OCF Device to Cloud Services Specification

VERSION 2.1.0 | November 2019

OPEN CONNECTIVITY FOUNDATION™

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*
Avaliable 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.

3.1.3
Resource Directory
a set of descriptions of Resources where the actual Resources are held on Servers external to the entity hosting the Resource Directory (3.1.3), allowing lookups to be performed for those Resources

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
- The requirements within this document including those for a Resource Directory
- 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 Architecture

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

- an account server function which is a logical entity that handles Device registration, Access Token validation and handles sign-in and token-refresh requests from the Device. An OCF Cloud User creates offline an account on the account server (by means of the mediator). The account server is then also used to register the Devices (Clients and Servers) per account. Note that all accounts are fully separated, e.g. logging into account A does not give access to Devices registered to account B.
- a Resource Directory as defined by this document. The Resource Directory exposes Resource information published by Devices. A Client, when discovering Devices, receives a response from the Resource Directory on behalf of the Device. With information included in the response from the Resource Directory, the Client may connect to the Device via the OCF Cloud.

This is illustrated in Figure 1.

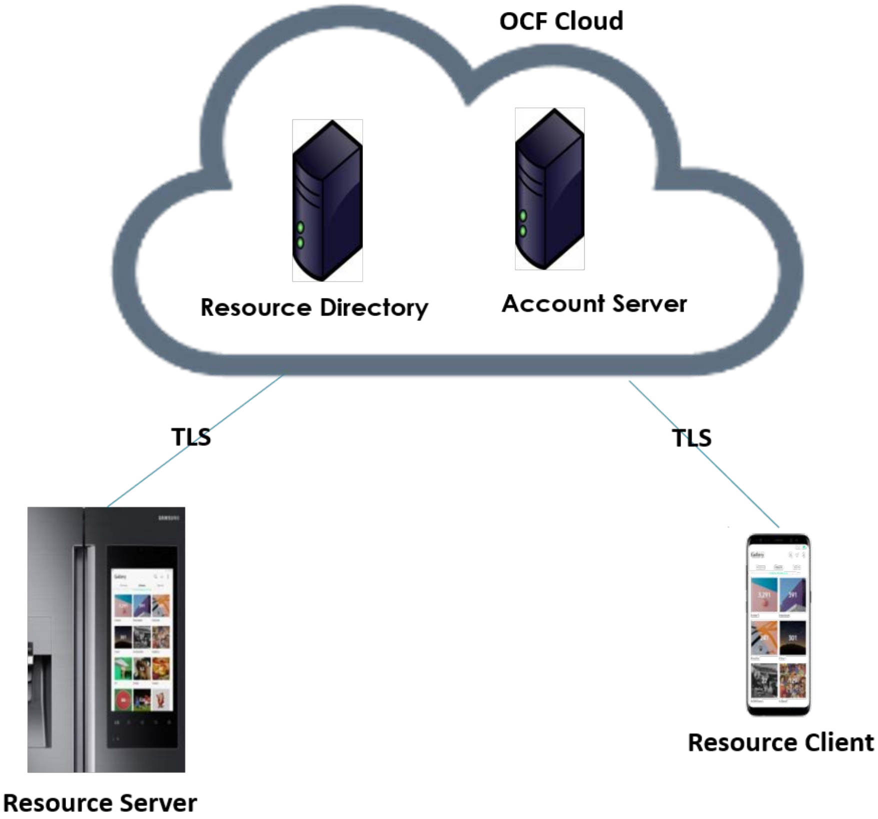

Figure 1 – OCF Cloud Architecture

5.3 Interaction Flow

This clause describes how the elements with the overall OCF Cloud interact. Figure 2 provides an overall introduction, Table 1 provides additional context to the elements in the flow.

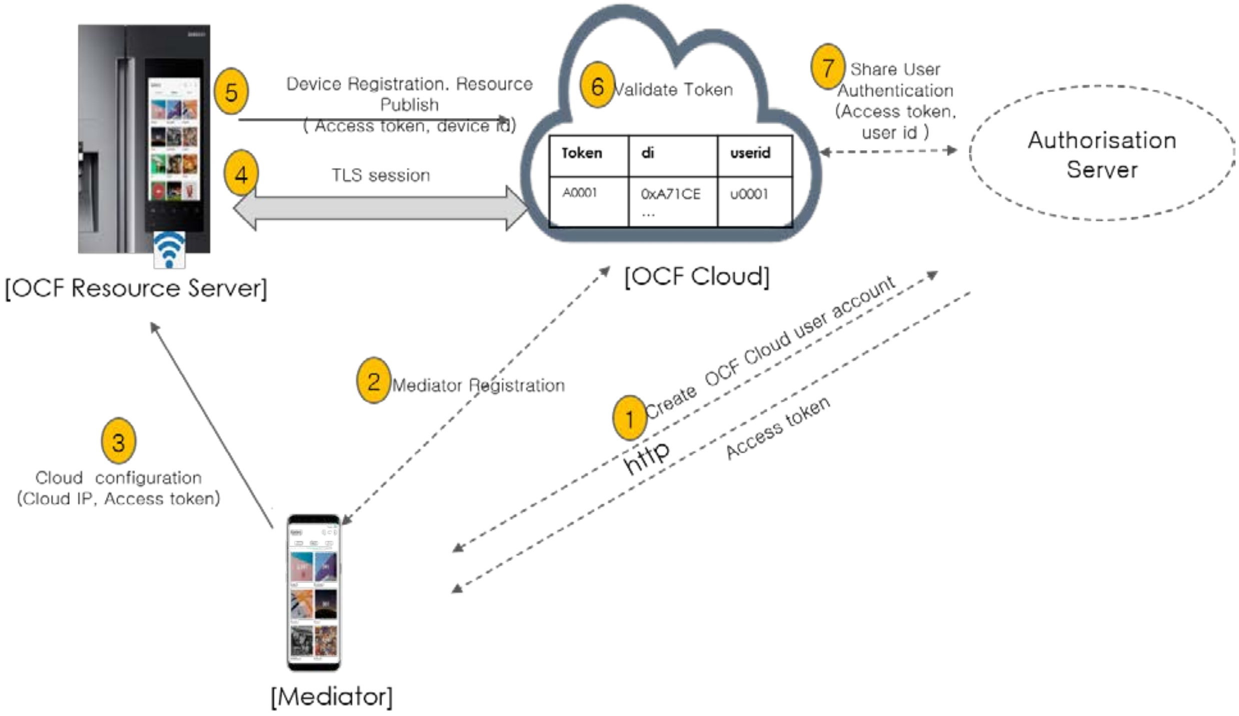

Figure 2 – OCF Cloud interaction model

Table 1 – OCF Cloud Interaction 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.

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

5.4 Cloud Operational Flow

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

The flow consists of the following high-level steps:

- Pre-requisites and OCF Cloud User account creation (see 5.4.1)
- Mediator registration with the OCF Cloud (see 5.4.2)
- Device provisioning by the Mediator (see 5.4.3)
- Device registration with the OCF Cloud (see 5.4.4)
- Device connection with the OCF Cloud (see 5.4.5)
- Devices Publishing Links to the OCF Cloud RD (see 5.4.6)
- Client to Server communication through the OCF Cloud (see 5.4.7)
- Device refreshing connection with the OCF Cloud (see 5.4.8)
- Device closing connection with the OCF Cloud (see 5.4.9)
- Device de-registering from the OCF Cloud (see 5.4.10)

5.4.1 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 makes use of a Mediator to provision the Device. A Mediator is a logical function that may be on the OCF Cloud User's personal device (e.g. phone) or elsewhere. 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)

5.4.2 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

5.4.3 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.

