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.
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

- Application
- Resource Model
- Encoding (CBOR)
  - CoAP
  - DTLS
  - UDP
  - IPv6
- L2 Connectivity

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)
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.

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

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

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

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.

Main SVRs:
- cred
- pstat
- doxm
- acl2

Device Provisioning Status (pstat) is updated by the OBT services to trigger the state machine transitions:
- Unowned: RFO TM, Reset
- Owned: RFPRO, SRESET, RFNOP
OCF Specification Overview

Technical Principles

OCF 2.2.2 Release
February 2021
Technical Principles for an Internet of Things Ecosystem
## Scope of IoT

### Vertical Profiles
- Smart Home
- Industrial
- Healthcare
- ...

### Baseline Functionality
- Group management
- ID & Addressing
- Protocol Bridge/GW
- Device management
- Security

### Connectivity
- Wi-Fi
- BT/BLE
- Thread
- ...

---

**Local Control**

- Things

**Remote Control**

- Controller

**Server to Server**

- Controller App
- Controller
- Cloud Interface
- Cloud Servers

---

- February 2021
- Copyright © 2021 Open Connectivity Foundation, Inc. All Rights Reserved.
Approaches to definition of various Things

- By defining resources of things and its properties
  - BinarySwitch
    - true(on), false(off)
  - Dimming
    - dimmingSetting (int)
    - step (int)
    - range [0-100]
  - Brightness
    - brightness (int)

- By defining functions/operations of things
  - SetSwitch
    - Power(bool)
  - SetDimmingLevel
    - dimmingSetting (int)
  - SetBrightness
    - brightness (int)

- (no Verbs) + Objects
  - *Fixed set of verbs (CRUDN) from transport layer will be used

- (Verbs + Objects)
  - Resource model in RESTful Architecture
    - (e.g., W3C, CSEP, etc.)
  - 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

Device under Test → Conformance Test → Certificate Issue & Logo Licensing

Mandatory
(in spec, cert & committed in Open Source Project)

Open Source

Optional Open Source Features

Optional Open Source Features

Tested Optional Spec Features

Optional Spec Features

February 2021

Copyright © 2021 Open Connectivity Foundation, Inc. All Rights Reserved.
Introduction to the Open Connectivity Foundation
Introduction to OCF - Optimized for IoT

Common Platform

RESTful Architecture

Certification Program

CoAP for Constrained Devices

Best In Class Security
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.
OCF Specification Overview
OCF Deliverables

Nomative Specifications

- See next slide

Resource Models via oneloTa

- 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
Speciation 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%2Fyaml
OneIoTa Tool

- Web based (see: http://oneiota.org) development tool
- Supports RAML, JSON, and OpenAPI2.0 syntax
- Populated to date with all OCF Resources defined using OpenAPI2.0; also all derived models for the defined ecosystem bridges.
- Supports multiple organizations
  - Each submitting organization defines their own license terms
OCF Specification Overview
Core Framework Specification &
Core Optional Framework Specification
OCF 2.2.2 Release
February 2021
Objectives
RESTful Architecture
OCF Roles
Resources
Semantic Tags
Basic Operations
Organization of an OCF Device
OCF Specification Features
Protocol Stack
Core Framework Topics Outline (2 of 2)

- Endpoint Overview
- Resource Discovery (CoAP Discovery)
- Block Transfer with CoAP Messaging
- Encoding Schemes
- Introspection
- Collection Resources
- Atomic Measurement Resources
- Device Reset
- Versioning
Core Optional Framework Topics Outline

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

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

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

```
{n: "myRoomTemperature",
 rt: "oic.r.temperature",
 if: "oic.if.a",
 id: "example_id_xyz",
 temperature: 23,
 units: "C",
 setValue: 25}
```
• 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
An OCF Server contains one or more Resources to describe a real world entity.

Each Resource contains Properties that describe 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.
Semantic Tags

- Semantic Tags are meta-information associated with a specific Resource instance, and provide a mechanism to add additional data that helps describe the Resource.
- The Semantic Tag will be present as a Link Parameter in all Links that are present that target the Resource. The Semantic Tag is further treated as a Common Property associated with the Resource.

<table>
<thead>
<tr>
<th>Positional Tags</th>
<th>Functional Behaviour Tags</th>
<th>Location Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>tag-pos-desc</td>
<td>tag-pos-rel</td>
<td>tag-func-desc</td>
</tr>
<tr>
<td>Descriptive text describing position (e.g. topleft)</td>
<td>Relative positions assuming a (1,1,1) cube</td>
<td>Functional description, can take any OCF defined mode value</td>
</tr>
<tr>
<td>Read Only</td>
<td>Read Only</td>
<td>Read Only</td>
</tr>
<tr>
<td>e.g. “tag-pos-desc”: “topleft”</td>
<td>e.g. “tag-pos-rel”: [0.5, 0.5, 0.5]</td>
<td>e.g. “tag-func-desc”: “freezer”</td>
</tr>
</tbody>
</table>
OCF Core Framework Basic Operation

**Discovery**
- Discover access policies, device info and resources on the devices

**Operation**
- Get device information by retrieving resources
- Control devices by changing resources
- Observe change on the properties of resources

**Basic common services**
- Device Monitoring
- Maintenance (e.g., reboot, factory reset, software update, etc.)
Organization of an OCF Device

- OCF Device concept

![Diagram showing the organization of an OCF Device]

Resource URI: `/oic/p`
- rt: oic.wk.p
- if: oic.if.r
- n: homePlatform
- pi: at1908
- mnmn: Samsung

*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
**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

```
[  
  {  "href": "/oic/res",
      "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017ea863989/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-017ea863989",
      "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-017ea863989",
      "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-017ea863989",
      "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"} ]}
]
```
Resource Discovery (Default Mechanism)

- OCF devices by default make use of CoAP Discovery using the IANA defined OCF Service 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)
Resource Discovery (Optional Mechanism)

- OCF devices may make use of CoAP Discovery using the IANA defined All CoAP Nodes Service Address (FF0X::FD).
- Multicast RETRIEVE (CoAP GET) sent to “.well-known/core”
- Response is encoded as defined by CoAP, and is an array with a single link; that link being the link to /oic/res with the extension that this contains a multi-value “rt”, and thus include also the device type of the device.

Example:

Request: GET coap://[FF02::FD]:5683/.well-known/core?rt=oic.wk.res
Response: 2.05 Content, Content-Format: 40
<coap://[fe80::b1d6]:1111/oic/res>;ct=10000;rt="oic.wk.res oic.d.sensor";if="oic.if.11 oic.if.baseline";
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

<table>
<thead>
<tr>
<th>CBOR</th>
<th>JSON</th>
<th>XML/EXI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Concise binary object representation based on JSON data model</td>
<td>Lightweight, text-based, language-independent data interchange format</td>
</tr>
<tr>
<td><strong>Standard</strong></td>
<td>IETF RFC 7049</td>
<td>IETF RFC 7159</td>
</tr>
<tr>
<td><strong>ContentType</strong></td>
<td>/application/vnd.ocf+cbor</td>
<td>/application/json</td>
</tr>
<tr>
<td><strong>OCF M/O</strong></td>
<td>Mandatory</td>
<td>Can be supported</td>
</tr>
</tbody>
</table>

*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
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
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

<table>
<thead>
<tr>
<th>Function</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVRs</td>
<td>As per Security Specification</td>
</tr>
<tr>
<td>Mandatory Core Resources</td>
<td>Properties in oic.wk.d, oic.wk.p set to defined defaults</td>
</tr>
<tr>
<td>Optional Core Resources</td>
<td>Properties in oic.wk.con set to defined defaults</td>
</tr>
<tr>
<td>Vertical Resources</td>
<td>Reset to vendor defined defaults</td>
</tr>
<tr>
<td>Created Resources</td>
<td>Deleted</td>
</tr>
<tr>
<td>Observe Transactions</td>
<td>Canceled with a (typically) 5.03</td>
</tr>
</tbody>
</table>
Payload Versioning

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

Device Level Versioning

- **Purpose**: OCF devices can be aware of each other's specification implementation version
- **Method**: icv (specification version) in /oic/d resource
Payload versioning

<table>
<thead>
<tr>
<th>Media Type</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>application/cbor</td>
<td>60</td>
</tr>
<tr>
<td>application/vnd.ocf+cbor</td>
<td>10000</td>
</tr>
</tbody>
</table>

Content-Formats

<table>
<thead>
<tr>
<th>CoAP Option Number</th>
<th>Name</th>
<th>Format</th>
<th>Length (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2049</td>
<td>Accept Version</td>
<td>uint</td>
<td>2</td>
</tr>
<tr>
<td>2053</td>
<td>Content-Format Version</td>
<td>uint</td>
<td>2</td>
</tr>
</tbody>
</table>

Option Numbers

Version Representation

<table>
<thead>
<tr>
<th>Major Version</th>
<th>Minor Version</th>
<th>Sub Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</td>
<td></td>
</tr>
</tbody>
</table>

Version Example

<table>
<thead>
<tr>
<th>OCF version</th>
<th>Binary representation</th>
<th>Integer value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0</td>
<td>0000 1000 0000 0000</td>
<td>2048</td>
</tr>
<tr>
<td>1.1.0</td>
<td>0000 1000 0100 0000</td>
<td>2112</td>
</tr>
</tbody>
</table>
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.
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, etc.

