OCF Specification Introduction and Overview

May 2019
Table of Contents

• Technical Principles for an Internet of Things Ecosystem
• Introduction to the Open Connectivity Foundation
• OCF Specification Overview
  - Core Framework
  - Core Extension: OCF Cloud
  - Core Extension: Wi-Fi Easy Setup
  - Security
  - Bridging
  - OCF to AllJoyn Mapping
  - OCF to OneM2M Mapping
  - Resource Type
  - Device Profile
Technical Principles for an Internet of Things Ecosystem
Scope of IoT

Vertical Profiles
- Smart Home
- Industrial
- Healthcare
- ... (Ellipses indicate additional profiles)

Baseline Functionality
- Group management
- ID & Addressing
- Protocol Bridge/GW
- Common Resource Model
- CRUDN
- Device management
- Discovery
- Messaging
- Streaming
- Security

Connectivity
- Wi-Fi
- BT/BLE
- Thread
- ... (Ellipses indicate additional connectivity options)

Diagram:
- Local Control
- Remote Control
- Server to Server

Controller
Cloud Interface
Controller App
Cloud Servers

May 14, 2019
Copyright © 2019 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)
  - Resources
    - properties

• By defining functions/operations of things
  - SetSwitch
    - Power(bool)
  - SetDimmingLevel
    - dimmingSetting (int)
  - SetBrightness
    - brightness (int)
  - Functions
    - Input & Output Parameters

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

- (Verbs + Objects)
  - RPC model
Support of Constrained Things
Class 2 Devices as Defined by RFC 7228

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

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

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

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

- Conformance test - Each device proves conformance to specifications

- Certification Scope

Open Source

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

Optional Open Source Features

Tested Optional Spec Features

Optional Spec Features

Certificate Issue & Logo Licensing
Licensing

• For Intellectual Property Rights (IPR) Policy: RAND-Z > RAND >> no IPR policy
• For Open Source: Apache 2.0 > Internet Systems Consortium (ISC)

• Due to the common nature of IoT connecting everything over the Internet, it’s most critical for manufacturers to avoid a licensing risk
  - Everything connected could be at potential risk

• Offering manufacturer-friendly Licensing and IPR Policy enables growth of market by attracting both start-ups and large enterprises; such an IPR policy must be clear and readily understandable ensuring that the terms are offered by all IP holders.
Introduction to the
Open Connectivity Foundation
Introduction to OCF - Optimized for IoT

- Common Platform
- CoAP for Constrained Devices
- Best In Class Security
- Certification Program
- RESTful Architecture

Copyright © 2019 Open Connectivity Foundation, Inc. All Rights Reserved.
OCF Areas of Technology Development

- Core Architecture
  - Fundamental resource framework
- Discovery
- CRUDN
- Transport Binds
- Security
- Resource Models (vertical agnostic)
- Device Profiles
  - Smart Home
  - Health
  - Automotive
- Ecosystem Bridging
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
OCF Specification Overview
OCF Deliverables

Normative Specifications
• See next slide

Resource Models via oneIoTa
• Domain agnostic resources
• Derived models for Ecosystem Mapping
  • To date: OCF-AllJoyn (CDM 16.4)

Certification Procedures
• Test Policy (Certification Procedure Requirements Document)
• Test Plans and Test Cases (Certification Test Requirements Document)
Specification Structure

Infrastructure

- Core Framework
- Security
- Bridging
- Device Specification

Resource Model

- Resource Specification (reflects OneIoTa content)
- OCF Resource to AllJoyn Interface Mapping Specification (reflects OneIoTa content)
Where can I find the specifications and Resource Type definitions?

**OCF Specifications:**
- [https://openconnectivity.org/developer/specifications](https://openconnectivity.org/developer/specifications)

**Resource Type Definitions**
- Core Resources: [https://github.com/openconnectivityfoundation/core](https://github.com/openconnectivityfoundation/core)
- Core Extension Resources: [https://github.com/openconnectivityfoundation/core-extensions](https://github.com/openconnectivityfoundation/core-extensions)
- Bridging Resources: [https://github.com/openconnectivityfoundation/bridging](https://github.com/openconnectivityfoundation/bridging)
- Security Resources: [https://github.com/openconnectivityfoundation/security-models](https://github.com/openconnectivityfoundation/security-models)
- Vertical Resources and Derived Models: [https://oneiota.org/documents?filter%5Bmedia_type%5D=application%2Framl%2Byaml](https://oneiota.org/documents?filter%5Bmedia_type%5D=application%2Framl%2Byaml)
Infrastructure:
Core Framework Specification
OCF 2.0.2 Release Overview
Core Framework Topics Outline (1 of 2)

- Objectives
- RESTful Architecture
- OCF Roles
- Resources
- Basic Operations
- Organization of an OCF Device
- OCF Specification Features
- Protocol Stack
- Device Example
- Endpoint Overview
Core Framework Topics Outline (2 of 2)

