OCF Cloud API for Cloud Services Specification

VERSION 2.2.4 | August 2021

CONTACT admin@openconnectivity.org

Legal Disclaimer

CONTENTS

Figures

Tables

Introduction

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

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

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

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

The OCF specification suite is made up of nineteen discrete documents, the documents fall into logical groupings as described herein:

- Core framework
  - Core Specification
  - Security Specification
  - Onboarding Tool Specification
- Bridging framework and bridges
  - Bridging Specification
  - Resource to Alljoyn Interface Mapping Specification
  - OCF Resource to oneM2M Resource Mapping Specification
  - OCF Resource to BLE Mapping Specification
  - OCF Resource to EnOcean Mapping Specification
  - OCF Resource to LWM2M Mapping Specification
  - OCF Resource to UPlus Mapping Specification
  - OCF Resource to Zigbee Cluster Mapping Specification
  - OCF Resource to Z-Wave Mapping Specification
- Resource and Device models
  - Resource Type Specification
  - Device Specification
- Core framework extensions
  - Easy Setup Specification
  - Core Optional Specification
- OCF Cloud
  - Cloud API for Cloud Services Specification

- Device to Cloud Services Specification
- Cloud Security Specification

Cloud API for Cloud Services Specification

1 Scope

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

2 Normative references

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

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

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

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

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

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

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

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

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

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

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

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

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

OCF Device to Cloud Services, *Open Connectivity Foundation Device to Cloud Services Specification, Version 2.2.0*

Available at:
https://openconnectivity.org/specs/OCF_Device_To_Cloud_Services_Specification_v2.2.0.pdf
Latest version available at:
https://openconnectivity.org/specs/OCF_Device_To_Cloud_Services_Specification.pdf

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

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

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 30118-1 and ISO/IEC 30118-2, OCF Device to Cloud Services and the following apply.

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

- ISO Online browsing platform: available at https://www.iso.org/obp
- IEC Electropedia: available at http://www.electropedia.org/

3.1.1
API Endpoint
defined URL to which requests defined in this document are sent

3.1.2
Bearer Token
OAuth2.0 access token as defined within IETF RFC 6750

3.1.3
Origin Cloud
OCF Cloud through which the user works with his OCF Devices

3.1.4
Subscription ID
unique identity that is associated with an instance of a subscription to an event (or events)

3.1.5
Target Cloud
OCF Cloud to which OCF Servers (OCF Devices) are connected which the user wants to control via the *Origin Cloud* (3.1.2)

3.2 Symbols and abbreviated terms

API	Application Programming Interface
HMAC	Hash-based Message Authentication Code

4 Document conventions and organization

4.1 Conventions

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

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

4.2 Notation

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

Required (or shall or mandatory)(M).

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

Recommended (or should)(S).

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

Allowed (may or allowed)(O).

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

DEPRECATED.

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

Conditionally allowed (CA)

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

Conditionally required (CR)

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

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

Words that are emphasized are printed in italic.

5 Overview

5.1 Introduction

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

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

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

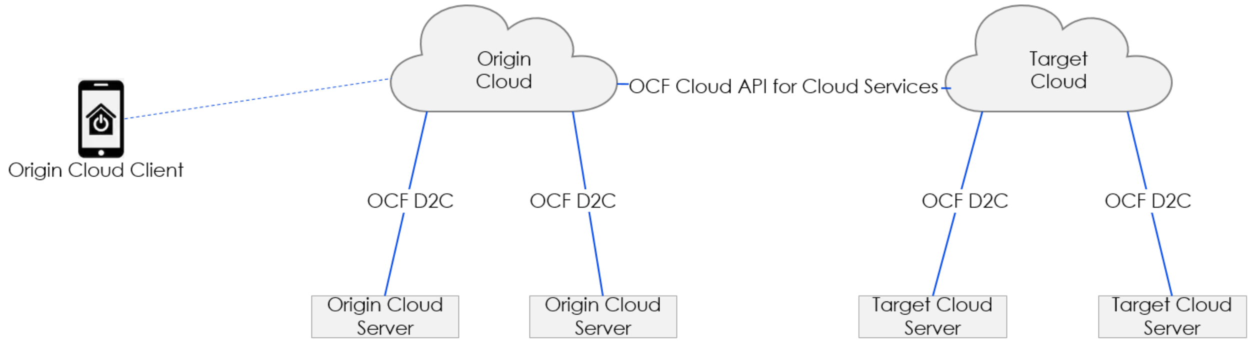

Figure 1 – OCF Cloud overview

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

- Account Linking API (clause 7)
  - Initial Account Linking
  - Removal of linked account
- Devices API (clause 8)
  - Retrieval of all Devices associated with a User (clause 8.3)
  - Retrieval of a single Device associated with a User (clause 8.4)
  - Retrieval of a single Resource (clause 8.5)
  - Update of a single Resource (clause 8.6)
- Events API (clause 9)
  - Subscription to an event: establishment of a subscription (clause 9.4.1)
  - Notification: event generated on an established subscription (clause 9.4.3)

5.2 OCF Cloud architecture alignment with ISO IEC 17789

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

OCF defines an application capabilities type cloud service, providing Communication as a Service (CaaS) (reference ISO/IEC 17788). This cloud service is provided by a cloud service provider, the mechanisms used by the cloud service provider in managing their overall cloud infrastructure are outside the scope of the OCF defined cloud service. The OCF definition is specific to the interface offered by the cloud service to the cloud service customer, specifically the cloud service user.

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

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

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

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

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

Credentials for the user of the cloud service are provided using OAUTH2.0 as defined by RFC 6749. The cloud service, either itself, or leveraging an external authorization server, provides a bearer token that is required in all requests from all cloud users. Please see clause 7 and OCF Cloud Security.

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

5.3 General OCF Cloud API for Cloud Services elements

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

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

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

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

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

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

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

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

5.4 Cloud to Cloud operational overview

5.4.1 Introduction

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

5.4.2 Conceptual architecture

Figure 2 describes the overall conceptual architecture.

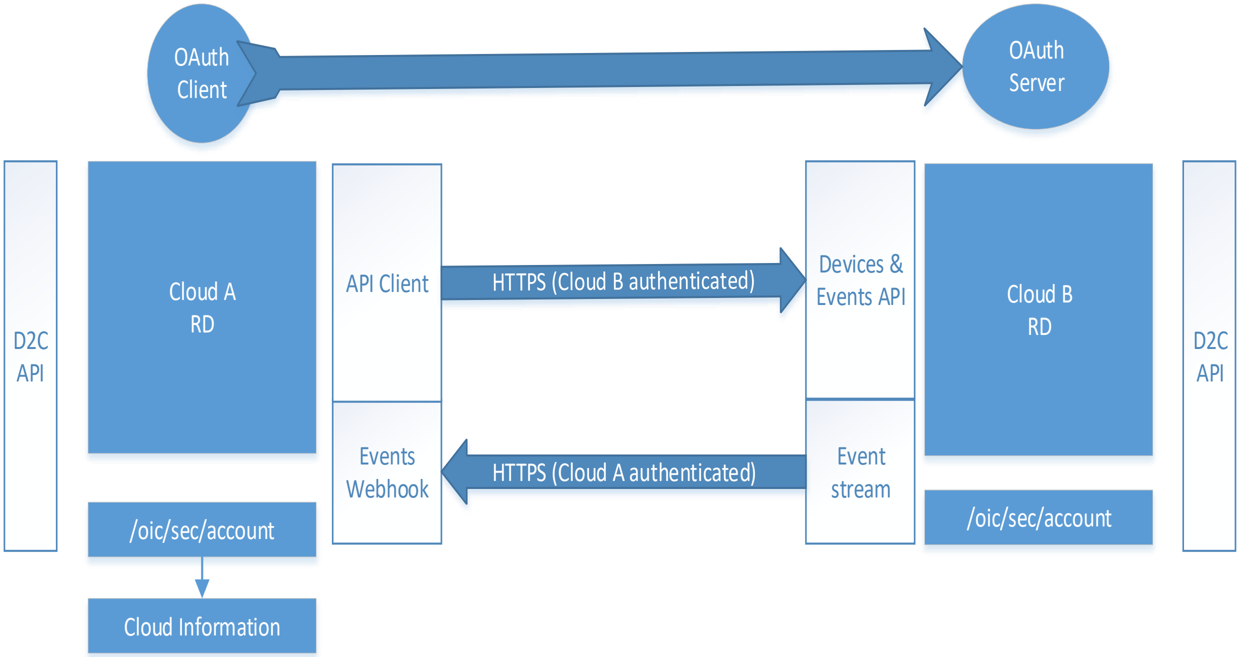

Figure 2 – Conceptual architecture

5.4.3 Authorizing OCF Cloud connectivity

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

5.4.4 Synchronization of user's set of Devices

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

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

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

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

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

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

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

5.4.6 Handling of requests and responses for connected Devices

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

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

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

6 Authentication and authorization

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

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

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

7 Account linking API

7.1 General

The account linking API is the mechanism by which Devices hosted on behalf of a user by the Target Cloud are linked with a user identity on the Origin Cloud. Account linking is established solely between the Origin Cloud and the Target Cloud; an Origin Cloud shall not proxy devices from the Target Cloud to another third-party cloud (whether that is an OCF Cloud or some other realization of cloud hosted functionality).

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

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

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

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

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

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

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

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

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

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

7.2 OAuth2.0 access token scopes

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

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

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

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

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:*

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

8 Devices API

8.1 Introduction

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

8.2 Parameters supported in Requests

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

Table 3 – Parameters used in Requests in the Device API

Friendly Name	Parameter Name	Location	Mandatory	Description
Accept	Accept	HTTP Header	Yes	An Accept request HTTP header advertises which content types, expressed as MIME types, the Origin Cloud is able to understand. The Target Cloud then selects one of the proposed content types and informs the Origin Cloud of its choice with the Content-Type response header.
Content Type	Content-Type	HTTP Header	No	The Content-Type header is used to indicate the media type of the payload. A Content-Type header tells the recipient what the content type of the returned payload actually is.

Correlation ID	Correlation-ID	HTTP Header	No	A Correlation ID, also known as a Transit ID, is a unique identifier value that is attached to requests and messages that allows reference to a particular transaction or event chain.
Content	content=[base, all]	Query String Parameter	No	When set to "base" this indicates to the recipient that the response payload Links are not resolved. When set to "all" this indicates to the recipient that the response payload is the resolved (i.e. resource representation) Link and not the Link itself. If not present "base" is assumed.

8.3 Retrieve all Devices

8.3.1 Summary

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

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

Table 4 provides a summary of the API.

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.

8.3.2 Request and response payload

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

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

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.

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

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

8.3.3 Responses

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

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

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

8.4 Retrieve one Device

8.4.1 Summary

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

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

Table 8 provides a summary of the API.

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

8.4.2 Request and response payload

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

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

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

8.4.3 Responses

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

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

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

8.5 Retrieve specific Resource

8.5.1 Summary

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

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

Table 10 provides a summary of the API.

