

OCF Cloud Security Specification

VERSION 2.0.5 | September 2019



OPEN CONNECTIVITY
FOUNDATION™

CONTACT admin@openconnectivity.org

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

1 **LEGAL DISCLAIMER**

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

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

15 Copyright © 2017-2019 Open Connectivity Foundation, Inc. All rights reserved.

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

17	1	Purpose and Role	1
18	2	Normative References	1
19	3	Terms, definitions, and abbreviated terms	2
20	3.1	Terms and definitions.....	2
21	3.2	Abbreviated terms.....	3
22	4	Document Conventions and Organization	3
23	4.1	Conventions.....	3
24	4.2	Notation.....	3
25	4.3	Data types	4
26	4.4	Document structure.....	4
27	5	Security overview	5
28	5.1	Preamble	5
29	5.2	Device Provisioning for OCF Cloud and Device Registration Overview.....	5
30	5.3	Credential overview	5
31	6	Device provisioning for OCF Cloud	5
32	6.1	Cloud Provisioning General	5
33	6.2	Device Provisioning by Mediator	6
34	7	Device authentication with OCF Cloud.....	9
35	7.1	Device Authentication with OCF Cloud General	9
36	7.2	Device Connection with the OCF Cloud	9
37	7.3	Security Considerations	11
38	8	Message integrity and confidentiality	12
39	8.1	Cloud Session Semantics	12
40	8.2	Cipher suites for OCF Cloud Credentials	12
41	9	Security resources.....	12
42	9.1	Account Resource.....	12
43	9.2	Account Session resource.....	14
44	9.3	Account Token Refresh Resource	15
45	10	Security hardening guidelines.....	16
46	10.1	Security hardening guidelines general	16
47	Annex A (normative) Resource Type definitions		18
48	A.1	Account Token	18
49	A.1.1	Introduction	18
50	A.1.2	Well-known URI	18
51	A.1.3	Resource type	18
52	A.1.4	OpenAPI 2.0 definition.....	18
53	A.1.5	Property definition	21
54	A.1.6	CRUDN behaviour	22
55	A.2	Session.....	22
56	A.2.1	Introduction	22
57	A.2.2	Well-known URI	22
58	A.2.3	Resource type	22
59	A.2.4	OpenAPI 2.0 definition.....	22

60	A.2.5	Property definition	24
61	A.2.6	CRUDN behaviour	25
62	A.3	Token Refresh	25
63	A.3.1	Introduction	25
64	A.3.2	Well-known URI	26
65	A.3.3	Resource type	26
66	A.3.4	OpenAPI 2.0 definition.....	26
67	A.3.5	Property definition	28
68	A.3.6	CRUDN behaviour	29
69			

70	FIGURES	
71	Figure 1 – OCF Interaction.....	3
72	Figure 4 – Device connection with OCF Cloud	10
73		
74		
75	Tables	
76	Table 1 – Mapping of Properties of the "oic.r.account" and "oic.r.coapcloudconf"	
77	Resources	9
78	Table 2 – Device connection with the OCF Cloud flow	11
79	Table 3 – Definition of the "oic.r.account" Resource.....	13
80	Table 4 – Properties of the "oic.r.account" Resource	14
81	Table 5 – Definition of the "oic.r.session" Resource	15
82	Table 6 – Properties of the "oic.r.session" Resource.....	15
83	Table 7 – Definition of the "oic.r.tokenrefresh" Resource	16
84	Table 8 – Properties of the "oic.r.tokenrefresh" Resource	16
85	Table 9 – Sensitive Data related to OCF Cloud	16
86	Table A.1 – Alphabetized list of security resources	18
87	Table A.2 – The Property definitions of the Resource with type "rt" = "oic.r.account".	21
88	Table A.3 – The CRUDN operations of the Resource with type "rt" = "oic.r.account".	22
89	Table A.4 – The Property definitions of the Resource with type "rt" = "oic.r.session".	24
90	Table A.5 – The CRUDN operations of the Resource with type "rt" = "oic.r.session".	25
91	Table A.6 – The Property definitions of the Resource with type "rt" = "oic.r.tokenrefresh".	28
92	Table A.7 – The CRUDN operations of the Resource with type "rt" = "oic.r.tokenrefresh".	29
93		

94 **1 Purpose and Role**

95 This document defines security objectives, philosophy, resources and mechanism that impacts
96 OCF base layers of ISO/IEC 30118-1:2018. ISO/IEC 30118-1:2018 contains informative security
97 content. The OCF Security Document contains security normative content and may contain
98 informative content related to the OCF base or other OCF documents.

99 **2 Normative References**

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

103 IETF RFC 7228, *Terminology for Constrained-Node Networks*, May 2014,
104 <https://tools.ietf.org/html/rfc7228>

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

110 OCF Security Document, Information technology – Open Connectivity Foundation (OCF)
111 Document, Latest version available
112 at:https://openconnectivity.org/specs/OCF_Security_Specification.pdf

113 OCF Device to Cloud Services Document, Information technology – Open Connectivity
114 Foundation (OCF) Document – Part 8: Device to Cloud Services, Latest version available at:
115 https://openconnectivity.org/specs/OCF_OCF_Device_To_Cloud_Services_Specification.pdf

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

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

120

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

123 oneM2M Release 3 Documents, <http://www.onem2m.org/technical/published-drafts>

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

126

127

128

129

130

131

132 **3 Terms, definitions, and abbreviated terms**

133 **3.1 Terms and definitions**

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

136 ISO and IEC maintain terminological databases for use in standardization at the following
137 addresses:

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

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

140 **3.1.1**

141 **Access Management Service (AMS)**

142 a service that dynamically constructs ACL Resources in response to a Device Resource request.

143 Note 1 to entry: An AMS can evaluate access policies remotely and supply the result to a Server which allows or denies
144 a pending access request. An AMS is authorised to provision ACL Resources.

145 **3.1.2**

146 **Trust Anchor**

147 a well-defined, shared authority, within a trust hierarchy, by which two cryptographic entities (e.g.
148 a Device and an onboarding tool) can assume trust

149 **3.1.3**

150 **OCF Security Domain**

151 a set of onboarded OCF Devices that are provisioned with credentialing information for confidential
152 communication with one another

153 **3.1.4**

154 **Access Token**

155 a credential used to authorize the connection with the OCF Cloud and access protected resources.
156 An Access Token is a string while the OCF Device has no internal logic based on its contents and
157 only forwards the token as-is

158 **3.1.5**

159 **Authorization Provider**

160 a Server issuing Access Tokens (3.1.4) to the Client after successfully authenticating the OCF
161 Cloud User (3.1.7) and obtaining authorization.

162 Note 1 to entry: Also known as authorization server in IETF RFC 6749.

163 **3.1.6**

164 **Device Registration**

165 a process by which Device is enrolled/registered to the OCF Cloud infrastructure (using Device
166 certificate and unique credential) and becomes ready for further remote operation through the cloud
167 interface (e.g. connection to remote Resources or publishing of its own Resources for access).

168 **3.1.7**

169 **OCF Cloud User**

170 a person or organization authorizing a set of Devices to interact with each other via an OCF Cloud.

171 Note 1 to entry: For each of the Devices, the OCF Cloud User is either the same as, or a delegate of, the person or
172 organization that onboarded that Device. The OCF Cloud User delegates, to the OCF Cloud authority, authority to route
173 between Devices registered by the OCF Cloud User. The OCF Cloud delegates, to the OCF Cloud User, authority to
174 select the set of Devices which can register and use the services of the OCF Cloud.

175 **3.2 Abbreviated terms**

176 **3.2.1**

177 **ACE**
178 Access Control Entry

179 **3.2.2**

180 **ACL**
181 Access Control List

182 **3.2.3**

183 **AMS**
184 Access Management Service

185 **3.2.4**

186 **CMS**
187 Credential Management Service

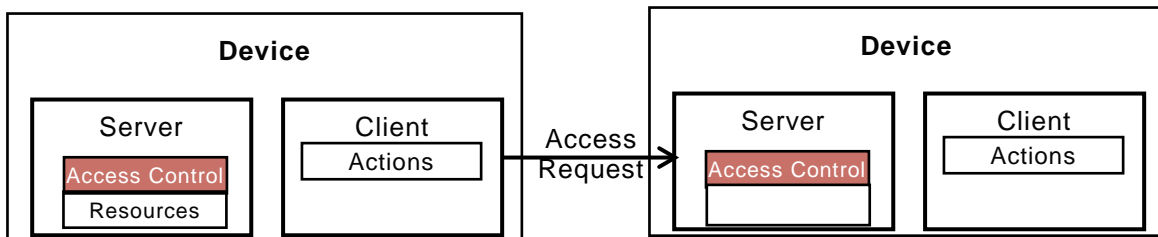
188 **4 Document Conventions and Organization**

189 **4.1 Conventions**

190 This document defines Resources, protocols and conventions used to implement security for OCF
191 core framework and applications.

