

# OCF Core Specification

VERSION 2.0.4 | July 2019



**OPEN** CONNECTIVITY  
FOUNDATION™

CONTACT [admin@openconnectivity.org](mailto:admin@openconnectivity.org)

Copyright Open Connectivity Foundation, Inc. © 2019  
All Rights Reserved.

## Legal Disclaimer

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

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.....	9
27	4.1	Conventions.....	9
28	4.2	Notation.....	9
29	4.3	Data types .....	10
30	5	Architecture.....	11
31	5.1	Overview .....	11
32	5.2	Principle .....	12
33	5.3	Functional block diagram .....	13
34	5.4	Framework.....	15
35	5.5	Example Scenario with roles .....	15
36	5.6	Example Scenario: Bridging to Non- OCF ecosystem.....	16
37	5.7	OCF Cloud architecture .....	17
38	6	Identification and addressing.....	19
39	6.1	Introduction.....	19
40	6.2	Identification .....	19
41	6.2.1	Overview .....	19
42	6.2.2	Resource identification and addressing .....	19
43	6.3	Namespace:.....	20
44	6.4	Network addressing .....	21
45	7	Resource model .....	21
46	7.1	Introduction.....	21
47	7.2	Resource .....	22
48	7.3	Property.....	22
49	7.3.1	Introduction .....	22
50	7.3.2	Common Properties .....	23
51	7.4	Resource Type .....	25
52	7.4.1	Introduction .....	25
53	7.4.2	Resource Type Property .....	25
54	7.4.3	Resource Type definition .....	25
55	7.4.4	Multi-value "rt" Resource .....	27
56	7.5	Device Type.....	27
57	7.6	OCF Interface .....	28
58	7.6.1	Introduction .....	28
59	7.6.2	OCF Interface Property.....	28
60	7.6.3	OCF Interface methods.....	29
61	7.7	Resource representation.....	43
62	7.8	Structure.....	43

63	7.8.1	Introduction .....	43
64	7.8.2	Resource Relationships .....	43
65	7.8.3	Collections.....	49
66	7.8.4	Atomic Measurement.....	51
67	7.9	Third (3 <sup>rd</sup> ) party specified extensions .....	53
68	7.10	Query Parameters.....	54
69	7.10.1	Introduction .....	54
70	7.10.2	Use of multiple parameters within a query .....	54
71	7.10.3	Application to multi-value "rt" Resources .....	55
72	7.10.4	OCF Interface specific considerations for queries .....	55
73	8	CRUDN .....	55
74	8.1	Overview .....	55
75	8.2	CREATE .....	56
76	8.2.1	Overview .....	56
77	8.2.2	CREATE request .....	56
78	8.2.3	Processing by the Server.....	57
79	8.2.4	CREATE response.....	57
80	8.3	RETRIEVE.....	57
81	8.3.1	Overview .....	57
82	8.3.2	RETRIEVE request.....	57
83	8.3.3	Processing by the Server.....	57
84	8.3.4	RETRIEVE response .....	58
85	8.4	UPDATE .....	58
86	8.4.1	Overview .....	58
87	8.4.2	UPDATE request .....	58
88	8.4.3	Processing by the Server.....	58
89	8.4.4	UPDATE response.....	59
90	8.5	DELETE.....	59
91	8.5.1	Overview .....	59
92	8.5.2	DELETE request.....	59
93	8.5.3	Processing by the Server.....	60
94	8.5.4	DELETE response .....	60
95	8.6	NOTIFY .....	60
96	8.6.1	Overview .....	60
97	8.6.2	NOTIFICATION response .....	60
98	9	Network and connectivity.....	60
99	9.1	Introduction.....	60
100	9.2	Architecture .....	61
101	9.3	IPv6 network layer requirements .....	62
102	9.3.1	Introduction .....	62
103	9.3.2	IPv6 node requirements.....	62
104	10	OCF Endpoint.....	62
105	10.1	OCF Endpoint definition.....	62
106	10.2	OCF Endpoint information.....	63

107	10.2.1	Introduction .....	63
108	10.2.2	"ep" .....	63
109	10.2.3	"pri" .....	64
110	10.2.4	OCF Endpoint information in "eps" Parameter .....	64
111	10.3	OCF Endpoint discovery .....	65
112	10.3.1	Introduction .....	65
113	10.3.2	Implicit discovery .....	65
114	10.3.3	Explicit discovery with "/oic/res" response .....	65
115	10.4	CoAP based OCF Endpoint discovery .....	70
116	11	Functional interactions .....	71
117	11.1	Introduction.....	71
118	11.2	Onboarding, Provisioning and Configuration .....	71
119	11.3	Resource discovery .....	73
120	11.3.1	Introduction .....	73
121	11.3.2	Resource based discovery: mechanisms .....	73
122	11.3.3	Resource based discovery: Information publication process .....	75
123	11.3.4	Resource based discovery: Finding information .....	76
124	11.3.5	Resource discovery using "/oic/res" .....	82
125	11.3.6	Resource Directory (RD) based discovery .....	84
126	11.4	Notification .....	95
127	11.4.1	Overview .....	95
128	11.4.2	Observe.....	96
129	11.5	Device management .....	97
130	11.5.1	Overview .....	97
131	11.5.2	Diagnostics and maintenance .....	97
132	11.5.3	Network monitoring.....	98
133	11.5.4	Software update Resource.....	102
134	11.6	Scenes .....	107
135	11.6.1	Introduction .....	107
136	11.6.2	Scenes .....	107
137	11.6.3	Security considerations.....	111
138	11.7	Icons.....	112
139	11.7.1	Overview .....	112
140	11.7.2	Resource .....	112
141	11.8	Introspection .....	112
142	11.8.1	Overview .....	112
143	11.8.2	Usage of Introspection.....	115
144	11.9	Alerts .....	116
145	11.9.1	Overview .....	116
146	11.9.2	Resource Types .....	117
147	11.9.3	Example of Use .....	118
148	12	Messaging.....	118
149	12.1	Introduction.....	118
150	12.2	Mapping of CRUDN to CoAP.....	119

151	12.2.1	Overview .....	119
152	12.2.2	URIs .....	119
153	12.2.3	CoAP method with request and response .....	119
154	12.2.4	Content-Format negotiation .....	121
155	12.2.5	OCF-Content-Format-Version information.....	121
156	12.2.6	Content-Format policy .....	122
157	12.2.7	CRUDN to CoAP response codes .....	124
158	12.2.8	CoAP block transfer.....	124
159	12.3	Mapping of CRUDN to CoAP serialization over TCP .....	125
160	12.3.1	Overview .....	125
161	12.3.2	URIs .....	125
162	12.3.3	CoAP method with request and response .....	125
163	12.3.4	Content-Format negotiation .....	125
164	12.3.5	OCF-Content-Format-Version information.....	125
165	12.3.6	Content-Format policy .....	125
166	12.3.7	CRUDN to CoAP response codes .....	125
167	12.3.8	CoAP block transfer.....	125
168	12.3.9	Keep alive (connection health).....	125
169	12.4	Payload Encoding in CBOR .....	126
170	13	Security .....	126
171	Annex A (informative)	Operation Examples .....	127
172	A.1	Introduction.....	127
173	A.2	When at home: From smartphone turn on a single light.....	127
174	A.3	GroupAction execution.....	128
175	A.4	When garage door opens, turn on lights in hall; also notify smartphone .....	128
176	A.5	Device management .....	128
177	Annex B (informative)	OCF interaction scenarios and deployment models .....	130
178	B.1	OCF interaction scenarios.....	130
179	B.2	Deployment model .....	131
180	Annex C (informative)	Other Resource models and OCF mapping .....	133
181	C.1	Multiple Resource models .....	133
182	C.2	OCF approach for support of multiple Resource models.....	133
183	C.3	Resource model indication .....	134
184	C.4	An Example Profile (OMA SpecWorks profile) .....	134
185	C.4.1	Overview .....	134
186	C.5	Conceptual equivalence.....	134
187	C.5.1	Resource Type: Light Control .....	135
188	Annex D (normative)	Resource Type definitions.....	137
189	D.1	List of Resource Type definitions .....	137
190	D.2	Atomic Measurement links list representation .....	137
191	D.2.1	Introduction .....	137
192	D.2.2	Example URI .....	137
193	D.2.3	Resource type .....	138
194	D.2.4	OpenAPI 2.0 definition.....	138

195	D.2.5	Property definition .....	144
196	D.2.6	CRUDN behaviour .....	145
197	D.3	Collection.....	145
198	D.3.1	Introduction .....	145
199	D.3.2	Example URI .....	145
200	D.3.3	Resource type .....	145
201	D.3.4	OpenAPI 2.0 definition.....	146
202	D.3.5	Property definition .....	153
203	D.3.6	CRUDN behaviour .....	154
204	D.4	Device Configuration.....	154
205	D.4.1	Introduction .....	154
206	D.4.2	Example URI .....	154
207	D.4.3	Resource type .....	154
208	D.4.4	OpenAPI 2.0 definition.....	154
209	D.4.5	Property definition .....	159
210	D.4.6	CRUDN behaviour .....	159
211	D.5	Platform Configuration .....	160
212	D.5.1	Introduction .....	160
213	D.5.2	Example URI .....	160
214	D.5.3	Resource type .....	160
215	D.5.4	OpenAPI 2.0 definition.....	160
216	D.5.5	Property definition .....	163
217	D.5.6	CRUDN behaviour .....	163
218	D.6	Device .....	163
219	D.6.1	Introduction .....	163
220	D.6.2	Well-known URI.....	164
221	D.6.3	Resource type .....	164
222	D.6.4	OpenAPI 2.0 definition.....	164
223	D.6.5	Property definition .....	167
224	D.6.6	CRUDN behaviour .....	168
225	D.7	Icon .....	168
226	D.7.1	Introduction .....	168
227	D.7.2	Example URI .....	168
228	D.7.3	Resource type .....	168
229	D.7.4	OpenAPI 2.0 definition.....	168
230	D.7.5	Property definition .....	170
231	D.7.6	CRUDN behaviour .....	170
232	D.8	Introspection Resource .....	171
233	D.8.1	Introduction .....	171
234	D.8.2	Well-known URI.....	171
235	D.8.3	Resource type .....	171
236	D.8.4	OpenAPI 2.0 definition.....	171
237	D.8.5	Property definition .....	173
238	D.8.6	CRUDN behaviour .....	173

239	D.9	Maintenance .....	174
240	D.9.1	Introduction .....	174
241	D.9.2	Well-known URI .....	174
242	D.9.3	Resource type .....	174
243	D.9.4	OpenAPI 2.0 definition.....	174
244	D.9.5	Property definition .....	177
245	D.9.6	CRUDN behaviour .....	177
246	D.10	Network Monitoring .....	177
247	D.10.1	Introduction .....	177
248	D.10.2	Example URI .....	177
249	D.10.3	Resource type .....	177
250	D.10.4	OpenAPI 2.0 definition.....	177
251	D.10.5	Property definition .....	180
252	D.10.6	CRUDN behaviour .....	181
253	D.11	Platform .....	181
254	D.11.1	Introduction .....	181
255	D.11.2	Well-known URI .....	181
256	D.11.3	Resource type .....	182
257	D.11.4	OpenAPI 2.0 definition.....	182
258	D.11.5	Property definition .....	184
259	D.11.6	CRUDN behaviour .....	185
260	D.12	Resource directory resource .....	185
261	D.12.1	Introduction .....	185
262	D.12.2	Well-known URI .....	185
263	D.12.3	Resource type .....	185
264	D.12.4	OpenAPI 2.0 definition.....	186
265	D.12.5	Property definition .....	190
266	D.12.6	CRUDN behaviour .....	190
267	D.13	Discoverable Resources .....	191
268	D.13.1	Introduction .....	191
269	D.13.2	Well-known URI .....	191
270	D.13.3	Resource type .....	191
271	D.13.4	OpenAPI 2.0 definition.....	191
272	D.13.5	Property definition .....	195
273	D.13.6	CRUDN behaviour .....	196
274	D.14	Scene List.....	196
275	D.14.1	Introduction .....	196
276	D.14.2	Example URI .....	196
277	D.14.3	Resource type .....	196
278	D.14.4	OpenAPI 2.0 definition.....	196
279	D.14.5	Property definition .....	200
280	D.14.6	CRUDN behaviour .....	200
281	D.15	Scene Collection.....	201
282	D.15.1	Introduction .....	201



283	D.15.2	Example URI .....	201
284	D.15.3	Resource type .....	201
285	D.15.4	OpenAPI 2.0 definition.....	201
286	D.15.5	Property definition .....	205
287	D.15.6	CRUDN behaviour .....	206
288	D.16	Scene Member.....	207
289	D.16.1	Introduction .....	207
290	D.16.2	Example URI .....	207
291	D.16.3	Resource type .....	207
292	D.16.4	OpenAPI 2.0 definition.....	207
293	D.16.5	Property definition .....	210
294	D.16.6	CRUDN behaviour .....	211
295	D.17	Alert.....	211
296	D.17.1	Introduction .....	211
297	D.17.2	Example URI .....	212
298	D.17.3	Resource type .....	212
299	D.17.4	OpenAPI 2.0 definition.....	212
300	D.17.5	Property definition .....	214
301	D.17.6	CRUDN behaviour .....	215
302	D.18	Alert Collection .....	215
303	D.18.1	Introduction .....	215
304	D.18.2	Example URI .....	215
305	D.18.3	Resource type .....	215
306	D.18.4	OpenAPI 2.0 definition.....	215
307	D.18.5	Property definition .....	219
308	D.18.6	CRUDN behaviour .....	220
309	D.19	software update .....	221
310	D.19.1	Introduction .....	221
311	D.19.2	Example URI .....	221
312	D.19.3	Resource type .....	221
313	D.19.4	OpenAPI 2.0 definition.....	221
314	D.19.5	Property definition .....	224
315	D.19.6	CRUDN behaviour .....	225
316	Annex E (informative)	OIC 1.1 Resource Type definitions .....	226
317	E.1	List of Resource Type Definitions.....	226
318	E.2	OCF Collection .....	226
319	E.2.1	Introduction .....	226
320	E.2.2	Wellknown URI .....	226
321	E.2.3	Resource type .....	226
322	E.2.4	OpenAPI 2.0 definition.....	226
323	E.2.5	Property definition .....	236
324	E.2.6	CRUDN behaviour .....	238
325	E.3	Discoverable Resources .....	238
326	E.3.1	Introduction .....	238

327	E.3.2	Wellknown URI .....	238
328	E.3.3	Resource type .....	238
329	E.3.4	OpenAPI 2.0 definition.....	238
330	E.3.5	Property definition .....	243
331	E.3.6	CRUDN behaviour .....	245
332	Annex F (informative) OpenAPI 2.0 Schema Extension .....		246
333	F.1	OpenAPI 2.0 Schema Reference.....	246
334	F.2	OpenAPI 2.0 Introspection empty file .....	246
335			
336			

337  
338  
339

## Figures

340	Figure 1 – Architecture - concepts .....	13
341	Figure 2 – Functional block diagram .....	14
342	Figure 3 – Communication layering model .....	15
343	Figure 4 – Example illustrating the roles .....	16
344	Figure 5 – Framework - Architecture Detail .....	17
345	Figure 6 – Server bridging to Non- OCF device.....	17
346	Figure 7 – OCF Cloud deployment architecture.....	18
347	Figure 8 – OCF Endpoint routing .....	18
348	Figure 9 – Example Resource .....	22
349	Figure 10 – CREATE operation.....	56
350	Figure 11 – RETRIEVE operation.....	57
351	Figure 12 – UPDATE operation .....	58
352	Figure 13 – DELETE operation .....	59
353	Figure 14 – High Level Network & Connectivity Architecture .....	61
354	Figure 15 – Resource based discovery: Information publication process.....	75
355	Figure 16 – Resource based discovery: Finding information.....	76
356	Figure 17 – Indirect discovery of Resources by via an RD.....	85
357	Figure 18 – RD discovery and RD supported query of Resources support.....	87
358	Figure 19 – Resource Direction Deployment Scenarios.....	88
359	Figure 20 – Observe Mechanism.....	96
360	Figure 21 – Interactions with the network monitoring Resource .....	101
361	Figure 22 – State transition diagram of collecting network information .....	102
362	Figure 23 – Typical state transitioning diagram for software update .....	104
363	Figure 24 – Typical sequence for none scheduled upgrading software .....	107
364	Figure 25 – Generic Scene Resource structure .....	107
365	Figure 26 – Interactions to check Scene support and setup of specific Scenes .....	108
366	Figure 27 – Client interactions on a specific Scene .....	109
367	Figure 28 – Interaction overview due to a Scene change .....	111
368	Figure 29 – Example usage of oneOf JSON schema .....	115
369	Figure 30 – Interactions to check Introspection support and download the Introspection Device Data.....	116
371	Figure 31 – Content-Format Policy for OCF Servers supporting error responses and backward compatibility responses.....	123
373	Figure 32 – Content-Format Policy for OCF Clients supporting error responses and backward compatibility responses.....	124
375	Figure 33 – Content-Format Policy for backward compatible OCF Clients negotiating lower OCF Content-Format-Version .....	124
377	Figure A.1 – When at home: from smartphone turn on a single light.....	128

378	Figure A.2 – Device management (maintenance) .....	129
379	Figure B.1 – Direct interaction between Server and Client .....	130
380	Figure B.2– Interaction between Client and Server using another Server .....	130
381	Figure B.3 – Interaction between Client and Server using Intermediary.....	130
382	Figure B.4 – Interaction between Client and Server using support from multiple Servers	
383	and Intermediary.....	131
384	Figure B.5 – Example of Devices .....	131

385

## Tables

386  
387

388	Table 1 – Additional OCF Types .....	10
389	Table 2 – Name Property Definition .....	24
390	Table 3 – Resource Identity Property Definition .....	24
391	Table 4 – Resource Type Common Property definition.....	25
392	Table 5 – Example foobar Resource Type.....	26
393	Table 6 – Example foobar Properties .....	26
394	Table 7 – Resource Interface Property definition.....	28
395	Table 8 – OCF standard OCF Interfaces .....	29
396	Table 9 – Batch OCF Interface Example .....	34
397	Table 10 – "bm" Property definition.....	45
398	Table 11 – Resource Types Property definition .....	48
399	Table 12 – Mandatory Resource Types Property definition.....	48
400	Table 13 – Common Properties for Collections (in addition to Common Properties defined	
401	in 7.3.2) .....	50
402	Table 14 – Common Properties for Atomic Measurement (in addition to Common	
403	Properties defined in 7.3.2).....	51
404	Table 15 – Atomic Measurement Resource Type .....	53
405	Table 16 – Properties for Atomic Measurement (in addition to Common Properties defined	
406	in 7.3.2) .....	53
407	Table 17 – 3rd party defined Resource elements .....	53
408	Table 18 – Parameters of CRUDN messages.....	56
409	Table 19 – "ep" value for Transport Protocol Suite.....	64
410	Table 20 – List of Core Resources .....	71
411	Table 21 – Configuration Resource.....	71
412	Table 22 – "oic.wk.con" Resource Type definition .....	72
413	Table 23 – "oic.wk.con.p" Resource Type definition .....	73
414	Table 24 – Mandatory discovery Core Resources .....	77
415	Table 25 – "oic.wk.res" Resource Type definition.....	77
416	Table 26 – Protocol scheme registry .....	78
417	Table 27 – "oic.wk.d" Resource Type definition.....	79

418	Table 28 – "oic.wk.p" Resource Type definition.....	81
419	Table 29 – "oic.wk.rd" Resource Type definition .....	86
420	Table 30 – "oic.wk.rd" Properties .....	86
421	Table 31 – Optional diagnostics and maintenance Device management Core Resources.....	97
422	Table 32 – "oic.wk.mnt" Resource Type definition.....	98
423	Table 33 – Optional monitoring Device management Core Resources.....	98
424	Table 34 – "oic.wk.nmon" Resource Type definition .....	99
425	Table 35 – Optional software update Resources .....	102
426	Table 36 – "oic.r.softwareupdate" Resource Type definition.....	103
427	Table 37 State definitions and state transitions of software update Resource .....	103
428	Table 38 Value definitions for the Property "swupdateaction" .....	104
429	Table 39 List of codes of the "swupdateresult" Property.....	105
430	Table 40 – list of Resource Types for Scenes .....	111
431	Table 41 – Optional Icon Core Resource.....	112
432	Table 42 – "oic.r.icon" Resource Type definition .....	112
433	Table 43 – Introspection Resource.....	115
434	Table 44 – "oic.wk.introspection" Resource Type definition.....	115
435	Table 45 – Optional Alert Core Resources .....	117
436	Table 46 – "oic.r.alert" Resource Type definition.....	117
437	Table 47 – "oic.r.alertcollection" Resource Type definition .....	118
438	Table 48 – CoAP request and response .....	119
439	Table 49 – OCF Content-Formats .....	121
440	Table 50 – OCF-Content-Format-Version and OCF-Accept-Content-Format-Version Option	
441	Numbers .....	121
442	Table 51 – OCF-Accept-Content-Format-Version and OCF-Content-Format-Version	
443	Representation .....	122
444	Table 52 – Examples of OCF-Content-Format-Version and OCF-Accept-Content-Format-	
445	Version Representation .....	122
446	Table A-1 – "oic.example.light" Resource Type definition.....	127
447	Table A-2 – oic.example.garagedoor Resource Type definition .....	127
448	Table C-1 – Light control Resource Type definition .....	135
449	Table C-2 – Light control Resource Type definition .....	135
450	Table D-1 – Alphabetized list of Core Resources .....	137
451	Table D-2 – The Property definitions of the Resource with type "rt" =	
452	"oic.wk.atomicmeasurement". .....	144
453	Table D-3 – The CRUDN operations of the Resource with type "rt" =	
454	"oic.wk.atomicmeasurement". .....	145
455	Table D-4 – The Property definitions of the Resource with type "rt" = "oic.wk.col".....	153
456	Table D-5 – The CRUDN operations of the Resource with type "rt" = "oic.wk.col". .....	154
457	Table D-6 – The Property definitions of the Resource with type "rt" = "oic.wk.con".....	159
458	Table D-7 – The CRUDN operations of the Resource with type "rt" = "oic.wk.con". .....	160

459	Table D-8 – The Property definitions of the Resource with type "rt" = "oic.wk.con.p".	163
460	Table D-9 – The CRUDN operations of the Resource with type "rt" = "oic.wk.con.p".	163
461	Table D-10 – The Property definitions of the Resource with type "rt" = "oic.wk.d".	167
462	Table D-11 – The CRUDN operations of the Resource with type "rt" = "oic.wk.d".	168
463	Table D-12 – The Property definitions of the Resource with type "rt" = "oic.r.icon".	170
464	Table D-13 – The CRUDN operations of the Resource with type "rt" = "oic.r.icon".	170
465	Table D-14 – The Property definitions of the Resource with type "rt" =	
466	"oic.wk.introspection".	173
467	Table D-15 – The CRUDN operations of the Resource with type "rt" =	
468	"oic.wk.introspection".	174
469	Table D-16 – The Property definitions of the Resource with type "rt" = "oic.wk.mnt".	177
470	Table D-17 – The CRUDN operations of the Resource with type "rt" = "oic.wk.mnt".	177
471	Table D-18 – The Property definitions of the Resource with type "rt" = "oic.wk.nmon".	181
472	Table D-19 – The CRUDN operations of the Resource with type "rt" = "oic.wk.nmon".	181
473	Table D-20 – The Property definitions of the Resource with type "rt" = "oic.wk.p".	184
474	Table D-21 – The CRUDN operations of the Resource with type "rt" = "oic.wk.p".	185
475	Table D-22 – The Property definitions of the Resource with type "rt" = "oic.wk.rd".	190
476	Table D-23 – The CRUDN operations of the Resource with type "rt" = "oic.wk.rd".	191
477	Table D-24 – The Property definitions of the Resource with type "rt" = "None".	195
478	Table D-25 – The CRUDN operations of the Resource with type "rt" = "None".	196
479	Table D-26 – The Property definitions of the Resource with type "rt" = "oic.wk.scenelist".	200
480	Table D-27 – The CRUDN operations of the Resource with type "rt" = "oic.wk.scenelist".	201
481	Table D-28 – The Property definitions of the Resource with type "rt" =	
482	"oic.wk.scenecollection".	205
483	Table D-29 – The CRUDN operations of the Resource with type "rt" =	
484	"oic.wk.scenecollection".	206
485	Table D-30 – The Property definitions of the Resource with type "rt" =	
486	"oic.wk.scenemember".	210
487	Table D-31 – The CRUDN operations of the Resource with type "rt" =	
488	"oic.wk.scenemember".	211
489	Table D-32 – The Property definitions of the Resource with type "rt" = "oic.r.alert".	214
490	Table D-33 – The CRUDN operations of the Resource with type "rt" = "oic.r.alert".	215
491	Table D-34 – The Property definitions of the Resource with type "rt" =	
492	"oic.r.alertcollection".	220
493	Table D-35 – The CRUDN operations of the Resource with type "rt" =	
494	"oic.r.alertcollection".	220
495	Table D-36 – The Property definitions of the Resource with type "rt" =	
496	"oic.r.softwareupdate".	224
497	Table D-37 – The CRUDN operations of the Resource with type "rt" =	
498	"oic.r.softwareupdate".	225
499	Table E-1 – Alphabetized list of referenced OIC 1.1 Core Resources.	226
500	Table E-2 – The Property definitions of the Resource with type "rt" = "oic.wk.col".	236

501 Table E-3 – The CRUDN operations of the Resource with type 'rt' = ['oic.wk.col'].....238  
502 Table E-4 – The Property definitions of the Resource with type "rt" = "oic.wk.res" .....243  
503 Table E-5 – The CRUDN operations of the Resource with type 'rt' = ['oic.wk.res'] .....245  
504  
505

506

## 507 **1 Scope**

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

- 509 – Core Specification documents: The Core Specification documents specify the Framework, i.e.,  
510 the OCF core architecture, interfaces, protocols and services to enable OCF profiles  
511 implementation for Internet of Things (IoT) usages and ecosystems.
- 512 – Vertical Domain Specification documents: The Vertical Domain Specification documents specify  
513 OCF Device profiles to enable IoT usages for different vertical market segments such as smart  
514 home, industrial, healthcare, and automotive. They also specify Resource definitions to enable  
515 vertical services and use case. Such specifications include ISO/IEC 30118-5:2018 which is built  
516 upon the interfaces and network security of the OCF core architecture defined in the Core  
517 Specification.

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

## 519 **2 Normative references**

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

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

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

527 ISO/IEC 30118-2:2018, Information technology – Open Connectivity Foundation (OCF)  
528 Specification – Part 2: Security specification  
529 <https://www.iso.org/standard/74239.html>  
530 Latest version available at: [https://openconnectivity.org/specs/OCF\\_Security\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Security_Specification.pdf)

531 ISO/IEC 30118-5:2018, Information technology – Open Connectivity Foundation (OCF)  
532 Specification – Part 5: Smart home device specification  
533 <https://www.iso.org/standard/74242.html>  
534 Latest version available at: [https://openconnectivity.org/specs/OCF\\_Device\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Device_Specification.pdf)

535 OCF Easy Wi-Fi Setup, Information technology – Open Connectivity Foundation (OCF)  
536 Specification – Part 7: Wi-Fi Easy Setup specification  
537 Latest version available at: [https://openconnectivity.org/specs/OCF\\_Wi-](https://openconnectivity.org/specs/OCF_Wi-Fi_Easy_Setup_Specification.pdf)  
538 [Fi\\_Easy\\_Setup\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Wi-Fi_Easy_Setup_Specification.pdf)

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

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

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

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

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



549 IETF RFC 4941, *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, September  
550 2007  
551 <https://www.rfc-editor.org/info/rfc4941>

552 IETF RFC 5424, *The Syslog Protocol*, March 2009  
553 <https://tools.ietf.org/html/rfc5424>IETF RFC 5646, *Tags for Identifying Languages*, September  
554 2009  
555 <https://www.rfc-editor.org/info/rfc5646>

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

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

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

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

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

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

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

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

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

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

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

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

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

585 IETF RFC 7959, *Block-Wise Transfers in the Constrained Application Protocol (CoAP)*, August  
586 2016  
587 <https://www.rfc-editor.org/info/rfc7959>

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

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

594 IANA ifType-MIB Definitions  
595 <https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib>

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

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

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

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

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

606 **3 Terms, definitions, and abbreviated terms**

607 **3.1 Terms and definitions**

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

609 ISO and IEC maintain terminological databases for use in standardization at the following  
610 addresses:  
611 – ISO Online browsing platform: available at <https://www.iso.org/obp>.  
612 – IEC Electropedia: available at <http://www.electropedia.org/>.

613 **3.1.1**  
614 **Alert**  
615 information provided by the Device (3.1.14) by means of a specialised Resource Type (3.1.36) that  
616 provides details with regard to potential problems, issues, or Device (3.1.14) status of interest on  
617 which action can be taken

618 **3.1.2**  
619 **Atomic Measurement**  
620 a design pattern that ensures that the Client (3.1.7) can only access the Properties (3.1.35) of  
621 linked Resources (3.1.32) atomically, that is as a single group

622 **3.1.3**  
623 **Bridged Client**  
624 logical entity that accesses data via a Bridged Protocol (3.1.5)

625 Note 1 to entry: For example, an AllJoyn Consumer application is a Bridged Client (3.1.3)

626 **3.1.4**  
627 **Bridged Device**  
628 Bridged Client (3.1.3) or Bridged Server (3.1.6)

629 **3.1.5**  
630 **Bridged Protocol**  
631 another protocol (e.g., AllJoyn) that is being translated to or from OCF protocols

632 **3.1.6**  
633 **Bridged Server**  
634 logical entity that provides data via a Bridged Protocol (3.1.5)

635 Note 1 to entry: For example an AllJoyn Producer is a Bridged Server (3.1.6).  
636 Note 2 to entry: More than one Bridged Server (3.1.6) can exist on the same physical platform.

637 **3.1.7**  
638 **Client**  
639 a logical entity that accesses a Resource (3.1.32) on a Server (3.1.42)

640 **3.1.8**  
641 **Collection**  
642 a Resource (3.1.32) that contains zero or more Links (3.1.22)

643 **3.1.9**  
644 **Common Properties**  
645 Properties (3.1.35) specified for all Resources (3.1.32)

646 **3.1.10**  
647 **Composite Device**  
648 a Device (3.1.14) that is modelled as multiple Device Types (3.1.15); with each component Device  
649 Type (3.1.15) being exposed as a Collection (3.1.8)

650 **3.1.11**  
651 **Configuration Source**  
652 a cloud or service network or a local read-only file which contains and provides configuration  
653 related information to the Devices (3.1.14)

654 **3.1.12**  
655 **Core Resources**  
656 those Resources (3.1.32) that are defined in this document

657 **3.1.13**  
658 **Default OCF Interface**  
659 an OCF Interface (3.1.19) used to generate the response when an OCF Interface (3.1.19) is omitted  
660 in a request

661 **3.1.14**  
662 **Device**  
663 a logical entity that assumes one or more roles, e.g., Client (3.1.7), Server (3.1.42)

664 Note 1 to entry: More than one Device (3.1.14) can exist on a Platform (3.1.31).

665 **3.1.15**  
666 **Device Type**  
667 a uniquely named definition indicating a minimum set of Resource Types (3.1.36) that a Device  
668 (3.1.14) supports

669 Note 1 to entry: A Device Type (3.1.15) provides a hint about what the Device (3.1.14) is, such as a light or a fan, for  
670 use during Resource (3.1.32) discovery.

671 **3.1.16**  
672 **Discoverable Resource**  
673 a Resource (3.1.32) that is listed in "/oic/res"

674 **3.1.17**  
675 **OCF Endpoint**  
676 entity participating in the OCF protocol, further identified as the source or destination of a request  
677 and response messages for a given Transport Protocol Suite

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

### 679 **3.1.18**

#### 680 **Framework**

681 a set of related functionalities and interactions defined in this document, which enable  
682 interoperability across a wide range of networked devices, including IoT

### 683 **3.1.19**

#### 684 **OCF Interface**

685 interface description in accordance with IETF RFC 6690 and as defined by OCF that provides a  
686 view to and permissible responses from a Resource (3.1.32)

### 687 **3.1.20**

#### 688 **Introspection**

689 mechanism to determine the capabilities of the hosted Resources (3.1.32) of a Device (3.1.14)

### 690 **3.1.21**

#### 691 **Introspection Device Data (IDD)**

692 data that describes the payloads per implemented method of the Resources (3.1.32) that make up  
693 the Device (3.1.14)

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

### 695 **3.1.22**

#### 696 **Links**

697 extends typed web links according to IETF RFC 5988

### 698 **3.1.23**

#### 699 **Non-Discoverable Resource**

700 a Resource (3.1.32) that is not listed in "/oic/res"

701 Note 1 to entry: The Resource (3.1.32) can be reached by a Link (3.1.22) which is conveyed by another Resource  
702 (3.1.32). For example a Resource (3.1.32) linked in a Collection (3.1.8) does not have to be listed in "/oic/res", since  
703 traversing the Collection (3.1.8) would discover the Resource (3.1.32) implemented on the Device (3.1.14).

### 704 **3.1.24**

#### 705 **Non-OCF Device**

706 a Device (3.1.14) which does not comply with the OCF Device (3.1.14) requirements

### 707 **3.1.25**

#### 708 **Notification**

709 the mechanism to make a Client (3.1.7) aware of state changes in a Resource (3.1.32)

### 710 **3.1.26**

#### 711 **Observe**

712 the act of monitoring a Resource (3.1.32) by sending a RETRIEVE operation which is cached by  
713 the Server (3.1.42) hosting the Resource (3.1.32) and reprocessed on every change to that  
714 Resource (3.1.32)

### 715 **3.1.27**

#### 716 **OpenAPI 2.0**

717 Resource (3.1.32) and Introspection Device Data (3.1.21) definitions used in this document as  
718 defined in the OpenAPI specification

### 719 **3.1.28**

#### 720 **Parameter**

721 an element that provides metadata about a Resource (3.1.32) referenced by the target URI of a  
722 Link (3.1.22)

723 **3.1.29**  
724 **Partial UPDATE**  
725 an UPDATE operation to a Resource (3.1.32) that includes a subset of the Properties (3.1.35) that  
726 are visible via the OCF Interface (3.1.19) being applied for the Resource Type (3.1.36)

727 **3.1.30**  
728 **Physical Device**  
729 the physical thing on which a Device(s) (3.1.14) is exposed

730 **3.1.31**  
731 **Platform**  
732 a Physical Device (3.1.30) containing one or more Devices (3.1.14)

733 **3.1.32**  
734 **Resource**  
735 represents an entity modelled and exposed by the Framework (3.1.18)

736 **3.1.33**  
737 **Resource Directory**  
738 a set of descriptions of Resources (3.1.32) where the actual Resources (3.1.32) are held on Servers  
739 (3.1.42) external to the Device (3.1.14) hosting the Resource Directory (3.1.33), allowing lookups  
740 to be performed for those Resources (3.1.32)

741 Note 1 to entry: This functionality can be used by sleeping Servers (3.1.42) or Servers (3.1.42) that choose not to  
742 listen/respond to multicast requests directly.

743 **3.1.34**  
744 **Resource Interface**  
745 a qualification of the permitted requests on a Resource (3.1.32)

746 **3.1.35**  
747 **Property**  
748 a significant aspect or Parameter (3.1.28) of a Resource (3.1.32), including metadata, that is  
749 exposed through the Resource (3.1.32)

750 **3.1.36**  
751 **Resource Type**  
752 a uniquely named definition of a class of Properties (3.1.35) and the interactions that are supported  
753 by that class

754 Note 1 to entry: Each Resource (3.1.32) has a Property (3.1.35) "rt" whose value is the unique name of the Resource  
755 Type (3.1.36).

756 **3.1.37**  
757 **Scene**  
758 a static entity that stores a set of defined Property (3.1.35) values for a Collection (3.1.8) of  
759 Resources (3.1.32)

760 Note 1 to entry: A Scene (3.1.37) is a prescribed setting of a set of Resources (3.1.32) with each having a predetermined  
761 value for the Property (3.1.35) that has to change.

762 **3.1.38**  
763 **Scene Collection**  
764 a Collection (3.1.8) that contains an enumeration of possible Scene Values (3.1.40) and the current  
765 Scene Value (3.1.40)

766 Note 1 to entry: The member values of the Scene Collection (3.1.38) are Scene Members (3.1.39).

767 **3.1.39**  
768 **Scene Member**  
769 a Resource (3.1.32) that contains mappings of Scene Values (3.1.40) to values of a Property  
770 (3.1.35) in the Resource (3.1.32)

771 **3.1.40**  
772 **Scene Value**  
773 a Scene (3.1.37) enumerator representing the state in which a Resource (3.1.32) can be

774 **3.1.41**  
775 **Secure OCF Endpoint**  
776 an OCF Endpoint (3.1.17) with a secure connection (e.g., CoAPS)

777 **3.1.42**  
778 **Server**  
779 a Device (3.1.14) with the role of providing Resource (3.1.32) state information and facilitating  
780 remote interaction with its Resources (3.1.32)

781 Note 1 to entry: A Server (3.1.42) can be implemented to expose Non-OCF Device (3.1.24) resources to Clients (3.1.7)  
782 (see 5.6).

783 **3.1.43**  
784 **Unsecure OCF Endpoint**  
785 an OCF Endpoint ( ) with an unsecure connection (e.g., CoAP)

786 **3.1.44**  
787 **Vertical Resource Type**  
788 a Resource Type (3.1.36) in a vertical domain specification

789 Note 1 to entry: An example of a Vertical Resource Type (3.1.44) would be "oic.r.switch.binary".

790 **3.1.45**  
791 **Virtual OCF Client**  
792 logical representation of a Bridged Client (3.1.3), which an Bridged Device (3.1.4) exposes to  
793 Servers (3.1.42)

794 **3.1.46**  
795 **Virtual OCF Device (or VOD)**  
796 Virtual OCF Client (3.1.45) or Virtual OCF Server (3.1.47)

797 **3.1.47**  
798 **Virtual OCF Server**  
799 logical representation of a Bridged Server (3.1.6), which an Bridged Device (3.1.4) exposes to  
800 Clients (3.1.7)

## 801 **3.2 Abbreviated terms**

802 **3.2.1**  
803 **ACL**  
804 Access Control List

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

806 **3.2.2**  
807 **BLE**  
808 Bluetooth Low Energy

809 **3.2.3**  
810 **CBOR**  
811 Concise Binary Object Representation

812 **3.2.4**  
813 **CoAP**  
814 Constrained Application Protocol

815 **3.2.5**  
816 **CoAPS**  
817 Secure Constrained Application Protocol

818 **3.2.6**  
819 **DTLS**  
820 Datagram Transport Layer Security

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

822 **3.2.7**  
823 **EXI**  
824 Efficient XML Interchange

825 **3.2.8**  
826 **IP**  
827 Internet Protocol

828 **3.2.9**  
829 **IRI**  
830 Internationalized Resource Identifiers

831 **3.2.10**  
832 **ISP**  
833 Internet Service Provider

834 **3.2.11**  
835 **JSON**  
836 JavaScript Object Notation

837 **3.2.12**  
838 **mDNS**  
839 Multicast Domain Name Service

840 **3.2.13**  
841 **MTU**  
842 Maximum Transmission Unit

843 **3.2.14**  
844 **NAT**  
845 Network Address Translation

846 **3.2.15**  
847 **OCF**  
848 Open Connectivity Foundation

849 the organization that created this document

850 **3.2.16**  
851 **REST**  
852 Representational State Transfer

853 **3.2.17**  
854 **RESTful**  
855 REST-compliant Web services

856 **3.2.18**  
857 **UDP**  
858 User Datagram Protocol

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

860 **3.2.19**  
861 **URI**  
862 Uniform Resource Identifier

863 **3.2.20**  
864 **URN**  
865 Uniform Resource Name

866 **3.2.21**  
867 **UTC**  
868 Coordinated Universal Time

869 **3.2.22**  
870 **UUID**  
871 Universal Unique Identifier

872 **3.2.23**  
873 **XML**  
874 Extensible Markup Language

## 875 **4 Document conventions and organization**

### 876 **4.1 Conventions**

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

### 881 **4.2 Notation**

882 In this document, features are described as required, recommended, allowed or DEPRECATED as  
883 follows:

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

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

888 Recommended (or should)(S).

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

895 Allowed (may or allowed)(O).



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

899 DEPRECATED.

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

906 Conditionally allowed (CA).

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

909 Conditionally required (CR).

910 – The definition or behaviour depends on a condition. If the specified condition is met, then the  
911 definition or behaviour is required. Otherwise the definition or behaviour is allowed as default  
912 unless specifically defined as not allowed.

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

914 Words that are emphasized are printed in italic.

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

### 919 4.3 Data types

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

923 Among other validation keywords, clause 7 in JSON Schema Validation defines a "format" keyword  
924 with a number of format attributes such as "uri" and "date-time", and a "pattern" keyword with a  
925 regular expression that can be used to validate a string. This clause defines patterns that are  
926 available for use in describing OCF Resources. The pattern names can be used in document text  
927 where JSON format names can occur. The actual JSON schemas shall use the JSON type and  
928 pattern instead.

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

930 **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)?))?)\$ ^P([0-9]+W)\$ ^P([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])\$ ^P([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.

931

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

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

## 936 5 Architecture

### 937 5.1 Overview

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

942 Specifically, the architecture provides:

- 943 – A communication and interoperability framework for multiple market segments (Consumer,  
 944 Enterprise, Industrial, Automotive, Health, etc.), OSs, platforms, modes of communication,  
 945 transports and use cases.

- 946 – A common and consistent model for describing the environment and enabling information and  
947 semantic interoperability.
- 948 – Common communication protocols for discovery and connectivity.
- 949 – Common security and identification mechanisms.
- 950 – Opportunity for innovation and product differentiation.
- 951 – A scalable solution addressing different Device capabilities, applicable to smart devices as well  
952 as the smallest connected things and wearable devices.

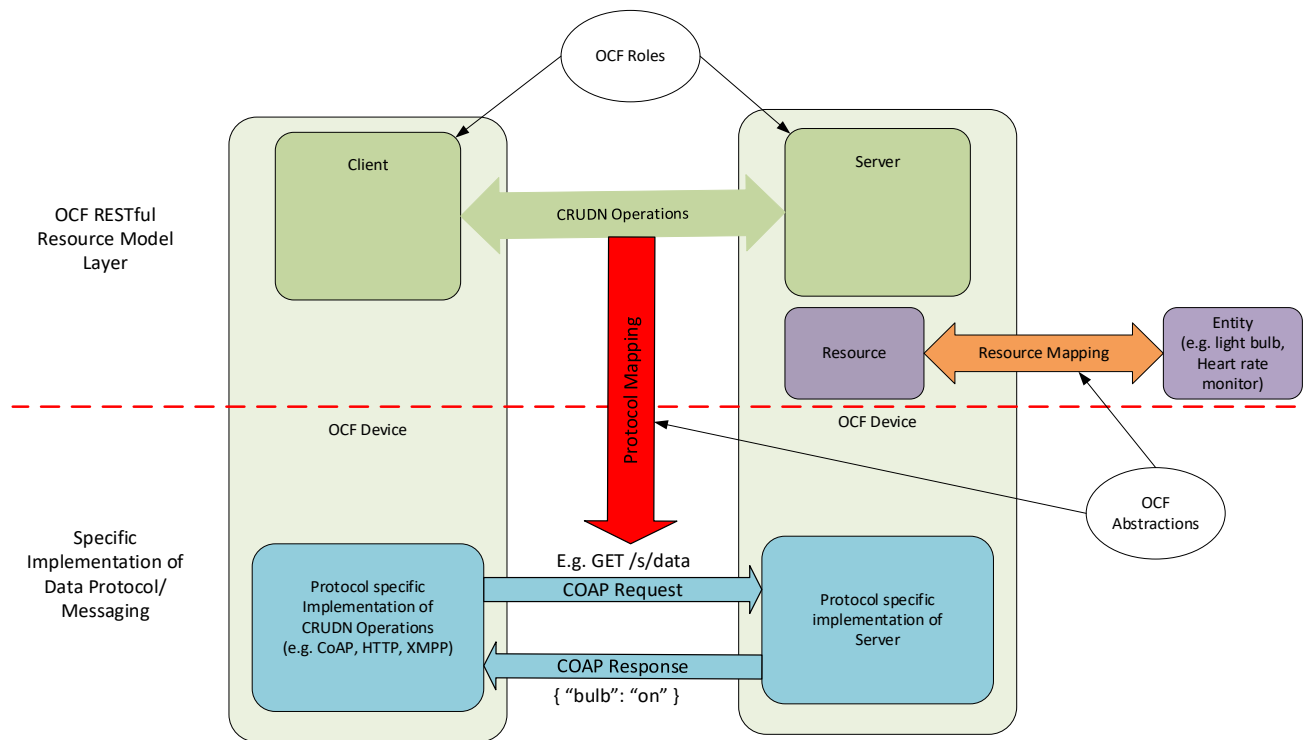
953 The architecture is based on the Resource Oriented Architecture design principles and described  
954 in the 5.2 through 5.6 respectively. 5.2 presents the guiding principles for OCF operations. 5.3  
955 defines the functional block diagram and Framework. 5.5 provides an example scenario with roles.  
956 5.6 provides an example scenario of bridging to non- OCF ecosystem.

## 957 **5.2 Principle**

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

962 The architecture defines the overall structure of the Framework as an information system and the  
963 interrelationships of the Entities that make up OCF. Entities are exposed as Resources, with their  
964 unique identifiers (URIs) and support interfaces that enable RESTful operations on the Resources.  
965 Every RESTful operation has an initiator of the operation (the Client) and a responder to the  
966 operation (the Server). In the Framework, the notion of the Client and Server is realized through  
967 roles (see 5.5). Any Device can act as a Client and initiate a RESTful operation on any Device  
968 acting as a Server. Likewise, any Device that exposes Entities as Resources acts as a Server.  
969 Conformant to the REST architectural style, each RESTful operation contains all the information  
970 necessary to understand the context of the interaction and is driven using a small set of generic  
971 operations, i.e., CREATE, RETRIEVE, UPDATE, DELETE and NOTIFY (CRUDN) defined in clause  
972 8, which include representations of Resources.

973 Figure 1 depicts the architecture.



974

975

**Figure 1 – Architecture - concepts**

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

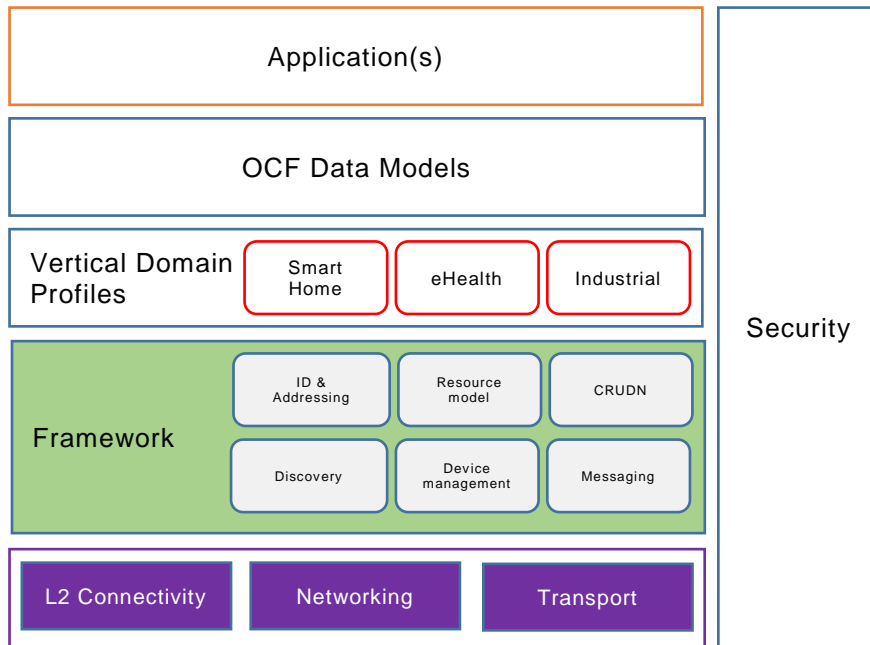
978 – Resource model: The Resource model provides the abstractions and concepts required to  
979 logically model, and logically operate on the application and its environment. The Core  
980 Resource model is common and agnostic to any specific application domain such as smart  
981 home, industrial or automotive. For example, the Resource model defines a Resource which  
982 abstracts an entity and the representation of a Resource maps the entity's state. Other  
983 Resource model concepts can be used to model other aspects, for example behaviour.

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

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

### 993 5.3 Functional block diagram

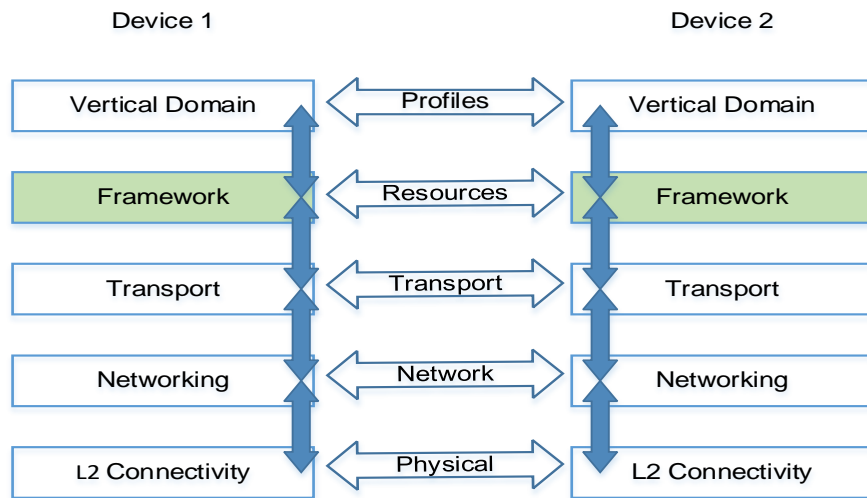
994 The functional block diagram encompasses all the functionalities required for operation. These  
995 functionalities are categorized as L2 connectivity, networking, transport, Framework, and  
996 application profiles. The functional blocks are depicted in Figure 2.



998

**Figure 2 – Functional block diagram**

- 999 – *L2 connectivity*: Provides the functionalities required for establishing physical and data link  
1000 layer connections (e.g., Wi-Fi™ or Bluetooth® connection) to the network.
- 1001 – *Networking*: Provides functionalities required for Devices to exchange data among themselves  
1002 over the network (e.g., Internet).
- 1003 – *Transport*: Provides end-to-end flow transport with specific QoS constraints. Examples of a  
1004 transport protocol include TCP and UDP or new Transport protocols under development in the  
1005 IETF, e.g., Delay Tolerant Networking (DTN).
- 1006 – *Framework*: Provides the core functionalities as defined in this document. The functional block  
1007 is the source of requests and responses that are the content of the communication between  
1008 two Devices.
- 1009 – *Vertical Domain profile*: Provides market segment specific functionalities, e.g., functions for the  
1010 smart home market segment.
- 1011 When two Devices communicate with each other, each functional block in a Device interacts with  
1012 its counterpart in the peer Device as shown in Figure 3.



1013

1014

**Figure 3 – Communication layering model**

1015 **5.4 Framework**

1016 Framework consists of functions which provide core functionalities for operation.

- 1017 – *Identification and addressing.* Defines the identifier and addressing capability. The Identification and addressing function is defined in clause 6.
- 1018
- 1019 – *Discovery.* Defines the process for discovering available.
  - 1020 – Devices (OCF Endpoint Discovery in clause 10) and
  - 1021 – Resources (Resource discovery in 11.3).
- 1022 – *Resource model.* Specifies the capability for representation of entities in terms of Resources and defines mechanisms for manipulating the Resources. The Resource model function is defined in clause 7.
- 1023
- 1024
- 1025 – *CRUDN.* Provides a generic scheme for the interactions between a Client and Server as defined in clause 8.
- 1026
- 1027 – *Messaging.* Provides specific message protocols for RESTful operation, i.e. CRUDN. For example, CoAP is a primary messaging protocol. The messaging function is defined in 12.
- 1028
- 1029 – *Device management.* Specifies the discipline of managing the capabilities of a Device, and includes Device provisioning and initial setup as well as Device monitoring and diagnostics. The Device management function is defined in 11.5.
- 1030
- 1031
- 1032 – *Security.* Includes authentication, authorization, and access control mechanisms required for secure access to Entities. The security function is defined in clause 13.
- 1033

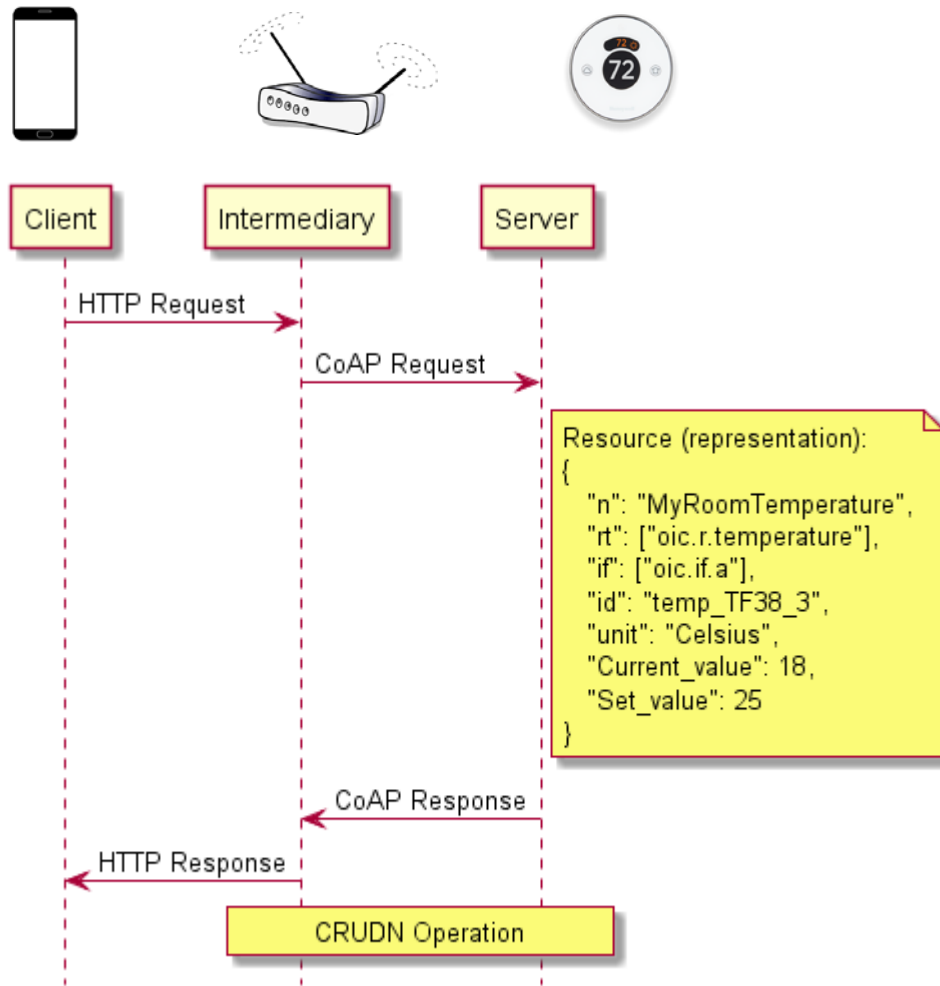
1034 **5.5 Example Scenario with roles**

1035 Interactions are defined between logical entities known as roles. Three roles are defined: Client, Server and Intermediary.

1037 Figure 4 illustrates an example of the roles in a scenario where a smart phone sends a request message to a thermostat; the original request is sent over HTTP, but is translated into a CoAP request message by a gateway in between, and then delivered to the thermostat. In this example, the smart phone takes the role of a Client, the gateway takes the role of an Intermediary and the thermostat takes the role of a Server.

1041

1042



1043

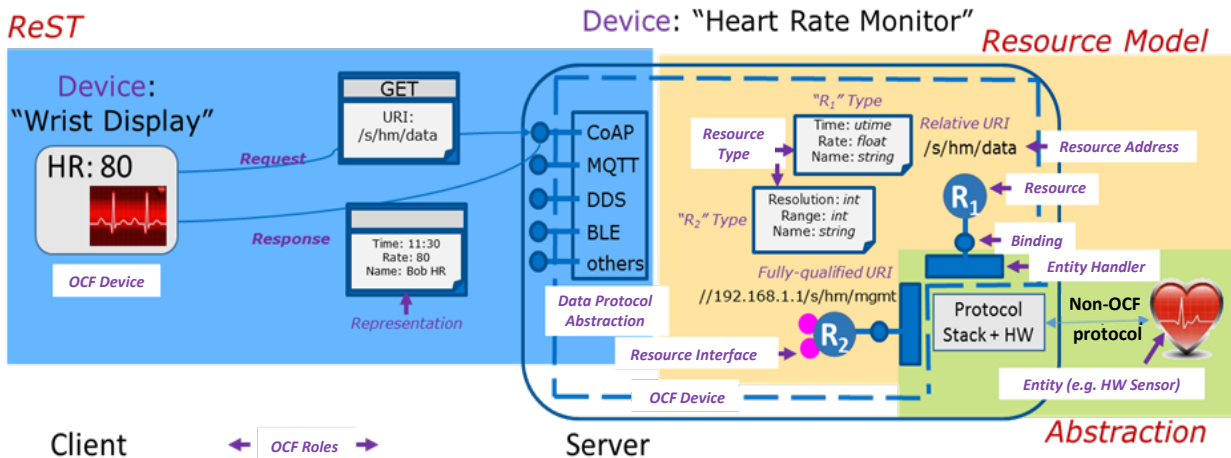
1044

**Figure 4 – Example illustrating the roles**

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

1046 The use case for this scenario is a display (like a wrist watch) that is used to monitor a heart rate  
1047 sensor that implements a protocol that is not OCF supported.

1048 Figure 5 provides a detailed logical view of the concepts described in Figure 1.



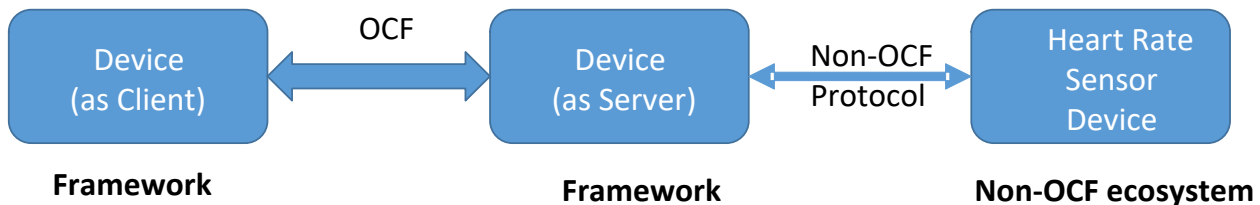
1049

1050

**Figure 5 – Framework - Architecture Detail**

1051  
1052

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.



1053

1054

**Figure 6 – Server bridging to Non- OCF device**

1055  
1056  
1057  
1058

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.

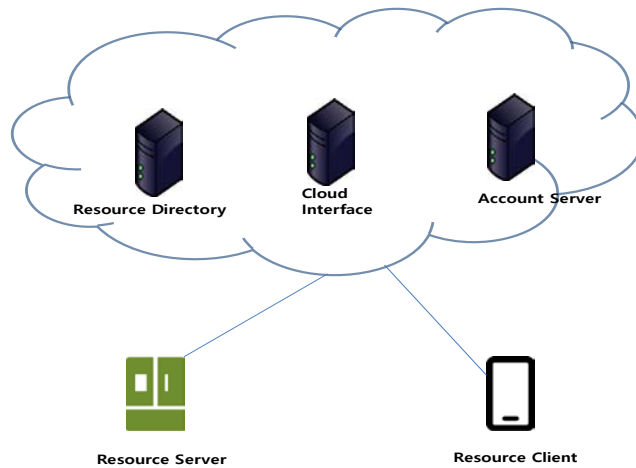
1059  
1060  
1061  
1062  
1063  
1064

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.

## 1065 5.7 OCF Cloud architecture

1066 This clause describes the architecture of OCF Cloud in Figure 7: and Figure 8





1067

1068

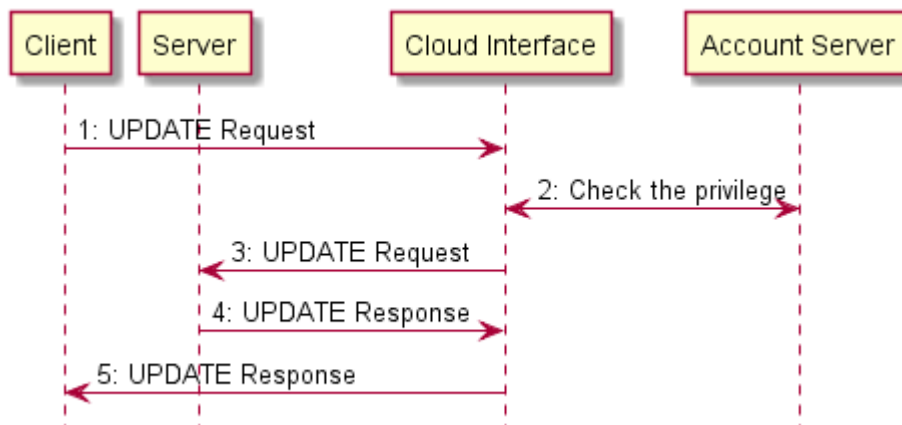
**Figure 7 – OCF Cloud deployment architecture**

1069 The Cloud architecture comprises of following three network entities:

- 1070 – *Cloud Interface Server* – A logical entity to which an OCF Device primarily. It encapsulates  
1071 *Account Server* and *Resource Directory* features. The Cloud Interface routes the packet  
1072 between OCF Devices based on the request URI in the packet header. The Client needs to  
1073 keep the persistent connection alive to the Server.
- 1074 – *Account Server* – A logical entity that handles Device registration, Auth Token validation and  
1075 handles sign-in and token-refresh requests from the Device.
- 1076 – *Resource Directory* – A logical entity holding Resource information published by Servers. A  
1077 Client when looking for a Resource receives a response from the Resource Directory on behalf  
1078 of the Server. Then with information included in the response form the Resource Directory, the  
1079 Client directly connects to the Server.

1080 When a Client try to access a Server, the Client connects to Cloud Interface Server then Cloud  
1081 Interface routes the received message to the indicated Server after checking the privilege.

1082



1083

1084

**Figure 8 – OCF Endpoint routing**

## 1085 **6 Identification and addressing**

### 1086 **6.1 Introduction**

1087 Facilitating proper and efficient interactions between elements in the Framework, requires a means  
1088 to identify, name and address these elements.

1089 The *identifier* unambiguously identifies an element in a context or domain. The context or domain  
1090 may be determined by the use or the application. The identifier is expected to be immutable over  
1091 the lifecycle of that element and is unambiguous within a context or domain.

1092 The *address* is used to define a place, way or means of reaching or accessing the element in order  
1093 to interact with it. An address may be mutable based on the context.

1094 The *name* is a handle that distinguishes the element from other elements in the Framework. The  
1095 name may be changed over the lifecycle of that element.

1096 There may be methods or resolution schemes that allow determining any of these based on the  
1097 knowledge of one or more of others (e.g., determine name from address or address from name).

1098 Each of these aspects may be defined separately for multiple contexts (e.g., a context could be a  
1099 layer in a stack). So an address may be a URL for addressing Resource and an IP address for  
1100 addressing at the connectivity layer. In some situations, both these addresses would be required.  
1101 For example, to do RETRIEVE (see 8.3) operation on a particular Resource representation, the  
1102 Client needs to know the address of the target Resource and the address of the Server through  
1103 which the Resource is exposed.

1104 In a context or domain of use, a name or address could be used as identifier or vice versa. For  
1105 example, a URL could be used as an identifier for a Resource and designated as a URI.

1106 The remainder of this clause discusses the identifier, address and naming from the point of view  
1107 of the Resource model and the interactions to be supported by the Resource model. Examples of  
1108 interactions are the RESTful interactions, i.e. CRUDN operation (clause 8) on a Resource. Also  
1109 the mapping of these to transport protocols, e.g., CoAP is described.