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.

8.5.2 Request and response payload

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

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

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

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

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

8.5.3 Responses

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

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

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

8.6 Update a Resource on a Device

8.6.1 Summary

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

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

Table 12 provides a summary of the API.

Table 12 – Update Resource API summary

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

				payload containing a reason string

8.6.2 Request and response payload

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

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

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

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

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

8.6.3 Responses

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

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

Table 13 – Resource update API non-success path responses

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

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

9 Events API

9.1 Introduction

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

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

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

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

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

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

9.2 Events authentication

9.2.1 Introduction

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

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

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

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

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

9.2.2 Create event signature

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

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

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

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

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

9.2.1 Verify the event signature

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

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

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

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

9.3 Parameters supported

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

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.

9.4 Events API subscription and notification payload definitions

9.4.1 Subscription request

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

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

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

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

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

Figure 3 is an example of such a payload.

```
{
"eventsUrl": "https://mynotificationuri",
"eventTypes": ["resource_contentchanged"],
"signingSecret": "DVDUEBe5nciVSXU85BPxrAjSsHenTzWY"
}
```

Figure 3 – Subscription Request example

9.4.2 Subscription response

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

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).

Figure 4 is an example of such a payload.

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

Figure 4 – Subscription Response example payload

9.4.3 Notification request

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

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

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

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

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

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

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

9.4.4 Notification response

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

9.5 Subscribe and unsubscribe to devices level event types

9.5.1 Summary

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

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

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

Table 20 provides a summary of the API.

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	

9.5.2 Request and response payload

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

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

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

There is no required payload for a DELETE unsubscribe response.

9.5.3 Responses

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

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

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

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

Response Code	Response scenario
400	May be sent by the Target Cloud if the request was malformed or badly constructed

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

9.6 Subscribe and unsubscribe to device level events

9.6.1 Summary

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

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

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

Table 22 provides a summary of the API.

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	

9.6.2 Request and response payload

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

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

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

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

There is no required payload for a DELETE unsubscribe response.

9.6.3 Responses

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

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

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

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

9.7 Subscribe and unsubscribe to resource level events

9.7.1 Summary

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

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

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

Table 24 provides a summary of the API.

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	

9.7.2 Request and response payload

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

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

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

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

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

There is no required payload for a DELETE unsubscribe response.

9.7.3 Responses

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

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

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

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

9.8 Notification of devices level events

9.8.1 Summary

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

Table 26 provides a summary of the API.

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	

9.8.2 Request and response payload

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

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

9.8.3 Responses

A 200 response shall be provided in a success case.

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

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

9.9 Notification of Device level events

9.9.1 Summary

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

Table 28 provides a summary of the API.

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	

9.9.2 Request and response payload

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

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

9.9.3 Responses

A 200 response shall be provided in a success case.

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

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

9.10 Notification of Resource level events

9.10.1 Summary

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

Table 30 provides a summary of the API.

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	

9.10.2 Request and response payload

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

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

9.10.3 Responses

A 200 response shall be provided in a success case.

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

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

Annex A
Representative flows

A.1 Introduction

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

A.2 OAuth2.0 application registration

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

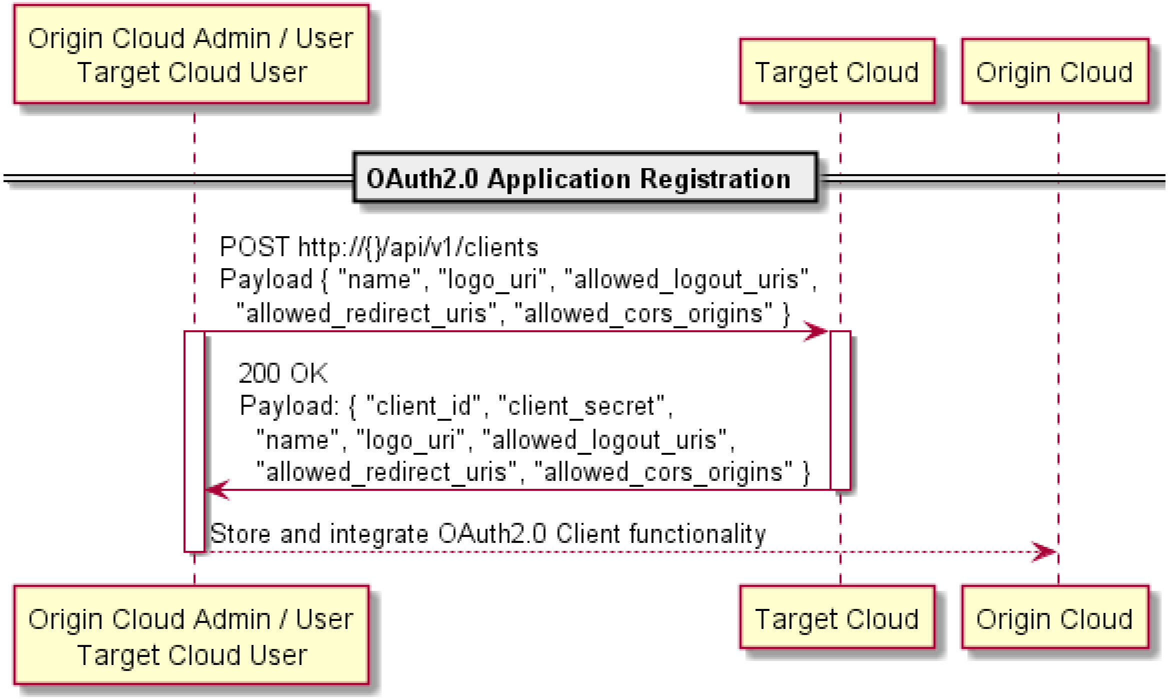
Origin Cloud Admin / User
Target Cloud User
Target Cloud
Origin Cloud
OAuth2.0 Application Registration
POST http://{}/api/v1/clients
Payload { "name", "logo_uri", "allowed_logout_uris",
"allowed_redirect_uris", "allowed_cors_origins" }
200 OK
Payload: { "client_id", "client_secret",
"name", "logo_uri", "allowed_logout_uris",
"allowed_redirect_uris", "allowed_cors_origins" }
Store and integrate OAuth2.0 Client functionality
Origin Cloud Admin / User
Target Cloud User
Target Cloud
Origin Cloud

Figure A.1 – Establish business relationship example flow

A.3 Account linking

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

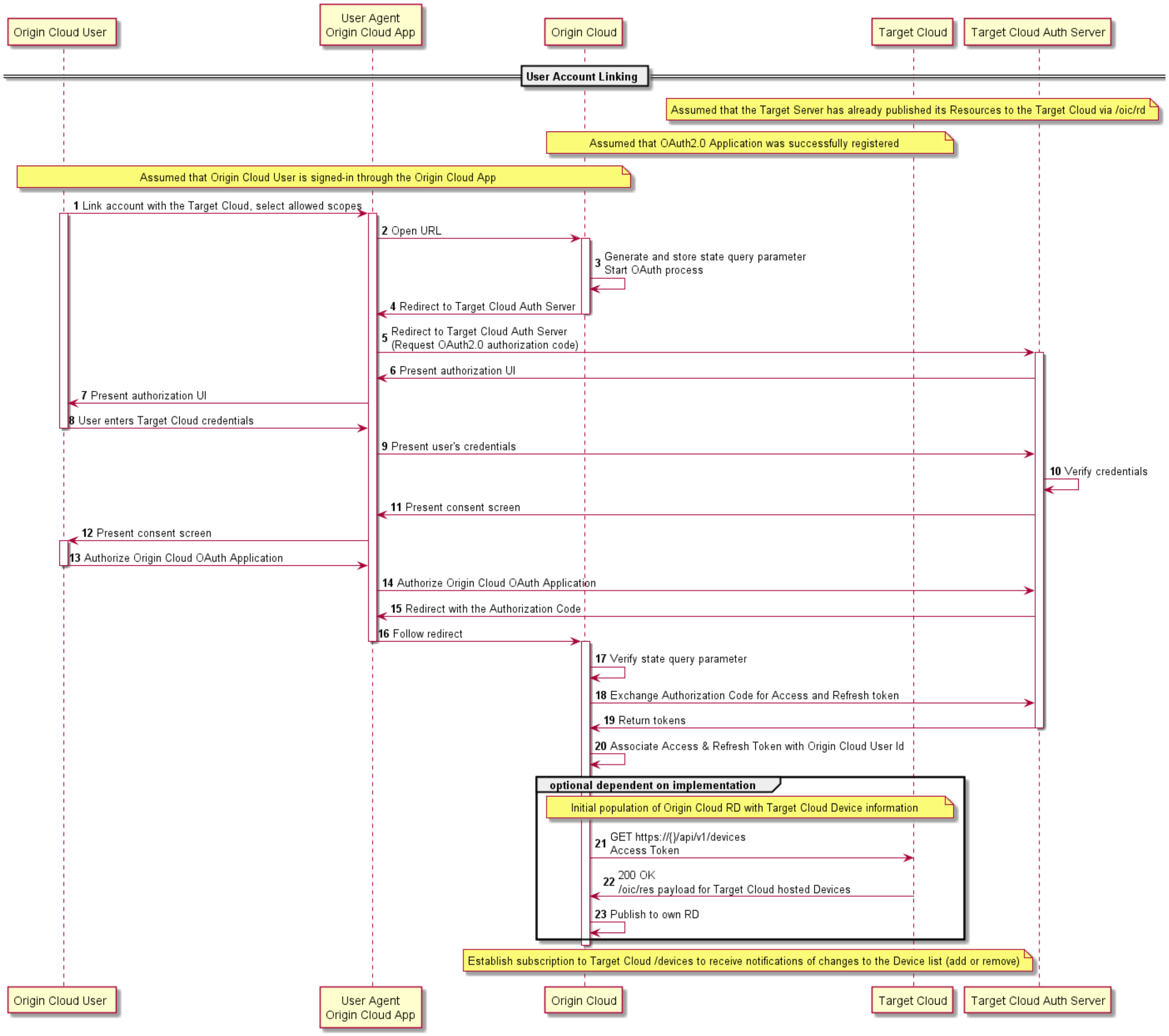
Origin Cloud User
User Agent
Origin Cloud App
Origin Cloud
Target Cloud
Target Cloud Auth Server
User Account Linking
Assumed that the Target Server has already published its Resources to the Target Cloud via /oic/rd
Assumed that OAuth2.0 Application was successfully registered
Assumed that Origin Cloud User is signed-in through the Origin Cloud App
1 Link account with the Target Cloud, select allowed scopes
2 Open URL
3 Generate and store state query parameter
Start OAuth process
4 Redirect to Target Cloud Auth Server
5 Redirect to Target Cloud Auth Server
(Request OAuth2.0 authorization code)
6 Present authorization UI
7 Present authorization UI
8 User enters Target Cloud credentials
9 Present user's credentials
10 Verify credentials
11 Present consent screen
12 Present consent screen
13 Authorize Origin Cloud OAuth Application
14 Authorize Origin Cloud OAuth Application
15 Redirect with the Authorization Code
16 Follow redirect
17 Verify state query parameter
18 Exchange Authorization Code for Access and Refresh token
19 Return tokens
20 Associate Access & Refresh Token with Origin Cloud User Id
optional dependent on implementation
Initial population of Origin Cloud RD with Target Cloud Device information
21 GET https://{}/api/v1/devices
Access Token
22 200 OK
/oic/res payload for Target Cloud hosted Devices
23 Publish to own RD
Establish subscription to Target Cloud /devices to receive notifications of changes to the Device list (add or remove)
Origin Cloud User
User Agent
Origin Cloud App
Origin Cloud
Target Cloud
Target Cloud Auth Server

