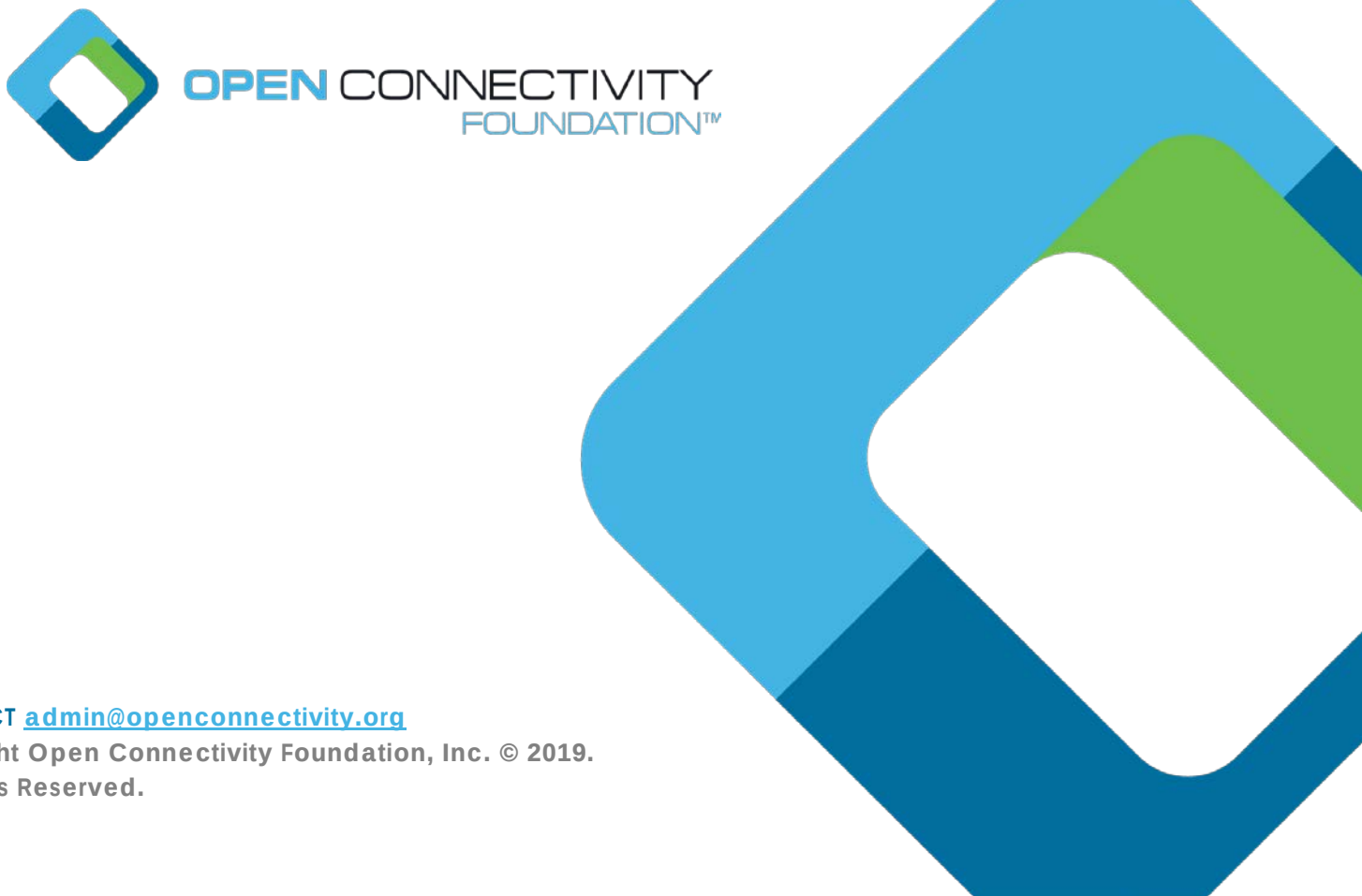
OCF Cloud Sercurity Specification

VERSION 2.0.5 | September 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 DOCUMENT 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 INTERCONNECT CONSORTIUM, INC. FURTHER DISCLAIMS ANY AND ALL WARRANTIES OF NON-INFRINGEMENT, ACCURACY OR LACK OF VIRUSES.

FIGURES

Tables

1 Purpose and Role

This document defines security objectives, philosophy, resources and mechanism that impacts OCF base layers of ISO/IEC 30118-1:2018. ISO/IEC 30118-1:2018 contains informative security content. The OCF Security Document contains security normative content and may contain informative content related to the OCF base or other OCF documents.

2 Normative References

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IETF RFC 7228, *Terminology for Constrained-Node Networks*, May 2014, https://tools.ietf.org/html/rfc7228

ISO/IEC 30118-1:2018 Information technology -- Open Connectivity Foundation (OCF) Document -- Part 1: Core document
https://www.iso.org/standard/53238.html
Latest version available at:
https://openconnectivity.org/specs/OCF_Core_Specification.pdf

OCF Security Document, Information technology – Open Connectivity Foundation (OCF) Document, Latest version available
at:https://openconnectivity.org/specs/OCF_Security_Specification.pdf

OCF Device to Cloud Services Document, Information technology – Open Connectivity Foundation (OCF) Document – Part 8: Device to Cloud Services, Latest version available at:
https://openconnectivity.org/specs/OCF_OCF_Device_To_Cloud_Services_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

oneM2M Release 3 Documents, http://www.onem2m.org/technical/published-drafts

OpenAPI document, aka *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 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
Access Management Service (AMS)

a service that dynamically constructs ACL Resources in response to a Device Resource request.

Note 1 to entry: An AMS can evaluate access policies remotely and supply the result to a Server which allows or denies a pending access request. An AMS is authorised to provision ACL Resources.

3.1.2
Trust Anchor

a well-defined, shared authority, within a trust hierarchy, by which two cryptographic entities (e.g. a Device and an onboarding tool) can assume trust

3.1.3
OCF Security Domain

a set of onboarded OCF Devices that are provisioned with credentialing information for confidential communication with one another

3.1.4
Access Token

a credential used to authorize the connection with the OCF Cloud and access protected resources. An Access Token is a string while the OCF Device has no internal logic based on its contents and only forwards the token as-is

3.1.5
Authorization Provider

a Server issuing Access Tokens (3.1.4) to the Client after successfully authenticating the OCF Cloud User (3.1.7) and obtaining authorization.

Note 1 to entry: Also known as authorization server in IETF RFC 6749.

3.1.6
Device Registration

a process by which Device is enrolled/registered to the OCF Cloud infrastructure (using Device certificate and unique credential) and becomes ready for further remote operation through the cloud interface (e.g. connection to remote Resources or publishing of its own Resources for access).

3.1.7
OCF Cloud User

a person or organization authorizing a set of Devices to interact with each other via an OCF Cloud.

Note 1 to entry: For each of the Devices, the OCF Cloud User is either the same as, or a delegate of, the person or organization that onboarded that Device. The OCF Cloud User delegates, to the OCF Cloud authority, authority to route between Devices registered by the OCF Cloud User. The OCF Cloud delegates, to the OCF Cloud User, authority to select the set of Devices which can register and use the services of the OCF Cloud.

3.2 Abbreviated terms

3.2.1
ACE
Access Control Entry

3.2.2
ACL
Access Control List

3.2.3
AMS
Access Management Service

3.2.4
CMS
Credential Management Service

4 Document Conventions and Organization

4.1 Conventions

This document defines Resources, protocols and conventions used to implement security for OCF core framework and applications.

