

OCF Cloud API for Cloud Services Specification

VERSION 2.2.1 | December 2020



OPEN CONNECTIVITY
FOUNDATION™

CONTACT admin@openconnectivity.org

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

Legal Disclaimer

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

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

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

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

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

CONTENTS

20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62

Introduction.....	viii
1 Scope.....	1
2 Normative references	1
3 Terms, definitions, and abbreviated terms	2
3.1 Terms and definitions.....	2
3.2 Symbols and abbreviated terms	2
4 Document conventions and organization.....	3
4.1 Conventions.....	3
4.2 Notation	3
5 Overview	5
5.1 Introduction.....	5
5.2 OCF Cloud Architecture Alignment with ISO IEC 17789	5
5.3 General OCF Cloud API for Cloud Services elements	6
5.4 Cloud to Cloud operational overview	7
5.4.1 Introduction	7
5.4.2 Conceptual architecture.....	7
5.4.3 Authorizing Cloud connectivity.....	7
5.4.4 Synchronization of User's set of Devices	7
5.4.5 Keeping up-to-date: Notifications of changes on other Clouds	8
5.4.6 Handling of Requests and Responses for connected Devices	8
6 Authentication and authorization.....	8
7 Account Linking API	8
7.1 General.....	8
7.2 OAuth2.0 access token scopes	10
8 Devices API.....	11
8.1 Introduction.....	11
8.2 Parameters supported in Requests	11
8.3 Retrieve all Devices	12
8.3.1 Summary	12
8.3.2 Request and Response payload	12
8.3.3 Responses	13
8.4 Retrieve one Device.....	14
8.4.1 Summary	14
8.4.2 Request and Response payload	14
8.4.3 Responses	15
8.5 Retrieve specific Resource	15
8.5.1 Summary	15
8.5.2 Request and Response payload	16
8.5.3 Responses	16
8.6 Update a Resource on a Device.....	17
8.6.1 Summary	17

63	8.6.2	Request and Response payload	17
64	8.6.3	Responses	18
65	9	Events API	18
66	9.1	Introduction.....	18
67	9.2	Events authentication.....	19
68	9.2.1	Introduction	19
69	9.2.2	Create event signature	19
70	9.2.1	Verify the event signature	20
71	9.3	Parameters supported.....	20
72	9.4	Events API subscription and notification payload definitions	21
73	9.4.1	Subscription request.....	21
74	9.4.2	Subscription response	22
75	9.4.3	Notification request.....	22
76	9.4.4	Notification response	24
77	9.5	Subscribe and unsubscribe to devices level event types	24
78	9.5.1	Summary	24
79	9.5.2	Request and Response payload	25
80	9.5.3	Responses	25
81	9.6	Subscribe and unsubscribe to device level events.....	25
82	9.6.1	Summary	25
83	9.6.2	Request and Response payload	26
84	9.6.3	Responses	26
85	9.7	Subscribe and unsubscribe to resource level events	27
86	9.7.1	Summary	27
87	9.7.2	Request and Response payload	27
88	9.7.3	Responses	27
89	9.8	Notification of devices level events	28
90	9.8.1	Summary	28
91	9.8.2	Request and Response payload	28
92	9.8.3	Responses	29
93	9.9	Notification of Device level events	29
94	9.9.1	Summary	29
95	9.9.2	Request and Response payload	29
96	9.9.3	Responses	29
97	9.10	Notification of Resource level events	30
98	9.10.1	Summary	30
99	9.10.2	Request and Response payload	30
100	9.10.3	Responses	30
101		Annex A Representative Flows	32
102	A.1	Introduction.....	32
103	A.2	OAuth2.0 application registration	32
104	A.3	Account linking	32
105	A.4	Retrieval of all Devices	33
106	A.4.1	Summary	33
107	A.4.2	Flow	33

108	A.4.3	Flow description	34
109	A.5	Retrieval of a single Device	34
110	A.5.1	Summary	34
111	A.5.2	Flow	34
112	A.5.3	Flow description	35
113	A.6	Retrieval of a single Resource	35
114	A.6.1	Summary	35
115	A.6.2	Flows.....	35
116	A.7	Update of a single Resource	37
117	A.7.1	Summary	37
118	A.7.2	Flows.....	37
119	A.8	Establishment of new subscription request.....	38
120	A.8.1	Summary	38
121	A.8.2	Flows.....	38
122	A.9	Event generated for a subscription.....	39
123	A.9.1	Summary	39
124	A.9.2	Flows.....	39
125	A.10	Addition of new registration.....	39
126	A.10.1	Summary	39
127	A.10.2	Flows.....	40
128	A.11	Removal of existing device registration	40
129	A.11.1	Summary	40
130	A.11.2	Flows.....	40
131	Annex B	Open API Definition	41
132	B.1	OCF Cloud API for Cloud Services.....	41
133	B.1.1	Supported APIs	41
134	B.1.2	OpenAPI 2.0 definition.....	42
135			
136			

137
138
139

Figures

140	Figure 1 – OCF Cloud Overview	5
141	Figure 2 – Conceptual Architecture	7
142	Figure 3 – Subscription Request Example.....	22
143	Figure 4 – Subscription Response Example Payload.....	22
144	Figure A.1 – Establish business relationship example flow.....	32
145	Figure A.2 – Initial association example flow.....	33
146	Figure A.3 – Retrieve all Devices example flow.....	34
147	Figure A.4 – Retrieve single Device example flow.....	35
148	Figure A.5 – Retrieve Resource (success) example flow.....	36
149	Figure A.6 – Retrieve Resource (timeout) example flow.....	37
150	Figure A.7 – Update Resource (success) example flow.....	37
151	Figure A.8 – Update Resource (timeout) example flow.....	38
152	Figure A.9 – Subscription establishment example flow.....	39
153	Figure A.10 – "resource_contentchanged" event example flow.....	39
154	Figure A.11 – Addition of new registered Device example flow.....	40
155	Figure A.12 – Removal of existing registration example flow.....	40

156
157

Tables

158		
159		
160	Table 1 – OAuth 2.0 access token scopes	10
161	Table 2 – Applicable OAuth2.0 access token scopes per API Endpoint	10
162	Table 3 – Parameters used in Requests in the Device API	11
163	Table 4 – Retrieve All Devices API Summary	12
164	Table 5 – Response payload Property definition	12
165	Table 6 – "device" Property definition	13
166	Table 7 – Devices API non-success path responses	13
167	Table 8 – Retrieve One Device API Summary	14
168	Table 9 – Device API non-success path responses	15
169	Table 10 – Retrieve Specific Resource API Summary	15
170	Table 11 – Resource Retrieval API non-success path responses	16
171	Table 12 – Update Resource API Summary	17
172	Table 13 – Resource Update API non-success path responses	18
173	Table 14 – Parameters used in the Events API	20
174	Table 15 – Event types and API Endpoints	21
175	Table 16 – Subscription Request Payload Properties	21
176	Table 17 – Subscription Response Properties	22
177	Table 18 – Notification request HTTP Headers	23
178	Table 19 – Event type to notification payload content	23
179	Table 20 – Subscription to /devices API Summary	24
180	Table 21 – Devices Event Subscription API non-success path responses	25
181	Table 22 – Subscription to Single Device API Summary	26
182	Table 23 – Device Event Subscription API non-success path responses	26
183	Table 24 – Subscription to Resource API Summary	27
184	Table 25 – Resource Event Subscription API non-success path responses	28
185	Table 26 – Notification of /devices API Summary	28
186	Table 27 – Devices Event Notification non-success path responses	29
187	Table 28 – Notification of Single Device API Summary	29
188	Table 29 – Device Event Notification non-success path responses	30
189	Table 30 – Notification of Resource API Summary	30
190	Table 31 – Resource Event Notification non-success path responses	31
191	Table A.1 – Retrieve all Devices flow summary	34
192	Table A.2 – Retrieve single Device flow summary	35
193	Table A.3 – Retrieve single Resource flow summary	36
194	Table A.4 – Update single Resource flow summary	38
195		
196		

Introduction

198 This document, and all the other parts associated with this document, were developed in response
199 to worldwide demand for smart home focused Internet of Things (IoT) devices, such as appliances,
200 door locks, security cameras, sensors, and actuators; these to be modelled and securely controlled,
201 locally and remotely, over an IP network.

202 While some inter-device communication existed, no universal language had been developed for
203 the IoT. Device makers instead had to choose between disparate frameworks, limiting their market
204 share, or developing across multiple ecosystems, increasing their costs. The burden then falls on
205 end users to determine whether the products they want are compatible with the ecosystem they
206 bought into, or find ways to integrate their devices into their network, and try to solve interoperability
207 issues on their own.

208 In addition to the smart home, IoT deployments in commercial environments are hampered by a
209 lack of security. This issue can be avoided by having a secure IoT communication framework, which
210 this standard solves.

211 The goal of these documents is then to connect the next 25 billion devices for the IoT, providing
212 secure and reliable device discovery and connectivity across multiple OSs and platforms. There
213 are multiple proposals and forums driving different approaches, but no single solution addresses
214 the majority of key requirements. This document and the associated parts enable industry
215 consolidation around a common, secure, interoperable approach.

216 **1 Scope**

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

219 **2 Normative references**

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

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

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

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

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

232 IETF RFC 7628, *A Set of Simple Authentication and Security Layer (SASL) Mechanisms for*
233 *OAuth*, August 2015
234 <https://www.rfc-editor.org/info/rfc7628>

235 IETF RFC 8075, *Guidelines for Mapping Implementations: HTTP to the Constrained Application*
236 *Protocol (CoAP)*, February 2017
237 <https://tools.ietf.org/html/rfc8075>

238 *A Set of Simple Authentication and Security Layer (SASL) Mechanisms for OAuth*, August 2015
239 <https://www.rfc-editor.org/info/rfc7628>

240 ISO/IEC 17788 *Information technology – Cloud computing – Overview and vocabulary*
241 <https://www.iso.org/standard/60544.html>

242 ISO/IEC 17789 *Information technology – Cloud computing – Reference architecture*
243 <https://www.iso.org/standard/60545.html>

244 ISO/IEC 30118-1 Information technology -- Open Connectivity Foundation (OCF) Specification --
245 Part 1: Core specification
246 <https://www.iso.org/standard/53238.html>
247 Latest version available at: https://openconnectivity.org/specs/OCF_Core_Specification.pdf

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

252 OCF Cloud Security, *Open Connectivity Foundation Cloud Security, Version 2.2.0*
253 Available at: https://openconnectivity.org/specs/OCF_Cloud_Security_Specification_v2.2.0.pdf
254 Latest version available at:
255 https://openconnectivity.org/specs/OCF_Cloud_Security_Specification.pdf

256
257 OCF Device to Cloud Services, *Open Connectivity Foundation Device to Cloud Services*
258 *Specification, Version 2.2.0*
259 Available at:
260 https://openconnectivity.org/specs/OCF_Device_To_Cloud_Services_Specification_v2.2.0.pdf
261 Latest version available at:

262 https://openconnectivity.org/specs/OCF_Device_To_Cloud_Services_Specification.pdf

263 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)
264 [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)

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

267 **3 Terms, definitions, and abbreviated terms**

268 **3.1 Terms and definitions**

269 For the purposes of this document, the terms and definitions given in ISO/IEC 30118-1 and ISO/IEC
270 30118-2 and the following apply.

271 ISO and IEC maintain terminological databases for use in standardization at the following
272 addresses:

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

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

275 **3.1.1**

276 **API Endpoint**

277 defined URL to which requests defined in this document are sent

278 **3.1.2**

279 **Bearer Token**

280 OAuth2.0 access token as defined within IETF RFC 6750

281 **3.1.3**

282 **Origin Cloud**

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

284 **3.1.4**

285 **Subscription ID**

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

287 **3.1.5**

288 **Target Cloud**

289 OCF Cloud to which OCF Servers (OCF Devices) are connected which the user wants to control
290 via the *Origin Cloud* (3.1.2)

291 **3.2 Symbols and abbreviated terms**

292 API Application Programming Interface

293 HMAC Hash-based Message Authentication Code

294

295 **4 Document conventions and organization**

296 **4.1 Conventions**

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

301 In this document, to be consistent with the IETF usages for RESTful operations, the RESTful
302 operation words CRUDN, CREATE, RETRIVE, UPDATE, DELETE, and NOTIFY will have all letters
303 capitalized. Any lowercase uses of these words have the normal technical English meaning.

304 **4.2 Notation**

305 In this document, features are described as required, recommended, allowed or DEPRECATED as
306 follows:

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

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

311 Recommended (or should)(S).

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

318 Allowed (may or allowed)(O).

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

322 DEPRECATED.

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

329 Conditionally allowed (CA)

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

332 Conditionally required (CR)

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

336

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

338 Words that are emphasized are printed in italic.

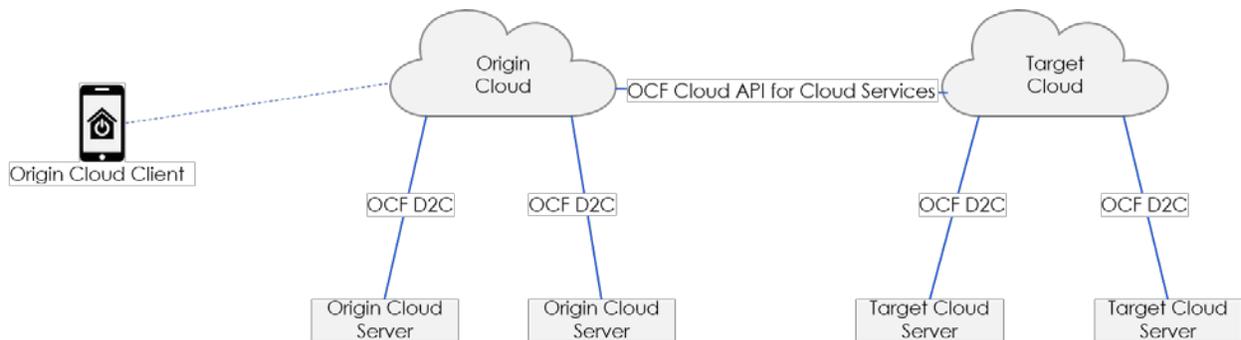
339 **5 Overview**

340 **5.1 Introduction**

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

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

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



350

351

Figure 1 – OCF Cloud Overview

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

- 353 – Account Linking API (clause 7)
 - 354 – Initial Account Linking
 - 355 – Removal of linked account
- 356 – Devices API (clause 8)
 - 357 – Retrieval of all Devices associated with a User (clause 8.3)
 - 358 – Retrieval of a single Device associated with a User (clause 8.4)
 - 359 – Retrieval of a single Resource (clause 8.5)
 - 360 – Update of a single Resource (clause 8.6)
- 361 – Events API (clause 9)
 - 362 – Subscription to an event: establishment of a subscription (clause 9.4.1)
 - 363 – Notification: event generated on an established subscription (clause 9.4.3)

364 **5.2 OCF Cloud Architecture Alignment with ISO IEC 17789**

365 Reference ISO/IEC 17789 defines a cloud computing reference architecture (CCRA) which can be
366 described in terms of one of four architectural viewpoints; user, functional, implementation, and
367 deployment. Of the four viewpoints, implementation and deployment are explicitly out of scope of
368 ISO/IEC 17789.

