Resource Types in Links included in the Collection.
rt	array: see schema	Yes	Read Only	Resource Type of the Resource.
if	array: see schema	Yes	Read Only	The OCF Interfaces

				supported by the Linked Resource.
rt	array: see schema	Yes	Read Only	The Resource Type of the Linked Resource.
anchor	multiple types: see schema	No	Read Write	
di	multiple types: see schema	No	Read Write	
eps	multiple types: see schema	No	Read Write	
href	multiple types: see schema	Yes	Read Write	
ins	multiple types: see schema	No	Read Write	
p	multiple types: see schema	No	Read Write	
rel	multiple types: see schema	No	Read Write	
title	multiple types: see schema	No	Read Write	
type	multiple types: see schema	No	Read Write	

8193 **D.14.6 CRUDN behaviour**

8194 Table D-27 defines the CRUDN operations that are supported on the "oic.wk.scenelist" Resource
8195 Type.

8196 **Table D-27 – The CRUDN operations of the Resource with type "rt" = "oic.wk.scenelist".**

Create	Read	Update	Delete	Notify
	get			observe

8197 **D.15 Scene Collection**

8198 **D.15.1 Introduction**

8199 Collection that models a set of Scenes.
8200 This Resource is a generic Collection Resource with additional Properties.
8201 The rts value contains oic.scenemember Resource Types.
8202 The additional Properties are
8203 lastScene, this is the Scene Value last set by any Client
8204 sceneValues, this is the list of available Scenes
8205 lastScene shall be listed in sceneValues.
8206

8207 **D.15.2 Example URI**

8208 /SceneCollectionResURI

8209 **D.15.3 Resource type**

8210 The Resource Type is defined as: "oic.wk.scenecollection".

8211 D.15.4 OpenAPI 2.0 definition

```
8212 {
8213   "swagger": "2.0",
8214   "info": {
8215     "title": "Scene Collection",
8216     "version": "2019-03-04",
8217     "license": {
8218       "name": "OCF Data Model License",
8219       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
8220       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
8221     },
8222     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
8223   },
8224   "schemes": [
8225     "http"
8226   ],
8227   "consumes": [
8228     "application/json"
8229   ],
8230   "produces": [
8231     "application/json"
8232   ],
8233   "paths": {
8234     "/SceneCollectionResURI?if=oic.if.ll" : {
8235       "get": {
8236         "description": "Collection that models a set of Scenes.\nThis Resource is a generic
8237 Collection Resource with additional Properties.\nThe rts value contains oic.scenemember Resource
8238 Types.\nThe additional Properties are\n lastScene, this is the Scene Value last set by any Client\n
8239 sceneValues, this is the list of available Scenes\n lastScene shall be listed in sceneValues.\n",
8240         "parameters": [
8241           {
8242             "$ref": "#/parameters/interface-all"
8243           }
8244         ],
8245         "responses": {
8246           "200": {
8247             "description": "",
8248             "x-example": [
8249               { "href": "/scenemember1", "rt": ["oic.wk.scenemember"], "if": ["oic.if.baseline"] },
8250               { "href": "/scenemember2", "rt": ["oic.wk.scenemember"], "if": ["oic.if.baseline"] }
8251             ],
8252             "schema": {
8253               "$ref": "#/definitions/slinks"
8254             }
8255           }
8256         }
8257       }
8258     },
8259     "/SceneCollectionResURI?if=oic.if.baseline" : {
8260       "get": {
8261         "description": "Collection that models a set of Scenes.\nThis Resource is a generic
8262 Collection Resource with additional Properties.\nThe rts value contains oic.scenemember Resource
8263 Types.\nThe additional Properties are\n lastScene, this is the Scene Value last set by any Client\n
8264 sceneValues, this is the list of available Scenes\n lastScene shall be listed in sceneValues.\n",
8265         "parameters": [
8266           {
8267             "$ref": "#/parameters/interface-all"
8268           }
8269         ],
8270         "responses": {
8271           "200": {
8272             "description": "",
8273             "x-example": {
8274               "lastScene": "off",
8275               "sceneValues": ["off","Reading","TVWatching"],
8276               "rt": ["oic.wk.scenecollection"],
8277               "n": "My Scenes for my living room",
8278               "rts": ["oic.wk.scenemember"],
8279               "links": [
8280                 { "href": "/scenemember1", "rt": ["oic.wk.scenemember"], "if": ["oic.if.baseline"] },
```

```

8281         {"href": "/scenemember2","rt": ["oic.wk.scenemember"],"if":["oic.if.baseline"]}
8282     ]
8283 },
8284     "schema": {
8285         "$ref": "#/definitions/SceneCollection"
8286     }
8287 }
8288 },
8289 },
8290     "post": {
8291         "description": "Provides the action to change the last set Scene selection.\nCalling this
8292 method shall update all Scene Members to the prescribed membervalue.\nWhen this method is called
8293 with the same value as the current lastScene value\nthen all Scene Members shall be updated.\n",
8294         "parameters": [
8295             {
8296                 "$ref": "#/parameters/interface-update"
8297             },
8298             {
8299                 "name": "body",
8300                 "in": "body",
8301                 "required": true,
8302                 "schema": {
8303                     "$ref": "#/definitions/SceneCollectionUpdate"
8304                 },
8305                 "x-example": {
8306                     "lastScene": "Reading"
8307                 }
8308             }
8309         ],
8310         "responses": {
8311             "200": {
8312                 "description" : "Indicates that the value is changed.\nThe changed Properties are
8313 provided in the response.\n",
8314                 "x-example": {
8315                     "lastScene": "Reading"
8316                 },
8317                 "schema": {
8318                     "$ref": "#/definitions/SceneCollectionUpdate"
8319                 }
8320             }
8321         }
8322     }
8323 },
8324 },
8325     "parameters": {
8326         "interface-update" : {
8327             "in" : "query",
8328             "name" : "if",
8329             "type" : "string",
8330             "enum" : ["oic.if.a"]
8331         },
8332         "interface-all" : {
8333             "in" : "query",
8334             "name" : "if",
8335             "type" : "string",
8336             "enum" : ["oic.if.ll", "oic.if.baseline"]
8337         }
8338     },
8339     "definitions": {
8340         "SceneCollection": {
8341             "properties": {
8342                 "rt": {
8343                     "description": "Resource Type of the Resource",
8344                     "items": {
8345                         "enum": ["oic.wk.scenecollection"],
8346                         "type": "string",
8347                         "maxLength": 64
8348                     },
8349                     "minItems": 1,
8350                     "readOnly": true,
8351                     "uniqueItems": true,

```

```

8352     "type": "array"
8353   },
8354   "lastScene": {
8355     "description": "Last selected Scene from the set of sceneValues",
8356     "type": "string"
8357   },
8358   "links": {
8359     "description": "A set of simple or individual OCF Links.",
8360     "items": {
8361       "$ref": "#/definitions/oic.oic-link"
8362     },
8363     "type": "array"
8364   },
8365   "sceneValues": {
8366     "description": "All available Scene Values",
8367     "items": {
8368       "type": "string"
8369     },
8370     "readOnly": true,
8371     "type": "array"
8372   },
8373   "n": {
8374     "$ref" :
8375     "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
8376     schema.json#/definitions/n"
8377   },
8378   "id": {
8379     "$ref" :
8380     "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
8381     schema.json#/definitions/id"
8382   },
8383   "rts": {
8384     "description": "Resource Type of the Resources within the Collection",
8385     "items": {
8386       "enum": ["oic.wk.scenemember"],
8387       "type": "string",
8388       "maxLength": 64
8389     },
8390     "minItems": 1,
8391     "readOnly": true,
8392     "uniqueItems": true,
8393     "type": "array"
8394   },
8395   "if" : {
8396     "description": "The OCF Interfaces supported by this Resource",
8397     "items": {
8398       "enum": [
8399         "oic.if.ll",
8400         "oic.if.baseline",
8401         "oic.if.a"
8402       ],
8403       "type": "string",
8404       "maxLength": 64
8405     },
8406     "minItems": 1,
8407     "uniqueItems": true,
8408     "readOnly": true,
8409     "type": "array"
8410   }
8411 },
8412 "type" : "object"
8413 },
8414 "SceneCollectionUpdate": {
8415   "properties": {
8416     "lastScene": {
8417       "description": "Last selected Scene from the set of sceneValues",
8418       "type": "string"
8419     }
8420   },
8421   "type" : "object"
8422 },

```

```

8423     "slinks" : {
8424         "type" : "array",
8425         "items" : {
8426             "$ref": "#/definitions/oic.oic-link"
8427         }
8428     },
8429     "oic.oic-link": {
8430         "type": "object",
8431         "properties": {
8432             "if": {
8433                 "description": "The OCF Interfaces supported by the Linked Resource",
8434                 "items": {
8435                     "enum": [
8436                         "oic.if.baseline"
8437                     ],
8438                     "type": "string",
8439                     "maxLength": 64
8440                 },
8441                 "minItems": 1,
8442                 "uniqueItems": true,
8443                 "readOnly": true,
8444                 "type": "array"
8445             },
8446             "rt": {
8447                 "description": "Resource Type of the Linked Resource",
8448                 "items": {
8449                     "enum": [ "oic.wk.scenemember" ],
8450                     "type": "string",
8451                     "maxLength": 64
8452                 },
8453                 "minItems": 1,
8454                 "uniqueItems": true,
8455                 "readOnly": true,
8456                 "type": "array"
8457             },
8458             "anchor": {
8459                 "$ref":
8460 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8461 schema.json#/definitions/anchor"
8462             },
8463             "di": {
8464                 "$ref":
8465 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8466 schema.json#/definitions/di"
8467             },
8468             "eps": {
8469                 "$ref":
8470 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8471 schema.json#/definitions/eps"
8472             },
8473             "href": {
8474                 "$ref":
8475 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8476 schema.json#/definitions/href"
8477             },
8478             "ins": {
8479                 "$ref":
8480 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8481 schema.json#/definitions/ins"
8482             },
8483             "p": {
8484                 "$ref":
8485 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8486 schema.json#/definitions/p"
8487             },
8488             "rel": {
8489                 "$ref":
8490 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8491 schema.json#/definitions/rel_array"
8492             },
8493             "title": {

```

```

8494     "$ref":
8495     "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8496     schema.json#/definitions/title"
8497     },
8498     "type": {
8499     "$ref":
8500     "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8501     schema.json#/definitions/type"
8502     }
8503     },
8504     "required": [
8505     "href",
8506     "rt",
8507     "if"
8508     ]
8509   }
8510 }
8511 }
8512

```

D.15.5 Property definition

Table D-28 defines the Properties that are part of the "oic.wk.scenecollection" Resource Type.

