OPEN CONNECTIVITY FOUNDATION®

OCF 2.2.4 Specification Introduction and Overview

October 2021

Disclaimer Notice

This Specification Overview document is for informational purposes only. It has not been adopted in full or in part by the Open Connectivity Foundation, and should not be relied upon for any purpose other than for review of the information contained herein. This document is subject to change, and the Open Connectivity Foundation and its members reserve the right without notice to you to change any or all portions hereof, delete portions hereof, make additions hereto, discard this document in its entirety or otherwise modify this document at any time. You should not and may not rely upon the contents of this document in any way, including but not limited to for the development of any products or services. In order to be considered a certified product, among other requirements, a product or service must be compliant with Final Specification of the Open Connectivity Foundation. To the extent this document references all or portions of a Draft or Final Specification of the Open Connectivity Foundation, or information related thereto, such references are made solely for informational purposes. This document is not a Draft or Final Specification of the Open Connectivity Foundation. Neither this document nor any of its contents is subject to or related to any licensing grants or commitments contained in the Open Connectivity Foundation Intellectual Property Rights (IPR) Policy or elsewhere. This document may contain or make reference to logos, brands, names, or other works of Open Connectivity Foundation, its members, or other third parties. The OCF logo is a trademark of Open Connectivity Foundation, Inc. in the United States or other countries. Other brands, names, or works contained herein may be claimed as the property of others. Any copying or other form of reproduction and/or distribution of this document or these works is strictly prohibited.

OPEN CONNECTIVITY FOUNDATION®

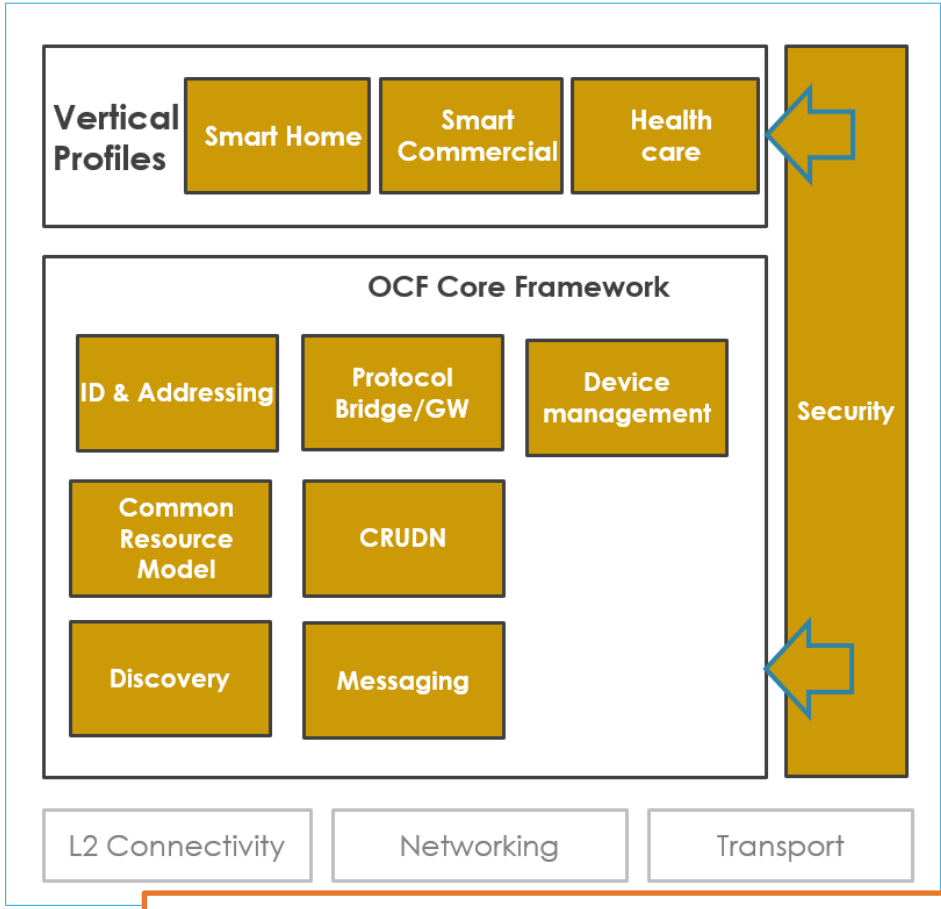
OCF Specification Fundamentals
Core Framework

OCF 2.2.4 Release

October 2021

Core Framework Fundamentals Building Blocks

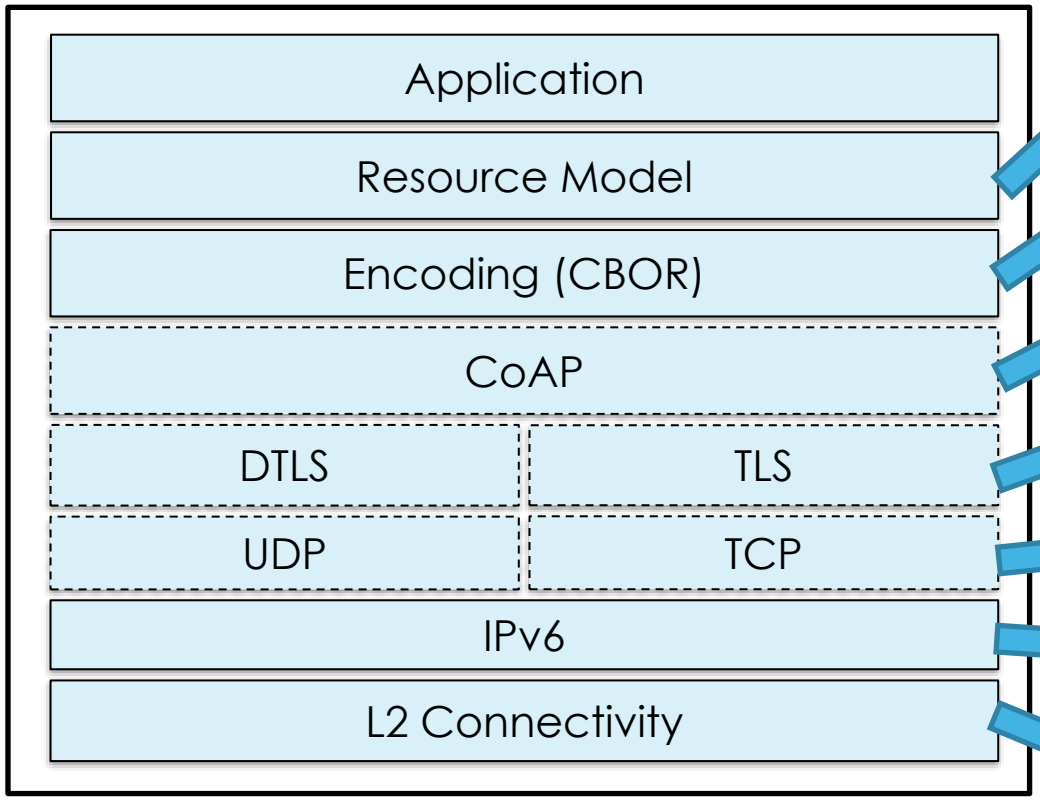
- Enable the development of vertical profiles (e.g. Smart Home, Smart Commercial) while maintaining fundamental interoperability via an Architecture that is scalable from resource constrained devices to resource rich devices



1. **Discovery:** Common method for device discovery (Multicast CoAP to All OCF Nodes Address)
2. **Messaging:** Constrained device support as default (IETF CoAP) as well as protocol translation via bridges
3. **Common Resource Model:** Real world entities defined as data models (resources)
4. **CRUDN:** Simple Request/Response mechanism with Create, Retrieve, Update, Delete, and Notify operations
5. **ID & Addressing:** Device Identifiers and OCF URIs (map to transport protocol)
6. **Protocol Bridge**: Framework provided by the Bridging Specification

Security is fundamental to the OCF ecosystem and applies to all elements

Core Framework Fundamentals Realized in the Protocol Stack



OCF Stack

- Common Resource Model defined using Open API 2.0
- Concise representation on the wire (binary encoded JSON)
- Constrained device support via use of CoAP as the transport layer
- Secure connection
- Connectionless or Connection Oriented
- IPv6 as the harmonization layer
- Agnostic of underlying physical layer technology (Wi-Fi, Ethernet, Thread)

OPEN CONNECTIVITY FOUNDATION®

Security Fundamentals of OCF

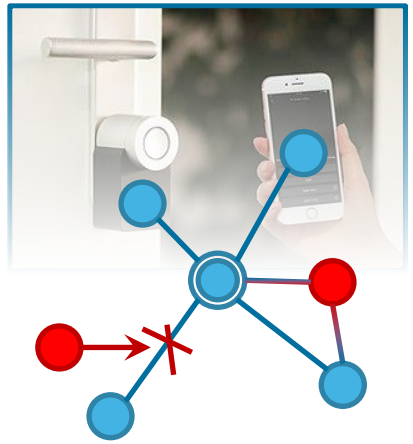
Secure device lifecycle

Secure operation

…is our end goal

Our objectives:

- ✓ Confidentiality
- ✓ Integrity
- ✓ Availability



The risks we face:

- Message interception/forgery
- Spoofing/privilege escalation
- Denial of service
- Device hijack

How do we get there?

(D)TLS sessions used for non-discovery connections

Authentication done as part of handshake

Randomized identity bound to credential:

- for PSK, identity is bound via 1-to-1 mapping
- for certificate, identity is in the Subject Name

Fine grained access control done for CRUDN operations on per-resource basis. Permission is denied by default

Wildcards and roles supported for scalability

* (D)TLS = (Datagram) Transport Layer Security
* PSK = Pairwise Secret Key
* CRUDN = Create, Retrieve, Update, Delete, Notify

Secure device lifecycle

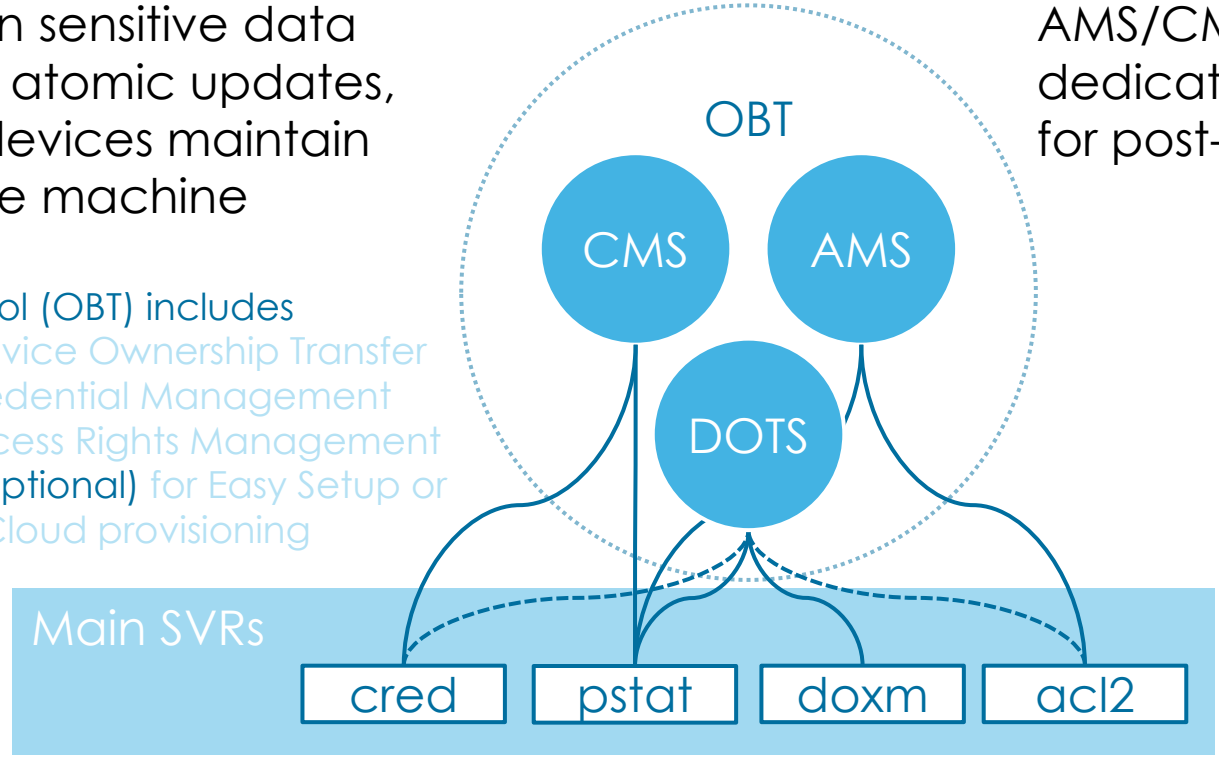
Secure provisioning

…is configuration of Security Virtual Resources by an authorized client, usually by the onboarding tool

Onboarding tool has full ownership and control over device during the ownership transfer procedure, which ends with installation of an Owner Credential

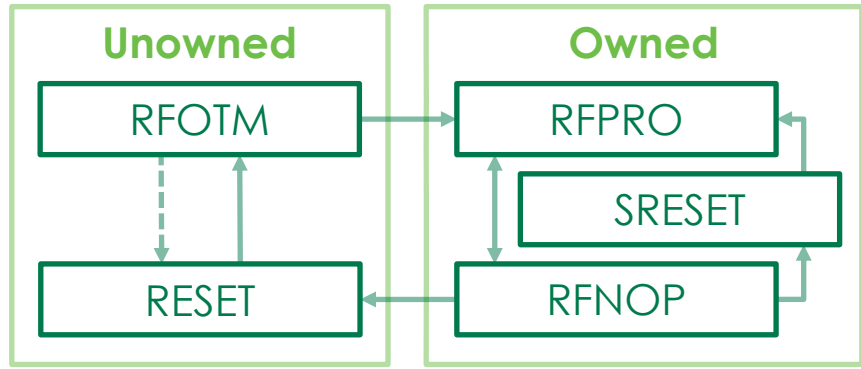
SVRs contain sensitive data and require atomic updates, so all OCF devices maintain internal state machine

Onboarding tool (OBT) includes
- DOTS for Device Ownership Transfer
- CMS for Credential Management
- AMS for Access Rights Management
- Mediator (optional) for Easy Setup or Device-to-Cloud provisioning



AMS/CMS have implicit control over their dedicated resources, and are responsible for post-onboarding provisioning

Device Provisioning Status (pstat) is updated by the OBT services to trigger the state machine transitions:



OPEN CONNECTIVITY FOUNDATION®

OCF Specification Overview Technical Principles

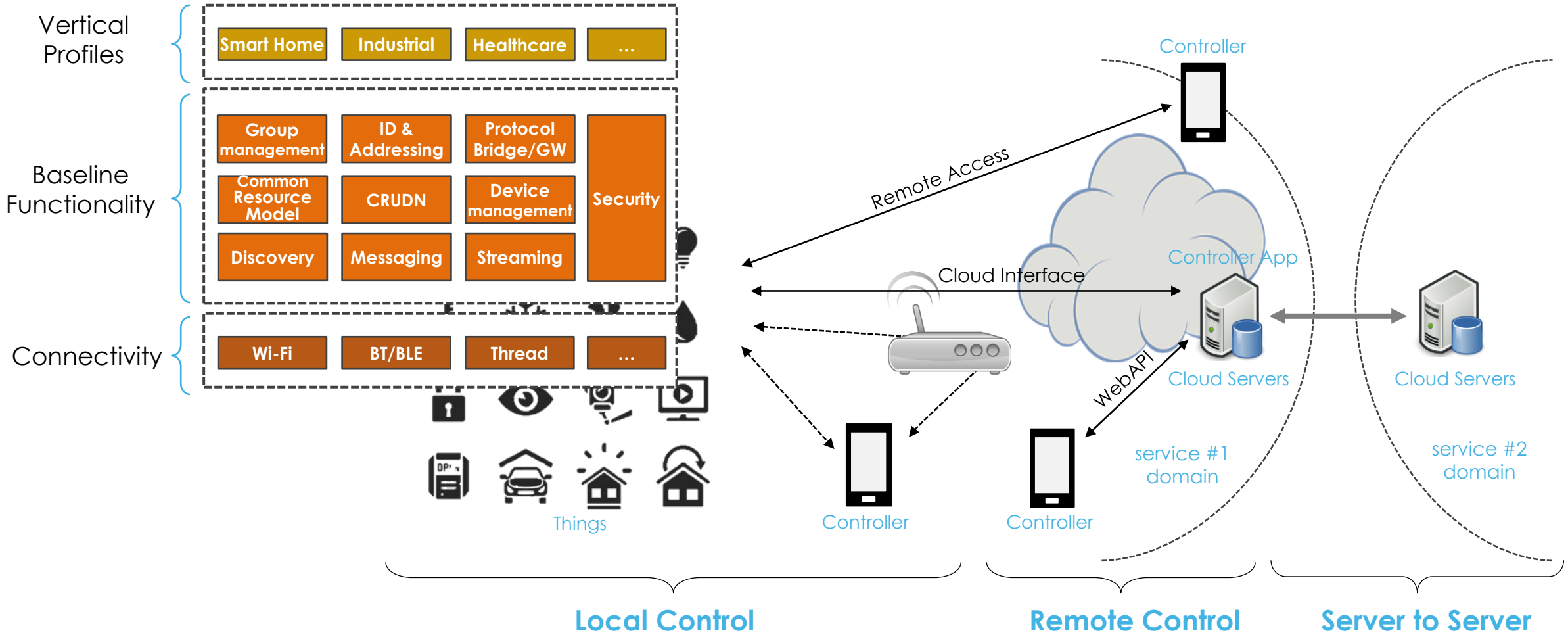
OCF 2.2.4 Release

October 2021

OPEN CONNECTIVITY FOUNDATION®

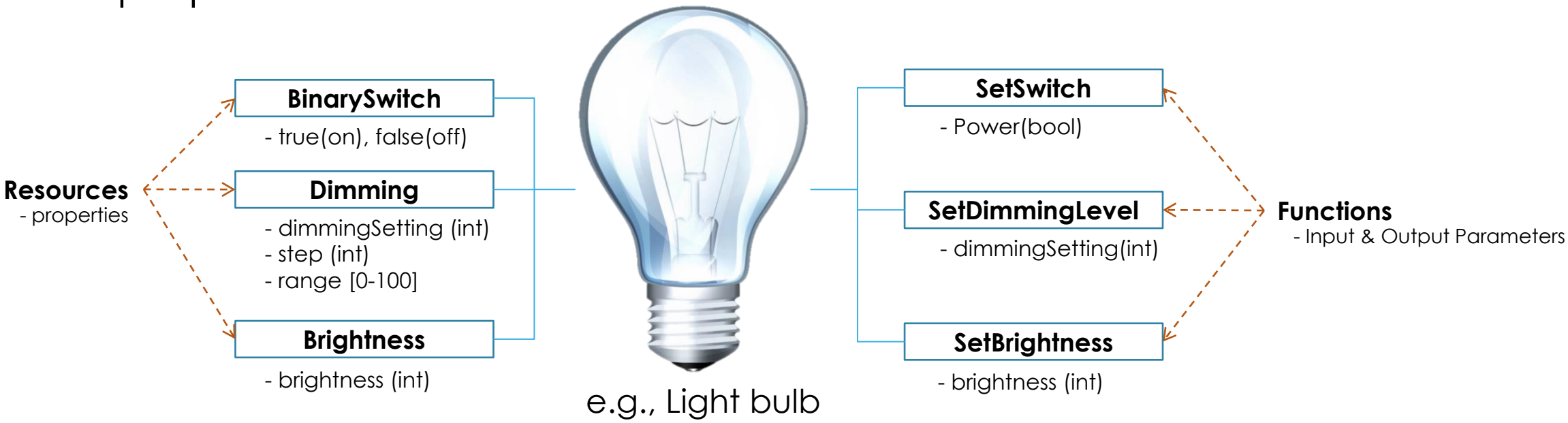
Technical Principles for an Internet of Things Ecosystem

Scope of IoT



Approaches to definition of various Things

- By defining resources of things and its properties
- By defining functions/operations of things



- (no Verbs) + Objects
  *Fixed set of verbs (CRUDN) from transport layer will be used
- Resource model in RESTful Architecture (e.g., W3C, CSEP, etc.)

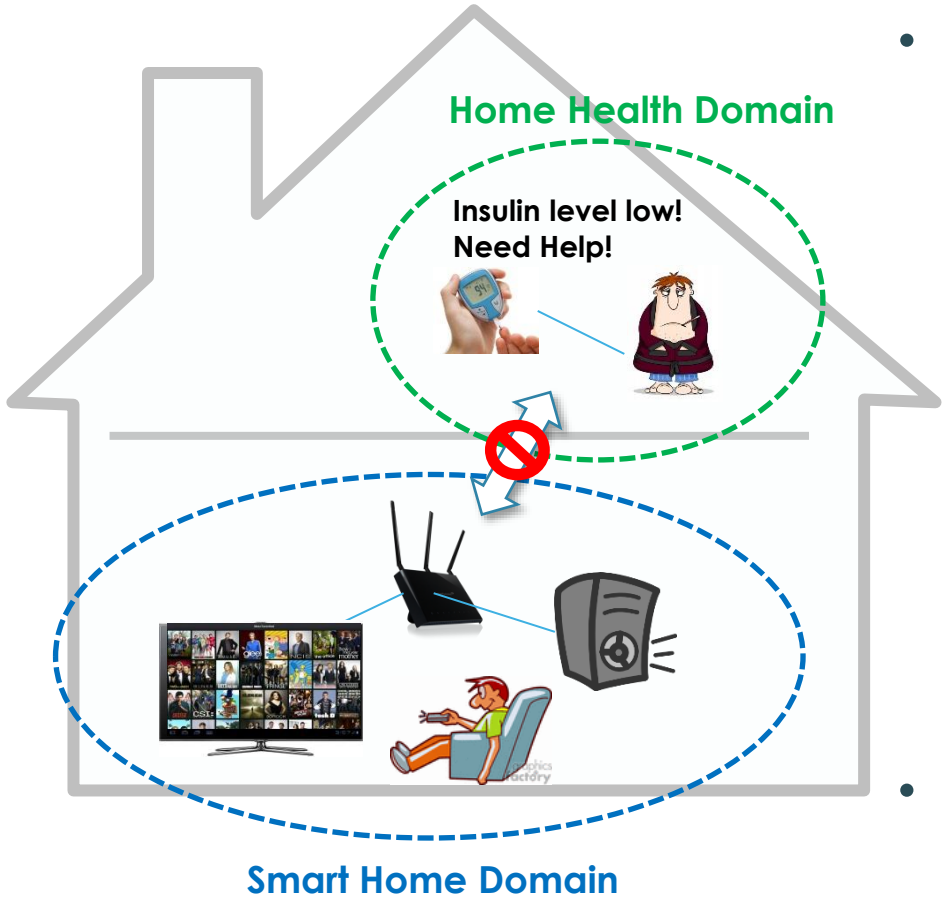
- (Verbs + Objects)
- RPC model

Support of Constrained Things
Class 2 Devices as Defined by RFC 7228

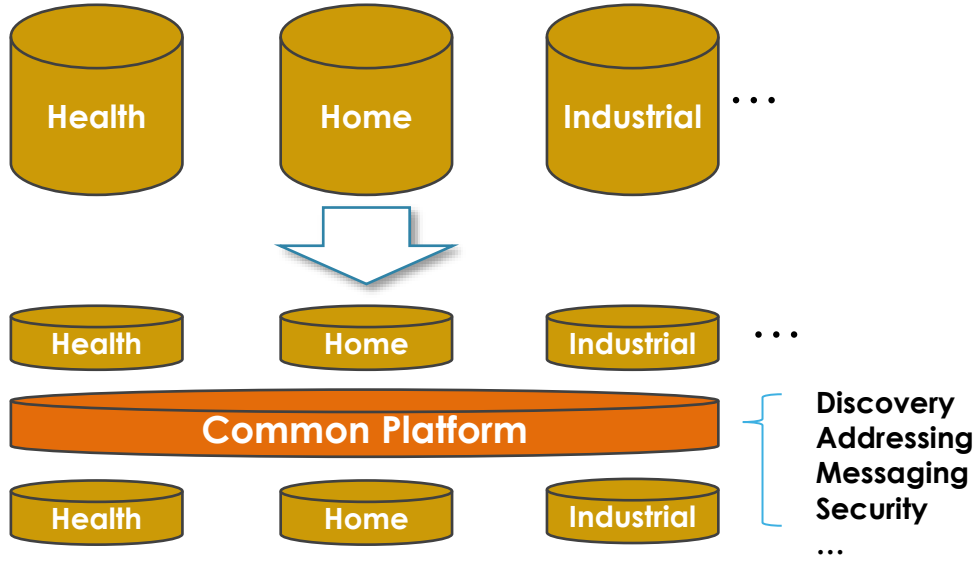
- Less overhead/ Less Traffic
  - Minimize CPU Load, Memory impacts, Traffic and Bandwidth
    - Compact header
    - Binary protocol
    - Compressed encoding of payload

- Low Complexity
    - Simple Resource Model
      > Short URI (Late Binding w/ resource type defined)
      > Broad and Shallow Hierarchy

Support of Multiple Verticals



- Legacy vertical services usually designed as silos
  → No common way to communicate among them

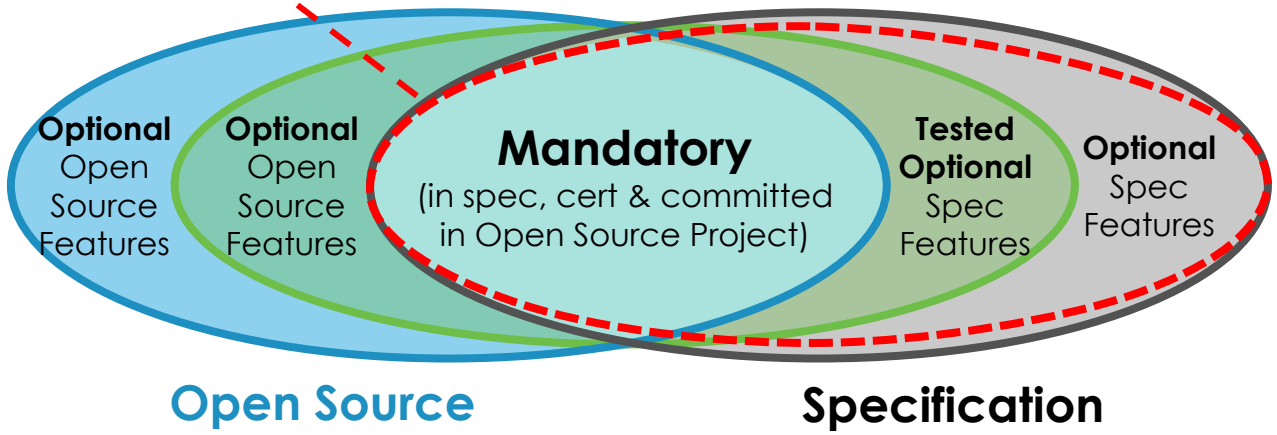


- A common platform provides a foundation for vertical services to collaborate and interwork by providing common services and data models

Conformance & Certification

- Conformance test - Each device proves conformance to specifications

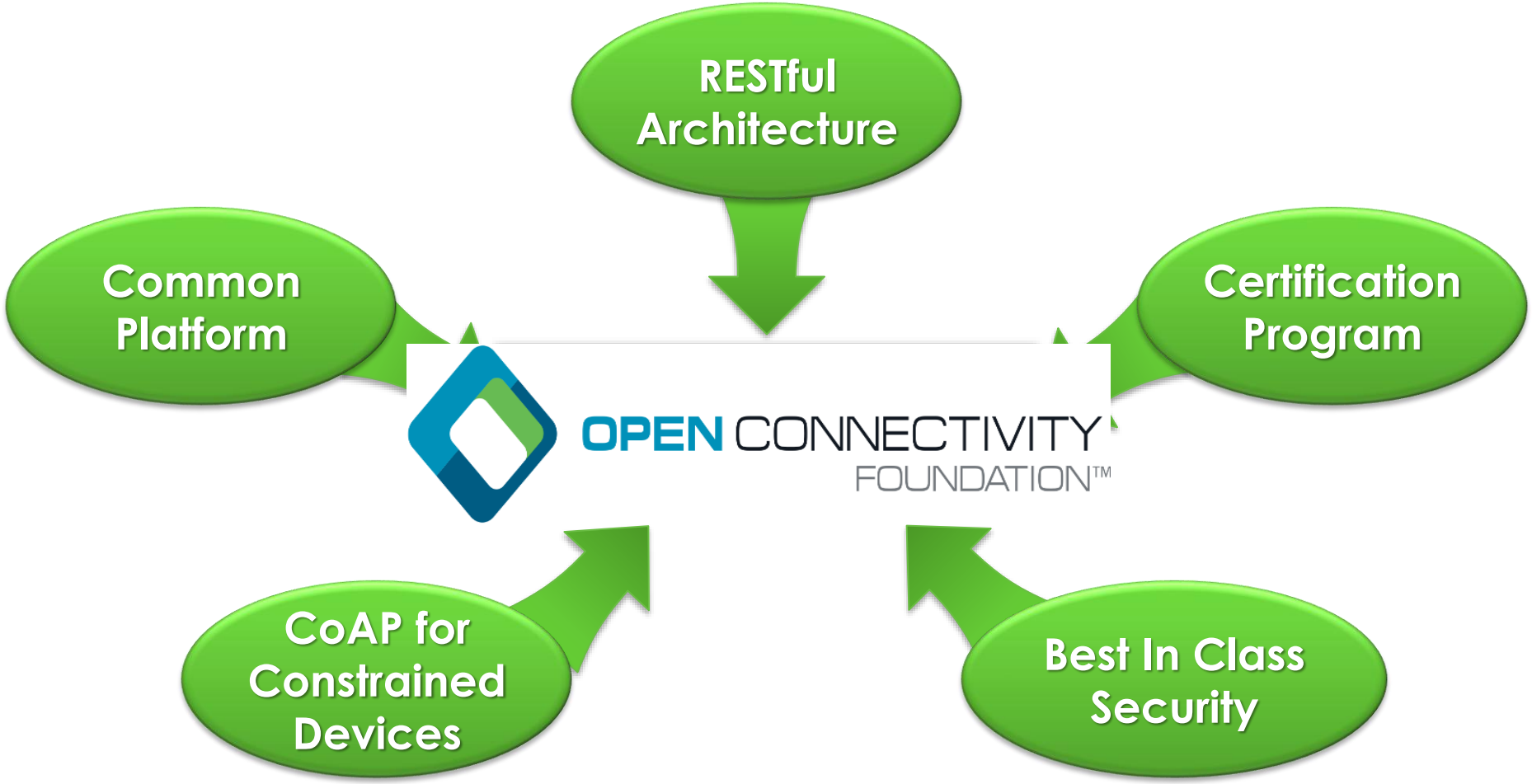
- Certification Scope



OPEN CONNECTIVITY FOUNDATION®

Introduction to the Open Connectivity Foundation

Introduction to OCF – Optimized for IoT



OCF Areas of Technology Development

- Core Architecture
  - Fundamental resource framework
  - Discovery
  - CRUDN
  - Transport Binds
- Security
- Resource Models (vertical agnostic)
- Device Profiles
  - Smart Home
  - Health
- Ecosystem Bridging
- Cloud Services

OCF Key Concepts (1/2)

- **Dedicated and optimized protocols for IoT (e.g. CoAP)**
  - Specific considerations for constrained devices
  - Fully compliant towards RESTful architecture
  - Built-in discovery and subscription mechanisms
- **Standards and Open Source to allow flexibility creating solutions**
  - Able to address all types of devices, form-factors, companies and markets with the widest possibility of options
  - Open Source is just one implementation to solve a problem

OCF Key Concepts (2/2)

- **Certification testing for interoperability**
  - Formal conformance testing for device validation to specifications
  - Plugfest testing for product interoperability
- **Certification and Logo program**
  - Products with the OCF Logo ensure OCF specifications are met
  - Logo reflects being part of an ecosystem of interoperable products

Licensing

- For OCF Core Technology
  - OCF Intellectual Property Rights (IPR) Policy : RAND-Z
  - IoTivity and plgd-dev Open Source : Apache 2.0
- For Domain Work Groups
  - Defined on a per WG basis
- The Licensing and IPR policy must be clear and readily understandable and ensures that these terms are offered by and to all member company IP holders
- This Licensing and IPR Policy enables growth of the entire market by attracting all levels of industry from start-ups to large enterprises
- Note: currently the domain working groups have the same IPR policy but have different scopes, e.g. smart home and smart commercial building.

OPEN CONNECTIVITY FOUNDATION®

OCF Specification Overview

OCF Deliverables

Normative Specifications

- See next slide

Resource Models via oneIoTa

- Domain agnostic resources
- Derived models for Ecosystem Mapping
  - To date: OCF-AllJoyn (CDM 16.4), OCF-oneM2M, OCF-Zigbee , OCF-UPlus, OCF-BLE, OCF-ZWave, OCF-EnOcean

Certification Procedures

- Test Policy (Certification Procedure Requirements Document)
- Test Plans and Test Cases (Certification Test Requirements Document)

Specification Structure

Infrastructure

- Core Mandatory Framework, Core Optional Extensions
- Security
- Bridging Framework
- Device Specification
- Cloud Services (Device to Cloud, Cloud to Cloud)

Resource Model

- Resource Specification (reflects OneIoTa content)
- Ecosystem Mapping Specifications (reflect OneIoTa content):
  - OCF to AllJoyn
  - OCF to BLE
  - OCF to EnOcean
  - OCF to oneM2M
  - OCF to UPlus
  - OCF to Zigbee
  - OCF to ZWave

Specification Location

Where can I find the specifications and Resource Type definitions?