369 OCF defines an application capabilities type cloud service, providing Communication as a Service
370 (CaaS) (reference ISO/IEC 17788). This cloud service is provided by a cloud service provider, the
371 mechanisms used by the cloud service provider in managing their overall cloud infrastructure are

372 outside the scope of the OCF defined cloud service. The OCF definition is specific to the interface
373 offered by the cloud service to the cloud service customer, specifically the cloud service user.

374
375 There are three different user views defined. In the case where the cloud service customer is an
376 OCF Device as specified in OCF Device to Cloud Services then the views provided are:

- 377 - Interface for the OCF Device to provide information to the cloud service
- 378 - Interface for the OCF Device to retrieve information that has been provided to the cloud
379 service

380
381 In the case where the cloud service customer is another instance of a cloud service as specified in
382 this document then the view provided is:

- 383 - Interface for the other cloud service instance to retrieve and update the information that is
384 provided via the cloud service

385
386 The OCF cloud service pertains specifically to a cloud service user, there is a single applicable
387 cloud service activity, that of "Use cloud service" defined in clause 8.2.21 of ISO/IEC 17789.

388
389 Credentials for the user of the cloud service are provided using OAuth2.0 as defined by RFC 6749.
390 The cloud service, either itself, or leveraging an external authorization server, provides a bearer
391 token that is required in all requests from all cloud users. Please see clause 7 and OCF Cloud
392 Security.

393
394 All connectivity between a cloud user and the cloud service is via mutually authenticated TLS; see
395 clause 7.1 of OCF Cloud Security.

396 **5.3 General OCF Cloud API for Cloud Services elements**

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

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

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

405 An Origin Cloud shall obtain a Bearer Token from the Target Cloud using standard OAuth2.0 (IETF
406 RFC 6749) mechanisms. All subsequent requests from an Origin Cloud to the Target Cloud shall
407 OCF Device to Cloud Services include this Bearer Token for the user in question.

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

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

415 All requests shall include an HTTP Accept header with the exception of a DELETE (as there is no
416 payload expected in the response). All requests or responses that carry content shall include an
417 HTTP Content-Type header. At a minimum media-types "application/json" and
418 "application/vnd.ocf+cbor" shall be supported. If the recipient of a request cannot provide a
419 response that is encoded according to the content of the Accept header, then a HTTP 406 (not
420 acceptable) response should be sent in accordance with IETF RFC 2818. On reception of a 406

421 response the originator of the request may re-attempt the request using an alternative Content-
422 Type if supported.

423 Any responses that are sent by an OCF Cloud may include a diagnostic payload (see ISO/IEC
424 30118-1). If a diagnostic payload is included in a response, the response shall have a Content-
425 Type header encoded as "text/plain", see also IETF RFC 8075 clause 6.6.

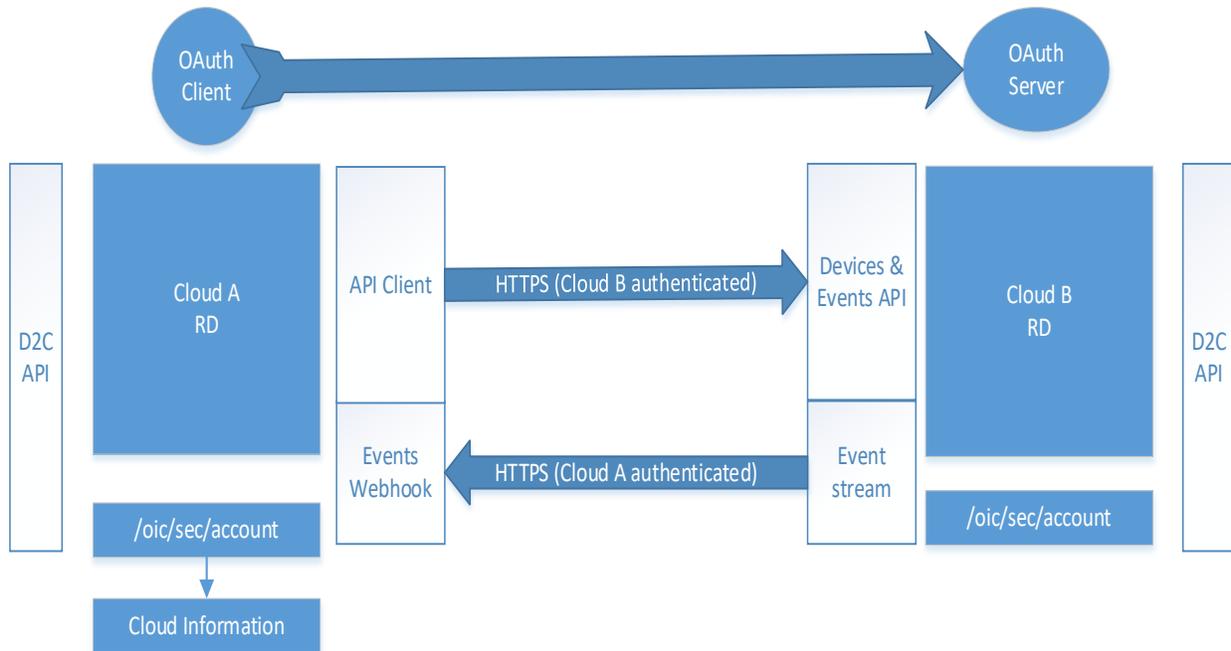
426 5.4 Cloud to Cloud operational overview

427 5.4.1 Introduction

428 This clause provides an informative overview of the flows that are enabled by the detailed API
429 defined in clauses 6, 7, 8, and 9. Clause 5.4 provides references to the applicable clauses within
430 this document that define the API specifics.

431 5.4.2 Conceptual architecture

432 Figure 2 describes the overall conceptual architecture.



433

434

Figure 2 – Conceptual Architecture

435 5.4.3 Authorizing Cloud connectivity

436 Consider a user who has accounts on two distinct, separately owned clouds, and devices
437 associated with each of those accounts on those clouds. The user wants to have a unified view of
438 all of their devices from a single client rather than having a client per cloud. The user via the client
439 they want to use for all devices indicates to the directly connected cloud (Origin Cloud) that they
440 want to link this account with an account on the other cloud (Target Cloud). This initiates a standard
441 OAuth2.0 authorization code grant type flow, see IETF RFC 6749, clause 1.3.1. Application of this
442 flow is described in clause 7.

443 5.4.4 Synchronization of User's set of Devices

444 After completion of the authorization code grant type flow from clause 5.4.3 the Origin Cloud (that
445 is the cloud to which the user is connected) is authorized to use the Device API to obtain on behalf
446 of the user the complete list of devices hosted on the Target Cloud for which the user has access.
447 The API is described in clause 8, and the flow is further illustrated in clause A.4.

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

450 **5.4.5 Keeping up-to-date: Notifications of changes on other Clouds**

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

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

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

458 When a new Device registers on the Target Cloud, and a subscription exists for that event, then a
459 notification is sent to the Origin Cloud with an event type of "devices_registered" and a payload
460 which contains the "di" of the newly registered device. The Origin Cloud may then RETRIEVE the
461 Links exposed by the newly added device using "/devices/{deviceid}" where "deviceid" was
462 provided in the payload of the notification. See clause A.10 for a flow illustrating this interaction.

463 **5.4.6 Handling of Requests and Responses for connected Devices**

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

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

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

476 **6 Authentication and authorization**

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

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

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

486 **7 Account Linking API**

487 **7.1 General**

488 The account linking API is the mechanism by which Devices hosted on behalf of a user by the
489 Target Cloud are linked with a user identity on the Origin Cloud. Account linking is established

490 solely between the Origin Cloud and the Target Cloud; an Origin Cloud shall not proxy devices
491 from the Target Cloud to another third-party Cloud.

492 The OAuth 2.0 Origin Cloud Client has to be registered with the Target Cloud as a prerequisite to
493 initiating the Authorization Code Grant Type flow, which allows the user to link his Origin Cloud
494 account with the Target Cloud. This process is named OAuth application registration and is beyond
495 the scope of this document. Successful registration of the OAuth 2.0 Origin Cloud Client in the
496 Target Cloud relies on the two entities establishing trust and obtaining the required client
497 parameters and OAuth2.0 Token Endpoints (e.g. client id, client secret, allowed redirect URIs).
498 See IETF RFC 6749, clause 2.

499 The linking is then achieved via the use of an OAuth2.0 authorization code grant type. Part of the
500 linking process is the end-user consent, which is very important in cross-domain identity federation,
501 ensuring that a malicious OAuth 2.0 Origin Cloud Client cannot obtain authorization without the
502 awareness and explicit consent of the resource owner (that is the user) of the Target Cloud. The
503 Target Cloud presents to the user linking the account the precise scope of authorization information
504 being requested by the Client. Details about scopes are available in clause 7.2. After the user's
505 consent and subsequent authorization code exchange, the Bearer Token and refresh tokens (see
506 IETF RFC 6749) shall be obtained from the Target Cloud by the Origin Cloud, following the format
507 and Content Type in IETF RFC 6750 clause 4. The Bearer Token identifies a user identity on the
508 Target Cloud. All requests for a Bearer Token or a refresh token shall include the "client_id" and
509 "client_secret" as defined by IETF RFC 6749. IETF RFC 6749 clause 2.3.1 describes two schemes
510 for inclusion of the "client_id" and "client_secret", one using an Authorization header with a "Basic"
511 scheme, and one that encodes the client credentials in the request body which is not recommended
512 by the referenced RFC. A Client shall provide an Authorization header in requests using the "Basic"
513 scheme, a Client should not encode the information in the request body.

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

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

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

523 Once such a Bearer Token has been acquired, the Origin Cloud shall link the OAuth2.0 access and
524 refresh token with its known local "userid". The user who linked his Target Cloud account with the
525 Origin Cloud account is from this moment able to request all his devices through the Origin Cloud,
526 because the Origin Cloud can make requests to the Target Cloud on behalf of the Target Cloud
527 user account. However, if an Origin Cloud makes a request that is not included in the OAuth2.0
528 access token scope granted by the Bearer Token, the Target Cloud shall reply with an appropriate
529 error response.

530 When a Bearer Token is first acquired, it is recommended that the Origin Cloud use the Device API
531 to retrieve the Device details for all Devices in OAuth2.0 access token scope of the Bearer Token.

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

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

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

548 **7.2 OAuth2.0 access token scopes**

549 This document defines a core set of OAuth2.0 access token scopes, see IETF RFC 6749. An Origin
 550 Cloud may request one or more of these scopes, a vendor extension thereof, or a vendor specific
 551 scope(s) as part of the account linking process. If the scope being provided by the Target Cloud is
 552 different from the requested scope, then that scope shall be included in the issued access token
 553 (see clause 5.1 of IETF RFC 6749). If the Target Cloud supports access token requests with no
 554 scopes provided, and an access token request with no scopes is received from the Origin Cloud,
 555 then the returned access token from the Target Cloud shall grant access to all of the OAuth2.0
 556 access token scopes defined in Table 1.

557 The description for each of the OAuth2.0 access token scopes shall be presented to the user during
 558 the account linking process by the OAuth2.0 server of the Target Cloud. The Target Cloud user
 559 sees a description on the consent screen and give an explicit consent that the Origin Cloud
 560 requesting that the Bearer Token is authorized to act on behalf of the user in the boundary of
 561 obtained OAuth2.0 access token scopes.

562 **Table 1 – OAuth 2.0 access token scopes**

OAuth2.0 access token scope name	OAuth2.0 access token scope description "The application will be able to:"
r:*	Read
w:*	Update

563
 564 Table 2 details the OAuth2.0 access token scopes that are applicable per API Endpoint. All API
 565 Endpoints that are listed in Table 2 shall be supported by a Target Cloud. So, for example, if an
 566 Origin Cloud sends a GET request to "/api/v1/devices?content=all" API Endpoint, the Origin Cloud
 567 must have a Bearer Token that contains OAuth2.0 access token scope "r:*" or a vendor extension
 568 thereof

569 **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:*

570 A vendor may extend the list of OAuth2.0 access token scopes beyond those listed in Table 2.
571 They are extended by adding additional vendor-specific information before the * in the OAuth2.0
572 access token scope name (e.g. "r:xyz:*"). How these extensions work is outside the scope of the
573 OCF but they may be present in the OAuth2.0 access token request. Note that if the user gives
574 consent to the Origin Cloud to "w:*", consent applies also to any derived OAuth2.0 access token
575 scopes (e.g. "w:xyz:*").

576 8 Devices API

577 8.1 Introduction

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

580 8.2 Parameters supported in Requests

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

582 **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

				<p>the response payload Links are not resolved.</p> <p>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.</p>
--	--	--	--	--

583

584 **8.3 Retrieve all Devices**

585 **8.3.1 Summary**

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

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

593 Table 4 provides a summary of the API.

594 **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.

595 **8.3.2 Request and Response payload**

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

603 **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
--------------	---------	---------	-----	-----	---	-----	--

604

605

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 UUID	"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.

606

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

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

615 8.3.3 Responses

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

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

622

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

623

624 8.4 Retrieve one Device

625 8.4.1 Summary

626 This request may be sent from the Origin Cloud to the Target Cloud in order to obtain information
627 on a specific Device that is registered for the user that is in scope as defined by the Bearer Token
628 on the Target Cloud.

629 A request to this API may be invoked at the Origin Cloud following reception of a notification that
630 a new Device has been added to a partner cloud, or alternatively as part of the flow following
631 account linking. Where the Origin Cloud supports OCF Device to Cloud Services, a request to this
632 API may also be invoked following reception of a RETRIEVE to "/oic/res" of the Origin Cloud
633 Resource Directory from an OCF Client with a query parameter that specifies a particular "deviceid"
634 (i.e. "?anchor=ocf://<some device uuid>").

635 Table 8 provides a summary of the API.

636 **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

637

638 8.4.2 Request and Response payload

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

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

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

644 **8.4.3 Responses**

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

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

651 **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

652

653 **8.5 Retrieve specific Resource**

654 **8.5.1 Summary**

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

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

660 Table 10 provides a summary of the API.

661 **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.