Figure A.2 – Initial association example flow

A.4 Retrieval of all Devices

A.4.1 Summary

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

A.4.2 Flow

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

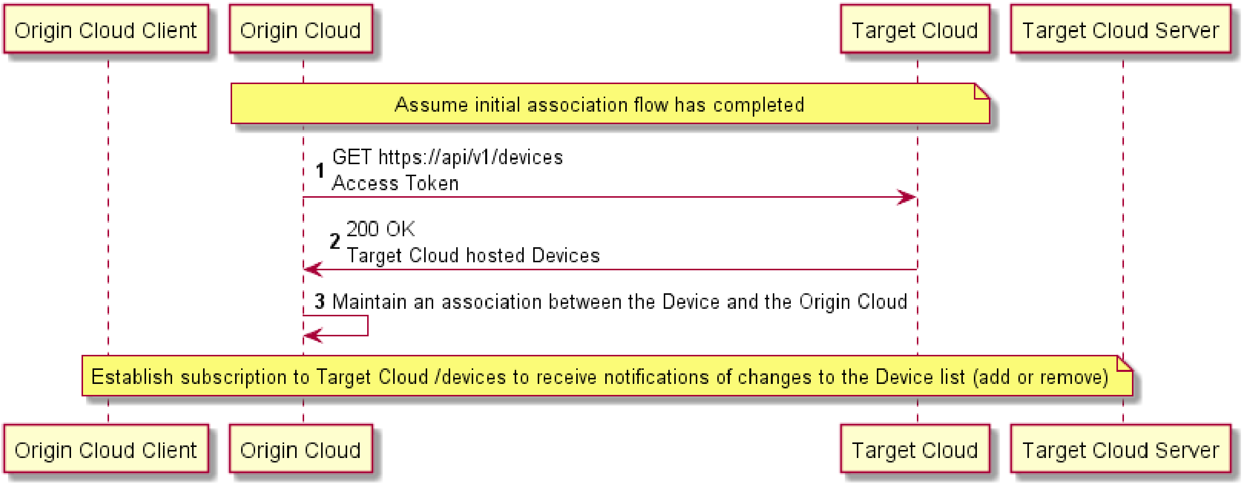
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server
Assume initial association flow has completed
1 GET https://api/v1/devices
Access Token
2 200 OK
Target Cloud hosted Devices
3 Maintain an association between the Device and the Origin Cloud
Establish subscription to Target Cloud /devices to receive notifications of changes to the Device list (add or remove)

Figure A.3 – Retrieve all Devices example flow

A.4.3 Flow description

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

Table A.1 – Retrieve all Devices flow summary

Number	Description
1	Origin 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 OCF Cloud functionality and Device status).
3	Origin Cloud maintains an association between the Device and the Target Cloud.

A.5 Retrieval of a single Device

A.5.1 Summary

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

A.5.2 Flow

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

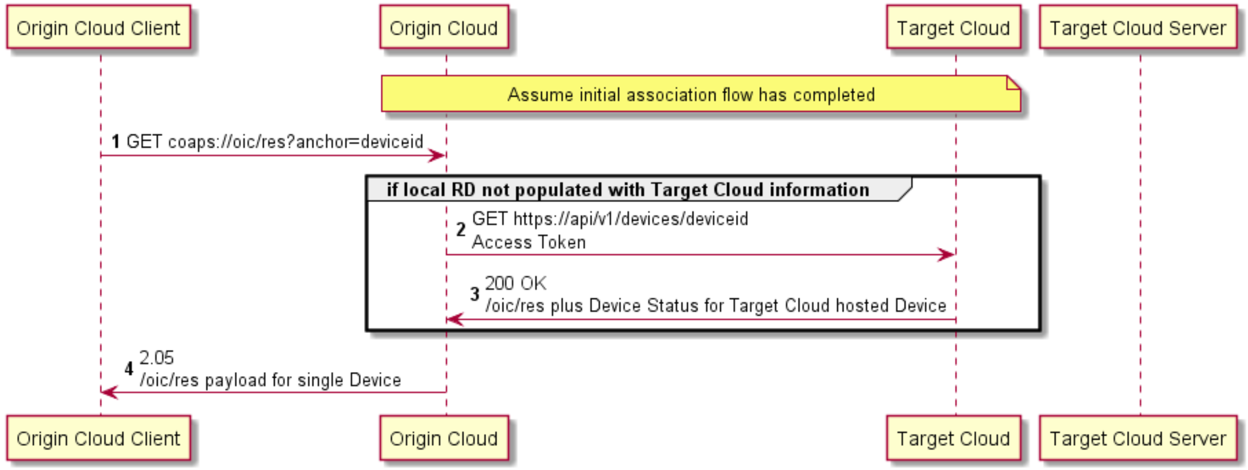
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server
Assume initial association flow has completed
1 GET coaps://oic/res?anchor=deviceid
if local RD not populated with Target Cloud information
2 GET https://api/v1/devices/deviceid
Access Token
3 200 OK
/oic/res plus Device Status for Target Cloud hosted Device
4 2.05
/oic/res payload for single Device
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server

Figure A.4 – Retrieve single Device example flow

A.5.3 Flow description

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

Table A.2 – Retrieve single Device flow summary

Number	Description
1	[OCF Device to Cloud] OCF Client role Device requests /oic/res from the Origin Cloud for a specific anchor (Device UUID).
2	[Assuming that the information hasn't been cached by the Origin Cloud] For the instance of /oic/sec/account that exists for the Device the Origin Cloud does a GET /devices/{deviceid} to the Target 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.

A.6 Retrieval of a single Resource

A.6.1 Summary

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

A.6.2 Flows

A.6.2.1 Success path

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

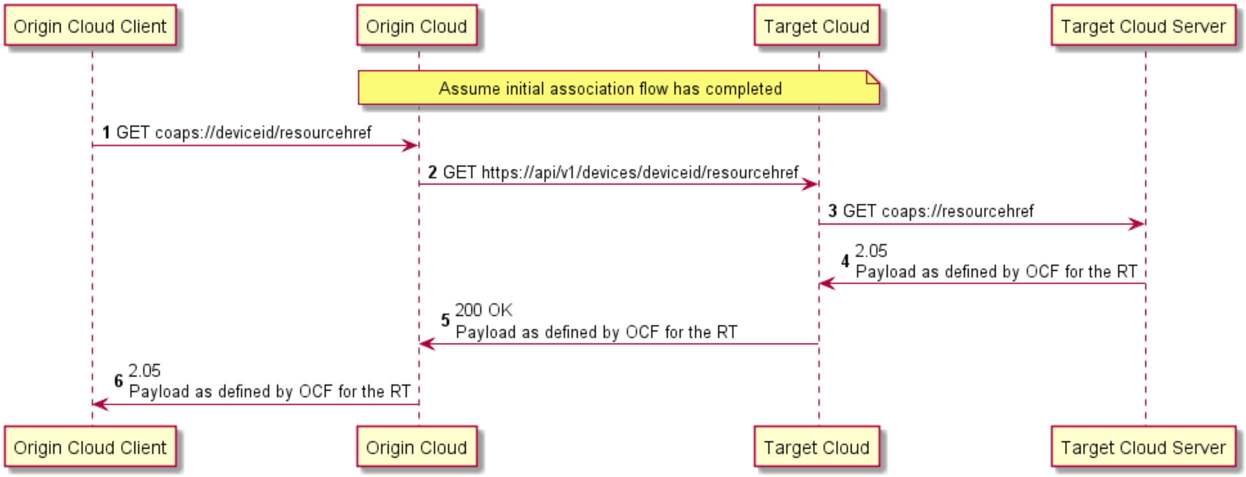
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server
Assume initial association flow has completed
1 GET coaps://deviceid/resourcehref
2 GET https://api/v1/devices/deviceid/resourcehref
3 GET coaps://resourcehref
4 2.05
Payload as defined by OCF for the RT
5 200 OK
Payload as defined by OCF for the RT
6 2.05
Payload as defined by OCF for the RT
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server

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 Origin 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 Origin Cloud] Origin Cloud identifies the Target 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.

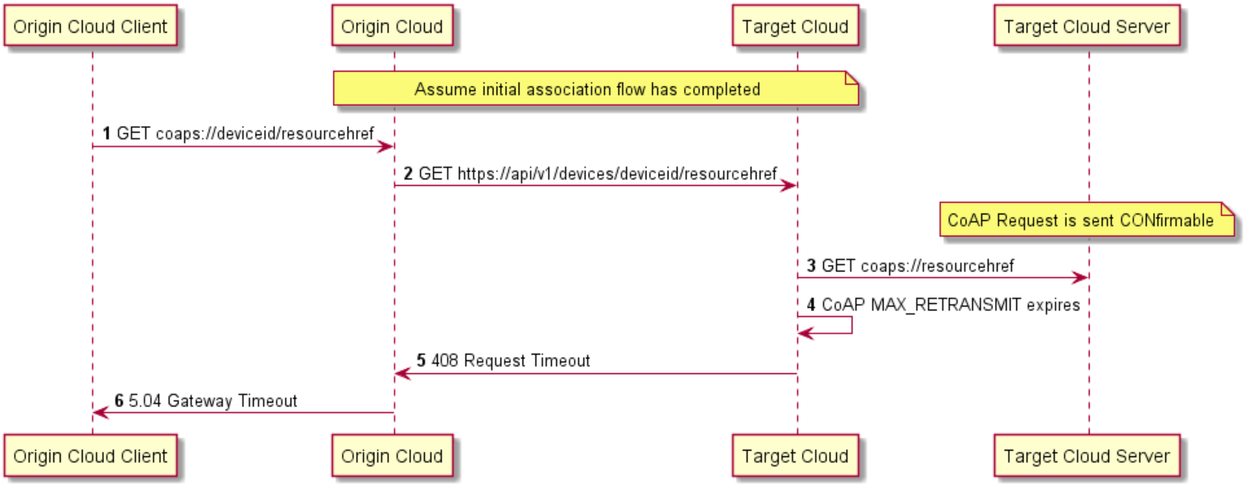
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server
Assume initial association flow has completed
1 GET coaps://deviceid/resourcehref
2 GET https://api/v1/devices/deviceid/resourcehref
CoAP Request is sent CONfirmable
3 GET coaps://resourcehref
4 CoAP MAX_RETRANSMIT expires
5 408 Request Timeout
6 5.04 Gateway Timeout