5.4.4 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.

5.4.5 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.

5.4.6 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.

5.4.7 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).

5.4.8 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.

5.4.9 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.

5.4.10 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 clause 5.4.3).

Figure 3 captures the state machine that is described by the informative operation flow provided in clause 5.4.

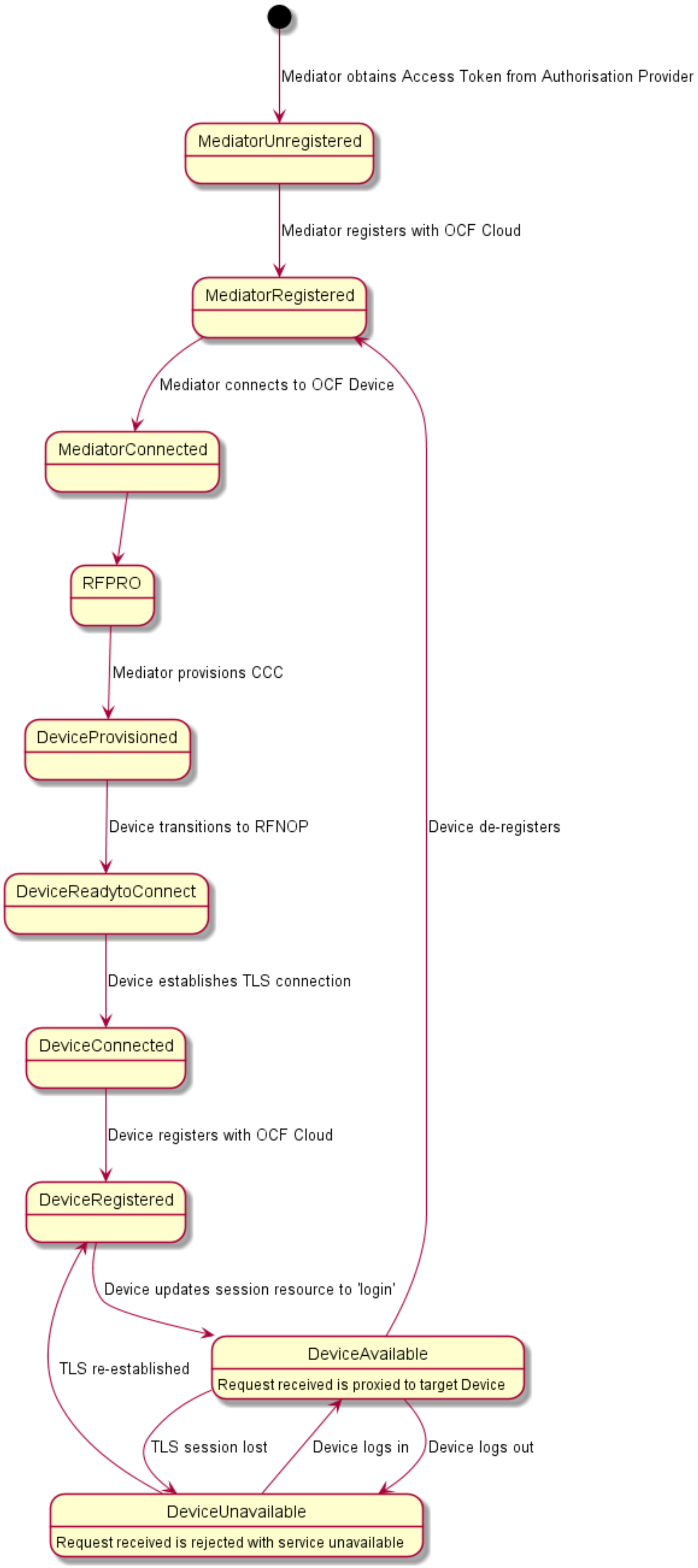
Mediator obtains Access Token from Authorisation Provider
MediatorUnregistered
Mediator registers with OCF Cloud
MediatorRegistered
Mediator connects to OCF Device
MediatorConnected
RFPRO
Mediator provisions CCC
DeviceProvisioned
Device transitions to RFNOP
Device de-registers
DeviceReadytoConnect
Device 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 3 – Overall Operational State Machine

6 Resource model

6.1 OCF Cloud Resource Directory

6.1.1 Indirect discovery for lookup of Resources

Indirect discovery is when a 3rd party, other than the discovering Device and the discovered Device, assists with the discovery process. The 3rd party, called a Resource Directory (RD), only provides information on Resources on behalf of another Device but does not host Resources on part of that Device.

In Figure 4, the OCF Cloud acts as Resource Directory for Device A and Device D which are both part of the same account. Device A and Device D publish their Resource information to the OCF Cloud. Device C which is also part of the same account as Devices A and D, may query the OCF Cloud to acquire the Resource information of Devices A and D.

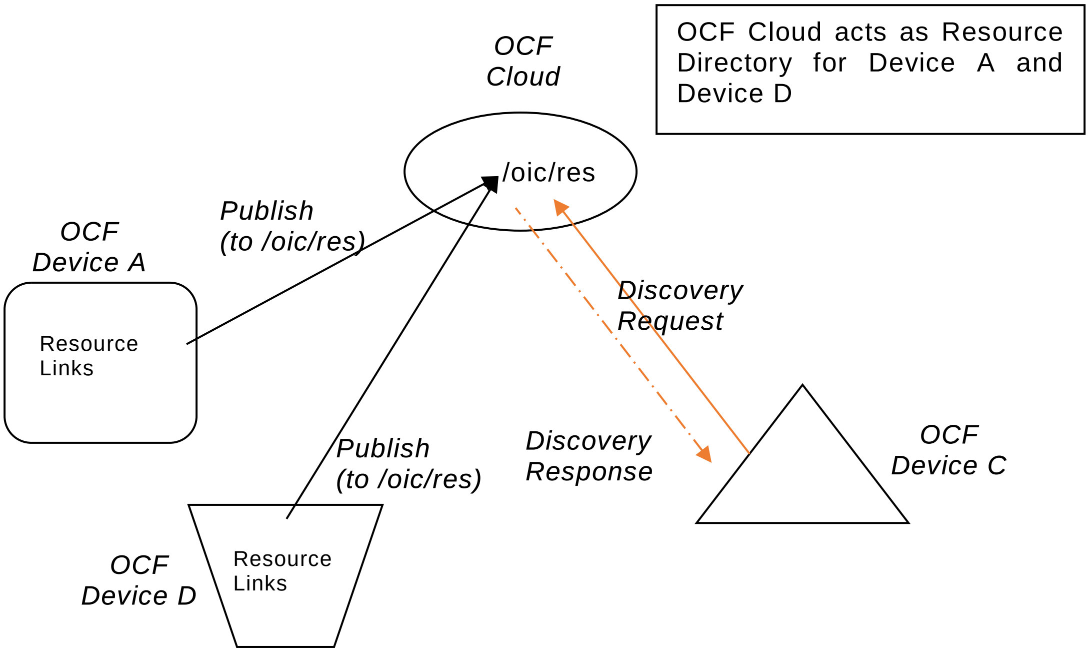

Figure 4 – Indirect discovery of Resources by via an RD

Indirect discovery is useful for when Devices may not be on the same network and require optimization for discovery or routing. Once Resources are discovered using indirect discovery, i.e., RD query, then the access to the Resource is done by a request sent to the endpoint exposed by the RD for the Resource.