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

Figure 1 depicts interaction between OCF Devices.

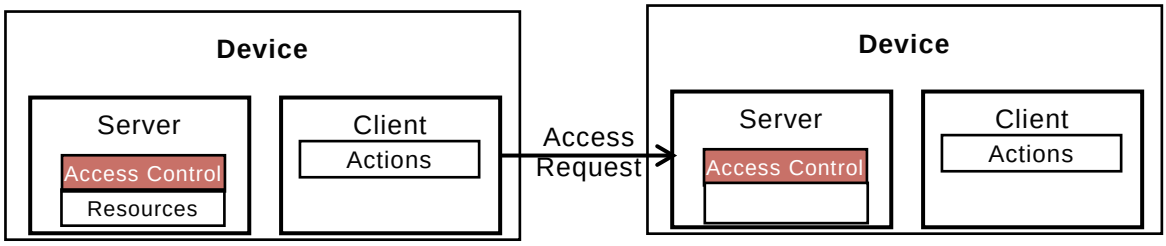


Figure 1 – OCF Interaction

Devices may implement a Client role that performs Actions on Servers. Actions access Resources managed by Servers. The OCF stack enforces access policies on Resources. End-to-end Device interaction can be protected using session protection protocol (e.g. DTLS) or with data encryption methods.

4.2 Notation

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

Required (or **shall** or **mandatory**).

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

Recommended (or **should**).

These features add functionality supported by OCF Core Architecture and should be implemented. Recommended features take advantage of the capabilities OCF 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 behavior that is permitted but not recommended.

Allowed (may or allowed).

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

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.

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.

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

Words that are emphasized are printed in italic.

4.3 Data types

See ISO/IEC 30118-1:2018.

4.4 Document structure

Informative clauses may be found in the Overview clauses, while normative clauses fall outside of those clauses.

The Security Document may use the oneM2M Release 3 Documents,
http://www.onem2m.org/technical/published-drafts

OpenAPI as the API definition language. The mapping of the CRUDN actions is specified in ISO/IEC 30118-1:2018.

5 Security overview

5.1 Preamble

A Device is authorized to communicate with an OCF Cloud if a trusted Mediator has provisioned the Device.

- Device and Mediator connect over DTLS using "/oic/sec/cred"
- Device is provisioned by Mediator with following information:
  - the URL of OCF Cloud
  - Authorization Provider Name to identify the origin of the Access Token
  - Access Token / Authorization Code that is validated / exchanged by the OCF Cloud
  - UUID of the OCF Cloud

The OpenAPI 2.0 definitions (Annex A) used in this document are normative. This includes that all defined payloads shall comply with the indicated OpenAPI 2.0 definitions. Annex A contains all of the OpenAPI 2.0 definitions for Resource Types defined in this document.

5.2 Device Provisioning for OCF Cloud and Device Registration Overview

As mentioned in the start of Clause 0, communication between a Device and OCF Cloud is subject to different criteria in comparison to Devices which are within a single local network. The Device is configured in order to connect to the OCF Cloud by a Mediator as specified in the CoAPCloudConf Resource clauses in OCF Cloud . Provisioning includes the remote connectivity and local details such as URL where the OCF Cloud hosting environment can be found, the OCF Cloud verifiable Access Token and optionally the name of the Authorization Provider which issued the Access Token.

NOTE a Device which connects to the OCF Cloud still retains the ownership established at onboarding with the DOTS.

5.3 Credential overview

Devices may use credentials to prove the identity and role(s) of the parties in bidirectional communication

Access Tokens are provided to an OCF Cloud once an authenticated session with an OCF Cloud is established, to verify the User ID with which the Device is to be associated.

6 Device provisioning for OCF Cloud

6.1 Cloud Provisioning General

The Device that connects to the OCF Cloud shall support the "oic.r.coapcloudconf" Resource on Device and following SVRs on the OCF Cloud: "/oic/sec/account", "/oic/sec/session", "/oic/sec/tokenrefresh".

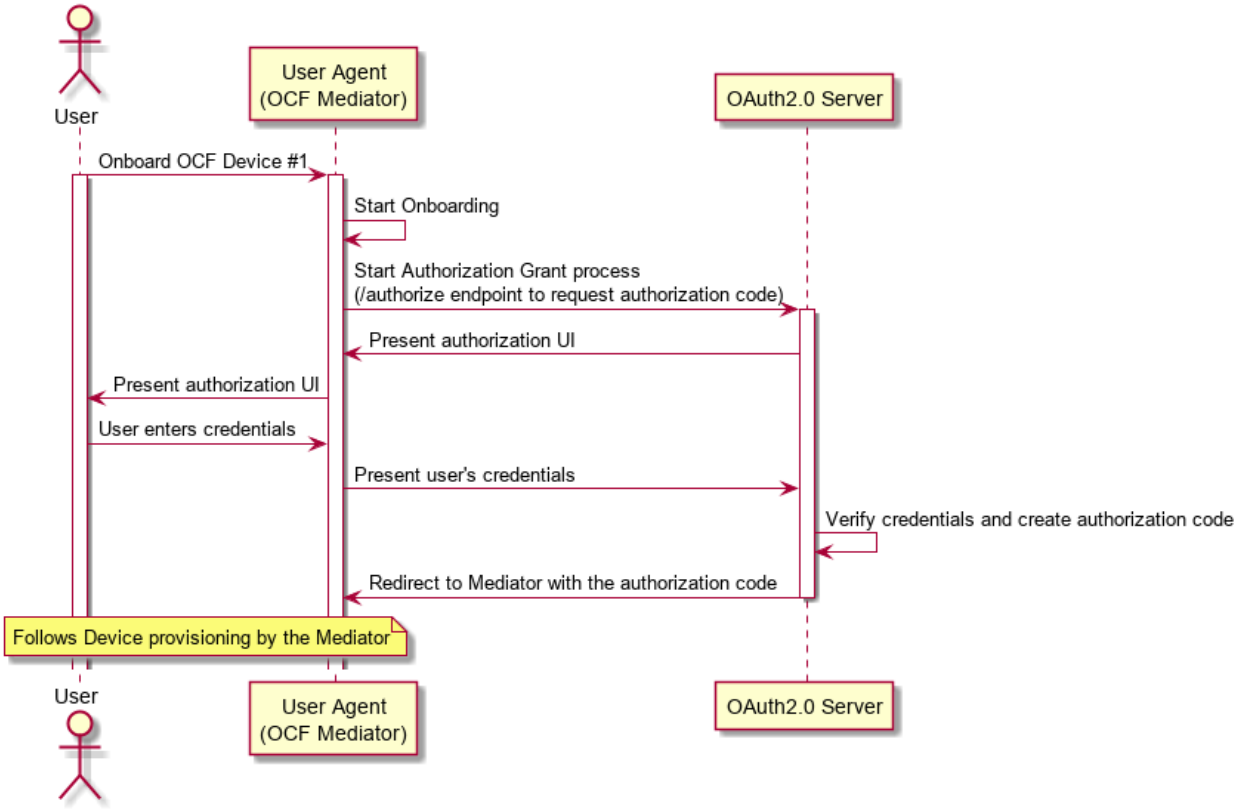
The OCF Cloud is expected to use a secure mechanism for associating a Mediator with an OCF Cloud User. The choice of mechanism is up to the OCF Cloud.Recommended solution is based on the OAuth2.0 Authorization Grant Type flow specified in IETF RFC 6749, where the Mediator act as a User-Agent and presents authorization UI to the user - see Figure 2. OCF Cloud is expected to ensure that the suitable authentication mechanism is used to authenticate the OCF Cloud User.

