

OCF Cloud API for Cloud Services

Hollywood | May 2020



OPEN CONNECTIVITY
FOUNDATION™

CONTACT admin@openconnectivity.org

Copyright Open Connectivity Foundation, Inc. © 2019-2020.
All Rights Reserved.

1 **OCF “Hollywood” – OCF Cloud API for Cloud Services Specification – Core Technology WG**
2 **CR 2931**

3
4
5 **Legal Disclaimer**
6

7 THIS IS A DRAFT SPECIFICATION DOCUMENT ONLY AND HAS NOT BEEN ADOPTED BY THE
8 OPEN CONNECTIVITY FOUNDATION. THIS DRAFT DOCUMENT MAY NOT BE RELIED UPON
9 FOR ANY PURPOSE OTHER THAN REVIEW OF THE CURRENT STATE OF THE DEVELOPMENT
10 OF THIS DRAFT DOCUMENT. THE OPEN CONNECTIVITY FOUNDATION AND ITS MEMBERS
11 RESERVE THE RIGHT WITHOUT NOTICE TO YOU TO CHANGE ANY OR ALL PORTIONS
12 HEREOF, DELETE PORTIONS HEREOF, MAKE ADDITIONS HERETO, DISCARD THIS DRAFT
13 DOCUMENT IN ITS ENTIRETY OR OTHERWISE MODIFY THIS DRAFT DOCUMENT AT ANY
14 TIME. YOU SHOULD NOT AND MAY NOT RELY UPON THIS DRAFT DOCUMENT IN ANY WAY,
15 INCLUDING BUT NOT LIMITED TO THE DEVELOPMENT OF ANY PRODUCTS OR SERVICES.
16 IMPLEMENTATION OF THIS DRAFT DOCUMENT IS DONE AT YOUR OWN RISK AMEND AND
17 IT IS NOT SUBJECT TO ANY LICENSING GRANTS OR COMMITMENTS UNDER THE OPEN
18 CONNECTIVITY FOUNDATION INTELLECTUAL PROPERTY RIGHTS POLICY OR OTHERWISE.
19 IN CONSIDERATION OF THE OPEN CONNECTIVITY FOUNDATION GRANTING YOU ACCESS
20 TO THIS DRAFT DOCUMENT, YOU DO HEREBY WAIVE ANY AND ALL CLAIMS ASSOCIATED
21 HERewith INCLUDING BUT NOT LIMITED TO THOSE CLAIMS DISCUSSED BELOW, AS WELL
22 AS CLAIMS OF DETRIMENTAL RELIANCE.

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

25 Copyright © 2020 Open Connectivity Foundation, Inc. All rights reserved.

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

28 CONTENTS

30 1 Scope 1

31 2 Normative references 1

32 3 Terms, definitions, and abbreviated terms 1

33 3.1 Terms and definitions..... 1

34 3.2 Abbreviated terms..... 2

35 4 Document conventions and organization..... 3

36 4.1 Conventions..... 3

37 4.2 Notation..... 3

38 5 Overview 4

39 5.1 Introduction..... 4

40 5.2 General OCF Cloud API for Cloud Services Elements 4

41 5.3 Cloud to Cloud Operational Overview 5

42 5.3.1 Introduction 5

43 5.3.2 Conceptual Architecture 5

44 5.3.3 Authorizing Cloud Connectivity 6

45 5.3.4 Synchronization of User's set of Devices 6

46 5.3.5 Keeping Up-to-Date: Notifications of changes on other Clouds 6

47 5.3.6 Handling of Requests and Responses for Connected Devices 7

48 6 Authentication & Authorization..... 7

49 7 Account Linking API 7

50 7.1 General..... 7

51 7.2 OAuth2.0 Access Token Scopes 8

52 8 Devices API..... 10

53 8.1 Introduction..... 10

54 8.2 Parameters Supported in Requests..... 10

55 8.3 Retrieve All Devices..... 10

56 8.3.1 Summary 10

57 8.3.2 Request and Response Payload 11

58 8.3.3 Responses 12

59 8.4 Retrieve One Device..... 12

60 8.4.1 Summary 12

61 8.4.2 Request and Response Payload 13

62 8.4.3 Responses 13

63 8.5 Retrieve Specific Resource..... 14

64 8.5.1 Summary 14

65 8.5.2 Request and Response Payload 14

66 8.5.3 Responses 15

67 8.6 Update a Resource on a Device..... 15

68 8.6.1 Summary 15

69	8.6.2	Request and Response Payload	16
70	8.6.3	Responses	16
71	9	Events API	17
72	9.1	Introduction.....	17
73	9.2	Events Authentication	18
74	9.2.1	Create Event Signature	18
75	9.2.2	Verify the Event Signature	Error! Bookmark not defined.
76	9.3	Parameters Supported	19
77	9.4	Events API subscription and notification payload definitions	19
78	9.4.1	Subscription request.....	19
79	9.4.2	Subscription response	20
80	9.4.3	Notification request.....	21
81	9.5	Subscribe and unsubscribe to devices level event types	22
82	9.5.1	Summary	23
83	9.5.2	Request and Response Payload	23
84	9.5.3	Responses	23
85	9.6	Subscribe and unsubscribe to device level events.....	24
86	9.6.1	Summary	24
87	9.6.2	Request and Response Payload	24
88	9.6.3	Responses	25
89	9.7	Subscribe and unsubscribe to resource level events	25
90	9.7.1	Summary	25
91	9.7.2	Request and Response Payload	26
92	9.7.3	Responses	26
93	9.8	Notification of devices level events	27
94	9.8.1	Summary	27
95	9.8.2	Request and Response Payload	27
96	9.8.3	Responses	27
97	9.9	Notification of Device level events	28
98	9.9.1	Summary	28
99	9.9.2	Request and Response Payload	28
100	9.9.3	Responses	28
101	9.10	Notification of Resource level events	29
102	9.10.1	Summary	29
103	9.10.2	Request and Response Payload	29
104	9.10.3	Responses	29
105		Annex A Representative Flows	30
106	A.1	Introduction.....	30
107	A.2	OAuth2.0 Application Registration.....	30
108	A.3	Account Linking	30
109	A.4	Retrieval of all Devices	31
110	A.4.1	Summary	31
111	A.4.2	Flow	31
112	A.4.3	Flow Description	32

113	A.5	Retrieval of a single Device	32
114	A.5.1	Summary	32
115	A.5.2	Flow	32
116	A.5.3	Flow Description	33
117	A.6	Retrieval of a single Resource	33
118	A.6.1	Summary	33
119	A.6.2	Flows.....	33
120	A.7	Update of a single Resource	35
121	A.7.1	Summary	35
122	A.7.2	Flows.....	35
123	A.8	Establishment of new subscription request.....	36
124	A.8.1	Summary	36
125	A.9	Event generated for a subscription.....	37
126	A.9.1	Summary	37
127	A.10	Addition of new registration.....	37
128	A.10.1	Summary	37
129	A.11	Removal of existing device registration	38
130	A.11.1	Summary	38
131	Annex B	Open API Definition	39
132	B.1.1	Supported APIs	39
133	B.1.2	OpenAPI 2.0 definition.....	40
134			
135			

136
137
138

Figures

139	Figure 1 – OCF Cloud Overview	4
140	Figure 2 – Conceptual Architecture	6
141	Figure 3 – Subscription Request Example.....	20
142	Figure 4 – Subscription Response Example Payload.....	21
143	Figure A.1 – Establish Business Relationship Example Flow.....	30
144	Figure A.2 – Initial Association Example Flow	31
145	Figure A.3 – Retrieve All Devices Example Flow	32
146	Figure A.4 – Retrieve Single Device Example Flow	33
147	Figure A.5 – Retrieve Resource (Success) Example Flow	34
148	Figure A.6 – Retrieve Resource (Timeout) Example Flow	35
149	Figure A.7 – Update Resource (Success) Example Flow	35
150	Figure A.8 – Update Resource (Timeout) Example Flow	36
151	Figure A.9 – Observe Establishment Example Flow	37
152	Figure A.10 – "resource_contentchanged" Event Example Flow	37
153	Figure A.11 – Addition of new registered Device example flow.....	38
154	Figure A.12 – Removal of existing registration example flow.....	38

155
156

Tables

157	
158	
159	Table 1 – OAuth 2.0 AccessToken Scopes..... 9
160	Table 2 – Applicable OAuth2.0 Access Token Scopes per API Endpoint 9
161	Table 3 – Parameters used in Requests in the Device API 10
162	Table 4 – Retrieve All Devices API Summary 11
163	Table 5 – Response payload Property definition 11
164	Table 6 – "device" Property definition 11
165	Table 7 – Devices API non-success path responses 12
166	Table 8 – Retrieve One Device API Summary 13
167	Table 9 – Device API non-success path responses 13
168	Table 10 – Retrieve Specific Resource API Summary 14
169	Table 11 – Resource Retrieval API non-success path responses 15
170	Table 12 – Update Resource API Summary 16
171	Table 13 – Resource Update API non-success path responses 17
172	Table 14 – Parameters used in the Events API 19
173	Table 15 – Event types and API Endpoints 19
174	Table 16 – Subscription Request Payload Properties 20
175	Table 17 – Subscription Response Properties 20
176	Table 18 – Notification request HTTP Headers 21
177	Table 19 – Event type to notification payload content 22
178	Table 20 – Subscription to /devices API Summary 23
179	Table 21 – Devices Event Subscription API non-success path responses 24
180	Table 22 – Subscription to Single Device API Summary 24
181	Table 23 – Device Event Subscription API non-success path responses 25
182	Table 24 – Subscription to Resource API Summary 25
183	Table 25 – Resource Event Subscription API non-success path responses 26
184	Table 26 – Notification of /devices API Summary 27
185	Table 27 – Devices Event Notification non-success path responses 27
186	Table 28 – Notification of Single Device API Summary 28
187	Table 29 – Device Event Notification non-success path responses 28
188	Table 30 – Notification of Resource API Summary 29
189	Table 31 – Resource Event Notification non-success path responses 29
190	Table A.1 – Retrieve all Devices Flow Summary 32
191	Table A.2 – Retrieve single Device Flow Summary 33
192	Table A.3 – Retrieve single Resource Flow Summary 34
193	Table A.4 – Update single Resource Flow Summary 36
194	

195 **1 Scope**

196 This document defines functional requirements for the OCF Cloud to Cloud Application
197 Programming Interface (API).

198 **2 Normative references**

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

203 IETF RFC 2818, *HTTP over TLS*, May 2000
204 <https://tools.ietf.org/html/rfc2818>

205 IETF RFC 5646, *Tags for Identifying Languages*, September 2009
206 <https://www.rfc-editor.org/info/rfc5646>

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

209 IETF RFC 6750, *The OAuth 2.0 Authorization Framework: Bearer Token Usage*, October 2012
210 <https://www.rfc-editor.org/info/rfc6750>

211 IETF RFC 7628, *A Set of Simple Authentication and Security Layer (SASL) Mechanisms for OAuth*,
212 August 2015
213 <https://www.rfc-editor.org/info/rfc7628> ISO/IEC 30118-1:2018 Information technology -- Open
214 Connectivity Foundation (OCF) Specification -- Part 1: Core specification
215 <https://www.iso.org/standard/53238.html>
216 Latest version available at: https://openconnectivity.org/specs/OCF_Core_Specification.pdf

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

221 OCF Device to Cloud Services Specification, *Open Connectivity Foundation Device to Cloud*
222 *Services Specification*,
223 Latest version available at:
224 https://openconnectivity.org/specs/OCF_Cloud_Specification.pdf

225 OCF Cloud API for Cloud Services [https://github.com/openconnectivityfoundation/core-](https://github.com/openconnectivityfoundation/core-extensions/blob/ocfcloud-openapi/swagger2.0/oic.r.cloudopenapi.swagger.json)
226 [extensions/blob/ocfcloud-openapi/swagger2.0/oic.r.cloudopenapi.swagger.json](https://github.com/openconnectivityfoundation/core-extensions/blob/ocfcloud-openapi/swagger2.0/oic.r.cloudopenapi.swagger.json)

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

229

230 **3 Terms, definitions, and abbreviated terms**

231 **3.1 Terms and definitions**

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

234 ISO and IEC maintain terminological databases for use in standardization at the following
235 addresses:

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

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

238

239 **3.1.1**

240 **API Endpoint**

241 a defined URL to which requests defined in this document are sent

242 **3.1.2**

243 **Bearer Token**

244 an OAuth2.0 access token as defined within IETF RFC 6750

245 **3.1.3**

246 **Origin Cloud**

247 the OCF Cloud or the 3rd party Cloud through which the user works with his OCF Devices

248 **3.1.4**

249 **Subscription ID**

250 a unique identity that is associated with an instance of a subscription to an event (or events)

251 **3.1.5**

252 **Target Cloud**

253 the OCF Cloud to which OCF Servers (OCF Devices) are connected which the user wants to control
254 via the Origin Cloud (3.1.2)

255 **3.2 Abbreviated terms**

256 **3.2.1**

257 **API**

258 Application Programming Interface

259 **3.2.2**

260 **HMAC**

261 Hash-based Message Authentication Code

262

263 4 Document conventions and organization

264 4.1 Conventions

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

269 4.2 Notation

270 In this document, features are described as required, recommended, allowed or DEPRECATED as
271 follows:

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

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

276 Recommended (or should)(S).

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

283 Allowed (may or allowed)(O).

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

287 DEPRECATED.

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

294 Conditionally allowed (CA)

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

297 Conditionally required (CR)

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

301

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

303 Words that are emphasized are printed in italic.

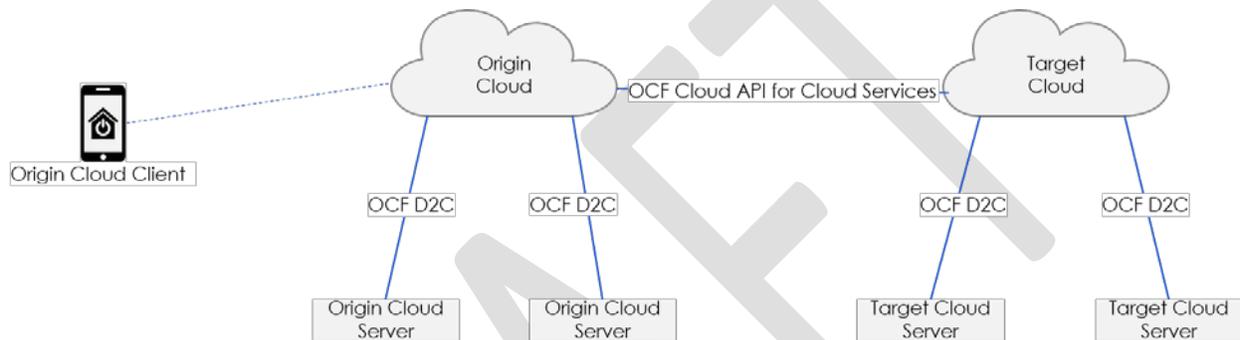
304 5 Overview

305 5.1 Introduction

306 This document defines the OCF Cloud API for Cloud Services. In this document Origin Cloud refers
307 to the OCF Cloud or the 3rd party Cloud through which the user works with his OCF Devices, Target
308 Cloud refers to the OCF Cloud to which OCF Servers (OCF Devices) are connected which the user
309 wants to control via the Origin Cloud.

310 An OCF Device is a collection of Resources, each Resource being an OpenAPI 2.0 defined object
311 that represents a physical property or characteristic of the Device (e.g. temperature sensed, light
312 colour, power on switch). The Device itself has an associated Device Type that provides an
313 indication of what the Device is, for example a Light is represented as a Device Type of "oic.d.light".

314 Please see Figure 1 for a representation of the target architecture.



315

316 **Figure 1 – OCF Cloud Overview**

317 The OCF Cloud API for Cloud Services supports the following cases:

- 318 – Account Linking API (clause 7)
 - 319 – Initial Account Linking
 - 320 – Removal of linked account
- 321 – Devices API (clause 8)
 - 322 – Retrieval of all Devices associated with a User (clause 8.3)
 - 323 – Retrieval of a single Device associated with a User (clause 8.4)
 - 324 – Retrieval of a single Resource (clause 8.5)
 - 325 – Update of a single Resource (clause 8.6)
- 326 – Events API (clause 9)
 - 327 – Subscription to an event: establishment of a subscription (clause 9.4.1)
 - 328 – Notification: event generated on an established subscription (clause 9.4.3)

329 5.2 General OCF Cloud API for Cloud Services Elements

330 The OCF Cloud API for Cloud Services is a RESTful API over HTTPS (IETF RFC 2818). The API
331 is defined using OpenAPI 2.0.

332 The Origin Cloud communicates with the Target Cloud using the domain name or URI it has
333 obtained from the initial OAuth 2.0 (IETF RFC 6749) Client Setup, covered in clause 7.
334 Communication between OCF Devices and OCF Clouds is defined in the OCF Device to Cloud
335 Services Specification.

336 All URIs presented within a "href" Link Parameter present in any payload shall be in the form
337 "/<deviceId>/<resourcehref>"; where <deviceId> is the identity of the Device as provided in the "di"
338 Property of "/oic/d" and "resourcehref" is the "href" of the Resource as provided by the Target Cloud.

339 An Origin Cloud shall obtain a Bearer Token from the Target Cloud using standard OAuth2.0 (IETF
340 RFC 6749) mechanisms. All subsequent requests from an Origin Cloud to the Target Cloud shall
341 include this Bearer Token for the user in question.

342 Any query parameters received by an Origin Cloud in a request from an OCF Client shall be passed
343 through clean (i.e. are part of the URI) in any request that is sent to a Target Cloud.

344 Each request may contain an optional HTTP Correlation-ID header, which carries a unique identifier
345 value that provides a reference to a particular transaction or event chain in the Target Cloud. If the
346 request does contain a Correlation-ID header, a Correlation-ID populated with the same value shall
347 be present in any response to that request. If the request does not contain a Correlation-ID header,
348 one should be present in the response.

349 All requests shall include an HTTP Accept header. All requests or responses that carry content
350 shall include an HTTP Content-Type header. At a minimum media-types "application/json" and
351 "application/vnd.ocf+cbor" shall be supported. If the recipient of a request cannot provide a
352 response that is encoded according to the content of the Accept header, then a HTTP 406 (not
353 acceptable) response should be sent in accordance with IETF RFC 2818. On reception of a 406
354 response the originator of the request may re-attempt the request using an alternative Content-
355 Type if supported.

