# OCF Cloud Specification

**VERSION 2.0.2 | April 2019**


**CONTACT** admin@openconnectivity.org

# Legal Disclaimer

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

# CONTENTS

# Figures

# Tables

# 1 Scope

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

# 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.

ISO/IEC 30118-1:2018 *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 30118-2:2018 *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 Wi-Fi Easy Setup, *Open Connectivity Foundation Wi-Fi Easy Setup, Version 2.0.1*
Available at: https://openconnectivity.org/specs/OCF_Wi-Fi_Easy_Setup_Specification_v2.0.1.pdf
Latest version available at:
https://openconnectivity.org/specs/OCF_Wi-Fi_Easy_Setup_Specification.pdf

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://tools.ietf.org/html/rfc6750

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

OpenAPI specification, *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 30118-1:2018 and ISO/IEC 30118-2:2018 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
**Cloud Provider**
entity or organization that hosts an OCF Cloud (3.1.2).

### 3.1.2
**OCF Cloud**
an OCF Cloud is not an OCF Device, but a logical entity that is owned by the Cloud Provider (3.1.1). An OCF Cloud is authorised to communicate with a Device on behalf of the OCF Cloud User.

## 3.2 Abbreviated terms

### 3.2.1
**UX**
User Experience

 

# 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.

## 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

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

– CoAP over TCP protocol defined in ISO/IEC 30118-1:2018

– Resource Directory defined in ISO/IEC 30118-1:2018

– The requirements within this document

– Security requirements and SVRs defined within the ISO/IEC 30118-2:2018

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

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

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

## 5.2 Interaction Flow

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

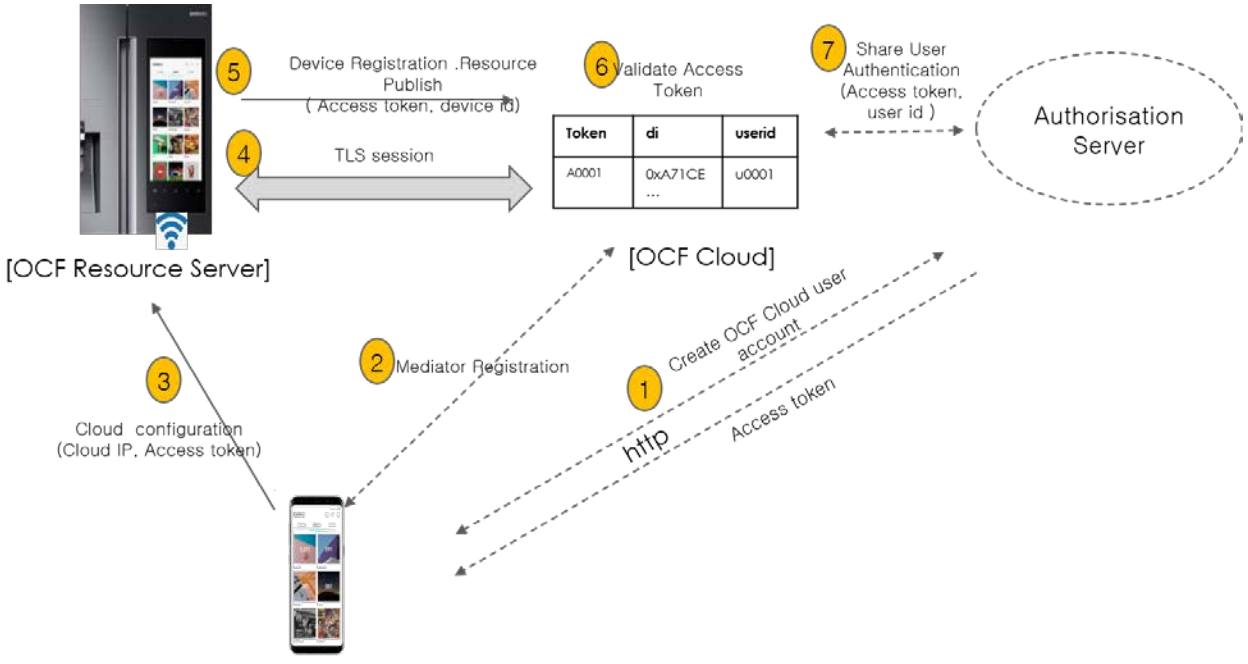


**Figure 1 – OCF Cloud deployment architecture**

 

212

213 **Table 1 – OCF Cloud Deployment Flow**

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

214

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

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

219 – an account server function which is a logical entity that handles Device registration, Access
220 Token validation and handles sign-in and token-refresh requests from the Device.