662

663 **8.5.2 Request and Response payload**

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

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

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

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

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

676 **8.5.3 Responses**

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

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

683 **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

684

685 8.6 Update a Resource on a Device

686 8.6.1 Summary

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

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

693 Table 12 provides a summary of the API.

694

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

695

696 8.6.2 Request and Response payload

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

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

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

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

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

712 8.6.3 Responses

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

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

719 **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

720

721 9 Events API

722 9.1 Introduction

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

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

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

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

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

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

749 **9.2 Events authentication**

750 **9.2.1 Introduction**

751 Hash-based Message Authentication Code (HMAC) signatures are a way to sign the notification
752 data using the "signingSecret" that only the Origin Cloud and Target Cloud know. The
753 "signingSecret" shall be created by the Origin Cloud and sent within the subscription request as
754 defined in the clause 9.4.1. After a successful subscription, the Target Cloud shall sign each
755 notification using the HMAC-SHA256 hashing algorithm following the formula from the clause 9.2.2.
756 The calculated signature shall be attached as the "Event-Signature" header with each notification
757 request sent to the Origin Cloud.

758 The signature shall be used by the Origin Cloud to verify the legitimacy of the source and data
759 itself. When the notification is received by the Origin Cloud it shall use its stored secret and the
760 notification to generate its own HMAC-SHA256 signature using the formula from clause 9.2.1 to
761 compare with the value from the "Event-Signature" header.

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

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

766 Detailed overview is provided in A.8.2, A.9.2, A.10.2, and A.11.2.

767 **9.2.2 Create event signature**

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

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

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

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

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

784 **9.2.1 Verify the event signature**

785 1) Create a string, that is made up of the concatenation of the encoded content of the following
 786 headers received in the notification, in order: "Content-Type", "Event-Type", "Subscription-ID",
 787 "Sequence-Number", and "Event-Timestamp". Between each value insert a colon (ASCII character
 788 value hex 3A) as a delimiter. If any one of the headers is not present, do not include that value but
 789 still include the delimiter (e.g. if "Content-Type" is not present include a ":" prior to encoding the
 790 "Event-Type"). All headers that are defined to be strings shall be handled as ASCII characters.

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

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

795 4) Compare the created signature to the "Event-Signature" header of the received notification and
 796 verify that they match.

797 **9.3 Parameters supported**

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

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

Friendly Name	Parameter Name	Location	Mandatory	Description
Accept	Accept	HTTP Header	Yes except for subscription cancellation (DELETE)	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.
---------------------	--------------	-------------	----	---

800 **9.4 Events API subscription and notification payload definitions**

801 **9.4.1 Subscription request**

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

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

811 **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

812
 813 Annex B provides a definition of the payload contained within the subscription request. The
 814 Properties that are contained in the payload are further clarified in Table 16.

815 **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

816

817 Figure 3 is an example of such a payload.

```

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

```

818 **Figure 3 – Subscription Request Example**

819 **9.4.2 Subscription response**

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

822 **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).

823

824 Figure 4 is an example of such a payload.

```

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

```

825 **Figure 4 – Subscription Response Example Payload**

826 **9.4.3 Notification request**

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

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

836 The Target Cloud shall send a notification with an event type of "subscription_cancelled" to the
837 "eventsUrl" provided by the Origin Cloud if there is a cancellation of the subscription. As there is
838 no defined payload for a "subscription_cancelled" event, a POST request that is sent for this event
839 type shall not include a "Content-Type" header. The cancellation may be through reception of a
840 DELETE from the Origin Cloud (see clauses 9.4.4, 9.6, and 9.7) or through internal logic on the
841 Target Cloud itself.

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

845 **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

846
 847 The format of the payload in a notification request depends on the event type for which the
 848 subscription was created. Table 19 defines the format of the payload provided in a notification per
 849 "eventType" (as received in the payload of the subscription request from the Origin Cloud) that may
 850 be sent by the Target Cloud. A Target Cloud shall populate the notification payload for the event
 851 type being signalled in the Event-Type HTTP header as defined in Table 19. The schema definitions
 852 for all payloads are provided in Annex B.

853 **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 UUIDs	Array containing Device UUIDs that have been registered since the previous notification was sent.
devices_unregistered	Empty array (i.e. [])	Array containing Device UUIDs for devices that have been de-registered since the previous notification was sent
devices_online	Array of all currently online Device UUIDs	Array containing Device UUIDs that have come

		online since the previous notification was sent.
devices_offline	Array of all currently offline Device UUIDs	Array containing Device UUIDs 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 UUID in the path	Array of Links of all Resources published by the Device UUID 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 UUID in the path since the previous notification was sent

854

855 **9.4.4 Notification response**

856 If the Target Cloud receives a non-success path response to a notification request it shall treat the
857 response as indicative of a request to cancel the subscription, and no further notifications for the
858 Subscription ID that was in the request shall be sent. See clauses 9.8.3, 9.9.3, and 9.10.3 for
859 further information.

860 **9.5 Subscribe and unsubscribe to devices level event types**

861 **9.5.1 Summary**

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

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

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

870 Table 20 provides a summary of the API.

871 **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	
--	--	--	-----------------------	--

872 **9.5.2 Request and Response payload**

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

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

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

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

879 **9.5.3 Responses**

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

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

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

888 **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

889

890 **9.6 Subscribe and unsubscribe to device level events**

891 **9.6.1 Summary**

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

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

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

899 Table 22 provides a summary of the API.

900

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	

901 **9.6.2 Request and Response payload**

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

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

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

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

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

910 **9.6.3 Responses**

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

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

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

919

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

920 **9.7 Subscribe and unsubscribe to resource level events**

921 **9.7.1 Summary**

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

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

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

929 Table 24 provides a summary of the API.

930 **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, Content-Type	201	See clause B.1 - /definitions/SubscribeResponse
			400, 401, 403, 404	
DELETE	/api/v1/devices/{deviceid}/{resourcehref}/subscriptions/{subscriptionId}	Correlation-ID	202	
			400, 401, 403, 404	

931 **9.7.2 Request and Response payload**

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

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

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

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

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

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

942 **9.7.3 Responses**

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

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

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

951 **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

952

953 **9.8 Notification of devices level events**

954 **9.8.1 Summary**

955 This request is sent from the Target Cloud to the Origin Cloud whenever there is an initial
 956 subscription to an event or an event for which a subscription exists occurs as defined in clause
 957 9.4.4.

958 Table 26 provides a summary of the API.

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

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

960 **9.8.2 Request and Response payload**

961 The "eventsUrl" in the URI shall be the value of the "eventsUrl" Property that was provided in the
 962 subscription request.

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

965 **9.8.3 Responses**

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

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

971 **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

972

973 **9.9 Notification of Device level events**

974 **9.9.1 Summary**

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

977 Table 28 provides a summary of the API.

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

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

979 **9.9.2 Request and Response payload**

980 The "eventsUrl" in the URI shall be the value of the "eventsUrl" Property that was provided in the
 981 subscription request.

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

984 **9.9.3 Responses**

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

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

990 **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

991

992 **9.10 Notification of Resource level events**

993 **9.10.1 Summary**

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

996 Table 30 provides a summary of the API.

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

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

998 **9.10.2 Request and Response payload**

999 The “eventsUrl” in the URI shall be the value of the "eventsUrl" Property that was provided in the
 1000 subscription request.

1001 The payload in the notification request depends on the Event-Type that is the subject of the
 1002 notification request; please see Table 19 for specifics and clause 9.4.3 for further information.

1003 **9.10.3 Responses**

1004 A 200 response shall be provided in a success case.

1005 A non-success path response that is indicative of the type of error shall be returned by an Origin
 1006 Cloud if an error scenario is detected. Table 31 lists possible non-success path responses and

1007 possible scenarios that may trigger their generation; an implementation may support additional
1008 responses as defined by IETF RFC 2818.

1009 **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

1010

1011
1012

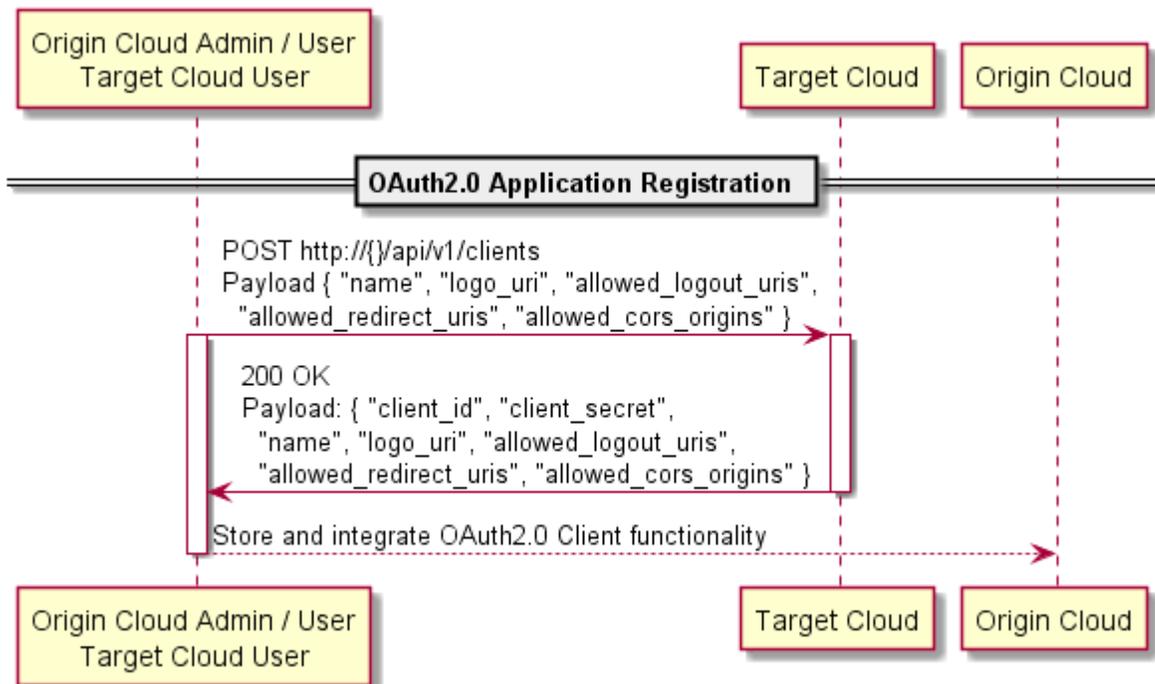
Annex A Representative Flows

1013 A.1 Introduction

1014 The flows illustrate use of the OCF Cloud API for Cloud Services using OCF Devices as the target
1015 servers where applicable and OCF Clouds as the two Clouds that are invoking/acting as API
1016 Endpoints. Note that this is for example use only and the API does not force this setup, which
1017 means non-OCF clouds with non-OCF devices may also use the API for interworking with other
1018 vendor's clouds.