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

A.7 Update of a single Resource

A.7.1 Summary

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

A.7.2 Flows

A.7.2.1 Success path

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

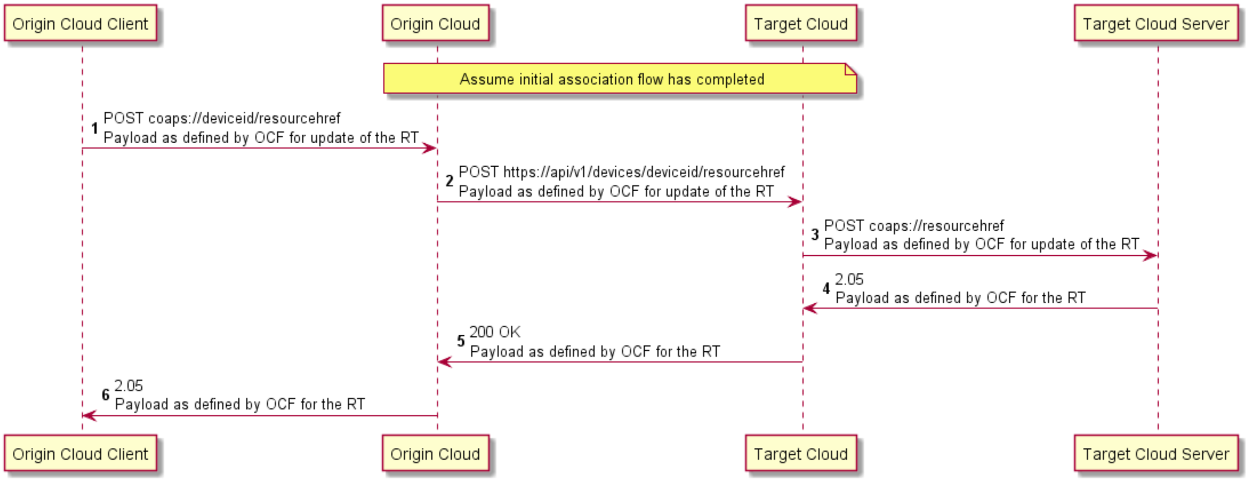
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server
Assume initial association flow has completed
1 POST coaps://deviceid/resourcehref
Payload as defined by OCF for update of the RT
2 POST https://api/v1/devices/deviceid/resourcehref
Payload as defined by OCF for update of the RT
3 POST coaps://resourcehref
Payload as defined by OCF for update of the RT
4 2.05
Payload as defined by OCF for the RT
5 200 OK
Payload as defined by OCF for the RT
6 2.05
Payload as defined by OCF for the RT

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

A.7.2.2 Success path flow description

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 Origin Cloud using the "href" exposed in the /oic/res response. This will be of the form "/deviceid/resourcehref"
2	Origin Cloud identifies the Target 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

A.7.2.3 Device is temporarily unavailable

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

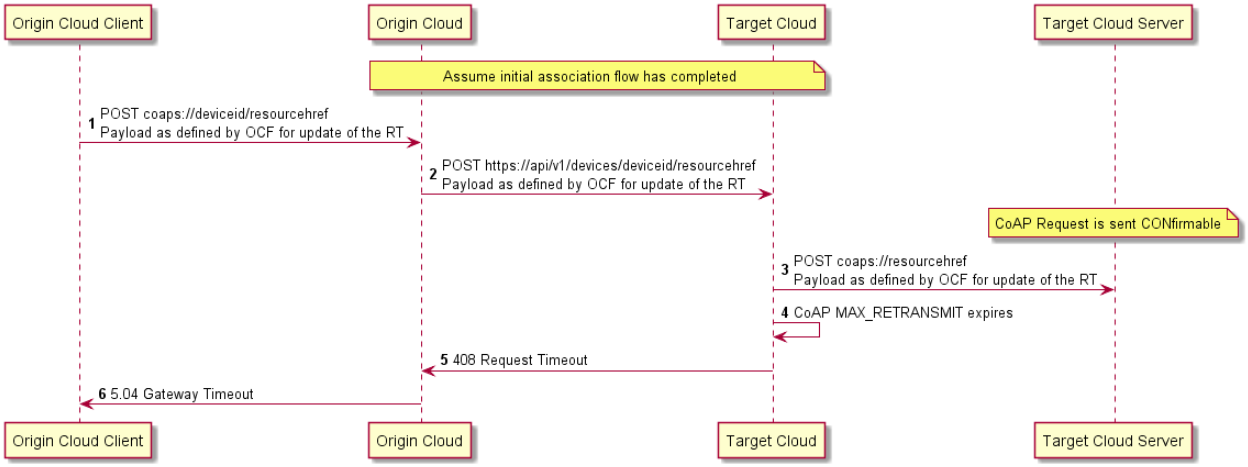
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server
Assume initial association flow has completed
1 POST coaps://deviceid/resourcehref
Payload as defined by OCF for update of the RT
2 POST https://api/v1/devices/deviceid/resourcehref
Payload as defined by OCF for update of the RT
CoAP Request is sent CONfirmable
3 POST coaps://resourcehref
Payload as defined by OCF for update of the RT
4 CoAP MAX_RETRANSMIT expires
5 408 Request Timeout
6 5.04 Gateway Timeout
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server

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

A.8 Establishment of new subscription request

A.8.1 Summary

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

A.8.2 Flows

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

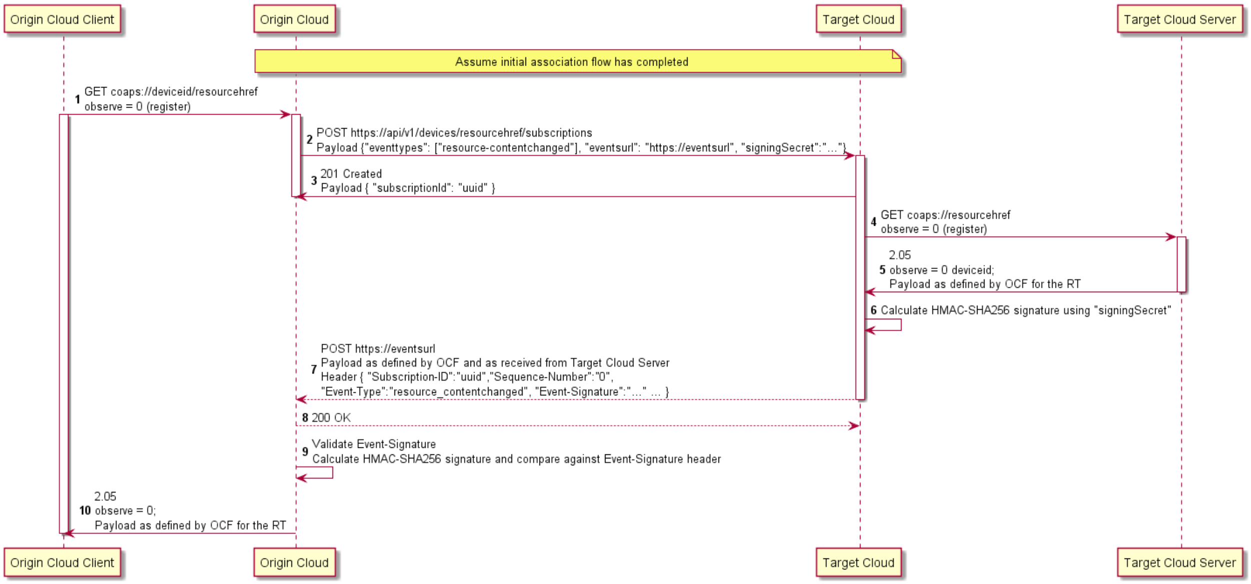
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server
Assume initial association flow has completed
1 GET coaps://deviceid/resourcehref
observe = 0 (register)
2 POST https://api/v1/devices/resourcehref/subscriptions
Payload {"eventtypes": ["resource-contentchanged"], "eventsurl": "https://eventsurl", "signingSecret":"…"}
3 201 Created
Payload { "subscriptionId": "uuid" }
4 GET coaps://resourcehref
observe = 0 (register)
2.05
5 observe = 0 deviceid;
Payload as defined by OCF for the RT
6 Calculate HMAC-SHA256 signature using "signingSecret"
POST https://eventsurl
7 Payload as defined by OCF and as received from Target Cloud Server
Header { "Subscription-ID":"uuid","Sequence-Number":"0",
"Event-Type":"resource_contentchanged", "Event-Signature":"…" … }
8 200 OK
9 Validate Event-Signature
Calculate HMAC-SHA256 signature and compare against Event-Signature header
2.05
10 observe = 0;
Payload as defined by OCF for the RT
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server

Figure A.9 – Subscription establishment example flow

A.9 Event generated for a subscription

A.9.1 Summary

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.

A.9.2 Flows

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

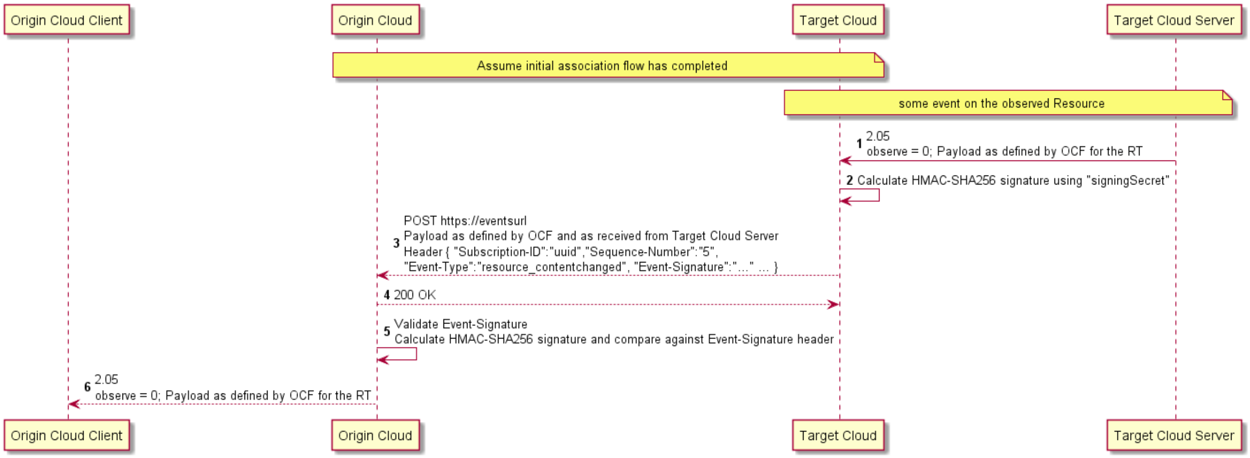
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server
Assume initial association flow has completed
some event on the observed Resource
1 2.05
observe = 0; Payload as defined by OCF for the RT
2 Calculate HMAC-SHA256 signature using "signingSecret"
POST https://eventsurl
3 Payload as defined by OCF and as received from Target Cloud Server
Header { "Subscription-ID":"uuid","Sequence-Number":"5",
"Event-Type":"resource_contentchanged", "Event-Signature":"…" … }
4 200 OK
5 Validate Event-Signature
Calculate HMAC-SHA256 signature and compare against Event-Signature header
6 2.05
observe = 0; Payload as defined by OCF for the RT
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server

Figure A.10 – "resource_contentchanged" event example flow

A.10 Addition of new registration

A.10.1 Summary

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.