221 – a Resource Directory as defined by the ISO/IEC 30118-1:2018. The Resource Directory
222 exposes Resource information published by Devices. A Client, when discovering Devices,
223 receives a response from the Resource Directory on behalf of the Device. With information
224 included in the response from the Resource Directory, the Client may connect to the Device via
225 the OCF Cloud.

226 ### 5.3 Cloud Operational Flow

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

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

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

 

### 5.3.1 Pre-requisites and OCF Cloud User Account Creation

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

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

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

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

### 5.3.2 Mediator registration with the OCF Cloud

See 8.1.2.2, 8.1.2.3.

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

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

### 5.3.3 Device provisioning by the Mediator

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

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

### 5.3.4 Device Registration with the OCF Cloud.

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

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

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

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

 

The Device is now registered with the OCF Cloud.

### 5.3.5 Connection with the OCF Cloud

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

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

### 5.3.6 Publishing Links to the OCF Cloud RD

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

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

### 5.3.7 Client to Server communication through the OCF Cloud

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

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

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

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

### 5.3.8 Refreshing connection with the OCF Cloud

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

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

### 5.3.9 Closing connection with the OCF Cloud

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

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

### 5.3.10 Deregistering from the OCF Cloud

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

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

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

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

 

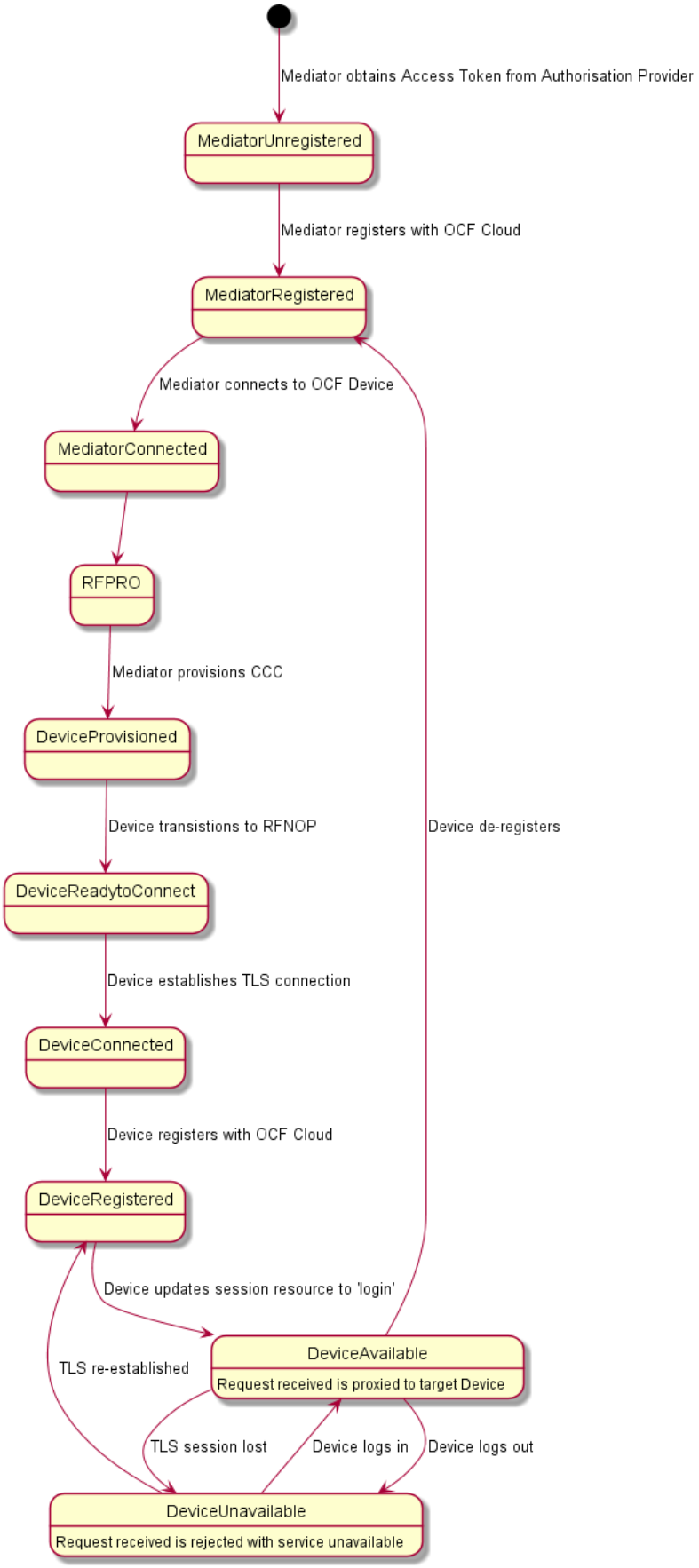
Mediator obtains Access Token from Authorisation Provider
MediatorUnregistered
Mediator registers with OCF Cloud
MediatorRegistered
Mediator connects to OCF Device
MediatorConnected
RFPRO
Mediator provisions CCC
DeviceProvisioned
Device transistions to RFNOP
Device de-registers
DeviceReadytoConnect
Device establishes TLS connection
DeviceConnected
Device registers with OCF Cloud
DeviceRegistered
Device updates session resource to 'login'
TLS re-established
DeviceAvailable
Request received is proxied to target Device
TLS session lost
Device logs in
Device logs out
DeviceUnavailable
Request received is rejected with service unavailable