1019 A.2 OAuth2.0 application registration

1020 Figure A.1 provides an example flow showing the registration of the OAuth 2.0 Origin Cloud Client.

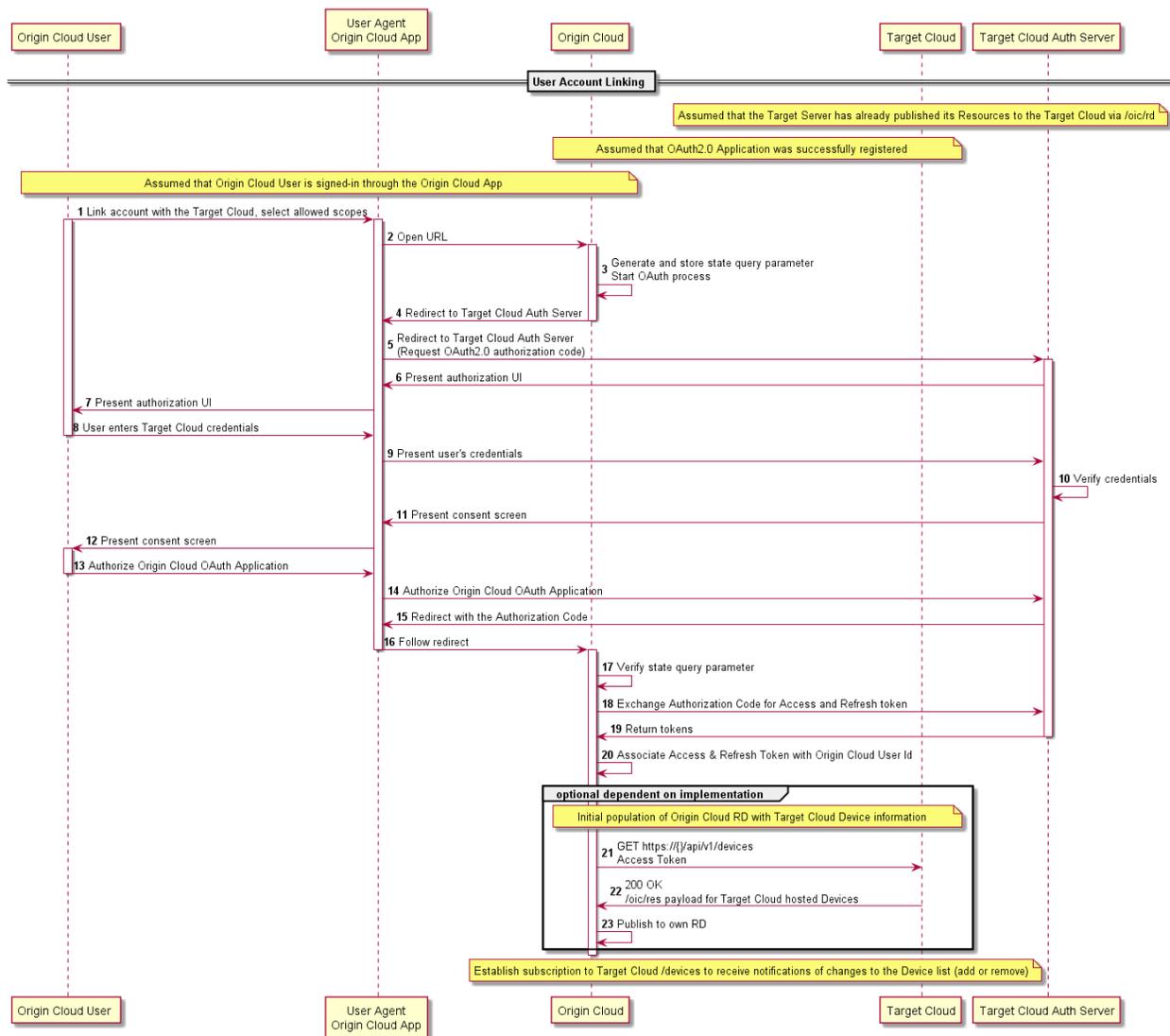


1021

1022 **Figure A.1 – Establish business relationship example flow**

1023 A.3 Account linking

1024 Figure A.2 provides an example flow of the account linking for a particular user.



1025

1026

Figure A.2 – Initial association example flow

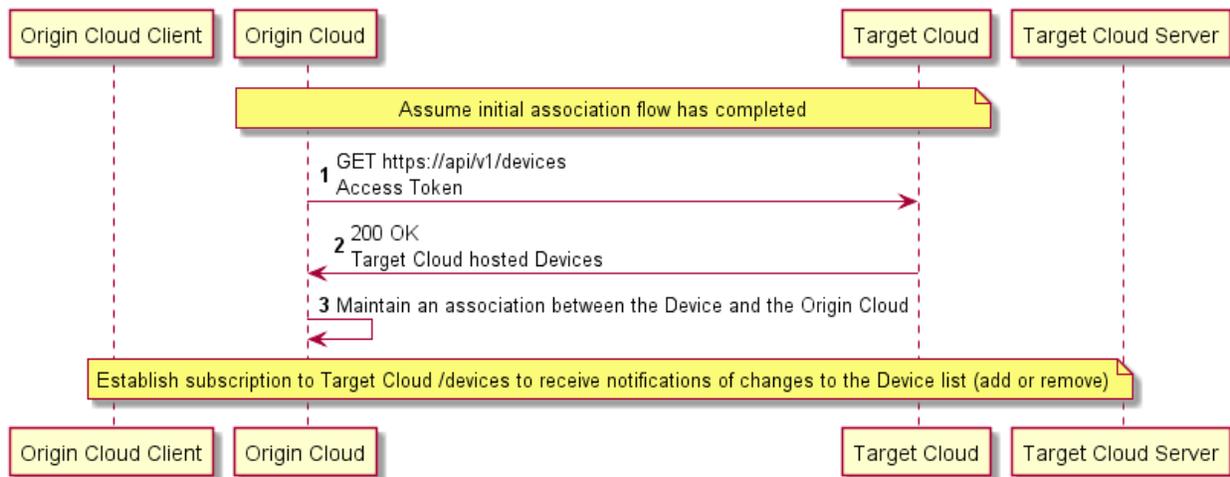
1027 **A.4 Retrieval of all Devices**

1028 **A.4.1 Summary**

1029 The Origin Cloud requests all Devices associated with a user (defined by the provided Bearer
 1030 Token). This may be invoked following account linking in order to retrieve the set of Devices for
 1031 the user.

1032 **A.4.2 Flow**

1033 Figure A.3 provides an example flow for the retrieval of all Devices.



1034

1035

Figure A.3 – Retrieve all Devices example flow

1036 **A.4.3 Flow description**

1037 Table A.1 explains each element in Figure A.3

1038

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.

1039

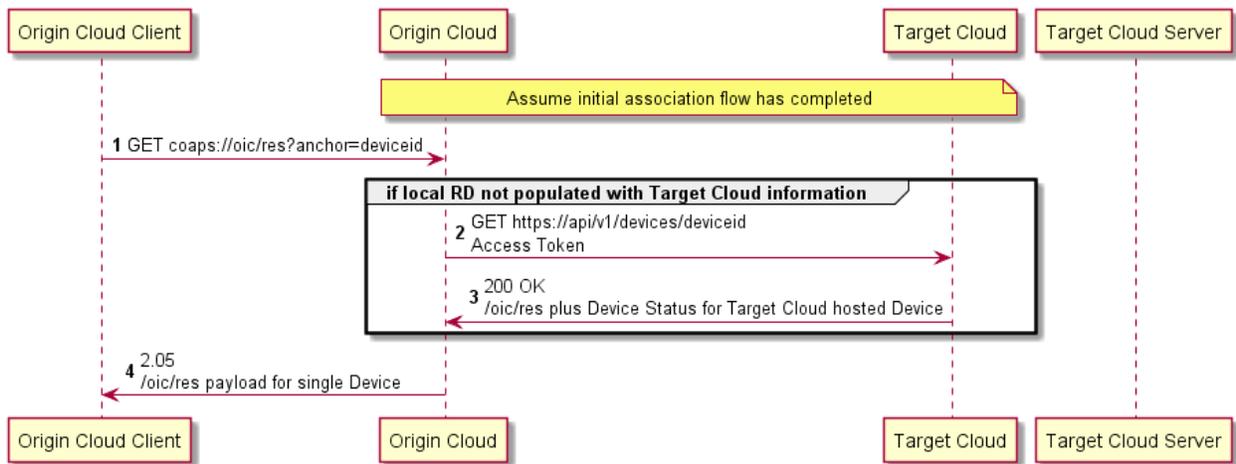
1040 **A.5 Retrieval of a single Device**

1041 **A.5.1 Summary**

1042 The Origin Cloud requests information for a single, specific Device associated with a user (defined
 1043 by the provided Bearer Token). This may be invoked by the Origin Cloud receiving a retrieve
 1044 request from a connected Client.

1045 **A.5.2 Flow**

1046 Figure A.4 provides an example flow for the retrieval of a single Device.



1047

1048

Figure A.4 – Retrieve single Device example flow

A.5.3 Flow description

1049 Table A.2 explains each element in Figure A.4.

1050

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 UUID).
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.

1051 **A.6 Retrieval of a single Resource**

1052 **A.6.1 Summary**

1053 The Origin Cloud requests information for a single, specific Resource exposed by a Device
1054 associated with a user (defined by the provided Bearer Token). This may be invoked by the Origin
1055 Cloud receiving a retrieve request from a connected Client.
1056

1057 **A.6.2 Flows**

1058 **A.6.2.1 Success path**

1059 Figure A.5 provides an example flow for the retrieval of a single Resource.

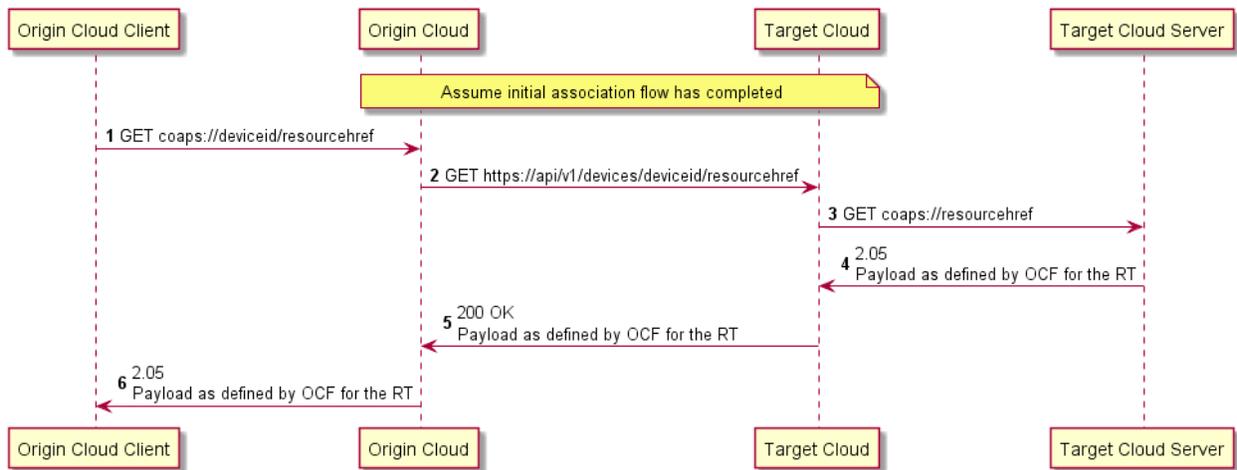


Figure A.5 – Retrieve Resource (success) example flow

A.6.2.2 Success path flow description

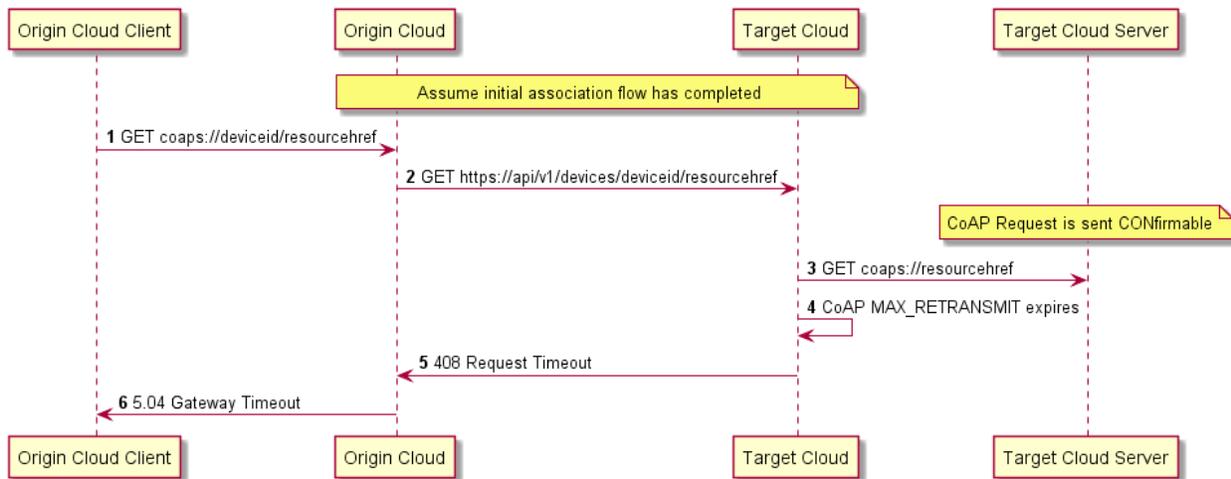
Table A.3 explains each element in Figure A.5.

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

A.6.2.3 Device is temporarily unavailable

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



1068
1069 **Figure A.6 – Retrieve Resource (timeout) example flow**

1070 **A.7 Update of a single Resource**

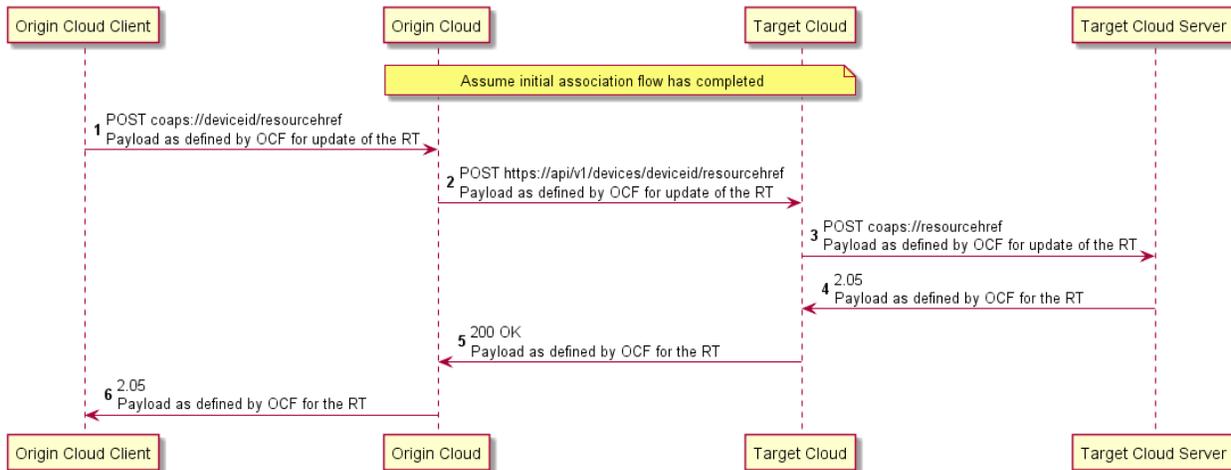
1071 **A.7.1 Summary**

1072 The Origin Cloud updates information for a single, specific Device associated with a user (defined
1073 by the provided Bearer Token). This may be invoked by the Origin Cloud receiving an update
1074 request from a connected Client.

1075 **A.7.2 Flows**

1076 **A.7.2.1 Success path**

1077 Figure A.7 provides an example flow for the updating of a single Resource.



1078
1079 **Figure A.7 – Update Resource (success) example flow**

1080 **A.7.2.2 Success path flow description**

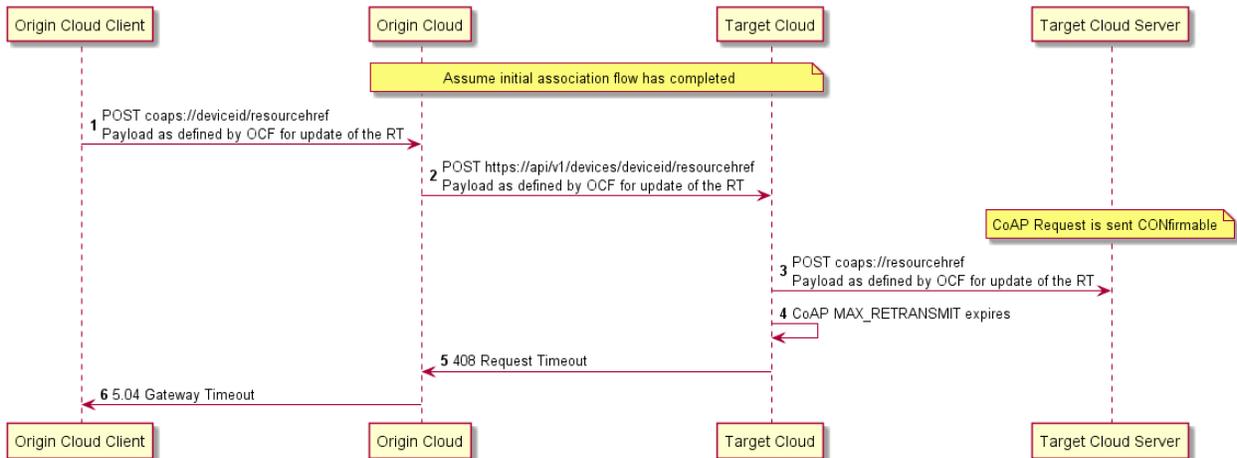
1081 Table A.4 explains each element in Figure A.7.