OCF Specifications:

- https://openconnectivity.org/developer/specifications

Resource Type Definitions

- Core Resources: https://github.com/openconnectivityfoundation/core
- Core Extension Resources: https://github.com/openconnectivityfoundation/core-extensions
- Bridging Resources: https://github.com/openconnectivityfoundation/bridging
- Cloud Resources: https://github.com/openconnectivityfoundation/cloud-services
- Security Resources: https://github.com/openconnectivityfoundation/security-models
- Vertical Resources and Derived Models: https://oneiota.org/documents?filter%5Bmedia_type%5D=application%2Framl%2Byaml

OPEN CONNECTIVITY FOUNDATION®

OCF Specification Overview
Core Framework Specification &
Core Optional Framework Specification

OCF 2.2.4 Release

October 2021

Core Framework Topics Outline (1 of 2)

- Objectives
- RESTful Architecture
- OCF Roles
- Actions
- Resources
- Basic Operations
- Organization of an OCF Device
- OCF Specification Features
- Protocol Stack
- Endpoint Overview

Core Framework Topics Outline (2 of 2)

- Resource Discovery (CoAP Discovery)
- Block Transfer with CoAP Messaging
- Encoding Schemes
- Introspection
- OCF Over MQTT
- Collection Resources
- Atomic Measurement Resources
- Device Reset
- Versioning

Core Optional Framework Topics Outline

- Objectives
- Device/Platform Configuration
- Device Management
- Alerts
- Icons
- Scenes/Rules

OPEN CONNECTIVITY FOUNDATION®

Mandatory Core Technology
Core Framework Specification

Core Framework Objectives

- Core Framework Specifications Scope
  - Specifies the technical specification(s) comprising of the core architectural framework, messaging, interfaces and protocols based on approved use-case scenarios
  - Enables the development of vertical profiles (e.g. Smart Home, Health) on top of the core while maintaining fundamental interoperability
- Architect a core framework that is scalable from resource constrained devices to resource rich devices
- Reuse open standards solutions (e.g. IETF) where they exist
- Ensure alignment with IoTivity Lite open source releases

RESTful Architecture

Entity handler

Physical entity

Client

Server

REST Architecture Style

- Addressable resources
- A uniform, constrained interface
- Representation based manipulation
- Communicate statelessly
- Hypermedia State Engine

Request

Response

CRUD & N operation

```
{
  "n": "myRoomTemperature",
  "rt": "oic.r.temperature",
  "if": "oic.if.a",
  "id": "example_id_xyz",
  "temperature": 23,
  "units": "C",
  "setValue": 25
}
```

Resource (representation)

Resource (representation)

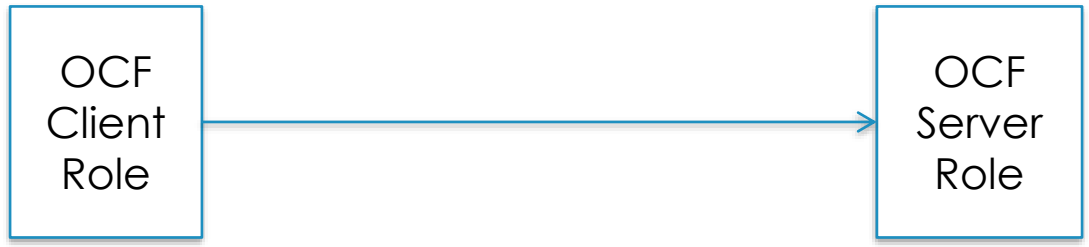
- RESTful Architecture (Representational State Transfer)
  - Resource based operation
    - Real world 'entity' is represented as 'Resource'
  - Resource manipulation via Request/ Response: CRUDN

OCF Actions

- A fundamental tenet OCF is the use of REST architectural style and the use of RESTful operations. However, there are use cases where a RESTful interaction is either awkward or otherwise problematic to realize.
- An action pattern provides support for where a Client needs to invoke an action or operation on a Server that is stateless or has an unknown state.
- The "action" Query Parameter may thus be realized by a Resource definition as part of UPDATE operation handling, where that Resource represents some subsequent action that is then taken.
- The generalized format of this Query Parameter is:
  - "action=<Resource defined value>".
  - E.g. POST /RemoteControlResURI?action=arrowup

OCF Roles

- Current OCF Architecture defines 2 logical roles that devices can take
  - OCF Server : A logical entity that exposes hosted resources, is discoverable, and responds to client initiated transactions
  - OCF Client : A logical entity that interacts with resources on an OCF Server via discovery and CRUDN actions
- An OCF Device implements one or both roles



Resources

- An OCF Server contains one or more Resources to describe a real world entity
- Each Resource contains Properties that describes an aspect that is exposed through a Resource including meta-information related to that Resource
- Each Resource contains Interface(s) that provides first a view into the Resource and then defines the requests and responses permissible on that view of the Resource

Resources - Interfaces

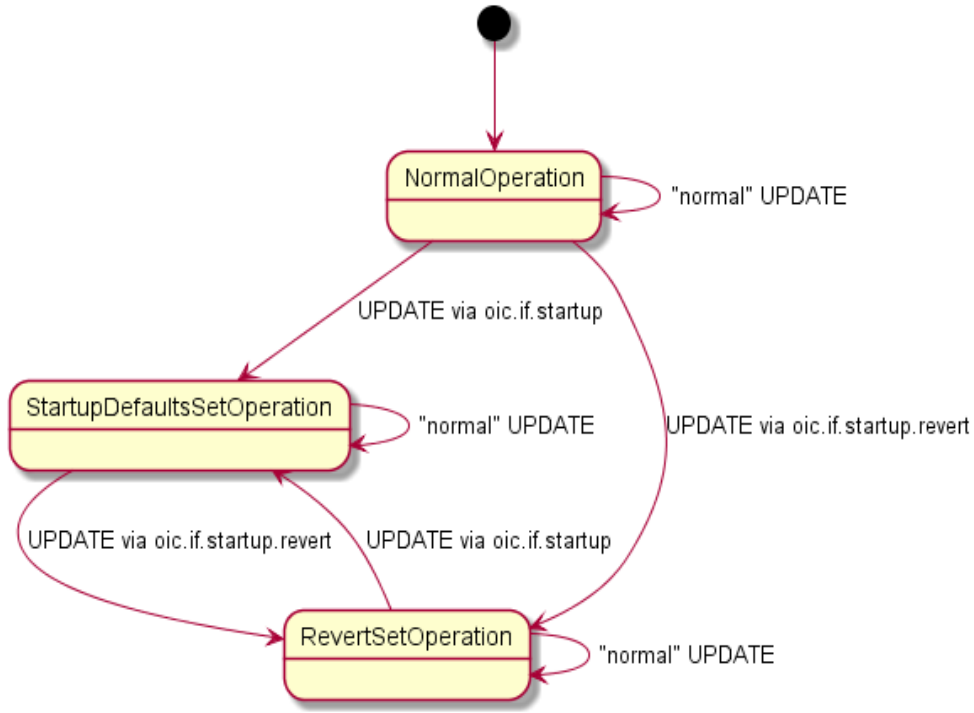
- All Resources have associated with them the Interfaces that they support. These describe the nature of the interaction, and shape the representation that will be provided by the Resource.
- All Resources have a Default Interface, defined in the OpenAPI definition:

```
"parameters": {
  "interface": {
    "in": "query",
    "name": "if",
    "type": "string",
    "enum": ["oic.if.a", "oic.if.baseline"]
  }
```

OCF Interface	Name	Description
baseline	"oic.if.baseline"	Defines a view into all Properties of a Resource including the Common Properties.
links list	"oic.if.ll"	Provides a view into Links in a Collection (Resource).
batch	"oic.if.b"	Provides the representation of the set of Resources linked within a collection
read-only	"oic.if.r"	Exposes the Properties of a Resource that may be read.
write-only	“oic.if.w"	Handles the Property Values of a Resource that may be ‘written'. When a RETRIEVE is issued on this Interface, only the Common Properties are provided.
read-write	"oic.if.rw"	Exposes only those Properties that may be read from a Resource during a RETRIEVE and that may be written to a Resource during and UPDATE.
actuator	"oic.if.a"	Used to read or write the Properties of an actuator.
sensor	"oic.if.s"	Used to read the Properties of a sensor.
create	"oic.if.create"	Used to create new Resources in a Collection.
property startup	"oic.if.startup"	Used to set the default values of Properties on a Resource that will be applied on power-up of the Device
Property revert	"oic.if.startup.revert"	Used to establish a revert state on a Resource such that on a power-up of the Device the target Properties are populated with the values to which they were set prior to the power-down.

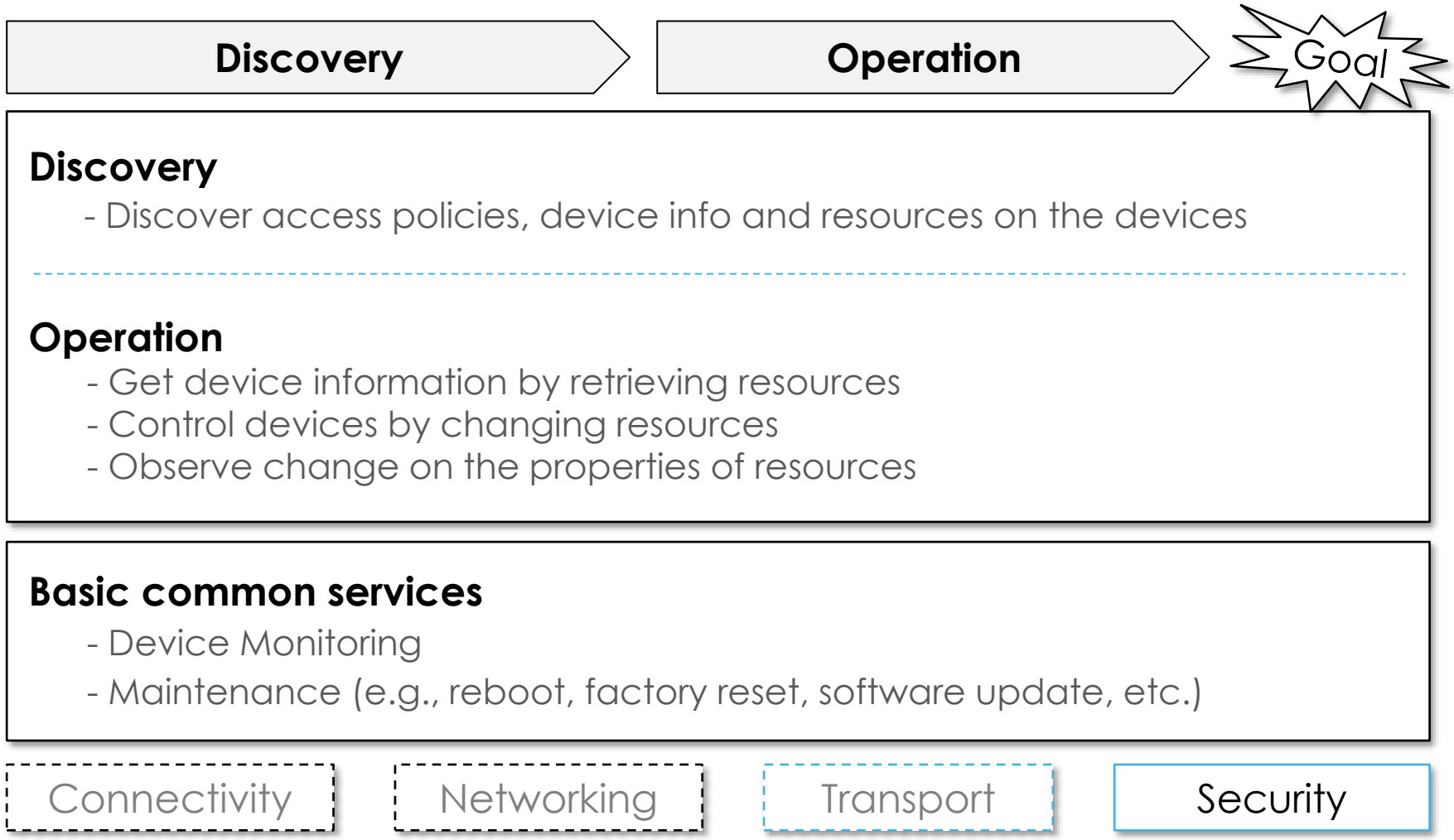
Resources – Default Values

- OCF offers two Interfaces that allow a client to set desired server behaviour on a power cycle.
  - Either to set the values that shall be set for the Resource
  - Or to ensure the values prior to the power cycle were maintained.
- These Interfaces are: oic.if.startup and oic.if.startup.revert



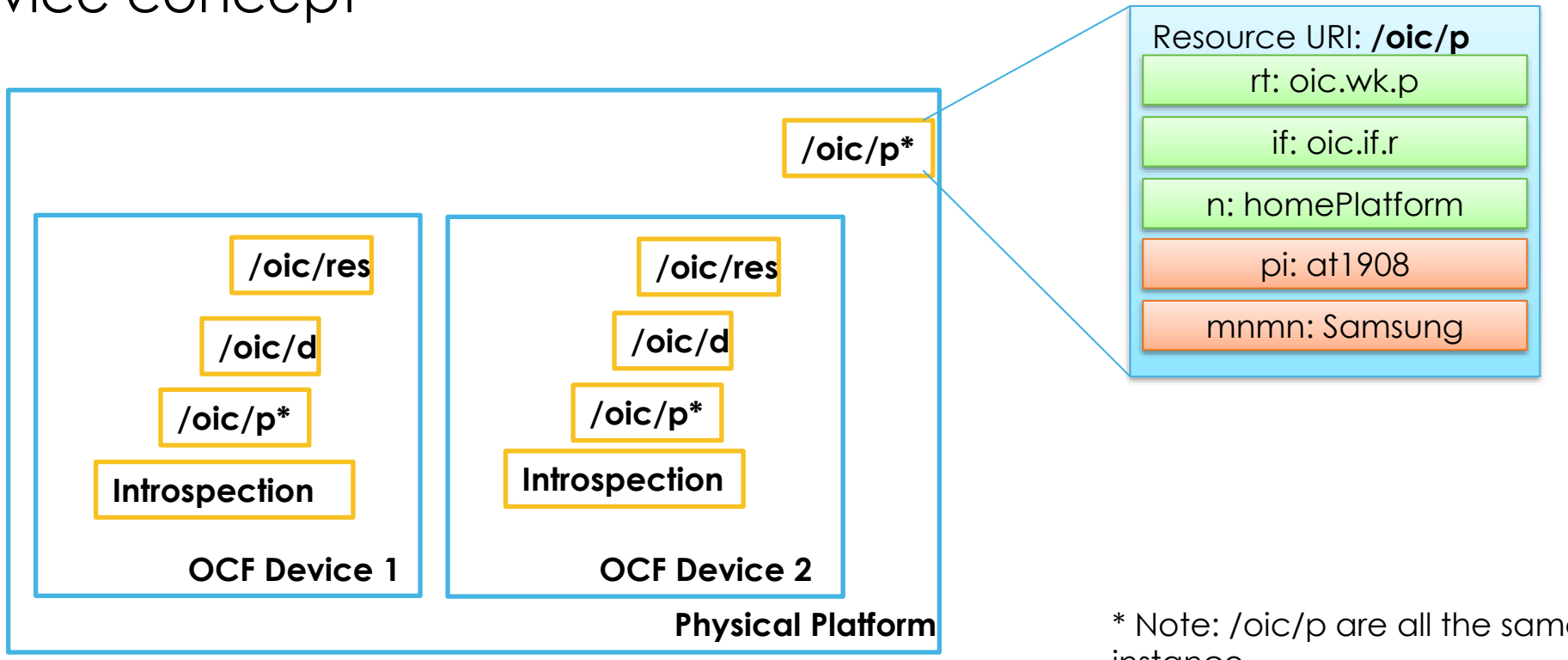
- If in "NormalOperation' Property values are implementation dependent.
- If in "StartupDefaultsSetOperation' Property Values shall be those set via the last UPDATE operation using "oic.if.startup"; values for Properties that are not included in the UPDATE operation are implementation dependent.
- If in "RevertSetOperation" Property Values shall be those prior to the power cycle.

OCF Core Framework Basic Operation



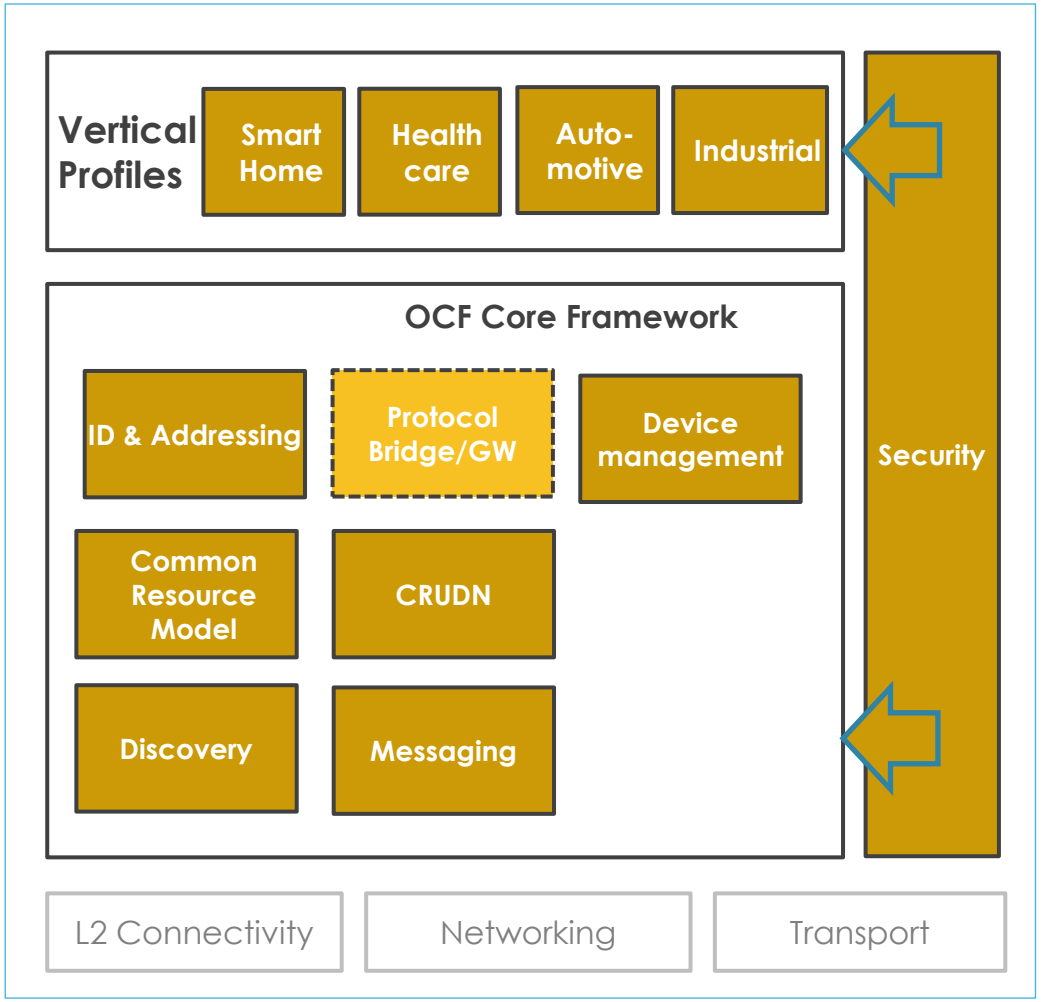
Organization of an OCF Device

- OCF Device concept



* Note: /oic/p are all the same instance

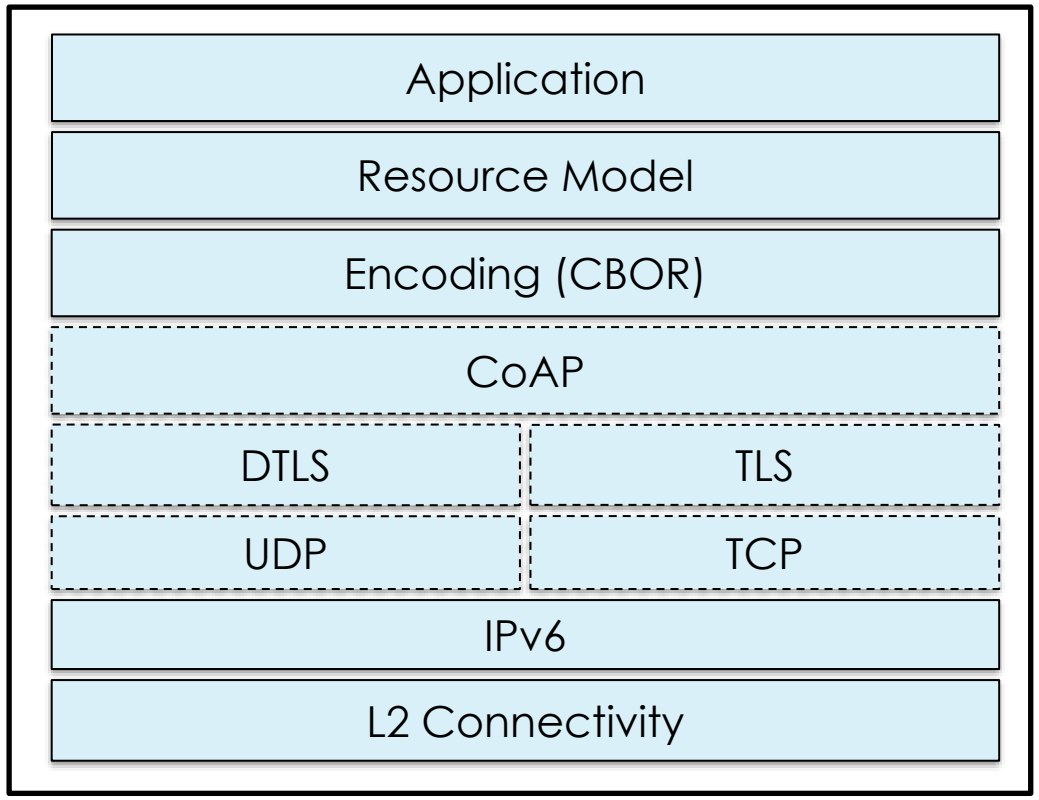
OCF Spec Features – Core Framework Spec



1. **Discovery:** Common method for device discovery (IETF CoRE)
2. **Messaging:** Constrained device support as default (IETF CoAP) as well as protocol translation via bridges
3. **Common Resource Model:** Real world entities defined as data models (resources)
4. **CRUDN:** Simple Request/Response mechanism with Create, Retrieve, Update, Delete and Notify operations
5. **ID & Addressing:** OCF IDs and addressing for OCF entities (Devices, Clients, Servers, Resources)
6. **Protocol Bridge/GW**: Handled by the Bridging Spec with some implications on the Core

Security is fundamental to the OCF ecosystem and applies to all elements

Protocol Stack



OCF Stack

Endpoint Overview

- Definition
  - An (OCF) Endpoint is defined as the source or destination of a request and response messages for a given Transport Protocol Suites (e.g. CoAP over UDP over IPv6). The specific definition of an Endpoint depends on the Transport Protocol Suites being used.
    - an address (e.g. IPv6 address + Port number) or an indirect identifier (e.g., DNS name) resolvable to an address.
    - (e.g.) For CoAP/UDP/IPv6, Endpoint is identified as IP address + port number or DNS name.
- Endpoint characteristics for OCF Device
  - Each OCF Device shall associate with at least one Endpoint with which it can exchange Request & Response messages.
    - When a message is sent to an Endpoint, it shall be delivered to the OCF Device which is associated with the Endpoint. When a Request message is delivered to an Endpoint, path component is enough to locate the target Resource.
  - OCF Device can be associated with multiple Endpoints.
    - E.g. OCF Device may support both CoAP & HTTP
  - An endpoint can be shared among multiple OCF Devices, only when there is a way to clearly indicate the target Resource with Request URI.

Endpoint information in /oic/res with “eps” Parameter

/oic/res

```
[
 { "href": "/oic/res",
   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989/oic/res",
   "rel": "self",
   "rt": ["oic.wk.res"],
   "if": ["oic.if.ll", "oic.if.baseline"],
   "p": {"bm": 3},
   "eps": [{"ep": "coaps://[fe80::b1d6]:44444"}] },
 { "href": "/oic/p",
   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
   "rt": ["oic.wk.p"],
   "if": ["oic.if.r", "oic.if.baseline"],
   "p": {"bm": 3},
   "eps": [{"ep": "coap://foo.bar.com:44444"}, {"ep": “coaps://foo.bar.com:11111"} ] },
 { "href": "/oic/d",
   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
   "rt": ["oic.wk.d", "oic.d.light"],
   "if": ["oic.if.r", "oic.if.baseline"],
   "p": {"bm": 3},
   "eps": [{"ep": "coap://[fe80::b1d6]:44444"}, {"ep": "coaps://[fe80::b1d6]:11111"} ] },
 { "href": "/myLight",
   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
   "rt": ["oic.r.switch.binary"],
   "if": ["oic.if.a", "oic.if.baseline"],
   "p": {"bm": 3},
   "eps": [{"ep": "coaps://[fe80::b1d6]:44444"}, {"ep": "coaps://foo.bar.com:11111"} ] }
]
```

Endpoint for each target resource.

Resource Discovery (CoAP Discovery)

- OCF devices make use of CoAP Discovery using IANA defined OCF Service Address (not the default CoAP address).
- Multicast RETRIEVE (CoAP GET) sent to well known URI /oic/res
- Response is an array of links; each link represents a Resource hosted by the responding server
- Links provide:
  - href
  - Relationship (self link, hosted link, bridged link)
  - Endpoint binds
  - Supported interfaces
  - Observability of the Resource
- Different response views on /oic/res can be realized by using different supported interfaces with the RETRIEVE operation (e.g., CoAP GET /oic/res?if=oic.if.baseline, CoAP GET /oic/res?if=oic.if.b)

Block Transfer with CoAP Messaging

- Basic CoAP messages work well for the small payloads we expect from light-weight, constrained IoT devices
- It is envisioned whereby an application will need to transfer larger payloads
- CoAP block wise transfer as defined in IETF RFC 7959 shall be used by all OCF Servers that receive a retrieve request for a content payload that would exceed the size of a CoAP datagram

Encoding Schemes – CBOR

- Everything in OCF is a Resource.
- All Resources are specified using OpenAPI 2.0 (aka Swagger) in JSON format to define the associated API
- OCF has mandated CBOR as the default encoding scheme on the wire

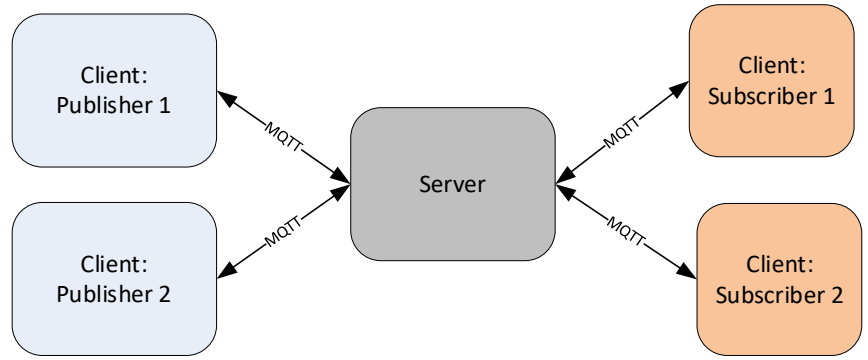
	CBOR	*JSON*	*XML/EXI*
Description	**- Concise binary object representation based on JSON data model**	*- Lightweight, text-based, language-independent data interchange format*	*- Binary compression standard for XML*
Standard	**IETF RFC 7049**	*IETF RFC 7159*	*W3C Efficient XML Interchange Format 1.0*
Content Type	**/application/vnd.ocf+cbor**	*/application/json*	*/application/exi*
OCF M/O	**Mandatory**	*Can be supported*	*Can be supported*

If needed in future revisions

Introspection

- What
  - Device description is available on the network
  - Device description:
    - List all end points
    - Per end point
      - Which method are implemented
        - » Query parameters per method
        - » Payloads definitions (request and response)
- How
  - Put the data described in OpenAPI 2.0 files on the wire as a CBOR encoded OpenAPI 2.0 (aka Swagger2.0) document.
    - Describes the payload on JSON level
      - Property names
      - Type
      - range

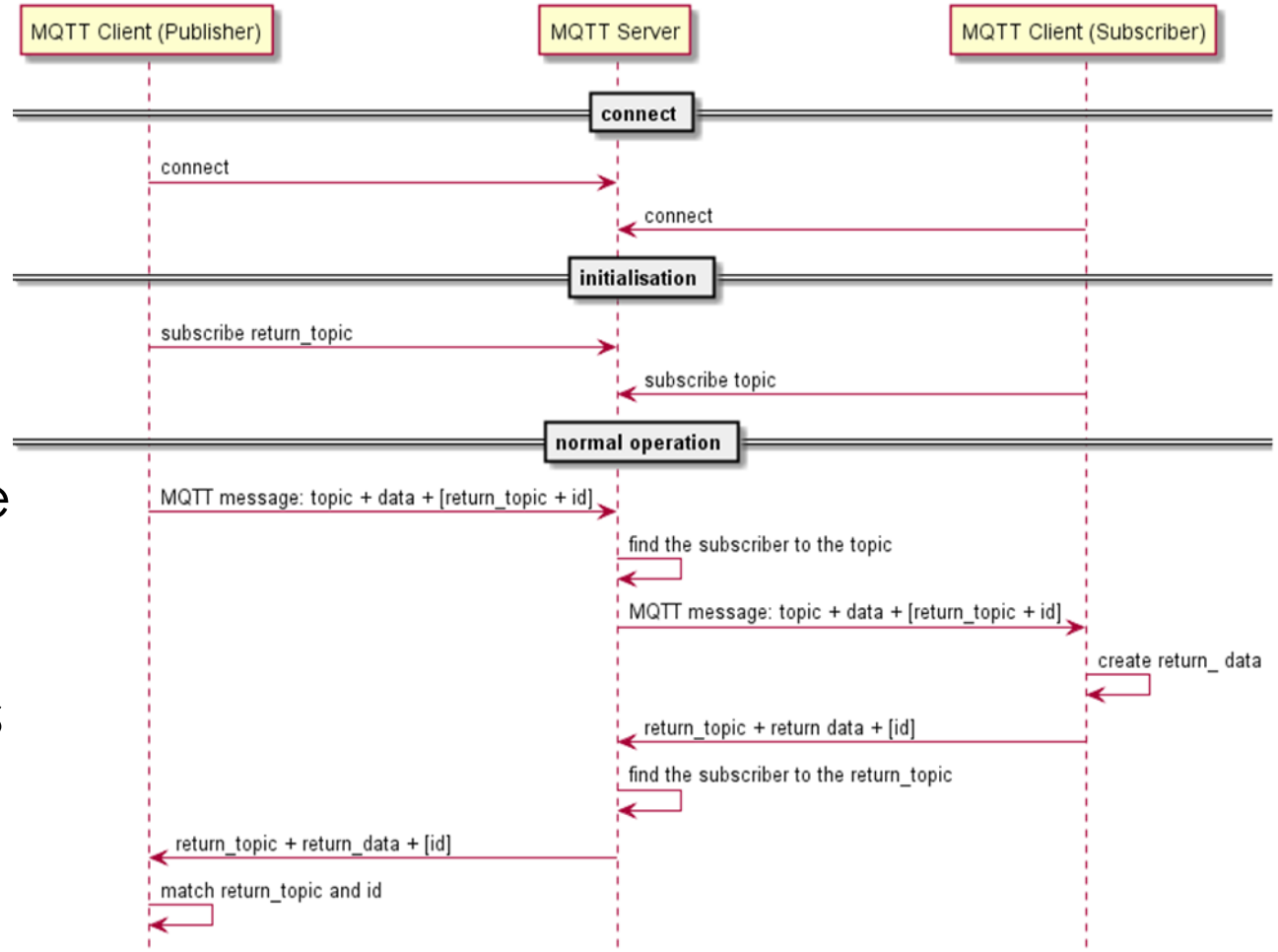
OCF over MQTT



- The communication (transport) of OCF for D2D and D2C is CoAP, with release 2.2.4 the communication (transport) can also be over MQTT.
- The MQTT topics to be used to represent OCF Devices are predefined
- This enables searching for OCF devices in the MQTT domain
- Security is provided by MQTT, all connections between an MQTT clients and MQTT server will adhere to the MQTT spec.

OCF over MQTT

- The MQTT clients will act as the Client or Server role as defined by the specifications.
- The interaction model using CRUDN operations is the same as with CoAP.
- The used resources, including the CBOR payload on the wire is the same as with CoAP
- Security is provided by MQTT, all connections between an MQTT clients and MQTT server will adhere to the MQTT spec.



Collection Resources

- An OCF Resource that contains one or more references (specified as OCF Links) to other OCF Resources, where each Link is individually addressable, is an OCF Collection
  - Client can request a server to create and add a Resource to any collection (except /oic/res) that indicates support for creating that Resource
- An OCF Link embraces and extends typed "web links" as specified in RFC 5988
- The primary example of a collection is /oic/res (Discovery Resource).
  - A small number of Resources in the Resource Model are also collections

Atomic Measurement Resources

- An OCF Resource that ensures a Client can only access the Properties of linked Resources (specified as OCF Links) atomically, as a whole, and read-only, using the "batch" interface
  - Atomically, meaning the value of all properties of the Atomic Measurement are sampled at the same time
  - As a whole, meaning that the values of all properties of the Atomic Measurement will be returned, or no value will be returned
  - Read-only, meaning that the properties of the Atomic Measurement can only be read, not written, using the batch interface. Any attempt to write to any property of the Atomic Measurement will result in an error.
- An OCF Link embraces and extends typed "web links" as specified in RFC 5988
- The primary example of Atomic Measurement Resources are with Healthcare vertical defined OCF Resources (e.g blood pressure measurement)

Device Reset – Hard Reset

- Hard Reset is characterized as a "hard" reset to manufacturer defaults. Hard reset also defines a state where the Device is ready to be transferred to another party.
- A hard reset can be triggered by an authorized OCF Client sending an UPDATE to /oic/sec/pstat setting dos.s = 0 (RESET) or an UPDATE to an instance of /oic/mnt (optional) that supports a 'Factory Reset'.
  - These are functionally equivalent
- Actions to be taken in the Core on an indication being provided to it that a hard reset has been requested:
  - Mandatory Resources: Properties reset to known defaults for oic.wk.d, oic.wk.p
  - Optional Resources: Properties reset to known defaults for oic.wk.con
  - Observes: All active Observe transactions canceled
  - Any resources created on the Device post transition to RFNOP deleted

Device Reset – Observe (established) or In Progress Request Note

- A Server handles an Observe request and informs the Client that the Observe transaction is no longer active by:
  - Sending an appropriate non-success response to the Observer (on the same transaction as the Observe request).
    - Typically a 5.03 (Service Unavailable) with Max-Age

Device Reset – Summary

Function	Actions
SVRs	As per Security Specification
Mandatory Core Resources	Properties in oic.wk.d, oic.wk.p set to defined defaults
Optional Core Resources	Properties in oic.wk.con set to defined defaults
Vertical Resources	Reset to vendor defined defaults
Created Resources	Deleted
Observe Transactions	Canceled with a (typically) 5.03

Versioning

Payload Versioning

- **Purpose**: client and server can understand each others payload.
- **Method**: resource model & encoding information in CoAP header

Device Level Versioning

- **Purpose**: OCF devices can be aware of each others specification implementation version
- **Method**: icv (specification version) in /oic/d resource

Payload versioning

Media Type	ID
application/cbor	60
application/vnd.ocf+cbor	10000

Content-Formats

CoAP Option Number	Name	Format	Length (bytes)
2049	Accept Version	uint	2
2053	Content-Format Version	uint	2

Option Numbers

Version Representation

	Major Version						Minor Version				Sub Version					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Version Example

OCF version	Binary representation	Integer value
1.0.0	0000 1000 0000 0000	2048
1.1.0	0000 1000 0100 0000	2112

OPEN CONNECTIVITY FOUNDATION®

Optional Core Technology
Core Optional Framework Specification

Core Optional Framework Objectives

- Specifies optional functionality (Resources) that can extend the Core Framework

Device/Platform Configuration (Optional)

- Should a Device support a Client to update configurable information in its “/oic/d” Device Resource, it will expose a device configuration Resource with the “oic.wk.con" Resource Type in the "/oic/res“ Discovery Resource
- Should a Device support a Client to update configurable information in its /oic/p Platform Resource, it will expose a platform configuration Resource with the “oic.wk.con" Resource Type in the "/oic/res“ Discovery Resource

Device Management (Optional)

- OCF Resources that provides an interested party (clients) to maintain and provide diagnostics for a Device
  - Resource to provide functionality for enabling factory reset, Device reboot, and last Device error information
  - Resource to enable monitoring the current network state of the Device
  - Resource to enable secure software updates for Devices in both managed and unmanaged networks
    - Download initiated by a vendor (e.g., Smart Home usage)
    - Download initiated by end user (e.g., Smart Home usage)
    - Download initiated by network manager (e.g., Industrial usage)

Alerts (Optional)

- An OCF Resource that provides an interested party (clients) with regard to error or other conditions that the Device is experiencing
  - An Alert contains human readable text that is dependent on the Device itself and the condition being reported
  - A Device may expose discrete instances of an Alert Resource
  - A Device may expose zero or more Alert Resources within an Alert Collection
- The primary example of Alerts Resources are for a managing client, such as a service provider, to observe all alerts from all managed Devices

Icons (Optional)

- Icons are a primitive that are needed by various OCF subsystems, such as bridging
- Resource Type of "oic.r.icon" has been defined to provide a common representation of an icon Resource that can be used by Devices and other ecosystems

Scenes/Rules (Optional)

- Scenes are a mechanism for automating certain operations by bundling user settings into a single Client operation.
  - A Scene is a static entity that stores a set of defined Property values for a Collection of Resources
  - Scenes provide a mechanism to store a setting over multiple Resources that may be hosted by multiple separate Servers.
  - Scenes, once set up, can be used by multiple Clients to recall a setup.
- Rules are Resources that implement autonomous decision logic according to a condition-action pattern
  - The Rule is evaluated based on the Property values of selected Resource instances. Rule Actions are triggered when a Rule Expression evaluates to "true" and consist of defined UPDATE operations that act upon Scene Collections by updating the "lastScene" Property to a defined value
  - Enable Devices to internally create Rules local to the Device itself and allow Clients minimal alterations to existing Rule Resources

OPEN CONNECTIVITY FOUNDATION®

References

Specification Location

Where can I find the specifications and Resource Type definitions?