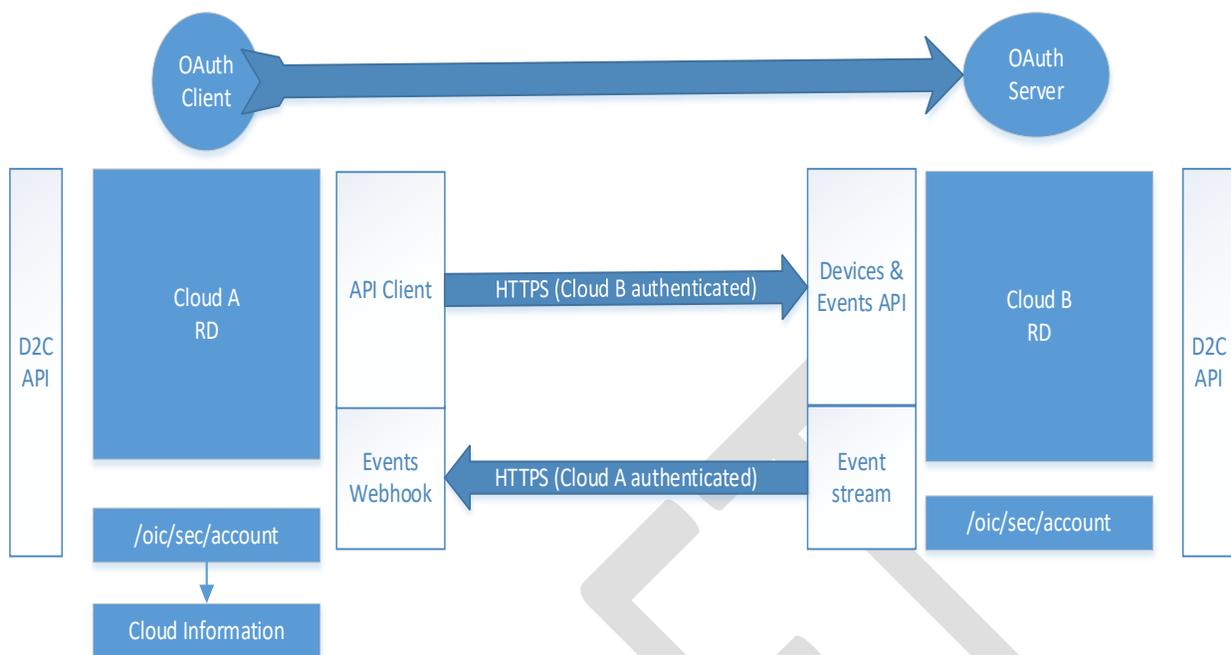
356 **5.3 Cloud to Cloud Operational Overview**

357 **5.3.1 Introduction**

358 This clause provides an informative overview of the flows that are enabled by the detailed API
359 defined in clauses 6, 7, 8, and 9. Clause 5.3 provides references to the applicable clauses within
360 this document that define the API specifics.

361 **5.3.2 Conceptual Architecture**

362 Figure 2 describes the overall conceptual architecture.



363
364 **Figure 2 – Conceptual Architecture**

365 **5.3.3 Authorizing Cloud Connectivity**

366 Consider a user who has accounts on two distinct, separately owned clouds, and devices
367 associated with each of those accounts on those clouds. The user wants to have a unified view of
368 all of their devices from a single client rather than having a client per cloud. The user via the client
369 they want to use for all devices indicates to the directly connected cloud (Origin Cloud) that they
370 want to link this account with an account on the other cloud (Target Cloud). This initiates a standard
371 OAuth2.0 Authorization Code Grant Type flow, see IETF RFC 6749, clause 1.3.1. Application of
372 this flow is described in clause 7.

373 **5.3.4 Synchronization of User's set of Devices**

374 After completion of the Authorization Code Grant Type flow from clause 5.3.3 the Origin Cloud (that
375 is the cloud to which the user is connected) is authorized to use the Device API to obtain on behalf
376 of the user the complete list of devices hosted on the Target Cloud for which the user has access.
377 The API is described in clause 8, and the flow is further illustrated in clause A.4.

378 The result of the invocation of the Device API is a complete set of device information that may then
379 be provided in a response to a RETRIEVE on "/oic/res" from the Origin Cloud.

380 **5.3.5 Keeping Up-to-Date: Notifications of changes on other Clouds**

381 Once the set of devices has been obtained, the Origin Cloud can subscribe to the events to which
382 it is interested across the user's complete device set ("/devices"), or per device in that set
383 ("/devices/{deviceid}"). See clause 9 for details of the API itself.

384 The subscription to "/devices" enables the Origin Cloud to be notified whenever a new device is
385 added or an existing device removed from the Target Cloud.

386 The subscription to "/devices/{deviceid}" enables the Origin Cloud to be notified whenever there is
387 a change in the state of a device (e.g. it has de-registered).

388 When a new Device registers on the Target Cloud, and a subscription exists for that event, then a
389 notification is sent to the Origin Cloud with an event type of "devices_registered" and a payload

390 which contains the "di" of the newly registered device. The Origin Cloud may then RETRIEVE the
391 Links exposed by the newly added device using "/devices/{deviceid}" where "deviceid" was
392 provided in the payload of the notification. See clause A.10 for a flow illustrating this interaction.

393 **5.3.6 Handling of Requests and Responses for Connected Devices**

394 From the perspective of the client connected to the Origin Cloud there is no distinction between
395 devices and their Resources hosted by the Origin Cloud itself and devices and their resources that
396 are hosted by a Target Cloud reached via this API.

397 Thus all requests for a target resource are formed using the mechanisms described in the OCF
398 Device to Cloud Services Specification.

399 The Origin Cloud identifies the Target Cloud for the requested Resource via the "deviceid" that is
400 in the request URI which is matched to the "di" Property in "/oic/sec/account". The request is then
401 effectively proxied to the Target Cloud via the "/devices/{deviceid}/{resourcehref}" API exposed by
402 the Target Cloud (see clause 8.5 and 8.6). Any query parameters received over the device to cloud
403 connection are included in the URI unaltered. The content-type of the payload in the request or
404 response is honoured. See clauses A.6 and A.7 for illustrative flows of this mechanism for both
405 RETRIEVE and UPDATE cases.

406 **6 Authentication & Authorization**

407 A Target Cloud shall only expose secure endpoints; any requests received over an unsecured
408 connection (i.e. HTTP) shall be redirected to the secure equivalent of that endpoint. The Origin
409 Cloud shall use the "Bearer" authentication scheme inside the "Authorization" request header field
410 to transmit the access token, as per IETF RFC 6750 clause 2.1. For definition of the "Authorization"
411 request header field, see IETF RFC 2818.

412 Bearer Tokens issued by the Target Cloud shall identify the user as well as the client that is sending
413 requests on behalf of the user to the Target Cloud.

414 On the OCF Server side there is no distinction between requests forwarded from the Origin Cloud
415 and requests coming via the Target Cloud.

416 **7 Account Linking API**

417 **7.1 General**

418 The account linking API is the mechanism by which Devices hosted on behalf of a user by the
419 Target Cloud are linked with a user identity on the Origin Cloud. Account linking is established
420 solely between the Origin Cloud and the Target Cloud; an Origin Cloud shall not proxy devices
421 from the Target Cloud to another third-party Cloud.

422 The OAuth 2.0 Origin Cloud Client has to be registered with the Target Cloud as a prerequisite to
423 initiating the Authorization Code Grant Type flow, which allows the user to link his Origin Cloud
424 account with the Target Cloud. This process is named OAuth Application registration and is beyond
425 the scope of this specification. Successful registration of the OAuth 2.0 Origin Cloud Client in the
426 Target Cloud relies on the two entities establishing trust and obtaining the required client
427 parameters and OAuth2.0 Token Endpoints (e.g. client id, client secret, allowed redirect URIs).
428 See IETF RFC 6749, clause 2.

429 The linking is then achieved via the use of an OAuth2.0 Authorization Code Grant Type. Part of the
430 linking process is the end-user consent, which is very important in cross-domain identity federation,
431 ensuring that a malicious OAuth 2.0 Origin Cloud Client cannot obtain authorization without the
432 awareness and explicit consent of the resource owner (that is the user) of the Target Cloud. The
433 Target Cloud presents to the user linking the account the precise scope of authorization information
434 being requested by the Client. Details about scopes are available in clause 7.2. After the user's

435 consent and subsequent authorization code exchange, the Bearer Token and refresh tokens (see
436 IETF RFC 6749) shall be obtained from the Target Cloud by the Origin Cloud, following the format
437 and Content Type in IETF RFC 6750 clause 4. The Bearer Token identifies a user identity on the
438 Target Cloud. All requests for a Bearer Token or a refresh token shall include the "client_id" and
439 "client_secret" as defined by IETF RFC 6749. IETF RFC 6749 clause 2.3.1 describes two schemes
440 for inclusion of the "client_id" and "client_secret", one using an Authorization header with a "Basic"
441 scheme, and one that encodes the client credentials in the request body which is not recommended
442 by the referenced RFC. **A Client shall provide an Authorization header in requests using the "Basic"
443 scheme, a Client should not encode the information in the request body.**

444 A Target Cloud may make use of the "offline_access" scope as defined by IETF RFC 7628, in such
445 an instance a Client requesting a token from such a Target Cloud shall include the scope in the
446 token request. How a Client determines what scopes the Target Cloud does or does not support
447 is outside of the scope of this document.

448 The "state" query parameter shall be present in each authorization request, see IETF RFC 6749
449 clause 4.1.1. State is an opaque value used by the Origin Cloud Client to maintain state between
450 the request and the callback during the account linking process, see clause A.3.

451 All requests, responses, and error codes that may be sent during Account Linking shall conform to
452 those defined in RFC 6749.

453 Once such a Bearer Token has been acquired, the Origin Cloud shall link the OAuth2.0 access and
454 refresh token with its known local "userid". The user who linked his Target Cloud account with the
455 Origin Cloud account is from this moment able to request all his devices through the Origin Cloud,
456 because the Origin Cloud can make requests to the Target Cloud on behalf of the Target Cloud
457 user account. However, if an Origin Cloud makes a request that is not included in the OAuth2.0
458 Access Token Scope granted by the Bearer Token, the Target Cloud shall reply with an appropriate
459 error response.

460 When a Bearer Token is first acquired, it is recommended that the Origin Cloud use the Device API
461 to retrieve the Device details for all Devices in OAuth2.0 Access Token Scope of the Bearer Token.

462 If the Origin Cloud supports the behaviour defined in the OCF Device to Cloud Services
463 Specification, then once the Origin Cloud has the set of Devices from the Target Cloud it creates
464 an instance of "/oic/sec/account" per Device. The optional Property "cloudid" in "/oic/sec/account"
465 is set to the OCF Cloud UUID of the Target Cloud available in the Common Name field of the End-
466 Entity certificate. If the Property is missing, empty, or contains the same value as the UUID of the
467 Origin Cloud, then the Device is local to the Origin Cloud.

468 The Origin Cloud may use the Events API to establish a subscription with the Device(s) on the
469 Target Cloud; such that addition or deletion of Devices on the Target Cloud can be correctly
470 reflected in the Origin Cloud. When the Device is deregistered from the Target Cloud, that Device
471 is no longer accessible via the Origin Cloud. When the Bearer Token obtained from the Target
472 Cloud expires and the refresh token is still valid, the Origin Cloud may ask for a new Bearer Token
473 through the OAuth2.0 token endpoint of the Target Cloud. Whenever the refresh token expires, is
474 not available, or the Bearer Token cannot be obtained, the Origin Cloud shall remove all
475 associations with the Devices hosted by the Target Cloud. See IETF RFC 6749 for further details.

476 It is recommended that the Origin Cloud subscribe to events of every Device that is hosted on the
477 Target Cloud by using the subscription mechanism described in clause 9.6.

478 **7.2 OAuth2.0 Access Token Scopes**

479 This document defines a core set of OAuth2.0 Access Token Scopes, see IETF RFC 6749. An
480 Origin Cloud may request one or more of these scopes, a vendor extension thereof, or a vendor
481 specific scope(s) as part of the account linking process. If the scope being provided by the Target
482 Cloud is different from the requested scope, then that scope shall be included in the issued Access

483 Token (see clause 5.1 of IETF RFC 6749). If the Target Cloud supports Access Token requests
 484 with no scopes provided, and an Access Token request with no scopes is received from the Origin
 485 Cloud, then the returned Access Token from the Target Cloud shall grant access to all of the
 486 OAuth2.0 Access Token Scopes defined in Table 1.

487 The description for each of the OAuth2.0 Access Token Scopes shall be presented to the user
 488 during the account linking process by the OAuth2.0 server of the Target Cloud. The Target Cloud
 489 user sees a description on the consent screen and give an explicit consent that the Origin Cloud
 490 requesting that the Bearer Token is authorized to act on behalf of the user in the boundary of
 491 obtained OAuth2.0 Access Token Scopes.

492 **Table 1 – OAuth 2.0 AccessToken Scopes**

OAuth2.0 Access Token Scope name	OAuth2.0 Access Token Scope description "The application will be able to:"
r:*	Read
w:*	Update

493
 494 Table 2 details the OAuth2.0 Access Token Scopes that are applicable per API Endpoint. All API
 495 Endpoints that are listed in Table 2 shall be supported by a Target Cloud. So, for example, if an
 496 Origin Cloud sends a GET request to "/api/v1/devices?content=all" API Endpoint, the Origin Cloud
 497 must have a Bearer Token that contains OAuth2.0 Access Token Scope "r:*" or a vendor extension
 498 thereof

499 **Table 2 – Applicable OAuth2.0 Access Token Scopes per API Endpoint**

API Endpoint	HTTP Request Type	Applicable scopes
/api/v1/devices	GET	r:*
/api/v1/devices?content=all	GET	r:*
/api/v1/devices/{deviceid}	GET	r:*
/api/v1/devices/{deviceid}?content=all	GET	r:*
/api/v1/devices/{deviceid}/{resourcehref}	GET	r:*
	POST	w:*
/api/v1/devices/subscriptions	POST	r:*
	DELETE	r:*
/api/v1/devices/{deviceid}/subscriptions	POST	r:*
	DELETE	r:*
/api/v1/devices/{deviceid}/{resourcehref}/subscriptions	POST	r:*
	DELETE	r:*

500 A vendor may extend the list of OAuth2.0 Access Token Scopes beyond those listed in Table 2.
 501 They are extended by adding additional vendor-specific information before the * in the OAuth2.0
 502 Access Token Scope name (e.g. "r:xyz:*"). How these extensions work is outside the scope of the
 503 OCF but they may be present in the OAuth2.0 Access Token request. Note that if the user gives
 504 consent to the Origin Cloud to "w:*", consent applies also to any derived OAuth2.0 Access Token
 505 Scopes (e.g. "w:xyz:*").

506 **8 Devices API**

507 **8.1 Introduction**

508 The Devices API supports the ability to retrieve and interact with the OCF Devices that are within
509 the scope of the provided Bearer Token.

510 **8.2 Parameters Supported in Requests**

511 Table 3 lists the parameters that may be provided as part of a request within the Device API.

512 **Table 3 – Parameters used in Requests in the Device API**

Friendly Name	Parameter Name	Location	Mandatory	Description
Accept	Accept	HTTP Header	Yes	An Accept request HTTP header advertises which content types, expressed as MIME types, the Origin Cloud is able to understand. The Target Cloud then selects one of the proposed content types and informs the Origin Cloud of its choice with the Content-Type response header.
Content Type	Content-Type	HTTP Header	No	The Content-Type header is used to indicate the media type of the payload. A Content-Type header tells the recipient what the content type of the returned payload actually is.
Correlation ID	Correlation-ID	HTTP Header	No	A Correlation ID, also known as a Transit ID, is a unique identifier value that is attached to requests and messages that allows reference to a particular transaction or event chain.
Content	content=[base, all]	Query String Parameter	No	When set to "base" this indicates to the recipient that the response payload Links are not resolved. When set to "all" this indicates to the recipient that the response payload is the resolved (i.e. resource representation) Link and not the Link itself. If not present "base" is assumed.

513

514 **8.3 Retrieve All Devices**

515 **8.3.1 Summary**

516 This request is sent from the Origin Cloud to the Target Cloud in order to obtain information on all
517 the Devices that are registered for the user that are in scope as defined by the Bearer Token on
518 the Target Cloud.

519 A request to this API may be invoked by the Origin Cloud on completion of account linking. Where
520 the Cloud supports the behaviour defined in the OCF Device to Cloud Services Specification this
521 may also be invoked by reception of a RETRIEVE to "/oic/res" of the Cloud Resource Directory
522 from an OCF Client.

523 Table 4 provides a summary of the API.

524

Table 4 – Retrieve All Devices API Summary

HTTP Request Type	API Endpoint	Parameters	Response Code	Response Payload
GET	/api/v1/devices	content=[base, all], Correlation-ID, Accept	200	See clause B.1 - array of /definitions/Device (for content=base) and /definitions/DeviceContentAll (for content=all)
			400, 401, 403, 503, 504	The response may include a diagnostic payload containing a reason string.

525 **8.3.2 Request and Response Payload**

526 There is no required payload in the request; if one is received at the Target Cloud it shall be ignored.
 527 The required response payload for a request that includes "content=base" or no "content"
 528 parameter shall be an array of objects; each object shall contain the Properties identified in the
 529 schema provided in Annex B, a "device" Property as defined by the schema, a status Property
 530 ("status") that indicates whether the Device is online or offline, and an array of Links (as defined
 531 for "/oic/res") for the Resources exposed by the specific Device. These Properties are further
 532 summarised in Table 5, with the specific Properties in the "device" Property summarised in Table 6.

533

Table 5 – Response payload Property definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Device	"device"	"object"	N/A	N/A	R	Yes	Set of Properties that defined the Device itself; see Table YYYY
Device Status	"status"	"string:"	Value from the enumeration {"online","offline"}	N/A	R	Yes	Status of the Device.
Device Links	"links"	"array"	N/A	N/A	R	Yes	The published set of Links exposed by the Device

534

535

Table 6 – "device" Property definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Resource Type	"rt"	"array"	N/A	N/A	R	Yes	Array contained the Device Type of the Device
(Device) Name	"n"	"string:"	N/A	N/A	R	Yes	Human friendly name defined by the vendor.
Device ID	"di"	"uuid"	N/A	N/A	R	Yes	Unique identifier for Device.
Manufacturer Name	"dmn"	"array"	N/A	N/A	R	Yes	Name of manufacturer of the Device, in one or more languages. This Property is an array of objects where each object has a "language" field (containing an IETF RFC 5646 language tag) and

							a "value" field containing the manufacturer name in the indicated language.
--	--	--	--	--	--	--	---

536
537 The minimum set of Resources that are exposed depends on the OCF Device Type of the Device;
538 this shall be the set defined in clause 6.1.3.2.1 of the OCF Device to Cloud Services Specification.

539 If the request includes "content=all" (analogous to a batch retrieval of /oic/res in the proximal
540 network) then the response payload shall be as defined for "content=base" with the exception that
541 instead of an array of Links to the hosted Resources, the response payload shall include an array
542 of the representations of the Resources themselves that are exposed for each Device that is
543 available. This is illustrated in the examples provided for the Device API in Annex B. See also the
544 definition of a batch response in ISO/IEC 30118-1:2018.

545 **8.3.3 Responses**

546 A 200 response shall be provided in a success case. The payload shall contain information for all
547 Devices that are in the scope of the Bearer Token.

548 A non-success path response that is indicative of the type of error shall be returned by a Target
549 Cloud if an error scenario is detected. Table 7 lists possible non-success path responses and
550 possible scenarios that trigger their generation; an implementation may support additional
551 responses as defined by IETF RFC 2818.

552 **Table 7 – Devices API non-success path responses**

Response Code	Response scenario
400	May be sent by the Target Cloud if the request was malformed or badly constructed
401	May be sent by the Target Cloud if the request is unauthorized (e.g. an invalid or missing Bearer Token)
403	May be sent by the Target Cloud if the requestor is known however the OAuth2.0 Access Token Scope of the request is forbidden
406	May be sent by the Target Cloud if the media type in the received Accept header is not supported/acceptable
503	May be sent by the Target Cloud if the service on the Target Cloud is unavailable
504	May be sent by the Target Cloud if the target Device is registered at the Target Cloud, however the Device itself is unavailable, offline, or otherwise unreachable. The response should include a Retry-After header containing the time after which the request may be re-attempted. Additional information may be indicated in a diagnostic payload

553
554 **8.4 Retrieve One Device**
555 **8.4.1 Summary**
556 This request may be sent from the Origin Cloud to the Target Cloud in order to obtain information
557 on a specific Device that is registered for the user that is in scope as defined by the Bearer Token
558 on the Target Cloud.

559 A request to this API may be invoked at the Origin Cloud following reception of a notification that
560 a new Device has been added to a partner cloud, or alternatively as part of the flow following
561 account linking. Where the Origin Cloud supports the OCF Device to Cloud Services Specification,
Copyright Open Connectivity Foundation, Inc. © 2020. All rights Reserved

562 a request to this API may also be invoked following reception of a RETRIEVE to "/oic/res" of the
 563 Origin Cloud Resource Directory from an OCF Client with a query parameter that specifies a
 564 particular "deviceid" (i.e. "?anchor=ocf://<some device id>").