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

193 Figure 1 depicts interaction between OCF Devices.



194

195 **Figure 1 – OCF Interaction**

196 Devices may implement a Client role that performs Actions on Servers. Actions access Resources
197 managed by Servers. The OCF stack enforces access policies on Resources. End-to-end Device
198 interaction can be protected using session protection protocol (e.g. DTLS) or with data encryption
199 methods.

200 **4.2 Notation**

201 In this document, features are described as required, recommended, allowed or DEPRECATED as
202 follows:

203 **Required** (or **shall** or **mandatory**).

204 These basic features shall be implemented to comply with OCF Core Architecture. The phrases
205 "shall not", and "PROHIBITED" indicate behavior that is prohibited, i.e. that if performed means the
206 implementation is not in compliance.

207 **Recommended** (or **should**).

208 These features add functionality supported by OCF Core Architecture and should be implemented.
209 Recommended features take advantage of the capabilities OCF Core Architecture, usually without
210 imposing major increase of complexity. Notice that for compliance testing, if a recommended
211 feature is implemented, it shall meet the specified requirements to be in compliance with these
212 guidelines. Some recommended features could become requirements in the future. The phrase
213 "should not" indicates behavior that is permitted but not recommended.

214 **Allowed** (may or allowed).

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

217 **Conditionally allowed** (CA)

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

220 **Conditionally required** (CR)

221 The definition or behaviour depends on a condition. If the specified condition is met, then the
222 definition or behaviour is required. Otherwise the definition or behaviour is allowed as default
223 unless specifically defined as not allowed.

224 **DEPRECATED**

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

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

232 Words that are emphasized are printed in italic.

233 **4.3 Data types**

234 See ISO/IEC 30118-1:2018.

235 **4.4 Document structure**

236 Informative clauses may be found in the Overview clauses, while normative clauses fall outside of
237 those clauses.

238 The Security Document may use the oneM2M Release 3 Documents,
239 <http://www.onem2m.org/technical/published-drafts>

240 OpenAPI as the API definition language. The mapping of the CRUDN actions is specified in
241 ISO/IEC 30118-1:2018.

242

243 5 Security overview

244 5.1 Preamble

245 A Device is authorized to communicate with an OCF Cloud if a trusted Mediator has provisioned
246 the Device.

- 247 – Device and Mediator connect over DTLS using "/oic/sec/cred"
- 248 – Device is provisioned by Mediator with following information:
 - 249 – the URL of OCF Cloud
 - 250 – Authorization Provider Name to identify the origin of the Access Token
 - 251 – Access Token / Authorization Code that is validated / exchanged by the OCF Cloud
 - 252 – UUID of the OCF Cloud

253 The OpenAPI 2.0 definitions (Annex A) used in this document are normative. This includes that all
254 defined payloads shall comply with the indicated OpenAPI 2.0 definitions. Annex A contains all of
255 the OpenAPI 2.0 definitions for Resource Types defined in this document.

256 5.2 Device Provisioning for OCF Cloud and Device Registration Overview

257 As mentioned in the start of Clause 0, communication between a Device and OCF Cloud is subject
258 to different criteria in comparison to Devices which are within a single local network. The Device is
259 configured in order to connect to the OCF Cloud by a Mediator as specified in the CoAPCloudConf
260 Resource clauses in OCF Cloud . Provisioning includes the remote connectivity and local details
261 such as URL where the OCF Cloud hosting environment can be found, the OCF Cloud verifiable
262 Access Token and optionally the name of the Authorization Provider which issued the Access
263 Token.

264 NOTE a Device which connects to the OCF Cloud still retains the ownership established at onboarding with the DOTS.

265 5.3 Credential overview

266 Devices may use credentials to prove the identity and role(s) of the parties in bidirectional
267 communication

268 Access Tokens are provided to an OCF Cloud once an authenticated session with an OCF Cloud
269 is established, to verify the User ID with which the Device is to be associated.

270 6 Device provisioning for OCF Cloud

271 6.1 Cloud Provisioning General

272 The Device that connects to the OCF Cloud shall support the "oic.r.coapcloudconf" Resource on
273 Device and following SVRs on the OCF Cloud: "/oic/sec/account", "/oic/sec/session",
274 "/oic/sec/tokenrefresh".

275 The OCF Cloud is expected to use a secure mechanism for associating a Mediator with an OCF
276 Cloud User. The choice of mechanism is up to the OCF Cloud. Recommended solution is based on
277 the OAuth2.0 Authorization Grant Type flow specified in IETF RFC 6749, where the Mediator act
278 as a User-Agent and presents authorization UI to the user - see Figure 2. OCF Cloud is expected
279 to ensure that the suitable authentication mechanism is used to authenticate the OCF Cloud User.

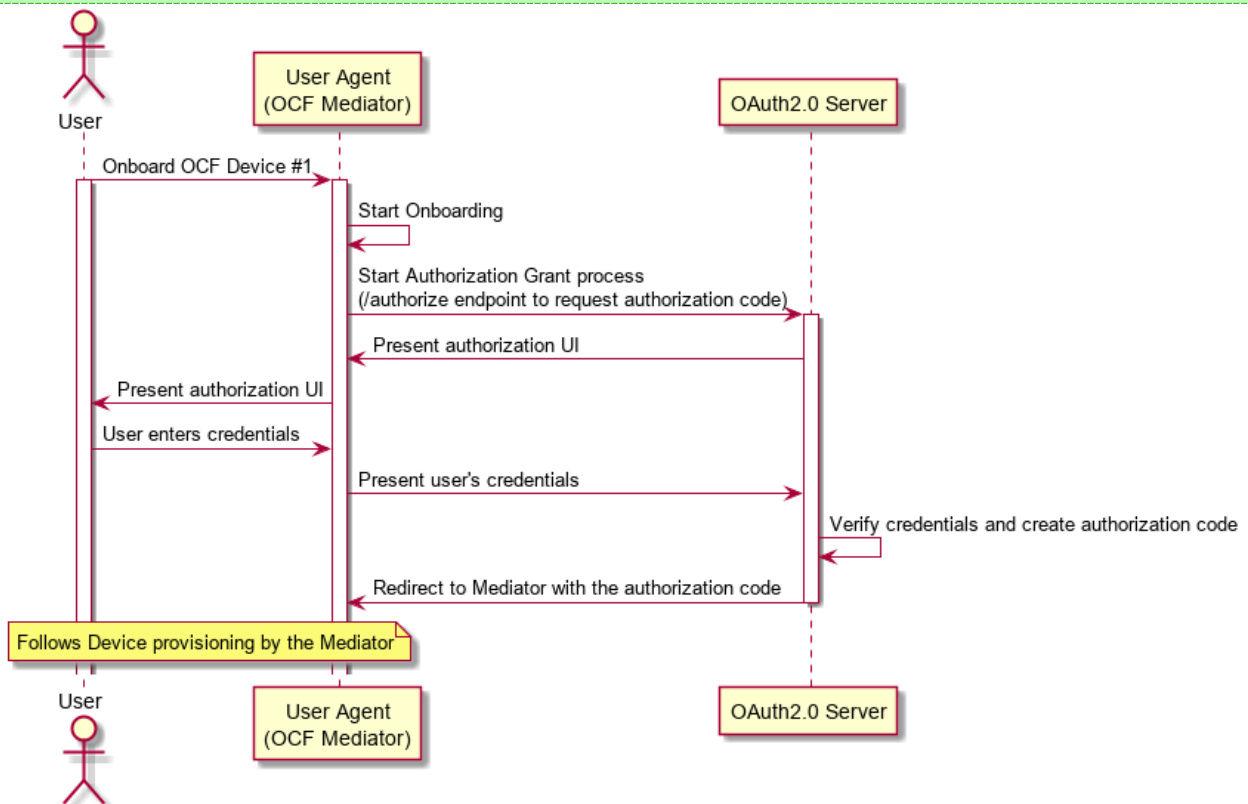
280 **Figure 2 User authorization and provisioning using Authorization Code Grant Flow**

```
281 @startuml
282
283 actor User
284 participant UserAgent as "User Agent\n(OCF Mediator)"
285 participant OAuthServer as "OAuth2.0 Server"
286 activate User
UserAgent->>OAuthServer: OAuth2.0 Server
UserAgent->>User: UserAgent: Onboard OCF Device #1
User->>UserAgent: activate User
```

```

287 activate UserAgent
288 UserAgent -> UserAgent: Start Onboarding
289 UserAgent -> OAuthServer: Start Authorization Grant process\n(/authorize endpoint to
290 request authorization code)
291 activate OAuthServer
292 OAuthServer -> UserAgent: Present authorization UI
293 UserAgent -> User: Present authorization UI
294 User -> UserAgent: User enters credentials
295 UserAgent -> OAuthServer: Present user's credentials
296 OAuthServer -> OAuthServer: Verify credentials and create authorization code
297 OAuthServer -> UserAgent: Redirect to Mediator with the authorization code
298 deactivate OAuthServer
299 note over User, UserAgent
300 Follows Device provisioning by the Mediator
301 end note
302
303 @enduml

