OPEN CONNECTIVITY FOUNDATION®

OCF Specification Introduction and Overview

June 2018

Table of Contents

- Summary of OCF2.0 additions/improvements with respect to OCF1.3
- 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
    - Resource Type
    - OCF to AllJoyn Mapping
    - Device Profile

Summary of additions and improvements of OCF 2.0 with respect to OCF 1.3

Core specification

- OCF cloud infrastructure
  - Remote access infrastructure: accessing devices at home while not at home
- Atomic measurement design pattern
  - Design pattern that treats a set of resources as a single addressable (atomic) unit
  - All data is retrieved atomically in one call; no access the individual resources
- WiFi easy setup enhancements; beaconing information using a softAP
- CoAP over TCP or TLS for interacting with resources (exposed per resource)

Security specification

- Onboarding Tool (OBT) definition and requirements
- PKI certificate extension requirements
- Simplified Access Control List for easier access control configuration
- Additional vendor attestations and best practices for Device security

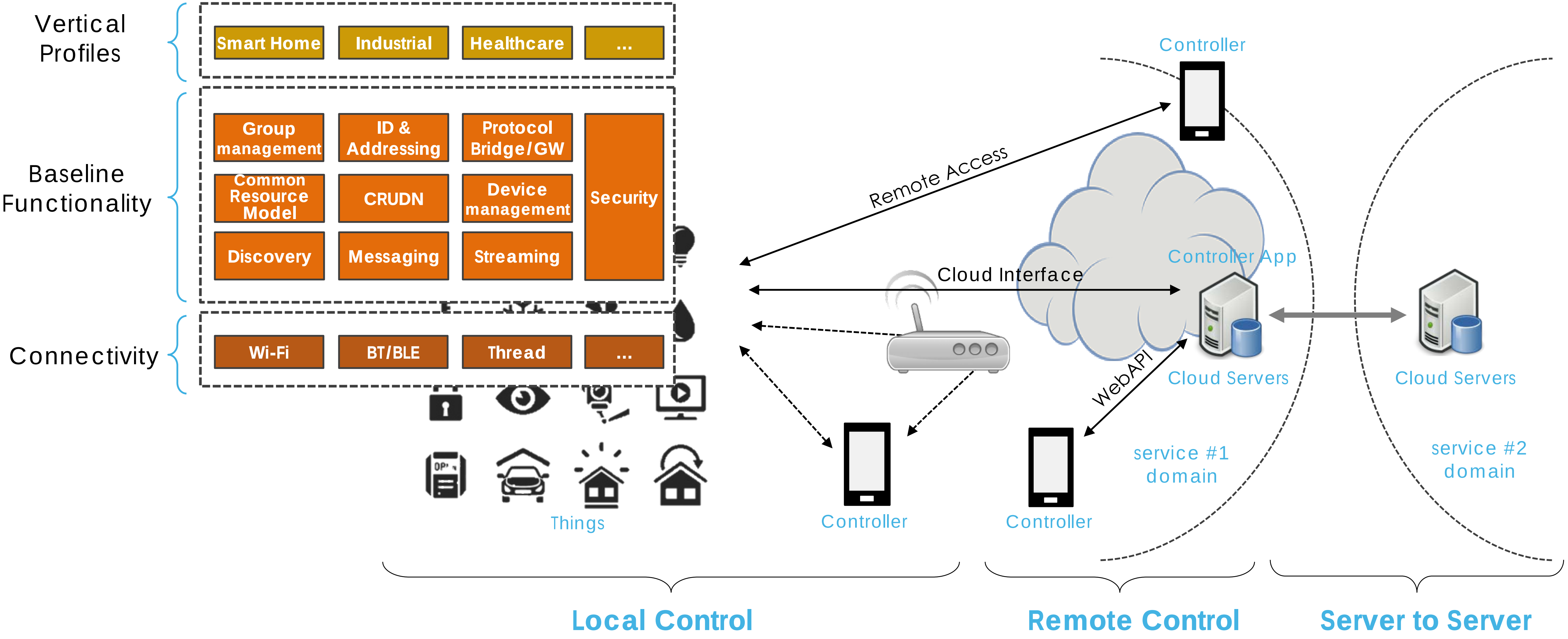
Resource/Device specification

- Additional resource types
  - For Healthcare and Industrial Use Cases
- Additional device types
  - Providing equivalency to oneM2M
- Device types are categorized
- New device categories added:
  - Healthcare device types
  - Industrial device types

OPEN CONNECTIVITY FOUNDATION®

Technical Principles for an Internet of Things Ecosystem

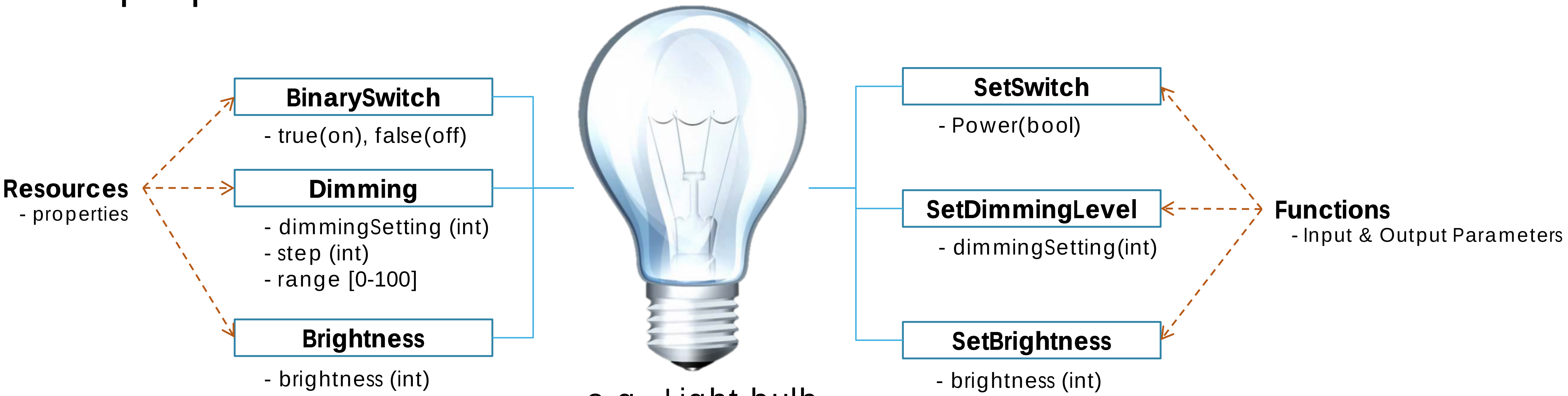
Scope of IoT


Vertical Profiles
Smart Home
Industrial
Healthcare
...
Baseline Functionality
Group management
ID & Addressing
Protocol Bridge/GW
Security
Common Resource Model
CRUDN
Device management
Discovery
Messaging
Streaming
Connectivity
Wi-Fi
BT/BLE
Thread
...
Things
Controller
Remote Access
Cloud Interface
Controller
Controller App
Cloud Servers
WebAPI
Controller
service #1 domain
Cloud Servers
service #2 domain
Local Control
Remote Control
Server to Server

Approaches to definition of various Things

- By defining resources of things and its properties

- By defining functions/operations of things



e.g., Light bulb

- (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

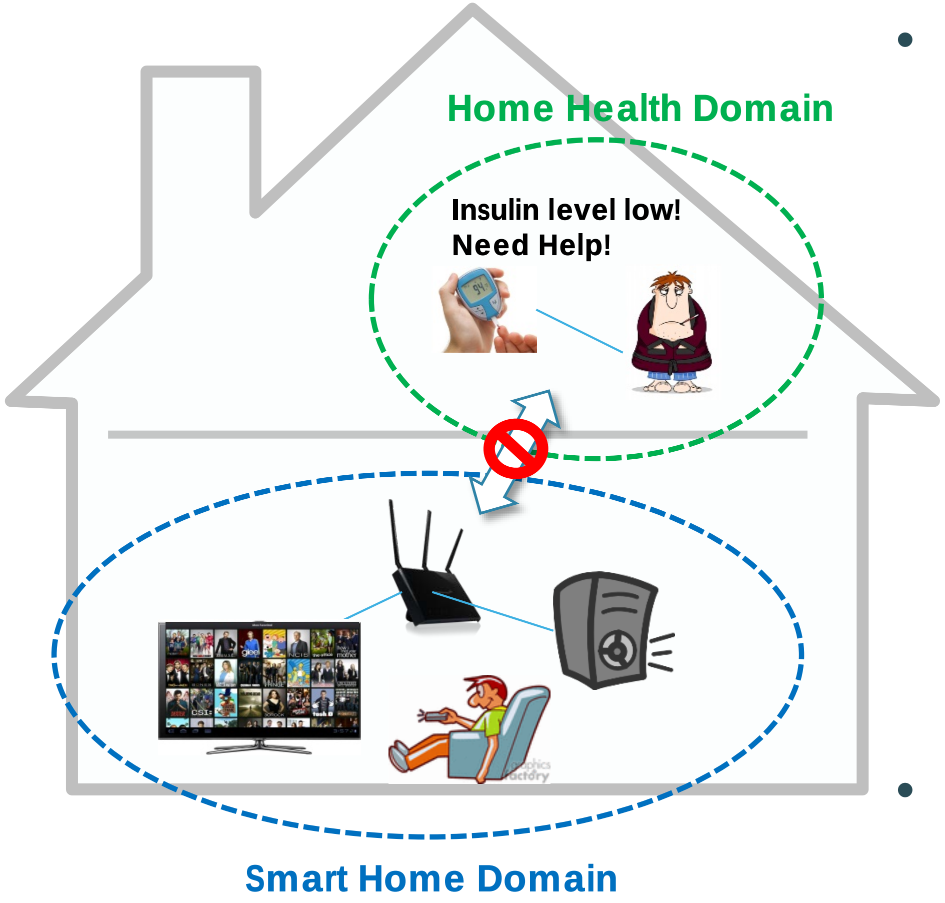
- Less overhead/ Less Traffic
  - Minimize CPU Load, Memory impacts, Traffic and Bandwidth
    - Compact header
    - Binary protocol
    - Compressed encoding of payload

- Low Complexity
  - Simple Resource Model

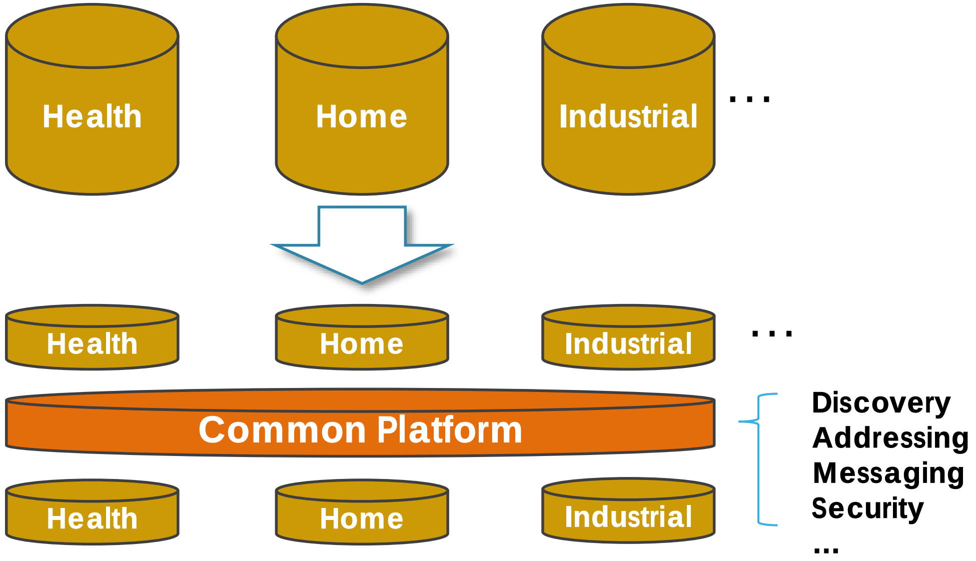
    > Short URI (Late Binding w/ resource type defined)

    > Broad and Shallow Hierarchy

Support of Multiple Verticals



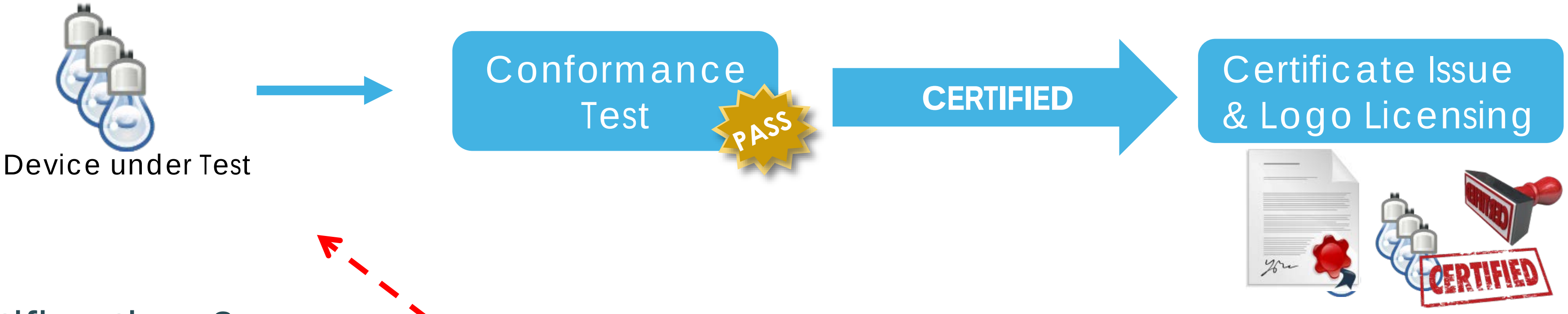
- Legacy vertical services usually designed as silos
  → No common way to communicate among them