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

1083 **A.7.2.3 Device is temporarily unavailable**

1084 Figure A.8 illustrates the case where the Device is temporarily unavailable.



1085

1086 **Figure A.8 – Update Resource (timeout) example flow**

1087 **A.8 Establishment of new subscription request**

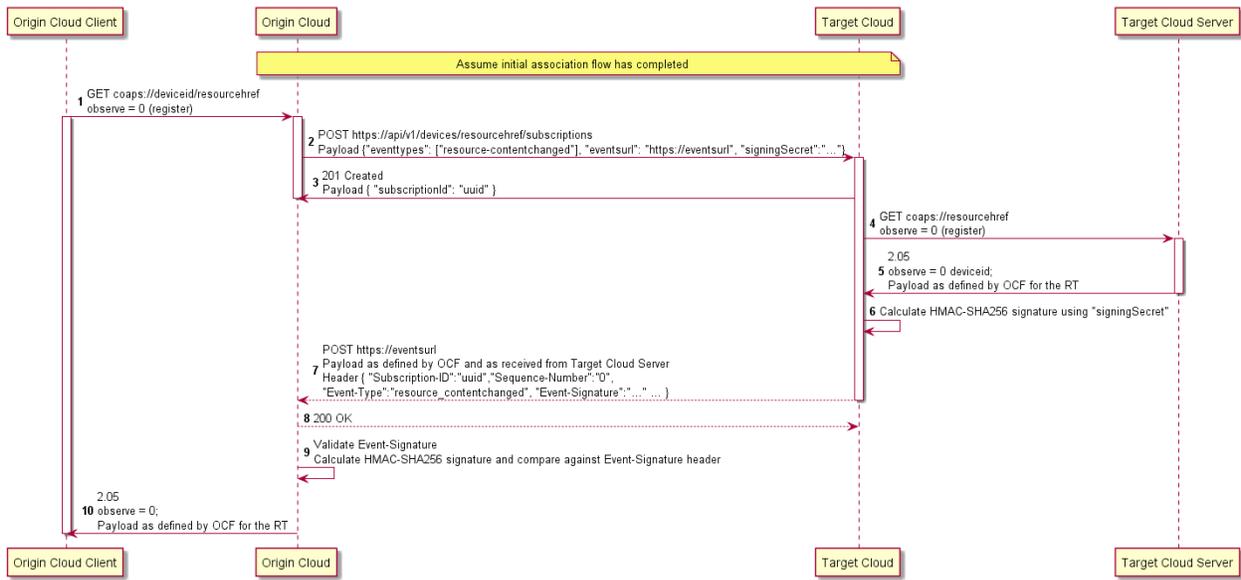
1088 **A.8.1 Summary**

1089 The Origin Cloud requests the establishment of an observe relationship with a single, specific
 1090 Resource on a Device associated with a user (defined by the provided Bearer Token). This may be
 1091 invoked by Origin Cloud receiving a retrieve request containing an observe option from a connected
 1092 Client.

1093 **A.8.2 Flows**

1094 Figure A.9 provides an example flow for the establishment of a subscription to the
 1095 "resource_contentchanged" event for a specific Resource.

1096



1097

1098

Figure A.9 – Subscription establishment example flow

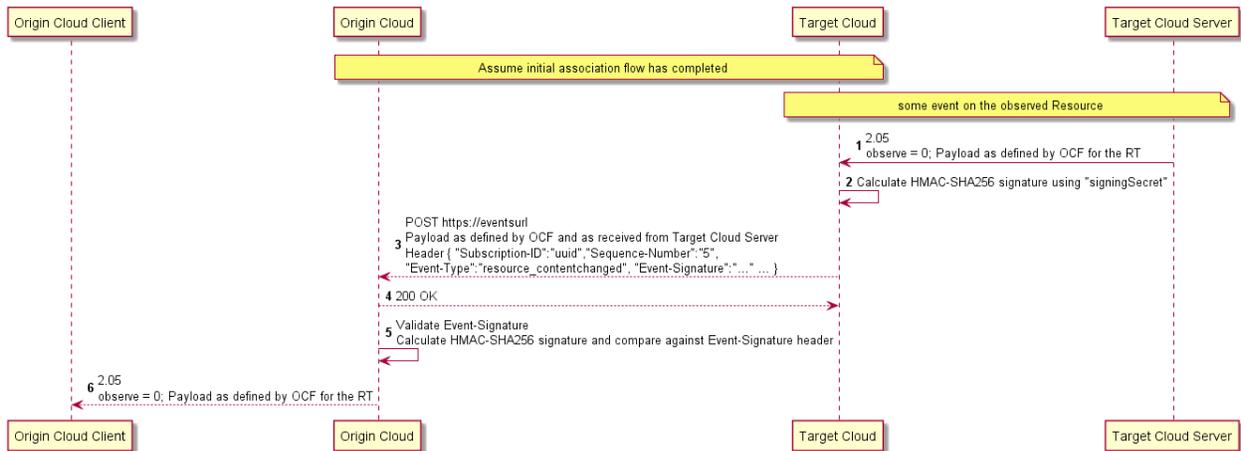
1099 **A.9 Event generated for a subscription**

1100 **A.9.1 Summary**

1101 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.

1102 **A.9.2 Flows**

1103 Figure A.10 provides an example flow for the handling of a generated "resource_contentchanged" event.



1106

1107

Figure A.10 – "resource_contentchanged" event example flow

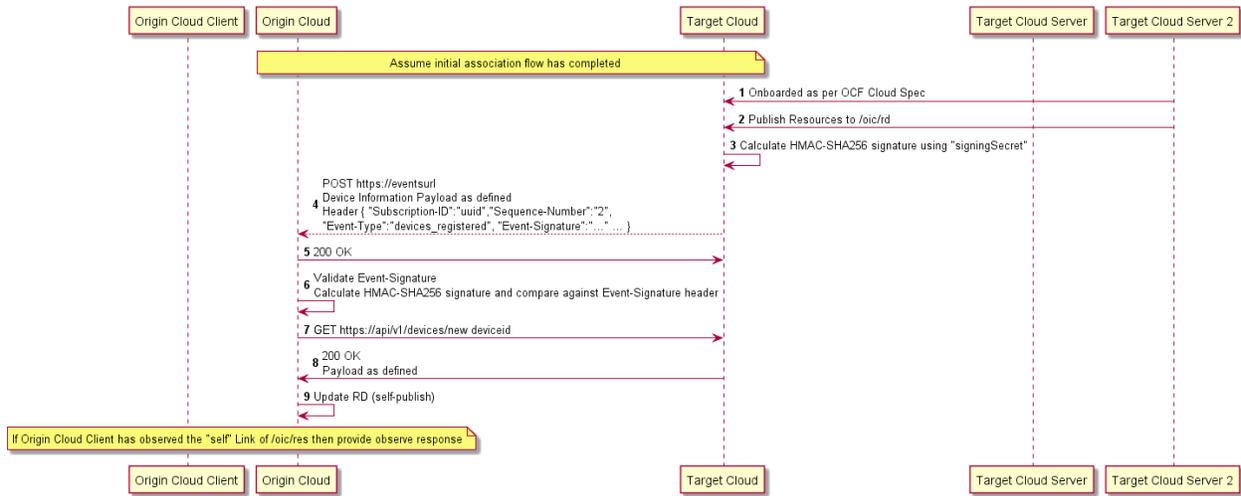
1108 **A.10 Addition of new registration**

1109 **A.10.1 Summary**

1110 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.

1113 **A.10.2 Flows**

1114 Figure A.11 provides an example flow for the generation of a notification (event) when a new Device
 1115 is registered.



1116

1117 **Figure A.11 – Addition of new registered Device example flow**

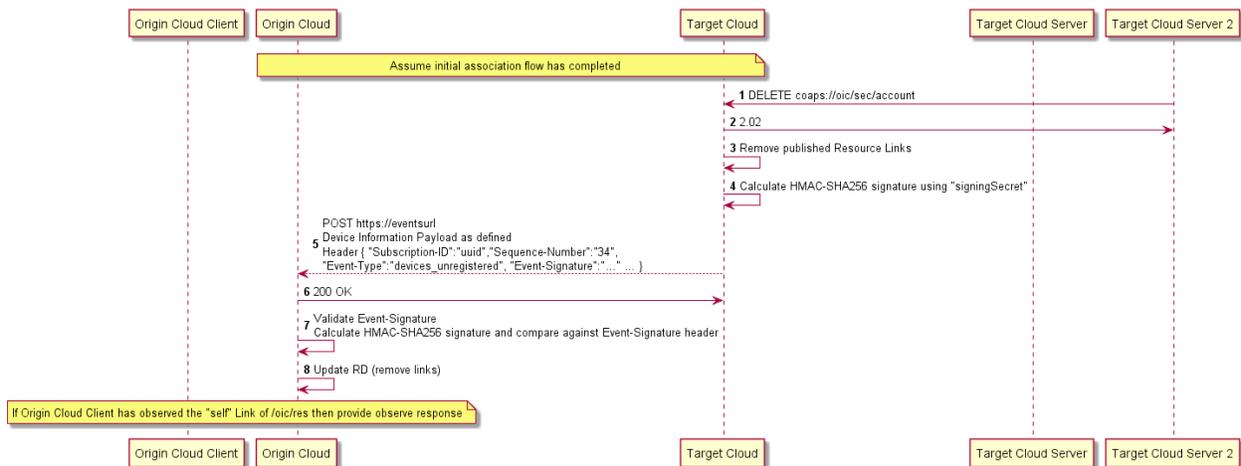
1118 **A.11 Removal of existing device registration**

1119 **A.11.1 Summary**

1120 The Origin Cloud has a priori established a subscription/event relationship with the set of Devices
 1121 associated with a user exposed by Target Cloud. The user then removes a Device from Target
 1122 Cloud.

1123 **A.11.2 Flows**

1124 Figure A.12 provides an example flow for the generation of a notification (event) when a Device is
 1125 removed.



1126

1127 **Figure A.12 – Removal of existing registration example flow**

Annex B Open API Definition

1128
1129
1130

1131 **B.1 OCF Cloud API for Cloud Services**

1132 **B.1.1 Supported APIs**

1133 **B.1.1.1 /api/v1/devices?content=base**

1134 Get meta-information, including Resource Links, for all Devices which are signed up to the OCF
1135 Cloud - either "online" or "offline". Devices which are "online" are signed in to the system and are
1136 accessible. Offline devices are signed up to the system, but currently disconnected.

1137 **B.1.1.2 /api/v1/devices?content=all**

1138 Get meta-information, including Resource Representations, for all Devices which are signed up to
1139 the OCF Cloud - either "online" or "offline". Devices which are "online" are signed in to the
1140 system and are accessible. Offline devices are signed up to the system, but currently
1141 disconnected.

1142 **B.1.1.3 /api/v1/devices/subscriptions**

1143 Subscribe to devices events by providing "eventTypes" you're interested in and an "eventsUrl"
1144 endpoint where notifications will be sent to as defined. A successful response contains a
1145 "subscriptionId" which identifies the registered subscription and is part of each notification. First
1146 notification for each registered event type is received immediately after subscription and contains
1147 the actual state of the resource, followed by new notifications in case of any change.
1148

1149 Supported events:

- 1150 - "devices_registered"
- 1151 - "devices_unregistered"
- 1152 - "devices_online"
- 1153 - "devices_offline"

1154 **B.1.1.4 /api/v1/devices/subscriptions/{subscriptionId}**

1155 Cancel the subscription identified by the provided "subscriptionId" that was returned in the
1156 response to the subscription request.

1157 **B.1.1.5 /api/v1/devices/{deviceId}?content=base**

1158 Get the meta-information for the Device given by the provided "deviceId" including Resource
1159 Links.

1160 **B.1.1.6 /api/v1/devices/{deviceId}?content=all**

1161 Get the meta-information for the Device given by the provided "deviceId" including Resource
1162 Representations.

1163 **B.1.1.7 /api/v1/devices/{deviceId}/subscriptions**

1164 Subscribe to Device level events by providing "eventTypes" you're interested in and an
1165 "eventsUrl" API Endpoint where notifications will be sent to as defined. A successful response
1166 contains a "subscriptionId" which identifies the registered subscription and is part of each
1167 notification. First notification for each registered event type is received immediately after
1168 subscription and contains the actual state of the resource, followed by new notifications in case
1169 of any change.

1170 Supported events:
1171

1172 - "resources_published"
1173 - "resources_unpublished"

1174 **B.1.1.8 /api/v1/devices/{deviceId}/subscriptions/{subscriptionId}**

1175 Cancel the subscription identified by the provided "subscriptionId" that was returned in the
1176 response to the subscription request.

1177 **B.1.1.9 /api/v1/devices/{deviceId}/{resourceLinkHref}**

1178 Get or update the Resource Representation of the Resource found at "resourceLinkHref" on the
1179 Device with the given "deviceId"

1180 **B.1.1.10 /api/v1/devices/{deviceId}/{resourceLinkHref}/subscriptions**

1181 Subscribe to Resource level events by providing "eventTypes" you're interested in and
1182 "eventsUrl" API Endpoint where notifications will be sent to as defined. A successful response
1183 contains a "subscriptionId" which identifies the registered subscription and is part of each event.
1184 First notification for each registered event type is received immediately after subscription and
1185 contains the actual state of the resource, followed by new notifications in case of any change.

1186 Supported events:

1187 - "resource_contentchanged"

1189 **B.1.1.11 /api/v1/devices/{deviceId}/{resourceLinkHref}/subscriptions/{subscriptionId}**

1190 Cancel the subscription identified by the provided "subscriptionId" that was returned in the
1191 response to the subscription request.

1192 **B.1.1.12 /{eventsUrl}**

1193 Events endpoint provided during subscription where notifications for the events specified in the
1194 subscription will be sent to as defined per event type. Confirmation of each notification sent to
1195 the "eventsUrl" endpoint is required with a "2xx" success code.