**Figure 2 – Overall Operational State Machine**

# 6 Resource model

## 6.1 CoAPCloudConf Resource

### 6.1.1 Introduction

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

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

### 6.1.2 Resource Definition

The CoAPCloudConf Resource is as defined in Table 2.

**Table 2 – CoAPCloudConf Resource**

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

 

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

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

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

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

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

### 6.1.3 Error Handling

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

– The Device shall set the "clec" Property to 1 if it receives an error response from the OCF Cloud (e.g. error response from the Cloud).

– The Device shall set the "clec" Property to 2 if there is a failure to connect to the OCF Cloud (e.g. no reply, timeout, or timeout).

– The Device shall set the "clec" Property to 3 if it fails to refresh the Access Token (e.g. if it receives an error response during the token refresh procedure).

 

# 7 Network and connectivity

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

 

# 8 Functional interactions

## 8.1 Onboarding, Provisioning, and Configuration

### 8.1.1 Overview

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

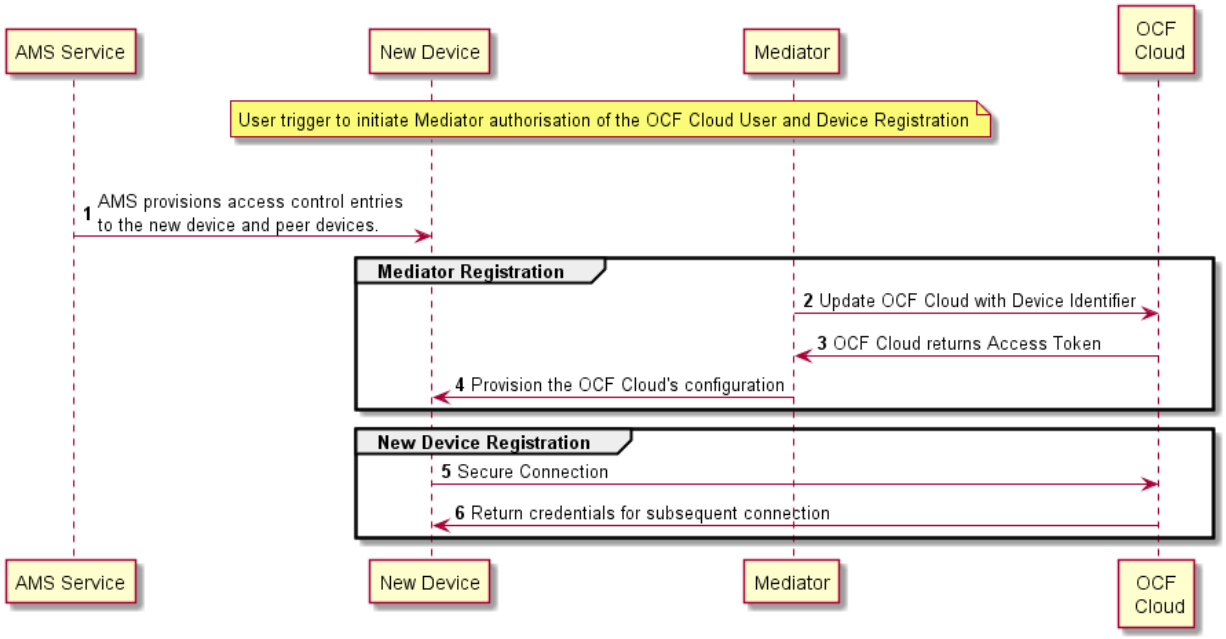


**Figure 3 – Registration with OCF Cloud**

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

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