565 Table 8 provides a summary of the API.

566 **Table 8 – Retrieve One Device API Summary**

HTTP Request Type	API Endpoint	Parameters	Response Code	Response Payload
GET	/api/v1/devices/{deviceid}	content=[base, all], Correlation-ID, Accept	200	See clause B.1 - /definitions/Device (for content=base) and /definitions/DeviceContentAll (for content=all)
			400, 401, 403, 404, 503, 504	The response may include a diagnostic payload containing a reason string

567

568 **8.4.2 Request and Response Payload**

569 There is no required payload in the request; if one is received at the Target Cloud it shall be ignored.

570 The "deviceid" in the URI of the request is the same as the "di" Property from /oic/d of the target
 571 OCF device.

572 The response payload shall be an object containing the mandatory Device information as defined
 573 in clause 8.3.2.

574 **8.4.3 Responses**

575 A 200 response shall be provided in a success case. The payload shall contain information for the
 576 requested Device.

577 A non-success path response that is indicative of the type of error shall be returned by a Target
 578 Cloud if an error scenario is detected. Table 9 lists possible non-success path responses and
 579 possible scenarios that may trigger their generation; an implementation may support additional
 580 responses as defined by IETF RFC 2818.

581 **Table 9 – Device API non-success path responses**

Response Code	Response scenario
400	May be sent by the Target Cloud if the request was malformed or badly constructed
401	May be sent by the Target Cloud if the request is unauthorized (e.g. an invalid or missing Bearer Token)
403	May be sent by the Target Cloud if the requestor is known however the OAuth2.0 Access Token Scope of the request is forbidden
404	May be sent by the Target Cloud if the indicated "deviceid" is not present on the Target Cloud
406	May be sent by the Target Cloud if the media type in the received Accept header is not supported/acceptable

503	May be sent by the Target Cloud if the service on the Target Cloud is unavailable
504	May be sent by the Target Cloud if the target Device is registered at the Target Cloud, however the Device itself is unavailable, offline, or otherwise unreachable. The response should include a Retry-After header containing the time after which the request may be re-attempted. Additional information may be indicated in a diagnostic payload

582

583 8.5 Retrieve Specific Resource

584 8.5.1 Summary

585 This request is sent from the Origin Cloud to the Target Cloud in order to obtain information on a
 586 specific Resource that is exposed by a Device that is registered for the user that is in scope as
 587 defined by the Bearer Token on the Target Cloud.

588 Where the Cloud supports the OCF Device to Cloud Services Specification this may be triggered
 589 by reception of a RETRIEVE to a URI exposed by a Link in the Cloud Resource Directory from an
 590 OCF Client.

591 Table 10 provides a summary of the API.

592

Table 10 – Retrieve Specific Resource API Summary

HTTP Request Type	API Endpoint	Parameters	Response Code	Response Payload
GET	/api/v1/devices/{deviceid}/{resourcehref}	Correlation-ID, Accept	200	Response payload as defined by OCF for the target Resource Type
			400, 401, 403, 404	The response may include a diagnostic payload containing a reason string
			503	The response may include a diagnostic payload containing a reason string
			504	Retry-After header and optionally a diagnostic payload containing a reason string.

593

594 8.5.2 Request and Response Payload

595 There is no required payload in the request; if one is received at the Target Cloud it shall be ignored.

596 The "deviceid" in the URI in the request is the same as the "di" Property from "/oic/d" of the target
 597 OCF device. The "resourcehref" in the URI is the same as the "href" Link Parameter for the target
 598 Resource instance.

599 The response payload shall be as defined by OCF for the Resource being received, or as defined
 600 by the vendor if the Resource is a 3rd party Resource.

601 The content-type of the response payload received from the target server is honoured; that is the
602 content and payload as received by the Target Cloud shall be proxied unaltered in the response.
603 Thus for example in the case where the target server is an OCF Device the content type would be
604 "application/vnd.ocf+cbor".

605 An Origin Cloud shall include unaltered in the requestURI of the request sent to the Target Cloud
606 any query parameters received over the device to cloud connection.

607 **8.5.3 Responses**

608 A 200 response shall be provided in a success case. The payload in the response shall be as
609 defined in <http://oneiota.org> for the target Resource Type.

610 A non-success path response that is indicative of the type of error shall be returned by a Target
611 Cloud if an error scenario is detected. Table 11 lists possible non-success path responses and
612 possible scenarios that may trigger their generation; an implementation may support additional
613 responses as defined by IETF RFC 2818.

614 **Table 11 – Resource Retrieval API non-success path responses**

Response Code	Response scenario
400	May be sent by the Target Cloud if the request was malformed or badly constructed
401	May be sent by the Target Cloud if the request is unauthorized (e.g. an invalid or missing Bearer Token)
403	May be sent by the Target Cloud if the requestor is known however the OAuth2.0 Access Token Scope of the request is forbidden
404	May be sent by the Target Cloud if the indicated "deviceid" is not present on the Target Cloud or the "resourcehref" is not found
406	May be sent by the Target Cloud if the media type in the received Accept header is not supported/acceptable
503	May be sent by the Target Cloud if the service on the Target Cloud is unavailable
504	May be sent by the Target Cloud if the target Device is registered at the Target Cloud, however the Device itself is unavailable, offline, or otherwise unreachable. The response should include a Retry-After header containing the time after which the request may be re-attempted. Additional information may be indicated in a diagnostic payload

615

616 **8.6 Update a Resource on a Device**

617 **8.6.1 Summary**

618 This request is sent from the Origin Cloud to the Target Cloud in order to update information
619 contained within a specific Resource exposed by a Device that is registered for the user that is in
620 scope as defined by the Bearer Token on the Target Cloud.

621 Where the Cloud supports the OCF Device to Cloud Services Specification a request to this API
622 may be triggered by reception of an UPDATE to a URI exposed by a Link in the Cloud Resource
623 Directory from an OCF Client.

624 Table 12 provides a summary of the API.

Table 12 – Update Resource API Summary

HTTP Request Type	API Endpoint	Parameters	Response Code	Response Payload
POST	/api/v1/devices/{deviceid}/{resourcehref}	payload, Correlation-ID, Accept, Content-Type	200	Optional resource representation
			400, 401, 403, 404, 415	The response may include a diagnostic payload containing a reason string.
			503	The response may include a diagnostic payload containing a reason string.
			504	Retry-After header and optionally a diagnostic payload containing a reason string

626

8.6.2 Request and Response Payload

628 The request payload shall be as defined by OCF for the Resource being updated, or as defined by
629 the vendor if the Resource is a 3rd party Resource.

630 The "deviceid" in the URI in the request is the same as the "di" Property from /oic/d of the target
631 OCF device. The "resourcehref" in the URI is the same as the "href" Link Parameter for the target
632 Resource instance.

633 The response payload shall be as defined by OCF for the Resource being received, or as defined
634 by the vendor if the Resource is a 3rd party Resource.

635 The Content-Type of the request is defined in an HTTP Content-Type header. In the case that the
636 request was initiated by another OCF Device, the CoAP content-format header value shall be
637 mapped to the HTTP Content-Type header to the Target Cloud. If the value is not present, the
638 Target Cloud shall forward the request as-is. Thus, for example, in the case where the origin client
639 is an OCF Device, the CoAP content-format option would be "application/vnd.ocf+cbor", which is
640 passed to the Target Cloud as an HTTP Content-Type header.

641 An Origin Cloud shall include unaltered in the requestURI of the request sent to the Target Cloud
642 any query parameters received over the device to cloud connection.

8.6.3 Responses

644 A 200 response shall be provided in a success case. The payload may optionally contain the
645 representation of the Resource that was updated.

646 A non-success path response that is indicative of the type of error shall be returned by a Target
647 Cloud if an error scenario is detected. Table 13 lists possible non-success path responses and
648 possible scenarios that may trigger their generation; an implementation may support additional
649 responses as defined by IETF RFC 2818.

Table 13 – Resource Update API non-success path responses

Response Code	Response scenario
400	May be sent by the Target Cloud if the request was malformed or badly constructed
401	May be sent by the Target Cloud if the request is unauthorized (e.g. an invalid or missing Bearer Token)
403	May be sent by the Target Cloud if the requestor is known however the OAuth2.0 Access Token Scope of the request is forbidden
404	May be sent by the Target Cloud if the indicated "deviceid" is not present on the Target Cloud or the "resourcehref" is not found
406	May be sent by the Target Cloud if the media type in the received Accept header is not supported/acceptable
415	May be sent by the Target Cloud if an unsupported media type was specified in the Content-Type header
503	May be sent by the Target Cloud if the service on the Target Cloud is unavailable
504	May be sent by the Target Cloud if the target Device is registered at the Target Cloud, however the Device itself is unavailable, offline, or otherwise unreachable. The response should include a Retry-After header containing the time after which the request may be re-attempted. Additional information may be indicated in a diagnostic payload

651

652 9 Events API

653 9.1 Introduction

654 The Events API supports the ability for an interested party to subscribe to events and subsequently
 655 receive notifications for those events. The events can be at the Resource level (like a CoAP
 656 observe) or at a more system level (such as for a change in the set of known Devices).

657 The Events API makes use of a mechanism whereby the Target Cloud notifies the Origin Cloud
 658 when a new event has occurred on the Target Cloud or any Device linked with the Target Cloud.
 659 This event stream (continual series of notifications) may be started by sending an initial
 660 subscription request to the Target Cloud specifying "eventTypes", "eventsUrl" (the API Endpoint to
 661 which notifications are sent), and the "signingSecret", the latter to verify whether requests from the
 662 Target Cloud are authentic. See clause 9.2. for details on the mechanism for how the
 663 "signingSecret" is used and clause 9.4.1 for details on the subscription request.

664 A Subscription ID shall be provided in the response to an initial subscription request. The
 665 Subscription ID is a unique string of type UUID, which shall be created and persisted by the Target
 666 Cloud. The created ID shall be part of each notification sent to the configured "eventsUrl". The
 667 Subscription ID shall also be used to DELETE this subscription. The Subscription ID is either
 668 present in a response payload, or within a HTTP header, or present as part of the request URI
 669 depending on the operation being undertaken. See clauses 9.4.2 and 9.4.3 for more details.

670 After the subscription is successful, the Target Cloud shall send an initial notification to the Origin
 671 Cloud "eventsUrl" (that was provided during establishment of the subscription) with the current
 672 state of the items to which the subscription applies. The Target Cloud shall send further
 673 notifications to the Origin Cloud whenever any changes occur (i.e. events) to the items to which
 674 the subscription applies.

675 Following the Origin Cloud's successful subscription to events of the Target Cloud, the Target
676 Cloud shall start sending notifications only after it establishes a new server-authenticated TLS
677 connection to the "eventsUrl" as specified by the Target Cloud.

678 Notifications generated by the Target Cloud in response to a subscription shall only be for devices
679 and system changes the Bearer Token authorizes.

680 **9.2 Events Authentication**

681 Hash-based Message Authentication Code (HMAC) signatures are a way to sign the notification
682 data using the "signingSecret" that only the Origin Cloud and Target Cloud know. The
683 "signingSecret" shall be created by the Origin Cloud and sent within the subscription request as
684 defined in the clause 9.4.1. After a successful subscription, the Target Cloud shall sign each
685 notification using the HMAC-SHA256 hashing algorithm following the formula from the clause 9.2.1.
686 The calculated signature shall be attached as the "Event-Signature" header with each notification
687 request sent to the Origin Cloud.

688 The signature shall be used by the Origin Cloud to verify the legitimacy of the source and data
689 itself. When the notification is received by the Origin Cloud it shall use its stored secret and the
690 notification to generate its own HMAC-SHA256 signature using the formula from the clause 9.2.2
691 to compare with the value from the "Event-Signature" header.

692 When the signing secret and notification request are the same on both sides then the HMAC
693 signature will match. This match proves the authenticity of the request and data.

694 When the HMAC signature does not match, the Origin Cloud shall ignore the notification request
695 message.

696 Detailed overview is provided in figures 0, 0, 0, and 0.

697 **9.2.1 Create Event Signature**

698 1) Get the current timestamp in the Unix time format; this is used to populate the "Event-Timestamp"
699 header.

700 2) Create a string, that is made up of the concatenation of the encoded content of the following
701 headers that are part of the notification that is to be sent, in order: "Content-Type", "Event-Type",
702 "Subscription-ID", "Sequence-Number", and "Event-Timestamp". Between each value insert a
703 colon (ASCII character value hex 3A) as a delimiter. If any one of the headers is not present, do
704 not include that value but still include the delimiter (e.g. if "Content-Type" is not present include a
705 ":" prior to encoding the "Event-Type"). All headers that are defined to be strings shall be handled
706 as ASCII characters.

707 3) After the encoding for "Event-Timestamp" add a final colon (ASCII character value hex 3A) and
708 the (i.e. as would be included in the HTTP request) raw bytes that make up the to be sent
709 notification body.

710 4) Hash the resulting string, using the "signingSecret" as a key using the HMAC-SHA256 hashing
711 algorithm, and taking the hex digest of the hash.

712 5) Include the resulting signature to the "Event-Signature" header of the notification and timestamp
713 to the "Event-Timestamp" header

714 **9.2.1 Verify the Event Signature**

715 1) Create a string, that is made up of the concatenation of the encoded content of the following
716 headers received in the notification, in order: "Content-Type", "Event-Type", "Subscription-ID",
717 "Sequence-Number", and "Event-Timestamp". Between each value insert a colon (ASCII character
718 value hex 3A) as a delimiter. If any one of the headers is not present, do not include that value but

719 still include the delimiter (e.g. if "Content-Type" is not present include a ":" prior to encoding the
720 "Event-Type"). All headers that are defined to be strings shall be handled as ASCII characters.

721 2) After the encoding for "Event-Timestamp" add a final colon (ASCII character value hex 3A) and
722 the received raw bytes (i.e. not subject to any decode) of the notification body.

723 3) Hash the resulting string, using the "signingSecret" as a key using the HMAC-SHA256 hashing
724 algorithm and take the hex digest of the hash.

725 4) Compare the resulting signature to the "Event-Signature" header of the received notification

726 9.3 Parameters Supported

727 Table 14 lists the parameters that may be provided within the Events API.

728 **Table 14 – Parameters used in the Events API**

Friendly Name	Parameter Name	Location	Mandatory	Description
Accept	Accept	HTTP Header	Yes	An Accept request HTTP header advertises which content types, expressed as MIME types, the client is able to understand. The resource server then selects one of the proposal and informs the client of its choice with the Content-Type response header. Each notification sent to the defined "eventsUrl" is then using this Accepted content type.
Correlation ID	Correlation-ID	HTTP Header	No	A Correlation ID, also known as a Transit ID, is a unique identifier value that is attached to requests and responses that allows reference to a particular transaction or notification.
Content Type	Content-Type	HTTP Header	No	The Content-Type header is used to indicate the media type of the payload. A Content-Type header tells the recipient what the content type of the returned payload actually is.

729 9.4 Events API subscription and notification payload definitions

730 9.4.1 Subscription request

731 A subscription request is sent by an Origin Cloud to the API Endpoint defined for the event type to
732 which the subscription is targeted. The set of event types and associated API Endpoints is provided
733 in Table 15. A Target Cloud should support "resources_published" and "resources_unpublished"
734 event types, a Target Cloud shall support all other event types listed in Table 15. If for whatever
735 reason a Target Cloud cannot honour the subscription request to an event type, it shall respond
736 with an appropriate non-success path final response.

737 Subscription to a "subscription_cancelled" event type is not done explicitly by an Origin Cloud; it
738 shall always be enabled at the Target Cloud whenever any other supported event type is the target
739 of a subscription.

740 **Table 15 – Event types and API Endpoints**

Event-Type	API Endpoint
subscription_cancelled	N/A as a subscription_cancelled event type is not explicitly subscribed to.
devices_registered	/api/v1/devices/subscriptions

devices_unregistered	/api/v1/devices/subscriptions
devices_online	/api/v1/devices/subscriptions
devices_offline	/api/v1/devices/subscriptions
resource_contentchanged	/api/v1/devices/{deviceid}/{resourcehref}/subscriptions
resources_published	/api/v1/devices/{deviceid}/subscriptions
resources_unpublished	/api/v1/devices/{deviceid}/subscriptions

741

742 Annex B provides a definition of the payload contained within the subscription request. The
 743 Properties that are contained in the payload are further clarified in Table 16.

744

Table 16 – Subscription Request Payload Properties

Payload Property Name	Value type	Mandatory	Description
eventsUrl	URI	Y	URI to which notifications are to be sent
eventTypes	array of enum	Y	Event type(s) for which the subscription is targeted. See Table 15
signingSecret	String of length 32	Y	Secret used to create HMAC signature for each event

745

746 Figure 3 is an example of such a payload.

```
{
  "eventsUrl": "https://mynotificationuri",
  "eventTypes": ["resource_contentchanged"],
  "signingSecret": "DVDUEBe5nciVSXU85BPxrAjsHentzWY"
}
```

747

Figure 3 – Subscription Request Example

748

9.4.2 Subscription response

749 The definition of the response to a subscription request is in Annex B. The Properties that are
 750 contained with the payload are further clarified in Table 17.

751

Table 17 – Subscription Response Properties

Payload Property Name	Value type	Mandatory	Description
subscriptionId	Uuid	Y	Identity of the subscription (the Subscription ID). May be mapped from other protocols if a unique identifier exists. Note this cannot be mapped from a CoAP Token as the Token in CoAP is Client-local in

		scope (i.e. not guaranteed unique beyond the Client issuing the request).
--	--	---

752

753 Figure 4 is an example of such a payload.

```
{
  "subscriptionId": "1eeb465c-5e8d-4305-a366-bbf035fff671"
}
```

754

Figure 4 – Subscription Response Example Payload

755 **9.4.3 Notification request**

756 When a subscription is first successfully established, the Target Cloud shall send a POST request
 757 to the "eventsUrl" that was provided in the subscription with the current state of the items to which
 758 the subscription applies. There shall be one POST request per subscribed event type; that is, if a
 759 subscription request contains multiple event types in the "eventTypes" Property, there is a
 760 notification request per identified event type, not one for all event types.

761 When there is a subsequent change (i.e. an event) that triggers a notification, the Target Cloud
 762 shall send a POST request to the "eventsUrl" that was provided in the subscription. The Target
 763 Cloud shall populate all headers defined in Table 18 in the POST that is sent to the "eventsUrl"
 764 provided by the Origin Cloud together with any notification payload.

765 The Target Cloud shall send a notification with an event type of "subscription_cancelled" to the
 766 "eventsUrl" provided by the Origin Cloud if there is a cancellation of the subscription. As there is
 767 no defined payload for a "subscription_cancelled" event, a POST request that is sent for this event
 768 type shall not include a "Content-Type" header. The cancellation may be through reception of a
 769 DELETE from the Origin Cloud (see clauses 9.5, 9.6, and 9.7) or through internal logic on the Target
 770 Cloud itself.

771 If the request that established the subscription contained a Correlation-ID header, then all
 772 notifications that are sent as a result of that subscription shall contain a Correlation-ID header
 773 populated with the same value as received in the original subscription request.