A.10.2 Flows

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

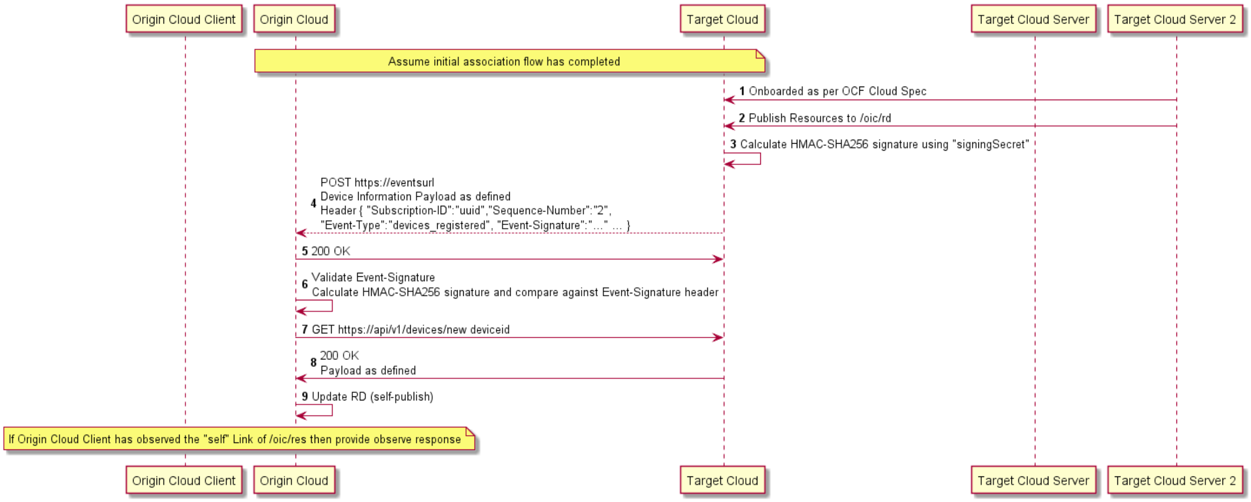
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server
Target Cloud Server 2
Assume initial association flow has completed
1 Onboarded as per OCF Cloud Spec
2 Publish Resources to /oic/rd
3 Calculate HMAC-SHA256 signature using "signingSecret"
4 POST https://eventsurl
Device Information Payload as defined
Header { "Subscription-ID":"uuid","Sequence-Number":"2",
"Event-Type":"devices_registered", "Event-Signature":"..." ... }
5 200 OK
6 Validate Event-Signature
Calculate HMAC-SHA256 signature and compare against Event-Signature header
7 GET https://api/v1/devices/new deviceid
8 200 OK
Payload as defined
9 Update RD (self-publish)
If Origin Cloud Client has observed the "self" Link of /oic/res then provide observe response
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server
Target Cloud Server 2

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

A.11 Removal of existing device registration

A.11.1 Summary

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 removes a Device from Target Cloud.

A.11.2 Flows

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

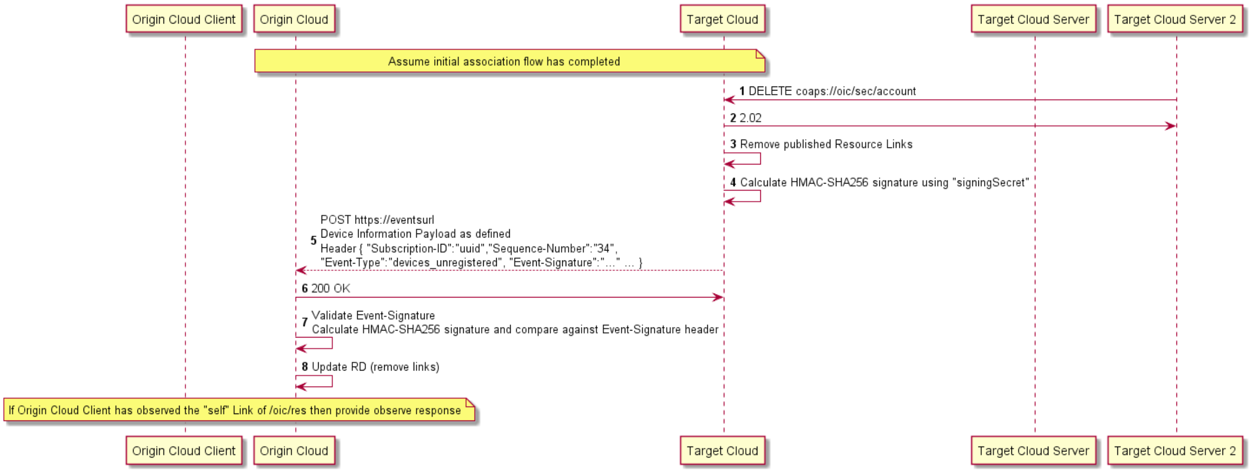
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server
Target Cloud Server 2
Assume initial association flow has completed
1 DELETE coaps://oic/sec/account
2 2.02
3 Remove published Resource Links
4 Calculate HMAC-SHA256 signature using "signingSecret"
5 POST https://eventsurl
Device Information Payload as defined
Header { "Subscription-ID":"uuid","Sequence-Number":"34",
"Event-Type":"devices_unregistered", "Event-Signature":"..." ... }
6 200 OK
7 Validate Event-Signature
Calculate HMAC-SHA256 signature and compare against Event-Signature header
8 Update RD (remove links)
If Origin Cloud Client has observed the "self" Link of /oic/res then provide observe response
Origin Cloud Client
Origin Cloud
Target Cloud
Target Cloud Server
Target Cloud Server 2

Figure A.12 – Removal of existing registration example flow

Annex B
Open API Definition

B.1 OCF Cloud API for Cloud Services

B.1.1 Supported APIs

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

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

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

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

B.1.1.3 /api/v1/devices/subscriptions

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

Supported events:
- "devices_registered"
- "devices_unregistered"
- "devices_online"
- "devices_offline"

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

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

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

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

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

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

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

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

Supported events:

- "resources_published"
- "resources_unpublished"

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

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

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

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

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

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

Supported events:
- "resource_contentchanged"

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

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

B.1.1.12 /{eventsUrl}

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

Notifications you may receive based on the event type you're subscribed to are:
- "subscription_cancelled": "SubscriptionCancelledEvent"
- "devices_registered": "DevicesRegisteredEvent"
- "devices_unregistered": "DevicesUnregisteredEvent"
- "resources_published": "ResourcesPublishedEvent"
- "resources_unpublished": "ResourcesUnpublishedEvent"
- "devices_online": "DevicesOnlineEvent"
- "devices_offline": "DevicesOfflineEvent"
- "resource_contentchanged": "ResourceContentChangedEvent"

B.1.2 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "OCF Cloud API for Cloud Services",
    "version": "0.0.3-20190828",
    "license": {
      "name": "Copyright 2019 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
```