Table D-28 – The Property definitions of the Resource with type "rt" = "oic.wk.scenecollection".

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema		Read Only	Resource Type of the Resource.
lastScene	string		Read Write	Last selected Scene from the set of sceneValues.
links	array: see schema		Read Write	A set of simple or individual OCF Links.
sceneValues	array: see schema		Read Only	All available Scene Values.
n	multiple types: see schema		Read Write	
id	multiple types: see schema		Read Write	
rts	array: see schema		Read Only	Resource Type of the Resources within the Collection.
if	array: see schema		Read Only	The OCF Interfaces supported by this Resource.
lastScene	string		Read Write	Last selected Scene from the set of sceneValues.
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by the Linked Resource.
rt	array: see schema	Yes	Read Only	Resource Type of the Linked Resource.
anchor	multiple types: see schema	No	Read Write	

di	multiple types: see schema	No	Read Write	
eps	multiple types: see schema	No	Read Write	
href	multiple types: see schema	Yes	Read Write	
ins	multiple types: see schema	No	Read Write	
p	multiple types: see schema	No	Read Write	
rel	multiple types: see schema	No	Read Write	
title	multiple types: see schema	No	Read Write	
type	multiple types: see schema	No	Read Write	

8517 **D.15.6 CRUDN behaviour**

8518 Table D-29 defines the CRUDN operations that are supported on the "oic.wk.scenecollection"
8519 Resource Type.

8520 **Table D-29 – The CRUDN operations of the Resource with type "rt" =**
8521 **"oic.wk.scenecollection".**

Create	Read	Update	Delete	Notify
	get	post		observe

8522 **D.16 Scene Member**

8523 **D.16.1 Introduction**

8524 Single Link that models a Scene Member.
8525

8526 **D.16.2 Example URI**

8527 /SceneMemberResURI

8528 **D.16.3 Resource type**

8529 The Resource Type is defined as: "oic.wk.scenemember".

8530 **D.16.4 OpenAPI 2.0 definition**

```
8531 {
8532   "swagger": "2.0",
8533   "info": {
8534     "title": "Scene Member",
8535     "version": "2019-03-04",
8536     "license": {
8537       "name": "OCF Data Model License",
8538       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
8539       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
8540     },
8541     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
8542   },
8543   "schemes": [
8544     "http"
8545   ],
8546   "consumes": [
8547     "application/json"
```

```

8548 ],
8549 "produces": [
8550   "application/json"
8551 ],
8552 "paths": {
8553   "/SceneMemberResURI" : {
8554     "get": {
8555       "description": "Single Link that models a Scene Member.\n",
8556       "parameters": [
8557         {
8558           "$ref": "#/parameters/interface-baseline"
8559         }
8560       ],
8561       "responses": {
8562         "200": {
8563           "description" : "",
8564           "x-example": {
8565             "rt": ["oic.wk.scenemember"],
8566             "id": "0685B960-FFFF-46F7-BEC0-9E6234671ADC1",
8567             "n": "my binary switch (for light bulb) mappings",
8568             "if": ["oic.if.baseline"],
8569             "link": {
8570               "href": "binarySwitch",
8571               "rt": ["oic.r.switch.binary"],
8572               "if": ["oic.if.a", "oic.if.baseline"],
8573               "eps": [
8574                 {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
8575                 {"ep": "coaps://[fe80::b1d6]:1122"},
8576                 {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
8577               ]
8578             },
8579             "SceneMappings": [
8580               {
8581                 "scene": "off",
8582                 "memberProperty": "value",
8583                 "memberValue": "true"
8584               },
8585               {
8586                 "scene": "Reading",
8587                 "memberProperty": "value",
8588                 "memberValue": "false"
8589               },
8590               {
8591                 "scene": "TVWatching",
8592                 "memberProperty": "value",
8593                 "memberValue": "true"
8594               }
8595             ]
8596           },
8597           "schema": {
8598             "$ref": "#/definitions/SceneMember"
8599           }
8600         }
8601       }
8602     }
8603   },
8604   "parameters": {
8605     "interface-baseline" : {
8606       "in" : "query",
8607       "name" : "if",
8608       "type" : "string",
8609       "enum" : ["oic.if.baseline"]
8610     }
8611   },
8612   "definitions": {
8613     "SceneMember": {
8614       "properties": {
8615         "rt": {
8616           "description": "Resource Type of the Resource",
8617           "items": {

```



```

8619         "enum": ["oic.wk.scenemember"],
8620         "type": "string",
8621         "maxLength": 64
8622     },
8623     "minItems": 1,
8624     "readOnly": true,
8625     "uniqueItems": true,
8626     "type": "array"
8627 },
8628 "SceneMappings": {
8629     "description": "Array of mappings per Scene, can be one(1)",
8630     "items": {
8631         "properties": {
8632             "memberProperty": {
8633                 "description": "Property name that will be mapped",
8634                 "readOnly": true,
8635                 "type": "string"
8636             },
8637             "memberValue": {
8638                 "description": "Value of the Member Property",
8639                 "readOnly": true,
8640                 "type": "string"
8641             },
8642             "scene": {
8643                 "description": "Specifies a Scene Value that will be acted upon",
8644                 "type": "string"
8645             }
8646         },
8647         "required": [
8648             "scene",
8649             "memberProperty",
8650             "memberValue"
8651         ],
8652         "type": "object"
8653     },
8654     "type": "array"
8655 },
8656 "n": {
8657     "$ref" :
8658     "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
8659     schema.json#/definitions/n"
8660 },
8661 "id": {
8662     "$ref" :
8663     "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
8664     schema.json#/definitions/id"
8665 },
8666 "link": {
8667     "$ref": "#/definitions/oic.oic-link"
8668 },
8669 "if": {
8670     "description": "The OCF Interfaces supported by this Resource",
8671     "items": {
8672         "enum": [
8673             "oic.if.baseline"
8674         ],
8675         "type": "string",
8676         "maxLength": 64
8677     },
8678     "minItems": 1,
8679     "readOnly": true,
8680     "uniqueItems": true,
8681     "type": "array"
8682 }
8683 },
8684 "type" : "object",
8685 "required": [
8686     "rt",
8687     "if",
8688     "SceneMappings"
8689 ]

```

```

8690     },
8691     "oic.oic-link": {
8692       "properties": {
8693         "if": {
8694           "description": "The OCF Interfaces supported by the target Resource",
8695           "items": {
8696             "enum": [
8697               "oic.if.baseline",
8698               "oic.if.ll",
8699               "oic.if.b",
8700               "oic.if.lb",
8701               "oic.if.rw",
8702               "oic.if.r",
8703               "oic.if.a",
8704               "oic.if.s"
8705             ],
8706             "type": "string",
8707             "maxLength": 64
8708           },
8709           "minItems": 1,
8710           "uniqueItems": true,
8711           "readOnly": true,
8712           "type": "array"
8713         },
8714         "rt": {
8715           "description": "Resource Type of the target Resource",
8716           "items": {
8717             "type": "string",
8718             "maxLength": 64
8719           },
8720           "minItems": 1,
8721           "readOnly": true,
8722           "uniqueItems": true,
8723           "type": "array"
8724         },
8725         "anchor": {
8726           "$ref":
8727             "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8728             schema.json#/definitions/anchor"
8729         },
8730         "di": {
8731           "$ref":
8732             "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8733             schema.json#/definitions/di"
8734         },
8735         "eps": {
8736           "$ref":
8737             "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8738             schema.json#/definitions/eps"
8739         },
8740         "href": {
8741           "$ref":
8742             "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8743             schema.json#/definitions/href"
8744         },
8745         "ins": {
8746           "$ref":
8747             "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8748             schema.json#/definitions/ins"
8749         },
8750         "p": {
8751           "$ref":
8752             "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8753             schema.json#/definitions/p"
8754         },
8755         "rel": {
8756           "$ref":
8757             "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8758             schema.json#/definitions/rel_array"
8759         },
8760         "title": {