- Resource Discovery (CoAP Discovery)
- Block Transfer with CoAP Messaging
- Encoding Schemes
- Defining OCF Components
- Vendor Extensions
- Introspection
- Collection Resources
- Atomic Measurement Resources
- Versioning
- Resource Discovery (Resource Directory)
Core Framework Objectives

- Core Framework Specification 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 open source releases
RESTful Architecture

- **Resource (representation)**

- **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

---

Client

Server

**Request**

**Response**

**CRUD & N operation**

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

Entity handler

Physical entity
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 describes an aspect that is exposed through a Resource including meta-information related to that Resource.
- Each Resource contains Interface(s) that provides first a view into the Resource and then defines the requests and responses permissible on that view of the Resource.
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, etc.)
Organization of an OCF Device

- OCF Device concept

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.
Protocol Stack

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

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

- Example overview
  - Smart light device with i) binary switch & ii) brightness resource
- Device type: Light device (oic.d.light) [Defined by the domain]
- Associated resources
  - Mandatory Core resources: oic/res, oic/p, oic/d
  - Mandatory Security Resources (not shown in the diagram)
  - Device specific resources: 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 Title</th>
<th>Device Type</th>
<th>Associated 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>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>

Example: Smart light device
Definition

An (OCF) Endpoint is defined as the source or destination of a request and response messages for a given Transport Protocol Suites (e.g. CoAP over UDP over IPv6). The specific definition of an Endpoint depends on the Transport Protocol Suites being used.

- an address (e.g. IPv6 address + Port number) or an indirect identifier (e.g., DNS name) resolvable to an address.
- (e.g.) For CoAP/UDP/IPv6, Endpoint is identified as IP address + port number or DNS name.

Endpoint characteristics for OCF Device

- Each OCF Device shall associate with at least one Endpoint with which it can exchange Request & Response messages.
  - When a message is sent to an Endpoint, it shall be delivered to the OCF Device which is associated with the Endpoint. When a Request message is delivered to an Endpoint, path component is enough to locate the target Resource.

- OCF Device can be associated with multiple Endpoints.
  - E.g. OCF Device may support both CoAP & HTTP