- 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
• 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

1. Create user account
2. Mediator Registration
3. Device Association (Device ID)
4. Cloud configuration (Cloud IP, Access token for the device)
5. TLS session
6. Device Registration, Resource Publish (Access token, Device ID)
7. Validate Token
8. Share User Authentication (Access token, user id)

[OCF Device]

[OCF Cloud Services]

[Mediator]

[Authorisation Server]

[Note: The dotted interface is out of OCF scope]
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.
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.
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 another entity. For the latter case, the OCF Cloud Services host also returns an Authorisation Provider Name (“apn”) in addition to the Access Token.
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").
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.
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.
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.
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).
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. "refresh_token"

2. The OCF Cloud Services host responds with a new
   1. "access_token"
   2. "refresh_token"
   3. "expiresin"
• 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
  • "login": false

Operational Flow Detail: Closing Connection with OCF Cloud Services
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.
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
Wi-Fi Easy Setup
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.
    • 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 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.
‘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

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Supported Interface</th>
<th>Example URI</th>
<th>Resource Type</th>
<th>CRUDN permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>EasySetup</td>
<td>Baseline, link-list, batch</td>
<td>/example/EasySetupResURI</td>
<td>oic.r.easysetup, oic.wk.col</td>
<td>RU</td>
</tr>
</tbody>
</table>

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

February 2021

Copyright © 2021 Open Connectivity Foundation, Inc. All Rights Reserved.
### Easy Setup - Wi-Fi Conf. Resource

- Contains Wi-Fi-related properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Name(key)</th>
<th>Value Type</th>
<th>Value Rule</th>
<th>Access Mode</th>
<th>M / O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported Wi-Fi Mode Type</td>
<td>swmt</td>
<td>array of string</td>
<td>enum</td>
<td>R</td>
<td>M</td>
<td>Indicates supported Wi-Fi mode types. It can be multiple. (i.e. “A”, “B”, “G”, “N”, “AC”)</td>
</tr>
<tr>
<td>Supported Wi-Fi Freq.</td>
<td>swf</td>
<td>array of string</td>
<td>R</td>
<td>M</td>
<td>Indicates supported Wi-Fi Frequency by Enrollee. Can be multiple. (i.e. “2.4G”, “5G”)</td>
<td></td>
</tr>
<tr>
<td>Target Network Name</td>
<td>tnn</td>
<td>string</td>
<td>RW</td>
<td>M</td>
<td>Indicates SSID of Wi-Fi AP i.e. Enroller.</td>
<td></td>
</tr>
<tr>
<td>Credential</td>
<td>cd</td>
<td>string</td>
<td>RW</td>
<td>M</td>
<td>Indicates credential information of Wi-Fi AP (password used to connect to enrollee).</td>
<td></td>
</tr>
<tr>
<td>Wi-Fi Auth Type</td>
<td>wat</td>
<td>string</td>
<td>enum</td>
<td>RW</td>
<td>M</td>
<td>Indicates Wi-Fi Auth Type (i.e. “None”, “WEP”, “WPA-PSK”, “WPA2-PSK”)</td>
</tr>
<tr>
<td>Wi-Fi Encryption Type</td>
<td>wet</td>
<td>string</td>
<td>enum</td>
<td>RW</td>
<td>M</td>
<td>Indicates Wi-Fi Encryption Type (i.e. “None”, “WEP_64”, “WEP_128”, “TKIP”, “AES”, “TKIP_AES”)</td>
</tr>
<tr>
<td>Supported Wi-Fi Auth Type</td>
<td>swat</td>
<td>array of string</td>
<td>enum</td>
<td>R</td>
<td>M</td>
<td>Supported Wi-Fi Auth types. Can be multiple. (“None”, “WEP”, “WPA_PS_K”, “WPA2_PS_K”)</td>
</tr>
<tr>
<td>Supported Wi-Fi Encryption Type</td>
<td>swet</td>
<td>array of string</td>
<td>enum</td>
<td>R</td>
<td>M</td>
<td>Supported Wi-Fi Encryption types. Can be multiple. (“None”, “WEP-64”, “WEP_128”, “TKIP”, “AES”, “TKIP_AES”)</td>
</tr>
</tbody>
</table>
Easy Setup - Dev Conf. Resource

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

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Supported Interface</th>
<th>Example URI</th>
<th>Resource Type</th>
<th>CRUDN permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Conf.</td>
<td>Read Only, Baseline</td>
<td>/example/DevConfResURI</td>
<td>oic.r.devconf</td>
<td>RU</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Name(key)</th>
<th>Value Type</th>
<th>Value Rule</th>
<th>Access Mode</th>
<th>M / O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Name</td>
<td>dn</td>
<td>one of: string or array of object</td>
<td>R</td>
<td>M</td>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>
Step 1: 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
• 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**

- **Standard**
- **Micro**
- **Nano**

**eSIM and Remote SIM Provisioning**

- **eSIM Profile**
- **SM-DP+ Server**
- **LPA (eSIM control Software)**
- **OCF Device**
- **eUICC (MFF2)**

**SM-DP+: Subscription Manager Data Preparation**

- LPA: Local Profile Assistant
- MFF2 - M2M Form Factor
  - 6mm x 5mm x 0.9mm

**eSIM Profile Download**

- GSMA Remote SIM Provisioning (RSP) standards
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.
5. From the assigned SM-DP+ server, the Device downloads an eSIM Profile with the Token.
6. Enrollee connects to the cellular network directly 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

- ‘eSIM EasySetup’ 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.

Resources
- eSIM EasySetup Resource (Collection)
  - RSP Capability Resource
  - RSP Conf Resource

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.

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Supported Interface</th>
<th>Example URI</th>
<th>Resource Type</th>
<th>CRUDN permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>eSIMEasySetup</td>
<td>Baseline, Link, Batch</td>
<td>/example/eSIMEasySetupResURI</td>
<td>oic.r.esimeasysetup</td>
<td>RUN</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Supported Interface</th>
<th>Example URI</th>
<th>Resource Type</th>
<th>CRUDN permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSPCapability</td>
<td>Read, Baseline</td>
<td>/example/RSPCapabilityResURI</td>
<td>oic.r.rspcapability</td>
<td>R</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Name(key)</th>
<th>Value Type</th>
<th>Value Rule</th>
<th>Access Mode</th>
<th>M / O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eUICC Information</td>
<td>euiccinfo</td>
<td>string</td>
<td>Max.1024 octets</td>
<td>R</td>
<td>M</td>
<td>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.</td>
</tr>
<tr>
<td>Device Information for RSP</td>
<td>deviceinfo</td>
<td>string</td>
<td>Max.128 octets</td>
<td>R</td>
<td>M</td>
<td>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.</td>
</tr>
</tbody>
</table>
eSIM Easy Setup - RSPConf Resource