```

```

8761         "$ref":
8762         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8763         schema.json#/definitions/title"
8764     },
8765     "type": {
8766         "$ref":
8767         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8768         schema.json#/definitions/type"
8769     }
8770 },
8771 "required": [
8772     "href",
8773     "rt",
8774     "if"
8775 ],
8776 "type": "object"
8777 }
8778 }
8779 }
8780

```

8781 D.16.5 Property definition

8782 Table D-30 defines the Properties that are part of the "oic.wk.scenemember" Resource Type.

8783 **Table D-30 – The Property definitions of the Resource with type "rt" =**
8784 **"oic.wk.scenemember".**

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	Yes	Read Only	Resource Type of the Resource.
SceneMappings	array: see schema	Yes	Read Write	Array of mappings per Scene, can be one(1).
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
link	multiple types: see schema	No	Read Write	
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by this Resource.
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by the target Resource.
rt	array: see schema	Yes	Read Only	Resource Type of the target Resource.
anchor	multiple types: see schema	No	Read Write	
di	multiple types: see schema	No	Read Write	
eps	multiple types: see schema	No	Read Write	
href	multiple types: see schema	Yes	Read Write	
ins	multiple types: see schema	No	Read Write	

p	multiple types: see schema	No	Read Write	
rel	multiple types: see schema	No	Read Write	
title	multiple types: see schema	No	Read Write	
type	multiple types: see schema	No	Read Write	

8785 **D.16.6 CRUDN behaviour**

8786 Table D-31 defines the CRUDN operations that are supported on the "oic.wk.scenemember"
8787 Resource Type.

8788 **Table D-31 – The CRUDN operations of the Resource with "rt" =**
8789 **"oic.wk.scenemember".**

Create	Read	Update	Delete	Notify
	get			observe

8790 **D.17 Alert**

8791 **D.17.1 Introduction**

8792 This Resource provides a mechanism for a Server to expose information to an
8793 interested party with regard to error or other conditions that the Device is experiencing (Alerts).
8794 category is a string that contains the Device defined category for the Alert.
8795 timestamp is an RFC3339 formatted time at which the Alert was generated.
8796 originatorid is a string that contains the identity of the originator of the Alert.
8797 severity is an integer that contains the RFC5424 defined severity of the Alert.
8798 subject is an array containing human readable text in one or more languages.
8799 accountid is a string containing the identity of the account with which the Device is associated.
8800

8801 **D.17.2 Example URI**

8802 /AlertResURI

8803 **D.17.3 Resource type**

8804 The Resource Type is defined as: "oic.r.alert".

8805 **D.17.4 OpenAPI 2.0 definition**

```
8806 {
8807   "swagger": "2.0",
8808   "info": {
8809     "title": "Alert",
8810     "version": "2019-02-28",
8811     "license": {
8812       "name": "OCF Data Model License",
8813       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
8814       "x-copyright": "Copyright 2019 Open Connectivity Foundation, Inc. All rights reserved."
8815     },
8816     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
8817   },
8818   "schemes": ["http"],
8819   "consumes": ["application/json"],
8820   "produces": ["application/json"],
8821   "paths": {
8822     "/AlertResURI" : {
8823       "get": {
8824         "description": "This Resource provides a mechanism for a Server to expose information to
```

```

8825 an\ninterested party with regard to error or other conditions that the Device is experiencing
8826 (Alerts).\ncategory is a string that contains the Device defined category for the Alert.\ntimestamp
8827 is an RFC3339 formatted time at which the Alert was generated.\noriginatorid is a string that
8828 contains the identity of the originator of the Alert.\nseverity is an integer that contains the
8829 RFC5424 defined severity of the Alert.\nsubject is an array containing human readable text in one or
8830 more languages.\naccountid is a string containing the identity of the account with which the Device
8831 is associated.\n",
8832     "parameters": [
8833         {"$ref": "#/parameters/interface"}
8834     ],
8835     "responses": {
8836         "200": {
8837             "description": "",
8838             "x-example":
8839                 {
8840                     "rt": ["oic.r.alert"],
8841                     "accountid": "MyAccountID",
8842                     "category": "MyCategory",
8843                     "timestamp": "2018-02-28T08:00:00Z",
8844                     "originatorid": "MyOriginatorID",
8845                     "severity": 3,
8846                     "subject": [{"language": "en-US", "value": "System error"}]
8847                 },
8848             "schema": {"$ref": "#/definitions/Alert"}
8849         }
8850     }
8851 }
8852 }
8853 },
8854 "parameters": {
8855     "interface": {
8856         "in": "query",
8857         "name": "if",
8858         "type": "string",
8859         "enum": ["oic.if.r", "oic.if.baseline"]
8860     }
8861 },
8862 "definitions": {
8863     "Alert": {
8864         "properties": {
8865             "category": {
8866                 "description": "Category into which the notification is classified",
8867                 "maxLength": 64,
8868                 "readOnly": true,
8869                 "type": "string"
8870             },
8871             "rt": {
8872                 "description": "Resource Type",
8873                 "items": {
8874                     "maxLength": 64,
8875                     "type": "string",
8876                     "enum": ["oic.r.alert"]
8877                 },
8878                 "minItems": 1,
8879                 "readOnly": true,
8880                 "uniqueItems": true,
8881                 "type": "array"
8882             },
8883             "severity": {
8884                 "description": "RFC 5424 severity of the alert",
8885                 "maximum": 7,
8886                 "minimum": 0,
8887                 "readOnly": true,
8888                 "type": "integer"
8889             },
8890             "timestamp": {
8891                 "description": "An RFC3339 formatted time indicating when the data was observed (e.g.:
8892 2016-02-15T09:19Z, 1996-12-19T16:39:57-08:00)",
8893                 "format": "date-time",
8894                 "readOnly": true,
8895                 "type": "string"

```

```

8896     },
8897     "subject": {
8898       "description": "Alert subject matter.",
8899       "items": {
8900         "properties": {
8901           "language": {
8902             "allOf": [
8903               {
8904                 "description": "An identifier formatted according to IETF RFC 5646 (language
8905 tag).",
8906                 "pattern": "^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$",
8907                 "type": "string"
8908               },
8909               {
8910                 "description": "An RFC 5646 language tag.",
8911                 "readOnly": true
8912               }
8913             ]
8914           },
8915           "value": {
8916             "description": "Alert subject matter in the indicated language.",
8917             "maxLength": 255,
8918             "readOnly": true,
8919             "type": "string"
8920           }
8921         },
8922         "type": "object"
8923       },
8924       "minItems": 1,
8925       "readOnly": true,
8926       "type": "array"
8927     },
8928     "originatorid": {
8929       "description": "ID of the creator of the event",
8930       "maxLength": 64,
8931       "readOnly": true,
8932       "type": "string"
8933     },
8934     "n": {
8935       "$ref" :
8936 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
8937 schema.json#/definitions/n"
8938     },
8939     "id": {
8940       "$ref" :
8941 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
8942 schema.json#/definitions/id"
8943     },
8944     "accountid": {
8945       "description": "ID of the account",
8946       "maxLength": 64,
8947       "readOnly": true,
8948       "type": "string"
8949     },
8950     "if": {
8951       "description": "The OCF Interfaces supported by this Resource",
8952       "items": {
8953         "enum": [
8954           "oic.if.r",
8955           "oic.if.baseline"
8956         ],
8957         "type": "string",
8958         "maxLength": 64
8959       },
8960       "minItems": 2,
8961       "readOnly": true,
8962       "uniqueItems": true,
8963       "type": "array"
8964     }
8965   },
8966   "type" : "object",

```

```

8967     "required": ["category", "timestamp", "originatorid", "severity"]
8968     }
8969   }
8970 }
8971

```

8972 **D.17.5 Property definition**

8973 Table D-32 defines the Properties that are part of the "oic.r.alert" Resource Type.

8974 **Table D-32 – The Property definitions of the Resource with type "rt" = "oic.r.alert".**

Property name	Value type	Mandatory	Access mode	Description
category	string	Yes	Read Only	Category into which the notification is classified.
rt	array: see schema	No	Read Only	Resource Type.
severity	integer	Yes	Read Only	RFC 5424 severity of the alert.
timestamp	string	Yes	Read Only	An RFC3339 formatted time indicating when the data was observed (e.g.: 2016-02-15T09:19Z, 1996-12-19T16:39:57-08:00).
subject	array: see schema	No	Read Only	Alert subject matter.
originatorid	string	Yes	Read Only	ID of the creator of the event.
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
accountid	string	No	Read Only	ID of the account.
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.