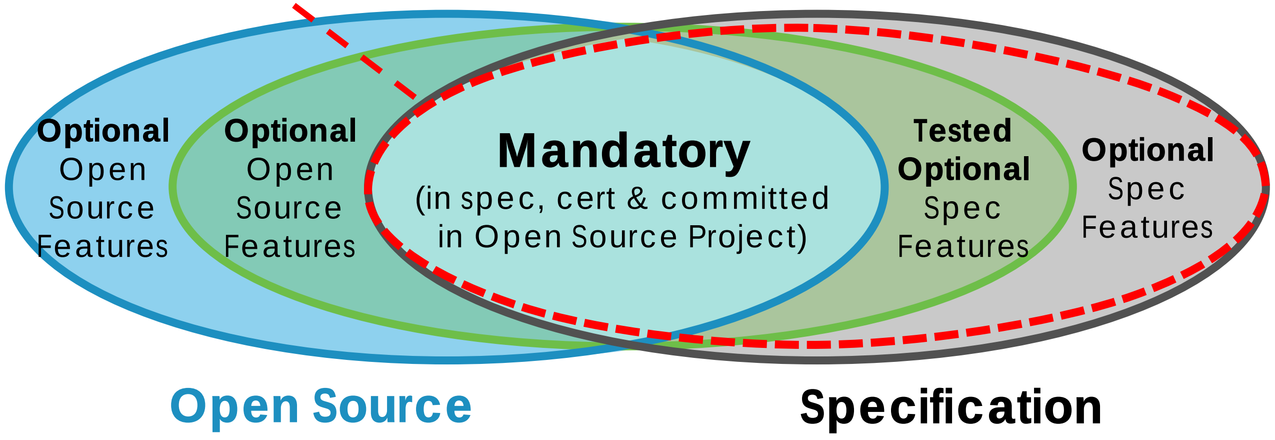
- A common platform provides a foundation for vertical services to collaborate and interwork by providing common services and data models

Conformance & Certification

- Conformance test - Each device proves conformance to specifications



- Certification Scope


Optional Open Source Features
Optional Open Source Features
Mandatory
(in spec, cert & committed in Open Source Project)
Tested Optional Spec Features
Optional Spec Features
Open Source
Specification

Licensing

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

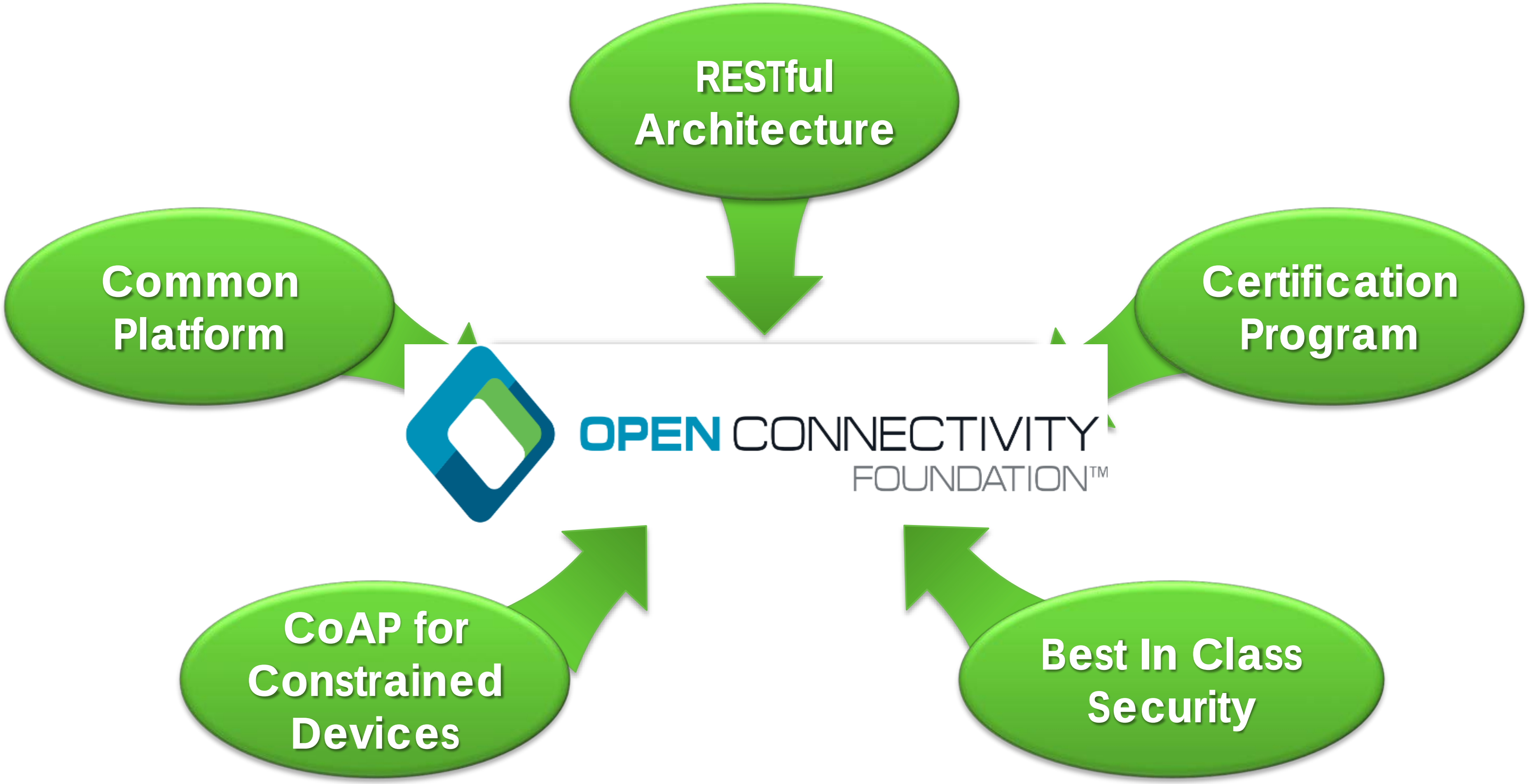
- Due to the common nature of IoT connecting everything over the Internet, it's most critical for manufacturers to avoid a licensing risk
  - Everything connected could be at potential risk

- 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.

OPEN CONNECTIVITY FOUNDATION®

Introduction to the Open Connectivity Foundation

Introduction to OCF – Optimized for IoT



RESTful Architecture

Common Platform

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
  - 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

OPEN CONNECTIVITY FOUNDATION®

OCF Specification Overview

OCF Deliverables

Normative Specifications

- See next slide

Resource Models via oneIoTa

- Domain agnostic resources
- Derived models for Ecosystem Mapping
  - To date: OCF-AllJoyn (CDM 16.4)

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)

Specification Location

Where can I find the specifications and Resource Type definitions?

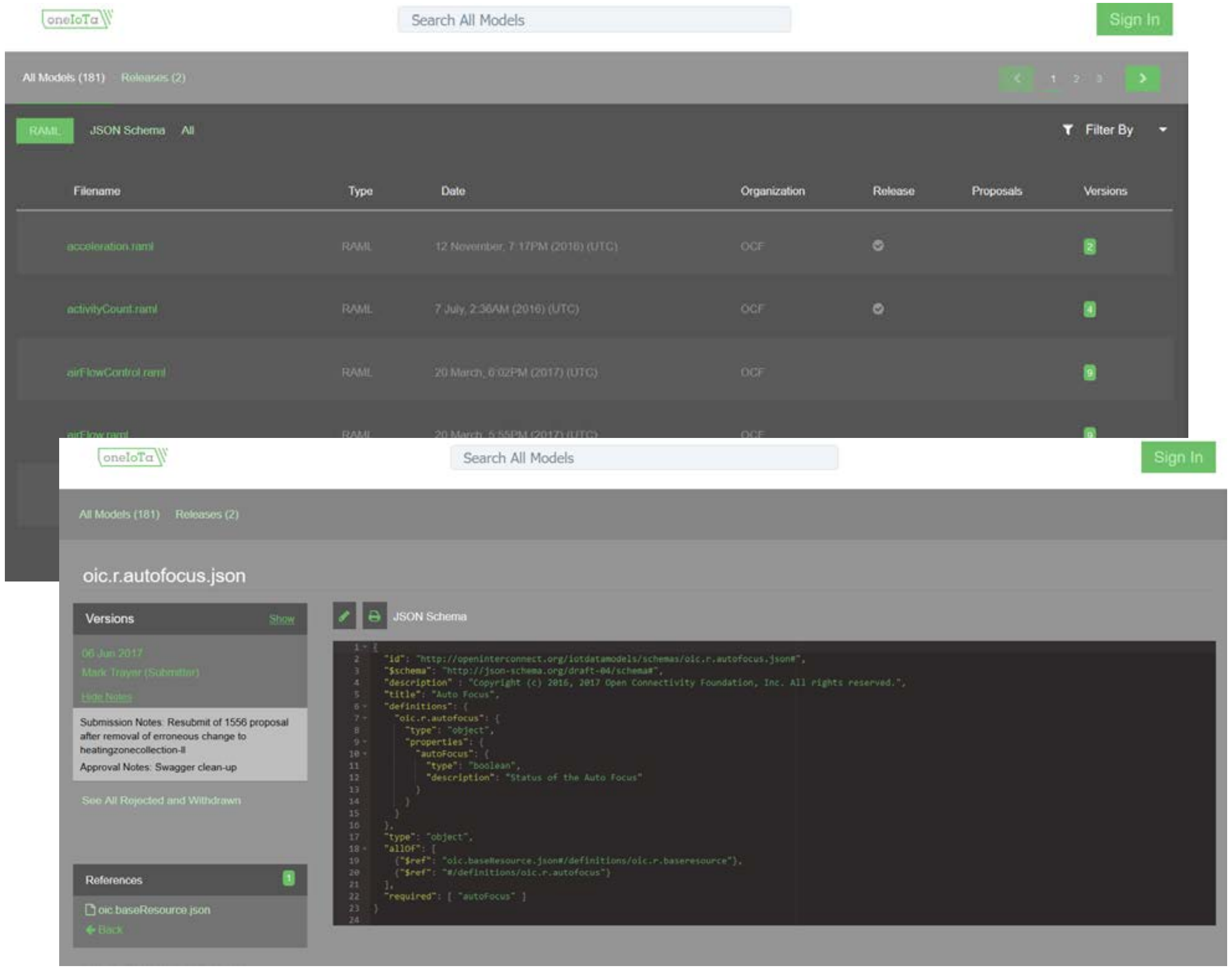
OCF Specifications:

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

Resource Type Definitions

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

OneIoTa Tool



Filename	Type	Date	Organization	Release	Proposals	Versions
acceleration.raml	RAML	12 November, 7:17PM (2016) (UTC)	OCF	✓		2
activityCount.raml	RAML	7 July, 2:36AM (2016) (UTC)	OCF	✓		4
airFlowControl.raml	RAML	20 March, 6:02PM (2017) (UTC)	OCF			9
airFlow.raml	RAML	20 March, 5:55PM (2017) (UTC)	OCF			9

oic.r.autofocus.json

Versions Show

06 Jun 2017

Mark Trayer (Submitter)

Hide Notes

Submission Notes: Resubmit of 1556 proposal after removal of erroneous change to heatingzonecollection-ll

Approval Notes: Swagger clean-up

See All Rejected and Withdrawn

References 1

oic.baseResource.json

Back

JSON Schema

```
{
  "id": "http://openinterconnect.org/iotdatamodels/schemas/oic.r.autofocus.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2016, 2017 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Auto Focus",
  "definitions": {
    "oic.r.autofocus": {
      "type": "object",
      "properties": {
        "autoFocus": {
          "type": "boolean",
          "description": "Status of the Auto Focus"
        }
      }
    }
  },
  "type": "object",
  "allOf": [
    {"$ref": "oic.baseResource.json#/definitions/oic.r.baseresource"},
    {"$ref": "#/definitions/oic.r.autofocus"}
  ],
  "required": [ "autoFocus" ]
}
```

- 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

Infrastructure: Core Framework Specification

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

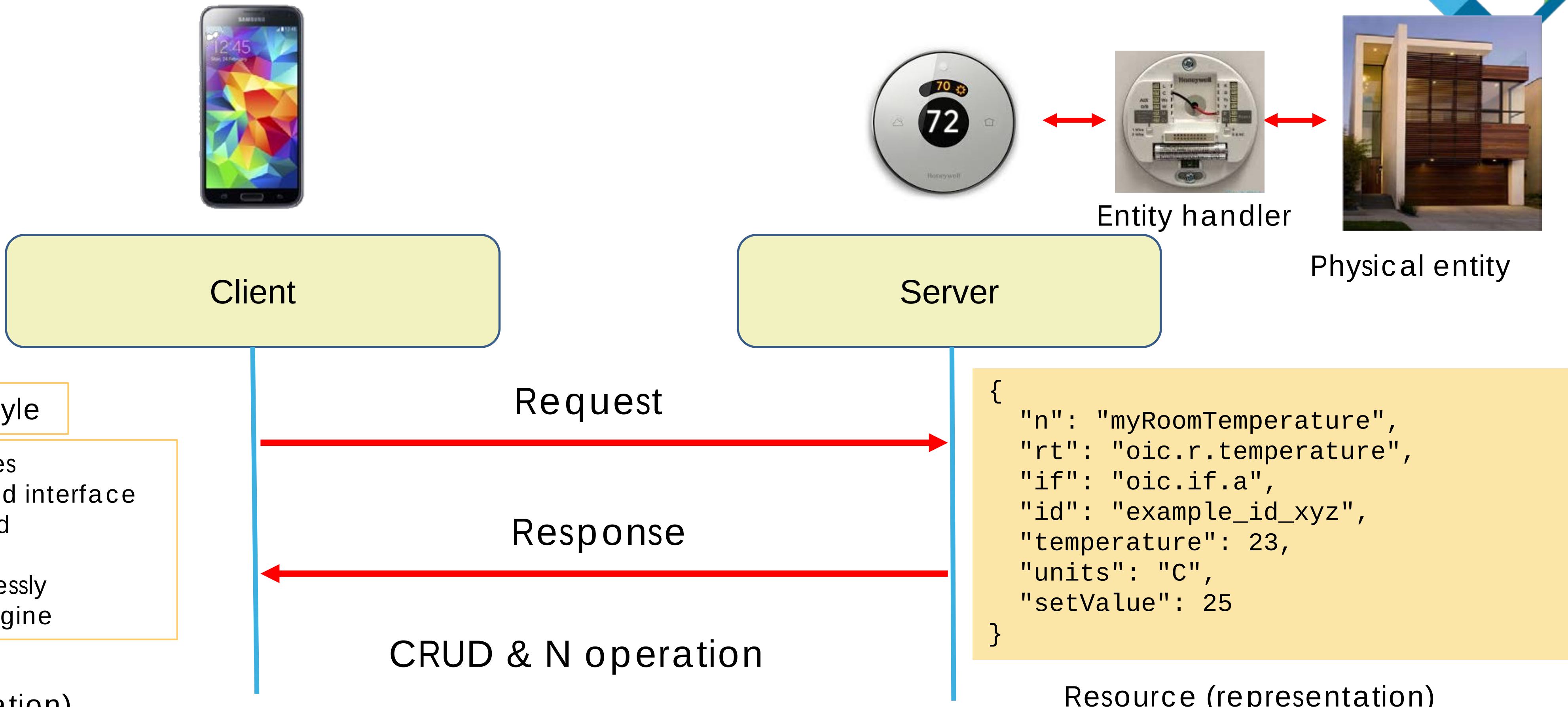
Core Framework Topics Outline (2 of 2)