774

Table 18 – Notification request HTTP Headers

HTTP Header	Value Type	Mandatory	Description
Correlation-ID	UUID	No	A Correlation ID, also known as a Transit ID, is a unique identifier value that is attached to requests and responses that allows reference to a particular transaction or event chain.
Content-Type	String	Yes, for notifications that include a payload	Indicates the media type of the notification payload
Event-Type	String	Yes	Type of the event
Subscription-ID	UUID	Yes	Subscription identifier for which this notification is being sent
Sequence-Number	String encoded Integer	Yes	Sequence number of the notification; the first notification shall have a value of 0, this value shall be incremented by 1 (one) for all subsequent notifications
Event-Timestamp	Unix time format	Yes	Time when the event occurred in standard Unix time format

Event-Signature	String	Yes	HMAC-SHA256 signature proving the authenticity of the request and data. See 9.2 Events Authentication
------------------------	--------	-----	---

775
776 The format of the payload in a notification request depends on the event type for which the
777 subscription was created. Table 19 defines the format of the payload provided in a notification per
778 "eventType" (as received in the payload of the subscription request from the Origin Cloud) that may
779 be sent by the Target Cloud. A Target Cloud shall populate the notification payload for the event
780 type being signalled in the Event-Type HTTP header as defined in Table 19. The schema definitions
781 for all payloads are provided in Annex B.

782 **Table 19 – Event type to notification payload content**

Event-Type header population	Notification payload on establishment of the subscription	Notification payload per subsequent notification
subscription_cancelled	Not present	Not applicable
devices_registered	Array of all currently registered Device IDs	Array containing Device IDs that have been registered since the previous notification was sent.
devices_unregistered	Empty array (i.e. [])	Array containing Device IDs for devices that have been de-registered since the previous notification was sent
devices_online	Array of all currently online Device IDs	Array containing Device IDs that have come online since the previous notification was sent.
devices_offline	Array of all currently offline Device IDs	Array containing Device IDs for devices that have gone offline since the previous notification was sent
resource_contentchanged	Current Resource Representation of the target Resource	Payload of the changed Resource as received by the Target Cloud
resources_published	Array of Links of all published Resources for the Device ID in the path	Array of Links of all Resources published by the Device ID in the path since the previous notification was sent
resources_unpublished	Empty array (i.e. [])	Array of Links of all Resources unpublished by the Device ID in the path since the previous notification was sent

783
784 **9.4.4 Notification response**
785 If the Target Cloud receives a non-success path response to a notification request it shall treat the
786 response as indicative of a request to cancel the subscription, and no further notifications for the

787 Subscription ID that was in the request shall be sent. See clauses 9.8.3, 9.9.3, and 9.10.3 for
788 further information.

789 **9.5 Subscribe and unsubscribe to devices level event types**

790 **9.5.1 Summary**

791 This request is sent from the Origin Cloud to the Target Cloud. An Origin Cloud may use this API
792 when it wants to receive notifications of events generated due to changes to the set of Devices that
793 are exposed.

794 Event types that may be subscribed to using this API are: devices_registered,
795 devices_unregistered, devices_online and devices_offline.

796 An Origin Cloud may establish a subscription by sending a POST request to the API Endpoint
797 shown in Table 20. To remove an existing subscription an Origin Cloud shall send a DELETE
798 request to the API Endpoint as shown in Table 20.

799 Table 20 provides a summary of the API.

800

Table 20 – Subscription to /devices API Summary

HTTP Request Type	API Endpoint	Parameters	Response Code	Response Payload
POST	/api/v1/devices/subscriptions	Correlation-ID, Accept, Content-Type	201	See clause B.1 - /definitions/SubscribeResponse
			400, 401, 403	
DELETE	/api/v1/devices/subscriptions/{subscriptionId}	Correlation-ID	202	
			400, 401, 403, 404	

801 **9.5.2 Request and Response Payload**

802 The request payload for the POST shall be as defined in clause 9.4.1.

803 The "subscriptionId" in the URI for the DELETE case shall be the "subscriptionId" that was returned
804 in the response to the subscription POST request.

805 The response payload for the subscription POST request shall contain the Subscription ID in a
806 "subscriptionId" Property as defined in clause 9.4.2.

807 There is no required payload for a DELETE unsubscribe response.

808 **9.5.3 Responses**

809 A 201 response shall be sent by the Target Cloud in a success case.

810 A 202 response shall be sent by the Target Cloud following a DELETE request and indicates that
811 the subscription was marked for cancellation; confirmation of the cancellation of the subscription
812 shall be provided by a subsequent notification with an Event-Type of "subscription_cancelled".

813 A non-success path response that is indicative of the type of error shall be returned by a Target
814 Cloud if an error scenario is detected. Table 21 lists possible non-success path responses and
815 possible scenarios that may trigger their generation; an implementation may support additional
816 responses as defined by IETF RFC 2818.

Table 21 – Devices Event Subscription API non-success path responses

Response Code	Response scenario
400	May be sent by the Target Cloud if the request was malformed or badly constructed
401	May be sent by the Target Cloud if the request is unauthorized (e.g. an invalid or missing Bearer Token)
403	May be sent by the Target Cloud if the requestor is known however the OAuth2.0 Access Token Scope of the request is forbidden
404	May be sent by the Target Cloud if the subscription was not found or the subscribed to Event-Type is not supported
406	May be sent by the Target Cloud if the media type in the received Accept header is not supported/acceptable

818

819 **9.6 Subscribe and unsubscribe to device level events**820 **9.6.1 Summary**

821 This request is sent from the Origin Cloud to the Target Cloud. This API is used when the Origin
822 Cloud wants to receive notifications for a specific Device on the Target Cloud.

823 Event types that may be subscribed to using this API are: resources_published and
824 resources_unpublished.

825 An Origin Cloud may establish a subscription by sending a POST request to the API Endpoint
826 shown in Table 22. To remove an existing subscription an Origin Cloud shall send a DELETE
827 request to the API Endpoint as shown in Table 22.

828 Table 22 provides a summary of the API.

829 **Table 22 – Subscription to Single Device API Summary**

HTTP Request Type	API Endpoint	Parameters	Response Code	Response Payload
POST	/api/v1/devices/{deviceid}/subscriptions	Correlation-ID, Accept, Content-Type	201	See clause B.1 - /definitions/SubscribeResponse
			400, 401, 403, 404	
DELETE	/api/v1/devices/{deviceid}/subscriptions/{subscriptionId}	Correlation-ID	202	
			400, 401, 403, 404	

830 **9.6.2 Request and Response Payload**

831 The request payload for the POST shall be as defined in clause 9.4.1.

832 The "deviceid" in the request URI shall be the same as the "di" Property from "/oic/d" of the target
833 OCF device.

834 The "subscriptionId" in the URI for the DELETE case shall be the "subscriptionId" that was returned
835 in the response to the subscription POST request.

836 The response payload for the subscription POST request shall contain the Subscription ID in a
837 "subscriptionId" Property as defined in clause 9.4.2.

838 There is no required payload for a DELETE unsubscribe response.

839 9.6.3 Responses

840 A 201 response shall be sent by the Target Cloud in a success case.

841 A 202 response shall be sent by the Target Cloud following a DELETE request and indicates that
842 the subscription was marked for cancellation; confirmation of the cancellation of the subscription
843 shall be provided by a subsequent notification with an Event-Type of "subscription_cancelled".

844 A non-success path response that is indicative of the type of error shall be returned by a Target
845 Cloud if an error scenario is detected. Table 23 lists possible non-success path responses and
846 possible scenarios that may trigger their generation; an implementation may support additional
847 responses as defined by IETF RFC 2818.

848 **Table 23 – Device Event Subscription API non-success path responses**

Response Code	Response scenario
400	May be sent by the Target Cloud if the request was malformed or badly constructed
401	May be sent by the Target Cloud if the request is unauthorized (e.g. an invalid or missing Bearer Token)
403	May be sent by the Target Cloud if the requestor is known however the OAuth2.0 Access Token Scope of the request is forbidden
404	May be sent by the Target Cloud if the subscription was not found or the subscribed to Event-Type is not supported
406	May be sent by the Target Cloud if the media type in the received Accept header is not supported/acceptable

849 9.7 Subscribe and unsubscribe to resource level events

850 9.7.1 Summary

851 This request is sent from the Origin Cloud to the Target Cloud. This API may be used by the Origin
852 Cloud to receive notifications from a specific observable Resource that exists on a specific Device
853 on the Target Cloud.

854 Events that may be subscribed to using this API are: resource_contentchanged.

855 An Origin Cloud may establish a subscription by sending a POST request to the API Endpoint
856 shown in Table 15. To remove an existing subscription an Origin Cloud shall send a DELETE
857 request to the API Endpoint as shown in Table 24.

858 Table 24 provides a summary of the API.

859 **Table 24 – Subscription to Resource API Summary**

HTTP Request Type	API Endpoint	Parameters	Response Code	Response Payload
POST	/api/v1/devices/{deviceid}/{resourcehref}/subscriptions	Correlation-ID, Accept,	201	See clause B.1 - /definitions/S

		Content-Type		unsubscribeResponse
			400, 401, 403, 404	
DELETE	/api/v1/devices/{deviceId}/{resourcehref}/subscriptions/{subscriptionId}	Correlation-ID	202	
			400, 401, 403, 404	

860 **9.7.2 Request and Response Payload**

861 The request payload for the POST shall be as defined in clause 9.4.1.

862 The "deviceId" in the URI in the request shall be the same as the "di" Property from /oic/d of the
863 target OCF device.

864 The "resourcehref" in the URI shall be the same as the "href" Link Parameter for the target
865 Resource instance.

866 The "subscriptionId" in the URI for the DELETE case shall be the "subscriptionId" that was returned
867 in the response to the subscription POST request.

868 The response payload for the subscription POST request shall contain the Subscription ID in a
869 "subscriptionId" Property as defined in clause 9.4.2.

870 There is no required payload for a DELETE unsubscribe response.

871 **9.7.3 Responses**

872 A 201 response shall be sent by the Target Cloud in a success case.

873 A 202 response shall be sent by the Target Cloud following a DELETE request and indicates that
874 the subscription was marked for cancellation; confirmation of the cancellation of the subscription
875 shall be provided by a subsequent notification with an Event-Type of "subscription_cancelled".

876 A non-success path response that is indicative of the type of error shall be returned by a Target
877 Cloud if an error scenario is detected. Table 25 lists possible non-success path responses and
878 possible scenarios that may trigger their generation; an implementation may support additional
879 responses as defined by IETF RFC 2818.

880 **Table 25 – Resource Event Subscription API non-success path responses**

Response Code	Response scenario
400	May be sent by the Target Cloud if the request was malformed or badly constructed
401	May be sent by the Target Cloud if the request is unauthorized (e.g. an invalid or missing Bearer Token)
403	May be sent by the Target Cloud if the requestor is known however the OAuth2.0 Access Token Scope of the request is forbidden
404	May be sent by the Target Cloud if the subscription was not found or the subscribed to Event-Type is not supported
406	May be sent by the Target Cloud if the media type in the received Accept header is not supported/acceptable

881

882 **9.8 Notification of devices level events**

883 **9.8.1 Summary**

884 This request is sent from the Target Cloud to the Origin Cloud whenever there is an initial
885 subscription to an event or an event for which a subscription exists occurs as defined in clause 9.5.

886 Table 26 provides a summary of the API.

887 **Table 26 – Notification of /devices API Summary**

HTTP Request Type	API Endpoint	Parameters	Response Code	Response Payload
POST	/{eventsUrl}	Correlation-ID,	200	
		Content-Type, Event-Type, Subscription-ID, Sequence-Number, Event-Signature, Event-Timestamp	400, 410	

888 **9.8.2 Request and Response Payload**

889 The “eventsUrl” in the URI shall be the value of the "eventsUrl" Property that was provided in the
890 subscription request.

891 The payload in the notification request depends on the Event-Type that is the subject of the
892 notification request; please see Table 19 for specifics and clause 9.4.3 for further information.

893 **9.8.3 Responses**

894 A 200 response shall be provided in a success case.

895 A non-success path response that is indicative of the type of error shall be returned by an Origin
896 Cloud if an error scenario is detected. Table 27 lists possible non-success path responses and
897 possible scenarios that may trigger their generation; an implementation may support additional
898 responses as defined by IETF RFC 2818.

899 **Table 27 – Devices Event Notification non-success path responses**

Response Code	Response scenario
400	May be sent by the Origin Cloud if the request was malformed or badly constructed
401	May be sent by the Origin Cloud if the request is unauthorized (e.g. an invalid or missing Bearer Token)
403	May be sent by the Origin Cloud if the requestor is known however the OAuth2.0 Access Token Scope of the request is forbidden
406	May be sent by the Origin Cloud if the media type in the received Accept header is not supported/acceptable
410	May be sent by the Origin Cloud if the subscription identified by the Subscription-ID header is no longer valid

900

901 **9.9 Notification of Device level events**

902 **9.9.1 Summary**

903 This request is sent from the Target Cloud to the Origin Cloud whenever there is an initial
904 subscription to an event or an event for which a subscription exists occurs as defined in clause 9.6.

905 Table 28 provides a summary of the API.

906 **Table 28 – Notification of Single Device API Summary**

HTTP Request Type	API Endpoint	Parameters	Response Code	Response Payload
POST	/{eventsUrl}	Correlation-ID,	200	
		Content-Type Event-Type, Subscription-ID, Sequence-Number, Event-Signature, Event-Timestamp	400, 410	

907 **9.9.2 Request and Response Payload**

908 The “eventsUrl” in the URI shall be the value of the "eventsUrl" Property that was provided in the
909 subscription request.

910 The payload in the notification request depends on the Event-Type that is the subject of the
911 notification request; please see Table 19 for specifics and clause 9.4.3 for further information.

912 **9.9.3 Responses**

913 A 200 response shall be provided in a success case.

914 A non-success path response that is indicative of the type of error shall be returned by an Origin
915 Cloud if an error scenario is detected. Table 29 lists possible non-success path responses and
916 possible scenarios that may trigger their generation; an implementation may support additional
917 responses as defined by IETF RFC 2818.

918 **Table 29 – Device Event Notification non-success path responses**

Response Code	Response scenario
400	May be sent by the Origin Cloud if the request was malformed or badly constructed
401	May be sent by the Origin Cloud if the request is unauthorized (e.g. an invalid or missing Bearer Token)
403	May be sent by the Origin Cloud if the requestor is known however the OAuth2.0 Access Token Scope of the request is forbidden
406	May be sent by the Origin Cloud if the media type in the received Accept header is not supported/acceptable
410	May be sent by the Origin Cloud if the subscription identified by the Subscription-ID header is no longer valid

919

920 **9.10 Notification of Resource level events**

921 **9.10.1 Summary**

922 This request is sent from the Target Cloud to the Origin Cloud whenever there is an initial
923 subscription to an event or an event for which a subscription exists occurs as defined in clause 9.7.

924 Table 30 provides a summary of the API.

925 **Table 30 – Notification of Resource API Summary**

HTTP Request Type	API Endpoint	Parameters	Response Code	Response Payload
POST	/{eventsUrl}	Correlation-ID,	200	
		Content-Type Event-Type, Subscription-ID, Sequence-Number, Event-Signature, Event-Timestamp	400, 410	

926 **9.10.2 Request and Response Payload**

927 The “eventsUrl” in the URI shall be the value of the "eventsUrl" Property that was provided in the
928 subscription request.

929 The payload in the notification request depends on the Event-Type that is the subject of the
930 notification request; please see Table 19 for specifics and clause 9.4.3 for further information.

931 **9.10.3 Responses**

932 A 200 response shall be provided in a success case.

933 A non-success path response that is indicative of the type of error shall be returned by an Origin
934 Cloud if an error scenario is detected. Table 31 lists possible non-success path responses and
935 possible scenarios that may trigger their generation; an implementation may support additional
936 responses as defined by IETF RFC 2818.

937 **Table 31 – Resource Event Notification non-success path responses**

Response Code	Response scenario
400	May be sent by the Origin Cloud if the request was malformed or badly constructed
401	May be sent by the Origin Cloud if the request is unauthorized (e.g. an invalid or missing Bearer Token)
403	May be sent by the Origin Cloud if the requestor is known however the OAuth2.0 Access Token Scope of the request is forbidden
406	May be sent by the Origin Cloud if the media type in the received Accept header is not supported/acceptable
410	May be sent by the Origin Cloud if the subscription identified by the Subscription-ID header is no longer valid

938

939
940

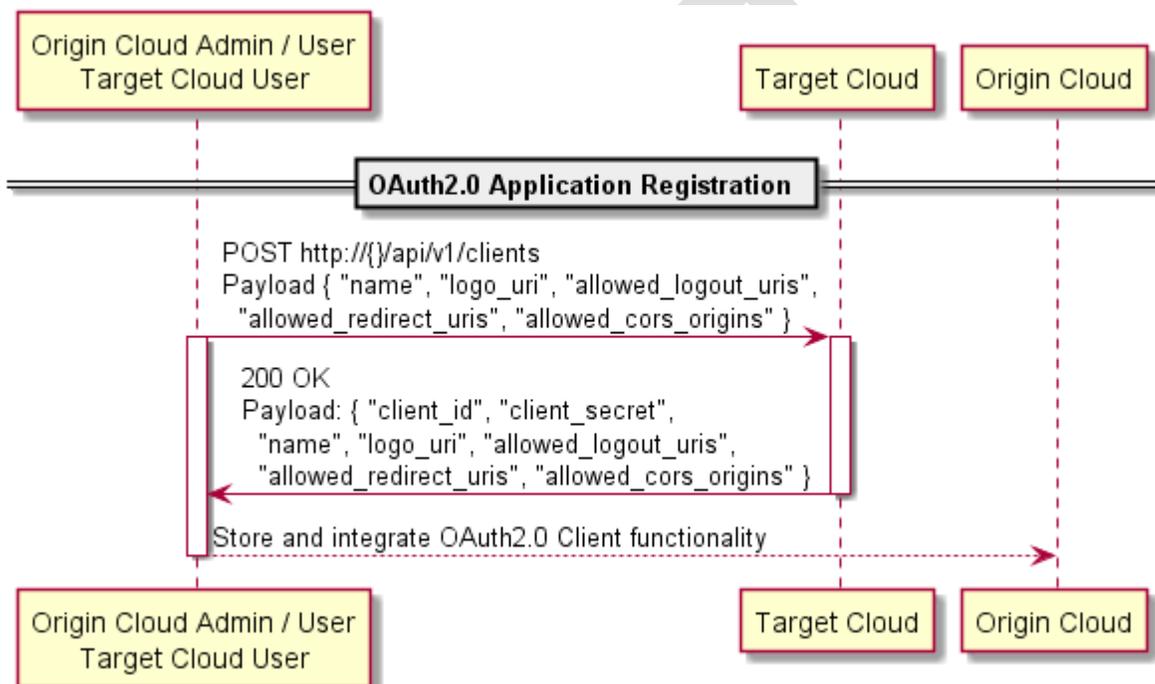
Annex A Representative Flows

941 A.1 Introduction

942 The flows illustrate use of the OCF Cloud API for Cloud Services using OCF Devices as the target
943 servers where applicable and OCF Clouds as the two Clouds that are invoking/acting as API
944 Endpoints. Note that this is for example use only and the API does not force this setup, which
945 means non-OCF clouds with non-OCF devices may also use the API for interworking with other
946 vendor's clouds.