8975 **D.17.6 CRUDN behaviour**

8976 Table D-33 defines the CRUDN operations that are supported on the "oic.r.alert" Resource Type.

8977 **Table D-33 – The CRUDN operations of the Resource with type "rt" = "oic.r.alert".**

Create	Read	Update	Delete	Notify
	get			observe

8978 D.18 Alert Collection

8979 D.18.1 Introduction

8980 This Resource is a Collection containing instances of Alerts (oic.r.alert).
8981 This is the response using the baseline interface.
8982

8983 D.18.2 Example URI

8984 /AlertCollectionResURI

8985 D.18.3 Resource type

8986 The Resource Type is defined as: "oic.r.alertcollection".

8987 D.18.4 OpenAPI 2.0 definition

```
8988 {
8989   "swagger": "2.0",
8990   "info": {
8991     "title": "Alert Collection",
8992     "version": "2019-03-04",
8993     "license": {
8994       "name": "OCF Data Model License",
8995       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
8996       "x-copyright": "Copyright 2019 Open Connectivity Foundation, Inc. All rights reserved."
8997     },
8998     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
8999   },
9000   "schemes": ["http"],
9001   "consumes": ["application/json"],
9002   "produces": ["application/json"],
9003   "paths": {
9004     "/AlertCollectionResURI?if=oic.if.ll" : {
9005       "get": {
9006         "description": "This Resource is a Collection containing instances of Alerts
9007 (oic.r.alert).\nThis is the response using the links list OCF Interface.\n",
9008         "parameters": [
9009           {"$ref": "#/parameters/interface-all"}
9010         ],
9011         "responses": {
9012           "200": {
9013             "description": "",
9014             "x-example": [
9015               {"href": "/myAlert1ResURI", "rt": ["oic.r.alert"], "if":
9016 ["oic.if.r","oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
9017               {"href": "/myAlert2ResURI", "rt": ["oic.r.alert"], "if":
9018 ["oic.if.r","oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
9019               {"href": "/myAlert3ResURI", "rt": ["oic.r.alert"], "if":
9020 ["oic.if.r","oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
9021               {"href": "/myAlert4ResURI", "rt": ["oic.r.alert"], "if":
9022 ["oic.if.r","oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]}
9023             ],
9024             "schema": {"$ref": "#/definitions/AlertCollection-ll" }
9025           }
9026         }
9027       }
9028     },
9029     "/AlertCollectionResURI?if=oic.if.b" : {
9030       "get": {
9031         "description": "This Resource is a Collection containing instances of Alerts
9032 (oic.r.alert).\nThis is the response using the Batch interface.\n",
9033         "parameters": [
9034           {"$ref": "#/parameters/interface-all"}
9035         ],
9036         "responses": {
9037           "200": {
9038             "description": "",
9039             "x-example": [
```



```

9040     {
9041         "href": "/Alert1ResURI",
9042         "rep":{
9043             "rt":          ["oic.r.alert"],
9044             "accountid":   "MyAccountID",
9045             "category":    "MyCategory",
9046             "timestamp":   "2018-02-28T08:00:00Z",
9047             "originatorid": "MyOriginatorID",
9048             "severity": 3,
9049             "subject": [{"language":"en-US", "value":"System error"}]
9050         }
9051     },
9052     {
9053         "href": "/Alert2ResURI",
9054         "rep":{
9055             "rt":          ["oic.r.alert"],
9056             "accountid":   "MyAccountID",
9057             "category":    "MyCategory",
9058             "timestamp":   "2018-02-28T08:15:00Z",
9059             "originatorid": "MyOriginatorID",
9060             "severity": 4,
9061             "subject": [{"language":"en-US", "value":"Network error"}]
9062         }
9063     }
9064 ],
9065 "schema": { "$ref": "#/definitions/AlertCollection-b" }
9066 }
9067 }
9068 }
9069 },
9070 "/AlertCollectionResURI?if=oic.if.baseline" : {
9071     "get": {
9072         "description": "This Resource is a Collection containing instances of Alerts
9073 (oic.r.alert).\nThis is the response using the baseline interface.\n",
9074         "parameters": [
9075             {"$ref": "#/parameters/interface-all"}
9076         ],
9077         "responses": {
9078             "200": {
9079                 "description" : "",
9080                 "x-example": {
9081                     "rt": ["oic.r.alertcollection"],
9082                     "rts": ["oic.r.alert"],
9083                     "if": ["oic.if.ll", "oic.if.b", "oic.if.baseline"],
9084                     "links": [
9085                         {"href": "/myAlert1ResURI", "rt": ["oic.r.alert"], "if":
9086 ["oic.if.r", "oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
9087                         {"href": "/myAlert2ResURI", "rt": ["oic.r.alert"], "if":
9088 ["oic.if.r", "oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
9089                         {"href": "/myAlert3ResURI", "rt": ["oic.r.alert"], "if":
9090 ["oic.if.r", "oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
9091                         {"href": "/myAlert4ResURI", "rt": ["oic.r.alert"], "if":
9092 ["oic.if.r", "oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]}
9093                     ]
9094                 },
9095                 "schema": { "$ref": "#/definitions/AlertCollection-baseline" }
9096             }
9097         }
9098     }
9099 },
9100 },
9101 "parameters": {
9102     "interface-all" : {
9103         "in" : "query",
9104         "name" : "if",
9105         "type" : "string",
9106         "enum" : ["oic.if.ll", "oic.if.b", "oic.if.baseline"]
9107     }
9108 },
9109 "definitions": {
9110     "AlertCollection-b" : {

```

```

9111     "type": "array",
9112     "minItems": 0,
9113     "uniqueItems": true,
9114     "items": {
9115         "type": "object",
9116         "additionalProperties": true,
9117         "properties": {
9118             "href": {
9119                 "$ref":
9120 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9121 schema.json#/definitions/href"
9122             },
9123             "rep": {
9124                 "$ref":
9125 "http://openconnectivityfoundation.github.io/core/swagger2.0/oic.r.alert.swagger.json#/definitions/A
9126 lert"
9127             }
9128         },
9129         "required": [
9130             "href",
9131             "rep"
9132         ]
9133     }
9134 },
9135 "AlertCollection-baseline" : {
9136     "properties": {
9137         "n": {
9138             "$ref" :
9139 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
9140 schema.json#/definitions/n"
9141         },
9142         "id": {
9143             "$ref" :
9144 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
9145 schema.json#/definitions/id"
9146         },
9147         "rt": {
9148             "items": {
9149                 "type": "string",
9150                 "enum": ["oic.r.alertcollection"],
9151                 "maxLength": 64
9152             },
9153             "minItems": 1,
9154             "type": "array",
9155             "uniqueItems": true,
9156             "readOnly": true
9157         },
9158         "rts": {
9159             "items": {
9160                 "type": "string",
9161                 "enum": ["oic.r.alert"],
9162                 "maxLength": 64
9163             },
9164             "minItems": 1,
9165             "type": "array",
9166             "uniqueItems": true,
9167             "readOnly": true
9168         },
9169         "if": {
9170             "description": "The OCF Interfaces supported by this Resource",
9171             "items": {
9172                 "enum": [
9173                     "oic.if.ll",
9174                     "oic.if.b",
9175                     "oic.if.baseline"
9176                 ],
9177                 "type": "string",
9178                 "maxLength": 64
9179             },
9180             "minItems": 3,
9181             "readOnly": true,

```

```

9182         "uniqueItems": true,
9183         "type": "array"
9184     },
9185     "links": {
9186         "description": "A set of simple or individual Links.",
9187         "items": {
9188             "$ref": "#/definitions/oic.oic-link"
9189         },
9190         "type": "array"
9191     }
9192 },
9193 "type" : "object",
9194 "required": ["rt","rts","if","links"]
9195 },
9196 "AlertCollection-ll" : {
9197     "type": "array",
9198     "items": {
9199         "$ref": "#/definitions/oic.oic-link"
9200     }
9201 },
9202 "oic.oic-link": {
9203     "type": "object",
9204     "properties": {
9205         "anchor": {
9206             "$ref":
9207 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9208 schema.json#/definitions/anchor"
9209         },
9210         "di": {
9211             "$ref":
9212 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9213 schema.json#/definitions/di"
9214         },
9215         "eps": {
9216             "$ref":
9217 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9218 schema.json#/definitions/eps"
9219         },
9220         "href": {
9221             "$ref":
9222 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9223 schema.json#/definitions/href"
9224         },
9225         "ins": {
9226             "$ref":
9227 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9228 schema.json#/definitions/ins"
9229         },
9230         "p": {
9231             "$ref":
9232 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9233 schema.json#/definitions/p"
9234         },
9235         "rel": {
9236             "$ref":
9237 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9238 schema.json#/definitions/rel_array"
9239         },
9240         "title": {
9241             "$ref":
9242 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9243 schema.json#/definitions/title"
9244         },
9245         "type": {
9246             "$ref":
9247 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9248 schema.json#/definitions/type"
9249         },
9250         "if": {
9251             "description": "The OCF Interfaces supported by the target Resource",
9252             "items": {

```

```

9253     "enum": [
9254         "oic.if.r",
9255         "oic.if.baseline"
9256     ],
9257     "type": "string",
9258     "maxLength": 64
9259 },
9260     "minItems": 2,
9261     "uniqueItems": true,
9262     "type": "array",
9263     "readOnly": true
9264 },
9265     "rt": {
9266         "description": "Resource Type of the target Resource",
9267         "items": {
9268             "maxLength": 64,
9269             "type": "string",
9270             "enum": ["oic.r.alert"]
9271         },
9272         "minItems": 1,
9273         "type": "array",
9274         "uniqueItems": true,
9275         "readOnly": true
9276     }
9277 },
9278     "required": [
9279         "href",
9280         "rt",
9281         "if"
9282     ]
9283 }
9284 }
9285 }
9286

```

9287 D.18.5 Property definition

9288 Table D-34 defines the Properties that are part of the "oic.r.alertcollection" Resource Type.

9289 **Table D-34 – The Property definitions of the Resource with type "rt" =**
9290 **"oic.r.alertcollection".**

Property name	Value type	Mandatory	Access mode	Description
href	multiple types: see schema	Yes	Read Write	
rep	multiple types: see schema	Yes	Read Write	
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
rt	array: see schema	Yes	Read Only	
rts	array: see schema	Yes	Read Only	
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by this Resource.
links	array: see schema	Yes	Read Write	A set of simple or individual Links.
anchor	multiple types: see schema	No	Read Write	
di	multiple types: see schema	No	Read Write	

eps	multiple types: see schema	No	Read Write	
href	multiple types: see schema	Yes	Read Write	
ins	multiple types: see schema	No	Read Write	
p	multiple types: see schema	No	Read Write	
rel	multiple types: see schema	No	Read Write	
title	multiple types: see schema	No	Read Write	
type	multiple types: see schema	No	Read Write	
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by the target Resource.
rt	array: see schema	Yes	Read Only	Resource Type of the target Resource.