```



304
305
306

307 6.2 Device Provisioning by Mediator

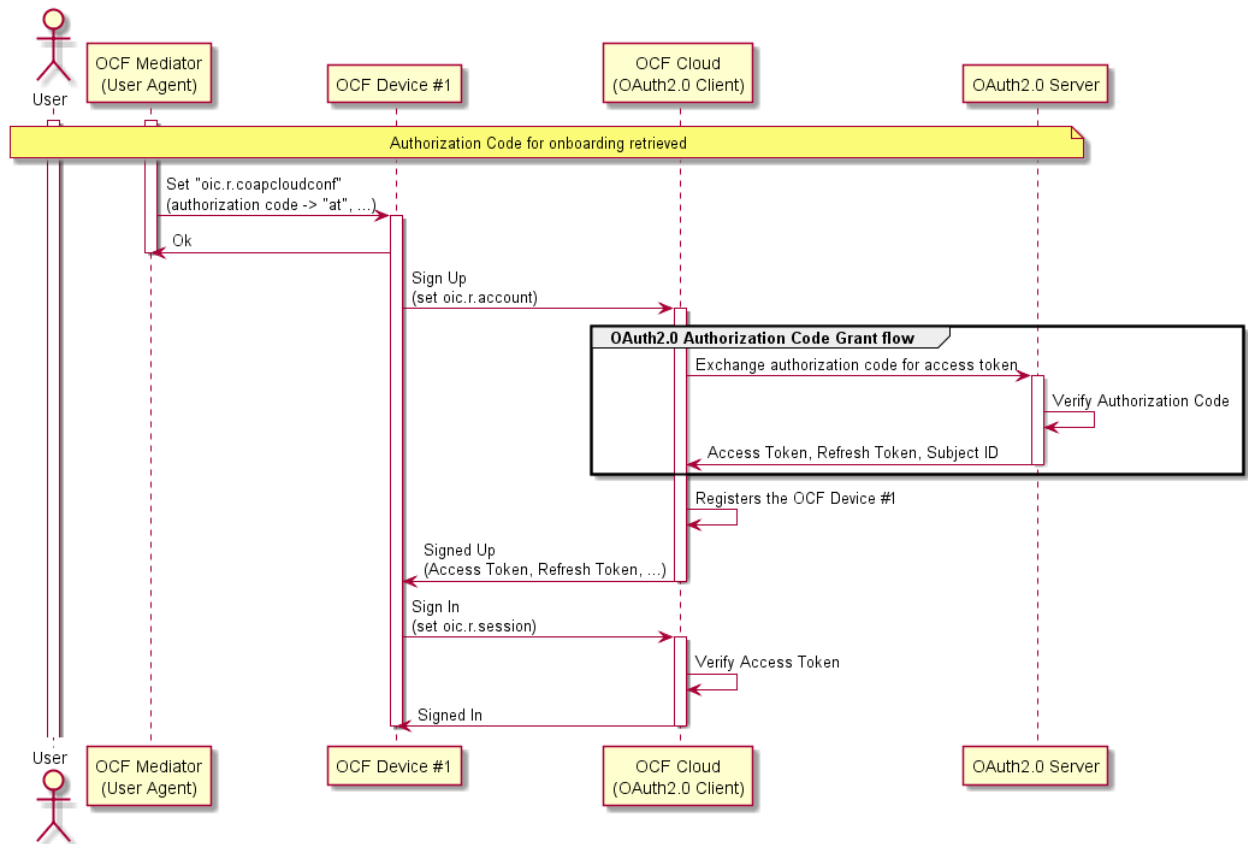
308 The Mediator and the Device shall use the secure session to provision the Device to connect with
309 the OCF Cloud.

310 The Mediator obtains an Authorization Code or directly an Access Token from the Authorization
311 Server as described in OCF Cloud . This value is then used by the Device for registering with the
312 OCF Cloud as described in clause 7. At the time of Device Registration OCF Cloud exchanges the
313 Authorization Code for the Access Token, returns it back to the OCF Device and associates the
314 TLS session with corresponding Device ID. The OCF Cloud maintains a map where Access Token
315 and Mediator provided Device ID are stored.

316 The Mediator provisions the Device, as described in OCF Cloud . The Mediator provisions OCF
317 Cloud URI to the "cis" Property of "oic.r.coapcloudconf" Resource, OCF Cloud UUID to the "sid"
318 Property of "oic.r.coapcloudconf" Resource and per-device Access Token or Authorization Code to
319 the "at" Property of "oic.r.coapcloudconf" Resource on Device. Exchanged and returned
320 provisioned Access Token is to be treated by Device as an Access Token with "Bearer" token type
321 as defined in IETF RFC 6750. The provisioned "at" value follows a proprietary data format, and
322 may include multiple values marshalled/concatenated together into a single string (e.g.
323 "{\"token\": \"abc\", \"client_id\": \"1234\", \"idp\": \"identityProvider1\"}" is a valid "at" Property value).
324 See Figure 3 for the detailed overview of the recommended flow, which includes optional OAuth
325 2.0 Authorization Code Grant

326 **Figure 3 Device Provisioning using Authorization Code Grant Flow**

```
327 @startuml
328
329 actor User
330 participant Mediator as "OCF Mediator\n(User Agent)"
331 participant Device as "OCF Device #1"
332 participant Cloud as "OCF Cloud\n(OAuth2.0 Client)"
333 participant OAuthServer as "OAuth2.0 Server"
334
335 activate User
336 activate Mediator
337
338 note over User, OAuthServer
339 Authorization Code for onboarding retrieved
340 end note
341
342 Mediator -> Device: Set "oic.r.coapcloudconf"\n(authorization code -> "at", ...)
343 activate Device
344 Device -> Mediator: Ok
345 deactivate Mediator
346 Device -> Cloud: Sign Up\n(set oic.r.account)
347 activate Cloud
348 group OAuth2.0 Authorization Code Grant flow
349     Cloud -> OAuthServer: Exchange authorization code for access token
350     activate OAuthServer
351     OAuthServer -> OAuthServer: Verify Authorization Code
352     OAuthServer -> Cloud: Access Token, Refresh Token, Subject ID
353     deactivate OAuthServer
354 end
355 Cloud -> Cloud: Registers the OCF Device #1
356 Cloud -> Device: Signed Up\n(Access Token, Refresh Token, ...)
357 deactivate Cloud
358 Device -> Cloud: Sign In\n(set oic.r.session)
359 activate Cloud
360 Cloud -> Cloud: Verify Access Token
361 Cloud -> Device: Signed In
362 deactivate Device
363 deactivate Cloud
364
365 @enduml
```



366
367
368

369 For the purposes of access control, the Device shall identify the OCF Cloud using the OCF Cloud
370 UUID in the Common Name field of the End-Entity certificate used to authenticate the OCF Cloud.

371 AMS should configure the ACE2 entries on a Device so that the Mediator(s) is the only Device(s)
372 with UPDATE permission for the "oic.r.coapcloudconf" Resource.