- An endpoint can be shared among multiple OCF Devices, only when there is a way to clearly indicate the target Resource with Request URI.
Endpoint information in /oic/res with “eps” Parameter

```
[/oic/res

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

- OCF devices make use of CoAP Discovery using IANA defined OCF Service Address (not the default CoAP address).
- Multicast RETRIEVE (CoAP GET) sent to well known URI /oic/res
- Response is an array of links; each link represents a Resource hosted by the responding server
- Links provide:
  - href
  - Relationship (self link, hosted link, bridged link)
  - Endpoint binds
  - Supported interfaces
  - Observability of the Resource
• 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
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.

---

**Encoding Schemes - CBOR**

- **Description**
  - Concise binary object representation based on JSON data model
  - Lightweight, text-based, language-independent data interchange format
  - Binary compression standard for XML

- **Standard**
  - IETF RFC 7049
  - IETF RFC 7159
  - W3C Efficient XML Interchange Format 1.0

- **ContentType**
  - /application/vnd.ocf+cbor
  - /application/json
  - /application/exi

- **OCFM/O**
  - Mandatory
  - Can be supported

---

If needed in future revisions

May 14, 2019

Copyright © 2019 Open Connectivity Foundation, Inc. All Rights Reserved.
Defining OCF Components (on top of CORE)

- **OCF Servers**
  - Defined by device identifier: standardized name of the device
  - List of mandatory OCF Resource Types per device
  - Note that OCF Clients are implicitly specified as "opposite" side of an OCF Server.
    - Currently OCF does not impose interaction sequences.
    - All instances of a Resource Type are allowed to talk to/from any OCF Client at any point in time

- **OCF Resource Type**
  - Defined by resource identifier: standardized name of the resource
  - List of mandatory properties per Resource Type
  - List of allowed actions (read/readwrite/..) per Resource Type
  - All OCF Resource Type IDs are IANA registered: http://www.iana.org/assignments/core-parameters/core-parameters.xhtml
Vendor Extensions

• 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
Introspection

• Why
  • On par with existing AllJoyn framework

• 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.
    • Describe the payload on JSON level
      – Property names
      – Type
      – range
Introspection: Underlying rationale

- Use OpenAPI 2.0 files it as input for the OpenAPI 2.0 definition that will go on the wire.
- Same restrictions as already investigated and part of the:
  - 1 file to be transferred: e.g. definition includes
    - All end points, methods, query parameters, payload definitions
  - Same kind of negotiation to download the file
Collection Resources (Optional)

• 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.

• 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 (Optional)

- 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).
Alerts Resources (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
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 version
- **Method**: icv (spec version), dmv (data model version) in /oic/d resource
## Content-Formats

<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>

## Option Numbers

<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>

## 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></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td></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 1000 0000</td>
<td>2112</td>
</tr>
</tbody>
</table>
Payload Versioning Use Case & Policies

Round 1

OCF 1.0 Client
GET /oic/res with application/vnd.ocf+cbor
Return an error

OIC 1.1 Server

Round 2

OCF 1.0 Client
GET /oic/res with application/vnd.ocf+cbor
Return /oic/res per OIC 1.1

OIC 1.1 Server

GET /oic/res with application/cbor
/oic/res response per OIC 1.1

OCF 1.0 Server

GET /oic/res with application/cbor

OIC 1.1 Server

GET /oic/res with application/vnd.ocf+cbor
/oic/res response per OCF 1.0

OCF 1.0 Server
Resource Discovery (Resource Directory) (Optional)

- OCF Devices may use Resource Directory to find the Resources hosted in the 3rd party Devices.
  - Publishing Devices register the Resources (i.e. Links) to a Resource Directory, to which a Client subsequently makes an inquiry to discover those Resources.

- Resource Directory
  - An OCF Device facilitating indirect discovery by exposing 3rd party Resources (i.e. Links) with the following features
    - RD discovery
      - Discover an RD and select one with oic.wk.rd
    - Resource publish
      - publish/update/delete Resource (i.e. Links) in /oic/res of an RD
    - Resource exposure
      - Expose published Resources via /oic/res of RD, which is aligned with CoAP discovery.
Use Case

• Remote Control/Manage OCF devices based on user authentication
  • User can access OCF devices which belong or are shared with him/her regardless of a location.
  • User can receive cloud providing service through the registered devices
    • EX) Device Management, Home security, Energy management and etc

• Usage & Operational Scenario
  1. Jane creates an account in the cloud
     - It is able to use the existing third-party account such as Facebook.
  2. Jane registers device resources under created account,
  3. She can control the device anywhere out of house
OCF Cloud operational Flow

1. Create user account