6.1.2 Resource Directory Definition

An OCF Cloud which acts as a Resource Directory (RD) will be involved in the following operations.

- *RD discovery* – the procedure by which publishing Devices discover an RD, in the case of the OCF Cloud this is a direct result of Device registration with an OCF Cloud.
- *Resource publish* – the procedures with which Devices publish their Resource information, i.e. Links.

– *Resource exposure* – the feature with which RDs expose the Links hosted by the 3rd party Devices via their own "/oic/res".

An RD makes use of Resource Type "oic.wk.rd" defined in Table 2 and Table 3. An OCF Cloud that supports the capability to host indirect discovery shall expose an instance of the "oic.wk.rd" Resource Type in its "/oic/res" to announce that it serves as an RD. The use of the "oic.wk.rd" Resource Type is restricted to OCF Clouds only, a proximal network Device shall not expose the "oic.wk.rd" Resource Type.

The discoverable instance of "oic.wk.rd" shall allow only secure connections (e.g. OCF Endpoint with a scheme of "coaps" or "coaps+tcp"). A publishing Device sends an UPDATE request to "/oic/rd" with its Links in the payload to publish the Links in "/oic/res" of the RD. A publishing Device is responsible for ensuring the RD has the correct published Links exposed via its "/oic/res".

Table 2 – "oic.wk.rd" Resource Type definition

Pre-defined URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
"/oic/rd"	Resource Directory	"oic.wk.rd"	"oic.if.baseline"	The Discoverable Resource Type through with which an RD 1) facilitates its discovery and provides the criteria to select an RD and 2) allows Devices to publish their Links in "/oic/res" of the RD.	Discovery

Table 3 – "oic.wk.rd" Properties

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Selector	"sel"	"integer"	N/A	N/A	R	Yes	Provides the criteria for RD selection. An integer representing a value calculated by the RD. The value is in the range of 0 to 100. The lower the value, the more preferable the RD is.

An RD may be queried at its "/oic/res" Resource to find Resources hosted on other Devices. A publishing Device may publish all or a partial list of Resources they host to an RD. The RD then responds to queries for Resource discovery on behalf of the publishing Device. Note that only Devices that belong to the same account as the querying Device are visible in the exposed instance of "/oic/res". For general Resource discovery, the RD behaves like any other Server in responding to requests to "/oic/res".

6.1.3 RD operational flows

6.1.3.1 Discovering an RD

In Figure 5, a Device that wishes to publish its Resources first registers with the OCF Cloud that hosts the RD and then publishes the desired Resource information.

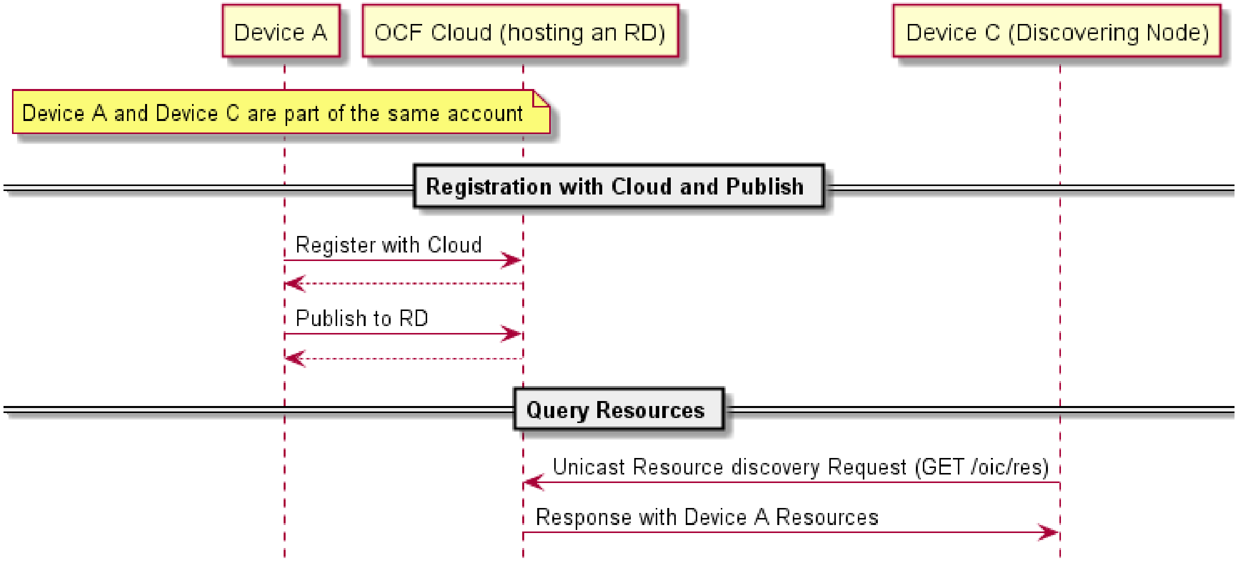
Device A
OCF Cloud (hosting an RD)
Device C (Discovering Node)
Device A and Device C are part of the same account
Registration with Cloud and Publish
Register with Cloud
Publish to RD
Query Resources
Unicast Resource discovery Request (GET /oic/res)
Response with Device A Resources

Figure 5 – RD discovery and RD supported query of Resources support

A Client that performs Resource discovery via an OCF Cloud RD does so via a unicast request to the RD; the Resource Directory defined in this document does not support the use of multicast queries to discover instances of an RD.

6.1.3.2 Publish Resources

6.1.3.2.1 Overview

After the selection process of an RD, a Device may push its Resource information to the selected RD, i.e., publish the Links in its "/oic/res" to the "/oic/res" of the RD.

The publishing Device may decide to publish all Resources or just a few of the Resources on the RD. At a minimum a publishing Device shall publish the mandatory Core Resources "/oic/d" and "/oic/p" as well as Resources that are defined as mandatory for the Device Type being published. The publishing Device should only publish Resources that are otherwise published to its own "/oic/res"; a publishing Device should not publish non-Discoverable Resources or Resources hosted by some other Device. A publishing Device shall respond to discovery requests on its "/oic/res" Resource unless all its Discoverable Resources have been published in an RD.

6.1.3.2.2 Publish: Push Resource information

Resource information may be published using an UPDATE request sent to "/oic/rd".

A Device which hosts a Resource may publish the Resource information, i.e. the Link targeting the Resource, to an RD by sending an UPDATE request with the Link in the payload. The published Link shall be exposed through the "/oic/res" of the RD.

When a Device first publishes a Link or Links, it shall send an UPDATE request to the "/oic/rd" Resource of the RD including the following key-value pairs in the payload:

– "di" –its value shall be the Device ID of the publishing Device, i.e. the "di" value of "/oic/d".