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

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Supported Interface</th>
<th>Example URI</th>
<th>Resource Type</th>
<th>CRUDN permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSPConf</td>
<td>Read Write, Baseline</td>
<td>/example /RSPConfResURI</td>
<td>oic.r.rspconf</td>
<td>RU</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Name(key)</th>
<th>Value Type</th>
<th>Value Rule</th>
<th>Access Mode</th>
<th>M / O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activation Code</td>
<td>ac</td>
<td>string</td>
<td>Max.256 characters</td>
<td>RW</td>
<td>M</td>
<td>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</td>
</tr>
<tr>
<td>eSIM Profile Metadata</td>
<td>pm</td>
<td>string</td>
<td>Max. 2048 octets</td>
<td>R</td>
<td>M</td>
<td>Refers to “ProfileInfo” in the GSMA RSP Technical Specification Annex H. This value type shall be encoded as Major Type 2.</td>
</tr>
<tr>
<td>Confirmation Code</td>
<td>cc</td>
<td>string</td>
<td>N/A</td>
<td>RW</td>
<td>O</td>
<td>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.</td>
</tr>
<tr>
<td>Confirmation Code Required</td>
<td>ccr</td>
<td>boolean</td>
<td>N/A</td>
<td>R</td>
<td>M</td>
<td>Indicates whether a Confirmation Code is required. Set to &quot;true&quot; if Confirmation Code is required and required a user to enter Confirmation Code</td>
</tr>
</tbody>
</table>
Example eSIM Easy Setup Flow with Wi-Fi Easy Setup (informative)

Soft AP Connect: Steps 1-3:
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:

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.
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 (OAuth)

2. Device and Resource Discovery

3. Resource Control (Retrieve and Update)

4. Event (Subscription and Notification)

Out-of-scope of OCF C2C

In-scope of OCF D2C

In-scope of OCF C2C

Cloud URI is known out of band

Secure Connection Setup (TLS)

Onboarding Device to Cloud

Resource Control

Observe

Notification

February 2021

Copyright © 2021 Open Connectivity Foundation, Inc. All Rights Reserved.
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:
  ectivityfoundation/core-extensions/ocfcloud-
  openapi/swagger2.0/oic.r.cloudopenapi.swagger.json
OCF Specification Overview

Security Specification

OCF 2.2.2 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:

Server – Responding Device

Application Resources

/oic/d  /oic/light/3  Etc...

Allow /Deny Resource Access
 acl(s)
 Subject
 Resource(s)
 Permission

Security Layer
 service(s)
 DeviceID
 SvcType
 CredID

cred(s)
 DeviceID
 CredType
 PrivateData

Session Layer

Connectivity Abstraction Layer

Secure Channel

Request Access

Client – Requesting Device

1  2  3

DTLS, OSCORE
Simplified Onboarding Sequence

- **Unowned Device boots**
- **Discovery (unsecured):**
  - DOTS sends multicast to discover unowned devices  
  - Unowned devices reply, including list of supported OTMs
- **Ownership Transfer:**
  - DOTS selects and configures this OTM to the new device  
  - DOTS & unowned Device perform OTM, inc. TLS handshake
  - DOTS config SVRs to authorize itself, CMS and AMS  
  - Device is now owned!
- **Provisioning:**
  - CMS provisions credentials, AMS provisions access policies  
  - Device is now provisioned and can commence normal operation
- **Normal Operation:**
  - Credentials and/or access policies can be updated by returning to Provisioning
Device Provisioning States

**RESET**
- Hard reset
- Owned?: No
- Provisioned?: No
- Permitted operations: Device prepares SVRs for OTM

**RFOTM**
- Ready For Ownership Transfer Mechanism
- Owned?: No
- Provisioned?: No
- Permitted operations: DOTS configures /doxm to claim ownership. Configures owners of SVRs & their credentials in /cred

**RFPRO**
- Ready For Provisioning
- Owned?: Yes
- Provisioned?: No/Partly → Yes
- Permitted operations: CMS updates /cred AMS updates /acl2

**RFNOP**
- Ready For Normal Operation
- Owned?: Yes
- Provisioned?: Yes
- Permitted operations: Normal Device-to-Device Operation subject to SVRs

**SRESET**
- Soft Reset
- Owned?: Yes
- Provisioned?: Some SVRs Corrupted
- Permitted Operations: DOTS attempts to repairs SVRs

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)

**Notes:**
- 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

**FEBRUARY 2021**

COPYRIGHT © 2021 OPEN CONNECTIVITY FOUNDATION, INC. ALL RIGHTS RESERVED.
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

Is Light On?

Request
Accept

 acl[0]
subject: DeviceID
rsrc: [/a/light]
pms: R

Turn Light Off

Request
Reject

 acl[0]
subject: DeviceID
rsrc: [/a/light]
pms: R

February 2021
Copyright © 2021 Open Connectivity Foundation, Inc. All Rights Reserved.
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 types of access control mechanisms 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 apply 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 to 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)

<table>
<thead>
<tr>
<th>oic.r.doxm</th>
<th>oic.r.cred</th>
<th>oic.r.pstat</th>
<th>oic.r.roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>oxm</td>
<td>creds</td>
<td>dos</td>
<td>roles</td>
</tr>
<tr>
<td>oxmsel</td>
<td>rowneruuid</td>
<td>isop</td>
<td></td>
</tr>
<tr>
<td>sct</td>
<td></td>
<td>cm</td>
<td></td>
</tr>
<tr>
<td>owned</td>
<td></td>
<td>tm</td>
<td></td>
</tr>
<tr>
<td>deviceuuid</td>
<td></td>
<td>om</td>
<td></td>
</tr>
<tr>
<td>devowneruuid</td>
<td></td>
<td>sm</td>
<td></td>
</tr>
<tr>
<td>rowneruuid</td>
<td></td>
<td>rowneruuid</td>
<td></td>
</tr>
<tr>
<td>aclist2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rowneruuid</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.

---

**Certificate-based Onboarding**

- **Shared Key Onboarding**
- **Anonymous Onboarding**
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.
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 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.
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.
• 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
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
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 the 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 Core Spec
- OCF Security Spec
- OCF Device Type Spec
- OCF Resource Type Spec

- OCF Bridging Spec
  - OCF Resource to AllJoyn Mapping Spec
  - OCF Resource to ZigBee Mapping Spec
  - OCF Resource to BLE Mapping Spec

- Ecosystem specific Data Model Translation rules
- Ecosystem specific Communication Message Translation rules
- Ecosystem Device specific Data Model Mapping rules

- OCF common Bridging Frameworks

February 2021
Copyright © 2021 Open Connectivity Foundation, Inc. All Rights Reserved.
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

Bridge Platform

Virtual OCF Devices

Asymmetric Client Bridge
- OCF Server
- OCF Client

Asymmetric Server Bridge
- OCF Server
- OCF Client

Bridge
- Virtual OCF Client
- Virtual OCF Server
- Bridging Function
- Virtual Bridged Server
- Virtual Bridged Client

Bridged
- Client
- Server

OCF
- Protocol

Non-OCF

February 2021

Copyright © 2021 Open Connectivity Foundation, Inc. All Rights Reserved.
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
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:

Resource Discovery
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**

- **Test case**
  - Test cases for normative text in CR (“shall” statement in CR)
# OCF Bridging: Bridging per Ecosystem

<table>
<thead>
<tr>
<th>BLE service-characteristic</th>
<th>OCF resource-property</th>
</tr>
</thead>
<tbody>
<tr>
<td>GATT-based profile(s)</td>
<td>OCF device</td>
</tr>
<tr>
<td>Service</td>
<td>OCF resource(s)</td>
</tr>
<tr>
<td>Characteristic</td>
<td>OCF resource property</td>
</tr>
<tr>
<td>Characteristic configuration descriptor</td>
<td>OCF notification on/off option</td>
</tr>
</tbody>
</table>

### BLE Example

- **GATT-based profile(s)**
  - Service #1
    - Characteristic
    - Characteristic
  - Service #2
    - Characteristic
    - Characteristic
  - Service #3
    - Characteristic
    - Characteristic
  - ...

- **OCF Device**
  - Resource #1
    - Property
    - Property
  - Resource #2
    - Property
    - Property
  - Resource #2
    - Property
    - Property
  - ...