2. Mediator Registration
3. Cloud configuration (Cloud IP, Access token)
4. TLS session
5. Device Registration. Resource Publish (Access token, device id)
6. Validate Token

Token | di | userid
-------|----|-------
A0001  | 0xA71CE | u0001

Share User Authentication (Access token, user id)

[OCF Cloud]

[OCF Resource Server]

[Mediator]

[Note: The dotted interface is out of OCF scope]
1. The OCF Cloud 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 OCF Cloud (e.g. the Mediator may be an App from the Cloud Provider).

3. The OCF Cloud User has access credentials for the Cloud (i.e. user name/password or similar)
   - User can use his 3rd party user account
1. The Mediator provides this Access Token to the OCF Cloud.

2. The OCF Cloud 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 User. This "uid" is the same for all Mediator instances that may be associated with the OCF Cloud User.

3. This same user ID can be used to assign multiple Devices to the same OCF Cloud User.
The Device is configured by the OBT by adding the required ACEs and creds to give the Mediator access to the CoAPCloudConf (CCC) Resource.

The Mediator connects to the Device through normal OCF Discovery processes.

The Mediator updates the CCC Resource on the Device with the Access Token ("at") and OCF Cloud URI ("cis"). The Mediator may also provide the Auth Provider Name ("apn").
1. The Device establishes a TLS connection with the OCF Cloud using the properties in CCC resource.

2. The Device sends an UPDATE request to the Account Resource on the OCF Cloud which includes the following Properties: "di","accesstoken", "authprovider"

3. The OCF Cloud ensures that the "di" and the "accesstoken" match its current values. The "accesstoken" value is the same one that the OCF Cloud or Auth provider provided to the Mediator

4. If the values match, the OCF Cloud sends the Account Resource Properties in the UPDATE response

5. If the Device sends a RETRIEVE request to any of the OCF Cloud Resources, the OCF Cloud responds with an appropriate error code.
V. Login with the OCF Cloud

1. In order to establish a TLS session and connect to the OCF Cloud 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 verifies that the values in the UPDATE request are correct and if so, the OCF Cloud sends a response message that includes the remaining session time ("expiresin").

3. The Device now has an active TCP connection and can exchange data.
VI. Publishing Links to the OCF Cloud RD

1. Once the TLS connection has been established to the OCF Cloud the Device publishes its Resources to the RD function of the OCF Cloud so that they can be seen/accessed remotely.

2. The acl2 and cred Resource of the OCF Device have to be provisioned by the OBT/AMS/CMS/DOTS to give the OCF Cloud the required CRUDN permissions.
VII. Client to Server communication through the OCF Cloud

1. Clients must go through this same process and register with the OCF Cloud. All of an OCF Cloud User's Devices (Clients and Servers) will be assigned the access control right associated with the User ID.

2. The OCF Cloud allows communication between all of a OCF Cloud 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, the OCF Cloud forwards those requests to the Device which responds to the OCF Cloud which then gets returned to the Client (i.e. Client -> OCF Cloud -> Device -> OCF Cloud -> Client).
When (or before) the "expiresin" timer expires, the Device should refresh its token by sending an UPDATE request to the Token Refresh Resource that includes:

1. "di"
2. "uid"
3. "refreshtoken"

The OCF Cloud responds with a new

1. "accesstoken"
2. "refreshtoken"
3. "expiresin"
IX. Closing connection with the OCF Cloud

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

X. Deregistering with the OCF Cloud

- The Device sends a DELETE request message to the Account Resource which includes: "accesstoken", "di"
- The OCF Cloud sends a response message confirming that the Device has been deregistered.
- To connect to the OCF Cloud again, the Device has to be provisioned by the Mediator again and then reregister with the OCF Cloud
Wi-Fi Easy Setup
OCF 2.0.2 Overview
Overview

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

- **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.
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

**Resources**

- EasySetup Resource (Collection)
- WiFiConf Resource
- DevConf Resource
- CoAPCloudConf Resource (optional)

**Properties**

- Provisioning status
- Last Error Code
- Connect
- Wi-Fi AP information
- Device Name (read only)
- Access Token
- Provider
- CI Server URL
### Resource Model: Easy Setup

- Indicates easy setup status

#### Resource Name: EasySetup

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

#### Properties

<table>
<thead>
<tr>
<th>Property Name(key)</th>
<th>Value Type</th>
<th>Access Mode</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy Setup Provisioning Status</td>
<td>integer</td>
<td>enum</td>
<td>R</td>
<td>Yes</td>
</tr>
<tr>
<td>Last Error Code</td>
<td>integer</td>
<td>enum</td>
<td>R</td>
<td>Yes</td>
</tr>
<tr>
<td>Connect</td>
<td>array of integer</td>
<td>RW</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

- **Easy Setup Provisioning Status**
  - Indicates the easy setup provisioning status of the device
  - (0: Need to Setup, 1: Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254: Reserved, 255: EOF)

- **Last Error Code**
  - Indicates a failure reason if it fails to connect to Enroller