– "links" –its value shall be the array of Links to be published. Links may omit the "ins" Parameter in which case the RD will assign a value for each Link. The supplied "ins" Parameter by the Client is allowed to be overruled by the RD, e.g. an RD can ignore the supplied "ins" value.

– "ttl" –its value indicates how long (in seconds) the publishing Device requests the RD to keep this published Link.

Notice that the payload shall carry the appropriate Content-Format of "application/vnd.ocf+cbor".

```
{
  "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
  "links": [
    {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9"
      "href": "/myLightSwitch",
      "rt": ["oic.r.switch.binary"],
      "if": ["oic.if.a", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
        {"ep": "coaps://[fe80::b1d6]:1122"},
        {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
      ]
    },
    {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
      "href": "/myLightBrightness",
      "rt": ["oic.r.brightness"],
      "if": ["oic.if.a", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[[2001:db8:a::123]:2222"}
      ]
    }
  ],
  "ttl": 600
}
```

When an RD receives this initial UPDATE request, it determines whether to grant the request or not. Upon granting the request, the RD shall send back an UPDATE response to the publishing Device. The response shall include a payload with the same information as the original UPDATE request with the following possible differences:

– For each Link, an "ins" Parameter shall be included in the response. The RD shall assign a unique "ins" value identifying the Link among all the Links it advertises. If the publishing Device included an "ins" value in the UPDATE request, the RD may use it as long as it doesn't match any existing "ins" value in the published Links.

– The "ttl" Property Value shall be assigned by the RD and it shall be included in the response. The RD should use the value included in the UPDATE request but may assign a value that is lower if it is not able to honour the requested "ttl" value. After this time elapses, the RD shall remove the Links. To keep a Link alive, the publishing Device may update the "ttl" using the UPDATE schema.

The RD shall add the new Links to its "/oic/res" and expose them to a valid discovery query, i.e. RETRIEVE request:

```
{
  "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
  "links": [
    {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
      "href": "/myLightSwitch",
      "rt": ["oic.r.switch.binary"],
      "if": ["oic.if.a", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
        {"ep": "coaps://[fe80::b1d6]:1122"},
        {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
      ],
      "ins": 11235
    },
    {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
      "href": "/myLightBrightness",
      "rt": ["oic.r.brightness"],
      "if": ["oic.if.a", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[[2001:db8:a::123]:2222"}
      ],
      "ins": 112358
     }
    ].
  "ttl": 600
}
```

6.1.3.3 Resource exposure

6.1.3.3.1 "/oic/res" and retrieving of the Resources

The "/oic/res" based discovery process for an OCF Cloud does not support the use of multicast. A registered Client may discover Resources by sending a unicast RETRIEVE to "/oic/res". Only those Resources for Devices that are registered with the same account as the Client are returned in a response to the RETRIEVE.

Interaction with Resources discovered using the RD is done using the same mechanism and methods as with Resources discovered by retrieving the "/oic/res" Resource of the Device hosting the Resources (e.g., connect to the exposed endpoint and perform CRUDN operations on the Resource).

The "/oic/res" response to a requesting Client includes the Links with the "anchor" Parameter containing an OCF URI. The "/oic/res" response has a single array of Links. Each Link shall contain the "anchor" Parameter of the value OCF URI where the authority component of <deviceID> indicates the Device hosting the target Resource.

For example, an RD may return the following to a Client.

```
[
  {
    "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
    "href": "/oic/res",
    "rel": "self",
    "rt": ["oic.wk.res"],
    "if": ["oic.if.ll", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coap://[2001:db8:a::b1d4]:77777"},
      {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
    ]
  },
  {
    "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
    "href": "/oic/d",
    "rt": ["oic.wk.d", "oic.d.fan"],
    "if": ["oic.if.r", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coap://[2001:db8:a::b1d4]:77777"},
      {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
    ]
  },
  {
    "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
    "href": "/oic/p",
    "rt": ["oic.wk.p"],
    "if": ["oic.if.r", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
    ]
  },
  {
    "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
    "href": "/myFanIntrospection",
    "rt": ["oic.wk.introspection"],
    "if": ["oic.if.r", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
    ]
  },
 {
    "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
    "href": "/oic/rd",
    "rt": ["oic.wk.rd"],
    "if": ["oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
    ]
  },
  {
    "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
    "href": "/myFanSwitch",
    "rt": ["oic.r.switch.binary"],
    "if": ["oic.if.a", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
    ]
  ,
  {
    "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
    "href": "/oic/d",
    "rt": ["oic.wk.d", "oic.d.light"],
    "if": ["oic.if.r", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coap://[2001:db8:b::c2e5]:66666"},
      {"ep": "coaps://[2001:db8:b::c2e5]:22222"}
    ]
  },
  {
    "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
    "href": "/oic/p",
    "rt": ["oic.wk.p"],
    "if": ["oic.if.r", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coaps://[2001:db8:b::c2e5]:22222"}
    ]
  },
  {
    "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
    "href": "/myLightSwitch",
    "rt": ["oic.r.switch.binary"],
    "if": ["oic.if.a", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coaps://[2001:db8:b::c2e5]:22222"}
    ]
  },
  {
    "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
    "href": "/myLightBrightness",
    "rt": ["oic.r.brightness"],
    "if": ["oic.if.a", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coaps://[2001:db8:b::c2e5]:22222"}
    ]
  }
]
```

6.2 CoAPCloudConf Resource

6.2.1 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.

6.2.2 Resource Definition

The CoAPCloudConf Resource is as defined in Table 4.

Table 4 – CoAPCloudConf Resource

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
"/example/CoapCloudConfResURI"	CoAPCloud Conf	"oic.r.coapcloudconf"	"oic.if.rw", "oic.if.baseline"	Configuration information for connecting to an OCF Cloud. The Resource properties exposed are listed in Table 5.	N/A

Table 5 defines the details for the "oic.r.coapcloudconf" Resource Type.

Table 5 – 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, 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
Cloud Provisioning Status	"cps"	string	enum	N/A	R	No	Cloud provisioning status of Device. One of: "uninitialized", "readytoregister", "registering", "registered", "failed"

[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").

6.2.3 Cloud status governing state machine

6.2.3.1 Introduction

The "cps" Property exposes the registration state of the Device with an OCF Cloud. The states supported are listed in Table 6.

Table 6 – Device registration states

State	Description
"uninitialized"	Device is not initialized (i.e. CoAPCloudConf Properties set) with information of the OCF Cloud to which it will connect.
"readytoregister"	Device has been configured but not registered with the target OCF Cloud.
"registering"	A TLS session is being established, or a TLS session has been established and the Device has sent an UPDATE operation to "/oic/sec/account" as defined in clause 8.1.4 and is waiting on a response.
"registered"	The Device has received a success path response from the UPDATE operation to "/oic/sec/account".
"failed"	The Device experiences a failure during Cloud Provisioning, e.g. the Device does not receive a Success path response from the UPDATE operation. The “clec” Property when in the "failed" state if exposed indicates the specific failure reason.

Figure 6 details the state machine which describes the transitions between the values that are exposed by the "cps" Property.