### 1110 **6.2 Identification**

#### 1111 **6.2.1 Overview**

1112 An identifier is unambiguous within the context or domain of use. There are many schemes that  
1113 may be used to generate an identifier that has the required properties. The identifier may be  
1114 context-specific in that the identifier is expected to be and guaranteed to be unambiguous only  
1115 within that context or domain. Identifier may also be context- independent where these identifiers  
1116 are guaranteed to be unambiguous across all contexts and domains both spatially and temporally.  
1117 The context-specific identifiers could be defined by simple schemes like monotonic enumeration or  
1118 may be defined by overloading an address or name, for example an IP address may be an identifier  
1119 within the private domain behind a gateway in a smart home. On the other hand, context-  
1120 independent identifiers require a stronger scheme that derives universally unique identities, for  
1121 example any one of the versions of Universally Unique Identifiers (UUIDs). Context independent  
1122 identifier may also be generated using hierarchy of domains where the root of the hierarchy is  
1123 identified with a UUID and sub-domains may generate context independent identifier by  
1124 concatenating context-specific identifiers for that domain to the context-independent identifier of  
1125 their parent.

#### 1126 **6.2.2 Resource identification and addressing**

1127 A Resource may be identified using a URI and addressed by the same URI if the URI is a URL. In  
1128 some cases a Resource may need an identifier that is different from a URI; in this case, the

1129 Resource may have a Property whose value is the identifier. When the URI is in the form of a URL,  
1130 then the URI may be used to address the Resource.

1131 An OCF URI is based on the general form of a URI as defined in IETF RFC 3986 as follows:

1132 `<scheme>://<authority>/<path>?<query>`

1133 Specifically the OCF URI is specified in the following form:

1134 `ocf://<authority>/<path>?<query>`

1135 The following is a description of values that each component takes.

1136 The *scheme* for the URI is "ocf". The "ocf" scheme represents the semantics, definitions and use  
1137 as defined in this document. If a URI has the portion preceding the "/" (double slash) omitted, then  
1138 the "ocf" scheme shall be assumed.

1139 Each transport binding is responsible for specifying how an OCF URI is converted to a transport  
1140 protocol URI before sending over the network by the requestor. Similarly on the receiver side, each  
1141 transport binding is responsible for specifying how an OCF URI is converted from a transport  
1142 protocol URI before handing over to the Resource model layer on the receiver.

1143 The authority of an OCF URI shall be the Device ID ("di") value, as defined in [OCF Security], of  
1144 the Server.

1145 The *path* is a string that unambiguously identifies or references a Resource within the context of  
1146 the Server. In this version of the document, a path shall not include pct-encoded non-ASCII  
1147 characters or NUL characters. A *path* shall be preceded by a "/" (slash). The *path* may have "/"  
1148 (slash) separated segments for human readability reasons. In the OCF context, the "/" (slash)  
1149 separated segments are treated as a single string that directly references the Resources (i.e. a flat  
1150 structure) and not parsed as a hierarchy. On the Server, the path or some substring in the path  
1151 may be shortened by using hashing or some other scheme provided the resulting reference is  
1152 unique within the context of the host.

1153 Once a path is generated, a Client accessing the Resource or recipient of the URI should use that  
1154 path as an opaque string and should not parse to infer a structure, organization or semantic.

1155 A query string shall contain a list of "<name>=<value>" segments (aka name-value pair) each  
1156 separated by a "&" (ampersand). The query string will be mapped to the appropriate syntax of the  
1157 protocol used for messaging. (e.g., CoAP).

1158 A URI may be either fully qualified or relative generation of URI.

1159 A URI may be defined by the Client which is the creator of that Resource. Such a URI may be  
1160 relative or absolute (fully qualified). A relative URI shall be relative to the Device on which it is  
1161 hosted. Alternatively, a URI may be generated by the Server of that Resource automatically based  
1162 on a pre-defined convention or organization of the Resources, based on an OCF Interface, based  
1163 on some rules or with respect to different roots or bases.

1164 The absolute path reference of a URI is to be treated as an opaque string and a Client should not  
1165 infer any explicit or implied structure in the URI – the URI is simply an address. It is also  
1166 recommended that Devices hosting a Resource treat the URI of each Resource as an opaque string  
1167 that addresses only that Resource. (e.g., URI's "/a" and "/a/b" are considered as distinct addresses  
1168 and Resource b cannot be construed as a child of Resource a).

### 1169 **6.3 Namespace:**

1170 The relative URI prefix "/oic/" is reserved as a namespace for URIs defined in OCF specifications  
1171 and shall not be used for URIs that are not defined in OCF specifications.

## 1172 **6.4 Network addressing**

1173 The following are the addresses used in this document:

1174 IP address

- 1175 – An IP address is used when the Device is using an IP configured interface.
- 1176 – When a Device only has the identity information of its peer, a resolution mechanism is needed  
1177 to map the identifier to the corresponding address.

## 1178 **7 Resource model**

### 1179 **7.1 Introduction**

1180 The Resource model defines concepts and mechanisms that provide consistency and core  
1181 interoperability between Devices in the OCF ecosystems. The Resource model concepts and  
1182 mechanisms are then mapped to the transport protocols to enable communication between the  
1183 Devices – each transport provides the communication protocol interoperability. The Resource  
1184 model, therefore, allows for interoperability to be defined independent of the transports.

1185 In addition, the concepts in the Resource model support modelling of the primary artefacts and  
1186 their relationships to one and another and capture the semantic information required for  
1187 interoperability in a context. In this way, OCF goes beyond simple protocol interoperability to  
1188 capture the rich semantics required for true interoperability in Wearable and Internet of Things  
1189 ecosystems.

1190 The primary concepts in the Resource model are: entity, Resources, Uniform Resource Identifiers  
1191 (URI), Resource Types, Properties, Representations, OCF Interfaces, Collections and Links. In  
1192 addition, the general mechanisms are CREATE, RETRIEVE, UPDATE, DELETE and NOTIFY.  
1193 These concepts and mechanisms may be composed in various ways to define the rich semantics  
1194 and interoperability needed for a diverse set of use cases that the Framework is applied to.

1195 In the OCF Resource model Framework, an entity needs to be visible, interacted with or  
1196 manipulated, it is represented by an abstraction called a Resource. A Resource encapsulates and  
1197 represents the state of an entity. A Resource is identified, addressed and named using URIs.

1198 Properties are "key=value" pairs and represent state of the Resource. A snapshot of these  
1199 Properties is the Representation of the Resource. A specific view of the Representation and the  
1200 mechanisms applicable in that view are specified as OCF Interfaces. Interactions with a Resource  
1201 are done as Requests and Responses containing Representations.

1202 A Resource instance is derived from a Resource Type. The uni-directional relationship between  
1203 one Resource and another Resource is defined as a Link. A Resource that has Properties and  
1204 Links is a Collection.

1205 A set of Properties can be used to define a state of a Resource. This state may be retrieved or  
1206 updated using appropriate Representations respectively in the response from and request to that  
1207 Resource.

1208 A Resource (and Resource Type) could represent and be used to expose a capability. Interactions  
1209 with that Resource can be used to exercise or use that capability. Such capabilities can be used to  
1210 define processes like discovery, management, advertisement etc. For example: *discovery of*  
1211 *Resources on a Device* can be defined as the retrieval of a representation of a specific Resource  
1212 where a Property or Properties have values that describe or reference the Resources on the Device.

1213 The information for Request or Response with the Representation may be communicated on the  
1214 wire by serializing using a transfer protocol or encapsulated in the payload of the transport protocol

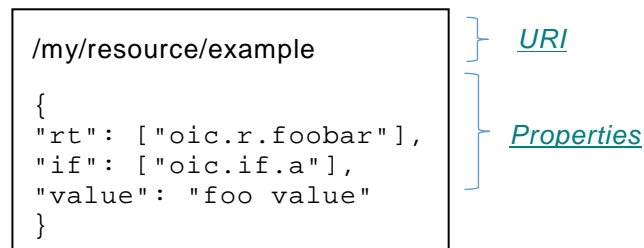
1215 – the specific method is determined by the normative mapping of the Request or Response to the  
1216 transport protocol. See 11.8 for transport protocols supported.

1217 The OpenAPI 2.0 definitions (Annex D) used in this document are normative. This includes that all  
1218 defined JSON payloads shall comply with the indicated OpeAPI 2.0 definitions. Annex D contains  
1219 all of the OpenAPI 2.0 definitions for Resource Types defined in this document.

## 1220 7.2 Resource

1221 A Resource shall be defined by one or more Resource Type(s) – see Annex D for Resource Type.  
1222 A request to CREATE a Resource shall specify one or more Resource Types that define that  
1223 Resource.

1224 A Resource is hosted in a Device. A Resource shall have a URI as defined in clause 6. The URI  
1225 may be assigned by the Authority at the creation of the Resource or may be pre-defined by the  
1226 specification of the Resource Type. An example Resource representation is depicted in Figure 9.



1227  
1228 **Figure 9 – Example Resource**

1229 Core Resources are the Resources defined in this document to enable functional interactions as  
1230 defined in clause 10 (e.g., Discovery, Device management, etc). Among the Core Resources,  
1231 "/oic/res", "/oic/p", and "/oic/d" shall be supported on all Devices. Devices may support other Core  
1232 Resources depending on the functional interactions they support.

## 1233 7.3 Property

### 1234 7.3.1 Introduction

1235 A Property describes an aspect that is exposed through a Resource including meta-information  
1236 related to that Resource.

1237 A Property shall have a name i.e. Property Name and a value i.e. Property Value. The Property is  
1238 expressed as a key-value pair where key is the Property Name and value the Property Value like  
1239 <Property Name> = <Property Value>. For example if the "temperature" Property has a Property  
1240 Name "temp" and a Property Value "30F", then the Property is expressed as "temp=30F". The  
1241 specific format of the Property depends on the encoding scheme. For example, in JSON, Property  
1242 is represented as "key": value (e.g., "temp": 30).

1243 In addition, the Property definition shall have a

1244 – *Value Type* – the Value Type defines the values that a Property Value may take. The Value  
1245 Type may be a simple data type (e.g. string, Boolean) as defined in 4.3 or may be a complex  
1246 data type defined with a schema. The Value Type may define

1247 – *Value Rules* define the rules for the set of values that the Property Value may take. Such  
1248 rules may define the range of values, the min-max, formulas, the set of enumerated values,  
1249 patterns, conditional values, and even dependencies on values of other Properties. The  
1250 rules may be used to validate the specific values in a Property Value and flag errors.

1251 – *Mandatory* – specifies if the Property is mandatory or not for a given Resource Type.

1252 – *Access modes* – specifies whether the Property may be read, written or both. Updates are  
1253 equivalent to a write. "r" is used for read and "w" is used for write – both may be specified.  
1254 Write does not automatically imply read.

1255 The definition of a Property may include the following additional information – these items are  
1256 informative:

1257 – *Property Title* - a human-friendly name to designate the Property; usually not sent over the wire.

1258 – *Description* – descriptive text defining the purpose and expected use of this Property.

1259 In general, a Property is meaningful only within the Resource to which it is associated. However a  
1260 base set of Properties that may be supported by all Resources, known as Common Properties,  
1261 keep their semantics intact across Resources i.e. their "key=value" pair means the same in any  
1262 Resource. Detailed tables for all Common Properties are defined in 7.3.2.

## 1263 **7.3.2 Common Properties**

### 1264 **7.3.2.1 Introduction**

1265 The Common Properties defined in this clause may be specified for all Resources. The following  
1266 Properties are defined as Common Properties: Resource Type, Resource Interface, Name, and  
1267 Resource Identity.

1268 The name of a Common Property shall be unique and shall not be used by other Properties. When  
1269 defining a new Resource Type, its non-common Properties shall not use the name of existing  
1270 Common Properties (e.g., "rt", "if", "n", "id"). When defining a new "Common Property", it should  
1271 be ensured that its name has not been used by any other Properties. The uniqueness of a new  
1272 Common Property name can be verified by checking all the Properties of all the existing OCF  
1273 defined Resource Types. However, this may become cumbersome as the number of Resource  
1274 Types grow. To prevent such name conflicts in the future, OCF may reserve a certain name space  
1275 for Common Property. Potential approaches are (1) a specific prefix (e.g. "oic") may be designated  
1276 and the name preceded by the prefix (e.g. "oic.psize") is only for Common Property; (2) the names  
1277 consisting of one or two letters are reserved for Common Property and all other Properties shall  
1278 have the name with the length larger than the 2 letters; (3) Common Properties may be nested  
1279 under specific object to distinguish themselves.

1280 The ability to UPDATE a Common Property (that supports write as an access mode) is restricted  
1281 to the "oic.if.rw" (read-write) OCF Interface; thus a Common Property shall be updatable using the  
1282 read-write OCF Interface if and only if the Property supports write access as defined by the Property  
1283 definition and the associated schema for the read-write OCF Interface.

1284 The following Common Properties for all Resources are specified in 7.3.2.2 through 7.3.2.6 and  
1285 summarized as follows:

1286 – *Resource Type* ("rt") – this Property is used to declare the Resource Type of that Resource.  
1287 Since a Resource could be define by more than one Resource Type the Property Value of the  
1288 Resource Type Property can be used to declare more than one Resource type. For example:  
1289 "rt": ["oic.wk.d", "oic.d.airconditioner"] declares that the Resource containing this Property is  
1290 defined by either the "oic.wk.d" Resource Type or the "oic.d.airconditioner" Resource Type.  
1291 See 7.3.2.3 for details.

1292 – *OCF Interface* ("if") – this Property declares the OCF Interfaces supported by the Resource.  
1293 The Property Value of the OCF Interface Property can be multi-valued and lists all the OCF  
1294 Interfaces supported. See 7.3.2.4 for details.

1295 – *Name* ("n") – the Property declares human-readable name assigned to the Resource. See  
1296 7.3.2.5.

1297 – *Resource Identity* ("id"): its Property Value shall be a unique (across the scope of the host  
 1298 Server) instance identifier for a specific instance of the Resource. The encoding of this identifier  
 1299 is Device and implementation dependent. See 7.3.2.6 for details.

1300 **7.3.2.2 Property Name and Property Value definitions**

1301 The Property Name and Property Value as used in this document:

- 1302 – *Property Name*– the key in "key=value" pair. Property Name is case sensitive and its data type  
 1303 is "string". Property names shall contain only letters A to Z, a to z, digits 0 to 9, hyphen, and  
 1304 dot, and shall not begin with a digit.
- 1305 – *Property Value* – the value in "key=value" pair. Property Value is case sensitive when its data  
 1306 type is "string".

1307 **7.3.2.3 Resource Type**

1308 Resource Type Property is specified in 7.4.

1309 **7.3.2.4 OCF Interface**

1310 OCF Interface Property is specified in 7.6.

1311 **7.3.2.5 Name**

1312 A human friendly name for the Resource, i.e. a specific resource instance name (e.g.,  
 1313 MyLivingRoomLight), The Name Property is as defined in Table 2

1314 **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.

1315  
 1316 The Name Property is read-write unless otherwise restricted by the Resource Type (i.e. the  
 1317 Resource Type does not support UPDATE or does not support UPDATE using read-write).

1318 **7.3.2.6 Resource Identity**

1319 The Resource Identity Property shall be a unique (across the scope of the host Server) instance  
 1320 identifier for a specific instance of the Resource. The encoding of this identifier is Device and  
 1321 implementation dependent as long as the uniqueness constraint is met, noting that an  
 1322 implementation may use a uuid as defined in 4.3. The Resource Identity Property is as defined in  
 1323 Table 3.

1324 **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)

1325

1326 **7.4 Resource Type**

1327 **7.4.1 Introduction**

1328 Resource Type is a class or category of Resources and a Resource is an instance of one or more  
1329 Resource Types.

1330 The Resource Types of a Resource is declared using the Resource Type Common Property as  
1331 described in 7.3.2.3 or in a Link using the Resource Type Parameter.

1332 A Resource Type may either be pre-defined by OCF or in custom definitions by manufacturers, end  
1333 users, or developers of Devices (vendor-defined Resource Types). Resource Types and their  
1334 definition details may be communicated out of band (i.e. in documentation) or be defined explicitly  
1335 using a meta-language which may be downloaded and used by APIs or applications. OCF has  
1336 adopted OpenAPI 2.0 as the specification method for OCF's RESTful interfaces and Resource  
1337 definitions.

1338 Every Resource Type shall be identified with a Resource Type ID which shall be represented using  
1339 the requirements and ABNF governing the Resource Type attribute in IETF RFC 6690 (clause 2 for  
1340 ABNF and clause 3.1 for requirements) with the caveat that segments are separated by a "."  
1341 (period). The entire string represents the Resource Type ID. When defining the ID each segment  
1342 may represent any semantics that are appropriate to the Resource Type. For example, each  
1343 segment could represent a namespace. Once the ID has been defined, the ID should be used  
1344 opaquely and implementations should not infer any information from the individual segments. The  
1345 string "oic", when used as the first segment in the definition of the Resource Type ID, is reserved  
1346 for OCF-defined Resource Types. All OCF defined Resource Types are to be registered with the  
1347 IANA Core Parameters registry as described also in IETF RFC 6690.

1348 **7.4.2 Resource Type Property**

1349 A Resource when instantiated or created shall have one or more Resource Types that are the  
1350 template for that Resource. The Resource Types that the Resource conforms to shall be declared  
1351 using the "rt" Common Property for the Resource as defined in Table 4. The Property Value for the  
1352 "rt" Common Property shall be the list of Resource Type IDs for the Resource Types used as  
1353 templates (i.e., "rt"=<list of Resource Type IDs>).

1354 **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

1355  
1356 Resource Types may be explicitly discovered or implicitly shared between the user (i.e. Client) and  
1357 the host (i.e. Server) of the Resource.

1358 **7.4.3 Resource Type definition**

1359 Resource Type is specified as follows:

- 1360 – *Pre-defined URI* (optional) – a pre-defined URI may be specified for a specific Resource Type  
1361 in an OCF specification. When a Resource Type has a pre-defined URI, all instances of that  
1362 Resource Type shall use only the pre-defined URI. An instance of a different Resource Type  
1363 shall not use the pre-defined URI.
- 1364 – *Resource Type Title* (optional) – a human friendly name to designate the Resource Type.



- 1365 – *Resource Type ID* – the value of "rt" Property which identifies the Resource Type, (e.g.,  
1366 "oic.wk.p").
- 1367 – *Resource Interfaces* – list of the OCF Interfaces that may be supported by the Resource Type.
- 1368 – *Properties* – definition of all the Properties that apply to the Resource Type. The Resource Type  
1369 definition shall define whether a property is mandatory, conditional mandatory, or optional.
- 1370 – *Related Resource Types* (optional) – the specification of other Resource Types that may be  
1371 referenced as part of the Resource Type, applicable to Collections.
- 1372 – *Mime Types* (optional) – mime types supported by the Resource including serializations (e.g.,  
1373 application/cbor, application/json, application/xml).

1374 Table 5 and Table 6 provides an example description of an illustrative foobar Resource Type and  
1375 its associated Properties.

1376 **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

1377

1378 **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

1379

1380 For example, an instance of the foobar Resource Type.

```
1381 {
1382   "rt": ["oic.r.foobar"],
1383   "if": ["oic.if.a"],
1384   "value": "foo value"
1385 }
```

1386

1387 For example, a schema representation for the foobar Resource Type.

```
1388 {
1389   "$schema": "http://json-schema.org/draft-04/schema",
1390   "type": "object",
1391   "properties": {
1392     "rt": {
1393       "type": "array",
1394       "items": {
1395         "type": "string",
1396         "maxLength": 64
1397       },

```

```

1398     "minItems" : 1,
1399     "readOnly": true,
1400     "description": "Resource Type of the Resource"
1401   },
1402   "if": {
1403     "type": "array",
1404     "items": {
1405       "type" : "string",
1406       "enum" : ["oic.if.baseline", "oic.if.ll", "oic.if.b", "oic.if.lb", "oic.if.rw",
1407 "oic.if.r", "oic.if.a", "oic.if.s"]
1408     },
1409     "value": {"type": "string"}
1410   },
1411   "required": ["rt", "if", "value"]
1412 }

```

#### 1413 7.4.4 Multi-value "rt" Resource

1414 Multi-value "rt" Resource means a Resource with multiple Resource Types where none of the  
1415 included Resource Types denote a well-known Resource Type (i.e. "oic.wk.<thing>"). Such a  
1416 Resource is associated with multiple Resource Types and so has an "rt" Property Value of multiple  
1417 Resource Type IDs (e.g. "rt": ["oic.r.switch.binary", "oic.r.light.brightness"]). The order of the  
1418 Resource Type IDs in the "rt" Property Value is meaningless. For example, "rt":  
1419 ["oic.r.switch.binary", "oic.r.light.brightness"] and "rt": ["oic.r.light.brightness", "oic.r.switch.binary"]  
1420 have the same meaning.

1421 Resource Types for multi-value "rt" Resources shall satisfy the following conditions:

- 1422 – Property Name – Property Names for each Resource Type shall be unique (within the scope of  
1423 the multi-value "rt" Resource) with the exception of Common Properties, otherwise there will be  
1424 conflicting Property semantics. If two Resource Types have a Property with the same Property  
1425 "Name, a multi-value "rt" Resource shall not be composed of these Resource Types.

1426 A multi-value "rt" Resource satisfies all the requirements for each Resource Type and conforms to  
1427 the OpenAPI 2.0 definitions for each component Resource Type. Thus the mandatory Properties  
1428 of a multi-value "rt" Resource shall be the union of all the mandatory Properties of each Resource  
1429 Type. For example, mandatory Properties of a Resource with "rt": ["oic.r.switch.binary",  
1430 "oic.r.light.brightness"] are "value" and "brightness", where the former is mandatory for  
1431 "oic.r.switch.binary" and the latter for "oic.r.light.brightness".

1432 The multi-value "rt" Resource Interface set shall be the union of the sets of OCF Interfaces from  
1433 the component Resource Types. The Resource Representation in response to a CRUDN action on  
1434 an OCF Interface shall be the union of the schemas that are defined for that OCF Interface. The  
1435 Default OCF Interface for a multi-value "rt" Resource shall be the baseline OCF Interface  
1436 ("oic.if.baseline") as that is the only guaranteed common OCF Interface between the Resource  
1437 Types.

1438 For clarity if each Resource Type supports the same set of OCF Interfaces, then the resultant multi-  
1439 value "rt" Resource has that same set of OCF Interfaces with a Default OCF Interface of baseline  
1440 ("oic.if.baseline").

1441 See 7.10.3 for the handling of query parameters as applied to a multi-value "rt" Resource.

#### 1442 7.5 Device Type

1443 A Device Type is a class of Device. Each Device Type defined will include a list of minimum  
1444 Resource Types that a Device shall implement for that Device Type. A Device may expose  
1445 additional standard and vendor defined Resource Types beyond the minimum list. The Device Type  
1446 is used in Resource discovery as specified in 11.3.4.

1447 Like a Resource Type, a Device Type can be used in the Resource Type Common Property or in a  
1448 Link using the Resource Type Parameter.

1449 A Device Type may either be pre-defined (ISO/IEC 30118-5:2018) or in custom definitions by  
1450 manufacturers, end users, or developers of Devices (vendor-defined Device Types). Device Types  
1451 and their definition details may be communicated out of band (like in documentation).

1452 Every Device Type shall be identified with a Resource Type ID using the same syntax constraints  
1453 as a Resource Type.

## 1454 7.6 OCF Interface

### 1455 7.6.1 Introduction

1456 An OCF Interface provides first a view into the Resource and then defines the requests and  
1457 responses permissible on that view of the Resource. So this view provided by an OCF Interface  
1458 defines the context for requests and responses on a Resource. Therefore, the same request to a  
1459 Resource when targeted to different OCF Interfaces may result in different responses.

1460 An OCF Interface may be defined by either this document (a Core OCF Interface), ISO/IEC 30118-  
1461 5:2018 (a vertical OCF Interface) or manufacturers, end users or developers of Devices (a vendor-  
1462 defined OCF Interface).

1463 The OCF Interface Property lists all the OCF Interfaces the Resource support. All Resources shall  
1464 have at least one OCF Interface. The Default OCF Interface shall be defined by an OCF  
1465 specification and inherited from the Resource Type definition. The Default OCF Interface  
1466 associated with all OCF-defined Resource Types shall be the supported OCF Interface listed first  
1467 within the *applicable enumeration* in the definition of the Resource Type (see Annex D for the OCF-  
1468 defined Resource Types defined in this document). The *applicable enumeration* is in the  
1469 "parameters" enumeration referenced from the first "get" method in the first "path" in the OpenAPI  
1470 2.0 file ("post" method if no "get" exists) for the Resource Type. All Default OCF Interfaces specified  
1471 in an OCF specification shall be mandatory.

1472 In addition to any OCF specification defined OCF Interface, all Resources shall support the  
1473 baseline OCF Interface ("oic.if.baseline") as defined in 7.6.3.2.

1474 See 7.10.4 for the use of queries to enable selection of a specific OCF Interface in a request.

1475 An OCF Interface may accept more than one media type. An OCF Interface may respond with more  
1476 than one media type. The accepted media types may be different from the response media types.  
1477 The media types are specified with the appropriate header parameters in the transfer protocol.  
1478 (NOTE: This feature has to be used judiciously and is allowed to optimize representations on the  
1479 wire) Each OCF Interface shall have at least one media type.

1480

### 1481 7.6.2 OCF Interface Property

1482 **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.

1483

1484 The OCF Interfaces supported by a Resource shall be declared using the OCF Interface Common  
 1485 Property (Table 7), e.g., ""if": ["oic.if.ll", "oic.if.baseline"]". The Property Value of an OCF Interface  
 1486 Property shall be a lower case string with segments separated by a "." (dot). The string "oic", when  
 1487 used as the first segment in the OCF Interface Property Value, is reserved for OCF-defined OCF  
 1488 Interfaces. The OCF Interface Property Value may also be a reference to an authority similar to  
 1489 IANA that may be used to find the definition of an OCF Interface. A Resource Type shall support  
 1490 one or more of the OCF Interfaces defined in 7.6.3.

1491 **7.6.3 OCF Interface methods**

1492 **7.6.3.1 Overview**

1493 OCF Interface methods shall not violate the defined OpenAPI 2.0 definitions for the Resources as  
 1494 defined in Annex D.

1495 The defined OCF Interfaces are listed in Table 8:

1496 **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.

1497

1498 **7.6.3.2 Baseline OCF Interface**

1499 **7.6.3.2.1 Overview**

1500 The Representation that is visible using the baseline OCF Interface includes all the Properties of  
 1501 the Resource including the Common Properties. The baseline OCF Interface shall be defined for  
 1502 all Resource Types. All Resources shall support the baseline OCF Interface.

### 1503 **7.6.3.2.2 Use of RETRIEVE**

1504 The baseline OCF Interface is used when a Client wants to retrieve all Properties of a Resource;  
1505 that is the Server shall respond with a Resource representation that includes all of the implemented  
1506 Properties of the Resource. When the Server is unable to send back the whole Resource  
1507 representation, it shall reply with an error message. The Server shall not return a partial Resource  
1508 representation.

1509 An example response to a RETRIEVE request using the baseline OCF Interface:

```
1510 {  
1511   "rt": ["oic.r.temperature"],  
1512   "if": ["oic.if.a", "oic.if.baseline"],  
1513   "temperature": 20,  
1514   "units": "C",  
1515   "range": [0,100]  
1516 }
```

### 1517 **7.6.3.2.3 Use of UPDATE**

1518 Using the baseline OCF Interface, all Properties of a Resource with the exception of Common  
1519 Properties may be modified using an UPDATE request with a list of Properties and their desired  
1520 values if a Resource Type has an associated schema for UPDATE using baseline. If the OCF  
1521 Interfaces exposed by a Resource in addition to the baseline OCF Interface do not support the  
1522 UPDATE semantic then UPDATE using the baseline OCF Interface is also not supported.

## 1523 **7.6.3.3 Links List OCF Interface**

### 1524 **7.6.3.3.1 Overview**

1525 The links list OCF Interface provides a view into the list of Links in a Resource. The Representation  
1526 visible through this OCF Interface has only the Links exposed as Property(-ies) that is(are) an array  
1527 (or arrays) of Links by the Resource – so this OCF Interface is used to manipulate or interact with  
1528 the list of Links. The Links list may be RETRIEVED using this OCF Interface.

1529 The links list OCF Interface is defined as follows:

- 1530 – The links list OCF Interface name is "oic.if.ll".
- 1531 – If there are no Links present in a Resource, then an empty list shall be returned in response to  
1532 a RETRIEVE request using the links list OCF Interface.
- 1533 – The Representation determined by this OCF Interface depends on the requesting Client. For a  
1534 Client that includes an OCF-Accept-Content-Format-Version option as defined in 12.2.5 in the  
1535 request the response only includes the Property value(s) of the Property(-ies) that are arrays  
1536 of Links, hence a Collection or "/oic/res" response with oic.if.ll is an array of Links. For a Client  
1537 that does not include an OCF-Accept-Content-Format-Version option the response is as defined  
1538 in Annex E.
- 1539 – The array of Links may be observed by a Client using the links list OCF Interface (i.e. by  
1540 following the procedures in clause 11.4.2 with the addition of a query parameter of "?if=oic.if.ll").
- 1541 – Any CREATE, UPDATE, or DELETE operation on any Link in the array of Links shall result in  
1542 the complete Resource representation for the links list OCF Interface as defined for the target  
1543 Resource (i.e. the full array of Links) subject to any applied filtering being provided in the  
1544 notification that is sent to the Client that initiated the Observe request.
- 1545 – If the act of deleting a Link results in no Links being present then an empty list shall be sent in  
1546 a notification.

### 1547 **7.6.3.3.2 Example: links list OCF Interface**

1548 A request to a Collection, where the request is to RETRIEVE the Links in room (the Links could be  
1549 referencing lights, fans, electric sockets etc).

```

1550 GET ocf://<devID>/a/room/1?if=oic.if.ll
1551 The response would be the array of OCF Links
1552
1553 [
1554   {
1555     "href": "/the/light/1",
1556     "rt": ["oic.r.switch.binary"],
1557     "if": ["oic.if.a", "oic.if.baseline"],
1558     "eps": [
1559       {"ep": "coaps://[2001:db8:a::b1d4]:55555"}]
1560     },
1561     {
1562       "href": "/the/light/2",
1563       "rt": ["oic.r.switch.binary"],
1564       "if": ["oic.if.a", "oic.if.baseline"],
1565       "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:55555"}]
1566     },
1567     {
1568       "href": "/my/fan/1",
1569       "rt": ["oic.r.switch.binary"],
1570       "if": ["oic.if.a", "oic.if.baseline"],
1571       "eps": [
1572         {"ep": "coaps://[2001:db8:a::b1d4]:55555"}]
1573         {
1574           "href": "/his/fan/2",
1575           "rt": ["oic.r.switch.binary"],
1576           "if": ["oic.if.a", "oic.if.baseline"],
1577           "eps": [
1578             {"ep": "coaps://[2001:db8:a::b1d4]:55555"}]
1579           }
1580       ]

```

### 1581 7.6.3.4 Batch OCF Interface

#### 1582 7.6.3.4.1 Overview

1583 The batch OCF Interface is used to interact with a Collection of Resources using a single/same  
 1584 Request. The batch OCF Interface can be used to RETRIEVE or UPDATE the Properties of the  
 1585 linked Resources with a single request.

1586 The batch OCF Interface is defined as follows:

- 1587 – The batch OCF Interface name is "oic.if.b"
- 1588 – A Collection Resource has linked Resources that are represented as URIs. In the "href"  
 1589 Property of the batch payload the URI shall be fully qualified for remote Resources and a  
 1590 relative reference for local Resources.
- 1591 – The original request is modified to create new requests targeting each of the linked Resources  
 1592 in the Collection by substituting the URI in the original request with the URI of the linked  
 1593 Resource. The payload in the original request is replicated in the payload of the new requests.
- 1594 – The requests shall be forwarded assuming use of the Default OCF Interface of the linked  
 1595 Resources.
- 1596 – Requests shall only be forwarded to linked Resources that are identified by relation types "item"  
 1597 or "hosts" ("hosts" is the default relation type value should the "rel" Link Parameter not be  
 1598 present). Requests shall not be forwarded to linked Resources that do not contain the "item" or  
 1599 "hosts" relation type values.
- 1600 – Properties of the Collection Resource itself may be included in payloads using "oic.if.b" OCF  
 1601 Interface by exposing a single Link with the link relation "self" along with "item" within the  
 1602 Collection, and ensuring that Link resolution cannot become an infinite loop due to recursive  
 1603 references. For example, if the Default OCF Interface of the Collection is "oic.if.b", then the

- 1604 Server might recursively include its batch representation within its batch representation, in an  
1605 endless loop. See 7.6.3.4.2 for an example of use of a Link containing "rel": ["self", "item"] to  
1606 include Properties of the Collection Resource, along with linked Resources, in "oic.if.b"  
1607 payloads.
- 1608 – If the Default OCF Interface of a Collection Resource is exposed using the Link relation "self",  
1609 and the Default OCF Interface contains Properties that expose any Links, those Properties shall  
1610 not be included in a batch representation which includes the "self" Link.
  - 1611 – Any request forwarded to a linked Resource that is a Collection (including a "self" Link reference)  
1612 shall have the Default OCF Interface of the linked Collection Resource applied.
  - 1613 – All the responses from the linked Resources shall be aggregated into a single Response to the  
1614 Client. The Server may timeout the response to a time window, the Server may choose any  
1615 appropriate window based on conditions.
  - 1616 – If a linked Resource cannot process the request, an empty response, i.e. a JSON object with  
1617 no content ("{}") as the representation for the "rep" Property, or error response should the linked  
1618 Resource Type provide an error schema or diagnostic payload, shall be returned by the linked  
1619 Resource. These empty or error responses for all linked Resources that exhibit an error shall  
1620 be included in the aggregated response to the original Client request. See the example in  
1621 7.6.3.4.2.
  - 1622 – If any of the linked Resources returns an error response, the aggregated response sent to the  
1623 Client shall also indicate an error (e.g. 4.xx in CoAP). If all of the linked Resources return  
1624 successful responses, the aggregated response shall include the success response code.
  - 1625 – The aggregated response shall be an array of objects representing the responses from each  
1626 linked Resource. Each object in the response shall include at least two items: (1) the URI of  
1627 the linked Resource (fully qualified for remote Resources, or a relative reference for local  
1628 Resources) as "href": <URI> and (2) the individual response object or array of objects if the  
1629 linked Resource is itself a Collection using "rep" as the key, e.g. "rep": { <representation of  
1630 individual response> }.
  - 1631 – If the Collection Resource is marked as Observable, linked Resources referenced in the  
1632 Collection may be Observed using the batch OCF Interface. If the Collection Resource is not  
1633 marked as Observable then the Collection cannot be Observed and Observe requests to the  
1634 Collection shall be handled as defined for the case where request validation fails in clause  
1635 11.4.2.4. The Observe mechanism shall work as defined in 11.4.2 with the Observe request  
1636 forwarded to each of the linked Resources. All responses to the request shall be aggregated  
1637 into a single response to the Client using the same representations and status codes as for  
1638 RETRIEVE operations using the batch OCF Interface.
  - 1639 – Should any one of the Observable linked Resources fail to honour the Observe request the  
1640 response to the batch Observe request shall also indicate that the entire request was not  
1641 honoured using the mechanism described in 11.4.2.4.
  - 1642 – If any of the Observable Resources in a request to a Collection using the batch OCF Interface  
1643 replies with an error or Observe Cancel, the Observations of all other linked Resources shall  
1644 be cancelled and the error or Observe Cancel status shall be returned to the Observing Client.
- 1645 NOTE Behavior may be different for Links that do network requests vs. local Resources.
- 1646 – All notifications to the Client that initiated an Observe request using the batch OCF Interface  
1647 shall use the batch representation for the Collection. This is the aggregation of any individual  
1648 Observe notifications received by the Device hosting the Collection from the individual Observe  
1649 requests that were forwarded to the linked Resources.
  - 1650 – Linked Resources which are not marked Observable in the Links of a Collection shall not trigger  
1651 Notifications, but may be included in the response to, and subsequent Notifications resulting  
1652 from, an Observe request to the batch OCF Interface of a Collection.

- 1653 – Each notification shall contain the most current values for all of the Linked Resources that would  
1654 be included if the original Observe request were processed again. The Server hosting the  
1655 Collection may choose to RETRIEVE all of the linked Resources each time, or may choose to  
1656 employ caching to avoid retrieving linked Resources on each Notification.
- 1657 – If a Linked Resource is Observable and has responded with a successful Observe response,  
1658 the most recently reported value of that Resource is considered to be the most current value  
1659 and may be reported in all subsequent Notifications.
- 1660 – Links in the Collection should be Observed by using the "oic.if.ll" OCF Interface. A notification  
1661 shall be sent any time the contents of the "oic.if.ll" OCF Interface representation are changed;  
1662 that is, if a Link is added, if a Link is removed, or if a Link is updated. Notifications on the  
1663 "oic.if.ll" OCF Interface shall contain all of the Links in the "oic.if.ll" OCF Interface representation.
- 1664 – Other Properties of the Collection Resource, if present, may be Observed by using the OCF  
1665 Interfaces defined in the definition for the Resource Type, including using the "oic.if.baseline"  
1666 OCF Interface.
- 1667 – The Client may choose to restrict the linked Resources to which the request is forwarded by  
1668 including additional query parameters in the request. The Server should process any additional  
1669 query parameters in a request that includes "oic.if.b" as selectors for linked Resources that are  
1670 to be processed by the request.
- 1671 – A Client shall perform UPDATE operations using the batch OCF Interface by creating a payload  
1672 that is similar to a RETRIEVE response payload from a batch OCF Interface request. The Server  
1673 shall send a separate UPDATE request to each of the linked Resources according to each "href"  
1674 Property and the corresponding value of the "rep" Property.
- 1675 – If the "href" value is empty, denoted by a zero length string or "" in JSON, the "rep" Property  
1676 shall be applied to linked Resources in the Collection.
- 1677 – Items with the empty "href" and link-specific "href" shall not be mixed in the same UPDATE  
1678 request.
- 1679 – All of the Properties in the UPDATE request may not be supported by the linked Resource. In  
1680 such cases, writable Properties in the UPDATE request that are supported by the linked  
1681 Resource shall be modified and Properties that are not supported shall be silently ignored.
- 1682 – The UPDATE response shall contain the updated values using the same payload schema as  
1683 RETRIEVE operations if provided by the linked Resource, along with the appropriate status  
1684 code. The aggregated response payload shall reflect the known state of the updated Properties  
1685 after the batch update was completed. If no payload is provided by the updated Resource then  
1686 an empty response (i.e. "rep": {}) shall be provided for that Resource.
- 1687 – A Collection shall not support the use of the UPDATE operation to add, modify or remove Links  
1688 in an existing Collection using the "oic.if.baseline", "ic.if.rw" or "oic.if.a" OCF Interfaces.

#### 1689 **7.6.3.4.2 Examples: Batch OCF Interface**

1690 Note that the examples provided in Table 9 are illustrative and do not include all mandatory schema  
1691 elements in all cases. It is assumed that the Default OCF Interface for the Resource Type  
1692 "x.org.example.rt.room" is specified in its Resource Type definition file as "oic.if.rw", which exposes  
1693 the Properties "x.org.example.colour" and "x.org.example.size".



**Table 9 – Batch OCF Interface Example**

<b>Resources</b>	<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><b>Use of batch, successful response</b></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><b>Use of batch, error response</b></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><b>Use of batch</b></p> <p><b>(UPDATE has POST semantics)</b></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>
--	--

<p><b>Use of batch (UPDATE has POST semantics)</b></p>	<pre>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"     }   } ]</pre> <p>This turns /the/light/1 off, turns /the/light/2 on, and sets the colour of /a/room/1 to "red".</p> <p>The response will be same as response for GET /a/room/1?if=oic.if.b with the updated Property values as shown.</p> <pre>[   {     "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}   } ]</pre> <p>Example use of additional query parameters to select items by matching Link Parameters.</p> <p>Turn on light 1 based on the "ins" Link Parameters value of "11111"</p> <pre>UPDATE /a/room/1?if=oic.if.b&amp;ins=11111 [   {     "href": "",     "rep": {       "value": false     }   } ]</pre> <p>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>
--	--

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

1695

1696 **7.6.3.5 Actuator OCF Interface**

1697 The actuator OCF Interface is the OCF Interface for viewing Resources that may be actuated i.e.  
1698 changes some value within or the state of the entity abstracted by the Resource:

- 1699 – The actuator OCF Interface name shall be "oic.if.a"
- 1700 – The actuator OCF Interface shall expose in the Resource Representation all mandatory  
1701 Properties as defined by the applicable OpenAPI 2.0 schema; the actuator OCF Interface may  
1702 also expose in the Resource Representation optional Properties as defined by the applicable  
1703 OpenAPI 2.0 schema that are implemented by the target Device.

1704 For example, a "Heater" Resource (for illustration only):

```
1705 /a/act/heater
1706 {
1707     "rt": ["acme.gas"],
1708     "if": ["oic.if.baseline", "oic.if.r", "oic.if.a", "oic.if.s"],
1709     "settemp": 10,
1710     "currenttemp" : 7
1711 }
```

1712 The actuator OCF Interface with respect to "Heater" Resource (for illustration only):

1713

1714 a) Retrieving values of an actuator.

1715 Request: GET /a/act/heater?if="oic.if.a"

1716

1717 Response:

```
1718 {
1719     "settemp": 10,
1720     "currenttemp" : 7
1721 }
```

1722 b) Correct use of actuator OCF Interface.

1723

1724 Request: POST /a/act/heater?if="oic.if.a"

```
1725 {
1726     "settemp": 20
1727 }
```

1728 Response:

```
1729     {
1730         Ok
1731     }
```

1732 c) Incorrect use of actuator OCF Interfance.

1733



1734 Request: POST /a/act/heater?if="oic.if.a"  
1735 {  
1736 "if": ["oic.if.s"] ← this is visible through baseline OCF Interface  
1737 }  
1738 Response:  
1739 {  
1740 Error  
1741 }

1742 – A RETRIEVE request using this OCF Interface shall return the Representation for this Resource  
1743 subject to any query and filter parameters that may also exist.

1744 – An UPDATE request using this OCF Interface shall provide a payload or body that contains the  
1745 Properties that will be updated on the target Resource.

1746 **7.6.3.6 Sensor OCF Interface**

1747 The sensor OCF Interface is the OCF Interface for retrieving measured, sensed or capability  
1748 specific information from a Resource that senses:

1749 – The sensor OCF Interface name shall be "oic.if.s".

1750 – The sensor OCF Interface shall expose in the Resource Representation all mandatory  
1751 Properties as defined by the applicable OpenAPI 2.0 schema; the sensor OCF Interface may  
1752 also expose in the Resource Representation optional Properties as defined by the applicable  
1753 OpenAPI 2.0 schema that are implemented by the target Device.

1754 – A RETRIEVE request using this OCF Interface shall return this representation for the Resource  
1755 subject to any query and filter parameters that may also exist.

1756 NOTE: The example here is with respect to retrieving values of a sensor

1757  
1758 Request: GET /a/act/heater?if="oic.if.s"  
1759  
1760 Response:  
1761 {  
1762 "currenttemp": 7  
1763 }  
1764

1765 **Incorrect use of the sensor.**

1766 Request: PUT /a/act/heater?if="oic.if.s" ← PUT is not allowed  
1767 {  
1768 "settemp": 20 ← this is possible through actuator OCF Interface  
1769 }  
1770 Response:  
1771 {  
1772 Error  
1773 }  
1774

1775 **Another incorrect use of the sensor.**

1776 Request: POST /a/act/heater?if="oic.if.s" ← POST is not allowed  
1777 {  
1778 "currenttemp": 15 ← this is possible through actuator OCF Interface  
1779 }  
1780 Response:  
1781 {  
1782 Error  
1783 }

### 1784 **7.6.3.7 Read-only OCF Interface**

1785 The read-only OCF Interface exposes only the Properties that may be read. This includes  
1786 Properties that may be read-only, read-write but not Properties that are write-only or set-only. The  
1787 applicable operations that can be applied to a Resource are only RETRIEVE and NOTIFY. An  
1788 attempt by a Client to apply a method other than RETRIEVE or NOTIFY to a Resource shall be  
1789 rejected with an error response code.

### 1790 **7.6.3.8 Read-write OCF Interface**

1791 The read-write OCF Interface is a generic OCF Interface to support reading and setting Properties  
1792 in a Resource. The applicable methods that can be applied to a Resource are only RETRIEVE,  
1793 NOTIFY, and UPDATE. For the RETRIEVE and NOTIFY operations, the behaviour is the same as  
1794 for the "oic.if.r" OCF Interface defined in 7.6.3.7. For the UPDATE operation, read-only Properties  
1795 (i.e. Properties tagged with "readOnly=True" in the OpenAPI 2.0 definition) shall not be in the  
1796 UPDATE payload. An attempt by a Client to apply a method other than RETRIEVE, NOTIFY, or  
1797 UPDATE to a Resource shall be rejected with an error response code.

## 1798 **7.7 Resource representation**

1799 Resource representation captures the state of a Resource at a particular time. The Resource  
1800 representation is exchanged in the request and response interactions with a Resource. A Resource  
1801 representation may be used to retrieve or update the state of a Resource.

1802 The Resource representation shall not be manipulated by the data connectivity protocols and  
1803 technologies (e.g., CoAP, UDP/IP or BLE).

## 1804 **7.8 Structure**

### 1805 **7.8.1 Introduction**

1806 In many scenarios and contexts, the Resources may have either an implicit or explicit structure  
1807 between them. A structure can, for example, be a tree, a mesh, a fan-out or a fan-in. The  
1808 Framework provides the means to model and map these structures and the relationships among  
1809 Resources. The primary building block for Resource structures in Framework is the Collection. A  
1810 Collection represents a container, which is extensible to model complex structures.

### 1811 **7.8.2 Resource Relationships**

#### 1812 **7.8.2.1 Introduction**

1813 Resource relationships are expressed as Links. A Link embraces and extends typed web links  
1814 concept as a means of expressing relationships between Resources. A Link consists of a set of  
1815 Parameters that define:

- 1816 – a context URI,
- 1817 – a target URI,
- 1818 – a relation from the context URI to the target URI, and
- 1819 – elements that provide metadata about the target URI, the relationship or the context of the Link.

1820 The target URI is mandatory and the other items in a Link are optional. Additional items in the Link  
1821 may be made mandatory based on the use of the links in different contexts (e.g. in Collections, in  
1822 discovery, in bridging etc.). OpenAPI 2.0 schema for the Link payload is provided in Annex D.

1823 An example of a Link is:

```
1824 {"href": "/switch", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a",  
1825 "oic.if.baseline"], "p": {"bm": 3}, "rel": "item"}
```

1826 Two Links are distinct from each other when at least one Parameter is different. For example the  
1827 two Links show here are distinct and can appear in the same list of Links.

```
1828 {"href": "/switch", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a",
1829 "oic.if.baseline"], "p": {"bm": 2}, "rel": "item"}
1830 {"href": "/switch", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a",
1831 "oic.if.baseline"], "p": {"bm": 2}}
```

1832 The document may mandate Parameters and Parameter values as required for certain capabilities.  
1833 For all Links returned in a response to a RETRIEVE on "/oic/res", if a Link does not explicitly include  
1834 the "rel" Parameter, a value of "rel"="hosts" shall be assumed. The relation value of "hosts" is  
1835 defined by IETF RFC 6690, the value of "item" by IETF RFC 6573, and the value of "self" by  
1836 IETF RFC 4287 and all are registered in the IANA Registry for Link Relations defined in  
1837 IANA Link Relations.

1838 As shown in Annex D the relation between the context URI and target URI in a Link is specified  
1839 using the "rel" JSON element and the value of this element specifies the particular relation.

1840 The context URI of the Link shall implicitly be the URI of the Resource (or specifically a Collection)  
1841 that contains the Link unless the Link specifies the "anchor" Parameter. The "anchor" Parameter  
1842 is used to change the context URI of a Link – the relationship with the target URI is based off the  
1843 anchor URI when the "anchor" is specified. "Anchor" Parameter uses transfer protocol URI for OIC  
1844 1.1 Link (e.g. "anchor": "coaps://[fe80::b1d6]:44444") and OCF URI defined in Sec 6 for OCF 1.0  
1845 Links (e.g. "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989"). For optional backward  
1846 compatibility with OIC 1.1, "anchor" Parameter uses transfer protocol URI for OIC 1.1 Link (e.g.  
1847 "anchor": "coaps://[fe80::b1d6]:44444").

1848 An example of using "anchors" in the context of Collections – a floor has rooms and rooms have  
1849 lights – the lights may be defined in floor as Links but the Links will have the "anchor" set to the  
1850 URI of the rooms that contain the lights (the relation is contains). This allows all lights in a floor to  
1851 be turned on or off together while still having the lights defined with respect to the rooms that  
1852 contain them (lights may also be turned on by using the room URI too). For example, here is the  
1853 use of "anchor" in Link:

```
1854 /a/floor {
1855   "links": [
1856     {
1857       "href": "/x/light1",
1858       "anchor": "/a/room1",   ** Note: /a/room1 has the item relationship with /x/light1; not /a/floor **
1859       "rel": "item"
1860     }
1861   ]
1862 }
1863
1864 /a/room1 {
1865   "links": [
1866     {
1867       ** Note: /a/room1 "contains" the /x/light since /a/room1 is the implicit context URI **
1868       "href": "/x/light1",
1869       "rel": "item"
1870     }
1871   ]
1872 }
```

## 1873 7.8.2.2 Parameters

### 1874 7.8.2.2.1 "ins" or Link Instance Parameter

1875 The "ins" Parameter identifies a particular Link instance in a list of Links. The "ins" Parameter may  
1876 be used to modify or delete a specific Link in a list of Links. The value of the "ins" Parameter is set  
1877 at instantiation of the Link by the OCF Device (Server) that is hosting the list of Links – once it has  
1878 been set, the "ins" Parameter shall not be modified for as long as the Link is a member of that list.

1879 **7.8.2.2.2 "p" or Policy Parameter**

1880 The Policy Parameter defines various rules for correctly accessing a Resource referenced by a  
 1881 target URI. The Policy rules are configured by a set of key-value pairs.

1882 The policy Parameter "p" is defined by:

- 1883 – "bm" key: The "bm" key corresponds to an integer value that is interpreted as an 8-bit bitmask.  
 1884 Each bit in the bitmask corresponds to a specific Policy rule. The rules are specified for "bm"  
 1885 in Table 10:

1886 **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 <sup>nd</sup> 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.

1887  
 1888 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  
 1889 measure. However, if any bit is set to value 1, then "bm" shall be included in "p" and all the bits shall be defined  
 1890 appropriately.

- 1891 – "sec" and "port" in the remaining bullets shall be used only in a response payload when the  
 1892 request does not include an OCF-Accept-Content-Format-Version option as defined in 12.2.5.  
 1893 In a payload sent in response to a request that includes an OCF-Accept-Content-Format-  
 1894 Version option "sec" and "port" shall not be used and instead the "eps" Parameter shall provide  
 1895 the information for an encrypted connection. See **Annex E** for the schema for the "p" Parameter  
 1896 that includes "sec" and "port".

- 1897 – "sec" key: The "sec" key corresponds to a Boolean value that indicates whether the Resource  
 1898 referenced by the target URI is accessed via an encrypted connection. If "sec" is true, the  
 1899 Resource is accessed via an encrypted connection, using the "port" specified. If "sec" is false,  
 1900 the Resource is accessed via an unencrypted connection, or via an encrypted connection (if  
 1901 such a connection is made using the "port" settings for another Resource, for which "sec" is  
 1902 true).

- 1903 – "port" key: The "port" key corresponds to an integer value that is used to indicate the port  
 1904 number where the Resource referenced by the target URI may be accessed via an encrypted  
 1905 connection.

- 1906 – If the Resource is only available via an encrypted connection (i.e. DTLS over IP), then

- 1907 – "p" shall include the "sec" key and its value shall be true.
- 1908 – "p" shall include the "port" key and its value shall be the port number where the encrypted
- 1909 connection may be established.
- 1910 – If the Resource is only available via an unencrypted connection, then
- 1911 – "p" shall include the "sec" key and its value shall be false or "p" shall omit the "sec" key; the
- 1912 default value of "sec" is false.
- 1913 – "p" shall omit the "port" key.
- 1914 – If the Resource is available via both an encrypted and unencrypted connection, then
- 1915 – "p" shall include the "sec" key and its value shall be false or "p" shall omit the "sec" key; the
- 1916 default value of "sec" is false.
- 1917 – "p" may omit the "port" key. If the "port" key is omitted, the Resource shall be available
- 1918 using the same "port" information as another Resource on the Device for which "sec" is true.
- 1919 – Access to the Resource on the port specified by the "port" key shall be made by an encrypted
- 1920 connection (e.g. "coaps://"). (Note that unencrypted connection to the Resource may be
- 1921 possible on a separate port discovered thru multicast discovery).
- 1922 – Note that access to the Resource is controlled by the ACL for the Resource. A successful
- 1923 encrypted connection does not ensure that the requested action will succeed. See
- 1924 ISO/IEC 30118-2:2018 clause 12 for more information.

1925 Example 1: This shows the Policy Parameter for a Resource that is discoverable but not Observable,  
 1926 and for which authenticated accesses shall be done via CoAPS port 33275.

```
1927 "p": {"bm": 1}
```

1928 Example 2: This shows a self-link, i.e. the "/oic/res" Link in itself that is discoverable and  
 1929 Observable.

```
1930 {
1931   "href": "/oic/res",
1932   "rel": "self",
1933   "rt": ["oic.wk.res"],
1934   "if": ["oic.if.ll", "oic.if.baseline"],
1935   "p": {"bm": 3}
1936 }
```

### 1937 7.8.2.2.3 "type" or Media Type Parameter

1938 The "type" Parameter may be used to specify the various media types that are supported by a  
 1939 specific target Resource. The default type of "application/vnd.ocf+cbor" shall be used when the  
 1940 "type" element is omitted. Once a Client discovers this information for each Resource, it may use  
 1941 one of the available representations in the appropriate header field of the Request or Response.

### 1942 7.8.2.2.4 "di" or Device ID Parameter

1943 The "di" Parameter specifies the Device ID of the Device that hosts the target Resource defined in  
 1944 the in the "href" Parameter.

1945 The Device ID may be used to qualify a relative reference used in the "href" or to lookup OCF  
 1946 Endpoint information for the relative reference.

### 1947 7.8.2.2.5 "eps" Parameter

1948 The "eps" Parameter indicates the OCF Endpoint information of the target Resource.

1949 "eps" shall have as its value an array of items and each item represents OCF Endpoint information  
 1950 with "ep" and "pri" as specified in 10.2. "ep" is mandatory but "pri" is optional.

1951 This is an example of "eps" with multiple OCF Endpoints.

```
1952 "eps": [  
1953   {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},  
1954   {"ep": "coaps://[fe80::b1d6]:1122"},  
1955   {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}  
1956 ]
```

1957 When "eps" is present in a link, the OCF Endpoint information in "eps" can be used to access the  
1958 target Resource referred by the "href" Parameter.

1959 Note that the type of OCF Endpoint – Secure or Unsecure – that a Resource exposes merely  
1960 determines the connection type(s) guaranteed to be available for sending requests to the Resource.  
1961 For example, if a Resource only exposes a single CoAP "ep", it does not guarantee that the  
1962 Resource cannot also be accessed via a Secure OCF Endpoint (e.g. via a CoAPS "ep" from another  
1963 Resource's "eps information). Nor does exposing a given type of OCF Endpoint ensure that access  
1964 to the Resource will be granted using the "ep" information. Whether requests to the Resource are  
1965 granted or denied by the Access Control layer is separate from the "eps" information, and is  
1966 determined by the configuration of the /acl2 Resource (see ISO/IEC 30118-2:2018 clause 13.5.3  
1967 for details).

1968 When present, max-age information (e.g. Max-Age option for CoAP defined in IETF RFC 7252)  
1969 determines the maximum time "eps" values may be cached before they are considered stale.

### 1970 **7.8.2.3 Formatting**

1971 When formatting in JSON, the list of Links shall be an array.

### 1972 **7.8.2.4 List of Links in a Collection**

1973 A Resource that exposes one or more Properties that are defined to be an array of Links where  
1974 each Link can be discretely accessed is a Collection. The Property Name "links" is recommended  
1975 for such an array of Links.

1976 This is an example of a Resource with a list of Links.

```
1977 /Room1  
1978 {  
1979   "rt": ["my.room"],  
1980   "if": ["oic.if.ll", "oic.if.baseline" ],  
1981   "color": "blue",  
1982   "links":  
1983   [  
1984     {  
1985       "href": "/oic/d",  
1986       "rt": ["oic.d.light", "oic.wk.d"],  
1987       "if": [ "oic.if.r", "oic.if.baseline" ],  
1988       "p": {"bm": 1}  
1989     },  
1990     {  
1991       "href": "/oic/p",  
1992       "rt": ["oic.wk.p"],  
1993       "if": [ "oic.if.r", "oic.if.baseline" ],  
1994       "p": {"bm": 1}  
1995     },  
1996     {  
1997       "href": "/switch",  
1998       "rt": ["oic.r.switch.binary"],  
1999       "if": [ "oic.if.a", "oic.if.baseline" ],  
2000       "p": {"bm": 3},  
2001       "mt": [ "application/vnd.ocf+cbor", "application/exi+xml" ]  
2002     },  
2003   ]  
2004 }
```

```

2003     {
2004         "href": "/brightness",
2005         "rt": [ "oic.r.light.brightness" ],
2006         "if": [ "oic.if.a", "oic.if.baseline" ],
2007         "p": { "bm": 3 }
2008     }
2009 ]
2010 }

```

2011 **7.8.2.5 Properties describing an array of Links**

2012 If a Resource Type that defines an array of Links (e.g. Collections, Atomic Measurements) has  
2013 restrictions on the "rt" values that can be within the array of Links, the Resource Type will define  
2014 the "rts" Property. The "rts" Property as defined in Table 11 will include all "rt" values allowed for  
2015 all Links in the array. If the Resource Type does not define the "rts" Property or the "rts" Property  
2016 is an empty array, then any "rt" value is permitted in the array of Links.

2017 For all instances of a Resource Type that defines the "rts" Property, the "rt" Link Parameter in  
2018 every Link in the array of Links shall be one of the "rt" values that is included in the "rts"  
2019 Property.

2020 **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.

2021

2022 If a Resource Type that defines an array of Links has "rt" values which are required to be in the  
2023 array, the Resource Type will define the "rts-m" Property, as defined in Table 12, which will contain  
2024 all of the "rt" values that are required to be in the array of Links. If "rts-m" is defined, and "rts" is  
2025 defined and is not an empty array, then the "rt" values present in "rts-m" will be part of the values  
2026 present in "rts". Moreover, if the "rts-m" Property is defined, it shall be mandated (i.e. included in  
2027 the "required" field of a JSON definition) in the Resource definition and Introspection Device Data  
2028 (see 11.8).

2029 For all instances of a Resource Type that defines the "rts-m" Property, there shall be at least one  
2030 Link in the array of Links corresponding to each one of the "rt" values in the "rts-m" Property; for  
2031 all such Links the "rt" Link Parameter shall contain at least one of the "rt" values in the "rts-m"  
2032 Property.

2033 **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.

2034

2035 **7.8.3 Collections**

2036 **7.8.3.1 Overview**

2037 A Resource that contains one or more references (specified as Links) to other Resources is a  
2038 Collection. These references may be related to each other or just be a list; the Collection provides  
2039 a means to refer to this set of references with a single handle (i.e. the URI). A simple Resource is  
2040 kept distinct from a Collection. Any Resource may be turned into a Collection by binding Resource  
2041 references as Links. Collections may be used for creating, defining or specifying hierarchies,  
2042 indexes, groups, and so on.

2043 A Collection shall have at least one Resource Type and at least one OCF Interface bound at all  
2044 times during its lifetime. During creation time of a Collection the Resource Type and OCF Interfaces  
2045 are specified. The initial defined Resource Types and OCF Interfaces may be updated during its  
2046 life time. These initial values may be overridden using mechanism used for overriding in the case  
2047 of a Resource. Additional Resource Types and OCF Interfaces may be bound to the Collection at  
2048 creation or later during the lifecycle of the Collection.

2049 A Collection shall define a Property that is an array with zero or more Links. The target URIs in the  
2050 Links may reference another Collection or another Resource. The referenced Collection or  
2051 Resource may reside on the same Device as the Collection that includes that Link (called a local  
2052 reference) or may reside on another Device (called a remote reference). The context URI of the  
2053 Links in the array shall (implicitly) be the Collection that contains that Property. The (implicit)  
2054 context URI may be overridden with explicit specification of the "anchor" Parameter in the Link  
2055 where the value of "anchor" is the new base of the Link.

2056 A Resource may be referenced in more than one Collection, therefore, a unique parent-child  
2057 relationship is not guaranteed. There is no pre-defined relationship between a Collection and the  
2058 Resource referenced in the Collection, i.e., the application may use Collections to represent a  
2059 relationship but none is automatically implied or defined. The lifecycles of the Collection and the  
2060 referenced Resource are also independent of one another.

2061 If the "drel" Property is defined for the Collection then all Links that don't explicitly specify a  
2062 relationship shall inherit this default relationship in the context of that Collection. The default  
2063 relationship defines the implicit relationship between the Collection and the target URI in the Link.

2064 In the following example a Property "links" represents the list of Links in a Collection. The "links"  
2065 Property has, as its value, an array of items and each item is a Link.

```
2066 /my/house ← This is IRI/URI of the Resosource
2067 {
2068   "rt": ["my.r.house"], ← This and the next 3 lines are the Properties of the Resource
2069   "color": "blue",
2070   "n": "myhouse",
2071   "links": [
2072     { ← This and the next 4 lines are the Parameters of a Liink
2073       "href": "/door",
2074       "rt": ["oic.r.door"],
2075       "if": ["oic.if.b", "oic.if.ll", "oic.if.baseline"]
2076     },
2077     {
2078       "href": "/door/lock",
2079       "rt": ["oic.r.lock"],
2080       "if": ["oic.if.b", "oic.if.baseline"],
2081       "type": ["application/cbor", "application/exi+xml"]
2082     },
2083     {
2084       "href": "/light",
```



```

2087     "rt": ["oic.r.light"],
2088     "if": ["oic.if.s", "oic.if.baseline"]
2089   },
2090   {
2091     "href": "/binarySwitch",
2092     "rt": ["oic.r.switch.binary"],
2093     "if": ["oic.if.a", "oic.if.baseline"],
2094     "type": ["application/cbor"]
2095   }
2096 ]
2097 }
2098 }
2099 }

```

2100 A Collection may be:

- 2101 – A pre-defined Collection where the Collection has been defined a priori and the Collection is  
2102 static over its lifetime. Such Collections may be used to model, for example, an appliance that  
2103 is composed of other Devices or fixed set of Resources representing fixed functions.
- 2104 – A Device local Collection where the Collection is used only on the Device that hosts the  
2105 Collection. Such Collections may be used as a short-hand on a Client for referring to many  
2106 Servers as one.
- 2107 – A centralized Collection where the Collection is hosted on a Device but other Devices may  
2108 access or update the Collection.
- 2109 – A hosted Collection where the Collection is centralized but is managed by an authorized agent  
2110 or party.

### 2111 7.8.3.2 Collection Properties

2112 A Collection shall define a Property that is an array of Links (the Property Name "links" is  
2113 recommended). In addition, other Properties may be defined for the Collection by the Resource  
2114 Type. The mandatory and recommended Common Properties for a Collection are shown in Table 13.  
2115 This list of Common Properties is in addition to those defined for Resources in 7.3.2.

2116 **Table 13 – Common Properties for Collections (in addition to Common Properties defined**  
2117 **in 7.3.2)**

Property	Description	Property Name	Value Type	Mandatory
<b>Links</b>	The array of Links in the Collection	Per Resource Type definition	json Array of Links	Yes
<b>Resource Types</b>	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
<b>Mandatory Resource Types</b>	The list of Resource Types for Links that are mandatory in the Collection.	As defined in Table 12	As defined in Table 12	No

2118

2119 **7.8.3.3 Default Resource Type**

2120 A default Resource Type, "oic.wk.col", is available for Collections. This Resource Type shall be  
2121 used only when another type has not been defined on the Collection or when no Resource Type  
2122 has been specified at the creation of the Collection.

2123 The default Resource Type provides support for the Common Properties including an array of Links  
2124 with the Property Name "links".

2125 **7.8.3.4 Default OCF Interface**

2126 All instances of a Collection shall support the links list ("oic.if.ll") OCF Interface in addition to the  
2127 baseline ("oic.if.baseline") OCF Interface. An instance of a Collection may optionally support  
2128 additional OCF Interfaces that are defined within this document. The Default OCF Interface for a  
2129 Collection shall be links list ("oic.if.ll") unless otherwise specified by the Resource Type definition.