1196

1197 Notifications you may receive based on the event type you're subscribed to are:

1198 - "subscription_cancelled": "SubscriptionCancelledEvent"

1199 - "devices_registered": "DevicesRegisteredEvent"

1200 - "devices_unregistered": "DevicesUnregisteredEvent"

1201 - "resources_published": "ResourcesPublishedEvent"

1202 - "resources_unpublished": "ResourcesUnpublishedEvent"

1203 - "devices_online": "DevicesOnlineEvent"

1204 - "devices_offline": "DevicesOfflineEvent"

1205 - "resource_contentchanged": "ResourceContentChangedEvent"

1206 **B.1.2 OpenAPI 2.0 definition**

```
1207 {
1208   "swagger": "2.0",
1209   "info": {
1210     "title": "OCF Cloud API for Cloud Services",
1211     "version": "0.0.3-20190828",
1212     "license": {
1213       "name": "Copyright 2019 Open Connectivity Foundation, Inc. All rights reserved.",
1214       "x-description": "Redistribution and use in source and binary forms, with or without
1215 modification, are permitted provided that the following conditions are met:\n      1.
1216 Redistributions of source code must retain the above copyright notice, this list of conditions and
1217 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
1218 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
1219 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
1220 Connectivity Foundation, INC. \\"AS IS\\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
1221 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
1222 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n\n      IN NO EVENT SHALL THE Open Connectivity
1223 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
```

```

1224 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
1225 SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n          HOWEVER CAUSED AND ON
1226 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
1227 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
1228 SUCH DAMAGE.\n"
1229 }
1230 },
1231 "host": "api.example.com",
1232 "schemes": [
1233   "https"
1234 ],
1235 "tags": [
1236   {
1237     "name": "Devices",
1238     "description": "Basic information about devices"
1239   },
1240   {
1241     "name": "Resources",
1242     "description": "Read or change the configuration of the device"
1243   },
1244   {
1245     "name": "Events",
1246     "description": "Be notified about changes occurring on the device"
1247   }
1248 ],
1249 "paths": {
1250   "/api/v1/devices?content=base": {
1251     "parameters": [
1252       {
1253         "$ref": "#/parameters/CorrelationId"
1254       },
1255       {
1256         "$ref": "#/parameters/Accept"
1257       },
1258       {
1259         "$ref": "#/parameters/BatchFormat"
1260       }
1261     ],
1262     "get": {
1263       "tags": [
1264         "Devices"
1265       ],
1266       "summary": "Get all devices with resource links",
1267       "description": "Get meta-information, including Resource Links, for all Devices which are
1268 signed up to the OCF Cloud - either \"online\" or \"offline\". Devices which are \"online\" are
1269 signed in to the system and are accessible. Offline devices are signed up to the system, but
1270 currently disconnected.",
1271       "produces": [
1272         "application/json"
1273       ],
1274       "responses": {
1275         "200": {
1276           "description": "An array of devices",
1277           "schema": {
1278             "type": "array",
1279             "items": {
1280               "$ref": "#/definitions/Device"
1281             }
1282           }
1283         },
1284         "400": {
1285           "$ref": "#/responses/BadRequest"
1286         },
1287         "401": {
1288           "$ref": "#/responses/Unauthorized"
1289         },
1290         "403": {
1291           "$ref": "#/responses/Forbidden"
1292         },
1293         "406": {
1294           "$ref": "#/responses/NotAcceptable"

```

```

1295         },
1296         "503": {
1297             "$ref": "#/responses/ServiceUnavailable"
1298         },
1299         "504": {
1300             "$ref": "#/responses/GatewayTimeout"
1301         }
1302     },
1303     "security": [
1304         {
1305             "oauth2": [
1306                 "r:*"
1307             ]
1308         }
1309     ]
1310 },
1311 },
1312 "/api/v1/devices?content=all": {
1313     "parameters": [
1314         {
1315             "$ref": "#/parameters/CorrelationId"
1316         },
1317         {
1318             "$ref": "#/parameters/Accept"
1319         },
1320         {
1321             "$ref": "#/parameters/BatchFormat"
1322         }
1323     ],
1324     "get": {
1325         "tags": [
1326             "Devices"
1327         ],
1328         "summary": "Get all devices with resource representations",
1329         "description": "Get meta-information, including Resource Representations, for all Devices
1330 which are signed up to the OCF Cloud - either \"online\" or \"offline\". Devices which are
1331 \"online\" are signed in to the system and are accessible. Offline devices are signed up to the
1332 system, but currently disconnected.",
1333         "produces": [
1334             "application/json"
1335         ],
1336         "responses": {
1337             "200": {
1338                 "description": "An array of devices",
1339                 "schema": {
1340                     "type": "array",
1341                     "items": {
1342                         "$ref": "#/definitions/DeviceContentAll"
1343                     }
1344                 }
1345             },
1346             "400": {
1347                 "$ref": "#/responses/BadRequest"
1348             },
1349             "401": {
1350                 "$ref": "#/responses/Unauthorized"
1351             },
1352             "403": {
1353                 "$ref": "#/responses/Forbidden"
1354             },
1355             "406": {
1356                 "$ref": "#/responses/NotAcceptable"
1357             },
1358             "503": {
1359                 "$ref": "#/responses/ServiceUnavailable"
1360             },
1361             "504": {
1362                 "$ref": "#/responses/GatewayTimeout"
1363             }
1364         },
1365         "security": [

```

```

1366         {
1367             "oauth2": [
1368                 "r:*"
1369             ]
1370         }
1371     ]
1372 }
1373 },
1374 "/api/v1/devices/subscriptions": {
1375     "parameters": [
1376         {
1377             "$ref": "#/parameters/CorrelationId"
1378         },
1379         {
1380             "$ref": "#/parameters/Accept"
1381         }
1382     ],
1383     "post": {
1384         "tags": [
1385             "Events"
1386         ],
1387         "summary": "Subscribe to events against the set of devices",
1388         "description": "Subscribe to devices events by providing \"eventTypes\" you're interested in
1389 and an \"eventsUrl\" endpoint where notifications will be sent to as defined. A successful response
1390 contains a \"subscriptionId\" which identifies the registered subscription and is part of each
1391 notification. First notification for each registered event type is received immediately after
1392 subscription and contains the actual state of the resource, followed by new notifications in case of
1393 any change.\n\nSupported events:\n- \"devices_registered\"\n- \"devices_unregistered\"\n-
1394 \"devices_online\"\n- \"devices_offline\"",
1395         "parameters": [
1396             {
1397                 "$ref": "#/parameters/ContentType"
1398             },
1399             {
1400                 "$ref": "#/parameters/SubscribeRequestDevices"
1401             }
1402         ],
1403         "consumes": [
1404             "application/json"
1405         ],
1406         "produces": [
1407             "application/json"
1408         ],
1409         "responses": {
1410             "201": {
1411                 "$ref": "#/definitions/SubscribeResponse"
1412             },
1413             "400": {
1414                 "$ref": "#/responses/BadRequest"
1415             },
1416             "401": {
1417                 "$ref": "#/responses/Unauthorized"
1418             },
1419             "403": {
1420                 "$ref": "#/responses/Forbidden"
1421             }
1422         },
1423         "security": [
1424             {
1425                 "oauth2": [
1426                     "r:*"
1427                 ]
1428             }
1429         ]
1430     }
1431 },
1432 "/api/v1/devices/subscriptions/{subscriptionId}": {
1433     "parameters": [
1434         {
1435             "$ref": "#/parameters/CorrelationId"
1436         },

```

```

1437     {
1438     "href": "#/parameters/SubscriptionIdPath"
1439     }
1440 ],
1441 "delete": {
1442     "tags": [
1443     "Events"
1444     ],
1445     "summary": "Unsubscribe from events against the set of devices",
1446     "description": "Cancel the subscription identified by the provided \"subscriptionId\" that
1447 was returned in the response to the subscription request.",
1448     "responses": {
1449     "202": {
1450     "description": "Subscription was marked for cancellation"
1451     },
1452     "400": {
1453     "$ref": "#/responses/BadRequest"
1454     },
1455     "401": {
1456     "$ref": "#/responses/Unauthorized"
1457     },
1458     "403": {
1459     "$ref": "#/responses/Forbidden"
1460     },
1461     "404": {
1462     "$ref": "#/responses/NotFound"
1463     },
1464     "406": {
1465     "$ref": "#/responses/NotAcceptable"
1466     }
1467     },
1468     "security": [
1469     {
1470     "oauth2": [
1471     "r:*"
1472     ]
1473     }
1474     ]
1475     }
1476 },
1477 "/api/v1/devices/{deviceId}?content=base": {
1478     "parameters": [
1479     {
1480     "$ref": "#/parameters/CorrelationId"
1481     },
1482     {
1483     "$ref": "#/parameters/Accept"
1484     },
1485     {
1486     "$ref": "#/parameters/DeviceId"
1487     },
1488     {
1489     "$ref": "#/parameters/BatchFormat"
1490     }
1491     ],
1492     "get": {
1493     "tags": [
1494     "Devices"
1495     ],
1496     "summary": "Get the device with resource links by ID",
1497     "description": "Get the meta-information for the Device given by the provided \"deviceId\"
1498 including Resource Links.",
1499     "consumes": [
1500     "application/json"
1501     ],
1502     "produces": [
1503     "application/json"
1504     ],
1505     "responses": {
1506     "200": {
1507     "description": "Device requested with content=base query parameter",

```

```

1508         "schema": {
1509             "$ref": "#/definitions/Device"
1510         },
1511     },
1512     "400": {
1513         "$ref": "#/responses/BadRequest"
1514     },
1515     "401": {
1516         "$ref": "#/responses/Unauthorized"
1517     },
1518     "403": {
1519         "$ref": "#/responses/Forbidden"
1520     },
1521     "404": {
1522         "$ref": "#/responses/NotFound"
1523     },
1524     "406": {
1525         "$ref": "#/responses/NotAcceptable"
1526     },
1527     "503": {
1528         "$ref": "#/responses/ServiceUnavailable"
1529     },
1530     "504": {
1531         "$ref": "#/responses/GatewayTimeout"
1532     }
1533 },
1534 "security": [
1535     {
1536         "oauth2": [
1537             "r:*"
1538         ]
1539     }
1540 ]
1541 }
1542 },
1543 "/api/v1/devices/{deviceId}?content=all": {
1544     "parameters": [
1545         {
1546             "$ref": "#/parameters/CorrelationId"
1547         },
1548         {
1549             "$ref": "#/parameters/Accept"
1550         },
1551         {
1552             "$ref": "#/parameters/DeviceId"
1553         },
1554         {
1555             "$ref": "#/parameters/BatchFormat"
1556         }
1557     ],
1558     "get": {
1559         "tags": [
1560             "Devices"
1561         ],
1562         "summary": "Get the device with resource representations by ID",
1563         "description": "Get the meta-information for the Device given by the provided \"deviceId\" including Resource Representations.",
1564         "consumes": [
1565             "application/json"
1566         ],
1567         "produces": [
1568             "application/json"
1569         ],
1570     },
1571     "responses": {
1572         "200": {
1573             "description": "Device requested with content=all query parameter",
1574             "schema": {
1575                 "$ref": "#/definitions/DeviceContentAll"
1576             }
1577         },
1578         "400": {

```

```

1579         "$ref": "#/responses/BadRequest"
1580     },
1581     "401": {
1582         "$ref": "#/responses/Unauthorized"
1583     },
1584     "403": {
1585         "$ref": "#/responses/Forbidden"
1586     },
1587     "404": {
1588         "$ref": "#/responses/NotFound"
1589     },
1590     "406": {
1591         "$ref": "#/responses/NotAcceptable"
1592     },
1593     "503": {
1594         "$ref": "#/responses/ServiceUnavailable"
1595     },
1596     "504": {
1597         "$ref": "#/responses/GatewayTimeout"
1598     }
1599 },
1600 "security": [
1601     {
1602         "oauth2": [
1603             "r:*"
1604         ]
1605     }
1606 ]
1607 },
1608 },
1609 "/api/v1/devices/{deviceId}/subscriptions": {
1610     "parameters": [
1611         {
1612             "$ref": "#/parameters/CorrelationId"
1613         },
1614         {
1615             "$ref": "#/parameters/DeviceId"
1616         },
1617         {
1618             "$ref": "#/parameters/Accept"
1619         }
1620     ],
1621     "post": {
1622         "tags": [
1623             "Events"
1624         ],
1625         "summary": "Subscribe to events against a specific device",
1626         "description": "Subscribe to Device level events by providing \"eventTypes\" you're
1627 interested in and an \"eventsUrl\" API Endpoint where notifications will be sent to as defined. A
1628 successful response contains a \"subscriptionId\" which identifies the registered subscription and
1629 is part of each notification. First notification for each registered event type is received
1630 immediately after subscription and contains the actual state of the resource, followed by new
1631 notifications in case of any change.\n\nSupported events:\n- \"resources_published\"\n-
1632 \"resources_unpublished\"",
1633         "parameters": [
1634             {
1635                 "$ref": "#/parameters/ContentType"
1636             },
1637             {
1638                 "$ref": "#/parameters/SubscribeRequestDevice"
1639             }
1640         ],
1641         "consumes": [
1642             "application/json"
1643         ],
1644         "produces": [
1645             "application/json"
1646         ],
1647         "responses": {
1648             "201": {
1649                 "$ref": "#/definitions/SubscribeResponse"

```

```

1650     },
1651     "400": {
1652       "$ref": "#/responses/BadRequest"
1653     },
1654     "401": {
1655       "$ref": "#/responses/Unauthorized"
1656     },
1657     "403": {
1658       "$ref": "#/responses/Forbidden"
1659     },
1660     "404": {
1661       "$ref": "#/responses/NotFound"
1662     },
1663     "406": {
1664       "$ref": "#/responses/NotAcceptable"
1665     }
1666   },
1667   "security": [
1668     {
1669       "oauth2": [
1670         "r:*"
1671       ]
1672     }
1673   ]
1674 },
1675 },
1676 "/api/v1/devices/{deviceId}/subscriptions/{subscriptionId}": {
1677   "parameters": [
1678     {
1679       "$ref": "#/parameters/CorrelationId"
1680     },
1681     {
1682       "$ref": "#/parameters/DeviceId"
1683     },
1684     {
1685       "$ref": "#/parameters/SubscriptionIdPath"
1686     }
1687   ],
1688   "delete": {
1689     "tags": [
1690       "Events"
1691     ],
1692     "summary": "Unsubscribe from events against a specific device",
1693     "description": "Cancel the subscription identified by the provided \"subscriptionId\" that
1694 was returned in the response to the subscription request.",
1695     "responses": {
1696       "202": {
1697         "description": "Subscription was marked for cancellation"
1698       },
1699       "400": {
1700         "$ref": "#/responses/BadRequest"
1701       },
1702       "401": {
1703         "$ref": "#/responses/Unauthorized"
1704       },
1705       "403": {
1706         "$ref": "#/responses/Forbidden"
1707       },
1708       "404": {
1709         "$ref": "#/responses/NotFound"
1710       }
1711     },
1712     "security": [
1713       {
1714         "oauth2": [
1715           "r:*"
1716         ]
1717       }
1718     ]
1719   }
1720 },

```

```

1721     "/api/v1/devices/{deviceId}/{resourceLinkHref}": {
1722         "parameters": [
1723             {
1724                 "$ref": "#/parameters/CorrelationId"
1725             },
1726             {
1727                 "$ref": "#/parameters/DeviceId"
1728             },
1729             {
1730                 "$ref": "#/parameters/ResourceLinkHref"
1731             },
1732             {
1733                 "$ref": "#/parameters/Accept"
1734             }
1735         ],
1736         "get": {
1737             "tags": [
1738                 "Resources"
1739             ],
1740             "summary": "Retrieve resource values",
1741             "description": "Get or update the Resource Representation of the Resource found at
1742 \"resourceLinkHref\" on the Device with the given \"deviceId\",
1743             "consumes": [
1744                 "application/json",
1745                 "application/vnd.ocf+cbor"
1746             ],
1747             "produces": [
1748                 "application/json",
1749                 "application/vnd.ocf+cbor"
1750             ],
1751             "responses": {
1752                 "200": {
1753                     "$ref": "#/definitions/ResourceRetrieveResponse"
1754                 },
1755                 "400": {
1756                     "$ref": "#/responses/BadRequest"
1757                 },
1758                 "401": {
1759                     "$ref": "#/responses/Unauthorized"
1760                 },
1761                 "403": {
1762                     "$ref": "#/responses/Forbidden"
1763                 },
1764                 "404": {
1765                     "$ref": "#/responses/NotFound"
1766                 },
1767                 "406": {
1768                     "$ref": "#/responses/NotAcceptable"
1769                 },
1770                 "503": {
1771                     "$ref": "#/responses/ServiceUnavailable"
1772                 },
1773                 "504": {
1774                     "$ref": "#/responses/GatewayTimeout"
1775                 }
1776             },
1777             "security": [
1778                 {
1779                     "oauth2": [
1780                         "r:*"
1781                     ]
1782                 }
1783             ],
1784         },
1785         "post": {
1786             "tags": [
1787                 "Resources"
1788             ],
1789             "summary": "Update resource values",
1790             "parameters": [
1791                 {