### 8.1.2 Use of Mediator

#### 8.1.2.1 Introduction

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

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

 

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

– OCF Cloud Interface URL ("cis") Property

– OCF Cloud UUID ("sid") Property (to verify Cloud identity)

– Access Token ("at") Property that is validated by the OCF Cloud

– Optionally the Authorisation Provider name ("apn") Property through which the Access Token was obtained

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

#### 8.1.2.2 OCF Cloud User Authorisation of the Mediator

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

A user authorisation mechanism is used to achieve the following:

– Obtain an Access Token that is validated by the Cloud

– OCF Cloud User authorisation via an Authorisation Provider; this provides consent to connect to the OCF Cloud.

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

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

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

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

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

 

#### 8.1.2.3 Device Provisioning by the Mediator

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

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

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

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

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

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

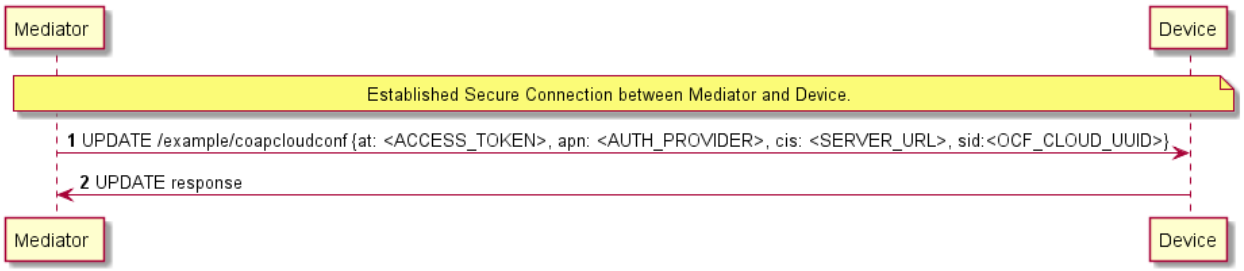


Figure 4 describes the flow for provisioning of the Device by a Mediator. Table 5 provides additional context around the flow.