Figure 2 User authorization and provisioning using Authorization Code Grant Flow

```
@startuml

actor User
participant UserAgent as "User Agent\n(OCF Mediator)"
participant OAuthServer as "OAuth2.0 Server"User -> UserAgent: Onboard OCF Device #1
activate User
```



```
activate UserAgent
UserAgent -> UserAgent: Start Onboarding
UserAgent -> OAuthServer: Start Authorization Grant process\n(/authorize endpoint to
request authorization code)
activate OAuthServer
OAuthServer -> UserAgent: Present authorization UI
UserAgent -> User: Present authorization UI
User -> UserAgent: User enters credentials
UserAgent -> OAuthServer: Present user's credentials
OAuthServer -> OAuthServer: Verify credentials and create authorization code
OAuthServer -> UserAgent: Redirect to Mediator with the authorization code
deactivate OAuthServer
note over User, UserAgent
Follows Device provisioning by the Mediator
end note

@enduml
```



6.2 Device Provisioning by Mediator

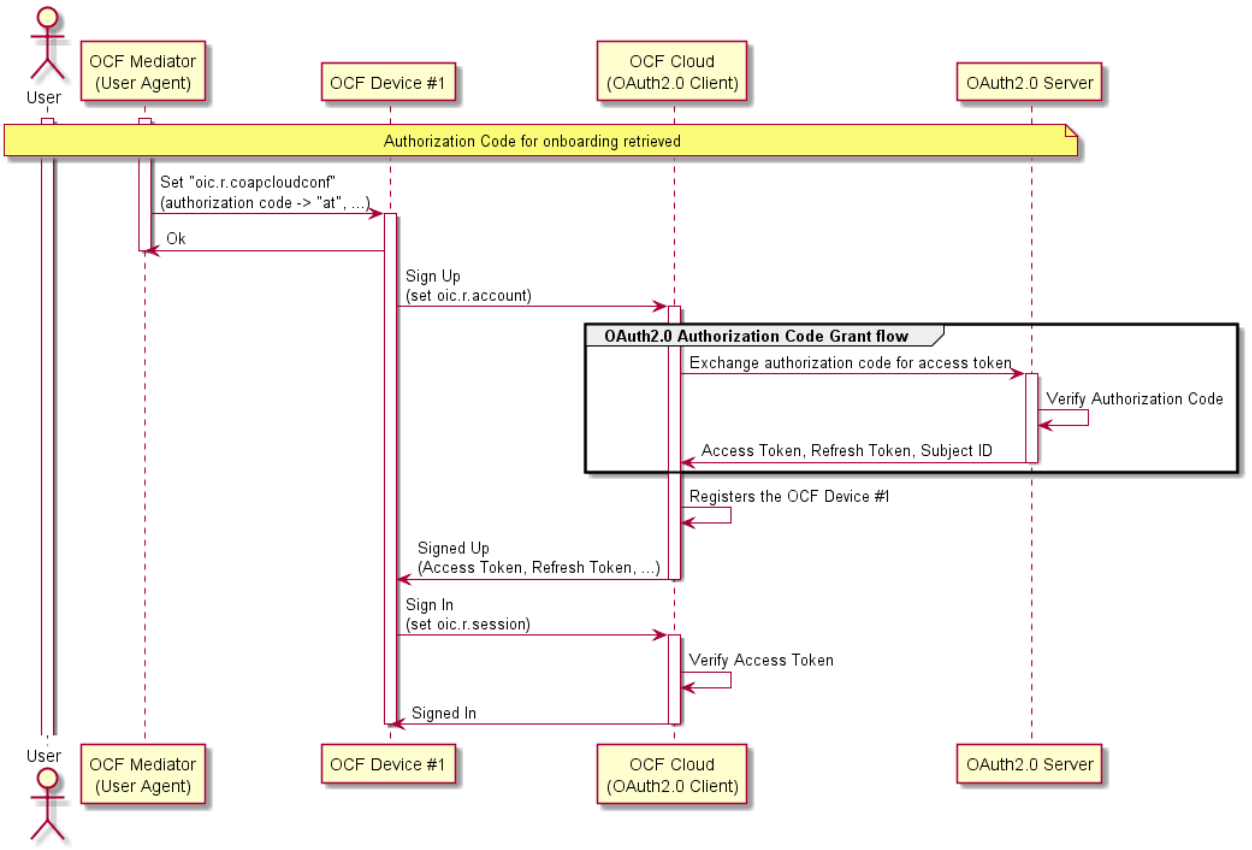
The Mediator and the Device shall use the secure session to provision the Device to connect with the OCF Cloud.

The Mediator obtains an Authorization Code or directly an Access Token from the Authorization Server as described in OCF Cloud . This value is then used by the Device for registering with the OCF Cloud as described in clause 7. At the time of Device Registration OCF Cloud exchanges the Authorization Code for the Access Token, returns it back to the OCF Device and associates the TLS session with corresponding Device ID. The OCF Cloud maintains a map where Access Token and Mediator provided Device ID are stored.

The Mediator provisions the Device, as described in OCF Cloud . The Mediator provisions OCF Cloud URI to the "cis" Property of "oic.r.coapcloudconf" Resource, OCF Cloud UUID to the "sid" Property of "oic.r.coapcloudconf" Resource and per-device Access Token or Authorization Code to the "at" Property of "oic.r.coapcloudconf" Resource on Device. Exchanged and returned provisioned Access Token is to be treated by Device as an Access Token with "Bearer" token type as defined in IETF RFC 6750. The provisioned "at" value follows a proprietary data format, and may include multiple values marshalled/concatenated together into a single string (e.g. "{\"token\":\"abc\", \"client_id\":\"1234\", \"idp\":\"identityProvider1\"}" is a valid "at" Property value). See Figure 3 for the detailed overview of the recommended flow, which includes optional OAuth 2.0 Authorization Code Grant

Figure 3 Device Provisioning using Authorization Code Grant Flow

```
@startuml

actor User
participant Mediator as "OCF Mediator\n(User Agent)"
participant Device as "OCF Device #1"
participant Cloud as "OCF Cloud\n(OAuth2.0 Client)"
participant OAuthServer as "OAuth2.0 Server"

activate User
activate Mediator

note over User, OAuthServer
Authorization Code for onboarding retrieved
end note

Mediator -> Device: Set "oic.r.coapcloudconf"\n(authorization code -> "at", ...)
activate Device
Device -> Mediator: Ok
deactivate Mediator
Device -> Cloud: Sign Up\n(set oic.r.account)
activate Cloud
group OAuth2.0 Authorization Code Grant flow
     Cloud -> OAuthServer: Exchange authorization code for access token
     activate OAuthServer
     OAuthServer -> OAuthServer: Verify Authorization Code
     OAuthServer -> Cloud: Access Token, Refresh Token, Subject ID
     deactivate OAuthServer
end
Cloud -> Cloud: Registers the OCF Device #1
Cloud -> Device: Signed Up\n(Access Token, Refresh Token, ...)
deactivate Cloud
Device -> Cloud: Sign In\n(set oic.r.session)
activate Cloud
Cloud -> Cloud: Verify Access Token
Cloud -> Device: Signed In
deactivate Device
deactivate Cloud

@enduml
```

For the purposes of access control, the Device shall identify the OCF Cloud using the OCF Cloud UUID in the Common Name field of the End-Entity certificate used to authenticate the OCF Cloud.

