

OCF Cloud Specification

VERSION 2.0.3 | June 2019



OPEN CONNECTIVITY
FOUNDATION™

CONTACT admin@openconnectivity.org

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

Legal Disclaimer

2
3

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

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

17 Copyright © 2018-2019 Open Connectivity Foundation, Inc. All rights reserved.

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

19

CONTENTS

20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61

1	Scope	1
2	Normative references	1
3	Terms, definitions, and abbreviated terms	2
3.1	Terms and definitions	2
3.2	Abbreviated terms	2
4	Document conventions and organization	3
4.1	Conventions	3
4.2	Notation	3
5	Overview	4
5.1	Introduction	4
5.2	Interaction Flow	4
5.3	Cloud Operational Flow	5
5.3.1	Pre-requisites and OCF Cloud User Account Creation	6
5.3.2	Mediator registration with the OCF Cloud	6
5.3.3	Device provisioning by the Mediator	6
5.3.4	Device Registration with the OCF Cloud	6
5.3.5	Connection with the OCF Cloud	7
5.3.6	Publishing Links to the OCF Cloud RD	7
5.3.7	Client to Server communication through the OCF Cloud	7
5.3.8	Refreshing connection with the OCF Cloud	7
5.3.9	Closing connection with the OCF Cloud	7
5.3.10	Deregistering from the OCF Cloud	7
6	Resource model	9
6.1	CoAPCloudConf Resource	9
6.1.1	Introduction	9
6.1.2	Resource Definition	9
6.1.3	Error Handling	10
7	Network and connectivity	11
8	Functional interactions	12
8.1	Onboarding, Provisioning, and Configuration	12
8.1.1	Overview	12
8.1.2	Use of Mediator	12
8.1.3	Device Connection to the OCF Cloud	15
8.1.4	Device Registration with the OCF Cloud	15
8.2	Resource Publication	16
8.3	Client Registration with the OCF Cloud	17
8.4	Resource Discovery	17
8.5	Device Deregistration from the OCF Cloud	19
9	Security	19
Annex A (normative)	Swagger2.0 definitions	20

62	A.1	List of Resource Type definitions	20
63	A.2	CoAP Cloud Configuration Resource	20
64	A.2.1	Introduction	20
65	A.2.2	Example URI	20
66	A.2.3	Resource type	20
67	A.2.4	OpenAPI 2.0 definition.....	20
68	A.2.5	Property definition	23
69	A.2.6	CRUDN behaviour	24
70			
71			

72
73
74
75
76
77
78
79
80
81
82
83

Figures

Figure 1 – OCF Cloud deployment architecture.....4

Figure 2 – Overall Operational State Machine9

Figure 3 – Registration with OCF Cloud 12

Figure 4 – Device Provisioning by the Mediator 14

Figure 5 – Resource publication to the OCF Cloud..... 17

Figure 6 – Resource discovery through OCF Cloud..... 18

Figure 7 – Request routing through OCF Cloud..... 19

Tables

84	
85	
86	Table 1 – OCF Cloud Deployment Flow 5
87	Table 2 – CoAPCloudConf Resource 9
88	Table 3 – oic.r.coapcloudconf Resource Type definition..... 10
89	Table 4 – Device to OCF Cloud Registration Flow..... 12
90	Table 5 – Device Provisioning by the Mediator..... 15
91	Table A.1 – Alphabetized list of resources 20
92	Table A.2 – The Property definitions of the Resource with type "rt" = "oic.r.coapcloudconf". . 23
93	Table A.3 – The CRUDN operations of the Resource with type "rt" = "oic.r.coapcloudconf"... 24
94	

95 **1 Scope**

96 This document defines functional extensions to the capabilities defined in ISO/IEC 30118-1:2018
97 to meet the requirements of the OCF Cloud. This document specifies new Resource Types to
98 enable the functionality and any extensions to the existing capabilities defined in ISO/IEC 30118-
99 1:2018.

100 **2 Normative references**

101 The following documents are referred to in the text in such a way that some or all of their content
102 constitutes requirements of this document. For dated references, only the edition cited applies. For
103 undated references, the latest edition of the referenced document (including any amendments)
104 applies.

105 ISO/IEC 30118-1:2018 *Information technology -- Open Connectivity Foundation (OCF)*
106 *Specification -- Part 1: Core specification*
107 <https://www.iso.org/standard/53238.html>
108 Latest version available at: https://openconnectivity.org/specs/OCF_Core_Specification.pdf

109 ISO/IEC 30118-2:2018 *Information technology -- Open Connectivity Foundation (OCF)*
110 *Specification -- Part 2: Security specification*
111 <https://www.iso.org/standard/74239.html>
112 Latest version available at: https://openconnectivity.org/specs/OCF_Security_Specification.pdf

113 OCF Wi-Fi Easy Setup, *Open Connectivity Foundation Wi-Fi Easy Setup, Version 2.0.1*
114 Available at:
115 https://openconnectivity.org/specs/OCF_Wi-Fi_Easy_Setup_Specification_v2.0.1.pdf
116 Latest version available at:
117 https://openconnectivity.org/specs/OCF_Wi-Fi_Easy_Setup_Specification.pdf

118 IETF RFC 6749, *The OAuth 2.0 Authorization Framework*, October 2012
119 <https://tools.ietf.org/html/rfc6749>

120 IETF RFC 6750, *The OAuth 2.0 Authorization Framework: Bearer Token Usage*, October 2012
121 <https://tools.ietf.org/html/rfc6750>

122 IETF RFC 8323, *CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets*,
123 February 2018
124 <https://tools.ietf.org/html/rfc8323>

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

127

128 **3 Terms, definitions, and abbreviated terms**

129 **3.1 Terms and definitions**

130 For the purposes of this document, the terms and definitions given in ISO/IEC 30118-1:2018 and
131 ISO/IEC 30118-2:2018 and the following apply.

132 ISO and IEC maintain terminological databases for use in standardization at the following
133 addresses:

134 – ISO Online browsing platform: available at <https://www.iso.org/obp>

135 – IEC Electropedia: available at <http://www.electropedia.org/>

136 **3.1.1**

137 **Cloud Provider**

138 entity or organization that hosts an OCF Cloud (3.1.2).

139 **3.1.2**

140 **OCF Cloud**

141 an OCF Cloud is not an OCF Device, but a logical entity that is owned by the Cloud Provider (3.1.1).

142 An OCF Cloud is authorised to communicate with a Device on behalf of the OCF Cloud User.

143 **3.2 Abbreviated terms**

144 **3.2.1**

145 **UX**

146 User Experience

147

148 **4 Document conventions and organization**

149 **4.1 Conventions**

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

154 **4.2 Notation**

155 In this document, features are described as required, recommended, allowed or DEPRECATED as
156 follows:

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

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

161 Recommended (or should)(S).

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

168 Allowed (may or allowed)(O).

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

172 DEPRECATED.

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

179 Conditionally allowed (CA)

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

182 Conditionally required (CR)

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

186

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

188 Words that are emphasized are printed in italic.

189 **5 Overview**

190 **5.1 Introduction**

191 An OCF Cloud extends the use of CoAP to enable a Device to interact with a cloud by utilizing
192 following features

- 193 – CoAP over TCP protocol defined in ISO/IEC 30118-1:2018
- 194 – Resource Directory defined in ISO/IEC 30118-1:2018
- 195 – The requirements within this document
- 196 – Security requirements and SVRs defined within the ISO/IEC 30118-2:2018