- 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) 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



REST Architecture Style

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

Resource (representation)

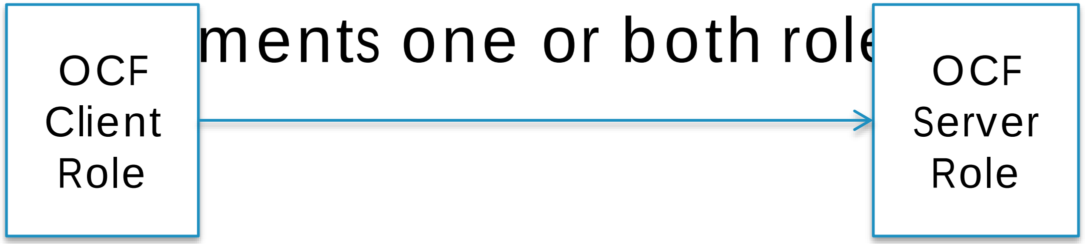
```
{
  "n": "myRoomTemperature",
  "rt": "oic.r.temperature",
  "if": "oic.if.a",
  "id": "example_id_xyz",
  "temperature": 23,
  "units": "C",
  "setValue": 25
}
```

Resource (representation)

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

OCF Roles

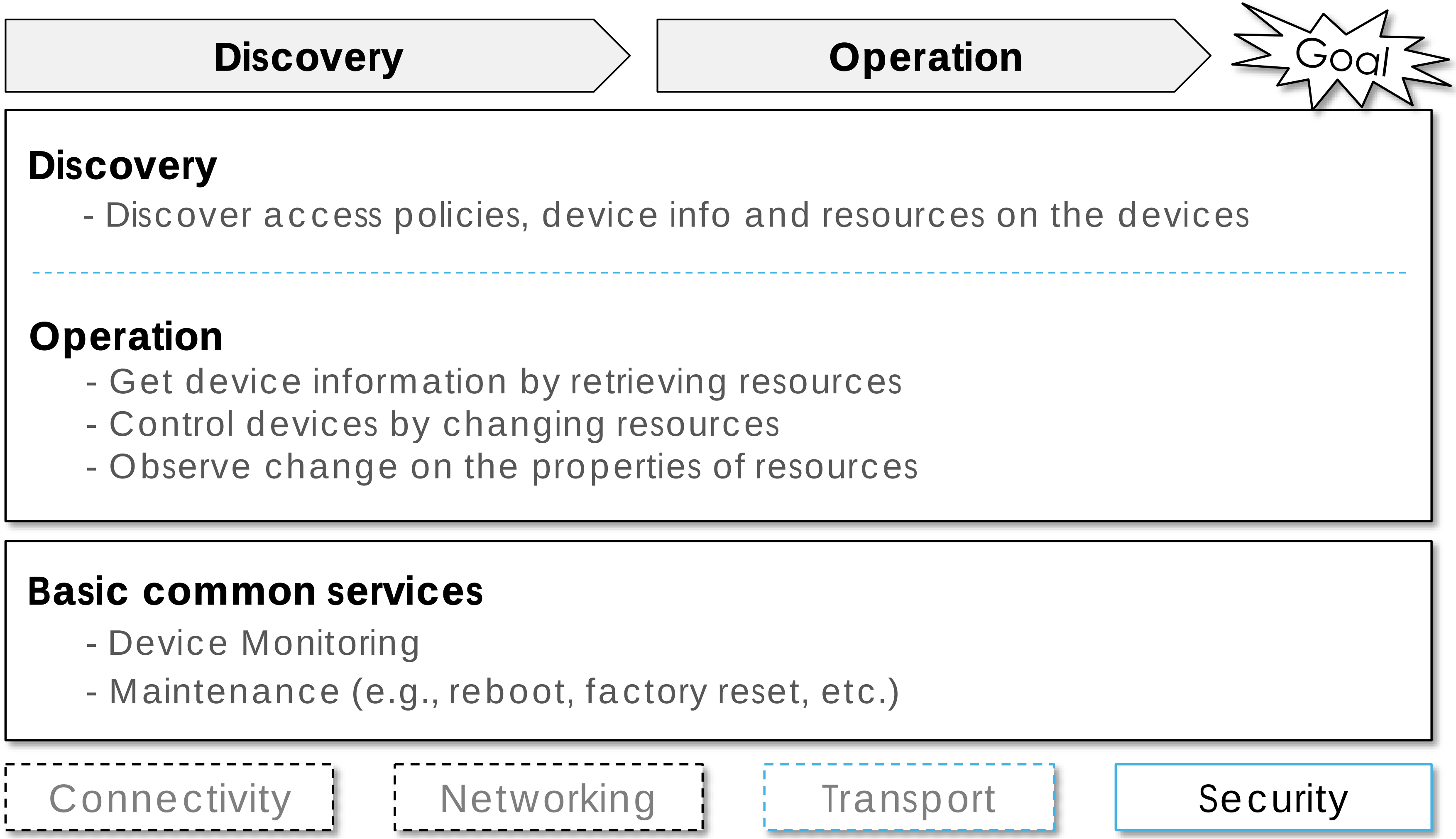
- Current OCF Architecture defines 2 logical roles that devices can take
  - OCF Server : A logical entity that exposes hosted resources, is discoverable, and responds to client initiated transactions
  - OCF Client : A logical entity that interacts with resources on an OCF Server via discovery and CRUDN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



Resources

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

OCF Core Framework Basic Operation



Discovery

Operation

Goal

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.)

Connectivity

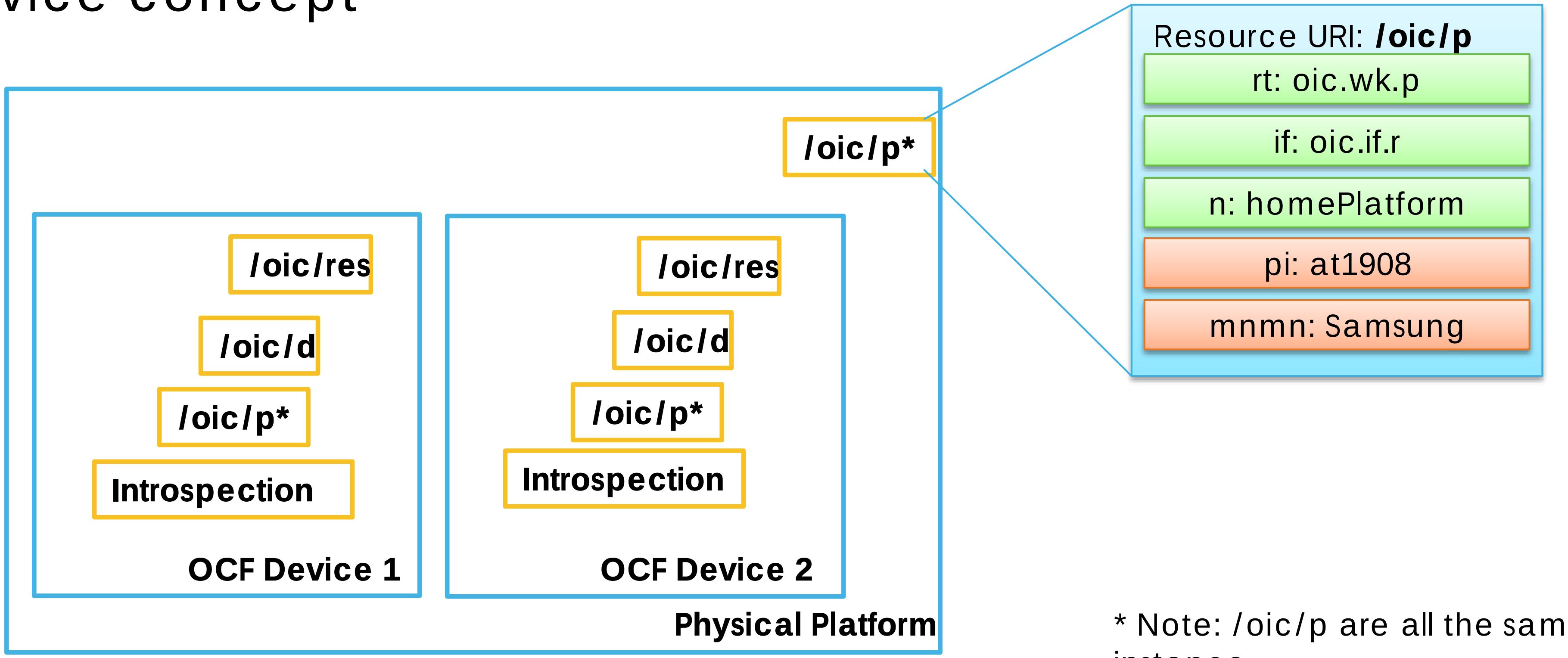
Networking

Transport

Security

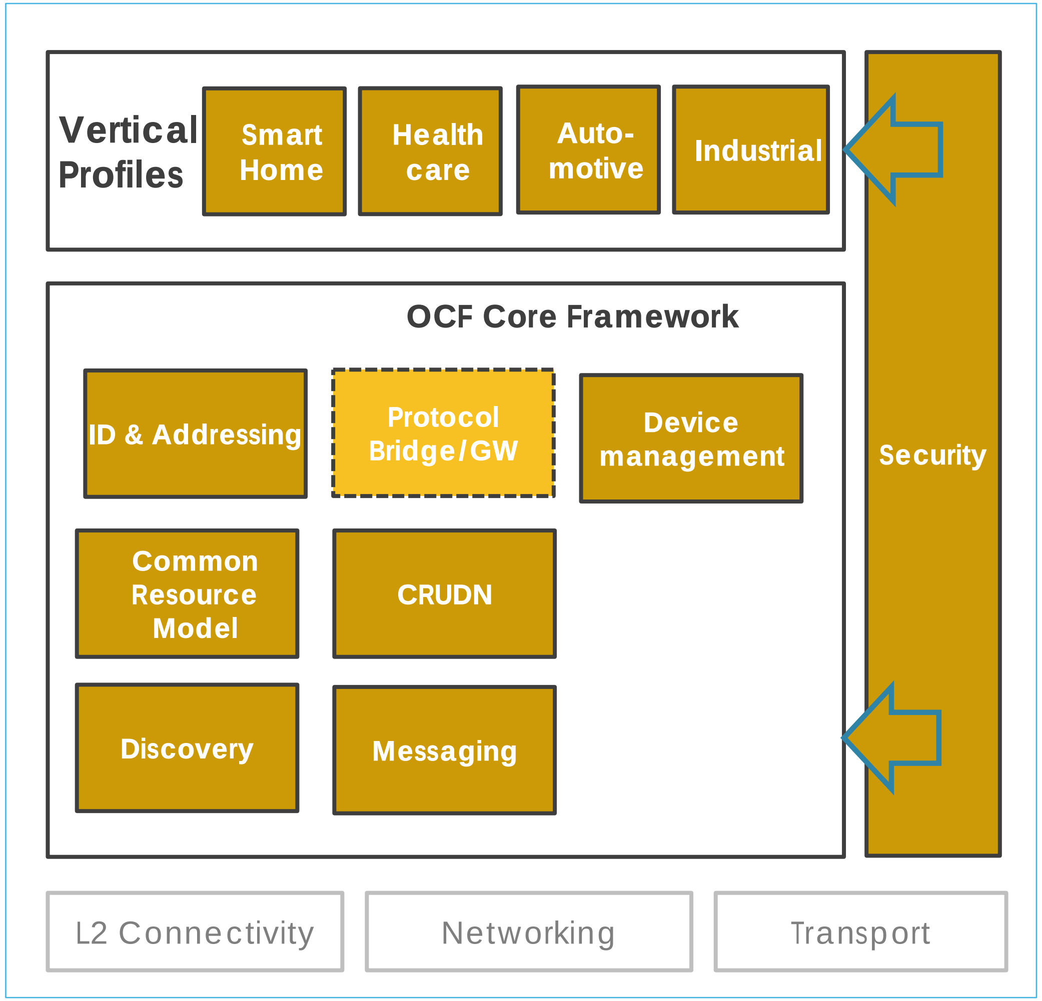
Organization of an OCF Device

- OCF Device concept



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

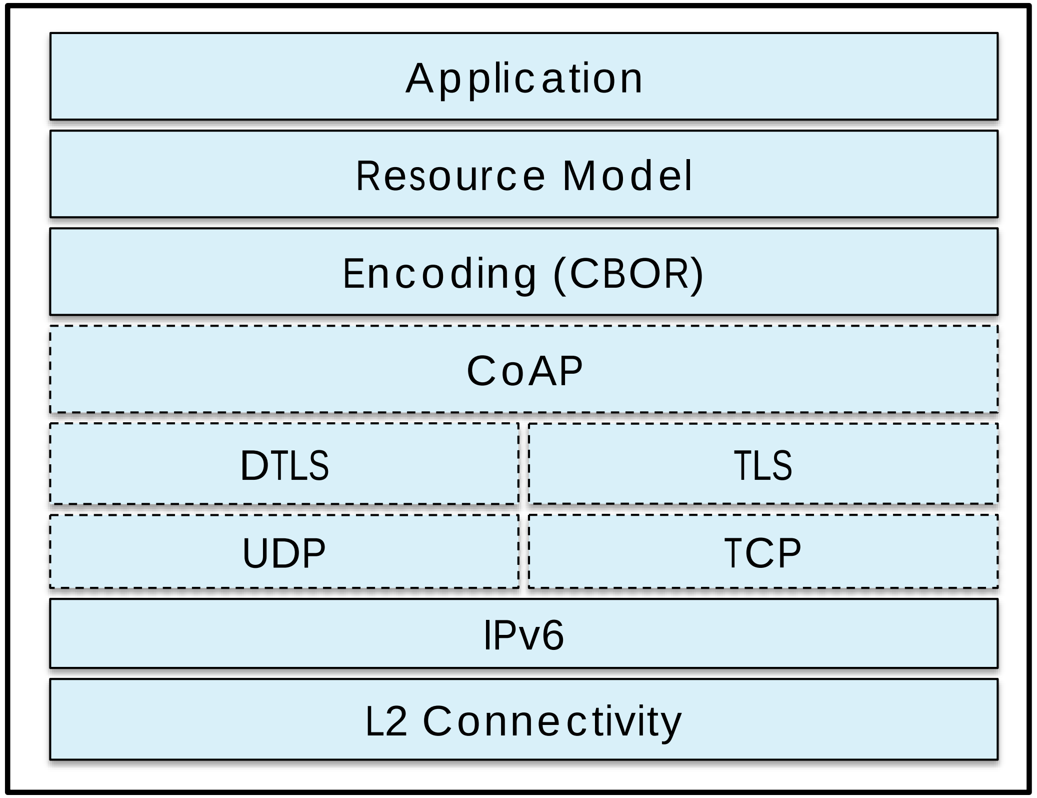
OCF Spec Features – Core Framework Spec



1. **Discovery:** Common method for device discovery (IETF CoRE)
2. **Messaging:** Constrained device support as default (IETF CoAP) as well as protocol translation via bridges
3. **Common Resource Model:** Real world entities defined as data models (resources)
4. **CRUDN:** Simple Request/Response mechanism with Create, Retrieve, Update, Delete and Notify commands
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 | TLS