---

February 2021

Copyright © 2021 Open Connectivity Foundation, Inc. All Rights Reserved.
**OCF Bridging : Bridging per Ecosystem**

- **Protocol Translation Mechanism b/w BLE and OCF - BLE example**

<table>
<thead>
<tr>
<th>BLE GATT feature</th>
<th>OCF CRUDN</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A (can’t create new service/characteristic)</td>
<td>CREATE</td>
</tr>
<tr>
<td>Characteristic Value Read</td>
<td>RETRIEVE</td>
</tr>
<tr>
<td>Characteristic Value Write</td>
<td>UPDATE</td>
</tr>
<tr>
<td>N/A (can’t delete existing service/characteristic)</td>
<td>DELETE</td>
</tr>
<tr>
<td>Characteristic Descriptor Value Write</td>
<td></td>
</tr>
<tr>
<td>Characteristic Value Notification</td>
<td>NOTIFY</td>
</tr>
</tbody>
</table>
### Table 5 - "ocn.w.k.d" Resource Type definition

<table>
<thead>
<tr>
<th>To OCF Property title</th>
<th>OCF Property name</th>
<th>OCF Description</th>
<th>OCF Mandatory?</th>
<th>From BLE Device Service Characteristic value</th>
<th>BLE Description</th>
<th>BLE Mandatory?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Device) Name</td>
<td>&quot;n&quot;</td>
<td>Human friendly name example: &quot;Eco- Thermostat&quot;</td>
<td>Y</td>
<td>Device Name</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Spec Version</td>
<td>&quot;ivs&quot;</td>
<td>ISO/IEC 30118-1:2015 the device is implemented to the syntax in &quot;core major.minor&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device ID</td>
<td>&quot;d&quot;</td>
<td>unique id for device. This value will be as defined in ISO/IEC 30118-2:2015 for DeviceID</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protocol-Independent</td>
<td>&quot;pilid&quot;</td>
<td>unique id for OCF Device (UUID)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 6 - "ocn.w.k.p" Resource Type definition

<table>
<thead>
<tr>
<th>To OCF Property title</th>
<th>OCF Property name</th>
<th>OCF Description</th>
<th>OCF Mandatory?</th>
<th>From BLE Device Service Characteristic value</th>
<th>BLE Description</th>
<th>BLE Mandatory?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform ID</td>
<td>&quot;pID&quot;</td>
<td>Unique identifier for the physical platform (UUID), this shall be a UUID in accordance with IETF RFC 4122</td>
<td>Y</td>
<td>(none)</td>
<td>(none)</td>
<td></td>
</tr>
</tbody>
</table>

### Table 7 - "ocn.w.k.cop.p" Resource Type definition

<table>
<thead>
<tr>
<th>To OCF Property title</th>
<th>OCF Property name</th>
<th>OCF Description</th>
<th>OCF Mandatory?</th>
<th>From BLE Device Service Characteristic value</th>
<th>BLE Description</th>
<th>BLE Mandatory?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer Name</td>
<td>&quot;mmnm&quot;</td>
<td>Name of manufacturer (not to exceed 16 characters)</td>
<td>Y</td>
<td>This characteristic represents the name of the manufacturer</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Manufacturer Name String</td>
<td>Device Information</td>
<td>This characteristic represents the name of the manufacturer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
OCF Bridging: Bridging per Ecosystem

- Translation *per each Device Type* - Derived Model

OCF Resource Model

Derived Model

Non-OCF Ecosystem Resource Model

Resource Spec

Device Spec

Mapping Spec

Other Ecosystem Spec
OCF Bridging : Bridging per Ecosystem

- Translation per each Device Type
  - Derived Model

Z-wave binary switch ↔ OCF Binary Switch

```
{
  "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 Class",
  "definitions": {
    "zwave.operation.binaryswitchcommandclass": {
      "type": "object",
      "properties": {
        "value": {
          "type": "boolean",
          "description": "On/off state at the receiving node",
          "x-ocf-conversion": {
            "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"]
}
```
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

```json
"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."
        ]
      }
    }
  }
```

AllJoyn Interface Name

AllJoyn Property Name

OCF Equivalent Resource Type

To OCF from AllJoyn

From OCF to AllJoyn

February 2021

Copyright © 2021 Open Connectivity Foundation, Inc. All Rights Reserved.
## OCF Bridging: Bridging per Ecosystem

<table>
<thead>
<tr>
<th>Ecosystem Name</th>
<th>BLE</th>
<th>ZigBee</th>
<th>Z-wave</th>
<th>U+</th>
<th>EnOcean</th>
<th>oneM2M</th>
<th>LwM2M (under development)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetric? or Asymmetric?</td>
<td>Asymmetric (Asymmetric Server Bridging)</td>
<td>Asymmetric (Asymmetric Server Bridging)</td>
<td>Asymmetric (Asymmetric Server Bridging)</td>
<td>Asymmetric (Asymmetric Server Bridging)</td>
<td>Asymmetric (Asymmetric Server Bridging)</td>
<td>Asymmetric (Asymmetric Client Bridging)</td>
<td>Asymmetric (Asymmetric Client Bridging)</td>
</tr>
</tbody>
</table>
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.
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:

<table>
<thead>
<tr>
<th>Optional or Mandatory</th>
<th>Statements</th>
<th>Compliance (Y/N)</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional</td>
<td>The Platform implements an OCF Bridge Device which bridges to the Zigbee Protocol</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 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.
Data Model per Ecosystem Bridges:

AllJoyn, oneM2M, ZigBee, BLE, Z-wave, U+, EnOcean
## AllJoyn Device Type Equivalency

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

- Yellow highlights identify Device Types that were added to support equivalency.
## AllJoyn Interface to Resource Mapping