OCF Specifications:

- https://openconnectivity.org/developer/specifications

Resource Type Definitions

- Core Resources: https://github.com/openconnectivityfoundation/core
- Core Extension Resources: https://github.com/openconnectivityfoundation/core-extensions
- Bridging Resources: https://github.com/openconnectivityfoundation/bridging
- Cloud Resources: https://github.com/openconnectivityfoundation/cloud-services
- Security Resources: https://github.com/openconnectivityfoundation/security-models
- Vertical Resources and Derived Models: https://oneiota.org/documents?filter%5Bmedia_type%5D=application%2Framl%2Byaml

OPEN CONNECTIVITY FOUNDATION®

OCF Specification Overview
Core Extension: OCF Device to Cloud Services

OCF 2.2.4 Release

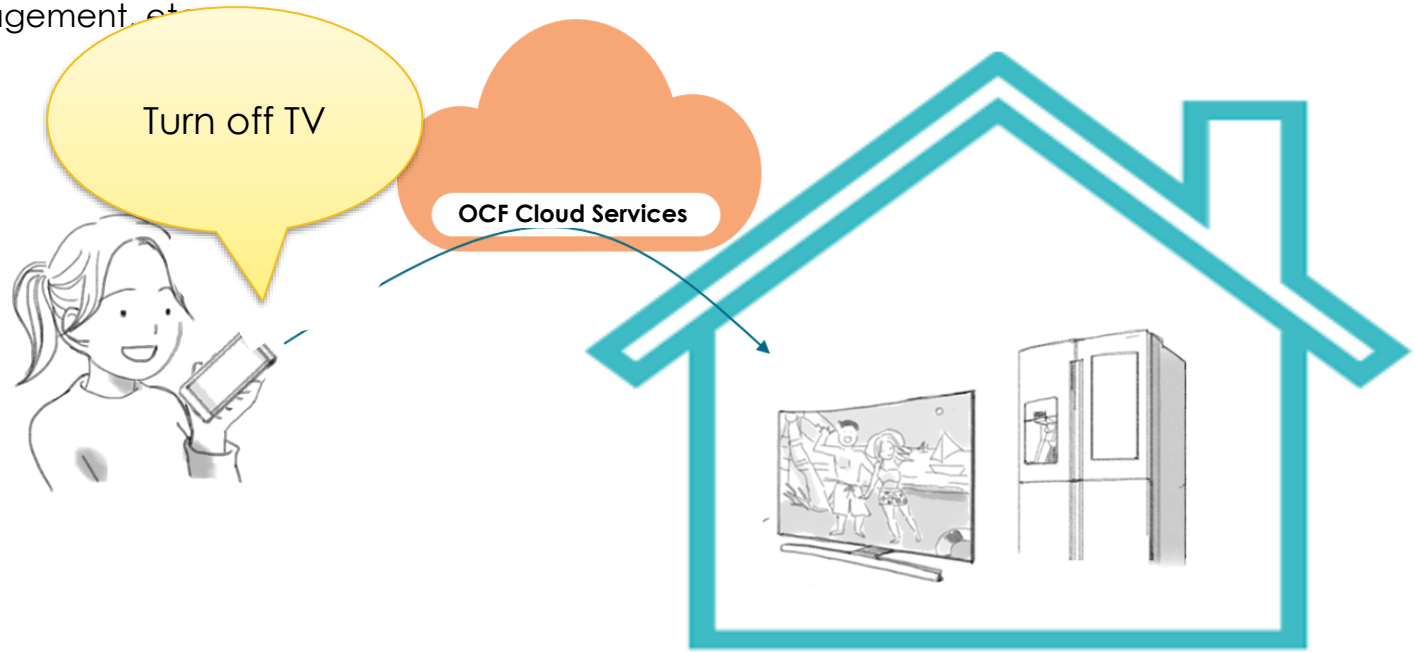
October 2021

Revision History

- 2.2.4
  - Cloud Proxy was added

Use Case

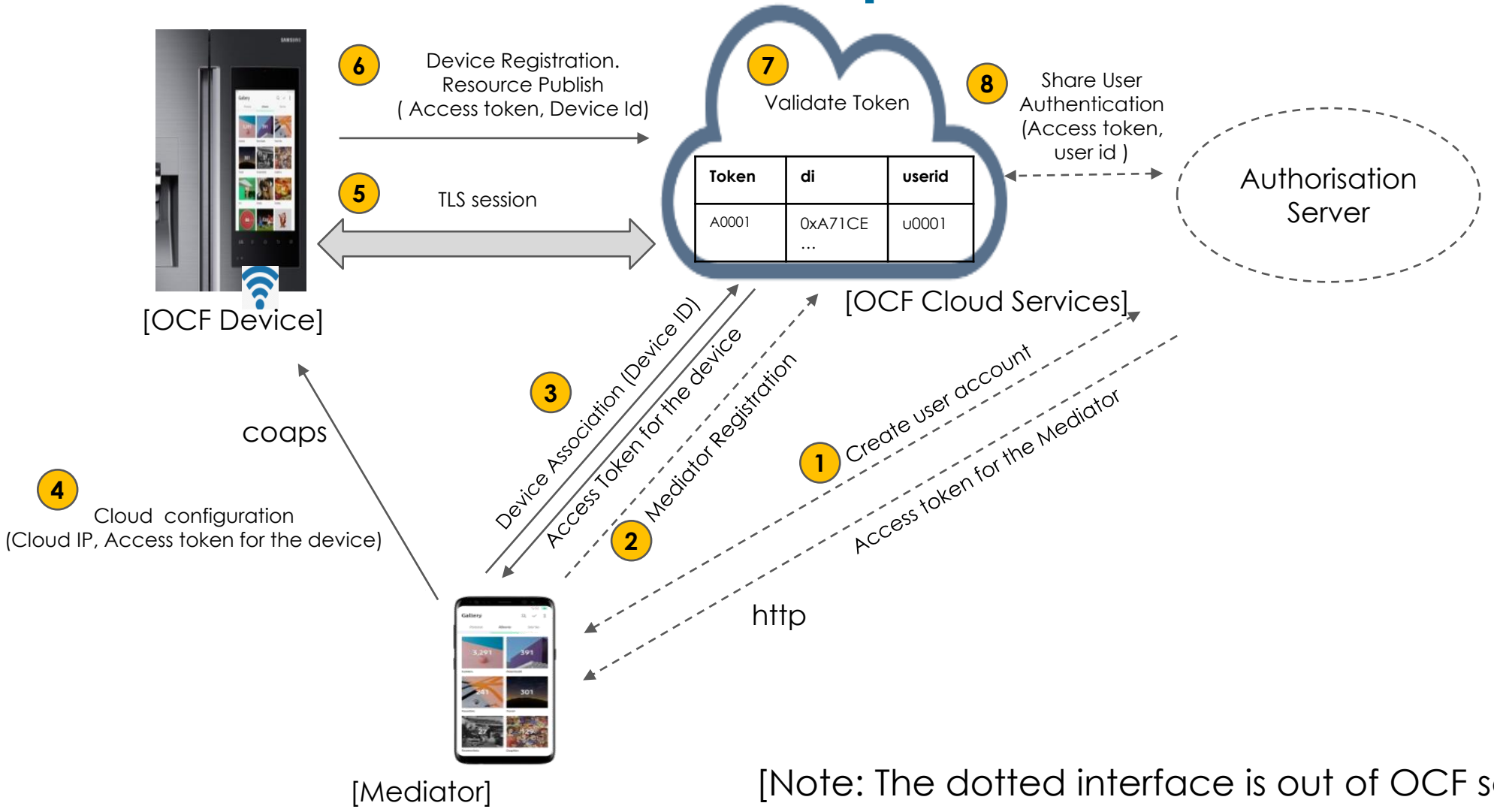
- Remote Control/Manage OCF devices based on user authentication
  - User can access OCF devices which belong to or are shared with them regardless of location.
  - User can receive cloud services in conjunction with the registered devices
    - E.g. Device Management, Home security, Energy management, et
- Usage & Operational Scenario
  1. Jane creates an account in the cloud
  2. Jane registers device resources under the created account
  3. She can control the device anywhere out of the proximal network



Solution Overview

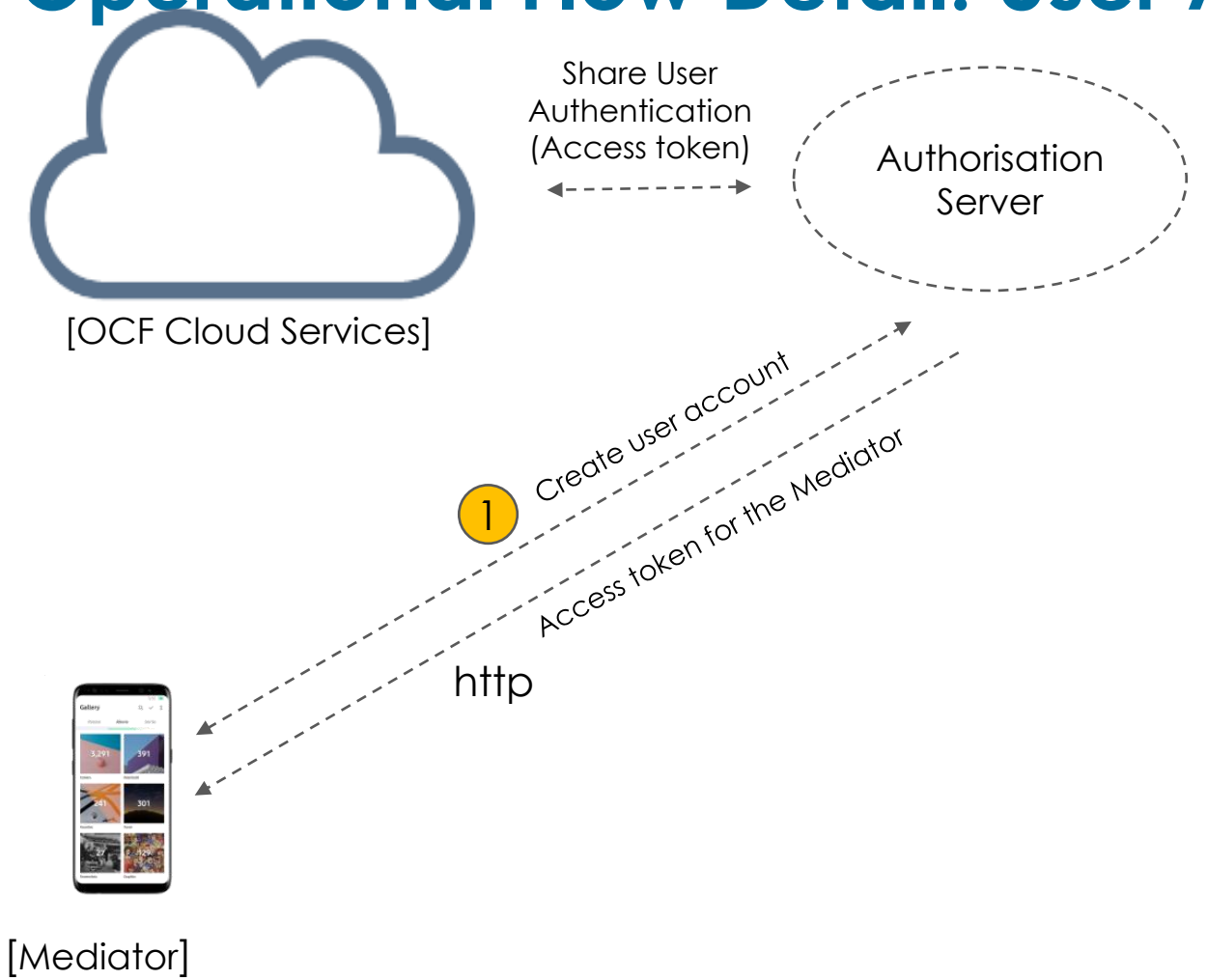
- OCF defines a logical function (termed "OCF Cloud Services" in this presentation) which allows a Device to register itself "in the Cloud" and thus support remote access applications.
- All communication between the Device and "OCF Cloud Services" is over TLS.
- The logical "OCF Cloud Services" instance is hosted by any vendor that wishes to support the capability.
- OCF itself does not act as an "OCF Cloud Services" host

OCF Device to Cloud Services: Operational Flow



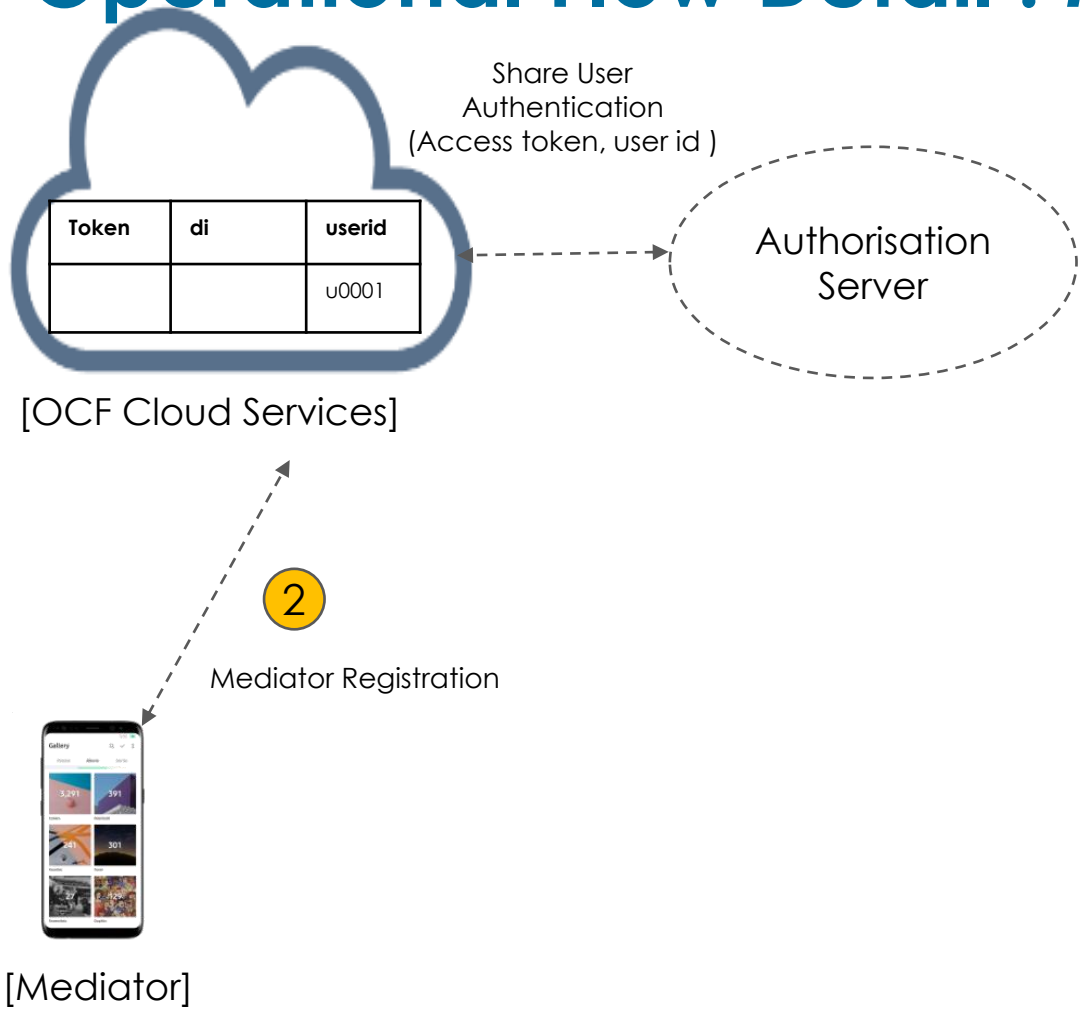
[Note: The dotted interface is out of OCF scope]

Operational Flow Detail: User Account Creation



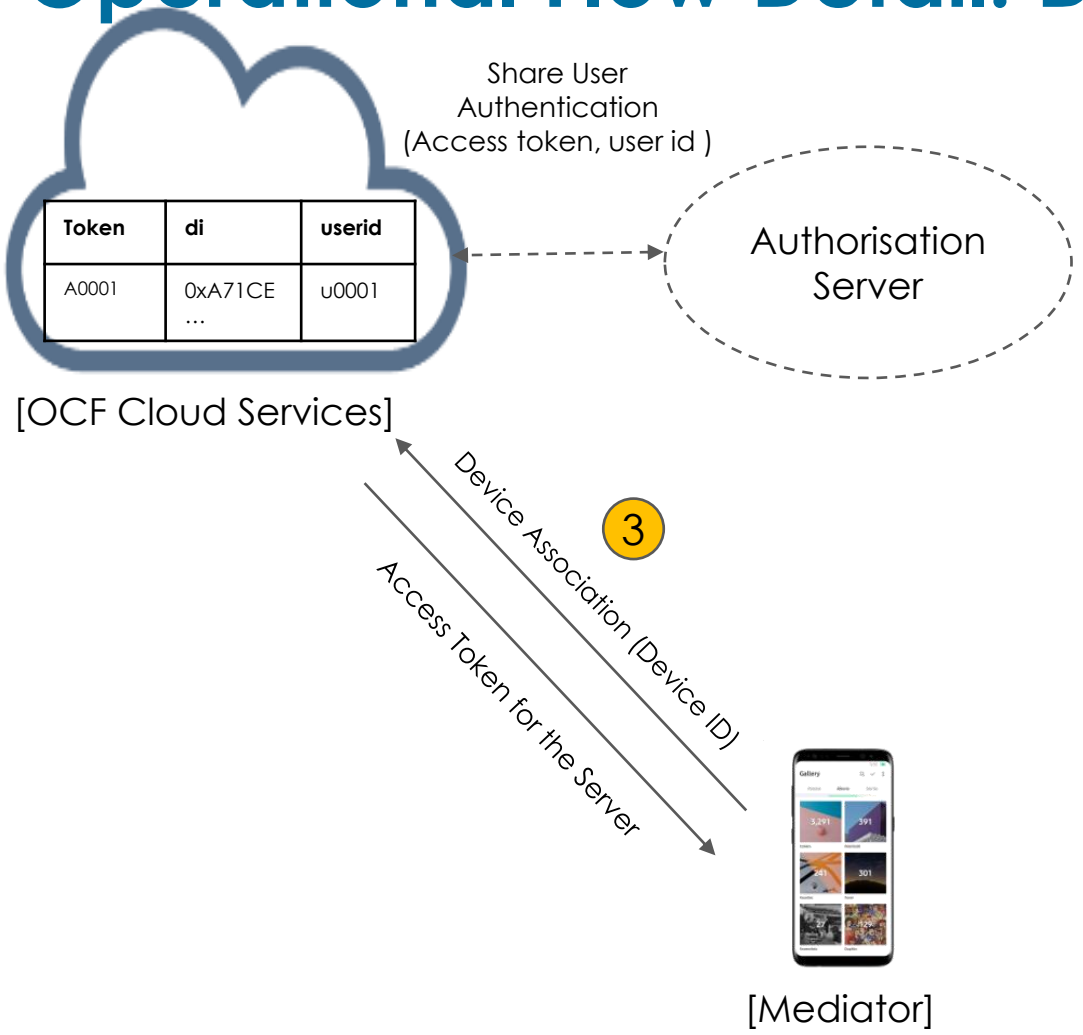
1. The OCF Cloud Services User downloads a Mediator onto their phone which will be used to Provision the Device.
2. The Mediator is configured with or through some out of band process to obtain the URL of the Cloud Service (e.g. the Mediator may be an App from the OCF Cloud Services Provider)
3. The OCF Cloud Services User has access credentials to use the OCF Cloud Services (i.e. user name/password or similar)
   - User can use their 3rd party user account

Operational Flow Detail : Mediator Registration



1. The Mediator provides this Access Token to the OCF Cloud Services host.
2. The OCF Cloud Service may also provide a new Access Token (that is different from the Access Token provided by the Mediator). The Mediator is now registered. The “uid" identifies the OCF Cloud Services User. This "uid" is the same for all Mediator instances that may be associated with the OCF Cloud Services User
3. This same user ID can be used to assign multiple Devices to the same OCF Cloud Services User

Operational Flow Detail: Device Association



1. The Mediator associated with a User ID requests that the OCF Cloud Services host associates an OCF device with the same User ID by providing the Device ID of desired device
2. The OCF Cloud Services host returns a unique Access Token for the device and maintains a map where the Access Token and the Device ID are stored
3. The Access Token can be created by the OCF Cloud Services host or by an other entity. For the latter case, the OCF Cloud Services host also returns an Authorisation Provider Name("apn") in addition to the Access Token

Operational Flow Detail: Device Provisioning by the Mediator

[OCF Device]

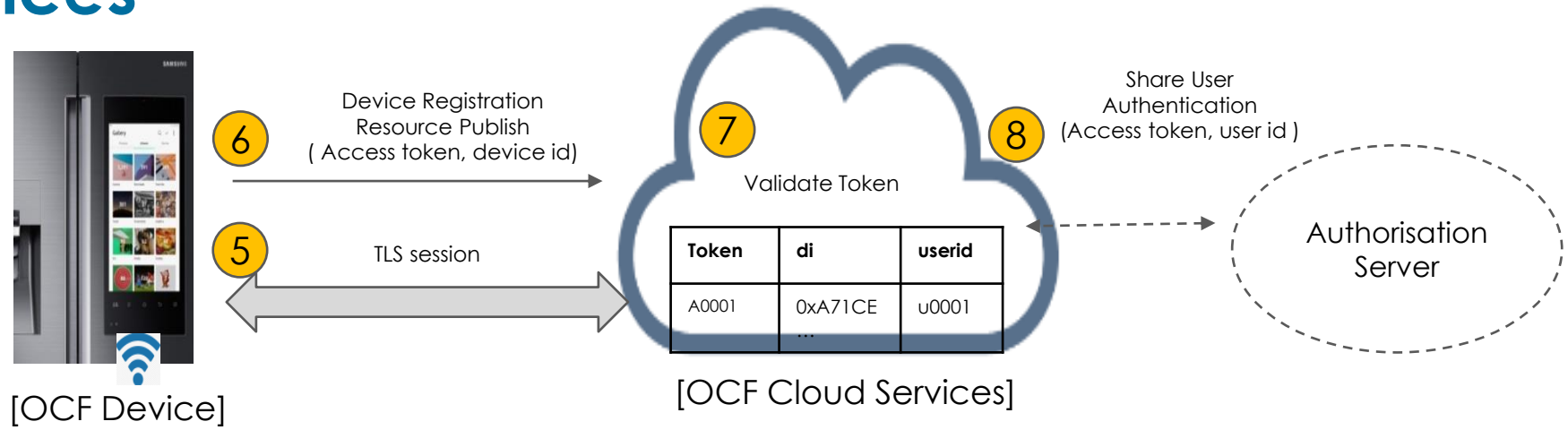
4

Cloud configuration
(Cloud URI Access token)

[Mediator]

1. The Device is configured by the OBT by adding the required ACEs and creds to give the Mediator access to the CoAPCloudConf (CCC) Resource
2. The Mediator connects to the Device through normal OCF Discovery processes.
3. The Mediator updates the CCC Resource on the Device with the Access Token ("at") and OCF Cloud Services URI ("cis"). The Mediator may also provide the Auth Provider Name ("apn").

Operational Flow Detail: Device Registration with OCF Cloud Services



1. The Device establishes a TLS connection with the OCF Cloud Services host using the properties in CCC resource.
2. The Device sends an UPDATE request to the Account Resource which includes the following Properties: "di","accesstoken", "authprovider"
3. The OCF Cloud Services host ensures that the "di" and the "accesstoken" match its current values. The "accesstoken" value is the same one that the OCF Cloud Services host or Auth provider provided to the Mediator
4. If the values match, the OCF Cloud Services host sends the Account Resource Properties in the UPDATE response
5. If the Device sends a RETRIEVE request to any of the OCF Cloud Services hosted Resources, the Cloud responds with an appropriate error code.

Operational Flow Detail: Login with OCF Cloud Services

1. In order to establish a TLS session and connect to the OCF Cloud Services host to enable passing data between the two, the Device sends an UPDATE request to the Session Resource which includes:
    1. "di" – The value of "di" from "/oic/d" of the Device
    2. "uid" as supplied from the Account Resource UPDATE response
    3. "accesstoken" as supplied from the Account Resource UPDATE response
    4. "login": true
2. The OCF Cloud Services host verifies that the values in the UPDATE request are correct and if so, the Cloud sends a response message that includes the remaining session time ("expiresin").
3. The Device now has an active TLS connection and can exchange data.

Operational Flow Detail: Publishing Links to the OCF Cloud Services RD

1. Once the TLS connection has been established to the OCF Cloud Services host, the Device publishes its Resources to the hosted RD function so that they can be seen/accessed remotely.
2. All Resources that are published to the hosted RD function are observable.
3. The acl2 and cred Resource of the OCF Device have to be provisioned by the OBT/AMS/CMS/DOTS to give the OCF Cloud Services host the required CRUDN permissions.

Operational Flow Detail: Client to Server Communication through OCF Cloud Services

1. Clients must go through this same process and register with the OCF Cloud Services host. All of an OCF Cloud Services User's Devices (Clients and Servers) will be assigned the access control right associated with the User ID
2. The OCF Cloud Service allows communication between all of a OCF Cloud Services User's Devices based on the fact that they have the same User ID.
3. When the Client attempts CRUDN actions on the Links hosted by the OCF Cloud Services RD, the OCF Cloud Services host forwards those requests to the Device which responds to the OCF Cloud Services host which then gets returned to the Client (i.e. Client -> OCF Cloud Services -> Device -> OCF Cloud Services -> Client).

Operational Flow Detail: Refreshing Connection with OCF Cloud Services

1. When (or before) the "expiresin" timer expires, the Device should refresh its token by sending an UPDATE request to the Token Refresh Resource that includes:
    1. "di"
    2. "uid"
    3. "refreshtoken"
2. The OCF Cloud Services host responds with a new
    1. "accesstoken"
    2. "refreshtoken"
    3. "expiresin"

Operational Flow Detail: Closing Connection with OCF Cloud Services

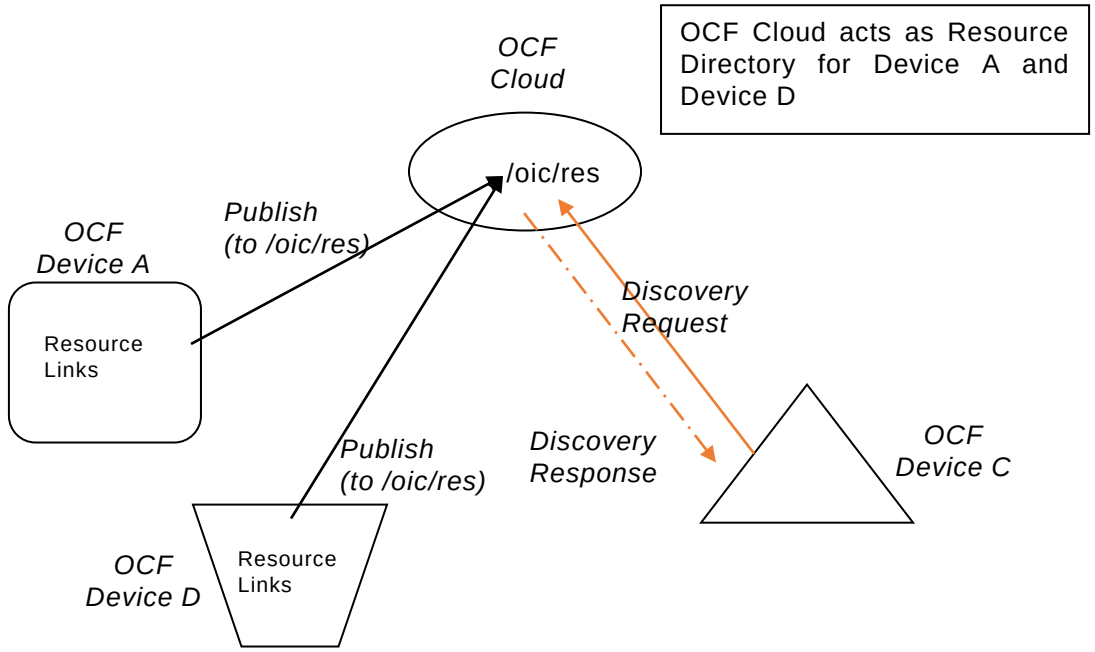
- If the Device wants to log out of OCF Cloud Services, it sends an UPDATE request to the Session Resource which includes:
  - "di", "uid“, and "accesstoken" as supplied from the Account Resource UPDATE response
  - "login": false

Operational Flow Detail: Deregistering from OCF Cloud Services

- The Device sends a DELETE request message to the Account Resource which includes: "accesstoken“,"di"
- The OCF Cloud Services host sends a response message confirming that the Device has been deregistered.
  - To connect to the OCF Cloud Services host again, the Device has to be provisioned by the Mediator again and then reregister with OCF Cloud Services

OCF Cloud Services: Resource Directory

- OCF Cloud Services host a Resource Directory to enable indirect discovery:
  - Indirect discovery is when a 3rd party (the Resource Directory), other than the discovering Device and the discovered Device, assists with the discovery process. The Resource Directory only provides information on Resources on behalf of another Device but does not host Resources on part of that Device.



- The OCF Cloud acts as Resource Directory for Device A and Device D.
- Device A and Device D publish their Resource information to the OCF Cloud.
- Device C queries the OCF Cloud to acquire the Resource information of Devices A and D

OCF Cloud Services: Resource Directory

- 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".
- The ability to host a Resource Directory is indicated by the OCF Cloud exposing an instance of the "oic.wk.rd" Resource Type in its "/oic/res".
  - The discoverable instance of "oic.wk.rd" shall allow only secure connections (e.g. OCF Endpoint with a scheme of "coaps" or "coaps+tcp").

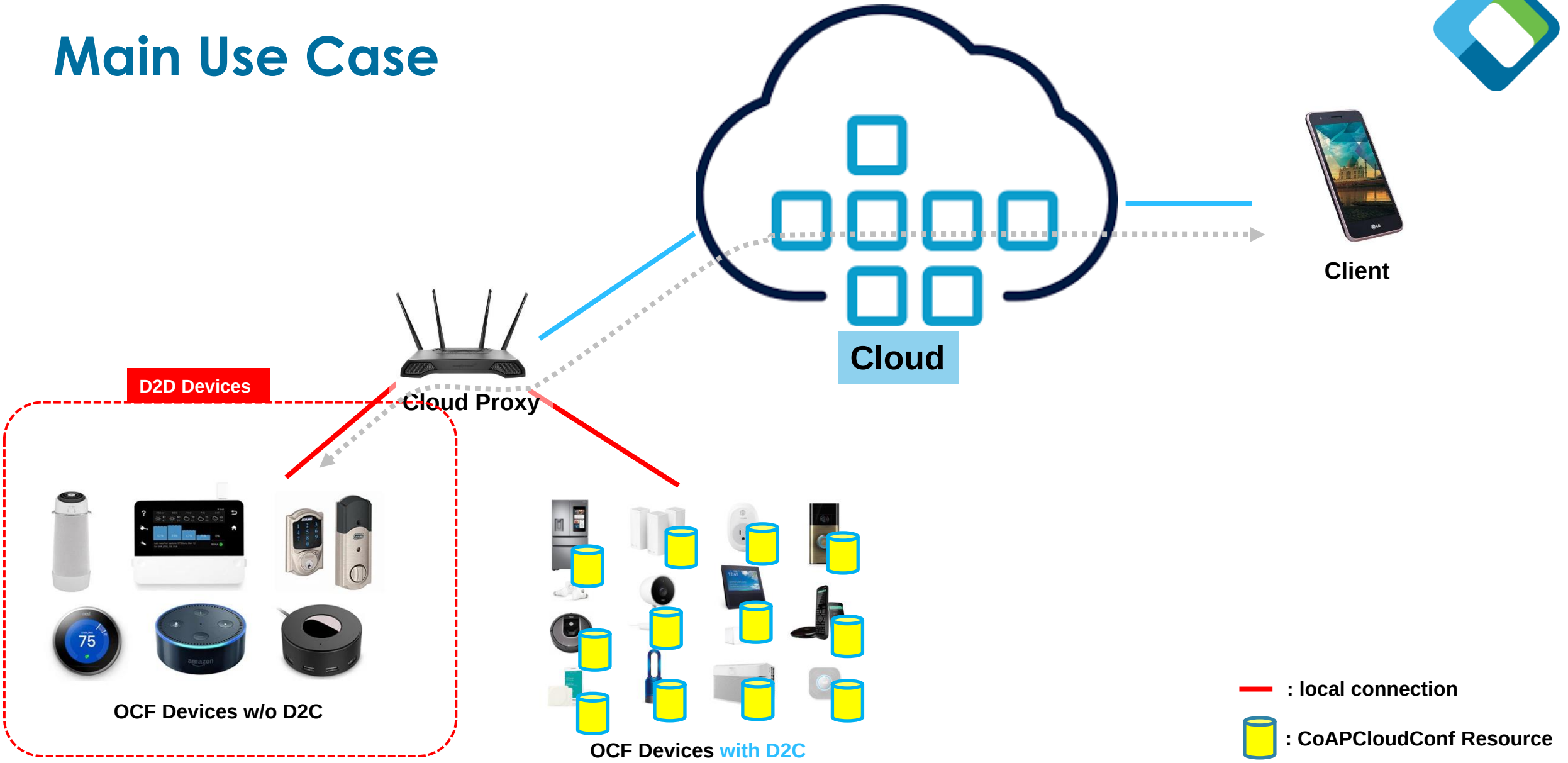
Device Reset – Hard Reset

- Hard Reset is characterized as a "hard" reset to manufacturer defaults. Hard reset also defines a state where the Device is ready to be transferred to another party.
- A hard reset can be triggered by an authorized OCF Client sending an UPDATE to /oic/sec/pstat setting dos.s = 0 (RESET) or an UPDATE to an instance of /oic/mnt (optional) that supports a 'Factory Reset'.
  - These are functionally equivalent
- Actions to be taken by a Device that is registered to OCF Cloud Services on an indication being provided to it that a hard reset has been requested:
  - The Device de-registers from OCF Cloud Services
  - The Properties provisioned by a Mediator in the CoAPCloudConf Resource are reset to known defaults

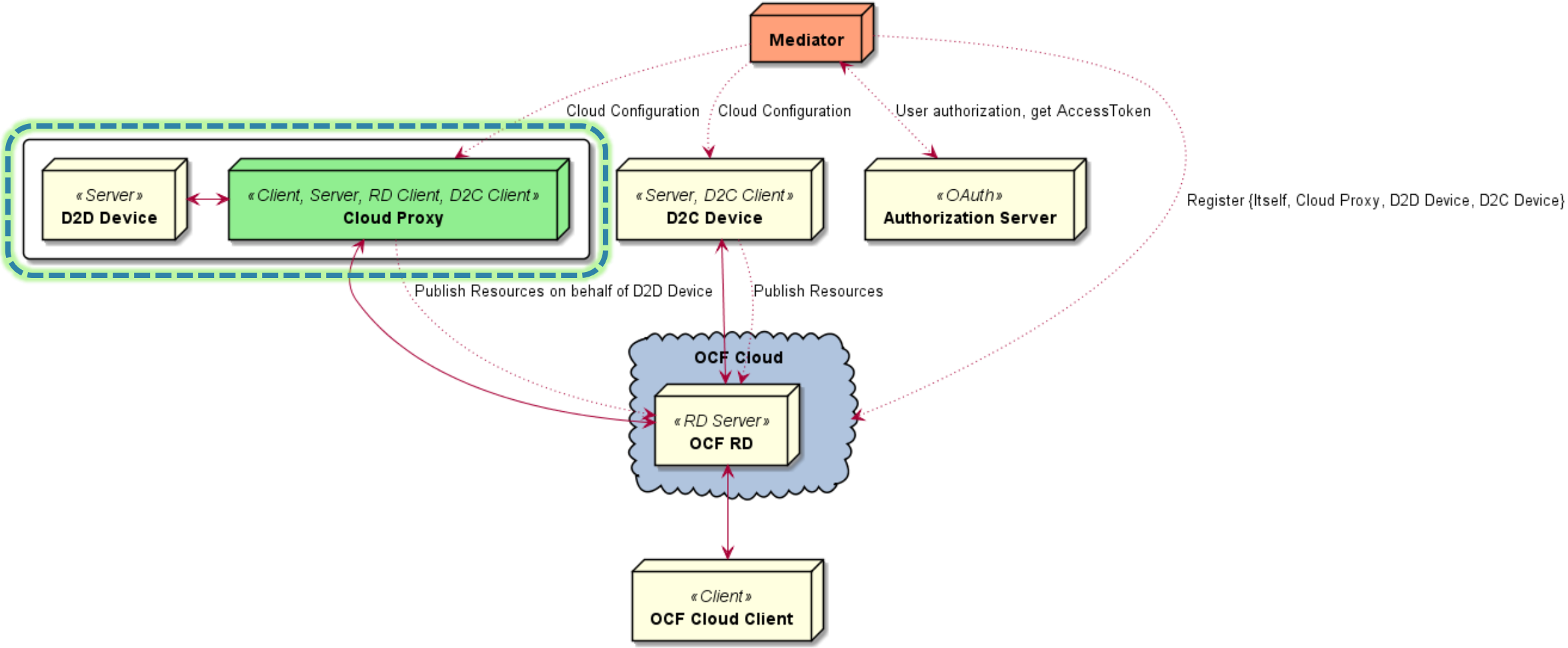
Cloud Proxy

- OCF Cloud Proxy
  - Provide **OCF Cloud connectivity** to OCF D2D Device
    - OCF D2D Device: OCF Device without D2C capability
  - It is not an OCF Bridge Platform
    - It doesn't maintain VOD for D2D Device
    - No Data Model translation happens
- **D2DServerList** Resource
  - OCF Resource that keeps the list of D2D Devices which are proxied through the Cloud Proxy
  - OCF Cloud Proxy maintains this
- Basic operation
  - 1) Cloud Proxy registers itself to the OCF Cloud
  - 2) Mediator updates D2DServerList with Device ID of D2D Device
  - 3) Cloud Proxy retrieves "/oic/res" of the D2D Device, and publishes them to the OCF Cloud