947 A.2 OAuth2.0 Application Registration

948 Figure A.1 provides an example flow showing the registration of the OAuth 2.0 Origin Cloud Client.



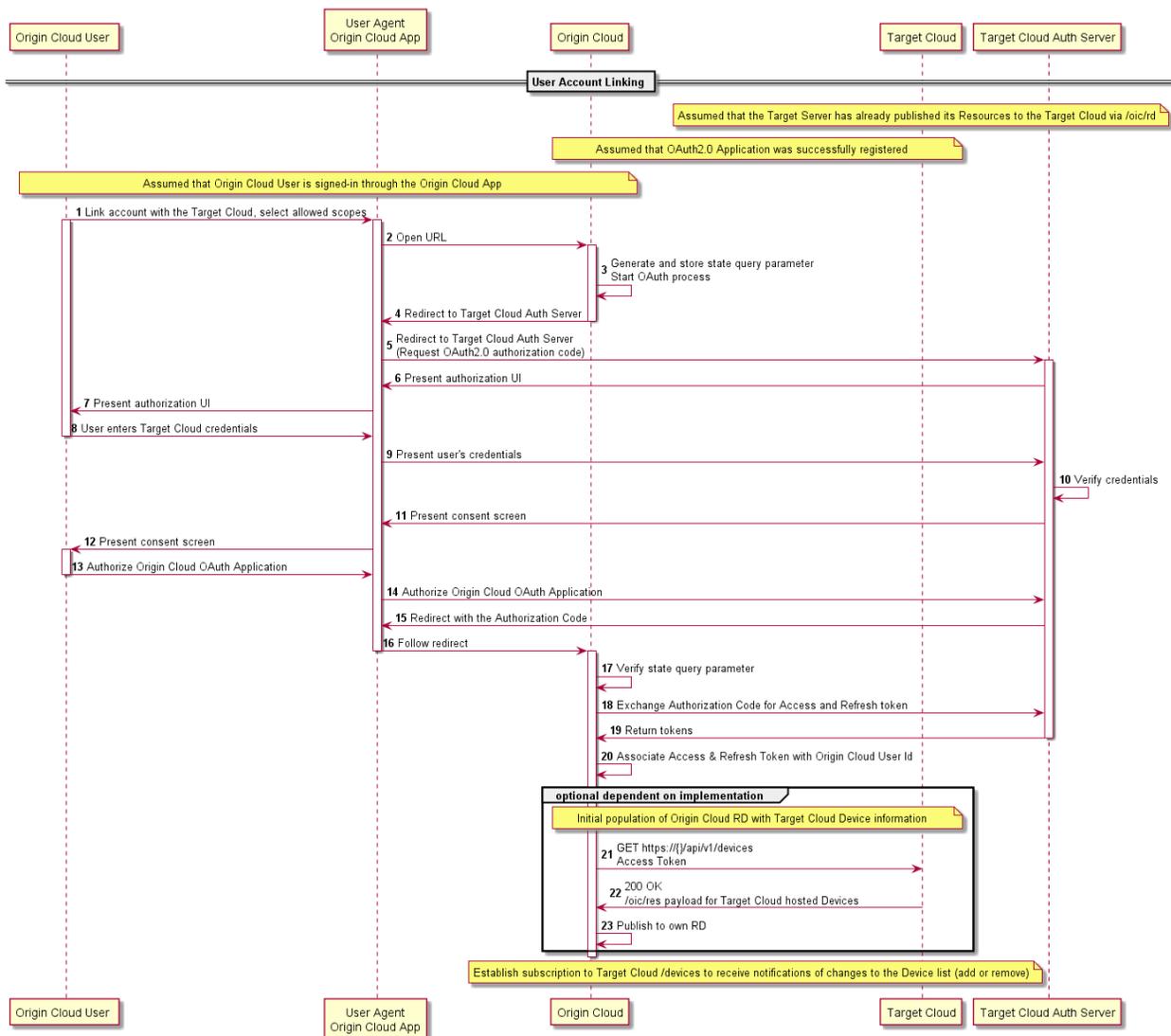
949

950

951 **Figure A.1 – Establish Business Relationship Example Flow**

952 A.3 Account Linking

953 Figure A.2 provides an example flow of the account linking for a particular user.



954

955

Figure A.2 – Initial Association Example Flow

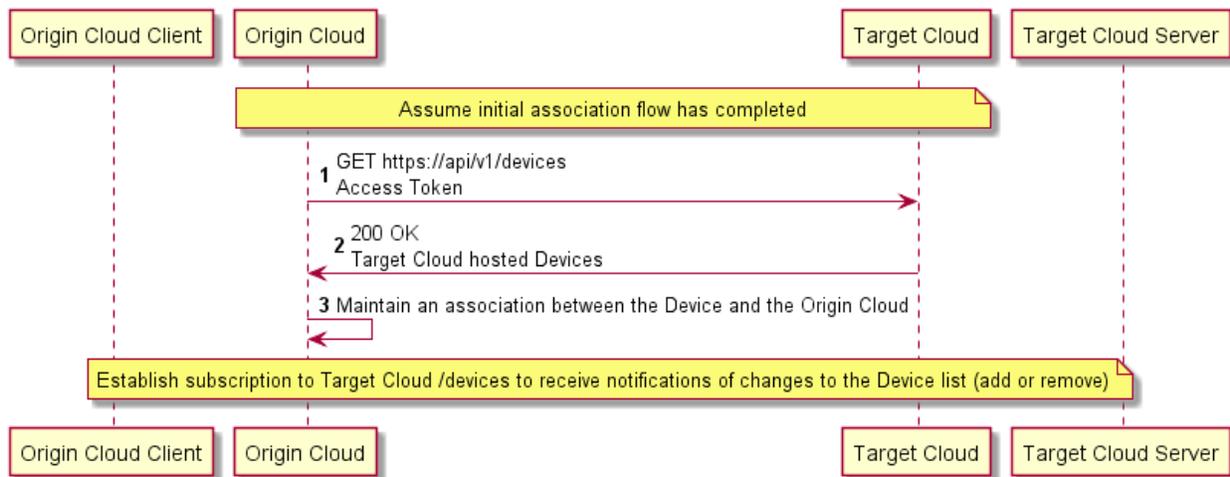
956 **A.4 Retrieval of all Devices**

957 **A.4.1 Summary**

958 The Origin Cloud requests all Devices associated with a user (defined by the provided Bearer
 959 Token). This may be invoked following account linking in order to retrieve the set of Devices for
 960 the user.

961 **A.4.2 Flow**

962 Figure A.3 provides an example flow for the retrieval of all Devices.



963

964

Figure A.3 – Retrieve All Devices Example Flow

A.4.3 Flow Description

966 Table A.1 explains each element in the above sequence diagram

967

Table A.1 – Retrieve all Devices Flow Summary

Number	Description
1	Cloud requests all Devices given by the scope in the Bearer Token that was obtained via OAuth.
2	Response is an array of Device information (Properties that are defined in /oic/d that are pertinent to Cloud functionality and Device status).
3	Cloud maintains an association between the Device and the host Cloud.

968

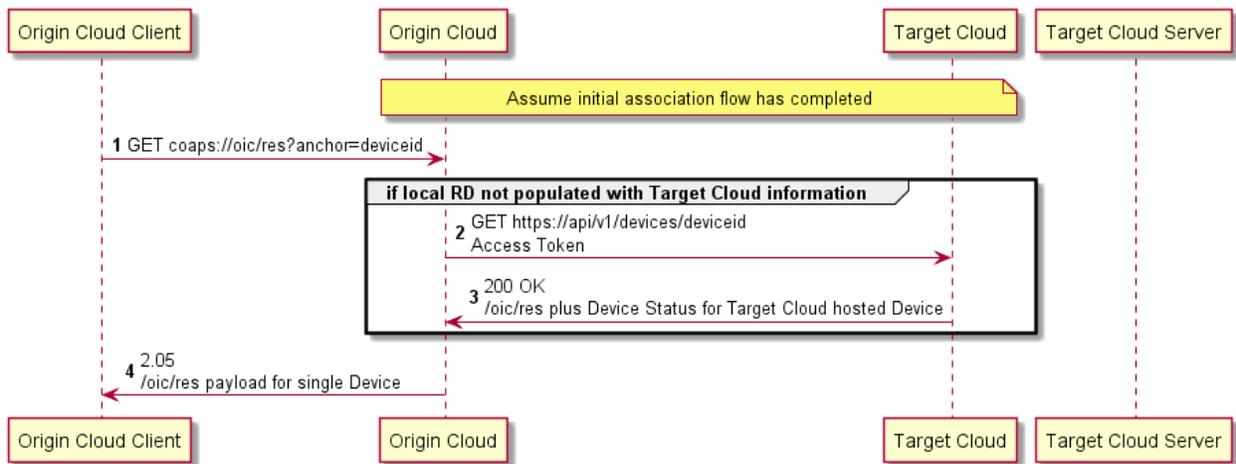
A.5 Retrieval of a single Device

A.5.1 Summary

971 The Origin Cloud requests information for a single, specific Device associated with a user (defined
 972 by the provided Bearer Token). This may be invoked by the Origin Cloud receiving a retrieve
 973 request from a connected Client.

A.5.2 Flow

975 Figure A.4 provides an example flow for the retrieval of a single Device.



976

977

Figure A.4 – Retrieve Single Device Example Flow

A.5.3 Flow Description

978 Table A.2 explains each element in the above sequence diagram

979

Table A.2 – Retrieve single Device Flow Summary

Number	Description
1	[OCF Device to Cloud] OCF Client role Device requests /oic/res from the Cloud for a specific anchor (device id).
2	[Assuming that the information hasn't been cached by the Cloud] For the instance of /oic/sec/account that exists for the Device the Cloud does a GET /devices/{deviceid} to the Cloud identified by the "clouded" in "/oic/sec/account". {deviceid} is also taken from /oic/sec/account.
3	Response is the Device information as well as an array of Links. The "href" in each Link will be of the form "/deviceid/resourcehref".
4	Response payload.

981 **A.6 Retrieval of a single Resource**

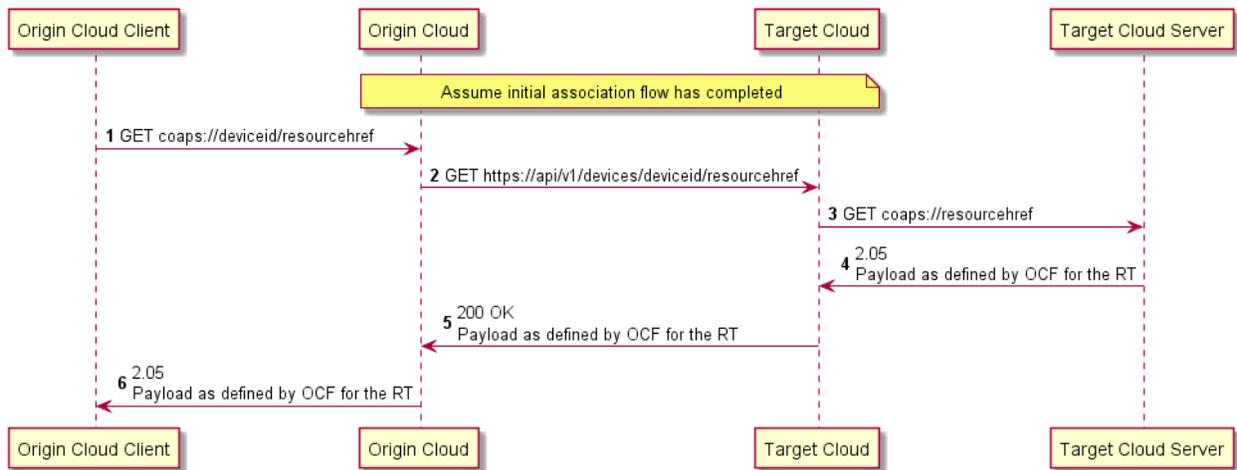
982 **A.6.1 Summary**

983 The Origin Cloud requests information for a single, specific Resource exposed by a Device
 984 associated with a user (defined by the provided Bearer Token). This may be invoked by the Origin
 985 Cloud receiving a retrieve request from a connected Client.

986 **A.6.2 Flows**

987 **A.6.2.1 Success Path**

988 Figure A.5 provides an example flow for the retrieval of a single Resource.



989
990 **Figure A.5 – Retrieve Resource (Success) Example Flow**

991 **A.6.2.2 Success Path Flow Description**

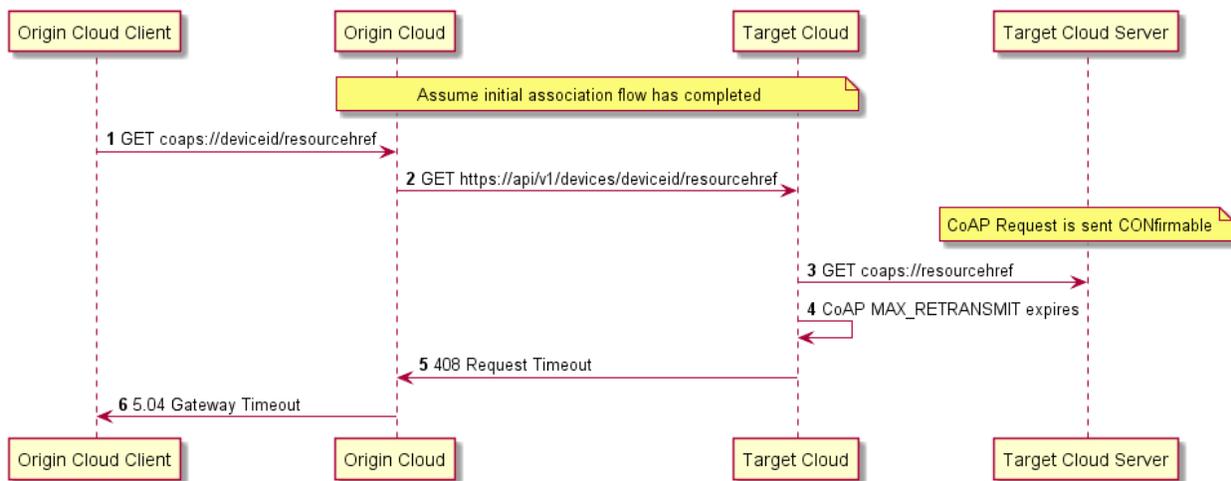
992 Table A.3 explains each element in the above sequence diagram

993 **Table A.3 – Retrieve single Resource Flow Summary**

Number	Description
1	[OCF Device to Cloud] OCF Client role Device requests a Resource from the Cloud using the "href" exposed in the /oic/res response. This will be of the form "/deviceid/resourcehref"
2	[Assuming that the resource representation hasn't been cached by the Cloud] Cloud identifies the host Cloud for the Resource via the instance of /oic/sec/account for the "deviceid". The request is then effectively proxied to the Target Cloud via a GET /devices/{deviceid}/{resourcehref}. Any query parameters received over CoAP are included in the URI unaltered.
3	[OCF Device to Cloud] Target Cloud identifies the TLS connection to the end Device via the {deviceid} and proxies the request.
4	Standard OCF response
5	Success path response including the response payload as received for the target Resource
6	Standard OCF response

994 **A.6.2.3 Device is Temporarily Unavailable**

995 Figure A.6 illustrates the case where the Device is temporarily unavailable.



996
997 **Figure A.6 – Retrieve Resource (Timeout) Example Flow**

998 **A.7 Update of a single Resource**

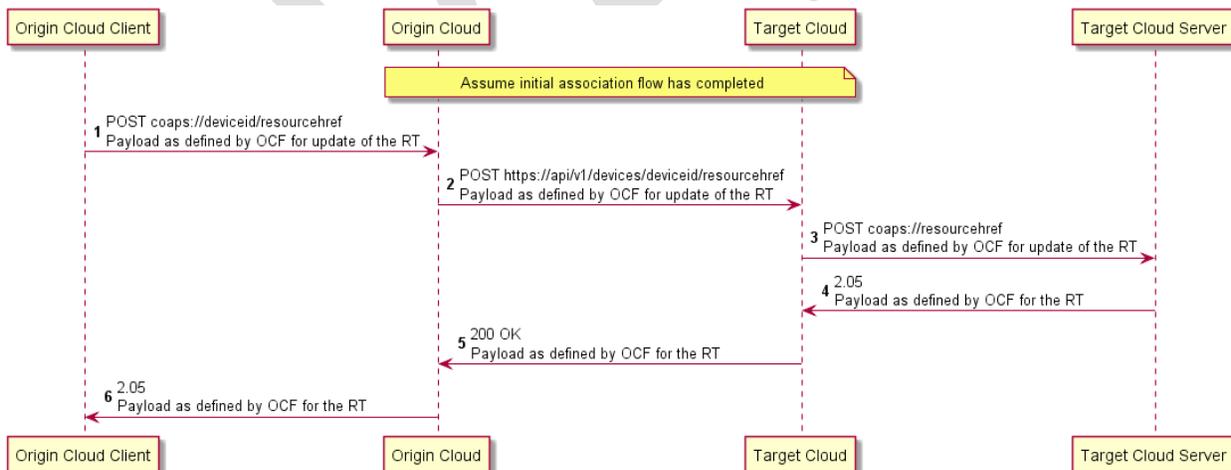
999 **A.7.1 Summary**

1000 The Origin Cloud updates information for a single, specific Device associated with a user (defined
1001 by the provided Bearer Token). This may be invoked by the Origin Cloud receiving an update
1002 request from a connected Client.

1003 **A.7.2 Flows**

1004 **A.7.2.1 Success Path**

1005 Figure A.7 provides an example flow for the updating of a single Resource.



1006
1007 **Figure A.7 – Update Resource (Success) Example Flow**

1008 **A.7.2.2 Success Path Flow Description**

1009 Table A.4 explains each element in the above sequence diagram

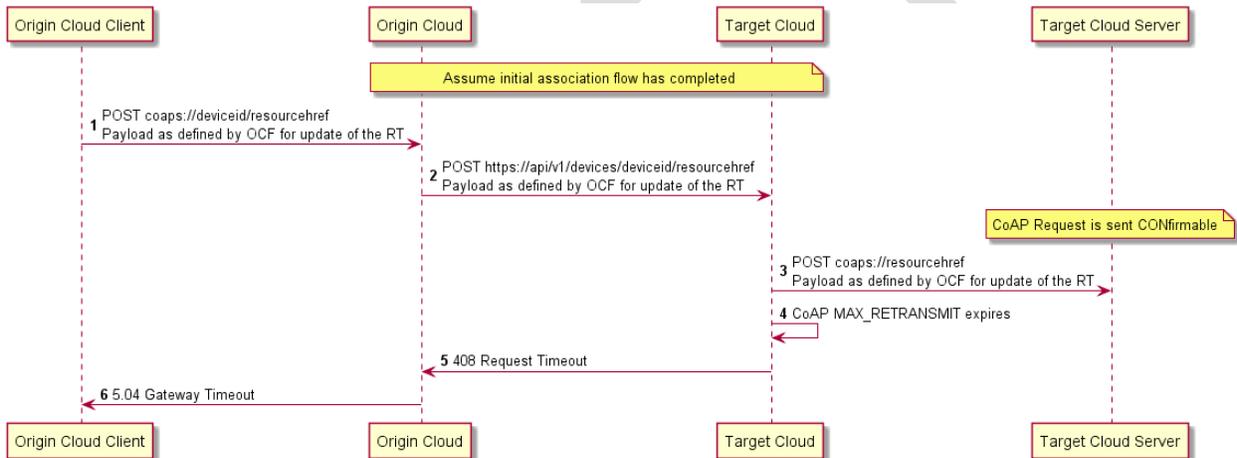
1010

Table A.4 – Update single Resource Flow Summary

Number	Description
1	[OCF Device to Cloud] OCF Client role Device requests a Resource from the Cloud using the "href" exposed in the /oic/res response. This will be of the form "/deviceid/resourcehref"
2	Cloud identifies the host Cloud for the Resource via the instance of /oic/sec/account for the "deviceid". The request is then effectively proxied to the Target Cloud via a POST /devices/{deviceid}/{resourcehref} including the payload from the original request. Any query parameters received over CoAP are included in the URI unaltered.
3	[OCF Device to Cloud] Target Cloud identifies the TLS connection to the end Device via the {deviceid} and proxies the request.
4	Standard OCF response
5	Success path response including the response payload as received for the target Resource
6	Standard OCF response

1011 **A.7.2.3 Device is Temporarily Unavailable**

1012 Figure A.8 illustrates the case where the Device is temporarily unavailable.



1013

1014 **Figure A.8 – Update Resource (Timeout) Example Flow**

1015 **A.8 Establishment of new subscription request**

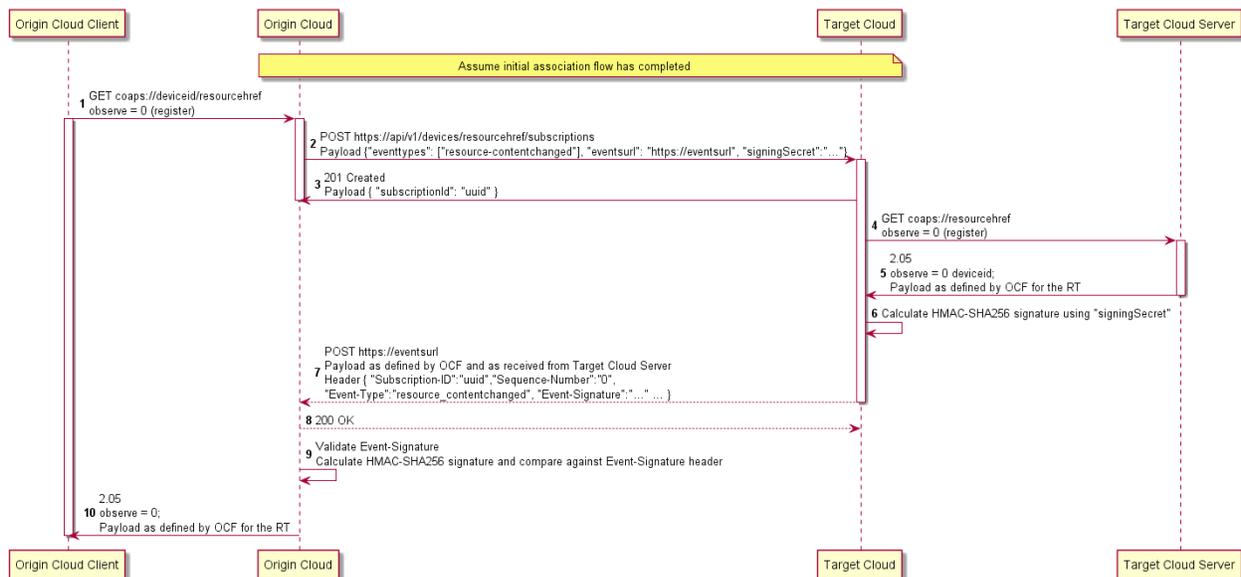
1016 **A.8.1 Summary**

1017 The Origin Cloud requests the establishment of an observe relationship with a single, specific
 1018 Resource on a Device associated with a user (defined by the provided Bearer Token). This may be
 1019 invoked by Origin Cloud receiving a retrieve request containing an observe option from a connected
 1020 Client.