  - (0: NO error, 1: A given SSID is not found, 2: Wi-Fi password is wrong, 3: IP address is not allocated, 4: NO internet connection, 5: Timeout, 6: Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not supported by the Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi Encryption Type is wrong (failure while connecting to the Enroller), 13~254: Reserved, 255: Unknown error)

- **Connect**
  - Indicates an array of connection types that trigger an attempt to connect to the Enroller to start
  - (1: Wi-Fi, 2: Other transport to be added in a future (e.g. BLE))
### Easy Setup - Wi-Fi Conf. Resource

- Contains Wi-Fi-related properties

<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”, “WPA2_PS”)</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>

---

**Example URI**

/example/WiFiConfResURI

**Resource Type**

oic.r.wificonf

**CRUDN permission**

RU
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>CRUD 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>
**Example Easy Setup Flow (informative)**

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
OCF Specification Overview
Security Specification
OCF 2.0.2 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)
  - **Available** (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)** creates and verifies access control permissions.
  - **Credential Management Service (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)** 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:

![Diagram showing the security layers and resource access]

Server – Responding Device

Application Resources

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

Security Layer

acl(s)
- Subject
- Resource(s)
- Permission

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

Allow /Deny Resource Access
Simplified Onboarding Sequence

• Unowned Device boots

• Discovery (unsecured):
  • DOTS sends multicast to discover unowned devices no TLS
  • Unowned devices reply, including list of supported OTMs no TLS

• Ownership Transfer:
  • DOTS selects and configures this OTM to the new device no TLS
  • DOTS & unowned Device perform OTM, inc. TLS handshake TLS
  • DOTS config SVRs to authorize itself, CMS and AMS TLS
  • Device is now owned!

• Provisioning:
  • CMS provisions credentials, AMS provisions access policies TLS
  • Device is now provisioned and can commence normal operation

• Normal Operation: TLS or no TLS
  • Credentials and/or access policies can be updated by returning to Provisioning
Device Provisioning States

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

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

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

- OCF devices can support the use of both symmetric and asymmetric credentials for establishing secure communication
  - Symmetric Key is mandatory
  - Certificates public/private keys are supported
  - CMS must be able to support both
- Missing credentials could be procured from a CMS
- Credentials may have an expiration period
  - Expired credentials can be refreshed
Access Control

Is Light On?

Turn Light Off

Request
Accept

Request
Reject

acl[0]

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

acl[0]

subject: DeviceID
rsrc: [/a/light]
pms: R
Access Control

- Protect Resources of the OCF Server to control CRUDN access for entity requesting access
  - Any request to the OCF Server is subject to ACL (Access Control List) policy check
  - ACE (Access Control Entry) policy applies to a OCF Server hosted Resource
  - Each ACE has a permission which allows read or write operation
- Two type of access control mechanism are supported:
  - Subject-based access control (SBAC)
    - ACE specifies the identity of requestor
  - Role-based Access Control (RBAC)
    - ACE specifies the role to accept of the entity requesting access
- ACL can be changed/updated via the AMS
  - Wildcards are supported to ease ACL management
  - ACL policies applies only at the OCF server side
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
  - For oic.sec.cred.trustca entries, also known as trust anchors for identity certificates, Subject ID has to match the ID 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>rowneruuuid</td>
<td>isop</td>
<td></td>
</tr>
<tr>
<td>sct</td>
<td>aclist2</td>
<td>cm</td>
<td></td>
</tr>
<tr>
<td>owned</td>
<td>rowneruuuid</td>
<td>tm</td>
<td></td>
</tr>
<tr>
<td>deviceuuuid</td>
<td>aclist2</td>
<td>om</td>
<td></td>
</tr>
<tr>
<td>devowneruuuid</td>
<td>rowneruuuid</td>
<td>sm</td>
<td></td>
</tr>
</tbody>
</table>
| rowneruuuid | }
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.
3 New Security Profiles for OCF 2.0

- **Optional and Certifiable improvements to Baseline Security Profile**
  - Black requires an audited CA, Blue/Purple require a vetted CA. All three include significant improvements to Device security such as hardware key storage, improved cipher suite support, etc.
  - A Device may be certified as conforming to any combination of Profiles. (e.g. Blue & Purple; Black only; Black & Blue & Purple; etc.)

- **Interoperable**: Devices of different Profiles can co-exist and interoperate.

- **Cryptographically Attestable**: Certificate extensions allow encoding of security attributes and OCF certification information.

- **Consistent**: No change to OCF branding due to Security Profiles.
Manufacturer Incentives to Use Security Profiles