9291 **D.18.6 CRUDN behaviour**

9292 Table D-35 defines the CRUDN operations that are supported on the "oic.r.alertcollection"
9293 Resource Type.

9294 **Table D-35 – The CRUDN operations of the Resource with type "rt" = "oic.r.alertcollection".**

Create	Read	Update	Delete	Notify
	get			observe

9295

Annex E
(informative)

OIC 1.1 Resource Type definitions

E.1 List of Resource Type Definitions

Table E.1 contains the list of OIC 1.1 defined Core Resources that are referenced in this document and so included herein to enable backwards compatibility. These definitions are only to be used when communicating with OIC 1.1 Devices where specifically referenced in this document.

Table E.1 – Alphabetized list of referenced OIC 1.1 Core Resources

Friendly Name (informative)	Resource Type (rt)	Clause
Collections	"oic.wk.col"	E.2
Discoverable Resources	"oic.wk.res"	E.3

E.2 OCF Collection

E.2.1 Introduction

OCF Collection Resource Type contains properties and links.

E.2.2 Wellknown URI

/CollectionResURI

E.2.3 Resource type

The Resource Type is defined as: "oic.wk.col".

E.2.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "OCF Collection",
    "version": "1.0",
    "license": {
      "name": "copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n      1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \AS IS\ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/CollectionResURI?if=oic.if.baseline" : {
      "get": {
```

```

9344         "description": "OCF Collection Resource Type contains properties and links.\nThe
9345 oic.if.baseline interface exposes a representation of\nthe links and the properties of the
9346 collection resource itself\nRetrieve on Baseline Interface\n",
9347         "parameters": [
9348             { "$ref": "#/parameters/interface-baseline" }
9349         ],
9350         "responses": {
9351             "200": {
9352                 "description": "",
9353                 "x-example":
9354                 {
9355                     "rt": ["oic.wk.col"],
9356                     "id": "unique_example_id",
9357                     "rts": [ "oic.r.switch.binary", "oic.r.airflow" ],
9358                     "rts-m": [ "oic.r.switch.binary" ],
9359                     "links": [
9360                         {
9361                             "href": "switch",
9362                             "rt": [ "oic.r.switch.binary" ],
9363                             "if": [ "oic.if.a", "oic.if.baseline" ],
9364                             "eps": [
9365                                 { "ep": "coap://[fe80::b1d6]:1111", "pri": 2 },
9366                                 { "ep": "coaps://[fe80::b1d6]:1122" },
9367                                 { "ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3 }
9368                             ]
9369                         },
9370                         {
9371                             "href": "airFlow",
9372                             "rt": [ "oic.r.airflow" ],
9373                             "if": [ "oic.if.a", "oic.if.baseline" ],
9374                             "eps": [
9375                                 { "ep": "coap://[fe80::b1d6]:1111", "pri": 2 },
9376                                 { "ep": "coaps://[fe80::b1d6]:1122" },
9377                                 { "ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3 }
9378                             ]
9379                         }
9380                     ]
9381                 },
9382                 "schema": { "$ref": "#/definitions/sbaseline" }
9383             }
9384         }
9385     },
9386     "post": {
9387         "description": "Update on Baseline Interface\n",
9388         "parameters": [
9389             { "$ref": "#/parameters/interface-baseline" },
9390             {
9391                 "name": "body",
9392                 "in": "body",
9393                 "required": true,
9394                 "schema": { "$ref": "#/definitions/sbaseline-update" }
9395             }
9396         ],
9397         "responses": {
9398             "200": {
9399                 "description": "",
9400                 "schema": { "$ref": "#/definitions/sbaseline-update" }
9401             }
9402         }
9403     }
9404 },
9405 "/CollectionResURI?if=oic.if.b" : {
9406     "get": {
9407         "description": "OCF Collection Resource Type contains properties and links.\nThe oic.if.b
9408 interface exposes a composite representation of the\nresources pointed to by the links\nRetrieve on
9409 Batch Interface\n",
9410         "parameters": [
9411             { "$ref": "#/parameters/interface-b" }
9412         ],
9413         "responses": {
9414             "200": {

```

```

9415         "description" : "All targets returned OK status (HTTP 200 or CoAP 2.05 Content)",
9416         "x-example":
9417             [
9418                 {
9419                     "href": "switch",
9420                     "rep":
9421                         {
9422                             "value": true
9423                         }
9424                 },
9425                 {
9426                     "href": "airFlow",
9427                     "rep":
9428                         {
9429                             "direction": "floor",
9430                             "speed":      3
9431                         }
9432                 }
9433             ],
9434         "schema": { "$ref": "#/definitions/sbatch-retrieve" }
9435     },
9436     "404": {
9437         "description" : "One or more targets did not return an OK status, return a
9438 representation containing returned properties from the targets that returned OK",
9439         "x-example":
9440             [
9441                 {
9442                     "href": "switch",
9443                     "rep":
9444                         {
9445                             "value": true
9446                         }
9447                 }
9448             ],
9449         "schema": { "$ref": "#/definitions/sbatch-retrieve" }
9450     }
9451 },
9452 },
9453 "post": {
9454     "description": "Update on Batch Interface\n",
9455     "parameters": [
9456         { "$ref": "#/parameters/interface-b" },
9457         {
9458             "name": "body",
9459             "in": "body",
9460             "required": true,
9461             "schema": { "$ref": "#/definitions/sbatch-update" },
9462             "x-example":
9463                 [
9464                     {
9465                         "href": "switch",
9466                         "rep":
9467                             {
9468                                 "value": true
9469                             }
9470                     },
9471                     {
9472                         "href": "airFlow",
9473                         "rep":
9474                             {
9475                                 "direction": "floor",
9476                                 "speed":      3
9477                             }
9478                     }
9479                 ]
9480         }
9481     ],
9482     "responses": {
9483         "200": {
9484             "description" : "all targets returned OK status (HTTP 200 or CoAP 2.04 Changed) return
9485 a representation of the current state of all targets",

```



```

9486     "x-example":
9487     [
9488     {
9489         "href": "switch",
9490         "rep":
9491         {
9492             "value": true
9493         }
9494     },
9495     {
9496         "href": "airFlow",
9497         "rep":
9498         {
9499             "direction": "demist",
9500             "speed": 5
9501         }
9502     }
9503 ],
9504     "schema": { "$ref": "#/definitions/sbatch-retrieve" }
9505 },
9506     "403": {
9507         "description": "one or more targets did not return OK status; return a retrieve
9508 representation of the current state of all targets in the batch",
9509         "x-example":
9510         [
9511         {
9512             "href": "switch",
9513             "rep":
9514             {
9515                 "value": true
9516             }
9517         },
9518         {
9519             "href": "airFlow",
9520             "rep":
9521             {
9522                 "direction": "floor",
9523                 "speed": 3
9524             }
9525         }
9526     ],
9527         "schema": { "$ref": "#/definitions/sbatch-retrieve" }
9528     }
9529 }
9530 },
9531 ],
9532 "/CollectionResURI?if=oic.if.ll" : {
9533     "get": {
9534         "description": "OCF Collection Resource Type contains properties and links.\nThe oic.if.ll
9535 interface exposes a representation of the links\nRetrieve on Link List Interface\n",
9536         "parameters": [
9537             { "$ref": "#/parameters/interface-ll" }
9538         ],
9539         "responses": {
9540             "200": {
9541                 "description": "",
9542                 "x-example":
9543                 {
9544                     "links": [
9545                         {
9546                             "href": "switch",
9547                             "rt": ["oic.r.switch.binary"],
9548                             "if": ["oic.if.a", "oic.if.baseline"],
9549                             "eps": [
9550                                 { "ep": "coap://[fe80:b1d6]:1111", "pri": 2 },
9551                                 { "ep": "coaps://[fe80:b1d6]:1122" },
9552                                 { "ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3 }
9553                             ]
9554                         },
9555                         {
9556                             "href": "airFlow",

```