2130 **7.8.4 Atomic Measurement**

2131 **7.8.4.1 Overview**

2132 Certain use cases require that the Properties of multiple Resources are only accessible as a group  
2133 and individual access to those Properties of each Resource by a Client is prohibited. The Atomic  
2134 Measurement Resource Type is defined to meet this requirement. This is accomplished through  
2135 the use of the Batch OCF Interface.

2136 **7.8.4.2 Atomic Measurement Properties**

2137 An Atomic Measurement shall define a Property that is an array of Links (the Property Name "links"  
2138 is recommended). In addition, other Properties may be defined for the Atomic Measurement by the  
2139 Resource Type. The mandatory and recommended Common Properties for an Atomic  
2140 Measurement are shown in Table 14. This list of Common Properties is in addition to those defined  
2141 for Resources in 7.3.2.

2142 **Table 14 – Common Properties for Atomic Measurement (in addition to Common Properties**  
2143 **defined in 7.3.2)**

Property	Description	Property Name	Value Type	Mandatory
<b>Links</b>	The array of Links in the Atomic Measurement	Per Resource Type definition	json Array of Links	Yes
<b>Resource Types</b>	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
<b>Mandatory Resource Types</b>	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

2144

2145 **7.8.4.3 Normative behaviour**

2146 The normative behaviour of an Atomic Measurement is as follows:

- 2147 – The behaviour of the Batch OCF Interface ("oic.if.b") on the Atomic Measurement is defined as  
2148 follows:
- 2149 – Only RETRIEVE and NOTIFY operations are supported, for Batch OCF Interface, on Atomic  
2150 Measurement; the behavior of the RETRIEVE and NOTIFY operations shall be the same as  
2151 specified in 7.6.3.4, with exceptions as provided for in 7.8.4.3.
  - 2152 – The UPDATE operation is not allowed, for Batch OCF Interface, on Atomic Measurement; if  
2153 an UPDATE operation is received, it shall result in a method not allowed error code.
  - 2154 – An error response shall not include any representation of a linked Resource (i.e. empty  
2155 response for all linked Resources).
- 2156 – Any linked Resource within an Atomic Measurement (i.e. the target Resource of a Link in an  
2157 Atomic Measurement) is subject to the following conditions:
- 2158 – Linked Resources within an Atomic Measurement and the Atomic Measurement itself shall  
2159 exist on a single Server.
  - 2160 – CRUDN operations shall not be allowed on linked Resources and shall result in a forbidden  
2161 error code.
  - 2162 – Linked Resources shall not expose the "oic.if.ll" OCF Interface. Since CRUDN operations  
2163 are not allowed on linked Resources, the "oic.if.ll" OCF Interface would never be accessible.
- 2164 – Links to linked Resources in an Atomic Measurement shall only be accessible through the  
2165 "oic.if.ll" or the "oic.if.baseline" OCF Interfaces of an Atomic Measurement.
- 2166 – The linked Resources shall not be listed in "/oic/res".
  - 2167 – A linked Resource in an Atomic Measurement shall have defined one of "oic.if.a", "oic.if.s",  
2168 "oic.if.r", or "oic.if.rw" as its Default OCF Interface.
  - 2169 – Not all linked Resources in an Atomic Measurement are required to be Observable. If an Atomic  
2170 Measurement is being Observed using the "oic.if.b" OCF Interface, notification responses shall  
2171 not be generated when the linked Resources which are not marked Observable are updated or  
2172 change state.
  - 2173 – All linked Resources in an Atomic Measurement shall be included in every RETRIEVE and  
2174 Observe response when using the "oic.if.b" OCF Interface.
  - 2175 – An Atomic Measurement shall support the "oic.if.b" and the "oic.if.ll" OCF Interfaces.
  - 2176 – Filtering of linked Resources in an Atomic Measurement is not allowed. Query parameters that  
2177 select one or more individual linked Resources in a request to an Atomic Measurement shall  
2178 result in a "forbidden" error code.
  - 2179 – If the "rel" Link Parameter is included in a Link contained in an Atomic Measurement, it shall  
2180 have either the "hosts" or the "item" value.
  - 2181 – The Default OCF Interface of an Atomic Measurement is "oic.if.b".

#### 2182 **7.8.4.4 Security considerations**

2183 Access rights to an Atomic Measurement Resource Type is as specified in clause 12.2.7.2 (ACL  
2184 considerations for batch request to the Atomic Measurement Resource Type) of ISO/IEC 30118-  
2185 2:2018).

#### 2186 **7.8.4.5 Default Resource Type**

2187 The Resource Type is defined as "oic.wk.atomicmeasurement" as defined in Table 15.

2188

**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

2189

2190 The Properties for Atomic Measurement are as defined in Table 16.

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

2191  
2192

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

2193

2194 **7.9 Third (3<sup>rd</sup>) party specified extensions**

2195 This clause describes how a 3<sup>rd</sup> party may add Device Types, Resource Types, 3<sup>rd</sup> party defined  
2196 Properties to an existing or 3<sup>rd</sup> party defined Resource Type, 3<sup>rd</sup> party defined enumeration values  
2197 to an existing enumeration and 3<sup>rd</sup> party defined Parameters to an existing defined Property.

2198 A 3<sup>rd</sup> party may specify additional (non-OCF) Resources within an OCF Device. A 3<sup>rd</sup> party may  
2199 also specify additional Properties within an existing OCF defined Resource Type. Further a 3<sup>rd</sup>  
2200 party may extend an OCF defined enumeration with 3<sup>rd</sup> party defined values.

2201 A 3<sup>rd</sup> party defined Device Type may expose both 3<sup>rd</sup> party and OCF defined Resource Types. A  
2202 3<sup>rd</sup> party defined Device Type must expose the mandatory Resources for all OCF Devices defined  
2203 within this document.

2204 A 3<sup>rd</sup> party defined Resource Type shall include any mandatory Properties defined in this  
2205 document and also any vertical specified mandatory Properties. All Properties defined within a 3<sup>rd</sup>  
2206 party defined Resource Type that are part of the OCF namespace that are not Common Properties  
2207 as defined in this document shall follow the 3<sup>rd</sup> party defined Property rules in Table 17.

2208 The following table defines the syntax rules for 3<sup>rd</sup> party defined Resource Type elements. Within  
2209 the table the term "Domain\_Name" refers to a domain name that is owned by the 3<sup>rd</sup> party that is  
2210 defining the new element.

2211

**Table 17 – 3rd party defined Resource elements**

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

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

2212

2213 With respect to the use of the Domain\_Name in this scheme the labels are reversed from how they  
 2214 appear in DNS or other resolution mechanisms. The 3<sup>rd</sup> party defined Device Type and Resource  
 2215 Type otherwise follow the rules defined in clause 7.4.2. 3<sup>rd</sup> party defined Resource Types should  
 2216 be registered in the IANA Constrained RESTful Environments (CoRE) Parameters registry.

2217 For example:

2218 x.com.samsung.galaxyphone.accelerator  
 2219 x.com.cisco.ciscorouterport  
 2220 x.com.hp.printerhead  
 2221 x.org.allseen.newinterface.newproperty

2222 **7.10 Query Parameters**

2223 **7.10.1 Introduction**

2224 Properties and Parameters (including those that are part of a Link) may be used in the query part  
 2225 of a URI (see 6.2.2) as one criterion for selection of a particular Resource. This is done by declaring  
 2226 the Property (i.e. <Property Name> = <desired Property Value>) as one of the segments of the  
 2227 query. Only ASCII strings are permitted in query filters, and NULL characters are disallowed in  
 2228 query filters. This means that only Property Values with ASCII characters may be matched in a  
 2229 query filter.

2230 The Resource is selected when all the declared Properties or Link Parameters in the query match  
 2231 the corresponding Properties or Link Parameters in the target.

2232 **7.10.2 Use of multiple parameters within a query**

2233 When a query contains multiple separate query parameters these are delimited by an "&" as  
 2234 described in 6.2.2.

2235 A Client may apply multiple separate query parameters, for  
 2236 example "?ins=11111&rt=oic.r.switch.binary". If such queries are supported by the Server this shall  
 2237 be accomplished by matching "all of" the different query parameter types ("rt", "ins", "if", etc)  
 2238 against the target of the query. In the example, this resolves to an instance of oic.r.switch.binary  
 2239 that also has an "ins" populated as "11111". There is no significance applied to the order of the  
 2240 query parameters.

2241 A Client may select more than one Resource Type using repeated query parameters, for example  
 2242 "?rt=oic.r.switch.binary&rt=oic.r.ramptime". If such queries are supported by the Server this shall  
 2243 be accomplished by matching "any of" the repeated query parameters against the target of the  
 2244 query. In the example, any instances of "oic.r.switch.binary" and/or "oic.r.ramptime" that may exist  
 2245 are selected.

2246 A Client may combine both multiple repeated parameters and multiple separate parameters in a  
 2247 single query, for example "?if=oic.if.b&ins=11111&rt=oic.r.switch.binary&rt=oic.r.ramptime". If  
 2248 such queries are supported by the Server this shall be accomplished by matching "any of" the  
 2249 repeated query parameters and then matching "all of" the different query parameter types. In the  
 2250 example any instances of "oic.r.switch.binary" and/or "oic.r.ramptime" that also have an "ins" of  
 2251 "11111" that may exist are selected in a batch response.

2252 NOTE The parameters within a query string are represented within the actual messaging protocol as defined in clause  
 2253 11.9.

### 2254 **7.10.3 Application to multi-value "rt" Resources**

2255 An "rt" query for a multi-value "rt" Resource with the Default OCF Interface of "oic.if.a", "oic.if.s",  
2256 "oic.if.r", "oic.if.rw" or "oic.if.baseline" is an extension of a generic "rt" query. When a Server  
2257 receives a RETRIEVE request for a multi-value "rt" Resource with an "rt" query, (i.e. GET  
2258 /ResExample?rt=oic.r.foo), the Server should respond only when the query value is an item of the  
2259 "rt" Property Value of the target Resource and should send back only the Properties associated  
2260 with the query value(s). For example, upon receiving GET /ResExample?rt=oic.r.switch.binary  
2261 targeting a Resource with "rt": ["oic.r.switch.binary", "oic.r.light.brightness"], the Server responds  
2262 with only the Properties of oic.r.switch.binary.

### 2263 **7.10.4 OCF Interface specific considerations for queries**

#### 2264 **7.10.4.1 OCF Interface selection**

2265 When an OCF Interface is to be selected for a request, it shall be specified as a query parameter  
2266 in the URI of the Resource in the request message. If no query parameter is specified, then the  
2267 Default OCF Interface shall be used. If the selected OCF Interface is not one of the permitted OCF  
2268 Interfaces on the Resource then selecting that OCF Interface is an error and the Server shall  
2269 respond with an error response code.

2270 For example, the baseline OCF Interface may be selected by adding "if=oic.if.baseline" to the list  
2271 of query parameters in the URI of the target Resource. For example: "GET  
2272 /oic/res?if=oic.if.baseline".

#### 2273 **7.10.4.2 Batch OCF Interface**

2274 See 7.6.3.4 for details on the batch OCF Interface itself. Query parameters may be used with the  
2275 batch OCF Interface in order to select particular Resources in a Collection for retrieval or update;  
2276 these parameters are used to select items in the Collection by matching Link Parameter Values.

2277 When Link selection query parameters are used with RETRIEVE operations applied using the batch  
2278 OCF Interface, only the Resources in the Collection with matching Link Parameters should be  
2279 returned.

2280 When Link selection query parameters are used with UPDATE operations applied using the batch  
2281 OCF Interface, only the Resources having matching Link Parameters should be updated.

2282 See 7.6.3.4.2 for examples of RETRIEVE and UPDATE operations that use Link selection query  
2283 parameters.

## 2284 **8 CRUDN**

### 2285 **8.1 Overview**

2286 CREATE, RETRIEVE, UPDATE, DELETE, and NOTIFY (CRUDN) are operations defined for  
2287 manipulating Resources. These operations are performed by a Client on the Resources contained  
2288 in n Server.

2289 On reception of a valid CRUDN operation a Server hosting the Resource that is the target of the  
2290 request shall generate a response depending on the OCF Interface included in the request; or  
2291 based on the Default OCF Interface for the Resource Type if no OCF Interface is included.

2292 CRUDN operations utilize a set of parameters that are carried in the messages and are defined in  
2293 Table 18. A Device shall use CBOR as the default payload (content) encoding scheme for Resource  
2294 representations included in CRUDN operations and operation responses; a Device may negotiate  
2295 a different payload encoding scheme (e.g, see in 12.2.4 for CoAP messaging). Clauses 8.2 through  
2296 8.6 respectively specify the CRUDN operations and use of the parameters. The type definitions for  
2297 these terms will be mapped in the clause 12 for each protocol.

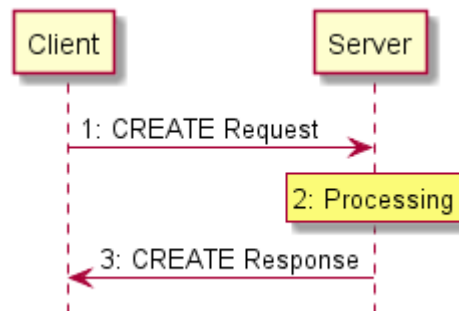
**Table 18 – Parameters of CRUDN messages**

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.

## 2299 8.2 CREATE

### 2300 8.2.1 Overview

2301 The CREATE operation is used to request the creation of new Resources on the Server. The  
 2302 CREATE operation is initiated by the Client and consists of three steps, as depicted in Figure 10.



2303

2304

**Figure 10 – CREATE operation**

### 2305 8.2.2 CREATE request

2306 The CREATE request message is transmitted by the Client to the Server to create a new Resource  
 2307 by the Server. The CREATE request message will carry the following parameters:

- 2308 – *fr*: Unique identifier of the Client
- 2309 – *to*: URI of the target Resource responsible for creation of the new Resource.
- 2310 – *ri*: Identifier of the CREATE request.
- 2311 – *cn*: Information of the Resource to be created by the Server.
- 2312 – *cn* will include the URI and Resource Type Property of the Resource to be created.
- 2313 – *cn* may include additional Properties of the Resource to be created.
- 2314 – *op*: CREATE

2315 **8.2.3 Processing by the Server**

2316 Following the receipt of a CREATE request, the Server may validate if the Client has the  
2317 appropriate rights for creating the requested Resource. If the validation is successful, the Server  
2318 creates the requested Resource. The Server caches the value of *ri* parameter in the CREATE  
2319 request for inclusion in the CREATE response message.

2320 **8.2.4 CREATE response**

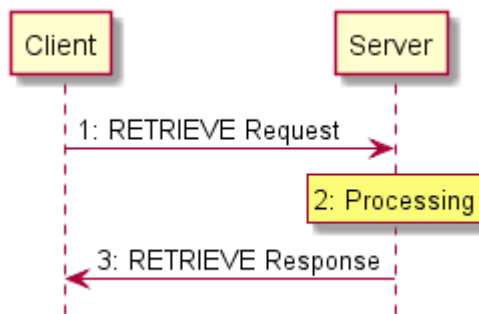
2321 The Server shall transmit a CREATE response message in response to a CREATE request  
2322 message from a Client. The CREATE response message will include the following parameters.

- 2323 – *fr*: Unique identifier of the Server
- 2324 – *to*: Unique identifier of the Client
- 2325 – *ri*: Identifier included in the CREATE request
- 2326 – *cn*: Information of the Resource as created by the Server.
  - 2327 – *cn* will include the URI of the created Resource.
  - 2328 – *cn* will include the Resource representation of the created Resource.
- 2329 – *rs*: The result of the CREATE operation.

2330 **8.3 RETRIEVE**

2331 **8.3.1 Overview**

2332 The RETRIEVE operation is used to request the current state or representation of a Resource. The  
2333 RETRIEVE operation is initiated by the Client and consists of three steps, as depicted in Figure 11.



2334

2335 **Figure 11 – RETRIEVE operation**

2336 **8.3.2 RETRIEVE request**

2337 RETRIEVE request message is transmitted by the Client to the Server to request the representation  
2338 of a Resource from a Server. The RETRIEVE request message will carry the following parameters.

- 2339 – *fr*: Unique identifier of the Client.
- 2340 – *to*: URI of the Resource the Client is targeting.
- 2341 – *ri*: Identifier of the RETRIEVE request.
- 2342 – *op*: RETRIEVE.

2343 **8.3.3 Processing by the Server**

2344 Following the receipt of a RETRIEVE request, the Server may validate if the Client has the  
2345 appropriate rights for retrieving the requested data and the Properties are readable. The Server  
2346 caches the value of *ri* parameter in the RETRIEVE request for use in the response



2347 **8.3.4 RETRIEVE response**

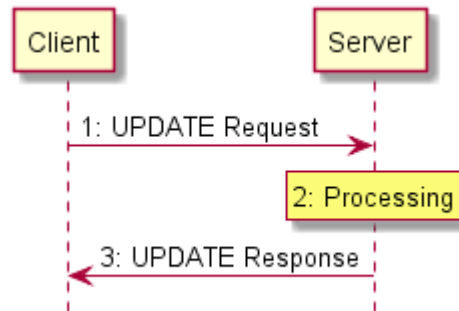
2348 The Server shall transmit a RETRIEVE response message in response to a RETRIEVE request  
2349 message from a Client. The RETRIEVE response message will include the following parameters.

- 2350 – *fr*: Unique identifier of the Server.
- 2351 – *to*: Unique identifier of the Client.
- 2352 – *ri*: Identifier included in the RETRIEVE request.
- 2353 – *cn*: Information of the Resource as requested by the Client.
- 2354 – *cn* should include the URI of the Resource targeted in the RETRIEVE request.
- 2355 – *rs*: The result of the RETRIEVE operation.

2356 **8.4 UPDATE**

2357 **8.4.1 Overview**

2358 The UPDATE operation is either a Partial UPDATE or a complete replacement of the information  
2359 in a Resource in conjunction with the OCF Interface that is also applied to the operation. The  
2360 UPDATE operation is initiated by the Client and consists of three steps, as depicted in Figure 12.



2361

2362 **Figure 12 – UPDATE operation**

2363 **8.4.2 UPDATE request**

2364 The UPDATE request message is transmitted by the Client to the Server to request the update of  
2365 information of a Resource on the Server. The UPDATE request message will carry the following  
2366 parameters.

- 2367 – *fr*: Unique identifier of the Client.
- 2368 – *to*: URI of the Resource targeted for the information update.
- 2369 – *ri*: Identifier of the UPDATE request.
- 2370 – *op*: UPDATE.
- 2371 – *cn*: Information, including Properties, of the Resource to be updated at the target Resource.

2372 **8.4.3 Processing by the Server**

2373 **8.4.3.1 Overview**

2374 Following the receipt of an UPDATE request, the Server may validate if the Client has the  
2375 appropriate rights for updating the requested data. If the validation is successful the Server updates  
2376 the target Resource information according to the information carried in *cn* parameter of the  
2377 UPDATE request message. The Server caches the value of *ri* parameter in the UPDATE request  
2378 for use in the response.

2379 An UPDATE request that includes Properties that are read-only shall be rejected by the Server with  
2380 an *rs* indicating a bad request.

2381 An UPDATE request shall be applied only to the Properties in the target Resource visible via the  
2382 applied OCF Interface that support the operation. An UPDATE of non-existent Properties is ignored.

2383 An UPDATE request shall be applied to the Properties in the target Resource even if those Property  
2384 Values are the same as the values currently exposed by the target Resource.

### 2385 **8.4.3.2 Resource monitoring by the Server**

2386 The Server shall monitor the state the Resource identified in the Observe request from the Client.  
2387 Anytime there is a change in the state of the Observed Resource or an UPDATE operation applied  
2388 to the Resource, the Server sends another RETRIEVE response with the Observe indication. The  
2389 mechanism does not allow the Client to specify any bounds or limits which trigger a notification,  
2390 the decision is left entirely to the Server.

### 2391 **8.4.3.3 Additional RETRIEVE responses with Observe indication**

2392 The Server shall transmit updated RETRIEVE response messages following Observed changes in  
2393 the state of the Resources requested by the Client. The RETRIEVE response message shall include  
2394 the parameters listed in 11.4.2.4.

### 2395 **8.4.4 UPDATE response**

2396 The UPDATE response message will include the following parameters:

- 2397 – *fr*: Unique identifier of the Server.
- 2398 – *to*: Unique identifier of the Client.
- 2399 – *ri*: Identifier included in the UPDATE request.
- 2400 – *rs*: The result of the UPDATE request.

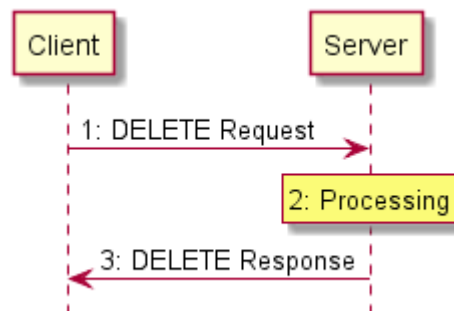
2401 The UPDATE response message may also include the following parameters:

- 2402 – *cn*: The Resource representation following processing of the UPDATE request.

## 2403 **8.5 DELETE**

### 2404 **8.5.1 Overview**

2405 The DELETE operation is used to request the removal of a Resource. The DELETE operation is  
2406 initiated by the Client and consists of three steps, as depicted in Figure 13.



2407

2408

**Figure 13 – DELETE operation**

### 2409 **8.5.2 DELETE request**

2410 DELETE request message is transmitted by the Client to the Server to delete a Resource on the  
2411 Server. The DELETE request message will carry the following parameters:

- 2412 – *fr*: Unique identifier of the Client.
- 2413 – *to*: URI of the target Resource which is the target of deletion.
- 2414 – *ri*: Identifier of the DELETE request.
- 2415 – *op*: DELETE.

### 2416 **8.5.3 Processing by the Server**

2417 Following the receipt of a DELETE request, the Server may validate if the Client has the appropriate  
2418 rights for deleting the identified Resource, and whether the identified Resource exists. If the  
2419 validation is successful, the Server removes the requested Resource and deletes all the associated  
2420 information. The Server caches the value of *ri* parameter in the DELETE request for use in the  
2421 response.

### 2422 **8.5.4 DELETE response**

2423 The Server shall transmit a DELETE response message in response to a DELETE request message  
2424 from a Client. The DELETE response message will include the following parameters.

- 2425 – *fr*: Unique identifier of the Server.
- 2426 – *to*: Unique identifier of the Client.
- 2427 – *ri*: Identifier included in the DELETE request.
- 2428 – *rs*: The result of the DELETE operation.

## 2429 **8.6 NOTIFY**

### 2430 **8.6.1 Overview**

2431 The NOTIFY operation is used to request asynchronous notification of state changes. Complete  
2432 description of the NOTIFY operation is provided in 11.4. The NOTIFY operation uses the  
2433 NOTIFICATION response message which is defined here.

### 2434 **8.6.2 NOTIFICATION response**

2435 The NOTIFICATION response message is sent by a Server to notify the URLs identified by the  
2436 Client of a state change. The NOTIFICATION response message carries the following parameters.

- 2437 – *fr*: Unique identifier of the Server.
- 2438 – *to*: URI of the Resource target of the NOTIFICATION message.
- 2439 – *ri*: Identifier included in the CREATE request.
- 2440 – *op*: NOTIFY.
- 2441 – *cn*: The updated state of the Resource.

## 2442 **9 Network and connectivity**

### 2443 **9.1 Introduction**

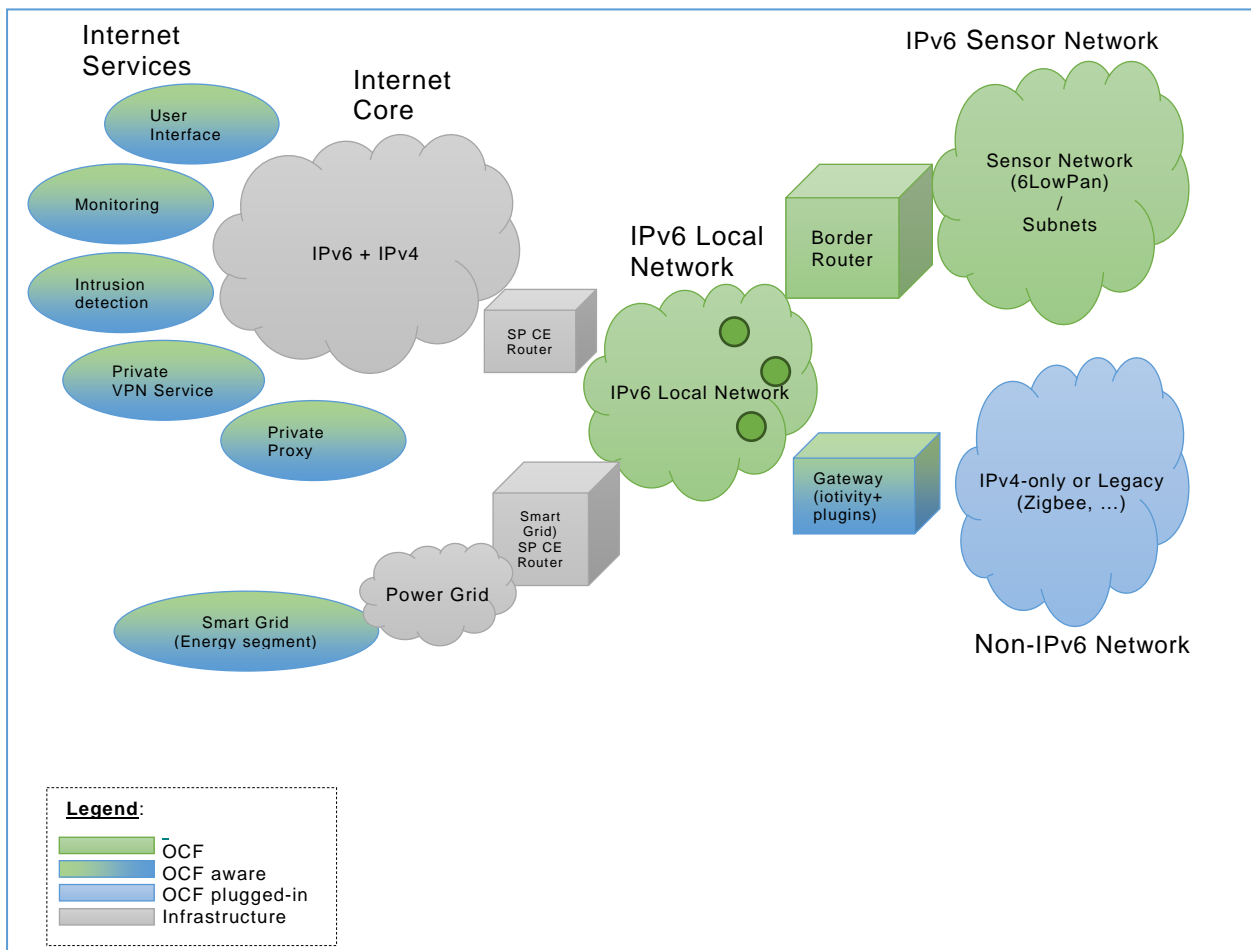
2444 The Internet of Things is comprised of a wide range of applications which sense and actuate the  
2445 physical world with a broad spectrum of device and network capabilities: from battery powered  
2446 nodes transmitting 100 bytes per day and able to last 10 years on a coin cell battery, to mains  
2447 powered nodes able to maintain Megabit video streams. It is estimated that many 10s of billions of  
2448 IoT devices will be deployed over the coming years.

2449 It is desirable that the connectivity options be adapted to the IP layer. To that end, IETF has  
2450 completed considerable work to adapt Bluetooth®, Wi-Fi, 802.15.4, LPWAN, etc. to IPv6. These  
2451 adaptations, plus the larger address space and improved address management capabilities, make  
2452 IPv6 the clear choice for the OCF network layer technology.

2453 **9.2 Architecture**

2454 While the aging IPv4 centric network has evolved to support complex topologies, its deployment  
 2455 was primarily provisioned by a single Internet Service Provider (ISP) as a single network. More  
 2456 complex network topologies, often seen in residential home, are mostly introduced through the  
 2457 acquisition of additional home network devices, which rely on technologies like private Network  
 2458 Address Translation (NAT). These technologies require expert assistance to set up correctly and  
 2459 should be avoided in a home network as they most often result in breakage of constructs like  
 2460 routing, naming and discovery services.

2461 The multi-segment ecosystem OCF addresses will not only cause a proliferation of new devices  
 2462 and associated routers, but also new services introducing additional edge routers. All these new  
 2463 requirements require advance architectural constructs to address complex network topologies like  
 2464 the one shown in Figure 14.



2465

2466 **Figure 14 – High Level Network & Connectivity Architecture**

2467 In terms of IETF RFC 6434, IPv6 nodes assume either a router or host role. Nodes may further  
 2468 implement various specializations of those roles:

- 2469 – A Router may implement Customer Edge Router capabilities as defined in IETF RFC 7084.
- 2470 – Nodes limited in processing power, memory, non-volatile storage or transmission capacity  
 2471 requires special IP adaptation layers (6LoWPAN) and/or dedicated routing protocols (RPL).  
 2472 Examples include devices transmitting over low power physical layer like IEEE 802.14.5, ITU  
 2473 G9959, Bluetooth Low Energy, DECT Ultra Low Energy, and Near Field Communication (NFC).

2474 – A node may translate and route messaging between IPv6 and non-IPv6 networks.

### 2475 **9.3 IPv6 network layer requirements**

#### 2476 **9.3.1 Introduction**

2477 Projections indicate that many 10s of billions of new IoT endpoints and related services will be  
2478 brought online in the next few years. These endpoint's capabilities will span from battery powered  
2479 nodes with limited compute, storage, and bandwidth to more richly resourced devices operating  
2480 over Ethernet and WiFi links.

2481 Internet Protocol version 4 (IPv4), deployed some 30 years ago, has matured to support a wide  
2482 variety of applications such as Web browsing, email, voice, video, and critical system monitoring  
2483 and control. However, the capabilities of IPv4 are at the point of exhaustion, not the least of which  
2484 is that available address space has been consumed.

2485 The IETF long ago saw the need for a successor to IPv4, thus the development of IPv6. OCF  
2486 recommends IPv6 at the network layer. Amongst the reasons for IPv6 recommendations are:

- 2487 – Larger address space. Side-effect: greatly reduce the need for NATs.
- 2488 – More flexible addressing architecture. Multiple addresses and types per interface: Link-local,  
2489 ULA, GUA, variously scoped Multicast addresses, etc. Better ability to support multi-homed  
2490 networks, better re-numbering capability, etc.
- 2491 – More capable auto configuration capabilities: DHCPv6, SLAAC, Router Discovery, etc.
- 2492 – Technologies enabling IP connectivity on constrained nodes are based upon IPv6.
- 2493 – All major consumer operating systems (iOS, Android, Windows, Linux) are already IPv6 enabled.
- 2494 – Major Service Providers around the globe are deploying IPv6.

#### 2495 **9.3.2 IPv6 node requirements**

##### 2496 **9.3.2.1 Introduction**

2497 In order to ensure network layer services interoperability from node to node, mandating a common  
2498 network layer across all nodes is vital. The protocol should enable the network to be: secure,  
2499 manageable, and scalable and to include constrained and self-organizing meshed nodes. OCF  
2500 mandates IPv6 as the common network layer protocol to ensure interoperability across all Devices.  
2501 More capable Devices may also include additional protocols creating multiple-stack Devices. The  
2502 remainder of this clause will focus on interoperability requirements for IPv6 hosts, IPv6 constrained  
2503 hosts and IPv6 routers. The various protocol translation permutations included in multi-stack  
2504 gateway devices may be addresses in subsequent addendums of this document.

##### 2505 **9.3.2.2 IP Layer**

2506 An IPv6 node shall support IPv6 and it shall conform to the requirements as specified in  
2507 IETF RFC 6434.

## 2508 **10 OCF Endpoint**

### 2509 **10.1 OCF Endpoint definition**

2510 The specific definition of an OCF Endpoint depends on the Transport Protocol Suite being used.  
2511 For the example of CoAP over UDP over IPv6, the OCF Endpoint is identified by an IPv6 address  
2512 and UDP port number.

2513 Each Device shall associate with at least one OCF Endpoint with which it can exchange request  
2514 and response messages. When a message is sent to an OCF Endpoint, it shall be delivered to the  
2515 Device which is associated with the OCF Endpoint. When a request message is delivered to an  
2516 OCF Endpoint, path component is enough to locate the target Resource.

2517 A Device can be associated with multiple OCF Endpoints. For example, an Device can have several  
2518 IP addresses or port numbers or support both CoAP and HTTP transfer protocol. Different  
2519 Resources in an Device may be accessed with the same OCF Endpoint or need different ones.  
2520 Some Resources may use one OCF Endpoint and others a different one. It depends on an  
2521 implementation.

2522 On the other hand, an OCF Endpoint can be shared among multiple Devices, only when there is a  
2523 way to clearly designate the target Resource with request URI. For example, when multiple CoAP  
2524 servers use uniquely different URI paths for all their hosted Resources, and the CoAP  
2525 implementation demultiplexes by path, they can share the same CoAP OCF Endpoint. However,  
2526 this is not possible in this version of the document, because a pre-determined URI (e.g. "/oic/d") is  
2527 mandatory for some mandatory Resources (e.g. "oic.wk.d").

## 2528 **10.2 OCF Endpoint information**

### 2529 **10.2.1 Introduction**

2530 OCF Endpoint is represented by OCF Endpoint information which consists of two items of key-  
2531 value pair, "ep" and "pri".

### 2532 **10.2.2 "ep"**

2533 "ep" represents Transport Protocol Suite and OCF Endpoint Locator specified as follows:

2534 – *Transport Protocol Suite* - a combination of protocols (e.g. CoAP + UDP + IPv6) with which  
2535 request and response messages can be exchanged for RESTful transaction (i.e. CRUDN). A  
2536 Transport Protocol Suite shall be indicated by a URI scheme name. All scheme names  
2537 supported by this document are IANA registered, these are listed in Table 19. A vendor may  
2538 also make use of a non-IANA registered scheme name for their own use (e.g.  
2539 "com.example.foo"), this shall follow the syntax for such scheme names defined by  
2540 IETF RFC 7595. The behaviour of a vendor-defined scheme name is undefined by this  
2541 document. All OCF defined Resource Types when exposing OCF Endpoint Information in an  
2542 "eps" (see 10.2.4) shall include at least one "ep" with a Transport Protocol Suite as defined in  
2543 Table 19.

2544 – *OCF Endpoint Locator* – an address (e.g. IPv6 address + Port number) or an indirect identifier  
2545 (e.g., DNS name) resolvable to an IP address, through which a message can be sent to the  
2546 OCF Endpoint and in turn associated Device. The OCF Endpoint Locator for "coap" and "coaps"  
2547 shall be specified as "IP address: port number". The OCF Endpoint Locator for "coap+tcp" or  
2548 "coaps+tcp" shall be specified as "IP address: port number" or "DNS name: port number" or  
2549 "DNS name" such that the DNS name shall be resolved to a valid IP address for the target  
2550 Resource with a name resolution service (i.e., DNS). For the 3rd case, when the port number  
2551 is omitted, the default port "5683" (and "5684") shall be assumed for "coap+tcp" (and for  
2552 "coaps+tcp") scheme respectively as defined in IETF RFC 8323. Temporary addresses should  
2553 not be used because OCF Endpoint Locators are for the purpose of accepting incoming  
2554 sessions, whereas temporary addresses are for initiating outgoing sessions (IETF RFC 4941).  
2555 Moreover, its inclusion in "/oic/res" can cause a privacy concern (IETF RFC 7721).

2556 "ep" shall have as its value a URI (as specified in IETF RFC 3986) with the scheme component  
2557 indicating Transport Protocol Suite and the authority component indicating the OCF Endpoint  
2558 Locator.

2559 An "ep" example for "coap" and "coaps" is as illustrated:

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

2560 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"

```

2561 The current list of "ep" with corresponding Transport Protocol Suite is shown in Table 19:

2562 **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"

2563

2564 **10.2.3 "pri"**

2565 When there are multiple OCF Endpoints, "pri" indicates the priority among them.

2566 "pri" shall be represented as a positive integer (e.g. "pri": 1) and the lower the value, the higher the  
2567 priority.

2568 The default "pri" value is 1, i.e. when "pri" is not present, it shall be equivalent to "pri": 1.

2569 **10.2.4 OCF Endpoint information in "eps" Parameter**

2570 To carry OCF Endpoint information, a new Link Parameter "eps" is defined in 7.8.2.2.5. "eps" has  
2571 an array of items as its value and each item represents OCF Endpoint information with two key-  
2572 value pairs, "ep" and "pri", of which "ep" is mandatory and "pri" is optional.

2573 OCF Endpoint Information in an "eps" Parameter is valid for the target Resource of the Link, i.e.,  
2574 the Resource referred by "href" Parameter. OCF Endpoint information in an "eps" Parameter may  
2575 be used to access other Resources on the Device, but such access is not guaranteed.

2576 A Client may resolve the "ep" value to an IP address for the target Resource, i.e., the address to  
2577 access the Device which hosts the target Resource. A valid (transfer protocol) URI for the target  
2578 Resource can be constructed with the scheme, host and port components from the "ep" value and  
2579 the "path" component from the "href" value.

2580 Links with an "eps":

```

2581 {
2582   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9 ",
2583   "href": "/myLightSwitch",
2584   "rt": ["oic.r.switch.binary"],
2585   "if": ["oic.if.a", "oic.if.baseline"],

```

```

2586     "p": {"bm": 3},
2587     "eps": [
2588         {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
2589         {"ep": "coaps://[fe80::b1d6]:1122"}
2590     ]
2591 }
2592
2593 {
2594     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2595     "href": "/myTemperature",
2596     "rt": ["oic.r.temperature"],
2597     "if": ["oic.if.a", "oic.if.baseline"],
2598     "p": {"bm": 3},
2599     "eps": [
2600         {"ep": "coap+tcp://foo.bar.com", "pri": 2},
2601         {"ep": "coaps+tcp://foo.bar.com:1122"}
2602     ]
2603 }

```

2604 In the previous example, "anchor" represents the hosting Device, "href", target Resource and "eps"  
2605 the two OCF Endpoints for the target Resource. The (fully-qualified) URIs for the target Resource  
2606 are as illustrated:

```

2607 coap://[fe80::b1d6]:1111/myLightSwitch
2608 coaps://[fe80::b1d6]:1122/myLightSwitch
2609 coap+tcp://foo.bar.com:5683/myTemperature

```

2610 coaps+tcp://foo.bar.com:1122/myTemperature If the target Resource of a Link requires a secure  
2611 connection (e.g. CoAPS), "eps" Parameter shall be used to indicate the necessary information (e.g.  
2612 port number) in OCF 1.0 payload. For optional backward compatibility with OIC 1.1, the "sec" and  
2613 "port" shall only be used in OIC 1.1 payload.

## 2614 10.3 OCF Endpoint discovery

### 2615 10.3.1 Introduction

2616 OCF Endpoint discovery is defined as the process for a Client to acquire the OCF Endpoint  
2617 information for Device or Resource.

### 2618 10.3.2 Implicit discovery

2619 If a Device is the source of a CoAP message (e.g. "/oic/res" response), the source IP address and  
2620 port number may be combined to form the OCF Endpoint Locator for the Device. Along with a  
2621 "coap" scheme and default "pri" value, OCF Endpoint information for the Device may be constructed.

2622 In other words, a "/oic/res" response message with CoAP may implicitly carry the OCF Endpoint  
2623 information of the responding Device and in turn all the hosted Resources, which may be accessed  
2624 with the same transfer protocol of CoAP. In the absence of an "eps" Parameter, a Client shall be  
2625 able to utilize implicit discovery to access the target Resource.

### 2626 10.3.3 Explicit discovery with "/oic/res" response

2627 OCF Endpoint information may be explicitly indicated with the "eps" Parameter of the Links in  
2628 "/oic/res".

2629 As in 10.3.2, an "/oic/res" response may implicitly indicate the OCF Endpoint information for some  
2630 Resources hosted by the responding Device. However implicit discovery, i.e., inference of OCF  
2631 Endpoint information from CoAP response message, may not work for some Resources on the  
2632 same Device. For example, some Resources may allow only secure access via CoAPS which  
2633 requires the "eps" Parameter to indicate the port number. Moreover "/oic/res" may expose a target  
2634 Resource which belongs to another Device.



2635 When the OCF Endpoint for a target Resource of a Link cannot be implicitly inferred, the "eps"  
2636 Parameter shall be included to provide explicit OCF Endpoint information with which a Client can  
2637 access the target Resource. In the presence of the "eps" Parameter, a Client shall be able to utilize  
2638 it to access the target Resource. For "coap" and "coaps", a Client may use the IP address in the  
2639 "ep" value in the "eps" Parameter to access the target Resource. For "coap+tcp" and "coaps+tcp",  
2640 a Client may use the IP address in the "eps" Parameter or resolve the DNS name in the "eps"  
2641 Parameter to acquire a valid IP address for the target Resource. If "eps" Parameter omits the port  
2642 number, then the default port "5683" (and "5684") shall be assumed for "coap+tcp" (and  
2643 "coaps+tcp") scheme as defined in IETF RFC 8323. To access the target Resource of a Link, a  
2644 Client may use the "eps" Parameter in the Link, if it is present and fall back on implicit discovery if  
2645 not.

2646 This applies to the case of "/oic/res" for a Resource Directory or Bridge Device which usually carries  
2647 the Links for Resources which another Device hosts.

2648 This is an example of an "/oic/res" response from a Bridge Device with two Bridged Devices, having  
2649 the "eps" Parameter in Links.

```
2650 [
2651   {
2652     "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2653     "href": "/oic/res",
2654     "rel": "self",
2655     "rt": ["oic.wk.res"],
2656     "if": ["oic.if.ll", "oic.if.baseline"],
2657     "p": {"bm": 3},
2658     "eps": [
2659       {"ep": "coap://[2001:db8:a::b1d4]:55555"},
2660       {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2661     ]
2662   },
2663   {
2664     "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2665     "href": "/oic/d",
2666     "rt": ["oic.wk.d", "oic.d.bridge"],
2667     "if": ["oic.if.r", "oic.if.baseline"],
2668     "p": {"bm": 3},
2669     "eps": [
2670       {"ep": "coap://[2001:db8:a::b1d4]:55555"},
2671       {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2672     ]
2673   },
2674   {
2675     "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2676     "href": "/oic/p",
2677     "rt": ["oic.wk.p"],
2678     "if": ["oic.if.r", "oic.if.baseline"],
2679     "p": {"bm": 3},
2680     "eps": [
2681       {"ep": "coap://[2001:db8:a::b1d4]:55555"},
2682       {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2683     ]
2684   },
2685   {
2686     "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2687     "href": "/mySecureMode",
2688     "rt": ["oic.r.securemode"],
2689     "if": ["oic.if.rw", "oic.if.baseline"],
2690     "p": {"bm": 3},
2691     "eps": [
```

```

2693     {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2694   ]
2695 },
2696 {
2697   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2698   "href": "/oic/sec/doxm",
2699   "rt": ["oic.r.doxm"],
2700   "if": ["oic.if.baseline"],
2701   "p": {"bm": 1},
2702   "eps": [
2703     {"ep": "coap://[2001:db8:a::b1d4]:55555"},
2704     {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2705   ]
2706 },
2707 {
2708   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2709   "href": "/oic/sec/pstat",
2710   "rt": ["oic.r.pstat"],
2711   "if": ["oic.if.baseline"],
2712   "p": {"bm": 1},
2713   "eps": [
2714     {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2715   ]
2716 },
2717 {
2718   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2719   "href": "/oic/sec/cred",
2720   "rt": ["oic.r.cred"],
2721   "if": ["oic.if.baseline"],
2722   "p": {"bm": 1},
2723   "eps": [
2724     {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2725   ]
2726 },
2727 {
2728   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2729   "href": "/oic/sec/acl2",
2730   "rt": ["oic.r.acl2"],
2731   "if": ["oic.if.baseline"],
2732   "p": {"bm": 1},
2733   "eps": [
2734     {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2735   ]
2736 },
2737 {
2738   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
2739   "href": "/myIntrospection",
2740   "rt": ["oic.wk.introspection"],
2741   "if": ["oic.if.r", "oic.if.baseline"],
2742   "p": {"bm": 3},
2743   "eps": [
2744     {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
2745   ]
2746 },
2747 {
2748   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2749   "href": "/oic/res",
2750   "rt": ["oic.wk.res"],
2751   "if": ["oic.if.ll", "oic.if.baseline"],
2752   "p": {"bm": 3},
2753   "eps": [
2754     {"ep": "coap://[2001:db8:a::b1d4]:66666"},
2755     {"ep": "coaps://[2001:db8:a::b1d4]:22222"}

```

```

2756     ]
2757   },
2758   {
2759     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2760     "href": "/oic/d",
2761     "rt": ["oic.wk.d", "oic.d.light", "oic.d.virtual"],
2762     "if": ["oic.if.r", "oic.if.baseline"],
2763     "p": {"bm": 3},
2764     "eps": [
2765       {"ep": "coap://[2001:db8:a::b1d4]:66666"},
2766       {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
2767     ]
2768   },
2769   {
2770     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2771     "href": "/oic/p",
2772     "rt": ["oic.wk.p"],
2773     "if": ["oic.if.r", "oic.if.baseline"],
2774     "p": {"bm": 3},
2775     "eps": [
2776       {"ep": "coap://[2001:db8:a::b1d4]:66666"},
2777       {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
2778     ]
2779   },
2780   {
2781     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2782     "href": "/myLight",
2783     "rt": ["oic.r.switch.binary"],
2784     "if": ["oic.if.a", "oic.if.baseline"],
2785     "p": {"bm": 3},
2786     "eps": [
2787       {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
2788     ]
2789   },
2790   {
2791     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2792     "href": "/oic/sec/doxm",
2793     "rt": ["oic.r.doxm"],
2794     "if": ["oic.if.baseline"],
2795     "p": {"bm": 1},
2796     "eps": [
2797       {"ep": "coap://[2001:db8:a::b1d4]:66666"},
2798       {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
2799     ]
2800   },
2801   {
2802     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2803     "href": "/oic/sec/pstat",
2804     "rt": ["oic.r.pstat"],
2805     "if": ["oic.if.baseline"],
2806     "p": {"bm": 1},
2807     "eps": [
2808       {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
2809     ]
2810   },
2811   {
2812     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2813     "href": "/oic/sec/cred",
2814     "rt": ["oic.r.cred"],
2815     "if": ["oic.if.baseline"],
2816     "p": {"bm": 1},
2817     "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:22222"}]
2818   },

```

```

2819 {
2820   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2821   "href": "/oic/sec/acl2",
2822   "rt": ["oic.r.acl2"],
2823   "if": ["oic.if.baseline"],
2824   "p": {"bm": 1},
2825   "eps": [
2826     {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
2827   ]
2828 },
2829 {
2830   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
2831   "href": "/myLightIntrospection",
2832   "rt": ["oic.wk.introspection"],
2833   "if": ["oic.if.r", "oic.if.baseline"],
2834   "p": {"bm": 3},
2835   "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:22222"}]
2836 },
2837 {
2838   "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
2839   "href": "/oic/res",
2840   "rt": ["oic.wk.res"],
2841   "if": ["oic.if.ll", "oic.if.baseline"],
2842   "p": {"bm": 3},
2843   "eps": [
2844     {"ep": "coap://[2001:db8:a::b1d4]:77777"},
2845     {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2846   ]
2847 },
2848 {
2849   "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
2850   "href": "/oic/d",
2851   "rt": ["oic.wk.d", "oic.d.fan", "oic.d.virtual"],
2852   "if": ["oic.if.r", "oic.if.baseline"],
2853   "p": {"bm": 3},
2854   "eps": [
2855     {"ep": "coap://[2001:db8:a::b1d4]:77777"},
2856     {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2857   ]
2858 },
2859 {
2860   "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
2861   "href": "/oic/p",
2862   "rt": ["oic.wk.p"],
2863   "if": ["oic.if.r", "oic.if.baseline"],
2864   "p": {"bm": 3},
2865   "eps": [
2866     {"ep": "coap://[2001:db8:a::b1d4]:77777"},
2867     {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2868   ]
2869 },
2870 {
2871   "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
2872   "href": "/myFan",
2873   "rt": ["oic.r.switch.binary"],
2874   "if": ["oic.if.a", "oic.if.baseline"],
2875   "p": {"bm": 3},
2876   "eps": [
2877     {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2878   ]
2879 },
2880 {
2881   "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",

```

```

2882     "href": "/oic/sec/doxm",
2883     "rt": ["oic.r.doxm"],
2884     "if": ["oic.if.baseline"],
2885     "p": {"bm": 1},
2886     "eps": [
2887       {"ep": "coap://[2001:db8:a::b1d4]:7777"},
2888       {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2889     ],
2890   },
2891   {
2892     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
2893     "href": "/oic/sec/pstat",
2894     "rt": ["oic.r.pstat"],
2895     "if": ["oic.if.baseline"],
2896     "p": {"bm": 1},
2897     "eps": [
2898       {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2899     ]
2900   },
2901   {
2902     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
2903     "href": "/oic/sec/cred",
2904     "rt": ["oic.r.cred"],
2905     "if": ["oic.if.baseline"],
2906     "p": {"bm": 1},
2907     "eps": [
2908       {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2909     ]
2910   },
2911   {
2912     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
2913     "href": "/oic/sec/acl2",
2914     "rt": ["oic.r.acl2"],
2915     "if": ["oic.if.baseline"],
2916     "p": {"bm": 1},
2917     "eps": [
2918       {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2919     ]
2920   },
2921   {
2922     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
2923     "href": "/myFanIntrospection",
2924     "rt": ["oic.wk.introspection"],
2925     "if": ["oic.if.r", "oic.if.baseline"],
2926     "p": {"bm": 3},
2927     "eps": [
2928       {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
2929     ]
2930   }
2931 ]
2932

```

2933 The exact format of the "/oic/res" response and a way for a Client to acquire a "/oic/res" response  
2934 message is specified in Annex D and 11.3.5 respectively.

#### 2935 **10.4 CoAP based OCF Endpoint discovery**

2936 The following describes CoAP based OCF Endpoint discovery:

- 2937 – Devices shall join the *All OCF Nodes* multicast groups (as defined in [IANA IPv6 Multicast  
2938 Address Space Registry]) with scopes 2, 3, and 5 (i.e., ff02::158, ff03::158 and ff05::158) and  
2939 shall listen on the port 5683. For compliance to IETF RFC 7252 a Device may additionally join  
2940 the *All CoAP Nodes* multicast groups.

- 2941 – Clients intending to discover Resources shall join the multicast groups as defined in a).
- 2942 – Devices shall expose "/oic/res" via an unsecured OCF Endpoint.
- 2943 – Clients shall send discovery requests (GET request) to the *All OCF Nodes* multicast group
- 2944 address with scope 2 ("ff02::158") at port "5683". The requested URI shall be "/oic/res". For
- 2945 compliance to IETF RFC 7252 a Client may additionally send to the *All CoAP Nodes* multicast
- 2946 groups.
- 2947 – If the discovery request is intended for a specific Resource Type, the query parameter "rt" shall
- 2948 be included in the request (see 6.2.2) with its value set to the desired Resource Type. Only
- 2949 Devices hosting the Resource Type shall respond to the discovery request.
- 2950 – When the "rt" query parameter is omitted, all Devices shall respond to the discovery request.
- 2951 – Handling of multicast requests shall be as described in clause 8 of IETF RFC 7252 and clause
- 2952 4.1 in IETF RFC 6690.
- 2953 – Devices which receive the request shall respond using CBOR payload encoding. A Device shall
- 2954 indicate support for CBOR payload encoding for multicast discovery as described in 12.4.

2955 **11 Functional interactions**

2956 **11.1 Introduction**

2957 The functional interactions between a Client and a Server are described in 11.2 through 11.9  
 2958 respectively. The functional interactions use CRUDN messages (clause 8) and include Discovery,  
 2959 Notification, and Device management. These functions require support of core defined Resources  
 2960 as defined in Table 20.

2961 **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

2962

2963 **11.2 Onboarding, Provisioning and Configuration**

2964 Onboarding and Provisioning are fully defined by the ISO/IEC 30118-2:2018.

2965 Should a Device support Client update of configurable information it shall do so via exposing an  
 2966 oic.wk.con Core Resource (Table 21) in "/oic/res".

2967

**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
--------------------	------------------------	----------------	-------------	---	---------------

2968

2969 Table 22 defines the "oic.wk.con" Resource Type.

2970

**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	Deg rees	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.
--	--	--	--	--	--	--	---

2971

2972 Table 23 defines the "oic.wk.con.p" Resource Type.

2973

**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,            [{"language": "en", "value": "Dave's Laptop"}]</p>

2974

### 2975 11.3 Resource discovery

#### 2976 11.3.1 Introduction

2977 Discovery is a function which enables OCF Endpoint discovery as well as Resource based  
 2978 discovery. OCF Endpoint discovery is described in detail in clause 10. This clause mainly describes  
 2979 the Resource based discovery.

#### 2980 11.3.2 Resource based discovery: mechanisms

##### 2981 11.3.2.1 Overview

2982 As part of discovery, a Client may find appropriate information about other OCF peers. This  
 2983 information could be instances of Resources, Resource Types or any other information represented  
 2984 in the Resource model that an OCF peer would want another OCF peer to discover.

2985 At the minimum, Resource based discovery uses the following:

- 2986 – A Resource to enable discovery shall be defined. The representation of that Resource shall  
 2987 contain the information that can be discovered.
- 2988 – The Resource to enable discovery shall be specified and commonly known a-priori. A Device  
 2989 for hosting the Resource to enable discovery shall be identified.
- 2990 – A mechanism and process to publish the information that needs to be discovered with the  
 2991 Resource to enable discovery.
- 2992 – A mechanism and process to access and obtain the information from the Resource to enable  
 2993 discovery. A query may be used in the request to limit the returned information.
- 2994 – A scope for the publication.
- 2995 – A scope for the access.
- 2996 – A policy for visibility of the information.



2997 Depending on the choice of the base aspects, the Framework defines three Resource based  
2998 discovery mechanisms:

- 2999 – Direct discovery, where the Resources are published locally at the Device hosting the  
3000 Resources and are discovered through peer inquiry.
- 3001 – Indirect discovery, where Resources are published at a third party assisting with the discovery  
3002 and peers publish and perform discovery against the Resource to enable discovery on the  
3003 assisting 3<sup>rd</sup> party.
- 3004 – Advertisement discovery, where the Resource to enable discovery is hosted local to the initiator  
3005 of the discovery inquiry but remote to the Devices that are publishing discovery information.

3006 A Device shall support direct discovery.

### 3007 **11.3.2.2 Direct discovery**

3008 In direct discovery,

- 3009 – The Device that is providing the information shall host the Resource to enable discovery.
- 3010 – The Device publishes the information available for discovery with the local Resource to enable  
3011 discovery (i.e. local scope).
- 3012 – Clients interested in discovering information about this Device shall issue RETRIEVE requests  
3013 directly to the Resource. The request may be made as a unicast or multicast. The request may  
3014 be generic or may be qualified or limited by using appropriate queries in the request.
- 3015 – The Server Device that receives the request shall send a response with the discovered  
3016 information directly back to the requesting Client Device.
- 3017 – The information that is included in the request is determined by the policies set for the Resource  
3018 to be discovered locally on the responding Device.

### 3019 **11.3.2.3 Indirect discovery of Resources (Resource Directory based discovery)**

3020 In indirect discovery the information about the resource to be discovered is hosted on a Server that  
3021 is not hosting the Resource. See 11.3.6 for details on Resource Directory based discovery.

3022 In indirect discovery:

- 3023 – The Resource to be discovered is hosted on a Device (e.g., an OCF Light Device) that is neither  
3024 the Client initiating the discovery nor the Device (e.g., Resource Directory (RD)) that is  
3025 providing the information to be discovered. This Device (e.g., RD) assisting indirect discovery  
3026 may use the same Resource to provide discovery for multiple agents looking to discover and  
3027 for multiple agents with information to be discovered.
- 3028 – The Device to be discovered or with information to discover, acting as the publishing Device,  
3029 requests to publish that information with Resource to be discovered on a different Device. The  
3030 policies on the information shared including the lifetime/validity are specified by the publishing  
3031 Device acting as the Resource Directory. The publishing Device may modify these policies as  
3032 required, e.g., the Resource Directory may shorten the lifetime/validity upon granting the  
3033 publishing request.
- 3034 – The Client doing the discovery may send a unicast discovery request to the Device hosting the  
3035 discovery information or send a multicast request that shall be monitored and responded to by  
3036 the Device. In both cases, the Device hosting the discovery information is acting on behalf of  
3037 the publishing Device.
- 3038 – The discovery policies may be set by the Device hosting the discovery information or by the  
3039 party that is publishing the information to be discovered. The discovery information that is  
3040 returned in the discovery response shall adhere to the policies that are in effect at the time of  
3041 the request.

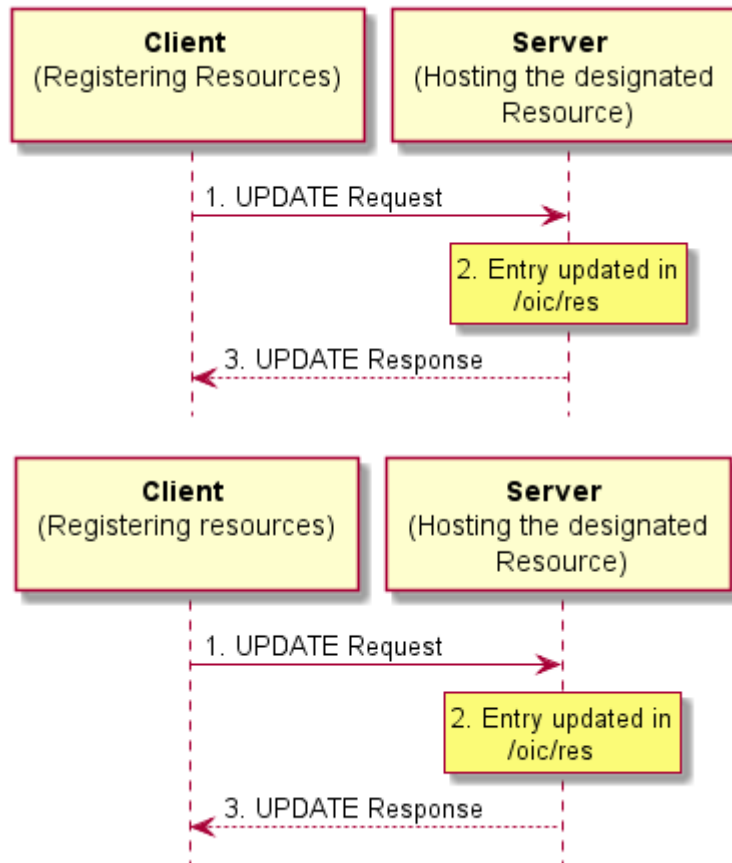
3042 **11.3.2.4 Advertisement Discovery**

3043 In advertisement discovery:

- 3044 – The Resource to enable discovery is hosted local to the Device that is initiating the discovery  
3045 request (Client). The Resource to enable discovery may be a Core Resource or discovered as  
3046 part of a bootstrap.
- 3047 – The request could be an implementation dependent lookup or be a local RETRIEVE request  
3048 against the Resource that enables discovery.
- 3049 – The Device with information to be discovered shall publish the appropriate information to the  
3050 Resource that enables discovery.
- 3051 – The publishing Device is responsible for the published information. The publishing Device may  
3052 UPDATE the information at the resource to enable discovery based on its needs by sending  
3053 additional publication requests. The policies on the information that is discovered including  
3054 lifetime is determined by the publishing Device.

3055 **11.3.3 Resource based discovery: Information publication process**

3056 The mechanism to publish information with the Resource to enable discovery can be done either  
3057 locally or remotely. The publication process is depicted in Figure 15. The Device which has  
3058 discovery information to publish shall a) either update the Resource that enables discovery if  
3059 hosted locally or b) issue an UPDATE request with the information to the Device which hosts the  
3060 Resource that enables discovery. The Device hosting the Resource to enable discovery  
3061 adds/updates the Resource to enable discovery with the provided information and then responds  
3062 to the Device which has requested the publication of the Resource with an UPDATE response.



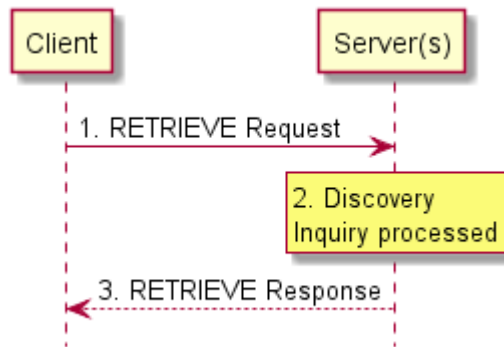
3063

3064

3065 **Figure 15 – Resource based discovery: Information publication process**

3066 **11.3.4 Resource based discovery: Finding information**

3067 The discovery process (Figure 16) is initiated as a RETRIEVE request to the Resource to enable  
3068 discovery. The request may be sent to a single Device (as in a Unicast) or to multiple Devices (as  
3069 in Multicast). The specific mechanisms used to do Unicast or Multicast are determined by the  
3070 support in the data connectivity layer. The response to the request has the information to be  
3071 discovered based on the policies for that information. The policies can determine which information  
3072 is shared, when and to which requesting agent. The information that can be discovered can be  
3073 Resources, types, configuration and many other standards or custom aspects depending on the  
3074 request to appropriate Resource and the form of request. Optionally the requester may narrow the  
3075 information to be returned in the request using query parameters in the URI query.



3076

3077 **Figure 16 – Resource based discovery: Finding information**

3078

3079

*Discovery Resources*

3080 The following Core Resources shall be implemented on all Devices to support discovery:

- 3081 – "/oic/res" for discovery of Resources.
- 3082 – "/oic/p" for discovery of Platform.
- 3083 – "/oic/d" for discovery of Device information.

3084 Devices shall expose each of "/oic/res", "/oic/d", and "/oic/p" via an unsecured OCF Endpoint.  
3085 Further details for these mandatory Core Resources are described in Table 24.

3086 *Platform Resource*

3087 The OCF recognizes that more than one instance of Device may be hosted on a single Platform.  
3088 Clients need a way to discover and access the information on the Platform. The Core Resource,  
3089 "/oic/p" exposes Platform specific Properties. All instances of Device on the same Platform shall  
3090 have the same values of any Properties exposed (i.e. a Device may choose to expose optional  
3091 Properties within "/oic/p" but when exposed the value of that Property should be the same as the  
3092 value of that Property on all other Devices on that Platform).

3093 *Device Resource*

3094 The Device Resource shall have the pre-defined URI "/oic/d". The Resource "/oic/d" exposes the  
3095 Properties pertaining to a Device as defined in Table 24. The Properties exposed are determined  
3096 by the specific instance of Device and defined by the Resource Type(s) of "/oic/d" on that Device.  
3097 Since all the Resource Types of "/oic/d" are not known a priori, the Resource Type(s) of "/oic/d"  
3098 shall be determined by discovery through the Core Resource "/oic/res". The Device Resource  
3099 "/oic/d" shall have a default Resource Type that helps in bootstrapping the interactions with this  
3100 Device (the default type is described in Table 24).

3101 *Protocol indication*

3102 A Device may need to support different messaging protocols depending on requirements for  
 3103 different vertical domain profiles. For example, a Smart Home profile may use CoAP and an  
 3104 Industrial profile may use DDS. To enable interoperability, a Device uses the protocol indication to  
 3105 indicate the transport protocols they support and can communicate over.

3106 **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

3107 Table 25 defines "oic.wk.res" Resource Type.

3108 **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.
-------	---------	-------	-----------	-----	---	-----	---

3109

3110 A Device shall support CoAP based discovery as the baseline discovery mechanism (see 10.4).

3111 The "/oic/res" shall list all Resources that are indicated as discoverable (see 11.3). Also the  
3112 following architecture Resource Types shall be listed:

- 3113 – Introspection Resource indicated with an "rt" value of "oic.wk.introspection".
- 3114 – "/oic/p" indicated with an "rt" value of "oic.wk.p".
- 3115 – "/oic/d" indicated with an "rt" value of "oic.wk.d"
- 3116 – "/oic/sec/doxm" indicated with an "rt" value of "oic.r.doxm" as defined in ISO/IEC 30118-2:2018.
- 3117 – "/oic/sec/pstat" indicated with an "rt" value of "oic.r.pstat" as defined in ISO/IEC 30118-2:2018.
- 3118 – "/oic/sec/acl2" indicated with an "rt" value of "oic.r.acl2" as defined in ISO/IEC 30118-2:2018.
- 3119 – "/oic/sec/cred" indicated with an "rt" value of "oic.r.cred" as defined in ISO/IEC 30118-2:2018.

3120 Conditionally required:

- 3121 – "/oic/res" with an "rt" value of "oic.wk.res" as self-reference, on the condition that "oic/res" has  
3122 to signal that it is Observable by a Client.

3123 The Introspection Resource is only applicable for Devices that host Vertical Resource Types (e.g.  
3124 "oic.r.switch.binary") or vendor-defined Resource Types. Devices that only host Resources  
3125 required to onboard the Device as a Client do not have to implement the Introspection Resource.

3126 Table 26 provides an OCF registry for protocol schemes.

3127 **Table 26 – Protocol scheme registry**

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

3128

3129 NOTE The discovery of an OCF Endpoint used by a specific protocol is out of scope. The mechanism used by a Client  
3130 to form requests in a different messaging protocol other than discovery is out of scope.

3131 The following applies to the use of "/oic/d":

- 3132 – A Device may choose to expose its Device Type(s) (e.g., refrigerator or A/C or composite of  
3133 multiple Device Types) by adding the Device Type to the list of Resource Types associated with  
3134 "/oic/d".

3135 – For example; "rt" of "/oic/d" becomes ["oic.wk.d", "oic.d.<thing1>", "oic.d.<thing2>"]; where  
 3136 "oic.d.<thing1>" and "oic.d.<thing2>" are defined in another specification such as  
 3137 ISO/IEC 30118-5:2018.

3138 – This implies that the Properties exposed by "/oic/d" are by default the mandatory Properties  
 3139 in Table 27.

3140 – A vertical may choose to extend the list of Properties defined by the Resource Type "oic.wk.d".  
 3141 In that case, the vertical shall assign a new Device Type specific Resource Type ID. The  
 3142 mandatory Properties defined in Table 27 shall always be present.

3143 – A Device may choose to expose a separate, Discoverable Resource with its Resource Type ID  
 3144 set to an OCF defined Device Type. In this case the Resource is equivalent to an instance of  
 3145 "oic.wk.d" and adheres to the definition thereof. As such the Resource shall at a minimum  
 3146 expose the mandatory Properties of "oic.wk.d". In the case where the Resource tagged in this  
 3147 manner is defined to be an instance of a Collection in accordance with 7.8.3 then the Resources  
 3148 that are part of that Collection shall at a minimum include the Resource Types mandated for  
 3149 the Device Type. For example, if a Collection Resource has an "rt" value of ["oic.d.light"], the  
 3150 Collection includes an instance of "oic.r.switch.binary" which is mandatory for an "oic.d.light"  
 3151 as per ISO/IEC 30118-5:2018.

3152 Table 27 "oic.wk.d" Resource Type definition defines the base Resource Type for the "/oic/d"  
 3153 Resource.

3154

**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.2.0.43".
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	"id"	"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.

Ecosystem Name	"econame"	"string"	enum	N/A	R	No	This is the name of ecosystem that a Bridged Device belongs to. If a Device has "oic.d.virtual" as one of Resource Type values ("rt") the Device shall contain this Property, otherwise this Property shall not be included.  This Property has enumeration values: ["BLE", "oneM2M", "UPlus", "Zigbee", "Z-Wave"].
Version of Ecosystem	"ecoversion"	"string"	N/A	N/A	R	No	This is the version of ecosystem that a Bridged Device belongs to. If a Device has "oic.d.virtual" as one of its Resource Type values ("rt") the Device should contain this Property, otherwise this Property shall not be included.

3155 The additional Resource Type(s) of the "/oic/d" Resource are defined by ISO/IEC 30118-5:2018.

3156 Table 28 defines "oic.wk.p" Resource Type.

3157

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

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
<b>Platform ID</b>	"pi"	"uuid"	N/A	N/A	R	Yes	Unique identifier for the physical Platform (UUID); 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.
<b>Manufacturer Name</b>	"mnmn"	"string"	N/A	N/A	R	Yes	Name of manufacturer.
<b>Manufacturer Details Link</b>	"mnml"	"uri"	N/A	N/A	R	No	Reference to manufacturer, represented as a URI.
<b>Model Number</b>	"mnmo"	"string"	N/A	N/A	R	No	Model number as designated by manufacturer.
<b>Date of Manufacture</b>	"mndt"	"date"	N/A	Time	R	No	Manufacturing date of Platform.
<b>Serial number</b>	"mnsel"	"string"	N/A	s	R	No	Serial number of the Platform, may be unique for each Platform of the same model number.
<b>Platform Version</b>	"mnpv"	"string"	N/A	N/A	R	No	Version of Platform – string (defined by manufacturer).
<b>OS Version</b>	"mnos"	"string"	N/A	N/A	R	No	Version of Platform resident OS – string



							(defined by manufacturer).
<b>Hardware Version</b>	"mnhw"	"string"	N/A	N/A	R	No	Version of Platform hardware.
<b>Firmware version</b>	"mnfv"	"string"	N/A	N/A	R	No	Version of Platform firmware.
<b>Support link</b>	"mnsi"	"uri"	N/A	N/A	R	No	URI that points to support information from manufacturer.
<b>SystemTime</b>	"st"	"date-time"	N/A	N/A	R	No	Reference time for the Platform.
<b>Vendor ID</b>	"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.
<b>Network Connectivity Type</b>	"mnct"	"array"	array of integer		R	No	An array of integer where each integer indicates the network connectivity type based on IANAIfType value as defined by IANA ifType-MIB Definitions, e.g., [71, 259] which represents Wi-Fi and Zigbee.

3158 **11.3.5 Resource discovery using "/oic/res"**

3159 Discovery using "/oic/res" is the default discovery mechanism that shall be supported by all Devices  
3160 as follows:

- 3161 – Every Device updates its local "/oic/res" with the Resources that are discoverable (see 7.3.2.2).  
3162 Every time a new Resource is instantiated on the Device and if that Resource is discoverable  
3163 by a remote Device then that Resource is published with the "/oic/res" Resource that is local to  
3164 the Device (as the instantiated Resource).
- 3165 – A Device wanting to discover Resources or Resource Types on one or more remote Devices  
3166 makes a RETRIEVE request to the "/oic/res" on the remote Devices. This request may be sent  
3167 multicast (default) or unicast if only a specific host is to be probed. The RETRIEVE request may  
3168 optionally be restricted using appropriate clauses in the query portion of the request. Queries  
3169 may select based on Resource Types, OCF Interfaces, or Properties.
- 3170 – The query applies to the representation of the Resources. "/oic/res" is the only Resource whose  
3171 representation has "rt". So "/oic/res" is the only Resource that can be used for Multicast  
3172 discovery at the transport protocol layer.
- 3173 – The Device receiving the RETRIEVE request responds with a list of Resources, the Resource  
3174 Type of each of the Resources and the OCF Interfaces that each Resource supports.  
3175 Additionally, information on the policies active on the Resource can also be sent. The policy  
3176 supported includes Observability and discoverability.
- 3177 – The receiving Device may do a deeper discovery based on the Resources returned in the  
3178 request to "/oic/res".

3179 The information that is returned on discovery against "/oic/res" is at the minimum:

- 3180 – The URI (relative or fully qualified URL) of the Resource.
- 3181 – The Resource Type(s) of each Resource. More than one Resource Type may be returned if the  
3182 Resource enables more than one type. To access Resources of multiple types, the specific  
3183 Resource Type that is targeted shall be specified in the request.

3184 – The OCF Interfaces supported by that Resource. Multiple OCF Interfaces may be returned. To  
3185 access a specific OCF Interface that OCF Interface shall be specified in the request. If the OCF  
3186 Interface is not specified, then the Default OCF Interface is assumed.

3187 Different "/oic/res" responses are returned according to requesting Clients, which indicate their  
3188 preference via inclusion or otherwise of an OCF-Accept-Content-Format-Version option.

3189 For Clients that do not include the OCF-Accept-Content-Format-Version option, an "/oic/res"  
3190 response shall use "sec" and "port" to provide the information for an encrypted connection. See  
3191 Annex E for the schema for the Link.

3192 For Clients that do include the OCF-Accept-Content-Format-Version option, an "/oic/res" response  
3193 includes an array of Links to conform to IETF RFC 6690. Each Link shall use an "eps" Parameter  
3194 to provide the information for an encrypted connection and carry "anchor" of the value OCF URI  
3195 where the authority component of <deviceID> indicates the Device hosting the target Resource.

3196 The OpenAPI 2.0 file for discovery using "/oic/res" is described in Annex D the schema that is  
3197 applicable to requesting Clients that do not include an OCF-Accept-Content-Format-Version option  
3198 is described in Annex E. Also refer to clause 10 (OCF Endpoint discovery) for details of Multicast  
3199 discovery using "/oic/res" on a CoAP transport.

3200 For example, a Light Device might return the following to OIC 1.1 Clients:

```
3201 [
3202   {
3203     "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
3204     "links": [
3205       {
3206         "href": "coaps://[fe80::b1d6]:44444/oic/res",
3207         "rel": "self",
3208         "rt": ["oic.wk.res"],
3209         "if": ["oic.if.ll", "oic.if.baseline"],
3210         "p": {"bm": 3}
3211       },
3212       {
3213         "href": "/oic/p",
3214         "rt": ["oic.wk.p"],
3215         "if": ["oic.if.r", "oic.if.baseline"],
3216         "p": {"bm": 3, "sec": true, "port": 11111}
3217       },
3218       {
3219         "href": "/oic/d",
3220         "rt": ["oic.wk.d", "oic.d.light"],
3221         "if": ["oic.if.r", "oic.if.baseline"],
3222         "p": {"bm": 3, "sec": true, "port": 11111}
3223       },
3224       {
3225         "href": "/myLight",
3226         "rt": ["oic.r.switch.binary"],
3227         "if": ["oic.if.a", "oic.if.baseline"],
3228         "p": {"bm": 3, "sec": true, "port": 11111}
3229       }
3230     ]
3231   }
3232 ]
```

3233 The light Device might return the following to Clients that request with the Content Format of  
3234 "application/vnd.ocf+cbor" in Accept Option:

```
3235 [
3236   {
3237     "href": "/oic/res",
```

```

3238     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989/oic/res",
3239     "rel": "self",
3240     "rt": ["oic.wk.res"],
3241     "if": ["oic.if.ll", "oic.if.baseline"],
3242     "p": {"bm": 3},
3243     "eps": [{"ep": "coap://[fe80::b1d6]:44444"}]
3244   },
3245   {
3246     "href": "/oic/p",
3247     "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
3248     "rt": ["oic.wk.p"],
3249     "if": ["oic.if.r", "oic.if.baseline"],
3250     "p": {"bm": 3},
3251     "eps": [{"ep": "coap://[fe80::b1d6]:44444"},
3252             {"ep": "coaps://[fe80::b1d6]:11111"}]
3253   },
3254 },
3255 {
3256   "href": "/oic/d",
3257   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
3258   "rt": ["oic.wk.d", "oic.d.light"],
3259   "if": ["oic.if.r", "oic.if.baseline"],
3260   "p": {"bm": 3},
3261   "eps": [{"ep": "coap://[fe80::b1d6]:44444"},
3262           {"ep": "coaps://[fe80::b1d6]:11111"}]
3263 },
3264 },
3265 {
3266   "href": "/myLight",
3267   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
3268   "rt": ["oic.r.switch.binary"],
3269   "if": ["oic.if.a", "oic.if.baseline"],
3270   "p": {"bm": 3},
3271   "eps": [{"ep": "coap://[fe80::b1d6]:44444"},
3272           {"ep": "coaps://[fe80::b1d6]:11111"}]
3273 },
3274 }
3275 ]

```

3276 After performing discovery using "/oic/res", Clients may discover additional details about Server by  
3277 performing discovery using "/oic/p", "/oic/rts" etc. If a Client already knows about Server it may  
3278 discover using other Resources without going through the discovery of "/oic/res".

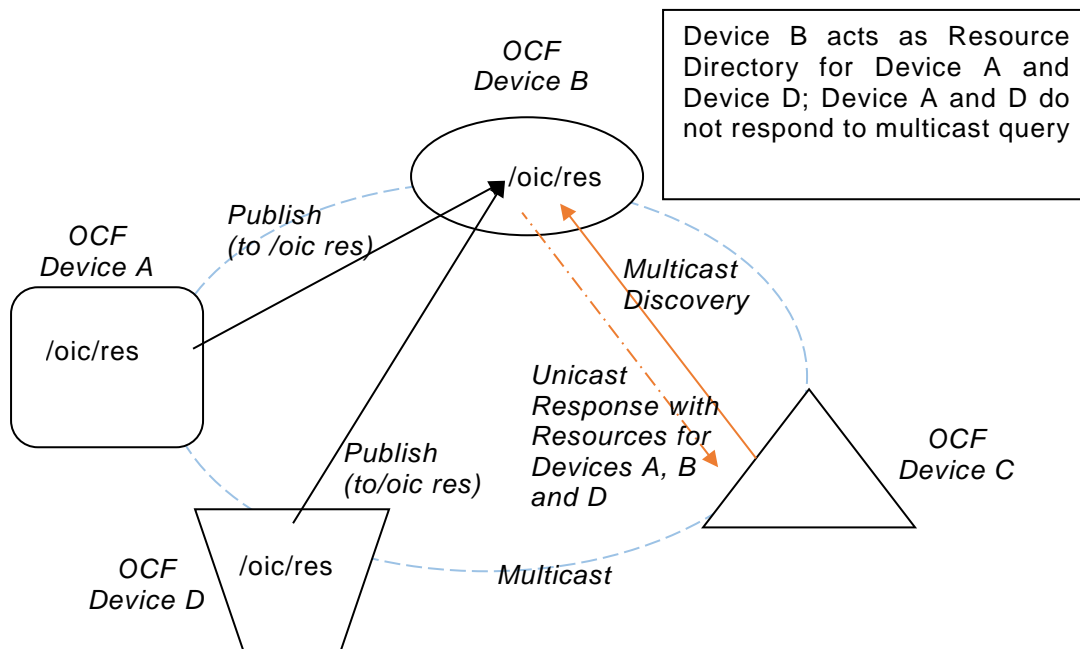
### 3279 **11.3.6 Resource Directory (RD) based discovery**

#### 3280 **11.3.6.1 Introduction**

##### 3281 **11.3.6.1.1 Indirect discovery for lookup of the Resources**

3282 Direct discovery is the mechanism used currently to find Resources in the network. When needed,  
3283 Resources are queried at a particular Device directly or a multicast packet is sent to all Devices.  
3284 Each queried Device responds directly with its Resources to the discovering Device. Resources  
3285 available locally are registered on the same Device.

3286 In some situations, one of the other mechanisms described in 11.3.2.3, called indirect discovery,  
3287 may be required. Indirect discovery is when a 3rd party Device, other than the discovering Device  
3288 and the discovered Device, assists with the discovery process. The 3rd party Device, called  
3289 Resource Directory (RD), only provides information on Resources on behalf of another Device but  
3290 does not host Resources on part of that Device.



3291

3292

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

3293 In Figure 17, Device B acts as Resource Directory for Device A and Device D. Device A and Device  
 3294 D publish their Resource information to Device B. Device C may query Device B to acquire the  
 3295 Resource information of Device A and Device D. Device A and Device D may not respond to a  
 3296 multicast query when Device B, as a Resource Directory, responds to the query on their behalf.

3297 Indirect discovery is useful for a constrained Device that needs to sleep to manage power and  
 3298 cannot process every discovery request, or when Devices may not be on the same network and  
 3299 requires optimization for discovery. Once Resources are discovered using indirect discovery, i.e.,  
 3300 RD query, then the access to the Resource is done by a request sent directly to the Device that  
 3301 hosts that Resource.

### 3302 11.3.6.1.2 Resource Directory

3303 A Resource Directory (RD) is a Device that assists with indirect discovery. A Device which acts as  
 3304 an RD will be involved in the following operations.

- 3305 – *RD discovery* – the procedure with which publishing Devices discover an RD and acquire the  
 3306 criteria to select from among multiple detected RDs.
- 3307 – *Resource publish* – the procedures with which Devices publish their Resource information, i.e.  
 3308 Links. Future revision of this document will allow modifying RD entries with UPDATE and  
 3309 DELETE operations. Any UPDATE or DELETE operations performed on an RD in this  
 3310 document should be either silently ignored or generate an error.
- 3311 – *Resource exposure* – the feature with which RDs expose the Links hosted by the 3<sup>rd</sup> party  
 3312 Devices via their own "/oic/res".

3313 The RDs make use of Resource Type "oic.wk.rd" defined in Table 29 and Table 30. A Device that  
 3314 supports the capability to host indirect discovery shall expose an instance of "oic.wk.rd" in its  
 3315 "/oic/res" to announce that it serves as an RD. The discoverable instance of "oic.wk.rd" shall allow  
 3316 only secure connections (e.g. OCF Endpoint with a scheme of "coaps" or "coaps+tcp"). A publishing  
 3317 Device may send a RETRIEVE request to "/oic/rd" to acquire the selection criteria among multiple  
 3318 RDs. Then it may send an UPDATE request to "/oic/rd" with its Links in the payload to publish the

3319 Links in "/oic/res" of the RD. A publishing Device is responsible to insure an RD has the correct  
 3320 published Links to expose via its "/oic/res".

3321 **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 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

3322

3323

3324 **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.

3325

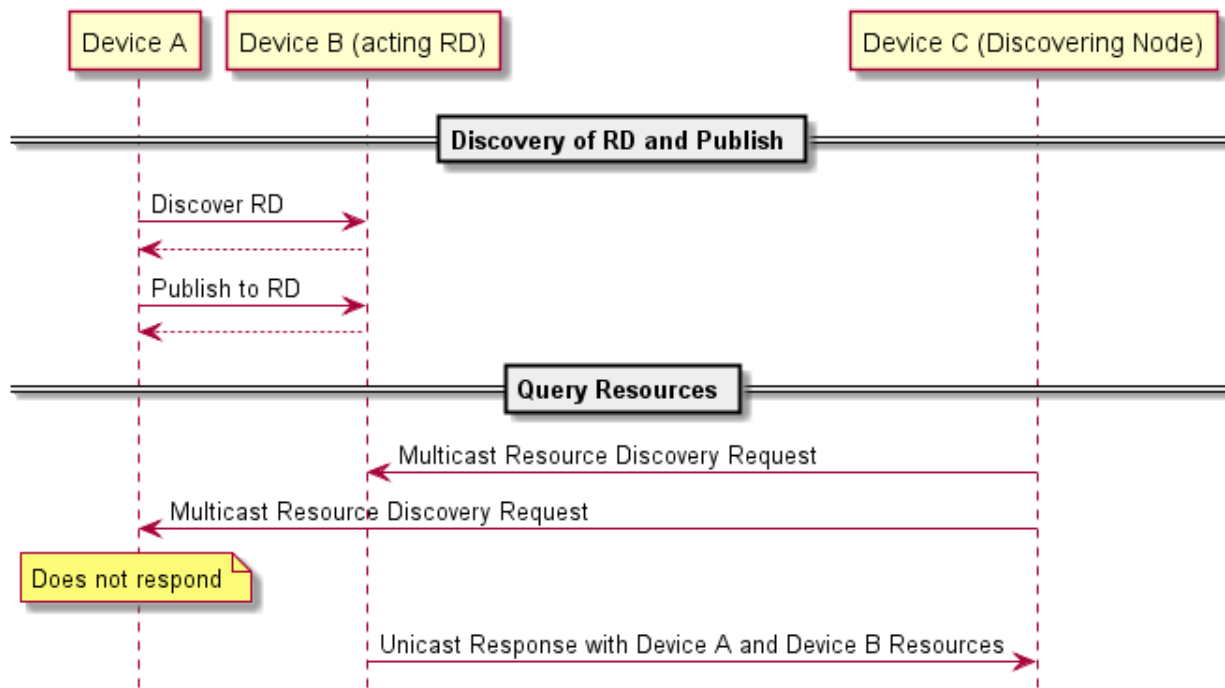
3326 An RD may be queried at its "/oic/res" Resource to find Resources hosted on other Devices. These  
 3327 Devices can be sleepy nodes or any other Device that cannot or may not respond to discovery  
 3328 requests. A publishing Device may publish all or a partial list of Resources they host to an RD. The  
 3329 RD then responds to queries for Resource discovery on behalf of the publishing Device (for  
 3330 example: when a Device may go to sleep). For general Resource discovery, the RD behaves like  
 3331 any other Server in responding to requests to "/oic/res".

3332 The remainder of 11.3.6 is divided into three parts. The first part covers "RD Discovery" (see  
 3333 11.3.6.2), i.e., discovering and selecting of an RD. The second part covers "Resource publish" (see  
 3334 11.3.6.3), i.e., publishing of Resources. The third part covers "Resource exposure" (see 11.3.6.4)  
 3335 where the RD replies to queries from Devices looking to discover Resources.

3336 **11.3.6.2 RD discovery**

3337 **11.3.6.2.1 Discovering an RD**

3338 An RD shall support RD discovery.



3339

3340

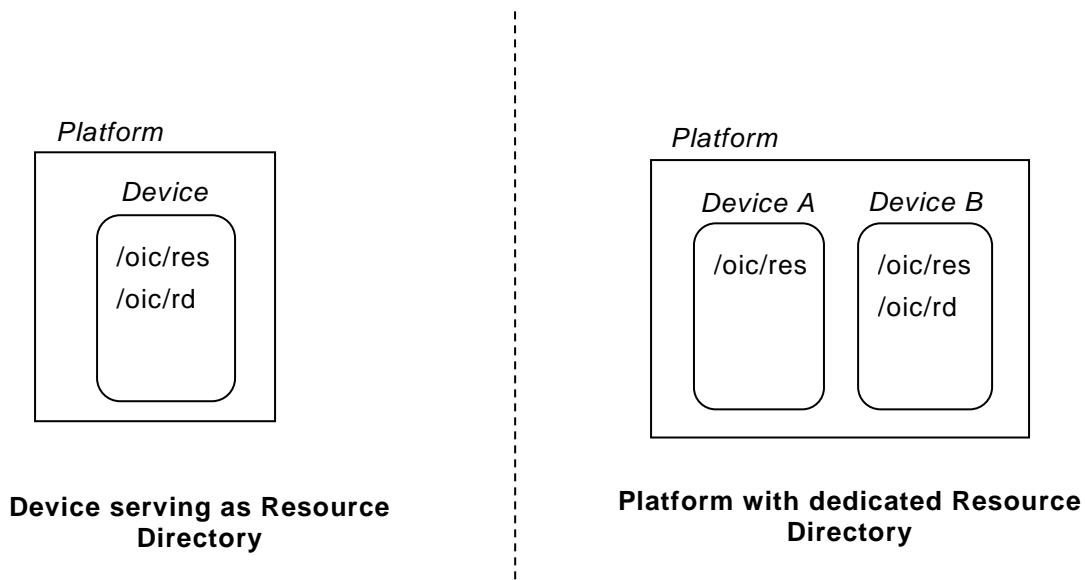
**Figure 18 – RD discovery and RD supported query of Resources support**

3341 In Figure 18 and Figure 19, a Device that wishes to publish its Resources first discovers an RD  
 3342 and then publishes the desired Resource information. Once a set of Resources have been  
 3343 published to an RD then the publishing Device should not respond to multicast Resource discovery  
 3344 queries for those published Resources when the RD is on the same multicast domain. In that case,  
 3345 only the RD should respond to multicast Resource discovery requests on the Resource published  
 3346 to it.

3347 It is allowed for more than one Device to act as an RD. The reason to have multiple RD support is  
 3348 to make networks scalable, handle network failures and prevent centralized Device failure  
 3349 bottlenecks. This does not preclude a scenario where a use case or deployment environment may  
 3350 require a single Device in the environment to be deployed as the only RD (e.g. gateway model).

3351 Discovering an RD may result in responses from more than one RD. If more than one RD responds,  
 3352 the discovering Device may select one of them based on the weighting parameter(s) provided in  
 3353 the response from the RD.

3354 A Client that performs Resource discovery uses an RD just like it uses any other Server for  
 3355 discovery. It may send a unicast request to the RD when it needs only the Resources published on  
 3356 the RD or do a multicast query when it does not require or have explicit knowledge of an RD.



3357  
3358 **Figure 19 – Resource Direction Deployment Scenarios**

3359 RDs may also be discovered in the following ways:

- 3360 – Pre-configuration: Devices wishing to publish Resource information may be configured a priori  
3361 with the information (e.g. IP address, port, transport etc.) of a specific RD. This pre-  
3362 configuration may be done at onboarding or may be updated on the Device using an out-of-  
3363 band method. This pre-configuration may be done by the manufacturer.
- 3364 – Query-oriented: A publishing Device wanting to discover Resource Directories using query-  
3365 oriented discovery may issue a multicast Resource discovery request for "/oic/res?rt=oic.wk.rd".  
3366 Only and all Devices that can be an RD shall respond to this query. The "/oic/rd" response shall  
3367 include information about the RD i.e., the presence of "oic.wk.rd" Link (as defined by the  
3368 Resource Type) and a subsequent query to "/oic/rd" would produce weighting parameters to  
3369 allow the discovering Device to select between RDs (see details in RD selection 11.3.6.2.2).  
3370 The "oic.wk.rd" Resource shall be instantiated on the Devices acting as RDs. The "oic.wk.rd"  
3371 schema is as defined in Annex D.

3372 **11.3.6.2.2 RD selection process**

3373 The Device that wants to use an RD will find zero or more RDs on the network. There may not be  
3374 an RD within the network. When discovering RDs, the Device needs to select an RD of all RDs  
3375 found on the network. The Device may send a RETRIEVE request to "/oic/rd/" of a specific RD, the  
3376 RD shall respond with the representation of "/oic/rd/" containing selection criteria as defined by the  
3377 "sel" Property. The lower the "sel" Property value is, the more preferable the responding RD is.  
3378 The creation of the "sel" value is vendor defined.

3379 For example an "/oic/rd" response may return the following.

```
3380 {
3381   "rt": ["oic.wk.rd"],
3382   "if": ["oic.if.baseline"],
3383   "sel": 50
3384 }
```

3385 The selection based on the "sel" Property value will ensure that a Device can judge if the found RD  
3386 is suitable for its needs.

3387 The following situations may occur during the selection of an RD:

3388 – A single or multiple RDs are present in the network.

3389 – No RD is present in the network.

3390 – An additional RD arrives on the network.

3391 In the first scenario, the RDs are already present. If a single RD is detected then that RD may be  
 3392 used. When multiple RDs are detected the Device may use the "sel" Property value to select the  
 3393 RD.

3394 In the second scenario, the publishing Device may continue looking for an RD until one is found or  
 3395 give up using an RD altogether.

3396 In the third scenario, the Device has already published its Resources to an existing RD, then  
 3397 discovers a new RD on the network. After judging the "sel" Property value, the Device may choose  
 3398 to move to the new RD. The Device should delete its Resource information from the currently used  
 3399 RD and publish the information to the new RD.

3400 **11.3.6.3 Resource publish**

3401 **11.3.6.3.1 Overview**

3402 An RD shall provide the facility to allow Devices to publish their Resource information to an RD.

3403 **11.3.6.3.2 Publish Resources**

3404 **11.3.6.3.2.1 Overview**

3405 After the selection process of an RD, a Device may push its Resource information to the selected  
 3406 RD, i.e., publish the Links in its "/oic/res" to the "/oic/res" of the RD.

3407 The publishing Device may decide to publish all Resources or just a few of the Resources on the  
 3408 RD. The publishing Device should only publish Resources that are otherwise published to its own  
 3409 "/oic/res"; a publishing Device should not publish non-Discoverable Resources or Resources  
 3410 hosted by some other Device. A publishing Device shall respond to discovery requests on its  
 3411 "/oic/res" Resource unless all its Discoverable Resources have been published in an RD.

3412 **11.3.6.3.2.2 Publish: Push Resource information**

3413 Resource information may be published using an UPDATE request sent to "/oic/rd".

3414 A Device which hosts a Resource may publish the Resource information, i.e. the Link targeting the  
 3415 Resource, to an RD by sending an UPDATE request with the Link in the payload. The published  
 3416 Link shall be exposed through the "/oic/res" of the RD.

3417 When a Device first publishes a Link or Links, it shall send an UPDATE request to the "/oic/rd"  
 3418 Resource of the RD including the following key-value pairs in the payload:

3419 – "di" –its value shall be the Device ID of the publishing Device, i.e. the "di" value of "/oic/d".

3420 – "links" –its value shall be the array of Links to be published. Links may omit the "ins" Parameter  
 3421 in which case the RD will assign a value for each Link. The supplied "ins" Parameter by the  
 3422 Client is allowed to be overruled by the RD, e.g. an RD can ignore the supplied "ins" value.

3423 – "ttl" –its value indicates how long (in seconds) the publishing Device requests the RD to keep  
 3424 this published Link.

3425 Notice that the payload shall carry the appropriate Content-Format of "application/vnd.ocf+cbor".

3426 {  
 3427 "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",  
 3428 "links": [  
 3429 {  
 3430 "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9"



```

3431     "href": "/myLightSwitch",
3432     "rt": ["oic.r.switch.binary"],
3433     "if": ["oic.if.a", "oic.if.baseline"],
3434     "p": {"bm": 3},
3435     "eps": [
3436         {"ep": "coaps://[fe80:b1d6]:1111", "pri": 2},
3437         {"ep": "coaps://[fe80:b1d6]:1122"},
3438         {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
3439     ]
3440 },
3441 {
3442     "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
3443     "href": "/myLightBrightness",
3444     "rt": ["oic.r.brightness"],
3445     "if": ["oic.if.a", "oic.if.baseline"],
3446     "p": {"bm": 3},
3447     "eps": [
3448         {"ep": "coaps://[[2001:db8:a::123]:2222"}
3449     ]
3450 }
3451 ],
3452 "ttl": 600
3453 }

```

3454 When an RD receives this initial UPDATE request, it determines whether to grant the request or  
3455 not. Upon granting the request, the RD shall send back an UPDATE response to the publishing  
3456 Device. The response shall include a payload with the same information as the original UPDATE  
3457 request with the following possible differences:

- 3458 – For each Link, an "ins" Parameter shall be included in the response. The RD shall assign a  
3459 unique "ins" value identifying the Link among all the Links it advertises. If the publishing Device  
3460 included an "ins" value in the UPDATE request, the RD may use it as long as it doesn't match  
3461 any existing "ins" value in the published Links.
- 3462 – The "ttl" Property Value shall be assigned by the RD and it shall be included in the response.  
3463 The RD should use the value included in the UPDATE request but may assign a value that is  
3464 lower if it is not able to honour the requested "ttl" value. After this time elapses, the RD shall  
3465 remove the Links. To keep a Link alive the publishing Device may update the "ttl" using the  
3466 UPDATE schema.

3467 The RD shall add the new Links to its "/oic/res" and expose them to a valid discovery query, i.e.  
3468 RETRIEVE request:

```

3469 {
3470     "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
3471     "links": [
3472         {
3473             "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
3474             "href": "/myLightSwitch",
3475             "rt": ["oic.r.switch.binary"],
3476             "if": ["oic.if.a", "oic.if.baseline"],
3477             "p": {"bm": 3},
3478             "eps": [
3479                 {"ep": "coaps://[fe80:b1d6]:1111", "pri": 2},
3480                 {"ep": "coaps://[fe80:b1d6]:1122"},
3481                 {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
3482             ],
3483             "ins": 11235
3484         },
3485         {
3486             "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
3487             "href": "/myLightBrightness",
3488             "rt": ["oic.r.brightness"],

```

```

3489     "if": ["oic.if.a", "oic.if.baseline"],
3490     "p": {"bm": 3},
3491     "eps": [
3492       {"ep": "coaps://[[2001:db8:a::123]:2222"}
3493     ],
3494     "ins": 112358
3495   }
3496 ].
3497   "ttl": 600
3498 }

```

3499 Once a publishing Device has published Resources to an RD, it may choose not respond to the  
3500 multicast discovery queries for the same Resources against its own `"/oic/res"`, especially when on  
3501 the same multicast domain as the RD. After publishing Resources, primarily it is the RDs  
3502 responsibility to reply to the queries for the published Resources.

3503 There is another possibility that the RD and the publishing Device both respond to the multicast  
3504 query from the discovering Device. This will create a duplication of the information but is an  
3505 alternative that may be used for non-robust networks. It is not a recommended option but for  
3506 industrial scenarios, this is one of the possibilities. Either way, discovering Clients need to always  
3507 be prepared to process duplicate information in responses to multicast discovery request. The  
3508 `"/oic/rd"` schema is as defined in Annex D to specify publishing to the `"/oic/rd"` Resource.

#### 3509 **11.3.6.4 Resource exposure**

##### 3510 **11.3.6.4.1 `"/oic/res"` and retrieving of the Resources**

3511 The `"/oic/res"` based discovery process remains the same as that in the absence of an RD.  
3512 Resources may be discovered by retrieving the `"/oic/res"` Resource by sending a multicast or  
3513 unicast request. In the case of a multicast discovery request, an RD shall include in its response  
3514 any published Resources on behalf of the Device that hosts the Resources. Clients should be  
3515 prepared to process duplicate Resource information from more than one RD responding with the  
3516 same information or from an RD and the hosting Device (publishing the Resource information) both  
3517 responding to the request. Interaction with Resources discovered using the RD is done using the  
3518 same mechanism and methods as with Resources discovered by retrieving the `"/oic/res"` Resource  
3519 of the Device hosting the Resources (e.g., connect to the hosting Device and perform CRUDN  
3520 operations on the Resource).

3521 Resource Directories provide different `"/oic/res"` responses according to the requesting Clients,  
3522 which indicate their preference with content format. OCF 1.0 Clients request with a Content Format  
3523 of `"application/vnd.ocf+cbor"` in the Accept Option, whereas the Content-Format `"application/cbor"`  
3524 in the Accept Option indicates OIC 1.1 Clients. For OIC 1.1 Clients, the `"/oic/res"` response includes  
3525 Links conforming to OIC 1.1 specification, which OIC 1.1 Clients can understand. In this case the  
3526 Resources hosted by the same Device shall be grouped together within a single JSON Object with  
3527 `"di"` indicating the hosting Device. For a 3rd party Resource, i.e., a Resource which doesn't belong  
3528 to the responding RD, its `"href"` value shall be a fully qualified transfer protocol URI with an IP  
3529 address and port number as its authority component (e.g.,  
3530 `coaps://[2001:db8:b::c2e5]:22222/myLightSwitch`).

3531 For example, an RD might return the following to an OIC 1.1 Clients:

```

3532 [
3533   {
3534     "di": "88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3535     "links": [
3536       {
3537         "href": "/oic/res",
3538         "rel": "self",
3539         "rt": ["oic.wk.res"],
3540         "if": ["oic.if.ll", "oic.if.baseline"],

```

```

3541     "p": {"bm": 3, "sec": false}
3542   },
3543   {
3544     "href": "/oic/d",
3545     "rt": ["oic.wk.d", "oic.d.fan"],
3546     "if": ["oic.if.r", "oic.if.baseline"],
3547     "p": {"bm": 3, "sec": false}
3548   },
3549   {
3550     "href": "/oic/p",
3551     "rt": ["oic.wk.p"],
3552     "if": ["oic.if.r", "oic.if.baseline"],
3553     "p": {"bm": 3, "sec": true, "port": 33333}
3554   },
3555   {
3556     "href": "/myFanIntrospection",
3557     "rt": ["oic.wk.introspection"],
3558     "if": ["oic.if.r", "oic.if.baseline"],
3559     "p": {"bm": 3, "sec": true, "port": 33333}
3560   },
3561   {
3562     "href": "/oic/rd",
3563     "rt": ["oic.wk.rd"],
3564     "if": ["oic.if.baseline"],
3565     "p": {"bm": 3, "sec": true, "port": 33333}
3566   },
3567   {
3568     "href": "/myFanSwitch",
3569     "rt": ["oic.r.switch.binary"],
3570     "if": ["oic.if.a", "oic.if.baseline"],
3571     "p": {"bm": 3, "sec": true, "port": 33333}
3572   },
3573   {
3574     "href": "/oic/sec/doxm",
3575     "rt": ["oic.r.doxm"],
3576     "if": ["oic.if.baseline"],
3577     "p": {"bm": 1, "sec": false}
3578   },
3579   {
3580     "href": "/oic/sec/pstat",
3581     "rt": ["oic.r.pstat"],
3582     "if": ["oic.if.baseline"],
3583     "p": {"bm": 1, "sec": true, "port": 33333}
3584   },
3585   {
3586     "href": "/oic/sec/cred",
3587     "rt": ["oic.r.cred"],
3588     "if": ["oic.if.baseline"],
3589     "p": {"bm": 1, "sec": true, "port": 33333}
3590   },
3591   {
3592     "href": "/oic/sec/acl2",
3593     "rt": ["oic.r.acl2"],
3594     "if": ["oic.if.baseline"],
3595     "p": {"bm": 1, "sec": true, "port": 33333}
3596   }
3597 ]
3598 },
3599 {
3600   "di": "dc70373c-1e8d-4fb3-962e-017eaa863989",
3601   "links": [
3602     {
3603       "href": "coap://[2001:db8:b::c2e5]:66666/oic/d",

```

```

3604         "rt": ["oic.wk.d", "oic.d.light", "oic.d.virtual"],
3605         "if": ["oic.if.r", "oic.if.baseline"],
3606         "p": {"bm": 3, "sec": false}
3607     },
3608     {
3609         "href": "coaps://[2001:db8:b::c2e5]:22222/oic/p",
3610         "rt": ["oic.wk.p"],
3611         "if": ["oic.if.r", "oic.if.baseline"],
3612         "p": {"bm": 3, "sec": true, "port": 22222}
3613     },
3614     {
3615         "href": "coaps://[2001:db8:b::c2e5]:22222/myLightSwitch",
3616         "rt": ["oic.r.switch.binary"],
3617         "if": ["oic.if.a", "oic.if.baseline"],
3618         "p": {"bm": 3, "sec": true, "port": 22222}
3619     },
3620     {
3621         "href": "coaps://[2001:db8:b::c2e5]:22222/myLightBrightness",
3622         "rt": ["oic.r.brightness"],
3623         "if": ["oic.if.a", "oic.if.baseline"],
3624         "p": {"bm": 3, "sec": true, "port": 22222}
3625     }
3626 ]
3627 }
3628 ]
3629

```

3630 For OCF 1.0 Clients, the "/oic/res" response includes the OCF 1.0 Links with the "anchor"  
3631 Parameter containing an OCF URI. The "/oic/res" response has a single array of Links to conform  
3632 to IETF RFC 6690. Each Link shall contain the "anchor" Parameter of the value OCF URI where  
3633 the authority component of <deviceId> indicates the Device hosting the target Resource.

3634 For example, an RD may return the following to an OCF 1.0 Client.

```

3635 [
3636   {
3637     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3638     "href": "/oic/res",
3639     "rel": "self",
3640     "rt": ["oic.wk.res"],
3641     "if": ["oic.if.ll", "oic.if.baseline"],
3642     "p": {"bm": 3},
3643     "eps": [
3644       {"ep": "coap://[2001:db8:a::b1d4]:77777"},
3645       {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3646     ]
3647   },
3648   {
3649     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3650     "href": "/oic/d",
3651     "rt": ["oic.wk.d", "oic.d.fan"],
3652     "if": ["oic.if.r", "oic.if.baseline"],
3653     "p": {"bm": 3},
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/p",
3662     "rt": ["oic.wk.p"],
3663     "if": ["oic.if.r", "oic.if.baseline"],

```

```

3664     "p": {"bm": 3},
3665     "eps": [
3666       {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3667     ],
3668   },
3669   {
3670     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3671     "href": "/myFanIntrospection",
3672     "rt": ["oic.wk.introspection"],
3673     "if": ["oic.if.r", "oic.if.baseline"],
3674     "p": {"bm": 3},
3675     "eps": [
3676       {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3677     ],
3678   },
3679   {
3680     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3681     "href": "/oic/rd",
3682     "rt": ["oic.wk.rd"],
3683     "if": ["oic.if.baseline"],
3684     "p": {"bm": 3},
3685     "eps": [
3686       {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3687     ],
3688   },
3689   {
3690     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3691     "href": "/myFanSwitch",
3692     "rt": ["oic.r.switch.binary"],
3693     "if": ["oic.if.a", "oic.if.baseline"],
3694     "p": {"bm": 3},
3695     "eps": [
3696       {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3697     ],
3698   },
3699   {
3700     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3701     "href": "/oic/sec/doxm",
3702     "rt": ["oic.r.doxm"],
3703     "if": ["oic.if.baseline"],
3704     "p": {"bm": 1},
3705     "eps": [
3706       {"ep": "coap://[2001:db8:a::b1d4]:77777"},
3707       {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3708     ],
3709   },
3710   {
3711     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3712     "href": "/oic/sec/pstat",
3713     "rt": ["oic.r.pstat"],
3714     "if": ["oic.if.baseline"],
3715     "p": {"bm": 1},
3716     "eps": [
3717       {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3718     ],
3719   },
3720   {
3721     "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3722     "href": "/oic/sec/cred",
3723     "rt": ["oic.r.cred"],
3724     "if": ["oic.if.baseline"],
3725     "p": {"bm": 1},
3726     "eps": [

```