1021 **A.8.2 Flows**

1022 Figure A.9 provides an example flow for the establishment of a subscription to the
 1023 "resource_contentchanged" event for a specific Resource.

1024



1025

1026

Figure A.9 – Observe Establishment Example Flow

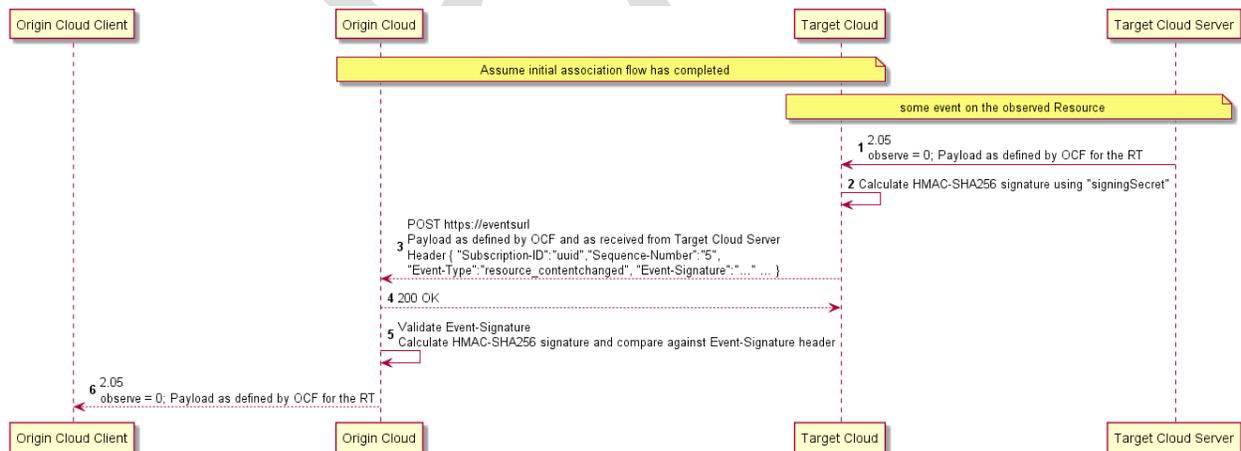
1027 **A.9 Event generated for a subscription**

1028 **A.9.1 Summary**

1029 An event occurs for a Resource with which the Origin Cloud has established a subscription/event relationship. This may be invoked by the target end Device being updated.

1031 **A.9.2 Flows**

1032 Figure A.10 provides an example flow for the handling of a generated "resource_contentchanged" event.



1034

Figure A.10 – "resource_contentchanged" Event Example Flow

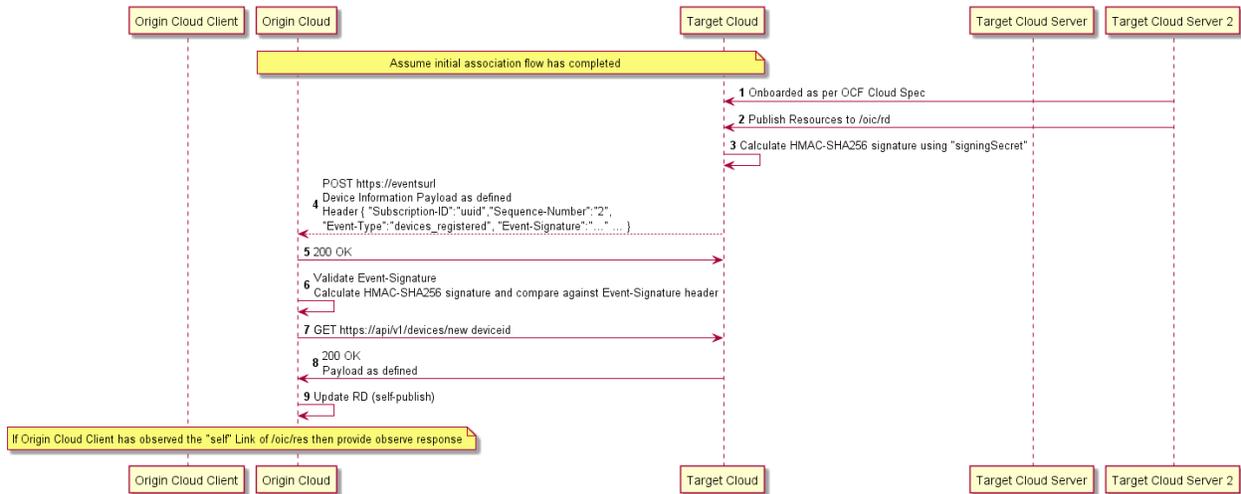
1036 **A.10 Addition of new registration**

1037 **A.10.1 Summary**

1038 The Origin Cloud has a priori established a subscription/event relationship with the set of Devices associated with a user exposed by Target Cloud. The user then registers a new Device with Target Cloud.

1041 **A.10.2 Flows**

1042 Figure A.11 provides an example flow for the generation of a notification (event) when a new Device
 1043 is registered.



1044

1045 **Figure A.11 – Addition of new registered Device example flow**

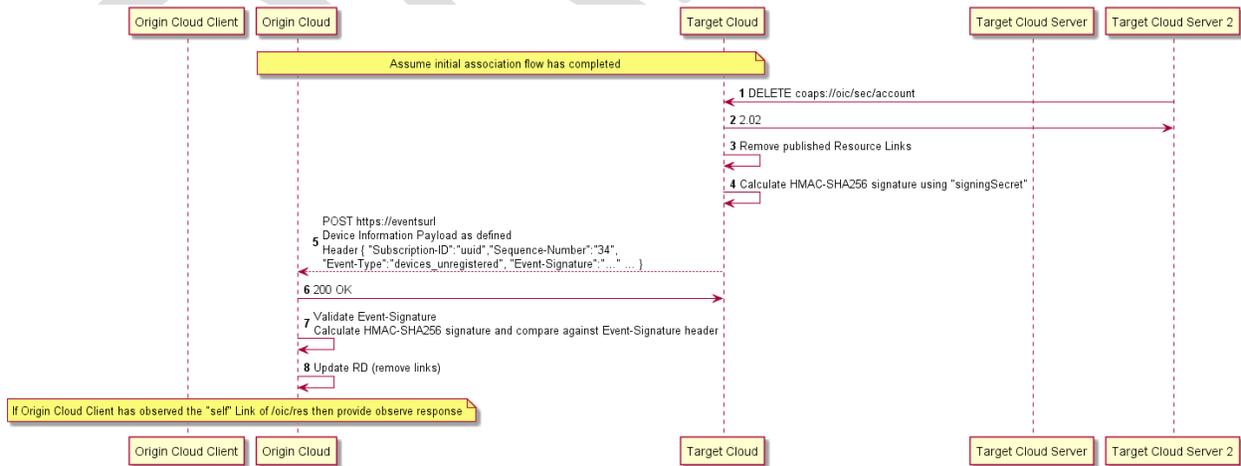
1046 **A.11 Removal of existing device registration**

1047 **A.11.1 Summary**

1048 The Origin Cloud has a priori established a subscription/event relationship with the set of Devices
 1049 associated with a user exposed by Target Cloud. The user then removes a Device from Target
 1050 Cloud.

1051 **A.11.2 Flows**

1052 Figure A.12 provides an example flow for the generation of a notification (event) when a Device is
 1053 removed.



1054

1055 **Figure A.12 – Removal of existing registration example flow**

Annex B Open API Definition

1056
1057
1058

1059 **B.1 OCF Cloud API for Cloud Services**

1060 **B.1.1 Supported APIs**

1061 **B.1.1.1 /api/v1/devices?content=base**

1062 Get meta-information, including Resource Links, for all Devices which are signed up to the OCF
1063 Cloud - either "online" or "offline". Devices which are "online" are signed in to the system and are
1064 accessible. Offline devices are signed up to the system, but currently disconnected.

1065 **B.1.1.2 /api/v1/devices?content=all**

1066 Get meta-information, including Resource Representations, for all Devices which are signed up to
1067 the OCF Cloud - either "online" or "offline". Devices which are "online" are signed in to the
1068 system and are accessible. Offline devices are signed up to the system, but currently
1069 disconnected.

1070 **B.1.1.3 /api/v1/devices/subscriptions**

1071 Subscribe to devices events by providing "eventTypes" you're interested in and an "eventsUrl"
1072 endpoint where notifications will be sent to as defined. A successful response contains a
1073 "subscriptionId" which identifies the registered subscription and is part of each notification. First
1074 notification for each registered event type is received immediately after subscription and contains
1075 the actual state of the resource, followed by new notifications in case of any change.

1076 Supported events:

- 1077 - "devices_registered"
- 1078 - "devices_unregistered"
- 1079 - "devices_online"
- 1080 - "devices_offline"
- 1081

1082 **B.1.1.4 /api/v1/devices/subscriptions/{subscriptionId}**

1083 Cancel the subscription identified by the provided "subscriptionId" that was returned in the
1084 response to the subscription request.

1085 **B.1.1.5 /api/v1/devices/{deviceId}?content=base**

1086 Get the meta-information for the Device given by the provided "deviceId" including Resource
1087 Links.

1088 **B.1.1.6 /api/v1/devices/{deviceId}?content=all**

1089 Get the meta-information for the Device given by the provided "deviceId" including Resource
1090 Representations.

1091 **B.1.1.7 /api/v1/devices/{deviceId}/subscriptions**

1092 Subscribe to Device level events by providing "eventTypes" you're interested in and an
1093 "eventsUrl" API Endpoint where notifications will be sent to as defined. A successful response
1094 contains a "subscriptionId" which identifies the registered subscription and is part of each
1095 notification. First notification for each registered event type is received immediately after
1096 subscription and contains the actual state of the resource, followed by new notifications in case
1097 of any change.

1098 Supported events:
1099

1100 - "resources_published"
1101 - "resources_unpublished"

1102 **B.1.1.8 /api/v1/devices/{deviceId}/subscriptions/{subscriptionId}**

1103 Cancel the subscription identified by the provided "subscriptionId" that was returned in the
1104 response to the subscription request.

1105 **B.1.1.9 /api/v1/devices/{deviceId}/{resourceLinkHref}**

1106 Get or update the Resource Representation of the Resource found at "resourceLinkHref" on the
1107 Device with the given "deviceId"

1108 **B.1.1.10 /api/v1/devices/{deviceId}/{resourceLinkHref}/subscriptions**

1109 Subscribe to Resource level events by providing "eventTypes" you're interested in and
1110 "eventsUrl" API Endpoint where notifications will be sent to as defined. A successful response
1111 contains a "subscriptionId" which identifies the registered subscription and is part of each event.
1112 First notification for each registered event type is received immediately after subscription and
1113 contains the actual state of the resource, followed by new notifications in case of any change.
1114

1115 Supported events:

1116 - "resource_contentchanged"

1117 **B.1.1.11 /api/v1/devices/{deviceId}/{resourceLinkHref}/subscriptions/{subscriptionId}**

1118 Cancel the subscription identified by the provided "subscriptionId" that was returned in the
1119 response to the subscription request.

1120 **B.1.1.12 /{eventsUrl}**

1121 Events endpoint provided during subscription where notifications for the events specified in the
1122 subscription will be sent to as defined per event type. Confirmation of each notification sent to the
1123 "eventsUrl" endpoint is required with a "2xx" success code.
1124

1125 Notifications you may receive based on the event type you're subscribed to are:

- 1126 - "subscription_cancelled": "SubscriptionCancelledEvent"
- 1127 - "devices_registered": "DevicesRegisteredEvent"
- 1128 - "devices_unregistered": "DevicesUnregisteredEvent"
- 1129 - "resources_published": "ResourcesPublishedEvent"
- 1130 - "resources_unpublished": "ResourcesUnpublishedEvent"
- 1131 - "devices_online": "DevicesOnlineEvent"
- 1132 - "devices_offline": "DevicesOfflineEvent"
- 1133 - "resource_contentchanged": "ResourceContentChangedEvent"

1134 **B.1.2 OpenAPI 2.0 definition**

```
1135 {  
1136   "swagger": "2.0",  
1137   "info": {  
1138     "title": "OCF Cloud API for Cloud Services",  
1139     "version": "0.0.3-20190828",  
1140     "license": {  
1141       "name": "Copyright 2019 Open Connectivity Foundation, Inc. All rights reserved.",  
1142       "x-description": "Redistribution and use in source and binary forms, with or without  
1143 modification, are permitted provided that the following conditions are met:\n      1.  
1144 Redistributions of source code must retain the above copyright notice, this list of conditions and  
1145 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above  
1146 copyright notice, this list of conditions and the following disclaimer in the documentation and/or  
1147 other materials provided with the distribution.\n      THIS SOFTWARE IS PROVIDED BY THE Open  
1148 Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT  
1149 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR  
1150 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity  
1151 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
```

```

1152 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
1153 SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n          HOWEVER CAUSED AND ON
1154 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
1155 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
1156 SUCH DAMAGE.\n"
1157 }
1158 },
1159 "host": "api.example.com",
1160 "schemes": [
1161   "https"
1162 ],
1163 "tags": [
1164   {
1165     "name": "Devices",
1166     "description": "Basic information about devices"
1167   },
1168   {
1169     "name": "Resources",
1170     "description": "Read or change the configuration of the device"
1171   },
1172   {
1173     "name": "Events",
1174     "description": "Be notified about changes occurring on the device"
1175   }
1176 ],
1177 "paths": {
1178   "/api/v1/devices?content=base": {
1179     "parameters": [
1180       {
1181         "$ref": "#/parameters/CorrelationId"
1182       },
1183       {
1184         "$ref": "#/parameters/Accept"
1185       },
1186       {
1187         "$ref": "#/parameters/BatchFormat"
1188       }
1189     ],
1190     "get": {
1191       "tags": [
1192         "Devices"
1193       ],
1194       "summary": "Get all devices with resource links",
1195       "description": "Get meta-information, including Resource Links, for all Devices which are
1196 signed up to the OCF Cloud - either \"online\" or \"offline\". Devices which are \"online\" are
1197 signed in to the system and are accessible. Offline devices are signed up to the system, but
1198 currently disconnected.",
1199       "produces": [
1200         "application/json"
1201       ],
1202       "responses": {
1203         "200": {
1204           "description": "An array of devices",
1205           "schema": {
1206             "type": "array",
1207             "items": {
1208               "$ref": "#/definitions/Device"
1209             }
1210           }
1211         },
1212         "400": {
1213           "$ref": "#/responses/BadRequest"
1214         },
1215         "401": {
1216           "$ref": "#/responses/Unauthorized"
1217         },
1218         "403": {
1219           "$ref": "#/responses/Forbidden"
1220         },
1221         "406": {
1222           "$ref": "#/responses/NotAcceptable"

```

```

1223     },
1224     "503": {
1225       "$ref": "#/responses/ServiceUnavailable"
1226     },
1227     "504": {
1228       "$ref": "#/responses/GatewayTimeout"
1229     }
1230   },
1231   "security": [
1232     {
1233       "oauth2": [
1234         "r:*"
1235       ]
1236     }
1237   ]
1238 },
1239 },
1240 "/api/v1/devices?content=all": {
1241   "parameters": [
1242     {
1243       "$ref": "#/parameters/CorrelationId"
1244     },
1245     {
1246       "$ref": "#/parameters/Accept"
1247     },
1248     {
1249       "$ref": "#/parameters/BatchFormat"
1250     }
1251   ],
1252   "get": {
1253     "tags": [
1254       "Devices"
1255     ],
1256     "summary": "Get all devices with resource representations",
1257     "description": "Get meta-information, including Resource Representations, for all Devices
1258 which are signed up to the OCF Cloud - either \"online\" or \"offline\". Devices which are
1259 \"online\" are signed in to the system and are accessible. Offline devices are signed up to the
1260 system, but currently disconnected.",
1261     "produces": [
1262       "application/json"
1263     ],
1264     "responses": {
1265       "200": {
1266         "description": "An array of devices",
1267         "schema": {
1268           "type": "array",
1269           "items": {
1270             "$ref": "#/definitions/DeviceContentAll"
1271           }
1272         }
1273       },
1274       "400": {
1275         "$ref": "#/responses/BadRequest"
1276       },
1277       "401": {
1278         "$ref": "#/responses/Unauthorized"
1279       },
1280       "403": {
1281         "$ref": "#/responses/Forbidden"
1282       },
1283       "406": {
1284         "$ref": "#/responses/NotAcceptable"
1285       },
1286       "503": {
1287         "$ref": "#/responses/ServiceUnavailable"
1288       },
1289       "504": {
1290         "$ref": "#/responses/GatewayTimeout"
1291       }
1292     },
1293     "security": [