373 The AMS should configure the ACE2 entries on the Device to allow request from the OCF Cloud.
374 By request from the Mediator, the AMS removes old ACL2 entries with previous OCF Cloud UUID.
375 This request happens before "oic.r.coapcloudconf" is configured by the Mediator for the new OCF
376 Cloud. The Mediator also requests AMS to set the OCF Cloud UUID as the "subject" Property for
377 the new ACL2 entries. AMS may use "sid" Property of "oic.r.coapcloudconf" Resource as the
378 current OCF Cloud UUID. AMS could either provision a wildcard entry for the OCF Cloud or
379 provision an entry listing each Resource published on the Device.

380 If OCF Cloud provides "redirecturi" Value as response during Device Registration, the redirected-
381 to OCF Cloud is assumed to have the same OCF Cloud UUID and to use the same trust anchor.
382 Otherwise, presented OCF Cloud UUID wouldn't match the provisioned ACL2 entries.

383 The Mediator should provision the "oic.r.coapcloudconf" Resource with the Properties in Table 1.
384 These details once provisioned are used by the Device to perform Device Registration to the OCF
385 Cloud. OCF Device is not expected to have any internal logic based on the values of "at" and "apn"
386 Properties. The values of these Properties are forwarded as-is to the OCF Cloud. After the initial
387 registration, the Device should use updated values received from the OCF Cloud instead. If OCF
388 Cloud User wants the Device to re-register with the OCF Cloud, they can use the Mediator to re-
389 provision the "oic.r.coapcloudconf" Resource with the new values.

390
391

Table 1 – Mapping of Properties of the "oic.r.account" and "oic.r.coapcloudconf" Resources

Property Title	oic.r.coapcloudconf	oic.r.account	Description
Authorization Provider Name	apn	authprovider	The name of Authorization Provider through which Access Token was obtained.
OCF Cloud URL	cis	-	This is the URL connection is established between Device and OCF Cloud.
Access Token	at	accesstoken	Access Token used to authorize the TLS connection for communication with the OCF Cloud, or the Authorization Code which is then verified and exchanged for the Access Token during Device Registration.
OCF Cloud UUID	sid	-	This is the identity of the OCF Cloud that the Device is configured to use.

392

7 Device authentication with OCF Cloud

393

7.1 Device Authentication with OCF Cloud General

394 The mechanisms for Device Authentication in clauses 10.2, 10.3 and 10.4 of OCF Security imply
395 that a Device is authorized to communicate with any other Device meeting the criteria provisioned
396 in "/oic/sec/cred"; the "/oic/sec/acl2" Resource (or "/oic/sec/acl1" resource of OIC1.1 Servers) are
397 additionally used to restrict access to specific Resources. The present clause describes Device
398 authentication for OCF Cloud, which uses slightly different criteria as described in clause 0. A
399 Device accessing an OCF Cloud shall establish a TLS session. The mutual authenticated TLS
400 session is established using Server certificate and Client certificate.

401 Each Device is identified by the Access Token obtained from the Device Registration response.
402 The OCF Cloud holds an OCF Cloud association table that maps Access Token, User ID and Device
403 ID. The Device Registration shall happen while the Device is in RFNOP state. After Device
404 Registration, the updated Access Token, Device ID and User ID are used by the Device for the
405 subsequent connection with the OCF Cloud.

7.2 Device Connection with the OCF Cloud

406 The Device should establish the TLS connection using the certificate based credential. The
407 connection should be established after Device is provisioned by Mediator.

408 The TLS session is established between Device and the OCF Cloud as specified in IETF RFC 8323.
409 The OCF Cloud is expected to provide certificate signed by trust anchor that is present in cred
410 entries of the Device. These cred entries are expected to be configured by the Mediator.
411

412 The Device shall validate the OCF Cloud's identity based on the credentials that are contained in
413 "/oic/sec/cred" Resource entries of the Device.

414 The OCF Cloud is expected to validate the manufacturer certificate provided by the Device.

415 The assumption is that the OCF Cloud User trusts the OCF Cloud that the Device connects. The
416 OCF Cloud connection should not happen without the consent of the OCF Cloud User. The

417 assumption is that the OCF Cloud User has either service agreement with the OCF Cloud provider
418 or uses manufacturer provided OCF Cloud.

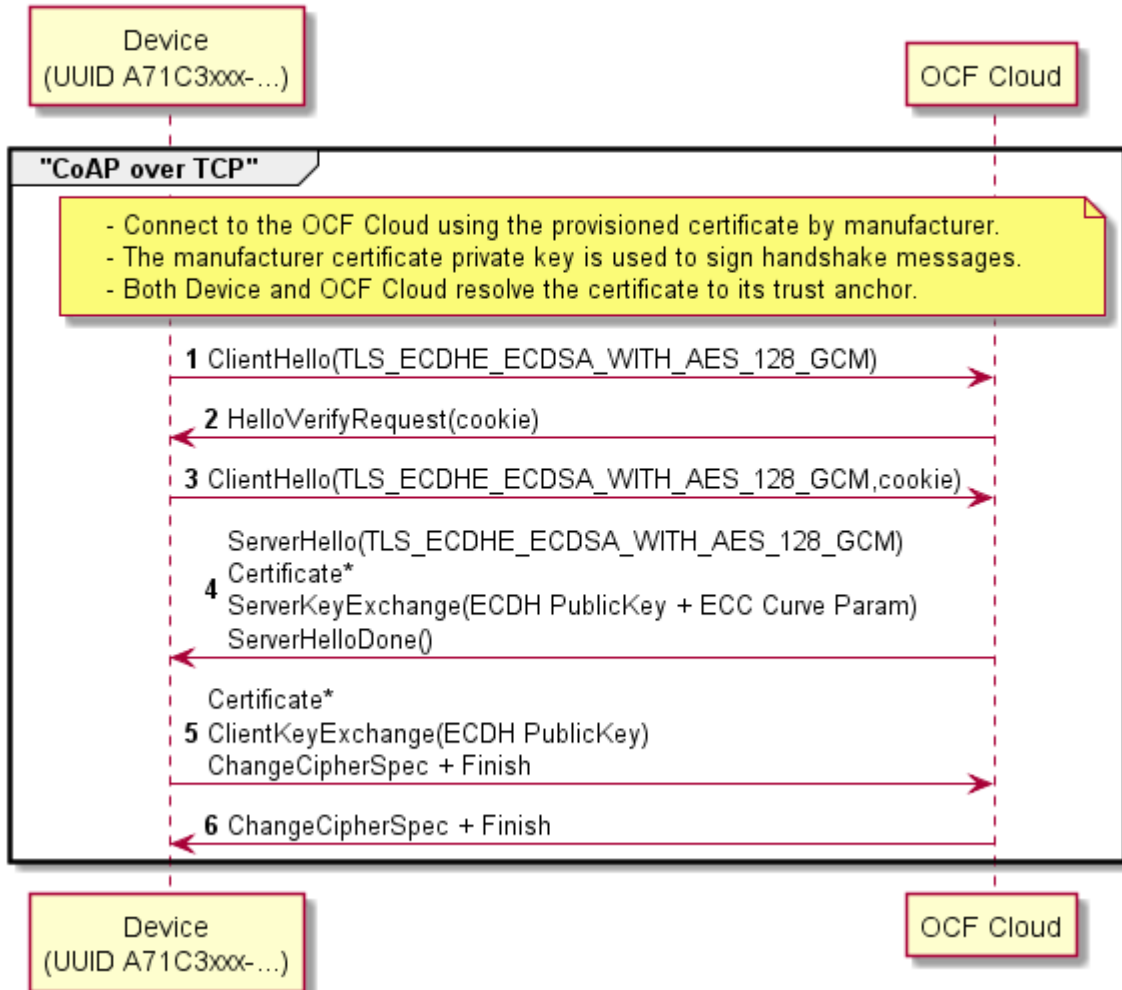
419 If authentication fails, the "clec" Property of "oic.r.coapcloudconf" Resource on the Device shall be
420 updated about the failed state, if it is supported by the Device. If authentication succeeds, the
421 Device and OCF Cloud should establish an encrypted link in accordance with the negotiated cipher
422 suite.