```

3727     {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3728   ]
3729 },
3730 {
3731   "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
3732   "href": "/oic/sec/acl2",
3733   "rt": ["oic.r.acl2"],
3734   "if": ["oic.if.baseline"],
3735   "p": {"bm": 1},
3736   "eps": [
3737     {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
3738   ]
3739 },
3740 {
3741   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
3742   "href": "/oic/d",
3743   "rt": ["oic.wk.d", "oic.d.light"],
3744   "if": ["oic.if.r", "oic.if.baseline"],
3745   "p": {"bm": 3},
3746   "eps": [
3747     {"ep": "coap://[2001:db8:b::c2e5]:66666"},
3748     {"ep": "coaps://[2001:db8:b::c2e5]:22222"}
3749   ]
3750 },
3751 {
3752   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
3753   "href": "/oic/p",
3754   "rt": ["oic.wk.p"],
3755   "if": ["oic.if.r", "oic.if.baseline"],
3756   "p": {"bm": 3},
3757   "eps": [
3758     {"ep": "coaps://[2001:db8:b::c2e5]:22222"}
3759   ]
3760 },
3761 {
3762   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
3763   "href": "/myLightSwitch",
3764   "rt": ["oic.r.switch.binary"],
3765   "if": ["oic.if.a", "oic.if.baseline"],
3766   "p": {"bm": 3},
3767   "eps": [
3768     {"ep": "coaps://[2001:db8:b::c2e5]:22222"}
3769   ]
3770 },
3771 {
3772   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
3773   "href": "/myLightBrightness",
3774   "rt": ["oic.r.brightness"],
3775   "if": ["oic.if.a", "oic.if.baseline"],
3776   "p": {"bm": 3},
3777   "eps": [
3778     {"ep": "coaps://[2001:db8:b::c2e5]:22222"}
3779   ]
3780 }
3781 ]

```

## 3782 11.4 Notification

### 3783 11.4.1 Overview

3784 A Server shall support NOTIFY operation to enable a Client to request and be notified of desired  
3785 states of one or more Resources in an asynchronous manner. 11.4.2 specifies the Observe  
3786 mechanism in which updates are delivered to the requester.

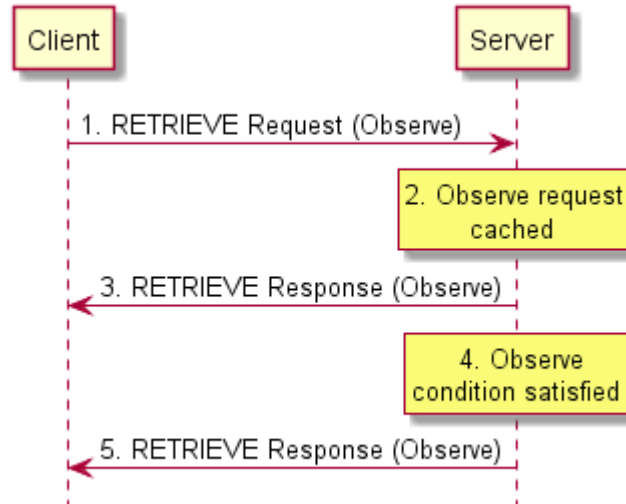
3787 **11.4.2 Observe**

3788 **11.4.2.1 Overview**

3789 In the Observe mechanism the Client utilizes the RETRIEVE operation to require the Server for  
3790 updates in case of Resource state changes. The Observe mechanism consists of five steps which  
3791 are depicted in Figure 20.

3792 NOTE the Observe mechanism can only be used for a resource with a Property of Observable (see 7.3.2.2).

3793



3794

3795

3796 **Figure 20 – Observe Mechanism**

3797 **11.4.2.2 RETRIEVE request with Observe indication**

3798 The Client transmits a RETRIEVE request message to the Server to request updates for the  
3799 Resource on the Server if there is a state change. The RETRIEVE request message carries the  
3800 following parameters:

- 3801 – *fr*: Unique identifier of the Client.
- 3802 – *to*: Resource that the Client is requesting to Observe.
- 3803 – *ri*: Identifier of the RETRIEVE operation.
- 3804 – *op*: RETRIEVE.
- 3805 – *obs*: Indication for Observe operation.

3806 **11.4.2.3 Processing by the Server**

3807 Following the receipt of the RETRIEVE request, the Server may validate if the Client has the  
3808 appropriate rights for the requested operation and the Properties are readable and Observable. If  
3809 the validation is successful, the Server caches the information related to the Observe request. The  
3810 Server caches the value of the *ri* parameter from the RETRIEVE request for use in the initial  
3811 response and future responses in case of a change of state.

3812 **11.4.2.4 RETRIEVE response with Observe indication**

3813 The Server shall transmit a RETRIEVE response message in response to a RETRIEVE request  
3814 message from a Client. The RETRIEVE response message shall include the following parameters.  
3815 If validation succeeded, the response includes an Observe indication. If not, the Observe indication

3816 is omitted from the response which signals to the requesting Client that registration for notification  
3817 was not allowed.

3818 The RETRIEVE response message shall include the following parameters:

- 3819 – *fr*: Unique identifier of the Server.
- 3820 – *to*: Unique identifier of the Client.
- 3821 – *ri*: Identifier included in the RETRIEVE operation.
- 3822 – *cn*: Information Resource representation as requested by the Client.
- 3823 – *rs*: The result of the RETRIEVE operation.
- 3824 – *obs*: Indication that the response is made to an Observe operation.

#### 3825 **11.4.2.5 Resource monitoring by the Server**

3826 The Server shall monitor the state the Resource identified in the Observe request from the Client.  
3827 Anytime there is a change in the state of the Observed Resource, the Server sends another  
3828 RETRIEVE response with the Observe indication. The mechanism does not allow the client to  
3829 specify any bounds or limits which trigger a notification, the decision is left entirely to the server.

#### 3830 **11.4.2.6 Additional RETRIEVE responses with Observe indication**

3831 The Server shall transmit updated RETRIEVE response messages following Observed changes in  
3832 the state of the Resources indicated by the Client. The RETRIEVE response message shall include  
3833 the parameters listed in 11.4.2.4.

#### 3834 **11.4.2.7 Cancelling Observe**

3835 The Client can explicitly cancel Observe by sending a RETRIEVE request without the Observe  
3836 indication field to the same Resource on the Server which it was Observing. For certain protocol  
3837 mappings, the Client may also be able to cancel an Observe by ceasing to respond to the  
3838 RETRIEVE responses.

### 3839 **11.5 Device management**

#### 3840 **11.5.1 Overview**

3841 The Device management includes the following functions:

- 3842 – Diagnostics and maintenance.

3843 The Device management functionalities specified in this version of document are intended to  
3844 address the basic Device management features. Addition of new Device management features in  
3845 the future versions of the document is expected.

#### 3846 **11.5.2 Diagnostics and maintenance**

3847 The Diagnostics and Maintenance function is intended for use by administrators to resolve issues  
3848 encountered with the Devices while operating in the field. If diagnostics and maintenance is  
3849 supported by a Device, the Core Resource `"/oic/mnt"` shall be supported as described in Table 31.

3850 **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
<code>"/oic/mnt"</code>	Maintenance	<code>"oic.wk.mnt"</code>	<code>"oic.if.rw"</code>	The Resource through which the Device is maintained and can be used for diagnostic purposes. The Properties exposed by <code>"/oic/mnt"</code> are listed in Table 32.	Device management



3851 Table 32 defines the "oic.wk.mnt" Resource Type. At least one of the Factory Reset, Reboot or last  
 3852 error Properties shall be implemented.

3853 **Table 32 – "oic.wk.mnt" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
<b>Factory_Reset</b>	"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.
<b>Reboot</b>	"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)
<b>Last error</b>	"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.

3854  
 3855 NOTE Default indicates the value of this Property as soon as the Device is rebooted or factory reset.

3856 **11.5.3 Network monitoring**

3857 Network monitoring is used for monitoring the current network state of the Device.

3858 The network monitoring Resource Type is "oic.wk.nmon" and is described in Table 33. The  
 3859 Resource Type may occur multiple times if more than 1 network interface is implemented. The  
 3860 Common Property "n" may be used to distinguish the different network interfaces, like distinguish  
 3861 the 2.4 and 5G Wi-Fi network interfaces.

3862 **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	Device management

				Resource Properties exposed by Resource Type "oic.wk.nmon" are listed in Table 34	
--	--	--	--	---	--

3863

3864 Table 34 defines "oic.wk.nmon" Resource Type.

3865

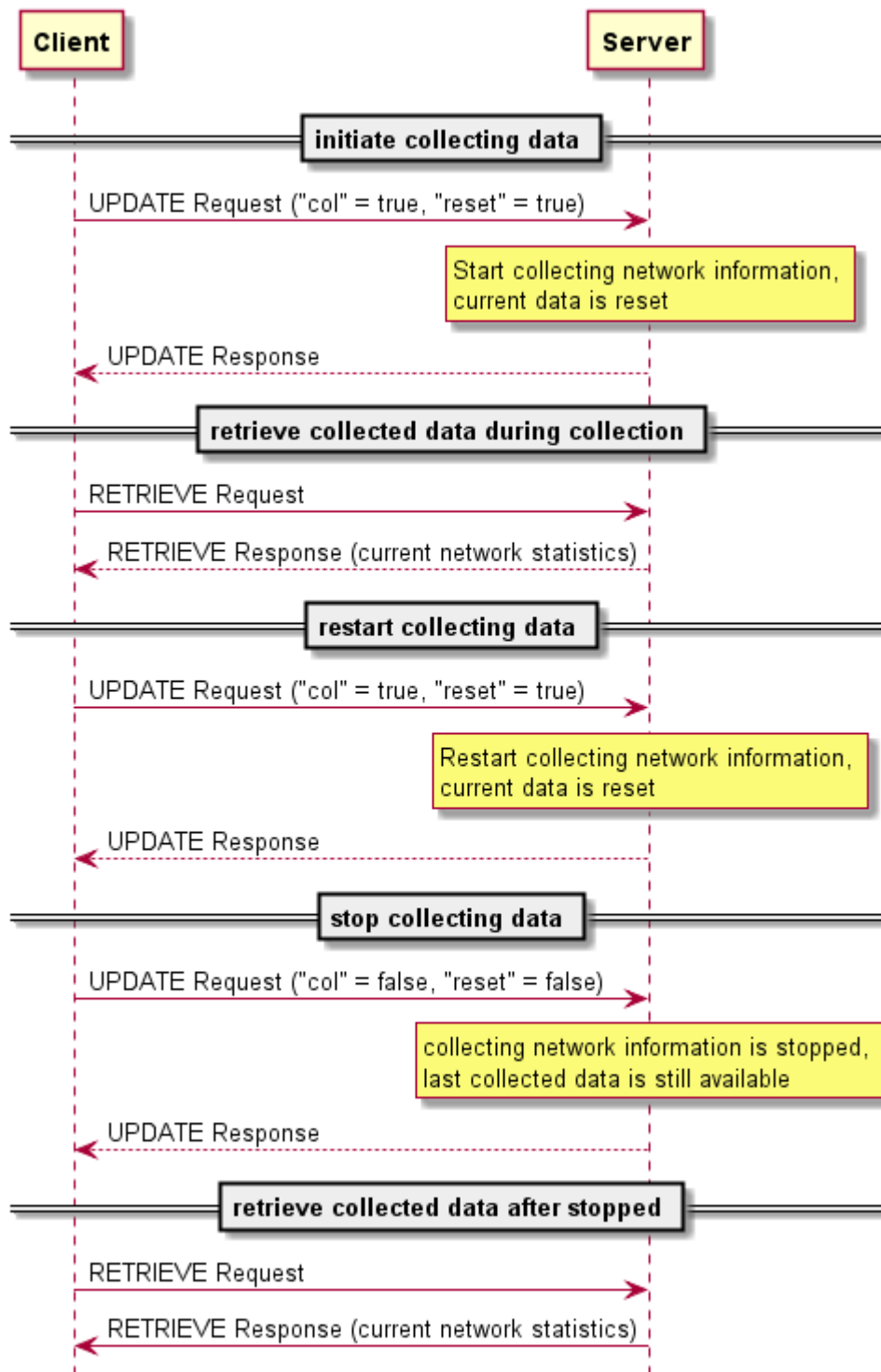
**Table 34 – "oic.wk.nmon" Resource Type definition**

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 IANA ifType-MIB Definitions.
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.

3866

3867 Examples of typical used values for ianaifType are 71 (ieee80211) for Wi-Fi and 6 (ethernetCsmacd)  
3868 for Ethernet.

3869 A Device should start collecting network monitoring data when receiving an UPDATE operation  
3870 with the parameter "col" = true. A Device should stop collecting network data when receiving an  
3871 UPDATE operation with parameter "col" = false. The collected network data should be reset when  
3872 an UPDATE operation with parameter "reset" = true is received, if the parameter "reset" is false  
3873 then the values should not be reset. Figure 21 illustrates the interactions with the network  
3874 monitoring Resource.

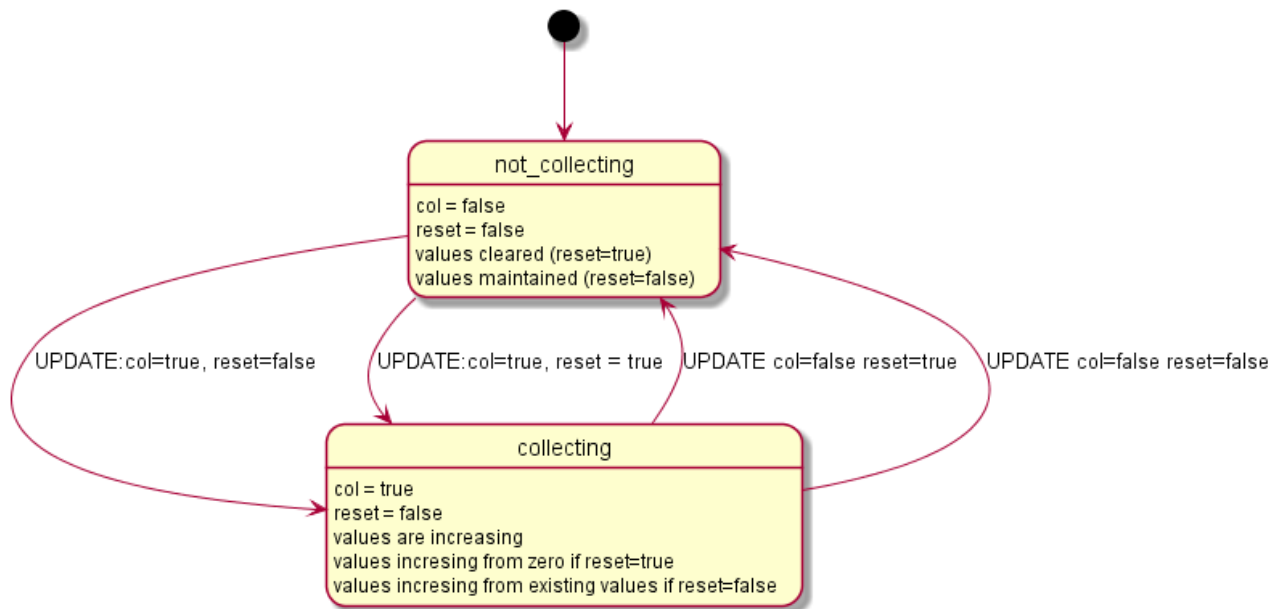


3875

3876

**Figure 21 – Interactions with the network monitoring Resource**

3877 The state transition diagram for collecting or not collecting network information is described by  
 3878 Figure 22.



3879

3880

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

3881 **11.5.4 Software update Resource**

3882 In ISO/IEC 30118-2:2018 there is already a manual triggered software update mechanism available.  
 3883 The triggering of the Client (manual) software update is achieved via the security Resource Type  
 3884 "oic.r.pstat" by using the appropriate bits in the "tm" Property. The software update triggering  
 3885 results in updates of the "cm" Property in the "oic.r.pstat" Resource Type (see ISO/IEC 30118-  
 3886 2:2018 clause 13.8). The software update Resource adds additional features to the security  
 3887 specified mechanism, like:

- 3888 – Specify the source to obtain the software package.
- 3889 – Time scheduled software update actions.
- 3890 – Status information, especially more info about various error situations.

3891 If the Device implements the software update Resource, it is required to implement the software  
 3892 update behaviour to actually update the software of the Device as indicated by the "oic.r.pstat"  
 3893 "cm" bits as defined in ISO/IEC 30118-2:2018 clause 13.8. Also the security defined software  
 3894 update process shall use the data that is set on the software update Resource like the "purl"  
 3895 Property.

3896 The software update Resource Type is "oic.r.softwareupdate" and is described in Table 35.

3897

**Table 35 – Optional software update Resources**

Example URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
"/example/oic/swupd"	Software Update	"oic.r.softwareupdate"	"oic.if.rw", "oic.if.baseline"	The Resource exposes Properties to control and monitor the software update mechanism. The Properties exposed by Resource Type "oic.r.softwareupdate" are listed in Table 36.	Device management

3898

3899 Table 36 defines the Properties of the "oic.r.softwareupdate" Resource Type.

3900

**Table 36 – "oic.r.softwareupdate" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
New version	"nv"	"string"	N/A	N/A	R	No	New available software version.
Package url	"purl"	"string"	URL	N/A	RW	Yes	Source of the software package, might be an HTTPS or a CoAPs URL.
Action	"swupdateaction"	"string"	enum (see Table 38)	N/A	RW	Yes	Scheduled action to do a software update.
State	"swupdatestate"	"string"	enum (see Table 37)	N/A	R	Yes	State of the software update.
Result	"swupdateresult"	"integer"	N/A	N/A	R	Yes	Result of the software update. List of error codes are as defined in Table 39.
Lastupdate	"lastupdate"	"string"	date-time	N/A	R	No	Time of the last software update according to IETF RFC 3339. Initial set on date of manufacturing.
Signage	"signed"	"string"	enum	N/A	R	No	Signage method of the software package, currently the only allowed value is "vendor".
Update time	"update time"	"string"	date-time	N/A	RW	Yes	Scheduled time, according to IETF RFC 3339, to do action which is specified in the "swupdateaction" Property.

3901

3902

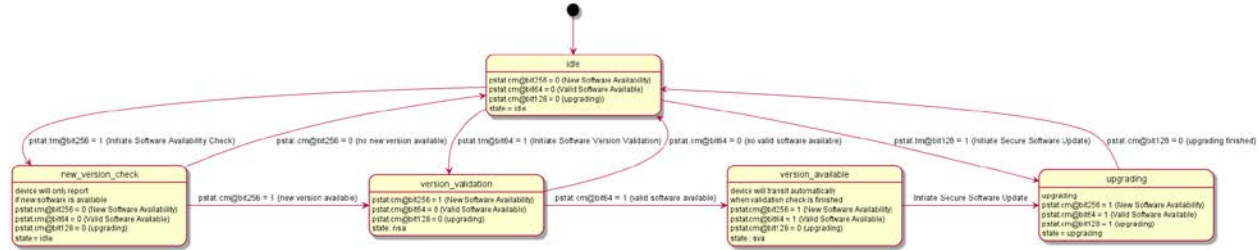
**Table 37 State definitions and state transitions of software update Resource**

Description	Value of Property "swupdatestate"	equivalent "cm" bit values in "pstat".	Transition allowed from state
Idle, waiting for updates	"idle"	Bit 64 = 0 Bit 128 = 0 Bit 256 = 0	"nsa", "svv", "sva", "upgrading"
New software available (after checking for new software being available on the url indicated by "purl"). This step does not download the new software	"nsa"	Bit 64 = 0 Bit 128 = 0 Bit 256 = 1	"idle", "svv", "sva", "upgrading"
Software version validation (during downloading and checking the software integrity)	"svv"	Bit 64 = 0 Bit 128 = 0 Bit 256 = 1	"idle", "nsa", "sva", "upgrading"
Software version available (The software is downloaded and deemed to be valid)	"sva"	Bit 64 = 1 Bit 128 = 0 Bit 256 = 1	"idle", "nsa", "svv", "upgrading"

Upgrading	"upgrading"	Bit 64 = 1 Bit 128 = 1 Bit 256 = 1	"idle", "nsa", "svv", "sva"
-----------	-------------	--	-----------------------------

3903

The typical state transitions are described by



3904

3905

3906

3907

3908

3909

3910

3911

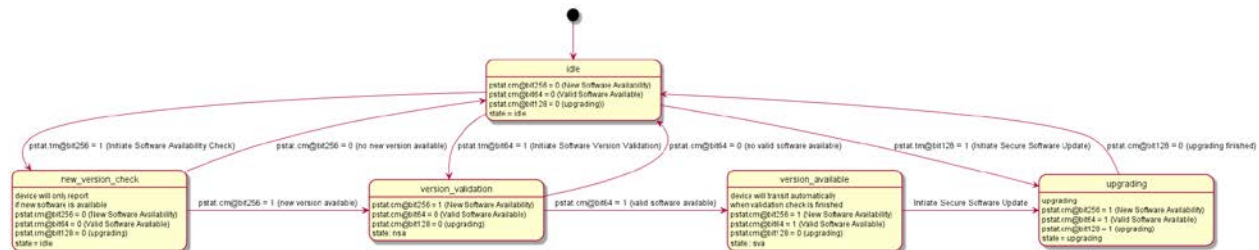
Figure 23. The state transitions can be triggered manually or by a timed action. The manual state triggers (i.e., "tm" Property of "oic.r.pstat") are described in ISO/IEC 30118-2:2018 clause 13.8. The timed state triggers are managed using the "swupdateaction" and "updatetime" Properties of the software update Resource to trigger software update actions at some future date and time. The action names for scheduled actions are listed in Table 38. When the "updatetime" for the timed action is in the past then the update shall not take place, it is implementation dependent if the UPDATE with an "updatetime" value in the past will give an error on the UPDATE operation.

3912

**Table 38 Value definitions for the Property "swupdateaction"**

Description	Value of Property "swupdateaction", for scheduled update actions.	Action taken	Equivalent "pstat" "tm" bits.
Nothing scheduled (not applicable).	"idle"	No action	
Initiate software availability check.	"isac"	Check on remote end point if a newer software version is available.	"tm" bit 256.
Initiate software versionValidation.	"isvv"	Downloads and verifies if the software version is valid.	"tm" bit 64.
Initiate secure software update.	"upgrade"	Upgrades the software in the Device. It uses the downloaded and validated software package. If no validated software package is available on the Device, the Device takes the necessary steps to obtain a validated software package, by downloading and verifying the software from the external source.	"tm" bit 128.

3913



3914

3915

**Figure 23 – Typical state transitioning diagram for software update**

3916

3917

3918

3919

The "purl" Property indicates the URL to obtain the software package from. This URL shall be a fully qualified URL. If the value is an empty string ("") then the Device will use the built in vendor defined URL (see security specification). If a built in URL is not implemented, setting the "purl" Property value to an empty string will result in an error code value of 6 as defined in Table 39.

3920

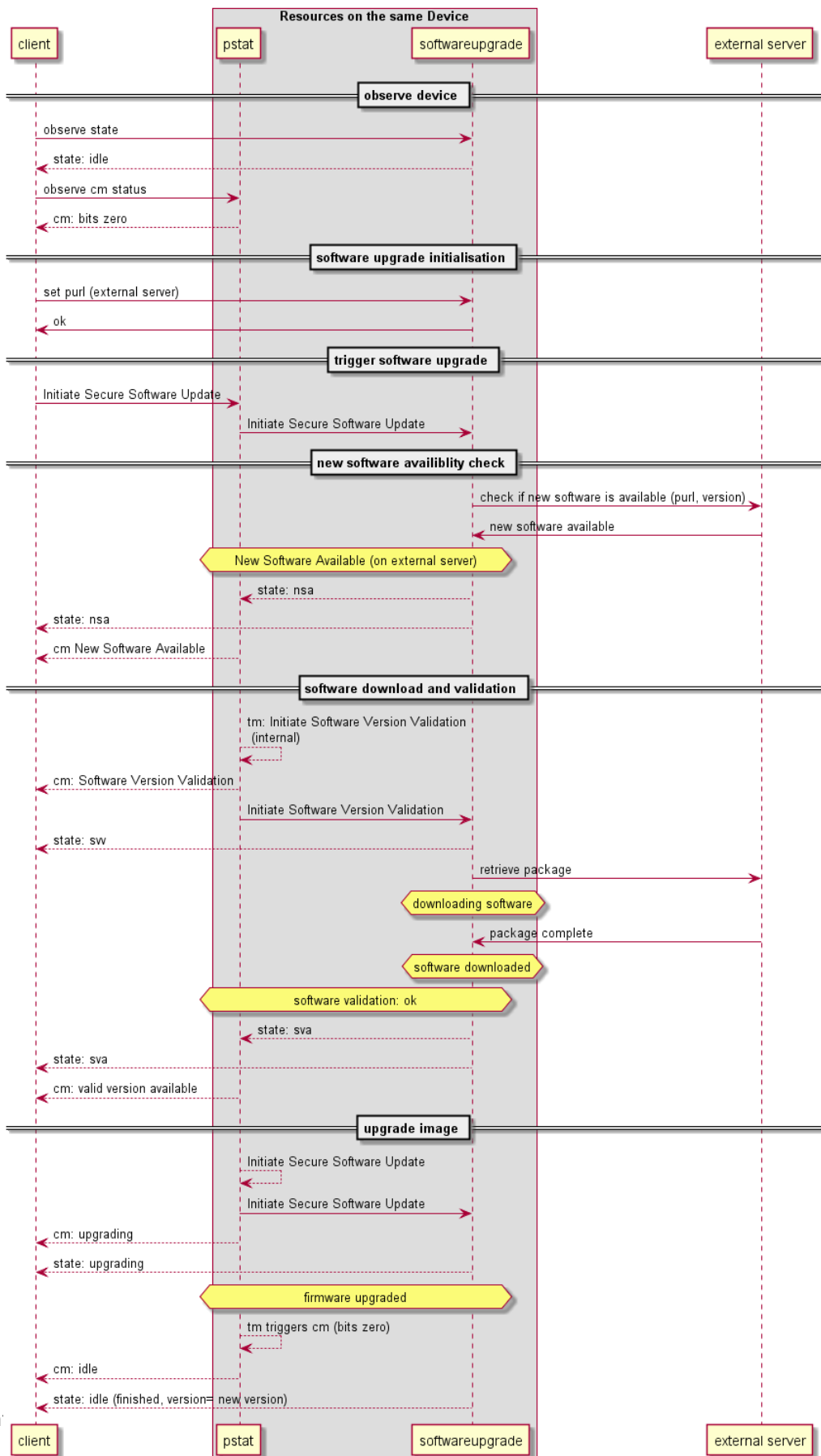
**Table 39 List of codes of the "swupdateresult" Property.**

Description	code
Idle.	0
Success, everything went well.	1
Not enough RAM.	2
Not enough Flash.	3
Connection lost.	4
Software validation failure.	5
Invalid URL to receive the software package.	6
Unsupported protocol for download URL.	7
Firmware update failed.	8
Software transport error codes. HTTP result codes when accessing the URL to download the software package.	400-600

3921

3922 Figure 24 depicts a typical update scenario. This scenario is using the observability of "pstat", so  
3923 that the Client is informed on the changes of the "cm" bit value to track the progress.





3925

**Figure 24 – Typical sequence for none scheduled upgrading software**

3926 **11.6 Scenes**

3927 **11.6.1 Introduction**

3928 Scenes are a mechanism for automating certain operations.

3929 A Scene is a static entity that stores a set of defined Property values for a Collection of Resources.  
3930 Scenes provide a mechanism to store a setting over multiple Resources that may be hosted by  
3931 multiple separate Servers. Scenes, once set up, can be used by multiple Clients to recall a setup.

3932 Scenes can be grouped and reused, a group of Scenes is also a Scene.

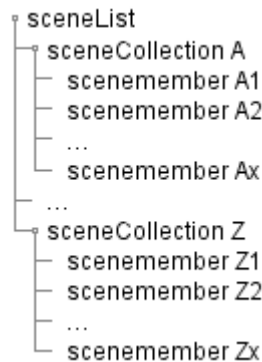
3933 In short, Scenes are bundled user settings.

3934 **11.6.2 Scenes**

3935 **11.6.2.1 Introduction**

3936 Scenes are described by means of Resources. The Scene Resources are hosted by a Server and  
3937 the top level Resource is listed in "/oic/res". This means that a Client can determine if the Scene  
3938 functionality is hosted on a Server via Resource discovery as defined in 11.3. The setup of Scenes  
3939 is driven by Client interactions. This includes creating new Scenes, and mappings of Server  
3940 Properties that are part of a Scene.

3941 The Scene functionality is created by multiple Resources and has the structure depicted in  
3942 Figure 25. The sceneList and sceneCollection Resources are overloaded Collection Resources.  
3943 The sceneCollection Resource contains a list of Scenes. This list contains zero or more Scenes.  
3944 The sceneMember Resource contains the mapping between a Scene and what needs to happen  
3945 according to that Scene on an indicated Resource.



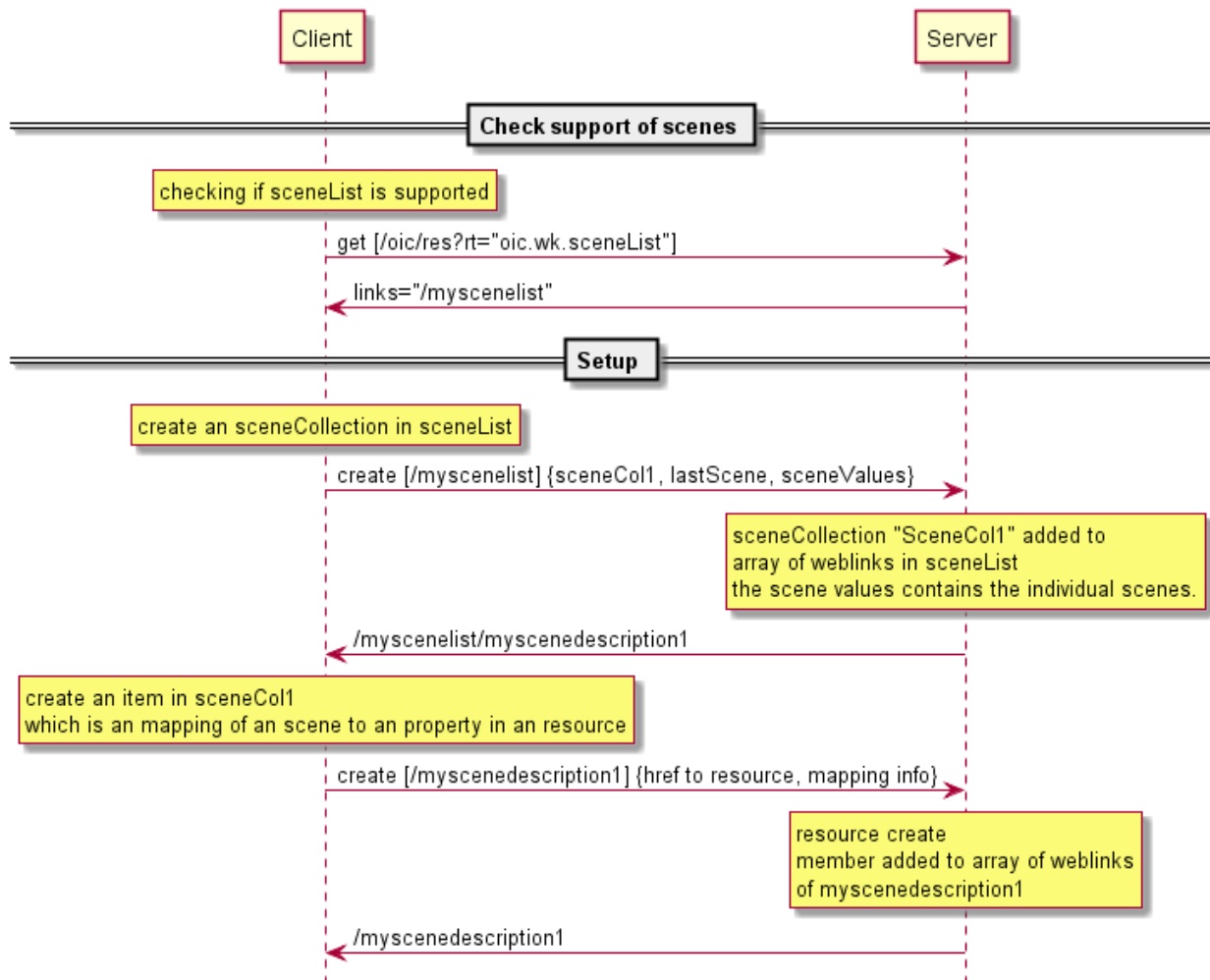
3946

3947 **Figure 25 – Generic Scene Resource structure**

3948 **11.6.2.2 Scene creation**

3949 A Client desiring to interact with Scenes needs to first determine if the Server supports the Scene  
3950 feature; the sceneMembers of a Scene that are Resources of end Device being updated by the  
3951 Scene change do not have to be co-located on the Server supporting the Scene feature. This can  
3952 be done by checking if "/oic/res" contains the "rt" of the sceneList Resource. This is depicted in  
3953 first steps of Figure 26. The sceneCollection Resource is created by the Server using some out of  
3954 bound mechanism, Client creation of Scenes is not supported at this time. This will entail defining  
3955 the Scene with an applicable list of Scene Values and the mappings for each Resource being part  
3956 of the Scene. The mapping for each Resource being part of the sceneCollection Resource is  
3957 described by a Resource called sceneMember. The sceneMember Resource contains the link to a

3958 Resource and the mapping between the Scene listed in the "sceneValues" Property and the actual  
 3959 Property value of the Resource indicated by the Link.



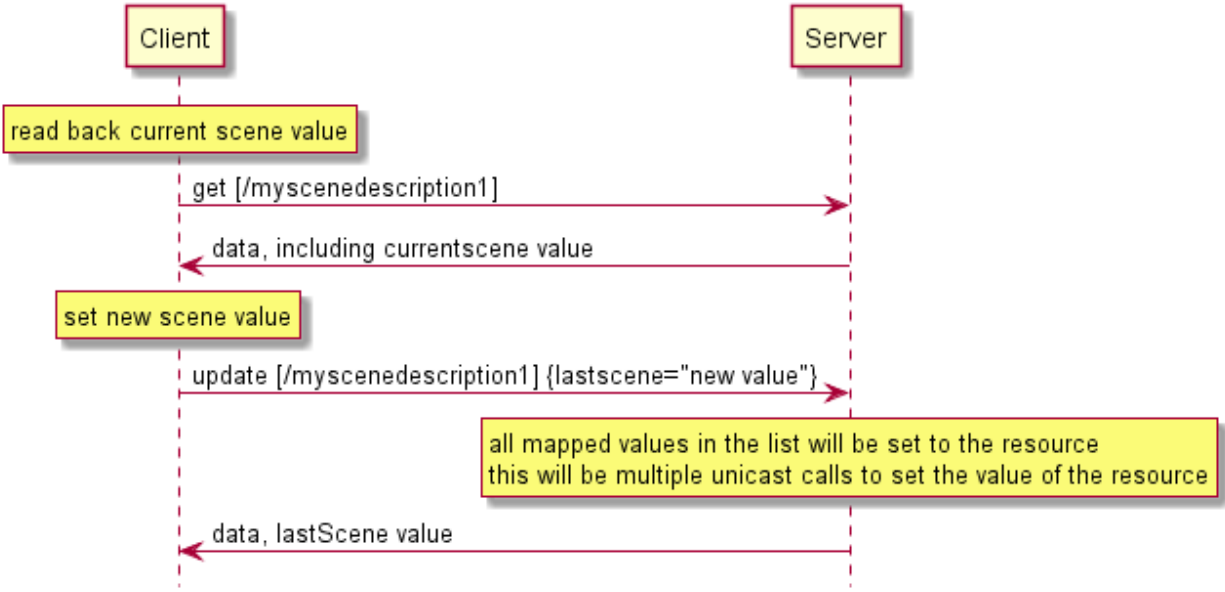
3960

3961 **Figure 26 – Interactions to check Scene support and setup of specific Scenes**

3962 **11.6.2.3 Interacting with Scenes**

3963 All capable Clients can interact with Scenes. The allowed Scene Values and the last applied Scene  
 3964 Value can be retrieved from the Server hosting the Scene. The Scene Value shall be changed by  
 3965 issuing an UPDATE operation with a payload that sets the "lastScene" Property to one of the listed  
 3966 allowed Scene Values. These steps are depicted in Figure 27. Note that the "lastScene" Property  
 3967 value does not imply that the current state of all Resources that are part of the Scene will be at the  
 3968 mapped value. This is due to that the setting the Scene Values are not modelled as actual states  
 3969 of the system. This means that another Client can change just one Resource being part of the  
 3970 Scene without having feedback that the state of the Scene is changed.

3971



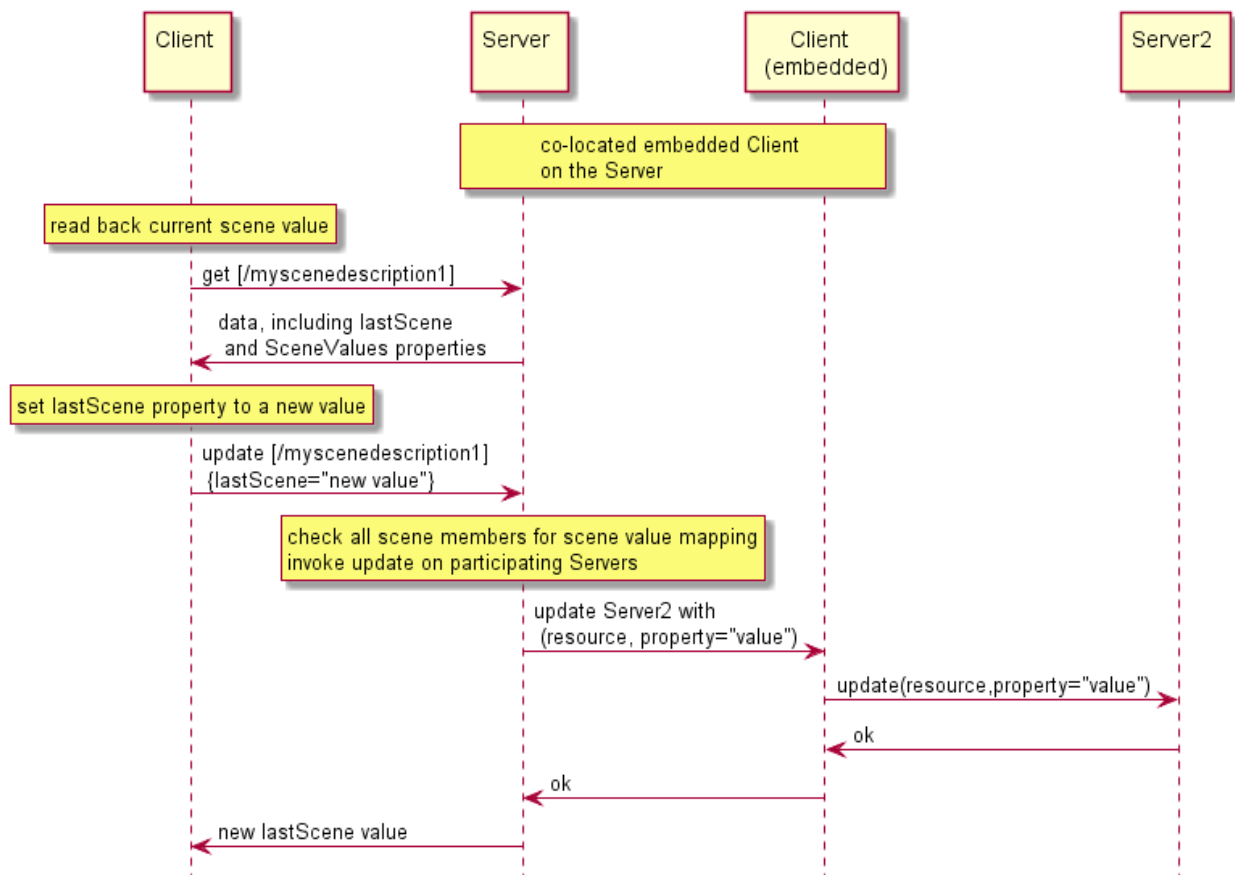
3972

3973

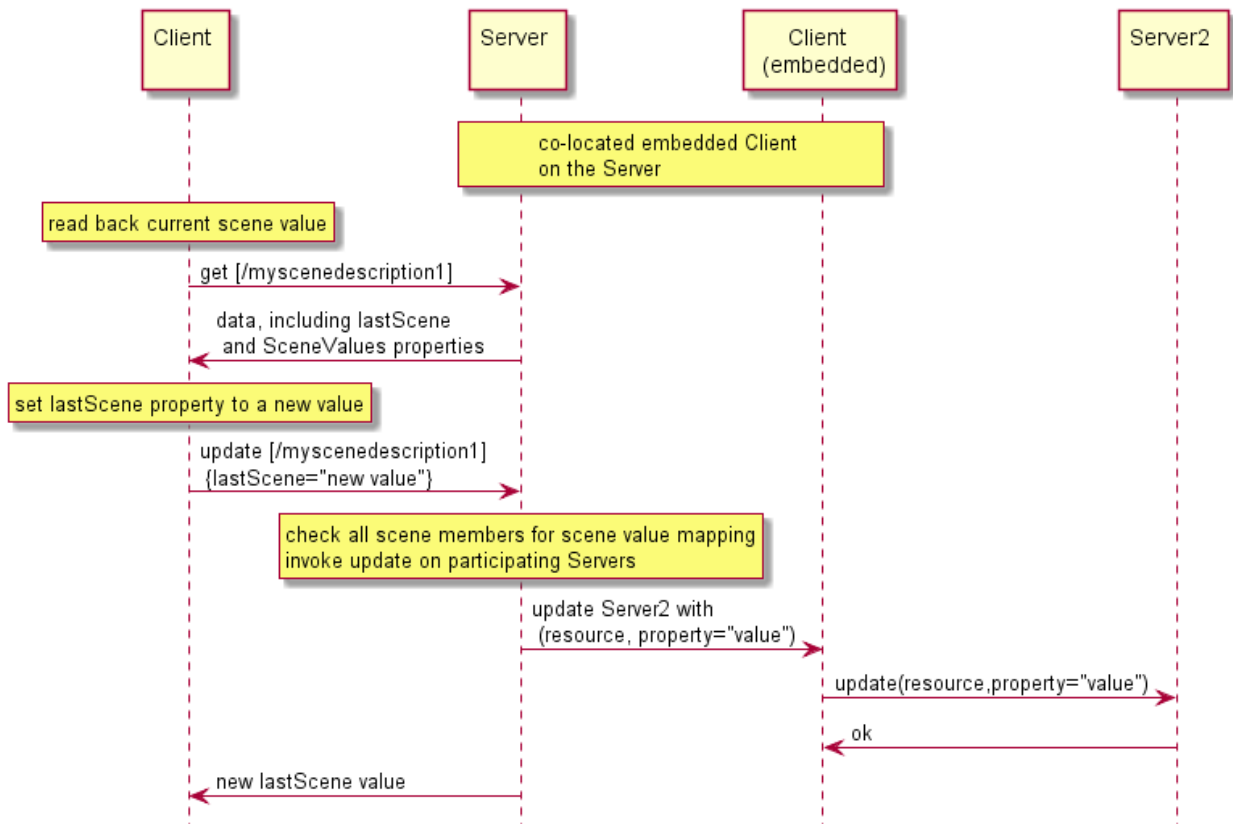
**Figure 27 – Client interactions on a specific Scene**

3974  
3975  
3976  
3977  
3978  
3979  
3980  
3981  
3982

As described previously, a Scene can reference one or more Resources (i.e., sceneMembers) that are present on one or more Servers. The Scene Members are re-evaluated each time a Scene change takes place. This evaluation is triggered by a Client that is either embedded as part of the Server hosting the Scene, or separate to the Server having knowledge of the Scene via a RETRIEVE operation, Observing the referenced Resources using the mechanism described in 11.4.2. The embedded Client located in the same Device with the Server is a general Client but interacts only with Scene functionalities. During the evaluation the mappings for the new Scene Value will be applied to the Servers which contain sceneMembers from the Scene that is being updated. This behaviour is depicted in Figure 28.



3983



3984

**Figure 28 – Interaction overview due to a Scene change**

3985

**11.6.2.4 Summary of Resource Types defined for Scene functionality**

3986

Table 40 summarizes the list of Resource Types that are part of Scenes.

3987

**Table 40 – list of Resource Types for Scenes**

3988

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

3989

**11.6.3 Security considerations**

3990

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.

3991

3992

3993

3994

3995

3996 See ISO/IEC 30118-2:2018 for details on the use of ACLs and also the mechanisms around Device  
 3997 Authentication that are necessary to ensure that the correct permissions exist for the Client to  
 3998 access the Scene Member Resource(s) on the Server.

3999 **11.7 Icons**

4000 **11.7.1 Overview**

4001 Icons are a primitive that are needed by various OCF subsystems, such as bridging. An optional  
 4002 Resource Type of "oic.r.icon" has been defined to provide a common representation of an icon  
 4003 Resource that can be used by Devices.

4004 **11.7.2 Resource**

4005 The icon Resource is as defined in Table 41.

4006 **Table 41 – 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 42.	Icon

4007

4008 Table 42 defines the details for the "oic.r.icon" Resource Type.

4009 **Table 42 – "oic.r.icon" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
<b>Mime Type</b>	"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
<b>Width</b>	"width"	"integer"	>= 1	pixels	R	Yes	Width of the icon in pixels greater than or equal to 1.
<b>Height</b>	"height"	"integer"	>= 1	pixels	R	Yes	Height of the icon in pixels greater than or equal to 1.
<b>Icon</b>	"media"	"uri"	N/A	N/A	R	Yes	URI to the location of the icon image.

4010

4011 **11.8 Introspection**

4012 **11.8.1 Overview**

4013 Introspection is a mechanism to announce the capabilities of Resources hosted on the Device.

4014 The intended usage of the Introspection Device Data (IDD) is to enable dynamic Clients e.g. Clients  
 4015 that can use the IDD) to generate dynamically a UI or dynamically create translations of the hosted  
 4016 Resources to another eco-system. Other usages of Introspection is that the information can be  
 4017 used to generate Client code. The IDD is designed to augment the existing data already on the

4018 wire. This means that existing mechanisms need to be used to get a full overview of what is  
4019 implemented in the Device. For example, the IDD does not convey information about Observability,  
4020 since that is already conveyed with the "p" Property on the Links in "/oic/res" (see 7.8.2.2.2).

4021 The IDD is recommended to be conveyed as static data. Meaning that the data does not change  
4022 during the uptime of a Device. However, when the IDD is not static, the Introspection Resource  
4023 shall be Observable and the url Property Value of "oic.wk.introspection" Resource shall change to  
4024 indicate that the IDD is changed.

4025 The IDD describes the Resources that make up the Device. For the complete list of included  
4026 Resources Table 20. The IDD is described as a OpenAPI 2.0 in JSON format file. The text in the  
4027 following bulleted list contains OpenAPI 2.0 terms, such as paths, methods etc. The OpenAPI 2.0  
4028 file shall contain the description of the Resources:

4029 – The IDD will use the HTTP syntax, e.g., define the CRUDN operation as HTTP methods and  
4030 use the HTTP status codes.

4031 – The IDD does not have to define all the status codes that indicate an error situation.

4032 – The IDD does not have to define a schema when the status code indicates that there is no  
4033 payload (see HTTP status code 204 as an example).

4034 – The paths (URLs) of the Resources in the IDD shall be without the OCF Endpoint description,  
4035 e.g. it shall not be a fully-qualified URL but only the relative path from the OCF Endpoint, aka  
4036 the "href". The relative path may include a query parameter (e.g. "?if=oic.if.ll"), in such cases  
4037 the text following (and including) the "?" delimiter shall be removed before equating to the "href"  
4038 that is conveyed by "/oic/res".

4039 – The following Resources shall be excluded in the IDD:

4040 – Resource with Resource Type: "oic.wk.res" unless 3rd party defined or optional Properties  
4041 are implemented.

4042 – Resource with Resource Type: "oic.wk.introspection".

4043 – Resources that handle Wi-Fi Easy Setup, see OCF Easy Wi-Fi Setup.

4044 – The following Resources shall be included in the IDD when optional or 3<sup>rd</sup> party defined  
4045 Properties are implemented:

4046 – Resources with type: "oic.wk.p" and "oic.wk.d" (e.g. discovery related Resources).

4047 – Security Virtual Resources from ISO/IEC 30118-2:2018.

4048 – When the Device does not expose instances of Vertical Resource Types, and does not have  
4049 any 3<sup>rd</sup> party defined Resources (see 7.8.4.4), and does not need to include Resources in the  
4050 IDD due to other clauses in this clause, then the IDD shall be an empty OpenAPI 2.0 file. An  
4051 example of an empty OpenAPI 2.0 file can be found in found in Annex F.2.

4052 – All other Resources that are individually addressable by a Client (i.e. the "href" can be resolved  
4053 and at least one operation is supported with a success path response) shall be listed in the IDD.

4054 – Per Resource the IDD shall include:

4055 – All implemented methods

4056 – For an OCF defined Resource Type, only the methods that are listed in the OpenAPI 2.0  
4057 definition are allowed to exist in the IDD. For an OCF defined Resource Type, methods  
4058 not listed in the OpenAPI 2.0 definition shall not exist in the IDD. The supported methods  
4059 contained in the IDD shall comply with the listed OCF Interfaces. For example, if the  
4060 POST method is listed in the IDD, then an OCF Interface that allows UPDATE will be  
4061 listed in the IDD.

4062 – Per supported method:

4063 – Implemented query parameters per method.



- 4064 – This includes the supported OCF Interfaces ("if") as enum values.
- 4065 – Schemas of the payload for the request and response bodies of the method.
- 4066 – Where the schema provides the representation of a batch request or response ("oic.if.b")
- 4067 the schema shall contain the representations for all Resource Types that may be
- 4068 included within the batch representation. The representations shall be provided within
- 4069 the IDD itself.
- 4070 – The schema data shall be conveyed by the OpenAPI 2.0 schema.
- 4071 – The OpenAPI 2.0 schema object shall comply with:
  - 4072 – The schemas shall be fully resolved, e.g. no references shall exist outside the
  - 4073 OpenAPI 2.0 file.
  - 4074 – The schemas shall list which OCF Interfaces are supported on the method.
  - 4075 – The schemas shall list if a Property is optional or required.
  - 4076 – The schemas shall include all Property validation keywords. Where an enum is
  - 4077 defined the enum shall contain the values supported by the Device. When vendor
  - 4078 defined extensions exist to the enum (defined in accordance to 7.8.4.4) these shall
  - 4079 be included in the enum.
  - 4080 – The schemas shall indicate if an Property is read only or read-write.
    - 4081 – By means of the readOnly schema tag belonging to the Property.
    - 4082 – Default value of readOnly is false as defined by OpenAPI 2.0.
  - 4083 – The default value of the "rt" Property shall be used to indicate the supported
  - 4084 Resource Types.
  - 4085 – oneOf and anyOf constructs are allowed to be used as part of a OpenAPI 2.0 schema
  - 4086 object. The OpenAPI 2.0 schema with oneOf and anyOf constructs can be found in
  - 4087 Annex **F.1**.
- 4088 – For Atomic Measurements (see clause 7.8.4), the following apply:
  - 4089 – The "rts" Property Value in the IDD shall include only the Resource Types the instance
  - 4090 contains and not the theoretical maximal set allowed by the schema definition.
  - 4091 – The Resources that are part of an Atomic Measurement, excluding the Atomic Measurement
  - 4092 Resource itself, shall not be added to their own individual path in the IDD, as they are not
  - 4093 individually addressable; however, the schemas for the composed Resource Types shall be
  - 4094 provided in the IDD as part of the batch response definition along with the "href" for the
  - 4095 Resource.

4096 Dynamic Resources (e.g. Resources that can be created on a request by a Client) shall have a  
 4097 URL definition which contains a URL identifier (e.g. using the {} syntax). A URL with {} identifies  
 4098 that the Resource definition applies to the whole group of Resources that may be created. The  
 4099 actual path may contain the Collection node that links to the Resource.

4100 Example of a URL with identifiers:

4101 /SceneListResURI/{SceneCollectionResURI}/{SceneMemberResURI}:

4102 When different Resource Types are allowed to be created in a Collection, then the different  
 4103 schemas for the CREATE method shall define all possible Resource Types that may be created.  
 4104 The schema construct oneOf allows the definition of a schema with selectable Resources. The  
 4105 oneOf construct allows the integration of all schemas and that only one existing sub schema shall  
 4106 be used to indicate the definition of the Resource that may be created.

4107 Example usage of oneOf JSON schema construct is shown in Figure 29:

4108 {

```

4109     "oneOf": [
4110         { <<subschema 1 definition>> },
4111         { << sub schema 2 definition >> }
4112     ...
4113     ]
4114 }

```

4115 **Figure 29 – Example usage of oneOf JSON schema**

4116 A Client using the IDD of a Device should check the version of the supported IDD of the Device.  
4117 The OpenAPI 2.0 version is indicated in each file with the tag "swagger". Example of the 2.0  
4118 supported version of the tag is: "swagger": "2.0". Later versions of the specification may reference  
4119 newer versions of the OpenAPI specification, for example 3.0.

4120 A Device shall support one Resource with a Resource Type of "oic.wk.introspection" as defined in  
4121 Table 43. The Resource with a Resource Type of "oic.wk.introspection" shall be included in the  
4122 Resource "/oic/res".

4123 An empty IDD file, e.g. no URLs are exposed, shall still have the mandatory OpenAPI 2.0 fields.  
4124 See OpenAPI specification. An example of an empty OpenAPI 2.0 file can be found in found in  
4125 Annex F.2.

4126 **Table 43 – 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

4127  
4128 Table 44 defines "oic.wk.introspection" Resource Type.

4129 **Table 44 – "oic.wk.introspection" Resource Type definition**

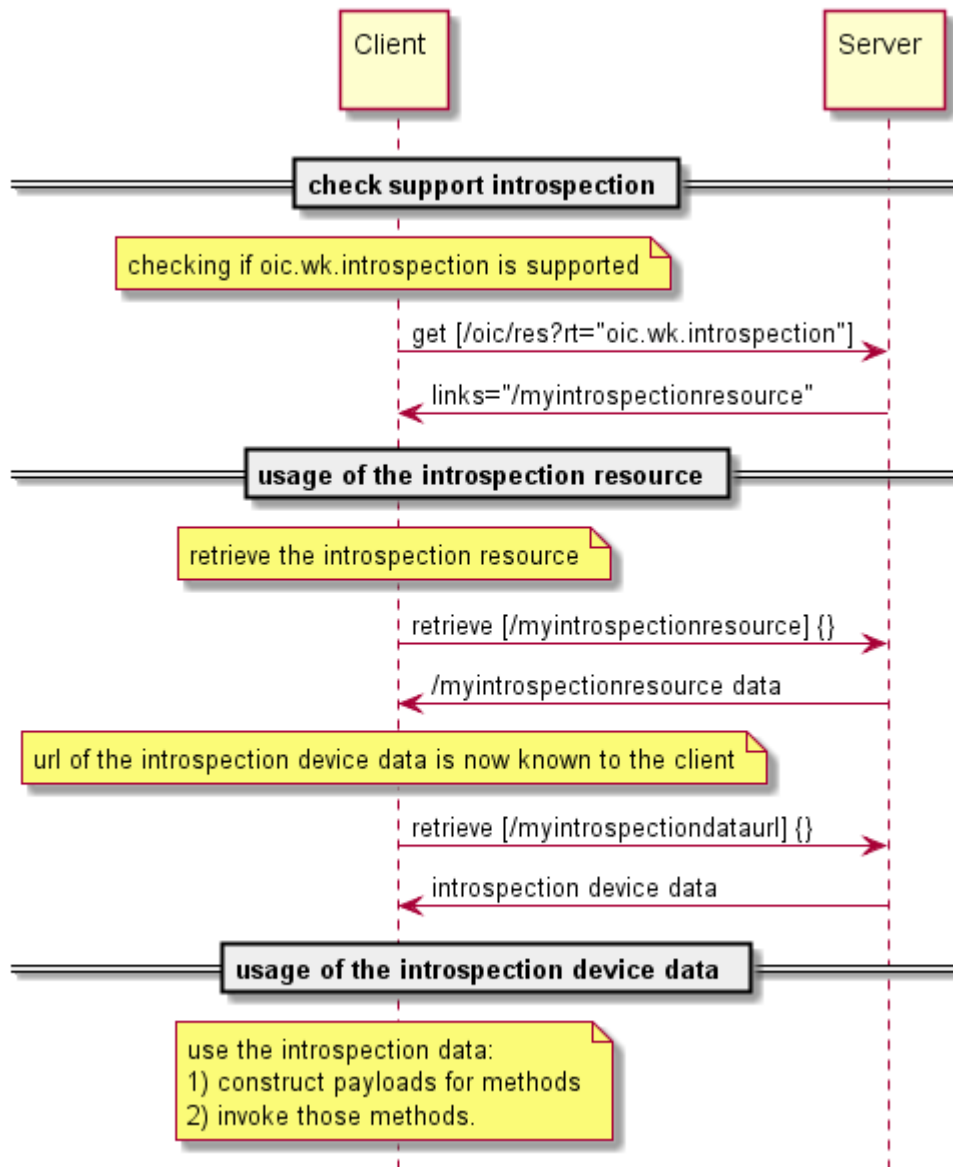
Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
urlInfo	"urlInfo"	"array"	N/A	N/A	R	Yes	array of objects
url	"url"	"string"	"uri"	N/A	R	Yes	URL to the hosted payload
protocol	"protocol"	"string"	"enum"	N/A	R	Yes	Protocol definition to retrieve the Introspection 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.

4130  
4131 **11.8.2 Usage of Introspection**

4132 The Introspection Device Data is retrieved in the following steps and as depicted in Figure 30:

- 4133 – Check if the Introspection Resource is supported and retrieve the URL of the Resource.

- 4134 – Retrieve the contents of the Introspection Resource
- 4135 – Download the Introspection Device Data from the URL specified the Introspection Resource.
- 4136 – Usage of the Introspection Device Data by the Client
- 4137



4138  
 4139 **Figure 30 – Interactions to check Introspection support and download the Introspection**  
 4140 **Device Data.**

4141 **11.9 Alerts**

4142 **11.9.1 Overview**

4143 Alerts provide a means by which a Device provides information to an interested party with regard  
 4144 to error or other conditions that the Device is experiencing. An Alert contains human readable text  
 4145 that is dependent on the Device itself and the condition being reported. A Device may expose  
 4146 discrete instances of an Alert Resource Type ("oic.r.alert") or may also expose Alerts within an

4147 Alert Collection ("oic.r.alertcollection"). If the instance of "oic.r.alertcollection" is Observable (see  
 4148 clause 7.8.2.2.2) then a Client may Observe the Collection using the mechanisms defined in clause  
 4149 11.4. As the Device adds and removes Alerts from the Collection notifications may be generated  
 4150 for any registered Observers, the format of which is dependent upon the OCF Interface used for  
 4151 the initial Observe, see clause 7.6.3.

4152 **11.9.2 Resource Types**

4153 The Alert and Alert Collection Resource Types are as defined in Table 45.

4154 **Table 45 – 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 46.	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

4155

4156 Table 46 defines the details for the "oic.r.alert" Resource Type.

4157 **Table 46 – "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.
--	--	--	--	--	--	--	--------------------------------------

4158  
 4159 The Alert Collection ("oic.r.alertcollection") Resource Type defines no Properties additional to  
 4160 those defined for all instances of a Collection in Table 9. However the Alert Collection does impose  
 4161 restrictions of the values that shall be populated in the "rt" and "rts" Properties. These are described  
 4162 in Table 47.

4163 **Table 47 – "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

4164  
 4165 **11.9.3 Example of Use**  
 4166 Consider a Device that is capable of generating Alerts; it exposes an empty instance of an Alert  
 4167 Collection ("oic.r.alertcollection"); that is the array of Links (the "links" Property) contains no items.

4168 As the Device under whatever conditions generates Alerts, the Device adds a Link to the Alert  
 4169 Resource in the instance of an Alert Collection. A Client that has discovered the Device and is  
 4170 Observing the Alert Collection using the links list OCF Interface ("oic.if.ll") will receive a notification  
 4171 containing the complete Alert Collection (not just any Links that were added). It is the responsibility  
 4172 of the Client to determine which Links were added (or removed if the Alert was removed); noting  
 4173 that the "generatedtime" Property may be used to determine the generated order. The Client then  
 4174 retrieves the Alert itself via a RETRIEVE to the "href" Link Parameter in the newly added Link to  
 4175 the Collection.

4176 See D.17 for an example of an Alert Resource and the applicable schema.

## 4177 **12 Messaging**

### 4178 **12.1 Introduction**

4179 This clause specifies the protocol messaging mapping to the CRUDN messaging operations (clause  
 4180 8) for each messaging protocol specified (e.g., CoAP.). Mapping to additional protocols is expected  
 4181 in later version of this document. All the Property information from the Resource model shall be  
 4182 carried within the message payload. This payload shall be generated in the Resource model layer  
 4183 and shall be encapsulated in the data connectivity layer. The message header shall only be used  
 4184 to describe the message payload (e.g., verb, mime-type, message payload format), in addition to  
 4185 the mandatory header fields defined in messaging protocol (e.g., CoAP) specification. If the  
 4186 message header does not support this, then this information shall also be carried in the message  
 4187 payload. Resource model information shall not be included in the message header structure unless  
 4188 the message header field is mandatory in the messaging protocol specification.

4189 When a Resource is specified with a RESTful description language like OpenAPI 2.0 then the HTTP  
 4190 syntax definitions are used in the description (e.g., HTTP syntax for the CRUDN operations, status  
 4191 codes, etc). The HTTP syntax will be mapped to the actual used web transfer protocol (e.g., CoAP).

4192 **12.2 Mapping of CRUDN to CoAP**

4193 **12.2.1 Overview**

4194 A Device implementing CoAP shall conform to IETF RFC 7252 for the methods specified in clause  
4195 12.2.3. A Device implementing CoAP shall conform to IETF RFC 7641 to implement the CoAP  
4196 Observe option. Support for CoAP block transfer when the payload is larger than the MTU is defined  
4197 in 12.2.8.

4198 **12.2.2 URIs**

4199 An OCF: URI is mapped to a coap: URI by replacing the scheme name "ocf" with "coap" if unsecure  
4200 or "coaps" if secure before sending over the network by the requestor. Similarly on the receiver  
4201 side, the scheme name is replaced with "ocf".

4202 Any query string that is present within the URI is encoded as one or more URI-Query Options as  
4203 defined in IETF RFC 7252 clause 6.4.

4204 **12.2.3 CoAP method with request and response**

4205 **12.2.3.1 Overview**

4206 Every request has a CoAP method that realizes the request. The primary methods and their  
4207 meanings are shown in Table 48, which provides the mapping of GET/POST/DELETE methods to  
4208 CREATE, RETRIEVE, UPDATE, and DELETE operations. The associated text provides the generic  
4209 behaviours when using these methods, however Resource OCF Interfaces may modify these  
4210 generic semantics. The HTTP codes in the RESTful descriptions will be translated as described in  
4211 IETF RFC 8075 clause 7 Response Code Mapping. CoAP methods not listed in Table 48 are not  
4212 supported.

4213 **Table 48 – CoAP request and response**

Method for CRUDN	(mandatory) Request data	(mandatory) Response data
<b>GET for RETRIEVE</b>	- <b>Method code:</b> GET (0.01). - <b>Request URI:</b> an existing URI for the Resource to be retrieved	- <b>Response code:</b> success (2.xx) or error (4.xx or 5.xx). - <b>Payload:</b> Resource representation of the target Resource (when successful).
<b>POST for CREATE</b>	- <b>Method code:</b> POST (0.02). - <b>Request URI:</b> an existing URI for the Resource responsible for the creation. - <b>Payload:</b> Resource presentation of the Resource to be created.	- <b>Response code:</b> success (2.xx) or error (4.xx or 5.xx). - <b>Payload:</b> the URI of the newly created Resource (when successful).
<b>POST for UPDATE</b>	- <b>Method code:</b> POST (0.02). - <b>Request URI:</b> an existing URI for the Resource to be updated. - <b>Payload:</b> representation of the Resource to be updated.	- <b>Response Code:</b> success (2.xx) or error (4.xx or 5.xx).
<b>DELETE for DELETE</b>	- <b>Method code:</b> DELETE (0.04). - <b>Request URI:</b> an existing URI for the Resource to be deleted.	- <b>Response code:</b> success (2.xx) or error (4.xx or 5.xx).

4214  
4215  
4216 **12.2.3.2 CREATE with POST**  
4217 POST shall be used only in situations where the request URI is valid, that is it is the URI of an  
4218 existing Resource on the Server that is processing the request. If no such Resource is present, the

4219 Server shall respond with an error response code of 4.xx. The use of POST for CREATE shall use  
4220 an existing request URI which identifies the Resource on the Server responsible for creation. The  
4221 URI of the created Resource is determined by the Server and provided to the Client in the response.

4222 A Client shall include the representation of the new Resource in the request payload. The new  
4223 resource representation in the payload shall have all the necessary Properties to create a valid  
4224 Resource instance, i.e. the created Resource should be able to properly respond to the valid  
4225 Request with mandatory OCF Interface (e.g., "GET with ?if=oc.if.baseline").

4226 Upon receiving the POST request, the Server shall either:

- 4227 – Create the new Resource with a new URI, respond with the new URI for the newly created  
4228 Resource and a success response code (2.xx); or
- 4229 – respond with an error response code (4.xx or 5.xx).

#### 4230 **12.2.3.3 RETRIEVE with GET**

4231 GET shall be used for the RETRIEVE operation. The GET method retrieves the representation of  
4232 the target Resource identified by the request URI.

4233 Upon receiving the GET request, the Server shall either:

- 4234 – Send back the response with the representation of the target Resource with a success response  
4235 code (2.xx); or
- 4236 – respond with an error response code (4.xx or 5.xx) or ignore it (e.g. non-applicable multicast  
4237 GET).

4238 GET is a safe method and is idempotent.

#### 4239 **12.2.3.4 UPDATE with POST**

4240 POST shall be used only in situations where the request URI is valid, that is it is the URI of an  
4241 existing Resource on the Server that is processing the request. If no such Resource is present, the  
4242 Server shall respond with an error response code of 4.xx. A client shall use POST to UPDATE  
4243 Property values of an existing Resource.

4244 Upon receiving the request, the Server shall either:

- 4245 – Apply the request to the Resource identified by the request URI in accordance with the applied  
4246 OCF Interface (i.e. POST for non-existent Properties is ignored) and send back a response with  
4247 a success response code (2.xx); or
- 4248 – respond with an error response code (4.xx or 5.xx). Note that if the representation in the payload  
4249 is incompatible with the target Resource for POST using the applied OCF Interface (i.e. the  
4250 overwrite semantic cannot be honored because of read-only Property in the payload), then the  
4251 error response code 4.xx shall be returned.

#### 4252 **12.2.3.5 DELETE with DELETE**

4253 DELETE shall be used for DELETE operation. The DELETE method requests that the Resource  
4254 identified by the request URI be deleted.

4255 Upon receiving the DELETE request, the Server shall either:

- 4256 – Delete the target Resource and send back a response with a success response code (2.xx); or
- 4257 – respond with an error response code (4.xx or 5.xx).

4258 DELETE is unsafe but idempotent (unless URIs are recycled for new instances).

4259 **12.2.4 Content-Format negotiation**

4260 The Framework mandates support of CBOR, however it allows for negotiation of the payload body  
4261 if more than one Content-Format (e.g. CBOR and JSON) is supported by an implementation. In this  
4262 case the Accept Option defined in clause 5.10.4 of IETF RFC 7252 shall be used to indicate which  
4263 Content-Format (e.g. JSON) is requested by the Client.

4264 The Content-Formats supported are shown in Table 49.

4265 **Table 49 – OCF Content-Formats**

Media Type	ID
"application/cbor"	60
"application/vnd.ocf+cbor"	10000

4266  
4267 Clients shall include a Content-Format Option in every message that contains a payload. Servers  
4268 shall include a Content-Format Option for all success (2.xx) responses with a payload body. Per  
4269 IETF RFC 7252 clause 5.5.1, Servers shall include a Content-Format Option for all error (4.xx or  
4270 5.xx) responses with a payload body unless they include a Diagnostic Payload; error responses  
4271 with a Diagnostic Payload do not include a Content-Format Option. The Content-Format Option  
4272 shall use the ID column numeric value from Table 49. An OCF vertical may mandate a specific  
4273 Content-Format Option.

4274 Clients shall also include an Accept Option in every request message. The Accept Option shall  
4275 indicate the required Content-Format as defined in Table 49 for response messages. The Server  
4276 shall return the required Content-Format if available. If the required Content-Format cannot be  
4277 returned, then the Server shall respond with an appropriate error message.

4278 **12.2.5 OCF-Content-Format-Version information**

4279 Servers and Clients shall include the OCF-Content-Format-Version Option in both request and  
4280 response messages with a payload. Clients shall include the OCF-Accept-Content-Format-Version  
4281 Option in request messages. The OCF-Content-Format-Version Option and OCF-Accept-Content-  
4282 Format-Version Option are specified as Option Numbers in the CoAP header as shown in Table 50.

4283 **Table 50 – OCF-Content-Format-Version and OCF-Accept-Content-Format-Version Option**  
4284 **Numbers**

CoAP Option Number	Name	Format	Length (bytes)
2049	OCF-Accept-Content-Format-Version	uint	2
2053	OCF-Content-Format-Version	uint	2

4285  
4286 The value of both the OCF-Accept-Content-Format-Version Option and the OCF-Content-Format-  
4287 Version Option is a two-byte unsigned integer that is used to define the major, minor and sub  
4288 versions. The major and minor versions are represented by 5 bits and the sub version is  
4289 represented by 6 bits as shown in Table 51.



4290  
4291

**Table 51 – OCF-Accept-Content-Format-Version and OCF-Content-Format-Version Representation**

	Major Version					Minor Version					Sub Version					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

4292

4293 Table 52 illustrates several examples:

4294  
4295

**Table 52 – Examples of OCF-Content-Format-Version and OCF-Accept-Content-Format-Version Representation**

OCF version	Binary representation	Integer value
"1.0.0"	"0000 1000 0000 0000"	2048
"1.1.0"	"0000 1000 0100 0000"	2112

4296

4297 The OCF-Accept-Content-Format-Version Option and OCF-Content-Format-Version Option for this  
4298 version of the document shall be "1.0.0" (i.e. "0b0000 1000 0000 0000").

4299 **12.2.6 Content-Format policy**

4300 All Devices shall support the current Content-Format Option, "application/vnd.ocf+cbor", and OCF-  
4301 Content-Format-Version "1.0.0".

4302 For backward compatibility with previous OCF-Content-Format-Version Options:

- 4303 – All Client Devices shall support OCF-Content-Format-Version Option set to "1.0.0" and higher.
- 4304 – All Client Devices shall support OCF-Accept-Content-Format-Version Option set to "1.0.0" and  
4305 higher.
- 4306 – A Client shall send a discovery request message with its Accept Option set to  
4307 "application/vnd.ocf+cbor", and its OCF-Accept-Content-Format-Version Option matching its  
4308 highest supported version.
- 4309 – A Server shall respond to a Client's discovery request that is higher than its OCF-Content-  
4310 Format-Version by responding with its Content-Format Option set to "application/vnd.ocf+cbor",  
4311 and OCF-Content-Format-Version matching its highest supported version. The response  
4312 representation shall be encoded with the OCF-Content-Format-Version matching the Server's  
4313 highest supported version.
- 4314 – A Server may support previous Content-Formats and OCF-Content-Format-Versions to support  
4315 backward compatibility with previous versions.
- 4316 – For a Server that supports multiple OCF-Content-Format-Version Options, the Server should  
4317 attempt to respond with an OCF-Content-Format-Version that matches the OCF-Accept-  
4318 Content-Format-Version of the request.

4319 For optional backward compatibility with OIC 1.1:

- 4320 – All Devices that claim backward compatibility to the OIC 1.1 specification shall support the  
4321 "application/cbor" media type.
- 4322 – For a Client supporting backward compatibility with OIC 1.1, the Client shall send a discovery  
4323 request with its Accept Option set to "application/cbor" in response to an error from an OIC 1.1  
4324 Server.

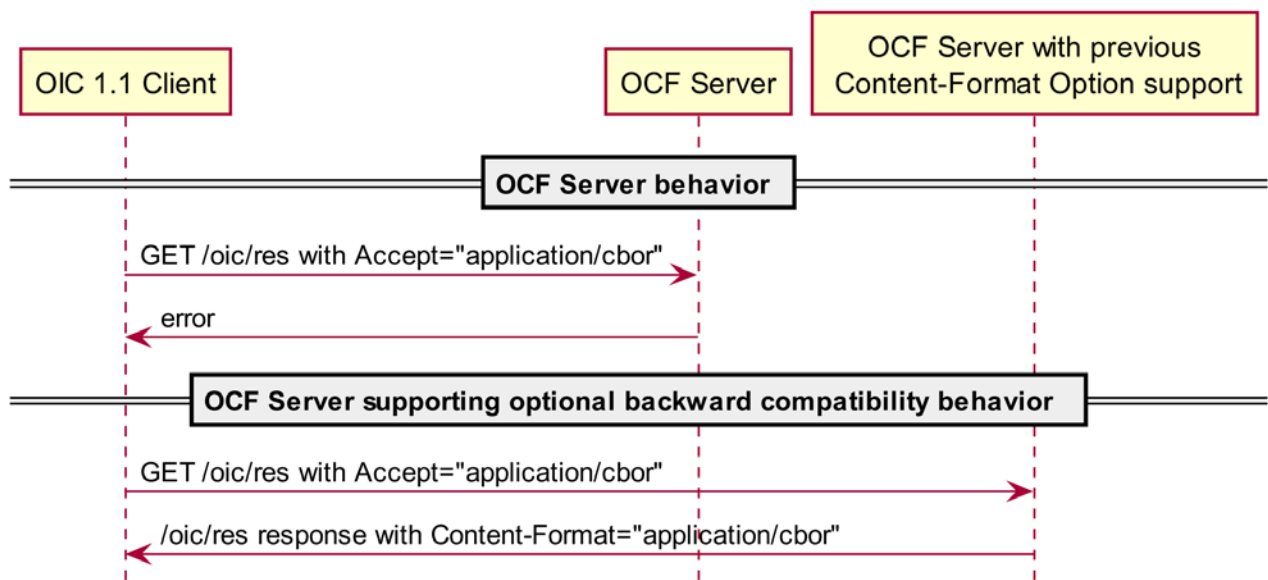
4325 – A Server supporting OIC 1.1 compatibility shall handle a Client request containing the Accept  
4326 Option ="application/cbor" by responding with its Content-Format Option set to  
4327 "application/cbor" and no OCF-Content-Format-Version Option.

4328 – For more OIC 1.1 information see Annex E.

4329 To maintain compatibility between Devices implemented to different versions of this document,  
4330 Devices should follow the policy as described in Figure 31, Figure 32 and Figure 33.

4331 The OIC 1.1 Clients and Servers represented in Figure 31 and Figure 32 support sending Content-  
4332 Format Option set to "application/cbor" and Accept Option set to "application/cbor". The OIC 1.1  
4333 Clients and Servers do not support OCF-Content-Format-Version Option, nor the OCF-Accept-  
4334 Content-Format-Version Option. The OCF Clients in Figure 32 and Figure 33 support sending  
4335 Content-Format Option set to "application/vnd.ocf+cbor", Accept Option set to  
4336 "application/vnd.ocf+cbor", OCF-Content-Format-Version Option set to "1.0.0", and OCF-Accept-  
4337 Content-Format-Version Option set to "1.0.0" (representing OCF 1.0 and later Clients). The OCF  
4338 Servers in Figure 31 and Figure 33 support sending Content-Format Option set to  
4339 "application/vnd.ocf+cbor" and OCF-Content-Format-Version Option set to "1.0.0" (representing  
4340 OCF 1.0 and later Servers).

4341

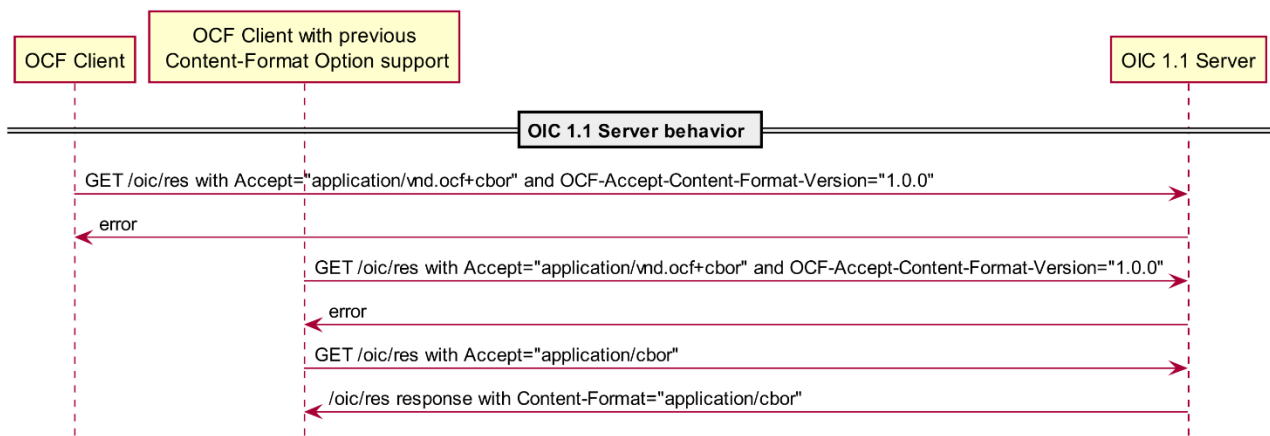


4342  
4343

4344 **Figure 31 – Content-Format Policy for OCF Servers supporting error responses and**  
4345 **backward compatibility responses**

4346

4347

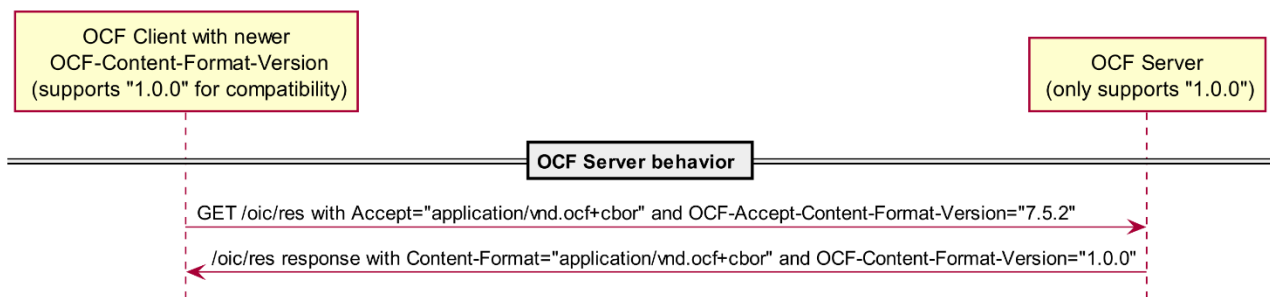


4348  
4349

**Figure 32 – Content-Format Policy for OCF Clients supporting error responses and backward compatibility responses**

4350  
4351

4352



4353

**Figure 33 – Content-Format Policy for backward compatible OCF Clients negotiating lower OCF Content-Format-Version**

4354  
4355

### 4356 12.2.7 CRUDN to CoAP response codes

4357 The mapping of CRUDN operations response codes to CoAP response codes are identical to the  
4358 response codes defined in IETF RFC 7252.

### 4359 12.2.8 CoAP block transfer

4360 Basic CoAP messages work well for the small payloads typical of light-weight, constrained IoT  
4361 devices. However scenarios can be envisioned in which an application needs to transfer larger  
4362 payloads.

4363 CoAP block-wise transfer as defined in IETF RFC 7959 shall be used by all Servers which generate  
4364 a content payload that would exceed the size of a CoAP datagram as the result of handling any  
4365 defined CRUDN operation.

4366 Similarly, CoAP block-wise transfer as defined in IETF RFC 7959 shall be supported by all Clients.  
4367 The use of block-wise transfer is applied to both the reception of payloads as well as transmission  
4368 of payloads that would exceed the size of a CoAP datagram.

4369 All blocks that are sent using this mechanism for a single instance of a transfer shall all have the  
4370 same reliability setting (i.e. all confirmable or all non-confirmable).

4371 A Client may support both the block1 (as descriptive) and block2 (as control) options as described  
4372 by IETF RFC 7959. A Server may support both the block1 (as control) and block2 (as descriptive)  
4373 options as described by IETF RFC 7959.

## 4374 **12.3 Mapping of CRUDN to CoAP serialization over TCP**

### 4375 **12.3.1 Overview**

4376 In environments where TCP is already available, CoAP can take advantage of it to provide reliability.  
4377 Also in some environments UDP traffic is blocked, so deployments may use TCP. For example,  
4378 consider a cloud application acting as a Client and the Server is located at the user's home. A  
4379 Server which already support CoAP as a messaging protocol could easily support CoAP  
4380 serialization over TCP rather than utilizing another messaging protocol. A Device implementing  
4381 CoAP Serialization over TCP shall conform to IETF RFC 8323.

### 4382 **12.3.2 URIs**

4383 When UDP is blocked, Clients are dependent on pre-configured details of the Device to determine  
4384 if the Device supports CoAP serialization over TCP. When UDP is not-blocked, a Device which  
4385 supports CoAP serialization over TCP shall populate the "eps" Parameter in the "/oic/res" response,  
4386 as defined in 10.2, with the URI scheme(s) as defined in clause 8.1 or 8.2 of IETF RFC 8323. For  
4387 the "coaps+tcp" URI scheme, as defined in clause 8.2 of IETF RFC 8323, IETF RFC 7301 shall be  
4388 used. In addition, the URIs used for CoAP serialization over TCP shall conform to 12.2.2 by  
4389 substituting the scheme names with the scheme names defined in clauses 8.1 and 8.2 of  
4390 IETF RFC 8323 respectively.

### 4391 **12.3.3 CoAP method with request and response**

4392 The CoAP methods used for CoAP serialization over TCP shall conform to 12.2.3.

### 4393 **12.3.4 Content-Format negotiation**

4394 The Content Format negotiation used for CoAP serialization over TCP shall conform to 12.2.4.

### 4395 **12.3.5 OCF-Content-Format-Version information**

4396 The OCF Content Format Version information used for CoAP serialization over TCP shall conform  
4397 to 12.2.5.

### 4398 **12.3.6 Content-Format policy**

4399 The Content Format policy used for CoAP serialization over TCP shall conform to 12.2.6.

### 4400 **12.3.7 CRUDN to CoAP response codes**

4401 The CRUDN to CoAP response codes for CoAP serialization over TCP shall conform to 12.2.7.

### 4402 **12.3.8 CoAP block transfer**

4403 The CoAP block transfer for CoAP serialization over TCP shall conform to clause 6 of  
4404 IETF RFC 8323.

### 4405 **12.3.9 Keep alive (connection health)**

4406 The Device that initiated the CoAP over TCP connection shall send a Ping message as described  
4407 in clause 5.4 in IETF RFC 8323. The Device to which the connection was made may send a Ping  
4408 message. The recipient of any Ping message shall send a Pong message as described in clause  
4409 5.4 in IETF RFC 8323.

4410 Both sides of an established CoAP over TCP connection may send subsequent Ping (and  
4411 corresponding Pong) messages.

4412 **12.4 Payload Encoding in CBOR**

4413 OCF implementations shall perform the conversion to CBOR from JSON defined schemas and to  
4414 JSON from CBOR in accordance with IETF RFC 7049 clause 4 unless otherwise specified in this  
4415 clause.

4416 Properties defined as a JSON integer shall be encoded in CBOR as an integer (CBOR major types  
4417 0 and 1). Properties defined as a JSON number shall be encoded as an integer, single- or double-  
4418 precision floating point (CBOR major type 7, sub-types 26 and 27); the choice is implementation  
4419 dependent. Half-precision floating point (CBOR major 7, sub-type 25) shall not be used. Integer  
4420 numbers shall be within the closed interval  $[-2^{53}, 2^{53}]$ . Properties defined as a JSON number  
4421 should be encoded as integers whenever possible; if this is not possible Properties defined as a  
4422 JSON number should use single-precision if the loss of precision does not affect the quality of  
4423 service, otherwise the Property shall use double-precision.

4424 On receipt of a CBOR payload, an implementation shall be able to interpret CBOR integer values  
4425 in any position. If a Property defined as a JSON integer is received encoded other than as an  
4426 integer, the implementation may reject this encoding using a final response as appropriate for the  
4427 underlying transport (e.g. 4.00 for CoAP) and thus optimise for the integer case. If a Property is  
4428 defined as a JSON number an implementation shall accept integers, single- and double-precision  
4429 floating point.

4430 **13 Security**

4431 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.

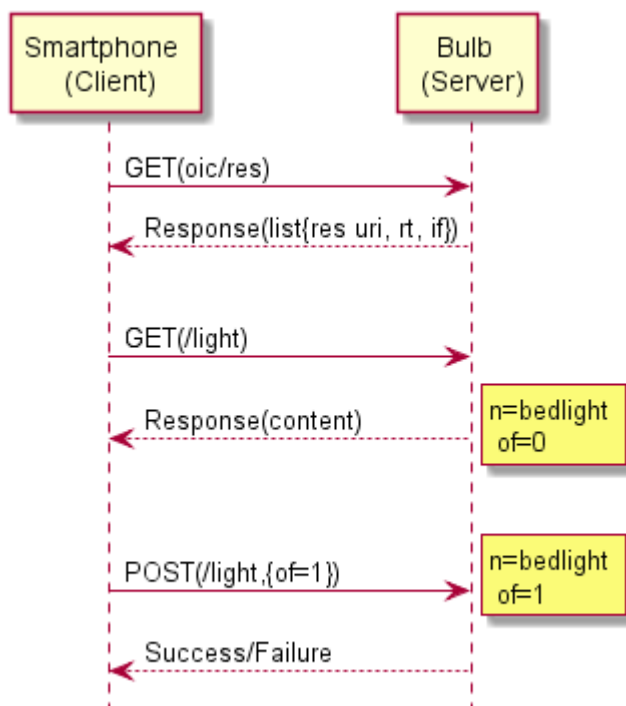
**Table A-2 – oc.example.garagedoor Resource Type definition**

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.



4452

4453

**Figure A.1 – When at home: from smartphone turn on a single light**

4454 Discovery request can be sent to *All OCF Nodes* Multicast address FF0X::158 or can be sent  
 4455 directly to the IP address of Device hosting the light Resource.

- 4456 – Smartphone sends a GET request to "/oic/res" Resource to discover all Resources hosted on  
 4457 targeted end point.
- 4458 – The end point (bulb) responds with the list of Resource URI, Resource Type and OCF Interfaces  
 4459 supported on the end point (one of the Resource is "/light" whose "rt=oic.example.light").
- 4460 – Smartphone sends a GET request to "/light" Resource to know its current state.
- 4461 – The end point responds with representation of light Resource ({n=bedlight;of=0}).
- 4462 – Smartphone changes the "of" Property of the light Resource by sending a POST request to  
 4463 "/light" Resource ("of=1").
- 4464 – On Successful execution of the request, the end point responds with the changed Resource  
 4465 representation. Else, error code is returned. Details of the error codes are defined in 12.2.7.

4466 **A.3 GroupAction execution**

4467 This example will be added when groups feature is added in later version of the document.

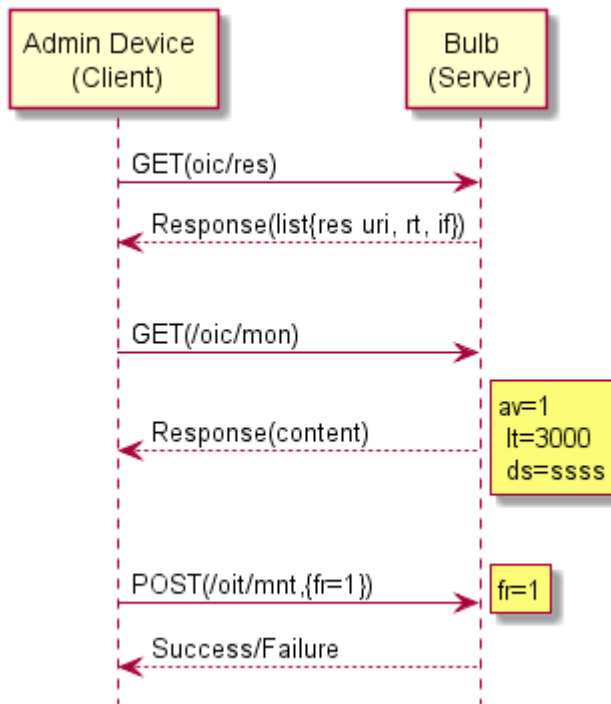
4468 **A.4 When garage door opens, turn on lights in hall; also notify smartphone**

4469 This example will be added when scripts feature is added in later version of the document.

4470 **A.5 Device management**

4471 This sequence highlights (Figure A.2) the Device management function of maintenance.

4472



4473

4474

**Figure A.2 – Device management (maintenance)**

4475 **Pre-Condition:** Admin Device has different security permissions and hence can perform Device  
 4476 management operations on the Device.

- 4477 – Admin Device sends a GET request to "/oic/res" Resource to discover all Resources hosted on  
 4478 a targeted end point (in this case Bulb).
- 4479 – The end point (bulb) responds with the list of Resource URI, Resource Type and OCF Interfaces  
 4480 supported on the end point (one of the Resources is "/oic/mnt" whose "rt=oic.wk.mnt").
- 4481 – Admin Device changes the "fr" Property of the maintenance Resource by sending a POST  
 4482 request to "/oic/mnt" Resource ("{fr=1}"). This triggers a factory reset of the end point (bulb).
- 4483 – On successful execution of the request, the end point responds with the changed Resource  
 4484 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.

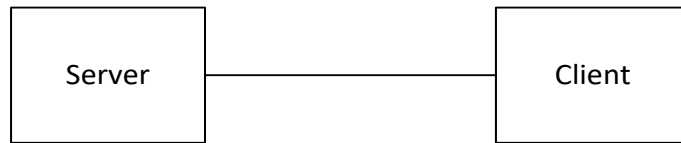


Figure B.1 – Direct interaction between Server and Client

- 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.

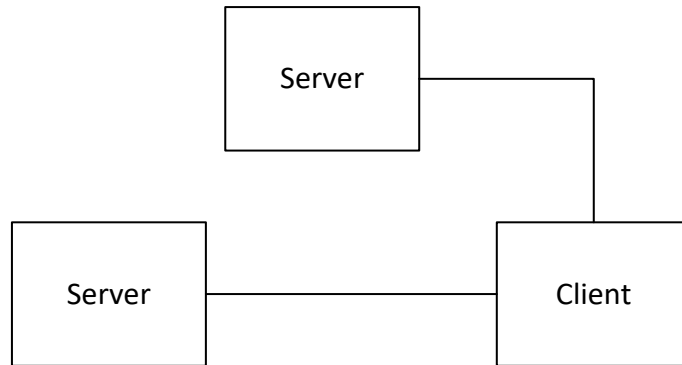


Figure B.2– Interaction between Client and Server using another Server

- 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.

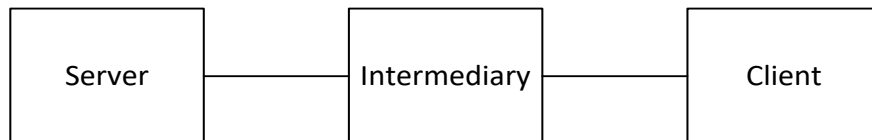
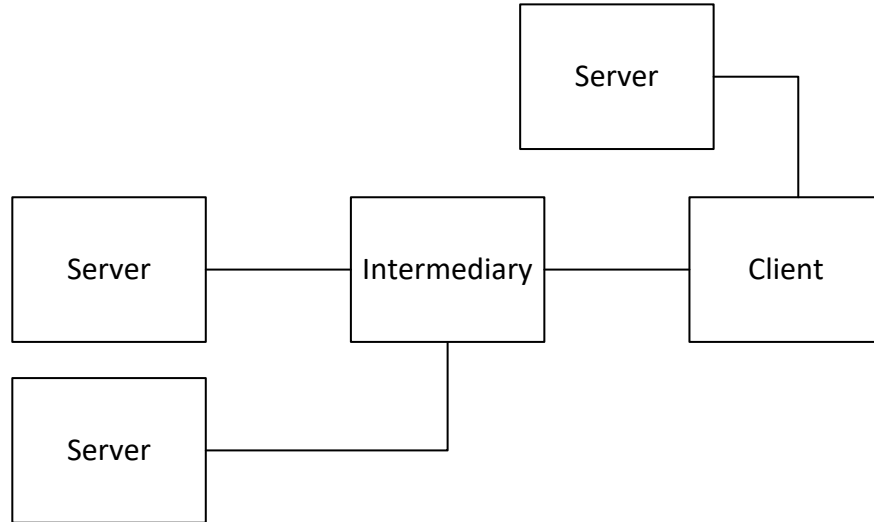


Figure B.3 – Interaction between Client and Server using Intermediary

- 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

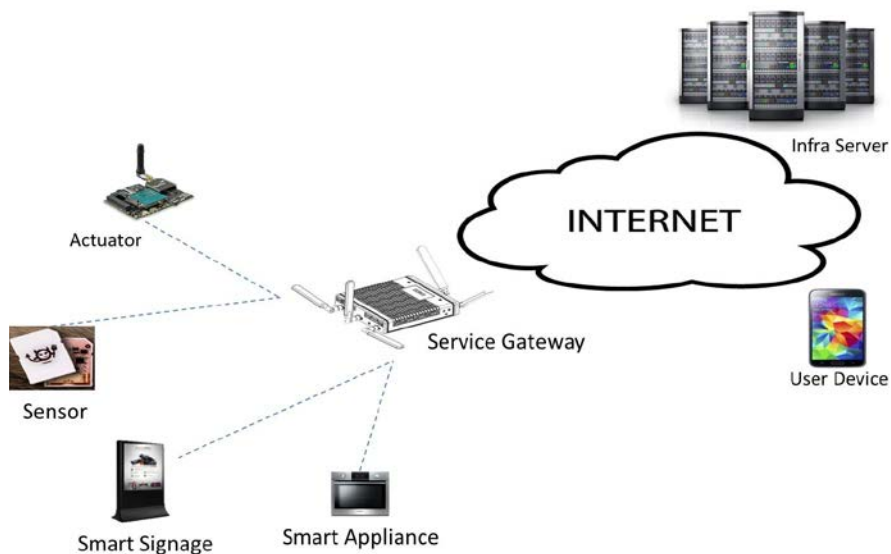
4512 smartphone first accesses a Resource Directory (RD) server to find the address to a specific  
 4513 appliance, then utilizes MQTT broker to deliver a command message to the appliance. The  
 4514 smartphone can utilize the mechanisms defined in CoRE Resource Directory such as default  
 4515 location, anycast address or DHCP to discover the Resource Directory information.



4516  
 4517 **Figure B.4 – Interaction between Client and Server using support from multiple Servers**  
 4518 **and Intermediary**

4519 **B.2 Deployment model**

4520 In deployment, Devices are deployed and interact via either wired or wireless connections. Devices  
 4521 are the physical entities that may host Resources and play one or more roles. There is no constraint  
 4522 on the structure of a deployment or number of Devices in it. Architecture is flexible and scalable  
 4523 and capable of addressing large number of Devices with different Device capabilities, including  
 4524 constrained Devices which have limited memory and capabilities. Constrained Devices are defined  
 4525 and categorized in [TCNN].



4526  
 4527 **Figure B.5 – Example of Devices**

4528 Figure B.5 depicts a typical deployment and set of Devices, which may be divided in the following  
4529 categories:

- 4530 – *Things*: Networked Devices which are able to interface with physical environments. Things are  
4531 the Devices which are primarily controlled and monitored. Examples include smart appliances,  
4532 sensors, and actuators. Things mostly take the role of Server but they may also take the role of  
4533 Client, for example in machine-to-machine communications.
- 4534 – *User Devices*: Devices employed by the users enabling the users to access Resources and  
4535 services. Examples include smart phones, tablets, and wearable devices. User Devices mainly  
4536 take the role of Client, but may also take the role of Server or Intermediary.
- 4537 – *Service Gateways*: Network equipment which take the role of Intermediary. Examples are home  
4538 gateways.
- 4539 – *Infra Servers*: Data centers residing in cloud infrastructure, which facilitate the interaction  
4540 among Devices by providing network services such as AAA, NAT traversal or discovery. It can  
4541 also play the role of Client or Intermediary.

## Annex C (informative)

### Other Resource models and OCF mapping

#### C.1 Multiple Resource models

RESTful interactions are defined dependent on the Resource model; hence, Devices require a common understanding of the Resource model for interoperability.

There are multiple Resource models defined by different organizations including OCF, OMA SpecWorks and oneM2M used in the industry, which may restrict interoperability among respective ecosystems. The main differences from Resource model are as follows:

- *Resource structure*: Resources may be defined to have Properties (e.g., oneM2M defined Resources), or may be defined as an atomic entity and not be decomposable into Properties (e.g., OMA SpecWorks defined Resources). For example, a smart light may be represented as a Resource with an on-off Property or a Resource Collection containing an on-off Resource. In the former, on-off Property doesn't have a URI of its own and can only be accessed indirectly via the Resource. In the latter, being a Resource itself, on-off Resource is assigned its own URI and can be directly manipulated.
- *Resource name & type*: Resources may be allowed to be named freely and have their characteristics indicated using a Resource Type Property (e.g., as defined in oneM2M). Alternatively, the name of Resources may be defined a priori in a way that the name by itself is indicative of its characteristic (e.g., as defined by OMA SpecWorks). For example, in oneM2M Resource model, a smart light can be named with no restrictions, such as "LivingRoomLight\_1" but in OMA SpecWorks Resource model it is required to have the fixed Object name with numerical Object ID of OMA SpecWorks Light Control ("3311"). Consequently, it's likely that in the former case the data path in URI is freely defined and in the latter case it is predetermined.
- *Resource hierarchy*: Resources may be allowed to be organized in hierarchy where a Resource contains another Resource with a parent-child relationship (e.g., in oneM2M definition of Resource model). Resources may also be required to have a flat structure and associate with other Resources only by referencing their links.

In addition, different organizations use different syntax and define different features (e.g., Resource OCF Interface), which preclude interoperability.

#### C.2 OCF approach for support of multiple Resource models

In order to expand the IoT ecosystem the Framework takes an inclusive approach for interworking with existing Resource models. Specifically, the Framework defines a Resource model while providing a mechanism to easily map to other models. By embracing existing Resource models OCF is inclusive of existing ecosystems while allowing for the transition toward definition of a comprehensive Resource model integrating all ecosystems.

The following OCF characteristics enable support of other Resource models:

- *Resource model is the superset of multiple models*: the Resource model is defined as the superset of existing Resource models. In other words, any existing Resource model can be mapped to a subset of Resource model concepts.
- *Framework may allow for Resource model negotiation*: the Client and Server exchange the information about what Resource model(s) each supports. Based on the exchanged information, the Client and Server choose a Resource model to perform RESTful interactions or to perform translation. This feature is out of scope of the current version of this document, however, the following is a high level description for Resource model negotiation.

### 4588 **C.3 Resource model indication**

4589 The Client and server exchange the information about what Resource model(s) each supports.  
4590 Based on the exchanged information, the Client and Server choose a Resource model to perform  
4591 RESTful interactions or to perform translation. The exchange could be part of discovery and  
4592 negotiation. Based on the exchange, the Client and Server follow a procedure to ensure  
4593 interoperability among them. They may choose a common Resource model or execute translation  
4594 between Resource models.

4595 – *Resource model schema exchange*: The Client and Server may share the Resource model  
4596 information when they initiate a RESTful interaction. They may exchange the information about  
4597 which Resource model they support as part of session establishment procedures. Alternatively,  
4598 each request or response message may carry the indication of which Resource model it is using.  
4599 For example, [COAP] defines Content-Format option to indicate the representation format such  
4600 as "application/json". It's possible to extend the Content-Format Option to indicate the Resource  
4601 model used with the representation format such as "application/ipsso-json".

4602 – *Ensuing procedures*: After the Client and Server exchange the Resource model information,  
4603 they perform a suitable procedure to ensure interoperability among them. The simplest way is  
4604 to choose a Resource model supported by both the Client and Server. In case there is no  
4605 common Resource model, the Client and Server may interact through a 3rd party.

4606 In addition to translation which can be resource intensive, a method based on profiles can be used  
4607 in which an OCF implementation can accommodate multiple profiles and hence multiple  
4608 ecosystems.

4609 – *Resource Model Profile*: the Framework defines Resource model profiles and implementers or  
4610 users choose the active profile. The chosen profile constraints the Device to strict rules in how  
4611 Resources are defined, instantiated and interacted with. This would allow for interoperation with  
4612 devices from the ecosystem identified by the profile (e.g., OMA SpecWorks, OneM2M etc.).  
4613 Although this enables a Device to participate in and be part of any given ecosystem, this scheme  
4614 does not allow for generic interoperability at runtime. While this approach may be suitable for  
4615 resource constrained devices, more resource capable devices are expected to support more  
4616 than one profile.

### 4617 **C.4 An Example Profile (OMA SpecWorks profile)**

#### 4618 **C.4.1 Overview**

4619 OMA SpecWorks defines smart objects that have specific Resources and they take values  
4620 determined by the data type of that Resource. The smart object specification defines a category of  
4621 such objects. Each Resource represents a characteristic of the smart object being modelled.

4622 While the terms may be different, there are equivalent concepts in OCF to represent these terms.  
4623 This clause provides the equivalent OCF terms and then frames the OMA SpecWorks smart object  
4624 in OCF terms.

4625 The OMA SpecWorks object Light Control defined in clause 16 of the OMA SpecWorks Smart  
4626 Objects 1.0 is used as the reference example.

### 4627 **C.5 Conceptual equivalence**

4628 The OMA SpecWorks smart object definition is equivalent to a Resource Type definition which  
4629 defines the relevant characteristics of an entity being modelled. The specific OMA SpecWorks  
4630 Resource is equivalent to a Property that like an OMA SpecWorks Resource has a defined data  
4631 type, enumeration of acceptable values, units, a general description and access modes (based on  
4632 the OCF Interface).

4633 The general method for developing the equivalent Resource Type from an OMA SpecWorks Smart  
 4634 Object definition is to ignore the Object ID and replace the Object URN with an OCF "." (dot)  
 4635 separated name that incorporates the OMA SpecWorks object. Alternatively the Object URN can  
 4636 be used as the Resource Type ID as is (as long as the URN does not contain any "." (dots)) – using  
 4637 the same Object URN as the Resource Type ID allows for compatibility when interacting with an  
 4638 OMA SpecWorks compliant device. The object URN based naming does not have any bearing for  
 4639 OCF to OCF interoperability and so the OCF format is preferred – for OCF to OCF interoperability  
 4640 only the data model consistency is required.

4641 Two models are available to render OMA SpecWorks objects into OCF.

- 4642 – One is where the OMA SpecWorks Smart Object represents a Resource. In this case, the IP  
 4643 Smart Object is regarded as a Resource with the Resource Type matching the description of  
 4644 the Smart Object. Furthermore, each Resource in the OMA SpecWorks definition is represented  
 4645 as a Property in the Resource Type (the OMA SpecWorks Resource ID is replaced with a string  
 4646 representing the Property). This is the preferred approach when the OMA SpecWorks Data  
 4647 Model is expressed in the Resource Model.
- 4648 – The other approach is to model an OMA SpecWorks Smart Object as a Collection. Each OMA  
 4649 SpecWorks Resource is then modelled as a Resource with a Resource Type that matches the  
 4650 definition of the OMA SpecWorks Resource. Each of these Resource instances are then bound  
 4651 to the Collection that represents this OMA SpecWorks Smart Object.

4652 C.5.1 is an example showing how an OMA SpecWorks LightControl Object is modelled as a  
 4653 Resource.

4654 **C.5.1 Resource Type: Light Control**

4655 Description: This Object is used to control a light source, such as a LED or other light. It allows a  
 4656 light to be turned on or off and its dimmer setting to be controlled as a percentage value between  
 4657 0 and 100. An optional colour setting enables a string to be used to indicate the desired colour.  
 4658 Table C-1 and Table C-2 define the Resource Type and its Properties, respectively.

4659 **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

4660

4661 **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

4662

4663

4664  
4665  
4666  
4667

## Annex D (normative)

### Resource Type definitions

#### 4668 D.1 List of Resource Type definitions

4669 All the clauses in Annex D and Annex E describe the Resource Types with a RESTful API definition  
4670 language. The Resource Type definitions presented in Annex D and Annex E are formatted for  
4671 readability, and so may appear to have extra line breaks. Table D-1 contains the list of defined  
4672 Core Common Resources in this document.

4673

**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
Software Update	"oic.r.softwareupdate"	D.19

4674  
4675

#### 4676 D.2 Atomic Measurement links list representation

##### 4677 D.2.1 Introduction

4678 The oic.if.baseline OCF Interface exposes a representation of the links and  
4679 the Common Properties of the Atomic Measurement Resource.  
4680

##### 4681 D.2.2 Example URI

4682 /AtomicMeasurementResURI



### 4683 D.2.3 Resource type

4684 The Resource Type is defined as: "oic.wk.atomicmeasurement".

### 4685 D.2.4 OpenAPI 2.0 definition