AMS should configure the ACE2 entries on a Device so that the Mediator(s) is the only Device(s) with UPDATE permission for the "oic.r.coapcloudconf" Resource.

The AMS should configure the ACE2 entries on the Device to allow request from the OCF Cloud. By request from the Mediator, the AMS removes old ACL2 entries with previous OCF Cloud UUID. This request happens before "oic.r.coapcloudconf" is configured by the Mediator for the new OCF Cloud. The Mediator also requests AMS to set the OCF Cloud UUID as the "subject" Property for the new ACL2 entries. AMS may use "sid" Property of "oic.r.coapcloudconf" Resource as the current OCF Cloud UUID. AMS could either provision a wildcard entry for the OCF Cloud or provision an entry listing each Resource published on the Device.

If OCF Cloud provides "redirecturi" Value as response during Device Registration, the redirected-to OCF Cloud is assumed to have the same OCF Cloud UUID and to use the same trust anchor. Otherwise, presented OCF Cloud UUID wouldn't match the provisioned ACL2 entries.

The Mediator should provision the "oic.r.coapcloudconf" Resource with the Properties in Table 1. These details once provisioned are used by the Device to perform Device Registration to the OCF Cloud. OCF Device is not expected to have any internal logic based on the values of "at" and "apn" Properties. The values of these Properties are forwarded as-is to the OCF Cloud. After the initial registration, the Device should use updated values received from the OCF Cloud instead. If OCF Cloud User wants the Device to re-register with the OCF Cloud, they can use the Mediator to re-provision the "oic.r.coapcloudconf" Resource with the new values.

Table 1 – Mapping of Properties of the "oic.r.account" and "oic.r.coapcloudconf" Resources

Property Title	oic.r.coapcloudconf	oic.r.account	Description
Authorization Provider Name	apn	authprovider	The name of Authorization Provider through which Access Token was obtained.
OCF Cloud URL	cis	-	This is the URL connection is established between Device and OCF Cloud.
Access Token	at	accesstoken	Access Token used to authorize the TLS connection for communication with the OCF Cloud, or the Authorization Code which is then verified and exchanged for the Access Token during Device Registration.
OCF Cloud UUID	sid	-	This is the identity of the OCF Cloud that the Device is configured to use.

7 Device authentication with OCF Cloud

7.1 Device Authentication with OCF Cloud General

The mechanisms for Device Authentication in clauses 10.2, 10.3 and 10.4 of OCF Security imply that a Device is authorized to communicate with any other Device meeting the criteria provisioned in "/oic/sec/cred"; the "/oic/sec/acl2" Resource (or "/oic/sec/acl1" resource of OIC1.1 Servers) are additionally used to restrict access to specific Resources. The present clause describes Device authentication for OCF Cloud, which uses slightly different criteria as described in clause 0. A Device accessing an OCF Cloud shall establish a TLS session. The mutual authenticated TLS session is established using Server certificate and Client certificate.

Each Device is identified by the Access Token obtained from the Device Registration response. The OCF Cloud holds an OCF Cloud association table that maps Access Token, User ID and Device ID. The Device Registration shall happen while the Device is in RFNOP state. After Device Registration, the updated Access Token, Device ID and User ID are used by the Device for the subsequent connection with the OCF Cloud.

7.2 Device Connection with the OCF Cloud

The Device should establish the TLS connection using the certificate based credential. The connection should be established after Device is provisioned by Mediator.

The TLS session is established between Device and the OCF Cloud as specified in IETF RFC 8323. The OCF Cloud is expected to provide certificate signed by trust anchor that is present in cred entries of the Device. These cred entries are expected to be configured by the Mediator.

The Device shall validate the OCF Cloud's identity based on the credentials that are contained in "/oic/sec/cred" Resource entries of the Device.

The OCF Cloud is expected to validate the manufacturer certificate provided by the Device.

The assumption is that the OCF Cloud User trusts the OCF Cloud that the Device connects. The OCF Cloud connection should not happen without the consent of the OCF Cloud User. The

assumption is that the OCF Cloud User has either service agreement with the OCF Cloud provider or uses manufacturer provided OCF Cloud.

If authentication fails, the "clec" Property of "oic.r.coapcloudconf" Resource on the Device shall be updated about the failed state, if it is supported by the Device. If authentication succeeds, the Device and OCF Cloud should establish an encrypted link in accordance with the negotiated cipher suite.

Figure 4 depicts sequence for Device connection with OCF Cloud and steps described in Table 2.

Figure 4 – Device connection with OCF Cloud

```
@startuml
autonumber
title Device Connection with OCF Cloud
participant "Device\n(UUID A71C3xxx-...)" as RS
participant "OCF Cloud" as CI

group "CoAP over TCP"
note over RS, CI
- Connect to the OCF Cloud using the provisioned certificate by manufacturer.
- The manufacturer certificate private key is used to sign handshake messages.
- Both Device and OCF Cloud resolve the certificate to its trust anchor.
end note

RS->CI: ClientHello(TLS_ECDHE_ECDSA_WITH_AES_128_GCM)
CI->RS: HelloVerifyRequest(cookie)
RS->CI: ClientHello(TLS_ECDHE_ECDSA_WITH_AES_128_GCM,cookie)
CI->RS:
ServerHello(TLS_ECDHE_ECDSA_WITH_AES_128_GCM)\nCertificate*\nServerKeyExchange(ECDH
PublicKey + ECC Curve Param)\nServerHelloDone()
RS->CI: Certificate*\nClientKeyExchange(ECDH PublicKey)\nChangeCipherSpec + Finish
CI->RS: ChangeCipherSpec + Finish

End
@enduml
```