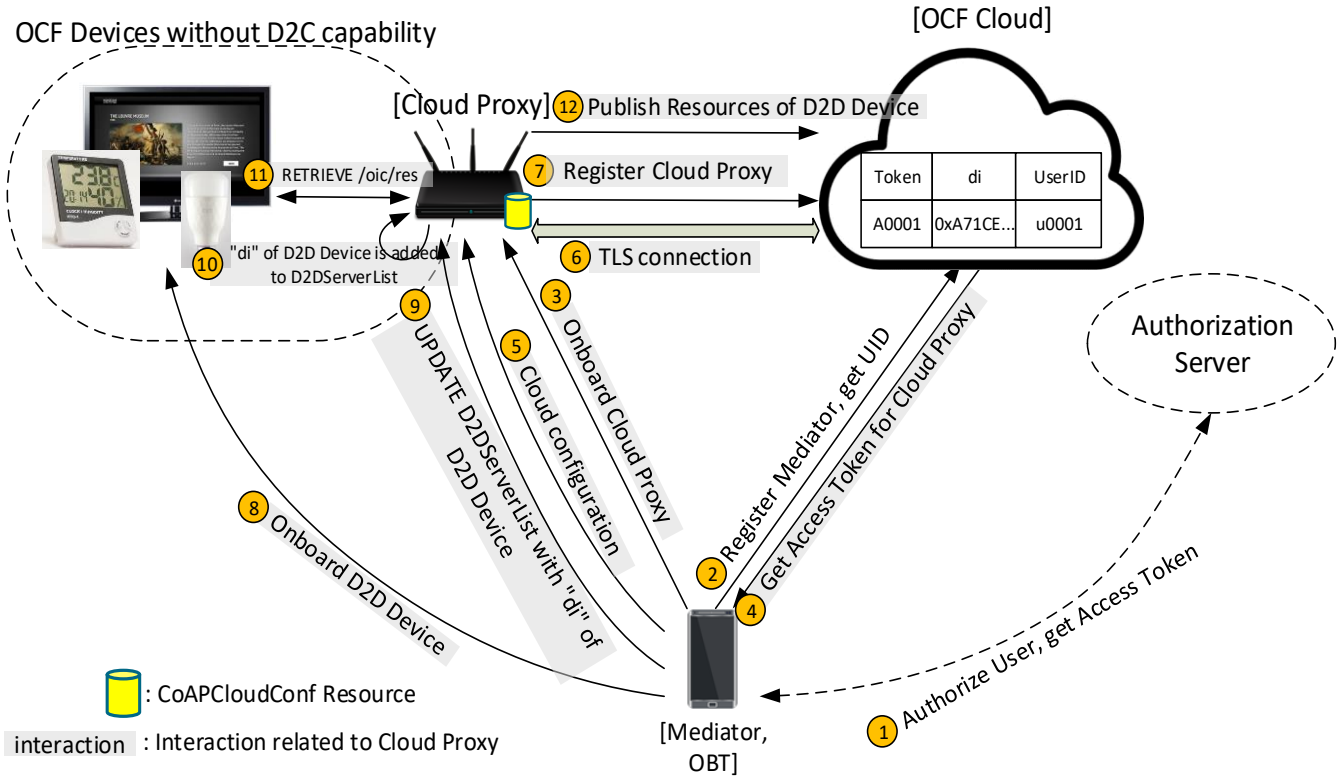
Main Use Case



Cloud Proxy Architecture



Cloud Proxy Interaction Flow



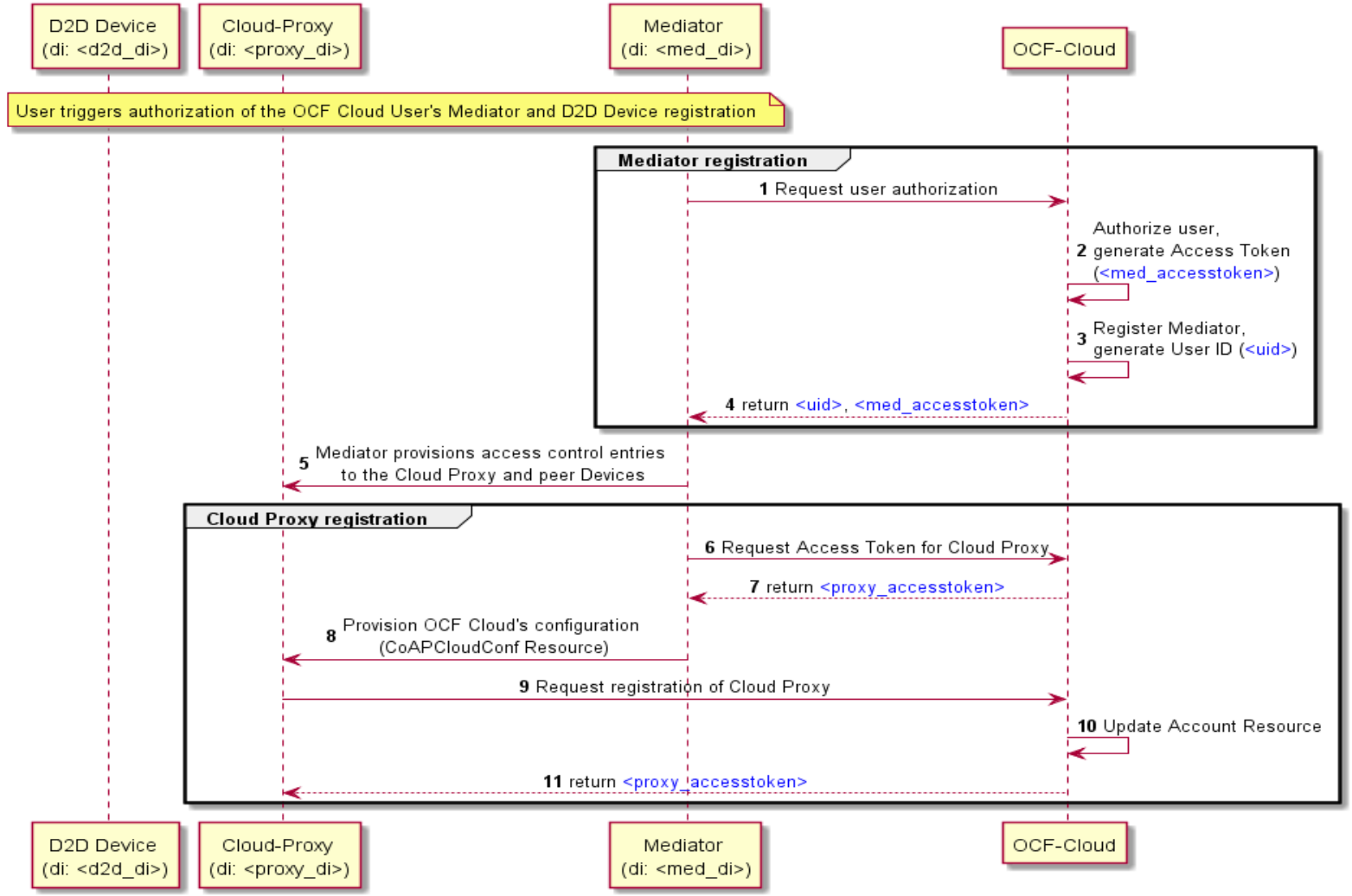
Cloud Proxy interaction flow

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, 4	The Mediator onboards the Cloud Proxy, and gets an Access Token from OCF Cloud
5	The Mediator provisions "oic.r.coapcloudconf" on the Cloud Proxy with the Access Token, the URL of the OCF Cloud, the identity (UUID) of the OCF Cloud, and optionally an Authorisation Provider Name.
6, 7	The Cloud Proxy establishes a TLS session to the OCF Cloud and subsequently registers with the OCF Cloud. After that the Cloud Proxy may publish its D2DServerList Resource to the OCF Cloud. The D2DServerList Resource can be used for management purposes.
8	The Mediator onboards the D2D Device. It is an implementation choice how to determine which OCF Device is a candidate D2D Device which will be proxied through the Cloud Proxy.

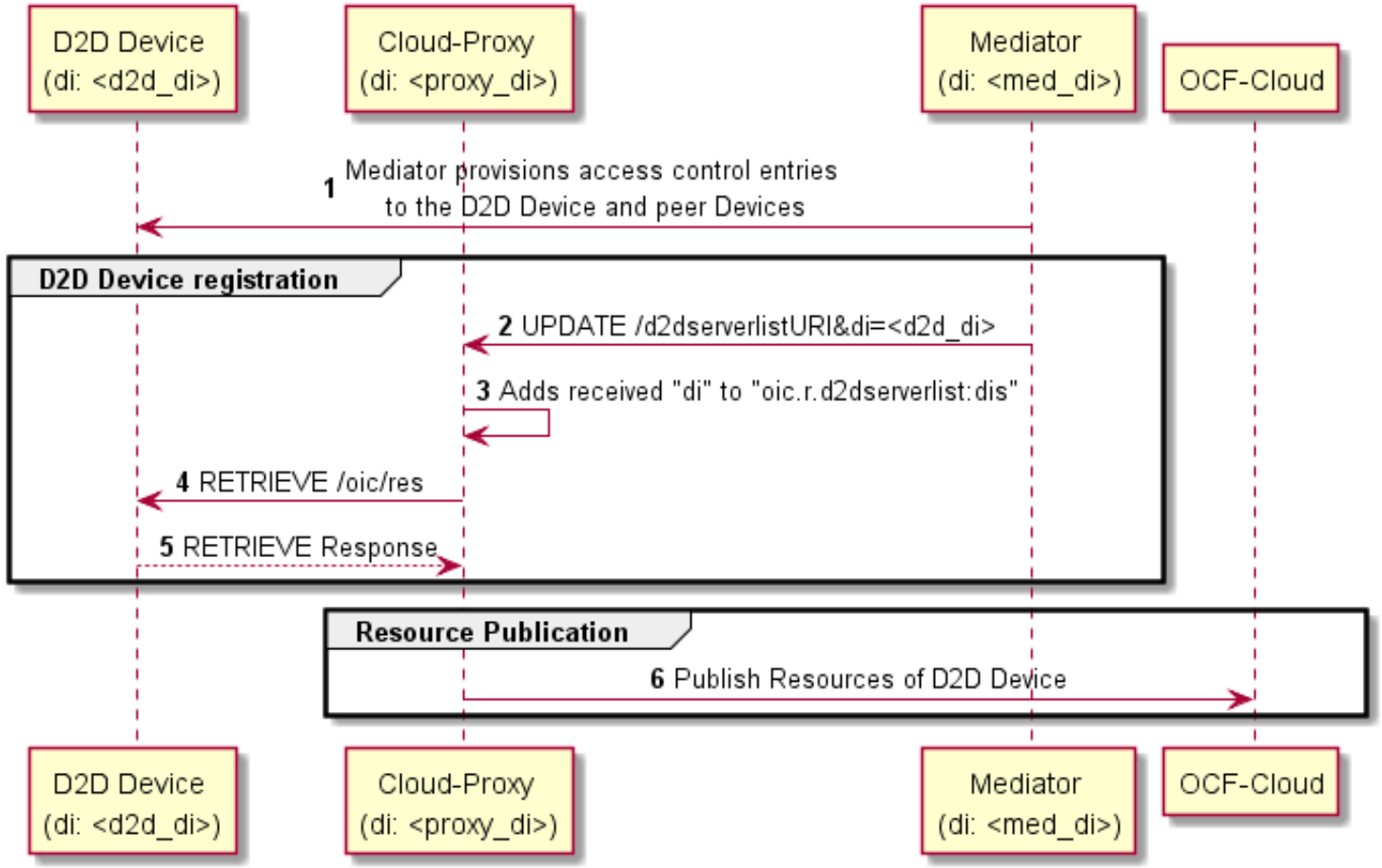
9, 10	The Mediator updates the D2DServerList Resource with the "di" parameter of the D2D Device. When the Cloud Proxy receives this UPDATE request, the Cloud Proxy adds "di" to "oic.r.d2dserverlist:dis".
	
	/example/d2dserver-listURI
	{
	"dis": [
	"0fa4f8d6-b246-11eb-bc27-0c9d928536d7"
	]
	}

11, 12	The Cloud retrieves "/oic/res" of the corresponding D2D Device, and publishes the Resources of the D2D Device to the OCF Cloud. At this time, the Cloud Proxy prepends the Device UUID of the D2D Device to the URI ("href") of all Resources.
	
	{
	"di": "<proxy_di>",
	"links": [
	"anchor": "<proxy_di>"
	"href": "/<d2d_di>/<href of D2D Device>"
	"eps": "eps" for the OCF Cloud
	…
	],
	"ttl": 600
	}

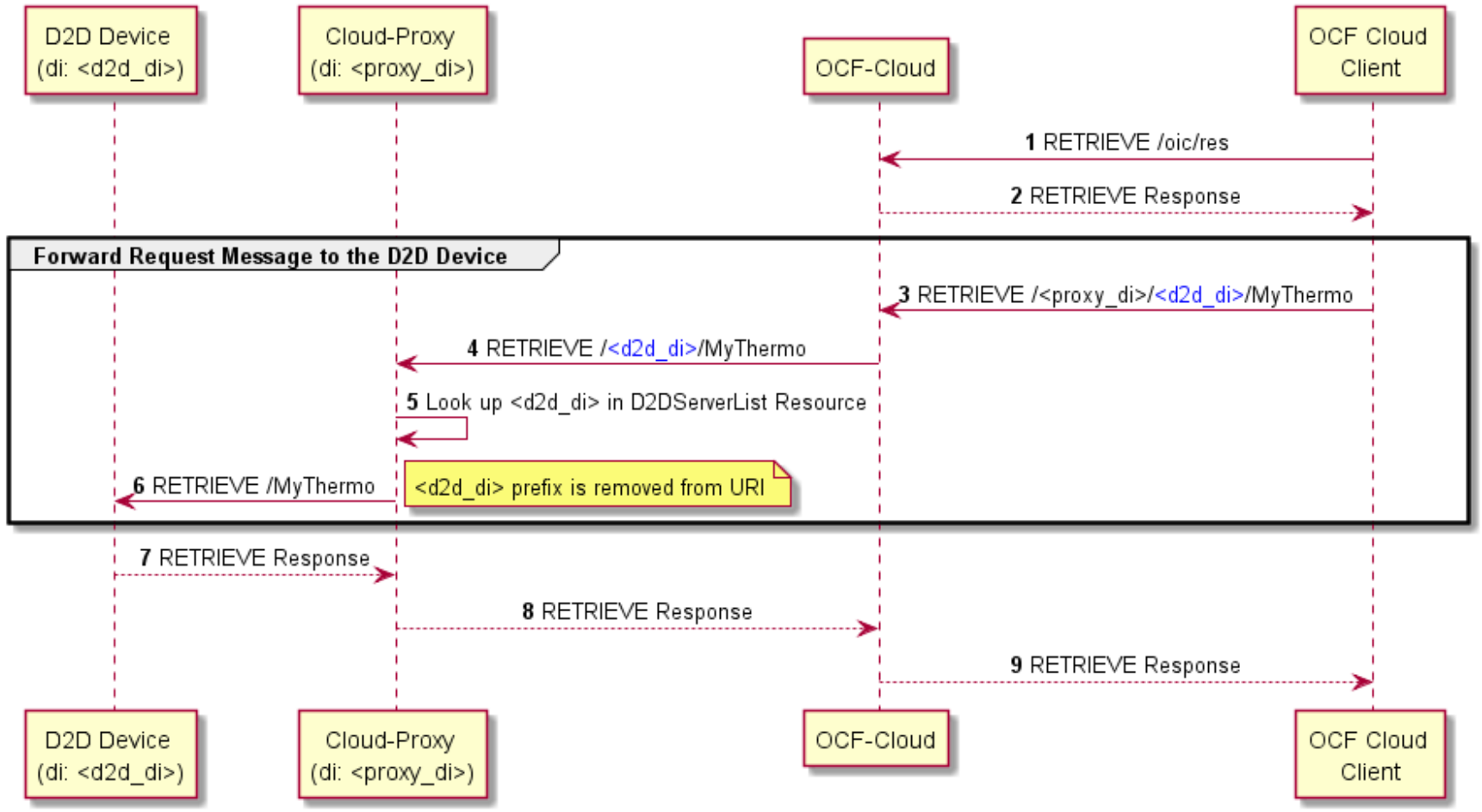
Cloud Proxy – Registration with OCF Cloud



Cloud Proxy – Resource publication to OCF Cloud



Cloud Proxy – Resource Interaction model through Cloud Proxy



OPEN CONNECTIVITY FOUNDATION®

OCF Specification Overview
Core Extension: Easy Setup

OCF 2.2.4 Release

October 2021

OPEN CONNECTIVITY FOUNDATION®

Wi-Fi Easy Setup

Overview

Overview

- Easy Setup is the 1st step when a device is unboxed. Specifically for UI-Less devices this is very important step. Wi-Fi Easy Setup spec defines interoperable data model that can be used to configure Wi-Fi connection on a device using a common communication channel. It also provides a standard way of a device proximally advertising its presence for discovery by clients that will perform the configuration. Other than Wi-Fi connection setup, OCF 2.0 specifications optionally provide a way to configure a connection with OCF Cloud Services.

- **Objectives:**
  - Define data model to be used for Easy Setup of an unboxed device.
  - Define spec with standard device beaconing and lost connection behavior.
  - Define Device roles and provide informative flow of operation.
  - Reuse existing security mechanism for Device Ownership and Access Control.

Scenario(s) / Use cases

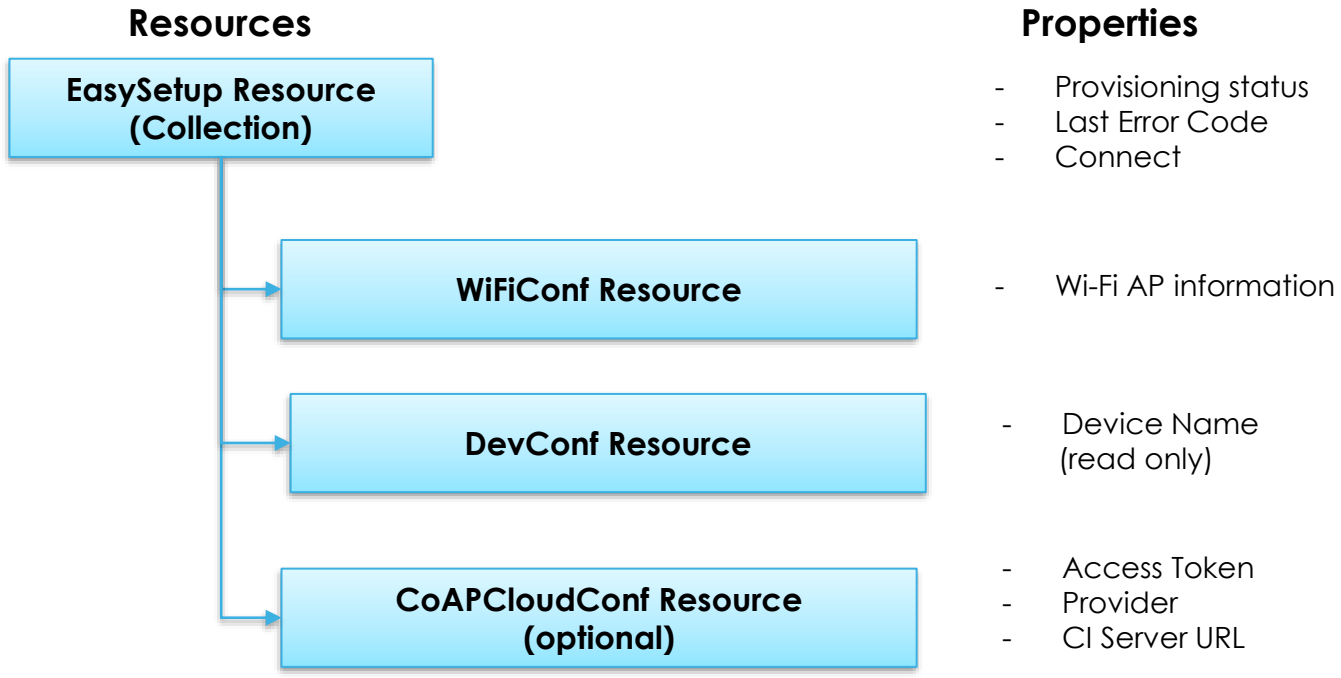
- Procedure
  - [1] A device is unboxed.
  - [2] Mobile connects to the unboxed device.
    - Using a Soft AP network when Wi-Fi transport is preferred.
    - Mobile transfers Home AP's information and other information.
      - SSID, password, security type of Home AP.
  - [3] Unboxed device connects to Home AP.
  - [4] (*Optionally*) Mobile transfers a cloud access information to the device via Home AP network.
  - [5] (*Optionally*) Unboxed device registers to OCF Cloud Services.

Roles & Definitions

- Easy Setup
  - Process of configuring an Enrollee to an Enroller using a Mediator (by transferring of essential information about the Enroller to the Enrollee).
- Mediator
  - Logical function that enables the Enrollee to connect to the target Network (Enroller). The Mediator transfers configuration information to the Enrollee.
  - Example: Mobile phone/PC
- Enrollee
  - The Device that needs to be configured and connected.
  - Example: Air-conditioner, Printer.
- Enroller
  - The target network entity to which the Enrollee connects.
  - Example: Wi-Fi Access Point
- Soft AP
  - Software Enabled Access Point hosted on the Enrollee which is not a dedicated Access Point.

Resource Model - Structure

- 'EasySetup' resource is a collection
  - Easier to get all resources' properties when a GET request with BATCH_INTERFACE is sent to *conf resources



Resource Model: Easy Setup

- Indicates easy setup status

Resource Name	Supported Interface	Example URI	Resource Type	CRUDN permission
EasySetup	Baseline, link-list, batch	/example/EasySetupResURI	oic.r.easysetup, oic.wk.col	RU

Property	Property Name(key)	Value Type	Value Rule	Access Mode	Mandatory	Description
Easy Setup Provisioning Status	ps	integer	enum	R	Yes	Indicates the easy setup provisioning status of the device (0: Need to Setup, 1: Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254: Reserved, 255: EOF)
Last Error Code	lec	integer	enum	R	Yes	Indicates a failure reason if it fails to connect to Enroller (0: NO error, 1: A given SSID is not found, 2: Wi-Fi password is wrong, 3: IP address is not allocated, 4: NO internet connection, 5: Timeout, 6: Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not supported by the Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi Encryption Type is wrong (failure while connecting to the Enroller), 13~254: Reserved, 255: Unknown error)
Connect	cn	array of integer		RW	Yes	Indicates an array of connection types that trigger an attempt to connect to the Enroller to start (1 : Wi-Fi, 2 : Other transport to be added in a future (e.g. BLE))

Easy Setup – Wi-Fi Conf. Resource

- Contains Wi-Fi-related properties

Resource Name	Supported Interface	Example URI	Resource Type	CRUDN permission
Wi-Fi Conf.	Read Write, Baseline	/example/WiFiConfResURI	oic.r.wificonf	RU

Property	Property Name(key)	Value Type	Value Rule	Access Mode	M / O	Description
Supported Wi-Fi Mode Type	swmt	array of string	enum	R	M	Indicates supported Wi-Fi mode types. It can be multiple. (i.e. “A”, “B”, “G”, “N”, “AC”)
Supported Wi-Fi Freq.	swf	array of string		R	M	Indicates supported Wi-Fi Frequency by Enrollee. Can be multiple. (i.e. “2.4G”, “5G”)
Target Network Name	tnn	string		RW	M	Indicates SSID of Wi-Fi AP i.e. Enroller.
Credential	cd	string		RW	M	Indicates credential information of Wi-Fi AP (password used to connect to enroller).
Wi-Fi Auth Type	wat	string	enum	RW	M	Indicates Wi-Fi Auth Type (i.e. “None”, “WEP”, “WPA-PSK”, “WPA2-PSK”)
Wi-Fi Encryption Type	wet	string	enum	RW	M	Indicates Wi-Fi Encryption Type (i.e. “None”, “WEP_64”, “WEP_128”, “TKIP”, “AES”, “TKIP_AES”)
Supported Wi-Fi Auth Type	swat	array of string	enum	R	M	Supported Wi-Fi Auth types. Can be multiple. ("None", "WEP", "WPA_PSK", "WPA2_PSK")
Supported Wi-Fi Encryption Type	swet	array of string	enum	R	M	Supported Wi-Fi Encryption types. Can be multiple. ("None", "WEP-64", "WEP_128", "TKIP", "AES", "TKIP_AES")

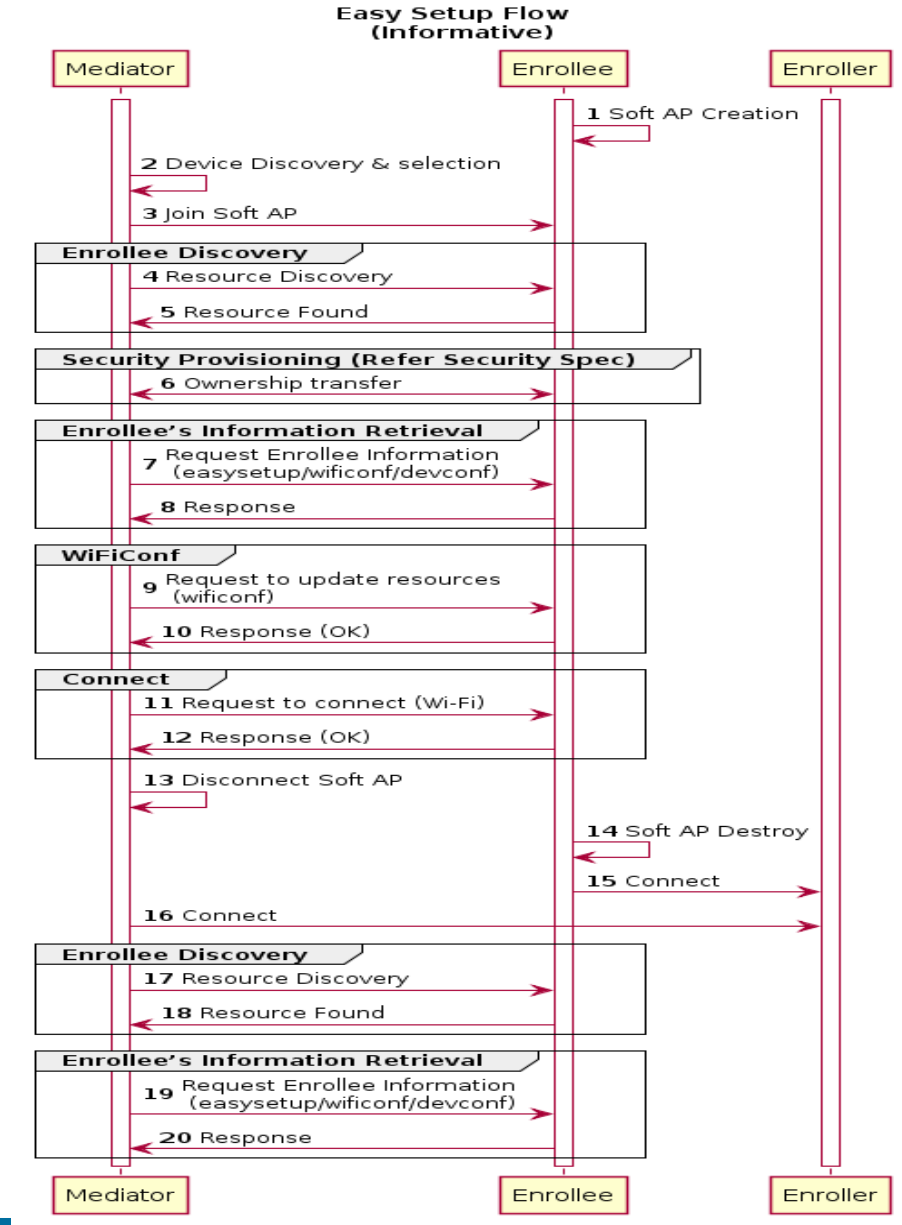
Easy Setup – Dev Conf. Resource

- Store all device configuration required in easy setup process
- Store a device name used in easy setup process

Resource Name	Supported Interface	Example URI	Resource Type	CRUDN permission
Device Conf.	Read Only, Baseline	/example/DevConfResURI	oic.r.devconf	RU

Property	Property Name(key)	Value Type	Value Rule	Access Mode	M / O	Description
Device Name	dn	one of: string or array of object		R	M	Indicates a pre-configured device name in language indicated by 'dl' in /oic/con. or An array of objects where each object has a 'language' field (containing an IETF RFC 5646 language tag) and a 'value' field containing the pre-configured device name in the indicated language. The pre-configured device name is presented by enrollee to mediator during easy-setup process.

Example Easy Setup Flow (informative)



Step1: Enrollee enables SoftAP
Steps 2-3: Mediator connects via the SoftAP
Enrollee Discovery: Steps 4-5:
- Mediator discovers the Enrollee OCF Resources

Security Provisioning: Step 6:
- Ownership Transfer

Enrollee Information Retrieval: Steps 7-8:
- Mediator Retrieves Configuration Resources

Wi-Fi Configuration: Steps 9-10:
- Mediator Updates Configuration Resources

Network Connect: Steps: 11-16
- Mediator instructs Enrollee to connect to configured Wi-Fi
- SoftAP disconnect and disablement

Enrollee Discovery and Retrieval: Steps 17-20:
Mediator discovers via Wi-Fi network

OPEN CONNECTIVITY FOUNDATION®

eSIM Easy Setup

Overview

Overview

- eSIM Easy Setup defines an interoperable data model that can be used to configure a cellular connection on a device using a common communication channel. Wi-Fi Easy Setup can also be used in conjunction with eSIM Easy Setup to configure IP connectivity for the OCF device to download SIM Information from an eSIM Server, known as a SM-DP+.

- **Objectives:**
  - Define data model to be used for eSIM Easy Setup of an unboxed device.
  - Define Device roles and provide informative flow of operation.
  - Reuse existing security mechanism for Device Ownership and Access Control.

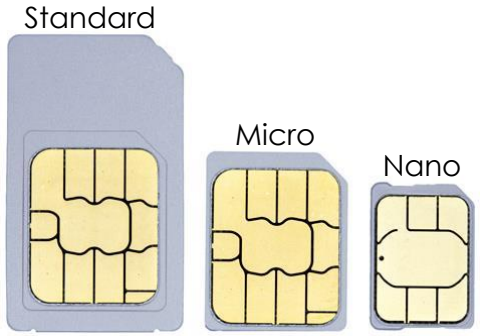
eSIM (standard term is eUICC), a remotely programmable SIM

- Operator is changed by downloading an eSIM Profile(1) from an eSIM Server, known as a SM-DP+.
- To download an eSIM Profile, LPA requires Activation Code which includes SM-DP+ address and Token(2).

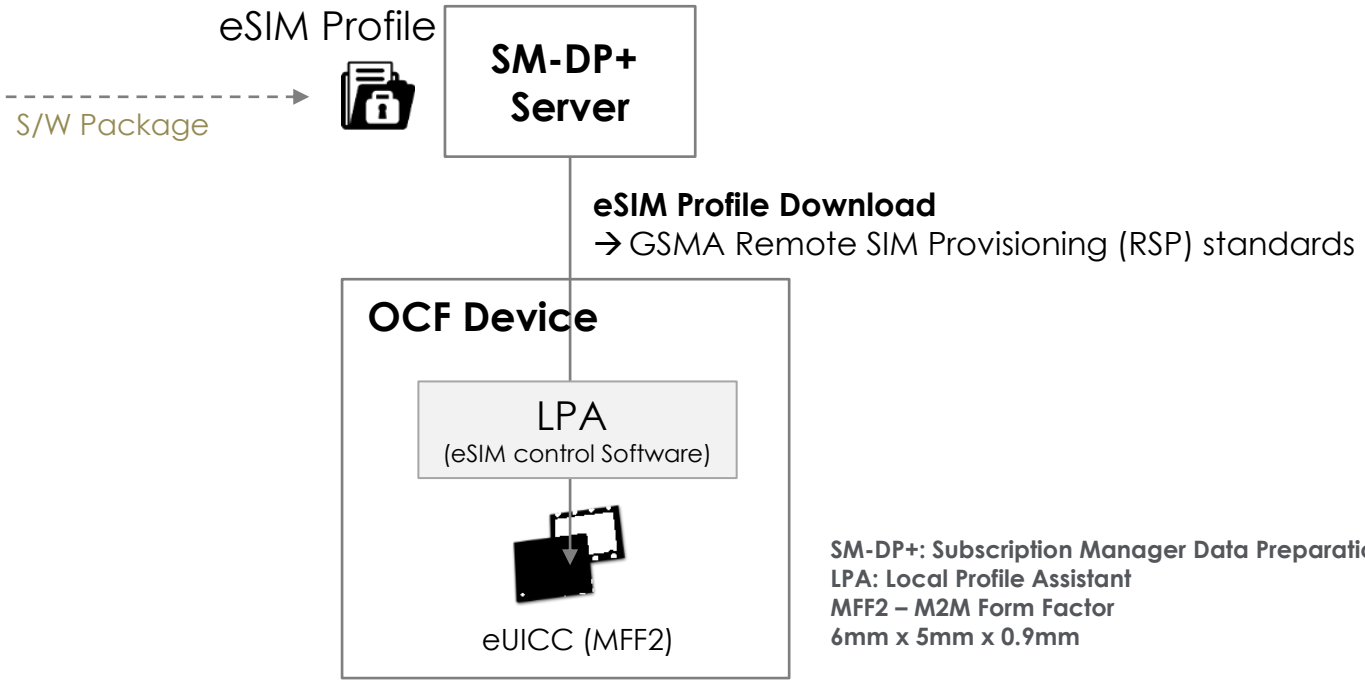
(1) A Profile is equivalent to one traditional SIM card

(2) Referred to as Activation Code Token in GSMA spec. Activation Code Token is an information provided by the Operator/Service Provider to reference a Subscription.

Traditional SIM Cards



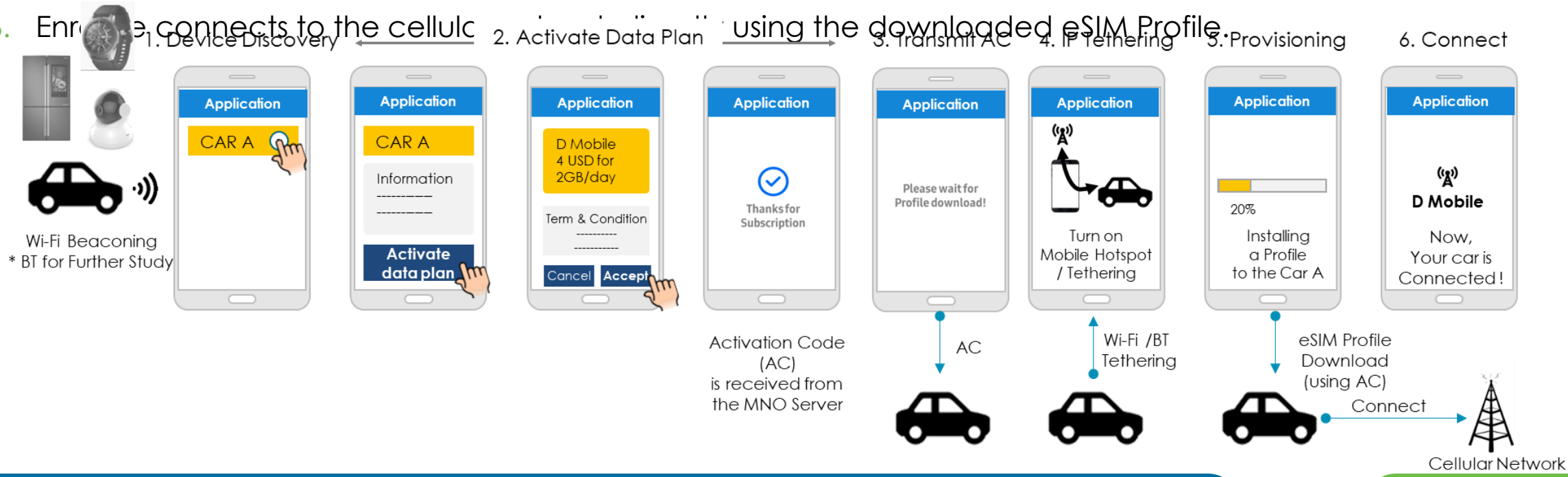
eSIM and Remote SIM Provisioning



SM-DP+: Subscription Manager Data Preparation
LPA: Local Profile Assistant
MFF2 – M2M Form Factor
6mm x 5mm x 0.9mm

Scenario(s) / Use cases

1. Device which supports eSIM Easy Setup is discovered by a Client (Mobile Device).
2. User subscribes to a data plan on Mobile. Service Provider returns an Activation Code(AC).
3. Mobile transmits the AC which includes SM-DP+ address and Token used for Profile download, to the Device.
4. [Optionally] Mobile provides IP connectivity to the Device for eSIM Profile download from SM-DP+ server.
5. From the assigned SM-DP+ server, the Device downloads an eSIM Profile with the Token.
6. Enr[illegible] connects to the cellulc[illegible] using the downloaded eSIM Profile.