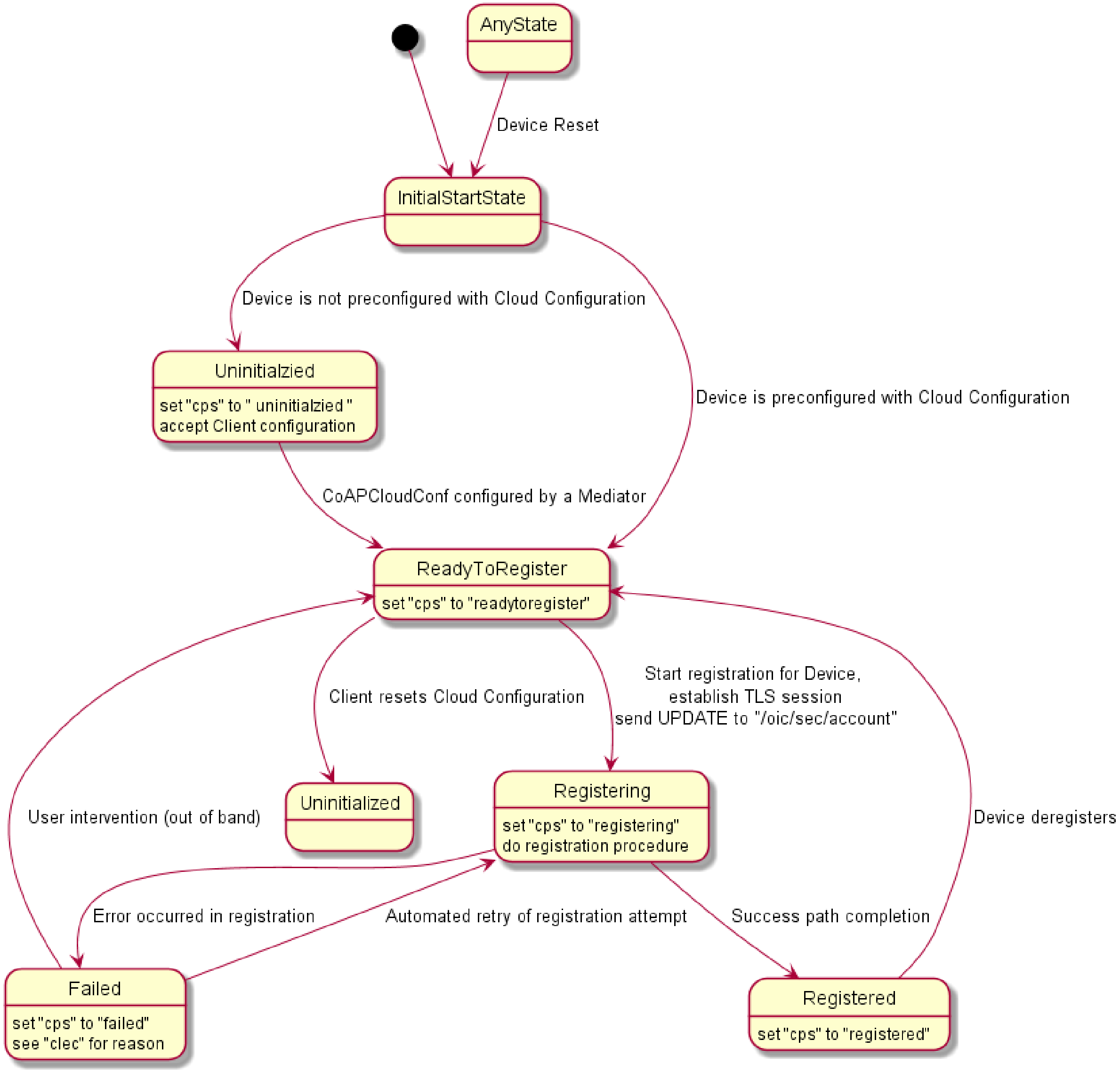
AnyState
Device Reset
InitialStartState
Device is not preconfigured with Cloud Configuration
Device is preconfigured with Cloud Configuration
Uninitialzied
set "cps" to " uninitialzied "
accept Client configuration
CoAPCloudConf configured by a Mediator
ReadyToRegister
set "cps" to "readytoregister"
Client resets Cloud Configuration
Start registration for Device,
establish TLS session
send UPDATE to "/oic/sec/account"
User intervention (out of band)
Uninitialized
Registering
set "cps" to "registering"
do registration procedure
Device deregisters
Error occurred in registration
Automated retry of registration attempt
Success path completion
Failed
set "cps" to "failed"
see "clec" for reason
Registered
set "cps" to "registered"

Figure 6 – Device registration status state machine

6.2.3.2 State definitions

6.2.3.2.1 "uninitialized" state

The Device has not been configured by a Mediator with resolvable information for the "cis", "sid", or "at" Properties of the "oic.r.coapcloudconf" Resource Type (i.e. the "cis" is a URI that cannot be resolved, and the "sid" is a null UUID). A Device may be in this state as an initial state. A Device shall transition into this state as a result of a Device reset (an appropriately privileged Client or OBT setting of "pstat") if there is no pre-configured information. It shall not be possible to perform an UPDATE operation to modify the Properties of the CoAPCloudConf Resource in any state other than "uninitialized", "readytoregister" or "failed" states.

6.2.3.2.2 "readytoregister" state

The Device has been configured by a Mediator with information for the "cis", "sid", and "at" Properties of the "oic.r.coapcloudconf" Resource Type, but has no connectivity to the OCF Cloud and is not in the process of establishing such connectivity. A Device may be in this state as an initial state. The Device shall transition to this state from the "uninitialized" state once it has been configured with values for the "cis", "at", and "sid" Properties in "oic.r.coapcloudconf". by a Mediator. A Device shall transition into this state as a result of a Device reset (Client setting of the "pstat" Property) if there is pre-configured information.

6.2.3.2.3 "registering" state

The Device shall transition to "registering" once the TLS handshake to the OCF Cloud is initiated. The Device shall transition from "registering" to "registered" on reception of a success path response to the UPDATE operation sent to the "/oic/sec/account" Resource as defined in clause 8.1.4. If a non-success path response is received to the UPDATE operation sent to the "/oic/sec/account" Resource the Device shall transition to the "failed" state, unless the Device autonomously re-attempts the registration by sending an UPDATE operation to the "/oic/sec/account" Resource as defined in clause 8.1.4. In this latter instance the Device shall remain in the "registering" state.

6.2.3.2.4 "registered" state

The Device has completed registration with the OCF Cloud as defined in clause 8.1.4. If the Device subsequently deregisters in accordance with clause 8.5 the Device shall transition to the "readytoregister" state.

6.2.3.2.5 "failed" state

The Device has received a non-success path response from the OCF Cloud during the registration procedure as defined in clause 8.1.4 and is not attempting an autonomous retry or re-attempt. The Device may offer some out of band means, or user intervention scheme, that allows the transition from the "failed" state to the "readytoregister" or the "uninitialized" state to enable re-attempt.

The "clec" Property, if exposed, shall be populated with the specific failure reason why the Device is in the "failed" state.

6.2.4 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 7 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.