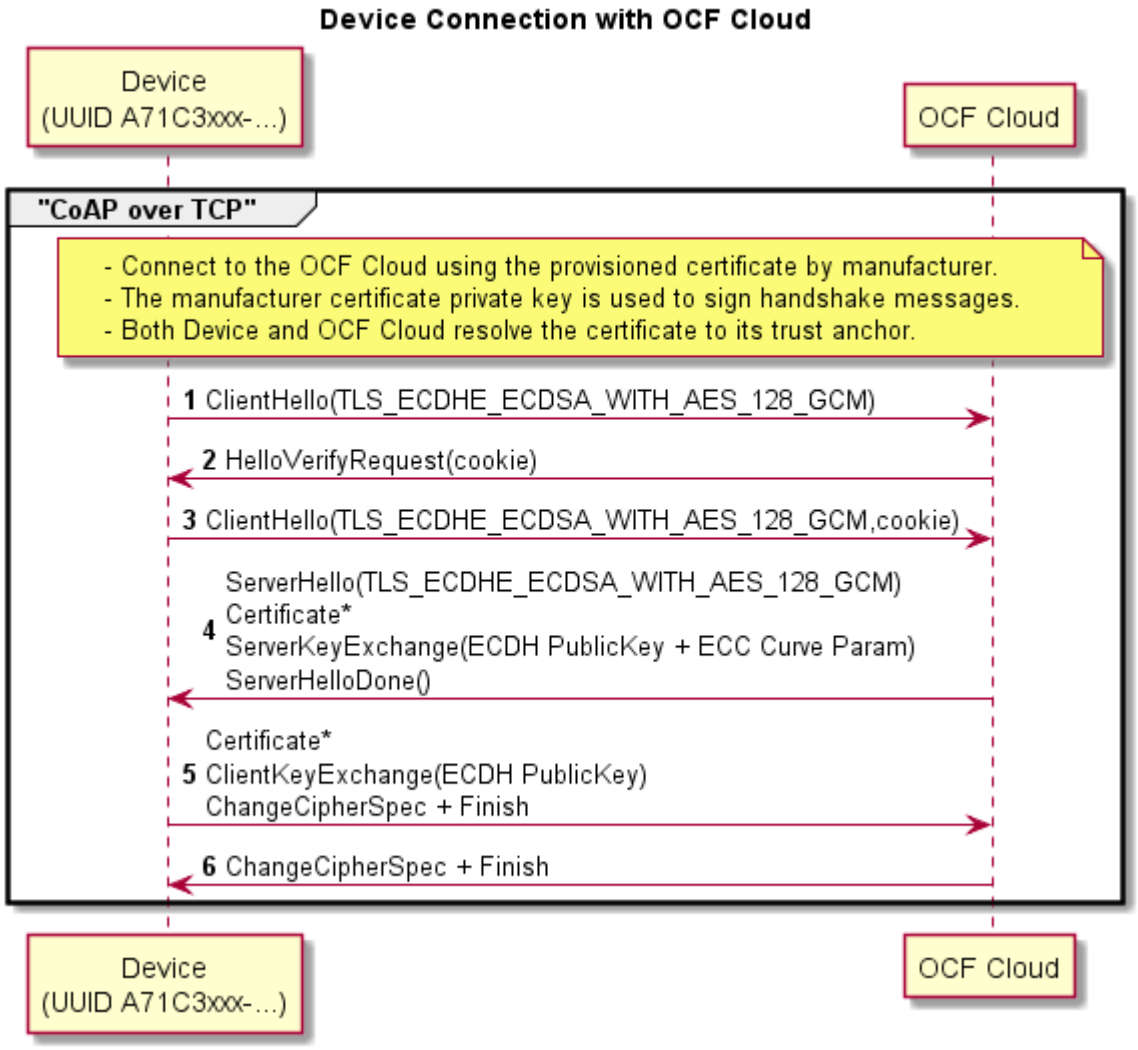



Table 2 – Device connection with the OCF Cloud flow

Steps	Description
1 - 6	TLS connection between the OCF Cloud and Device. The Device's manufacturer certificate may contain data attesting to the Device hardening and security properties

7.3 Security Considerations

When an OCF Server receives a request sent via the OCF Cloud, then the OCF Server permits that request using the identity of the OCF Cloud rather than the identity of the OCF Client. If there is no mechanism through which the OCF Cloud permits only those interactions which the user intends between OCF Clients and OCF Server via the OCF Cloud, and denies all other interactions, then OCF Clients might get elevated privileges by submitting a request via the OCF Cloud. This is highly undesirable from the security perspective. Consequently, OCF Cloud implementations are expected to provide some mechanism through which the OCF Cloud prevents OCF Clients getting elevated privileges when submitting a request via the OCF Cloud. In the present document release, the details of the mechanism are left to the implementation.

The security considerations about the manufacturer certificate as described in clause 7.3.6.5 of OCF Security are also applicable in the Device authentication with the OCF Cloud.

The Device should validate the OCF Cloud's TLS certificate as defined by IETF RFC 6125 and in accordance with its requirements for Server identity authentication.

The "uid" and "di" Property Value of "/oic/d" Resource may be considered personally identifiable information in some regulatory regions, and the OCF Cloud is expected to provide protections appropriate to its governing regulatory bodies.

8 Message integrity and confidentiality

8.1 Cloud Session Semantics

The messages between the OCF Cloud and Device shall be exchanged only if the Device and OCF Cloud authenticate each other as described in 7. The asymmetric cipher suites as described in 8.2 shall be employed for establishing a secured session and for encrypting/decrypting between the OCF Cloud and the Device. The OCF Endpoint sending the message shall encrypt and authenticate the message using the cipher suite as described in 8.2 and the OCF Endpoint shall verify and decrypt the message before processing it.

8.2 Cipher suites for OCF Cloud Credentials

All Devices supporting OCF Cloud Certificate Credentials shall implement:

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

All Devices supporting OCF Cloud Certificate Credentials should implement:

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

9 Security resources

9.1 Account Resource

The Account Resource specifies the Properties based on IETF RFC 6749 Access Token based account creation. The mechanism to obtain credentials is described in Clause 6. The Account Resource is used for Device Registration. The Account Resource is instantiated on the OCF Cloud as "oic/sec/account" SVR and is used by cloud-enabled Devices to register with the OCF Cloud. It should be only accessible on a secure channel; non-secure channel should not be able access this Resource.

During the Device Registration process, an OCF Cloud can provide a distinct URI of another OCF Cloud ("redirected-to" OCF Cloud). Both initial and redirected-to OCF Clouds are expected to belong to the same Vendor; they are assumed to have the same UUID and are assumed to have an Out-of-Band Communication Channel established. Device does not have to perform the Device Registration on the redirected-to OCF Cloud and the OCF Cloud may ignore such attempts. Redirected-to OCF Cloud is expected to accept the Access Token, provided to the Device by the initial OCF Cloud.

The RETRIEVE operation on OCF Cloud’s "/oic/sec/account" Resource is not allowed and the OCF Cloud is expected to reject all attempts to perform such operation.

The UPDATE operation on the OCF Cloud's "/oic/sec/account" Resource behaves as follows:

– A Device intending to register with the OCF Cloud shall send UPDATE with following Properties "di" ("di" Property Value of "/oic/d" Resource), and "accesstoken" as configured by the Mediator ("at" Property Value of "oic.r.coapcloudconf" Resource). The OCF Cloud verifies it is the same "accesstoken" which was assigned to the Mediator for the corresponding "di" Property Value. The "accesstoken" is the permission for the Device to access the OCF Cloud. If the "apn" was included when the Mediator UPDATED the "oic.r.coapcloudconf" Resource, the Device shall also include "authprovider" Property when registering with the OCF Cloud. If no "apn" is specified, then the "authprovider" Property shall not be included in the UPDATE request.

– OCF Cloud returns "accesstoken", "uid", "refreshtoken", and "expiresin" It may also return "redirecturi". Received "accesstoken" is to be treated by Device as an Access Token with "Bearer" token type as defined in IETF RFC 6750. This "accesstoken" shall be used for the following Account Session start using "oic/sec/session" SVR. Received "refreshtoken" is to be treated by Device as a Refresh Token as defined in IETF RFC 6749. The Device stores the OCF Cloud's Response values. If "redirecturi" is received, Device shall use received value as a new OCF Cloud URI instead of "cis" Property Value of "oic.r.coapcloudconf" Resource for further connections.

The DELETE operation on the OCF Cloud’s "/oic/sec/account" Resource should behave as follows:

– To deregister with the OCF Cloud, a DELETE operation shall be sent with the "accesstoken" and either "uid", or "di" to be deregistered with the OCF Cloud. On DELETE with the OCF Cloud, the Device should also delete values internally stored. Once deregister with an OCF Cloud, Device can connect to any other OCF Cloud. Device deregistered need to go through the steps in 6 again to be registered with the OCF Cloud.