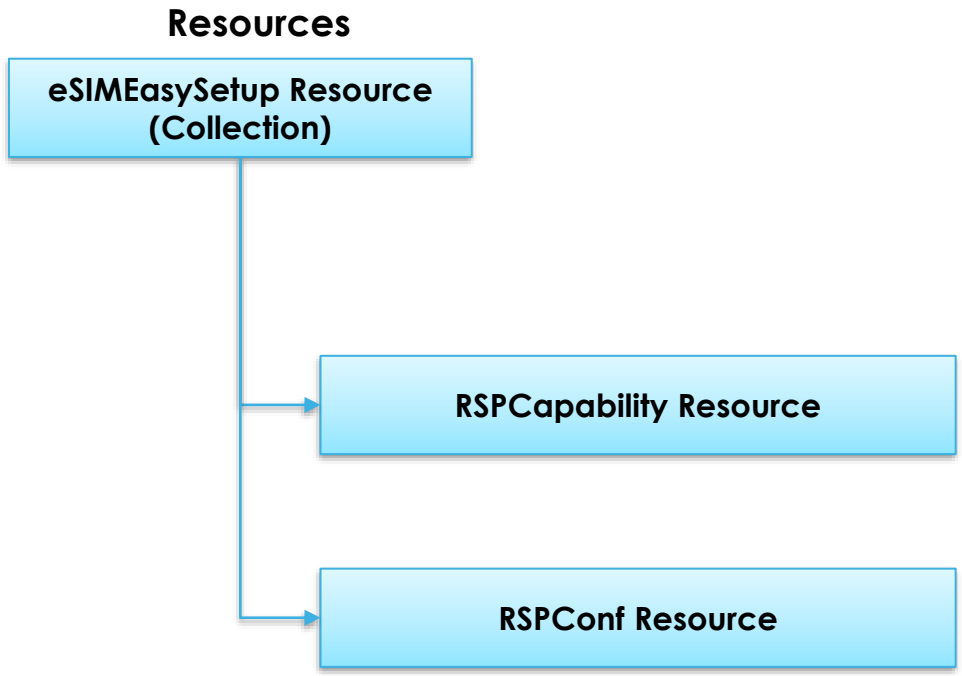


Roles & Definitions

- Mediator
  - Logical function that enables the Enrollee to connect to the target Network. Regarding eSIM Easy Setup, the Mediator transfers configuration information (e.g. Activation Code) to the Enrollee.
  - In case that Wi-Fi Easy Setup is used to provide IP connectivity for Remote SIM Provisioning on to the Enrollee, Mediator for Wi-Fi Easy Setup also acts as an Enroller.
  - Example: Mobile phone/PC
- Enrollee
  - The Device that needs to be configured and connected.
  - Example: eSIM equipped Device such as Watch/Tablet

Resource Model - Structure

- ‘eSIMEasySetup’ resource is a collection
  - Easier to get all resources’ properties when a GET request with “oic.if.b” (Batch Interface) is sent to *conf resources.



Properties

- RSP(Remote SIM Provisioning) Procedure Status
- RSP Last Error Reason
- RSP Last Error Code
- RSP Last Error Description
- RSP End User Consent
- Links

- eUICC Information
- Device Information for RSP

- Activation Code
- eSIM Profile Metadata
- Confirmation Code
- Confirmation Code Required

Resource Model: eSIM Easy Setup

- Indicates Remote SIM Provisioning (RSP) status, and RSP last error code which was produced in the process of eSIM Easy Setup.

Resource Name	Supported Interface	Example URI	Resource Type	CRUDN permission
eSIMEasySetup	Baseline, Link, Batch	/example/eSIMEasySetupResURI	oic.r.esimeasysetup	RUN

Property	Property Name(key)	Value Type	Value Rule	Access Mode	Mandatory	Description
RSP Procedure Status	ps	string	enum	R	Yes	Steps in Remote SIM Provisioning ("Undefined", "Initiated", "User confirmation pending", "Confirmation received", "Downloaded", "Installed", "Error")
RSP Last Error Reason	ler	string	N/A	R	Yes	Error Reason returned during eSIM Easy Setup. It indicates where it occurred (e.g.ES9+.GetBoundProfilePackage(Fail), ES10b.LoadBoundProfilePackage(Fail))
RSP Last Error Code	lec	string	N/A	R	Yes	Error Code returned during eSIM Easy Setup. It indicates why it occurred (e.g. "8.8.1–3.8", "7", "6A 80")
RSP Last Error Description	led	string	N/A	R	No	Optional error description returned during eSIM Easy Setup (e.g. Invalid SM-DP+ Address)
RSP End User Consent	euc	string	enum	RW	Yes	End User Consent for Remote SIM Provisioning ("Undefined", "Timeout", "Download Reject", "Download Postponed", "Download OK", "Download and Enable OK")
Links	links	array	N/A	R	Yes	Array of web links that are RSPCapability Resource and RSPConf Resource

eSIM Easy Setup – RSPCapability Resource

- Contains properties to help a service provider to provide appropriate cellular plans to an end user.

Resource Name	Supported Interface	Example URI	Resource Type	CRUDN permission
RSPCapability	Read, Baseline	/example/RSPCapabilityResURI	oic.r.rspcapability	R

Property	Property Name(key)	Value Type	Value Rule	Access Mode	M / O	Description
eUICC Information	euiccinfo	string	Max.1024 octets	R	M	eUICC information used for the eSIM Profile download and installation procedure. Refers to "EUICCInfo2" defined in the GSMA RSP Technical Specification Annex H. This value type shall be encoded as Major Type 2.
Device Information for RSP	deviceinfo	string	Max.128 octets	R	M	Device information used for the eSIM Profile download and installation procedure. Refers to "DeviceInfo" defined in the GSMA RSP Technical Specification Annex H. This value type shall be encoded as Major Type 2.

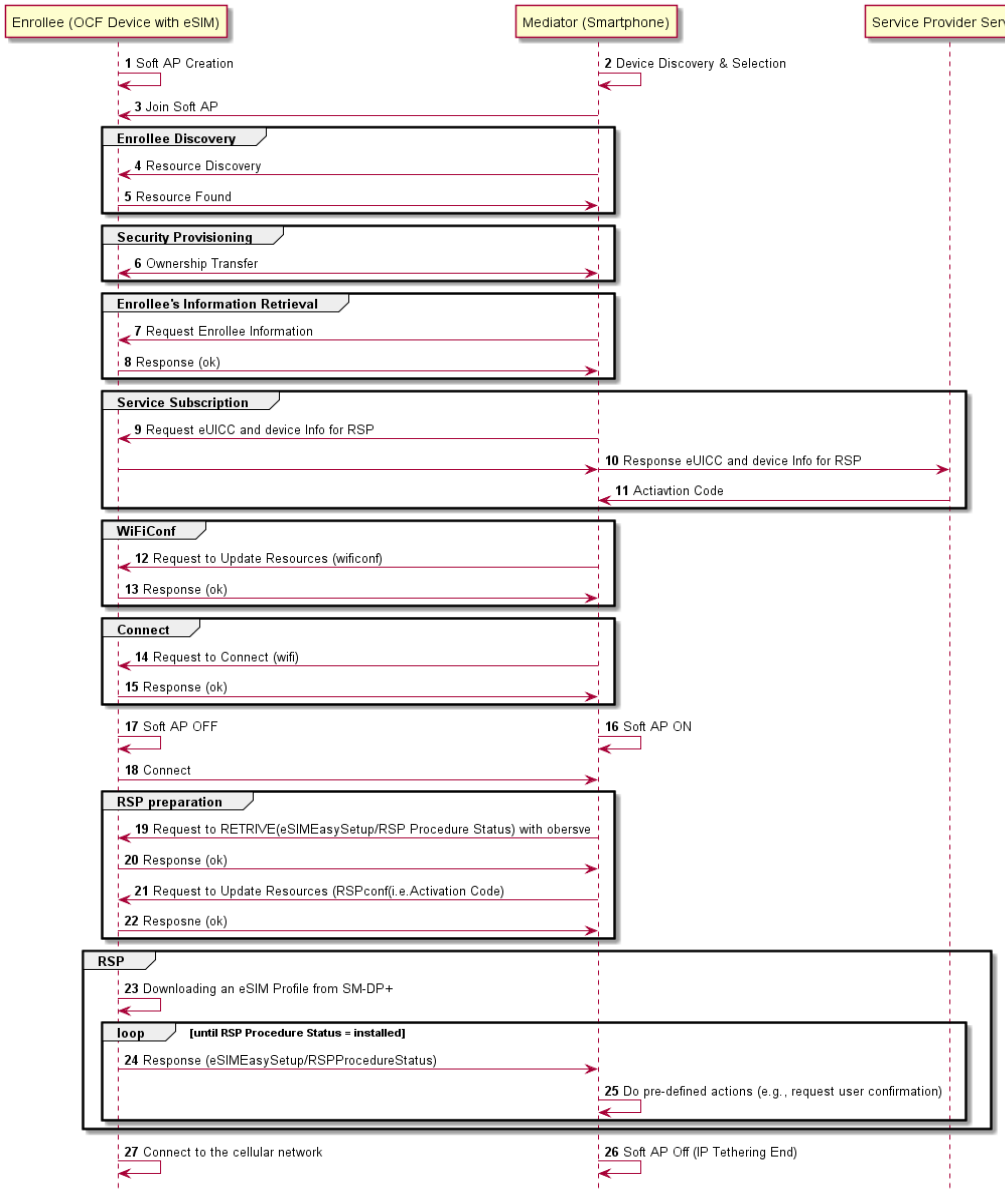
eSIM Easy Setup – RSPConf Resource

- Contains the properties used to download and install an eSIM Profile to an eSIM capable OCF device.

Resource Name	Supported Interface	Example URI	Resource Type	CRUDN permission
RSPConf	Read Write, Baseline	/example /RSPConfResURI	oic.r.rspconf	RU

Property	Property Name(key)	Value Type	Value Rule	Access Mode	M / O	Description
Activation Code	ac	string	Max.256 characters	RW	M	The information needed to provision an eSIM device. Comprises SM-DP+ server FQDN(Fully Qualified Domain Name) and Activation Code Token binding to a specific subscription as defined by the GSMA RSP Technical Specification
eSIM Profile Metadata	pm	string	Max. 2048 octets	R	M	Refers to "ProfileInfo" in the GSMA RSP Technical Specification Annex H. This value type shall be encoded as Major Type 2.
Confirmation Code	cc	string	N/A	RW	O	A code entered by an end user required by the SM-DP+ to confirm the download and installation of an eSIM Profile. The Confirmation Code is provided from a service provider to the end user.
Confirmation Code Required	ccr	boolean	N/A	R	M	Indicates whether a Confirmation Code is required. Set to "true" if Confirmation Code is required and required a user to enter Confirmation Code

Example eSIM Easy Setup Flow with Wi-Fi Easy Setup (informative)



Soft AP Connect: Steps1-3:

Mediator connects Enrollee via the SoftAP

Enrollee Discovery: Steps 4-5:

Mediator discovers the Enrollee OCF Resources (e.g. eSIM Easy Setup, Wi-Fi Easy Setup)

Security Provisioning: Step 6:

Ownership Transfer

Enrollee Information Retrieval: Steps 7-8:

Mediator Retrieves the eSIM Easy Setup and Easy Setup Resources

Service Subscription on the Mobile: Steps 9-11:

Mediator Retrieves Device and eUICC information which is used for a service provider to provide cellular plans to select from. As a result of the contract, the service provider server sends an Activation Code.

Network Connect for IP tethering: Steps 12-18:

Mediator Updates Wi-Fi Easy Setup Configuration Resources.
Mediator instructs the Enrollee to connect to configured Wi-Fi SoftAP.
Enrollee terminates SoftAP, and connect Mediator via the Wi-Fi SoftAP.

RSP(Remote SIM Provisioning) Preparation: Steps 19-22:

Mediator uses Notify operation to the eSIMEasySetup Resource.
Mediator Updates eSIM Easy Setup Resource with Activation Code.

RSP(Remote SIM Provisioning) in Progress: Steps 23-25:

As receiving Activation Code, Enrollee starts downloading an eSIM Profile from the SM-DP+ server. When the RSP Procedure Status ("ps") Resource value changes, Enrollee sends a Notification to the Mediator.

Cellular Network Connect from Enrollee: Steps 26-27:

If successfully Installed (i.e. when "ps" turns to "installed"), Mediator terminates the Soft AP, and then leaves the eSIM Easy Setup mode. Enrollee connects to the cellular network directly.

OPEN CONNECTIVITY FOUNDATION®

OCF Specification Overview
OCF Cloud API for Cloud Services

OCF 2.2.4 Release

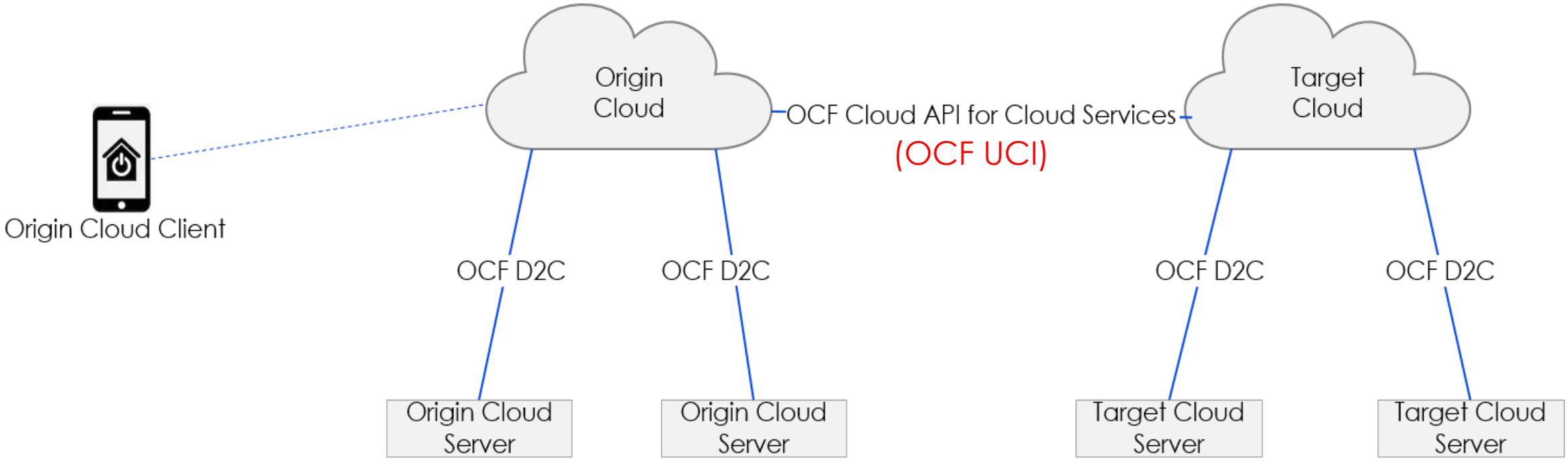
October 2021

Overview

- OCF Cloud Services Overview
- API Design Concept
- List of OCF Cloud APIs for Cloud Services
- General Flows
  1. Initial Association
  2. Device/Resource Discovery
  3. Resource Control
  4. Event Subscription & Notification
- Document Links

OCF Cloud Services Overview

- A representation of the target architecture



C2C = Cloud to Cloud as defined by the OCF Cloud API for Cloud Services
D2C = Device to Cloud as defined by the OCF Device to Cloud Services Spec

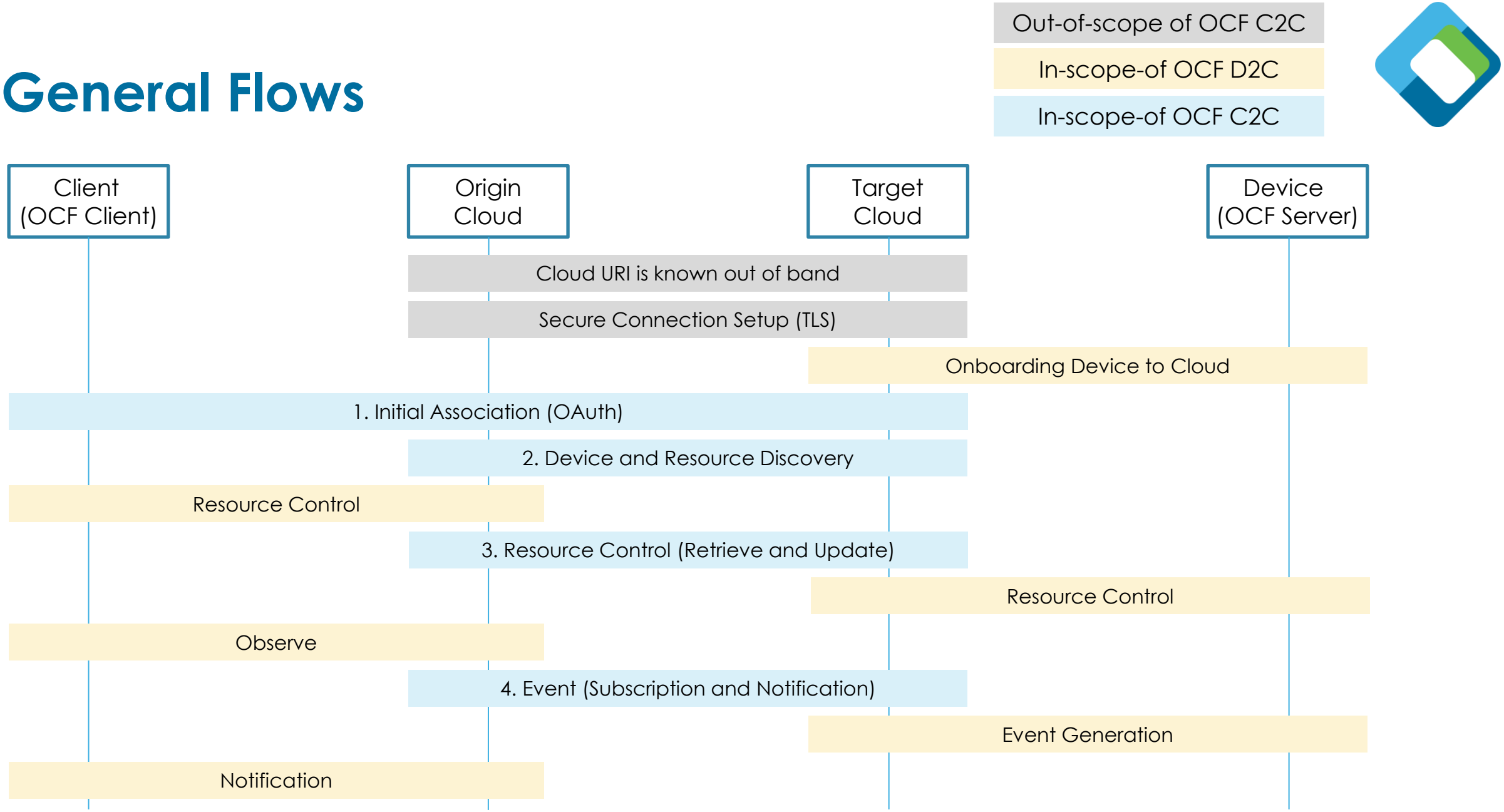
API Design Concept

- OAuth 2.0 (IETF RFC 6749) based Cloud Account Linking
- Secure connection (TLS) between Clouds
- HTTPS Request/Response Message for retrieving and updating devices/resources
- JSON defined payload to carry OCF-defined Device/Resource Models

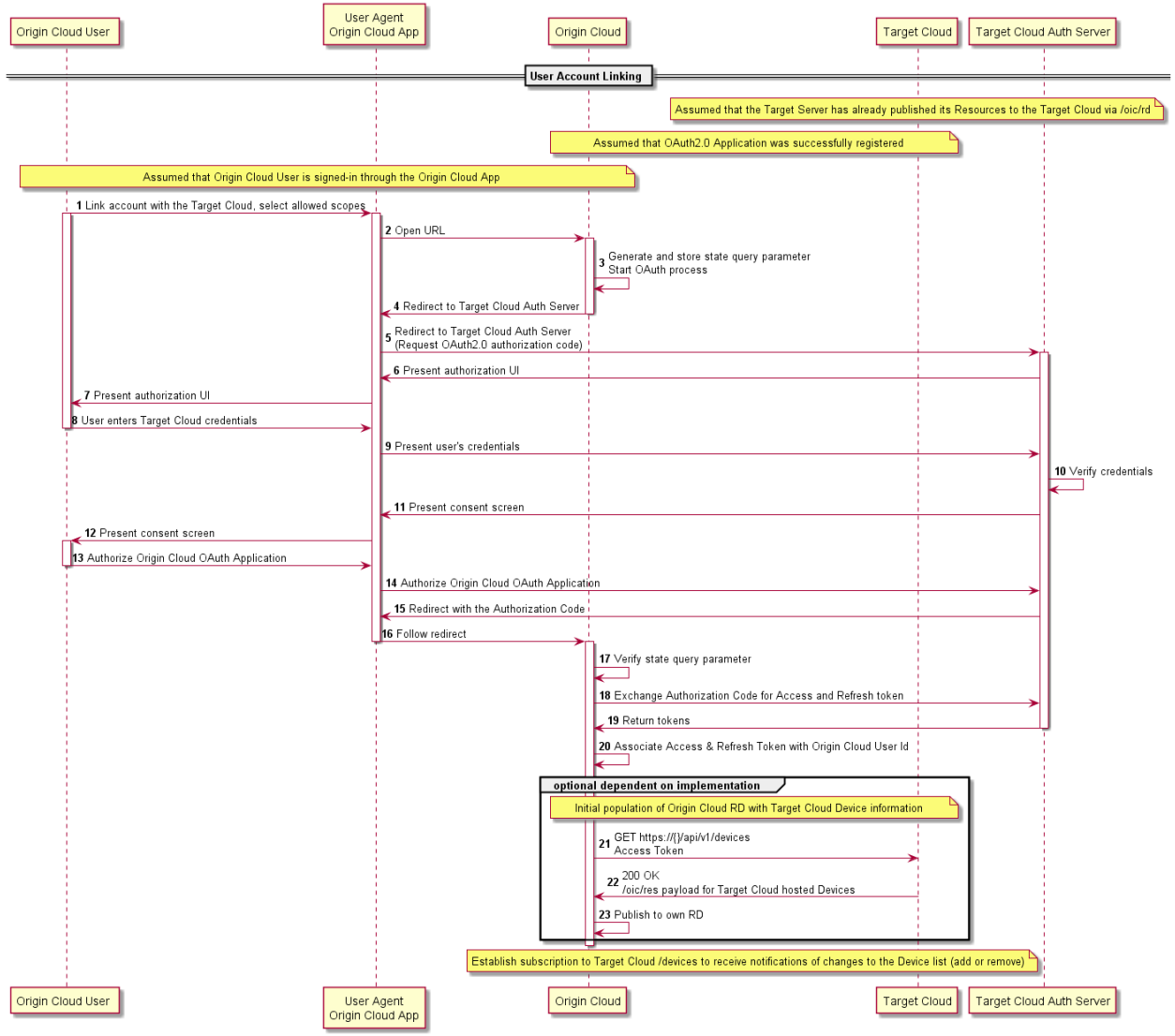
List of OCF Cloud APIs for Cloud Services

- Account Linking API
  - Initial Account Linking
  - Removal of linked account
- Devices API
  - Retrieval of all Devices associated with a User
  - Retrieval of a single Device associated with a User
  - Retrieval of a single Resource
  - Update of a single Resource
- Events API
  - Subscription to events: establishment and cancellation of a subscription
  - Notification: event generated on an established subscription
    - Event set: device registered/unregistered, device online/offline, resource published/unpublished, resource content changed

General Flows

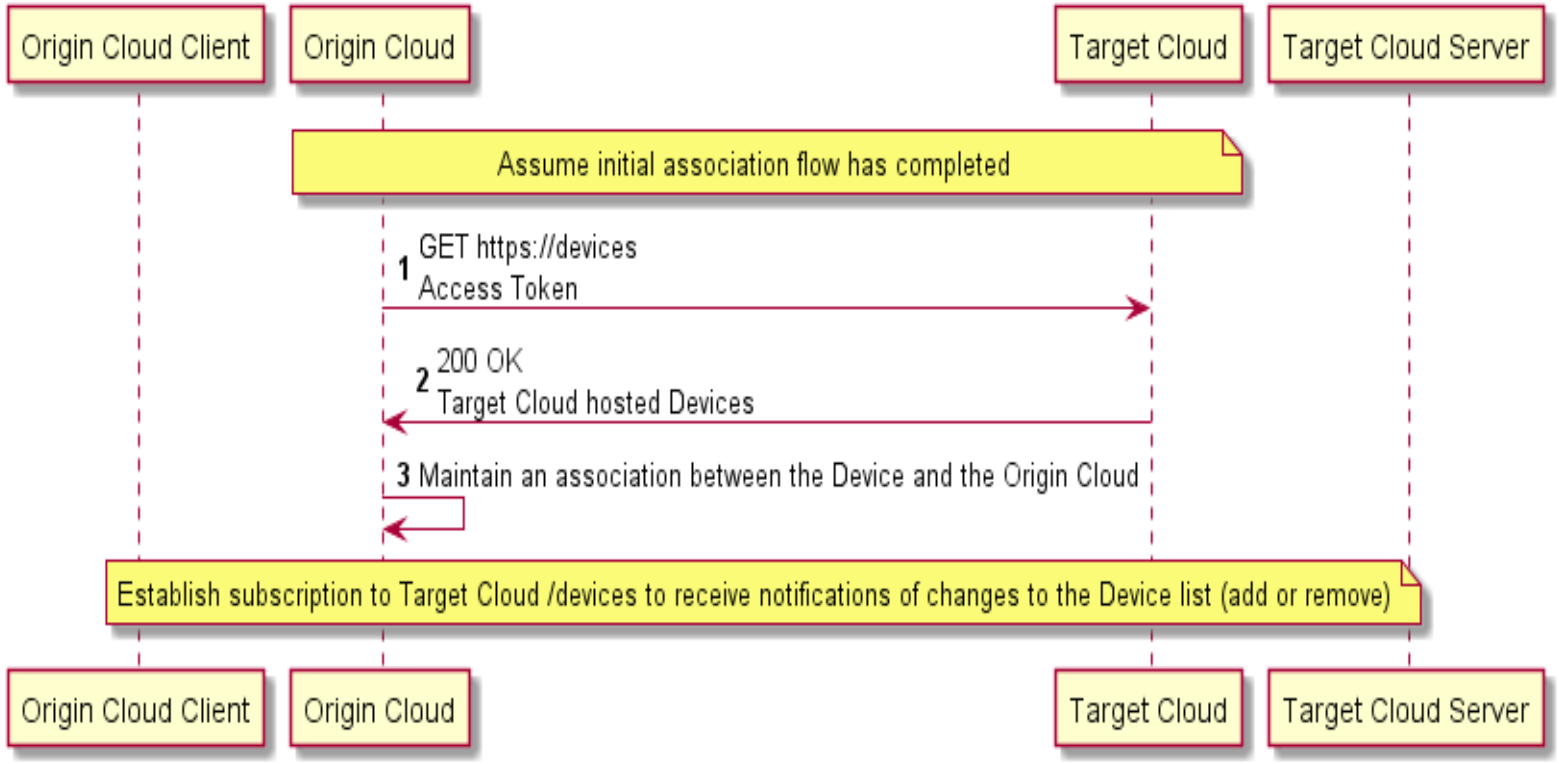


1. Initial Association



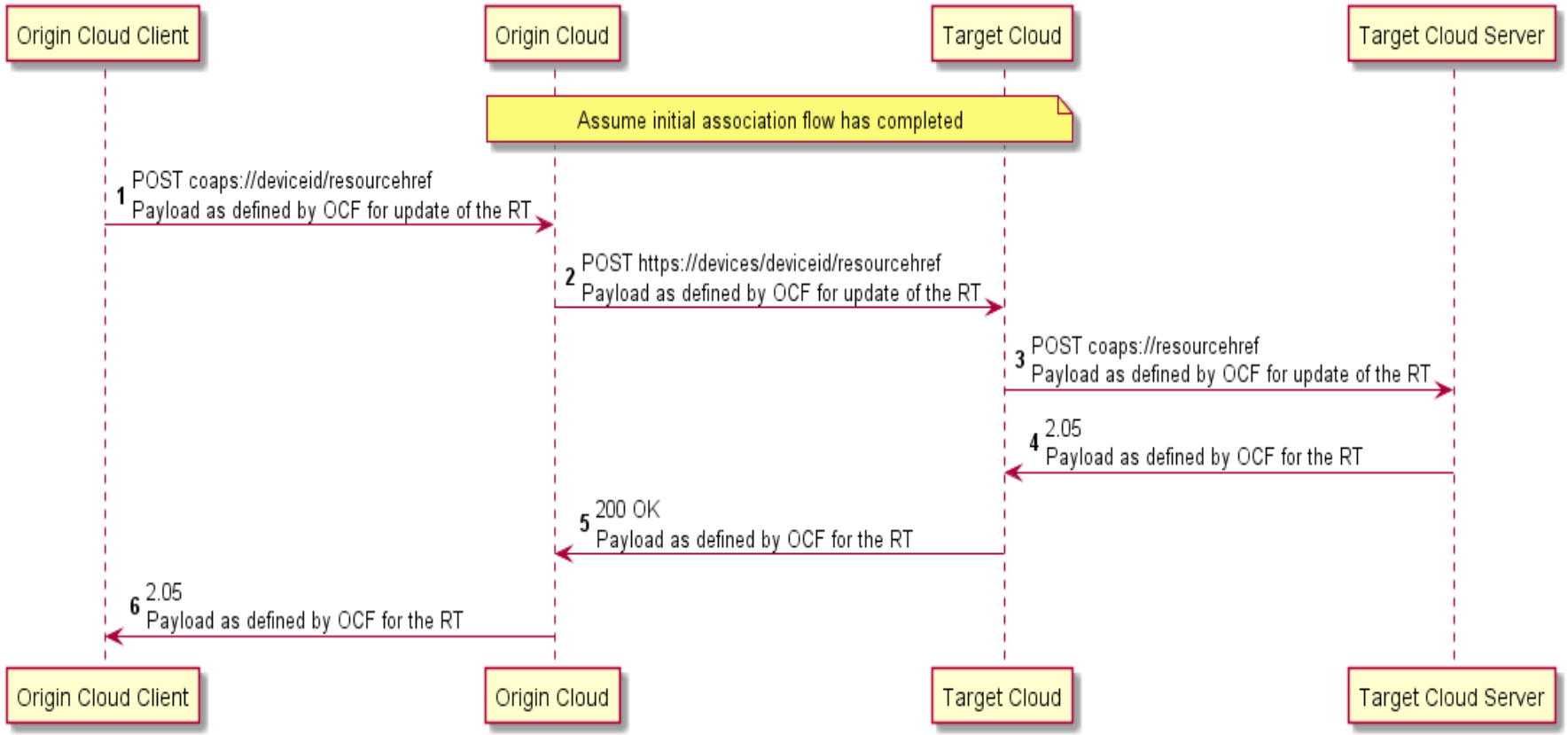
2. Device/Resource Discovery

- Retrieval of all Devices example



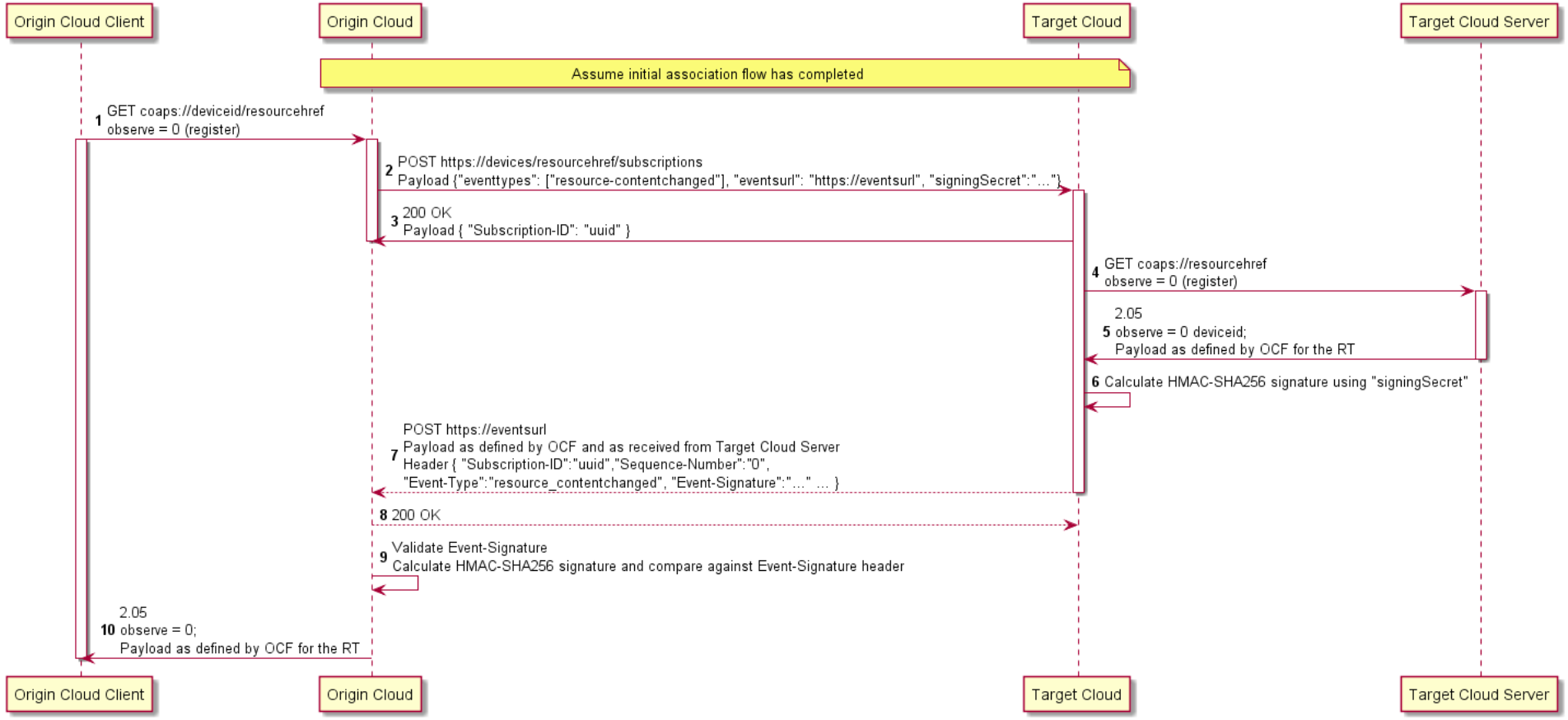
3. Resource Control

- Update of a single Resource example



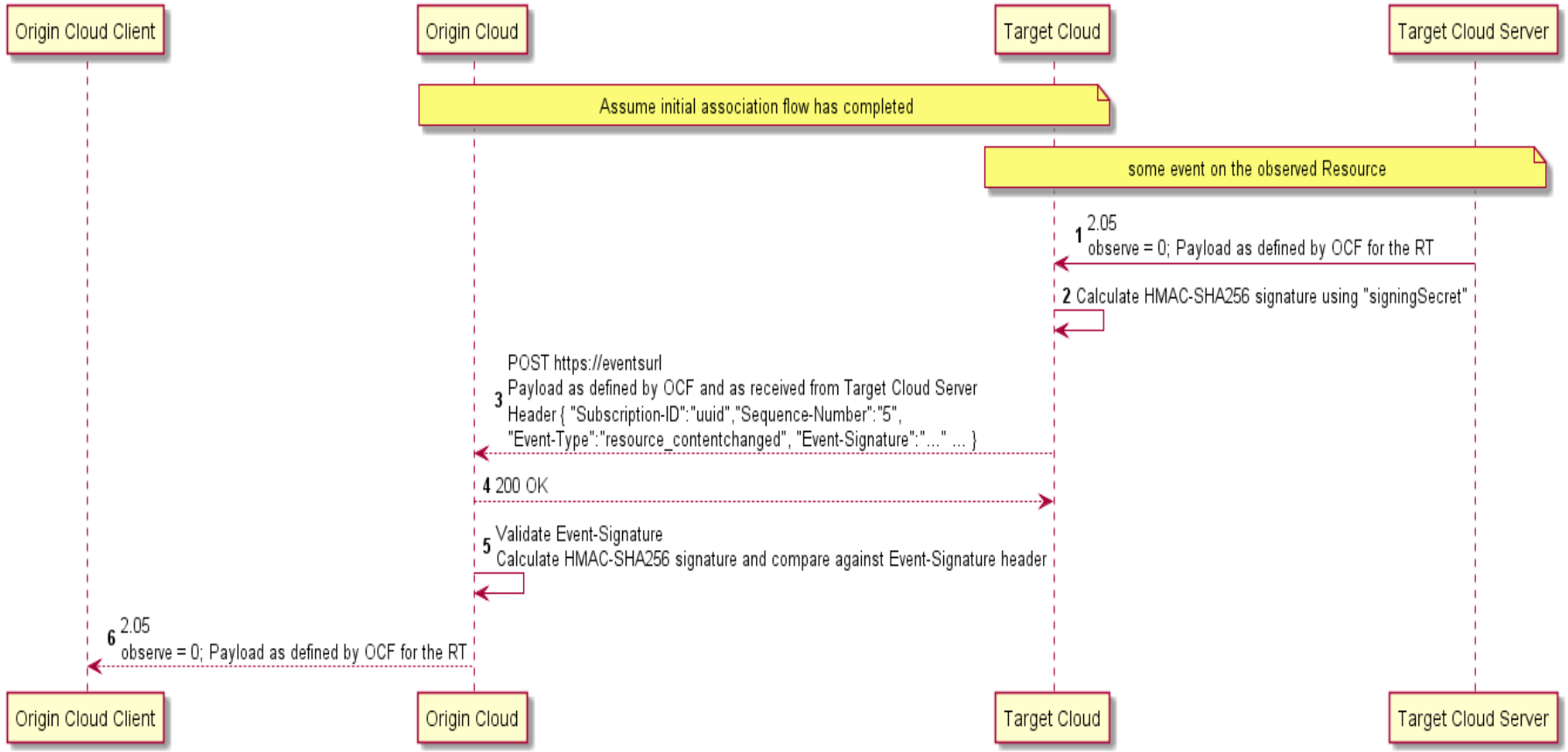
4-1. Event Subscription

- Observe Establishment Example



4-2. Event Notification

- "Resource_Contentchanged" Event Example



Document Links

- OCF Specifications:
  https://openconnectivity.org/developer/specifications

- OpenAPI definition of the API (Github):
  https://github.com/openconnectivityfoundation/core-extensions/blob/ocfcloud-openapi/swagger2.0/oic.r.cloudopenapi.swagger.json