```
4686 {
4687   "swagger": "2.0",
4688   "info": {
4689     "title": "Atomic Measurement links list representation",
4690     "version": "2019-03-04",
4691     "license": {
4692       "name": "OCF Data Model License",
4693       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
4694       "x-copyright": "Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights reserved."
4695     },
4696     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
4697   },
4698   "schemes": ["http"],
4699   "consumes": ["application/json"],
4700   "produces": ["application/json"],
4701   "paths": {
4702     "/AtomicMeasurementResURI?if=oic.if.ll": {
4703       "get": {
4704         "description": "The oic.if.ll OCF Interface exposes a representation
4705 of the Links",
4706         "parameters": [
4707           {
4708             "$ref": "#/parameters/interface-all"
4709           }
4710         ],
4711         "responses": {
4712           "200": {
4713             "description": "",
4714             "x-example": [{
4715               "href": "/temperature",
4716               "rt": ["oic.r.temperature"],
4717               "if": ["oic.if.s", "oic.if.baseline"]
4718             }],
4719             {
4720               "href": "/bodylocation",
4721               "rt": ["oic.r.body.location.temperature"],
4722               "if": ["oic.if.s", "oic.if.baseline"]
4723             },
4724             {
4725               "href": "/timestamp",
4726               "rt": ["oic.r.time.stamp"],
4727               "if": ["oic.if.s", "oic.if.baseline"]
4728             }
4729           ],
4730           "schema": {
4731             "$ref": "#/definitions/links"
4732           }
4733         }
4734       }
4735     },
4736     "/AtomicMeasurementResURI?if=oic.if.b": {
4737       "get": {
4738         "description": "The oic.if.b OCF Interface returns data items
4739 retrieved from Resources pointed to by the Links.\n",
4740         "parameters": [
4741           {
4742             "$ref": "#/parameters/interface-all"
4743           }
4744         ],
4745         "responses": {
4746           "200": {
4747             "description": "Normal response, no errors, all
4748 Properties are returned correctly\n",
4749             "x-example": [{
4750               "href": "/temperature",
```

```

4751         "rep": {
4752             "temperature": 38,
4753             "units": "C",
4754             "range": [25, 45]
4755         }
4756     },
4757     {
4758         "href": "/bodylocation",
4759         "rep": {
4760             "bloc": "ear"
4761         }
4762     },
4763     {
4764         "href": "/timestamp",
4765         "rep": {
4766             "timestamp": "2007-04-05T14:30+09:00"
4767         }
4768     }
4769 ],
4770 "schema": {
4771     "$ref": "#/definitions/batch-retrieve"
4772 }
4773 },
4774 }
4775 },
4776 "/AtomicMeasurementResURI?if=oic.if.baseline": {
4777     "get": {
4778         "description": "The oic.if.baseline OCF Interface exposes a
4779 representation of the links and\nthe Common Properties of the Atomic Measurement Resource.\n",
4780         "parameters": [
4781             {
4782                 "$ref": "#/parameters/interface-all"
4783             }
4784 ],
4785         "responses": {
4786             "200": {
4787                 "description": "",
4788                 "x-example": {
4789                     "rt": ["oic.wk.atomicmeasurement"],
4790                     "if": ["oic.if.b", "oic.if.ll",
4791 "oic.if.baseline"],
4792                     "rts": ["oic.r.temperature",
4793 "oic.r.body.location.temperature", "oic.r.time.stamp"],
4794                     "rts-m": ["oic.r.temperature",
4795 "oic.r.body.location.temperature", "oic.r.time.stamp"],
4796                     "links": [{
4797                         "href": "/temperature",
4798                         "rt": ["oic.r.temperature"],
4799                         "if": ["oic.if.s", "oic.if.baseline"]
4800                     },
4801                     {
4802                         "href": "/bodylocation",
4803                         "rt":
4804 ["oic.r.body.location.temperature"],
4805                         "if": ["oic.if.s", "oic.if.baseline"]
4806                     },
4807                     {
4808                         "href": "/timestamp",
4809                         "rt": ["oic.r.time.stamp"],
4810                         "if": ["oic.if.s", "oic.if.baseline"]
4811                     }
4812                 ]
4813             },
4814             "schema": {
4815                 "$ref": "#/definitions/baseline"
4816             }
4817         }
4818     }
4819 },
4820 "parameters": {

```

```

4822         "interface-all": {
4823             "in": "query",
4824             "name": "if",
4825             "type": "string",
4826             "enum": ["oic.if.b", "oic.if.ll", "oic.if.baseline"]
4827         }
4828     },
4829     "definitions": {
4830         "links": {
4831             "type": "array",
4832             "items": {
4833                 "$ref": "#/definitions/oic.oic-link"
4834             }
4835         },
4836         "batch-retrieve": {
4837             "title": "Collection Batch Retrieve Format (auto merged)",
4838             "minItems": 1,
4839             "items": {
4840                 "additionalProperties": true,
4841                 "properties": {
4842                     "href": {
4843                         "$ref":
4844 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4845 schema.json#/definitions/href"
4846                     },
4847                     "rep": {
4848                         "oneOf": [{
4849                             "description": "The response payload from a
4850 single Resource",
4851                             "type": "object"
4852                         },
4853                         {
4854                             "description": " The response payload from a
4855 Collection (batch) Resource",
4856                             "items": {
4857                                 "properties": {
4858                                     "anchor": {
4859                                         "$ref":
4860 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4861 schema.json#/definitions/anchor"
4862                                     },
4863                                     "di": {
4864                                         "$ref":
4865 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4866 schema.json#/definitions/di"
4867                                     },
4868                                     "eps": {
4869                                         "$ref":
4870 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4871 schema.json#/definitions/eps"
4872                                     },
4873                                     "href": {
4874                                         "$ref":
4875 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4876 schema.json#/definitions/href"
4877                                     },
4878                                     "if": {
4879                                         "description": "The OCF
4880 Interface set supported by this Resource",
4881                                         "items": {
4882                                             "enum": [
4883                                                 "oic.if.baseline",
4884                                                 "oic.if.ll",
4885                                                 "oic.if.b",
4886                                                 "oic.if.rw",
4887                                                 "oic.if.r",
4888                                                 "oic.if.a",
4889                                                 "oic.if.s"],
4890                                             "type":
4891 "string"
4892

```

```

4893                                     },
4894                                     "minItems": 1,
4895                                     "uniqueItems": true,
4896                                     "type": "array"
4897                                     },
4898                                     "ins": {
4899                                         "$ref":
4900 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4901 schema.json#/definitions/ins"
4902                                     },
4903                                     "p": {
4904                                         "$ref":
4905 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4906 schema.json#/definitions/p"
4907                                     },
4908                                     "rel": {
4909                                         "description": "The relation of the target URI
4910 referenced by the Link to the context URI",
4911                                         "oneOf": [
4912                                             {
4913                                                 "$ref":
4914 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4915 schema.json#/definitions/rel_array"
4916                                             },
4917                                             {
4918                                                 "$ref":
4919 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4920 schema.json#/definitions/rel_string"
4921                                             }
4922                                         ]
4923                                     },
4924                                     "rt": {
4925                                         "description":
4926 "Resource Type of the Resource",
4927                                         "items": {
4928                                             "maxLength":
4929 64,
4930                                             "type":
4931 "string"
4932                                         },
4933                                         "minItems": 1,
4934                                         "uniqueItems": true,
4935                                         "type": "array"
4936                                     },
4937                                     "title": {
4938                                         "$ref":
4939 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4940 schema.json#/definitions/title"
4941                                     },
4942                                     "type": {
4943                                         "$ref":
4944 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
4945 schema.json#/definitions/type"
4946                                     }
4947                                     },
4948                                     "required": [
4949                                         "href",
4950                                         "rt",
4951                                         "if"
4952                                     ],
4953                                     "type": "object"
4954                                 },
4955                                 "type": "array"
4956                            }
4957                    }
4958            },
4959            "required": [
4960                "href",
4961                "rep"
4962            ],
4963            "type": "object"

```

```

4964         },
4965         "type": "array"
4966     },
4967     "baseline": {
4968         "properties": {
4969             "links": {
4970                 "description": "A set of simple or individual Links.",
4971                 "items": {
4972                     "$ref": "#/definitions/oic.oic-link"
4973                 },
4974                 "type": "array"
4975             },
4976             "n": { "$ref" :
4977 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
4978 schema.json#/definitions/n"},
4979             "id": { "$ref" :
4980 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
4981 schema.json#/definitions/id"},
4982             "rt": {
4983                 "description": "Resource Type of this Resource",
4984                 "items": {
4985                     "enum": ["oic.wk.atomicmeasurement"],
4986                     "type": "string",
4987                     "maxLength": 64
4988                 },
4989                 "minItems": 1,
4990                 "readOnly": true,
4991                 "uniqueItems": true,
4992                 "type": "array"
4993             },
4994             "rts": {
4995                 "description": "An array of Resource Types that are supported
4996 within an array of Links exposed by the Resource",
4997                 "items": {
4998                     "maxLength": 64,
4999                     "type": "string"
5000                 },
5001                 "minItems": 1,
5002                 "readOnly": true,
5003                 "uniqueItems": true,
5004                 "type": "array"
5005             },
5006             "rts-m": {
5007                 "description": "An array of Resource Types that are mandatory
5008 to be exposed within an array of Links exposed by the Resource",
5009                 "items": {
5010                     "maxLength": 64,
5011                     "type": "string"
5012                 },
5013                 "minItems": 1,
5014                 "readOnly": true,
5015                 "uniqueItems": true,
5016                 "type": "array"
5017             },
5018             "if": {
5019                 "description": "The OCF Interface set supported by this
5020 Resource",
5021                 "items": {
5022                     "enum": ["oic.if.b", "oic.if.ll", "oic.if.baseline"],
5023                     "type": "string"
5024                 },
5025                 "minItems": 3,
5026                 "readOnly": true,
5027                 "uniqueItems": true,
5028                 "type": "array"
5029             }
5030         },
5031         "type": "object",
5032         "required": [
5033             "rt",
5034             "if",

```

```

5035     "links"
5036   ]
5037     },
5038     "oic.oic-link": {
5039       "properties": {
5040         "anchor": {
5041           "$ref":
5042 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5043 schema.json#/definitions/anchor"
5044         },
5045         "di": {
5046           "$ref":
5047 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5048 schema.json#/definitions/di"
5049         },
5050         "eps": {
5051           "$ref":
5052 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5053 schema.json#/definitions/eps"
5054         },
5055         "href": {
5056           "$ref":
5057 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5058 schema.json#/definitions/href"
5059         },
5060         "if": {
5061           "description": "The OCF Interface set supported by this
5062 Resource",
5063           "items": {
5064             "enum": [
5065               "oic.if.baseline",
5066               "oic.if.ll",
5067               "oic.if.b",
5068               "oic.if.rw",
5069               "oic.if.r",
5070               "oic.if.a",
5071               "oic.if.s"],
5072             "type": "string"
5073           },
5074           "minItems": 1,
5075           "uniqueItems": true,
5076           "type": "array"
5077         },
5078         "ins": {
5079           "$ref":
5080 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5081 schema.json#/definitions/ins"
5082         },
5083         "p": {
5084           "$ref":
5085 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5086 schema.json#/definitions/p"
5087         },
5088         "rel": {
5089           "description": "The relation of the target URI referenced by the Link to the context URI",
5090           "oneOf": [
5091             {
5092               "$ref":
5093 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5094 schema.json#/definitions/rel_array"
5095             },
5096             {
5097               "$ref":
5098 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5099 schema.json#/definitions/rel_string"
5100             }
5101           ]
5102         },
5103         "rt": {
5104           "description": "Resource Type of the Resource",
5105           "items": {

```

```

5106         "maxLength": 64,
5107         "type": "string"
5108     },
5109     "minItems": 1,
5110     "uniqueItems": true,
5111     "type": "array"
5112 },
5113     "title": {
5114         "$ref":
5115         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5116         schema.json#/definitions/title"
5117     },
5118     "type": {
5119         "$ref":
5120         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5121         schema.json#/definitions/type"
5122     }
5123 },
5124     "required": [
5125         "href",
5126         "rt",
5127         "if"
5128     ],
5129     "type": "object"
5130 }
5131 }
5132 }
5133 }

```

## 5134 D.2.5 Property definition

5135 Table D-2 defines the Properties that are part of the "oic.wk.atomicmeasurement" Resource Type.

5136 **Table D-2 – The Property definitions of the Resource with type "rt" =**  
5137 **"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	

5138 **D.2.6 CRUDN behaviour**

5139 Table D-3 defines the CRUDN operations that are supported on the "oic.wk.atomicmeasurement"  
5140 Resource Type.

5141 **Table D-3 – The CRUDN operations of the Resource with type "rt" =**  
5142 **"oic.wk.atomicmeasurement".**

Create	Read	Update	Delete	Notify
	get			observe

5143 **D.3 Collection**

5144 **D.3.1 Introduction**

5145 Collection Resource Type contains Properties and Links.  
5146 The oic.if.baseline OCF Interface exposes a representation of  
5147 the Links and the Properties of the Collection Resource itself  
5148

5149 **D.3.2 Example URI**

5150 /CollectionResURI

5151 **D.3.3 Resource type**

5152 The Resource Type is defined as: "oic.wk.col".



#### 5153 D.3.4 OpenAPI 2.0 definition

```
5154 {
5155   "swagger": "2.0",
5156   "info": {
5157     "title": "Collection",
5158     "version": "2019-03-04",
5159     "license": {
5160       "name": "OCF Data Model License",
5161       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
5162       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
5163     },
5164     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
5165   },
5166   "schemes": [
5167     "http"
5168   ],
5169   "consumes": [
5170     "application/json"
5171   ],
5172   "produces": [
5173     "application/json"
5174   ],
5175   "paths": {
5176     "/CollectionResURI?if=oic.if.ll" : {
5177       "get": {
5178         "description": "Collection Resource Type contains Properties and Links.\n\nThe oic.if.ll OCF
5179 Interface exposes a representation of the Links\n",
5180         "parameters": [
5181           {
5182             "$ref": "#/parameters/interface-all"
5183           }
5184         ],
5185         "responses": {
5186           "200": {
5187             "description" : "",
5188             "x-example": [
5189               {
5190                 "href": "/switch",
5191                 "rt": ["oic.r.switch.binary"],
5192                 "if": ["oic.if.a", "oic.if.baseline"],
5193                 "eps": [
5194                   {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
5195                   {"ep": "coaps://[fe80::b1d6]:1122"},
5196                   {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
5197                 ]
5198               },
5199               {
5200                 "href": "/airFlow",
5201                 "rt": ["oic.r.airflow"],
5202                 "if": ["oic.if.a", "oic.if.baseline"],
5203                 "eps": [
5204                   {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
5205                   {"ep": "coaps://[fe80::b1d6]:1122"},
5206                   {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
5207                 ]
5208               }
5209             ],
5210             "schema": {
5211               "$ref": "#/definitions/slinks"
5212             }
5213           }
5214         }
5215       }
5216     },
5217     "/CollectionResURI?if=oic.if.baseline" : {
5218       "get": {
5219         "description": "Collection Resource Type contains Properties and Links.\n\nThe oic.if.baseline
5220 OCF Interface exposes a representation of\n\nthe Links and the Properties of the Collection Resource
5221 itself\n",
5222         "parameters": [
```

```

5223     {
5224         "$ref": "#/parameters/interface-all"
5225     }
5226 ],
5227 "responses": {
5228     "200": {
5229         "description": "",
5230         "x-example": {
5231             "rt": ["oic.wk.col"],
5232             "if": ["oic.if.ll", "oic.if.b", "oic.if.baseline"],
5233             "rts": ["oic.r.switch.binary", "oic.r.airflow"],
5234             "rts-m": ["oic.r.switch.binary"],
5235             "links": [
5236                 {
5237                     "href": "/switch",
5238                     "rt": ["oic.r.switch.binary"],
5239                     "if": ["oic.if.a", "oic.if.baseline"],
5240                     "eps": [
5241                         {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
5242                         {"ep": "coaps://[fe80::b1d6]:1122"},
5243                         {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
5244                     ]
5245                 },
5246                 {
5247                     "href": "/airFlow",
5248                     "rt": ["oic.r.airflow"],
5249                     "if": ["oic.if.a", "oic.if.baseline"],
5250                     "eps": [
5251                         {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
5252                         {"ep": "coaps://[fe80::b1d6]:1122"},
5253                         {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
5254                     ]
5255                 }
5256             ]
5257         },
5258         "schema": {
5259             "$ref": "#/definitions/sbaseline"
5260         }
5261     }
5262 },
5263 },
5264 "post": {
5265     "description": "Update on Baseline OCF Interface\n",
5266     "parameters": [
5267         {
5268             "$ref": "#/parameters/interface-update"
5269         },
5270         {
5271             "name": "body",
5272             "in": "body",
5273             "required": true,
5274             "schema": {
5275                 "$ref": "#/definitions/sbaseline-update"
5276             }
5277         }
5278     ],
5279     "responses": {
5280         "200": {
5281             "description": "",
5282             "schema": {
5283                 "$ref": "#/definitions/sbaseline"
5284             }
5285         }
5286     }
5287 },
5288 },
5289 "/CollectionResURI?if=oic.if.b" : {
5290     "get": {
5291         "description": "Collection Resource Type contains Properties and Links.\nThe oic.if.b OCF
5292 Interface exposes a composite representation of the\nResources pointed to by the Links\n",
5293         "parameters": [

```

```

5294     {
5295         "$ref": "#/parameters/interface-all"
5296     }
5297 ],
5298 "responses": {
5299     "200": {
5300         "description": "All targets returned OK status",
5301         "x-example": [
5302             {
5303                 "href": "/switch",
5304                 "rep": {
5305                     "value": true
5306                 }
5307             },
5308             {
5309                 "href": "/airFlow",
5310                 "rep": {
5311                     "direction": "floor",
5312                     "speed": 3
5313                 }
5314             }
5315         ],
5316         "schema": {
5317             "$ref": "#/definitions/sbatch-retrieve"
5318         }
5319     },
5320     "404": {
5321         "description": "One or more targets did not return an OK status, return a
5322 representation containing returned Properties from the targets that returned OK",
5323         "x-example": [
5324             {
5325                 "href": "/switch",
5326                 "rep": {
5327                     "value": true
5328                 }
5329             }
5330         ],
5331         "schema": {
5332             "$ref": "#/definitions/sbatch-retrieve"
5333         }
5334     }
5335 },
5336 },
5337 "post": {
5338     "description": "Update on Batch OCF Interface\n",
5339     "parameters": [
5340         {
5341             "$ref": "#/parameters/interface-update"
5342         },
5343         {
5344             "name": "body",
5345             "in": "body",
5346             "required": true,
5347             "schema": {
5348                 "$ref": "#/definitions/sbatch-update"
5349             },
5350             "x-example": [
5351                 {
5352                     "href": "/switch",
5353                     "rep": {
5354                         "value": true
5355                     }
5356                 },
5357                 {
5358                     "href": "/airFlow",
5359                     "rep": {
5360                         "direction": "floor",
5361                         "speed": 3
5362                     }
5363                 }
5364             ]
5365         }
5366     ]
5367 }

```

```

5365     }
5366   ],
5367   "responses": {
5368     "200": {
5369       "description" : "All targets returned OK status, return a representation of the current
5370 state of all targets",
5371       "x-example": [
5372         {
5373           "href": "/switch",
5374           "rep": {
5375             "value": true
5376           }
5377         },
5378         {
5379           "href": "/airFlow",
5380           "rep": {
5381             "direction": "demist",
5382             "speed": 5
5383           }
5384         }
5385       ],
5386       "schema": {
5387         "$ref": "#/definitions/sbatch-retrieve"
5388       }
5389     },
5390     "403": {
5391       "description" : "One or more targets did not return OK status; return a retrieve
5392 representation of the current state of all targets in the batch",
5393       "x-example": [
5394         {
5395           "href": "/switch",
5396           "rep": {
5397             "value": true
5398           }
5399         },
5400         {
5401           "href": "/airFlow",
5402           "rep": {
5403             "direction": "floor",
5404             "speed": 3
5405           }
5406         }
5407       ],
5408       "schema": {
5409         "$ref": "#/definitions/sbatch-retrieve"
5410       }
5411     }
5412   }
5413 }
5414 },
5415 },
5416 "parameters": {
5417   "interface-all" : {
5418     "in" : "query",
5419     "name" : "if",
5420     "type" : "string",
5421     "enum" : ["oic.if.ll", "oic.if.b", "oic.if.baseline"]
5422   },
5423   "interface-update" : {
5424     "in" : "query",
5425     "name" : "if",
5426     "type" : "string",
5427     "enum" : ["oic.if.b", "oic.if.baseline"]
5428   }
5429 },
5430 "definitions": {
5431   "sbaseline" : {
5432     "properties": {
5433       "links" : {
5434         "description": "A set of simple or individual Links.",
5435         "items": {

```

```

5436         "$ref": "#/definitions/oic.oic-link"
5437     },
5438     "type": "array"
5439 },
5440 "n": {
5441     "$ref":
5442 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
5443 schema.json#/definitions/n"
5444 },
5445 "id": {
5446     "$ref":
5447 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
5448 schema.json#/definitions/id"
5449 },
5450 "rt": {
5451     "$ref": "#/definitions/oic.core.rt-col"
5452 },
5453 "rts": {
5454     "$ref": "#/definitions/oic.core.rt"
5455 },
5456 "rts-m": {
5457     "$ref": "#/definitions/oic.core.rt"
5458 },
5459 "if": {
5460     "description": "The OCF Interfaces supported by this Resource",
5461     "items": {
5462         "enum": [
5463             "oic.if.ll",
5464             "oic.if.baseline",
5465             "oic.if.b"
5466         ],
5467         "type": "string",
5468         "maxLength": 64
5469     },
5470     "minItems": 2,
5471     "uniqueItems": true,
5472     "readOnly": true,
5473     "type": "array"
5474 },
5475 },
5476 "additionalProperties": true,
5477 "type": "object",
5478 "required": [
5479     "rt",
5480     "if",
5481     "links"
5482 ],
5483 },
5484 "sbaseline-update": {
5485     "additionalProperties": true
5486 },
5487     "oic.core.rt-col": {
5488         "description": "Resource Type of the Resource",
5489         "items": {
5490             "enum": ["oic.wk.col"],
5491             "type": "string",
5492             "maxLength": 64
5493         },
5494         "minItems": 1,
5495         "uniqueItems": true,
5496         "readOnly": true,
5497         "type": "array"
5498     },
5499 "oic.core.rt": {
5500     "description": "Resource Type or set of Resource Types",
5501     "items": {
5502         "type": "string",
5503         "maxLength": 64
5504     },
5505     "minItems": 1,
5506     "uniqueItems": true,

```

```

5507         "readOnly": true,
5508         "type": "array"
5509     },
5510     "sbatch-retrieve" : {
5511         "minItems" : 1,
5512         "items" : {
5513             "additionalProperties": true,
5514             "properties": {
5515                 "href": {
5516                     "$ref":
5517 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5518 schema.json#/definitions/href"
5519                 },
5520                 "rep": {
5521                     "oneOf": [
5522                         {
5523                             "description": "The response payload from a single Resource",
5524                             "type": "object"
5525                         },
5526                         {
5527                             "description": " The response payload from a Collection (batch) Resource",
5528                             "items": {
5529                                 "$ref": "#/definitions/oic.oic-link"
5530                             },
5531                             "type": "array"
5532                         }
5533                     ]
5534                 }
5535             },
5536             "required": [
5537                 "href",
5538                 "rep"
5539             ],
5540             "type": "object"
5541         },
5542         "type" : "array"
5543     },
5544     "sbatch-update" : {
5545         "title" : "Collection Batch Update Format",
5546         "minItems" : 1,
5547         "items" : {
5548             "$ref": "#/definitions/sbatch-update.item"
5549         },
5550         "type" : "array"
5551     },
5552     "sbatch-update.item" : {
5553         "additionalProperties": true,
5554         "description": "Array of Resource representations to apply to the batch Collection, using href
5555 to indicate which Resource(s) in the batch to update. If the href Property is empty, effectively
5556 making the URI reference to the Collection itself, the representation is to be applied to all
5557 Resources in the batch",
5558         "properties": {
5559             "href": {
5560                 "$ref":
5561 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5562 schema.json#/definitions/href"
5563             },
5564             "rep": {
5565                 "oneOf": [
5566                     {
5567                         "description": "The payload for a single Resource",
5568                         "type": "object"
5569                     },
5570                     {
5571                         "description": " The payload for a Collection (batch) Resource",
5572                         "items": {
5573                             "$ref": "#/definitions/oic.oic-link"
5574                         },
5575                         "type": "array"
5576                     }
5577                 ]

```

```

5578     }
5579   },
5580   "required": [
5581     "href",
5582     "rep"
5583   ],
5584   "type": "object"
5585 },
5586 "slinks" : {
5587   "type": "array",
5588   "items": {
5589     "$ref": "#/definitions/oic.oic-link"
5590   }
5591 },
5592 "oic.oic-link": {
5593   "properties": {
5594     "if": {
5595       "description": "The OCF Interfaces supported by the Linked target",
5596       "items": {
5597         "enum": [
5598           "oic.if.baseline",
5599           "oic.if.ll",
5600           "oic.if.b",
5601           "oic.if.rw",
5602           "oic.if.r",
5603           "oic.if.a",
5604           "oic.if.s"
5605         ],
5606         "type": "string",
5607         "maxLength": 64
5608       },
5609       "minItems": 1,
5610       "uniqueItems": true,
5611       "readOnly": true,
5612       "type": "array"
5613     },
5614     "rt": {
5615       "$ref": "#/definitions/oic.core.rt"
5616     },
5617     "anchor": {
5618       "$ref":
5619 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5620 schema.json#/definitions/anchor"
5621     },
5622     "di": {
5623       "$ref":
5624 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5625 schema.json#/definitions/di"
5626     },
5627     "eps": {
5628       "$ref":
5629 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5630 schema.json#/definitions/eps"
5631     },
5632     "href": {
5633       "$ref":
5634 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5635 schema.json#/definitions/href"
5636     },
5637     "ins": {
5638       "$ref":
5639 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5640 schema.json#/definitions/ins"
5641     },
5642     "p": {
5643       "$ref":
5644 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5645 schema.json#/definitions/p"
5646     },
5647     "rel": {
5648       "$ref":

```

```

5649 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5650 schema.json#/definitions/rel_array"
5651 },
5652 "title": {
5653   "$ref":
5654 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5655 schema.json#/definitions/title"
5656 },
5657 "type": {
5658   "$ref":
5659 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
5660 schema.json#/definitions/type"
5661 }
5662 },
5663 "required": [
5664   "href",
5665   "rt",
5666   "if"
5667 ],
5668 "type": "object"
5669 }
5670 }
5671 }
5672

```

### 5673 D.3.5 Property definition

5674 Table D-4 defines the Properties that are part of the "oic.wk.col" Resource Type.

5675 **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	

5676 **D.3.6 CRUDN behaviour**

5677 Table D-5 defines the CRUDN operations that are supported on the "oic.wk.col" Resource Type.

5678 **Table D-5 – The CRUDN operations of the Resource with type "rt" = "oic.wk.col".**

Create	Read	Update	Delete	Notify
	get	post		observe

5679 **D.4 Device Configuration**

5680 **D.4.1 Introduction**

5681 Resource that allows for Device specific information to be configured.

5682

5683 **D.4.2 Example URI**

5684 /exampleDeviceConfigurationResURI

5685 **D.4.3 Resource type**

5686 The Resource Type is defined as: "oic.wk.con".

5687 **D.4.4 OpenAPI 2.0 definition**

```
5688 {
5689   "swagger": "2.0",
5690   "info": {
5691     "title": "Device Configuration",
5692     "version": "2019-02-28",
5693     "license": {
5694       "name": "OCF Data Model License",
5695       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
5696       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
5697     },
5698     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
5699   },
5700   "schemes": [
5701     "http"
5702   ],
5703   "consumes": [
```

```

5704     "application/json"
5705 ],
5706 "produces": [
5707     "application/json"
5708 ],
5709 "paths": {
5710     "/exampleDeviceConfigurationResURI" : {
5711         "get": {
5712             "description": "Resource that allows for Device specific information to be configured.\n",
5713             "parameters": [
5714                 {
5715                     "$ref": "#/parameters/interface-all"
5716                 }
5717             ],
5718             "responses": {
5719                 "200": {
5720                     "description": "",
5721                     "x-example": {
5722                         "n": "My Friendly Device Name",
5723                         "rt": ["oic.wk.con"],
5724                         "loc": [32.777,-96.797],
5725                         "locn": "My Location Name",
5726                         "c": "USD",
5727                         "r": "MyRegion",
5728                         "dl": "en"
5729                     },
5730                     "schema": {
5731                         "$ref": "#/definitions/Configuration"
5732                     }
5733                 }
5734             }
5735         },
5736         "post": {
5737             "description": "Update the information about the Device\n",
5738             "parameters": [
5739                 {
5740                     "$ref": "#/parameters/interface-rw"
5741                 },
5742                 {
5743                     "name": "body",
5744                     "in": "body",
5745                     "required": true,
5746                     "schema": {
5747                         "$ref": "#/definitions/Update"
5748                     },
5749                     "x-example": {
5750                         "n": "Nuevo Nombre Amistoso",
5751                         "r": "MyNewRegion",
5752                         "ln": [ { "language": "es", "value": "Nuevo Nombre Amistoso" } ],
5753                         "dl": "es"
5754                     }
5755                 }
5756             ],
5757             "responses": {
5758                 "200": {
5759                     "description": "",
5760                     "x-example": {
5761                         "n": "Nuevo Nombre Amistoso",
5762                         "r": "MyNewRegion",
5763                         "ln": [ { "language": "es", "value": "Nuevo Nombre Amistoso" } ],
5764                         "dl": "es"
5765                     },
5766                     "schema": {
5767                         "$ref": "#/definitions/Update"
5768                     }
5769                 }
5770             }
5771         }
5772     }
5773 },
5774 "parameters": {

```

```

5775     "interface-rw" : {
5776         "in" : "query",
5777         "name" : "if",
5778         "type" : "string",
5779         "enum" : ["oic.if.rw"]
5780     },
5781     "interface-all" : {
5782         "in" : "query",
5783         "name" : "if",
5784         "type" : "string",
5785         "enum" : ["oic.if.rw", "oic.if.baseline"]
5786     }
5787 },
5788 "definitions": {
5789     "Configuration": {
5790         "properties": {
5791             "rt": {
5792                 "description": "Resource Type of the Resource",
5793                 "items": {
5794                     "enum": ["oic.wk.con"],
5795                     "type": "string",
5796                     "maxLength": 64
5797                 },
5798                 "minItems": 1,
5799                 "uniqueItems": true,
5800                 "readOnly": true,
5801                 "type": "array"
5802             },
5803             "loc": {
5804                 "description": "Location information (lat, long)",
5805                 "items": {
5806                     "type": "number"
5807                 },
5808                 "maxItems": 2,
5809                 "minItems": 2,
5810                 "type": "array"
5811             },
5812             "c": {
5813                 "description": "Currency",
5814                 "maxLength": 64,
5815                 "type": "string"
5816             },
5817             "ln": {
5818                 "description": "Localized names",
5819                 "items": {
5820                     "properties": {
5821                         "language": {
5822                             "allOf": [
5823                                 {
5824                                     "description": "Format pattern according to IETF RFC 5646 (language tag).",
5825                                     "pattern": "^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$",
5826                                     "type": "string"
5827                                 },
5828                                 {
5829                                     "description": "An RFC 5646 language tag."
5830                                 }
5831                             ]
5832                         },
5833                         "value": {
5834                             "description": "The Device name in the indicated language.",
5835                             "maxLength": 64,
5836                             "type": "string"
5837                         }
5838                     },
5839                     "type": "object"
5840                 },
5841                 "minItems": 1,
5842                 "type": "array"
5843             },
5844             "locn": {
5845                 "description": "Human Friendly Name for location",

```

```

5846         "maxLength": 64,
5847         "type": "string"
5848     },
5849     "dl": {
5850         "allOf": [
5851             {
5852                 "description": "Format pattern according to IETF RFC 5646 (language tag).",
5853                 "pattern": "^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$",
5854                 "type": "string"
5855             },
5856             {
5857                 "description": "Default Language as an RFC 5646 language tag."
5858             }
5859         ]
5860     },
5861     "n": {
5862         "$ref" :
5863         "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
5864         schema.json#/definitions/n"
5865     },
5866     "id": {
5867         "$ref" :
5868         "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
5869         schema.json#/definitions/id"
5870     },
5871     "r": {
5872         "description": "Region",
5873         "maxLength": 64,
5874         "type": "string"
5875     },
5876     "if" : {
5877         "description": "The OCF Interfaces supported by this Resource",
5878         "items": {
5879             "enum": [
5880                 "oic.if.baseline",
5881                 "oic.if.rw"
5882             ],
5883             "type": "string",
5884             "maxLength": 64
5885         },
5886         "minItems": 1,
5887         "uniqueItems": true,
5888         "readOnly": true,
5889         "type": "array"
5890     }
5891 },
5892 "type" : "object",
5893 "required": ["n"]
5894 },
5895 "Update" : {
5896     "properties": {
5897         "loc": {
5898             "description": "Location information (lat, long)",
5899             "items": {
5900                 "type": "number"
5901             },
5902             "maxItems": 2,
5903             "minItems": 2,
5904             "type": "array"
5905         },
5906         "c": {
5907             "description": "Currency",
5908             "maxLength": 64,
5909             "type": "string"
5910         },
5911         "ln": {
5912             "description": "Localized names",
5913             "items": {
5914                 "properties": {
5915                     "language": {
5916

```

```

5917         {
5918             "description": "Format pattern according to IETF RFC 5646 (language tag).",
5919             "pattern": "^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$",
5920             "type": "string"
5921         },
5922         {
5923             "description": "An RFC 5646 language tag."
5924         }
5925     ],
5926 },
5927 "value": {
5928     "description": "The Device name in the indicated language.",
5929     "maxLength": 64,
5930     "type": "string"
5931 },
5932 },
5933 "type": "object"
5934 },
5935 "minItems": 1,
5936 "type": "array"
5937 },
5938 "locn": {
5939     "description": "Human Friendly Name for location",
5940     "maxLength": 64,
5941     "type": "string"
5942 },
5943 "dl": {
5944     "allOf": [
5945         {
5946             "description": "Format pattern according to IETF RFC 5646 (language tag).",
5947             "pattern": "^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$",
5948             "type": "string"
5949         },
5950         {
5951             "description": "Default Language as an RFC 5646 language tag."
5952         }
5953     ]
5954 },
5955 "n": {
5956     "description": "The human friendly name to be set on the Resource, this is also reflected
5957 in the same Property in oic.wk.d",
5958     "maxLength": 64,
5959     "type": "string"
5960 },
5961 "r": {
5962     "description": "Region",
5963     "maxLength": 64,
5964     "type": "string"
5965 }
5966 },
5967 "anyOf": [
5968     {
5969         "required": ["loc"]
5970     },
5971     {
5972         "required": ["locn"]
5973     },
5974     {
5975         "required": ["c"]
5976     },
5977     {
5978         "required": ["r"]
5979     },
5980     {
5981         "required": ["ln"]
5982     },
5983     {
5984         "required": ["dl"]
5985     },
5986     {
5987         "required": ["n"]

```

```

5988     }
5989     ],
5990     "type" : "object"
5991   }
5992 }
5993 }
5994

```

5995 **D.4.5 Property definition**

5996 Table D-6 defines the Properties that are part of the "oic.wk.con" Resource Type.

5997 **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.

5998 **D.4.6 CRUDN behaviour**

5999 Table D-7 defines the CRUDN operations that are supported on the "oic.wk.con" Resource Type.

6000 **Table D-7 – The CRUDN operations of the Resource with type "rt" = "oic.wk.con".**

Create	Read	Update	Delete	Notify
	get	post		observe

6001 **D.5 Platform Configuration**

6002 **D.5.1 Introduction**

6003 Resource that allows for Platform specific information to be configured.  
6004

6005 **D.5.2 Example URI**

6006 /examplePlatformConfigurationResURI

6007 **D.5.3 Resource type**

6008 The Resource Type is defined as: "oic.wk.con.p".

6009 **D.5.4 OpenAPI 2.0 definition**

```

6010 {
6011   "swagger": "2.0",
6012   "info": {
6013     "title": "Platform Configuration",
6014     "version": "2019-03-04",
6015     "license": {
6016       "name": "OCF Data Model License",
6017       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
6018       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
6019     },
6020     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
6021   },
6022   "schemes": [
6023     "http"
6024   ],
6025   "consumes": [
6026     "application/json"
6027   ],
6028   "produces": [
6029     "application/json"
6030   ],
6031   "paths": {
6032     "/examplePlatformConfigurationResURI": {
6033       "get": {
6034         "description": "Resource that allows for Platform specific information to be configured.\n",
6035         "parameters": [
6036           {
6037             "$ref": "#/parameters/interface-all"
6038           }
6039         ],
6040         "responses": {
6041           "200": {
6042             "description": "",
6043             "x-example": {
6044               "rt": ["oic.wk.con.p"],
6045               "mnpn": [ { "language": "en", "value": "My Friendly Device Name" } ]
6046             },
6047             "schema": { "$ref": "#/definitions/Conf_Platform" }
6048           }
6049         }
6050       },
6051       "post": {
6052         "description": "Update the information about the Platform\n",
6053         "parameters": [
6054           {
6055             "$ref": "#/parameters/interface-rw"

```

```

6056     },
6057     {
6058         "name": "body",
6059         "in": "body",
6060         "required": true,
6061         "schema": { "$ref": "#/definitions/Update_Platform" },
6062         "x-example": {
6063             "n": "Nuevo nombre",
6064             "mnpn": [ { "language": "es", "value": "Nuevo nombre de Plataforma Amigable" } ]
6065         }
6066     }
6067 ],
6068 "responses": {
6069     "200": {
6070         "description": "",
6071         "x-example": {
6072             "n": "Nuevo nombre",
6073             "mnpn": [ { "language": "es", "value": "Nuevo nombre de Plataforma Amigable" } ]
6074         },
6075         "schema": { "$ref": "#/definitions/Update_Platform" }
6076     }
6077 }
6078 }
6079 },
6080 },
6081 "parameters": {
6082     "interface-rw": {
6083         "in": "query",
6084         "name": "if",
6085         "type": "string",
6086         "enum": [ "oic.if.rw" ]
6087     },
6088     "interface-all": {
6089         "in": "query",
6090         "name": "if",
6091         "type": "string",
6092         "enum": [ "oic.if.rw", "oic.if.baseline" ]
6093     }
6094 },
6095 "definitions": {
6096     "Conf_Platform": {
6097         "properties": {
6098             "rt": {
6099                 "description": "Resource Type of the Resource",
6100                 "items": {
6101                     "enum": [ "oic.wk.con.p" ],
6102                     "type": "string",
6103                     "maxLength": 64
6104                 },
6105                 "minItems": 1,
6106                 "uniqueItems": true,
6107                 "readOnly": true,
6108                 "type": "array"
6109             },
6110             "n": {
6111                 "$ref":
6112 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6113 schema.json#/definitions/n"
6114             },
6115             "mnpn": {
6116                 "description": "Platform names",
6117                 "items": {
6118                     "properties": {
6119                         "language": {
6120                             "allOf": [
6121                                 {
6122                                     "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
6123 schema.json#/definitions/language-tag"
6124                                 },
6125                                 {
6126                                     "description": "An RFC 5646 language tag."

```



```

6127         }
6128     ]
6129 },
6130     "value": {
6131         "description": "The Platform description in the indicated language.",
6132         "maxLength": 64,
6133         "type": "string"
6134     }
6135 },
6136     "type": "object"
6137 },
6138     "minItems": 1,
6139     "type": "array"
6140 },
6141     "id": {
6142         "$ref":
6143 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6144 schema.json#/definitions/id"
6145     },
6146     "if": {
6147         "description": "The OCF Interfaces supported by this Resource",
6148         "items": {
6149             "enum": [
6150                 "oic.if.rw",
6151                 "oic.if.baseline"
6152             ],
6153             "type": "string",
6154             "maxLength": 64
6155         },
6156         "minItems": 1,
6157         "readOnly": true,
6158         "uniqueItems": true,
6159         "type": "array"
6160     }
6161 },
6162     "type" : "object"
6163 },
6164     "Update_Platform": {
6165         "properties": {
6166             "n": {
6167                 "description": "The human friendly name to be set on the Resource, this is also reflected
6168 in the same Property in oic.wk.p",
6169                 "maxLength": 64,
6170                 "type": "string"
6171             },
6172             "mnpn" : {
6173                 "description": "Platform names",
6174                 "items": {
6175                     "properties": {
6176                         "language": {
6177                             "allOf": [
6178                                 {
6179                                     "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
6180 schema.json#/definitions/language-tag"
6181                                 },
6182                                 {
6183                                     "description": "An RFC 5646 language tag."
6184                                 }
6185                             ]
6186                         },
6187                         "value": {
6188                             "description": "The Platform description in the indicated language.",
6189                             "maxLength": 64,
6190                             "type": "string"
6191                         }
6192                     },
6193                     "type": "object"
6194                 },
6195                 "minItems": 1,
6196                 "type": "array"
6197             }

```

```

6198     },
6199     "type": "object",
6200     "anyOf": [
6201       {
6202         "required": ["mnpn"]
6203       },
6204       {
6205         "required": ["n"]
6206       }
6207     ]
6208   }
6209 }
6210 }
6211

```

### 6212 D.5.5 Property definition

6213 Table D-8 defines the Properties that are part of the "oic.wk.con.p" Resource Type.

6214 **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.

### 6215 D.5.6 CRUDN behaviour

6216 Table D-9 defines the CRUDN operations that are supported on the "oic.wk.con.p" Resource Type.

6217 **Table D-9 – The CRUDN operations of the Resource with type "rt" = "oic.wk.con.p".**

Create	Read	Update	Delete	Notify
	get	post		observe

## 6218 D.6 Device

### 6219 D.6.1 Introduction

6220 Known Resource that is hosted by every Server.  
6221 Allows for logical Device specific information to be discovered.  
6222

6223 **D.6.2 Well-known URI**

6224 /oic/d

6225 **D.6.3 Resource type**

6226 The Resource Type is defined as: "oic.wk.d".

6227 **D.6.4 OpenAPI 2.0 definition**

```
6228 {
6229   "swagger": "2.0",
6230   "info": {
6231     "title": "Device",
6232     "version": "2019-03-13",
6233     "license": {
6234       "name": "OCF Data Model License",
6235       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
6236       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
6237     },
6238     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
6239   },
6240   "schemes": [
6241     "http"
6242   ],
6243   "consumes": [
6244     "application/json"
6245   ],
6246   "produces": [
6247     "application/json"
6248   ],
6249   "paths": {
6250     "/oic/d" : {
6251       "get": {
6252         "description": "Known Resource that is hosted by every Server.\nAllows for logical Device
6253 specific information to be discovered.\n",
6254         "parameters": [
6255           {
6256             "$ref": "#/parameters/interface"
6257           }
6258         ],
6259         "responses": {
6260           "200": {
6261             "description": "",
6262             "x-example":
6263             {
6264               "n": "Device 1",
6265               "rt": ["oic.wk.d"],
6266               "di": "54919CA5-4101-4AE4-595B-353C51AA983C",
6267               "icv": "ocf.2.0.2",
6268               "dmv": "ocf.res.1.0.0, ocf.sh.1.0.0",
6269               "piid": "6F0AAC04-2BB0-468D-B57C-16570A26AE48"
6270             },
6271             "schema": {
6272               "$ref": "#/definitions/Device"
6273             }
6274           }
6275         }
6276       }
6277     }
6278   },
6279   "parameters": {
6280     "interface" : {
6281       "in": "query",
6282       "name": "if",
6283       "type": "string",
6284       "enum": ["oic.if.r", "oic.if.baseline"]
6285     }
6286   },
6287   "definitions": {
6288     "Device": {
```

```

6289     "properties": {
6290       "rt": {
6291         "description": "Resource Type of the Resource",
6292         "items": {
6293           "type": "string",
6294           "maxLength": 64
6295         },
6296         "minItems": 1,
6297         "readOnly": true,
6298         "uniqueItems": true,
6299         "type": "array"
6300       },
6301       "ld": {
6302         "description": "Localized Descriptions.",
6303         "items": {
6304           "properties": {
6305             "language": {
6306               "allOf": [
6307                 {
6308                   "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
6309 schema.json#/definitions/language-tag"
6310                 },
6311                 {
6312                   "description": "An RFC 5646 language tag.",
6313                   "readOnly": true
6314                 }
6315               ]
6316             },
6317             "value": {
6318               "description": "Device description in the indicated language.",
6319               "maxLength": 64,
6320               "readOnly": true,
6321               "type": "string"
6322             }
6323           },
6324           "type": "object"
6325         },
6326         "minItems": 1,
6327         "readOnly": true,
6328         "type": "array"
6329       },
6330       "pid": {
6331         "allOf": [
6332           {
6333             "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
6334 schema.json#/definitions/uuid"
6335           },
6336           {
6337             "description": "Protocol independent unique identifier for the Device that is
6338 immutable.",
6339             "readOnly": true
6340           }
6341         ]
6342       },
6343       "di": {
6344         "allOf": [
6345           {
6346             "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
6347 schema.json#/definitions/uuid"
6348           },
6349           {
6350             "description": "Unique identifier for the Device",
6351             "readOnly": true
6352           }
6353         ]
6354       },
6355       "dmno": {
6356         "description": "Model number as designated by manufacturer.",
6357         "maxLength": 64,
6358         "readOnly": true,
6359         "type": "string"

```

```

6360     },
6361     "sv": {
6362         "description": "Software version.",
6363         "maxLength": 64,
6364         "readOnly": true,
6365         "type": "string"
6366     },
6367     "dmn": {
6368         "description": "Manufacturer Name.",
6369         "items": {
6370             "properties": {
6371                 "language": {
6372                     "allOf": [
6373                         {
6374                             "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
6375 schema.json#/definitions/language-tag"
6376                         },
6377                         {
6378                             "description": "An RFC 5646 language tag.",
6379                             "readOnly": true
6380                         }
6381                     ]
6382                 },
6383                 "value": {
6384                     "description": "Manufacturer name in the indicated language.",
6385                     "maxLength": 64,
6386                     "readOnly": true,
6387                     "type": "string"
6388                 }
6389             },
6390             "type": "object"
6391         },
6392         "minItems": 1,
6393         "readOnly": true,
6394         "type": "array"
6395     },
6396     "icv": {
6397         "description": "The version of the Device",
6398         "maxLength": 64,
6399         "readOnly": true,
6400         "type": "string"
6401     },
6402     "dmv": {
6403         "description": "Specification versions of the Resource and Device Specifications to which
6404 this device data model is implemented",
6405         "maxLength": 256,
6406         "readOnly": true,
6407         "type": "string"
6408     },
6409     "n": {
6410         "$ref": "
6411 https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6412 schema.json#/definitions/n"
6413     },
6414     "id": {
6415         "$ref": "
6416 https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6417 schema.json#/definitions/id"
6418     },
6419     "if": {
6420         "description": "The OCF Interfaces supported by this Resource",
6421         "items": {
6422             "enum": [
6423                 "oic.if.r",
6424                 "oic.if.baseline"
6425             ],
6426             "type": "string",
6427             "maxLength": 64
6428         },
6429         "minItems": 2,
6430         "uniqueItems": true,

```