Format of "oic.r.account" Resource is defined in Table 3.

Table 3 – Definition of the "oic.r.account" Resource

Fixed URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
/oic/sec/account	Account	oic.r.account	oic.if.baseline	Resource used for a device to add itself under a given credential	N/A

Table 4 defines the Properties of "oic.r.account ".

Table 4 – Properties of the "oic.r.account" Resource

Property Title	Property Name	Value Type	Value Rule	Access Mode	Mandatory	Description
Device ID	di	string	uuid	W	Yes	Unique Device identifier. Format pattern according to IETF RFC 4122.
Authorization Provider Name	authprovider	string	N/A	W	No	The name of Authorization Provider through which Access Token was obtained.
Access Token	accesstoken	string	Non-empty string	W	Yes	Access Token used to authorize and associate the TLS connection for communication with the OCF Cloud with the Device ID, or the Authorization Code which is then verified and exchanged for the Access Token during Device Registration.
Access Token	accesstoken	string	Non-empty string	R	Yes	Access Token used to authorize and associate the TLS connection for communication with the OCF Cloud with the Device ID.
Refresh Token	refreshtoken	string	Non-empty string	R	Yes	Refresh token can be used to refresh the Access Token before getting expired.
Token Expiration	expiresin	integer	-	R	Yes	Access Token life time in seconds (-1 if permanent).
User ID	uid	string	uuid	R	Yes	Unique OCF Cloud User identifier. Format pattern according to IETF RFC 4122.
Redirect URI	redirecturi	string	-	R	No	Using this URI, the Client needs to reconnect to a redirected OCF Cloud. If provided, this value shall be used by the Device instead of Mediator-provided URI during the Device Registration.

9.2 Account Session resource

The "/oic/sec/session" Resource hosted on the OCF Cloud is used for creating connections with the OCF Cloud subsequent to Device registration though "/oic/sec/account" Resource. The "/oic/sec/session" Resource requires the device ID, User ID and Access Token which are stored securely on the Device.

The "/oic/sec/session" Resource is exposed by the OCF Cloud. It should be only accessible on a secure channel; non-secure channel cannot access this Resource.

The RETRIEVE operation on OCF Cloud's "/oic/sec/session" Resource is not allowed and the OCF Cloud is expected to reject all attempts to perform such operation.

The UPDATE operation is defined as follows for OCF Cloud's "/oic/sec/session" Resource:

– The Device connecting to the OCF Cloud shall send an UPDATE request message to the OCF Cloud's "/oic/sec/session" Resource. The message shall include the "di" Property Value of "/oic/d" Resource and "uid", "login" Value ("true" to establish connection; "false" to disconnect) and "accesstoken" as returned by OCF Cloud during Device Registration. The OCF Cloud verifies it is the same Access Token which was returned to the Device during Device Registration process or during Token Refresh. If Device was attempting to establish the connection and provided values were verified as correct by the OCF Cloud, OCF Cloud sends a response with remaining lifetime of the associated Access Token ("expiresin" Property Value).

"oic.r.session" Resource is defined in Table 5.

549 **Table 5 – Definition of the "oic.r.session" Resource**

Fixed URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
/oic/sec/session	Account Session	oic.r.session	oic.if.baseline	Resource that enables a device to manage its session using login or logout	N/A

550 Table 6 defines the Properties of "oic.r.session".

551 **Table 6 – Properties of the "oic.r.session" Resource**

Property Title	Property Name	Value Type	Value Rule	Access Mode	Mandatory	Description
User ID	uid	string	uuid	W	Yes	User ID provided by Device Registration process. Format pattern according to IETF RFC 4122.
Device ID	di	string	uuid	W	Yes	Unique device id registered for a Device.Format pattern according to IETF RFC 4122.
Access Token	accesstoken	string	A string of at least one character	W	Yes	Access Token used to authorize and associate the TLS connection for communication with the OCF Cloud with the Device ID
Login Status	login	boolean	N/A	W	Yes	Action for the request: true = login, false = logout
Token Expiration	expiresin	integer	N/A	R	Yes	Remaining Access Token life time in seconds (-1 if permanent) This Property is only provided to Device during connection establishment (when "login" Property Value equals "true"), it's not available otherwise

552 ### 9.3 Account Token Refresh Resource

553 The "/oic/sec/tokenrefresh" Resource is used by the Device for refreshing the Access Token.

554 The "/oic/sec/tokenrefresh" Resource is hosted by the OCF Cloud. It should be only accessible on
555 a secure channel; non-secure channel cannot access this Resource.

556 The Device should use "/oic/sec/tokenrefresh" to refresh the Access Token with the OCF Cloud,
557 when the time specified in "expiresin" is near.

558 The RETRIEVE operation on OCF Cloud's "/oic/sec/ tokenrefresh" Resource is not allowed and the
559 OCF Cloud is expected to reject all attempts to perform such operation.

560 The UPDATE operation is defined as follows for "/oic/sec/tokenrefresh" Resource

561 – The Device attempting to refresh the Access Token shall send an UPDATE request message
562 to the OCF Cloud's "/oic/sec/tokenrefresh" Resource. The message shall include the "di"
563 Property Value of "/oic/d" Resource, "uid" and "refreshtoken", as returned by OCF Cloud.

564 – OCF Cloud response is expected to include a "refreshtoken", new "accesstoken", and
565 "expiresin". Received "accesstoken" is to be treated by Device as an Access Token with
566 "Bearer" token type as defined in IETF RFC 6750. This Access Token is the permission for the
567 Device to access the OCF Cloud. Received "refreshtoken" is to be treated by Device as a
568 Refresh Token as defined in IETF RFC 6749. Received "refreshtoken" may be the new Refresh

Token or the same one as provided by the Device in the UPDATE request. In case when new distinct "refreshtoken" is provided by the OCF Cloud, the Device shall discard the old value. The OCF Cloud's response values "refreshtoken", "acesstoken" and "expiresin" are securely stored on the Device.

"oic.r.tokenrefresh" Resource is defined in Table 7.

Table 7 – Definition of the "oic.r.tokenrefresh" Resource

Fixed URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
/oic/sec/tokenrefresh	Token Refresh	oic.r.tokenrefresh	oic.if.baseline	Resource to manage the access-token using refresh token	N/A

Table 8 defines the Properties of "oic.r.tokenrefresh".

Table 8 – Properties of the "oic.r.tokenrefresh" Resource

Property Title	Property Name	Value Type	Value Rule	Access Mode	Mandatory	Description
User ID	uid	string	uuid	W	Yes	User ID provided by Sign-up process. Format pattern according to IETF RFC 4122.
Device ID	di	string	uuid	W	Yes	Unique device id registered for an OCF Cloud User account. Format pattern according to IETF RFC 4122.
Refresh Token	refreshtoken	string	A string of at least one character	RW	Yes	Refresh token can be used to refresh the Access Token before getting expired.
Access Token	accesstoken	string	A string of at least one character	R	Yes	Access Token used to authorize and associate the TLS connection for communication with the OCF Cloud with the Device ID.
Token Expiration	expiresin	integer	-	R	Yes	Access Token life time in seconds (-1 if permanent).

10 Security hardening guidelines

10.1 Security hardening guidelines general

In addition to the Sensitive Data list outlined in Table 75 of Security Document, any Device implementing OCF Cloud connection capabilities should also provide reasonable protection for the information in Table 9.