423 Figure 4 depicts sequence for Device connection with OCF Cloud and steps described in Table 2.

424 **Figure 4 – Device connection with OCF Cloud**

```
425 @startuml
426 autonumber
427 title Device Connection with OCF Cloud
428 participant "Device\n(UUID A71C3xxx-...)" as RS
429 participant "OCF Cloud" as CI
430
431 group "CoAP over TCP"
432 note over RS, CI
433 - Connect to the OCF Cloud using the provisioned certificate by manufacturer.
434 - The manufacturer certificate private key is used to sign handshake messages.
435 - Both Device and OCF Cloud resolve the certificate to its trust anchor.
436 end note
437
438 RS->CI: ClientHello(TLS_ECDHE_ECDSA_WITH_AES_128_GCM)
439 CI->RS: HelloVerifyRequest(cookie)
440 RS->CI: ClientHello(TLS_ECDHE_ECDSA_WITH_AES_128_GCM,cookie)
441 CI->RS:
442 ServerHello(TLS_ECDHE_ECDSA_WITH_AES_128_GCM)\nCertificate*\nServerKeyExchange(ECDH
443 PublicKey + ECC Curve Param)\nServerHelloDone()
444 RS->CI: Certificate*\nClientKeyExchange(ECDH PublicKey)\nChangeCipherSpec + Finish
445 CI->RS: ChangeCipherSpec + Finish
446
447 End
448 @enduml
```

Device Connection with OCF Cloud



449

450

Table 2 – Device connection with the OCF Cloud flow

Steps	Description
1 - 6	TLS connection between the OCF Cloud and Device. The Device's manufacturer certificate may contain data attesting to the Device hardening and security properties

451 7.3 Security Considerations

452 When an OCF Server receives a request sent via the OCF Cloud, then the OCF Server permits
 453 that request using the identity of the OCF Cloud rather than the identity of the OCF Client. If there
 454 is no mechanism through which the OCF Cloud permits only those interactions which the user
 455 intends between OCF Clients and OCF Server via the OCF Cloud, and denies all other interactions,
 456 then OCF Clients might get elevated privileges by submitting a request via the OCF Cloud. This is
 457 highly undesirable from the security perspective. Consequently, OCF Cloud implementations are
 458 expected to provide some mechanism through which the OCF Cloud prevents OCF Clients getting
 459 elevated privileges when submitting a request via the OCF Cloud. In the present document release,
 460 the details of the mechanism are left to the implementation.

461 The security considerations about the manufacturer certificate as described in clause 7.3.6.5 of
 462 OCF Security are also applicable in the Device authentication with the OCF Cloud.

463 The Device should validate the OCF Cloud's TLS certificate as defined by IETF RFC 6125 and in
464 accordance with its requirements for Server identity authentication.

465 The "uid" and "di" Property Value of "/oic/d" Resource may be considered personally identifiable
466 information in some regulatory regions, and the OCF Cloud is expected to provide protections
467 appropriate to its governing regulatory bodies.

468 **8 Message integrity and confidentiality**

469 **8.1 Cloud Session Semantics**

470 The messages between the OCF Cloud and Device shall be exchanged only if the Device and OCF
471 Cloud authenticate each other as described in 7. The asymmetric cipher suites as described in 8.2
472 shall be employed for establishing a secured session and for encrypting/decrypting between the
473 OCF Cloud and the Device. The OCF Endpoint sending the message shall encrypt and authenticate
474 the message using the cipher suite as described in 8.2 and the OCF Endpoint shall verify and
475 decrypt the message before processing it.

476 **8.2 Cipher suites for OCF Cloud Credentials**

477 All Devices supporting OCF Cloud Certificate Credentials shall implement:

478 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

479 All Devices supporting OCF Cloud Certificate Credentials should implement:

480 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,

481 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,

482 TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,

483 TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,

484 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

485

486 **9 Security resources**

487 **9.1 Account Resource**

488 The Account Resource specifies the Properties based on IETF RFC 6749 Access Token based
489 account creation. The mechanism to obtain credentials is described in Clause 6. The Account
490 Resource is used for Device Registration. The Account Resource is instantiated on the OCF Cloud
491 as "/oic/sec/account" SVR and is used by cloud-enabled Devices to register with the OCF Cloud. It
492 should be only accessible on a secure channel; non-secure channel should not be able access this
493 Resource.

494 During the Device Registration process, an OCF Cloud can provide a distinct URI of another OCF
495 Cloud ("redirected-to" OCF Cloud). Both initial and redirected-to OCF Clouds are expected to
496 belong to the same Vendor; they are assumed to have the same UUID and are assumed to have
497 an Out-of-Band Communication Channel established. Device does not have to perform the Device
498 Registration on the redirected-to OCF Cloud and the OCF Cloud may ignore such attempts.
499 Redirected-to OCF Cloud is expected to accept the Access Token, provided to the Device by the
500 initial OCF Cloud.

501 The RETRIEVE operation on OCF Cloud's "/oic/sec/account" Resource is not allowed and the OCF
502 Cloud is expected to reject all attempts to perform such operation.

503 The UPDATE operation on the OCF Cloud's "/oic/sec/account" Resource behaves as follows:

- 504 – A Device intending to register with the OCF Cloud shall send UPDATE with following Properties
 505 "di" ("di" Property Value of "/oic/d" Resource), and "accesstoken" as configured by the Mediator
 506 ("at" Property Value of "oic.r.coapcloudconf" Resource). The OCF Cloud verifies it is the same
 507 "accesstoken" which was assigned to the Mediator for the corresponding "di" Property Value.
 508 The "accesstoken" is the permission for the Device to access the OCF Cloud. If the "apn" was
 509 included when the Mediator UPDATED the "oic.r.coapcloudconf" Resource, the Device shall
 510 also include "authprovider" Property when registering with the OCF Cloud. If no "apn" is
 511 specified, then the "authprovider" Property shall not be included in the UPDATE request.
- 512 – OCF Cloud returns "accesstoken", "uid", "refreshtoken", and "expiresin" It may also return
 513 "redirecturi". Received "accesstoken" is to be treated by Device as an Access Token with
 514 "Bearer" token type as defined in IETF RFC 6750. This "accesstoken" shall be used for the
 515 following Account Session start using "oic/sec/session" SVR. Received "refreshtoken" is to be
 516 treated by Device as a Refresh Token as defined in IETF RFC 6749. The Device stores the
 517 OCF Cloud's Response values. If "redirecturi" is received, Device shall use received value as
 518 a new OCF Cloud URI instead of "cis" Property Value of "oic.r.coapcloudconf" Resource for
 519 further connections.
- 520 The DELETE operation on the OCF Cloud's "/oic/sec/account" Resource should behave as follows:
- 521 – To deregister with the OCF Cloud, a DELETE operation shall be sent with the "accesstoken"
 522 and either "uid", or "di" to be deregistered with the OCF Cloud. On DELETE with the OCF Cloud,
 523 the Device should also delete values internally stored. Once deregister with an OCF Cloud,
 524 Device can connect to any other OCF Cloud. Device deregistered need to go through the steps
 525 in 6 again to be registered with the OCF Cloud.

526 Format of "oic.r.account" Resource is defined in Table 3.

527 **Table 3 – Definition of the "oic.r.account" Resource**

Fixed URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
/oic/sec/account	Account	oic.r.account	oic.if.basel ine	Resource used for a device to add itself under a given credential	N/A

528 Table 4 defines the Properties of "oic.r.account".

Table 4 – Properties of the "oic.r.account" Resource

Property Title	Property Name	Value Type	Value Rule	Access Mode	Mandatory	Description
Device ID	di	string	uuid	W	Yes	Unique Device identifier. Format pattern according to IETF RFC 4122.
Authorization Provider Name	authprovider	string	N/A	W	No	The name of Authorization Provider through which Access Token was obtained.
Access Token	accesstoken	string	Non-empty string	W	Yes	Access Token used to authorize and associate the TLS connection for communication with the OCF Cloud with the Device ID, or the Authorization Code which is then verified and exchanged for the Access Token during Device Registration.
Access Token	accesstoken	string	Non-empty string	R	Yes	Access Token used to authorize and associate the TLS connection for communication with the OCF Cloud with the Device ID.
Refresh Token	refreshtoken	string	Non-empty string	R	Yes	Refresh token can be used to refresh the Access Token before getting expired.
Token Expiration	expiresin	integer	-	R	Yes	Access Token life time in seconds (-1 if permanent).
User ID	uid	string	uuid	R	Yes	Unique OCF Cloud User identifier. Format pattern according to IETF RFC 4122.
Redirect URI	redirecturi	string	-	R	No	Using this URI, the Client needs to reconnect to a redirected OCF Cloud. If provided, this value shall be used by the Device instead of Mediator-provided URI during the Device Registration.