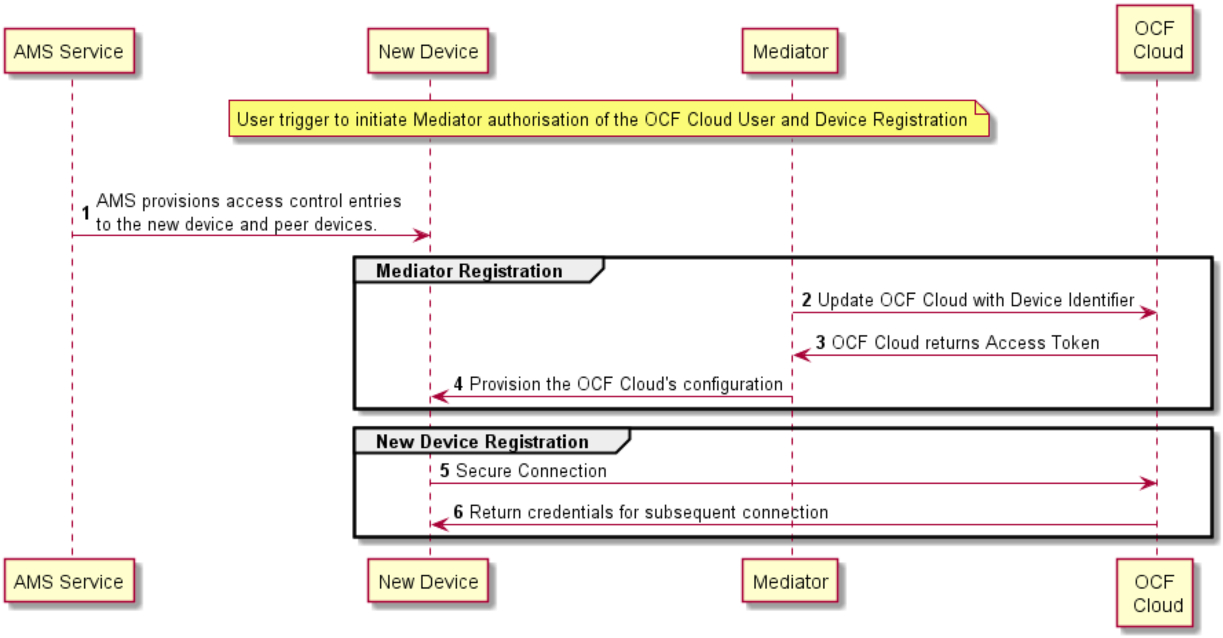
AMS Service
New Device
Mediator
OCF Cloud
User trigger to initiate Mediator authorisation of the OCF Cloud User and Device Registration
1 AMS provisions access control entries to the new device and peer devices.
Mediator Registration
2 Update OCF Cloud with Device Identifier
3 OCF Cloud returns Access Token
4 Provision the OCF Cloud's configuration
New Device Registration
5 Secure Connection
6 Return credentials for subsequent connection
AMS Service
New Device
Mediator
OCF Cloud

Figure 7 – Registration with OCF Cloud

Table 7 – 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 8 describes the flow for provisioning of the Device by a Mediator. Table 8 provides additional context around the flow.

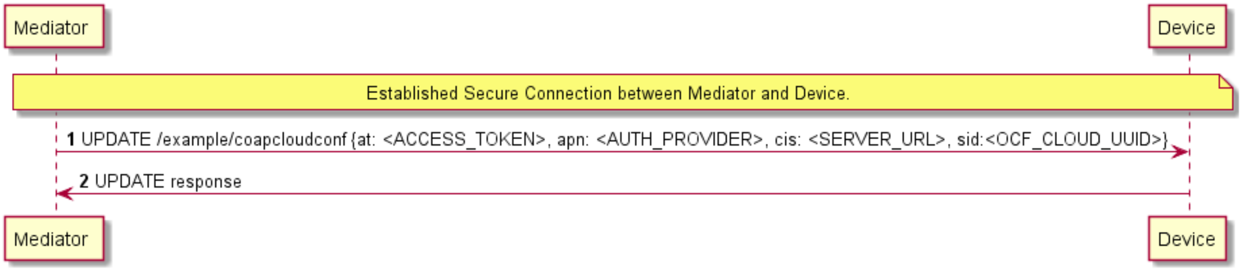
Mediator
Device
Established Secure Connection between Mediator and Device.
1 UPDATE /example/coapcloudconf {at: <ACCESS_TOKEN>, apn: <AUTH_PROVIDER>, cis: <SERVER_URL>, sid:<OCF_CLOUD_UUID>}
2 UPDATE response
Mediator
Device

Figure 8 – Device Provisioning by the Mediator

Table 8 – 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 9 provides an example showing the provided Device ID from the Device; Figure 10 shows the pre-pending of the Device ID to the "href" Link Parameter in the Resource Directory itself.

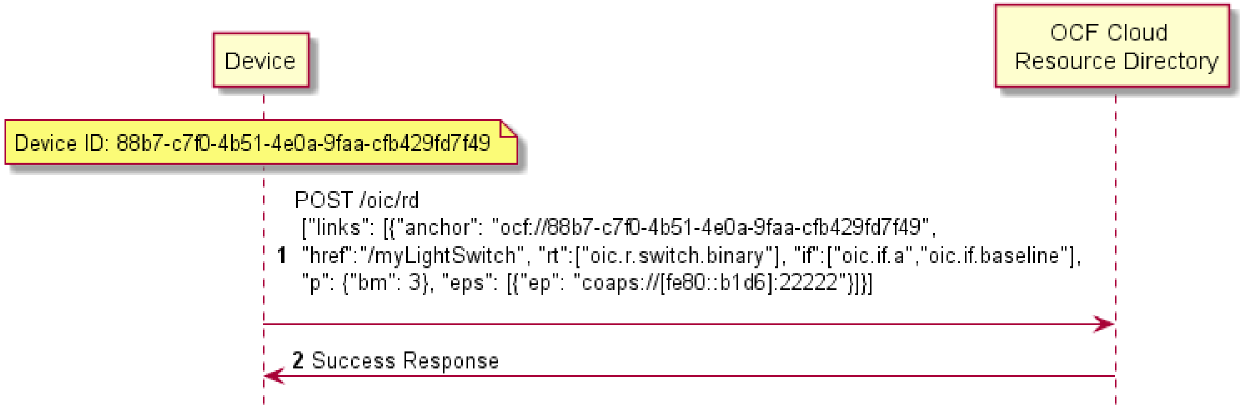
Device
OCF Cloud
Resource Directory
Device ID: 88b7-c7f0-4b51-4e0a-9faa-cfb429fd7f49
1
POST /oic/rd
["links": [{"anchor": "ocf://88b7-c7f0-4b51-4e0a-9faa-cfb429fd7f49",
"href":"/myLightSwitch", "rt":["oic.r.switch.binary"], "if":["oic.if.a","oic.if.baseline"],
"p": {"bm": 3}, "eps": [{"ep": "coaps://[fe80::b1d6]:22222"}]}]
2 Success Response

Figure 9 – 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 is for the OCF Cloud and not the publishing Device.