```

```

1792         "$ref": "#/parameters/ResourceUpdateRequest"
1793     },
1794     {
1795         "$ref": "#/parameters/ContentType"
1796     }
1797 ],
1798 "consumes": [
1799     "application/json",
1800     "application/vnd.ocf+cbor"
1801 ],
1802 "produces": [
1803     "application/json",
1804     "application/vnd.ocf+cbor"
1805 ],
1806 "responses": {
1807     "200": {
1808         "$ref": "#/definitions/ResourceRetrieveResponse"
1809     },
1810     "400": {
1811         "$ref": "#/responses/BadRequest"
1812     },
1813     "401": {
1814         "$ref": "#/responses/Unauthorized"
1815     },
1816     "403": {
1817         "$ref": "#/responses/Forbidden"
1818     },
1819     "404": {
1820         "$ref": "#/responses/NotFound"
1821     },
1822     "415": {
1823         "$ref": "#/responses/UnsupportedMediaType"
1824     },
1825     "503": {
1826         "$ref": "#/responses/ServiceUnavailable"
1827     },
1828     "504": {
1829         "$ref": "#/responses/GatewayTimeout"
1830     }
1831 },
1832 "security": [
1833     {
1834         "oauth2": [
1835             "r:*",
1836             "w:*"
1837         ]
1838     }
1839 ]
1840 },
1841 },
1842 "/api/v1/devices/{deviceId}/{resourceLinkHref}/subscriptions": {
1843     "parameters": [
1844         {
1845             "$ref": "#/parameters/CorrelationId"
1846         },
1847         {
1848             "$ref": "#/parameters/DeviceId"
1849         },
1850         {
1851             "$ref": "#/parameters/ResourceLinkHref"
1852         },
1853         {
1854             "$ref": "#/parameters/Accept"
1855         }
1856     ],
1857     "post": {
1858         "tags": [
1859             "Events"
1860         ],
1861         "summary": "Subscribe to events against a specific resource",
1862         "description": "Subscribe to Resource level events by providing \"eventTypes\" you're

```

```

1863 interested in and \eventsUrl\ API Endpoint where notifications will be sent to as defined. A
1864 successful response contains a \subscriptionId\ which identifies the registered subscription and
1865 is part of each event. First notification for each registered event type is received immediately
1866 after subscription and contains the actual state of the resource, followed by new notifications in
1867 case of any change.\n \nSupported events:\n- \resource_contentchanged\",
1868     "parameters": [
1869         {
1870             "$ref": "#/parameters/ContentType"
1871         },
1872         {
1873             "$ref": "#/parameters/SubscribeRequestResources"
1874         }
1875     ],
1876     "consumes": [
1877         "application/json"
1878     ],
1879     "produces": [
1880         "application/json"
1881     ],
1882     "responses": {
1883         "201": {
1884             "$ref": "#/definitions/SubscribeResponse"
1885         },
1886         "400": {
1887             "$ref": "#/responses/BadRequest"
1888         },
1889         "401": {
1890             "$ref": "#/responses/Unauthorized"
1891         },
1892         "403": {
1893             "$ref": "#/responses/Forbidden"
1894         },
1895         "404": {
1896             "$ref": "#/responses/NotFound"
1897         },
1898         "406": {
1899             "$ref": "#/responses/NotAcceptable"
1900         }
1901     },
1902     "security": [
1903         {
1904             "oauth2": [
1905                 "r:*"
1906             ]
1907         }
1908     ]
1909 },
1910 "/api/v1/devices/{deviceId}/{resourceLinkHref}/subscriptions/{subscriptionId}": {
1911     "parameters": [
1912         {
1913             "$ref": "#/parameters/CorrelationId"
1914         },
1915         {
1916             "$ref": "#/parameters/DeviceId"
1917         },
1918         {
1919             "$ref": "#/parameters/ResourceLinkHref"
1920         },
1921         {
1922             "$ref": "#/parameters/SubscriptionIdPath"
1923         }
1924     ],
1925     "delete": {
1926         "tags": [
1927             "Events"
1928         ],
1929         "summary": "Unsubscribe from events against a specific resource",
1930         "description": "Cancel the subscription identified by the provided \subscriptionId\ that
1931 was returned in the response to the subscription request.",
1932         "responses": {
1933

```

```

1934         "202": {
1935             "description": "Subscription was marked for cancellation"
1936         },
1937         "400": {
1938             "$ref": "#/responses/BadRequest"
1939         },
1940         "401": {
1941             "$ref": "#/responses/Unauthorized"
1942         },
1943         "403": {
1944             "$ref": "#/responses/Forbidden"
1945         },
1946         "404": {
1947             "$ref": "#/responses/NotFound"
1948         }
1949     },
1950     "security": [
1951         {
1952             "oauth2": [
1953                 "r:*"
1954             ]
1955         }
1956     ]
1957 },
1958 },
1959 "/{eventsUrl}": {
1960     "post": {
1961         "tags": [
1962             "Events"
1963         ],
1964         "summary": "Events endpoint provided by the subscriber, where events are delivered",
1965         "description": "Events endpoint provided during subscription where notifications for the
1966 events specified in the subscription will be sent to as defined per event type. Confirmation of
1967 each notification sent to the \"{eventsUrl}\" endpoint is required with a \"2xx\" success
1968 code.\n\nNotifications you may receive based on the event type you're subscribed to are:\n -
1969 \"subscription_cancelled\": \"SubscriptionCancelledEvent\"\n - \"devices_registered\":
1970 \"DevicesRegisteredEvent\"\n - \"devices_unregistered\": \"DevicesUnregisteredEvent\"\n -
1971 \"resources_published\": \"ResourcesPublishedEvent\"\n - \"resources_unpublished\":
1972 \"ResourcesUnpublishedEvent\"\n - \"devices_online\": \"DevicesOnlineEvent\"\n -
1973 \"devices_offline\": \"DevicesOfflineEvent\"\n - \"resource_contentchanged\":
1974 \"ResourceContentChangedEvent\"",
1975         "parameters": [
1976             {
1977                 "$ref": "#/parameters/CorrelationId"
1978             },
1979             {
1980                 "$ref": "#/parameters/ContentType"
1981             },
1982             {
1983                 "$ref": "#/parameters/EventType"
1984             },
1985             {
1986                 "$ref": "#/parameters/SubscriptionId"
1987             },
1988             {
1989                 "$ref": "#/parameters/SequenceNumber"
1990             },
1991             {
1992                 "$ref": "#/parameters/EventSignature"
1993             },
1994             {
1995                 "$ref": "#/parameters/EventTimestamp"
1996             },
1997             {
1998                 "$ref": "#/parameters/EventsUrl"
1999             },
2000             {
2001                 "$ref": "#/parameters/Event"
2002             }
2003         ],
2004         "consumes": [

```

```

2005         "application/json",
2006         "application/vnd.ocf+cbor"
2007     ],
2008     "responses": {
2009         "200": {
2010             "description": "Event successfully recieved"
2011         },
2012         "400": {
2013             "$ref": "#/responses/BadRequest"
2014         },
2015         "410": {
2016             "description": "The subscription identified by the Subscription-ID header is no more in
2017 demand and shall be cancelled"
2018         }
2019     }
2020 }
2021 }
2022 },
2023 "securityDefinitions": {
2024     "oauth2": {
2025         "type": "oauth2",
2026         "flow": "accessToken",
2027         "authorizationUrl": "https://example.com/api/oauth/dialog",
2028         "tokenUrl": "https://example.com/api/oauth/token",
2029         "scopes": {
2030             "r:*": "Read device data",
2031             "w:*": "Update content of published resource"
2032         }
2033     }
2034 },
2035 "parameters": {
2036     "CorrelationId": {
2037         "name": "Correlation-ID",
2038         "in": "header",
2039         "type": "string",
2040         "format": "uuid",
2041         "description": "A Correlation ID, also known as a Transit ID, is a unique identifier value
2042 that is attached to requests and messages that allow reference to a particular transaction or event
2043 chain.\n"
2044     },
2045     "ContentType": {
2046         "name": "Content-Type",
2047         "in": "header",
2048         "type": "string",
2049         "enum": [
2050             "application/json",
2051             "application/vnd.ocf+cbor"
2052         ],
2053         "required": true,
2054         "description": "The Content-Type header is used to indicate the media type of the resource. In
2055 responses, a Content-Type header tells the client what the content type of the returned content
2056 actually is. In requests, (such as POST), the client tells the server what type of data is actually
2057 sent.\n"
2058     },
2059     "Accept": {
2060         "name": "Accept",
2061         "in": "header",
2062         "type": "string",
2063         "enum": [
2064             "application/json",
2065             "application/vnd.ocf+cbor"
2066         ],
2067         "description": "The Accept request header can be used to specify certain media types which are
2068 acceptable for the response. Accept headers can be used to indicate that the request is specifically
2069 limited to a small set of desired types.\n"
2070     },
2071     "SubscriptionId": {
2072         "name": "Subscription-ID",
2073         "in": "header",
2074         "description": "Unique id of the subscription",
2075         "type": "string",

```

```

2076         "format": "uuid",
2077         "required": true
2078     },
2079     "SequenceNumber": {
2080         "name": "Sequence-Number",
2081         "in": "header",
2082         "description": "Sequence number of the event; first event starting with number 0",
2083         "type": "string",
2084         "required": true
2085     },
2086     "EventSignature": {
2087         "name": "Event-Signature",
2088         "in": "header",
2089         "description": "The signature created by combining the `signingSecret` from the subscription
2090 request, headers and the body of the request using a stanard HMAC-SHA256 keyed hash.",
2091         "type": "string",
2092         "required": true
2093     },
2094     "EventTimestamp": {
2095         "name": "Event-Timestamp",
2096         "in": "header",
2097         "description": "Time when the event occurred in standard Unix time format",
2098         "type": "string",
2099         "required": true
2100     },
2101     "EventType": {
2102         "name": "Event-Type",
2103         "in": "header",
2104         "type": "string",
2105         "enum": [
2106             "subscription_cancelled",
2107             "devices_registered",
2108             "devices_unregistered",
2109             "resource_contentchanged",
2110             "resources_published",
2111             "resources_unpublished",
2112             "devices_online",
2113             "devices_offline"
2114         ],
2115         "required": true
2116     },
2117     "DeviceType": {
2118         "description": "Filter devices by device type",
2119         "name": "rt",
2120         "in": "query",
2121         "type": "array",
2122         "items": {
2123             "type": "string"
2124         }
2125     },
2126     "ResourceLinkHref": {
2127         "description": "Path to resource",
2128         "name": "resourceLinkHref",
2129         "in": "path",
2130         "type": "string",
2131         "required": true
2132     },
2133     "DeviceId": {
2134         "description": "Id of the device",
2135         "name": "deviceId",
2136         "in": "path",
2137         "type": "string",
2138         "format": "uuid",
2139         "required": true
2140     },
2141     "SubscriptionIdPath": {
2142         "name": "subscriptionId",
2143         "in": "path",
2144         "type": "string",
2145         "format": "uuid",
2146         "required": true

```

```

2147     },
2148     "BatchFormat": {
2149         "name": "content",
2150         "in": "query",
2151         "description": "Indicates to the recipient that the response payload shall be the resolved
2152 (i.e. resource representation) Link and not the Link itself. Default is `base`. When requesting
2153 `all`, additional scope `r:*` is required",
2154         "type": "string",
2155         "enum": [
2156             "base",
2157             "all"
2158         ]
2159     },
2160     "EventsUrl": {
2161         "name": "eventsUrl",
2162         "type": "string",
2163         "in": "path",
2164         "required": true
2165     },
2166     "ResourceUpdateRequest": {
2167         "description": "Map of resource values encoded to application/vnd.ocf+cbor type",
2168         "name": "content",
2169         "in": "body",
2170         "schema": {
2171             "$ref": "#/definitions/ResourceUpdateRequest"
2172         },
2173         "required": true
2174     },
2175     "SubscribeRequestDevices": {
2176         "name": "content",
2177         "in": "body",
2178         "schema": {
2179             "$ref": "#/definitions/SubscribeRequestDevices"
2180         },
2181         "required": true
2182     },
2183     "SubscribeRequestDevice": {
2184         "name": "content",
2185         "in": "body",
2186         "schema": {
2187             "$ref": "#/definitions/SubscribeRequestDevice"
2188         },
2189         "required": true
2190     },
2191     "SubscribeRequestResources": {
2192         "name": "content",
2193         "in": "body",
2194         "schema": {
2195             "$ref": "#/definitions/SubscribeRequestResources"
2196         },
2197         "required": true
2198     },
2199     "Event": {
2200         "description": "Event of a specific type, based on what you are subscribed to",
2201         "name": "content",
2202         "in": "body",
2203         "schema": {
2204             "$ref": "#/definitions/ResourceContentChangedEvent"
2205         },
2206         "required": true
2207     }
2208 },
2209 "responses": {
2210     "Unauthorized": {
2211         "description": "Unauthorized"
2212     },
2213     "NotFound": {
2214         "description": "Not found"
2215     },
2216     "SubscriptionCancellationPending": {
2217         "description": "Subscription was marked for cancellation"