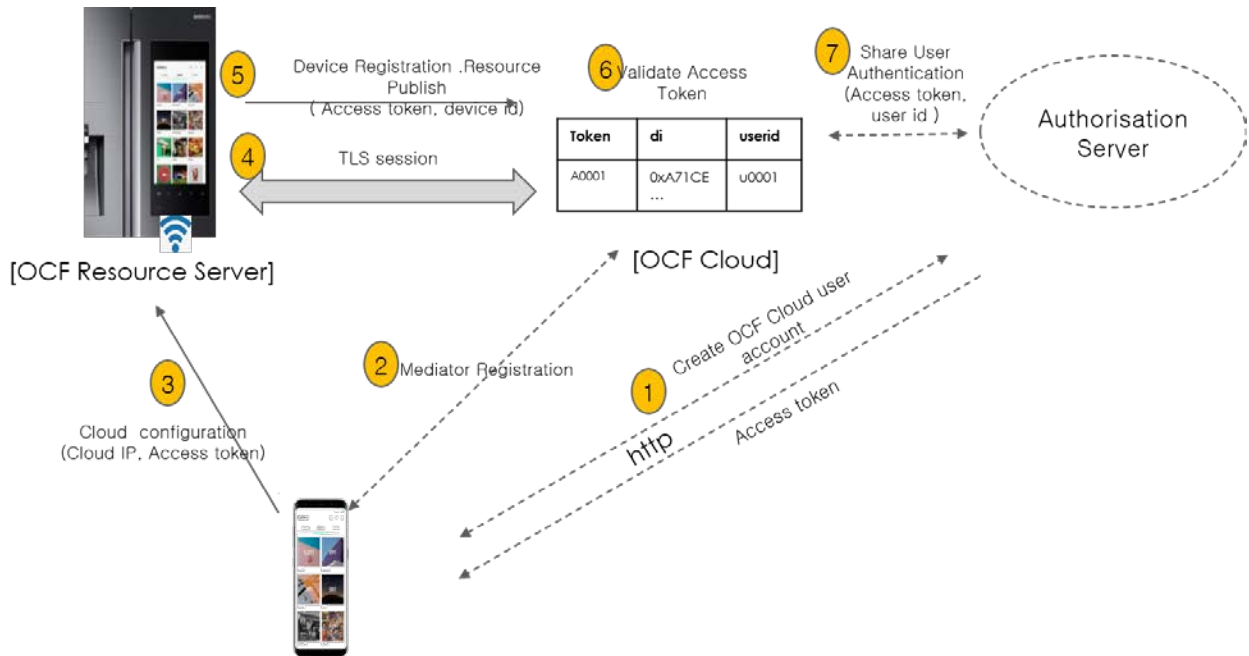
197 Devices which are not within a single local network may interact with each other using CoAP over
198 TCP (see ISO/IEC 30118-1:2018) via an OCF Cloud. At any point in time, a Device is configured
199 to use at most one OCF Cloud. The OCF Cloud groups Devices that belong to same OCF Cloud
200 User under an OCF Cloud created User ID. All the Devices registered to the OCF Cloud and
201 belonging to the same User ID can communicate with each other subject to the Device(s)
202 authorising the OCF Cloud in the ACE2 policies.

203 Annex A specifies the Resource Type definitions using the schema defined in the OpenAPI
204 specification as the API definition language that shall be followed by an OCF Device realizing the
205 Resources specified in this document.

206 Note that an OCF Cloud is not an OCF Device, but a logical entity that is owned by the Cloud
207 Provider. An OCF Cloud is authorized to communicate with a Device by the OCF Cloud User

208 **5.2 Interaction Flow**

209 This clause describes how the elements with the overall OCF Cloud interact. Figure 1 provides an
210 overall introduction, Table 1 provides additional context to the elements in the flow.



211

212

Figure 1 – OCF Cloud deployment architecture

213

214

Table 1 – OCF Cloud Deployment Flow

Steps	Description
1	The Mediator obtains an Access Token for the OCF Cloud User from an Authorisation Provider
2	The Mediator registers with the OCF Cloud
3	The Mediator provisions "oic.r.coapcloudconf" on the Device with an Access Token, the URL of the OCF Cloud, the identity (UUID) of the OCF Cloud, and optionally an Authorisation Provider Name.
4, 5	The Device establishes a TLS session to the OCF Cloud and subsequently registers with the OCF Cloud
6, 7	The OCF Cloud validates the registration request and authorises the Access Token. Returning information to the Device in the "uid" of the OCF Cloud User and the expiration information of the Access Token.

215

216 In the case where the OCF Cloud also acts as the Authorisation Server step 1 from Table 1 may
217 be between the Mediator and the OCF Cloud in which case step 7 is not required.

218 The OCF Cloud is a logical entity to which an OCF Device communicates via a persistent TLS
219 connection. It encapsulates two functions:

- 220 – an account server function which is a logical entity that handles Device registration, Access
221 Token validation and handles sign-in and token-refresh requests from the Device.
- 222 – a Resource Directory as defined by the ISO/IEC 30118-1:2018. The Resource Directory
223 exposes Resource information published by Devices. A Client, when discovering Devices,
224 receives a response from the Resource Directory on behalf of the Device. With information
225 included in the response from the Resource Directory, the Client may connect to the Device via
226 the OCF Cloud.

227 **5.3 Cloud Operational Flow**

228 The sub-clauses listed provide an informative overview of the flow which results on a Device being
229 registered with an OCF Cloud and Client interaction with that Device. The clauses provide
230 references to the applicable clauses within this document and other documents that provide
231 normative details.

232 The flow consists of the following high-level steps:

- 233 – Pre-requisites and OCF Cloud User account creation (see 5.3.1)
- 234 – Mediator registration with the OCF Cloud (see 5.3.2)
- 235 – Device provisioning by the Mediator (see 5.3.3)
- 236 – Device registration with the OCF Cloud (see 5.3.4)
- 237 – Device connection with the OCF Cloud (see 5.3.5)
- 238 – Devices Publishing Links to the OCF Cloud RD (see 5.3.6)
- 239 – Client to Server communication through the OCF Cloud (see 5.3.7)
- 240 – Device refreshing connection with the OCF Cloud (see 5.3.8)
- 241 – Device closing connection with the OCF Cloud (see 5.3.9)
- 242 – Device de-registering from the OCF Cloud (see 5.3.10)

243 **5.3.1 Pre-requisites and OCF Cloud User Account Creation**

244 The OCF Cloud User has a Device that they want to hook up to the OCF Cloud so that they can
245 access it remotely.

246 The Device is onboarded to the OCF Network as defined in ISO/IEC 30118-2:2018.

247 The OCF Cloud User downloads a Mediator onto their personal device (e.g. phone) which will be
248 used to provision the Device. The Mediator is configured with or through some out of band process
249 to obtain the URL of the OCF Cloud (e.g. the Mediator may be an application from the Cloud
250 Provider).

251 The OCF Cloud User has access credentials for authenticating the OCF Cloud User to the
252 Authorisation Provider (i.e. user name/password or similar)

253 **5.3.2 Mediator registration with the OCF Cloud**

254 See 8.1.2.2, 8.1.2.3.

255 Via some trigger (e.g. a UX or other out of bounds mechanism), the Mediator authenticates the
256 OCF Cloud User to the Authorisation Provider and requests Access Token from an Authorisation
257 Provider.

258 The Mediator registers by providing its Access Token to the OCF Cloud which verifies the token
259 and creates a User ID with which the Mediator is associated. All instances of a Mediator for the
260 same OCF Cloud User will be associated with the same User ID. Similarly, this same User ID may
261 be used to assign multiple Devices to the same OCF Cloud User

262 **5.3.3 Device provisioning by the Mediator**

263 See 8.1.2.3; see also ISO/IEC 30118-2:2018 clause 7.5.2

264 The Mediator connects to the Device through normal OCF processes. The Mediator then requests
265 an Access Token from the OCF Cloud for the Device being provisioned. The Mediator updates the
266 "oic.r.coapcloudconf" Resource on the Device with the Access Token received from the OCF Cloud,
267 the OCF Cloud URI, and the OCF Cloud UUID. The Mediator may also provide the Auth Provider
268 Name. Note that this Access Token may only be used one time for the initial Device Registration
269 with the OCF Cloud.