530 9.2 Account Session resource

531 The "/oic/sec/session" Resource hosted on the OCF Cloud is used for creating connections with
 532 the OCF Cloud subsequent to Device registration though "/oic/sec/account" Resource. The
 533 "/oic/sec/session" Resource requires the device ID, User ID and Access Token which are stored
 534 securely on the Device.

535 The "/oic/sec/session" Resource is exposed by the OCF Cloud. It should be only accessible on a
 536 secure channel; non-secure channel cannot access this Resource.

537 The RETRIEVE operation on OCF Cloud's "/oic/sec/session" Resource is not allowed and the OCF
 538 Cloud is expected to reject all attempts to perform such operation.

539 The UPDATE operation is defined as follows for OCF Cloud's "/oic/sec/session" Resource:

- 540 – The Device connecting to the OCF Cloud shall send an UPDATE request message to the OCF
 541 Cloud's "/oic/sec/session" Resource. The message shall include the "di" Property Value of
 542 "/oic/d" Resource and "uid", "login" Value ("true" to establish connection; "false" to disconnect)
 543 and "accesstoken" as returned by OCF Cloud during Device Registration. The OCF Cloud
 544 verifies it is the same Access Token which was returned to the Device during Device
 545 Registration process or during Token Refresh. If Device was attempting to establish the
 546 connection and provided values were verified as correct by the OCF Cloud, OCF Cloud sends
 547 a response with remaining lifetime of the associated Access Token ("expiresin" Property Value).

548 "oic.r.session" Resource is defined in Table 5.

549

Table 5 – Definition of the "oic.r.session" Resource

Fixed URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
/oic/sec/session	Account Session	oic.r.session	oic.if.baseline	Resource that enables a device to manage its session using login or logout	N/A

550 Table 6 defines the Properties of "oic.r.session".

551

Table 6 – Properties of the "oic.r.session" Resource

Property Title	Property Name	Value Type	Value Rule	Access Mode	Mandatory	Description
User ID	uid	string	uuid	W	Yes	User ID provided by Device Registration process. Format pattern according to IETF RFC 4122.
Device ID	di	string	uuid	W	Yes	Unique device id registered for a Device.Format pattern according to IETF RFC 4122.
Access Token	accesstoken	string	A string of at least one character	W	Yes	Access Token used to authorize and associate the TLS connection for communication with the OCF Cloud with the Device ID
Login Status	login	boolean	N/A	W	Yes	Action for the request: true = login, false = logout
Token Expiration	expiresin	integer	N/A	R	Yes	Remaining Access Token life time in seconds (-1 if permanent) This Property is only provided to Device during connection establishment (when "login" Property Value equals "true"), it's not available otherwise

552 **9.3 Account Token Refresh Resource**

553 The "/oic/sec/tokenrefresh" Resource is used by the Device for refreshing the Access Token.

554 The "/oic/sec/tokenrefresh" Resource is hosted by the OCF Cloud. It should be only accessible on
555 a secure channel; non-secure channel cannot access this Resource.

556 The Device should use "/oic/sec/tokenrefresh" to refresh the Access Token with the OCF Cloud,
557 when the time specified in "expiresin" is near.

558 The RETRIEVE operation on OCF Cloud's "/oic/sec/ tokenrefresh" Resource is not allowed and the
559 OCF Cloud is expected to reject all attempts to perform such operation.

560 The UPDATE operation is defined as follows for "/oic/sec/tokenrefresh" Resource

561 – The Device attempting to refresh the Access Token shall send an UPDATE request message
562 to the OCF Cloud's "/oic/sec/tokenrefresh" Resource. The message shall include the "di"
563 Property Value of "/oic/d" Resource, "uid" and "refreshtoken", as returned by OCF Cloud.

564 – OCF Cloud response is expected to include a "refreshtoken", new "accesstoken", and
565 "expiresin". Received "accesstoken" is to be treated by Device as an Access Token with
566 "Bearer" token type as defined in IETF RFC 6750. This Access Token is the permission for the
567 Device to access the OCF Cloud. Received "refreshtoken" is to be treated by Device as a
568 Refresh Token as defined in IETF RFC 6749. Received "refreshtoken" may be the new Refresh

569 Token or the same one as provided by the Device in the UPDATE request. In case when new
 570 distinct "refreshtoken" is provided by the OCF Cloud, the Device shall discard the old value.
 571 The OCF Cloud's response values "refreshtoken", "acesstoken" and "expiresin" are securely
 572 stored on the Device.

573 "oic.r.tokenrefresh" Resource is defined in Table 7.

574 **Table 7 – Definition of the "oic.r.tokenrefresh" Resource**

Fixed URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
/oic/sec/tokenrefresh	Token Refresh	oic.r.tokenrefresh	oic.if.baseline	Resource to manage the access-token using refresh token	N/A

575 Table 8 defines the Properties of "oic.r.tokenrefresh".

576 **Table 8 – Properties of the "oic.r.tokenrefresh" Resource**

Property Title	Property Name	Value Type	Value Rule	Access Mode	Mandatory	Description
User ID	uid	string	uuid	W	Yes	User ID provided by Sign-up process. Format pattern according to IETF RFC 4122.
Device ID	di	string	uuid	W	Yes	Unique device id registered for an OCF Cloud User account. Format pattern according to IETF RFC 4122.
Refresh Token	refreshtoken	string	A string of at least one character	RW	Yes	Refresh token can be used to refresh the Access Token before getting expired.
Access Token	acesstoken	string	A string of at least one character	R	Yes	Access Token used to authorize and associate the TLS connection for communication with the OCF Cloud with the Device ID.
Token Expiration	expiresin	integer	-	R	Yes	Access Token life time in seconds (-1 if permanent).

577 **10 Security hardening guidelines**

578 **10.1 Security hardening guidelines general**

579 In addition to the Sensitive Data list outlined in Table 75 of Security Document, any Device
 580 implementing OCF Cloud connection capabilities should also provide reasonable protection for the
 581 information in Table 9.

582 **Table 9 – Sensitive Data related to OCF Cloud**

Data	Integrity protection	Confidentiality protection
OCF Cloud URL	Yes	Not required
OCF Cloud Identity	Yes	Not required

583

584

585

586

587

**Annex A
(normative)
Resource Type definitions**

588
589
590
591

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

593 **Table A.1 – Alphabetized list of security resources**

Friendly Name (informative)	Resource Type (rt)	Clause
Account	oic.r.account	A.1
Account Session	oic.r.session	A.2
Account Token Refresh	oic.r.tokenrefresh	A.3

594 **A.1 Account Token**

595 **A.1.1 Introduction**

596 Sign-up using generic account provider.

597 **A.1.2 Well-known URI**

598 /oic/sec/account

599 **A.1.3 Resource type**

600 The Resource Type is defined as: "oic.r.account".

601 **A.1.4 OpenAPI 2.0 definition**

```

602 {
603   "swagger": "2.0",
604   "info": {
605     "title": "Account Token",
606     "version": "20190111",
607     "license": {
608       "name": "OCF Data Model License",
609       "url":
610 "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
611 CENSE.md",
612       "x-copyright": "copyright 2016-2017, 2019 Open Connectivity Foundation, Inc. All rights
613 reserved."
614     },
615     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
616   },
617   "schemes": ["http"],
618   "consumes": ["application/json"],
619   "produces": ["application/json"],
620   "paths": {
621     "/oic/sec/account" : {
622       "post": {
623         "description": "Sign-up using generic account provider.\n",
624         "parameters": [
625           { "$ref": "#/parameters/interface" },
626           {
627             "name": "body",
628             "in": "body",
629             "required": true,
630             "schema": { "$ref": "#/definitions/Account-request" },
631             "x-example":
632             {
633               "di" : "9cfbeb8e-5ale-4dlc-9d01-00c04fd430c8",
634               "authprovider" : "github",
635               "accesstoken" : "8802f2eaf8b5e147a936"
636             }
637           }
638         ]
639       }
640     }
641   }