```

```

1294         {
1295             "oauth2": [
1296                 "r:*"
1297             ]
1298         }
1299     ]
1300 }
1301 },
1302 "/api/v1/devices/subscriptions": {
1303     "parameters": [
1304         {
1305             "$ref": "#/parameters/CorrelationId"
1306         },
1307         {
1308             "$ref": "#/parameters/Accept"
1309         }
1310     ],
1311     "post": {
1312         "tags": [
1313             "Events"
1314         ],
1315         "summary": "Subscribe to events against the set of devices",
1316         "description": "Subscribe to devices events by providing \"eventTypes\" you're interested in
1317 and an \"eventsUrl\" endpoint where notifications will be sent to as defined. A successful response
1318 contains a \"subscriptionId\" which identifies the registered subscription and is part of each
1319 notification. First notification for each registered event type is received immediately after
1320 subscription and contains the actual state of the resource, followed by new notifications in case of
1321 any change.\n\nSupported events:\n- \"devices_registered\"\n- \"devices_unregistered\"\n-
1322 \"devices_online\"\n- \"devices_offline\"",
1323         "parameters": [
1324             {
1325                 "$ref": "#/parameters/ContentType"
1326             },
1327             {
1328                 "$ref": "#/parameters/SubscribeRequestDevices"
1329             }
1330         ],
1331         "consumes": [
1332             "application/json"
1333         ],
1334         "produces": [
1335             "application/json"
1336         ],
1337         "responses": {
1338             "201": {
1339                 "$ref": "#/definitions/SubscribeResponse"
1340             },
1341             "400": {
1342                 "$ref": "#/responses/BadRequest"
1343             },
1344             "401": {
1345                 "$ref": "#/responses/Unauthorized"
1346             },
1347             "403": {
1348                 "$ref": "#/responses/Forbidden"
1349             }
1350         },
1351         "security": [
1352             {
1353                 "oauth2": [
1354                     "r:*"
1355                 ]
1356             }
1357         ]
1358     }
1359 },
1360 "/api/v1/devices/subscriptions/{subscriptionId}": {
1361     "parameters": [
1362         {
1363             "$ref": "#/parameters/CorrelationId"
1364         },

```

```

1365     {
1366         "$ref": "#/parameters/SubscriptionIdPath"
1367     }
1368 ],
1369 "delete": {
1370     "tags": [
1371         "Events"
1372     ],
1373     "summary": "Unsubscribe from events against the set of devices",
1374     "description": "Cancel the subscription identified by the provided \"subscriptionId\" that
1375 was returned in the response to the subscription request.",
1376     "responses": {
1377         "202": {
1378             "description": "Subscription was marked for cancellation"
1379         },
1380         "400": {
1381             "$ref": "#/responses/BadRequest"
1382         },
1383         "401": {
1384             "$ref": "#/responses/Unauthorized"
1385         },
1386         "403": {
1387             "$ref": "#/responses/Forbidden"
1388         },
1389         "404": {
1390             "$ref": "#/responses/NotFound"
1391         },
1392         "406": {
1393             "$ref": "#/responses/NotAcceptable"
1394         }
1395     },
1396     "security": [
1397         {
1398             "oauth2": [
1399                 "r:*"
1400             ]
1401         }
1402     ]
1403 },
1404 "/api/v1/devices/{deviceId}?content=base": {
1405     "parameters": [
1406         {
1407             "$ref": "#/parameters/CorrelationId"
1408         },
1409         {
1410             "$ref": "#/parameters/Accept"
1411         },
1412         {
1413             "$ref": "#/parameters/DeviceId"
1414         },
1415         {
1416             "$ref": "#/parameters/BatchFormat"
1417         }
1418     ],
1419 },
1420 "get": {
1421     "tags": [
1422         "Devices"
1423     ],
1424     "summary": "Get the device with resource links by ID",
1425     "description": "Get the meta-information for the Device given by the provided \"deviceId\"
1426 including Resource Links.",
1427     "consumes": [
1428         "application/json"
1429     ],
1430     "produces": [
1431         "application/json"
1432     ],
1433     "responses": {
1434         "200": {
1435             "description": "Device requested with content=all query parameter",

```

```

1436         "schema": {
1437             "$ref": "#/definitions/DeviceContentAll"
1438         },
1439     },
1440     "400": {
1441         "$ref": "#/responses/BadRequest"
1442     },
1443     "401": {
1444         "$ref": "#/responses/Unauthorized"
1445     },
1446     "403": {
1447         "$ref": "#/responses/Forbidden"
1448     },
1449     "404": {
1450         "$ref": "#/responses/NotFound"
1451     },
1452     "406": {
1453         "$ref": "#/responses/NotAcceptable"
1454     },
1455     "503": {
1456         "$ref": "#/responses/ServiceUnavailable"
1457     },
1458     "504": {
1459         "$ref": "#/responses/GatewayTimeout"
1460     }
1461 },
1462 "security": [
1463     {
1464         "oauth2": [
1465             "r:*"
1466         ]
1467     }
1468 ]
1469 },
1470 },
1471 "/api/v1/devices/{deviceId}?content=all": {
1472     "parameters": [
1473         {
1474             "$ref": "#/parameters/CorrelationId"
1475         },
1476         {
1477             "$ref": "#/parameters/Accept"
1478         },
1479         {
1480             "$ref": "#/parameters/DeviceId"
1481         },
1482         {
1483             "$ref": "#/parameters/BatchFormat"
1484         }
1485     ],
1486     "get": {
1487         "tags": [
1488             "Devices"
1489         ],
1490         "summary": "Get the device with resource representations by ID",
1491         "description": "Get the meta-information for the Device given by the provided \"deviceId\" including Resource Representations.",
1492         "consumes": [
1493             "application/json"
1494         ],
1495         "produces": [
1496             "application/json"
1497         ],
1498         "responses": {
1499             "200": {
1500                 "description": "Device requested with content=all query parameter",
1501                 "schema": {
1502                     "$ref": "#/definitions/DeviceContentAll"
1503                 }
1504             },
1505             "400": {

```

```

1507         "$ref": "#/responses/BadRequest"
1508     },
1509     "401": {
1510         "$ref": "#/responses/Unauthorized"
1511     },
1512     "403": {
1513         "$ref": "#/responses/Forbidden"
1514     },
1515     "404": {
1516         "$ref": "#/responses/NotFound"
1517     },
1518     "406": {
1519         "$ref": "#/responses/NotAcceptable"
1520     },
1521     "503": {
1522         "$ref": "#/responses/ServiceUnavailable"
1523     },
1524     "504": {
1525         "$ref": "#/responses/GatewayTimeout"
1526     }
1527 },
1528 "security": [
1529     {
1530         "oauth2": [
1531             "r:*"
1532         ]
1533     }
1534 ]
1535 },
1536 },
1537 "/api/v1/devices/{deviceId}/subscriptions": {
1538     "parameters": [
1539         {
1540             "$ref": "#/parameters/CorrelationId"
1541         },
1542         {
1543             "$ref": "#/parameters/DeviceId"
1544         },
1545         {
1546             "$ref": "#/parameters/Accept"
1547         }
1548     ],
1549     "post": {
1550         "tags": [
1551             "Events"
1552         ],
1553         "summary": "Subscribe to events against a specific device",
1554         "description": "Subscribe to Device level events by providing \"eventTypes\" you're
1555 interested in and an \"eventsUrl\" API Endpoint where notifications will be sent to as defined. A
1556 successful response contains a \"subscriptionId\" which identifies the registered subscription and
1557 is part of each notification. First notification for each registered event type is received
1558 immediately after subscription and contains the actual state of the resource, followed by new
1559 notifications in case of any change.\n\nSupported events:\n- \"resources_published\"\n-
1560 \"resources_unpublished\"",
1561         "parameters": [
1562             {
1563                 "$ref": "#/parameters/ContentType"
1564             },
1565             {
1566                 "$ref": "#/parameters/SubscribeRequestDevice"
1567             }
1568         ],
1569         "consumes": [
1570             "application/json"
1571         ],
1572         "produces": [
1573             "application/json"
1574         ],
1575         "responses": {
1576             "201": {
1577                 "$ref": "#/definitions/SubscribeResponse"

```

```

1578     },
1579     "400": {
1580       "$ref": "#/responses/BadRequest"
1581     },
1582     "401": {
1583       "$ref": "#/responses/Unauthorized"
1584     },
1585     "403": {
1586       "$ref": "#/responses/Forbidden"
1587     },
1588     "404": {
1589       "$ref": "#/responses/NotFound"
1590     },
1591     "406": {
1592       "$ref": "#/responses/NotAcceptable"
1593     }
1594   },
1595   "security": [
1596     {
1597       "oauth2": [
1598         "r:*"
1599       ]
1600     }
1601   ]
1602 },
1603 ],
1604 "/api/v1/devices/{deviceId}/subscriptions/{subscriptionId}": {
1605   "parameters": [
1606     {
1607       "$ref": "#/parameters/CorrelationId"
1608     },
1609     {
1610       "$ref": "#/parameters/DeviceId"
1611     },
1612     {
1613       "$ref": "#/parameters/SubscriptionIdPath"
1614     }
1615   ],
1616   "delete": {
1617     "tags": [
1618       "Events"
1619     ],
1620     "summary": "Unsubscribe from events against a specific device",
1621     "description": "Cancel the subscription identified by the provided \"subscriptionId\" that
1622 was returned in the response to the subscription request.",
1623     "responses": {
1624       "202": {
1625         "description": "Subscription was marked for cancellation"
1626       },
1627       "400": {
1628         "$ref": "#/responses/BadRequest"
1629       },
1630       "401": {
1631         "$ref": "#/responses/Unauthorized"
1632       },
1633       "403": {
1634         "$ref": "#/responses/Forbidden"
1635       },
1636       "404": {
1637         "$ref": "#/responses/NotFound"
1638       }
1639     },
1640     "security": [
1641       {
1642         "oauth2": [
1643           "r:*"
1644         ]
1645       }
1646     ]
1647   }
1648 },

```

```

1649     "/api/v1/devices/{deviceId}/{resourceLinkHref}": {
1650         "parameters": [
1651             {
1652                 "$ref": "#/parameters/CorrelationId"
1653             },
1654             {
1655                 "$ref": "#/parameters/DeviceId"
1656             },
1657             {
1658                 "$ref": "#/parameters/ResourceLinkHref"
1659             },
1660             {
1661                 "$ref": "#/parameters/Accept"
1662             }
1663         ],
1664         "get": {
1665             "tags": [
1666                 "Resources"
1667             ],
1668             "summary": "Retrieve resource values",
1669             "description": "Get or update the Resource Representation of the Resource found at
1670 \"resourceLinkHref\" on the Device with the given \"deviceId\",
1671             "consumes": [
1672                 "application/json",
1673                 "application/vnd.ocf+cbor"
1674             ],
1675             "produces": [
1676                 "application/json",
1677                 "application/vnd.ocf+cbor"
1678             ],
1679             "responses": {
1680                 "200": {
1681                     "$ref": "#/definitions/ResourceRetrieveResponse"
1682                 },
1683                 "400": {
1684                     "$ref": "#/responses/BadRequest"
1685                 },
1686                 "401": {
1687                     "$ref": "#/responses/Unauthorized"
1688                 },
1689                 "403": {
1690                     "$ref": "#/responses/Forbidden"
1691                 },
1692                 "404": {
1693                     "$ref": "#/responses/NotFound"
1694                 },
1695                 "406": {
1696                     "$ref": "#/responses/NotAcceptable"
1697                 },
1698                 "503": {
1699                     "$ref": "#/responses/ServiceUnavailable"
1700                 },
1701                 "504": {
1702                     "$ref": "#/responses/GatewayTimeout"
1703                 }
1704             },
1705             "security": [
1706                 {
1707                     "oauth2": [
1708                         "r:*"
1709                     ]
1710                 }
1711             ],
1712         },
1713         "post": {
1714             "tags": [
1715                 "Resources"
1716             ],
1717             "summary": "Update resource values",
1718             "parameters": [
1719                 {

```

```

1720         "$ref": "#/parameters/ResourceUpdateRequest"
1721     },
1722     {
1723         "$ref": "#/parameters/ContentType"
1724     }
1725 ],
1726 "consumes": [
1727     "application/json",
1728     "application/vnd.ocf+cbor"
1729 ],
1730 "produces": [
1731     "application/json",
1732     "application/vnd.ocf+cbor"
1733 ],
1734 "responses": {
1735     "200": {
1736         "$ref": "#/definitions/ResourceRetrieveResponse"
1737     },
1738     "400": {
1739         "$ref": "#/responses/BadRequest"
1740     },
1741     "401": {
1742         "$ref": "#/responses/Unauthorized"
1743     },
1744     "403": {
1745         "$ref": "#/responses/Forbidden"
1746     },
1747     "404": {
1748         "$ref": "#/responses/NotFound"
1749     },
1750     "415": {
1751         "$ref": "#/responses/UnsupportedMediaType"
1752     },
1753     "503": {
1754         "$ref": "#/responses/ServiceUnavailable"
1755     },
1756     "504": {
1757         "$ref": "#/responses/GatewayTimeout"
1758     }
1759 },
1760 "security": [
1761     {
1762         "oauth2": [
1763             "r:*",
1764             "w:*"
1765         ]
1766     }
1767 ]
1768 },
1769 },
1770 "/api/v1/devices/{deviceId}/{resourceLinkHref}/subscriptions": {
1771     "parameters": [
1772         {
1773             "$ref": "#/parameters/CorrelationId"
1774         },
1775         {
1776             "$ref": "#/parameters/DeviceId"
1777         },
1778         {
1779             "$ref": "#/parameters/ResourceLinkHref"
1780         },
1781         {
1782             "$ref": "#/parameters/Accept"
1783         }
1784     ],
1785     "post": {
1786         "tags": [
1787             "Events"
1788         ],
1789         "summary": "Subscribe to events against a specific resource",
1790         "description": "Subscribe to Resource level events by providing \"eventTypes\" you're

```

```

1791 interested in and \"eventsUrl\" API Endpoint where notifications will be sent to as defined. A
1792 successful response contains a \"subscriptionId\" which identifies the registered subscription and
1793 is part of each event. First notification for each registered event type is received immediately
1794 after subscription and contains the actual state of the resource, followed by new notifications in
1795 case of any change.\n\nSupported events:\n- \"resource_contentchanged\",
1796     \"parameters\": [
1797         {
1798             \"$ref\": \"#/parameters/ContentType\"
1799         },
1800         {
1801             \"$ref\": \"#/parameters/SubscribeRequestResources\"
1802         }
1803     ],
1804     \"consumes\": [
1805         \"application/json\"
1806     ],
1807     \"produces\": [
1808         \"application/json\"
1809     ],
1810     \"responses\": {
1811         \"201\": {
1812             \"$ref\": \"#/definitions/SubscribeResponse\"
1813         },
1814         \"400\": {
1815             \"$ref\": \"#/responses/BadRequest\"
1816         },
1817         \"401\": {
1818             \"$ref\": \"#/responses/Unauthorized\"
1819         },
1820         \"403\": {
1821             \"$ref\": \"#/responses/Forbidden\"
1822         },
1823         \"404\": {
1824             \"$ref\": \"#/responses/NotFound\"
1825         },
1826         \"406\": {
1827             \"$ref\": \"#/responses/NotAcceptable\"
1828         }
1829     },
1830     \"security\": [
1831         {
1832             \"oauth2\": [
1833                 \"r:*\"
1834             ]
1835         }
1836     ]
1837 },
1838 },
1839 \"/api/v1/devices/{deviceId}/{resourceLinkHref}/subscriptions/{subscriptionId}\": {
1840     \"parameters\": [
1841         {
1842             \"$ref\": \"#/parameters/CorrelationId\"
1843         },
1844         {
1845             \"$ref\": \"#/parameters/DeviceId\"
1846         },
1847         {
1848             \"$ref\": \"#/parameters/ResourceLinkHref\"
1849         },
1850         {
1851             \"$ref\": \"#/parameters/SubscriptionIdPath\"
1852         }
1853     ],
1854     \"delete\": {
1855         \"tags\": [
1856             \"Events\"
1857         ],
1858         \"summary\": \"Unsubscribe from events against a specific resource\",
1859         \"description\": \"Cancel the subscription identified by the provided \"subscriptionId\" that
1860 was returned in the response to the subscription request.\",
1861         \"responses\": {

```

```

1862         "202": {
1863             "description": "Subscription was marked for cancellation"
1864         },
1865         "400": {
1866             "$ref": "#/responses/BadRequest"
1867         },
1868         "401": {
1869             "$ref": "#/responses/Unauthorized"
1870         },
1871         "403": {
1872             "$ref": "#/responses/Forbidden"
1873         },
1874         "404": {
1875             "$ref": "#/responses/NotFound"
1876         }
1877     },
1878     "security": [
1879         {
1880             "oauth2": [
1881                 "r:*"
1882             ]
1883         }
1884     ]
1885 },
1886 },
1887 "/{eventsUrl}": {
1888     "post": {
1889         "tags": [
1890             "Events"
1891         ],
1892         "summary": "Events endpoint provided by the subscriber, where events are delivered",
1893         "description": "Events endpoint provided during subscription where notifications for the
1894 events specified in the subscription will be sent to as defined per event type. Confirmation of
1895 each notification sent to the \"{eventsUrl}\" endpoint is required with a \"2xx\" success
1896 code.\n\nNotifications you may receive based on the event type you're subscribed to are:\n -
1897 \"subscription_cancelled\": \"SubscriptionCancelledEvent\"\n - \"devices_registered\":
1898 \"DevicesRegisteredEvent\"\n - \"devices_unregistered\": \"DevicesUnregisteredEvent\"\n -
1899 \"resources_published\": \"ResourcesPublishedEvent\"\n - \"resources_unpublished\":
1900 \"ResourcesUnpublishedEvent\"\n - \"devices_online\": \"DevicesOnlineEvent\"\n -
1901 \"devices_offline\": \"DevicesOfflineEvent\"\n - \"resource_contentchanged\":
1902 \"ResourceContentChangedEvent\"",
1903         "parameters": [
1904             {
1905                 "$ref": "#/parameters/CorrelationId"
1906             },
1907             {
1908                 "$ref": "#/parameters/ContentType"
1909             },
1910             {
1911                 "$ref": "#/parameters/EventType"
1912             },
1913             {
1914                 "$ref": "#/parameters/SubscriptionId"
1915             },
1916             {
1917                 "$ref": "#/parameters/SequenceNumber"
1918             },
1919             {
1920                 "$ref": "#/parameters/EventSignature"
1921             },
1922             {
1923                 "$ref": "#/parameters/EventTimestamp"
1924             },
1925             {
1926                 "$ref": "#/parameters/EventsUrl"
1927             },
1928             {
1929                 "$ref": "#/parameters/Event"
1930             }
1931         ],
1932         "consumes": [

```

```

1933         "application/json",
1934         "application/vnd.ocf+cbor"
1935     ],
1936     "responses": {
1937         "200": {
1938             "description": "Event successfully recieved"
1939         },
1940         "400": {
1941             "$ref": "#/responses/BadRequest"
1942         },
1943         "410": {
1944             "description": "The subscription identified by the Subscription-ID header is no more in
1945 demand and shall be cancelled"
1946         }
1947     }
1948 }
1949 },
1950 },
1951 "securityDefinitions": {
1952     "oauth2": {
1953         "type": "oauth2",
1954         "flow": "accessToken",
1955         "authorizationUrl": "https://example.com/api/oauth/dialog",
1956         "tokenUrl": "https://example.com/api/oauth/token",
1957         "scopes": {
1958             "r:*": "Read device data",
1959             "w:*": "Update content of published resource"
1960         }
1961     }
1962 },
1963 "parameters": {
1964     "CorrelationId": {
1965         "name": "Correlation-ID",
1966         "in": "header",
1967         "type": "string",
1968         "format": "uuid",
1969         "description": "A Correlation ID, also known as a Transit ID, is a unique identifier value
1970 that is attached to requests and messages that allow reference to a particular transaction or event
1971 chain.\n"
1972     },
1973     "ContentType": {
1974         "name": "Content-Type",
1975         "in": "header",
1976         "type": "string",
1977         "enum": [
1978             "application/json",
1979             "application/vnd.ocf+cbor"
1980         ],
1981         "required": true,
1982         "description": "The Content-Type header is used to indicate the media type of the resource. In
1983 responses, a Content-Type header tells the client what the content type of the returned content
1984 actually is. In requests, (such as POST), the client tells the server what type of data is actually
1985 sent.\n"
1986     },
1987     "Accept": {
1988         "name": "Accept",
1989         "in": "header",
1990         "type": "string",
1991         "enum": [
1992             "application/json",
1993             "application/vnd.ocf+cbor"
1994         ],
1995         "description": "The Accept request header can be used to specify certain media types which are
1996 acceptable for the response. Accept headers can be used to indicate that the request is specifically
1997 limited to a small set of desired types.\n"
1998     },
1999     "SubscriptionId": {
2000         "name": "Subscription-ID",
2001         "in": "header",
2002         "description": "Unique id of the subscription",
2003         "type": "string",