270 **5.3.4 Device Registration with the OCF Cloud.**

271 See 8.1.3 and 8.1.4; see also ISO/IEC 30118-2:2018 clauses 10.5, 13.11, 13.12

272 On configuration of the "oic.r.coapcloudconf" Resource by the Mediator, the Device establishes a
273 TLS connection with the OCF Cloud using the URI that was provisioned, and the Device's
274 manufacturer certificate and the trust anchor certificate(s) for OCF Cloud certificate validation, both
275 of which were installed by the Device manufacturer. The combination of the Device's manufacturer
276 certificate and OCF Cloud User's Access Token ensures the interactions between the OCF Cloud
277 and OCF Devices are within the OCF Cloud User's domain.

278 To register with the OCF Cloud, the Device then sends an UPDATE operation to the Account
279 Resource on the OCF Cloud which includes the Access Token that was provisioned in the
280 "oic.r.coapcloudconf" Resource. Note that the OCF Cloud maintains a unique instance of the
281 Account Resource for every Device.

282 If the UPDATE is successfully validated, then the OCF Cloud provides an UPDATE response that
283 may provide updated values for the Access Token and details on the lifetime (expiration) of that
284 Token. The OCF Cloud also includes the User ID to which the Device is associated. All values
285 returned are stored securely on the Device. The returned Access Token is not written to the
286 "oic.r.coapcloudconf" Resource.

287 The Device is now registered with the OCF Cloud.

288 **5.3.5 Connection with the OCF Cloud**

289 See 8.1.4, see also ISO/IEC 30118-2:2018 clause 13.12

290 In order to enable passing data between the Device and the OCF Cloud, the Device sends an
291 UPDATE request to the Session Resource; once validated, the OCF Cloud sends a response
292 message that includes the remaining lifetime of the associated Access Token. The Device now has
293 an active connection and can exchange data.

294 **5.3.6 Publishing Links to the OCF Cloud RD**

295 See 8.2; see also ISO/IEC 30118-2:2018 clause 10.5, ISO/IEC 30118-1:2018 clause 11.3.6.

296 Once the TLS connection has been established to the OCF Cloud the Device exposes its Resources
297 in the Resource Directory in the OCF Cloud so that they may be seen/accessed remotely.

298 **5.3.7 Client to Server communication through the OCF Cloud**

299 See 8.3, 8.4; see also ISO/IEC 30118-2:2018 clause 10.5.

300 As for a Server, Clients follow this same process and register with the OCF Cloud.

301 The OCF Cloud allows communication between all of an OCF Cloud User's Devices based on the
302 fact that they have the same User ID.

303 When the Client attempts CRUDN actions on the Links hosted by the OCF Cloud, the OCF Cloud
304 forwards those requests to the Device. The Device responds to the OCF Cloud which then proxies
305 the response to the Client (i.e. Client -> OCF Cloud -> Device -> OCF Cloud -> Client).

306 **5.3.8 Refreshing connection with the OCF Cloud**

307 See ISO/IEC 30118-2:2018 clause 13.13.

308 When (or before) the Access Token expires, the Device refreshes its token by sending an UPDATE
309 request to the Token Refresh Resource.

310 **5.3.9 Closing connection with the OCF Cloud**

311 See ISO/IEC 30118-2:2018 clause 13.12.

312 To log out of the OCF Cloud the Device sends an UPDATE request to the Session Resource
313 indicating a "login" status of "false". This does not delete or remove any of the Device Registration
314 information. The Device may log back into the OCF Cloud at any point prior to expiration of the
315 Access Token.

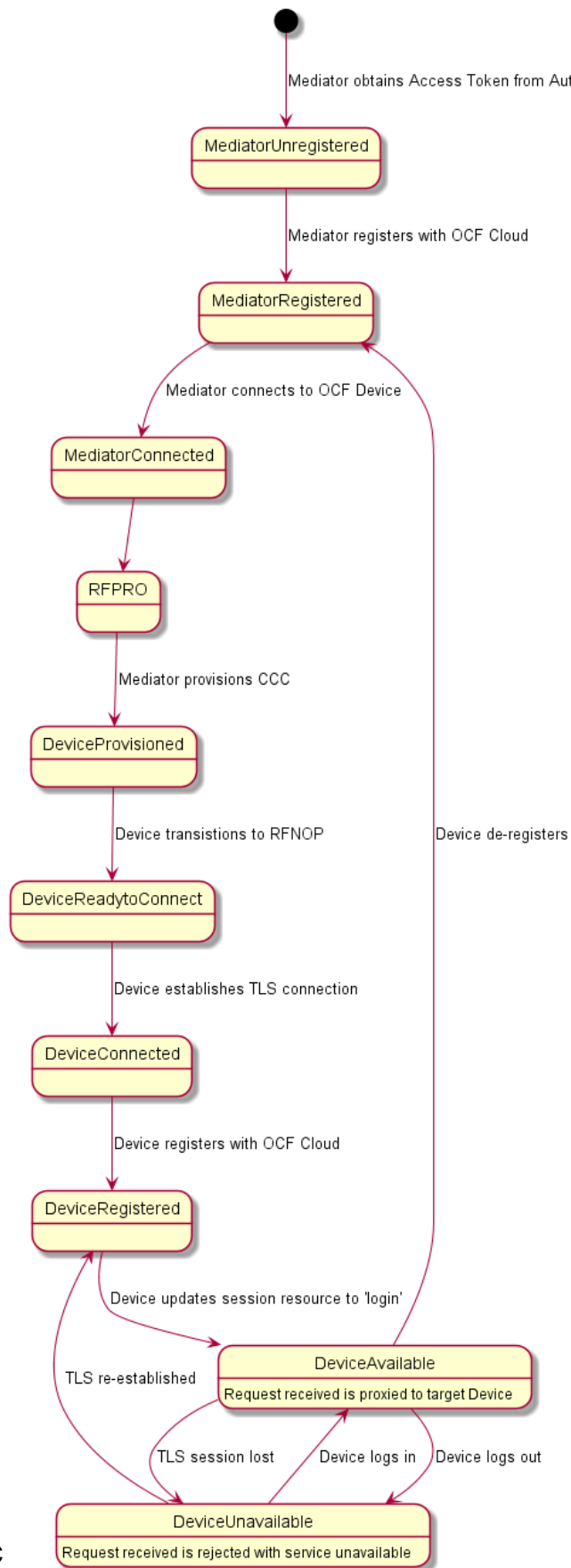
316 **5.3.10 Deregistering from the OCF Cloud**

317 See 8.5; see also ISO/IEC 30118-2:2018 clause 13.10.

318 To deregister with the OCF Cloud, the Device sends a DELETE request message to the Account
319 Resource including its Access Token. The OCF Cloud sends a response message confirming that
320 the Device has been deregistered.

321 To connect to the OCF Cloud again, the Device has to re-follow the flow starting with Mediator
322 provisioning (see 5.3.3).

323 Figure 2 captures the state machine that is described by the informative operation flow provided in
324 5.3.



326

Figure 2 – Overall Operational State Machine

327

6 Resource model

328

6.1 CoAPCloudConf Resource

329

6.1.1 Introduction

330

The CoAPCloudConf resource exposes configuration information for connecting to an OCF Cloud. This is an optional discoverable Resource, which may additionally be included within the Easy Setup Collection ("oic.r.easysetup") and so used during the Easy Setup process as defined in OCF Wi-Fi Easy Setup.

334

The CoAPCloudConf Resource shall expose only secure Endpoints (e.g. CoAPS); see the ISO/IEC 30118-1:2018, clause 10.

335

336

6.1.2 Resource Definition

337

The CoAPCloudConf Resource is as defined in Table 2.