```

```

637     }
638   ],
639   "responses": {
640     "204": {
641       "description": "2.04 Changed respond with required and optional information\n",
642       "x-example":
643         {
644           "rt": ["oic.r.account"],
645           "accesstoken": "0f3d9f7fe5491d54077d",
646           "refreshtoken": "00fe4644a6fbe5324eec",
647           "expiresin": 3600,
648           "uid": "123e4567-e89b-12d3-a456-d6e313b71d9f",
649           "redirecturi": "coaps+tcp://example.com:443"
650         },
651       "schema": { "$ref": "#/definitions/Account-response" }
652     }
653   }
654 },
655 "delete": {
656   "description": "Delete a device. This also removes all resources in the device on cloud
657 side.\nexample: /oic/account?di=9cfbeb8e-5ale-4dlc-9d01-
658 00c04fd430c8&accesstoken=0f3d9f7fe5491d54077d\n",
659   "parameters": [
660     { "$ref": "#/parameters/interface" }
661   ],
662   "responses": {
663     "202": {
664       "description": "2.02 Deleted response informing the device is successfully
665 deleted.\n"
666     }
667   }
668 }
669 }
670 },
671 "parameters": {
672   "interface": {
673     "in": "query",
674     "name": "if",
675     "type": "string",
676     "enum": ["oic.if.baseline"]
677   }
678 },
679 "definitions": {
680   "Account-request": {
681     "properties": {
682       "authprovider": {
683         "description": "The name of Authorization Provider through which Access Token was
684 obtained",
685         "type": "string"
686       },
687       "accesstoken": {
688         "description": "Access-Token used for communication with OCF Cloud after account
689 creation",
690         "pattern": "(?!$|\\s+).*",
691         "type": "string"
692       },
693       "di": {
694         "description": "Format pattern according to IETF RFC 4122.",
695         "pattern": "^-[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
696 9]{12}$",
697         "type": "string"
698       }
699     },
700     "type": "object",
701     "required": ["di", "accesstoken"]
702   },
703   "Account-response": {
704     "properties": {
705       "expiresin": {
706         "description": "Access-Token remaining life time in seconds (-1 if permanent)",
707         "readOnly": true,

```



```

708         "type": "integer"
709     },
710     "rt": {
711         "description": "Resource Type of the Resource",
712         "items": {
713             "maxLength": 64,
714             "type": "string",
715             "enum" : ["oic.r.account"]
716         },
717         "minItems": 1,
718         "maxItems": 1,
719         "readOnly": true,
720         "type": "array"
721     },
722     "refreshToken" : {
723         "description": "Refresh token can be used to refresh the Access Token before getting
724 expired",
725         "pattern": "(?!$|\\s+).*",
726         "readOnly": true,
727         "type": "string"
728     },
729     "uid" : {
730         "description": "Format pattern according to IETF RFC 4122.",
731         "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
732 9]{12}$",
733         "type": "string"
734     },
735     "accessToken" : {
736         "description": "Access-Token used for communication with cloud after account creation",
737         "pattern": "(?!$|\\s+).*",
738         "type": "string"
739     },
740     "n": {
741         "$ref":
742 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
743 schema.json#/definitions/n"
744     },
745     "id": {
746         "$ref":
747 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
748 schema.json#/definitions/id"
749     },
750     "redirecturi" : {
751         "description": "Using this URI, the Client needs to reconnect to a redirected OCF Cloud.
752 If provided, this value shall be used by the Device instead of Mediator-provided URI during the
753 Device Registration.",
754         "readOnly": true,
755         "type": "string"
756     },
757     "if": {
758         "description": "The interface set supported by this resource",
759         "items": {
760             "enum": [
761                 "oic.if.baseline"
762             ],
763             "type": "string"
764         },
765         "minItems": 1,
766         "maxItems": 1,
767         "uniqueItems": true,
768         "readOnly": true,
769         "type": "array"
770     }
771 },
772 "type" : "object",
773 "required": ["accessToken", "refreshToken", "expiresin", "uid"]
774 }
775 }
776 }
777

```

778 **A.1.5 Property definition**

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

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

Property name	Value type	Mandatory	Access mode	Description
di	string	Yes	Write Only	Unique Device identifier. Format pattern according to IETF RFC 4122.
authprovider	string	No	Write Only	The name of Authorization Provider through which Access Token was obtained.
acesstoken	string	Yes	Write Only	Access Token used to authorize and associate the TLS connection for communication with the OCF Cloud with the Device ID, or the Authorization Code which is then verified and exchanged for the Access Token during Device Registration.
id	multiple types: see schema	No	Read Write	
refreshtoken	string	Yes	Read Only	Refresh token can be used to refresh the Access Token before getting expired.
rt	array: see schema	No	Read Only	Resource Type of the Resource
acesstoken	string	Yes	Read Only	Access Token used to authorize and associate the TLS connection for communication with the OCF Cloud with the Device ID.
uid	string	Yes	Read Only	Unique OCF Cloud User identifier. Format

				pattern according to IETF RFC 4122.
expiresin	integer	Yes	Read Only	Access-Token life time in seconds (-1 if permanent)
if	array: see schema	No	Read Only	The interface set supported by this resource
redirecturi	string	No	Read Only	Using this URI, the Client needs to reconnect to a redirected OCF Cloud. If provided, this value shall be used by the Device instead of Mediator-provided URI during the Device Registration.
n	multiple types: see schema	No	Read Write	

781 **A.1.6 CRUDN behaviour**

782 Table A.3 defines the CRUDN operations that are supported on the "oic.r.account" Resource Type.

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

Create	Read	Update	Delete	Notify
		post	delete	

784 **A.2 Session**

785 **A.2.1 Introduction**

786 Resource that manages the persistent session between a Device and OCF Cloud.

787 **A.2.2 Well-known URI**

788 /oic/sec/session

789 **A.2.3 Resource type**

790 The Resource Type is defined as: "oic.r.session".

791 **A.2.4 OpenAPI 2.0 definition**

```

792 {
793   "swagger": "2.0",
794   "info": {
795     "title": "Session",
796     "version": "v1.0-20181001",
797     "license": {
798       "name": "OCF Data Model License",
799       "url":
800 "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
801 CENSE.md",
802       "x-copyright": "copyright 2016-2017, 2019 Open Connectivity Foundation, Inc. All rights
803 reserved."
804     }
805   }

```

```

805     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
806   },
807   "schemes": ["http"],
808   "consumes": ["application/json"],
809   "produces": ["application/json"],
810   "paths": {
811     "/oic/sec/session" : {
812       "post": {
813         "description": "Resource that manages the persistent session between a Device and OCF
814 Cloud.",
815         "parameters": [
816           { "$ref": "#/parameters/interface" },
817           {
818             "name": "body",
819             "in": "body",
820             "required": true,
821             "schema": { "$ref": "#/definitions/Account-Session-Request" },
822             "x-example":
823               {
824                 "uid" : "123e4567-e89b-12d3-a456-d6e313b71d9f",
825                 "di" : "9cfbeb8e-5ale-4dlc-9d01-00c04fd430c8",
826                 "accesstoken" : "0f3d9f7fe5491d54077d",
827                 "login" : true
828               }
829           }
830         ],
831         "responses": {
832           "204": {
833             "description": "",
834             "x-example":
835               {
836                 "rt": ["oic.r.session"],
837                 "expiresin" : 3600
838               },
839             "schema": { "$ref": "#/definitions/Account-Session-Response" }
840           }
841         }
842       }
843     }
844   },
845   "parameters": {
846     "interface" : {
847       "in" : "query",
848       "name" : "if",
849       "type" : "string",
850       "enum" : ["oic.if.baseline"]
851     }
852   },
853   "definitions": {
854     "Account-Session-Request" : {
855       "properties": {
856         "uid": {
857           "description": "Format pattern according to IETF RFC 4122.",
858           "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
859 9]{12}$",
860           "type": "string"
861         },
862         "di": {
863           "description": "The Device ID\nFormat pattern according to IETF RFC 4122.",
864           "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
865 9]{12}$",
866           "type": "string"
867         },
868         "accesstoken": {
869           "description": "Access-Token used to grant access right for the Device to sign-in.",
870           "pattern": "(?!$|\\s+).*",
871           "type": "string"
872         },
873         "login": {
874           "description": "Action for the request: true = login, false = logout.",
875           "type": "boolean"