UDP | TCP

IPv6

L2 Connectivity

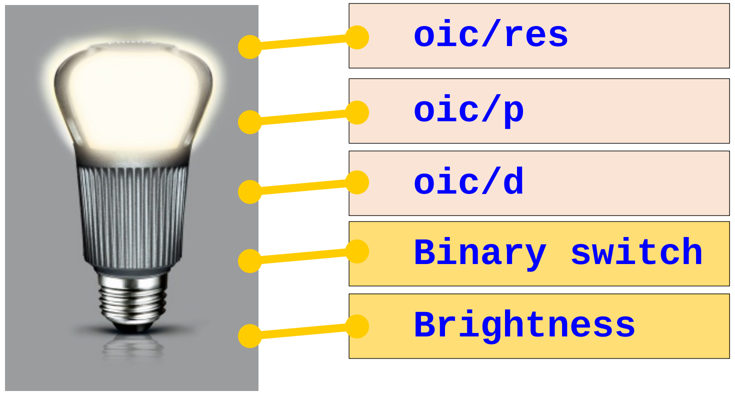
OCF Stack

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)

Example: Smart light device

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



oic/res

oic/p

oic/d

Binary switch

Brightness

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.
    - (e.g.) For CoAP/UDP/IPv6, Endpoint is identified as IP address + port number.
- 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://[fe80::b1d6]:44444"}, {"ep": "coaps://[fe80::b1d6]: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": "coap://[fe80::b1d6]:44444"}, {"ep": "coaps://[fe80::b1d6]:11111"} ] }
]
```

Endpoint for each target resource.

Resource Discovery (CoAP Discovery)

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

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 JSON schema plus RAML to define the associated API
- OCF has mandated CBOR as the default encoding scheme on the wire

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

If needed in future revisions

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 RAML and JSON on wire as a CBOR encoded Swagger2.0 document.
    - Describes the payload on JSON level
      - Property names
      - Type
      - range

Introspection: Underlying rationale

- Leave the current way of working intact: e.g. use RAML+JSON as is: use it as input for the Swagger2.0 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)

Versioning

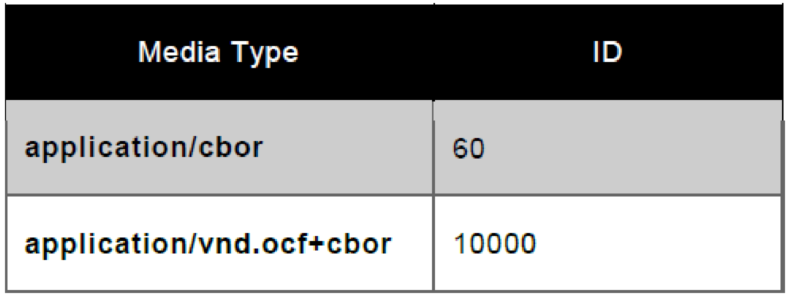
Payload Versioning

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

Device Level Versioning

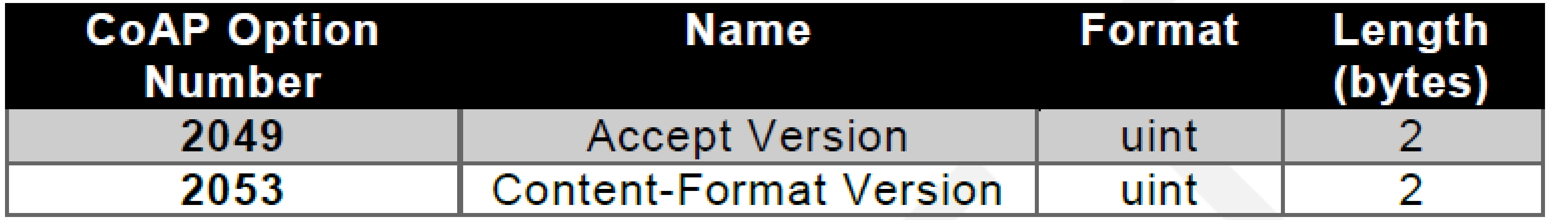
- **Purpose**: OCF devices can be aware of each others version
- **Method**: icv (spec version), dmv (data model version) in /oic/d resource

Payload versioning



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

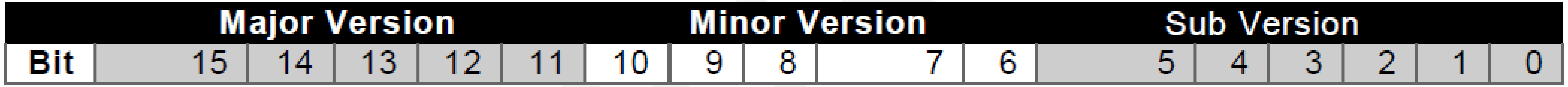
Content-Formats



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

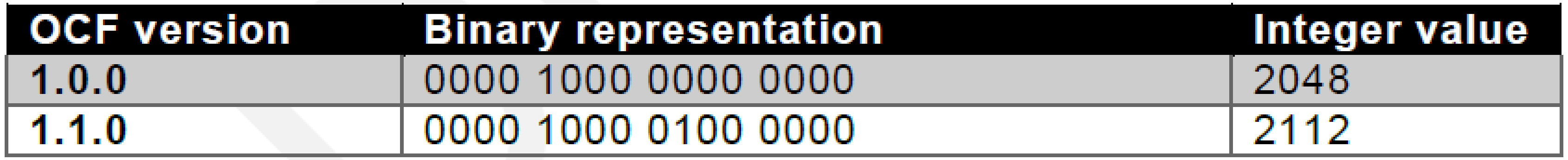
Option Numbers

Version Representation



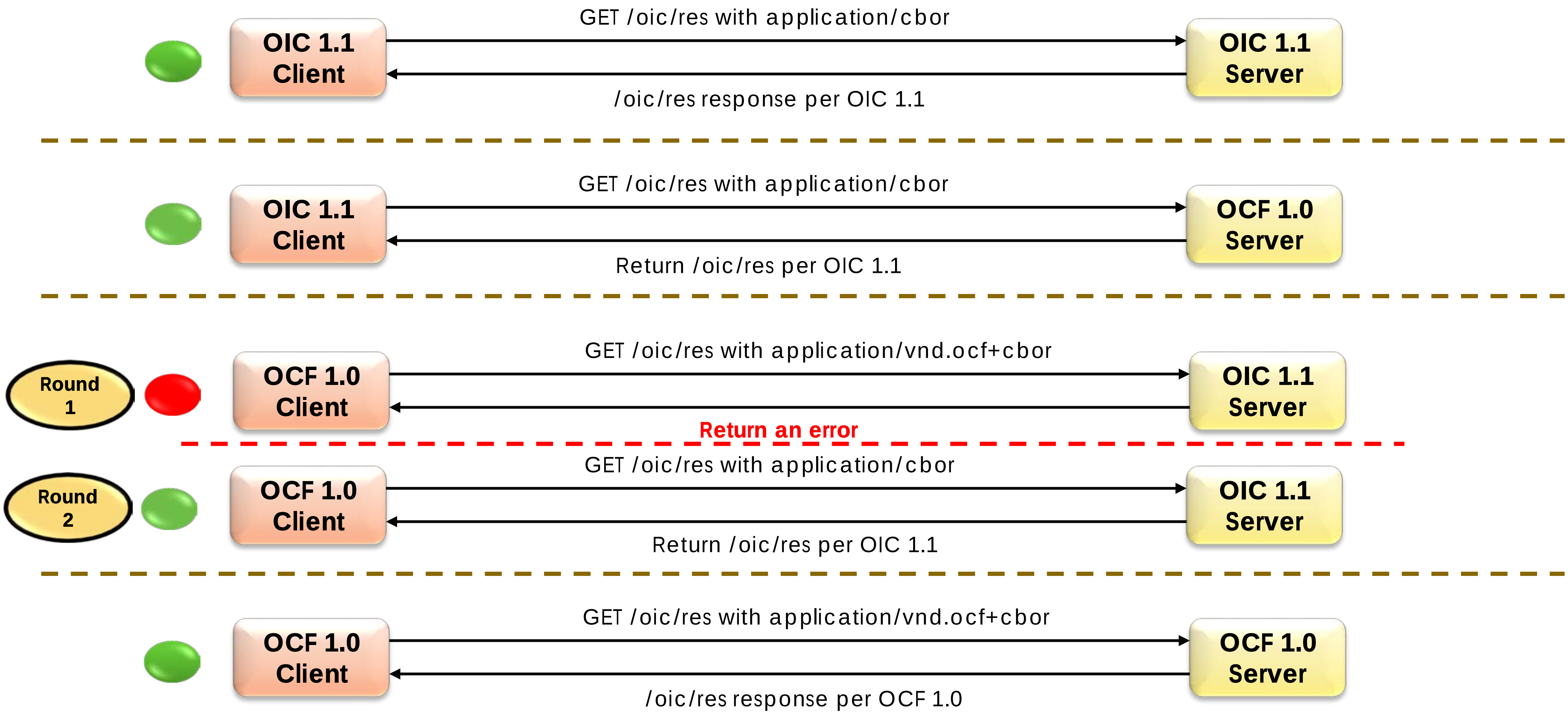
	Major Version					Minor Version					Sub Version					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Version Example



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

Payload Versioning Use Case & Policies



🟢 **OIC 1.1 Client** → **OIC 1.1 Server**

GET /oic/res with application/cbor

/oic/res response per OIC 1.1

🟢 **OIC 1.1 Client** → **OCF 1.0 Server**

GET /oic/res with application/cbor

Return /oic/res per OIC 1.1

Round 1 🔴 **OCF 1.0 Client** → **OIC 1.1 Server**

GET /oic/res with application/vnd.ocf+cbor

Return an error

Round 2 🟢 **OCF 1.0 Client** → **OIC 1.1 Server**

GET /oic/res with application/cbor

Return /oic/res per OIC 1.1

🟢 **OCF 1.0 Client** → **OCF 1.0 Server**

GET /oic/res with application/vnd.ocf+cbor

/oic/res response per OCF 1.0

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.

OCF Specification Overview Core Extension: OCF Cloud

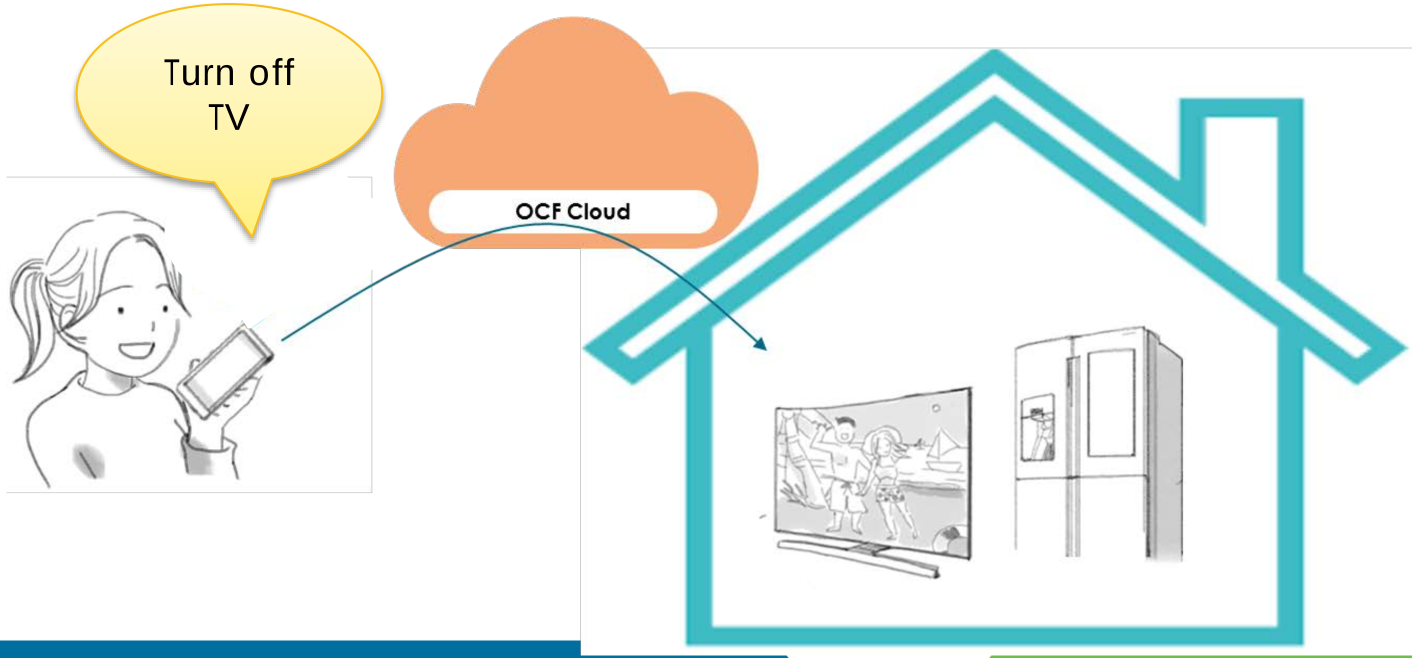
OCF 2.0 Release

Use Case

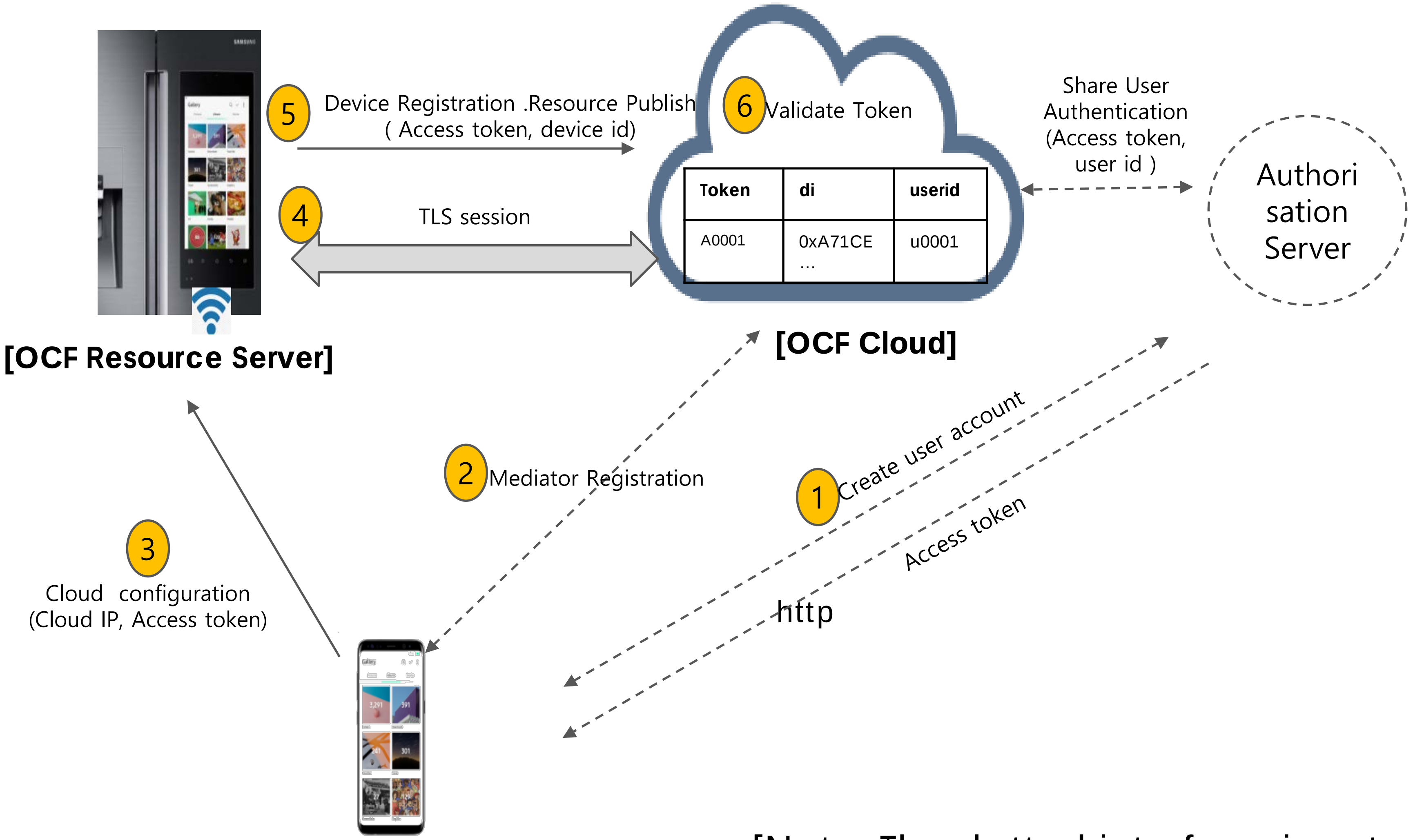
- Remote Control/Manage OCF devices based on user authentication
  - User can access OCF devices which belong or are shared with him/her regardless of a location.
  - User can receive cloud providing service through the registered devices
    - EX) Device Management, Home security, Energy management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



5 Device Registration .Resource Publish (Access token, device id)

4 TLS session

6 Validate Token

Token	di	userid
A0001	0xA71CE …	u0001

Share User Authentication (Access token, user id)