```

9557         "rt": ["oic.r.airflow"],
9558         "if": ["oic.if.a", "oic.if.baseline"],
9559         "eps": [
9560             {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
9561             {"ep": "coaps://[fe80::b1d6]:1122"},
9562             {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
9563         ]
9564     }
9565 ]
9566 },
9567     "schema": { "$ref": "#/definitions/slinks" }
9568 }
9569 }
9570 }
9571 },
9572 },
9573 "parameters": {
9574     "interface-11" : {
9575         "in" : "query",
9576         "name" : "if",
9577         "type" : "string",
9578         "enum" : ["oic.if.11"]
9579     },
9580     "interface-b" : {
9581         "in" : "query",
9582         "name" : "if",
9583         "type" : "string",
9584         "enum" : ["oic.if.b"]
9585     },
9586     "interface-baseline" : {
9587         "in" : "query",
9588         "name" : "if",
9589         "type" : "string",
9590         "enum" : ["oic.if.baseline"]
9591     },
9592     "interface-all" : {
9593         "in" : "query",
9594         "name" : "if",
9595         "type" : "string",
9596         "enum" : ["oic.if.11", "oic.if.baseline", "oic.if.b"]
9597     }
9598 },
9599 "definitions": {
9600     "sbaseline" : {
9601         "properties": {
9602             "links" : {
9603                 "description": "A set of simple or individual OIC Links.",
9604                 "items": {
9605                     "$ref": "#/definitions/oic.oic-link"
9606                 },
9607                 "type": "array"
9608             },
9609             "n": {
9610                 "type": "string",
9611                 "description": "User friendly name of the collection"
9612             },
9613             "id": {
9614                 "anyOf": [
9615                     {
9616                         "type": "integer",
9617                         "description": "A number that is unique to that collection; like an ordinal number
9618 that is not repeated"
9619                     },
9620                     {
9621                         "type": "string",
9622                         "description": "A unique string that could be a hash or similarly unique"
9623                     },
9624                     {
9625                         "description": "An identifier formatted according to IETF RFC 4122.",
9626                         "type": "string",
9627                         "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-

```

```

9628 9]{12}$"
9629     },
9630   ],
9631   "description": "ID for the collection. Can be an value that is unique to the use context
9632 or a UUIDv4"
9633   },
9634   "rt": { "$ref": "#/definitions/oic.core/properties/rt" },
9635   "rts": { "$ref": "#/definitions/oic.core/properties/rt" },
9636   "if": {
9637     "description": "The interface set supported by this resource",
9638     "items": {
9639       "enum": [ "oic.if.baseline",
9640                "oic.if.ll",
9641                "oic.if.b",
9642                "oic.if.rw",
9643                "oic.if.r",
9644                "oic.if.a",
9645                "oic.if.s" ],
9646       "type": "string"
9647     },
9648     "minItems": 1,
9649     "type": "array"
9650   },
9651   },
9652   "type" : "object"
9653 },
9654 "sbaseline-update": {
9655   "additionalProperties": true
9656 },
9657   "oic.core": {
9658     "properties": {
9659       "rt": {
9660         "description": "Resource Type of the Resource",
9661         "items": {
9662           "maxLength": 64,
9663           "type": "string"
9664         },
9665         "minItems": 1,
9666         "readOnly": true,
9667         "type": "array"
9668       },
9669     },
9670     "type": "object"
9671   },
9672   "sbatch-retrieve" : {
9673     "title" : "Collection Batch Retrieve Format (auto merged)",
9674     "minItems" : 1,
9675     "items" : {
9676       "additionalProperties": true,
9677       "properties": {
9678         "href": {
9679           "description": "URI of the target resource relative assuming the collection URI as
9680 anchor",
9681           "format": "uri",
9682           "maxLength": 256,
9683           "type": "string"
9684         },
9685         "rep": {
9686           "oneOf": [
9687             {
9688               "description": "The response payload from a single resource",
9689               "type": "object"
9690             },
9691             {
9692               "description": " The response payload from a collection (batch) resource",
9693               "items": {
9694                 "properties": {
9695                   "anchor": {
9696                     "description": "This is used to override the context URI e.g. override the URI
9697 of the containing collection.",
9698                     "format": "uri",

```

```

9699         "maxLength": 256,
9700         "type": "string"
9701     },
9702     "di": {
9703         "allOf": [
9704             {
9705                 "description": "Format pattern according to IETF RFC 4122.",
9706                 "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-
9707 [a-fA-F0-9]{12}$",
9708                 "type": "string"
9709             },
9710             {
9711                 "description": "The device ID"
9712             }
9713         ]
9714     },
9715     "eps": {
9716         "description": "the Endpoint information of the target Resource",
9717         "items": {
9718             "properties": {
9719                 "ep": {
9720                     "description": "Transport Protocol Suite + Endpoint Locator",
9721                     "format": "uri",
9722                     "type": "string"
9723                 },
9724                 "pri": {
9725                     "description": "The priority among multiple Endpoints",
9726                     "minimum": 1,
9727                     "type": "integer"
9728                 }
9729             },
9730             "type": "object"
9731         },
9732         "type": "array"
9733     },
9734     "href": {
9735         "description": "This is the target URI, it can be specified as a Relative
9736 Reference or fully-qualified URI.",
9737         "format": "uri",
9738         "maxLength": 256,
9739         "type": "string"
9740     },
9741     "if": {
9742         "description": "The interface set supported by this resource",
9743         "items": {
9744             "enum": [
9745                 "oic.if.baseline",
9746                 "oic.if.ll",
9747                 "oic.if.b",
9748                 "oic.if.rw",
9749                 "oic.if.r",
9750                 "oic.if.a",
9751                 "oic.if.s"
9752             ],
9753             "type": "string"
9754         },
9755         "minItems": 1,
9756         "type": "array"
9757     },
9758     "ins": {
9759         "description": "The instance identifier for this web link in an array of web
9760 links - used in collections",
9761         "type": "integer"
9762     },
9763     "p": {
9764         "description": "Specifies the framework policies on the Resource referenced by
9765 the target URI",
9766         "properties": {
9767             "bm": {
9768                 "description": "Specifies the framework policies on the Resource
9769 referenced by the target URI for e.g. observable and discoverable",

```

```

9770         "type": "integer"
9771     }
9772 },
9773     "required": [
9774         "bm"
9775     ],
9776     "type": "object"
9777 },
9778     "rel": {
9779         "description": "The relation of the target URI referenced by the link to the
context URI",
9780         "oneOf": [
9781             {
9782                 "default": [
9783                     "hosts"
9784                 ],
9785                 "items": {
9786                     "maxLength": 64,
9787                     "type": "string"
9788                 },
9789                 "minItems": 1,
9790                 "type": "array"
9791             },
9792             {
9793                 "default": "hosts",
9794                 "maxLength": 64,
9795                 "type": "string"
9796             }
9797         ]
9798     },
9799 },
9800     "rt": {
9801         "description": "Resource Type of the Resource",
9802         "items": {
9803             "maxLength": 64,
9804             "type": "string"
9805         },
9806         "minItems": 1,
9807         "type": "array"
9808     },
9809     "title": {
9810         "description": "A title for the link relation. Can be used by the UI to
provide a context.",
9811         "maxLength": 64,
9812         "type": "string"
9813     },
9814 },
9815     "type": {
9816         "default": "application/cbor",
9817         "description": "A hint at the representation of the resource referenced by the
target URI. This represents the media types that are used for both accepting and emitting.",
9818         "items": {
9819             "maxLength": 64,
9820             "type": "string"
9821         },
9822         "minItems": 1,
9823         "type": "array"
9824     }
9825 }
9826 },
9827     "required": [
9828         "href",
9829         "rt",
9830         "if"
9831     ],
9832     "type": "object"
9833 },
9834     "type": "array"
9835 }
9836 ]
9837 }
9838 },
9839     "required": [
9840         "href",

```

```

9841         "rep"
9842     ],
9843     "type": "object"
9844 },
9845     "type" : "array"
9846 },
9847     "sbatch-update" : {
9848         "title" : "Collection Batch Update Format (auto merged)",
9849         "minItems" : 1,
9850         "items" : { "$ref": "#/definitions/oic.batch-update.item" },
9851         "type" : "array"
9852     },
9853     "slinks" : {
9854         "type": "object",
9855         "properties": {
9856             "links": {
9857                 "type" : "array",
9858                 "items" : {
9859                     "$ref": "#/definitions/oic.oic-link"
9860                 }
9861             }
9862         }
9863     },
9864     "oic.batch-update.item" : {
9865         "additionalProperties": true,
9866         "description": "array of resource representations to apply to the batch collection, using href
9867 to indicate which resource(s) in the batch to update. If the href property is empty, effectively
9868 making the URI reference to the collection itself, the representation is to be applied to all
9869 resources in the batch",
9870         "properties": {
9871             "href": {
9872                 "description": "URI of the target resource relative assuming the collection URI as
9873 anchor",
9874                 "format": "uri",
9875                 "maxLength": 256,
9876                 "type": "string"
9877             },
9878             "rep": {
9879                 "oneOf": [
9880                     {
9881                         "description": "The response payload from a single resource",
9882                         "type": "object"
9883                     },
9884                     {
9885                         "description": " The response payload from a collection (batch) resource",
9886                         "items": {
9887                             "$ref": "#/definitions/oic.oic-link"
9888                         },
9889                         "type": "array"
9890                     }
9891                 ]
9892             }
9893         },
9894         "required": [
9895             "href",
9896             "rep"
9897         ],
9898         "type": "object"
9899     },
9900     "oic.oic-link" : {
9901         "properties": {
9902             "anchor": {
9903                 "description": "This is used to override the context URI e.g. override the URI of the
9904 containing collection.",
9905                 "format": "uri",
9906                 "maxLength": 256,
9907                 "type": "string"
9908             },
9909             "di": {
9910                 "description": "The Device ID formatted according to IETF RFC 4122.",
9911                 "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-

```