338

Table 2 – CoAPCloudConf Resource

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
"/example/CoapCloudConfResURI"	CoAPCloudConf	"oic.r.coapcloudconf"	"oic.if.rw", "oic.if.baseline"	Configuration information for connecting to an OCF Cloud. The Resource properties exposed are listed in Table 3.	N/A

339

340

341 Table 3 defines the details for the "oic.r.coapcloudconf" Resource Type.

342 **Table 3 – oic.r.coapcloudconf Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Auth Provider Name	apn	String	N/A	N/A	RW	No	The name of the Authorisation Provider through which access token was obtained.
OCF Cloud interface URL	cis	String	uri	N/A	RW	Yes	URL of OCF Cloud.
Access Token	at	String	The Access Token is a string of at least one character	N/A	W ¹	Yes (in an UPDATE only)	Access token which is returned by an Authorisation Provider or OCF Cloud.
OCF Cloud UUID	sid	uuid	N/A	N/A	RW	Yes	The identity of the OCF Cloud
Last Error Code during Cloud Provisioning	clec	integer	enum	N/A	R	No	0: No Error, 1: Error response from the OCF Cloud, 2: Failed to connect to the OCF Cloud, 3: Failed to refresh Access Token, 4~254: Reserved, 255: Unknown error

¹ The Access Token is not included in a RETRIEVE response payload. It can only be the target of an UPDATE.

343

344 If the "clec" Property is implemented by a Device it shall have an initial value of 0 ("No error").

345 **6.1.3 Error Handling**

346 The "clec" Property of the CoAPCloudConf Resource (i.e. "oic.r.coapcloudconf") is used to indicate
 347 any error that occurred in the cloud configuration process while trying to connect to the OCF Cloud
 348 (using the information populated by the Mediator in the CoAPCloudConf Resource). This is an
 349 optional Property and if implemented, is set by the Device:

- 350 – The Device shall set the "clec" Property to 1 if it receives an error response from the OCF Cloud
 351 (e.g. error response from the Cloud).
- 352 – The Device shall set the "clec" Property to 2 if there is a failure to connect to the OCF Cloud
 353 (e.g. no reply, timeout, or timeout).
- 354 – The Device shall set the "clec" Property to 3 if it fails to refresh the Access Token (e.g. if it
 355 receives an error response during the token refresh procedure).

356 **7 Network and connectivity**

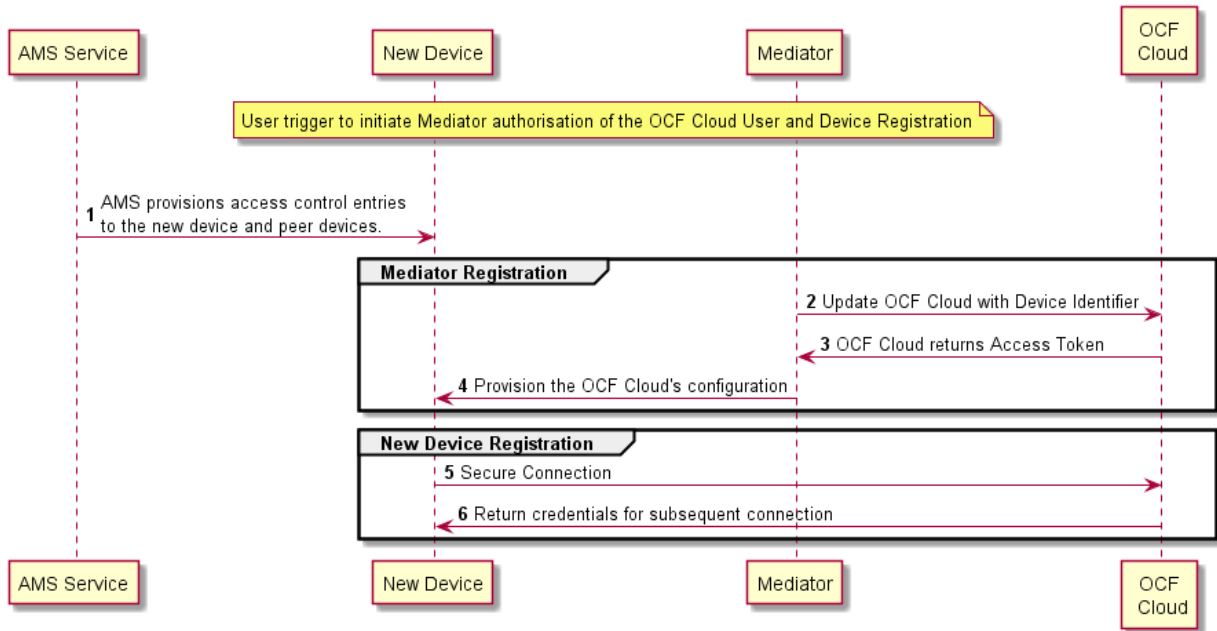
357 A TLS session exists between a Device and the OCF Cloud as specified in IETF RFC 8323; this is
358 established following device configuration as detailed in 8.1.2.3.

359 **8 Functional interactions**

360 **8.1 Onboarding, Provisioning, and Configuration**

361 **8.1.1 Overview**

362 Figure 3 provides an overview of the interaction between the different entities to get the Device
 363 registered with the OCF Cloud. A summary of the flow is provided in Table 4.



364 **Figure 3 – Registration with OCF Cloud**

365 **Table 4 – Device to OCF Cloud Registration Flow**

366

367

Steps	Description
1	AMS provisions access control entries to the new device and peer devices.
2-3	Mediator obtains the OCF Cloud User's information and authorisation.
4	Mediator provisions the credentials for the Device to connect to the OCF Cloud
5-6	Device connects to the OCF Cloud using manufacturer certificate. The OCF Cloud returns credentials to the Device, used for subsequent connection to the OCF Cloud.

368

369 **8.1.2 Use of Mediator**

370 **8.1.2.1 Introduction**

371 The Mediator is a specialised service that is used for provisioning the "oic.r.coapcloudconf"
 372 Resource, and enabling connection of a headless Device to an OCF Cloud. The Mediator is
 373 specified in OCF Wi-Fi Easy Setup.

374 The Mediator is implemented as part of the OBT (Onboarding Tool); and so could be part of any
 375 Device that itself hosts an OBT. A Device is authorized to communicate with an OCF Cloud if a
 376 trusted Mediator has provisioned the Device. The Device and Mediator connect over DTLS using
 377 credentials from "/oic/sec/cred".

378 As part of Device provisioning, the Mediator sets the following information in the
379 "oic.r.coapcloudconf" Resource exposed by the Device:

- 380 – OCF Cloud Interface URL ("cis") Property
- 381 – OCF Cloud UUID ("sid") Property (to verify Cloud identity)
- 382 – Access Token ("at") Property that is validated by the OCF Cloud
- 383 – Optionally the Authorisation Provider name ("apn") Property through which the Access Token
384 was obtained

385 If an error occurs during the process of registering and authenticating a Device with the OCF Cloud
386 the Mediator may RETRIEVE the "clec" Property if implemented by the "oic.r.coapcloudconf"
387 Resource on the Device to obtain a hint as to the cause of the error.

388 **8.1.2.2 OCF Cloud User Authorisation of the Mediator**

389 The Mediator uses a user authorisation mechanism to enable the OCF Cloud to validate the OCF
390 Cloud User's authorisation and obtain the OCF Cloud User's identity. The Authorisation Provider
391 should be trusted by both the OCF Cloud User and the OCF Cloud. The Mediator may use OAUTH
392 2.0 (see IETF RFC 6749) or another user authentication mechanism to obtain an Access Token as
393 a form of authorisation from an OCF Cloud User via an Authorisation Provider. This authorisation
394 achieves a variety of purposes. Firstly, the authorisation shows OCF Cloud User consent for
395 Mediator to connect to the OCF Cloud. Secondly, the authorisation is used to obtain information to
396 map the Devices to the same OCF Cloud User.