```

6431         "readOnly": true,
6432         "type": "array"
6433     },
6434     "econame" : {
6435         "description": "Ecosystem Name of the Bridged Device which is exposed by this VOD.",
6436         "type": "string",
6437         "enum": ["BLE", "oneM2M", "UPlus", "Zigbee", "Z-Wave"],
6438         "readOnly": true
6439     },
6440     "ecoversion" : {
6441         "description": "Version of ecosystem that a Bridged Device belongs to. Typical version
6442 string format is like n.n (e.g. 5.0).",
6443         "type": "string",
6444         "maxLength": 64,
6445         "readOnly": true
6446     }
6447 },
6448 "type": "object",
6449 "required": ["n", "di", "icv", "dmv", "piid"]
6450 }
6451 }
6452 }
6453

```

#### 6454 D.6.5 Property definition

6455 Table D-10 defines the Properties that are part of the "oic.wk.d" Resource Type.

6456 **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.

econame	string	No	Read Only	Ecosystem Name of the Bridged Device which is exposed by this VOD.
ecoversion	string	No	Read Only	Version of ecosystem that a Bridged Device belongs to. Typical version string format is like n.n (e.g. 5.0).

6457 **D.6.6 CRUDN behaviour**

6458 Table D-11 defines the CRUDN operations that are supported on the "oic.wk.d" Resource Type.

6459 **Table D-11 – The CRUDN operations of the Resource with type "rt" = "oic.wk.d".**

Create	Read	Update	Delete	Notify
	get			observe

6460 **D.7 Icon**

6461 **D.7.1 Introduction**

6462 This Resource describes the attributes associated with an Icon.  
6463

6464 **D.7.2 Example URI**

6465 /IconResURI

6466 **D.7.3 Resource type**

6467 The Resource Type is defined as: "oic.r.icon".

6468 **D.7.4 OpenAPI 2.0 definition**

```
6469 {
6470   "swagger": "2.0",
6471   "info": {
6472     "title": "Icon",
6473     "version": "2019-02-26",
6474     "license": {
6475       "name": "OCF Data Model License",
6476       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
6477       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
6478     },
6479     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
6480   },
6481   "schemes": [
6482     "http"
6483   ],
6484   "consumes": [
6485     "application/json"
6486   ],
6487   "produces": [
6488     "application/json"
6489   ],
6490   "paths": {
6491     "/IconResURI" : {
6492       "get": {
6493         "description": "This Resource describes the attributes associated with an Icon.\n",
6494         "parameters": [
6495           {
6496             "$ref": "#/parameters/interface"
6497           }
6498         ]
6499       }
7000     }
7001   }
7002 }
```

```

6498     ],
6499     "responses": {
6500         "200": {
6501             "description": "",
6502             "x-example": {
6503                 "rt": ["oic.r.icon"],
6504                 "mimetype": "image/png",
6505                 "width": 256,
6506                 "height": 256,
6507                 "media": "http://findbetter.ru/public/uploads/1481662800/2043.png"
6508             },
6509             "schema": {
6510                 "$ref": "#/definitions/Icon"
6511             }
6512         }
6513     }
6514 }
6515 },
6516 },
6517 "parameters": {
6518     "interface": {
6519         "in": "query",
6520         "name": "if",
6521         "type": "string",
6522         "enum": ["oic.if.r", "oic.if.baseline"]
6523     }
6524 },
6525 "definitions": {
6526     "Icon": {
6527         "properties": {
6528             "mimetype": {
6529                 "description": "The Media Type of the icon",
6530                 "maxLength": 64,
6531                 "readOnly": true,
6532                 "type": "string"
6533             },
6534             "rt": {
6535                 "description": "Resource Type of the Resource",
6536                 "items": {
6537                     "enum": ["oic.r.icon"],
6538                     "type": "string",
6539                     "maxLength": 64
6540                 },
6541                 "minItems": 1,
6542                 "uniqueItems": true,
6543                 "readOnly": true,
6544                 "type": "array"
6545             },
6546             "media": {
6547                 "description": "Specifies the URI to the icon",
6548                 "format": "uri",
6549                 "maxLength": 256,
6550                 "readOnly": true,
6551                 "type": "string"
6552             },
6553             "n": {
6554                 "$ref":
6555 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6556 schema.json#/definitions/n"
6557             },
6558             "id": {
6559                 "$ref":
6560 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6561 schema.json#/definitions/id"
6562             },
6563             "width": {
6564                 "description": "The width in pixels",
6565                 "minimum": 1,
6566                 "readOnly": true,
6567                 "type": "integer"
6568             }

```



```

6569     "height": {
6570         "description": "The height in pixels",
6571         "minimum": 1,
6572         "readOnly": true,
6573         "type": "integer"
6574     },
6575     "if": {
6576         "description": "The OCF Interfaces supported by this Resource",
6577         "items": {
6578             "enum": [
6579                 "oic.if.r",
6580                 "oic.if.baseline"
6581             ],
6582             "maxLength": 64,
6583             "type": "string"
6584         },
6585         "minItems": 2,
6586         "uniqueItems": true,
6587         "readOnly": true,
6588         "type": "array"
6589     }
6590 },
6591 "type": "object",
6592 "required": ["mimetype", "width", "height", "media"]
6593 }
6594 }
6595 }
6596

```

### 6597 D.7.5 Property definition

6598 Table D-12 defines the Properties that are part of the "oic.r.icon" Resource Type.

6599 **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.

### 6600 D.7.6 CRUDN behaviour

6601 Table D-13 defines the CRUDN operations that are supported on the "oic.r.icon" Resource Type.

6602 **Table D-13 – The CRUDN operations of the Resource with type "rt" = "oic.r.icon".**

Create	Read	Update	Delete	Notify
	get			observe

## 6603 **D.8 Introspection Resource**

### 6604 **D.8.1 Introduction**

6605 This Resource provides the means to get the Introspection Device Data (IDD) specifying all the  
6606 OCF Endpoints of the Device.  
6607 The url hosted by this Resource is either a local or an external url.  
6608

### 6609 **D.8.2 Well-known URI**

6610 /IntrospectionResURI

### 6611 **D.8.3 Resource type**

6612 The Resource Type is defined as: "oic.wk.introspection".

### 6613 **D.8.4 OpenAPI 2.0 definition**

```
6614 {
6615   "swagger": "2.0",
6616   "info": {
6617     "title": "Introspection Resource",
6618     "version": "2019-03-04",
6619     "license": {
6620       "name": "OCF Data Model License",
6621       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
6622       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
6623     },
6624     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
6625   },
6626   "schemes": [
6627     "http"
6628   ],
6629   "consumes": [
6630     "application/json"
6631   ],
6632   "produces": [
6633     "application/json"
6634   ],
6635   "paths": {
6636     "/IntrospectionResURI": {
6637       "get": {
6638         "description": "This Resource provides the means to get the Introspection Device Data (IDD)
6639 specifying all the OCF Endpoints of the Device.\n\nThe url hosted by this Resource is either a local
6640 or an external url.\n",
6641         "parameters": [
6642           {
6643             "$ref": "#/parameters/interface"
6644           }
6645         ],
6646         "responses": {
6647           "200": {
6648             "description": "",
6649             "x-example": {
6650               "rt": ["oic.wk.introspection"],
6651               "urlInfo": [
6652                 {
6653                   "content-type": "application/cbor",
6654                   "protocol": "coap",
6655                   "url": "coap://[fe80::1]:1234/IntrospectionExampleURI"
6656                 }
6657               ]
6658             },
6659             "schema": {
6660               "$ref": "#/definitions/oic.wk.introspectionInfo"
6661             }
6662           }
6663         }
6664       }
6665     }
6666   }
6667 }
```

```

6664     }
6665     }
6666   },
6667   "parameters": {
6668     "interface": {
6669       "in": "query",
6670       "name": "if",
6671       "type": "string",
6672       "enum": ["oic.if.r", "oic.if.baseline"]
6673     }
6674   },
6675   "definitions": {
6676     "oic.wk.introspectionInfo": {
6677       "properties": {
6678         "rt": {
6679           "description": "Resource Type of the Resource",
6680           "items": {
6681             "enum": ["oic.wk.introspection"],
6682             "type": "string",
6683             "maxLength": 64
6684           },
6685           "minItems": 1,
6686           "readOnly": true,
6687           "uniqueItems": true,
6688           "type": "array"
6689         },
6690         "n": {
6691           "$ref":
6692 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6693 schema.json#/definitions/n"
6694         },
6695         "urlInfo": {
6696           "description": "Information on the location of the Introspection Device Data (IDD).",
6697           "items": {
6698             "properties": {
6699               "content-type": {
6700                 "default": "application/cbor",
6701                 "description": "content-type of the Introspection Device Data",
6702                 "enum": [
6703                   "application/json",
6704                   "application/cbor"
6705                 ],
6706                 "type": "string"
6707             },
6708             "protocol": {
6709               "description": "Identifier for the protocol to be used to obtain the Introspection
6710 Device Data",
6711               "enum": [
6712                 "coap",
6713                 "coaps",
6714                 "http",
6715                 "https",
6716                 "coap+tcp",
6717                 "coaps+tcp"
6718               ],
6719               "type": "string"
6720             },
6721             "url": {
6722               "description": "The URL of the Introspection Device Data.",
6723               "format": "uri",
6724               "type": "string"
6725             },
6726             "version": {
6727               "default": 1,
6728               "description": "The version of the Introspection Device Data that can be
6729 downloaded",
6730               "enum": [
6731                 1
6732               ],
6733               "type": "integer"
6734             }

```

```

6735     },
6736     "required": [
6737         "url",
6738         "protocol"
6739     ],
6740     "type": "object"
6741 },
6742 "minItems": 1,
6743 "readOnly": true,
6744 "type": "array"
6745 },
6746 "id": {
6747     "$ref":
6748 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6749 schema.json#/definitions/id"
6750 },
6751 "if": {
6752     "description": "The OCF Interfaces supported by this Resource",
6753     "items": {
6754         "enum": [
6755             "oic.if.r",
6756             "oic.if.baseline"
6757         ],
6758         "type": "string",
6759         "maxLength": 64
6760     },
6761     "minItems": 2,
6762     "readOnly": true,
6763     "uniqueItems": true,
6764     "type": "array"
6765 }
6766 },
6767 "type" : "object",
6768 "required": ["urlInfo"]
6769 }
6770 }
6771 }
6772

```

### 6773 D.8.5 Property definition

6774 Table D-14 defines the Properties that are part of the "oic.wk.introspection" Resource Type.

6775 **Table D-14 – The Property definitions of the Resource with type "rt" =**  
6776 **"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.

### 6777 D.8.6 CRUDN behaviour

6778 Table D-15 defines the CRUDN operations that are supported on the "oic.wk.introspection"  
6779 Resource Type.

6780  
6781

**Table D-15 – The CRUDN operations of the Resource with type "rt" = "oic.wk.introspection".**

Create	Read	Update	Delete	Notify
	get			observe

6782 **D.9 Maintenance**

6783 **D.9.1 Introduction**

6784 The Resource through which a Device is maintained and can be used for diagnostic purposes.  
6785 fr (Factory Reset) is a boolean.  
6786 The value 0 means No action (Default), the value 1 means Start Factory Reset  
6787 After factory reset, this value shall be changed back to the default value  
6788 rb (Reboot) is a boolean.  
6789 The value 0 means No action (Default), the value 1 means Start Reboot  
6790 After Reboot, this value shall be changed back to the default value  
6791

6792 **D.9.2 Well-known URI**

6793 /oic/mnt

6794 **D.9.3 Resource type**

6795 The Resource Type is defined as: "oic.wk.mnt".

6796 **D.9.4 OpenAPI 2.0 definition**

```
6797 {  
6798   "swagger": "2.0",  
6799   "info": {  
6800     "title": "Maintenance",  
6801     "version": "2019-03-04",  
6802     "license": {  
6803       "name": "OCF Data Model License",  
6804       "url":  
6805         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI  
6806         CENSE.md",  
6807       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."  
6808     },  
6809     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"  
6810   },  
6811   "schemes": ["http"],  
6812   "consumes": ["application/json"],  
6813   "produces": ["application/json"],  
6814   "paths": {  
6815     "/oic/mnt" : {  
6816       "get": {  
6817         "description": "The Resource through which a Device is maintained and can be used for  
6818         diagnostic purposes.\nfr (Factory Reset) is a boolean.\n The value 0 means No action (Default), the  
6819         value 1 means Start Factory Reset\nAfter factory reset, this value shall be changed back to the  
6820         default value\nrb (Reboot) is a boolean.\n The value 0 means No action (Default), the value 1 means  
6821         Start Reboot\nAfter Reboot, this value shall be changed back to the default value\n",  
6822         "parameters": [  
6823           {"$ref": "#/parameters/interface-all"}  
6824         ],  
6825         "responses": {  
6826           "200": {  
6827             "description": "",  
6828             "x-example": {  
6829               "rt": ["oic.wk.mnt"],  
6830               "fr": false,  
6831               "rb": false,  
6832               "err" : 503  
6833             },  
6834           }  
6835         }  
6836       }  
6837     }  
6838   }  
6839 }
```

```

6834         "schema": { "$ref": "#/definitions/mnt" }
6835     }
6836 },
6837 },
6838 "post": {
6839     "description": "Set the maintenance action(s)\n",
6840     "parameters": [
6841         { "$ref": "#/parameters/interface-rw" },
6842         {
6843             "name": "body",
6844             "in": "body",
6845             "required": true,
6846             "schema": { "$ref": "#/definitions/mnt-update" },
6847             "x-example": {
6848                 "fr": false,
6849                 "rb": false
6850             }
6851         }
6852     ],
6853     "responses": {
6854         "200": {
6855             "description": "",
6856             "x-example": {
6857                 "fr": false,
6858                 "rb": false
6859             },
6860             "schema": { "$ref": "#/definitions/mnt" }
6861         }
6862     }
6863 },
6864 },
6865 },
6866 "parameters": {
6867     "interface-all" : {
6868         "in" : "query",
6869         "name" : "if",
6870         "type" : "string",
6871         "enum" : ["oic.if.rw", "oic.if.baseline"]
6872     },
6873     "interface-rw" : {
6874         "in" : "query",
6875         "name" : "if",
6876         "type" : "string",
6877         "enum" : ["oic.if.rw"]
6878     }
6879 },
6880 "definitions": {
6881     "mnt" : {
6882         "properties": {
6883             "rt" : {
6884                 "description": "Resource Type of the Resource",
6885                 "items": {
6886                     "enum": ["oic.wk.mnt"],
6887                     "type": "string",
6888                     "maxLength": 64
6889                 },
6890                 "minItems": 1,
6891                 "uniqueItems": true,
6892                 "readOnly": true,
6893                 "type": "array"
6894             },
6895             "fr" : {
6896                 "description": "Factory Reset",
6897                 "type": "boolean"
6898             },
6899             "err" : {
6900                 "description": "Last HTTP occurred error",
6901                 "maximum": 599,
6902                 "minimum": 399,
6903                 "readOnly": true,
6904                 "type": "integer"

```

```

6905     },
6906     "n": {
6907       "$ref":
6908       "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6909       schema.json#/definitions/n"
6910     },
6911     "rb": {
6912       "description": "Reboot Action",
6913       "type": "boolean"
6914     },
6915     "id": {
6916       "$ref":
6917       "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6918       schema.json#/definitions/id"
6919     },
6920     "if": {
6921       "description": "The OCF Interfaces supported by this Resource",
6922       "items": {
6923         "enum": [
6924           "oic.if.rw",
6925           "oic.if.baseline"
6926         ],
6927         "type": "string",
6928         "maxLength": 64
6929       },
6930       "minItems": 1,
6931       "readOnly": true,
6932       "uniqueItems": true,
6933       "type": "array"
6934     }
6935   },
6936   "anyOf" : [
6937     {
6938       "required": [ "fr" ]
6939     },
6940     {
6941       "required": [ "rb" ]
6942     },
6943     {
6944       "required": [ "err" ]
6945     }
6946   ],
6947   "type" : "object"
6948 },
6949 "mnt-update" : {
6950   "properties": {
6951     "fr" : {
6952       "description": "Factory Reset",
6953       "type": "boolean"
6954     },
6955     "n": {
6956       "$ref":
6957       "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6958       schema.json#/definitions/n"
6959     },
6960     "rb": {
6961       "description": "Reboot Action",
6962       "type": "boolean"
6963     }
6964   },
6965   "anyOf" : [
6966     {
6967       "required": [
6968         "fr"
6969       ]
6970     },
6971     {
6972       "required": [
6973         "rb"
6974       ]
6975     }

```

```

6976     ],
6977     "type" : "object"
6978   }
6979 }
6980 }
6981

```

## 6982 D.9.5 Property definition

6983 Table D-16 defines the Properties that are part of the "oic.wk.mnt" Resource Type.

6984 **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.

## 6985 D.9.6 CRUDN behaviour

6986 Table D-17 defines the CRUDN operations that are supported on the "oic.wk.mnt" Resource Type.

6987 **Table D-17 – The CRUDN operations of the Resource with type "rt" = "oic.wk.mnt".**

Create	Read	Update	Delete	Notify
	get	post		observe

## 6988 D.10 Network Monitoring

### 6989 D.10.1 Introduction

6990 The Resource through which a Device can monitor network traffic.  
6991

### 6992 D.10.2 Example URI

6993 /nmonResURI

### 6994 D.10.3 Resource type

6995 The Resource Type is defined as: "oic.wk.nmon".

### 6996 D.10.4 OpenAPI 2.0 definition

```

6997 {
6998   "swagger": "2.0",

```



```

6999     "info": {
7000         "title": "Network Monitoring",
7001         "version": "2019-03-27",
7002         "license": {
7003             "name": "OCF Data Model License",
7004             "url":
7005 "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
7006 CENSE.md",
7007             "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
7008         },
7009         "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
7010     },
7011     "schemes": ["http"],
7012     "consumes": ["application/json"],
7013     "produces": ["application/json"],
7014     "paths": {
7015         "/nmonResURI" : {
7016             "get": {
7017                 "description": "The Resource through which a Device can monitor network traffic.\n",
7018                 "parameters": [
7019                     {"$ref": "#/parameters/interface-all"}
7020                 ],
7021                 "responses": {
7022                     "200": {
7023                         "description": "",
7024                         "x-example": {
7025                             "rt": ["oic.wk.nmon"],
7026                             "ianaifType": 71,
7027                             "reset": false,
7028                             "col" : false,
7029                             "tx" : 10,
7030                             "rx" : 15,
7031                             "mmstx" : 50,
7032                             "amstx" : 35,
7033                             "mmsrx" : 35,
7034                             "amsrx" : 20
7035                         },
7036                         "schema": { "$ref": "#/definitions/nmon" }
7037                     }
7038                 }
7039             },
7040             "post": {
7041                 "description": "Start/Stop collecting and reset the networking monitor Resource\n",
7042                 "parameters": [
7043                     {"$ref": "#/parameters/interface-rw"},
7044                     {
7045                         "name": "body",
7046                         "in": "body",
7047                         "required": true,
7048                         "schema": { "$ref": "#/definitions/nmon-update" },
7049                         "x-example": {
7050                             "col": true,
7051                             "reset": true
7052                         }
7053                     }
7054                 ],
7055                 "responses": {
7056                     "200": {
7057                         "description": "",
7058                         "x-example": {
7059                             "rt": ["oic.wk.nmon"],
7060                             "ianaifType": 71,
7061                             "reset": false,
7062                             "col" : true,
7063                             "tx" : 0,
7064                             "rx" : 0,
7065                             "mmstx" : 0,
7066                             "amstx" : 0,
7067                             "mmsrx" : 0,
7068                             "amsrx" : 0
7069                         },

```

```

7070         "schema": { "$ref": "#/definitions/nmon" }
7071     }
7072 }
7073 }
7074 }
7075 },
7076 "parameters": {
7077     "interface-rw" : {
7078         "in" : "query",
7079         "name" : "if",
7080         "type" : "string",
7081         "enum" : ["oic.if.rw"]
7082     },
7083     "interface-all" : {
7084         "in" : "query",
7085         "name" : "if",
7086         "type" : "string",
7087         "enum" : ["oic.if.rw", "oic.if.baseline"]
7088     }
7089 },
7090 "definitions": {
7091     "nmon" : {
7092         "properties": {
7093             "amstx" : {
7094                 "description": "Average transmitted message size in bytes (tx) in the collection period",
7095                 "readOnly": true,
7096                 "type": "integer"
7097             },
7098             "reset" : {
7099                 "description": "True: reset the collected values",
7100                 "readOnly": false,
7101                 "type": "boolean"
7102             },
7103             "mmsrx" : {
7104                 "description": "Maximum received message size in bytes (rx) in the collection period",
7105                 "readOnly": true,
7106                 "type": "integer"
7107             },
7108             "mmstx" : {
7109                 "description": "Maximum transmitted message size in bytes (tx) in the collection period",
7110                 "readOnly": true,
7111                 "type": "integer"
7112             },
7113             "tx" : {
7114                 "description": "Amount of transmitted kilo bytes from the collection",
7115                 "readOnly": true,
7116                 "type": "integer"
7117             },
7118             "rt" : {
7119                 "description": "Resource Type of the Resource",
7120                 "items": {
7121                     "enum": ["oic.wk.nmon"],
7122                     "type": "string",
7123                     "maxLength": 64
7124                 },
7125                 "minItems": 1,
7126                 "uniqueItems": true,
7127                 "readOnly": true,
7128                 "type": "array"
7129             },
7130             "ianaifType" : {
7131                 "description": "The type of the network connection, as defined by iana
7132 https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib",
7133                 "readOnly": true,
7134                 "type": "integer"
7135             },
7136             "rx" : {
7137                 "description": "Amount of received kilobytes from the collection",
7138                 "readOnly": true,
7139                 "type": "integer"
7140             },

```

```

7141         "id" : {
7142             "$ref":
7143 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7144 schema.json#/definitions/id"
7145         },
7146         "amsrx" : {
7147             "description": "Average received message size in bytes (rx) in the collection period",
7148             "readOnly": true,
7149             "type": "integer"
7150         },
7151         "n" : {
7152             "$ref":
7153 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7154 schema.json#/definitions/n"
7155         },
7156         "col" : {
7157             "description": "True: Device is collecting values",
7158             "readOnly": false,
7159             "type": "boolean"
7160         },
7161         "if" : {
7162             "description": "The OCF Interfaces supported by this Resource",
7163             "items": {
7164                 "enum": [
7165                     "oic.if.rw",
7166                     "oic.if.baseline"
7167                 ],
7168                 "type": "string",
7169                 "maxLength": 64
7170             },
7171             "minItems": 1,
7172             "readOnly": true,
7173             "uniqueItems": true,
7174             "type": "array"
7175         }
7176     },
7177     "type" : "object",
7178     "required": ["reset", "col", "ianaifType"]
7179 },
7180 "nmon-update" : {
7181     "properties": {
7182         "reset" : {
7183             "description": "True: reset the collected values",
7184             "readOnly": false,
7185             "type": "boolean"
7186         },
7187         "n" : {
7188             "$ref":
7189 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7190 schema.json#/definitions/n"
7191         },
7192         "col" : {
7193             "description": "True: Device is collecting values",
7194             "readOnly": false,
7195             "type": "boolean"
7196         }
7197     },
7198     "type" : "object",
7199     "required": ["reset", "col"]
7200 }
7201 }
7202 }
7203

```

## 7204 D.10.5 Property definition

7205 Table D-18 defines the Properties that are part of the "oic.wk.nmon" Resource Type.

**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 <a href="https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib">https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib</a>
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

7207 **D.10.6 CRUDN behaviour**

7208 Table D-19 defines the CRUDN operations that are supported on the "oic.wk.nmon" Resource Type.

7209 **Table D-19 – The CRUDN operations of the Resource with type "rt" = "oic.wk.nmon".**

Create	Read	Update	Delete	Notify
	get	post		observe

7210 **D.11 Platform**7211 **D.11.1 Introduction**

7212 Known Resource that is defines the Platform on which an Server is hosted.  
 7213 Allows for Platform specific information to be discovered.  
 7214

7215 **D.11.2 Well-known URI**

7216 /oic/p

### 7217 D.11.3 Resource type

7218 The Resource Type is defined as: "oic.wk.p".

### 7219 D.11.4 OpenAPI 2.0 definition

```
7220 {
7221   "swagger": "2.0",
7222   "info": {
7223     "title": "Platform",
7224     "version": "2019-03-04",
7225     "license": {
7226       "name": "OCF Data Model License",
7227       "url":
7228 "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
7229 CENSE.md",
7230       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
7231     },
7232     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
7233   },
7234   "schemes": ["http"],
7235   "consumes": ["application/json"],
7236   "produces": ["application/json"],
7237   "paths": {
7238     "/oic/p" : {
7239       "get": {
7240         "description": "Known Resource that is defines the Platform on which an Server is
7241 hosted.\nAllows for Platform specific information to be discovered.\n",
7242         "parameters": [
7243           {"$ref": "#/parameters/interface"}
7244         ],
7245         "responses": {
7246           "200": {
7247             "description" : "",
7248             "x-example": {
7249               "pi": "54919CA5-4101-4AE4-595B-353C51AA983C",
7250               "rt": ["oic.wk.p"],
7251               "mnmn": "Acme, Inc"
7252             },
7253             "schema": { "$ref": "#/definitions/Platform" }
7254           }
7255         }
7256       }
7257     }
7258   },
7259   "parameters": {
7260     "interface" : {
7261       "in": "query",
7262       "name": "if",
7263       "type": "string",
7264       "enum": ["oic.if.r", "oic.if.baseline"]
7265     }
7266   },
7267   "definitions": {
7268     "Platform" : {
7269       "properties": {
7270         "rt" : {
7271           "description": "Resource Type of the Resource",
7272           "items": {
7273             "enum": ["oic.wk.p"],
7274             "type": "string",
7275             "maxLength": 64
7276           },
7277           "minItems": 1,
7278           "uniqueItems": true,
7279           "readOnly": true,
7280           "type": "array"
7281         },
7282         "pi" : {
7283           "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
7284 9]{12}$",
```

```

7285         "type": "string",
7286         "description": "Platform Identifier",
7287         "readOnly": true
7288     },
7289     "mnfv" : {
7290         "description": "Manufacturer's firmware version",
7291         "maxLength": 64,
7292         "readOnly": true,
7293         "type": "string"
7294     },
7295     "vid" : {
7296         "description": "Manufacturer's defined information for the Platform. The content is
7297 freeform, with population rules up to the manufacturer",
7298         "maxLength": 64,
7299         "readOnly": true,
7300         "type": "string"
7301     },
7302     "mnmn" : {
7303         "description": "Manufacturer name",
7304         "maxLength": 64,
7305         "readOnly": true,
7306         "type": "string"
7307     },
7308     "mnmo" : {
7309         "description": "Model number as designated by the manufacturer",
7310         "maxLength": 64,
7311         "readOnly": true,
7312         "type": "string"
7313     },
7314     "mnhw" : {
7315         "description": "Platform Hardware Version",
7316         "maxLength": 64,
7317         "readOnly": true,
7318         "type": "string"
7319     },
7320     "mnos" : {
7321         "description": "Platform Resident OS Version",
7322         "maxLength": 64,
7323         "readOnly": true,
7324         "type": "string"
7325     },
7326     "mndt" : {
7327         "pattern": "^[0-9]{4}-(1[0-2]|0[1-9])-(3[0-1]|2[0-9]|1[0-9]|0[1-9])$",
7328         "type": "string",
7329         "description": "Manufacturing Date.",
7330         "readOnly": true
7331     },
7332     "id" : {
7333         "$ref":
7334 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7335 schema.json#/definitions/id"
7336     },
7337     "mns1" : {
7338         "description": "Manufacturer's Support Information URL",
7339         "format": "uri",
7340         "maxLength": 256,
7341         "readOnly": true,
7342         "type": "string"
7343     },
7344     "mnpv" : {
7345         "description": "Platform Version",
7346         "maxLength": 64,
7347         "readOnly": true,
7348         "type": "string"
7349     },
7350     "st" : {
7351         "description": "The date-time format pattern according to IETF RFC 3339.",
7352         "format": "date-time",
7353         "readOnly": true,
7354         "type": "string"
7355     },

```

```

7356     "n" : {
7357         "$ref":
7358         "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7359         schema.json#/definitions/n"
7360     },
7361     "mnmml" : {
7362         "description": "Manufacturer's URL",
7363         "format": "uri",
7364         "maxLength": 256,
7365         "readOnly": true,
7366         "type": "string"
7367     },
7368     "mnmnl" : {
7369         "description": "Serial number as designated by the manufacturer",
7370         "maxLength": 64,
7371         "readOnly": true,
7372         "type": "string"
7373     },
7374     "if" : {
7375         "description": "The OCF Interfaces supported by this Resource",
7376         "items": {
7377             "enum": [
7378                 "oic.if.r",
7379                 "oic.if.baseline"
7380             ],
7381             "type": "string",
7382             "maxLength": 64
7383         },
7384         "minItems": 2,
7385         "readOnly": true,
7386         "uniqueItems": true,
7387         "type": "array"
7388     },
7389     "mnmnct" : {
7390         "description": "An array of integers and each integer indicates the network connectivity
7391         type based on IANAIfType value as defined by: https://www.iana.org/assignments/ianaiftype-
7392         mib/ianaiftype-mib, e.g., [71, 259] which represents Wi-Fi and Zigbee.",
7393         "items": {
7394             "type": "integer",
7395             "minimum": 1,
7396             "description": "The network connectivity type based on IANAIfType value as defined by:
7397         https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib."
7398         },
7399         "minItems": 1,
7400         "readOnly": true,
7401         "type": "array"
7402     }
7403 },
7404 "type" : "object",
7405 "required": ["pi", "mnmn"]
7406 }
7407 }
7408 }
7409

```

#### 7410 D.11.5 Property definition

7411 Table D-20 defines the Properties that are part of the "oic.wk.p" Resource Type.

7412 **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.
mnmfv	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.
mnmn	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.
mnct	array: see schema	No	Read Only	An array of integers and each integer indicates the network connectivity type based on IANAIfType value as defined by: <a href="https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib">https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib</a> , e.g., [71, 259] which represents Wi-Fi and Zigbee.

7413 **D.11.6 CRUDN behaviour**

7414 Table D-21 defines the CRUDN operations that are supported on the "oic.wk.p" Resource Type.

7415 **Table D-21 – The CRUDN operations of the Resource with type "rt" = "oic.wk.p".**

Create	Read	Update	Delete	Notify
	get			observe

7416 **D.12 Resource directory resource**

7417 **D.12.1 Introduction**

7418 Resource to be exposed by any Device that can act as a Resource Directory.

7419 1) Provides selector criteria (e.g., integer) with GET request

7420 2) Publish a Link in /oic/res with POST request

7421

7422 **D.12.2 Well-known URI**

7423 /oic/rd

7424 **D.12.3 Resource type**

7425 The Resource Type is defined as: "oic.wk.rd".



## 7426 D.12.4 OpenAPI 2.0 definition

```
7427 {
7428   "swagger": "2.0",
7429   "info": {
7430     "title": "Resource directory resource",
7431     "version": "2019-02-22",
7432     "license": {
7433       "name": "OCF Data Model License",
7434       "url":
7435         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
7436 CENSE.md",
7437       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
7438     },
7439     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
7440   },
7441   "schemes": ["http"],
7442   "consumes": ["application/json"],
7443   "produces": ["application/json"],
7444   "paths": {
7445     "/oic/rd" : {
7446       "get": {
7447         "description": "Resource to be exposed by any Device that can act as a Resource
7448 Directory.\n1) Provides selector criteria (e.g., integer) with GET request\n2) Publish a Link in
7449 /oic/res with POST request\n",
7450         "parameters": [
7451           {"$ref": "#/parameters/rdgetinterface"}
7452         ],
7453         "responses": {
7454           "200": {
7455             "description": "Respond with the selector criteria - either the set of attributes or
7456 the bias factor\n",
7457             "x-example": {
7458               "rt": ["oic.wk.rd"],
7459               "if": ["oic.if.baseline"],
7460               "sel": 50
7461             },
7462             "schema": { "$ref": "#/definitions/rdSelection" }
7463           }
7464         }
7465       },
7466       "post": {
7467         "description": "Publish the Resource information for the first time in /oic/res. Updates to
7468 existing entries are not allowed.\nAppropriates parts of the information, i.e., Links of the
7469 published Resources will be discovered through /oic/res.\n1) When a Device first publishes a Link,
7470 the request payload to RD may include the Links without an \"ins\" Parameter.\n2) Upon granting the
7471 request, the RD assigns a unique instance value identifying the Link among all the Links it
7472 advertises\n and sends back the instance value in the \"ins\" Parameter in the Link to the
7473 publishing Device.\n",
7474         "parameters": [
7475           {"$ref": "#/parameters/rdpostinterface"},
7476           {
7477             "name": "body",
7478             "in": "body",
7479             "required": true,
7480             "schema": { "$ref": "#/definitions/rdPublish" },
7481             "x-example": {
7482               "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
7483               "links": [
7484                 {
7485                   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
7486                   "href": "/myLightSwitch",
7487                   "rt": [ "oic.r.switch.binary" ],
7488                   "if": [ "oic.if.a", "oic.if.baseline" ],
7489                   "p": { "bm": 3 },
7490                   "eps": [
7491                     { "ep": "coaps://[2001:db8:a::b1d6]:1111", "pri": 2 },
7492                     { "ep": "coaps://[2001:db8:a::b1d6]:1122" },
7493                     { "ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3 }
7494                   ]
7495                 }
7496               ]
7497             }
7498           }
7499         ]
7500       }
7501     }
7502   }
7503 }
```

```

7496         {
7497             "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
7498             "href": "/myLightBrightness",
7499             "rt": [ "oic.r.brightness" ],
7500             "if": [ "oic.if.a", "oic.if.baseline" ],
7501             "p": { "bm": 3 },
7502             "eps": [
7503                 { "ep": "coaps://[[2001:db8:a::123]:2222" }
7504             ]
7505         },
7506     ],
7507     "ttl": 600
7508 }
7509 },
7510 ],
7511 "responses": {
7512     "200": {
7513         "description" : "Respond with the same schema as publish with the additional \"ins\"
7514 Parameter in the Link.\n",
7515         "x-example": {
7516             "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
7517             "links": [
7518                 {
7519                     "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
7520                     "href": "/myLightSwitch",
7521                     "rt": [ "oic.r.switch.binary" ],
7522                     "if": [ "oic.if.a", "oic.if.baseline" ],
7523                     "p": { "bm": 3 },
7524                     "eps": [
7525                         { "ep": "coaps://[2001:db8:a::b1d6]:1111", "pri": 2 },
7526                         { "ep": "coaps://[2001:db8:a::b1d6]:1122" },
7527                         { "ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3 }
7528                     ],
7529                     "ins": 11235
7530                 },
7531                 {
7532                     "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
7533                     "href": "/myLightBrightness",
7534                     "rt": [ "oic.r.brightness" ],
7535                     "if": [ "oic.if.a", "oic.if.baseline" ],
7536                     "p": { "bm": 3 },
7537                     "eps": [
7538                         { "ep": "coaps://[2001:db8:a::123]:2222" }
7539                     ],
7540                     "ins": 112358
7541                 }
7542             ],
7543             "ttl": 600
7544         },
7545         "schema": { "$ref": "#/definitions/rdPublish" }
7546     }
7547 }
7548 },
7549 },
7550 ],
7551 "parameters": {
7552     "rdgetinterface" : {
7553         "in" : "query",
7554         "name" : "if",
7555         "type" : "string",
7556         "enum" : [ "oic.if.baseline" ]
7557     },
7558     "rdpostinterface" : {
7559         "in" : "query",
7560         "name" : "if",
7561         "type" : "string",
7562         "enum" : [ "oic.if.baseline" ]
7563     }
7564 },
7565 "definitions": {
7566     "rdSelection" : {

```

```

7567     "properties": {
7568       "rt" : {
7569         "description": "Resource Type of the Resource",
7570         "items": {
7571           "enum": ["oic.wk.rd"],
7572           "type": "string",
7573           "maxLength": 64
7574         },
7575         "minItems": 1,
7576         "uniqueItems": true,
7577         "readOnly": true,
7578         "type": "array"
7579       },
7580       "n" : {
7581         "$ref":
7582 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7583 schema.json#/definitions/n"
7584       },
7585       "sel" : {
7586         "description": "A bias factor calculated by the Resource Directory",
7587         "maximum": 100,
7588         "minimum": 0,
7589         "readOnly": true,
7590         "type": "integer"
7591       },
7592       "id" : {
7593         "$ref":
7594 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7595 schema.json#/definitions/id"
7596       },
7597       "if" : {
7598         "description": "The OCF Interfaces supported by this Resource",
7599         "items": {
7600           "enum": [
7601             "oic.if.baseline"
7602           ],
7603           "type": "string",
7604           "maxLength": 64
7605         },
7606         "minItems": 1,
7607         "readOnly": true,
7608         "uniqueItems": true,
7609         "type": "array"
7610       }
7611     },
7612     "type" : "object",
7613     "required": ["sel"]
7614   },
7615   "rdPublish" : {
7616     "properties": {
7617       "di" : {
7618         "$ref":
7619 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7620 schema.json#/definitions/di"
7621       },
7622       "ttl" : {
7623         "description": "Time to indicate a RD, i.e. how long to keep this published item.",
7624         "type": "integer"
7625       },
7626       "links" : {
7627         "description": "A set of simple or individual OCF Links.",
7628         "items": {
7629           "properties": {
7630             "anchor": {
7631               "$ref":
7632 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7633 schema.json#/definitions/anchor"
7634             },
7635             "di": {
7636               "$ref":
7637 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-

```

```

7638 schema.json#/definitions/di"
7639     },
7640     "eps": {
7641       "$ref":
7642       "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7643       schema.json#/definitions/eps"
7644     },
7645     "href": {
7646       "$ref":
7647       "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7648       schema.json#/definitions/href"
7649     },
7650     "if": {
7651       "description": "The interface set supported by the published resource",
7652       "items": {
7653         "enum": [
7654           "oic.if.baseline",
7655           "oic.if.ll",
7656           "oic.if.b",
7657           "oic.if.rw",
7658           "oic.if.r",
7659           "oic.if.a",
7660           "oic.if.s"
7661         ],
7662         "type": "string",
7663         "maxLength": 64
7664       },
7665       "minItems": 1,
7666       "uniqueItems": true,
7667       "type": "array"
7668     },
7669     "ins": {
7670       "$ref":
7671       "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7672       schema.json#/definitions/ins"
7673     },
7674     "p": {
7675       "$ref":
7676       "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7677       schema.json#/definitions/p"
7678     },
7679     "rel": {
7680       "description": "The relation of the target URI referenced by the Link to the context
7681       URI",
7682       "oneOf": [
7683         {
7684           "default": [
7685             "hosts"
7686           ],
7687           "items": {
7688             "maxLength": 64,
7689             "type": "string"
7690           },
7691           "minItems": 1,
7692           "type": "array"
7693         },
7694         {
7695           "default": "hosts",
7696           "maxLength": 64,
7697           "type": "string"
7698         }
7699       ]
7700     },
7701     "rt": {
7702       "description": "Resource Type of the published Resource",
7703       "items": {
7704         "maxLength": 64,
7705         "type": "string"
7706       },
7707       "minItems": 1,
7708       "maxItems": 1,

```

```

7709         "uniqueItems": true,
7710         "type": "array"
7711     },
7712     "title": {
7713         "$ref":
7714         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7715         schema.json#/definitions/title"
7716     },
7717     "type": {
7718         "$ref":
7719         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7720         schema.json#/definitions/type"
7721     }
7722 },
7723 "required": [
7724     "href",
7725     "rt",
7726     "if"
7727 ],
7728 "type": "object"
7729 },
7730 "type": "array"
7731 }
7732 },
7733 "type": "object",
7734 "required": ["di", "links", "ttl"]
7735 }
7736 }
7737 }
7738

```

7739 **D.12.5 Property definition**

7740 Table D-22 defines the Properties that are part of the "oic.wk.rd" Resource Type.

7741 **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.

7742 **D.12.6 CRUDN behaviour**

7743 Table D-23 defines the CRUDN operations that are supported on the "oic.wk.rd" Resource Type.

7744 **Table D-23 – The CRUDN operations of the Resource with type "rt" = "oic.wk.rd".**

Create	Read	Update	Delete	Notify
	get	post		observe

7745 **D.13 Discoverable Resources**

7746 **D.13.1 Introduction**

7747 Baseline representation of /oic/res; list of discoverable Resources  
7748

7749 **D.13.2 Well-known URI**

7750 /oic/res

7751 **D.13.3 Resource type**

7752 The Resource Type is defined as: "oic.wk.res".

7753 **D.13.4 OpenAPI 2.0 definition**

```
7754 {
7755   "swagger": "2.0",
7756   "info": {
7757     "title": "Discoverable Resources",
7758     "version": "2019-03-13",
7759     "license": {
7760       "name": "OCF Data Model License",
7761       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
7762       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
7763     },
7764     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
7765   },
7766   "schemes": [
7767     "http"
7768   ],
7769   "consumes": [
7770     "application/json"
7771   ],
7772   "produces": [
7773     "application/json"
7774   ],
7775   "paths": {
7776     "/oic/res?if=oic.if.ll": {
7777       "get": {
7778         "description": "Links list representation of /oic/res; list of discoverable Resources\n",
7779         "parameters": [
7780           {
7781             "$ref": "#/parameters/interface-all"
7782           }
7783         ],
7784         "responses": {
7785           "200": {
7786             "description": "",
7787             "x-example": [
7788               {
7789                 "href": "/humidity",
7790                 "rt": ["oic.r.humidity"],
7791                 "if": ["oic.if.s", "oic.if.baseline"],
7792                 "p": {"bm": 3},
7793                 "eps": [
7794                   {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
7795                   {"ep": "coaps://[fe80::b1d6]:1122"},
7796                   {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
7797                 ]
7798               }
7799             ]
7800           }
7801         }
7802       }
7803     }
7804   }
7805 }
```

```

7800         "href": "/temperature",
7801         "rt": ["oic.r.temperature"],
7802         "if": ["oic.if.s", "oic.if.baseline"],
7803         "p": {"bm": 3},
7804         "eps": [
7805             {"ep": "coaps://[[2001:db8:a::123]:2222"}
7806         ]
7807     },
7808 ],
7809     "schema": {
7810         "$ref": "#/definitions/slinklist"
7811     }
7812 }
7813 }
7814 },
7815 },
7816 "/oic/res?if=oic.if.baseline": {
7817     "get": {
7818         "description": "Baseline representation of /oic/res; list of discoverable Resources\n",
7819         "parameters": [
7820             {
7821                 "$ref": "#/parameters/interface-all"
7822             }
7823         ],
7824         "responses": {
7825             "200": {
7826                 "description": "",
7827                 "x-example": [
7828                     {
7829                         "rt": ["oic.wk.res"],
7830                         "if": ["oic.if.ll", "oic.if.baseline"],
7831                         "links": [
7832                             {
7833                                 "href": "/humidity",
7834                                 "rt": ["oic.r.humidity"],
7835                                 "if": ["oic.if.s", "oic.if.baseline"],
7836                                 "p": {"bm": 3},
7837                                 "eps": [
7838                                     {"ep": "coaps://[fe80:b1d6]:1111", "pri": 2},
7839                                     {"ep": "coaps://[fe80:b1d6]:1122"},
7840                                     {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
7841                                 ]
7842                             },
7843                             {
7844                                 "href": "/temperature",
7845                                 "rt": ["oic.r.temperature"],
7846                                 "if": ["oic.if.s", "oic.if.baseline"],
7847                                 "p": {"bm": 3},
7848                                 "eps": [
7849                                     {"ep": "coaps://[[2001:db8:a::123]:2222"}
7850                                 ]
7851                             }
7852                         ]
7853                     }
7854                 ],
7855                 "schema": {
7856                     "$ref": "#/definitions/sbaseline"
7857                 }
7858             }
7859         }
7860     }
7861 },
7862 },
7863 "parameters": {
7864     "interface-all": {
7865         "in": "query",
7866         "name": "if",
7867         "type": "string",
7868         "enum": ["oic.if.ll", "oic.if.baseline"]
7869     }
7870 },

```

```

7871     "definitions": {
7872       "oic.oic-link": {
7873         "type": "object",
7874         "properties": {
7875           "anchor": {
7876             "$ref":
7877 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7878 schema.json#/definitions/anchor"
7879           },
7880           "di": {
7881             "$ref":
7882 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7883 schema.json#/definitions/di"
7884           },
7885           "eps": {
7886             "$ref":
7887 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7888 schema.json#/definitions/eps"
7889           },
7890           "href": {
7891             "$ref":
7892 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7893 schema.json#/definitions/href"
7894           },
7895           "if": {
7896             "description": "The OCF Interfaces supported by the Linked Resource",
7897             "items": {
7898               "enum": [
7899                 "oic.if.baseline",
7900                 "oic.if.ll",
7901                 "oic.if.b",
7902                 "oic.if.rw",
7903                 "oic.if.r",
7904                 "oic.if.a",
7905                 "oic.if.s"
7906               ],
7907               "type": "string",
7908               "maxLength": 64
7909             },
7910             "minItems": 1,
7911             "uniqueItems": true,
7912             "type": "array"
7913           },
7914           "ins": {
7915             "$ref":
7916 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7917 schema.json#/definitions/ins"
7918           },
7919           "p": {
7920             "$ref":
7921 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7922 schema.json#/definitions/p"
7923           },
7924           "rel": {
7925             "description": "The relation of the target URI referenced by the Link to the context URI",
7926             "oneOf": [
7927               {
7928                 "$ref":
7929 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7930 schema.json#/definitions/rel_array"
7931               },
7932               {
7933                 "$ref":
7934 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7935 schema.json#/definitions/rel_string"
7936               }
7937             ]
7938           },
7939           "rt": {
7940             "description": "Resource Type of the Linked Resource",
7941             "items": {

```



```

7942         "maxLength": 64,
7943         "type": "string"
7944     },
7945     "minItems": 1,
7946     "uniqueItems": true,
7947     "type": "array"
7948 },
7949 "title": {
7950     "$ref":
7951 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7952 schema.json#/definitions/title"
7953 },
7954 "type": {
7955     "$ref":
7956 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
7957 schema.json#/definitions/type"
7958 }
7959 },
7960 "required": [
7961     "href",
7962     "rt",
7963     "if"
7964 ]
7965 },
7966 "slinklist": {
7967     "type": "array",
7968     "readOnly": true,
7969     "items": {
7970         "$ref": "#/definitions/oic.oic-link"
7971     }
7972 },
7973 "sbaseline": {
7974     "type": "array",
7975     "minItems": 1,
7976     "maxItems": 1,
7977     "items": {
7978         "type": "object",
7979         "properties": {
7980             "n": {
7981                 "$ref":
7982 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7983 schema.json#/definitions/n"
7984             },
7985             "id": {
7986                 "$ref":
7987 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7988 schema.json#/definitions/id"
7989             },
7990             "rt": {
7991                 "description": "Resource Type of this Resource",
7992                 "items": {
7993                     "enum": ["oic.wk.res"],
7994                     "type": "string",
7995                     "maxLength": 64
7996                 },
7997                 "minItems": 1,
7998                 "readOnly": true,
7999                 "uniqueItems": true,
8000                 "type": "array"
8001             },
8002             "if": {
8003                 "description": "The OCF Interfaces supported by this Resource",
8004                 "items": {
8005                     "enum": [
8006                         "oic.if.ll",
8007                         "oic.if.baseline"
8008                     ],
8009                     "type": "string",
8010                     "maxLength": 64
8011                 },
8012                 "minItems": 2,

```

```

8013         "readOnly": true,
8014         "uniqueItems": true,
8015         "type": "array"
8016     },
8017     "links": {
8018         "type": "array",
8019         "items": {
8020             "$ref": "#/definitions/oic.oic-link"
8021         }
8022     }
8023 },
8024 },
8025 "required": [
8026     "rt",
8027     "if",
8028     "links"
8029 ]
8030 }
8031 }
8032 }
8033 }

```

### 8034 D.13.5 Property definition

8035 Table D-24 defines the Properties that are part of the "oic.wk.res" Resource Type.

8036 **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	

8037 **D.13.6 CRUDN behaviour**

8038 Table D-25 defines the CRUDN operations that are supported on the "None" Resource Type.

8039 **Table D-25 – The CRUDN operations of the Resource with type "rt" = "None".**

Create	Read	Update	Delete	Notify
	get			observe

8040 **D.14 Scene List**

8041 **D.14.1 Introduction**

8042 Toplevel Scene Resource.  
8043 This Resource is a generic Collection Resource.  
8044 The rts value contains oic.wk.scenecollection Resource Types.  
8045

8046 **D.14.2 Example URI**

8047 /SceneListResURI

8048 **D.14.3 Resource type**

8049 The Resource Type is defined as: "oic.wk.scenelist".

8050 **D.14.4 OpenAPI 2.0 definition**

```

8051 {
8052   "swagger": "2.0",
8053   "info": {
8054     "title": "Scene List",
8055     "version": "2019-03-04",
8056     "license": {
8057       "name": "OCF Data Model License",
8058       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
8059       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
8060     },
8061     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
8062   },
8063   "schemes": [
8064     "http"
8065   ],
8066   "consumes": [
8067     "application/json"
8068   ],
8069   "produces": [
8070     "application/json"
8071   ],
8072   "paths": {
8073     "/SceneListResURI?if=oic.if.ll": {
8074       "get": {
8075         "description": "Toplevel Scene Resource.\nThis Resource is a generic Collection
8076 Resource.\nThe rts value contains oic.wk.scenecollection Resource Types.\n",
8077         "parameters": [
8078           {
8079             "$ref": "#/parameters/interface-all"
8080           }
8081         ]
8082       }
8083     }
8084   }

```

```

8081     ],
8082     "responses": {
8083         "200": {
8084             "description" : "",
8085             "x-example": [
8086                 {"href": "/scenecollection1", "rt": ["oic.wk.scenecollection"], "if": ["oic.if.ll",
"oic.if.baseline"]},
8087                 {"href": "/scenecollection2", "rt": ["oic.wk.scenecollection"], "if": ["oic.if.ll",
"oic.if.baseline"]}
8088             ],
8089             "schema": {
8090                 "$ref": "#/definitions/slinks"
8091             }
8092         }
8093     }
8094 },
8095 },
8096 },
8097 },
8098 "/SceneListResURI?if=oic.if.baseline": {
8099     "get": {
8100         "description": "Toplevel Scene Resource.\nThis Resource is a generic Collection
8101 Resource.\nThe rts value contains oic.wk.scenecollection Resource Types.\n",
8102         "parameters": [
8103             {
8104                 "$ref": "#/parameters/interface-all"
8105             }
8106         ],
8107         "responses": {
8108             "200": {
8109                 "description" : "",
8110                 "x-example": {
8111                     "rt": ["oic.wk.scenelist"],
8112                     "if": ["oic.if.ll", "oic.if.baseline"],
8113                     "n": "list of scene collections",
8114                     "rts": ["oic.wk.scenecollection"],
8115                     "links": [
8116                         {"href": "/scenecollection1", "rt": ["oic.wk.scenecollection"], "if": ["oic.if.ll",
"oic.if.baseline"]},
8117                         {"href": "/scenecollection2", "rt": ["oic.wk.scenecollection"], "if": ["oic.if.ll",
"oic.if.baseline"]}
8118                     ]
8119                 }
8120             },
8121             "schema": { "$ref": "#/definitions/Collection" }
8122         }
8123     }
8124 },
8125 },
8126 },
8127 },
8128 "parameters": {
8129     "interface-all" : {
8130         "in" : "query",
8131         "name" : "if",
8132         "type" : "string",
8133         "enum" : ["oic.if.ll", "oic.if.baseline"]
8134     }
8135 },
8136 "definitions": {
8137     "Collection": {
8138         "properties": {
8139             "links": {
8140                 "description": "A set of simple or individual OCF Links.",
8141                 "items": {
8142                     "$ref": "#/definitions/oic.oic-link"
8143                 },
8144                 "type": "array"
8145             },
8146             "n": {
8147                 "$ref" :
8148                 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
8149                 schema.json#/definitions/n"
8150             },
8151             "id": {

```

```

8152         "$ref" :
8153         "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
8154         schema.json#/definitions/id"
8155     },
8156     "if": {
8157         "type": "array",
8158         "description": "The OCF Interfaces supported by this Resource",
8159         "items": {
8160             "enum": [
8161                 "oic.if.ll",
8162                 "oic.if.baseline"
8163             ],
8164             "type": "string",
8165             "maxLength": 64
8166         },
8167         "minItems": 2,
8168         "uniqueItems": true,
8169         "readOnly": true
8170     },
8171     "rts": {
8172         "description": "The list of allowable Resource Types in Links included in the Collection",
8173         "items": {
8174             "enum": ["oic.wk.scenecollection"],
8175             "type": "string",
8176             "maxLength": 64
8177         },
8178         "minItems": 1,
8179         "uniqueItems": true,
8180         "readOnly": true,
8181         "type": "array"
8182     },
8183     "rt": {
8184         "description": "Resource Type of the Resource",
8185         "items": {
8186             "enum": ["oic.wk.scenelist"],
8187             "type": "string",
8188             "maxLength": 64
8189         },
8190         "minItems": 1,
8191         "readOnly": true,
8192         "uniqueItems": true,
8193         "type": "array"
8194     }
8195 },
8196 "type": "object",
8197 "required": [
8198     "rt",
8199     "if",
8200     "links"
8201 ]
8202 },
8203 "slinks" : {
8204     "type" : "array",
8205     "items" : {
8206         "$ref": "#/definitions/oic.oic-link"
8207     }
8208 },
8209 "oic.oic-link": {
8210     "properties": {
8211         "if": {
8212             "description": "The OCF Interfaces supported by the Linked Resource",
8213             "items": {
8214                 "enum": [
8215                     "oic.if.ll",
8216                     "oic.if.baseline"
8217                 ],
8218                 "type": "string",
8219                 "maxLength": 64
8220             },
8221             "minItems": 1,
8222             "uniqueItems": true,

```

```

8223         "readOnly": true,
8224         "type": "array"
8225     },
8226     "rt": {
8227         "description": "The Resource Type of the Linked Resource",
8228         "items": {
8229             "enum": ["oic.wk.scenecollection"],
8230             "type": "string",
8231             "maxLength": 64
8232         },
8233         "minItems": 1,
8234         "uniqueItems": true,
8235         "readOnly": true,
8236         "type": "array"
8237     },
8238     "anchor": {
8239         "$ref":
8240         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8241         schema.json#/definitions/anchor"
8242     },
8243     "di": {
8244         "$ref":
8245         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8246         schema.json#/definitions/di"
8247     },
8248     "eps": {
8249         "$ref":
8250         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8251         schema.json#/definitions/eps"
8252     },
8253     "href": {
8254         "$ref":
8255         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8256         schema.json#/definitions/href"
8257     },
8258     "ins": {
8259         "$ref":
8260         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8261         schema.json#/definitions/ins"
8262     },
8263     "p": {
8264         "$ref":
8265         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8266         schema.json#/definitions/p"
8267     },
8268     "rel": {
8269         "$ref":
8270         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8271         schema.json#/definitions/rel_array"
8272     },
8273     "title": {
8274         "$ref":
8275         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8276         schema.json#/definitions/title"
8277     },
8278     "type": {
8279         "$ref":
8280         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8281         schema.json#/definitions/type"
8282     }
8283 },
8284 "required": [
8285     "href",
8286     "rt",
8287     "if"
8288 ],
8289 "type": "object"
8290 }
8291 }
8292 }
8293

```

8294 **D.14.5 Property definition**

8295 Table D-26 defines the Properties that are part of the "oic.wk.scenelist" Resource Type.

8296 **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	

8297 **D.14.6 CRUDN behaviour**

8298 Table D-27 defines the CRUDN operations that are supported on the "oic.wk.scenelist" Resource Type.  
8299

8300 **Table D-27 – The CRUDN operations of the Resource with type "rt" = "oic.wk.scenelist".**

Create	Read	Update	Delete	Notify
	get			observe

8301 **D.15 Scene Collection**

8302 **D.15.1 Introduction**

8303 Collection that models a set of Scenes.  
 8304 This Resource is a generic Collection Resource with additional Properties.  
 8305 The rts value contains oic.scenemember Resource Types.  
 8306 The additional Properties are  
 8307 lastScene, this is the Scene Value last set by any Client  
 8308 sceneValues, this is the list of available Scenes  
 8309 lastScene shall be listed in sceneValues.  
 8310

8311 **D.15.2 Example URI**

8312 /SceneCollectionResURI

8313 **D.15.3 Resource type**

8314 The Resource Type is defined as: "oic.wk.sceneecollection".

8315 **D.15.4 OpenAPI 2.0 definition**

```

8316 {
8317   "swagger": "2.0",
8318   "info": {
8319     "title": "Scene Collection",
8320     "version": "2019-03-04",
8321     "license": {
8322       "name": "OCF Data Model License",
8323       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
8324       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
8325     },
8326     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
8327   },
8328   "schemes": [
8329     "http"
8330   ],
8331   "consumes": [
8332     "application/json"
8333   ],
8334   "produces": [
8335     "application/json"
8336   ],
8337   "paths": {
8338     "/SceneCollectionResURI?if=oic.if.ll" : {
8339       "get": {
8340         "description": "Collection that models a set of Scenes.\nThis Resource is a generic
8341 Collection Resource with additional Properties.\nThe rts value contains oic.scenemember Resource
8342 Types.\nThe additional Properties are\n lastScene, this is the Scene Value last set by any Client\n
8343 sceneValues, this is the list of available Scenes\n lastScene shall be listed in sceneValues.\n",
8344         "parameters": [
8345           {
8346             "$ref": "#/parameters/interface-all"
8347           }
8348         ],
8349         "responses": {
8350           "200": {
8351             "description": "",
8352             "x-example": [
8353               {"href": "/scenemember1", "rt": ["oic.wk.scenemember"], "if": ["oic.if.baseline"]},
8354               {"href": "/scenemember2", "rt": ["oic.wk.scenemember"], "if": ["oic.if.baseline"]}
8355             ]
8356           }
8357         }
8358       }
8359     }
8360   }
8361 }
    
```



```

8355         ],
8356         "schema": {
8357             "$ref": "#/definitions/slinks"
8358         }
8359     }
8360 }
8361 },
8362 },
8363 "/SceneCollectionResURI?if=oic.if.baseline" : {
8364     "get": {
8365         "description": "Collection that models a set of Scenes.\nThis Resource is a generic
8366 Collection Resource with additional Properties.\nThe rts value contains oic.scenemember Resource
8367 Types.\nThe additional Properties are\n lastScene, this is the Scene Value last set by any Client\n
8368 sceneValues, this is the list of available Scenes\n lastScene shall be listed in sceneValues.\n",
8369         "parameters": [
8370             {
8371                 "$ref": "#/parameters/interface-all"
8372             }
8373         ],
8374         "responses": {
8375             "200": {
8376                 "description" : "",
8377                 "x-example": {
8378                     "lastScene": "off",
8379                     "sceneValues": ["off","Reading","TVWatching"],
8380                     "rt": ["oic.wk.scenecollection"],
8381                     "n": "My Scenes for my living room",
8382                     "rts": ["oic.wk.scenemember"],
8383                     "links": [
8384                         {"href": "/scenemember1","rt": ["oic.wk.scenemember"],"if":["oic.if.baseline"]},
8385                         {"href": "/scenemember2","rt": ["oic.wk.scenemember"],"if":["oic.if.baseline"]}
8386                     ]
8387                 },
8388                 "schema": {
8389                     "$ref": "#/definitions/SceneCollection"
8390                 }
8391             }
8392         },
8393     },
8394     "post": {
8395         "description": "Provides the action to change the last set Scene selection.\nCalling this
8396 method shall update all Scene Members to the prescribed membervalue.\nWhen this method is called
8397 with the same value as the current lastScene value\nthen all Scene Members shall be updated.\n",
8398         "parameters": [
8399             {
8400                 "$ref": "#/parameters/interface-update"
8401             },
8402             {
8403                 "name": "body",
8404                 "in": "body",
8405                 "required": true,
8406                 "schema": {
8407                     "$ref": "#/definitions/SceneCollectionUpdate"
8408                 },
8409                 "x-example": {
8410                     "lastScene": "Reading"
8411                 }
8412             }
8413         ],
8414         "responses": {
8415             "200": {
8416                 "description" : "Indicates that the value is changed.\nThe changed Properties are
8417 provided in the response.\n",
8418                 "x-example": {
8419                     "lastScene": "Reading"
8420                 },
8421                 "schema": {
8422                     "$ref": "#/definitions/SceneCollectionUpdate"
8423                 }
8424             }
8425         }
8426     }
8427 }

```

```

8426     }
8427   }
8428 },
8429 "parameters": {
8430   "interface-update" : {
8431     "in" : "query",
8432     "name" : "if",
8433     "type" : "string",
8434     "enum" : ["oic.if.a"]
8435   },
8436   "interface-all" : {
8437     "in" : "query",
8438     "name" : "if",
8439     "type" : "string",
8440     "enum" : ["oic.if.ll", "oic.if.baseline"]
8441   }
8442 },
8443 "definitions": {
8444   "SceneCollection": {
8445     "properties": {
8446       "rt": {
8447         "description": "Resource Type of the Resource",
8448         "items": {
8449           "enum": ["oic.wk.scenecollection"],
8450           "type": "string",
8451           "maxLength": 64
8452         },
8453         "minItems": 1,
8454         "readOnly": true,
8455         "uniqueItems": true,
8456         "type": "array"
8457       },
8458       "lastScene": {
8459         "description": "Last selected Scene from the set of sceneValues",
8460         "type": "string"
8461       },
8462       "links": {
8463         "description": "A set of simple or individual OCF Links.",
8464         "items": {
8465           "$ref": "#/definitions/oic.oic-link"
8466         },
8467         "type": "array"
8468       },
8469       "sceneValues": {
8470         "description": "All available Scene Values",
8471         "items": {
8472           "type": "string"
8473         },
8474         "readOnly": true,
8475         "type": "array"
8476       },
8477       "n": {
8478         "$ref" :
8479 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
8480 schema.json#/definitions/n"
8481       },
8482       "id": {
8483         "$ref" :
8484 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
8485 schema.json#/definitions/id"
8486       },
8487       "rts": {
8488         "description": "Resource Type of the Resources within the Collection",
8489         "items": {
8490           "enum": ["oic.wk.scenemember"],
8491           "type": "string",
8492           "maxLength": 64
8493         },
8494         "minItems": 1,
8495         "readOnly": true,
8496         "uniqueItems": true,

```

```

8497         "type": "array"
8498     },
8499     "if" : {
8500         "description": "The OCF Interfaces supported by this Resource",
8501         "items": {
8502             "enum": [
8503                 "oic.if.ll",
8504                 "oic.if.baseline",
8505                 "oic.if.a"
8506             ],
8507             "type": "string",
8508             "maxLength": 64
8509         },
8510         "minItems": 1,
8511         "uniqueItems": true,
8512         "readOnly": true,
8513         "type": "array"
8514     }
8515 },
8516 "type" : "object"
8517 },
8518 "SceneCollectionUpdate": {
8519     "properties": {
8520         "lastScene": {
8521             "description": "Last selected Scene from the set of sceneValues",
8522             "type": "string"
8523         }
8524     },
8525     "type" : "object"
8526 },
8527 "slinks" : {
8528     "type" : "array",
8529     "items" : {
8530         "$ref": "#/definitions/oic.oic-link"
8531     }
8532 },
8533 "oic.oic-link": {
8534     "type": "object",
8535     "properties": {
8536         "if": {
8537             "description": "The OCF Interfaces supported by the Linked Resource",
8538             "items": {
8539                 "enum": [
8540                     "oic.if.baseline"
8541                 ],
8542                 "type": "string",
8543                 "maxLength": 64
8544             },
8545             "minItems": 1,
8546             "uniqueItems": true,
8547             "readOnly": true,
8548             "type": "array"
8549         },
8550         "rt": {
8551             "description": "Resource Type of the Linked Resource",
8552             "items": {
8553                 "enum": ["oic.wk.scenemember"],
8554                 "type": "string",
8555                 "maxLength": 64
8556             },
8557             "minItems": 1,
8558             "uniqueItems": true,
8559             "readOnly": true,
8560             "type": "array"
8561         },
8562         "anchor": {
8563             "$ref":
8564 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8565 schema.json#/definitions/anchor"
8566         },
8567         "di": {

```