```
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.\n"
    }
  },
  "host": "api.example.com",
  "schemes": [
    "https"
  ],
  "tags": [
    {
      "name": "Devices",
      "description": "Basic information about devices"
    },
    {
      "name": "Resources",
      "description": "Read or change the configuration of the device"
    },
    {
      "name": "Events",
      "description": "Be notified about changes occuring on the device"
    }
  ],
  "paths": {
    "/api/v1/devices?content=base": {
      "parameters": [
        {
          "$ref": "#/parameters/CorrelationId"
        },
        {
          "$ref": "#/parameters/Accept"
        },
        {
          "$ref": "#/parameters/BatchFormat"
        }
      ],
      "get": {
        "tags": [
          "Devices"
        ],
        "summary": "Get all devices with resource links",
        "description": "Get meta-information, including Resource Links, for all Devices which are
signed up to the OCF Cloud - either \"online\" or \"offline\". Devices which are \"online\" are
signed in to the system and are accessible. Offline devices are signed up to the system, but
currently disconnected.",
        "produces": [
          "application/json"
        ],
        "responses": {
          "200": {
            "description": "An array of devices",
            "schema": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/Device"
              }
            }
          },
          "400": {
            "$ref": "#/responses/BadRequest"
          },
          "401": {
            "$ref": "#/responses/Unauthorized"
          },
          "403": {
            "$ref": "#/responses/Forbidden"
          },
          "406": {
            "$ref": "#/responses/NotAcceptable"
          },
          "503": {
            "$ref": "#/responses/ServiceUnavailable"
          },
          "504": {
            "$ref": "#/responses/GatewayTimeout"
          }
        },
        "security": [
          {
            "oauth2": [
              "r:*"
            ]
          }
        ]
      }
    },
    "/api/v1/devices?content=all": {
      "parameters": [
        {
          "$ref": "#/parameters/CorrelationId"
        },
        {
          "$ref": "#/parameters/Accept"
        },
        {
          "$ref": "#/parameters/BatchFormat"
        }
      ],
      "get": {
        "tags": [
          "Devices"
        ],
        "summary": "Get all devices with resource representations",
        "description": "Get meta-information, including Resource Representations, for all Devices
which are signed up to the OCF Cloud - either \"online\" or \"offline\". Devices which are
\"online\" are signed in to the system and are accessible. Offline devices are signed up to the
system, but currently disconnected.",
        "produces": [
          "application/json"
        ],
        "responses": {
          "200": {
            "description": "An array of devices",
            "schema": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/DeviceContentAll"
              }
            }
          },
          "400": {
            "$ref": "#/responses/BadRequest"
          },
          "401": {
            "$ref": "#/responses/Unauthorized"
          },
          "403": {
            "$ref": "#/responses/Forbidden"
          },
          "406": {
            "$ref": "#/responses/NotAcceptable"
          },
          "503": {
            "$ref": "#/responses/ServiceUnavailable"
          },
          "504": {
            "$ref": "#/responses/GatewayTimeout"
          }
        },
        "security": [
          {
            "oauth2": [
              "r:*"
            ]
          }
        ]
      }
    },
    "/api/v1/devices/subscriptions": {
      "parameters": [
        {
          "$ref": "#/parameters/CorrelationId"
        },
        {
          "$ref": "#/parameters/Accept"
        }
      ],
      "post": {
        "tags": [
          "Events"
        ],
        "summary": "Subscribe to events against the set of devices",
        "description": "Subscribe to devices events by providing \"eventTypes\" you're interested in
and an \"eventsUrl\" endpoint where notifications will be sent to as defined. A successful response
contains a \"subscriptionId\" which identifies the registered subscription and is part of each
notification. First notification for each registered event type is received immediately after
subscription and contains the actual state of the resource, followed by new notifications in case of
any change.\n\nSupported events:\n- \"devices_registered\"\n- \"devices_unregistered\"\n-
\"devices_online\"\n- \"devices_offline\"",
        "parameters": [
          {
            "$ref": "#/parameters/ContentType"
          },
          {
            "$ref": "#/parameters/SubscribeRequestDevices"
          }
        ],
        "consumes": [
          "application/json"
        ],
        "produces": [
          "application/json"
        ],
        "responses": {
          "201": {
            "$ref": "#/definitions/SubscribeResponse"
          },
          "400": {
            "$ref": "#/responses/BadRequest"
          },
          "401": {
            "$ref": "#/responses/Unauthorized"
          },
          "403": {
            "$ref": "#/responses/Forbidden"
          }
        },
        "security": [
          {
            "oauth2": [
              "r:*"
            ]
          }
        ]
      }
    },
    "/api/v1/devices/subscriptions/{subscriptionId}": {
      "parameters": [
        {
          "$ref": "#/parameters/CorrelationId"
        },
        {
          "$ref": "#/parameters/SubscriptionIdPath"
        }
      ],
      "delete": {
        "tags": [
          "Events"
        ],
        "summary": "Unsubscribe from events against the set of devices",
        "description": "Cancel the subscription identified by the provided \"subscriptionId\" that
was returned in the response to the subscription request.",
        "responses": {
          "202": {
            "description": "Subscription was marked for cancellation"
          },
          "400": {
            "$ref": "#/responses/BadRequest"
          },
          "401": {
            "$ref": "#/responses/Unauthorized"
          },
          "403": {
            "$ref": "#/responses/Forbidden"
          },
          "404": {
            "$ref": "#/responses/NotFound"
          },
          "406": {
            "$ref": "#/responses/NotAcceptable"
          }
        },
        "security": [
          {
            "oauth2": [
              "r:*"
            ]
          }
        ]
      }
    },
    "/api/v1/devices/{deviceId}?content=base": {
      "parameters": [
        {
          "$ref": "#/parameters/CorrelationId"
        },
        {
          "$ref": "#/parameters/Accept"
        },
        {
          "$ref": "#/parameters/DeviceId"
        },
        {
          "$ref": "#/parameters/BatchFormat"
        }
      ],
      "get": {
        "tags": [
          "Devices"
        ],
        "summary": "Get the device with resource links by ID",
        "description": "Get the meta-information for the Device given by the provided \"deviceId\"
including Resource Links.",
        "consumes": [
          "application/json"
        ],
        "produces": [
          "application/json"
        ],
        "responses": {
          "200": {
            "description": "Device requested with content=base query parameter",
            "schema": {
              "$ref": "#/definitions/Device"
            }
          },
          "400": {
            "$ref": "#/responses/BadRequest"
          },
          "401": {
            "$ref": "#/responses/Unauthorized"
          },
          "403": {
            "$ref": "#/responses/Forbidden"
          },
          "404": {
            "$ref": "#/responses/NotFound"
          },
          "406": {
            "$ref": "#/responses/NotAcceptable"
          },
          "503": {
            "$ref": "#/responses/ServiceUnavailable"
          },
          "504": {
            "$ref": "#/responses/GatewayTimeout"
          }
        },
        "security": [
          {
            "oauth2": [
              "r:*"
            ]
          }
        ]
      }
    },
    "/api/v1/devices/{deviceId}?content=all": {
      "parameters": [
        {
          "$ref": "#/parameters/CorrelationId"
        },
        {
          "$ref": "#/parameters/Accept"
        },
        {
          "$ref": "#/parameters/DeviceId"
        },
        {
          "$ref": "#/parameters/BatchFormat"
        }
      ],
      "get": {
        "tags": [
          "Devices"
        ],
        "summary": "Get the device with resource representations by ID",
        "description": "Get the meta-information for the Device given by the provided \"deviceId\"
including Resource Representations.",
        "consumes": [
          "application/json"
        ],
        "produces": [
          "application/json"
        ],
        "responses": {
```

```json
          "200": {
            "description": "Device requested with content=all query parameter",
            "schema": {
              "$ref": "#/definitions/DeviceContentAll"
            }
          },
          "400": {
            "$ref": "#/responses/BadRequest"
          },
          "401": {
            "$ref": "#/responses/Unauthorized"
          },
          "403": {
            "$ref": "#/responses/Forbidden"
          },
          "404": {
            "$ref": "#/responses/NotFound"
          },
          "406": {
            "$ref": "#/responses/NotAcceptable"
          },
          "503": {
            "$ref": "#/responses/ServiceUnavailable"
          },
          "504": {
            "$ref": "#/responses/GatewayTimeout"
          }
        },
        "security": [
          {
            "oauth2": [
              "r:*"
            ]
          }
        ]
      }
    },
    "/api/v1/devices/{deviceId}/subscriptions": {
      "parameters": [
        {
          "$ref": "#/parameters/CorrelationId"
        },
        {
          "$ref": "#/parameters/DeviceId"
        },
        {
          "$ref": "#/parameters/Accept"
        }
      ],
      "post": {
        "tags": [
          "Events"
        ],
        "summary": "Subscribe to events against a specific device",
        "description": "Subscribe to Device level events by providing \"eventTypes\" you're
interested in and an \"eventsUrl\" API Endpoint where notifications will be sent to as defined. A
successful response contains a \"subscriptionId\" which identifies the registered subscription and
is part of each notification. First notification for each registered event type is received
immediately after subscription and contains the actual state of the resource, followed by new
notifications in case of any change.\n\nSupported events:\n- \"resources_published\"\n-
\"resources_unpublished\"",
        "parameters": [
          {
            "$ref": "#/parameters/ContentType"
          },
          {
            "$ref": "#/parameters/SubscribeRequestDevice"
          }
        ],
        "consumes": [
          "application/json"
        ],
        "produces": [
          "application/json"
        ],
        "responses": {
          "201": {
            "$ref": "#/definitions/SubscribeResponse"
          },
          "400": {
            "$ref": "#/responses/BadRequest"
          },
          "401": {
            "$ref": "#/responses/Unauthorized"
          },
          "403": {
            "$ref": "#/responses/Forbidden"
          },
          "404": {
            "$ref": "#/responses/NotFound"
          },
          "406": {
            "$ref": "#/responses/NotAcceptable"
          }
        },
        "security": [
          {
            "oauth2": [
              "r:*"
            ]
          }
        ]
      }
    },
    "/api/v1/devices/{deviceId}/subscriptions/{subscriptionId}": {
      "parameters": [
        {
          "$ref": "#/parameters/CorrelationId"
        },
        {
          "$ref": "#/parameters/DeviceId"
        },
        {
          "$ref": "#/parameters/SubscriptionIdPath"
        }
      ],
      "delete": {
        "tags": [
          "Events"
        ],
        "summary": "Unsubscribe from events against a specific device",
        "description": "Cancel the subscription identified by the provided \"subscriptionId\" that
was returned in the response to the subscription request.",
        "responses": {
          "202": {
            "description": "Subscription was marked for cancellation"
          },
          "400": {
            "$ref": "#/responses/BadRequest"
          },
          "401": {
            "$ref": "#/responses/Unauthorized"
          },
          "403": {
            "$ref": "#/responses/Forbidden"
          },
          "404": {
            "$ref": "#/responses/NotFound"
          }
        },
```