```

```

876     },
877   },
878   "type" : "object",
879   "required": ["uid", "di", "accesstoken", "login"]
880 },
881 "Account-Session-Response" : {
882   "properties": {
883     "expiresin": {
884       "description": "Access-Token remaining life time in seconds (-1 if permanent).",
885       "readOnly": true,
886       "type": "integer"
887     },
888     "rt": {
889       "description": "Resource Type of the Resource.",
890       "items": {
891         "maxLength": 64,
892         "type": "string",
893         "enum": ["oic.r.session"]
894       },
895       "minItems": 1,
896       "readOnly": true,
897       "type": "array"
898     },
899     "n": {
900       "$ref":
901       "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
902       schema.json#/definitions/n"
903     },
904     "id": {
905       "$ref":
906       "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
907       schema.json#/definitions/id"
908     },
909     "if": {
910       "description": "The interface set supported by this Resource.",
911       "items": {
912         "enum": [
913           "oic.if.baseline"
914         ],
915         "type": "string"
916       },
917       "minItems": 1,
918       "readOnly": true,
919       "type": "array"
920     }
921   },
922   "type" : "object",
923   "required" : ["expiresin"]
924 }
925 }
926 }
927

```

928 A.2.5 Property definition

929 Table A.4 defines the Properties that are part of the "oic.r.session" Resource Type.

930 **Table A.4 – The Property definitions of the Resource with type "rt" = "oic.r.session".**

Property name	Value type	Mandatory	Access mode	Description
if	array: see schema	No	Read Only	The interface set supported by this Resource.
expiresin	integer	Yes	Read Only	Remaining Access Token life time in seconds (-1 if permanent). This Property is only

				provided to Device during connection establishment (when "login" Property Value equals "true"), it's not available otherwise.
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	
di	string	Yes	Write Only	Unique device id registered for a Device. Format pattern according to IETF RFC 4122.
accesstoken	string	Yes	Write Only	Access Token used to authorize and associate the TLS connection for communication with the OCF Cloud with the Device ID.
uid	string	Yes	Write Only	User ID provided by Device Registration process. Format pattern according to IETF RFC 4122.
login	boolean	Yes	Write Only	Action for the request: true = login, false = logout.

931 **A.2.6 CRUDN behaviour**

932 Table A.5 defines the CRUDN operations that are supported on the "oic.r.session" Resource Type.

933 **Table A.5 – The CRUDN operations of the Resource with type "rt" = "oic.r.session".**

Create	Read	Update	Delete	Notify
		post		

934 **A.3 Token Refresh**

935 **A.3.1 Introduction**

936 Obtain fresh Access Token using the refresh token, client should refresh Access Token before it
937 expires.

938 **A.3.2 Well-known URI**

939 /oic/sec/tokenrefresh

940 **A.3.3 Resource type**

941 The Resource Type is defined as: "oic.r.tokenrefresh".

942 **A.3.4 OpenAPI 2.0 definition**

```
943 {
944   "swagger": "2.0",
945   "info": {
946     "title": "Token Refresh",
947     "version": "v1.0-20181001",
948     "license": {
949       "name": "OCF Data Model License",
950       "url":
951 "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
952 CENSE.md",
953     "x-copyright": "copyright 2016-2017, 2019 Open Connectivity Foundation, Inc. All rights
954 reserved."
955   },
956   "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
957 },
958 "schemas": ["http"],
959 "consumes": ["application/json"],
960 "produces": ["application/json"],
961 "paths": {
962   "/oic/sec/tokenrefresh" : {
963     "post": {
964       "description": "Obtain fresh access-token using the refresh token, client should refresh
965 access-token before it expires.\n",
966       "parameters": [
967         { "$ref": "#/parameters/interface" },
968         {
969           "name": "body",
970           "in": "body",
971           "required": true,
972           "schema": { "$ref": "#/definitions/TokenRefresh-Request" },
973           "x-example":
974             {
975               "uid" : "123e4567-e89b-12d3-a456-d6e313b71d9f",
976               "di" : "9cfbeb8e-5ale-4d1c-9d01-00c04fd430c8",
977               "refreshtoken" : "00fe4644a6fbe5324eec"
978             }
979         }
980       ],
981       "responses": {
982         "204": {
983           "description": "2.04 Changed respond with new access-token.\n",
984           "x-example":
985             {
986               "rt": ["oic.r.tokenrefresh"],
987               "accesstoken" : "8ce598980761869837be",
988               "refreshtoken" : "d4922312b6df0518e146",
989               "expiresin" : 3600
990             }
991           ,
992           "schema": { "$ref": "#/definitions/TokenRefresh-Response" }
993         }
994       }
995     }
996   }
997 },
998 "parameters": {
999   "interface" : {
1000     "in" : "query",
1001     "name" : "if",
1002     "type" : "string",
1003     "enum" : ["oic.if.baseline"]
1004   }
}
```

```

1005     },
1006     "definitions": {
1007         "TokenRefresh-Request" : {
1008             "properties": {
1009                 "refreshToken": {
1010                     "description": "Refresh token received by account management or during token refresh
1011 procedure.",
1012                     "pattern": "(?!$|\\s+).*",
1013                     "type": "string"
1014                 },
1015                 "uid": {
1016                     "description": "Format pattern according to IETF RFC 4122.",
1017                     "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
1018 9]{12}$",
1019                     "type": "string"
1020                 },
1021                 "di": {
1022                     "description": "Format pattern according to IETF RFC 4122.",
1023                     "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
1024 9]{12}$",
1025                     "type": "string"
1026                 }
1027             },
1028             "type": "object",
1029             "required": ["uid", "di", "refreshToken"]
1030         },
1031         "TokenRefresh-Response" : {
1032             "properties": {
1033                 "expiresin": {
1034                     "description": "Access-Token life time in seconds (-1 if permanent).",
1035                     "readOnly": true,
1036                     "type": "integer"
1037                 },
1038                 "rt": {
1039                     "description": "Resource Type of the Resource.",
1040                     "items": {
1041                         "maxLength": 64,
1042                         "type": "string",
1043                         "enum": ["oic.r.tokenrefresh"]
1044                     },
1045                     "minItems": 1,
1046                     "readOnly": true,
1047                     "type": "array"
1048                 },
1049                 "refreshToken": {
1050                     "description": "Refresh token received by account management or during token refresh
1051 procedure.",
1052                     "pattern": "(?!$|\\s+).*",
1053                     "type": "string"
1054                 },
1055                 "accesstoken": {
1056                     "description": "Granted Access-Token.",
1057                     "pattern": "(?!$|\\s+).*",
1058                     "readOnly": true,
1059                     "type": "string"
1060                 },
1061                 "n": {
1062                     "$ref":
1063 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1064 schema.json#/definitions/n"
1065                 },
1066                 "id": {
1067                     "$ref":
1068 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1069 schema.json#/definitions/id"
1070                 },
1071                 "if" :
1072                 {
1073                     "description": "The interface set supported by this Resource.",
1074                     "items": {
1075                         "enum": [

```



```

1076         "oic.if.baseline"
1077     ],
1078     "type": "string"
1079 },
1080 "minItems": 1,
1081 "readOnly": true,
1082 "type": "array"
1083 }
1084 },
1085 "type" : "object",
1086 "required": ["accesstoken", "refreshtoken", "expiresin"]
1087 }
1088 }
1089 }
1090

```

1091 A.3.5 Property definition

1092 Table A.6 defines the Properties that are part of the "oic.r.tokenrefresh" Resource Type.

1093 **Table A.6 – The Property definitions of the Resource with type "rt" = "oic.r.tokenrefresh".**

Property name	Value type	Mandatory	Access mode	Description
refreshtoken	string	Yes	Write Only	Refresh token can be used to refresh the Access Token before getting expired.
uid	string	Yes	Write Only	User ID provided by Sign-up process. Format pattern according to IETF RFC 4122.
di	string	Yes	Write Only	Unique device id registered for an OCF Cloud User account. Format pattern according to IETF RFC 4122.
if	array: see schema	No	Read Only	The interface set supported by this Resource.
expiresin	integer	Yes	Read Only	Access Token life time in seconds (-1 if permanent).
accesstoken	string	Yes	Read Only	Access Token used to authorize and associate the TLS connection for communication with the OCF Cloud with the Device ID.
refreshtoken	string	Yes	Read Only	Refresh token can be used to

				refresh the Access Token before getting expired.
n	multiple types: see schema	No	Read Write	
rt	array: see schema	No	Read Only	Resource Type of the Resource.
id	multiple types: see schema	No	Read Write	

1094 **A.3.6 CRUDN behaviour**

1095 Table A.7 defines the CRUDN operations that are supported on the "oic.r.tokenrefresh" Resource
 1096 Type.

1097 **Table A.7 – The CRUDN operations of the Resource with type "rt" = "oic.r.tokenrefresh".**

Create	Read	Update	Delete	Notify
		post		

1098

1099