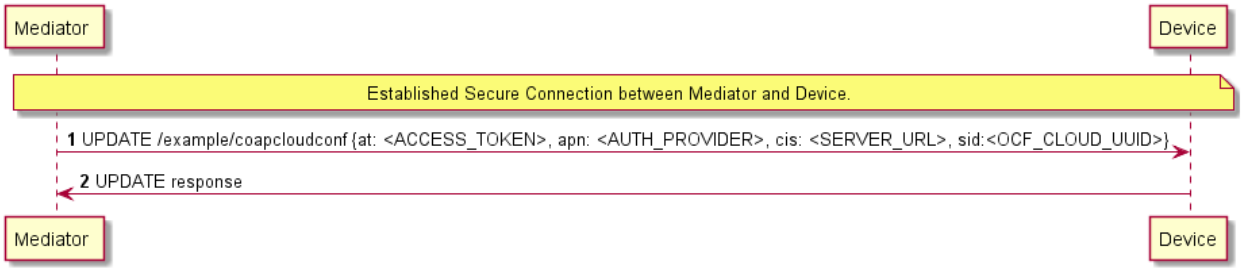


**Figure 4 – Device Provisioning by the Mediator**

 

**Table 5 – Device Provisioning by the Mediator**

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

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

### 8.1.3 Device Connection to the OCF Cloud

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

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

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

### 8.1.4 Device Registration with the OCF Cloud

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

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

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

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

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

 

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

## 8.2 Resource Publication

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

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

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

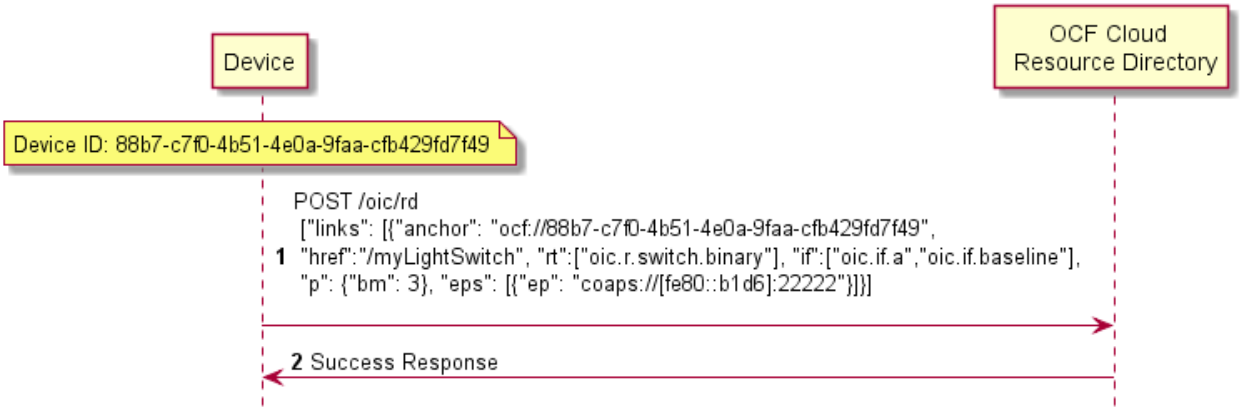


Figure 5 provides an example showing the provided Device ID from the Device; Figure 6 shows the pre-pending of the Device ID to the "href" Link Parameter in the Resource Directory itself.

 