- Rendered for easy go through:
  https://petstore.swagger.io/?url=https://raw.githubusercontent.com/openconnectivityfoundation/core-extensions/ocfcloud-openapi/swagger2.0/oic.r.cloudopenapi.swagger.json

OPEN CONNECTIVITY FOUNDATION®

OCF Specification Overview Security Specification

OCF 2.2.4 Release

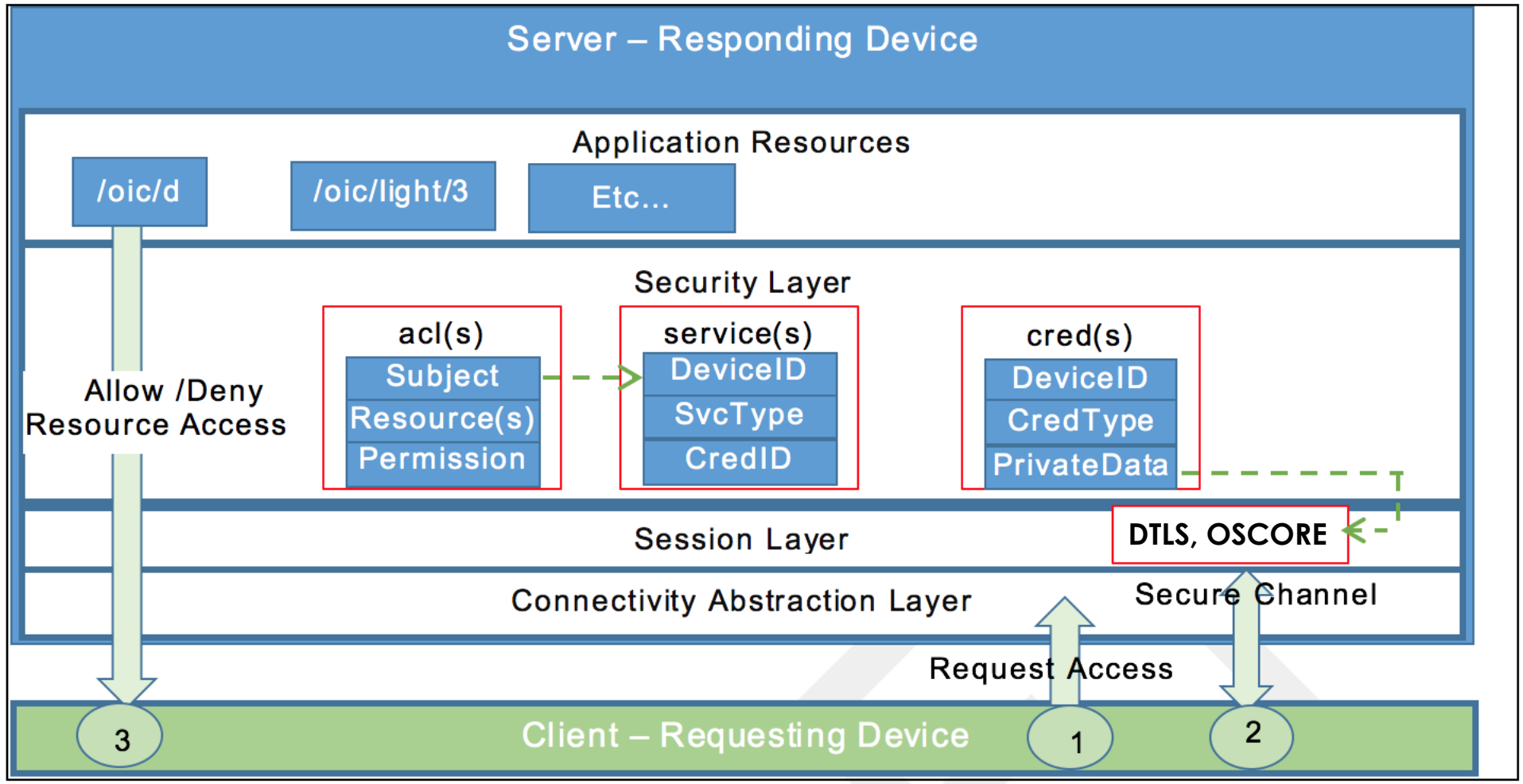
OCF Security Summary

- OCF is concerned with
  - **Device Identity** (Immutable, Unique, Attestable)
  - **Onboarding** (including **Authentication, Authorization, & Auditing (AAA)**)
  - **Confidentiality** (Protect data and communications)
  - **Integrity** (Resources, device state, and transitions are all managed)
  - **Availability** (not only at the device level but also secured so they don't impact the networks within which they operate)
  - **Lifecycle Management** (Including secure software update and verifications mechanisms)
  - **Future Security** (Looking at credential types, algorithms, and adapting to changes in the security landscape as it relates to the security of OCF devices, now and in the future)
- OCF key management supports device protection and authentication
- OCF uses Access Control Lists (ACLs) to manage authorization
- Secure device ownership transfer helps prevent attacks when devices are added to the network

Security Principals

- **Resources**: a data structure that defines the types, data and interfaces of a device; each can be Created/Retrieved/Updated/Deleted or to which Notification can be set based on appropriate access control
- **Access Control Entries (ACEs) and Access Control Lists (ACLs)** are entries and collections, respectively, of permissions granting one device access to a Resource.
- **Onboarding Tools (OBTs)** are OCF Devices that help bring other OCF Devices into the local network. The OBTs are collections of services, some of those are listed below:
  - **Access Manager Service (AMS,** oic.d.ams**)** creates and verifies access control permissions.
  - **Credential Management Service (CMS ,** oic.d.cms**)** is the name and resource type for a device which is granted permission to create and manage security credentials.
  - **Mediator** provisions the OCF Device with information necessary for remote service management.
  - **Device Ownership Transfer Service (DOTS ,** oic.d.dots**)** onboards the OCF Device**.**
  - **Ownership Transfer Mechanism (OTM)** is method of onboarding (e.g. using cert for authentication).
- **Secure Virtual Resources (SVRs)** are special security resources with severely restricted permissions and access management.

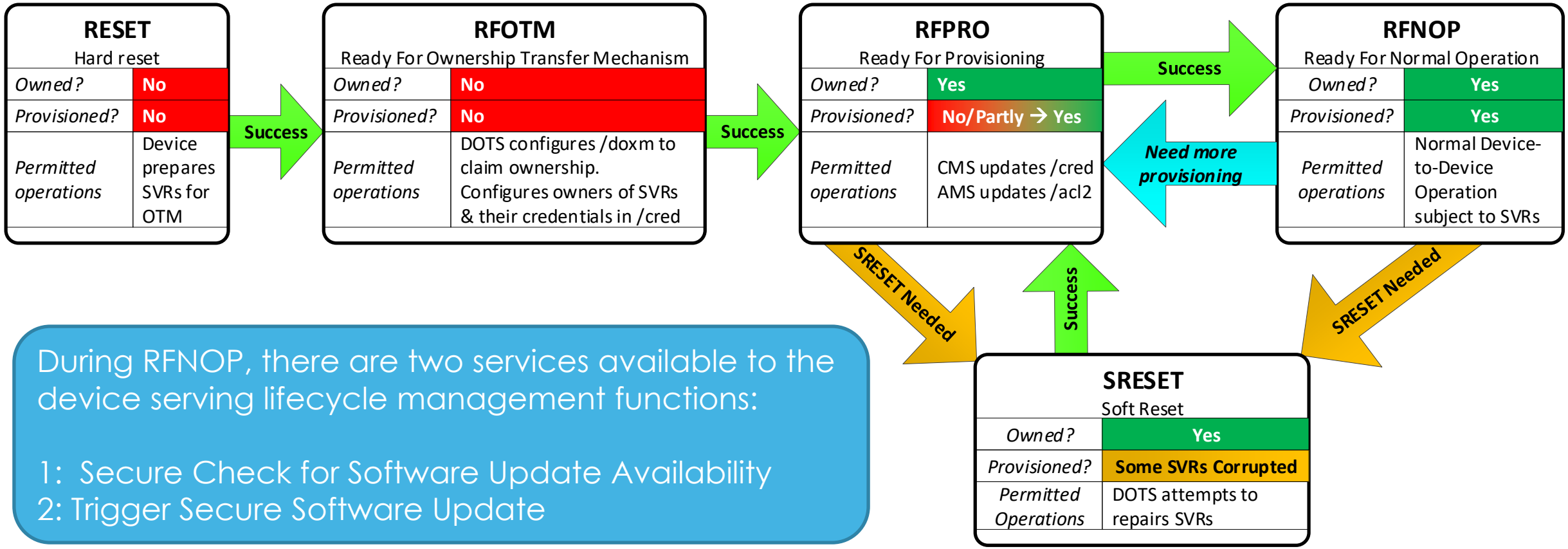
How OCF Security Protects Device Resources:



Simplified Onboarding Sequence

- *Unowned Device boots*
- **Discovery (unsecured):**
  - DOTS sends multicast to discover unowned devices — no TLS
  - Unowned devices reply, including list of supported OTMs — no TLS
- **Ownership Transfer:**
  - DOTS selects and configures this OTM to the new device — no TLS
  - DOTS & unowned Device perform OTM, inc. TLS handshake — TLS
  - DOTS configs SVRs to authorize itself, CMS and AMS — TLS
  - *Device is now owned!*
- **Provisioning:**
  - CMS provisions credentials, AMS provisions access policies — TLS
  - *Device is now provisioned and can commence normal operation*
- **Normal Operation:** — TLS , TLS+OSCORE or no TLS
  - *Credentials and/or access policies can be updated by returning to Provisioning*

Device Provisioning States



During RFNOP, there are two services available to the device serving lifecycle management functions:

1: Secure Check for Software Update Availability
2: Trigger Secure Software Update

*Device can transition to **RESET** from any state (these transitions are not shown)*

Credential Management

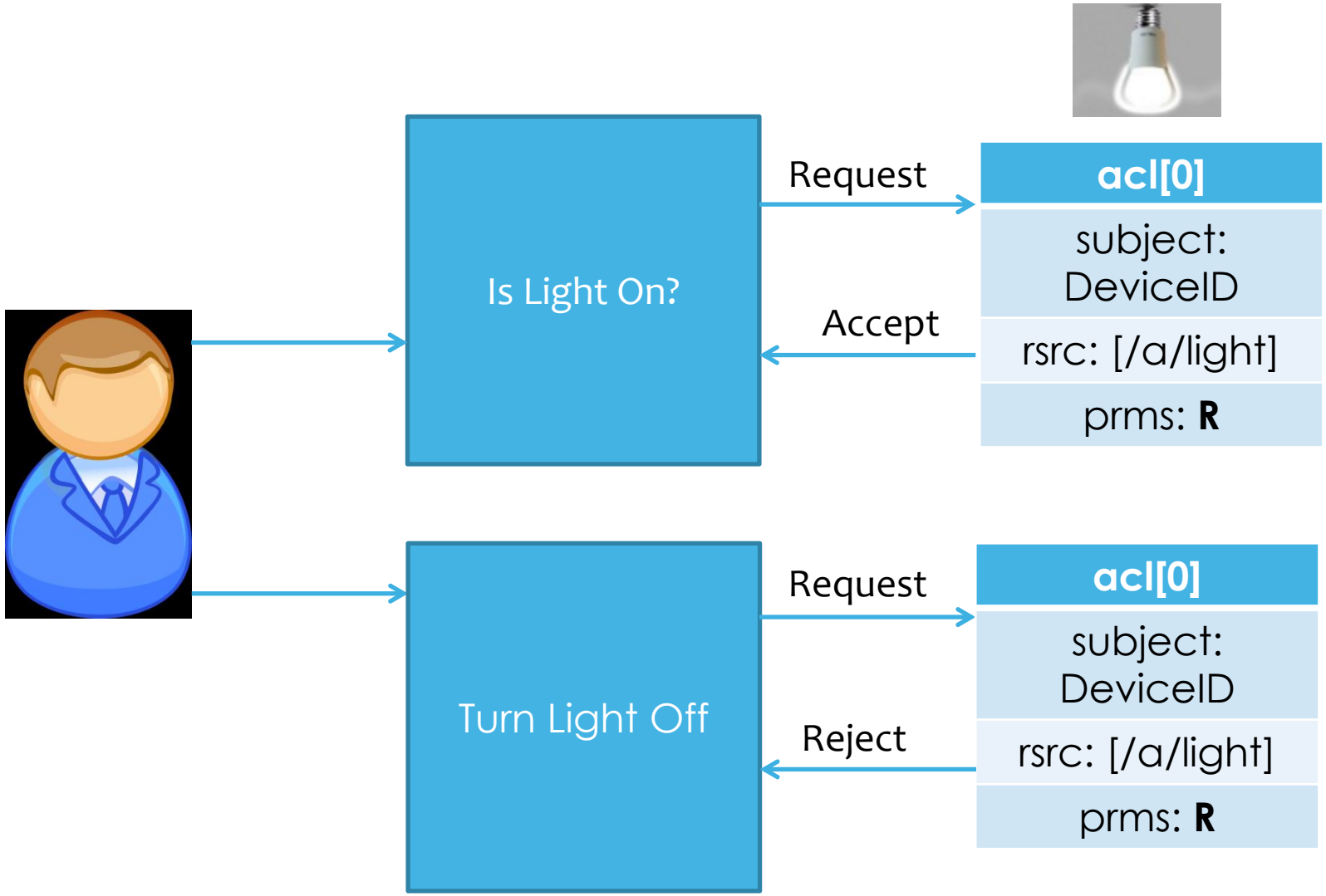
OCF devices support multiple credential types:

- Pairwise Secret Key
- Asymmetric credentials:
  - Raw asymmetric key
  - Identity certificate w/ key pair – used to establish a secure session
  - Role certificate w/ key pair – used to assert a role within a session

Missing credentials could be procured from a CMS

Credentials can be configured to have an expiration period

Access Control



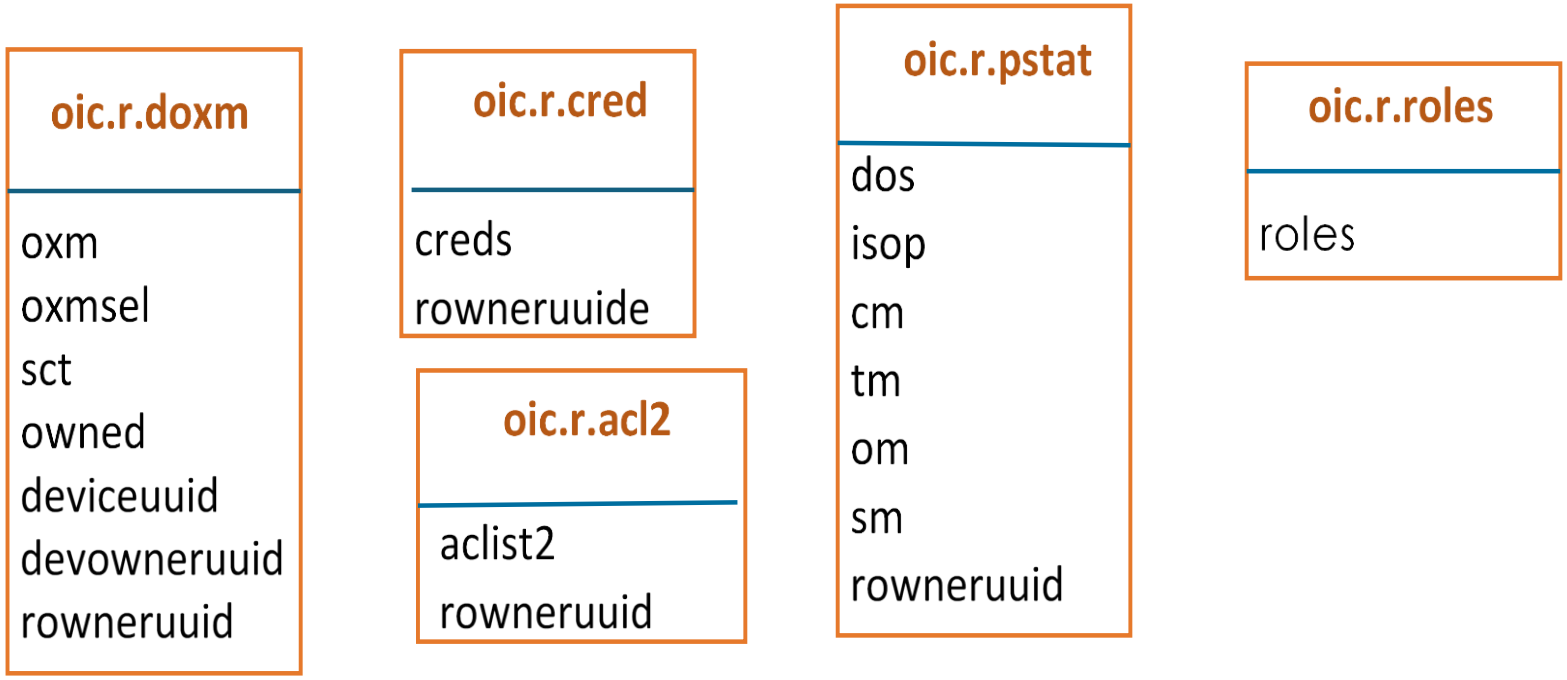
Access Control

- Protect Resources of the OCF Server to control CRUDN access for entity requesting access
  - Any request to the OCF Server is subject to ACL(Access Control List) policy check
  - ACE (Access Control Entry) policy applies to a OCF Server hosted Resource
  - Each ACE has a permission which allows read or write operation
- Two type of access control mechanism are supported:
  - Subject-based access control (SBAC)
    - ACE specifies the identity of requestor
  - Role-based Access Control (RBAC)
    - ACE specifies the role to accept of the entity requesting access
- ACL can be changed/updated via the AMS
  - Wildcards are supported to ease ACL management
- ACL policies applies only at the OCF server side
- When Client is authorized to CREATE a Resource in a Collection, Server autonomously changes ACL to give Client full read/write access that Resource

Security Virtual Resource (SVR)

- OCF defines SVRs (Security Virtual Resource) to perform OCF security related functionality
  - "Virtual" is an artefact of legacy resource naming. It is in fact a full-fledged OCF resource
- Device Ownership Transfer Resource (/oic/sec/doxm) manage Device Ownership status
- Provisioning Resource (/oic/sec/pstat) manage Device Provisioning status
- Credential Resource (/oic/sec/cred) manages Device credentials
  - Credential Resource is used for establishing secure communication
  - Subject ID has to match the ID of connecting OCF Device
  - For oic.sec.cred.cert entries, also known as identity certificates, Common Name of the End-Entity certificate has to match the UUID of the OCF Device presenting it
  - For oic.sec.cred.trustca entries, also known as trust anchors for identity certificates, Subject ID has to match the UUID of connecting OCF Device
  - Subject ID is used to verify identity of the OCF Devices and can be matched to ACLs
- Access Control List (/oic/sec/acl2) manages the Access Control Entry for the Resource Server

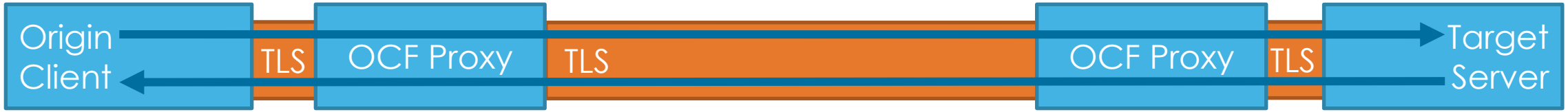
Security Virtual Resource (SVR)



Message Integrity and Confidentiality

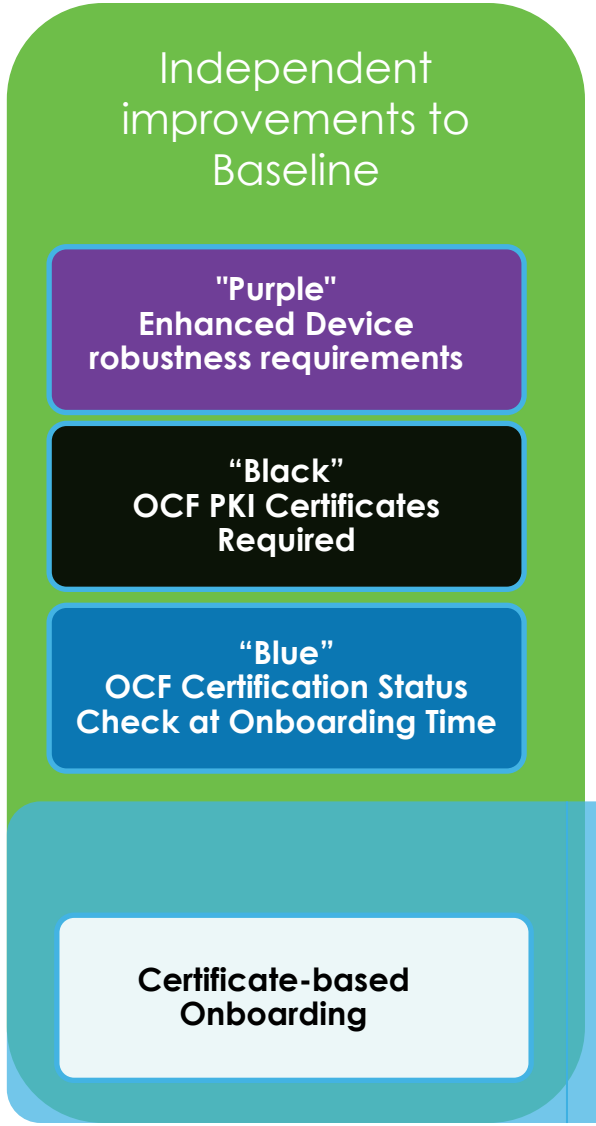
- Secured communications between OCF Devices are protected against eavesdropping, tampering, and message replay.
- Unicast messages are secured using DTLS or TLS. Multicast messages are not secured.
- All communications are signed and encrypted.
- Communicating devices are required to authenticate each other. Communicating devices need to have useable credentials to talk to each other. If they are missing, the devices could contact the CMS to get them.

End-to-End security of unicast messages using OSCORE



- OCF Proxies (e.g. OCF Cloud) can interpret OCF compliant URIs of unicast request messages intended for resources on another Target Server and route accordingly
- Even if DTLS/TLS applied, message content is neither encrypted nor integrity protected in OCF Proxies
- In some scenarios, users may be reluctant to trust OCF Proxies with confidentiality, integrity and freshness of OCF CRUDN messages
- OSCORE provides end-to-end security (Origin Client to Target Server) so OCF Proxies do not need to be trusted
- OSCORE encrypts and integrity protects OCF CRUDN messages
  - Exceptions are fields used by OCF Proxies for delivery purposes (e.g. routing).
- OSCORE credential: symmetric key provisioned by CMS to Origin Client & Target Server
- Target Server applies access control using identity configured in OSCORE credential

3 New Security Profiles for OCF 2.0



- **Optional and Certifiable improvements to Baseline Security Profile**
  - Black requires an audited CA, Blue/Purple require a vetted CA. All three include significant improvements to Device security such as hardware key storage, improved cipher suite support, etc.
  - A Device may be certified as conforming to any combination of Profiles. (e.g. Blue & Purple; Black only; Black & Blue & Purple; etc.)
- **Interoperable:** Devices of different Profiles can co-exist and interoperate.
- **Cryptographically Attestable:** Certificate extensions allow encoding of security attributes and OCF certification information.
- **Consistent:** No change to OCF branding due to Security Profiles.

Manufacturer Incentives to Use Security Profiles

- Purple: Manufacturer building a Device with requirement for measured boot and secure SW update, to improve device integrity (e.g. connect to cloud, healthcare or government).
- Black: Manufacturer wishing to require use of OCF PKI, which ensures certificates are signed by OCF PKI, and meet OCF Certificate Policy.
- Blue: Manufacturer wishing to use its own (or other non-OCF) CA, which must conform to OCF-defined CA vetting criteria.

"Purple"
Enhanced Device robustness requirements

"Black"
OCF PKI Certificates Required

"Blue"
OCF Certification Status Check at Onboarding Time

Additional Commentary on Security Profiles

- The **BLACK** profile requires use of the OCF PKI, using certificates that all share a common OCF root. Issuance of a black-profile Certificate requires that the Device has passed OCF certification, and requires that the CA issuing the certificates undergoes a successful audit of their CA process (OCF Certificate Policy based on WebTrust for Certificate Authorities v2.1).
- The **BLUE** profile has certificates issued by CAs that have passed a successful audit of their CA process. The Certification Status of the Device is verified at onboarding time against the OCF's Certification Management System. Currently, an extensible model for distributing audited CA's public roots to the OBTs is under design, but shorter-term, a list of vetted Roots can be found in the OCF Security Specification.
- The **PURPLE** profile adds some additional attestations that the manufacturer is asserting, related to the integrity of the device. These attestations are on file with the OCF Certification Working Group, and are identified specifically inside the Certificate.
- For further details on the granular differences between these profiles, please see the OCF Security Specification v2.0.2

OCF Cloud (optional feature)

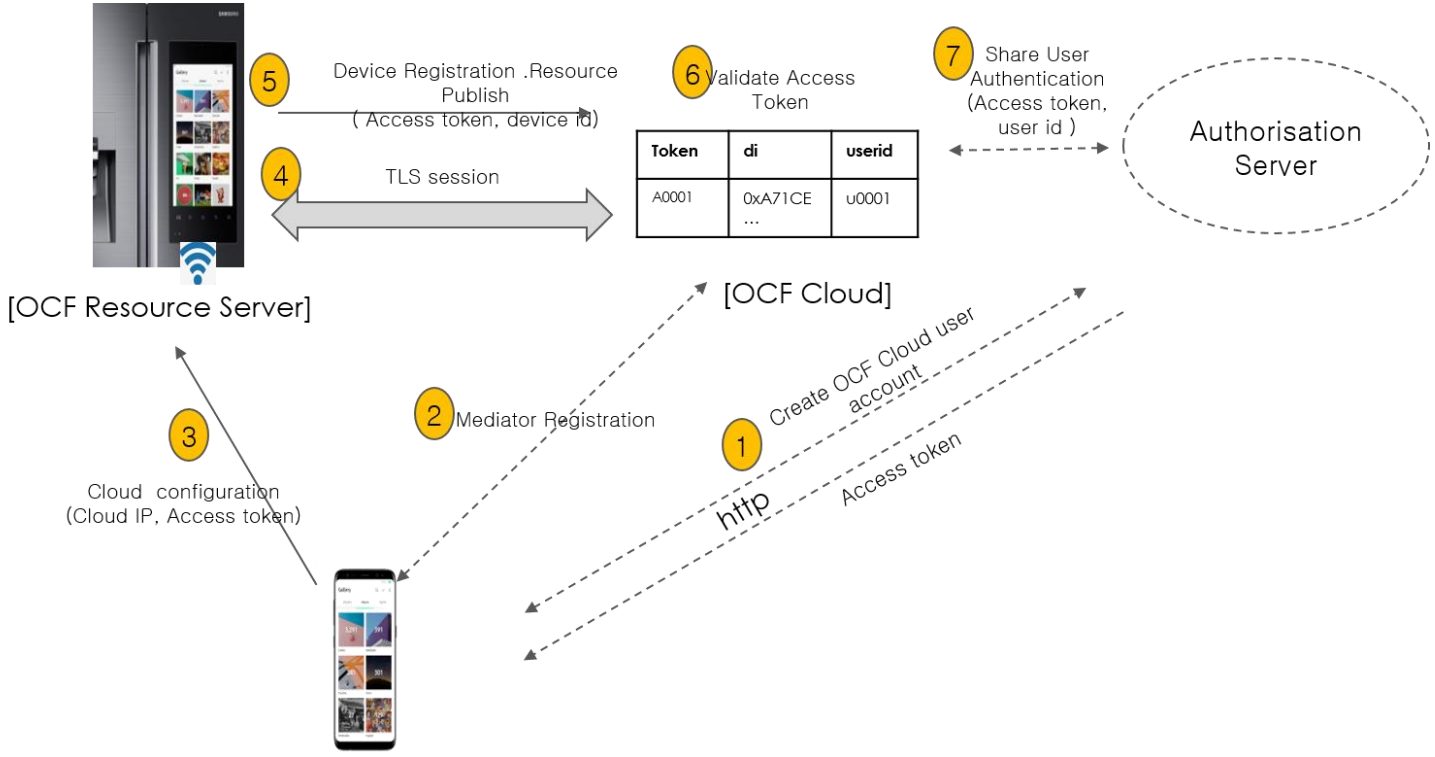
OCF Cloud enables Device interaction with cloud-hosted Resource Directory

3 SVRs are hosted on the cloud:

- Account
- Session
- Token refresh

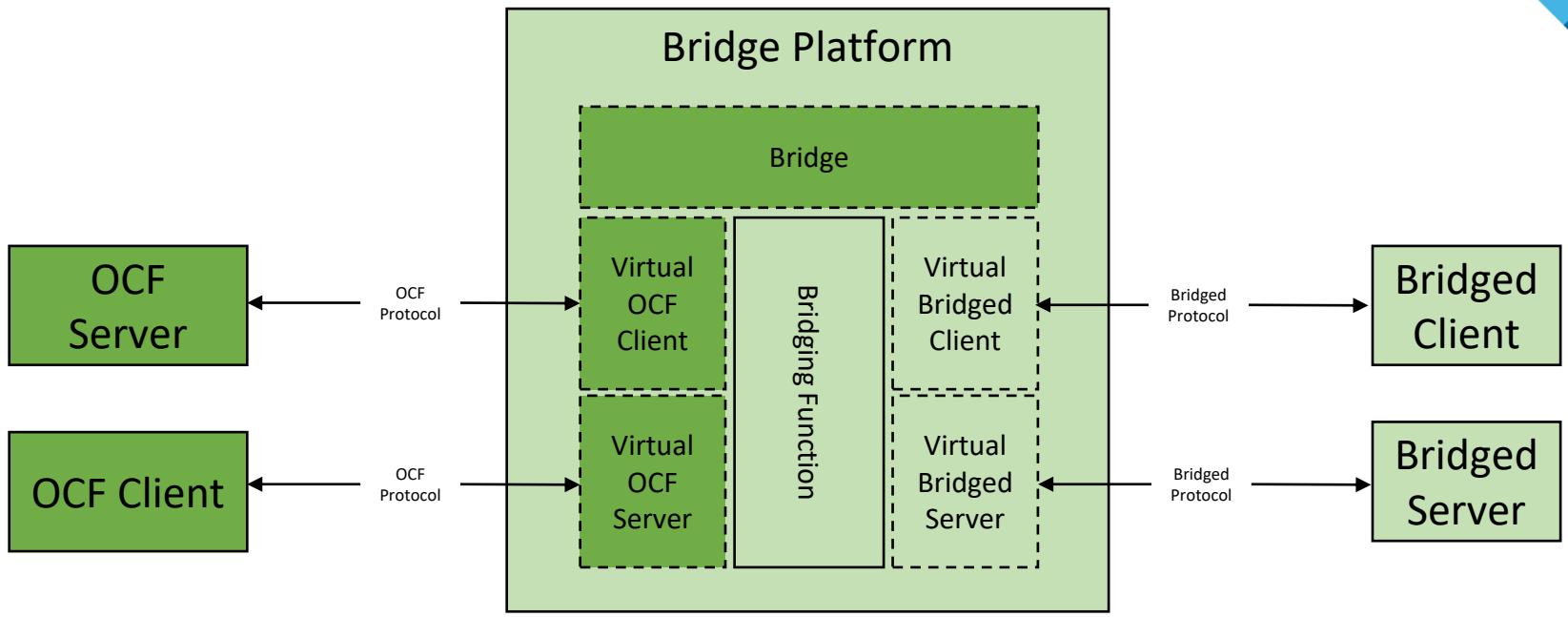
User ID is used as a basis for access control; authentication is performed using token received from cloud during Device registration

Initial Device provisioning for cloud connection is performed by the Mediator service. Mediator is usually hosted by OBT



OCF Bridging

OCF provides opportunity for OCF-vetted ecosystems to integrate into OCF ecosystem by implementing the **Bridging Function**



Bridge and Virtual OCF Device (VOD) are mostly independent OCF Devices, except:

- VOD can not be Owned, if Bridge is Unowned
- When Bridge becomes Unowned, Unowned VODs shall drop DTLS connections
- VOD may use manufacturer credentials hosted by Bridge

VOD is indistinguishable from OCF device, but has additional "oic.d.virtual" Device Type

Best Practices and Attestations

- Some security practices fall outside of our ability to test as part of OCF certification process.
- These are included in the Best Practices section of the Security specification. This section is not intended to be comprehensive, but is intended to provide guidance.
- Certification process requires signing of an OCF Attestation Document, which addresses specific security practices to which the manufacturer asserts compliance.