```

```

2004     "format": "uuid",
2005     "required": true
2006   },
2007   "SequenceNumber": {
2008     "name": "Sequence-Number",
2009     "in": "header",
2010     "description": "Sequence number of the event; first event starting with number 0",
2011     "type": "string",
2012     "required": true
2013   },
2014   "EventSignature": {
2015     "name": "Event-Signature",
2016     "in": "header",
2017     "description": "The signature created by combining the `signingSecret` from the subscription
2018 request, headers and the body of the request using a stanard HMAC-SHA256 keyed hash.",
2019     "type": "string",
2020     "required": true
2021   },
2022   "EventTimestamp": {
2023     "name": "Event-Timestamp",
2024     "in": "header",
2025     "description": "Time when the event occurred in standard Unix time format",
2026     "type": "string",
2027     "required": true
2028   },
2029   "EventType": {
2030     "name": "Event-Type",
2031     "in": "header",
2032     "type": "string",
2033     "enum": [
2034       "subscription_cancelled",
2035       "devices_registered",
2036       "devices_unregistered",
2037       "resource_contentchanged",
2038       "resources_published",
2039       "resources_unpublished",
2040       "devices_online",
2041       "devices_offline"
2042     ],
2043     "required": true
2044   },
2045   "DeviceType": {
2046     "description": "Filter devices by device type",
2047     "name": "rt",
2048     "in": "query",
2049     "type": "array",
2050     "items": {
2051       "type": "string"
2052     }
2053   },
2054   "ResourceLinkHref": {
2055     "description": "Path to resource",
2056     "name": "resourceLinkHref",
2057     "in": "path",
2058     "type": "string",
2059     "required": true
2060   },
2061   "DeviceId": {
2062     "description": "Id of the device",
2063     "name": "deviceId",
2064     "in": "path",
2065     "type": "string",
2066     "format": "uuid",
2067     "required": true
2068   },
2069   "SubscriptionIdPath": {
2070     "name": "subscriptionId",
2071     "in": "path",
2072     "type": "string",
2073     "format": "uuid",
2074     "required": true

```

```

2075     },
2076     "BatchFormat": {
2077         "name": "content",
2078         "in": "query",
2079         "description": "Indicates to the recipient that the response payload shall be the resolved
2080 (i.e. resource representation) Link and not the Link itself. Default is `base`. When requesting
2081 `all`, additional scope `r:*` is required",
2082         "type": "string",
2083         "enum": [
2084             "base",
2085             "all"
2086         ]
2087     },
2088     "EventsUrl": {
2089         "name": "eventsUrl",
2090         "type": "string",
2091         "in": "path",
2092         "required": true
2093     },
2094     "ResourceUpdateRequest": {
2095         "description": "Map of resource values encoded to application/vnd.ocf+cbor type",
2096         "name": "content",
2097         "in": "body",
2098         "schema": {
2099             "$ref": "#/definitions/ResourceUpdateRequest"
2100         },
2101         "required": true
2102     },
2103     "SubscribeRequestDevices": {
2104         "name": "content",
2105         "in": "body",
2106         "schema": {
2107             "$ref": "#/definitions/SubscribeRequestDevices"
2108         },
2109         "required": true
2110     },
2111     "SubscribeRequestDevice": {
2112         "name": "content",
2113         "in": "body",
2114         "schema": {
2115             "$ref": "#/definitions/SubscribeRequestDevice"
2116         },
2117         "required": true
2118     },
2119     "SubscribeRequestResources": {
2120         "name": "content",
2121         "in": "body",
2122         "schema": {
2123             "$ref": "#/definitions/SubscribeRequestResources"
2124         },
2125         "required": true
2126     },
2127     "Event": {
2128         "description": "Event of a specific type, based on what you are subscribed to",
2129         "name": "content",
2130         "in": "body",
2131         "schema": {
2132             "$ref": "#/definitions/ResourceContentChangedEvent"
2133         },
2134         "required": true
2135     }
2136 },
2137 "responses": {
2138     "Unauthorized": {
2139         "description": "Unauthorized"
2140     },
2141     "NotFound": {
2142         "description": "Not found"
2143     },
2144     "SubscriptionCancellationPending": {
2145         "description": "Subscription was marked for cancellation"

```

```

2146     },
2147     "Forbidden": {
2148       "description": "Insufficient permissions"
2149     },
2150     "BadRequest": {
2151       "description": "The request was malformed or badly constructed"
2152     },
2153     "ServiceUnavailable": {
2154       "description": "The service on the Target Cloud is unavailable for the reason indicated in the
2155 diagnostic payload"
2156     },
2157     "GatewayTimeout": {
2158       "description": "The target Device is registered at the target Cloud, however the Device itself
2159 is unavailable, offline, or otherwise unreachable. The response should include a Retry-After header
2160 containing the time after which the request may be re-attempted. Additional information is indicated
2161 in the diagnostic payload."
2162     },
2163     "UnsupportedMediaType": {
2164       "description": "The request contained an unsupported media type in the Content-Type header"
2165     },
2166     "NotAcceptable": {
2167       "description": "The server cannot honour the Content-Type requested in the Accept header"
2168     }
2169   },
2170   "definitions": {
2171     "DeviceProperties": {
2172       "type": "object",
2173       "required": ["rt", "di", "dmn", "n"],
2174       "properties": {
2175         "rt": {
2176           "description": "Resource Type of the Resource",
2177           "items": {
2178             "type": "string",
2179             "maxLength": 64
2180           },
2181           "minItems": 1,
2182           "readOnly": true,
2183           "uniqueItems": true,
2184           "type": "array"
2185         },
2186         "di": {
2187           "allOf": [
2188             {
2189               "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
2190 schema.json#/definitions/uuid"
2191             },
2192             {
2193               "description": "Unique identifier for the Device",
2194               "readOnly": true
2195             }
2196           ]
2197         },
2198         "dmn": {
2199           "description": "Manufacturer Name.",
2200           "items": {
2201             "properties": {
2202               "language": {
2203                 "allOf": [
2204                   {
2205                     "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
2206 schema.json#/definitions/language-tag"
2207                   },
2208                   {
2209                     "description": "An RFC 5646 language tag.",
2210                     "readOnly": true
2211                   }
2212                 ]
2213               },
2214               "value": {
2215                 "description": "Manufacturer name in the indicated language.",
2216                 "maxLength": 64,

```

```

2217         "readOnly": true,
2218         "type": "string"
2219     },
2220 },
2221     "type": "object"
2222 },
2223     "minItems": 1,
2224     "readOnly": true,
2225     "type": "array"
2226 },
2227     "n": {
2228         "$ref": :
2229 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
2230 schema.json#/definitions/n"
2231     }
2232 },
2233 },
2234 "Device": {
2235     "type": "object",
2236     "required": ["device", "status", "links"],
2237     "properties": {
2238         "device": {
2239             "$ref": "#/definitions/DeviceProperties"
2240         },
2241         "status": {
2242             "$ref": "#/definitions/DeviceStatus"
2243         },
2244         "links": {
2245             "type": "array",
2246             "items": {
2247                 "$ref": :
2248 "http://openconnectivityfoundation.github.io/core/swagger2.0/oic.wk.res.swagger.json#/definitions/oi
2249 c.oic-link"
2250             }
2251         },
2252     },
2253     "example": {
2254         "device": {
2255             "rt": ["oic.wk.d", "oic.d.sensor"],
2256             "dmn": "Open Connectivity Foundation",
2257             "n": "Food safety sensor",
2258             "di": "53080a4f-5e3e-4291-802f-3436238232d2"
2259         },
2260         "status": "online",
2261         "links": [
2262             {
2263                 "href": "/53080a4f-5e3e-4291-802f-3436238232d2/oic/d",
2264                 "rt": [
2265                     "oic.wk.d",
2266                     "oic.d.sensor"
2267                 ],
2268                 "if": [
2269                     "oic.if.r",
2270                     "oic.if.baseline"
2271                 ]
2272             },
2273             {
2274                 "href": "/53080a4f-5e3e-4291-802f-3436238232d2/oic/p",
2275                 "rt": [
2276                     "oic.wk.p"
2277                 ],
2278                 "if": [
2279                     "oic.if.r",
2280                     "oic.if.baseline"
2281                 ]
2282             },
2283             {
2284                 "href": "/53080a4f-5e3e-4291-802f-3436238232d2/humidity",
2285                 "rt": [
2286                     "oic.r.humidity"
2287                 ],

```

```

2288         "if": [
2289             "oic.if.s",
2290             "oic.if.baseline"
2291         ]
2292     },
2293     {
2294         "href": "/53080a4f-5e3e-4291-802f-3436238232d2/temperature",
2295         "rt": [
2296             "oic.r.temperature"
2297         ],
2298         "if": [
2299             "oic.if.s",
2300             "oic.if.baseline"
2301         ]
2302     }
2303 ]
2304 }
2305 },
2306 "DeviceContentAll": {
2307     "type": "object",
2308     "required": ["device", "status", "links"],
2309     "properties": {
2310         "device": {
2311             "$ref": "#/definitions/DeviceProperties"
2312         },
2313         "status": {
2314             "$ref": "#/definitions/DeviceStatus"
2315         },
2316         "links": {
2317             "type": "array",
2318             "items": {
2319                 "type": "object",
2320                 "properties": {
2321                     "href": {
2322                         "type": "string"
2323                     },
2324                     "rep": {
2325                         "oneOf": [
2326                             {
2327                                 "type": "object"
2328                             },
2329                             {
2330                                 "type": "array"
2331                             }
2332                         ]
2333                     }
2334                 }
2335             }
2336         }
2337     },
2338     "example": {
2339         "device": {
2340             "rt": ["oic.wk.d", "oic.d.sensor"],
2341             "dmn": "Open Connectivity Foundation",
2342             "n": "Food safety sensor",
2343             "di": "53080a4f-5e3e-4291-802f-3436238232d2"
2344         },
2345         "status": "online",
2346         "links": [
2347             {
2348                 "href": "/53080a4f-5e3e-4291-802f-3436238232d2/oic/d",
2349                 "rep": {
2350                     "rt": ["oic.wk.d", "oic.d.sensor"],
2351                     "dmn": "Open Connectivity Foundation",
2352                     "n": "Food safety sensor",
2353                     "di": "53080a4f-5e3e-4291-802f-3436238232d2",
2354                     "icv": "ocf.2.0.5",
2355                     "dmv": "ocf.res.1.3.0, ocf.sh.1.3.0",
2356                     "piid": "6F0AAC04-2BB0-468D-B57C-16570A26AE48"
2357                 }
2358             }
2359         ]
2360     }
2361 }

```

```

2359     {
2360         "href": "/53080a4f-5e3e-4291-802f-3436238232d2/oic/p",
2361         "rep": {
2362             "pi": "54919CA5-4101-4AE4-595B-353C51AA983C",
2363             "mnfv": "1.1.20"
2364         }
2365     },
2366     {
2367         "href": "/53080a4f-5e3e-4291-802f-3436238232d2/humidity",
2368         "rep": {
2369             "humidity": 62,
2370             "desiredHumidity": 65
2371         }
2372     },
2373     {
2374         "href": "/53080a4f-5e3e-4291-802f-3436238232d2/temperature",
2375         "rep": {
2376             "temperature": 21,
2377             "units": "C"
2378         }
2379     }
2380 ]
2381 }
2382 },
2383 "DeviceStatus": {
2384     "description": "Device status available from the OCF Cloud, which tracks if the device has
opened TCP connection and is signed in",
2385     "type": "string",
2386     "enum": [
2387         "online",
2388         "offline"
2389     ]
2390 },
2391 },
2392 "ResourceUpdateRequest": {
2393     "type": "string",
2394     "description": "Desired content of the resource",
2395     "example": "o29kZXNpcmVkSHVtaWRpdHkYPGV0eXBlc4Fub2ljLnIuaHVtaWRpdHloaHVtaWRpdHkYKA=="
2396 },
2397 "ResourceRetrieveResponse": {
2398     "type": "string",
2399     "description": "Content of the resource returned from the device",
2400     "example": "o29kZXNpcmVkSHVtaWRpdHkYPGV0eXBlc4Fub2ljLnIuaHVtaWRpdHloaHVtaWRpdHkYKA=="
2401 },
2402 "EventType": {
2403     "type": "string",
2404     "enum": [
2405         "subscription_cancelled",
2406         "devices_registered",
2407         "devices_unregistered",
2408         "resource_contentchanged",
2409         "resources_published",
2410         "resources_unpublished",
2411         "devices_online",
2412         "devices_offline"
2413     ]
2414 },
2415 "EventTypeDevices": {
2416     "type": "string",
2417     "enum": [
2418         "devices_registered",
2419         "devices_unregistered",
2420         "devices_online",
2421         "devices_offline"
2422     ]
2423 },
2424 "EventTypeDevice": {
2425     "type": "string",
2426     "enum": [
2427         "resources_published",
2428         "resources_unpublished"
2429     ]

```

```

2430 },
2431 "EventTypeResources": {
2432   "type": "string",
2433   "enum": [
2434     "resource_contentchanged"
2435   ]
2436 },
2437 "SubscriptionId": {
2438   "description": "Unique id of the subscription",
2439   "type": "string",
2440   "format": "uuid"
2441 },
2442 "SubscribeRequestDevices": {
2443   "type": "object",
2444   "properties": {
2445     "eventsUrl": {
2446       "$ref": "#/definitions/EventsUrl"
2447     },
2448     "eventTypes": {
2449       "type": "array",
2450       "items": {
2451         "$ref": "#/definitions/EventTypeDevices"
2452       }
2453     },
2454     "signingSecret": {
2455       "type": "string",
2456       "maxLength": 32,
2457       "minLength": 32
2458     }
2459   },
2460   "required": [
2461     "eventsUrl",
2462     "eventTypes",
2463     "signingSecret"
2464   ],
2465   "example": {
2466     "eventsUrl": "https://events.example.com/",
2467     "eventTypes": [
2468       "devices_registered",
2469       "devices_unregistered"
2470     ],
2471     "signingSecret": "3BZ6oI9xbRJzOUvUoRb5RgaZjPqHrmql"
2472   }
2473 },
2474 "SubscribeRequestDevice": {
2475   "type": "object",
2476   "properties": {
2477     "eventsUrl": {
2478       "$ref": "#/definitions/EventsUrl"
2479     },
2480     "eventTypes": {
2481       "type": "array",
2482       "items": {
2483         "$ref": "#/definitions/EventTypeDevice"
2484       }
2485     },
2486     "signingSecret": {
2487       "type": "string",
2488       "maxLength": 32,
2489       "minLength": 32
2490     }
2491   },
2492   "required": [
2493     "eventsUrl",
2494     "eventTypes",
2495     "signingSecret"
2496   ],
2497   "example": {
2498     "eventsUrl": "https://events.example.com/",
2499     "eventTypes": [
2500       "resource_published",

```

```

2501         "resource_unpublished"
2502     ],
2503     "signingSecret": "3BZ6oI9xbRJzOUvUoRb5RgaZjPqHrmql"
2504 }
2505 },
2506 "SubscribeRequestResources": {
2507     "type": "object",
2508     "properties": {
2509         "eventsUrl": {
2510             "$ref": "#/definitions/EventsUrl"
2511         },
2512         "eventTypes": {
2513             "type": "array",
2514             "items": {
2515                 "$ref": "#/definitions/EventTypeResources"
2516             }
2517         },
2518         "signingSecret": {
2519             "type": "string",
2520             "maxLength": 32,
2521             "minLength": 32
2522         }
2523     },
2524     "required": [
2525         "eventsUrl",
2526         "eventTypes",
2527         "signingSecret"
2528     ],
2529     "example": {
2530         "eventsUrl": "https://events.example.com/",
2531         "eventTypes": [
2532             "resource_contentchanged"
2533         ],
2534         "signingSecret": "3BZ6oI9xbRJzOUvUoRb5RgaZjPqHrmql"
2535     }
2536 },
2537 "SubscribeResponse": {
2538     "description": "Subscription was registered, waiting for verification",
2539     "type": "object",
2540     "properties": {
2541         "subscriptionId": {
2542             "$ref": "#/definitions/SubscriptionId"
2543         }
2544     },
2545     "required": [
2546         "subscriptionId"
2547     ],
2548     "example": {
2549         "subscriptionId": "1eeb465c-5e8d-4305-a366-bbf035fff671"
2550     }
2551 },
2552 "EventsUrl": {
2553     "type": "string",
2554     "format": "url",
2555     "example": "https://events.example.com/"
2556 },
2557 "SubscriptionCancelledEvent": {
2558     "type": "object",
2559     "description": "Subscription with provided id was cancelled"
2560 },
2561 "DevicesRegisteredEvent": {
2562     "description": "Device was successfully signed up to the OCF Cloud, as defined in the
2563 `oic.sec.account`",
2564     "type": "object",
2565     "properties": {
2566         "content": {
2567             "type": "array",
2568             "items": {
2569                 "properties": {
2570                     "di": {
2571                         "type": "string",

```

```

2572         "format": "uuid"
2573     }
2574 }
2575 }
2576 }
2577 }
2578 },
2579 "DevicesUnregisteredEvent": {
2580     "description": "Device was successfully signed off from the OCF Cloud, as defined in the
2581 `oic.sec.account`,
2582     "type": "object",
2583     "properties": {
2584         "content": {
2585             "type": "array",
2586             "items": {
2587                 "properties": {
2588                     "di": {
2589                         "type": "string",
2590                         "format": "uuid"
2591                     }
2592                 }
2593             }
2594         }
2595     },
2596     "ResourcesPublishedEvent": {
2597         "type": "object",
2598         "properties": {
2599             "content": {
2600                 "type": "array",
2601                 "items": {
2602                     "$ref":
2603 "http://openconnectivityfoundation.github.io/core/swagger2.0/oic.wk.res.swagger.json#/definitions/oi
2604 c.oic-link"
2605                 }
2606             }
2607         }
2608     },
2609 },
2610 "ResourcesUnpublishedEvent": {
2611     "type": "object",
2612     "properties": {
2613         "content": {
2614             "type": "array",
2615             "items": {
2616                 "$ref":
2617 "http://openconnectivityfoundation.github.io/core/swagger2.0/oic.wk.res.swagger.json#/definitions/oi
2618 c.oic-link"
2619             }
2620         }
2621     },
2622 },
2623 "DevicesOnlineEvent": {
2624     "type": "object",
2625     "properties": {
2626         "content": {
2627             "type": "array",
2628             "items": {
2629                 "properties": {
2630                     "di": {
2631                         "type": "string",
2632                         "format": "uuid"
2633                     }
2634                 }
2635             }
2636         }
2637     },
2638 },
2639 "DevicesOfflineEvent": {
2640     "type": "object",
2641     "properties": {
2642         "content": {

```

```
2643     "type": "array",
2644     "items": {
2645       "properties": {
2646         "di": {
2647           "type": "string",
2648           "format": "uuid"
2649         }
2650       }
2651     }
2652   }
2653 }
2654 },
2655 "ResourceContentChangedEvent": {
2656   "type": "string",
2657   "description": "New Content of the resource returned from the device",
2658   "example": "o29kZXNpcmVkSHVtaWRpdHkYPGV0eXBlc4Fub2ljLnIuaHVtaWRpdHloaHVtaWRpdHkYKA=="
2659 }
2660 }
2661 }
2662 }
```

DRAFT