• Purple: Manufacturer building a Device with requirement for measured boot and secure SW update, to improve device integrity (e.g. connect to cloud, healthcare or government).

• Black: Manufacturer wishing to require use of OCF PKI, which ensures certificates are signed by OCF PKI, and meet OCF Certificate Policy.

• Blue: Manufacturer wishing to use its own (or other non-OCF) CA, which must conform to OCF-defined CA vetting criteria.
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
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.
Infrastructure: Bridging Specification
OCF 2.0.2 Overview
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

- Provides **specific requirements** for translation between OCF and specific ecosystems
  - Current specification supports AllJoyn (Symmetric Bridging) and oneM2M (Asymmetric Bridging), support for other ecosystems will be added
    - Requirements for mapping core resources, propagating errors, and in the case of AllJoyn, algorithmically translating custom resource types.
The detailed mapping to/from OCF Resources and the equivalent constructs in a bridged ecosystem are provided in what are known as Mapping Specifications.

These specifications define the mapping between Device representations (OCF Device Types) and Resource Representations.

The following Mapping Specifications currently exist:

- OCF to AllJoyn Mapping specification for translating OCF Resources to AllJoyn Interfaces.
- OCF to oneM2M Mapping specification for translating OCF Resources to oneM2M Module Classes.
OCF Bridge – Definition

- An OCF Bridge is a device that represents one or more non-OCF Devices (Bridged Devices) as “Virtual OCF Devices” on the OCF network and represents one or more OCF-Devices as “Virtual Bridged Devices” on the non-OCF network.
- The Bridged Devices themselves are out of the scope of OCF.
- A “regular” OCF Device and a Virtual OCF Device are exactly same except that the Virtual OCF Device has “oic.d.virtual” as its "rt" value.
- An OCF Bridge is indicated on the network with an “rt” of “oic.d.bridge”.
**OCF Bridge - Definition**

- 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
Bridging Concept - Operation

Virtual OCF Devices

OCF Server

OCF Client

Bridge Platform

Bridge

Virtual OCF Server

Virtual OCF Client

Virtual Bridged Server

Virtual Bridged Client

Bridged Devices

Bridged Client

Bridged Server

OCF Protocol

OCF Protocol

Bridged Protocol

Bridged Protocol

OCF

Non-OCF
Bridging Concept - Data Model

OCF Resource Model

Derived Model

Non-OCF Ecosystem Resource Model

Resource Spec Device Spec

Mapping Spec

Other Ecosystem Spec
Bridging Security

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

Overview
Overview

• Models the interworking between OCF and the other ecosystem
• Predicated on OCF being the superset model; so any Device Types and Resource Types (as equivalents to constructs in the other ecosystem) that were missing from OCF were defined in the equivalent OCF Specifications.
• Defines the mapping in terms of:
  • Device Type equivalency
  • Resource <-> other ecosystem equivalency
  • Detailed Property by Property mapping (Derived Models)
Derived models use standard JSON schema syntax. Fundamentally, derived models provide a conversion mapping between OCF data models and the data models in the other ecosystem. The example below is for AllJoyn.

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

- **AllJoyn Interface Name**
- **AllJoyn Property Name**
- **OCF Equivalent Resource Type**

**To OCF Block**

**From OCF Block**
Ecosystem Bridges:
OCF to AllJoyn
Overview
## Device Type Equivalency

- Yellow highlights identify Device Types that were added to support 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>AirPurifier</td>
<td>Air Purifier</td>
<td>oic.d.airpurifier</td>
</tr>
<tr>
<td></td>
<td>AirQualityMonitor</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>ElectricFan</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>

May 14, 2019

Copyright © 2019 Open Connectivity Foundation, Inc. All Rights Reserved.
## 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></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></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></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.operationalstate</td>
<td>oic.if.s</td>
</tr>
<tr>
<td>Operation.FanSpeedLevel</td>
<td>Air Flow</td>
<td>oic.r.airflow</td>
<td>oic.if.a</td>
</tr>
<tr>
<td>Operation.HeatingZone</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></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: OCF to oneM2M

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

<table>
<thead>
<tr>
<th>oneM2M Device Type</th>
<th>OCF Device Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>deviceMicrogeneration</td>
<td>oic.d.energygenerator</td>
</tr>
<tr>
<td>deviceMultiFunctionPrinter</td>
<td>oic.d.multifunctionprinter</td>
</tr>
<tr>
<td>deviceOutdoorLamp</td>
<td>oic.d.light</td>
</tr>
<tr>
<td>deviceOven</td>
<td>oic.d.oven</td>
</tr>
<tr>
<td>devicePrinter</td>
<td>oic.d.printer</td>
</tr>
<tr>
<td>deviceRefrigerator</td>
<td>oic.d.refrigerator</td>
</tr>
<tr>
<td>deviceRobotCleaner</td>
<td>oic.d.robotcleaner</td>
</tr>
<tr>
<td>deviceScanner</td>
<td>oic.d.scanner</td>
</tr>
<tr>
<td>deviceSecurityPanel</td>
<td>oic.d.securitypanel</td>
</tr>
<tr>
<td>deviceSetTopBox</td>
<td>oic.d.stb</td>
</tr>
<tr>
<td>deviceSmartElectricMeter</td>
<td>oic.d.electricmeter</td>
</tr>
<tr>
<td>deviceSmartPlug</td>