```

```

2218     },
2219     "Forbidden": {
2220       "description": "Insufficient permissions"
2221     },
2222     "BadRequest": {
2223       "description": "The request was malformed or badly constructed"
2224     },
2225     "ServiceUnavailable": {
2226       "description": "The service on the Target Cloud is unavailable for the reason indicated in the
2227 diagnostic payload"
2228     },
2229     "GatewayTimeout": {
2230       "description": "The target Device is registered at the target Cloud, however the Device itself
2231 is unavailable, offline, or otherwise unreachable. The response should include a Retry-After header
2232 containing the time after which the request may be re-attempted. Additional information is indicated
2233 in the diagnostic payload."
2234     },
2235     "UnsupportedMediaType": {
2236       "description": "The request contained an unsupported media type in the Content-Type header"
2237     },
2238     "NotAcceptable": {
2239       "description": "The server cannot honour the Content-Type requested in the Accept header"
2240     }
2241   },
2242   "definitions": {
2243     "DeviceProperties": {
2244       "type": "object",
2245       "required": ["rt", "di", "dmn", "n"],
2246       "properties": {
2247         "rt": {
2248           "description": "Resource Type of the Resource",
2249           "items": {
2250             "type": "string",
2251             "maxLength": 64
2252           },
2253           "minItems": 1,
2254           "readOnly": true,
2255           "uniqueItems": true,
2256           "type": "array"
2257         },
2258         "di": {
2259           "allOf": [
2260             {
2261               "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
2262 schema.json#/definitions/uuid"
2263             },
2264             {
2265               "description": "Unique identifier for the Device",
2266               "readOnly": true
2267             }
2268           ]
2269         },
2270         "dmn": {
2271           "description": "Manufacturer Name.",
2272           "items": {
2273             "properties": {
2274               "language": {
2275                 "allOf": [
2276                   {
2277                     "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
2278 schema.json#/definitions/language-tag"
2279                   },
2280                   {
2281                     "description": "An RFC 5646 language tag.",
2282                     "readOnly": true
2283                   }
2284                 ]
2285               },
2286               "value": {
2287                 "description": "Manufacturer name in the indicated language.",
2288                 "maxLength": 64,

```

```

2289         "readOnly": true,
2290         "type": "string"
2291     },
2292 },
2293     "type": "object"
2294 },
2295     "minItems": 1,
2296     "readOnly": true,
2297     "type": "array"
2298 },
2299     "n": {
2300         "$ref" :
2301 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
2302 schema.json#/definitions/n"
2303     }
2304 },
2305 },
2306 "Device": {
2307     "type": "object",
2308     "required": ["device", "status", "links"],
2309     "properties": {
2310         "device": {
2311             "$ref": "#/definitions/DeviceProperties"
2312         },
2313         "status": {
2314             "$ref": "#/definitions/DeviceStatus"
2315         },
2316         "links": {
2317             "type": "array",
2318             "items": {
2319                 "$ref":
2320 "http://openconnectivityfoundation.github.io/core/swagger2.0/oic.wk.res.swagger.json#/definitions/oi
2321 c.oic-link"
2322             }
2323         }
2324     },
2325     "example": {
2326         "device": {
2327             "rt": ["oic.wk.d", "oic.d.sensor"],
2328             "dmn": [{"language": "en", "value": "Open Connectivity Foundation"}],
2329             "n": "Food safety sensor",
2330             "di": "53080a4f-5e3e-4291-802f-3436238232d2"
2331         },
2332         "status": "online",
2333         "links": [
2334             {
2335                 "href": "/53080a4f-5e3e-4291-802f-3436238232d2/oic/d",
2336                 "rt": [
2337                     "oic.wk.d",
2338                     "oic.d.sensor"
2339                 ],
2340                 "if": [
2341                     "oic.if.r",
2342                     "oic.if.baseline"
2343                 ]
2344             },
2345             {
2346                 "href": "/53080a4f-5e3e-4291-802f-3436238232d2/oic/p",
2347                 "rt": [
2348                     "oic.wk.p"
2349                 ],
2350                 "if": [
2351                     "oic.if.r",
2352                     "oic.if.baseline"
2353                 ]
2354             },
2355             {
2356                 "href": "/53080a4f-5e3e-4291-802f-3436238232d2/humidity",
2357                 "rt": [
2358                     "oic.r.humidity"
2359                 ],

```

```

2360         "if": [
2361             "oic.if.s",
2362             "oic.if.baseline"
2363         ]
2364     },
2365     {
2366         "href": "/53080a4f-5e3e-4291-802f-3436238232d2/temperature",
2367         "rt": [
2368             "oic.r.temperature"
2369         ],
2370         "if": [
2371             "oic.if.s",
2372             "oic.if.baseline"
2373         ]
2374     }
2375 ]
2376 },
2377 },
2378 "DeviceContentAll": {
2379     "type": "object",
2380     "required": ["device", "status", "links"],
2381     "properties": {
2382         "device": {
2383             "$ref": "#/definitions/DeviceProperties"
2384         },
2385         "status": {
2386             "$ref": "#/definitions/DeviceStatus"
2387         },
2388         "links": {
2389             "type": "array",
2390             "items": {
2391                 "type": "object",
2392                 "properties": {
2393                     "href": {
2394                         "type": "string"
2395                     },
2396                     "rep": {
2397                         "oneOf": [
2398                             {
2399                                 "type": "object"
2400                             },
2401                             {
2402                                 "type": "array"
2403                             }
2404                         ]
2405                     }
2406                 }
2407             }
2408         }
2409     },
2410     "example": {
2411         "device": {
2412             "rt": ["oic.wk.d", "oic.d.sensor"],
2413             "dmn": [{"language": "en", "value": "Open Connectivity Foundation"}],
2414             "n": "Food safety sensor",
2415             "di": "53080a4f-5e3e-4291-802f-3436238232d2"
2416         },
2417         "status": "online",
2418         "links": [
2419             {
2420                 "href": "/53080a4f-5e3e-4291-802f-3436238232d2/oic/d",
2421                 "rep": {
2422                     "rt": ["oic.wk.d", "oic.d.sensor"],
2423                     "dmn": [{"language": "en", "value": "Open Connectivity Foundation"}],
2424                     "n": "Food safety sensor",
2425                     "di": "53080a4f-5e3e-4291-802f-3436238232d2",
2426                     "icv": "ocf.2.0.5",
2427                     "dmv": "ocf.res.1.3.0, ocf.sh.1.3.0",
2428                     "piid": "6F0AAC04-2BB0-468D-B57C-16570A26AE48"
2429                 }
2430             }
2431         ]
2432     }
2433 }

```

```

2431     {
2432         "href": "/53080a4f-5e3e-4291-802f-3436238232d2/oic/p",
2433         "rep": {
2434             "pi": "54919CA5-4101-4AE4-595B-353C51AA983C",
2435             "mnfv": "1.1.1.20"
2436         }
2437     },
2438     {
2439         "href": "/53080a4f-5e3e-4291-802f-3436238232d2/humidity",
2440         "rep": {
2441             "humidity": 62,
2442             "desiredHumidity": 65
2443         }
2444     },
2445     {
2446         "href": "/53080a4f-5e3e-4291-802f-3436238232d2/temperature",
2447         "rep": {
2448             "temperature": 21,
2449             "units": "C"
2450         }
2451     }
2452 ]
2453 }
2454 },
2455 "DeviceStatus": {
2456     "description": "Device status available from the OCF Cloud, which tracks if the device has
2457 opened TCP connection and is signed in",
2458     "type": "string",
2459     "enum": [
2460         "online",
2461         "offline"
2462     ]
2463 },
2464 "ResourceUpdateRequest": {
2465     "type": "string",
2466     "description": "Desired content of the resource",
2467     "example": "o29kZXNpcmVkSHVtaWRpdHkYPGV0eXBlc4Fub2ljLnIuaHVtaWRpdHloaHVtaWRpdHkYKA=="
2468 },
2469 "ResourceRetrieveResponse": {
2470     "type": "string",
2471     "description": "Content of the resource returned from the device",
2472     "example": "o29kZXNpcmVkSHVtaWRpdHkYPGV0eXBlc4Fub2ljLnIuaHVtaWRpdHloaHVtaWRpdHkYKA=="
2473 },
2474 "EventType": {
2475     "type": "string",
2476     "enum": [
2477         "subscription_cancelled",
2478         "devices_registered",
2479         "devices_unregistered",
2480         "resource_contentchanged",
2481         "resources_published",
2482         "resources_unpublished",
2483         "devices_online",
2484         "devices_offline"
2485     ]
2486 },
2487 "EventTypeDevices": {
2488     "type": "string",
2489     "enum": [
2490         "devices_registered",
2491         "devices_unregistered",
2492         "devices_online",
2493         "devices_offline"
2494     ]
2495 },
2496 "EventTypeDevice": {
2497     "type": "string",
2498     "enum": [
2499         "resources_published",
2500         "resources_unpublished"
2501     ]

```

```

2502     },
2503     "EventTypeResources": {
2504         "type": "string",
2505         "enum": [
2506             "resource_contentchanged"
2507         ]
2508     },
2509     "SubscriptionId": {
2510         "description": "Unique id of the subscription",
2511         "type": "string",
2512         "format": "uuid"
2513     },
2514     "SubscribeRequestDevices": {
2515         "type": "object",
2516         "properties": {
2517             "eventsUrl": {
2518                 "$ref": "#/definitions/EventsUrl"
2519             },
2520             "eventTypes": {
2521                 "type": "array",
2522                 "items": {
2523                     "$ref": "#/definitions/EventTypeDevices"
2524                 }
2525             },
2526             "signingSecret": {
2527                 "type": "string",
2528                 "maxLength": 32,
2529                 "minLength": 32
2530             }
2531         },
2532         "required": [
2533             "eventsUrl",
2534             "eventTypes",
2535             "signingSecret"
2536         ],
2537         "example": {
2538             "eventsUrl": "https://events.example.com/",
2539             "eventTypes": [
2540                 "devices_registered",
2541                 "devices_unregistered"
2542             ],
2543             "signingSecret": "3BZ6oI9xbRJzOUvUoRb5RgaZjPqHrmql"
2544         }
2545     },
2546     "SubscribeRequestDevice": {
2547         "type": "object",
2548         "properties": {
2549             "eventsUrl": {
2550                 "$ref": "#/definitions/EventsUrl"
2551             },
2552             "eventTypes": {
2553                 "type": "array",
2554                 "items": {
2555                     "$ref": "#/definitions/EventTypeDevice"
2556                 }
2557             },
2558             "signingSecret": {
2559                 "type": "string",
2560                 "maxLength": 32,
2561                 "minLength": 32
2562             }
2563         },
2564         "required": [
2565             "eventsUrl",
2566             "eventTypes",
2567             "signingSecret"
2568         ],
2569         "example": {
2570             "eventsUrl": "https://events.example.com/",
2571             "eventTypes": [
2572                 "resource_published",

```

```

2573         "resource_unpublished"
2574     ],
2575     "signingSecret": "3BZ6oI9xbRJzOUvUoRb5RgaZjPqHrmql"
2576 }
2577 },
2578 "SubscribeRequestResources": {
2579     "type": "object",
2580     "properties": {
2581         "eventsUrl": {
2582             "$ref": "#/definitions/EventsUrl"
2583         },
2584         "eventTypes": {
2585             "type": "array",
2586             "items": {
2587                 "$ref": "#/definitions/EventTypeResources"
2588             }
2589         },
2590         "signingSecret": {
2591             "type": "string",
2592             "maxLength": 32,
2593             "minLength": 32
2594         }
2595     },
2596     "required": [
2597         "eventsUrl",
2598         "eventTypes",
2599         "signingSecret"
2600     ],
2601     "example": {
2602         "eventsUrl": "https://events.example.com/",
2603         "eventTypes": [
2604             "resource_contentchanged"
2605         ],
2606         "signingSecret": "3BZ6oI9xbRJzOUvUoRb5RgaZjPqHrmql"
2607     }
2608 },
2609 "SubscribeResponse": {
2610     "description": "Subscription was registered, waiting for verification",
2611     "type": "object",
2612     "properties": {
2613         "subscriptionId": {
2614             "$ref": "#/definitions/SubscriptionId"
2615         }
2616     },
2617     "required": [
2618         "subscriptionId"
2619     ],
2620     "example": {
2621         "subscriptionId": "1eeb465c-5e8d-4305-a366-bbf035fff671"
2622     }
2623 },
2624 "EventsUrl": {
2625     "type": "string",
2626     "format": "url",
2627     "example": "https://events.example.com/"
2628 },
2629 "SubscriptionCancelledEvent": {
2630     "type": "object",
2631     "description": "Subscription with provided id was cancelled"
2632 },
2633 "DevicesRegisteredEvent": {
2634     "description": "Device was successfully signed up to the OCF Cloud, as defined in the
2635 `oic.sec.account`",
2636     "type": "object",
2637     "properties": {
2638         "content": {
2639             "type": "array",
2640             "items": {
2641                 "properties": {
2642                     "di": {
2643                         "type": "string",

```

```

2644         "format": "uuid"
2645     }
2646 }
2647 }
2648 }
2649 }
2650 },
2651 "DevicesUnregisteredEvent": {
2652     "description": "Device was successfully signed off from the OCF Cloud, as defined in the
2653 `oic.sec.account`,
2654     "type": "object",
2655     "properties": {
2656         "content": {
2657             "type": "array",
2658             "items": {
2659                 "properties": {
2660                     "di": {
2661                         "type": "string",
2662                         "format": "uuid"
2663                     }
2664                 }
2665             }
2666         }
2667     }
2668 },
2669 "ResourcesPublishedEvent": {
2670     "type": "object",
2671     "properties": {
2672         "content": {
2673             "type": "array",
2674             "items": {
2675                 "$ref":
2676 "http://openconnectivityfoundation.github.io/core/swagger2.0/oic.wk.res.swagger.json#/definitions/oi
2677 c.oic-link"
2678             }
2679         }
2680     }
2681 },
2682 "ResourcesUnpublishedEvent": {
2683     "type": "object",
2684     "properties": {
2685         "content": {
2686             "type": "array",
2687             "items": {
2688                 "$ref":
2689 "http://openconnectivityfoundation.github.io/core/swagger2.0/oic.wk.res.swagger.json#/definitions/oi
2690 c.oic-link"
2691             }
2692         }
2693     }
2694 },
2695 "DevicesOnlineEvent": {
2696     "type": "object",
2697     "properties": {
2698         "content": {
2699             "type": "array",
2700             "items": {
2701                 "properties": {
2702                     "di": {
2703                         "type": "string",
2704                         "format": "uuid"
2705                     }
2706                 }
2707             }
2708         }
2709     }
2710 },
2711 "DevicesOfflineEvent": {
2712     "type": "object",
2713     "properties": {
2714         "content": {

```

```
2715         "type": "array",
2716         "items": {
2717             "properties": {
2718                 "di": {
2719                     "type": "string",
2720                     "format": "uuid"
2721                 }
2722             }
2723         }
2724     },
2725 },
2726 },
2727 "ResourceContentChangedEvent": {
2728     "type": "string",
2729     "description": "New Content of the resource returned from the device",
2730     "example": "o29kZZNpcmVkSHVtaWRpdHkYPGV0eXBlc4Fub2ljLnIuaHVtaWRpdHloaHVtaWRpdHkYKA=="
2731 }
2732 }
2733 }
2734 }
```