<table>
<thead>
<tr>
<th>AllJoyn Interface</th>
<th>OCF Resource Type Name</th>
<th>OCF Resource Type ID</th>
<th>OCF Interface(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment.CurrentAirQuality</td>
<td>Air Quality Collection</td>
<td>oic.r.airqualitycollection</td>
<td>oic.if.s</td>
</tr>
<tr>
<td>Environment.CurrentAirQualityLevel</td>
<td>Air Quality Collection</td>
<td>oic.r.airqualitycollection</td>
<td>oic.if.s</td>
</tr>
<tr>
<td>Environment.CurrentHumidity</td>
<td>Humidity</td>
<td>oic.r.humidity</td>
<td>oic.if.s</td>
</tr>
<tr>
<td>Environment.CurrentTemperature</td>
<td>Temperature</td>
<td>oic.r.temperature</td>
<td>oic.if.s</td>
</tr>
<tr>
<td>Environment.TargetHumidity</td>
<td>Humidity</td>
<td>oic.r.humidity, oic.r.selectablelevels</td>
<td>oic.if.a</td>
</tr>
<tr>
<td>Environment.TargetTemperature</td>
<td>Temperature</td>
<td>oic.r.temperature</td>
<td>oic.if.a</td>
</tr>
<tr>
<td>Operation.AudioVolume</td>
<td>Audio Controls</td>
<td>oic.r.audio</td>
<td>oic.if.a</td>
</tr>
<tr>
<td>Operation.Channel</td>
<td>Not mapped</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operation.ClimateControlMode</td>
<td>Mode</td>
<td>oic.r.mode</td>
<td>oic.if.a</td>
</tr>
<tr>
<td>Operation.ClosedStatus</td>
<td>Operational State</td>
<td>oic.r.operational.state</td>
<td>oic.if.s</td>
</tr>
<tr>
<td>Operation.CycleControl</td>
<td>Door</td>
<td>oic.r.door</td>
<td>oic.if.s</td>
</tr>
<tr>
<td>Operation.FanSpeedLevel</td>
<td>Operational State</td>
<td>oic.r.operational.state</td>
<td>oic.if.s</td>
</tr>
<tr>
<td>Operation.HeatingZone</td>
<td>Air Flow</td>
<td>oic.r.airflow</td>
<td>oic.if.a</td>
</tr>
<tr>
<td>Operation.OvenCyclePhase</td>
<td>Heating Zone Collection</td>
<td>oic.r.heatingzonecollection</td>
<td>oic.if.s</td>
</tr>
<tr>
<td>Operation.HvacFanMode</td>
<td>Mode</td>
<td>oic.r.mode</td>
<td>oic.if.a</td>
</tr>
<tr>
<td>Operation.OnOffStatus</td>
<td>Binary Switch</td>
<td>oic.r.switch.binary</td>
<td>oic.if.s</td>
</tr>
<tr>
<td>Operation.OvenCyclePhase</td>
<td>Operational State</td>
<td>oic.r.operationalstate</td>
<td>oic.if.s</td>
</tr>
</tbody>
</table>
Ecosystem Bridges:  
AllJoyn, oneM2M, ZigBee, BLE, Z-wave, U+, EnOcean
<table>
<thead>
<tr>
<th>oneM2M Device Type</th>
<th>OCF Device Type</th>
<th>oneM2M Device Type</th>
<th>OCF Device Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>device3DPrinter</td>
<td>oic.d.3dprinter</td>
<td>deviceMicrogeneration</td>
<td>oic.d.energygenerator</td>
</tr>
<tr>
<td>deviceAirConditioner</td>
<td>oic.d.airconditioner</td>
<td>deviceMultiFunctionPrinter</td>
<td>oic.d.multipfunctionprinter</td>
</tr>
<tr>
<td>deviceAirPurifier</td>
<td>oic.d.airpurifier</td>
<td>deviceOutdoorLamp</td>
<td>oic.d.light</td>
</tr>
<tr>
<td>deviceAirQualityMonitor</td>
<td>oic.d.airqualitymonitor</td>
<td>deviceOven</td>
<td>oic.d.oxygen</td>
</tr>
<tr>
<td>deviceAudioReceiver</td>
<td>oic.d.audioreceiver</td>
<td>devicePrinter</td>
<td>oic.d.printer</td>
</tr>
<tr>
<td>deviceBloodPressureMonitor</td>
<td>oic.d.bloodpressuremonitor</td>
<td>deviceRefrigerator</td>
<td>oic.d.refrigerator</td>
</tr>
<tr>
<td>deviceCamera</td>
<td>oic.d.camera</td>
<td>deviceRobotCleaner</td>
<td>oic.d.robotcleaner</td>
</tr>
<tr>
<td>deviceClothesDryer</td>
<td>oic.d.clothesdryer</td>
<td>deviceScanner</td>
<td>oic.d.scanner</td>
</tr>
<tr>
<td>deviceClothesWasher</td>
<td>oic.d.clotheswasher</td>
<td>deviceSecurityPanel</td>
<td>oic.d.securitypanel</td>
</tr>
<tr>
<td>deviceCoffeeMachine</td>
<td>oic.d.coffeemachine</td>
<td>deviceSetTopBox</td>
<td>oic.d.stb</td>
</tr>
<tr>
<td>deviceCookerHood</td>
<td>oic.d.cookerhood</td>
<td>deviceSmartElectricMeter</td>
<td>oic.d.electricmeter</td>
</tr>
<tr>
<td>deviceCooktop</td>
<td>oic.d.cooktop</td>
<td>deviceSmartPlug</td>
<td>oic.d.smartplug</td>
</tr>
<tr>
<td>deviceDehumidifier</td>
<td>oic.d.dehumidifier</td>
<td>deviceSteamCloset</td>
<td>oic.d.steamcloset</td>
</tr>
<tr>
<td>deviceDishWasher</td>
<td>oic.d.dishwasher</td>
<td>deviceStorageBattery</td>
<td>oic.d.battery</td>
</tr>
<tr>
<td>deviceDoor</td>
<td>oic.d.door</td>
<td>deviceSwitch</td>
<td>oic.d.switch</td>
</tr>
<tr>
<td>deviceDoorLock</td>
<td>oic.d.smartlock</td>
<td>deviceTelevision</td>
<td>oic.d.tv</td>
</tr>
<tr>
<td>deviceElectricVehicleCharger</td>
<td>oic.d.electricvehiclecharger</td>
<td>deviceThermostat</td>
<td>oic.d.thermostat</td>
</tr>
<tr>
<td>deviceFan</td>
<td>oic.d.fan</td>
<td>deviceWaterHeater</td>
<td>oic.d.waterheater</td>
</tr>
<tr>
<td>deviceFoodProbe</td>
<td>oic.d.foodprobe</td>
<td>deviceWaterValve</td>
<td>oic.d.watervalve</td>
</tr>
<tr>
<td>deviceFreezer</td>
<td>oic.d.freezer</td>
<td>deviceWeightScaleAndBodyCompositionAnalyzer</td>
<td>oic.d.bodyscale</td>
</tr>
<tr>
<td>deviceGlucosemeter</td>
<td>oic.d.glucosemeter</td>
<td>deviceWindowShade</td>
<td>oic.d.blind</td>
</tr>
<tr>
<td>deviceHumidifier</td>
<td>oic.d.humidifier</td>
<td>deviceThermometer</td>
<td>oic.d.bodythermometer</td>
</tr>
<tr>
<td>deviceKettle</td>
<td>oic.d.kettle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>deviceLight</td>
<td>oic.d.light</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Yellow highlights identify Device Types that were added to support equivalency.
## OCF Resources to oneM2M Module Classes

<table>
<thead>
<tr>
<th>oneM2M Module Class</th>
<th>OCF Resource Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3Dprinter</td>
<td>oic.r.3dprinter</td>
</tr>
<tr>
<td>acousticsensor</td>
<td>oic.r.soundpressure</td>
</tr>
<tr>
<td>airflow</td>
<td>oic.r.airflow</td>
</tr>
<tr>
<td>airpurifierjobmode</td>
<td>oic.r.operational.state</td>
</tr>
<tr>
<td>airqualitysensor</td>
<td>oic.r.airquality</td>
</tr>
<tr>
<td>alaramspeaker</td>
<td>oic.r.audiovolume</td>
</tr>
<tr>
<td>audioVolume</td>
<td>oic.r.audio</td>
</tr>
<tr>
<td>autodocumentfeeder</td>
<td>oic.r.operational.state</td>
</tr>
<tr>
<td>battery</td>
<td>oic.r.energy.battery</td>
</tr>
<tr>
<td>binaryswitch</td>
<td>oic.r.switch.binary</td>
</tr>
<tr>
<td>boiler</td>
<td>oic.r.sensor</td>
</tr>
<tr>
<td>brewing</td>
<td>oic.r.brewing</td>
</tr>
<tr>
<td>brightness</td>
<td>oic.r.light.brightness</td>
</tr>
<tr>
<td>clock</td>
<td>oic.r.clock</td>
</tr>
<tr>
<td>clothesdryerjobmode</td>
<td>oic.r.operational.state</td>
</tr>
<tr>
<td>colour</td>
<td>oic.r.colour</td>
</tr>
<tr>
<td>coloursaturation</td>
<td>oic.r.colour.saturation</td>
</tr>
<tr>
<td>credentials</td>
<td>oic.r.userinfo</td>
</tr>
<tr>
<td>dehumidifierjobmode</td>
<td>oic.r.operational.state</td>
</tr>
<tr>
<td>doorStatus</td>
<td>oic.r.door</td>
</tr>
<tr>
<td>electricvehicleconnector</td>
<td>oic.r.vehicle.connector</td>
</tr>
<tr>
<td>energyconsumption</td>
<td>oic.r.energy.electrical</td>
</tr>
<tr>
<td>energygeneration</td>
<td>oic.r.energy.generation</td>
</tr>
<tr>
<td>filterinfo</td>
<td>oic.r.consumable</td>
</tr>
<tr>
<td>foaming</td>
<td>oic.r.foaming</td>
</tr>
<tr>
<td>grinder</td>
<td>oic.r.grinder</td>
</tr>
<tr>
<td>heatingzone</td>
<td>oic.r.heatingzone</td>
</tr>
<tr>
<td>height</td>
<td>oic.r.height</td>
</tr>
<tr>
<td>hotwaternsupply</td>
<td>oic.r.switch.binary</td>
</tr>
<tr>
<td>impactsensor</td>
<td>oic.r.impactsensor</td>
</tr>
<tr>
<td>keepwarm</td>
<td>oic.r.time_period</td>
</tr>
<tr>
<td>Keypad</td>
<td>oic.r.keypadchar</td>
</tr>
<tr>
<td>liquidlevel</td>
<td>oic.r.liquid.level</td>
</tr>
<tr>
<td>liquidremaining</td>
<td>oic.r.liquid.level</td>
</tr>
<tr>
<td>lock</td>
<td>oic.r.lock</td>
</tr>
<tr>
<td>motionSensor</td>
<td>oic.r.sensor.motion</td>
</tr>
<tr>
<td>openlevel</td>
<td>oic.r.openlevel</td>
</tr>
<tr>
<td>operationmode</td>
<td>oic.r.switch.binary</td>
</tr>
<tr>
<td>overcurrentsensor</td>
<td>oic.r.time_period</td>
</tr>
<tr>
<td>powersave</td>
<td>oic.r.switch.binary</td>
</tr>
<tr>
<td>printqueue</td>
<td>oic.r.printer.queue</td>
</tr>
<tr>
<td>pushbutton</td>
<td>oic.r.button</td>
</tr>
<tr>
<td>refrigeration</td>
<td>oic.r.refrigeration</td>
</tr>
<tr>
<td>relativeHumidity</td>
<td>oic.r.humidity</td>
</tr>
<tr>
<td>robotcleanerjobmode</td>
<td>oic.r.operational.state</td>
</tr>
<tr>
<td>steamclosetjobmode</td>
<td>oic.r.operational.state</td>
</tr>
<tr>
<td>temperature</td>
<td>oic.r.temperature</td>
</tr>
<tr>
<td>uvsensor</td>
<td>oic.r.sensor.radiation.uv</td>
</tr>
<tr>
<td>watersensor</td>
<td>oic.r.sensor.water</td>
</tr>
<tr>
<td>weight</td>
<td>oic.r.weight</td>
</tr>
</tbody>
</table>