Authori sation Server

[OCF Resource Server]

[OCF Cloud]

2 Mediator Registration

1 Create user account

Access token

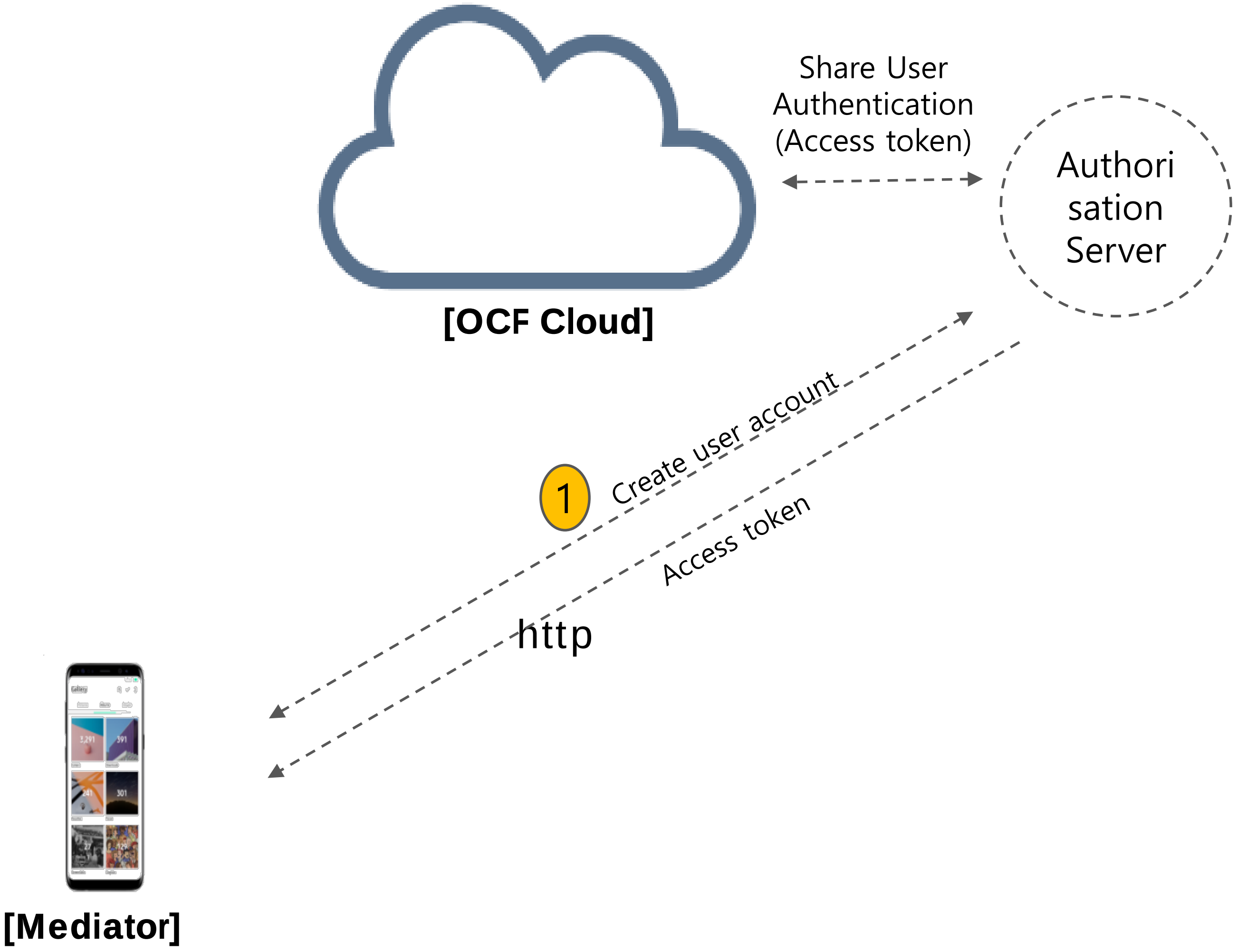
http

3 Cloud configuration (Cloud IP, Access token)

[Mediator]

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

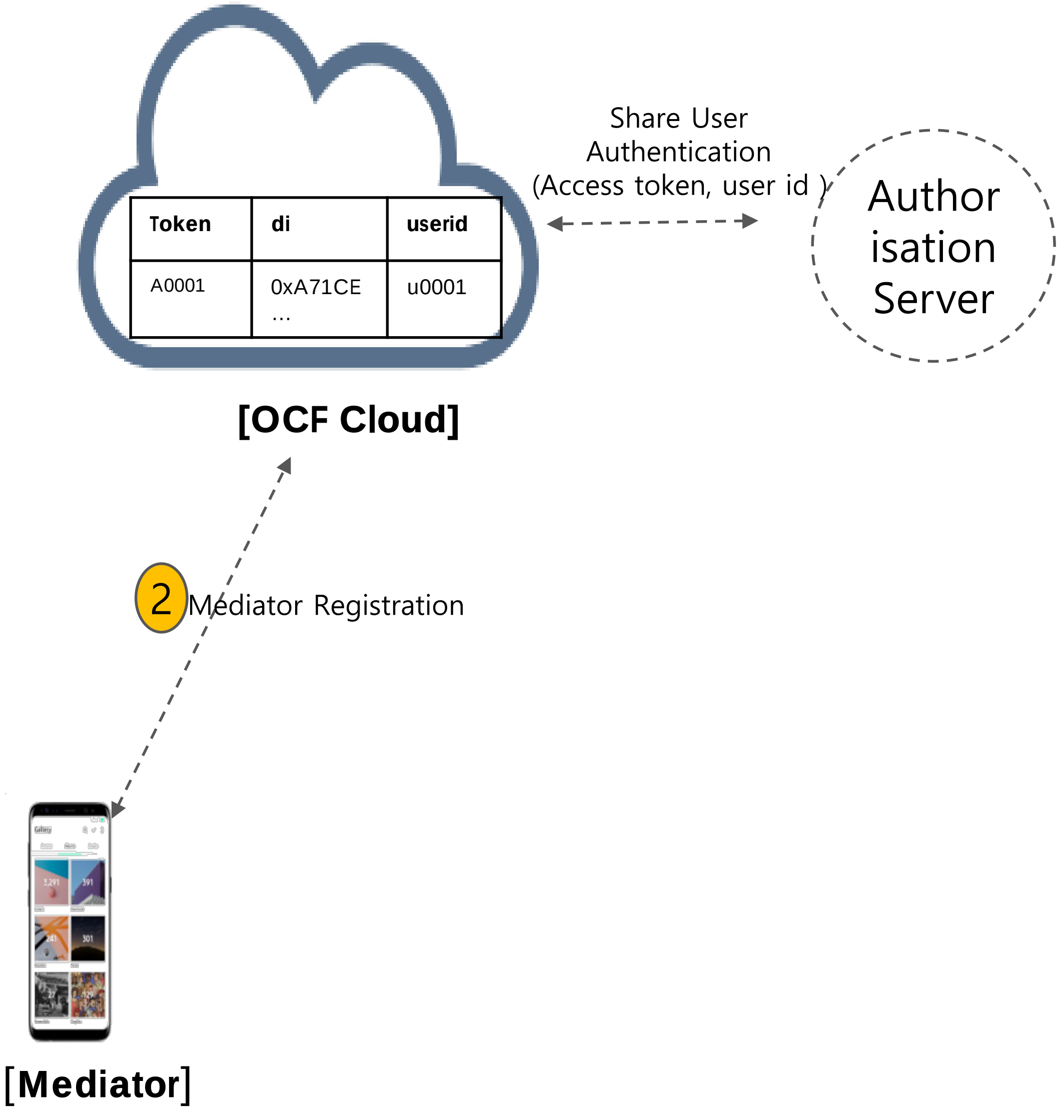
Ⅰ. OCF Cloud User Account Creation



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

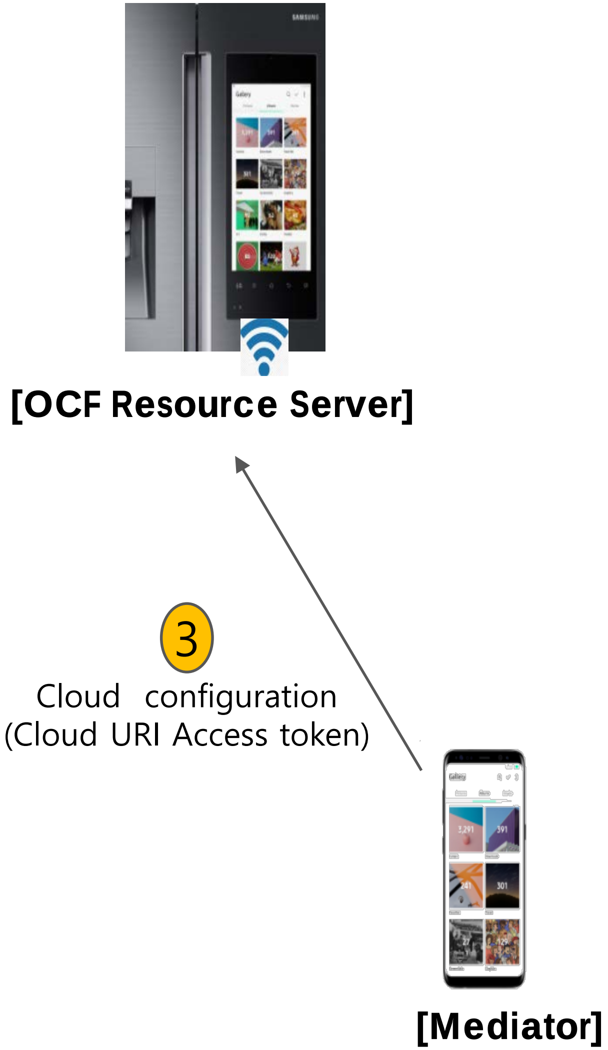
Ⅱ. Mediator registration with the OCF Cloud



Token	di	userid
A0001	0xA71CE …	u0001

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

Ⅲ. Device provisioning by the Mediator



[OCF Resource Server]

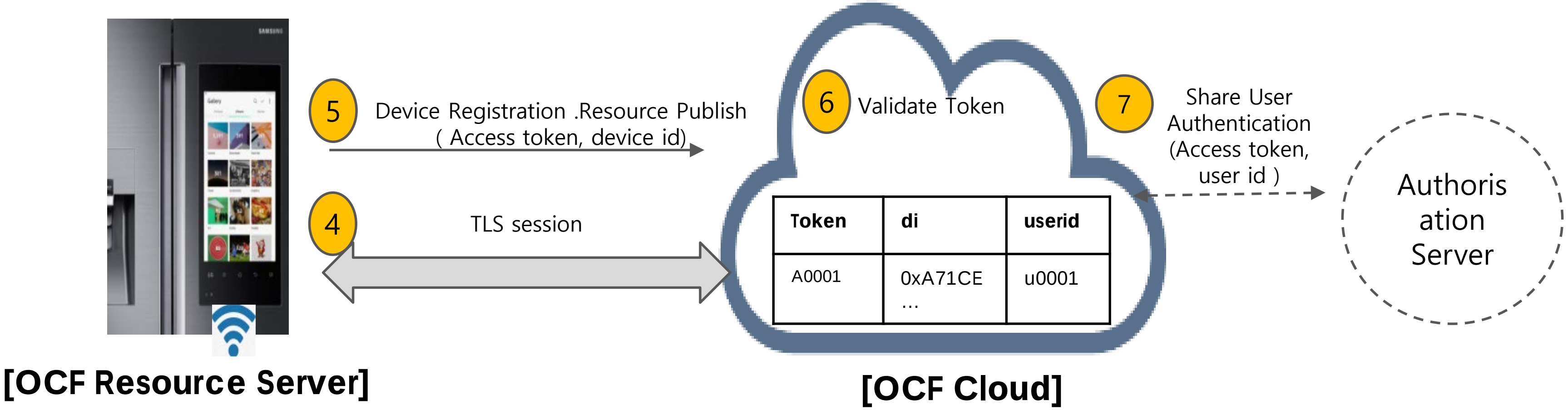
3

Cloud configuration
(Cloud URI Access token)

[Mediator]

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

Ⅳ. Device registration with the OCF Cloud



Token	di	userid
A0001	0xA71CE …	u0001

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 d.di value 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).

Ⅷ. Refreshing connection with the OCF Cloud

1. When (or before) the "expiresin" timer expires, the Device should refresh its token by sending an UPDATE request to the Token Refresh Resource that includes:

   1) "di"

   2) "uid"

   3) "refreshtoken“

2. The OCF Cloud responds with a new

   1) "accesstoken"

   2) "refreshtoken"

   3) "expiresin"

Ⅸ. Closing connection with the OCF Cloud

1. 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 UPDATE response
   - "login": false

Ⅹ. Deregistering with the OCF Cloud

1. The Device sends a DELETE request message to the Account Resource which includes: "accesstoken“, "di"
2. 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

OCF Specification Overview
Core Extension: Wi-Fi Easy Setup

OCF 2.0 Release

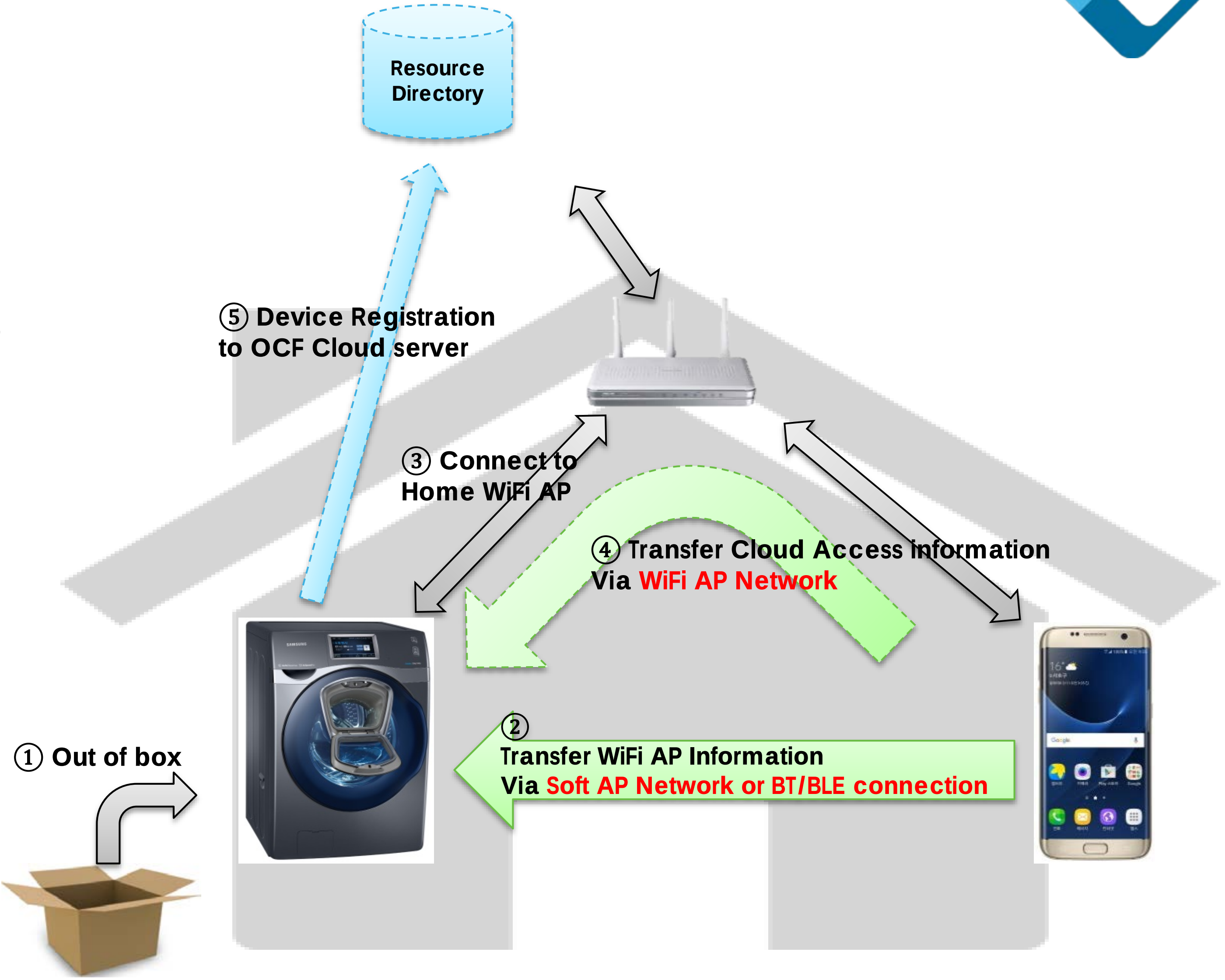
Overview

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

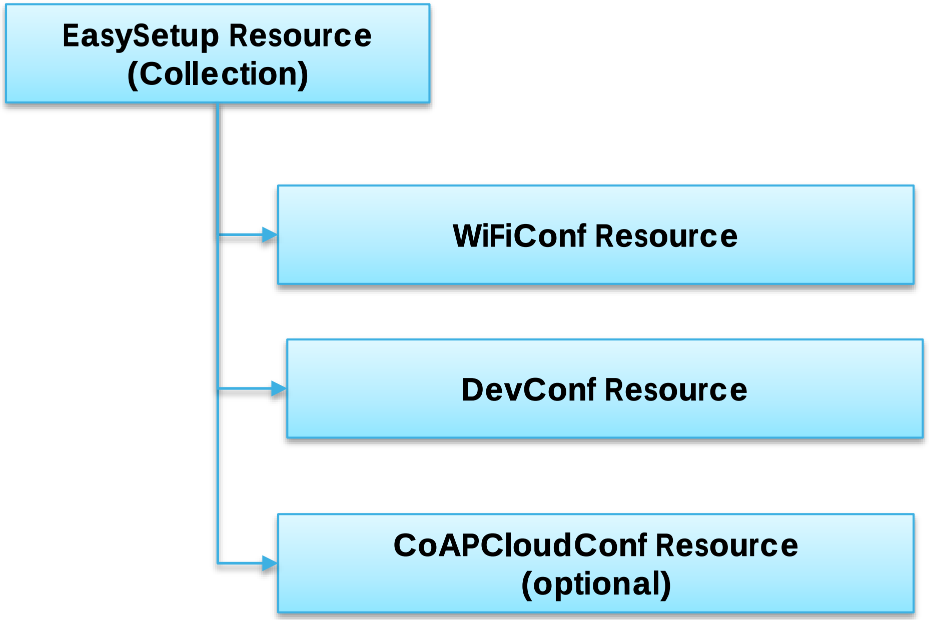

Resource Directory
⑤ Device Registration to OCF Cloud server
③ Connect to Home WiFi AP
④ Transfer Cloud Access information Via WiFi AP Network
② Transfer WiFi AP Information Via Soft AP Network or BT/BLE connection
① Out of box

Roles & Definitions

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

Resource Model - Structure

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



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

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

Easy Setup – Wi-Fi Conf. Resource

- Contains Wi-Fi-related properties

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

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

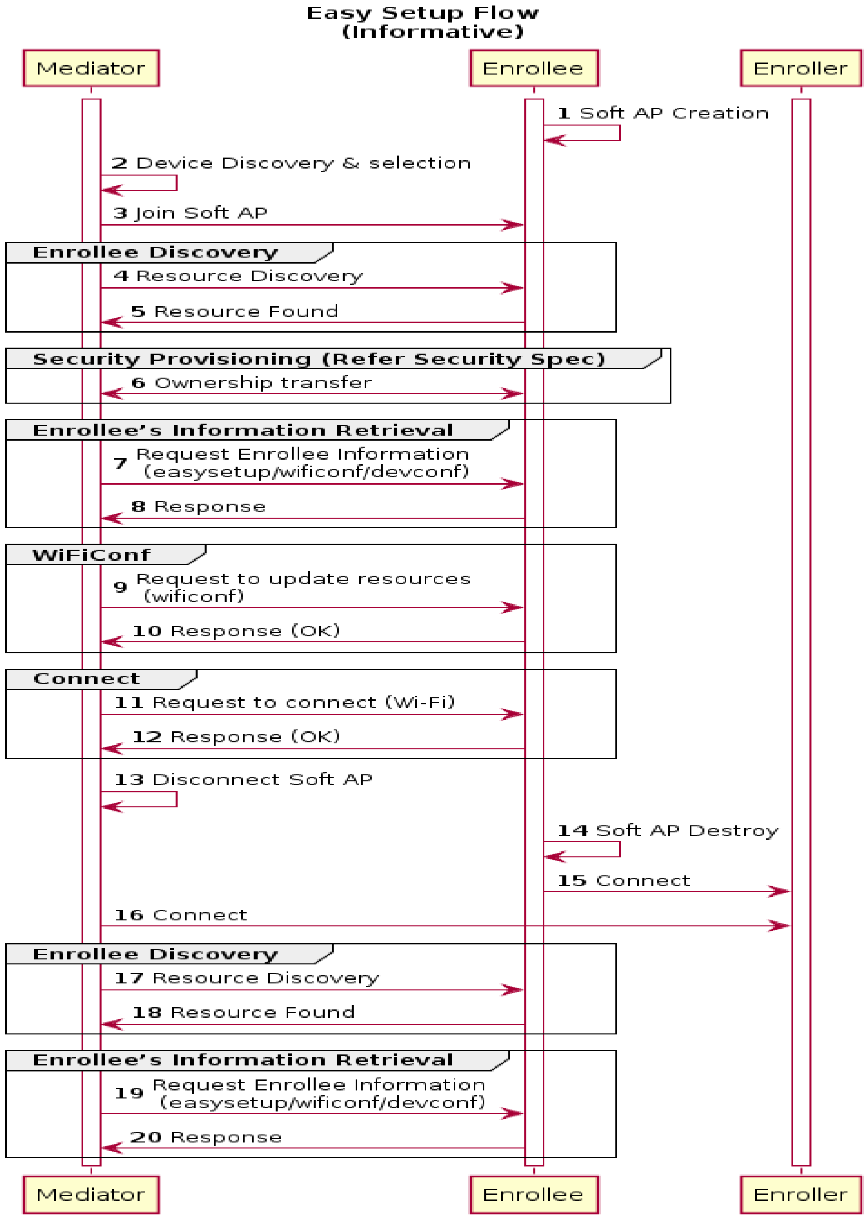
Easy Setup – Dev Conf. Resource

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

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

Property	Property Name(key)	Value Type	Value Rule	Access Mode	M / O	Description
Device Name	dn	one of: string or array of object		R	M	Indicates a pre-configured device name in language indicated by 'dl' in /oic/con. or An array of objects where each object has a 'language' field (containin g 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 d uring easy-setup process.

Example Easy Setup Flow (informative)



Easy Setup Flow
(Informative)

Mediator
Enrollee
Enroller

1 Soft AP Creation

2 Device Discovery & selection

3 Join Soft AP

Enrollee Discovery

4 Resource Discovery

5 Resource Found

Security Provisioning (Refer Security Spec)

6 Ownership transfer

Enrollee's Information Retrieval

7 Request Enrollee Information (easysetup/wificonf/devconf)

8 Response

WiFiConf