OPEN CONNECTIVITY FOUNDATION®

OCF Specification Overview
Bridging and Bridges

OCF 2.2.4 Release

October 2021

Revision History

- 2.1.0
  - Split “OCF Bridging Spec” into:
    - OCF Bridging Framework (OCF Bridging specification)
    - Ecosystem-specific Bridging (OCF Resource to XXX Mapping specification)
  - New Ecosystem Bridgings were added
    - BLE, Z-wave, U+
- 2.1.1
  - New Ecosystem Bridging was added
    - EnOcean alliance
- 2.2.1
  - Revise whole contents
- 2.2.4
  - New Ecosystem Bridging was added
    - LwM2M

Contents

- OCF Bridging
  - Bridging Overview
  - Bridging Framework
  - Bridging per Ecosystem
  - Security
  - Testing
- Data Model per Ecosystem Bridge
  - AllJoyn
  - oneM2M
  - ZigBee
  - BLE
  - Z-wave
  - U+
  - EnOcean
  - LwM2M

OPEN CONNECTIVITY FOUNDATION®

OCF Bridging

Bridging Overview

Bridging Framework

Bridging per Ecosystem

Security

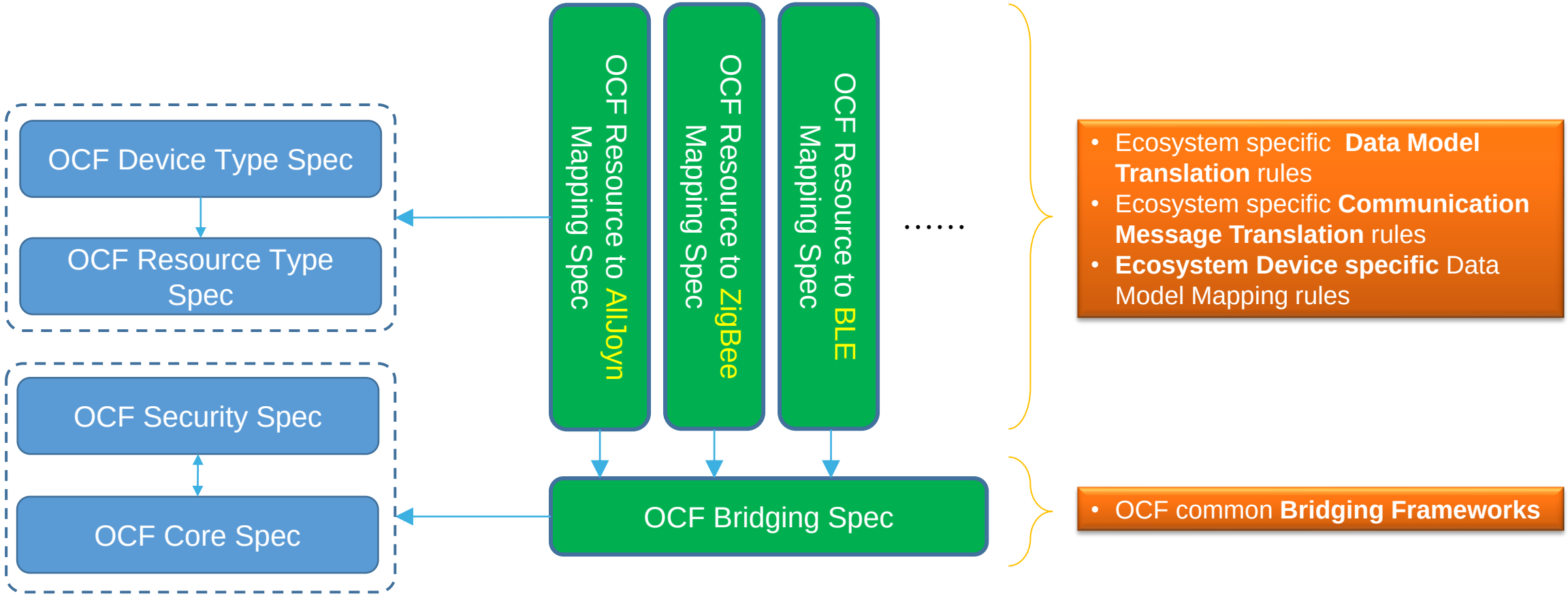
Testing

OCF Bridging : Bridging Overview

- **OCF Bridging Specification**
  - Specifies a **framework** for **Asymmetric** / **Symmetric** translation between devices in OCF and non-OCF ecosystems.
    - In symmetric bridging, a bridge device exposes OCF Server(s) to another ecosystem and exposes other ecosystem's server(s) to OCF. In asymmetric bridging, a bridge device exposes OCF Server(s) to another ecosystem or exposes another ecosystem's server(s) to OCF, but not both.
  - Provides **general requirements** for translation between OCF and non-OCF ecosystems
    - Requirements for resource discovery, message translation, security, etc
- **OCF Resource to XXX Mapping Specifications**
  - Provides **specific requirements** for translation between OCF and **specific ecosystems**
    - Ecosystem specific Data Model Translation rules to/from OCF
    - Ecosystem specific Communication Message Translation rules to/from OCF
    - Ecosystem Device specific Data Model Mapping rules to/from OCF
  - The Mapping Specifications for following ecosystems currently exist:
    - AllJoyn
    - oneM2M
    - ZigBee
    - BLE
    - Z-wave
    - (Haier) U+
    - EnOcean alliance

OCF Bridging : Bridging Overview

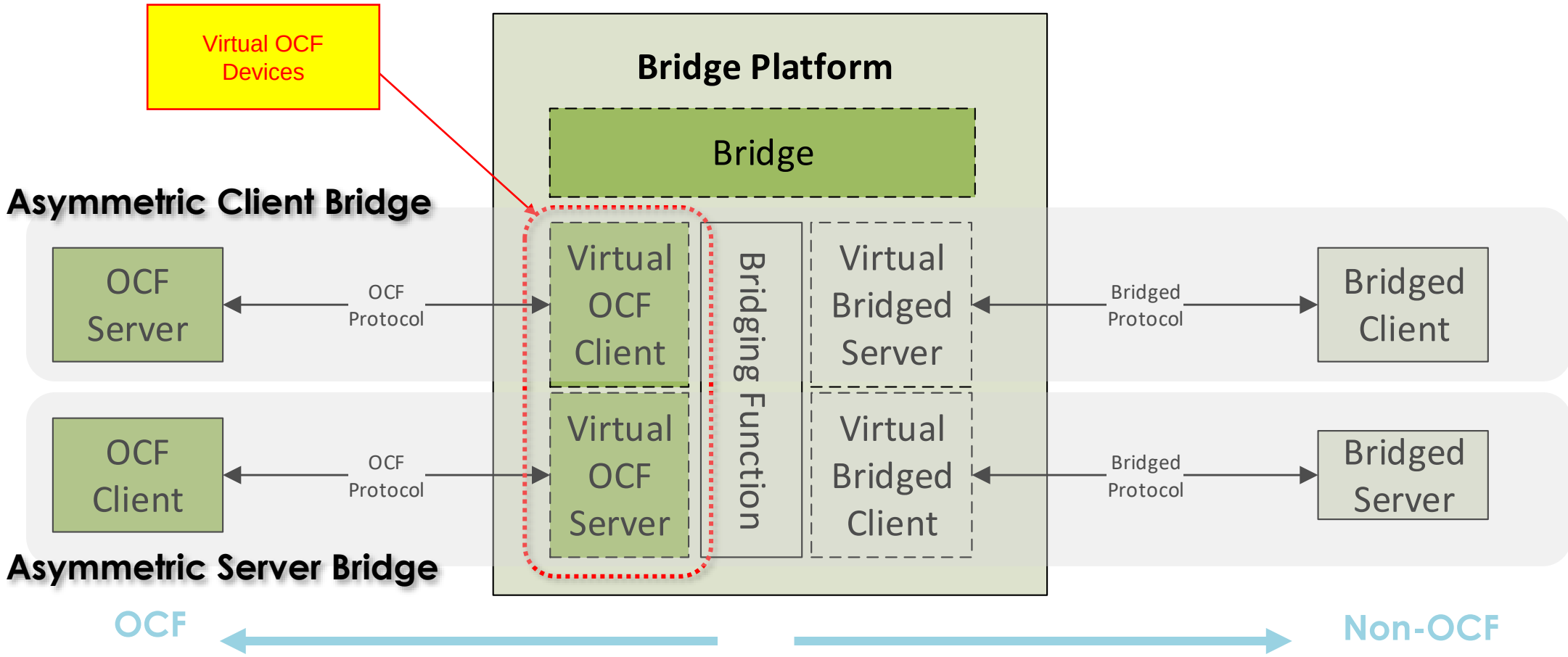
- Relationship between OCF Bridging Specs and Others



OCF Bridging : Bridging Framework

- **Bridge**
  - OCF Device that has a Device Type of "**oic.d.bridge**", provides information on the set of **Virtual OCF Devices** that are resident on the same Bridge Platform.
- **Virtual OCF Device** = Virtual OCF Client + Virtual OCF Server
  - Virtual OCF Client
    - logical representation of a Bridged Client, which an OCF Bridge Device exposes to OCF Servers
  - Virtual OCF Server
    - logical representation of a Bridged Server, which an OCF Bridge Device exposes to OCF Clients
- **Bridge Platform**
  - Entity on which the **Bridge** and **Virtual OCF Devices** are resident
- **Symmetric**, **Asymmetric** Bridging
  - In symmetric bridging, a bridge device exposes OCF Server(s) to another ecosystem and exposes other ecosystem's server(s) to OCF. In asymmetric bridging, a bridge device exposes OCF Server(s) to another ecosystem or exposes another ecosystem's server(s) to OCF, but not both.
- Asymmetric Bridging: **Asymmetric Client Bridging** vs. **Asymmetric Server Bridging**
  - Asymmetric Client Bridging: non-OCF Client is exposed to "Virtual OCF Client"
  - Asymmetric Server Bridging: non-OCF Server is exposed to "Virtual OCF Server" ➔ **Most of OCF Bridging !**

OCF Bridging : Bridging Framework



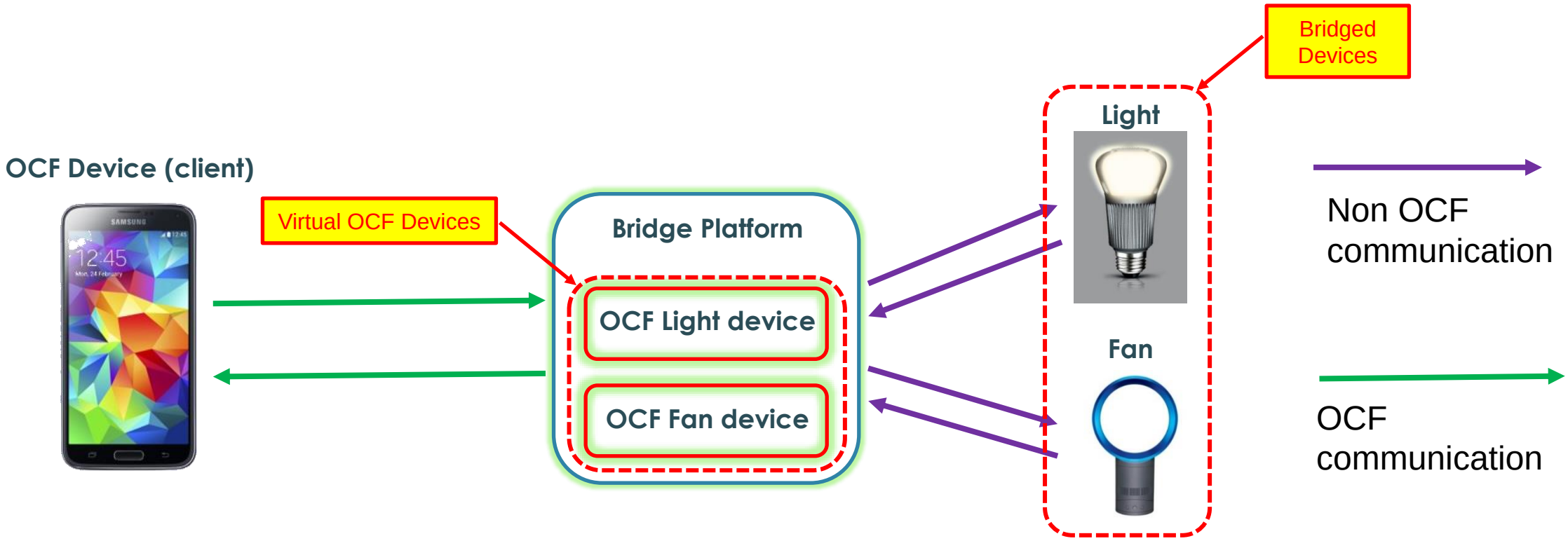
OCF Bridging : Bridging Framework

- General Requirements
  - Bridge shall have "**oic.d.bridge**" as its Device Type
  - VOD shall have "**oic.d.virtual**" as its Device Type
  - Except for these requirements, Bridge and VOD are same as normal OCF Device
- Requirement for "Bridge"
  - VOD List : For maintenance purpose, an OCF Bridge exposes an instance of a **VODList** (Virtual OCF Device List) Resource ("**oic.r.vodlist**") which contains information about all Virtual OCF Devices (Clients or Servers) that are exposed by the OCF Bridge


```
"oic.r.vodlist":
{
  "vods": [
    {
      "n": "My Zigbee Thermostat",
      "di": "54919CA5-4101-4AE4-595B-353C51AA5678",
      "econame": "Zigbee"
    }
  ]
}
```

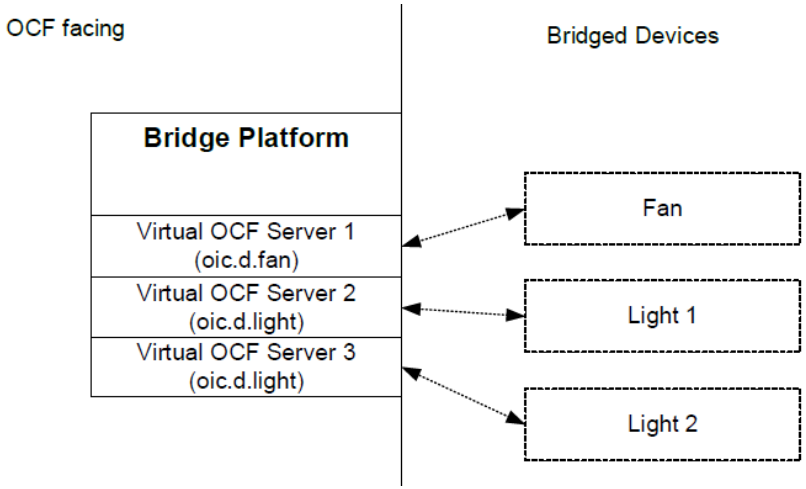
OCF Bridging : Bridging Framework

- Bridging example (Asymmetric Server Bridge)
  - Light and Fan are non-OCF Devices
  - Light and Fan are exposed as "**Virtual OCF Devices**" to OCF Devices by the Bridge



OCF Bridging : Bridging Framework

- Resource Discovery



Asymmetric Server Bridging

Response from the Bridge:

```
[
 {
   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
   "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]:55555"},
           {"ep": "coaps://[2001:db8:a::b1d4]:11111"}]
 },
 {
   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
   "href": "/oic/d",
   "rt": ["oic.wk.d", "oic.d.bridge"],
   "if": ["oic.if.r", "oic.if.baseline"],
   "p": {"bm": 3},
   "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:11111"}]
 },
 {
   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
   "href": "
   "rt": ["o
   "if": ["o
   "p": {"bm
   "eps": [{
 },
```

Response from the Fan VOD:

```
[
 {
   "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
   "href": "/oic/res",
   "rt": ["oic.wk.res"],
   "if": ["oic.if.ll", "oic.if.baseline"],
   "p": {"bm": 3},
   "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:22222"}]
 },
 {
   "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
   "href": "/oic/d",
   "rt": ["oic.wk.d", "oic.d.fan", "oic.d.virtual"],
   "if": ["oic.if.r", "oic.if.baseline"],
   "p": {"bm": 3},
   "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:22222"}]
```

Response from the first Light VOD:

```
[
 {
   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
   "href": "/oic/res",
   "rt": ["oic.wk.res"],
   "if": ["oic.if.ll", "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", "oic.d.virtual"],
   "if": ["oic.if.r", "oic.if.baseline"],
   "p": {"bm": 3},
   "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:33333"}]
 },
 {
   "anchor": "ocf://dc70373c-1e8d-4fb3-96
   "href": "/oic/p",
```

Response from the second Light VOD:

```
[
 {
   "anchor": "ocf://8202138e-aa22-452c-b512-9ebad02bef7c",
   "href": "/oic/res",
   "rt": ["oic.wk.res"],
   "if": ["oic.if.ll", "oic.if.baseline"],
```

OCF Bridging : Bridging per Ecosystem

1

Separate spec for "OCF Resource to X Mapping spec"

- **Translation mechanism** b/w two ecosystems
  - Protocol Translation Mechanism (e.g. "GATT feature of BLE" and "CRUDN of OCF")
  - Data Model Translation Mechanism (e.g. "BLE service/characteristic" and "OCF resource/property")
- Mapping rule for **OCF core Resource**
  - How to build OCF core Resources based on other eco's info? (e.g. /oic/d based on "BLE Device Information Service")
- Translation rule for Resources per each **Device type**
  - Derived models for data model mapping b/w devices in different ecosystems

2

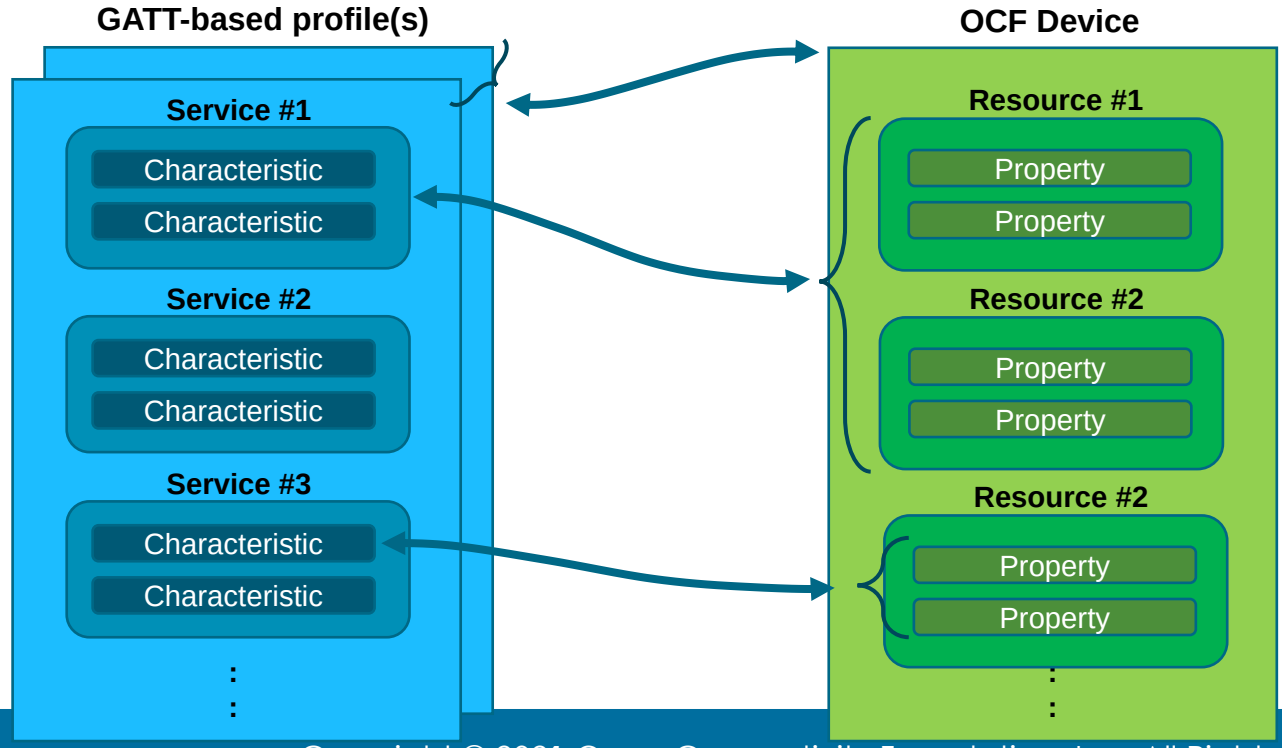
Attestation document

- Test case
  - Test cases for normative text in CR ("shall" statement in CR)

OCF Bridging : Bridging per Ecosystem

- Data Model Translation Mechanism b/w BLE and OCF – BLE example

BLE service-characteristic	OCF resource-property
GATT-based profile(s)	OCF device
Service	OCF resource(s)
Characteristic	OCF resource property
Characteristic configuration descriptor	OCF notification on/off option



OCF Bridging : Bridging per Ecosystem

- Protocol Translation Mechanism b/w BLE and OCF – BLE example
  - OCF CRUDN can be translated into GATT features (and vice versa)

BLE GATT feature		OCF CRUDN
N/A (can't create new service/characteristic)	⟺	CREATE
Characteristic Value Read	⟺	RETRIEVE
Characteristic Value Write	⟺	UPDATE
N/A (can't delete existing service/characteristic)	⟺	DELETE
Characteristic Descriptor Value Write **Characteristic Value Notification**	⟺	NOTIFY

OCF Bridging : Bridging per Ecosystem

- Mapping rule for OCF core Resource

Table 5 – "oic.wk.d" Resource Type definition

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory ?	From BLE Device Service Characteristic value	BLE Description	BLE Mandatory ?
(Device) Name	"n"	Human friendly name For example, "Bob's	Y	Device Name		Y
Spec Version	"icv"	Spec version of ISO/IEC 30118-1:20 this Device is implemented to, The syntax is "core.major.minor"]				
Device ID	"di"	Unique identifier for Device. This value s be as defined in ISO/IEC 30118-2:20 for DeviceID.				
Protocol-Independent	"piid"	Unique identifier for OCF Device (UUID)				

Table 6 – "oic.wk.p" Resource Type definition

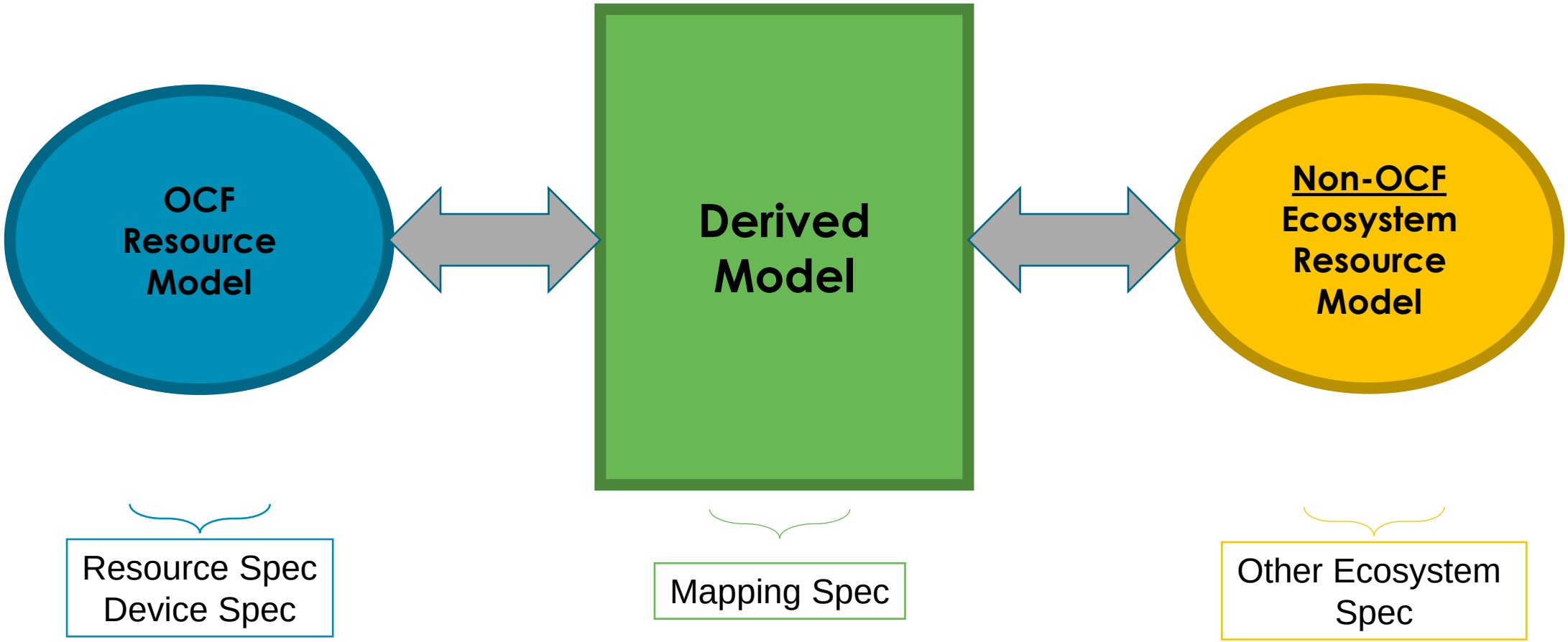
To OCF Property title	OCF Property name	OCF Description	OCF Mandatory?	From BLE Device Service Characteristic value	BLE Description	BLE Mandatory?
Platform ID	"pi"	Unique identifier for the physical platform (UIUID); this shall be a UUID in accordance with IETF RFC 4122. It is recommended that the UUID be created using the random generation scheme (version 4 UUID) specific in the RFC.	Y	(none)	(none)	
Manufacturer Name	"mnmn"	Name of manufacturer (not to exceed 16 characters)	Y	Manufacturer Name String (Device Information). if Device Information	This characteristic represents the name of	N

Table 7 – "oic.wk.con.p" Resource Type definition

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory?	From BLE Device Service Characteristic value	BLE Description	BLE Mandatory?
Platform Names	"mnpn"	Platform Identifier	N	Manufacturer Name String (Device Information)	This characteristic represents the name of the manufacturer of the Device.	

OCF Bridging : Bridging per Ecosystem

- Translation **per each Device Type** – Derived Model



OCF Bridging : Bridging per Ecosystem

- Translation **per each Device Type** – Derived Model

```
{
  "id": "http://openinterconnect.org/zwavemapping/schemas/zwave.operation.binaryswitchcommandclass.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Binary Switch Command Classs",
  "definitions": {
    "zwave.operation.binaryswitchcommandclass": {
      "type": "object",
      "properties": {
        "Value": {
          "type" :  "boolean",
          "description": "On/Off state at the receiving node",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.switch.binary",
            "x-to-ocf": [
              "if Value = 255, ocf.r.switch.binary.value = true.",
              "if Value != 255, ocf.r.switch.binary.value = false."
            ],
            "x-from-ocf": [
              "if ocf.r.switch.binary.value = false, Value = 0",
              "if ocf.r.switch.binary.value = true, Value = 255"

                          ]
          }
        }
      }
    }
  },
  "type": "object",
   "allOf": [
    {"$ref": "#/definitions/zwave.operation.binaryswitchcommandclass"}
  ],
  "required": ["Value"]
}
```

Z-wave binary switch ←→ OCF Binary Switch

OCF Bridging : Bridging per Ecosystem

- Derived models use standard JSON schema syntax. Fundamentally, derived models provide a conversion mapping between OCF data models and the data models in the other ecosystem.
  - The example below is for AllJoyn

```
"asa.environment.targethumidity": {
    "type": "object",
    "properties": {
      "targetvalue": {
        "type": "number",
        "description": "Measured value",
        "x-ocf-conversion": {
          "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
          "x-to-ocf": [
            "if minvalue != maxvalue, ocf.desiredhumidity = targetvalue;ocf.targetlevel = selectablehumiditylevels[0].",
            "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
          ],
          "x-from-ocf": [
            "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
            "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
          ]
        }
      }
}
```

AllJoyn Interface Name

AllJoyn Property Name

OCF Equivalent Resource Type

To OCF from AllJoyn

From OCF to AllJoyn

OCF Bridging : Bridging per Ecosystem

Ecosystem Name	BLE	ZigBee	Z-wave	U+	EnOcean	oneM2M	LwM2M (under development)
Symmetric? or Asymmetric?	Asymmetric (Asymmetric Server Bridging)	Asymmetric (Asymmetric Server Bridging)	Asymmetric (Asymmetric Server Bridging)	Asymmetric (Asymmetric Server Bridging)	Asymmetric (Asymmetric Server Bridging)	Asymmetric (Asymmetric Client Bridging)	Asymmetric (Asymmetric Client Bridging)

OCF Bridging : Security

- The Bridge needs to be a trusted entity as it translates message payloads.
- The Bridge itself and all Virtual Devices that it exposes must be onboarded (transfer of ownership) and provisioned for secure operation.
- Each Virtual Device exposed by the Bridge must implement the security requirements of the ecosystem that it is connected to.
- Bridging specifies mechanisms to selectively block communications between the Bridge and OCF Devices and between the Bridge and Bridged Devices. This fine-grained control enables an administrator to control communications across ecosystems that may not have similar security capabilities.

OCF Bridging : Testing

- How to test ? → Vendor Attestation

- **Considerations for Zigbee Bridging**

  If the IUT implements a Bridge for Zigbee bridging as defined in the OCF Bridging Specification, then by signing below, Applicant attests, represents and warrants that each of the following are true and correct, and will remain true and correct for the IUT:

Optional or Mandatory	Statements	Compliance (Y/N)	Response
Optional	The Platform implements an OCF Bridge Device which bridges to the Zigbee Protocol		

- ☐ Applicant confirms that the Bridge implements Zigbee 3.0 Stack and above for communications with the Zigbee 3.0 Server.
- ☐ Applicant confirms that the Bridge act as a Zigbee Coordinator.
- ☐ Applicant confirms that the Bridge implements OCF to Zigbee Data Model Mapping Specification.

est cases for "Bridge"

cases

Vendor Attestation example (Zigbee)

OPEN CONNECTIVITY FOUNDATION®

Data Model per Ecosystem Bridges:

AllJoyn, oneM2M, ZigBee, BLE, Z-wave, U+, EnOcean, LwM2M

AllJoyn Device Type Equivalency

Classification	ASA Device Type	OCF Device Type	OCF Device Type ID
Air Care	Air Conditioner	Air Conditioner	oic.d.airconditioner
	AirPurifier	Air Purifier	oic.d.airpurifier
	AirQualityMonitor	Air Quality Monitor	oic.d.aqm
	Dehumidifier	Dehumidifier	oic.d.dehumidifier
	Humidifier	Humidifier	oic.d.humidifier
	ElectricFan	Fan	oic.d.fan
	Thermostat	Thermostat	oic.d.thermostat
Fabric Care	Clothes Washer	Washer	oic.d.washer
	Clothers Dryer	Dryer	oic.d.dryer
	Clothers Washer-Dryer	Washer-Dryer	oic.d.washerdryer
Food Preservation	Refrigerator	Refrigerator	oic.d.refrigerator
	Ice Maker	Ice Maker (Resource)	oic.r.icemaker
	Freezer	Freezer	oic.d.freezer
Food Preparation	Oven	Oven	oic.d.oven
	Cooktop	Cooktop	oic.d.cooktop
	Cookerhood	Cooker Hood	oic.d.cookerhood
	Foodprobe	Food Probe	oic.d.foodprobe
Dish Care	Dishwasher	Dishwasher	oic.d.dishwasher
Floor Care	Robot Cleaner	Robot Cleaner	oic.d.robotcleaner
Entertainment	TV	Television	oic.d.tv
	Set Top box (STB)	Set Top Box	oic.d.stb

- Yellow highlights identify Device Types that were added to support equivalency

AllJoyn Interface to Resource Mapping

AllJoyn Interface	OCF Resource Type Name	OCF Resource Type ID	OCF Interface(s)
Environment.CurrentAirQuality	Air Quality Collection	oic.r.airqualitycollection	oic.if.s
Environment.CurrentAirQualityLevel	Air Quality Collection	oic.r.airqualitycollection	oic.if.s
Environment.CurrentHumidity	Humidity	oic.r.humidity	oic.if.s
Environment.CurrentTemperature	Temperature	oic.r.temperature	oic.if.s
Environment.TargetHumidity	Humidity	oic.r.humidity, oic.r.selectablelevels	oic.if.a
Environment.TargetTemperature	Temperature	oic.r.temperature	oic.if.a
Operation.AudioVolume	Audio Controls	oic.r.audio	oic.if.a
Operation.Channel	Not mapped		
Operation.ClimateControlMode	Mode	oic.r.mode	oic.if.a
	Operational State	oic.r.operational.state	oic.if.s
Operation.ClosedStatus	Door	oic.r.door	oic.if.s
Operation.CycleControl	Operational State	oic.r.operational.state	oic.if.s
Operation.FanSpeedLevel	Air Flow	oic.r.airflow	oic.if.a
Operation.HeatingZone	Heating Zone Collection	oic.r.heatingzonecollection	oic.if.s
Operation.HvacFanMode	Mode	oic.r.mode	oic.if.a
Operation.OnOffStatus	Binary Switch	oic.r.switch.binary	oic.if.s
Operation.OvenCyclePhase	Operational State	oic.r.operationalstate	oic.if.s

OPEN CONNECTIVITY FOUNDATION®

Ecosystem Bridges:

AllJoyn, **oneM2M**, ZigBee, BLE, Z-wave, U+, EnOcean , LwM2M

Asymmetric Client Bridging

OCF to oneM2M Device Type Mapping

oneM2M Device Type	OCF Device Type
device3DPrinter	oic.d.3dprinter
deviceAirConditioner	oic.d.airconditioner
deviceAirPurifier	oic.d.airpurifier
deviceAirQualityMonitor	oic.d.airqualitymonitor
deviceAudioReceiver	oic.d.receiver
deviceBloodPressureMonitor	oic.d.bloodpressuremonitor
deviceCamera	oic.d.camera
deviceClothesDryer	oic.d.dryer
deviceClothesWasher	oic.d.washer
deviceCoffeeMachine	oic.d.coffeemachine
deviceCookerHood	oic.d.cookerhood
deviceCooktop	oic.d.cooktop
deviceDehumidifier	oic.d.dehumidifier
deviceDishWasher	oic.d.dishwasher
deviceDoor	oic.d.door
deviceDoorLock	oic.d.smartlock
deviceElectricVehicleCharger	oic.d.electricvehiclecharger
deviceFan	oic.d.fan
deviceFoodProbe	oic.d.foodprobe
deviceFreezer	oic.d.freezer
deviceGlucosemeter	oic.d.glucosemeter
deviceHumidifier	oic.d.humidifier
deviceKettle	oic.d.kettle
deviceLight	oic.d.light

oneM2M Device Type	OCF Device Type
deviceMicrogeneration	oic.d.energygenerator
deviceMultiFunctionPrinter	oic.d.multifunctionprinter
deviceOutdoorLamp	oic.d.light
deviceOven	oic.d.oven
devicePrinter	oic.d.printer
deviceRefrigerator	oic.d.refrigerator
deviceRobotCleaner	oic.d.robotcleaner
deviceScanner	oic.d.scanner
deviceSecurityPanel	oic.d.securitypanel
deviceSetTopBox	oic.d.stb
deviceSmartElectricMeter	oic.d.electricmeter
deviceSmartPlug	oic.d.smartplug
deviceSteamCloset	oic.d.steamcloset
deviceStorageBattery	oic.d.battery
deviceSwitch	oic.d.switch
deviceTelevision	oic.d.tv
deviceThermostat	oic.d.thermostat
deviceWaterHeater	oic.d.waterheater
deviceWaterValve	oic.d.watervalve
deviceWeightScaleAndBodyCompositionAnalyzer	oic.d.bodyscale
deviceWindowShade	oic.d.blind
deviceThermometer	oic.d.bodythermometer

- Yellow highlights identify Device Types that were added to support equivalency

OCF Resources to oneM2M Module Classes

oneM2M Module Class	OCF Resource Type
3Dprinter	oic.r.3dprinter
acousticsensor	oic.r.soundpressure
airconjobmode	oic.r.operational.state
airflow	oic.r.airflow
airpurifierjobmode	oic.r.operational.state
airqualitysensor	oic.r.airquality oic.r.switch.binary oic.r.humidity
alarmspeaker	oic.r.audiovolume oic.r.switch.binary oic.r.light.dimming
audioVolume	oic.r.audio
autodocumentfeeder	oic.r.operational.state
battery	oic.r.energy.battery
binaryswitch	oic.r.swtich.binary
boiler	oic.r.sensor
brewing	oic.r.brewing
brightness	oic.r.light.brightness
clock	oic.r.clock
clothesdryerjobmode	oic.r.operational.state
colour	oic.r.colour
coloursaturation	oic.r.colour.saturation
credentials	oic.r.userinfo
dehumidiiferjobmode	oic.r.operational.state
doorStatus	oic.r.door
electricvehicleconnector	oic.r.vehicle.connector
energyconsumption	oic.r.energy.electrical oic.r.energy.consumption
energygeneration	oic.r.energy.generation

oneM2M Module Class	OCF Resource Type
filterinfo	oic.r.consumable oic.r.sensor
foaming	oic.r.foaming
grinder	oic.r.grinder oic.r.switch.binary
heatingzone	oic.r.heatingzone
height	oic.r.height
hotwatersupply	oic.r.switch.binary oic.r.sensor
impactsensor	oic.r.impactsensor
keepwarm	oic.r.time.period
Keypad	oic.r.keypadchar
liquidlevel	oic.r.liquid.level
liquidremaining	oic.r.liquid.level
lock	oic.r.lock
motionSensor	oic.r.sensor.motion oic.r.sensor.props
openlevel	oic.r.openlevel
operationmode	oic.r.switch.binary
overcurrentsensor	oic.r.time.period oic.r.sensor
powersave	oic.r.switch.binary
printqueue	oic.r.printer.queue
pushbutton	oic.r.button
refrigeration	oic.r.refrigeration
relativeHumidity	oic.r.humidity
robotcleanerjobmode	oic.r.operational.state
steamclosetjobmode	oic.r.operational.state
temperature	oic.r.temperature
uvsensor	oic.r.sensor.radiation.uv
watersensor	oic.r.sensor.water
weight	oic.r.weight

- Yellow highlights identify Resource Types that were added to support equivalency

Ecosystem Bridges:

AllJoyn, oneM2M, **ZigBee**, BLE, Z-wave, U+, EnOcean , LwM2M

Asymmetric Server Bridging

OCF to Zigbee Device Type Mapping

Zigbee Device Type	Zigbee Device ID	OCF Device Type
On/off Output	0x0002	oic.d.smartplug
Mains Power Outlet	0x0009	oic.d.smartplug
Smart Plug	0x0051	oic.d.smartplug
On/Off Light	0x0100	oic.d.light
Dimmable Light	0x0101	oic.d.light
Color Dimmable Light	0x0102	oic.d.light
Color Temperature Light	0x010c	oic.d.light
Extended Color Light	0x010d	oic.d.light
Window Covering Device	0x0202	oic.d.blind
Thermostat	0x0301	oic.d.thermostat
Temperature Sensor	0x0302	oic.d.sensor
Occupancy Sensor	0x0107	oic.d.sensor
IAS Zone	0x0402	oic.d.sensor