Figure 10 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:

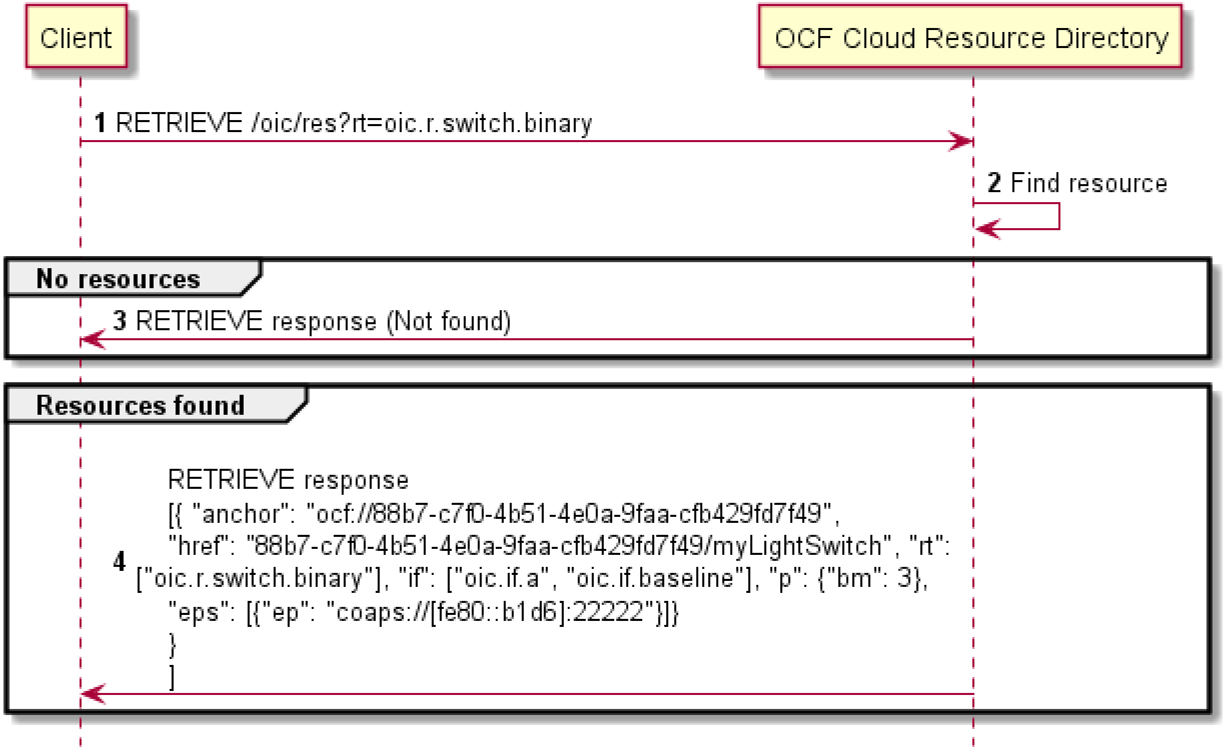
Client
OCF Cloud Resource Directory
1 RETRIEVE /oic/res?rt=oic.r.switch.binary
2 Find resource
No resources
3 RETRIEVE response (Not found)
Resources found
4 RETRIEVE response
[{ "anchor": "ocf://88b7-c7f0-4b51-4e0a-9faa-cfb429fd7f49",
"href": "88b7-c7f0-4b51-4e0a-9faa-cfb429fd7f49/myLightSwitch", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a", "oic.if.baseline"], "p": {"bm": 3},
"eps": [{"ep": "coaps://[fe80::b1d6]:22222"}]}
}
]

Figure 10 – 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 prepended 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 11 illustrates request routing via the OCF Cloud

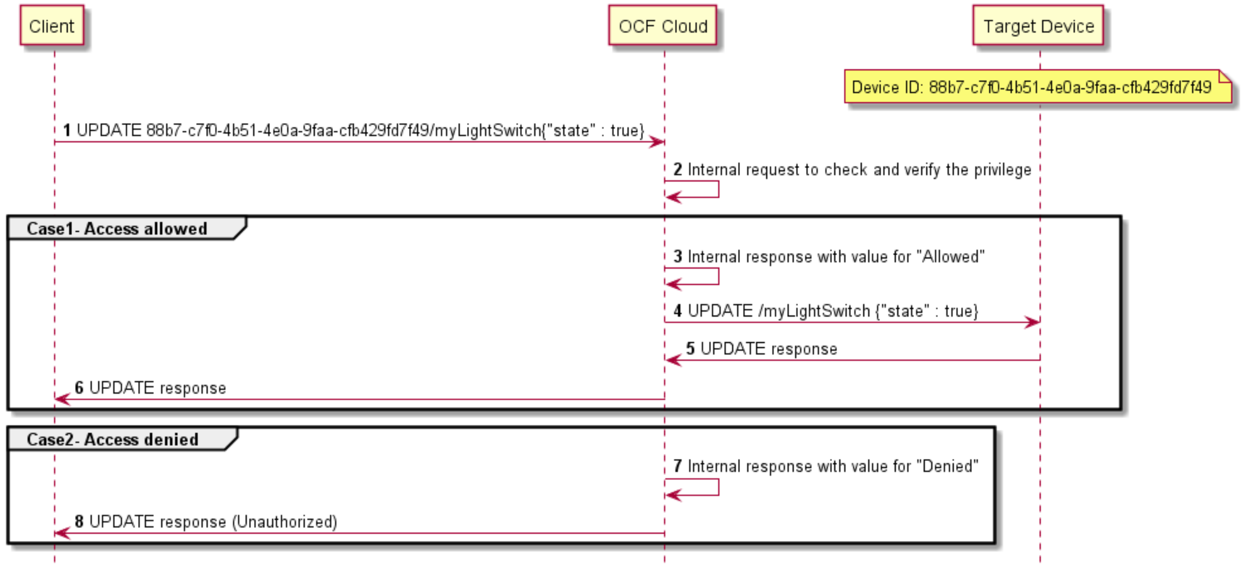
Client
OCF Cloud
Target Device
Device ID: 88b7-c7f0-4b51-4e0a-9faa-cfb429fd7f49
1 UPDATE 88b7-c7f0-4b51-4e0a-9faa-cfb429fd7f49/myLightSwitch{"state" : true}
2 Internal request to check and verify the privilege
Case1- Access allowed
3 Internal response with value for "Allowed"
4 UPDATE /myLightSwitch {"state" : true}
5 UPDATE response
6 UPDATE response
Case2- Access denied
7 Internal response with value for "Denied"
8 UPDATE response (Unauthorized)

Figure 11 – 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
Resource Directory	"oic.wk.rd"	A.2
CoAP Cloud Configuration	"oic.r.coapcloudconf"	A.3

A.2 Resource directory resource

A.2.1 Introduction

Resource to be exposed by any Device that can act as a Resource Directory.
1) Provides selector criteria (e.g., integer) with GET request
2) Publish a Link in /oic/res with POST request

A.2.2 Well-known URI

/oic/rd

A.2.3 Resource type

The Resource Type is defined as: "oic.wk.rd".