9 Request to update resources (wificonf)

10 Response (OK)

Connect

11 Request to connect (Wi-Fi)

12 Response (OK)

13 Disconnect Soft AP

14 Soft AP Destroy

15 Connect

16 Connect

Enrollee Discovery

17 Resource Discovery

18 Resource Found

Enrollee's Information Retrieval

19 Request Enrollee Information (easysetup/wificonf/devconf)

20 Response

Mediator
Enrollee
Enroller

Step1: Enrollee enables SoftAP

Steps 2-3: Mediator connects via the SoftAP

Enrollee Discovery: Steps 4-5:

Mediator discovers the Enrollee OCF Resources

Security Provisioning: Step 6:

Ownership Transfer

Enrollee Information Retrieval: Steps 7-8:

Mediator Retrieves Configuration Resources

Wi-Fi Configuration: Steps 9-10:

Mediator Updates Configuration Resources

Network Connect: Steps: 11-16

Mediator instructs Enrollee to connect to configured Wi-Fi

SoftAP disconnect and disablement

Enrollee Discovery and Retrieval: Steps 17-20:

Mediator discovers via Wi-Fi network

OPEN CONNECTIVITY FOUNDATION®

OCF Specification Overview
Security Specification

OCF 2.0 Release

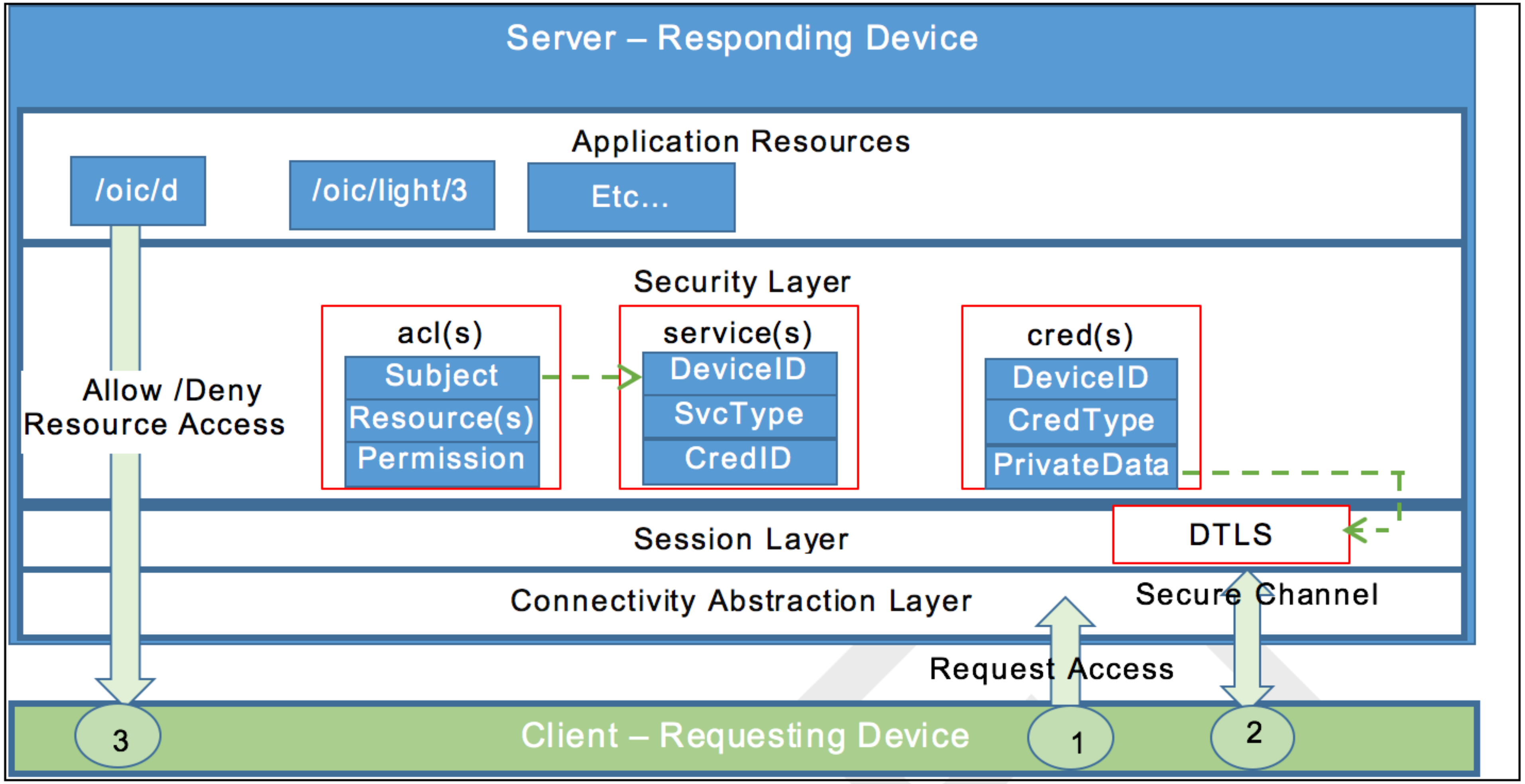
OCF Security Summary

- 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.
- **Secure Virtual Resources (SVRs)** are special security resources with severely restricted permissions and access management.

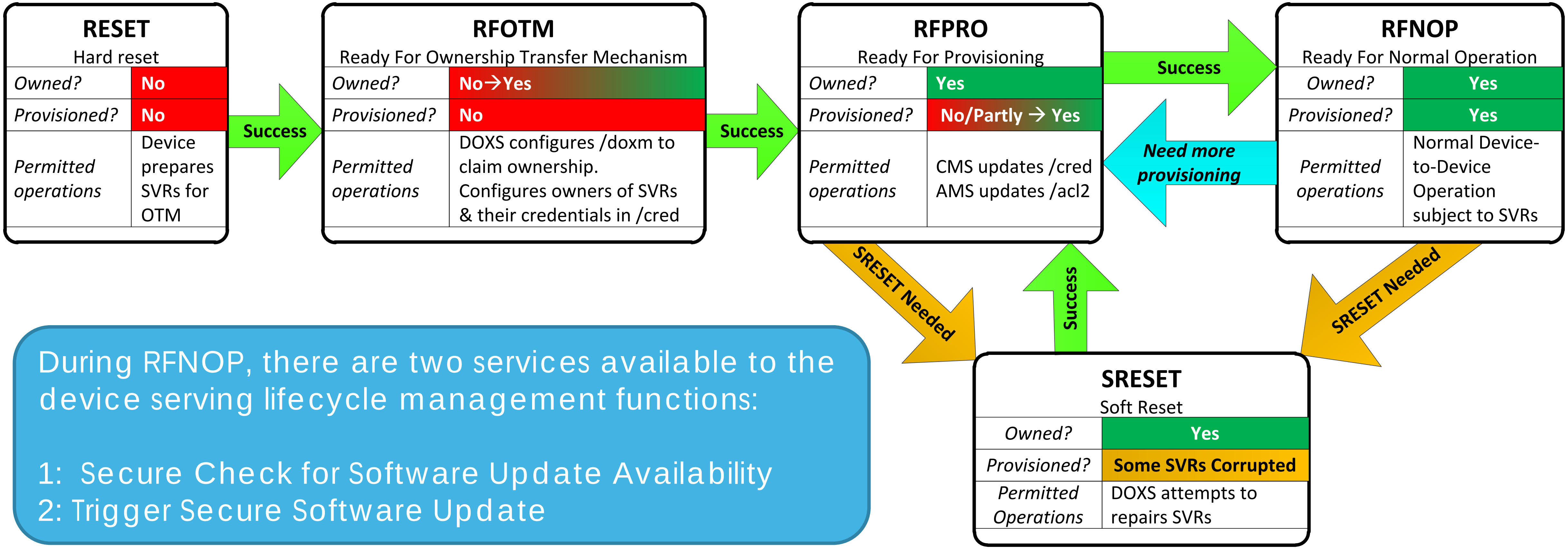
How OCF Security Protects Device Resources:



Simplified Onboarding Sequence

- *Unowned Device boots*
- **Discovery (unsecured):**
  - DOXS sends multicast to discover unowned devices no TLS
  - Unowned devices reply, including list of supported OTMs no TLS
- **Ownership Transfer:**
  - DOXS selects and configures this OTM to the new device no TLS
  - DOXS & unowned Device perform OTM, inc. TLS handshake TLS
  - DOXS configs SVRs to authorize itself, CMS and AMS TLS
  - *Device is now owned!*
- **Provisioning:**
  - CMS provisions credentials, AMS provisions access policies TLS
  - *Device is now provisioned and can commence normal operation*
- **Normal Operation:** TLS or no TLS
  - *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→Yes
Provisioned?	No
Permitted operations	DOXS 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	DOXS attempts to repairs SVRs

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

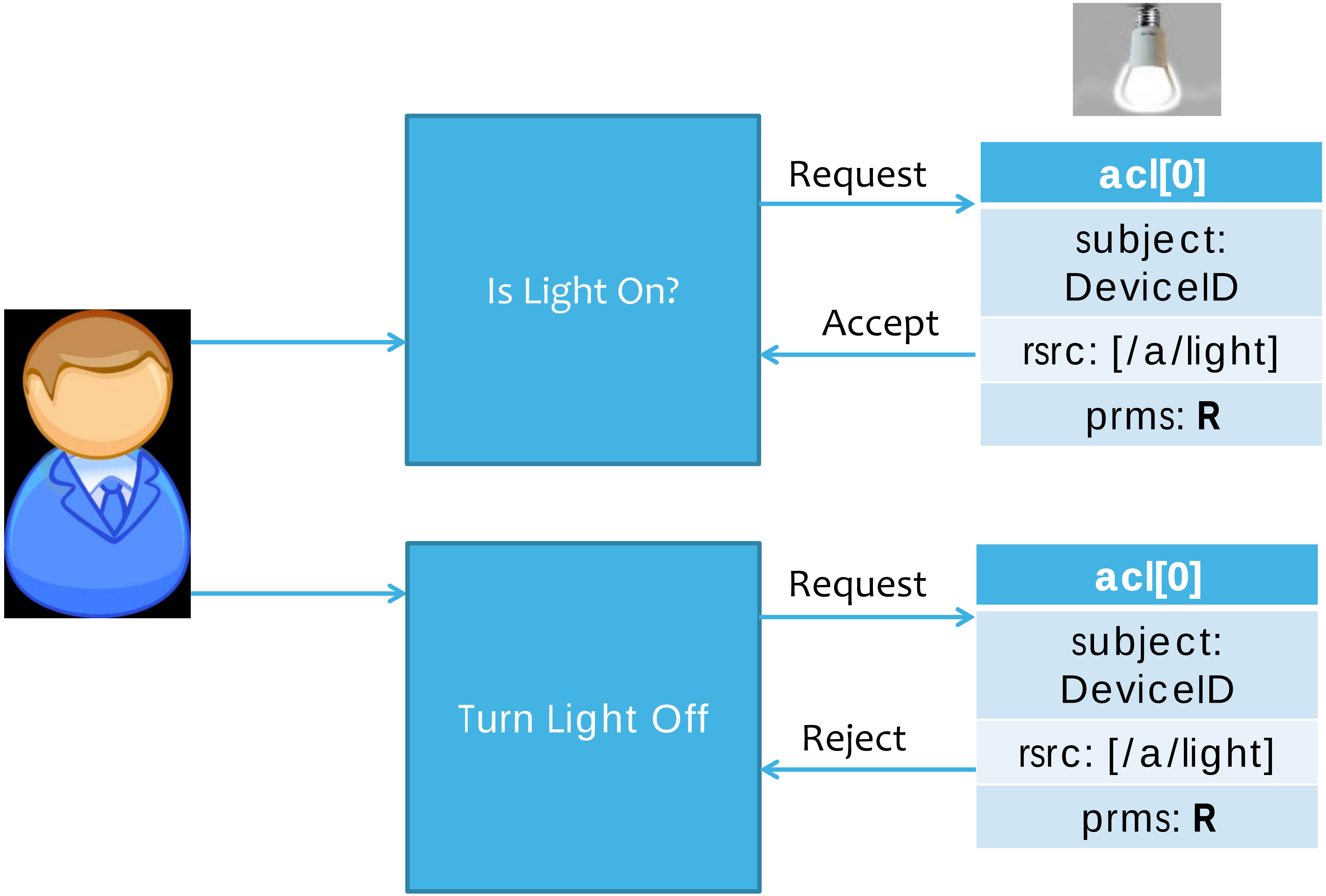
1: Secure Check for Software Update Availability
2: Trigger Secure Software Update

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

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.
- Missing credentials could be procured from a CMS
- Credentials may have an expiration period
  - Expired credentials can be refreshed

Access Control



Is Light On?

Request

Accept

acl[0]

subject: DeviceID

rsrc: [/a/light]

prms: **R**

Turn Light Off

Request

Reject

acl[0]

subject: DeviceID

rsrc: [/a/light]

prms: **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
  - Subject ID is used to verify identity of the OCF Devices and can be matched to ACLs
- Access Control List (/oic/sec/acl) manages the Access Control Entry for the Resource Server
  - Access Manager ACL (/oic/sec/amacl) Resource specified an AMS to enforce ACL
  - Signed ACL (/oic/sec/sacl) Resource to sign ACL policies

Security Virtual Resource (SVR)

oic.r.acl2 Resource

aclist2
rowneruuid

oic.r.acl Resource

aclist
rowneruuid

oic.r.amacl Resource

resources

oic.r.sacl Resource

aclist2
signature

oic.r.doxm Resource

oxm
oxmsel
sct
owned
deviceuuid
devowneruuid
rowneruuid

oic.r.cred Resource

creds
rowneruuid

oic.r.pstat Resource

dos
isop
cm
tm
om
sm
rowneruuid

oic.r.roles Resource

roles

oic.r.crl Resource