OCF Resources to Zigbee Clusters

Zigbee Cluster	OCF Resource Type Name	OCF Resource Type ID	OCF Interface(s)
On/off	Binary Switch	oic.r.switch.binary	oic.if.a
Level Control	Dimming	oic.r.light.dimming	oic.if.a
Color Control	Colour Hue and Saturation, Colour Space Coordinates, Colour Temperature	oic.r.colour.hs, oic.r.colour.csc, oic.r.colour.colourtemperature,	oic.if.a
Thermostat	Temperature (3)	oic.r.temperature (3) * 1 for sensor, 2 for heater and cooler	oic.if.s oic.if.a
Window Covering	Window Covering	oic.r.windowcovering, oic.r.openlevel (4) * 2 for lift (percentage scale and cm scale), 2 for tilt (percentage scale and cm scale)	oic.if.rw oic.if.a
Temperature Measurement	Temperature	oic.r.temperature	oic.if.s
Occupancy Sensing	Presence Sensor	oic.r.sensor.presence	oic.if.s
IAS Zone	IAS Zone	oic.r.ias.zone	oic.if.rw

- Yellow highlights identify Resource Types that were added to support equivalency

Ecosystem Bridges:

AllJoyn, oneM2M, ZigBee, **BLE**, Z-wave, U+, EnOcean , LwM2M

Asymmetric Server Bridging

OCF to BLE Device Type Mapping

BLE GATT-based Profile	OCF Device Type
Blood Pressure Profile	"oic.d.bloodpressuremonitor"
Glucose Profile	"oic.d.glucosemeter"
Health Thermometer Profile	"oic.d.bodythermometer"
Weight Scale Profile	"oic.d.bodyscale"

OCF Resources to BLE Services

BLE GATT-based Profile	BLE Service	OCF Resource		OCF Device Type
		Atomic Measurement Resource Type	Resource Type	
Blood Pressure Profile	Blood Pressure Service	"oic.r. bloodpressuremonitor-am"	"oic.r.blood.pressure"	"oic.d.bloodpressuremonitor"
			"oic.r.pulserate"	
	Device Information Service		"oic.wk.d"	
			"oic.wk.p"	
Glucose Profile	Glucose Service	"oic.r.glucosemeter-am"	"oic.r.glucose"	"oic.d.glucosemeter"
			"oic.r.glucose.carb"	
			"oic.r.glucose.exercise"	
			"oic.r.glucose.hba1c"	
			"oic.r.glucose.health"	
			"oic.r.glucose.meal"	
			"oic.r.glucose.medication"	
			"oic.r.glucose.samplelocation"	
			"oic.r.glucose.tester"	
	Device Information Service		"oic.wk.d"	
			"oic.wk.p"	
Health Thermometer Profile	Health Thermometer Service	"oic.r.bodythermometer-am"	"oic.r.temperature"	"oic.d.bodythermometer"
			"oic.r.body.location.temperature"	
	Device Information Service		"oic.wk.d"	
			"oic.wk.p"	
Weight Scale Profile	Weight Scale Service	"oic.r.bodyscale-am"	"oic.r.weight"	"oic.d.bodyscale"
			"oic.r.bmi"	
			"oic.r.height"	
			"oic.r.body.fat"	
			"oic.r.body.water"	
			"oic.r.body.slm"	
			"oic.r.body.ffm"	
	Device Information Service		"oic.wk.d"	
			"oic.wk.p"	

Ecosystem Bridges:

AllJoyn, oneM2M, ZigBee, BLE, **Z-wave**, U+, EnOcean , LwM2M

Asymmetric Server Bridging

OCF to Z-Wave Device Mapping

Z- Wave Plus Device	OCF Device Type	OCF Device Name
Light Dimmer Switch	oic.d.light	Light
Door Lock – Keypad	oic.d.smartlock	Smart Lock
On/Off Power Switch	oic.d.switch	Switch
Sensor - Multilevel	oic.d.sensor	Generic Sensor
Sensor – Notification	oic.d.sensor	Generic Sensor

OCF Resources to Z-Wave Command Class

Z- Wave Plus Device	Z-Wave Command Class	OCF Resource Type	OCF Device Type	OCF Device Name
Light Dimmer Switch	Multilevel Switch Command Class	oic.r.switch.binary	oic.d.light	Light
	Multilevel Switch Command Class	oic.r.light.dimming		
	Manufacturer Specific Command Class	oic.wk.d		
	Version Command Class Z-Wave Plus Info Command Class	oic.wk.p		
Door Lock – Keypad	Door Lock Command Class	oic.r.lock.status	oic.d.smartlock	Smart Lock
	User Code Command Class	oic.r.lock.code		
	Battery Command Class	oic.r.energy.battery.		
	Manufacturer Specific Command Class	oic.wk.d		
	Version Command Class Z-Wave Plus Info Command Class	oic.wk.p		
On/Off Power Switch	Binary Switch Command Class	oic.r.switch.binary	oic.d.switch	Switch
	Battery Command Class	oic.r.energy.battery.		
	Manufacturer Specific Command Class	oic.wk.d		
	Version Command Class Z-Wave Plus Info Command Class	oic.wk.p		
Sensor - Multilevel	Multilevel Sensor Command Class	oic.r.sensor.carbondioxide	oic.d.sensor	Generic Sensor
	Multilevel Sensor Command Class	oic.r.sensor.carbonmonoxide		
	Multilevel Sensor Command Class	oic.r.sensor.water		
	Multilevel Sensor Command Class	oic.r.sensor.smoke		
	Battery Command Class	oic.r.energy.battery.		
	Manufacturer Specific Command Class	oic.wk.d		
	Version Command Class Z-Wave Plus Info Command Class	oic.wk.p		
Sensor - Notification	Notification Command Class	oic.r.sensor.carbondioxide	oic.d.sensor	Generic Sensor
	Notification Command Class	oic.r.sensor.carbonmonoxide		
	Notification Command Class	oic.r.sensor.water		
	Notification Command Class	oic.r.sensor.smoke		
	Battery Command Class	oic.r.energy.battery.		
	Manufacturer Specific Command Class	oic.wk.d		
	Version Command Class Z-Wave Plus Info Command Class	oic.wk.p		

OPEN CONNECTIVITY FOUNDATION®

Ecosystem Bridges:

AllJoyn, oneM2M, ZigBee, BLE, Z-wave, U+, EnOcean , LwM2M

Asymmetric Server Bridging

OCF to U+ Device Mapping

U+ Device	OCF Device Name	OCF Device Type ("rt")
Air Conditioner	Air Conditioner	"oic.d.airconditioner"
Water Heater	Water Heater	"oic.d.waterheater"
Air Purifier	Air Purifier	"oic.d.airpurifier"

OCF Resources to U+ Properties

U+ Device	U+ Property	OCF Resource Type	OCF Device Name	OCF Device Type ("rt")
Air Conditioner	"onOffStatus"	"oic.r.switch.binary"	Air Conditioner	"oic.d.airconditioner"
	"targetTemperature"	"oic.r.temperature"		
	"windSpeed"	"oic.r.selectablelevels"		
	"operationMode"	"oic.r.mode"		
Water Heater	"onOffStatus"	oic.r.switch.binary	Water Heater	"oic.d.waterheater"
	"targetTemperature"	"oic.r.temperature"		
Air Purifier	"onOffStatus"	"oic.r.switch.binary"	Air Purifier	"oic.d.airpurifier"
	"mode"	"oic.r.operational.state"		
	"windSpeed"	"oic.r.selectablelevels"		

Ecosystem Bridges:

AllJoyn, oneM2M, ZigBee, BLE, Z-wave, U+, **EnOcean**, LwM2M

Asymmetric Server Bridging

OCF to EnOcean Device Mapping

EnOcean Device Name (EEP)	OCF Device Type	OCF Device Name
Push Button (F6-01-01)	oic.d.sensor	Generic Sensor
Rocker Switch, 2 Rocker (F6-02-XX)	oic.d.sensor	Generic Sensor
Rocker Switch, 4 Rocker (F6-03-XX)	oic.d.sensor	Generic Sensor
Position Switch (F6-04-01)	oic.d.sensor	Generic Sensor
Position Switch (F6-04-02)	oic.d.sensor	Generic Sensor
Liquid Leakage Detector (Water) (F6-05-01)	oic.d.sensor	Generic Sensor
Smoke Detector (F6-05-02)	oic.d.sensor	Generic Sensor
Single Input Contact (D5-00-01)	oic.d.sensor	Generic Sensor
Temperature Sensor (A5-02-XX)	oic.d.sensor	Generic Sensor
Temperature and Humidity Sensor (A5-04-XX)	oic.d.sensor	Generic Sensor
Barometric Sensor (A5-05-01)	oic.d.sensor	Generic Sensor
Light Sensor (A5-06-XX)	oic.d.sensor	Generic Sensor
Occupancy Sensor (A5-07-XX)	oic.d.sensor	Generic Sensor
Light, Temperature and Occupancy Sensor (A5-08-XX)	oic.d.sensor	Generic Sensor

OCF Resources to EnOcean Telegram Parameter

EnOcean Device Name (EEP)	EnOcean Telegram Parameters	OCF Resource Type(s)	OCF Device Type	OCF Device Name
Push Button (F6-01-01)	Push Button Released Push Button Pressed	oic.r.button	oic.d.sensor	Generic Sensor
Rocker Switch, 2 Rocker (F6-02-XX)	Rocker 1st Action AI Rocker 1st Action AO Rocker 1st Action BI Rocker 1st Action BO	oic.r.button oic.r.button	oic.d.sensor	Generic Sensor
Rocker Switch, 4 Rocker (F6-03-XX)	Rocker 1st Action AI Rocker 1st Action AO Rocker 1st Action BI Rocker 1st Action BO Rocker 1st Action CI Rocker 1st Action CO Rocker 1st Action DI Rocker 1st Action DO	oic.r.button oic.r.button oic.r.button oic.r.button	oic.d.sensor	Generic Sensor
Position Switch (F6-04-01)	Key Card activated Key Card taken out	oic.r.keycardswitch	oic.d.sensor	Generic Sensor
Position Switch (F6-04-02)	Key Card inserted Key Card taken out	oic.r.keycardswitch	oic.d.sensor	Generic Sensor
Liquid Leakage Detector (Water) (F6-05-01)	Alert Signal	oic.r.sensor.water	oic.d.sensor	Generic Sensor
Smoke Detector (F6-05-02)	Smoke Alarm ON Smoke Alarm OFF	oic.r.sensor.smoke	oic.d.sensor	Generic Sensor
Single Input Contact (D5-00-01)	Open Closed	oic.r.sensor.contact	oic.d.sensor	Generic Sensor
Temperature Sensor (A5-02-XX)	Temperature value Unit (defined by spec) Range (by type spec)	oic.r.temperature	oic.d.sensor	Generic Sensor
Temperature and Humidity Sensor (A5-04-XX)	Temperature value Temperature unit (by spec) Temperature range (by type spec) Humidity (%)	oic.r.temperature oic.r.humidity	oic.d.sensor	Generic Sensor
Barometric Sensor (A5-05-01)	Barometer value	oic.r.sensor. atmosphericpressure	oic.d.sensor	Generic Sensor
Light Sensor (A5-06-XX)	Illumination value (linear, lx) range (by type Spec)	oic.r.sensor.illuminance	oic.d.sensor	Generic Sensor
Occupancy Sensor (A5-07-XX)	PIR Status Uncertain PIR Status Motion detected	oic.r.sensor.presence	oic.d.sensor	Generic Sensor
Light, Temperature and Occupancy Sensor (A5-08-XX)	Temperature value Temp Unit (by spec Temp Range (by TYPE spec) Illumination value Illumination range (by type spec) Occupancy	oic.r.temperature oic.r.sensor.illuminance oic.r.sensor.presence	oic.d.sensor	Generic Sensor

OPEN CONNECTIVITY FOUNDATION®

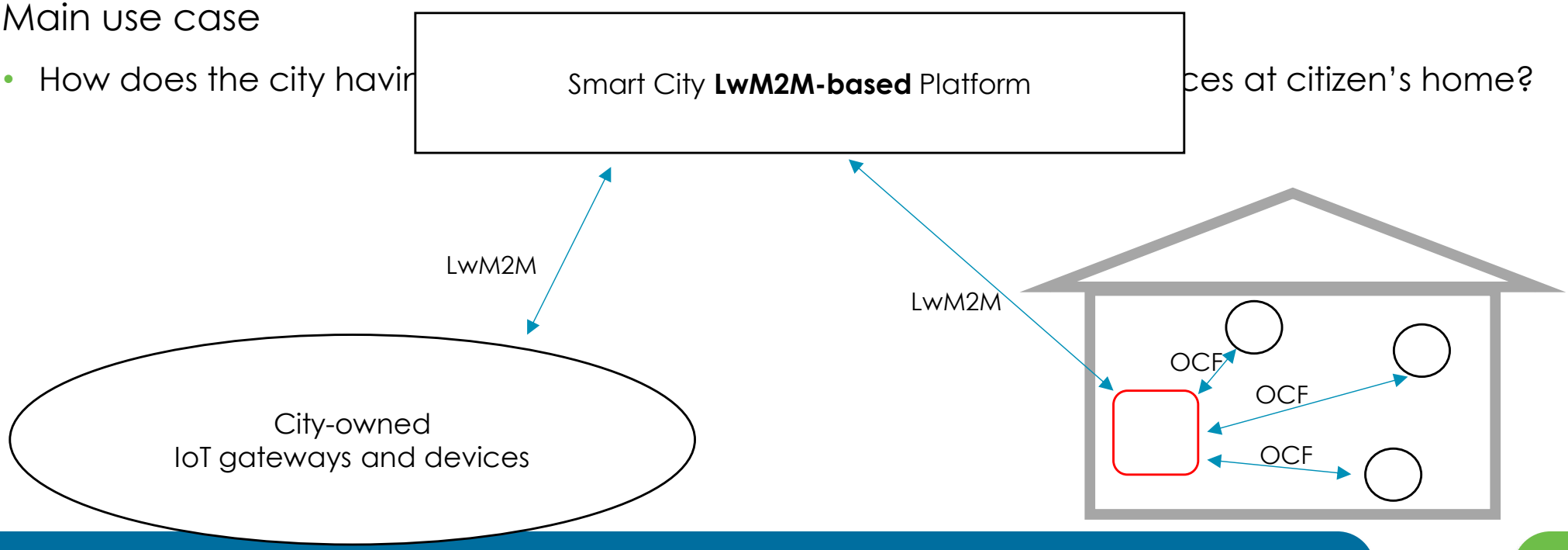
Ecosystem Bridges:

AllJoyn, oneM2M, ZigBee, BLE, Z-wave, U+, EnOcean, **LwM2M**

Asymmetric Client Bridging

LwM2M Bridging

- LwM2M Bridging
  - **Asymmetric Client** Bridging
    - LwM2M Client accesses OCF Servers
- Main use case
  - How does the city havir ces at citizen's home?



Supported OCF Device Types for OCF Resources to LWM2M Object Mapping

LWM2M Object name	OCF Resource Type	OCF Device Type
On/Off switch	oic.r.switch.binary	oic.d.airconditioner
Temperature	oic.r.temperature	
On/Off switch	oic.r.switch.binary	oic.d.airpurifier
On/Off switch	oic.r.switch.binary	oic.d.washerdryer
On/Off switch	oic.r.switch.binary	oic.d.dehumidifier
Power	oic.r.energy.consumption	oic.d.electricmeter
Power	oic.r.energy.consumption	oic.d.energymonitor
On/Off switch	oic.r.switch.binary	oic.d.fan
Temperature	oic.r.temperature	oic.d.refrigerator
On/Off switch	oic.r.switch.binary	oic.d.humidifier
On/Off switch	oic.r.switch.binary	oic.d.light
On/Off switch	oic.r.switch.binary	oic.d.oven
Temperature	oic.r.temperature	
On/Off switch	oic.r.switch.binary	oic.d.stb
Lock	oic.r.lock.status	oic.d.smartlock
On/Off switch	oic.r.switch.binary	oic.d.smartplug
Temperature	oic.r.temperature	oic.d.thermostat
On/Off switch	oic.r.switch.binary	oic.d.waterheater
Temperature	oic.r.temperature	

LWM2M Object to OCF Resource Type Mapping

LWM2M Object Name	LWM2M Object ID	OCF Resource Type
Actuation	3306	oic.r.audio
Buzzer	3338	oic.r.door
Device	3	oic.wk.d
		oic.wk.p
		oic.r.energy.battery
Digital Input	3300	oic.r.vehicleconnector
Door	10351	oic.r.door
Energy	3331	oic.r.energy.consumption
Humidity	3304	oic.r.humidity
Load Control	3310	oic.r.time.period
Lock	10359	oic.r.lock.status
On/Off switch	3342	oic.r.switch.binary
Positioner	3337	oic.r.openlevel
Power	3328	oic.r.energy.consumption
Temperature	3303	oic.r.temperature

OPEN CONNECTIVITY FOUNDATION®

References

Specification Location

- Where can I find the specifications and Resource Type definitions?
  - OCF Specifications: https://openconnectivity.org/developer/specifications
- Resource Type Definitions
  - Core Resources: https://github.com/openconnectivityfoundation/core
  - Core Extension Resources: https://github.com/openconnectivityfoundation/core-extensions
  - Bridging Resources: https://github.com/openconnectivityfoundation/bridging
    - EnOcean Derived Models: https://github.com/openconnectivityfoundation/OCF-EnOcean
    - Z-Wave Derived Models: https://github.com/openconnectivityfoundation/OCF-Z-Wave
    - Zigbee Derived Models: https://github.com/openconnectivityfoundation/OCF-Zigbee
    - U+ Derived Models: https://github.com/openconnectivityfoundation/OCF-UPlus
    - oneM2M Derived Models: https://github.com/openconnectivityfoundation/OCF-oneM2M
    - BLE Derived Models: https://github.com/openconnectivityfoundation/OCF-BLE
    - AllJoyn Derived Models: https://github.com/openconnectivityfoundation/OCF-AllJoyn
    - LwM2M Derived Models: https://github.com/openconnectivityfoundation/OCF-LWM2M
  - Cloud Resources: https://github.com/openconnectivityfoundation/cloud-services
  - Security Resources: https://github.com/openconnectivityfoundation/security
  - Device Types: https://github.com/openconnectivityfoundation/devicemodels
    - Resource Types: https://github.com/openconnectivityfoundation/IoTDataModels

OPEN CONNECTIVITY FOUNDATION®

OCF Specification Overview
Device and Resource Modeling

OCF 2.2.4 Release

October 2021

OPEN CONNECTIVITY FOUNDATION®

Resource Model: Resource Type Specification

Overview

Resource Specification

- List of reusable resources that are used in an OCF Device
  - More than 100 Resource Types are defined as of OCF 2.2.4; enabling Smart Home, Healthcare, Smart Factory, and Photovoltaic System applications.
  - All Resource Types build on the Core definitions
- Each resource definition contains:
  - A unique identifier (rt)
  - Identification of the default interface and other supported interfaces
  - List of supported methods

Resources are specified in OpenAPI2.0 (formerly known as 'Swagger 2.0')

See https://oneiota.org for the complete set of OCF defined Resource Types

Resource Type Map

- All OCF Resource Types are available in: https://oneiota.org
- A list of all currently accepted Resource Types with links to the OpenAPI definitions that are found in oneIoTa may be found here: https://openconnectivityfoundation.github.io/devicemodels/docs/resource.html
- All OCF Resource Type IDs are IANA registered: http://www.iana.org/assignments/core-parameters/core-parameters.xhtml

If an OCF Device hosts an OCF defined resource then it shall follow all normative requirements in the Resource Specification applicable to that Resource.

Newly Defined Resource Types – OCF 2.2.4

Resource Type	Use Case Category	Use Case Detail
Remote Control	Media Control	Models a generic keypad based remote control, which can be used by a Client to interact with any device that supports such inputs.
TV Apps	Media Control	Properties associated with the launch and control of apps that may be resident on a TV (e.g. Netflix, Hulu etc).
Vendor List	Smart Home	Properties associated with a list of possible vendors that may be associated with a device or application.
DALI	Smart Lighting	Properties associated with a DALI based light
DALI Config	Smart Lighting	Properties associated with the configuration of a DALI based light

OPEN CONNECTIVITY FOUNDATION®

Device Specification

Overview

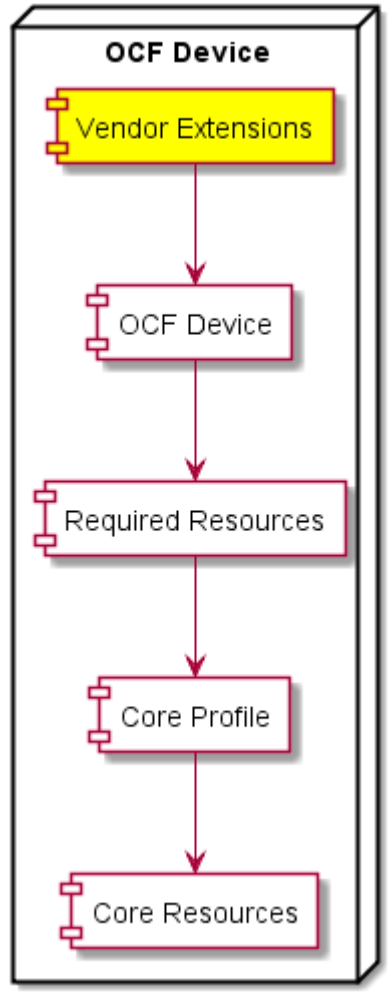
Higher Layer Specifications

- Specifications are split into 2 documents:
  - Device specification (per vertical Annexes if needed)
  - Resource specification (vertical agnostic)

The Device specification uses the Resources defined in the Resource Specification

Device Specification

- Contains profiles of
  - Core specification
  - Security specification
- Contains list of OCF devices
- Each OCF device definition contains:
  - Human friendly name
  - Unique identifier (rt) in the form oic.d.<thing>
    - This is exposed in an OCF Device as part of



Exposure of an OCF Device Type is Mandatory. If an OCF Server hosts an OCF known device then it shall follow all normative requirements in the Device Specification applicable to that Device.

Device Categories

- All OCF devices are grouped into Device Categories based on the Universal Device Classification (UDC) that was developed by LBNL.
  - https://eta-intranet.lbl.gov/sites/default/files/lbnl-classification-v1.pdf

Device Category Name	Description
LBNL Categories	
Space Conditioning	Heating and cooling systems
Lighting	
Appliance	Also known as "white goods"; covers major appliances only.
Electronics	Personal electronics
Miscellaneous	Small appliances, other
Infrastructure	Physical building and infrastructure
Transportation	Vehicles, fixed devices that provide movement (e.g. Escalators)
Other	
OCF Added Categories	
Fitness	Includes lifestyle
Medical	
Personal Health	

Mandatory Resources per Device Type

- A vertical may specify a set of Resources that are mandatory to expose by a specific Device Type.
  - Note: a Device is free to expose any number of optional Resources that it requires
  - Currently defined verticals: Smart Home, Healthcare, Industrial, Photo Voltaic System
- The complete set of Device Types and any associated mandatory resources that exist for a vertical are all available in github:
  - https://github.com/openconnectivityfoundation/devicemodels/blob/master/oic.devicemap-content.json

Vendor Extensions

- A vendor is allowed to:
  - Create their own defined (non-OCF standardized) Resource Types
  - Create their own defined (non-OCF standardized) Device Types
  - Extend existing devices with additional (not mandated) Resource Types
    - With standardized resource types
    - With vendor defined resource types
- All vendor extensions follow an OCF defined naming scheme

Some Example Device Types

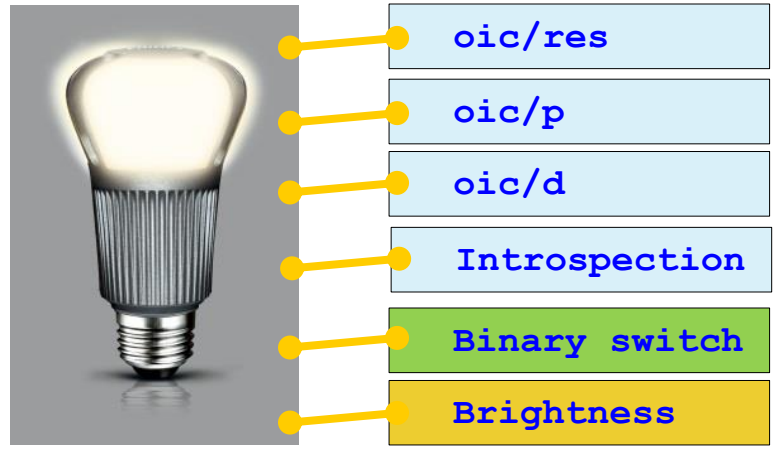
Category	Name	Device Type	Mandatory Resources
Appliance	Refrigerator	oic.d.refrigerator	Temperature (x2)
Electronics	3D Printer	oic.d.3dprinter	Binary Switch, 3D Printer, Temperature, Printer Queue, Operational State
Miscellaneous	Optical Augmented RFID Reader	oic.d.orfid	RFID Tag, RFID Station
Personal Health	Body Scale	oic.d.bodyscale	Body Scale Atomic Measurement

Note: All defined Device Types are of the form “oic.d.<thing>” where <thing> is a single alphanumeric string (lower case [a..z],[0..9] only) no more than 24 characters in length giving a total maximum length of the Device Type of 32 characters

Device example: light device (oic.d.light)

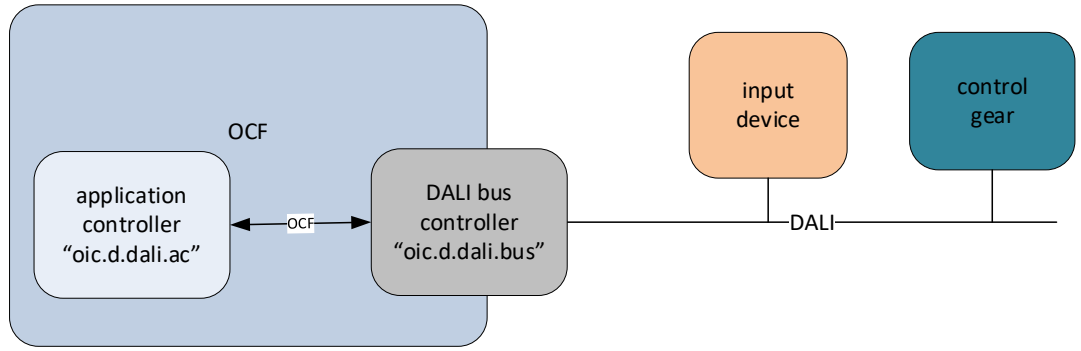
- Example overview
  - Smart light device exposing a binary switch (mandatory) and a brightness (optional) Resource
- Device type: Light (oic.d.light)
- Associated resources
  - Mandatory Core resources: /oic/res, /oic/p, /oic/d, Introspection
  - Mandatory Security Resources (not shown in the diagram)
  - Mandatory Resources for the Device Type: Binary switch (oic.r.switch.binary),
  - Other optional resources can be exposed, in this example Brightness resource (oic.r.light.brightness)

Device Name	Device Type	Resource Type	M/O
Light	oic.d.light	oic/res (oic.wk.res)	M
		oic/p (oic.wk.p)	M
		oic/d (oic.d.light)	M
		Introspection (oic.wk.introspection)	M
		Binary switch (oic.r.switch.binary)	M
		Brightness (oic.r.light.brightness)	O



Enabling DALI over OCF

- Goal: reuse existing DALI devices, enable the DAIL devices via IP.
- Keep all DALI instructions the same



- Designed as set of Resources, to convey the DALI FF/BF info
- Payload for the resources include the BF/FF "as is"
- Allows to add additional information over the FF/BF, for example the "send twice" instruction, making the DALI commands time insensitive on IP.

Complete Set of OCF Defined Device Types (1/3)

Friendly Name	Device Type
3D Printer	oic.d.3dprinter
A/C Assistant	oic.d.acassistant
AFCI	oic.d.afci
Activity Tracker	oic.d.activitytracker
Air Conditioner	oic.d.airconditioner
Air Purifier	oic.d.airpurifier
Air Quality Monitor	oic.d.airqualitymonitor
Airer	oic.d.airer
Battery	oic.d.battery
Blind	oic.d.blind
Blood Pressure Monitor	oic.d.bloodpressuremonitor
Body Composition Analyser	oic.d.bodycompositionanalyser
Body Scale	oic.d.bodyscale

Friendly Name	Device Type
Body Thermometer	oic.d.bodythermometer
Camera	oic.d.camera
Circuit Breaker	oic.d.circuitbreaker
Clothes Dryer	oic.d.dryer
Clothes Washer	oic.d.washer
Clothes Washer/Dryer	oic.d.washerdryer
Coffee Machine	oic.d.coffeemachine
Continuous Glucose Meter	oic.d.cgm
Cooker Hood	oic.d.cookerhood
Cooktop	oic.d.cooktop
Cycling Cadence Sensor	oic.d.cyclingcadencesensor
Cycling Power Meter	oic.d.cyclingpowermeter
Cycling Speed Sensor	oic.d.cyclingspeedsensor

Items in red are new in OCF 2.1.x and OCF 2.2.x

Complete Set of OCF Defined Device Types (2/3)

Friendly Name	Device Type
DALI	oic.d.dali
DALI Application Controller	oic.d.dali.ac
DALI Bus	oic.d.dali.bus
Dehumidifier	oic.d.dehumidifier
Dishwasher	oic.d.dishwasher
Door	oic.d.door
Electric Meter	oic.d.electricmeter
Electric Vehicle Charger	oic.d.electricvehiclecharger
Energy Generator	oic.d.energygenerator
Energy Monitor	oic.d.energymonitor
Fan	oic.d.fan
Food Probe	oic.d.foodprobe
Freezer	oic.d.freezer

Friendly Name	Device Type
Garage Door	oic.d.garagedoor
Generic Sensor	oic.d.sensor
Glucose Meter	oic.d.glucosemeter
Grinder	oic.d.grinder
GFCI	oic.d.gfci
Heart Rate Monitor	oic.d.heartratemonitor
Humidifier	oic.d.humidifier
Humidifier	oic.d.humidifier
Indoor Garden	oic.d.indoorgarden
Inverter	oic.d.inverter
Kettle	oic.d.kettle
Light	oic.d.light
Mattress	oic.d.mattress

Friendly Name	Device Type
Microwave Oven	oic.d.microwave
Muscle Oxygen Monitor	oic.d.muscleoxygenmonitor
Optical Augmented RFID Reader	oic.d.orfid
Oven	oic.d.oven
PV Array System	oic.d.pvarraysystem
Printer	oic.d.printer
Printer (Multi-Function)	oic.d.multifunctionprinter
Robot Cleaner	oic.d.robotcleaner
Scanner	oic.d.scanner
Security Panel	oic.d.securitypanel
Set Top Box	oic.d.stb
Sleep Monitor	oic.d.sleepmonitor
Smart Light	oic.d.light.smart
Smart Lock	oic.d.lock

Complete Set of OCF Defined Device Types (3/3)

Friendly Name	Device Type
Smart Plug	oic.d.smartplug
Speaker	oic.d.speaker
Steam Closet	oic.d.steamcloset
Switch	oic.d.switch
Television	oic.d.tv
Thermostat	oic.d.thermostat
Water Heater	oic.d.waterheater
Pulse Oximeter	oic.d.pulseoximeter
Receiver	oic.d.receiver
Refrigerator	oic.d.refrigerator
Wallpad	oic.d.wallpad
Water Purifier	oic.d.waterpurifier
Water Valve	oic.d.watervalve
Window	oic.d.window

OPEN CONNECTIVITY FOUNDATION®

References

Specification Location

Where can I find the specifications and Resource Type definitions?

OCF Specifications:

- https://openconnectivity.org/developer/specifications

Resource Type Definitions

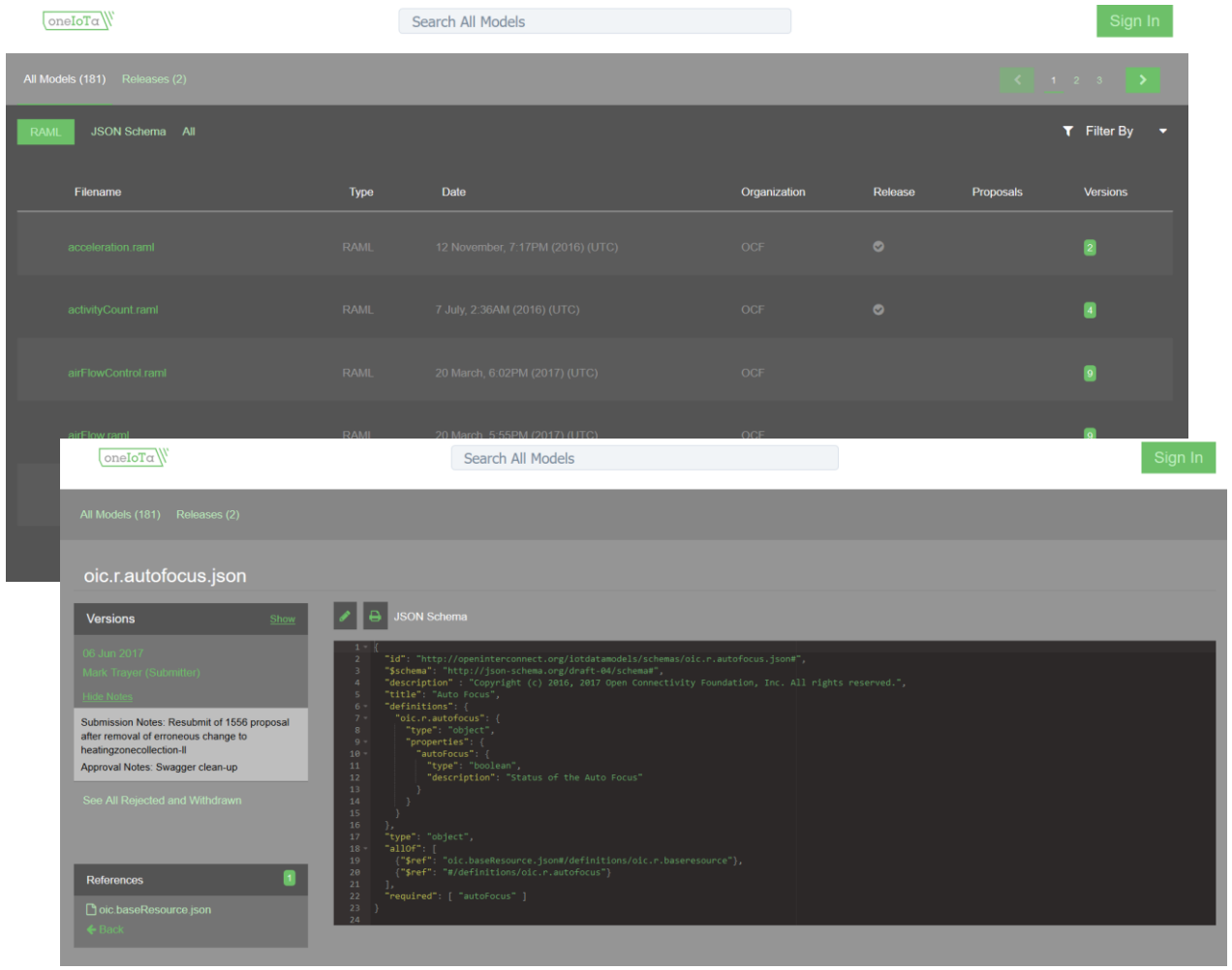
- Core Resources: https://github.com/openconnectivityfoundation/core
- Core Optional Resources and Extensions: https://github.com/openconnectivityfoundation/core-extensions
- Bridging Resources: https://github.com/openconnectivityfoundation/bridging
- Cloud Resources: https://github.com/openconnectivityfoundation/cloud-services
- Security Resources: https://github.com/openconnectivityfoundation/security
- Vertical Resources and Derived Models: https://oneiota.org/documents?filter%5Bmedia_type%5D=application%2Fswagger%2Bjson

Device and Resource Maps

A web front end to the github hosted device and resource maps that are maintained by OCF may be found here:

- https://openconnectivityfoundation.github.io/devicemodels/docs/index.html

OneIoTa Tool



- Web based (see: http://oneiota.org) development tool
- Supports RAML, JSON, and OpenAPI2.0 syntax
- Populated to date with all OCF Resources (in OpenAPI2.0), plus OCF-AllJoyn, OCF-BLE, OCF-LWM2M, OCF-oneM2M, OCF-Uplus, OCF-Zigbee, OCF-ZWave, and OCF-EnOcean derived models.
- Supports multiple organizations
  - Each submitting organization defines their own license terms

Thank you!

- Access the OCF specifications https://openconnectivity.org/developer/specifications/
- Contact OCF at admin@openconnectivity.org

OPEN CONNECTIVITY
FOUNDATION™