```json
        "security": [
          {
            "oauth2": [
              "r:*"
            ]
          }
        ]
      }
    },
    "/api/v1/devices/{deviceId}/{resourceLinkHref}": {
      "parameters": [
        {
          "$ref": "#/parameters/CorrelationId"
        },
        {
          "$ref": "#/parameters/DeviceId"
        },
        {
          "$ref": "#/parameters/ResourceLinkHref"
        },
        {
          "$ref": "#/parameters/Accept"
        }
      ],
      "get": {
        "tags": [
          "Resources"
        ],
        "summary": "Retrieve resource values",
        "description": "Get or update the Resource Representation of the Resource found at
\"resourceLinkHref\" on the Device with the given \"deviceId\"",
        "consumes": [
          "application/json",
          "application/vnd.ocf+cbor"
        ],
        "produces": [
          "application/json",
          "application/vnd.ocf+cbor"
        ],
        "responses": {
          "200": {
            "$ref": "#/definitions/ResourceRetrieveResponse"
          },
          "400": {
            "$ref": "#/responses/BadRequest"
          },
          "401": {
            "$ref": "#/responses/Unauthorized"
          },
          "403": {
            "$ref": "#/responses/Forbidden"
          },
          "404": {
            "$ref": "#/responses/NotFound"
          },
          "406": {
            "$ref": "#/responses/NotAcceptable"
          },
          "503": {
            "$ref": "#/responses/ServiceUnavailable"
          },
          "504": {
            "$ref": "#/responses/GatewayTimeout"
          }
        },
        "security": [
          {
            "oauth2": [
              "r:*"
            ]
          }
        ]
      },
      "post": {
        "tags": [
          "Resources"
        ],
        "summary": "Update resource values",
        "parameters": [
          {
            "$ref": "#/parameters/ResourceUpdateRequest"
          },
          {
            "$ref": "#/parameters/ContentType"
          }
        ],
        "consumes": [
          "application/json",
          "application/vnd.ocf+cbor"
        ],
        "produces": [
          "application/json",
          "application/vnd.ocf+cbor"
        ],
        "responses": {
          "200": {
            "$ref": "#/definitions/ResourceRetrieveResponse"
          },
          "400": {
            "$ref": "#/responses/BadRequest"
          },
          "401": {
            "$ref": "#/responses/Unauthorized"
          },
          "403": {
            "$ref": "#/responses/Forbidden"
          },
          "404": {
            "$ref": "#/responses/NotFound"
          },
          "415": {
            "$ref": "#/responses/UnsupportedMediaType"
          },
          "503": {
            "$ref": "#/responses/ServiceUnavailable"
          },
          "504": {
            "$ref": "#/responses/GatewayTimeout"
          }
        },
        "security": [
          {
            "oauth2": [
              "r:*",
              "w:*"
            ]
          }
        ]
      }
    },
    "/api/v1/devices/{deviceId}/{resourceLinkHref}/subscriptions": {
      "parameters": [
        {
          "$ref": "#/parameters/CorrelationId"
        },
        {
          "$ref": "#/parameters/DeviceId"
        },
        {
          "$ref": "#/parameters/ResourceLinkHref"
        },
        {
          "$ref": "#/parameters/Accept"
        }
      ],
      "post": {
        "tags": [
          "Events"
        ],
        "summary": "Subscribe to events against a specific resource",
        "description": "Subscribe to Resource level events by providing \"eventTypes\" you're
interested in and \"eventsUrl\" API Endpoint where notifications will be sent to as defined. A
successful response contains a \"subscriptionId\" which identifies the registered subscription and
is part of each event. First notification for each registered event type is received immediately
after subscription and contains the actual state of the resource, followed by new notifications in
case of any change.\n \nSupported events:\n- \"resource_contentchanged\"",
        "parameters": [
          {
            "$ref": "#/parameters/ContentType"
          },
          {
            "$ref": "#/parameters/SubscribeRequestResources"
          }
        ],
        "consumes": [
          "application/json"
        ],
        "produces": [
          "application/json"
        ],
        "responses": {
          "201": {
            "$ref": "#/definitions/SubscribeResponse"
          },
          "400": {
            "$ref": "#/responses/BadRequest"
          },
          "401": {
            "$ref": "#/responses/Unauthorized"
          },
          "403": {
            "$ref": "#/responses/Forbidden"
          },
          "404": {
            "$ref": "#/responses/NotFound"
          },
          "406": {
            "$ref": "#/responses/NotAcceptable"
          }
        },
        "security": [
          {
            "oauth2": [
              "r:*"
            ]
          }
        ]
      }
    },
    "/api/v1/devices/{deviceId}/{resourceLinkHref}/subscriptions/{subscriptionId}": {
      "parameters": [
        {
          "$ref": "#/parameters/CorrelationId"
        },
        {
          "$ref": "#/parameters/DeviceId"
        },
        {
          "$ref": "#/parameters/ResourceLinkHref"
        },
        {
          "$ref": "#/parameters/SubscriptionIdPath"
        }
      ],
      "delete": {
        "tags": [
          "Events"
        ],
        "summary": "Unsubscribe from events against a specific resource",
        "description": "Cancel the subscription identified by the provided \"subscriptionId\" that
was returned in the response to the subscription request.",
        "responses": {
          "202": {
            "description": "Subscription was marked for cancellation"
          },
          "400": {
            "$ref": "#/responses/BadRequest"
          },
          "401": {
            "$ref": "#/responses/Unauthorized"
          },
          "403": {
            "$ref": "#/responses/Forbidden"
          },
          "404": {
            "$ref": "#/responses/NotFound"
          }
        },
        "security": [
          {
            "oauth2": [
              "r:*"
            ]
          }
        ]
      }
    },
    "/{eventsUrl}": {
      "post": {
        "tags": [
          "Events"
        ],
        "summary": "Events endpoint provided by the subscriber, where events are delivered",
        "description": "Events endpoint provided during subscription where notifications for the
events specified in the subscription will be sent to  as defined per event type. Confirmation of
each notification sent to the \"eventsUrl\" endpoint is required with a \"2xx\" success
code.\n\nNotifications you may receive based on the event type you're subscribed to are:\n -
\"subscription_cancelled\": \"SubscriptionCancelledEvent\"\n - \"devices_registered\":
\"DevicesRegisteredEvent\"\n - \"devices_unregistered\": \"DevicesUnregisteredEvent\"\n -
\"resources_published\": \"ResourcesPublishedEvent\"\n - \"resources_unpublished\":
\"ResourcesUnpublishedEvent\"\n - \"devices_online\": \"DevicesOnlineEvent\"\n -
\"devices_offline\": \"DevicesOfflineEvent\"\n - \"resource_contentchanged\":
\"ResourceContentChangedEvent\"",
        "parameters": [
          {
            "$ref": "#/parameters/CorrelationId"
          },
          {
            "$ref": "#/parameters/ContentType"
          },
          {
            "$ref": "#/parameters/EventType"
          },
          {
            "$ref": "#/parameters/SubscriptionId"
          },
          {
            "$ref": "#/parameters/SequenceNumber"
          },
          {
            "$ref": "#/parameters/EventSignature"
          },
          {
            "$ref": "#/parameters/EventTimestamp"
          },
          {
            "$ref": "#/parameters/EventsUrl"
          },
          {
            "$ref": "#/parameters/Event"
          }
        ],
        "consumes": [
          "application/json",
          "application/vnd.ocf+cbor"
        ],
        "responses": {
          "200": {
            "description": "Event successfully recieved"
          },
          "400": {
            "$ref": "#/responses/BadRequest"
          },
          "410": {
            "description": "The subscription identified by the Subscription-ID header is no more in
demand and shall be cancelled"
          }
        }
      }
    }
  },
  "securityDefinitions": {
    "oauth2": {
      "type": "oauth2",
      "flow": "accessCode",
      "authorizationUrl": "https://example.com/api/oauth/dialog",
      "tokenUrl": "https://example.com/api/oauth/token",
      "scopes": {
        "r:*": "Read device data",
        "w:*": "Update content of published resource"
      }
    }
  },
  "parameters": {
    "CorrelationId": {
      "name": "Correlation-ID",
      "in": "header",
      "type": "string",
      "format": "uuid",
      "description": "A Correlation ID, also known as a Transit ID, is a unique identifier value
that is attached to requests and messages that allow reference to a particular transaction or event
chain.\n"
    },
    "ContentType": {
      "name": "Content-Type",
      "in": "header",
      "type": "string",
      "enum": [
        "application/json",
        "application/vnd.ocf+cbor"
      ],
      "required": true,
      "description": "The Content-Type header is used to indicate the media type of the resource. In
responses, a Content-Type header tells the client what the content type of the returned content
actually is. In requests, (such as POST), the client tells the server what type of data is actually
sent.\n"
    },
    "Accept": {
      "name": "Accept",
      "in": "header",
      "type": "string",
      "enum": [
        "application/json",
        "application/vnd.ocf+cbor"
      ],
      "description": "The Accept request header can be used to specify certain media types which are
acceptable for the response. Accept headers can be used to indicate that the request is specifically
limited to a small set of desired types.\n"
    },
    "SubscriptionId": {
      "name": "Subscription-ID",
      "in": "header",
      "description": "Unique id of the subscription",
      "type": "string",
      "format": "uuid",
      "required": true
    },
    "SequenceNumber": {
      "name": "Sequence-Number",
      "in": "header",
      "description": "Sequence number of the event; first event starting with number 0",
      "type": "string",
      "required": true
    },
    "EventSignature": {
      "name": "Event-Signature",
      "in": "header",
      "description": "The signature created by combining the `signingSecret` from the subscription
request, headers and the body of the request using a stanard HMAC-SHA256 keyed hash.",
      "type": "string",
      "required": true
    },
    "EventTimestamp": {
      "name": "Event-Timestamp",
      "in": "header",
      "description": "Time when the event occurred in standard Unix time format",
      "type": "string",
      "required": true
    },
    "EventType": {
      "name": "Event-Type",
      "in": "header",
      "type": "string",
      "enum": [
        "subscription_cancelled",
        "devices_registered",
        "devices_unregistered",
        "resource_contentchanged",
        "resources_published",
        "resources_unpublished",
        "devices_online",
        "devices_offline"
      ],
      "required": true
    },
    "DeviceType": {
      "description": "Filter devices by device type",
      "name": "rt",
      "in": "query",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "ResourceLinkHref": {
      "description": "Path to resource",
      "name": "resourceLinkHref",
      "in": "path",
      "type": "string",
      "required": true
    },
    "DeviceId": {
      "description": "Id of the device",
      "name": "deviceId",
      "in": "path",
      "type": "string",
      "format": "uuid",
      "required": true
    },
    "SubscriptionIdPath": {
      "name": "subscriptionId",
      "in": "path",
      "type": "string",
      "format": "uuid",
      "required": true
    },
    "BatchFormat": {
      "name": "content",
      "in": "query",
      "description": "Indicates to the recipient that the response payload shall be the resolved
(i.e. resource representation) Link and not the Link itself. Default is `base`. When requesting
`all`, additional scope `r:*` is required",
      "type": "string",
      "enum": [
        "base",
        "all"
      ]
    },
    "EventsUrl": {
      "name": "eventsUrl",
      "type": "string",
      "in": "path",
      "required": true
    },
    "ResourceUpdateRequest": {
      "description": "Map of resource values encoded to application/vnd.ocf+cbor type",
      "name": "content",
      "in": "body",
      "schema": {
        "$ref": "#/definitions/ResourceUpdateRequest"
      },
      "required": true
    },
    "SubscribeRequestDevices": {
      "name": "content",
      "in": "body",
      "schema": {
        "$ref": "#/definitions/SubscribeRequestDevices"
      },
      "required": true
    },
    "SubscribeRequestDevice": {
      "name": "content",
      "in": "body",
      "schema": {
        "$ref": "#/definitions/SubscribeRequestDevice"
      },
      "required": true
    },
    "SubscribeRequestResources": {
      "name": "content",
      "in": "body",
      "schema": {
        "$ref": "#/definitions/SubscribeRequestResources"
      },
      "required": true
    },
    "Event": {
      "description": "Event of a specific type, based on what you are subscribed to",
      "name": "content",
      "in": "body",
      "schema": {
        "$ref": "#/definitions/ResourceContentChangedEvent"
      },
      "required": true
    }
  },
  "responses": {