crlid
thisupdate
crldata

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.

Best Practices and Attestations

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

OPEN CONNECTIVITY FOUNDATION®

OCF Specification Overview Bridging and Bridges

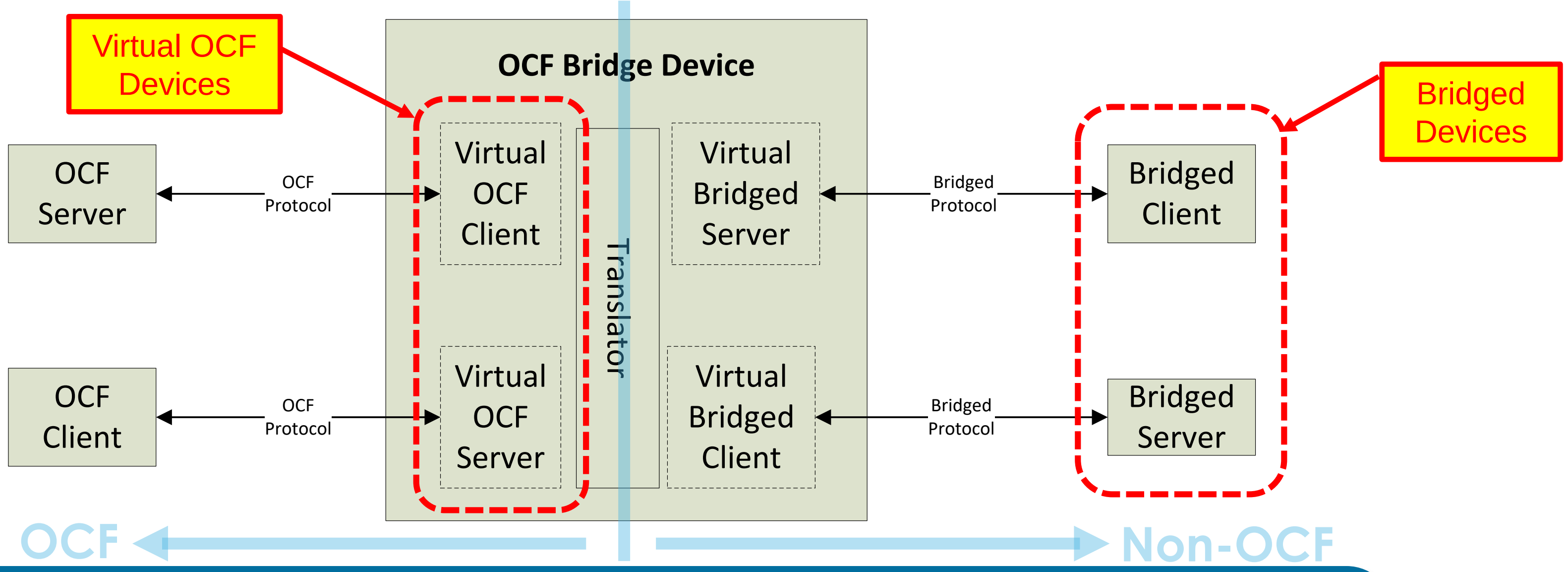
OCF 2.0 Release

Bridging Specification

- Specifies a framework for bi-directional translation between devices in OCF and non-OCF ecosystems.
  - Bi-directional translation: not only exposes non-OCF server to OCF, but also exposes OCF server to non-OCF ecosystem
- Specifies general requirements for translation between OCF and non-OCF ecosystems
  - Requirements for resource discovery, message translation, security, and handling of multiple bridges.
- Specifies specific requirements for translation between OCF and AllJoyn ecosystems
  - Requirements for mapping core resources, propagating errors, and algorithmically translating custom resource types.
  - Refers to OCF to AllJoyn Mapping specification for translating well-known resource types.

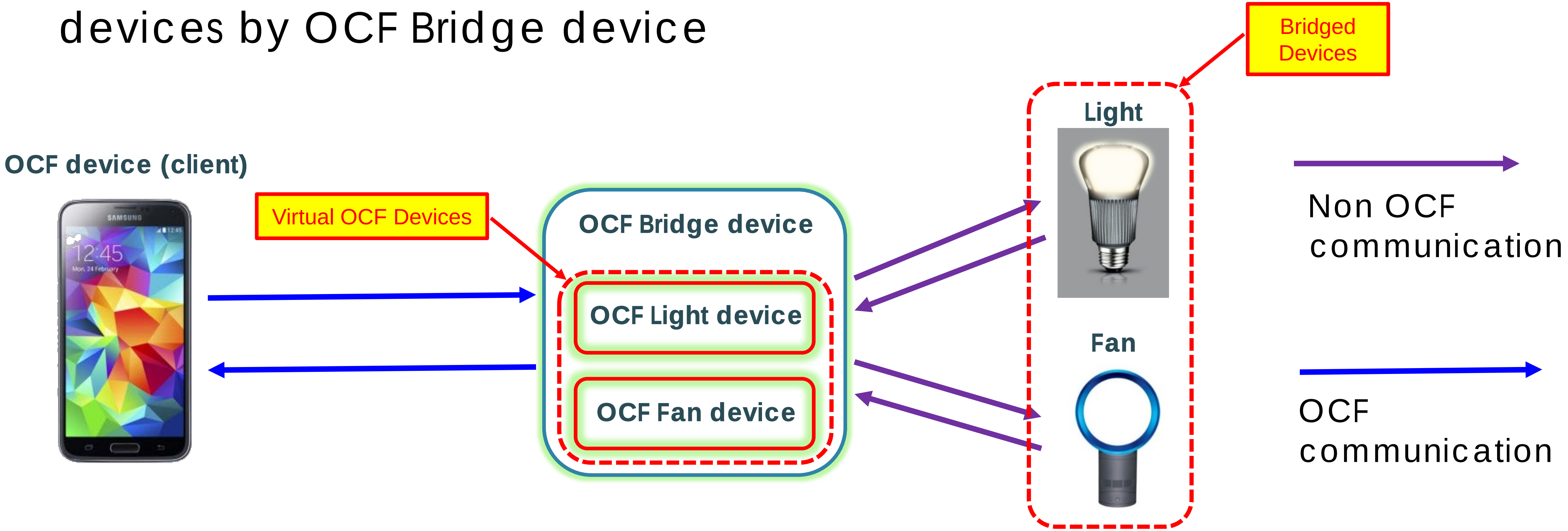
OCF Bridge – Definition

- 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 non-OCF network
- The bridged devices themselves are out of the scope of OCF.
- The only difference between a ‘regular’ OCF device and a virtual OCF device is that the latter is encapsulated in an OCF Bridge device.
- An OCF Bridge device is indicated on the network with an “rt” of “oic.d.bridge”. When such a device is discovered, its discoverable resources would describe the devices that it bridges.

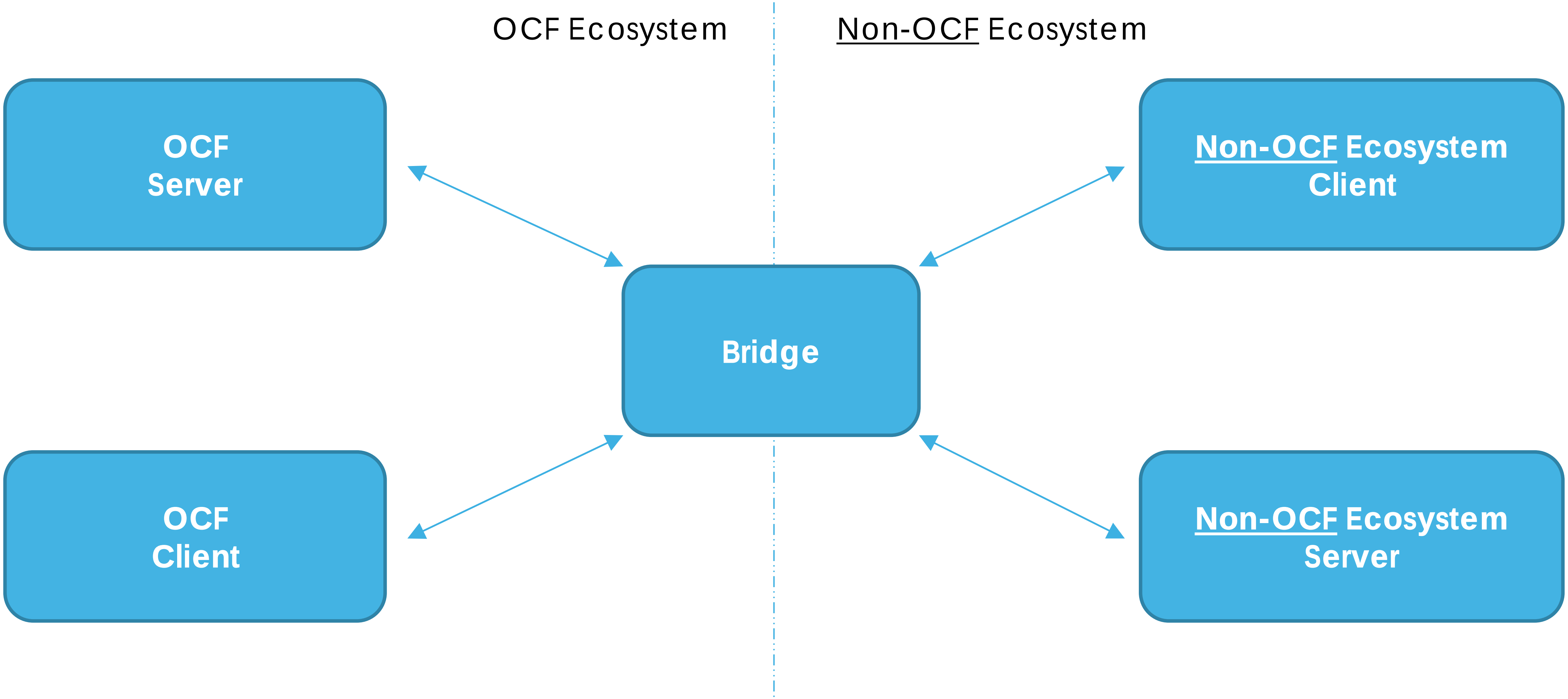


OCF Bridge – Definition

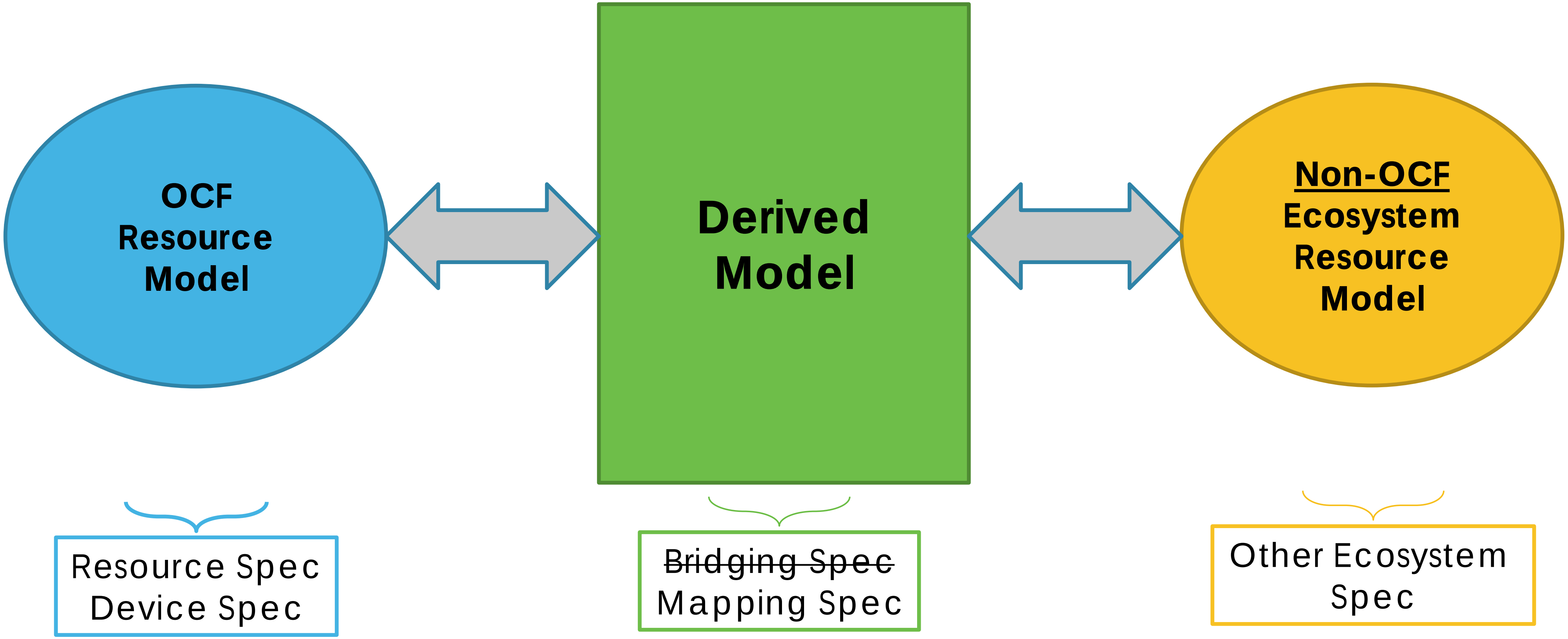
- Bridging example
  - Light and Fan are non-OCF devices
  - Light and Fan are exposed as “Virtual OCF Devices” to OCF devices by OCF Bridge device



Bridging Concept – Bidirectional Operation



Bridging Concept – Data Model



OCF Resource Model

Derived Model

Non-OCF Ecosystem Resource Model

Resource Spec
Device Spec

~~Bridging Spec~~
Mapping Spec

Other Ecosystem Spec

Bridging Security

- OCF Bridge needs to be a trusted entity as it translates message payloads.
- OCF 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 OCF Bridge must implement the security requirements of the ecosystem that it is connected to.
- Bridging specifies mechanisms to selectively block communications between the OCF Bridge and OCF devices and between the OCF Bridge and bridged devices. This fine-grained control enables an administrator to control communications across ecosystems that may not have similar security capabilities.

OPEN CONNECTIVITY FOUNDATION®

Bridges:
Derived Modeling – OCF to AllJoyn Mapping

Overview

Overview

- Models the interworking between OCF and AllJoyn
- Makes use of derived model syntax as defined (with some small changes) in the OCF White Paper here: https://www.iab.org/wp-content/IAB-uploads/2016/03/OCF-Derived-Models-for-Interoperability-Between-IoT-Ecosystems_v2-examples.pdf
- Predicated on OCF being the superset model; so any Device Types and Resource Types (as equivalents to AllJoyn interfaces) that were missing from OCF were defined in the equivalent OCF Specifications.
- Defines the mapping in terms of:
  - Device Type equivalency
  - Resource <-> Interface equivalency
  - Detailed Property by Property mapping on a per Interface Basis (Derived Models)

Derived Model Syntax

- Derived models use standard JSON schema syntax. Fundamentally, derived models provide a conversion mapping between OCF data models and the data models in 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[illegible]