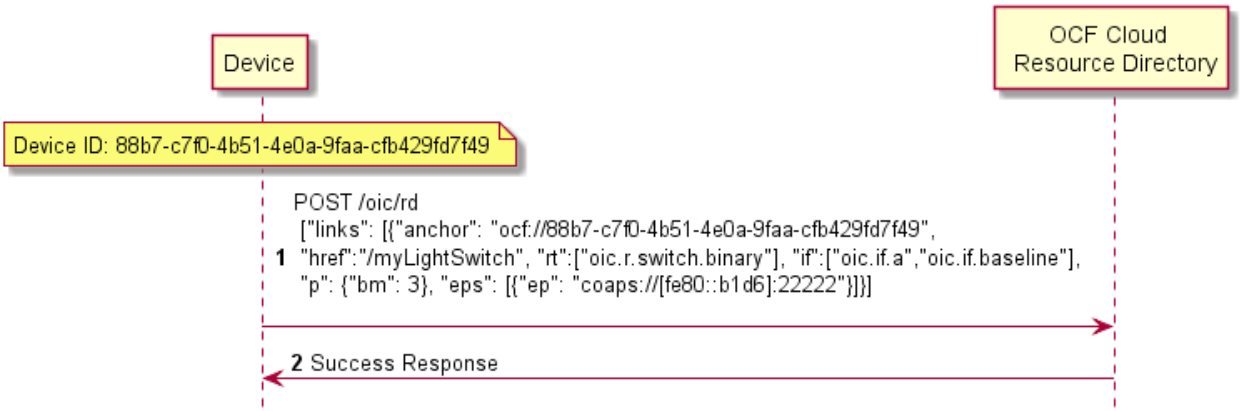


**Figure 5 – Resource publication to the OCF Cloud**

## 8.3 Client Registration with the OCF Cloud

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

## 8.4 Resource Discovery

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

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

 

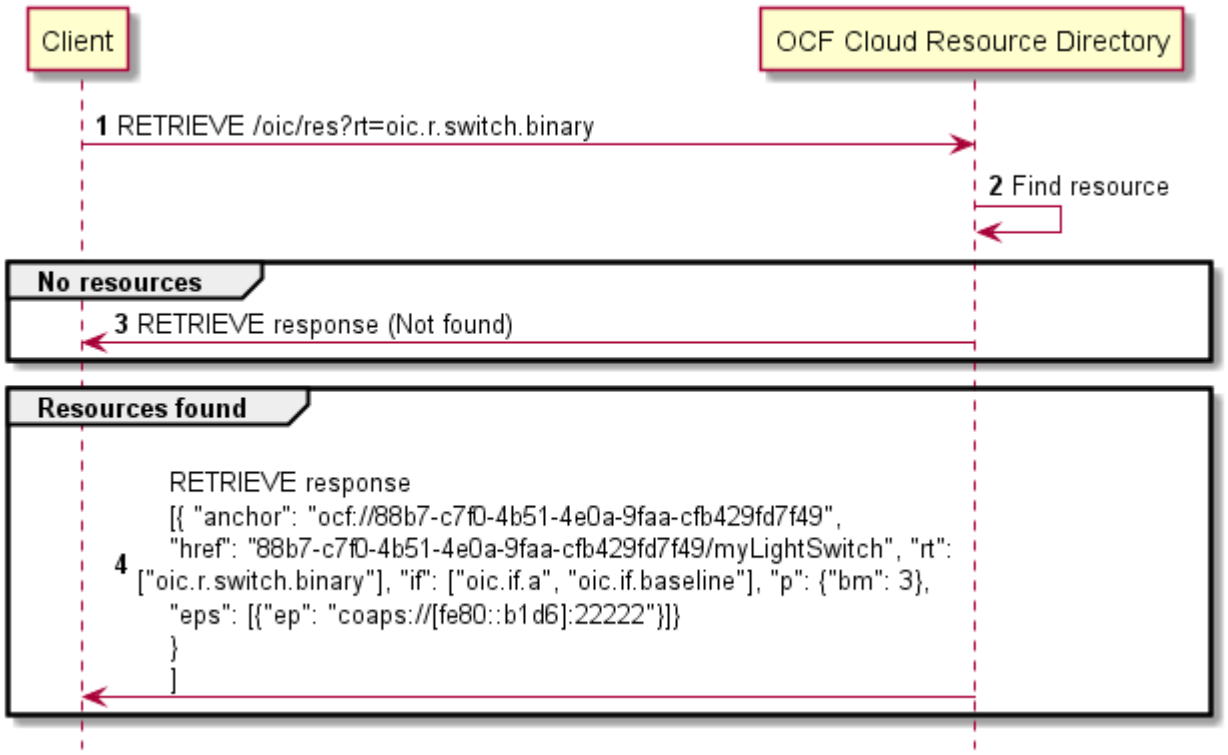


**Figure 6 – Resource discovery through OCF Cloud**

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

Figure 7 illustrates request routing via the OCF Cloud

 