397 A user authorisation mechanism is used to achieve the following:

- 398 – Obtain an Access Token that is validated by the Cloud
- 399 – OCF Cloud User authorisation via an Authorisation Provider; this provides consent to connect
400 to the OCF Cloud.

401 If a different Mediator is used by the same OCF Cloud User, a new Access Token may be obtained
402 from an Authorisation Provider. Mediator Registration with the OCF Cloud

403 The Mediator connects to the OCF Cloud using a provisioned certificate on the Mediator to establish
404 a TLS connection.

405 On its first connection, the Mediator starts the registration process with the OCF Cloud. The
406 Mediator provides the OCF Cloud with the Mediator's Access Token received from the Authorisation
407 Provider in 8.1.2.2 in order to register with the OCF Cloud.

408 The OCF Cloud then verifies the Access Token with the Authorisation Provider. If the Authorisation
409 Provider validates the Access Token successfully, then it will return information about the OCF
410 Cloud User to whom the Access Token belongs. The OCF Cloud generates a unique Access Token
411 for the Mediator (which may be the original Access Token from the Mediator or a new Access Token)
412 and a User ID (i.e. "uid" Property of "oic.r.account") if this is the first instance of registering a
413 Mediator with this OCF Cloud User. The User ID acts as a unique identity for the OCF Cloud User.
414 All instances of a Mediator for the same OCF Cloud User will be associated with the same User ID.
415 This information is returned to the Mediator over TLS. The returned Access Token and User ID are
416 used by the OCF Cloud to identify the Mediator. This returned Access Token is used by the
417 Mediator in subsequent interactions with the OCF Cloud.

418 All Devices registering with the OCF Cloud receive the same User ID from the OCF Cloud when
419 registering with the same Mediator.

420 **8.1.2.3 Device Provisioning by the Mediator**

421 The Mediator obtains the OCF Cloud User's permission before the Mediator and OCF Cloud interact to
422 preregister the Device with the OCF Cloud. This clause provides an informative description of
423 the expected subsequent exchange between a Mediator and an OCF Cloud.

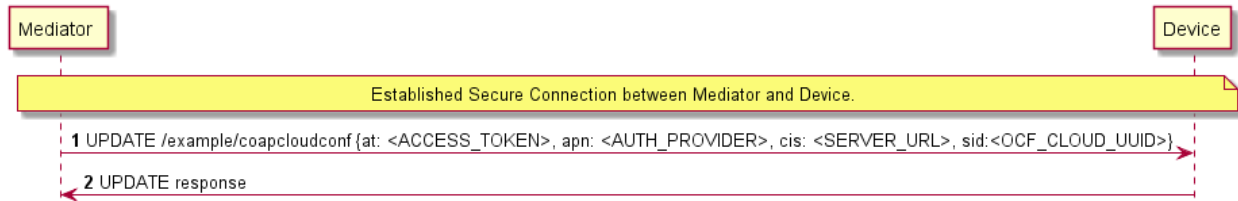
424 Once the OCF Cloud has associated the Mediator with a User ID, the Mediator can request the
425 OCF Cloud to associate OCF Devices with the same User ID. To register the Device with the OCF
426 Cloud, the Mediator first requests an Access Token for the Device from the OCF Cloud. The
427 Mediator may provide the following information to the OCF Cloud to obtain an Access Token for
428 the Device:

- 429 – Device ID (i.e. "di" Property Value of "/oic/d" of the Device)

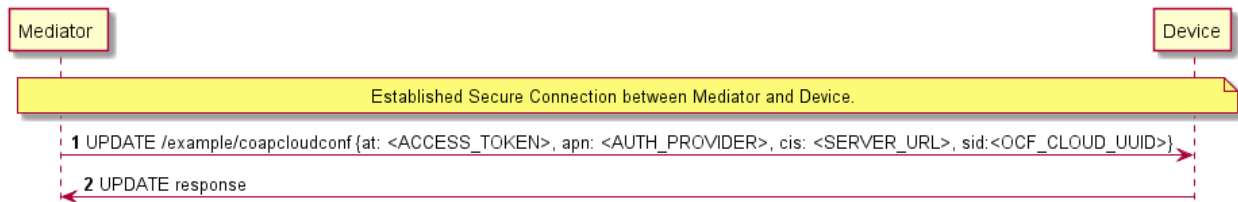
430 The OCF Cloud then returns a unique Access Token for the Device. The OCF Cloud maintains a
431 map where Access Token and Mediator-provided Device ID are stored. At the time of Device
432 Registration OCF Cloud validates the Access Token and associates the TLS session with
433 corresponding Device ID. The OCF Cloud may also return an Authorisation Provider Name
434 associated with the Access Token if the Access Token for the Device was created by an entity
435 other than the OCF Cloud.

436 The Mediator provides this Access Token to the Device ("at" Property) via an UPDATE to the
437 Device's "oic.r.coapcloudconf" Resource. The provisioned Access Token is to be treated by Device
438 as an Access Token with "Bearer" token type as defined in IETF RFC 6750. The Mediator also
439 provisions the OCF Cloud URI ("cis" Property), where the OCF Cloud URI can be either pre-
440 configured or provided to the Mediator via OCF Cloud User input. The Mediator further provisions
441 the OCF Cloud UUD ("sid" Property) to the identity of the OCF Cloud. If the OCF Cloud also
442 returned an Authorisation Provider Name in association with the Access Token for the Device then
443 this is also provisioned by the Mediator on the Device ("apn" Property of "oic.r.coapcloudconf").

444 See ISO/IEC 30118-2:2018 clause 7.5.2 for details on the population of ACE2 entries on the Device
445 to allow CRUDN operations from the Mediator and OCF Cloud.



446
447 Figure 4 describes the flow for provisioning of the Device by a Mediator. Table 5 provides additional
448 context around the flow.



449
450 **Figure 4 – Device Provisioning by the Mediator**

451

452

Table 5 – Device Provisioning by the Mediator

Steps	Description
1 - 2	Mediator updates the "oic.r.coapcloudconf" Resource on the Device with configuration information to enable the Device to connect to the OCF Cloud

453

454 Please see ISO/IEC 30118-2:2018 clause 7.5.2 for further details on the mapping of Properties
455 between the Device and OCF Cloud.

456 **8.1.3 Device Connection to the OCF Cloud**

457 On conclusion of Device provisioning as defined in 8.1.2.3 and after transitioning to a state of
458 RFNOP (if not already in RFNOP) the Device shall establish a TLS connection with the OCF Cloud
459 as defined in the ISO/IEC 30118-2:2018 clause 10.5. Further see the ISO/IEC 30118-2:2018 clause
460 10.5.3 for additional security considerations.

461 If authentication of the TLS session being established as defined in the ISO/IEC 30118-2:2018 fails,
462 the "clec" Property of the "oic.r.coapcloudconf" Resource on the Device (if supported) shall be
463 updated about the failed state. If authentication succeeds, the Device and OCF Cloud establish an
464 encrypted link in accordance with the negotiated cipher suite. Further, if the TLS connection is lost
465 due to a failure the "clec" Property of the "oic.r.coapcloudconf" Resource on the Device (if
466 supported) should be updated about the failed state (value of "2").

467 If the TLS connection is lost either via a failure or closed by the OCF Cloud then it may be re-
468 established by following the procedures in the ISO/IEC 30118-2:2018 clause 10.5. A Device may
469 automatically attempt to re-establish the TLS connection, alternatively a Device may require some
470 user trigger to initiate the re-establishment of the TLS connection.

471 **8.1.4 Device Registration with the OCF Cloud**

472 The OCF Cloud maintains a map of User IDs ("uid" Property of "oic.r.account"), Device IDs ("di"
473 Property of "oic.r.account") and Access Tokens ("accesstoken" Property of "oic.r.account";
474 populated with the same value as the "at" Property obtained from "oic.r.coapcloudconf") to
475 authenticate Devices connecting to the OCF Cloud.