```

8568         "$ref":
8569         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8570         schema.json#/definitions/di"
8571     },
8572     "eps": {
8573         "$ref":
8574         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8575         schema.json#/definitions/eps"
8576     },
8577     "href": {
8578         "$ref":
8579         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8580         schema.json#/definitions/href"
8581     },
8582     "ins": {
8583         "$ref":
8584         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8585         schema.json#/definitions/ins"
8586     },
8587     "p": {
8588         "$ref":
8589         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8590         schema.json#/definitions/p"
8591     },
8592     "rel": {
8593         "$ref":
8594         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8595         schema.json#/definitions/rel_array"
8596     },
8597     "title": {
8598         "$ref":
8599         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8600         schema.json#/definitions/title"
8601     },
8602     "type": {
8603         "$ref":
8604         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8605         schema.json#/definitions/type"
8606     }
8607 },
8608 "required": [
8609     "href",
8610     "rt",
8611     "if"
8612 ]
8613 }
8614 }
8615 }
8616

```

### 8617 D.15.5 Property definition

8618 Table D-28 defines the Properties that are part of the "oic.wk.scenecollection" Resource Type.

8619 **Table D-28 – The Property definitions of the Resource with type "rt" =**  
8620 **"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	

8621 **D.15.6 CRUDN behaviour**

8622 Table D-29 defines the CRUDN operations that are supported on the "oic.wk.scenecollection"  
8623 Resource Type.

8624 **Table D-29 – The CRUDN operations of the Resource with type "rt" =**  
8625 **"oic.wk.scenecollection".**

Create	Read	Update	Delete	Notify
	get	post		observe

## 8626 **D.16 Scene Member**

### 8627 **D.16.1 Introduction**

8628 Single Link that models a Scene Member.  
8629

### 8630 **D.16.2 Example URI**

8631 /SceneMemberResURI

### 8632 **D.16.3 Resource type**

8633 The Resource Type is defined as: "oic.wk.scenemember".

### 8634 **D.16.4 OpenAPI 2.0 definition**

```
8635 {  
8636   "swagger": "2.0",  
8637   "info": {  
8638     "title": "Scene Member",  
8639     "version": "2019-03-04",  
8640     "license": {  
8641       "name": "OCF Data Model License",  
8642       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",  
8643       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."  
8644     },  
8645     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"  
8646   },  
8647   "schemes": [  
8648     "http"  
8649   ],  
8650   "consumes": [  
8651     "application/json"  
8652   ],  
8653   "produces": [  
8654     "application/json"  
8655   ],  
8656   "paths": {  
8657     "/SceneMemberResURI" : {  
8658       "get": {  
8659         "description": "Single Link that models a Scene Member.\n",  
8660         "parameters": [  
8661           {  
8662             "$ref": "#/parameters/interface-baseline"  
8663           }  
8664         ],  
8665         "responses": {  
8666           "200": {  
8667             "description" : "",  
8668             "x-example": {  
8669               "rt": ["oic.wk.scenemember"],  
8670               "id": "0685B960-FFFF-46F7-BEC0-9E6234671ADC1",  
8671               "n": "my binary switch (for light bulb) mappings",  
8672               "if": ["oic.if.baseline"],  
8673               "link": {  
8674                 "href": "binarySwitch",  
8675                 "rt": ["oic.r.switch.binary"],  
8676                 "if": ["oic.if.a", "oic.if.baseline"],  
8677                 "eps": [  
8678                   {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},  
8679                   {"ep": "coaps://[fe80::b1d6]:1122"},  
8680                   {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}  
8681                 ]  
8682               },  
8683               "SceneMappings": [  
8684                 {  
8685                   "scene": "off",  
8686                   "memberProperty": "value",  
8687                   "memberValue": "true"
```

```

8688         },
8689         {
8690             "scene":          "Reading",
8691             "memberProperty": "value",
8692             "memberValue":   "false"
8693         },
8694         {
8695             "scene":          "TVWatching",
8696             "memberProperty": "value",
8697             "memberValue":   "true"
8698         }
8699     ]
8700 },
8701 "schema": {
8702     "$ref": "#/definitions/SceneMember"
8703 }
8704 }
8705 }
8706 }
8707 }
8708 },
8709 "parameters": {
8710     "interface-baseline" : {
8711         "in" : "query",
8712         "name" : "if",
8713         "type" : "string",
8714         "enum" : ["oic.if.baseline"]
8715     }
8716 },
8717 "definitions": {
8718     "SceneMember": {
8719         "properties": {
8720             "rt": {
8721                 "description": "Resource Type of the Resource",
8722                 "items": {
8723                     "enum": ["oic.wk.scenemember"],
8724                     "type": "string",
8725                     "maxLength": 64
8726                 },
8727                 "minItems": 1,
8728                 "readOnly": true,
8729                 "uniqueItems": true,
8730                 "type": "array"
8731             },
8732             "SceneMappings": {
8733                 "description": "Array of mappings per Scene, can be one(1)",
8734                 "items": {
8735                     "properties": {
8736                         "memberProperty": {
8737                             "description": "Property name that will be mapped",
8738                             "readOnly": true,
8739                             "type": "string"
8740                         },
8741                         "memberValue": {
8742                             "description": "Value of the Member Property",
8743                             "readOnly": true,
8744                             "type": "string"
8745                         },
8746                         "scene": {
8747                             "description": "Specifies a Scene Value that will be acted upon",
8748                             "type": "string"
8749                         }
8750                     },
8751                     "required": [
8752                         "scene",
8753                         "memberProperty",
8754                         "memberValue"
8755                     ],
8756                     "type": "object"
8757                 },
8758                 "type": "array"

```

```

8759     },
8760     "n": {
8761       "$ref" :
8762       "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
8763       schema.json#/definitions/n"
8764     },
8765     "id": {
8766       "$ref" :
8767       "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
8768       schema.json#/definitions/id"
8769     },
8770     "link": {
8771       "$ref": "#/definitions/oic.oic-link"
8772     },
8773     "if": {
8774       "description": "The OCF Interfaces supported by this Resource",
8775       "items": {
8776         "enum": [
8777           "oic.if.baseline"
8778         ],
8779         "type": "string",
8780         "maxLength": 64
8781       },
8782       "minItems": 1,
8783       "readOnly": true,
8784       "uniqueItems": true,
8785       "type": "array"
8786     }
8787   },
8788   "type" : "object",
8789   "required": [
8790     "rt",
8791     "if",
8792     "SceneMappings"
8793   ]
8794 },
8795 "oic.oic-link": {
8796   "properties": {
8797     "if": {
8798       "description": "The OCF Interfaces supported by the target Resource",
8799       "items": {
8800         "enum": [
8801           "oic.if.baseline",
8802           "oic.if.ll",
8803           "oic.if.b",
8804           "oic.if.lb",
8805           "oic.if.rw",
8806           "oic.if.r",
8807           "oic.if.a",
8808           "oic.if.s"
8809         ],
8810         "type": "string",
8811         "maxLength": 64
8812       },
8813       "minItems": 1,
8814       "uniqueItems": true,
8815       "readOnly": true,
8816       "type": "array"
8817     },
8818     "rt": {
8819       "description": "Resource Type of the target Resource",
8820       "items": {
8821         "type": "string",
8822         "maxLength": 64
8823       },
8824       "minItems": 1,
8825       "readOnly": true,
8826       "uniqueItems": true,
8827       "type": "array"
8828     },
8829     "anchor": {

```



```

8830         "$ref":
8831         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8832         schema.json#/definitions/anchor"
8833     },
8834     "di": {
8835         "$ref":
8836         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8837         schema.json#/definitions/di"
8838     },
8839     "eps": {
8840         "$ref":
8841         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8842         schema.json#/definitions/eps"
8843     },
8844     "href": {
8845         "$ref":
8846         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8847         schema.json#/definitions/href"
8848     },
8849     "ins": {
8850         "$ref":
8851         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8852         schema.json#/definitions/ins"
8853     },
8854     "p": {
8855         "$ref":
8856         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8857         schema.json#/definitions/p"
8858     },
8859     "rel": {
8860         "$ref":
8861         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8862         schema.json#/definitions/rel_array"
8863     },
8864     "title": {
8865         "$ref":
8866         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8867         schema.json#/definitions/title"
8868     },
8869     "type": {
8870         "$ref":
8871         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
8872         schema.json#/definitions/type"
8873     }
8874 },
8875 "required": [
8876     "href",
8877     "rt",
8878     "if"
8879 ],
8880 "type": "object"
8881 }
8882 }
8883 }
8884

```

### 8885 D.16.5 Property definition

8886 Table D-30 defines the Properties that are part of the "oic.wk.scenemember" Resource Type.

8887 **Table D-30 – The Property definitions of the Resource with type "rt" =**  
8888 **"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	

8889 **D.16.6 CRUDN behaviour**

8890 Table D-31 defines the CRUDN operations that are supported on the "oic.wk.scenemember"  
8891 Resource Type.

8892 **Table D-31 – The CRUDN operations of the Resource with type "rt" =**  
8893 **"oic.wk.scenemember".**

Create	Read	Update	Delete	Notify
	get			observe

8894 **D.17 Alert**

8895 **D.17.1 Introduction**

8896 This Resource provides a mechanism for a Server to expose information to an  
8897 interested party with regard to error or other conditions that the Device is experiencing (Alerts).  
8898 category is a string that contains the Device defined category for the Alert.  
8899 timestamp is an RFC3339 formatted time at which the Alert was generated.

8900 originatorid is a string that contains the identity of the originator of the Alert.  
8901 severity is an integer that contains the RFC5424 defined severity of the Alert.  
8902 subject is an array containing human readable text in one or more languages.  
8903 accountid is a string containing the identity of the account with which the Device is associated.  
8904

## 8905 **D.17.2 Example URI**

8906 /AlertResURI

## 8907 **D.17.3 Resource type**

8908 The Resource Type is defined as: "oic.r.alert".

## 8909 **D.17.4 OpenAPI 2.0 definition**

```
8910 {
8911   "swagger": "2.0",
8912   "info": {
8913     "title": "Alert",
8914     "version": "2019-02-28",
8915     "license": {
8916       "name": "OCF Data Model License",
8917       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
8918       "x-copyright": "Copyright 2019 Open Connectivity Foundation, Inc. All rights reserved."
8919     },
8920     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
8921   },
8922   "schemes": ["http"],
8923   "consumes": ["application/json"],
8924   "produces": ["application/json"],
8925   "paths": {
8926     "/AlertResURI" : {
8927       "get": {
8928         "description": "This Resource provides a mechanism for a Server to expose information to
8929 an\ninterested party with regard to error or other conditions that the Device is experiencing
8930 (Alerts).\nncategory is a string that contains the Device defined category for the Alert.\ntimestamp
8931 is an RFC3339 formatted time at which the Alert was generated.\noriginatorid is a string that
8932 contains the identity of the originator of the Alert.\nseverity is an integer that contains the
8933 RFC5424 defined severity of the Alert.\nsubject is an array containing human readable text in one or
8934 more languages.\naccountid is a string containing the identity of the account with which the Device
8935 is associated.\n",
8936         "parameters": [
8937           { "$ref": "#/parameters/interface" }
8938         ],
8939         "responses": {
8940           "200": {
8941             "description": "",
8942             "x-example":
8943               {
8944                 "rt": ["oic.r.alert"],
8945                 "accountid": "MyAccountID",
8946                 "category": "MyCategory",
8947                 "timestamp": "2018-02-28T08:00:00Z",
8948                 "originatorid": "MyOriginatorID",
8949                 "severity": 3,
8950                 "subject": [{"language": "en-US", "value": "System error"}]
8951               },
8952             "schema": { "$ref": "#/definitions/Alert" }
8953           }
8954         }
8955       }
8956     }
8957   },
8958   "parameters": {
8959     "interface" : {
8960       "in" : "query",
8961       "name" : "if",
8962       "type" : "string",
```

```

8963     "enum" : ["oic.if.r", "oic.if.baseline"]
8964   }
8965 },
8966 "definitions": {
8967   "Alert" : {
8968     "properties": {
8969       "category": {
8970         "description": "Category into which the notification is classified",
8971         "maxLength": 64,
8972         "readOnly": true,
8973         "type": "string"
8974       },
8975       "rt": {
8976         "description": "Resource Type",
8977         "items": {
8978           "maxLength": 64,
8979           "type": "string",
8980           "enum": ["oic.r.alert"]
8981         },
8982         "minItems": 1,
8983         "readOnly": true,
8984         "uniqueItems": true,
8985         "type": "array"
8986       },
8987       "severity": {
8988         "description": "RFC 5424 severity of the alert",
8989         "maximum": 7,
8990         "minimum": 0,
8991         "readOnly": true,
8992         "type": "integer"
8993       },
8994       "timestamp": {
8995         "description": "An RFC3339 formatted time indicating when the data was observed (e.g.:
8996 2016-02-15T09:19Z, 1996-12-19T16:39:57-08:00)",
8997         "format": "date-time",
8998         "readOnly": true,
8999         "type": "string"
9000       },
9001       "subject": {
9002         "description": "Alert subject matter.",
9003         "items": {
9004           "properties": {
9005             "language": {
9006               "allOf": [
9007                 {
9008                   "description": "An identifier formatted according to IETF RFC 5646 (language
9009 tag).",
9010                   "pattern": "^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$",
9011                   "type": "string"
9012                 },
9013                 {
9014                   "description": "An RFC 5646 language tag.",
9015                   "readOnly": true
9016                 }
9017               ]
9018             },
9019             "value": {
9020               "description": "Alert subject matter in the indicated language.",
9021               "maxLength": 255,
9022               "readOnly": true,
9023               "type": "string"
9024             }
9025           },
9026           "type": "object"
9027         },
9028         "minItems": 1,
9029         "readOnly": true,
9030         "type": "array"
9031       },
9032       "originatorid": {
9033         "description": "ID of the creator of the event",

```

```

9034         "maxLength": 64,
9035         "readOnly": true,
9036         "type": "string"
9037     },
9038     "n": {
9039         "$ref" :
9040         "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
9041         schema.json#/definitions/n"
9042     },
9043     "id": {
9044         "$ref" :
9045         "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
9046         schema.json#/definitions/id"
9047     },
9048     "accountid": {
9049         "description": "ID of the account",
9050         "maxLength": 64,
9051         "readOnly": true,
9052         "type": "string"
9053     },
9054     "if": {
9055         "description": "The OCF Interfaces supported by this Resource",
9056         "items": {
9057             "enum": [
9058                 "oic.if.r",
9059                 "oic.if.baseline"
9060             ],
9061             "type": "string",
9062             "maxLength": 64
9063         },
9064         "minItems": 2,
9065         "readOnly": true,
9066         "uniqueItems": true,
9067         "type": "array"
9068     }
9069 },
9070 "type" : "object",
9071 "required": ["category", "timestamp", "originatorid", "severity"]
9072 }
9073 }
9074 }
9075

```

9076 **D.17.5 Property definition**

9077 Table D-32 defines the Properties that are part of the "oic.r.alert" Resource Type.

9078 **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.

9079 **D.17.6 CRUDN behaviour**

9080 Table D-33 defines the CRUDN operations that are supported on the "oic.r.alert" Resource Type.

9081 **Table D-33 – The CRUDN operations of the Resource with type "rt" = "oic.r.alert".**

Create	Read	Update	Delete	Notify
	get			observe

9082 **D.18 Alert Collection**

9083 **D.18.1 Introduction**

9084 This Resource is a Collection containing instances of Alerts (oic.r.alert).  
 9085 This is the response using the baseline interface.  
 9086

9087 **D.18.2 Example URI**

9088 /AlertCollectionResURI

9089 **D.18.3 Resource type**

9090 The Resource Type is defined as: "oic.r.alertcollection".

9091 **D.18.4 OpenAPI 2.0 definition**

```

9092 {
9093   "swagger": "2.0",
9094   "info": {
9095     "title": "Alert Collection",
9096     "version": "2019-03-04",
9097     "license": {
9098       "name": "OCF Data Model License",
9099       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
9100       "x-copyright": "Copyright 2019 Open Connectivity Foundation, Inc. All rights reserved."
9101     },
9102     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
9103   },
9104   "schemes": ["http"],
9105   "consumes": ["application/json"],
9106   "produces": ["application/json"],
9107   "paths": {
9108     "/AlertCollectionResURI?if=oic.if.ll" : {
9109       "get": {
9110         "description": "This Resource is a Collection containing instances of Alerts
(oic.r.alert).\nThis is the response using the links list OCF Interface.\n",

```

```

9112     "parameters": [
9113         {"$ref": "#/parameters/interface-all"}
9114     ],
9115     "responses": {
9116         "200": {
9117             "description": "",
9118             "x-example": [
9119                 {"href": "/myAlert1ResURI", "rt": ["oic.r.alert"], "if":
9120 ["oic.if.r", "oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
9121                 {"href": "/myAlert2ResURI", "rt": ["oic.r.alert"], "if":
9122 ["oic.if.r", "oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
9123                 {"href": "/myAlert3ResURI", "rt": ["oic.r.alert"], "if":
9124 ["oic.if.r", "oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
9125                 {"href": "/myAlert4ResURI", "rt": ["oic.r.alert"], "if":
9126 ["oic.if.r", "oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]}
9127             ],
9128             "schema": { "$ref": "#/definitions/AlertCollection-11" }
9129         }
9130     }
9131 },
9132 },
9133 "/AlertCollectionResURI?if=oic.if.b" : {
9134     "get": {
9135         "description": "This Resource is a Collection containing instances of Alerts
9136 (oic.r.alert).\nThis is the response using the Batch interface.\n",
9137         "parameters": [
9138             {"$ref": "#/parameters/interface-all"}
9139         ],
9140         "responses": {
9141             "200": {
9142                 "description": "",
9143                 "x-example": [
9144                     {
9145                         "href": "/Alert1ResURI",
9146                         "rep": {
9147                             "rt": ["oic.r.alert"],
9148                             "accountid": "MyAccountID",
9149                             "category": "MyCategory",
9150                             "timestamp": "2018-02-28T08:00:00Z",
9151                             "originatorid": "MyOriginatorID",
9152                             "severity": 3,
9153                             "subject": [{"language": "en-US", "value": "System error"}]}
9154                     },
9155                     {
9156                         "href": "/Alert2ResURI",
9157                         "rep": {
9158                             "rt": ["oic.r.alert"],
9159                             "accountid": "MyAccountID",
9160                             "category": "MyCategory",
9161                             "timestamp": "2018-02-28T08:15:00Z",
9162                             "originatorid": "MyOriginatorID",
9163                             "severity": 4,
9164                             "subject": [{"language": "en-US", "value": "Network error"}]}
9165                     }
9166                 ],
9167             }
9168         },
9169         "schema": { "$ref": "#/definitions/AlertCollection-b" }
9170     }
9171 },
9172 },
9173 },
9174 "/AlertCollectionResURI?if=oic.if.baseline" : {
9175     "get": {
9176         "description": "This Resource is a Collection containing instances of Alerts
9177 (oic.r.alert).\nThis is the response using the baseline interface.\n",
9178         "parameters": [
9179             {"$ref": "#/parameters/interface-all"}
9180         ],
9181         "responses": {
9182             "200": {

```

```

9183         "description" : "",
9184         "x-example": {
9185             "rt": ["oic.r.alertcollection"],
9186             "rts": ["oic.r.alert"],
9187             "if": ["oic.if.ll", "oic.if.b", "oic.if.baseline"],
9188             "links": [
9189                 {"href": "/myAlert1ResURI", "rt": ["oic.r.alert"], "if":
9190 ["oic.if.r","oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
9191                 {"href": "/myAlert2ResURI", "rt": ["oic.r.alert"], "if":
9192 ["oic.if.r","oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
9193                 {"href": "/myAlert3ResURI", "rt": ["oic.r.alert"], "if":
9194 ["oic.if.r","oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
9195                 {"href": "/myAlert4ResURI", "rt": ["oic.r.alert"], "if":
9196 ["oic.if.r","oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]}
9197             ]
9198         },
9199         "schema": { "$ref": "#/definitions/AlertCollection-baseline" }
9200     }
9201 }
9202 }
9203 }
9204 },
9205 "parameters": {
9206     "interface-all" : {
9207         "in" : "query",
9208         "name" : "if",
9209         "type" : "string",
9210         "enum" : ["oic.if.ll", "oic.if.b", "oic.if.baseline"]
9211     }
9212 },
9213 "definitions": {
9214     "AlertCollection-b" : {
9215         "type": "array",
9216         "minItems": 0,
9217         "uniqueItems": true,
9218         "items": {
9219             "type": "object",
9220             "additionalProperties": true,
9221             "properties": {
9222                 "href": {
9223                     "$ref":
9224 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9225 schema.json#/definitions/href"
9226                 },
9227                 "rep": {
9228                     "$ref":
9229 "http://openconnectivityfoundation.github.io/core/swagger2.0/oic.r.alert.swagger.json#/definitions/A
9230 lert"
9231                 }
9232             },
9233             "required": [
9234                 "href",
9235                 "rep"
9236             ]
9237         }
9238     },
9239     "AlertCollection-baseline" : {
9240         "properties": {
9241             "n": {
9242                 "$ref" :
9243 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
9244 schema.json#/definitions/n"
9245             },
9246             "id": {
9247                 "$ref" :
9248 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
9249 schema.json#/definitions/id"
9250             },
9251             "rt": {
9252                 "items": {
9253                     "type": "string",

```



```

9254         "enum": ["oic.r.alertcollection"],
9255         "maxLength": 64
9256     },
9257     "minItems": 1,
9258     "type": "array",
9259     "uniqueItems": true,
9260     "readOnly": true
9261 },
9262 "rts": {
9263     "items": {
9264         "type": "string",
9265         "enum": ["oic.r.alert"],
9266         "maxLength": 64
9267     },
9268     "minItems": 1,
9269     "type": "array",
9270     "uniqueItems": true,
9271     "readOnly": true
9272 },
9273 "if": {
9274     "description": "The OCF Interfaces supported by this Resource",
9275     "items": {
9276         "enum": [
9277             "oic.if.ll",
9278             "oic.if.b",
9279             "oic.if.baseline"
9280         ],
9281         "type": "string",
9282         "maxLength": 64
9283     },
9284     "minItems": 3,
9285     "readOnly": true,
9286     "uniqueItems": true,
9287     "type": "array"
9288 },
9289 "links": {
9290     "description": "A set of simple or individual Links.",
9291     "items": {
9292         "$ref": "#/definitions/oic.oic-link"
9293     },
9294     "type": "array"
9295 }
9296 },
9297 "type" : "object",
9298 "required": ["rt", "rts", "if", "links"]
9299 },
9300 "AlertCollection-ll" : {
9301     "type": "array",
9302     "items": {
9303         "$ref": "#/definitions/oic.oic-link"
9304     }
9305 },
9306 "oic.oic-link": {
9307     "type": "object",
9308     "properties": {
9309         "anchor": {
9310             "$ref":
9311 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9312 schema.json#/definitions/anchor"
9313         },
9314         "di": {
9315             "$ref":
9316 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9317 schema.json#/definitions/di"
9318         },
9319         "eps": {
9320             "$ref":
9321 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9322 schema.json#/definitions/eps"
9323         },
9324         "href": {

```

```

9325         "$ref":
9326         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9327         schema.json#/definitions/href"
9328     },
9329     "ins": {
9330         "$ref":
9331         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9332         schema.json#/definitions/ins"
9333     },
9334     "p": {
9335         "$ref":
9336         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9337         schema.json#/definitions/p"
9338     },
9339     "rel": {
9340         "$ref":
9341         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9342         schema.json#/definitions/rel_array"
9343     },
9344     "title": {
9345         "$ref":
9346         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9347         schema.json#/definitions/title"
9348     },
9349     "type": {
9350         "$ref":
9351         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
9352         schema.json#/definitions/type"
9353     },
9354     "if": {
9355         "description": "The OCF Interfaces supported by the target Resource",
9356         "items": {
9357             "enum": [
9358                 "oic.if.r",
9359                 "oic.if.baseline"
9360             ],
9361             "type": "string",
9362             "maxLength": 64
9363         },
9364         "minItems": 2,
9365         "uniqueItems": true,
9366         "type": "array",
9367         "readOnly": true
9368     },
9369     "rt": {
9370         "description": "Resource Type of the target Resource",
9371         "items": {
9372             "maxLength": 64,
9373             "type": "string",
9374             "enum": ["oic.r.alert"]
9375         },
9376         "minItems": 1,
9377         "type": "array",
9378         "uniqueItems": true,
9379         "readOnly": true
9380     }
9381 },
9382 "required": [
9383     "href",
9384     "rt",
9385     "if"
9386 ]
9387 }
9388 }
9389 }
9390

```

### 9391 **D.18.5 Property definition**

9392 Table D-34 defines the Properties that are part of the "oic.r.alertcollection" Resource Type.

9393  
9394

**Table D-34 – The Property definitions of the Resource with type "rt" = "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.

9395 **D.18.6 CRUDN behaviour**

9396 Table D-35 defines the CRUDN operations that are supported on the "oic.r.alertcollection"  
9397 Resource Type.

9398 **Table D-35 – The CRUDN operations of the Resource with type "rt" = "oic.r.alertcollection".**

Create	Read	Update	Delete	Notify
	get			observe

9399

## 9400 **D.19 software update**

### 9401 **D.19.1 Introduction**

9402 The Resource performing scheduled software update.

### 9403 **D.19.2 Example URI**

9404 /softwareupdateResURI

### 9405 **D.19.3 Resource type**

9406 The Resource Type is defined as: "oic.r.softwareupdate".

### 9407 **D.19.4 OpenAPI 2.0 definition**

```
9408 {
9409   "swagger": "2.0",
9410   "info": {
9411     "title": "software update",
9412     "version": "20190408",
9413     "license": {
9414       "name": "OCF Data Model License",
9415       "url":
9416         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
9417         CENSE.md",
9418       "x-copyright": "Copyright 2019 Open Connectivity Foundation, Inc. All rights reserved."
9419     },
9420     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
9421   },
9422   "schemes": ["http"],
9423   "consumes": ["application/json"],
9424   "produces": ["application/json"],
9425   "paths": {
9426     "/softwareupdateResURI" : {
9427       "get": {
9428         "description": "The Resource performing scheduled software update.",
9429         "parameters": [
9430           {"$ref": "#/parameters/interface"}
9431         ],
9432         "responses": {
9433           "200": {
9434             "description": "Schedule an software update.",
9435             "x-example":
9436               {
9437                 "rt": ["oic.r.softwareupdate"],
9438                 "if": ["oic.if.rw", "oic.if.baseline"],
9439                 "nv" : "my version",
9440                 "purl": "https://myvendor/myexampleurl",
9441                 "swupdateaction": "idle",
9442                 "swupdatestate" : "idle",
9443                 "swupdateresult" : 0,
9444                 "lastupdate" : "2015-01-09T14:30:00Z",
9445                 "signed" : "vendor",
9446                 "updatetime" : "2015-01-09T14:30:00Z"
9447               },
9448             "schema": { "$ref": "#/definitions/swupdate" }
9449           }
9450         }
9451       },
9452       "post": {
9453         "description": "Mechanism to schedule a start of the software update.",
9454         "parameters": [
9455           {"$ref": "#/parameters/interface"},
9456           {
9457             "name": "body",
9458             "in": "body",
9459             "required": true,
9460             "schema": { "$ref": "#/definitions/swupdate-update" },
9461             "x-example":
```

```

9462         {
9463             "purl": "https://myvendor/newversion",
9464             "swupdateaction": "upgrade",
9465             "updatetime" : "2030-01-09T14:30:00Z"
9466         }
9467     },
9468 ],
9469 "responses": {
9470     "200": {
9471         "description": "",
9472         "x-example":
9473             {
9474                 "nv": "my new version",
9475                 "purl": "https://myvendor/myexampleurl",
9476                 "swupdateaction": "upgrade",
9477                 "swupdatestate" : "idle",
9478                 "swupdateresult" : 0,
9479                 "lastupdate" : "2015-01-09T14:30:00Z",
9480                 "signed" : "vendor",
9481                 "updatetime" : "2030-01-09T14:30:00Z"
9482             },
9483         "schema": { "$ref": "#/definitions/swupdate" }
9484     }
9485 }
9486 }
9487 },
9488 },
9489 "parameters": {
9490     "interface": {
9491         "in": "query",
9492         "name": "if",
9493         "type": "string",
9494         "enum": ["oic.if.rw", "oic.if.baseline"]
9495     }
9496 },
9497 "definitions": {
9498     "swupdate": {
9499         "properties": {
9500             "rt": {
9501                 "items": {
9502                     "enum": [
9503                         "oic.r.softwareupdate"
9504                     ],
9505                     "type": "string",
9506                     "maxLength": 64
9507                 },
9508                 "minItems": 1,
9509                 "type": "array",
9510                 "readOnly": true,
9511                 "uniqueItems": true
9512             },
9513             "nv": {
9514                 "description": "New available Software version",
9515                 "maxLength": 64,
9516                 "type": "string",
9517                 "readOnly" : true
9518             },
9519             "purl": {
9520                 "description": "Source of the software package, might be a HTTPS or CoAPs URL",
9521                 "maxLength": 64,
9522                 "type": "string",
9523                 "format": "uri"
9524             },
9525             "swupdateaction": {
9526                 "description": "Scheduled action to do a software update",
9527                 "maxLength": 64,
9528                 "type": "string",
9529                 "enum": [
9530                     "idle",
9531                     "isac",
9532                     "isvv",

```

```

9533         "upgrade"
9534     ]
9535 },
9536 "swupdatestate": {
9537     "description": "State of the software update",
9538     "readOnly": true,
9539     "type": "string",
9540     "enum": [
9541         "idle",
9542         "nsa",
9543         "svv",
9544         "sva",
9545         "upgrading"
9546     ]
9547 },
9548 "swupdateresult": {
9549     "description": "Result of the software update, list of result codes",
9550     "readOnly": true,
9551     "type": "integer"
9552 },
9553 "lastupdate": {
9554     "description": "Time of the last software update (in RFC3339 format), Initial set on date
of manufacturing",
9555     "readOnly": true,
9556     "maxLength": 64,
9557     "type": "string",
9558     "format": "date-time"
9559 },
9560 "signed": {
9561     "description": "Signage method of the software package, currently the only allowed value
is 'vendor'.",
9562     "readOnly": true,
9563     "type": "string",
9564     "enum": [
9565         "vendor"
9566     ]
9567 },
9568 "updateaction": {
9569     "description": "Scheduled time (in RFC3339 format) to do action which is specified in
'swupdateaction' Property.",
9570     "maxLength": 64,
9571     "type": "string",
9572     "format": "date-time"
9573 },
9574 "n": {
9575     "$ref":
9576     "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
9577 },
9578 "id": {
9579     "$ref":
9580     "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
9581 },
9582 "if": {
9583     "description": "The interface set supported by this resource",
9584     "items": {
9585         "enum": [
9586             "oic.if.rw",
9587             "oic.if.baseline"
9588         ],
9589         "type": "string"
9590     },
9591     "minItems": 2,
9592     "maxItems": 2,
9593     "type": "array",
9594     "readOnly": true,
9595     "uniqueItems": true
9596 }
9597 },
9598 "required": ["purl", "swupdateaction", "swupdatestate", "swupdateresult", "updateaction"]
9599

```

```

9604     },
9605     "swupdate-update": {
9606       "properties": {
9607         "purl": {
9608           "$ref": "#/definitions/swupdate/properties/purl"
9609         },
9610         "swupdateaction" : {
9611           "$ref": "#/definitions/swupdate/properties/swupdateaction"
9612         },
9613         "updatetime": {
9614           "$ref": "#/definitions/swupdate/properties/updatetime"
9615         }
9616       },
9617       "required": ["purl", "swupdateaction", "updatetime"]
9618     }
9619   }
9620 }
9621

```

9622 **D.19.5 Property definition**

9623 Table D-36 defines the Properties that are part of the "oic.r.softwareupdate" Resource Type.

9624 **Table D-36 – The Property definitions of the Resource with type "rt" =**  
9625 **"oic.r.softwareupdate".**

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	
nv	string	No	Read Only	New available Software version
purl	string	Yes	Read Write	Source of the software package, might be a HTTPS or CoAPs URL
swupdateaction	string	Yes	Read Write	Scheduled action to do a software update
swupdatestate	string	Yes	Read Only	State of the software update
swupdateresult	integer	Yes	Read Only	Result of the software update, list of result codes
lastupdate	string	No	Read Only	Time of the last software update (in RFC3339 format), Initial set on date of manufacturing
signed	string	No	Read Only	Signage method of the software package, currently the only allowed value is 'vendor'.
updatetime	string	Yes	Read Write	Scheduled time (in RFC3339 format) to do action which is specified in 'swupdateaction' Property.
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	

if	array: see schema	No	Read Only	The interface set supported by this resource
purl	multiple types: see schema	Yes	Read Write	
swupdateaction	multiple types: see schema	Yes	Read Write	
update-time	multiple types: see schema	Yes	Read Write	

9626 **D.19.6 CRUDN behaviour**

9627 Table D-37 defines the CRUDN operations that are supported on the "oic.r.softwareupdate"  
 9628 Resource Type.

9629 **Table D-37 – The CRUDN operations of the Resource with type "rt" =**  
 9630 **"oic.r.softwareupdate".**

Create	Read	Update	Delete	Notify
	get	post		observe

9631



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": {

```

```

9680     "description": "OCF Collection Resource Type contains properties and links.\nThe
9681 oic.if.baseline interface exposes a representation of\nthe links and the properties of the
9682 collection resource itself\nRetrieve on Baseline Interface\n",
9683     "parameters": [
9684       {"$ref": "#/parameters/interface-baseline"}
9685     ],
9686     "responses": {
9687       "200": {
9688         "description": "",
9689         "x-example":
9690           {
9691             "rt": ["oic.wk.col"],
9692             "id": "unique_example_id",
9693             "rts": [ "oic.r.switch.binary", "oic.r.airflow" ],
9694             "rts-m": [ "oic.r.switch.binary" ],
9695             "links": [
9696               {
9697                 "href": "switch",
9698                 "rt": [ "oic.r.switch.binary" ],
9699                 "if": [ "oic.if.a", "oic.if.baseline" ],
9700                 "eps": [
9701                   { "ep": "coap://[fe80::b1d6]:1111", "pri": 2 },
9702                   { "ep": "coaps://[fe80::b1d6]:1122" },
9703                   { "ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3 }
9704                 ]
9705               },
9706               {
9707                 "href": "airFlow",
9708                 "rt": [ "oic.r.airflow" ],
9709                 "if": [ "oic.if.a", "oic.if.baseline" ],
9710                 "eps": [
9711                   { "ep": "coap://[fe80::b1d6]:1111", "pri": 2 },
9712                   { "ep": "coaps://[fe80::b1d6]:1122" },
9713                   { "ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3 }
9714                 ]
9715               }
9716             ]
9717           },
9718         "schema": { "$ref": "#/definitions/sbaseline" }
9719       }
9720     }
9721   },
9722   "post": {
9723     "description": "Update on Baseline Interface\n",
9724     "parameters": [
9725       {"$ref": "#/parameters/interface-baseline"},
9726       {
9727         "name": "body",
9728         "in": "body",
9729         "required": true,
9730         "schema": { "$ref": "#/definitions/sbaseline-update" }
9731       }
9732     ],
9733     "responses": {
9734       "200": {
9735         "description": "",
9736         "schema": { "$ref": "#/definitions/sbaseline-update" }
9737       }
9738     }
9739   },
9740 },
9741 "/CollectionResURI?if=oic.if.b" : {
9742   "get": {
9743     "description": "OCF Collection Resource Type contains properties and links.\nThe oic.if.b
9744 interface exposes a composite representation of the\nresources pointed to by the links\nRetrieve on
9745 Batch Interface\n",
9746     "parameters": [
9747       {"$ref": "#/parameters/interface-b"}
9748     ],
9749     "responses": {
9750       "200": {

```

```

9751         "description" : "All targets returned OK status (HTTP 200 or CoAP 2.05 Content)",
9752         "x-example":
9753             [
9754                 {
9755                     "href": "switch",
9756                     "rep":
9757                         {
9758                             "value": true
9759                         }
9760                 },
9761                 {
9762                     "href": "airFlow",
9763                     "rep":
9764                         {
9765                             "direction": "floor",
9766                             "speed":      3
9767                         }
9768                 }
9769             ],
9770         "schema": { "$ref": "#/definitions/sbatch-retrieve" }
9771     },
9772     "404": {
9773         "description" : "One or more targets did not return an OK status, return a
9774 representation containing returned properties from the targets that returned OK",
9775         "x-example":
9776             [
9777                 {
9778                     "href": "switch",
9779                     "rep":
9780                         {
9781                             "value": true
9782                         }
9783                 }
9784             ],
9785         "schema": { "$ref": "#/definitions/sbatch-retrieve" }
9786     }
9787 }
9788 },
9789 "post": {
9790     "description": "Update on Batch Interface\n",
9791     "parameters": [
9792         { "$ref": "#/parameters/interface-b" },
9793         {
9794             "name": "body",
9795             "in": "body",
9796             "required": true,
9797             "schema": { "$ref": "#/definitions/sbatch-update" },
9798             "x-example":
9799                 [
9800                     {
9801                         "href": "switch",
9802                         "rep":
9803                             {
9804                                 "value": true
9805                             }
9806                     },
9807                     {
9808                         "href": "airFlow",
9809                         "rep":
9810                             {
9811                                 "direction": "floor",
9812                                 "speed":      3
9813                             }
9814                     }
9815                 ]
9816         }
9817     ],
9818     "responses": {
9819         "200": {
9820             "description" : "all targets returned OK status (HTTP 200 or CoAP 2.04 Changed) return
9821 a representation of the current state of all targets",

```

```

9822     "x-example":
9823     [
9824         {
9825             "href": "switch",
9826             "rep":
9827             {
9828                 "value": true
9829             }
9830         },
9831         {
9832             "href": "airFlow",
9833             "rep":
9834             {
9835                 "direction": "demist",
9836                 "speed": 5
9837             }
9838         }
9839     ],
9840     "schema": { "$ref": "#/definitions/sbatch-retrieve" }
9841 },
9842 "403": {
9843     "description": "one or more targets did not return OK status; return a retrieve
9844 representation of the current state of all targets in the batch",
9845     "x-example":
9846     [
9847         {
9848             "href": "switch",
9849             "rep":
9850             {
9851                 "value": true
9852             }
9853         },
9854         {
9855             "href": "airFlow",
9856             "rep":
9857             {
9858                 "direction": "floor",
9859                 "speed": 3
9860             }
9861         }
9862     ],
9863     "schema": { "$ref": "#/definitions/sbatch-retrieve" }
9864 }
9865 }
9866 },
9867 ],
9868 "/CollectionResURI?if=oic.if.ll" : {
9869     "get": {
9870         "description": "OCF Collection Resource Type contains properties and links.\nThe oic.if.ll
9871 interface exposes a representation of the links\nRetrieve on Link List Interface\n",
9872         "parameters": [
9873             {"$ref": "#/parameters/interface-ll"}
9874         ],
9875         "responses": {
9876             "200": {
9877                 "description": "",
9878                 "x-example":
9879                 {
9880                     "links": [
9881                         {
9882                             "href": "switch",
9883                             "rt": ["oic.r.switch.binary"],
9884                             "if": ["oic.if.a", "oic.if.baseline"],
9885                             "eps": [
9886                                 {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
9887                                 {"ep": "coaps://[fe80::b1d6]:1122"},
9888                                 {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
9889                             ]
9890                         },
9891                         {
9892                             "href": "airFlow",

```

```

9893         "rt": ["oic.r.airflow"],
9894         "if": ["oic.if.a", "oic.if.baseline"],
9895         "eps": [
9896             {"ep": "coap://[fe80::bld6]:1111", "pri": 2},
9897             {"ep": "coaps://[fe80::bld6]:1122"},
9898             {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
9899         ]
9900     }
9901 }
9902 },
9903     "schema": { "$ref": "#/definitions/slinks" }
9904 }
9905 }
9906 }
9907 }
9908 },
9909 "parameters": {
9910     "interface-ll" : {
9911         "in" : "query",
9912         "name" : "if",
9913         "type" : "string",
9914         "enum" : ["oic.if.ll"]
9915     },
9916     "interface-b" : {
9917         "in" : "query",
9918         "name" : "if",
9919         "type" : "string",
9920         "enum" : ["oic.if.b"]
9921     },
9922     "interface-baseline" : {
9923         "in" : "query",
9924         "name" : "if",
9925         "type" : "string",
9926         "enum" : ["oic.if.baseline"]
9927     },
9928     "interface-all" : {
9929         "in" : "query",
9930         "name" : "if",
9931         "type" : "string",
9932         "enum" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"]
9933     }
9934 },
9935 "definitions": {
9936     "sbaseline" : {
9937         "properties": {
9938             "links" : {
9939                 "description": "A set of simple or individual OIC Links.",
9940                 "items": {
9941                     "$ref": "#/definitions/oic.oic-link"
9942                 },
9943                 "type": "array"
9944             },
9945             "n": {
9946                 "type": "string",
9947                 "description": "User friendly name of the collection"
9948             },
9949             "id": {
9950                 "anyOf": [
9951                     {
9952                         "type": "integer",
9953                         "description": "A number that is unique to that collection; like an ordinal number
9954 that is not repeated"
9955                     },
9956                     {
9957                         "type": "string",
9958                         "description": "A unique string that could be a hash or similarly unique"
9959                     },
9960                     {
9961                         "description": "An identifier formatted according to IETF RFC 4122.",
9962                         "type": "string",
9963                         "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]"

```

```

9964 9]{12}$"
9965     }
9966   ],
9967   "description": "ID for the collection. Can be an value that is unique to the use context
9968 or a UUIDv4"
9969   },
9970   "rt": {"$ref": "#/definitions/oic.core/properties/rt"},
9971   "rts": {"$ref": "#/definitions/oic.core/properties/rt"},
9972   "if": {
9973     "description": "The interface set supported by this resource",
9974     "items": {
9975       "enum": ["oic.if.baseline",
9976               "oic.if.ll",
9977               "oic.if.b",
9978               "oic.if.rw",
9979               "oic.if.r",
9980               "oic.if.a",
9981               "oic.if.s"],
9982       "type": "string"
9983     },
9984     "minItems": 1,
9985     "type": "array"
9986   }
9987 },
9988 "type" : "object"
9989 },
9990 "sbaseline-update": {
9991   "additionalProperties": true
9992 },
9993   "oic.core": {
9994     "properties": {
9995       "rt": {
9996         "description": "Resource Type of the Resource",
9997         "items": {
9998           "maxLength": 64,
9999           "type": "string"
10000         },
10001         "minItems": 1,
10002         "readOnly": true,
10003         "type": "array"
10004       }
10005     },
10006     "type": "object"
10007   },
10008   "sbatch-retrieve" : {
10009     "title" : "Collection Batch Retrieve Format (auto merged)",
10010     "minItems" : 1,
10011     "items" : {
10012       "additionalProperties": true,
10013       "properties": {
10014         "href": {
10015           "description": "URI of the target resource relative assuming the collection URI as
10016 anchor",
10017           "format": "uri",
10018           "maxLength": 256,
10019           "type": "string"
10020         },
10021         "rep": {
10022           "oneOf": [
10023             {
10024               "description": "The response payload from a single resource",
10025               "type": "object"
10026             },
10027             {
10028               "description": " The response payload from a collection (batch) resource",
10029               "items": {
10030                 "properties": {
10031                   "anchor": {
10032                     "description": "This is used to override the context URI e.g. override the URI
10033 of the containing collection.",
10034                     "format": "uri",

```

```

10035         "maxLength": 256,
10036         "type": "string"
10037     },
10038     "di": {
10039         "allOf": [
10040             {
10041                 "description": "Format pattern according to IETF RFC 4122.",
10042                 "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-
10043 [a-fA-F0-9]{12}$",
10044                 "type": "string"
10045             },
10046             {
10047                 "description": "The device ID"
10048             }
10049         ]
10050     },
10051     "eps": {
10052         "description": "the Endpoint information of the target Resource",
10053         "items": {
10054             "properties": {
10055                 "ep": {
10056                     "description": "Transport Protocol Suite + Endpoint Locator",
10057                     "format": "uri",
10058                     "type": "string"
10059                 },
10060                 "pri": {
10061                     "description": "The priority among multiple Endpoints",
10062                     "minimum": 1,
10063                     "type": "integer"
10064                 }
10065             },
10066             "type": "object"
10067         },
10068         "type": "array"
10069     },
10070     "href": {
10071         "description": "This is the target URI, it can be specified as a Relative
10072 Reference or fully-qualified URI.",
10073         "format": "uri",
10074         "maxLength": 256,
10075         "type": "string"
10076     },
10077     "if": {
10078         "description": "The interface set supported by this resource",
10079         "items": {
10080             "enum": [
10081                 "oic.if.baseline",
10082                 "oic.if.ll",
10083                 "oic.if.b",
10084                 "oic.if.rw",
10085                 "oic.if.r",
10086                 "oic.if.a",
10087                 "oic.if.s"
10088             ],
10089             "type": "string"
10090         },
10091         "minItems": 1,
10092         "type": "array"
10093     },
10094     "ins": {
10095         "description": "The instance identifier for this web link in an array of web
10096 links - used in collections",
10097         "type": "integer"
10098     },
10099     "p": {
10100         "description": "Specifies the framework policies on the Resource referenced by
10101 the target URI",
10102         "properties": {
10103             "bm": {
10104                 "description": "Specifies the framework policies on the Resource
10105 referenced by the target URI for e.g. observable and discoverable",

```

```

10106         "type": "integer"
10107     }
10108 },
10109 "required": [
10110     "bm"
10111 ],
10112 "type": "object"
10113 },
10114 "rel": {
10115     "description": "The relation of the target URI referenced by the link to the
context URI",
10116     "oneOf": [
10117         {
10118             "default": [
10119                 "hosts"
10120             ],
10121             "items": {
10122                 "maxLength": 64,
10123                 "type": "string"
10124             },
10125             "minItems": 1,
10126             "type": "array"
10127         },
10128         {
10129             "default": "hosts",
10130             "maxLength": 64,
10131             "type": "string"
10132         }
10133     ]
10134 },
10135 },
10136 "rt": {
10137     "description": "Resource Type of the Resource",
10138     "items": {
10139         "maxLength": 64,
10140         "type": "string"
10141     },
10142     "minItems": 1,
10143     "type": "array"
10144 },
10145 "title": {
10146     "description": "A title for the link relation. Can be used by the UI to
provide a context.",
10147     "maxLength": 64,
10148     "type": "string"
10149 },
10150 },
10151 "type": {
10152     "default": "application/cbor",
10153     "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.",
10154     "items": {
10155         "maxLength": 64,
10156         "type": "string"
10157     },
10158     "minItems": 1,
10159     "type": "array"
10160 }
10161 },
10162 },
10163 "required": [
10164     "href",
10165     "rt",
10166     "if"
10167 ],
10168 "type": "object"
10169 },
10170 "type": "array"
10171 }
10172 ]
10173 }
10174 },
10175 "required": [
10176     "href",

```



```

10177         "rep"
10178     ],
10179     "type": "object"
10180 },
10181 "type" : "array"
10182 },
10183 "sbatch-update" : {
10184     "title" : "Collection Batch Update Format (auto merged)",
10185     "minItems" : 1,
10186     "items" : { "$ref": "#/definitions/oic.batch-update.item" },
10187     "type" : "array"
10188 },
10189 "slinks" : {
10190     "type": "object",
10191     "properties": {
10192         "links": {
10193             "type" : "array",
10194             "items" : {
10195                 "$ref": "#/definitions/oic.oic-link"
10196             }
10197         }
10198     }
10199 },
10200 "oic.batch-update.item" : {
10201     "additionalProperties": true,
10202     "description": "array of resource representations to apply to the batch collection, using href
10203 to indicate which resource(s) in the batch to update. If the href property is empty, effectively
10204 making the URI reference to the collection itself, the representation is to be applied to all
10205 resources in the batch",
10206     "properties": {
10207         "href": {
10208             "description": "URI of the target resource relative assuming the collection URI as
10209 anchor",
10210             "format": "uri",
10211             "maxLength": 256,
10212             "type": "string"
10213         },
10214         "rep": {
10215             "oneOf": [
10216                 {
10217                     "description": "The response payload from a single resource",
10218                     "type": "object"
10219                 },
10220                 {
10221                     "description": " The response payload from a collection (batch) resource",
10222                     "items": {
10223                         "$ref": "#/definitions/oic.oic-link"
10224                     },
10225                     "type": "array"
10226                 }
10227             ]
10228         }
10229     },
10230     "required": [
10231         "href",
10232         "rep"
10233     ],
10234     "type": "object"
10235 },
10236 "oic.oic-link" : {
10237     "properties": {
10238         "anchor": {
10239             "description": "This is used to override the context URI e.g. override the URI of the
10240 containing collection.",
10241             "format": "uri",
10242             "maxLength": 256,
10243             "type": "string"
10244         },
10245         "di": {
10246             "description": "The Device ID formatted according to IETF RFC 4122.",
10247             "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]"

```

```

10248 9]{12}$",
10249     "type": "string"
10250   },
10251   "eps": {
10252     "description": "the Endpoint information of the target Resource",
10253     "items": {
10254       "properties": {
10255         "ep": {
10256           "description": "Transport Protocol Suite + Endpoint Locator",
10257           "format": "uri",
10258           "type": "string"
10259         },
10260         "pri": {
10261           "description": "The priority among multiple Endpoints",
10262           "minimum": 1,
10263           "type": "integer"
10264         }
10265       },
10266       "type": "object"
10267     },
10268     "type": "array"
10269   },
10270   "href": {
10271     "description": "This is the target URI, it can be specified as a Relative Reference or
10272 fully-qualified URI.",
10273     "format": "uri",
10274     "maxLength": 256,
10275     "type": "string"
10276   },
10277   "if": {
10278     "description": "The interface set supported by this resource",
10279     "items": {
10280       "enum": [
10281         "oic.if.baseline",
10282         "oic.if.ll",
10283         "oic.if.b",
10284         "oic.if.rw",
10285         "oic.if.r",
10286         "oic.if.a",
10287         "oic.if.s"
10288       ],
10289       "type": "string"
10290     },
10291     "minItems": 1,
10292     "type": "array"
10293   },
10294   "ins": {
10295     "description": "The instance identifier for this web link in an array of web links - used
10296 in collections",
10297     "type": "integer"
10298   },
10299   "p": {
10300     "description": "Specifies the framework policies on the Resource referenced by the target
10301 URI",
10302     "properties": {
10303       "bm": {
10304         "description": "Specifies the framework policies on the Resource referenced by the
10305 target URI for e.g. observable and discoverable",
10306         "type": "integer"
10307       }
10308     },
10309     "required": [
10310       "bm"
10311     ],
10312     "type": "object"
10313   },
10314   "rel": {
10315     "description": "The relation of the target URI referenced by the link to the context URI",
10316     "oneOf": [
10317       {
10318         "default": [

```

```

10319         "hosts"
10320     ],
10321     "items": {
10322         "maxLength": 64,
10323         "type": "string"
10324     },
10325     "minItems": 1,
10326     "type": "array"
10327 },
10328 {
10329     "default": "hosts",
10330     "maxLength": 64,
10331     "type": "string"
10332 }
10333 ]
10334 },
10335 "rt": {
10336     "description": "Resource Type of the Resource",
10337     "items": {
10338         "maxLength": 64,
10339         "type": "string"
10340     },
10341     "minItems": 1,
10342     "type": "array"
10343 },
10344 "title": {
10345     "description": "A title for the link relation. Can be used by the UI to provide a
10346 context.",
10347     "maxLength": 64,
10348     "type": "string"
10349 },
10350 "type": {
10351     "default": "application/cbor",
10352     "description": "A hint at the representation of the resource referenced by the target URI.
10353 This represents the media types that are used for both accepting and emitting.",
10354     "items": {
10355         "maxLength": 64,
10356         "type": "string"
10357     },
10358     "minItems": 1,
10359     "type": "array"
10360 }
10361 },
10362 "required": [
10363     "href",
10364     "rt",
10365     "if"
10366 ],
10367 "type": "object"
10368 }
10369 }
10370 }
10371

```

10372 **E.2.5 Property definition**

10373 Table E-2 defines the Properties that are part of the "oic.wk.col" Resource Type

10374 **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.

10375 **E.2.6 CRUDN behaviour**

10376 Table E-3 defines the CRUDN operations that are supported on the ['oic.wk.col'] Resource Type

10377 **Table E-3 – The CRUDN operations of the Resource with type 'rt' = ['oic.wk.col']**

Create	Read	Update	Delete	Notify
	get	post		observe

10378 **E.3 Discoverable Resources**

10379 **E.3.1 Introduction**

10380 List of discoverable resources.

10381 **E.3.2 Wellknown URI**

10382 /oic/res

10383 **E.3.3 Resource type**

10384 The Resource Type is defined as: "oic.wk.res"

10385 **E.3.4 OpenAPI 2.0 definition**

```

10386 {
10387   "swagger": "2.0",
10388   "info": {
10389     "title": "Discoverable Resources Link List interface",
10390     "version": "v1-20160622",
10391     "license": {
10392       "name": "copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved.",
10393       "x-description": "Redistribution and use in source and binary forms, with or without
10394 modification, are permitted provided that the following conditions are met:\n      1.
10395 Redistributions of source code must retain the above copyright notice, this list of conditions and
10396 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
10397 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
10398 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
10399 Connectivity Foundation, INC. \\"AS IS\\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
10400 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
10401 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n\n      IN NO EVENT SHALL THE Open Connectivity
10402 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
10403 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
10404 SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n\n      HOWEVER CAUSED AND ON
10405 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
10406 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

```

```

10407 SUCH DAMAGE.\n"
10408 }
10409 },
10410 "schemes": ["http"],
10411 "consumes": ["application/json"],
10412 "produces": ["application/json"],
10413 "paths": {
10414   "/oic/res?if=oic.if.ll" : {
10415     "get": {
10416       "description": "Link list representation of /oic/res; list of discoverable
10417 resources\nRetrieve the discoverable resource set, link list interface\n",
10418       "parameters": [
10419         {"$ref": "#/parameters/interface-ll"}
10420       ],
10421       "responses": {
10422         "200": {
10423           "description": "",
10424           "x-example":
10425             [
10426               {
10427                 "di": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1",
10428                 "links": [
10429                   {
10430                     "href": "/humidity",
10431                     "rt": ["oic.r.humidity"],
10432                     "if": ["oic.if.s"],
10433                     "p": {"bm": 3},
10434                     "eps": [
10435                       {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
10436                       {"ep": "coaps://[fe80::b1d6]:1122"},
10437                       {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
10438                     ]
10439                   },
10440                   {
10441                     "href": "/temperature",
10442                     "rt": ["oic.r.temperature"],
10443                     "if": ["oic.if.s"],
10444                     "p": {"bm": 3},
10445                     "eps": [
10446                       {"ep": "coaps://[[2001:db8:a::123]:2222"}
10447                     ]
10448                   }
10449                 ]
10450               }
10451             ],
10452           "schema": {"$ref": "#/definitions/slinklist"}
10453         }
10454       }
10455     }
10456   },
10457   "/oic/res?if=oic.if.baseline" : {
10458     "get": {
10459       "description": "Baseline representation of /oic/res; list of discoverable
10460 resources\nRetrieve the discoverable resource set, baseline interface\n",
10461       "parameters": [
10462         {"$ref": "#/parameters/interface-baseline"}
10463       ],
10464       "responses": {
10465         "200": {
10466           "description": "",
10467           "x-example":
10468             [
10469               {
10470                 "rt": ["oic.wk.res"],
10471                 "if": ["oic.if.baseline", "oic.if.ll"],
10472                 "di": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1",
10473                 "links": [
10474                   {
10475                     "href": "/humidity",
10476                     "rt": ["oic.r.humidity"],
10477                     "if": ["oic.if.s"],

```

```

10478         "p": { "bm": 3},
10479         "eps": [
10480             { "ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
10481             { "ep": "coaps://[fe80::b1d6]:1122"},
10482             { "ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
10483         ]
10484     },
10485     {
10486         "href": "/temperature",
10487         "rt": ["oic.r.temperature"],
10488         "if": ["oic.if.s"],
10489         "p": { "bm": 3},
10490         "eps": [
10491             { "ep": "coaps://[[2001:db8:a::123]:2222" }
10492         ]
10493     }
10494 ]
10495 }
10496 ],
10497 "schema": { "$ref": "#/definitions/sbaseline" }
10498 }
10499 }
10500 }
10501 }
10502 },
10503 "parameters": {
10504     "interface-ll" : {
10505         "in" : "query",
10506         "name" : "if",
10507         "type" : "string",
10508         "enum" : ["oic.if.ll"]
10509     },
10510     "interface-baseline" : {
10511         "in" : "query",
10512         "name" : "if",
10513         "type" : "string",
10514         "enum" : ["oic.if.baseline"]
10515     },
10516     "interface-all" : {
10517         "in" : "query",
10518         "name" : "if",
10519         "type" : "string",
10520         "enum" : ["oic.if.ll", "oic.if.baseline"]
10521     }
10522 },
10523 "definitions": {
10524     "slinklist" : {
10525         "type": "array",
10526         "items" : {
10527             "type": "object",
10528             "properties": {
10529                 "di": {
10530                     "description": "An identifier formatted according to IETF RFC 4122.",
10531                     "type": "string",
10532                     "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}$",
10533                     "readOnly": true
10534                 },
10535                 "links": {
10536                     "type": "array",
10537                     "items": {
10538                         "$ref": "#/definitions/oic.oic-link"
10539                     }
10540                 }
10541             }
10542         }
10543     },
10544     "sbaseline" : {
10545         "type": "array",
10546         "items" : {
10547             "type": "object",

```

```

10549     "properties": {
10550         "n": {
10551             "description": "Human friendly name",
10552             "maxLength": 64,
10553             "readOnly": true,
10554             "type": "string"
10555         },
10556         "rt": {
10557             "description": "Resource Type of the Resource",
10558             "items": {
10559                 "maxLength": 64,
10560                 "type": "string"
10561             },
10562             "minItems": 1,
10563             "readOnly": true,
10564             "type": "array"
10565         },
10566         "if": {
10567             "description": "The interface set supported by this resource",
10568             "items": {
10569                 "enum": [
10570                     "oic.if.baseline",
10571                     "oic.if.ll"
10572                 ],
10573                 "type": "string"
10574             },
10575             "minItems": 1,
10576             "readOnly": true,
10577             "type": "array"
10578         },
10579         "di": {
10580             "description": "An identifier formatted according to IETF RFC 4122.",
10581             "type": "string",
10582             "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
10583 9]{12}$",
10584             "readOnly": true
10585         },
10586         "mpro": {
10587             "readOnly": true,
10588             "description": "Supported messaging protocols",
10589             "type": "string",
10590             "maxLength": 64
10591         },
10592         "links": {
10593             "type": "array",
10594             "items": {
10595                 "$ref": "#/definitions/oic.oic-link"
10596             }
10597         }
10598     },
10599     "required": [
10600         "rt",
10601         "if",
10602         "links"
10603     ]
10604 },
10605 "oic.oic-link": {
10606     "type": "object",
10607     "properties": {
10608         "anchor": {
10609             "description": "This is used to override the context URI e.g. override the URI of the
10610 containing collection.",
10611             "format": "uri",
10612             "maxLength": 256,
10613             "type": "string"
10614         },
10615         "di": {
10616             "description": "The Device ID formatted according to IETF RFC 4122.",
10617             "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
10618 9]{12}$",
10619

```



```

10620         "type": "string"
10621     },
10622     "eps": {
10623         "description": "the Endpoint information of the target Resource",
10624         "items": {
10625             "properties": {
10626                 "ep": {
10627                     "description": "Transport Protocol Suite + Endpoint Locator",
10628                     "format": "uri",
10629                     "type": "string"
10630                 },
10631                 "pri": {
10632                     "description": "The priority among multiple Endpoints",
10633                     "minimum": 1,
10634                     "type": "integer"
10635                 }
10636             },
10637             "type": "object"
10638         },
10639         "type": "array"
10640     },
10641     "href": {
10642         "description": "This is the target URI, it can be specified as a Relative Reference or
10643 fully-qualified URI.",
10644         "format": "uri",
10645         "maxLength": 256,
10646         "type": "string"
10647     },
10648     "if": {
10649         "description": "The interface set supported by this resource",
10650         "items": {
10651             "enum": [
10652                 "oic.if.baseline",
10653                 "oic.if.ll",
10654                 "oic.if.b",
10655                 "oic.if.rw",
10656                 "oic.if.r",
10657                 "oic.if.a",
10658                 "oic.if.s"
10659             ],
10660             "type": "string"
10661         },
10662         "minItems": 1,
10663         "type": "array"
10664     },
10665     "ins": {
10666         "description": "The instance identifier for this web link in an array of web links - used
10667 in collections",
10668         "type": "integer"
10669     },
10670     "p": {
10671         "description": "Specifies the framework policies on the Resource referenced by the target
10672 URI",
10673         "properties": {
10674             "bm": {
10675                 "description": "Specifies the framework policies on the Resource referenced by the
10676 target URI for e.g. observable and discoverable",
10677                 "type": "integer"
10678             }
10679         },
10680         "required": [
10681             "bm"
10682         ],
10683         "type": "object"
10684     },
10685     "rel": {
10686         "description": "The relation of the target URI referenced by the link to the context URI",
10687         "oneOf": [
10688             {
10689                 "default": [
10690                     "hosts"

```

```

10691         ],
10692         "items": {
10693             "maxLength": 64,
10694             "type": "string"
10695         },
10696         "minItems": 1,
10697         "type": "array"
10698     },
10699     {
10700         "default": "hosts",
10701         "maxLength": 64,
10702         "type": "string"
10703     }
10704 ]
10705 },
10706 "rt": {
10707     "description": "Resource Type of the Resource",
10708     "items": {
10709         "maxLength": 64,
10710         "type": "string"
10711     },
10712     "minItems": 1,
10713     "type": "array"
10714 },
10715 "title": {
10716     "description": "A title for the link relation. Can be used by the UI to provide a
10717 context.",
10718     "maxLength": 64,
10719     "type": "string"
10720 },
10721 "type": {
10722     "default": "application/cbor",
10723     "description": "A hint at the representation of the resource referenced by the target URI.
10724 This represents the media types that are used for both accepting and emitting.",
10725     "items": {
10726         "maxLength": 64,
10727         "type": "string"
10728     },
10729     "minItems": 1,
10730     "type": "array"
10731 }
10732 },
10733 "required": ["href", "rt", "if"]
10734 }
10735 }
10736 }
10737

```

### 10738 E.3.5 Property definition

10739 Table E-4 defines the Properties that are part of the "oic.wk.res" Resource Type.

10740 **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

10741 **E.3.6 CRUDN behaviour**

10742 Table E-5 defines the CRUDN operations that are supported on the None Resource Type

10743 **Table E-5 – The CRUDN operations of the Resource with type 'rt' = ['oic.wk.res']**

Create	Read	Update	Delete	Notify
	get			observe

10744

10745

10746  
10747  
10748  
10749

## Annex F (informative)

### OpenAPI 2.0 Schema Extension

10750

#### F.1 OpenAPI 2.0 Schema Reference

10751 OpenAPI 2.0 does not support allOf and anyOf JSON schema validation constructs; this document  
10752 has extended the underlying OpenAPI 2.0 schema to enable these, all OpenAPI 2.0 files are valid  
10753 against the extended schema. Reference the following location for a copy of the extended schema:

10754 <https://github.com/openconnectivityfoundation/OCFswagger2.0-schema>

10755

#### F.2 OpenAPI 2.0 Introspection empty file

10756 Reference the following location for a copy of an empty OpenAPI 2.0 file:

10757 [https://github.com/openconnectivityfoundation/DeviceBuilder/blob/master/introspection-](https://github.com/openconnectivityfoundation/DeviceBuilder/blob/master/introspection-examples/introspection-empty.txt)  
10758 [examples/introspection-empty.txt](https://github.com/openconnectivityfoundation/DeviceBuilder/blob/master/introspection-examples/introspection-empty.txt)  
10759