A.2.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Resource directory resource",
    "version": "2019-02-22",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.md",
      "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/oic/rd" : {
      "get": {
        "description": "Resource to be exposed by any Device that can act as a Resource
Directory.\n1) Provides selector criteria (e.g., integer) with GET request\n2) Publish a Link in
/oic/res with POST request\n",
        "parameters": [
          {"$ref": "#/parameters/rdgetinterface"}
        ],
        "responses": {
            "200": {
              "description" : "Respond with the selector criteria - either the set of attributes or
the bias factor\n",
              "x-example": {
                "rt": ["oic.wk.rd"],
                "if": ["oic.if.baseline"],
                "sel": 50
              },
              "schema": { "$ref": "#/definitions/rdSelection" }
            }
        }
      },
      "post": {
        "description": "Publish the Resource information for the first time in /oic/res. Updates to
existing entries are not allowed.\nAppropriates parts of the information, i.e., Links of the
published Resources will be discovered through /oic/res.\n1) When a Device first publishes a Link,
the request payload to RD may include the Links without an \"ins\" Parameter.\n2) Upon granting the
request, the RD assigns a unique instance value identifying the Link among all the Links it
advertises\n   and sends back the instance value in the \"ins\" Parameter in the Link to the
publishing Device.\n",
        "parameters": [
          {"$ref": "#/parameters/rdpostinterface"},
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/rdPublish" },
            "x-example": {
              "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
              "links": [
                {
                  "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
                  "href":   "/myLightSwitch",
                  "rt":     [ "oic.r.switch.binary" ],
                  "if":     [ "oic.if.a", "oic.if.baseline" ],
                  "p":      { "bm": 3 },
                  "eps": [
                    { "ep": "coaps://[2001:db8:a::b1d6]:1111", "pri": 2 },
                    { "ep": "coaps://[2001:db8:a::b1d6]:1122" },
                    { "ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3 }
                  ]
                },
                {
                  "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
                  "href":   "/myLightBrightness",
                  "rt":     [ "oic.r.brightness" ],
                  "if":     [ "oic.if.a", "oic.if.baseline" ],
                  "p":      { "bm": 3 },
                  "eps": [
                    { "ep": "coaps://[[2001:db8:a::123]:2222" }
                  ]
                }
              ],
              "ttl": 600
            }
          }
        ],
        "responses": {
          "200": {
            "description" : "Respond with the same schema as publish with the additional \"ins\"
Parameter in the Link.\n",
            "x-example": {
              "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
              "links": [
                {
                  "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
                  "href":   "/myLightSwitch",
                  "rt":     [ "oic.r.switch.binary" ],
                  "if":     [ "oic.if.a", "oic.if.baseline" ],
                  "p":      { "bm": 3 },
                  "eps": [
                    { "ep": "coaps://[2001:db8:a::b1d6]:1111", "pri": 2 },
                    { "ep": "coaps://[2001:db8:a::b1d6]:1122" },
                    { "ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3 }
                  ],
                  "ins":    11235
                },
                {
                  "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
                  "href":   "/myLightBrightness",
                  "rt":     ["oic.r.brightness"],
                  "if":     ["oic.if.a", "oic.if.baseline"],
                  "p":      {"bm": 3},
                  "eps": [
                      {"ep": "coaps://[2001:db8:a::123]:2222"}
                  ],
                  "ins":    112358
                }
              ],
              "ttl": 600
            },
            "schema": { "$ref": "#/definitions/rdPublish" }
          }
        }
      }
    }
  },
  "parameters": {
    "rdgetinterface" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.baseline"]
    },
    "rdpostinterface" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.baseline"]
    }
  },
  "definitions": {
    "rdSelection" : {
      "properties": {
        "rt" : {
          "description": "Resource Type of the Resource",
          "items": {
            "enum": ["oic.wk.rd"],
            "type": "string",
            "maxLength": 64
          },
          "minItems": 1,
          "uniqueItems": true,
          "readOnly": true,
          "type": "array"
        },
        "n" : {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
        },
        "sel" : {
          "description": "A bias factor calculated by the Resource Directory",
          "maximum": 100,
          "minimum": 0,
          "readOnly": true,
          "type": "integer"
        },
        "id" : {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
        },
        "if" : {
          "description": "The OCF Interfaces supported by this Resource",
          "items": {
            "enum": [
              "oic.if.baseline"
            ],
            "type": "string",
            "maxLength": 64
          },
          "minItems": 1,
          "readOnly": true,
          "uniqueItems": true,
          "type": "array"
        }
      },
      "type" : "object",
      "required": ["sel"]
    },
    "rdPublish" : {
      "properties": {
        "di" : {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/di"
        },
        "ttl" : {
          "description": "Time to indicate a RD, i.e. how long to keep this published item.",
          "type": "integer"
        },
        "links" : {
          "description": "A set of simple or individual OCF Links.",
          "items": {
            "properties": {
              "anchor": {
                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/anchor"
              },
              "di": {
                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/di"
              },
              "eps": {
                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/eps"
              },
              "href": {
                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
              },
              "if": {
                "description": "The interface set supported by the published resource",
                "items": {
                  "enum": [
                    "oic.if.baseline",
                    "oic.if.ll",
                    "oic.if.b",
                    "oic.if.rw",
                    "oic.if.r",
                    "oic.if.a",
                    "oic.if.s"
                  ],
                  "type": "string",
                  "maxLength": 64
                },
                "minItems": 1,
                "uniqueItems": true,
                "type": "array"
              },
              "ins": {
                "$ref":
```

```
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/ins"
              },
              "p": {
                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/p"
              },
              "rel": {
                "description": "The relation of the target URI referenced by the Link to the context
URI",
                "oneOf": [
                  {
                    "default": [
                      "hosts"
                    ],
                    "items": {
                      "maxLength": 64,
                      "type": "string"
                    },
                    "minItems": 1,
                    "type": "array"
                  },
                  {
                    "default": "hosts",
                    "maxLength": 64,
                    "type": "string"
                  }
                ]
              },
              "rt": {
                "description": "Resource Type of the published Resource",
                "items": {
                  "maxLength": 64,
                  "type": "string"
                },
                "minItems": 1,
                "maxItems": 1,
                "uniqueItems": true,
                "type": "array"
              },
              "title": {
                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/title"
              },
              "type": {
                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/type"
              }
            },
            "required": [
              "href",
              "rt",
              "if"
            ],
            "type": "object"
          },
          "type": "array"
        }
      },
      "type" : "object",
      "required": ["di", "links", "ttl"]
    }
  }
}
```

A.2.5 Property definition

Table A-2 defines the Properties that are part of the "oic.wk.rd" Resource Type.

Table A-2 – The Property definitions of the Resource with type "rt" = "oic.wk.rd".

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	Resource Type of the Resource.
n	multiple types: see schema	No	Read Write	
sel	integer	Yes	Read Only	A bias factor calculated by the Resource Directory.
id	multiple types: see schema	No	Read Write	
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.
di	multiple types: see schema	Yes	Read Write	
ttl	integer	Yes	Read Write	Time to indicate a RD, i.e. how long to keep this published item.
links	array: see schema	Yes	Read Write	A set of simple or individual OCF Links.

A.2.6 CRUDN behaviour

Table A-3 defines the CRUDN operations that are supported on the "oic.wk.rd" Resource Type.

Table A-3 – The CRUDN operations of the Resource with type "rt" = "oic.wk.rd".

Create	Read	Update	Delete	Notify
	get	post		observe

A.3 CoAP Cloud Configuration Resource

A.3.1 Introduction

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

A.3.2 Example URI

/CoAPCloudConfResURI

A.3.3 Resource type

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

A.3.4 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.3.5 Property definition

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

Table A.4 – 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.3.6 CRUDN behaviour

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

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

Create	Read	Update	Delete	Notify
	get	post		observe