- Yellow highlights identify Resource Types that were added to support equivalency.
Ecosystem Bridges:
AllJoyn, oneM2M, ZigBee, BLE, Z-wave, U+, EnOcean
## OCF to Zigbee Device Type Mapping

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

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

- Yellow highlights identify Resource Types that were added to support equivalency.
Ecosystem Bridges:
AllJoyn, oneM2M, ZigBee, BLE, Z-wave, U+, EnOcean
## OCF to BLE Device Type Mapping

<table>
<thead>
<tr>
<th>BLE GATT-based Profile</th>
<th>OCF Device Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blood Pressure Profile</td>
<td>&quot;oic.d.bloodpressuremonitor&quot;</td>
</tr>
<tr>
<td>Glucose Profile</td>
<td>&quot;oic.d.glucosemeter&quot;</td>
</tr>
<tr>
<td>Health Thermometer Profile</td>
<td>&quot;oic.d.bodythermometer&quot;</td>
</tr>
<tr>
<td>Weight Scale Profile</td>
<td>&quot;oic.d.bodyscale&quot;</td>
</tr>
</tbody>
</table>
## OCF Resources to BLE Services

<table>
<thead>
<tr>
<th>BLE GATT-based Profile</th>
<th>BLE Service</th>
<th>Atomic Measurement Resource Type</th>
<th>Resource Type</th>
<th>OCF Device Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blood Pressure Profile</td>
<td>Blood Pressure Service</td>
<td>&quot;oic.r.bloodpressuremonitor-am&quot;</td>
<td>&quot;oic.r.blood.pressure&quot;</td>
<td>&quot;oic.d.bloodpressuremonitor&quot;</td>
</tr>
<tr>
<td></td>
<td>Device Information</td>
<td>&quot;oic.r.pulserate&quot;</td>
<td>&quot;oic.wk.d&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Service</td>
<td>&quot;oic.r.pulse.rate&quot;</td>
<td>&quot;oic.wk.p&quot;</td>
<td></td>
</tr>
<tr>
<td>Glucose Profile</td>
<td>Glucose Service</td>
<td>&quot;oic.r.glucosemeter-am&quot;</td>
<td>&quot;oic.r.glucose&quot;</td>
<td>&quot;oic.d.glucosemeter&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;oic.r.glucose.carb&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;oic.r.glucose.exercise&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;oic.r.glucose.hba1c&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;oic.r.glucose.health&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;oic.r.glucose.meal&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;oic.r.glucose.medication&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;oic.r.glucose.samplelocation&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;oic.r.glucose.tester&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device Information</td>
<td>&quot;oic.wk.d&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Service</td>
<td>&quot;oic.wk.p&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Health Thermometer</td>
<td>Health Thermometer</td>
<td>&quot;oic.r.bodythermometer-am&quot;</td>
<td>&quot;oic.r.temperature&quot;</td>
<td>&quot;oic.d.bodythermometer&quot;</td>
</tr>
<tr>
<td>Profile</td>
<td>Service</td>
<td></td>
<td>&quot;oic.r.body.location.temperature&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device Information</td>
<td>&quot;oic.wk.d&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Service</td>
<td>&quot;oic.wk.p&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight Scale Profile</td>
<td>Weight Scale Service</td>
<td>&quot;oic.r.bodyyscale-am&quot;</td>
<td>&quot;oic.r.weight&quot;</td>
<td>&quot;oic.d.bodyyscale&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;oic.r.bmi&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;oic.r.height&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;oic.r.body.fat&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;oic.r.body.water&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;oic.r.body.slim&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;oic.r.body.ffm&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device Information</td>
<td>&quot;oic.wk.d&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Service</td>
<td>&quot;oic.wk.p&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Ecosystem Bridges:
AllJoyn, oneM2M, ZigBee, BLE, Z-wave, U+, EnOcean
### OCF to Z-Wave Device Mapping

<table>
<thead>
<tr>
<th>Z-Wave Plus Device</th>
<th>OCF Device Type</th>
<th>OCF Device Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light Dimmer Switch</td>
<td>oic.d.light</td>
<td>Light</td>
</tr>
<tr>
<td>Door Lock – Keypad</td>
<td>oic.d.smartlock</td>
<td>Smart Lock</td>
</tr>
<tr>
<td>On/Off Power Switch</td>
<td>oic.d.switch</td>
<td>Switch</td>
</tr>
<tr>
<td>Sensor - Multilevel</td>
<td>oic.d.sensor</td>
<td>Generic Sensor</td>
</tr>
<tr>
<td>Sensor – Notification</td>
<td>oic.d.sensor</td>
<td>Generic Sensor</td>
</tr>
</tbody>
</table>
## OCF Resources to Z-Wave Command Class