<td>oic.d.smartplug</td>
</tr>
<tr>
<td>deviceSteamCloset</td>
<td>oic.d.steamcloset</td>
</tr>
<tr>
<td>deviceStorageBattery</td>
<td>oic.d.battery</td>
</tr>
<tr>
<td>deviceSwitch</td>
<td>oic.d.switch</td>
</tr>
<tr>
<td>deviceTelevision</td>
<td>oic.d.tv</td>
</tr>
<tr>
<td>deviceThermostat</td>
<td>oic.d.thermostat</td>
</tr>
<tr>
<td>deviceWaterHeater</td>
<td>oic.d.waterheater</td>
</tr>
<tr>
<td>deviceWaterValve</td>
<td>oic.d.watervalve</td>
</tr>
<tr>
<td>deviceWeightScaleAndBodyCompositionAnalyzer</td>
<td>oic.d.bodyscale</td>
</tr>
<tr>
<td>deviceWindowShade</td>
<td>oic.d.blind</td>
</tr>
<tr>
<td>deviceThermometer</td>
<td>oic.d.bodythermometer</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>acoustic sensor</td>
<td>oic.r.sound.pressure</td>
</tr>
<tr>
<td>aircon jobmode</td>
<td>oic.r.operational.state</td>
</tr>
<tr>
<td>airflow</td>
<td>oic.r.airflow</td>
</tr>
<tr>
<td>air purifier jobmode</td>
<td>oic.r.operational.state</td>
</tr>
<tr>
<td>air quality sensor</td>
<td>oic.r.air quality</td>
</tr>
<tr>
<td></td>
<td>oic.r.switch.binary</td>
</tr>
<tr>
<td></td>
<td>oic.r.humidity</td>
</tr>
<tr>
<td>alarmspeaker</td>
<td>oic.r.audio</td>
</tr>
<tr>
<td></td>
<td>oic.r.switch.binary</td>
</tr>
<tr>
<td></td>
<td>oic.r.light.dimming</td>
</tr>
<tr>
<td>audioVolume</td>
<td>oic.r.audio</td>
</tr>
<tr>
<td>auto documentfeeder</td>
<td>oic.r.operational.state</td>
</tr>
<tr>
<td>battery</td>
<td>oic.r.energy.battery</td>
</tr>
<tr>
<td>binary switch</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>clothes dryer jobmode</td>
<td>oic.r.operational.state</td>
</tr>
<tr>
<td>colour</td>
<td>oic.r.colour</td>
</tr>
<tr>
<td>colour saturation</td>
<td>oic.r.colour.saturation</td>
</tr>
<tr>
<td>credentials</td>
<td>oic.ruserinfo</td>
</tr>
<tr>
<td>dehumidifier jobmode</td>
<td>oic.r.operational.state</td>
</tr>
<tr>
<td>door status</td>
<td>oic.r.door</td>
</tr>
<tr>
<td>electric vehicle connector</td>
<td>oic.r.vehicle.connector</td>
</tr>
<tr>
<td>energy consumption</td>
<td>oic.r.energy.electrical oic.r.energy.consumption</td>
</tr>
<tr>
<td>energy generation</td>
<td>oic.r.energy.generation</td>
</tr>
<tr>
<td>filter info</td>
<td>oic.r.consumable</td>
</tr>
<tr>
<td></td>
<td>oic.r.sensor</td>
</tr>
<tr>
<td>foaming</td>
<td>oic.r.foaming</td>
</tr>
<tr>
<td>grinder</td>
<td>oic.r.grinder</td>
</tr>
<tr>
<td></td>
<td>oic.r.switch.binary</td>
</tr>
<tr>
<td>heating zone</td>
<td>oic.r.heatingzone</td>
</tr>
<tr>
<td>height</td>
<td>oic.r.height</td>
</tr>
<tr>
<td>hotwater supply</td>
<td>oic.r.switch.binary</td>
</tr>
<tr>
<td></td>
<td>oic.r.sensor</td>
</tr>
<tr>
<td>impact sensor</td>
<td>oic.r.impact.sensor</td>
</tr>
<tr>
<td>keep warm</td>
<td>oic.r.time.period</td>
</tr>
<tr>
<td>keypad</td>
<td>oic.r.keypad.char</td>
</tr>
<tr>
<td>liquid level</td>
<td>oic.r.liquid.level</td>
</tr>
<tr>
<td>liquid remaining</td>
<td>oic.r.liquid.level</td>
</tr>
<tr>
<td>lock</td>
<td>oic.r.lock</td>
</tr>
<tr>
<td>motion sensor</td>
<td>oic.r.sensor.motion</td>
</tr>
<tr>
<td></td>
<td>oic.r.sensor.props</td>
</tr>
<tr>
<td>open level</td>
<td>oic.r.openlevel</td>
</tr>
<tr>
<td>operation mode</td>
<td>oic.r.switch.binary</td>
</tr>
<tr>
<td>over current sensor</td>
<td>oic.r.time.period</td>
</tr>
<tr>
<td></td>
<td>oic.r.sensor</td>
</tr>
<tr>
<td>power save</td>
<td>oic.r.switch.binary</td>
</tr>
<tr>
<td>print queue</td>
<td>oic.r.printer.queue</td>
</tr>
<tr>
<td>push button</td>
<td>oic.r.button</td>
</tr>
<tr>
<td>refrigeration</td>
<td>oic.r.refrigeration</td>
</tr>
<tr>
<td>relative humidity</td>
<td>oic.r.humidity</td>
</tr>
<tr>
<td>robot cleaner jobmode</td>
<td>oic.r.operational.state</td>
</tr>
<tr>
<td>steam cleaner jobmode</td>
<td>oic.r.operational.state</td>
</tr>
<tr>
<td>temperature</td>
<td>oic.r.temperature</td>
</tr>
<tr>
<td>uvc sensor</td>
<td>oic.r.sensor.radiation.uv</td>
</tr>
<tr>
<td>water sensor</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.
OCF Specification Overview
Device and Resource Modeling
OCF 2.0 Release
Resource Specification

- List of reusable resources that are used in an OCF Device
- More than 100 Resource Types defined as of OCF 2.0.2 enabling Smart Home, Healthcare, and Industrial applications.
- All Resource Types build on the Core definitions
- Each resource definition contains:
  - unique identifier (rt)
  - Identification of the default interface and other supported interfaces
  - List of supported methods
  - List 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
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
### Newly Defined Resource Types - OCF 2.0

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>Use Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D Printer</td>
<td>Device Control</td>
</tr>
<tr>
<td>Blood Pressure</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Body Mass Index</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Body Fat</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Body Fat Free Mass</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Body Location</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Body Location Temperature</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Body Soft Lean Mass</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Body Water</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Calorific Value</td>
<td>Smart Energy</td>
</tr>
<tr>
<td>Conversion Factor</td>
<td>Smart Energy</td>
</tr>
<tr>