    "Unauthorized": {
      "description": "Unauthorized"
    },
    "NotFound": {
      "description": "Not found"
    },
    "SubscriptionCancellationPending": {
      "description": "Subscription was marked for cancellation"
    },
    "Forbidden": {
      "description": "Insufficient permissions"
    },
    "BadRequest": {
      "description": "The request was malformed or badly constructed"
    },
    "ServiceUnavailable": {
      "description": "The service on the Target Cloud is unavailable for the reason indicated in the
diagnostic payload"
    },
    "GatewayTimeout": {
      "description": "The target Device is registered at the target Cloud, however the Device itself
is unavailable, offline, or otherwise unreachable. The response should include a Retry-After header
containing the time after which the request may be re-attempted. Additional information is indicated
in the diagnostic payload."
    },
    "UnsupportedMediaType": {
      "description": "The request contained an unsupported media type in the Content-Type header"
    },
    "NotAcceptable": {
      "description": "The server cannot honour the Content-Type requested in the Accept header"
    }
  },
  "definitions": {
    "DeviceProperties": {
      "type": "object",
      "required": ["rt", "di", "dmn", "n"],
      "properties": {
        "rt":  {
          "description": "Resource Type of the Resource",
          "items": {
            "type": "string",
            "maxLength": 64
          },
          "minItems": 1,
          "readOnly": true,
          "uniqueItems": true,
          "type": "array"
        },
        "di": {
          "allOf": [
            {
              "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
schema.json#/definitions/uuid"
            },
            {
              "description": "Unique identifier for the Device",
              "readOnly": true
            }
          ]
        },
        "dmn":  {
          "description": "Manufacturer Name.",
          "items": {
            "properties": {
              "language": {
                "allOf": [
                  {
                    "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
schema.json#/definitions/language-tag"
                  },
                  {
                    "description": "An RFC 5646 language tag.",
                    "readOnly": true
                  }
                ]
              },
              "value": {
                "description": "Manufacturer name in the indicated language.",
                "maxLength": 64,
                "readOnly": true,
                "type": "string"
              }
            },
            "type": "object"
          },
          "minItems": 1,
          "readOnly": true,
          "type": "array"
        },
        "n":  {
          "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
        }
      }
    },
    "Device": {
      "type": "object",
      "required": ["device", "status", "links"],
      "properties": {
        "device": {
          "$ref":"#/definitions/DeviceProperties"
        },
        "status": {
          "$ref": "#/definitions/DeviceStatus"
        },
        "links": {
          "type": "array",
          "items": {
            "$ref":
"http://openconnectivityfoundation.github.io/core/swagger2.0/oic.wk.res.swagger.json#/definitions/oi
c.oic-link"
          }
        }
      },
      "example": {
        "device": {
          "rt": ["oic.wk.d","oic.d.sensor"],
          "dmn": [{"language":"en", "value":"Open Connectivity Foundation"}],
          "n": "Food safety sensor",
          "di": "53080a4f-5e3e-4291-802f-3436238232d2"
        },
        "status": "online",
        "links": [
          {
            "href": "/53080a4f-5e3e-4291-802f-3436238232d2/oic/d",
            "rt": [
              "oic.wk.d",
              "oic.d.sensor"
            ],
            "if": [
              "oic.if.r",
              "oic.if.baseline"
            ]
          },
          {
            "href": "/53080a4f-5e3e-4291-802f-3436238232d2/oic/p",
            "rt": [
              "oic.wk.p"
            ],
            "if": [
              "oic.if.r",
              "oic.if.baseline"
            ]
          },
          {
            "href": "/53080a4f-5e3e-4291-802f-3436238232d2/humidity",
            "rt": [
              "oic.r.humidity"
            ],
            "if": [
              "oic.if.s",
              "oic.if.baseline"
            ]
          },
          {
            "href": "/53080a4f-5e3e-4291-802f-3436238232d2/temperature",
            "rt": [
              "oic.r.temperature"
            ],
            "if": [
              "oic.if.s",
              "oic.if.baseline"
            ]
          }
        ]
      }
    },
    "DeviceContentAll": {
      "type": "object",
      "required": ["device", "status", "links"],
      "properties": {
        "device": {
          "$ref":"#/definitions/DeviceProperties"
        },
        "status": {
          "$ref": "#/definitions/DeviceStatus"
        },
        "links": {
          "type": "array",
          "items": {
            "type": "object",
            "properties": {
              "href": {
                "type": "string"
              },
              "rep": {
                "oneOf": [
                  {
                    "type": "object"
                  },
                  {
                    "type": "array"
                  }
                ]
              }
            }
          }
        }
      },
      "example": {
        "device": {
          "rt": ["oic.wk.d","oic.d.sensor"],
          "dmn": [{"language":"en", "value":"Open Connectivity Foundation"}],
          "n": "Food safety sensor",
          "di": "53080a4f-5e3e-4291-802f-3436238232d2"
        },
        "status": "online",
        "links": [
          {
            "href": "/53080a4f-5e3e-4291-802f-3436238232d2/oic/d",
            "rep": {
              "rt": ["oic.wk.d","oic.d.sensor"],
              "dmn": [{"language":"en", "value":"Open Connectivity Foundation"}],
              "n": "Food safety sensor",
              "di": "53080a4f-5e3e-4291-802f-3436238232d2",
              "icv":  "ocf.2.0.5",
              "dmv":  "ocf.res.1.3.0, ocf.sh.1.3.0",
              "piid": "6F0AAC04-2BB0-468D-B57C-16570A26AE48"
            }
          },
          {
            "href": "/53080a4f-5e3e-4291-802f-3436238232d2/oic/p",
            "rep": {
              "pi":   "54919CA5-4101-4AE4-595B-353C51AA983C",
              "mnfv": "1.1.20"
            }
          },
          {
            "href": "/53080a4f-5e3e-4291-802f-3436238232d2/humidity",
            "rep": {
              "humidity": 62,
              "desiredHumidity": 65
            }
          },
          {
            "href": "/53080a4f-5e3e-4291-802f-3436238232d2/temperature",
            "rep": {
              "temperature": 21,
              "units": "C"
            }
          }
        ]
      }
    },
    "DeviceStatus": {
      "description": "Device status available from the OCF Cloud, which tracks if the device has
opened TCP connection and is signed in",
      "type": "string",
      "enum": [
        "online",
        "offline"
      ]
    },
    "ResourceUpdateRequest": {
      "type": "string",
      "description": "Desired content of the resource",
      "example": "o29kZXNpcmVkSHVtaWRpdHkYPGV0eXBlc4Fub2ljLnIuaHVtaWRpdHloaHVtaWRpdHkYKA=="
    },
    "ResourceRetrieveResponse": {
      "type": "string",
      "description": "Content of the resource returned from the device",
      "example": "o29kZXNpcmVkSHVtaWRpdHkYPGV0eXBlc4Fub2ljLnIuaHVtaWRpdHloaHVtaWRpdHkYKA=="
    },
    "EventType": {
      "type": "string",
      "enum": [
        "subscription_cancelled",
        "devices_registered",
        "devices_unregistered",
        "resource_contentchanged",
        "resources_published",
        "resources_unpublished",
        "devices_online",
        "devices_offline"
      ]
    },
    "EventTypeDevices": {
      "type": "string",
      "enum": [
        "devices_registered",
        "devices_unregistered",
        "devices_online",
        "devices_offline"
      ]
    },
    "EventTypeDevice": {
      "type": "string",
      "enum": [
        "resources_published",
        "resources_unpublished"
      ]
    },
    "EventTypeResources": {
      "type": "string",
      "enum": [
        "resource_contentchanged"
      ]
    },
    "SubscriptionId": {
      "description": "Unique id of the subscription",
      "type": "string",
      "format": "uuid"
    },
    "SubscribeRequestDevices": {
      "type": "object",
      "properties": {
        "eventsUrl": {
          "$ref": "#/definitions/EventsUrl"
        },
        "eventTypes": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/EventTypeDevices"
          }
        },
        "signingSecret": {
          "type": "string",
          "maxLength": 32,
          "minLength": 32
        }
      },
      "required": [
        "eventsUrl",
        "eventTypes",
        "signingSecret"
      ],
      "example": {
        "eventsUrl": "https://events.example.com/",
        "eventTypes": [
          "devices_registered",
          "devices_unregistered"
        ],
        "signingSecret": "3BZ6oI9xbRJzOUvUoRb5RgaZjPqHrmql"
      }
    },
    "SubscribeRequestDevice": {
      "type": "object",
      "properties": {
        "eventsUrl": {
          "$ref": "#/definitions/EventsUrl"
        },
        "eventTypes": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/EventTypeDevice"
          }
        },
        "signingSecret": {
          "type": "string",
          "maxLength": 32,
          "minLength": 32
        }
      },
      "required": [
        "eventsUrl",
        "eventTypes",
        "signingSecret"
      ],
      "example": {
        "eventsUrl": "https://events.example.com/",
        "eventTypes": [
          "resource_published",
          "resource_unpublished"
        ],
        "signingSecret": "3BZ6oI9xbRJzOUvUoRb5RgaZjPqHrmql"
      }
    },
    "SubscribeRequestResources": {
      "type": "object",
      "properties": {
        "eventsUrl": {
          "$ref": "#/definitions/EventsUrl"
        },
        "eventTypes": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/EventTypeResources"
          }
        },
        "signingSecret": {
          "type": "string",
          "maxLength": 32,
          "minLength": 32
        }
      },
      "required": [
        "eventsUrl",
        "eventTypes",
        "signingSecret"
      ],
      "example": {
        "eventsUrl": "https://events.example.com/",
        "eventTypes": [
          "resource_contentchanged"
        ],
        "signingSecret": "3BZ6oI9xbRJzOUvUoRb5RgaZjPqHrmql"
      }
    },
    "SubscribeResponse": {
      "description": "Subscription was registered, waiting for verification",
      "type": "object",
      "properties": {
        "subscriptionId": {
          "$ref": "#/definitions/SubscriptionId"
        }
      },
      "required": [
        "subscriptionId"
      ],
      "example": {
        "subscriptionId": "1eeb465c-5e8d-4305-a366-bbf035fff671"
      }
    },
    "EventsUrl": {
      "type": "string",
      "format": "url",
      "example": "https://events.example.com/"
    },
    "SubscriptionCancelledEvent": {
      "type": "object",
      "description": "Subscription with provided id was cancelled"
    },
    "DevicesRegisteredEvent": {
      "description": "Device was successfully signed up to the OCF Cloud, as defined in the
`oic.sec.account`",
      "type": "object",
      "properties": {
        "content": {
          "type": "array",
          "items": {
            "properties": {
              "di": {
                "type": "string",
                "format": "uuid"
              }
            }
          }
        }
      }
    },
    "DevicesUnregisteredEvent": {
      "description": "Device was successfully signed off from the OCF Cloud, as defined in the
`oic.sec.account`",
      "type": "object",
      "properties": {
        "content": {
          "type": "array",
          "items": {
            "properties": {
              "di": {
                "type": "string",
                "format": "uuid"
              }
            }
          }
        }
      }
    },
    "ResourcesPublishedEvent": {
      "type": "object",
      "properties": {
        "content": {
          "type": "array",
          "items": {
            "$ref":
"http://openconnectivityfoundation.github.io/core/swagger2.0/oic.wk.res.swagger.json#/definitions/oi
c.oic-link"
          }
        }
      }
    },
    "ResourcesUnpublishedEvent": {
      "type": "object",
      "properties": {
        "content": {
          "type": "array",
          "items": {
            "$ref":
"http://openconnectivityfoundation.github.io/core/swagger2.0/oic.wk.res.swagger.json#/definitions/oi
c.oic-link"
          }
        }
      }
    },
    "DevicesOnlineEvent": {
      "type": "object",
      "properties": {
        "content": {
          "type": "array",
          "items": {
            "properties": {
              "di": {
                "type": "string",
                "format": "uuid"
              }
            }
          }
        }
      }
    },
    "DevicesOfflineEvent": {
      "type": "object",
      "properties": {
        "content": {
          "type": "array",
          "items": {
            "properties": {
              "di": {
                "type": "string",
                "format": "uuid"
              }
            }
          }
        }
      }
    },
    "ResourceContentChangedEvent": {
      "type": "string",
      "description": "New Content of the resource returned from the device",
      "example": "o29kZXNpcmVkSHVtaWRpdHkYPGV0eXBlc4Fub2ljLnIuaHVtaWRpdHloaHVtaWRpdHkYKA=="
    }
  }
}
```