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

AllJoyn, oneM2M, ZigBee, BLE, Z-wave, U+, EnOcean
<table>
<thead>
<tr>
<th>U+ Device</th>
<th>OCF Device Name</th>
<th>OCF Device Type (&quot;rt&quot;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air Conditioner</td>
<td>Air Conditioner</td>
<td>&quot;oic.d.airconditioner&quot;</td>
</tr>
<tr>
<td>Water Heater</td>
<td>Water Heater</td>
<td>&quot;oic.d.waterheater&quot;</td>
</tr>
<tr>
<td>Air Purifier</td>
<td>Air Purifier</td>
<td>&quot;oic.d.airpurifier&quot;</td>
</tr>
<tr>
<td>U+ Device</td>
<td>U+ Property</td>
<td>OCF Resource Type</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Air Conditioner</td>
<td>&quot;onOffStatus&quot;</td>
<td>&quot;oic.r.switch.binary&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;targetTemperature&quot;</td>
<td>&quot;oic.r.temperature&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;windSpeed&quot;</td>
<td>&quot;oic.r.selectablelevels&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;operationMode&quot;</td>
<td>&quot;oic.r.mode&quot;</td>
</tr>
<tr>
<td>Water Heater</td>
<td>&quot;onOffStatus&quot;</td>
<td>oic.r.switch.binary</td>
</tr>
<tr>
<td></td>
<td>&quot;targetTemperature&quot;</td>
<td>&quot;oic.r.temperature&quot;</td>
</tr>
<tr>
<td>Air Purifier</td>
<td>&quot;onOffStatus&quot;</td>
<td>&quot;oic.r.switch.binary&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;mode&quot;</td>
<td>&quot;oic.r.operational.state&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;windSpeed&quot;</td>
<td>&quot;oic.r.selectablelevels&quot;</td>
</tr>
</tbody>
</table>
Ecosystem Bridges:
AllJoyn, oneM2M, ZigBee, BLE, Z-wave, U+, EnOcean
<table>
<thead>
<tr>
<th>Device Name</th>
<th>OCF Device Type</th>
<th>OCF Device Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push Button (F6-01-01)</td>
<td>oic.d.sensor</td>
<td>Generic Sensor</td>
</tr>
<tr>
<td>Rocker Switch, 2 Rocker (F6-02-XX)</td>
<td>oic.d.sensor</td>
<td>Generic Sensor</td>
</tr>
<tr>
<td>Rocker Switch, 4 Rocker (F6-03-XX)</td>
<td>oic.d.sensor</td>
<td>Generic Sensor</td>
</tr>
<tr>
<td>Position Switch (F6-04-01)</td>
<td>oic.d.sensor</td>
<td>Generic Sensor</td>
</tr>
<tr>
<td>Position Switch (F6-04-02)</td>
<td>oic.d.sensor</td>
<td>Generic Sensor</td>
</tr>
<tr>
<td>Liquid Leakage Detector (Water) (F6-05-01)</td>
<td>oic.d.sensor</td>
<td>Generic Sensor</td>
</tr>
<tr>
<td>Smoke Detector (F6-05-02)</td>
<td>oic.d.sensor</td>
<td>Generic Sensor</td>
</tr>
<tr>
<td>Single Input Contact (D5-00-01)</td>
<td>oic.d.sensor</td>
<td>Generic Sensor</td>
</tr>
<tr>
<td>Temperature Sensor (A5-02-XX)</td>
<td>oic.d.sensor</td>
<td>Generic Sensor</td>
</tr>
<tr>
<td>Temperature and Humidity Sensor (A5-04-XX)</td>
<td>oic.d.sensor</td>
<td>Generic Sensor</td>
</tr>
<tr>
<td>Barometric Sensor (A5-05-01)</td>
<td>oic.d.sensor</td>
<td>Generic Sensor</td>
</tr>
<tr>
<td>Light Sensor (A5-06-XX)</td>
<td>oic.d.sensor</td>
<td>Generic Sensor</td>
</tr>
<tr>
<td>Occupancy Sensor (A5-07-XX)</td>
<td>oic.d.sensor</td>
<td>Generic Sensor</td>
</tr>
<tr>
<td>Light, Temperature and Occupancy Sensor (A5-08-XX)</td>
<td>oic.d.sensor</td>
<td>Generic Sensor</td>
</tr>
<tr>
<td>EnOcean Device Name (EEP)</td>
<td>EnOcean Telegram Parameters</td>
<td>OCF Resource Type(s)</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>----------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Push Button (F6-01-01)</td>
<td>Push Button Released</td>
<td>oic.r.button</td>
</tr>
<tr>
<td></td>
<td>Push Button Pressed</td>
<td>oic.r.button</td>
</tr>
<tr>
<td>Rocker Switch, 2 Rocker (F6-02-XX)</td>
<td>Rocker 1st Action AI</td>
<td>oic.r.button</td>
</tr>
<tr>
<td></td>
<td>Rocker 1st Action AO</td>
<td>oic.r.button</td>
</tr>
<tr>
<td></td>
<td>Rocker 1st Action BI</td>
<td>oic.r.button</td>
</tr>
<tr>
<td></td>
<td>Rocker 1st Action BO</td>
<td>oic.r.button</td>
</tr>
<tr>
<td>Rocker Switch, 4 Rocker (F6-03-XX)</td>
<td>Rocker 1st Action AI</td>
<td>oic.r.button</td>
</tr>
<tr>
<td></td>
<td>Rocker 1st Action AO</td>
<td>oic.r.button</td>
</tr>
<tr>
<td></td>
<td>Rocker 1st Action BI</td>
<td>oic.r.button</td>
</tr>
<tr>
<td></td>
<td>Rocker 1st Action BO</td>
<td>oic.r.button</td>
</tr>
<tr>
<td></td>
<td>Rocker 1st Action CI</td>
<td>oic.r.button</td>
</tr>
<tr>
<td></td>
<td>Rocker 1st Action CO</td>
<td>oic.r.button</td>
</tr>
<tr>
<td></td>
<td>Rocker 1st Action DI</td>
<td>oic.r.button</td>
</tr>
<tr>
<td></td>
<td>Rocker 1st Action DO</td>
<td>oic.r.button</td>
</tr>
<tr>
<td>Position Switch (F6-04-01)</td>
<td>Key Card activated</td>
<td>oic.r.keycardswitch</td>
</tr>
<tr>
<td></td>
<td>Key Card taken out</td>
<td></td>
</tr>
<tr>
<td>Position Switch (F6-04-02)</td>
<td>Key Card inserted</td>
<td>oic.r.keycardswitch</td>
</tr>
<tr>
<td></td>
<td>Key Card taken out</td>
<td></td>
</tr>
<tr>
<td>Liquid Leakage Detector (Water) (F6-05-01)</td>
<td>Alert Signal</td>
<td>oic.r.sensor.water</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smoke Detector (F6-05-02)</td>
<td>Smoke Alarm ON</td>
<td>oic.r.sensor.smoke</td>
</tr>
<tr>
<td></td>
<td>Smoke Alarm OFF</td>
<td></td>
</tr>
<tr>
<td>Single Input Contact (D5-00-01)</td>
<td>Open</td>
<td>oic.r.sensor.contact</td>
</tr>
<tr>
<td></td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>Temperature Sensor (A5-02-XX)</td>
<td>Temperature value</td>
<td>oic.r.temperature</td>
</tr>
<tr>
<td></td>
<td>Unit (defined by spec)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Range (by type spec)</td>
<td></td>
</tr>
<tr>
<td>Temperature and Humidity Sensor (A5-04-XX)</td>
<td>Temperature value</td>
<td>oic.r.temperature</td>
</tr>
<tr>
<td></td>
<td>Temperature unit (by spec)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Temperature range (by type spec)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Humidity (%)</td>
<td></td>
</tr>
<tr>
<td>Barometric Sensor (A5-05-01)</td>
<td>Barometer value</td>
<td>oic.r.sensor.atmosphericpressure</td>
</tr>
<tr>
<td>Light Sensor (A5-06-XX)</td>
<td>Illumination value (linear, lx)</td>
<td>oic.r.sensor.illuminance</td>
</tr>
<tr>
<td></td>
<td>range (by type Spec)</td>
<td></td>
</tr>
<tr>
<td>Occupancy Sensor (A5-07-XX)</td>
<td>PIR Status Uncertain</td>
<td>oic.r.sensor.presence</td>
</tr>
<tr>
<td></td>
<td>PIR Status Motion detected</td>
<td></td>
</tr>
<tr>
<td>Light, Temperature and Occupancy Sensor</td>
<td>Temperature value</td>
<td>oic.r.temperature</td>
</tr>
<tr>
<td></td>
<td>Temp Unit (by spec)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Temp Range (by TYPE spec)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Illumination value</td>
<td>oic.r.sensor.illuminance</td>
</tr>
<tr>
<td></td>
<td>Temperature value</td>
<td>oic.r.temperature</td>
</tr>
<tr>
<td></td>
<td>range (by type Spec)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Occupancy</td>
<td>oic.r.sensor.presence</td>
</tr>
</tbody>
</table>
References
• 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
      • ZWave Derived Models: https://github.com/openconnectivityfoundation/OCF-ZWave
      • 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
    • 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
OCF Specification Overview
Device and Resource Modeling

OCF 2.2.1 Release
December 2020
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 2.2.1; 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
  • Captures per method the JSON schema defining the supported payload
  • Detailed list of the Property(-ies) the resource exposes

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 2.2.1

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>Use Case Category</th>
<th>Use Case Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Civic Location</td>
<td>Location</td>
<td>Properties associated with providing a Civic Location as defined by RFC RFC 4119 and clarified by RFC 5774</td>
</tr>
<tr>
<td>Colour Rendering Index</td>
<td>Smart Devices</td>
<td>Properties associated with the colour rendering index of a device</td>
</tr>
<tr>
<td>Generic Measurement Sensor</td>
<td>IoT Sensors</td>
<td>Properties associated with sensed measurement</td>
</tr>
<tr>
<td>Sound Pressure</td>
<td>IoT Sensors</td>
<td>Properties associated with providing sound pressure in Pascal</td>
</tr>
<tr>
<td>Sound Pressure Level</td>
<td>IoT Sensors</td>
<td>Properties associated with providing sound pressure level (dB)</td>
</tr>
<tr>
<td>UVA Radiation</td>
<td>IoT Sensors</td>
<td>Properties associated with the measurement of UVA Radiation</td>
</tr>
<tr>
<td>UVB Radiation</td>
<td>IoT Sensors</td>
<td>Properties associated with the measurement of UVB Radiation</td>
</tr>
</tbody>
</table>
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.