<td>Gas Consumption</td>
<td>Smart Energy</td>
</tr>
<tr>
<td>Gas Usage</td>
<td>Smart Energy</td>
</tr>
<tr>
<td>Glucose</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Glucose Carbohydrates</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Glucose Exercise</td>
<td>Personal Health</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>Use Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glucose HbA1c</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Glucose Health</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Glucose Meal</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Glucose Medication</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Glucose Sample Location</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Glucose Tester</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Impact Sensor</td>
<td>Sensor Modeling</td>
</tr>
<tr>
<td>Keypad Char</td>
<td>Home Security</td>
</tr>
<tr>
<td>Opaque Data</td>
<td>Smart Home</td>
</tr>
<tr>
<td>ORFID Station</td>
<td>Smart Factory</td>
</tr>
<tr>
<td>ORFID Tag</td>
<td>Smart Factory</td>
</tr>
<tr>
<td>Power Source</td>
<td>Smart Energy</td>
</tr>
<tr>
<td>Print Queue</td>
<td>Device Control</td>
</tr>
<tr>
<td>Pulse Rate</td>
<td>Personal Health</td>
</tr>
<tr>
<td>Sensor Properties</td>
<td>Generic Sensor Modeling</td>
</tr>
<tr>
<td>User ID</td>
<td>Personal Health</td>
</tr>
<tr>
<td>User Info</td>
<td>Smart Home</td>
</tr>
</tbody>
</table>

Resource Types are Conditionally Mandatory. If an OCF Server hosts an OCF known resource then it shall follow all normative requirements in the Resource Specification applicable to that Resource.
Device Specification

OCF 2.0.2 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>

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 on 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

- 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)
## 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.
### Complete Set of OCF Defined Device Types (1/2)

<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>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>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.bloodpressuremonitor</td>
</tr>
<tr>
<td>Body Scale</td>
<td>oic.d.bodyscale</td>
</tr>
<tr>
<td>Body Thermometer</td>
<td>oic.d.bodyscale</td>
</tr>
<tr>
<td>Camera</td>
<td>oic.d.camera</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>Cooker Hood</td>
<td>oic.d.cookerhood</td>
</tr>
<tr>
<td>Cooktop</td>
<td>oic.d.cooktop</td>
</tr>
<tr>
<td>Dehumidifier</td>
<td>oic.d.dehumidifier</td>
</tr>
<tr>
<td>Dishwasher</td>
<td>oic.d.dishwasher</td>
</tr>
<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 Charger</td>
<td>oic.d.electricvehiclecharger</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>Humidifier</td>
<td>oic.d.humidifier</td>
</tr>
<tr>
<td>Kettle</td>
<td>oic.d.kettle</td>
</tr>
<tr>
<td>Light</td>
<td>oic.d.light</td>
</tr>
<tr>
<td>Microwave Oven</td>
<td>oic.d.microwave</td>
</tr>
<tr>
<td>Oven</td>
<td>oic.d.oven</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>Receiver</td>
<td>oic.d.receiver</td>
</tr>
</tbody>
</table>
## Complete Set of OCF Defined Device Types (2/2)

<table>
<thead>
<tr>
<th>Friendly Name</th>
<th>Device Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refrigerator</td>
<td>oic.d.refrigerator</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>Smart Lock</td>
<td>oic.d.lock</td>
</tr>
<tr>
<td>Smart Plug</td>
<td>oic.d.smartplug</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>
<tr>
<td>Water Valve</td>
<td>oic.d.watervalve</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Friendly Name</th>
<th>Device Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window</td>
<td>oic.d.window</td>
</tr>
</tbody>
</table>

Items in red are new in OCF 2.0
Thank you!

• Access the OCF specifications
  https://openconnectivity.org/resources/specifications
• Contact OCF at admin@openconnectivity.org
References
Where can I find the specifications and Resource Type definitions?

OCF Specifications:
- [https://openconnectivity.org/developer/specifications](https://openconnectivity.org/developer/specifications)

Resource Type Definitions
- Core Resources: [https://github.com/openconnectivityfoundation/core](https://github.com/openconnectivityfoundation/core)
- Core Extension Resources: [https://github.com/openconnectivityfoundation/core-extensions](https://github.com/openconnectivityfoundation/core-extensions)
- Bridging Resources: [https://github.com/openconnectivityfoundation/bridging](https://github.com/openconnectivityfoundation/bridging)
- Security Resources: [https://github.com/openconnectivityfoundation/security-models](https://github.com/openconnectivityfoundation/security-models)
- Vertical Resources and Derived Models: [https://oneiota.org/documents?filter%5Bmedia_type%5D=application%2Framl%2Byaml](https://oneiota.org/documents?filter%5Bmedia_type%5D=application%2Framl%2Byaml)
A web front end to the github hosted device and resource maps that are maintained by OCF may be found here:

• https://openconnectivityfoundation.github.io/devicemodels/docs/index.html
OneIoTa Tool

- Web based (see: http://oneiota.org) development tool
- Supports RAML, JSON, and Swagger2.0 syntax
- Populated to date with all OCF Resources, Swagger2.0 versions of all such Resources, and OCF-AllJoyn derived models.
- Supports multiple organizations
  - Each submitting organization defines their own license terms