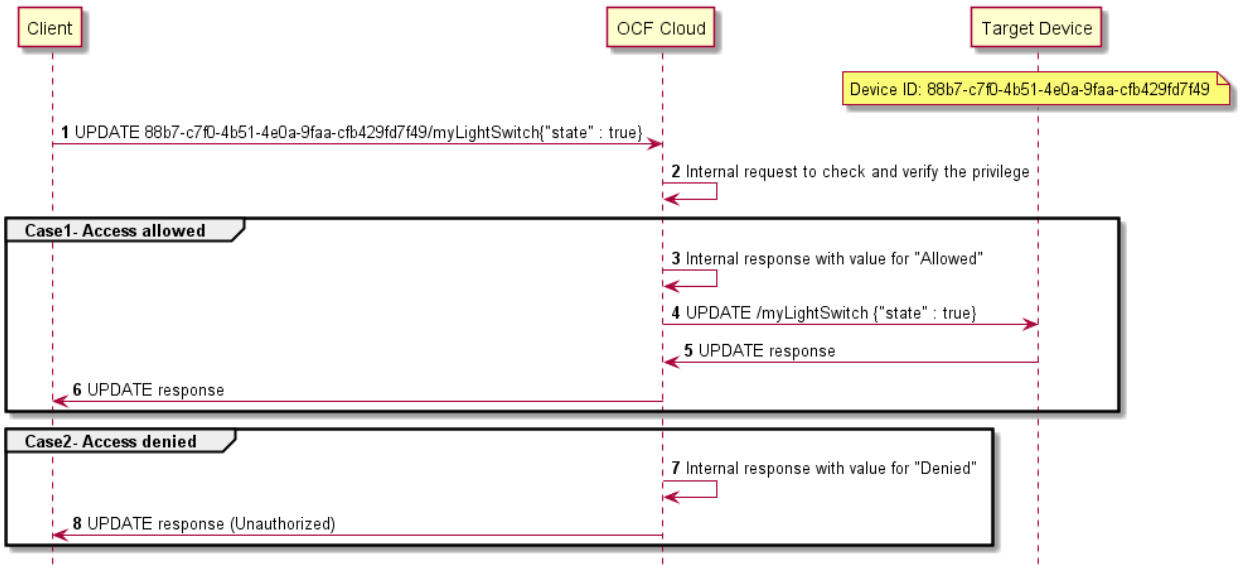


**Figure 7 – Request routing through OCF Cloud**

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

### 8.5 Device Deregistration from the OCF Cloud

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

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

## 9 Security

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

 

# Annex A (normative)

# Swagger2.0 definitions

## A.1 List of Resource Type definitions

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

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

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

## A.2 CoAP Cloud Configuration Resource

### A.2.1 Introduction

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

### A.2.2 Example URI

/CoAPCloudConfResURI

### A.2.3 Resource type

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

### A.2.4 OpenAPI 2.0 definition

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

```
                  },
                "schema": { "$ref": "#/definitions/CoAPCloudConf" }
              }
          }
        },
        "post": {
          "description": "Update properties of the CoAPCloudConf Resource.\n",
          "parameters": [
            {"$ref": "#/parameters/interface-all"},
            {
              "name": "body",
              "in": "body",
              "required": true,
              "schema": { "$ref": "#/definitions/CoAPCloudConfUpdate" },
              "x-example":
                {
                  "at": "0f3d9f7fe5491d54077d",
                  "apn": "github",
                  "cis": "coaps+tcp://example.com:443",
                  "sid" : "987e6543-a21f-10d1-a112-421345746237"
                }
            }
          ],
          "responses": {
              "200": {
                "description" : "",
                "x-example":
                  {
                    "apn": "github",
                    "cis": "coaps+tcp://example.com:443",
                    "sid" : "987e6543-a21f-10d1-a112-421345746237",
                    "clec": 0
                  },
                "schema": { "$ref": "#/definitions/CoAPCloudConf" }
              }
          }
        }
      },
      "/CoAPCloudConfResURI?if=oic.if.baseline" : {
        "get": {
          "description": "The CoAPCloudConf Resource exposes configuration information for connecting
to an OCF Cloud.\n",
          "parameters": [
            {"$ref": "#/parameters/interface-all"}
          ],
          "responses": {
              "200": {
                "description" : "",
                "x-example":
                  {
                    "rt": ["oic.r.coapcloudconf"],
                    "if" : ["oic.if.rw","oic.if.baseline"],
                    "apn": "github",
                    "cis": "coaps+tcp://example.com:443",
                    "sid" : "987e6543-a21f-10d1-a112-421345746237",
                    "clec": 0
                  },
                "schema": { "$ref": "#/definitions/CoAPCloudConf" }
              }
          }
        },
        "post": {
          "description": "Update Properties of the CoAPCloudConf Resource.\n",
          "parameters": [
            {"$ref": "#/parameters/interface-all"},
            {
              "name": "body",
              "in": "body",
              "required": true,
              "schema": { "$ref": "#/definitions/CoAPCloudConfUpdate" },
              "x-example":
```