Table 9 – Sensitive Data related to OCF Cloud

Data	Integrity protection	Confidentiality protection
OCF Cloud URL	Yes	Not required
OCF Cloud Identity	Yes	Not required

586

587

Annex A
(normative)
Resource Type definitions

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

Table A.1 – Alphabetized list of security resources

Friendly Name (informative)	Resource Type (rt)	Clause
Account	oic.r.account	A.1
Account Session	oic.r.session	A.2
Account Token Refresh	oic.r.tokenrefresh	A.3

A.1 Account Token

A.1.1 Introduction

Sign-up using generic account provider.

A.1.2 Well-known URI

/oic/sec/account

A.1.3 Resource type

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

A.1.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Account Token",
    "version": "20190111",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.md",
      "x-copyright": "copyright 2016-2017, 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/sec/account" : {
      "post": {
        "description": "Sign-up using generic account provider.\n",
        "parameters": [
          {"$ref": "#/parameters/interface"},
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/Account-request" },
            "x-example":
              {
                "di" : "9cfbeb8e-5a1e-4d1c-9d01-00c04fd430c8",
                "authprovider" : "github",
                "accesstoken" : "8802f2eaf8b5e147a936"
              }
```



```
            }
          ],
          "responses": {
              "204": {
                "description" : "2.04 Changed respond with required and optional information\n",
                "x-example":
                  {
                    "rt": ["oic.r.account"],
                    "accesstoken" : "0f3d9f7fe5491d54077d",
                    "refreshtoken" : "00fe4644a6fbe5324eec",
                    "expiresin" : 3600,
                    "uid" : "123e4567-e89b-12d3-a456-d6e313b71d9f",
                    "redirecturi" : "coaps+tcp://example.com:443"
                  },
                "schema": { "$ref": "#/definitions/Account-response" }
              }
          }
        },
        "delete": {
          "description": "Delete a device. This also removes all resources in the device on cloud
side.\nexample: /oic/account?di=9cfbeb8e-5a1e-4d1c-9d01-
00c04fd430c8&accesstoken=0f3d9f7fe5491d54077d\n",
          "parameters": [
            {"$ref": "#/parameters/interface"}
          ],
          "responses": {
              "202": {
                "description" : "2.02 Deleted response informing the device is successfully
deleted.\n"
              }
          }
        }
      }
    },
    "parameters": {
      "interface" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.baseline"]
      }
    },
    "definitions": {
      "Account-request" : {
        "properties": {
          "authprovider": {
            "description": "The name of Authorization Provider through which Access Token was
obtained",
            "type": "string"
          },
          "accesstoken" : {
            "description": "Access-Token used for communication with OCF Cloud after account
creation",
            "pattern": "(?!$|\\s+).*",
            "type": "string"
          },
          "di": {
            "description": "Format pattern according to IETF RFC 4122.",
            "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
9]{12}$",
            "type": "string"
          }
        },
        "type" : "object",
        "required": ["di", "accesstoken"]
      },
      "Account-response": {
        "properties": {
          "expiresin" : {
            "description": "Access-Token remaining life time in seconds (-1 if permanent)",
            "readOnly": true,
```



```
            "type": "integer"
          },
          "rt": {
            "description": "Resource Type of the Resource",
            "items": {
              "maxLength": 64,
              "type": "string",
              "enum" : ["oic.r.account"]
            },
            "minItems": 1,
            "maxItems": 1,
            "readOnly": true,
            "type": "array"
          },
          "refreshtoken" : {
            "description": "Refresh token can be used to refresh the Access Token before getting
    expired",
            "pattern": "(?!$|\\s+).*",
            "readOnly": true,
            "type": "string"
          },
          "uid" : {
            "description": "Format pattern according to IETF RFC 4122.",
            "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
    9]{12}$",
            "type": "string"
          },
          "accesstoken" : {
            "description": "Access-Token used for communication with cloud after account creation",
            "pattern": "(?!$|\\s+).*",
            "type": "string"
          },
          "n": {
            "$ref":
    "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
    schema.json#/definitions/n"
          },
          "id": {
            "$ref":
    "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
    schema.json#/definitions/id"
          },
          "redirecturi" : {
            "description": "Using this URI, the Client needs to reconnect to a redirected OCF Cloud.
    If provided, this value shall be used by the Device instead of Mediator-provided URI during the
    Device Registration.",
            "readOnly": true,
            "type": "string"
          },
          "if": {
            "description": "The interface set supported by this resource",
            "items": {
              "enum": [
                "oic.if.baseline"
              ],
              "type": "string"
            },
            "minItems": 1,
            "maxItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          }
        },
        "type" : "object",
        "required": ["accesstoken", "refreshtoken", "expiresin", "uid"]
      }
    }
    }
```


A.1.5 Property definition

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

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

Property name	Value type	Mandatory	Access mode	Description
di	string	Yes	Write Only	Unique Device identifier. Format pattern according to IETF RFC 4122.
authprovider	string	No	Write Only	The name of Authorization Provider through which Access Token was obtained.
accesstoken	string	Yes	Write Only	Access Token used to authorize and associate the TLS connection for communication with the OCF Cloud with the Device ID, or the Authorization Code which is then verified and exchanged for the Access Token during Device Registration.
id	multiple types: see schema	No	Read Write	
refreshtoken	string	Yes	Read Only	Refresh token can be used to refresh the Access Token before getting expired.
rt	array: see schema	No	Read Only	Resource Type of the Resource
accesstoken	string	Yes	Read Only	Access Token used to authorize and associate the TLS connection for communication with the OCF Cloud with the Device ID.
uid	string	Yes	Read Only	Unique OCF Cloud User identifier. Format

				pattern according to IETF RFC 4122.
expiresin	integer	Yes	Read Only	Access-Token life time in seconds (-1 if permanent)
if	array: see schema	No	Read Only	The interface set supported by this resource
redirecturi	string	No	Read Only	Using this URI, the Client needs to reconnect to a redirected OCF Cloud. If provided, this value shall be used by the Device instead of Mediator-provided URI during the Device Registration.
n	multiple types: see schema	No	Read Write	

A.1.6 CRUDN behaviour

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

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

Create	Read	Update	Delete	Notify
		post	delete	

A.2 Session

A.2.1 Introduction

Resource that manages the persistent session between a Device and OCF Cloud.

A.2.2 Well-known URI

/oic/sec/session

A.2.3 Resource type

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

A.2.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Session",
    "version": "v1.0-20181001",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.md",
      "x-copyright": "copyright 2016-2017, 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/sec/session" : {
      "post": {
        "description": "Resource that manages the persistent session between a Device and OCF
Cloud.",
        "parameters": [
          {"$ref": "#/parameters/interface"},
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/Account-Session-Request" },
            "x-example":
              {
                "uid" : "123e4567-e89b-12d3-a456-d6e313b71d9f",
                "di" : "9cfbeb8e-5a1e-4d1c-9d01-00c04fd430c8",
                "accesstoken" : "0f3d9f7fe5491d54077d",
                "login" : true
              }
          }
        ],
        "responses": {
            "204": {
              "description" : "",
              "x-example":
                {
                  "rt": ["oic.r.session"],
                  "expiresin" : 3600
                },
              "schema": { "$ref": "#/definitions/Account-Session-Response" }
            }
        }
      }
    }
  },
  "parameters": {
    "interface" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.baseline"]
    }
  },
  "definitions": {
    "Account-Session-Request" : {
      "properties": {
        "uid": {
          "description": "Format pattern according to IETF RFC 4122.",
          "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
9]{12}$",
          "type": "string"
        },
        "di": {
          "description": "The Device ID\nFormat pattern according to IETF RFC 4122.",
          "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
9]{12}$",
          "type": "string"
        },
        "accesstoken": {
          "description": "Access-Token used to grant access right for the Device to sign-in.",
          "pattern": "(?!$|\\s+).*",
          "type": "string"
        },
        "login": {
          "description": "Action for the request: true = login, false = logout.",
          "type": "boolean"
```



```
        }
      },
      "type" : "object",
      "required": ["uid", "di", "accesstoken", "login"]
    },
    "Account-Session-Response" : {
      "properties": {
        "expiresin": {
          "description": "Access-Token remaining life time in seconds (-1 if permanent).",
          "readOnly": true,
          "type": "integer"
        },
        "rt": {
          "description": "Resource Type of the Resource.",
          "items": {
            "maxLength": 64,
            "type": "string",
            "enum": ["oic.r.session"]
          },
          "minItems": 1,
          "readOnly": true,
          "type": "array"
        },
        "n": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
        },
        "id": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
        },
        "if": {
          "description": "The interface set supported by this Resource.",
          "items": {
            "enum": [
              "oic.if.baseline"
            ],
            "type": "string"
          },
          "minItems": 1,
          "readOnly": true,
          "type": "array"
        }
      },
      "type" : "object",
      "required" : ["expiresin"]
    }
  }
}
```