```

9912 9]{12}$",
9913     "type": "string"
9914   },
9915   "eps": {
9916     "description": "the Endpoint information of the target Resource",
9917     "items": {
9918       "properties": {
9919         "ep": {
9920           "description": "Transport Protocol Suite + Endpoint Locator",
9921           "format": "uri",
9922           "type": "string"
9923         },
9924         "pri": {
9925           "description": "The priority among multiple Endpoints",
9926           "minimum": 1,
9927           "type": "integer"
9928         }
9929       },
9930       "type": "object"
9931     },
9932     "type": "array"
9933   },
9934   "href": {
9935     "description": "This is the target URI, it can be specified as a Relative Reference or
9936 fully-qualified URI.",
9937     "format": "uri",
9938     "maxLength": 256,
9939     "type": "string"
9940   },
9941   "if": {
9942     "description": "The interface set supported by this resource",
9943     "items": {
9944       "enum": [
9945         "oic.if.baseline",
9946         "oic.if.ll",
9947         "oic.if.b",
9948         "oic.if.rw",
9949         "oic.if.r",
9950         "oic.if.a",
9951         "oic.if.s"
9952       ],
9953       "type": "string"
9954     },
9955     "minItems": 1,
9956     "type": "array"
9957   },
9958   "ins": {
9959     "description": "The instance identifier for this web link in an array of web links - used
9960 in collections",
9961     "type": "integer"
9962   },
9963   "p": {
9964     "description": "Specifies the framework policies on the Resource referenced by the target
9965 URI",
9966     "properties": {
9967       "bm": {
9968         "description": "Specifies the framework policies on the Resource referenced by the
9969 target URI for e.g. observable and discoverable",
9970         "type": "integer"
9971       }
9972     },
9973     "required": [
9974       "bm"
9975     ],
9976     "type": "object"
9977   },
9978   "rel": {
9979     "description": "The relation of the target URI referenced by the link to the context URI",
9980     "oneOf": [
9981       {
9982         "default": [

```

```

9983         "hosts"
9984     ],
9985     "items": {
9986         "maxLength": 64,
9987         "type": "string"
9988     },
9989     "minItems": 1,
9990     "type": "array"
9991 },
9992 {
9993     "default": "hosts",
9994     "maxLength": 64,
9995     "type": "string"
9996 }
9997 ]
9998 },
9999 "rt": {
10000     "description": "Resource Type of the Resource",
10001     "items": {
10002         "maxLength": 64,
10003         "type": "string"
10004     },
10005     "minItems": 1,
10006     "type": "array"
10007 },
10008 "title": {
10009     "description": "A title for the link relation. Can be used by the UI to provide a
10010 context.",
10011     "maxLength": 64,
10012     "type": "string"
10013 },
10014 "type": {
10015     "default": "application/cbor",
10016     "description": "A hint at the representation of the resource referenced by the target URI.
10017 This represents the media types that are used for both accepting and emitting.",
10018     "items": {
10019         "maxLength": 64,
10020         "type": "string"
10021     },
10022     "minItems": 1,
10023     "type": "array"
10024 }
10025 },
10026 "required": [
10027     "href",
10028     "rt",
10029     "if"
10030 ],
10031 "type": "object"
10032 }
10033 }
10034 }
10035

```

10036 E.2.5 Property definition

10037 Table E.2 defines the Properties that are part of the "oic.wk.col" Resource Type

10038 **Table E.2 – The Property definitions of the Resource with type "rt" = "oic.wk.col"**

Property name	Value type	Mandatory	Access mode	Description
rep	multiple types: see schema	Yes	Read Write	
href	string	Yes	Read Write	URI of the target resource relative assuming the collection URI as anchor.

rt	array: see schema		Read Only	Resource Type of the Resource.
links	array: see schema		Read Write	
if	array: see schema		Read Write	The interface set supported by this resource.
rts	multiple types: see schema		Read Write	
id	multiple types: see schema		Read Write	ID for the collection. Can be an value that is unique to the use context or a UUIDv4.
rt	multiple types: see schema		Read Write	
n	string		Read Write	User friendly name of the collection.
links	array: see schema		Read Write	A set of simple or individual OIC Links.
di	string	No	Read Write	The Device ID formatted according to IETF RFC 4122.
anchor	string	No	Read Write	This is used to override the context URI e.g. override the URI of the containing collection.
if	array: see schema	Yes	Read Write	The interface set supported by this resource.
rel	multiple types: see schema	No	Read Write	The relation of the target URI referenced by the link to the context URI.
eps	array: see schema	No	Read Write	the Endpoint information of the target Resource.
ins	integer	No	Read Write	The instance identifier for this web link in an array of web links - used in collections.
rt	array: see schema	Yes	Read Write	Resource Type of the Resource.
type	array: see schema	No	Read Write	A hint at the representation of the resource referenced by the target URI. This represents the media types that are used for both accepting and emitting.
title	string	No	Read Write	A title for the link relation. Can be used by the UI to provide a context.

href	string	Yes	Read Write	This is the target URI, it can be specified as a Relative Reference or fully-qualified URI.
p	object: see schema	No	Read Write	Specifies the framework policies on the Resource referenced by the target URI.
rep	multiple types: see schema	Yes	Read Write	
href	string	Yes	Read Write	URI of the target resource relative assuming the collection URI as anchor.

10039 **E.2.6 CRUDN behaviour**

10040 Table E.3 defines the CRUDN operations that are supported on the ['oic.wk.col'] Resource Type

10041 **Table E.3 – The CRUDN operations of the Resource with type 'rt' = ['oic.wk.col']**

Create	Read	Update	Delete	Notify
	get	post		observe

10042 **E.3 Discoverable Resources**

10043 **E.3.1 Introduction**

10044 List of discoverable resources.

10045 **E.3.2 Wellknown URI**

10046 /oic/res

10047 **E.3.3 Resource type**

10048 The Resource Type is defined as: "oic.wk.res"

10049 **E.3.4 OpenAPI 2.0 definition**

```

10050 {
10051   "swagger": "2.0",
10052   "info": {
10053     "title": "Discoverable Resources Link List interface",
10054     "version": "v1-20160622",
10055     "license": {
10056       "name": "copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved.",
10057       "x-description": "Redistribution and use in source and binary forms, with or without
10058 modification, are permitted provided that the following conditions are met:\n      1.
10059 Redistributions of source code must retain the above copyright notice, this list of conditions and
10060 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
10061 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
10062 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
10063 Connectivity Foundation, INC. \AS IS\ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
10064 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
10065 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
10066 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
10067 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
10068 SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND ON
10069 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
10070 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

```

```

10071 SUCH DAMAGE.\n"
10072 }
10073 },
10074 "schemes": ["http"],
10075 "consumes": ["application/json"],
10076 "produces": ["application/json"],
10077 "paths": {
10078   "/oic/res?if=oic.if.ll" : {
10079     "get": {
10080       "description": "Link list representation of /oic/res; list of discoverable
10081 resources\nRetrieve the discoverable resource set, link list interface\n",
10082       "parameters": [
10083         {"$ref": "#/parameters/interface-ll"}
10084       ],
10085       "responses": {
10086         "200": {
10087           "description": "",
10088           "x-example":
10089             [
10090               {
10091                 "di": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1",
10092                 "links": [
10093                   {
10094                     "href": "/humidity",
10095                     "rt": ["oic.r.humidity"],
10096                     "if": ["oic.if.s"],
10097                     "p": {"bm": 3},
10098                     "eps": [
10099                       {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
10100                       {"ep": "coaps://[fe80::b1d6]:1122"},
10101                       {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
10102                     ]
10103                   },
10104                   {
10105                     "href": "/temperature",
10106                     "rt": ["oic.r.temperature"],
10107                     "if": ["oic.if.s"],
10108                     "p": {"bm": 3},
10109                     "eps": [
10110                       {"ep": "coaps://[[2001:db8:a::123]:2222"}
10111                     ]
10112                   }
10113                 ]
10114               }
10115             ],
10116           "schema": {"$ref": "#/definitions/slinklist"}
10117         }
10118       }
10119     }
10120   },
10121   "/oic/res?if=oic.if.baseline" : {
10122     "get": {
10123       "description": "Baseline representation of /oic/res; list of discoverable
10124 resources\nRetrieve the discoverable resource set, baseline interface\n",
10125       "parameters": [
10126         {"$ref": "#/parameters/interface-baseline"}
10127       ],
10128       "responses": {
10129         "200": {
10130           "description": "",
10131           "x-example":
10132             [
10133               {
10134                 "rt": ["oic.wk.res"],
10135                 "if": ["oic.if.baseline", "oic.if.ll"],
10136                 "di": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1",
10137                 "links": [
10138                   {
10139                     "href": "/humidity",
10140                     "rt": ["oic.r.humidity"],
10141                     "if": ["oic.if.s"],

```