476 After the TLS connection is established with the OCF Cloud, the Device shall register with the OCF
477 Cloud by sending an UPDATE request to "/oic/sec/account" as defined in clause 13.10 of the
478 ISO/IEC 30118-2:2018. The OCF Cloud consequently associates the TLS connection with the
479 corresponding "uid" and "di" Properties populated in the "/oic/sec/account/" Resource. Any other
480 Device registering with the OCF Cloud is assigned the same User ID by the OCF Cloud when
481 registering with any Mediator associated with that User ID. Device Registration permits a Client to
482 access Resources on the OCF Cloud which are associated with the same User ID as the Client.

483 If the Property values in the UPDATE to "/oic/sec/account" do not match the equivalents provided
484 to the Mediator by the OCF Cloud the OCF Cloud should close the TLS connection with the Device.
485 Note that the OCF Cloud may also apply additional out-of-band measures, for example the OCF
486 Cloud may send an email to the OCF Cloud User for additional verification to register the Device.

487 If the UPDATE operation is accepted by the OCF Cloud, the OCF Cloud responds as defined in
488 clause 13.10 of the ISO/IEC 30118-2:2018.

489 The "accesstoken" Property that is returned in the UPDATE response may be valid for limited
490 duration; in this instance the Device may use the "/oic/sec/tokenrefresh" Resource to renew the
491 "accesstoken" before the Access Token expires at the time specified in the "expiresin" Property.

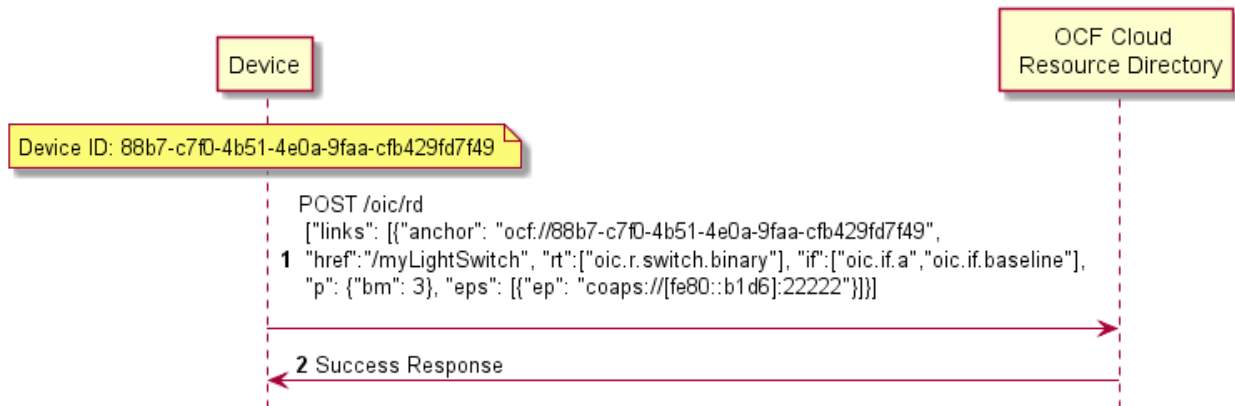
492 On completion of Device Registration the Device shall send an UPDATE to "/oic/sec/session" as
493 defined in clause 13.11 of the ISO/IEC 30118-2:2018 to ensure that the established TLS session
494 is maintained for subsequent interaction with the OCF Cloud Resource Directory as defined in
495 clause 8.2.

496 8.2 Resource Publication

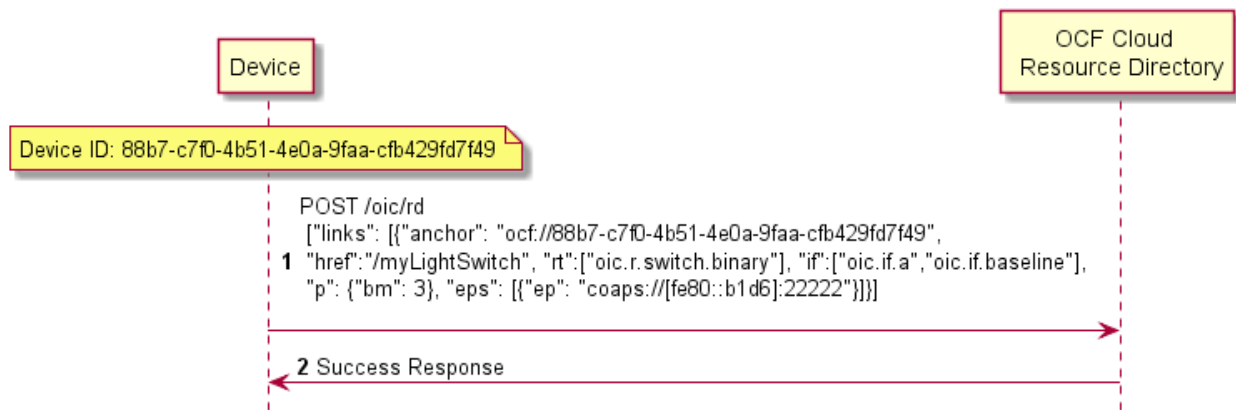
497 An OCF Cloud exposes a Resource Directory as defined in the ISO/IEC 30118-1:2018 clause
498 11.3.6. After a Device is registered with an OCF Cloud, the Device should publish its Resources to
499 the OCF Cloud's Resource Directory following the procedures defined in the ISO/IEC 30118-1:2018
500 clause 11.3.6. The Device and OCF Cloud maintain a persistent TLS connection over which
501 requests received by the OCF Cloud for the Device are routed.

502 The OCF Cloud maintains an internal association between the published Endpoint information from
503 the Device and the Endpoint information that it (the OCF Cloud) exposes in the Links within the
504 OCF Cloud's Resource Directory. The Endpoint exposed by the OCF Cloud for all Resources
505 published to it is that of the OCF Cloud itself and not the publishing Device. These Endpoints use
506 a scheme of "coaps+tcp". The Links within the OCF Cloud's Resource Directory are only identified
507 per the OCF Cloud User Account (User ID). For example, the registered Links are only returned to
508 Client under same User ID with a Server, and not returned to any other Client under a different
509 User ID with the Server.

510 There is potential ambiguity where different instances of Devices from the same vendor (e.g.
511 multiple lights) publish their Resources; this is because the local "href" Link Parameter that is
512 provided to the RD is likely to be the same in each case. In order to avoid this ambiguity the
513 Resource Directory shall prepend the "href" that is published with the Device ID for the publishing
514 Device. Thus ensuring that all requests received by the OCF Cloud have a unique URI per
515 published Resource.



516
517 Figure 5 provides an example showing the provided Device ID from the Device; Figure 6 shows the
518 pre-pending of the Device ID to the "href" Link Parameter in the Resource Directory itself.