A.2.5 Property definition

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

Table A.4 – The Property definitions of the Resource with type "rt" = "oic.r.session".

Property name	Value type	Mandatory	Access mode	Description
if	array: see schema	No	Read Only	The interface set supported by this Resource.
expiresin	integer	Yes	Read Only	Remaining Access Token life time in seconds (-1 if permanent). This Property is only

				provided to Device during connection establishment (when "login" Property Value equals "true"), it's not available otherwise.
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	
di	string	Yes	Write Only	Unique device id registered for a Device. Format pattern according to IETF RFC 4122.
accesstoken	string	Yes	Write Only	Access Token used to authorize and associate the TLS connection for communication with the OCF Cloud with the Device ID.
uid	string	Yes	Write Only	User ID provided by Device Registration process. Format pattern according to IETF RFC 4122.
login	boolean	Yes	Write Only	Action for the request: true = login, false = logout.

A.2.6 CRUDN behaviour

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

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

Create	Read	Update	Delete	Notify
		post		

A.3 Token Refresh

A.3.1 Introduction

Obtain fresh Access Tokenusing the refresh token, client should refresh Access Token before it expires.

A.3.2 Well-known URI

/oic/sec/tokenrefresh

A.3.3 Resource type

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

A.3.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Token Refresh",
    "version": "v1.0-20181001",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.md",
      "x-copyright": "copyright 2016-2017, 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/sec/tokenrefresh" : {
      "post": {
        "description": "Obtain fresh access-token using the refresh token, client should refresh
access-token before it expires.\n",
        "parameters": [
          {"$ref": "#/parameters/interface"},
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/TokenRefresh-Request" },
            "x-example":
              {
                "uid" : "123e4567-e89b-12d3-a456-d6e313b71d9f",
                "di" : "9cfbeb8e-5a1e-4d1c-9d01-00c04fd430c8",
                "refreshtoken" : "00fe4644a6fbe5324eec"
              }
          }
        ],
        "responses": {
            "204": {
              "description" : "2.04 Changed respond with new access-token.\n",
              "x-example":
                {
                  "rt": ["oic.r.tokenrefresh"],
                  "accesstoken" : "8ce598980761869837be",
                  "refreshtoken" : "d4922312b6df0518e146",
                  "expiresin" : 3600
                }
                ,
              "schema": { "$ref": "#/definitions/TokenRefresh-Response" }
            }
        }
      }
    }
  },
  "parameters": {
    "interface" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.baseline"]
    }
```



```
  },
  "definitions": {
    "TokenRefresh-Request" : {
      "properties": {
        "refreshtoken": {
          "description": "Refresh token received by account management or during token refresh
procedure.",
          "pattern": "(?!$|\\s+).*",
          "type": "string"
        },
        "uid": {
          "description": "Format pattern according to IETF RFC 4122.",
          "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
9]{12}$",
          "type": "string"
        },
        "di": {
          "description": "Format pattern according to IETF RFC 4122.",
          "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
9]{12}$",
          "type": "string"
        }
      },
      "type" : "object",
      "required": ["uid", "di", "refreshtoken"]
    },
    "TokenRefresh-Response" : {
      "properties": {
        "expiresin": {
          "description": "Access-Token life time in seconds (-1 if permanent).",
          "readOnly": true,
          "type": "integer"
        },
        "rt": {
          "description": "Resource Type of the Resource.",
          "items": {
            "maxLength": 64,
            "type": "string",
            "enum": ["oic.r.tokenrefresh"]
          },
          "minItems": 1,
          "readOnly": true,
          "type": "array"
        },
        "refreshtoken": {
          "description": "Refresh token received by account management or during token refresh
procedure.",
          "pattern": "(?!$|\\s+).*",
          "type": "string"
        },
        "accesstoken": {
          "description": "Granted Access-Token.",
          "pattern": "(?!$|\\s+).*",
          "readOnly": true,
          "type": "string"
        },
        "n": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
        },
        "id": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
        },
        "if" :
                {
          "description": "The interface set supported by this Resource.",
          "items": {
            "enum": [
```



```
                "oic.if.baseline"
              ],
              "type": "string"
            },
            "minItems": 1,
            "readOnly": true,
            "type": "array"
          }
        },
        "type" : "object",
        "required": ["accesstoken", "refreshtoken", "expiresin"]
      }
    }
}
```

A.3.5 Property definition

Table A.6 defines the Properties that are part of the "oic.r.tokenrefresh" Resource Type.

Table A.6 – The Property definitions of the Resource with type "rt" = "oic.r.tokenrefresh".

Property name	Value type	Mandatory	Access mode	Description
refreshtoken	string	Yes	Write Only	Refresh token can be used to refresh the Access Token before getting expired.
uid	string	Yes	Write Only	User ID provided by Sign-up process. Format pattern according to IETF RFC 4122.
di	string	Yes	Write Only	Unique device id registered for an OCF Cloud User account. Format pattern according to IETF RFC 4122.
if	array: see schema	No	Read Only	The interface set supported by this Resource.
expiresin	integer	Yes	Read Only	Access Token life time in seconds (-1 if permanent).
accesstoken	string	Yes	Read Only	Access Token used to authorize and associate the TLS connection for communication with the OCF Cloud with the Device ID.
refreshtoken	string	Yes	Read Only	Refresh token can be used to

				refresh the Access Token before getting expired.
n	multiple types: see schema	No	Read Write	
rt	array: see schema	No	Read Only	Resource Type of the Resource.
id	multiple types: see schema	No	Read Write	

A.3.6 CRUDN behaviour

Table A.7 defines the CRUDN operations that are supported on the "oic.r.tokenrefresh" Resource Type.

Table A.7 – The CRUDN operations of the Resource with type "rt" = "oic.r.tokenrefresh".

Create	Read	Update	Delete	Notify
		post		