<table>
<thead>
<tr>
<th>Device Category Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LBNL Categories</strong></td>
<td></td>
</tr>
<tr>
<td>Space Conditioning</td>
<td>Heating and cooling systems</td>
</tr>
<tr>
<td>Lighting</td>
<td></td>
</tr>
<tr>
<td>Appliance</td>
<td>Also known as “white goods”; covers major appliances only.</td>
</tr>
<tr>
<td>Electronics</td>
<td>Personal electronics</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>Small appliances, other</td>
</tr>
<tr>
<td>Infrastructure</td>
<td>Physical building and infrastructure</td>
</tr>
<tr>
<td>Transportation</td>
<td>Vehicles, fixed devices that provide movement (e.g. Escalators)</td>
</tr>
<tr>
<td><strong>OCF Added Categories</strong></td>
<td></td>
</tr>
<tr>
<td>Fitness</td>
<td>Includes lifestyle</td>
</tr>
<tr>
<td>Medical</td>
<td></td>
</tr>
<tr>
<td>Personal Health</td>
<td></td>
</tr>
</tbody>
</table>
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](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

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

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)

<table>
<thead>
<tr>
<th>Device Name</th>
<th>Device Type</th>
<th>Resource Type</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light</td>
<td>oic.d.light</td>
<td>oic/res (oic.wk.res)</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>oic/p (oic.wk.p)</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>oic/d (oic.d.light)</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Introspection (oic.wk.introspection)</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Binary switch (oic.r.switch.binary)</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Brightness (oic.r.light.brightness)</td>
<td>O</td>
</tr>
</tbody>
</table>
### Complete Set of OCF Defined Device Types (1/3)

<table>
<thead>
<tr>
<th>Friendly Name</th>
<th>Device Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D Printer</td>
<td>oic.d.3dprinter</td>
</tr>
<tr>
<td>AFCI</td>
<td>oic.d.afci</td>
</tr>
<tr>
<td>Activity Tracker</td>
<td>oic.d.activitytracker</td>
</tr>
<tr>
<td>Air Conditioner</td>
<td>oic.d.airconditioner</td>
</tr>
<tr>
<td>Air Purifier</td>
<td>oic.d.airpurifier</td>
</tr>
<tr>
<td>Air Quality Monitor</td>
<td>oic.d.airqualitymonitor</td>
</tr>
<tr>
<td>Airer</td>
<td>oic.d.airer</td>
</tr>
<tr>
<td>Battery</td>
<td>oic.d.battery</td>
</tr>
<tr>
<td>Blind</td>
<td>oic.d.blind</td>
</tr>
<tr>
<td>Blood Pressure Monitor</td>
<td>oic.d.bloodpressurermonitor</td>
</tr>
<tr>
<td>Body Composition</td>
<td>oic.d.bodycomposition</td>
</tr>
<tr>
<td>Body Scale</td>
<td>oic.d.bodyscale</td>
</tr>
<tr>
<td>Body Thermometer</td>
<td>oic.d.bodythermometer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Friendly Name</th>
<th>Device Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera</td>
<td>oic.d.camera</td>
</tr>
<tr>
<td>Circuit Breaker</td>
<td>oic.d.circuitbreaker</td>
</tr>
<tr>
<td>Clothes Dryer</td>
<td>oic.d.dryer</td>
</tr>
<tr>
<td>Clothes Washer</td>
<td>oic.d.washer</td>
</tr>
<tr>
<td>Clothes Washer/Dryer</td>
<td>oic.d.washerdryer</td>
</tr>
<tr>
<td>Coffee Machine</td>
<td>oic.d.coffeemachine</td>
</tr>
<tr>
<td>Continuous Glucose Meter</td>
<td>oic.d.cgm</td>
</tr>
<tr>
<td>Cooker Hood</td>
<td>oic.d.cookerhood</td>
</tr>
<tr>
<td>Cooktop</td>
<td>oic.d.cooktop</td>
</tr>
<tr>
<td>Cycling Cadence Sensor</td>
<td>oic.d.cyclingcadencesensor</td>
</tr>
<tr>
<td>Cycling Power Meter</td>
<td>oic.d.cyclingpowermeter</td>
</tr>
<tr>
<td>Cycling Speed Sensor</td>
<td>oic.d.cyclingspeedsensor</td>
</tr>
<tr>
<td>Dehumidifier</td>
<td>oic.d.dehumidifier</td>
</tr>
<tr>
<td>Dishwasher</td>
<td>oic.d.dishwasher</td>
</tr>
</tbody>
</table>

Items in red are new in OCF 2.1.x and OCF 2.2.x
<table>
<thead>
<tr>
<th>Friendly Name</th>
<th>Device Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Door</td>
<td>oic.d.door</td>
</tr>
<tr>
<td>Electric Meter</td>
<td>oic.d.electricmeter</td>
</tr>
<tr>
<td>Electric Vehicle</td>
<td>oic.d.electricvehicle</td>
</tr>
<tr>
<td>Charger</td>
<td></td>
</tr>
<tr>
<td>Energy Generator</td>
<td>oic.d.energygenerator</td>
</tr>
<tr>
<td>Energy Monitor</td>
<td>oic.d.energymonitor</td>
</tr>
<tr>
<td>Fan</td>
<td>oic.d.fan</td>
</tr>
<tr>
<td>Food Probe</td>
<td>oic.d.foodprobe</td>
</tr>
<tr>
<td>Freezer</td>
<td>oic.d.freezer</td>
</tr>
<tr>
<td>Garage Door</td>
<td>oic.d.garagedoor</td>
</tr>
<tr>
<td>Generic Sensor</td>
<td>oic.d.sensor</td>
</tr>
<tr>
<td>Glucose Meter</td>
<td>oic.d.glucosemeter</td>
</tr>
<tr>
<td>Grinder</td>
<td>oic.d.grinder</td>
</tr>
<tr>
<td>GFCI</td>
<td>oic.d.gfci</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Friendly Name</th>
<th>Device Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heart Rate Monitor</td>
<td>oic.d.heartratemonitor</td>
</tr>
<tr>
<td>Humidifier</td>
<td>oic.d.humidifier</td>
</tr>
<tr>
<td>Humidifier</td>
<td>oic.d.humidifier</td>
</tr>
<tr>
<td>Indoor Garden</td>
<td>oic.d.indoorgarden</td>
</tr>
<tr>
<td>Inverter</td>
<td>oic.d.inverter</td>
</tr>
<tr>
<td>Kettle</td>
<td>oic.d.kettle</td>
</tr>
<tr>
<td>Light</td>
<td>oic.d.light</td>
</tr>
<tr>
<td>Mattress</td>
<td>oic.d.mattress</td>
</tr>
<tr>
<td>Microwave Oven</td>
<td>oic.d.microwave</td>
</tr>
<tr>
<td>Muscle Oxygen Monitor</td>
<td>oic.d.muscleoxyg enmonitor</td>
</tr>
<tr>
<td>Optical Augmented RFID Reader</td>
<td>oic.d.orfid</td>
</tr>
<tr>
<td>Oven</td>
<td>oic.d.oven</td>
</tr>
<tr>
<td>PV Array System</td>
<td>oic.d.pvarraysystem</td>
</tr>
<tr>
<td>Printer</td>
<td>oic.d.printer</td>
</tr>
<tr>
<td>Printer (Multi-Function)</td>
<td>oic.d.multifunctionprinter</td>
</tr>
<tr>
<td>Robot Cleaner</td>
<td>oic.d.robotcleaner</td>
</tr>
<tr>
<td>Scanner</td>
<td>oic.d.scanner</td>
</tr>
<tr>
<td>Security Panel</td>
<td>oic.d.securitypanel</td>
</tr>
<tr>
<td>Set Top Box</td>
<td>oic.d.stb</td>
</tr>
<tr>
<td>Sleep Monitor</td>
<td>oic.d.sleepmonitor</td>
</tr>
<tr>
<td>Smart Lock</td>
<td>oic.d.lock</td>
</tr>
<tr>
<td>Smart Plug</td>
<td>oic.d.smartplug</td>
</tr>
<tr>
<td>Speaker</td>
<td>oic.d.speaker</td>
</tr>
<tr>
<td>Steam Closet</td>
<td>oic.d.steamcloset</td>
</tr>
<tr>
<td>Switch</td>
<td>oic.d.switch</td>
</tr>
<tr>
<td>Television</td>
<td>oic.d.tv</td>
</tr>
<tr>
<td>Thermostat</td>
<td>oic.d.thermostat</td>
</tr>
<tr>
<td>Water Heater</td>
<td>oic.d.waterheater</td>
</tr>
</tbody>
</table>

February 2021

Copyright © 2021 Open Connectivity Foundation, Inc. All Rights Reserved.
### Complete Set of OCF Defined Device Types (3/3)

<table>
<thead>
<tr>
<th>Friendly Name</th>
<th>Device Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pulse Oximeter</td>
<td>oic.d.pulseoximeter</td>
</tr>
<tr>
<td>Receiver</td>
<td>oic.d.receiver</td>
</tr>
<tr>
<td>Refrigerator</td>
<td>oic.d.refrigerator</td>
</tr>
<tr>
<td>Water Purifier</td>
<td>oic.d.waterpurifier</td>
</tr>
<tr>
<td>Water Valve</td>
<td>oic.dwatervalve</td>
</tr>
<tr>
<td>Window</td>
<td>oic.d.window</td>
</tr>
</tbody>
</table>
Thank you!

• Access the OCF specifications
  https://openconnectivity.org/resources/specifications
• Contact OCF at admin@openconnectivity.org