519
520 **Figure 5 – Resource publication to the OCF Cloud**

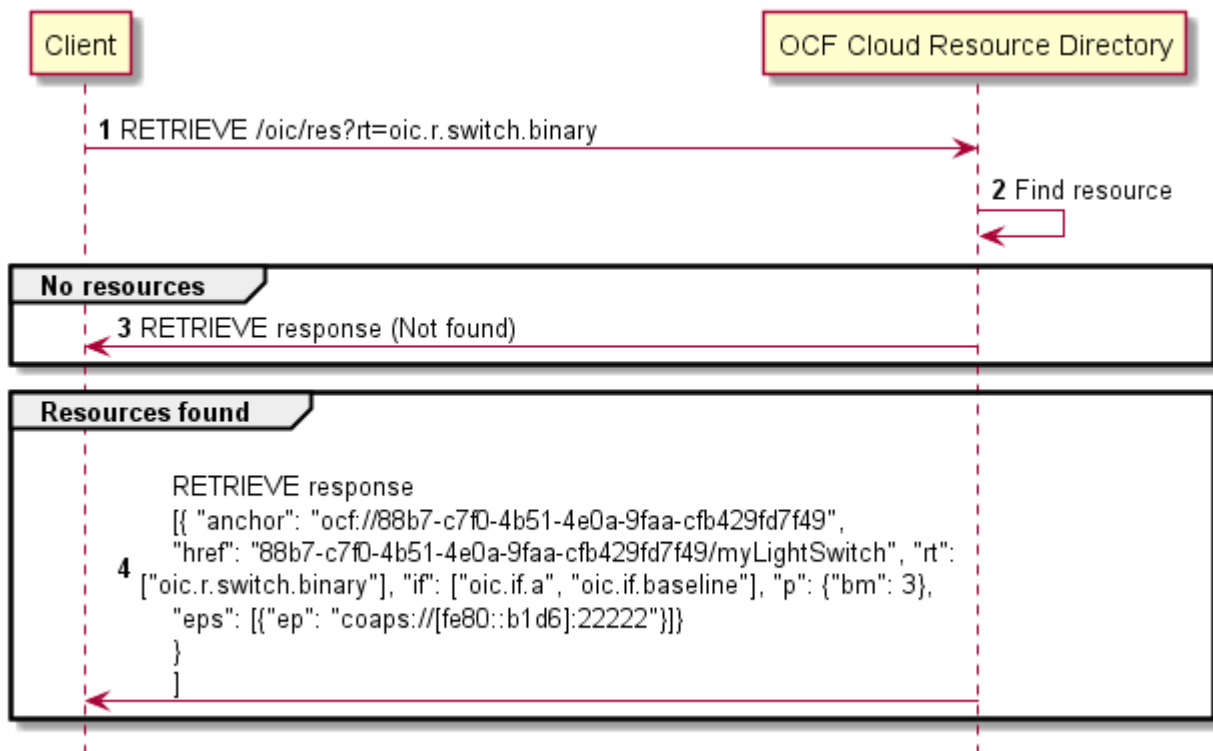
521 **8.3 Client Registration with the OCF Cloud**

522 A Device acting in the Client role follows the same procedures as a Device in the Server role
523 registering with the OCF Cloud. This Client is associated with a User ID in the same manner in
524 which a Server is associated with the same User ID

525 **8.4 Resource Discovery**

526 A remote Device may query "/oic/res" to discover Resources published to the OCF Cloud. The OCF
527 Cloud's Resource Directory responds with Links for the Resources published to the OCF Cloud by
528 Devices that are registered to the OCF Cloud for the User ID with which the remote Device is
529 associated. The "eps" Link Parameter in the "/oic/res" response are for the OCF Cloud and not the
530 publishing Device.

531 Figure 6 provides an illustrative flow for Resource Discovery, note the population of the 'href' for
532 instance of "oic.r.switch.binary" including the Device ID of the target Device in accordance with 8.2:



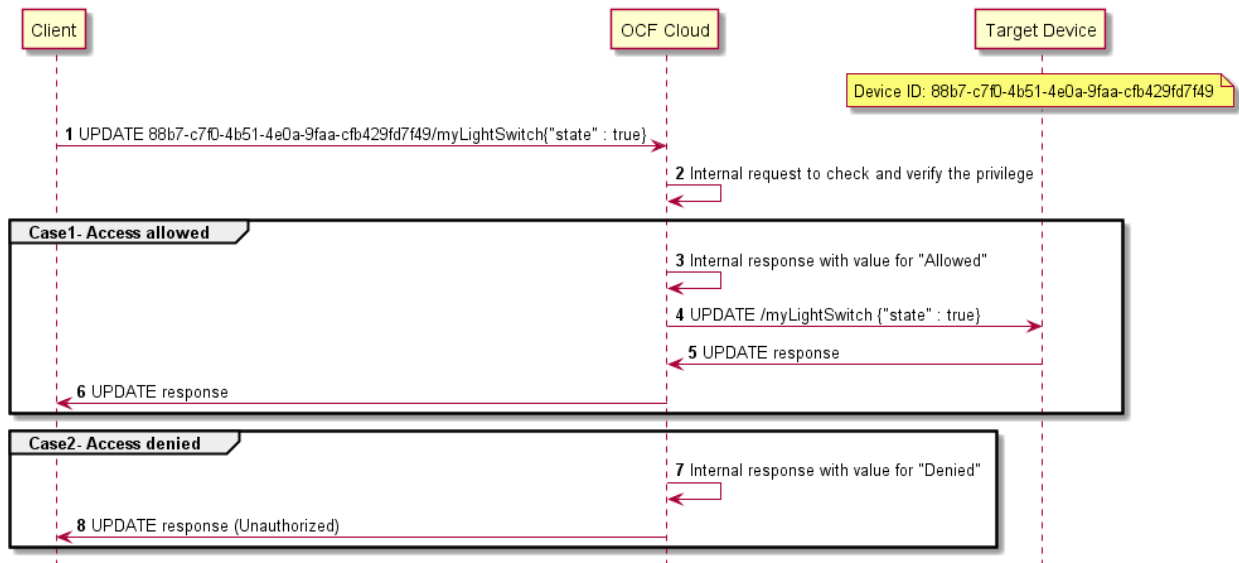
533

534

Figure 6 – Resource discovery through OCF Cloud

535 The OCF Cloud acts as a simple proxy, forwarding the messages to the publishing Devices. The
 536 remote Device sends a RETRIEVE to the OCF Cloud to obtain the content of the Server's published
 537 Resources, the OCF Cloud will route the message to the target Device after first removing the
 538 Device ID that had been prepended to the 'href' Link Parameter by the Cloud RD. Similarly, other
 539 CRUDN operations originated by a Client are routed to the Server via the OCF Cloud. The
 540 publishing Device treats the forwarded request message as a request from the OCF Cloud. The
 541 publishing Device authorises the request as specified in ISO/IEC 30118-2:2018, using the UUID of
 542 the OCF Cloud configured in the "sid" Property of "oic.r.coapcloudconf". The publishing Device
 543 sends a response message to the OCF Cloud, and the OCF Cloud forwards the response to the
 544 Client which sent the corresponding request.

545 Figure 7 illustrates request routing via the OCF Cloud



546

547

Figure 7 – Request routing through OCF Cloud

548 If it is not possible for whatever reason for the OCF Cloud to route a Client request to the Server
 549 that OCF Cloud may reject the request with a final response (e.g. "Service Unavailable").

550 **8.5 Device Deregistration from the OCF Cloud**

551 To deregister from the OCF Cloud the Device first sends a DELETE operation to the
 552 "/oic/sec/account" Resource as defined in the ISO/IEC 30118-2:2018 clause 13.11.

553 Upon completion of deregistration of the Device the OCF Cloud deletes the links for the
 554 deregistered Device from the Resource Directory that is exposed by the OCF Cloud.

555 **9 Security**

556 OCF Cloud shall follow the security requirements captured in the ISO/IEC 30118-2:2018.

557

Annex A (normative)

Swagger2.0 definitions

A.1 List of Resource Type definitions

Table A.1 contains the list of defined resources in this document.

Table A.1 – Alphabetized list of resources

Friendly Name (informative)	Resource Type (rt)	Clause
CoAP Cloud Configuration	"oic.r.coapcloudconf"	A.2

A.2 CoAP Cloud Configuration Resource

A.2.1 Introduction

The CoAPCloudConf Resource exposes configuration information for connecting to an OCF Cloud.

A.2.2 Example URI

/CoAPCloudConfResURI

A.2.3 Resource type

The Resource Type is defined as: "oic.r.coapcloudconf".