```

10142         "p": { "bm": 3 },
10143         "eps": [
10144             { "ep": "coaps://[fe80::bld6]:1111", "pri": 2 },
10145             { "ep": "coaps://[fe80::bld6]:1122",
10146               { "ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3 }
10147             ]
10148         },
10149         {
10150             "href": "/temperature",
10151             "rt": [ "oic.r.temperature" ],
10152             "if": [ "oic.if.s" ],
10153             "p": { "bm": 3 },
10154             "eps": [
10155                 { "ep": "coaps://[[2001:db8:a::123]:2222" ]
10156             ]
10157         }
10158     ]
10159 },
10160 ],
10161     "schema": { "$ref": "#/definitions/sbaseline" }
10162 }
10163 }
10164 }
10165 }
10166 },
10167 "parameters": {
10168     "interface-ll" : {
10169         "in" : "query",
10170         "name" : "if",
10171         "type" : "string",
10172         "enum" : [ "oic.if.ll" ]
10173     },
10174     "interface-baseline" : {
10175         "in" : "query",
10176         "name" : "if",
10177         "type" : "string",
10178         "enum" : [ "oic.if.baseline" ]
10179     },
10180     "interface-all" : {
10181         "in" : "query",
10182         "name" : "if",
10183         "type" : "string",
10184         "enum" : [ "oic.if.ll", "oic.if.baseline" ]
10185     }
10186 },
10187 "definitions": {
10188     "slinklist" : {
10189         "type": "array",
10190         "items" : {
10191             "type": "object",
10192             "properties": {
10193                 "di": {
10194                     "description": "An identifier formatted according to IETF RFC 4122.",
10195                     "type": "string",
10196                     "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
10197 9]{12}$",
10198                     "readOnly": true
10199                 },
10200                 "links": {
10201                     "type": "array",
10202                     "items": {
10203                         "$ref": "#/definitions/oic.oic-link"
10204                     }
10205                 }
10206             }
10207         }
10208     },
10209     "sbaseline" : {
10210         "type": "array",
10211         "items" : {
10212             "type": "object",

```

```

10213     "properties": {
10214         "n": {
10215             "description": "Human friendly name",
10216             "maxLength": 64,
10217             "readOnly": true,
10218             "type": "string"
10219         },
10220         "rt": {
10221             "description": "Resource Type of the Resource",
10222             "items": {
10223                 "maxLength": 64,
10224                 "type": "string"
10225             },
10226             "minItems": 1,
10227             "readOnly": true,
10228             "type": "array"
10229         },
10230         "if": {
10231             "description": "The interface set supported by this resource",
10232             "items": {
10233                 "enum": [
10234                     "oic.if.baseline",
10235                     "oic.if.ll"
10236                 ],
10237                 "type": "string"
10238             },
10239             "minItems": 1,
10240             "readOnly": true,
10241             "type": "array"
10242         },
10243         "di": {
10244             "description": "An identifier formatted according to IETF RFC 4122.",
10245             "type": "string",
10246             "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
10247 9]{12}$",
10248             "readOnly": true
10249         },
10250         "mpro": {
10251             "readOnly": true,
10252             "description": "Supported messaging protocols",
10253             "type": "string",
10254             "maxLength": 64
10255         },
10256         "links": {
10257             "type": "array",
10258             "items": {
10259                 "$ref": "#/definitions/oic.oic-link"
10260             }
10261         }
10262     },
10263     "required": [
10264         "rt",
10265         "if",
10266         "links"
10267     ]
10268 }
10269 },
10270 "oic.oic-link": {
10271     "type": "object",
10272     "properties": {
10273         "anchor": {
10274             "description": "This is used to override the context URI e.g. override the URI of the
10275 containing collection.",
10276             "format": "uri",
10277             "maxLength": 256,
10278             "type": "string"
10279         },
10280         "di": {
10281             "description": "The Device ID formatted according to IETF RFC 4122.",
10282             "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
10283 9]{12}$",

```

```

10284     "type": "string"
10285   },
10286   "eps": {
10287     "description": "the Endpoint information of the target Resource",
10288     "items": {
10289       "properties": {
10290         "ep": {
10291           "description": "Transport Protocol Suite + Endpoint Locator",
10292           "format": "uri",
10293           "type": "string"
10294         },
10295         "pri": {
10296           "description": "The priority among multiple Endpoints",
10297           "minimum": 1,
10298           "type": "integer"
10299         }
10300       },
10301       "type": "object"
10302     },
10303     "type": "array"
10304   },
10305   "href": {
10306     "description": "This is the target URI, it can be specified as a Relative Reference or
10307 fully-qualified URI.",
10308     "format": "uri",
10309     "maxLength": 256,
10310     "type": "string"
10311   },
10312   "if": {
10313     "description": "The interface set supported by this resource",
10314     "items": {
10315       "enum": [
10316         "oic.if.baseline",
10317         "oic.if.ll",
10318         "oic.if.b",
10319         "oic.if.rw",
10320         "oic.if.r",
10321         "oic.if.a",
10322         "oic.if.s"
10323       ],
10324       "type": "string"
10325     },
10326     "minItems": 1,
10327     "type": "array"
10328   },
10329   "ins": {
10330     "description": "The instance identifier for this web link in an array of web links - used
10331 in collections",
10332     "type": "integer"
10333   },
10334   "p": {
10335     "description": "Specifies the framework policies on the Resource referenced by the target
10336 URI",
10337     "properties": {
10338       "bm": {
10339         "description": "Specifies the framework policies on the Resource referenced by the
10340 target URI for e.g. observable and discoverable",
10341         "type": "integer"
10342       }
10343     },
10344     "required": [
10345       "bm"
10346     ],
10347     "type": "object"
10348   },
10349   "rel": {
10350     "description": "The relation of the target URI referenced by the link to the context URI",
10351     "oneOf": [
10352       {
10353         "default": [
10354           "hosts"

```

```

10355         ],
10356         "items": {
10357             "maxLength": 64,
10358             "type": "string"
10359         },
10360         "minItems": 1,
10361         "type": "array"
10362     },
10363     {
10364         "default": "hosts",
10365         "maxLength": 64,
10366         "type": "string"
10367     }
10368 ],
10369 },
10370 "rt": {
10371     "description": "Resource Type of the Resource",
10372     "items": {
10373         "maxLength": 64,
10374         "type": "string"
10375     },
10376     "minItems": 1,
10377     "type": "array"
10378 },
10379 "title": {
10380     "description": "A title for the link relation. Can be used by the UI to provide a
10381 context.",
10382     "maxLength": 64,
10383     "type": "string"
10384 },
10385 "type": {
10386     "default": "application/cbor",
10387     "description": "A hint at the representation of the resource referenced by the target URI.
10388 This represents the media types that are used for both accepting and emitting.",
10389     "items": {
10390         "maxLength": 64,
10391         "type": "string"
10392     },
10393     "minItems": 1,
10394     "type": "array"
10395 }
10396 },
10397 "required": ["href", "rt", "if"]
10398 }
10399 }
10400 }
10401

```

E.3.5 Property definition

Table E.4 defines the Properties that are part of the "oic.wk.res" Resource Type.

Table E.4 – The Property definitions of the Resource with type "rt" = "oic.wk.res"

Property name	Value type	Mandatory	Access mode	Description
di	string		Read Only	An identifier formatted according to IETF RFC 4122.
links	array: see schema		Read Write	
links	array: see schema	Yes	Read Write	
if	array: see schema	Yes	Read Only	The interface set supported by this resource
rt	array: see schema	Yes	Read Only	Resource Type of the Resource

n	string	No	Read Only	Human friendly name
di	string	No	Read Only	An identifier formatted according to IETF RFC 4122.
mpro	string	No	Read Only	Supported messaging protocols
ins	integer	No	Read Write	The instance identifier for this web link in an array of web links - used in collections
type	array: see schema	No	Read Write	A hint at the representation of the resource referenced by the target URI. This represents the media types that are used for both accepting and emitting.
eps	array: see schema	No	Read Write	the Endpoint information of the target Resource
if	array: see schema	Yes	Read Write	The interface set supported by this resource
rel	multiple types: see schema	No	Read Write	The relation of the target URI referenced by the link to the context URI
rt	array: see schema	Yes	Read Write	Resource Type of the Resource
anchor	string	No	Read Write	This is used to override the context URI e.g. override the URI of the containing collection.
di	string	No	Read Write	The Device ID formatted according to IETF RFC 4122.
href	string	Yes	Read Write	This is the target URI, it can be specified as a Relative Reference or fully-qualified URI.
title	string	No	Read Write	A title for the link relation. Can be used by the UI to provide a context.
p	object: see schema	No	Read Write	Specifies the framework policies on the Resource referenced by the target URI

10405 **E.3.6 CRUDN behaviour**

10406 Table E.5 defines the CRUDN operations that are supported on the None Resource Type

10407 **Table E.5 – The CRUDN operations of the Resource with type 'rt' = ['oic.wk.res']**

Create	Read	Update	Delete	Notify
	get			observe

10408

10409

10410 Annex F
10411 (informative)

10412 **OpenAPI 2.0 Schema Extension**
10413

10414 **F.1 OpenAPI 2.0 Schema Reference**

10415 OpenAPI 2.0 does not support allOf and anyOf JSON schema validation constructs; this document
10416 has extended the underlying OpenAPI 2.0 schema to enable these, all OpenAPI 2.0 files are valid
10417 against the extended schema. Reference the following location for a copy of the extended schema:

10418 <https://github.com/openconnectivityfoundation/OCFswagger2.0-schema>

10419 **F.2 OpenAPI 2.0 Introspection empty file**

10420 Reference the following location for a copy of an empty OpenAPI 2.0 file:

10421 [https://github.com/openconnectivityfoundation/DeviceBuilder/blob/master/examples/introspection-](https://github.com/openconnectivityfoundation/DeviceBuilder/blob/master/examples/introspection-empty.txt)
10422 [empty.txt](https://github.com/openconnectivityfoundation/DeviceBuilder/blob/master/examples/introspection-empty.txt)
10423