 

```
                {
                  "at": "0f3d9f7fe5491d54077d",
                  "apn": "github",
                  "cis": "coaps+tcp://example.com:443",
                  "sid" : "987e6543-a21f-10d1-a112-421345746237"
                }
            }
          ],
          "responses": {
              "200": {
                "description" : "",
                "x-example":
                  {
                    "apn": "github",
                    "cis": "coaps+tcp://example.com:443",
                    "sid" : "987e6543-a21f-10d1-a112-421345746237",
                    "clec": 0
                  },
                "schema": { "$ref": "#/definitions/CoAPCloudConf" }
              }
          }
        }
      },
      "parameters": {
        "interface-all" : {
          "in" : "query",
          "name" : "if",
          "type" : "string",
          "enum" : ["oic.if.rw","oic.if.baseline"]
        }
      },
      "definitions": {
        "CoAPCloudConf" : {
          "properties": {
            "rt" : {
              "description": "Resource Type of the Resource",
              "items": {
                "enum": ["oic.r.coapcloudconf"],
                "type": "string",
                "maxLength": 64
              },
              "minItems": 1,
              "uniqueItems": true,
              "readOnly": true,
              "type": "array"
            },
            "n" : {
              "$ref":
    "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
    schema.json#/definitions/n"
            },
            "cis" : {
              "description": "URL of OCF Cloud",
              "format": "uri",
              "type": "string"
            },
            "apn" : {
              "description": "The Authorisation Provider through which an Access Token was obtained.",
              "type": "string"
            },
            "sid" : {
              "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
    schema.json#/definitions/uuid"
            },
            "clec" : {
              "description": "Last Error Code during Cloud Provisioning (0: No Error, 1: Error response
    from the OCF Cloud, 2: Failed to connect to the OCF Cloud, 3: Failed to refresh Access Token, 4~254:
    Reserved, 255: Unknown error)",
              "enum": [
                0,
```

 

```
              1,
              2,
              3,
              255
            ],
            "readOnly": true
          },
          "id" : {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
          },
          "if" : {
            "description": "The OCF Interfaces supported by this Resource",
            "items": {
              "enum": [
                "oic.if.rw",
                "oic.if.baseline"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 2,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          }
        },
        "type" : "object",
        "required":["cis", "sid"]
      },
      "CoAPCloudConfUpdate" : {
        "properties": {
          "cis" : {
            "description": "URL of OCF Cloud",
            "format": "uri",
            "type": "string"
          },
          "apn" : {
            "description": "The Authorisation Provider through which an Access Token was obtained.",
            "type": "string"
          },
          "at" : {
            "description": "Access Token which is returned by an Authorisation Provider or OCF
Cloud.",
            "type": "string"
          },
          "sid" : {
            "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
schema.json#/definitions/uuid"
          }
        },
        "type" : "object",
        "required":["cis", "at", "sid"]
      }
    }
}
```

### A.2.5 Property definition

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

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

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

 

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

### A.2.6 CRUDN behaviour

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

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

| Create | Read | Update | Delete | Notify |
|---|---|---|---|---|
| | get | post | | observe |

 