A.2.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "CoAP Cloud Configuration Resource",
    "version": "20190327",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.md",
      "x-copyright": "Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/CoAPCloudConfResURI?if=oic.if.rw" : {
      "get": {
        "description": "The CoAPCloudConf Resource exposes configuration information for connecting
to an OCF Cloud.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-all"}
        ],
        "responses": {
          "200": {
            "description": "",
            "x-example":
            {
              "rt" : ["oic.r.coapcloudconf"],
              "apn": "github",
              "cis": "coaps+tcp://example.com:443",
              "sid" : "987e6543-a21f-10d1-a112-421345746237",
              "clec": 0
            }
          }
        }
      }
    }
  }
}
```

```

608         },
609         "schema": { "$ref": "#/definitions/CoAPCloudConf" }
610     }
611 }
612 },
613 "post": {
614     "description": "Update properties of the CoAPCloudConf Resource.\n",
615     "parameters": [
616         { "$ref": "#/parameters/interface-all" },
617         {
618             "name": "body",
619             "in": "body",
620             "required": true,
621             "schema": { "$ref": "#/definitions/CoAPCloudConfUpdate" },
622             "x-example":
623                 {
624                     "at": "0f3d9f7fe5491d54077d",
625                     "apn": "github",
626                     "cis": "coaps+tcp://example.com:443",
627                     "sid": "987e6543-a21f-10d1-a112-421345746237"
628                 }
629         }
630     ],
631     "responses": {
632         "200": {
633             "description": "",
634             "x-example":
635                 {
636                     "apn": "github",
637                     "cis": "coaps+tcp://example.com:443",
638                     "sid": "987e6543-a21f-10d1-a112-421345746237",
639                     "clec": 0
640                 },
641             "schema": { "$ref": "#/definitions/CoAPCloudConf" }
642         }
643     }
644 }
645 },
646 "/CoAPCloudConfResURI?if=oic.if.baseline": {
647     "get": {
648         "description": "The CoAPCloudConf Resource exposes configuration information for connecting
649 to an OCF Cloud.\n",
650         "parameters": [
651             { "$ref": "#/parameters/interface-all" }
652         ],
653         "responses": {
654             "200": {
655                 "description": "",
656                 "x-example":
657                     {
658                         "rt": ["oic.r.coapcloudconf"],
659                         "if": ["oic.if.rw", "oic.if.baseline"],
660                         "apn": "github",
661                         "cis": "coaps+tcp://example.com:443",
662                         "sid": "987e6543-a21f-10d1-a112-421345746237",
663                         "clec": 0
664                     },
665                 "schema": { "$ref": "#/definitions/CoAPCloudConf" }
666             }
667         }
668     },
669     "post": {
670         "description": "Update Properties of the CoAPCloudConf Resource.\n",
671         "parameters": [
672             { "$ref": "#/parameters/interface-all" },
673             {
674                 "name": "body",
675                 "in": "body",
676                 "required": true,
677                 "schema": { "$ref": "#/definitions/CoAPCloudConfUpdate" },
678                 "x-example":

```

```

679         {
680             "at": "0f3d9f7fe5491d54077d",
681             "apn": "github",
682             "cis": "coaps+tcp://example.com:443",
683             "sid" : "987e6543-a21f-10d1-a112-421345746237"
684         }
685     },
686 ],
687 "responses": {
688     "200": {
689         "description" : "",
690         "x-example":
691         {
692             "apn": "github",
693             "cis": "coaps+tcp://example.com:443",
694             "sid" : "987e6543-a21f-10d1-a112-421345746237",
695             "clec": 0
696         },
697         "schema": { "$ref": "#/definitions/CoAPCloudConf" }
698     }
699 }
700 }
701 },
702 },
703 "parameters": {
704     "interface-all" : {
705         "in" : "query",
706         "name" : "if",
707         "type" : "string",
708         "enum" : ["oic.if.rw", "oic.if.baseline"]
709     },
710 },
711 "definitions": {
712     "CoAPCloudConf" : {
713         "properties": {
714             "rt" : {
715                 "description": "Resource Type of the Resource",
716                 "items": {
717                     "enum": ["oic.r.coapcloudconf"],
718                     "type": "string",
719                     "maxLength": 64
720                 },
721                 "minItems": 1,
722                 "uniqueItems": true,
723                 "readOnly": true,
724                 "type": "array"
725             },
726             "n" : {
727                 "$ref":
728 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
729 schema.json#/definitions/n"
730             },
731             "cis" : {
732                 "description": "URL of OCF Cloud",
733                 "format": "uri",
734                 "type": "string"
735             },
736             "apn" : {
737                 "description": "The Authorisation Provider through which an Access Token was obtained.",
738                 "type": "string"
739             },
740             "sid" : {
741                 "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
742 schema.json#/definitions/uuid"
743             },
744             "clec" : {
745                 "description": "Last Error Code during Cloud Provisioning (0: No Error, 1: Error response
746 from the OCF Cloud, 2: Failed to connect to the OCF Cloud, 3: Failed to refresh Access Token, 4~254:
747 Reserved, 255: Unknown error)",
748                 "enum": [
749                     0,

```

```

750         1,
751         2,
752         3,
753         255
754     ],
755     "readOnly": true
756 },
757     "id" : {
758         "$ref":
759 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
760 schema.json#/definitions/id"
761     },
762     "if" : {
763         "description": "The OCF Interfaces supported by this Resource",
764         "items": {
765             "enum": [
766                 "oic.if.rw",
767                 "oic.if.baseline"
768             ],
769             "type": "string",
770             "maxLength": 64
771         },
772         "minItems": 2,
773         "uniqueItems": true,
774         "readOnly": true,
775         "type": "array"
776     }
777 },
778 "type" : "object",
779 "required":["cis", "sid"]
780 },
781 "CoAPCloudConfUpdate" : {
782     "properties": {
783         "cis" : {
784             "description": "URL of OCF Cloud",
785             "format": "uri",
786             "type": "string"
787         },
788         "apn" : {
789             "description": "The Authorisation Provider through which an Access Token was obtained.",
790             "type": "string"
791         },
792         "at" : {
793             "description": "Access Token which is returned by an Authorisation Provider or OCF
794 Cloud.",
795             "type": "string"
796         },
797         "sid" : {
798             "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
799 schema.json#/definitions/uuid"
800         }
801     },
802     "type" : "object",
803     "required":["cis", "at", "sid"]
804 }
805 }
806 }
807

```

808 A.2.5 Property definition

809 Table A.2 defines the Properties that are part of the "oic.r.coapcloudconf" Resource Type.

810 **Table A.2 – The Property definitions of the Resource with type "rt" = "oic.r.coapcloudconf".**

Property name	Value type	Mandatory	Access mode	Description
sid	multiple types: see schema	Yes	Read Write	

rt	array: see schema	No	Read Only	Resource Type of the Resource.
id	multiple types: see schema	No	Read Write	
n	multiple types: see schema	No	Read Write	
cis	string	Yes	Read Write	URL of OCF Cloud.
apn	string	No	Read Write	The Authorisation Provider through which an Access Token was obtained.
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.
clec	multiple types: see schema	No	Read Only	Last Error Code during Cloud Provisioning (0: No Error, 1: Error response from the OCF Cloud, 2: Failed to connect to the OCF Cloud, 3: Failed to refresh Access Token, 4~254: Reserved, 255: Unknown error).
sid	multiple types: see schema	Yes	Read Write	
at	string	Yes	Read Write	Access Token which is returned by an Authorisation Provider or OCF Cloud.
apn	string	No	Read Write	The Authorisation Provider through which an Access Token was obtained.
cis	string	Yes	Read Write	URL of OCF Cloud.

811 **A.2.6 CRUDN behaviour**

812 Table A.3 defines the CRUDN operations that are supported on the "oic.r.coapcloudconf" Resource
813 Type.

814 **Table A.3 – The CRUDN operations of the Resource with type "rt" = "oic.r.coapcloudconf".**

Create	Read	Update	Delete	Notify
	get	post		observe

815