          "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

Device Type Equivalency

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

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

Interface to Resource Mapping

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

OPEN CONNECTIVITY
FOUNDATION®

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 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 RESTful API Modelling Language (RAML) and OpenAPI (formerly known as 'Swagger 2.0')

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

Defined Resource Types – OIC 1.1 to OCF 1.3.1

Resource Types	Use Case
Air Flow	Indoor Environment Control
Air Flow Control	
Battery	Device Control
Binary switch	Device Control
Brightness	Lighting Control
Colour Chroma	
Colour RGB	
Dimming	
Door	Indoor Environment Control
Energy Consumption	Energy Management
Energy Usage	
Humidity	Indoor Environment Control
Icemaker	Device Control

Resource Types	Use Case
Lock	Keyless Entry
Lock Code	
Mode	Device Control
Open Level	
Operational State	
Ramp Time	Lighting Control
Refrigeration	Device Control
Temperature	Indoor Environment Control
Time Period	Device Control

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.

Defined Resource Types – OIC 1.1 to OCF 1.3.1

Resource Type	Use Case
Audio	TV, Home Entertainment
Auto Focus	IP Camera
Auto White Balance	IP Camera
Automatic Document Feeder	Scanner Support
Button	Device Control
Colour Saturation	IP Camera
DRLC	Smart Energy
Energy Overload	Smart Energy
Media	IP Camera
Media Source List	TV, Home Entertainment
Movement (Linear)	Robot Cleaner
Night Mode	IP Camera
PTZ	IP Camera
Signal Strength	Proximity

Sensor Resource Type	Use Case
Acceleration	Extended Sensor Set (for a Generic Sensor Device)
Activity Count	
Atmospheric Pressure	
Carbon Dioxide	
Carbon Monoxide	
Contact	
Glass Break	
Heart Rate Zone	
Illuminance	
Magnetic Field Direction	
Presence	
Radiation (UV)	
Sleep	
Smoke	
Three Axis	
Touch	
Water	

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.

Defined Resource Types – OIC 1.1 to OCF 1.3.1

Resource Type	Use Case
Air Quality	Indoor Environment Control
Air Quality Collection	Indoor Environment Control
Consumable	Device Control
Consumable Collection	Device Control
Delay Defrost	Energy Star
Ecomode	Device Control
Heating Zone	Device Control
Heating Zone Collection	Device Control
Selectable Levels	Device Control
Value Conditional	Notifications

Resource Type	Use Case
Battery Material	Device Control
Brewing	Device Control
Colour Space Coordinates	Lighting Control
Colour Temperature	Lighting Control
Colour Hue/Saturation	Lighting Control
Energy	Energy Star
Energy Generation	Energy Star
Foaming	Device Control
Grinder	Device Control
Liquid Level	Device Control
Vehicle Connector	Device Control
Time Stamp	Multiple

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.

New Resource Types – OCF 2.0

Resource Type	Use Case
3D Printer	Device Control
Blood Pressure	Personal Health
Body Mass Index	Personal Health
Body Fat	Personal Health
Body Fat Free Mass	Personal Health
Body Location	Personal Health
Body Location Temperature	Personal Health
Body Soft Lean Mass	Personal Health
Body Water	Personal Health
Glucose	Personal Health
Glucose Carbohydrates	Personal Health
Glucose Exercise	Personal Health
Glucose HbA1c	Personal Health
Glucose Health	Personal Health
Glucose Meal	Personal Health
Glucose Medication	Personal Health
Glucose Sample Location	Personal Health
Glucose Tester	Personal Health
ORFID Station	Smart Factory
ORFID Tag	Smart Factory
Power Source	Energy Star
Print Queue	Device Control
Pulse Rate	Personal Health
Sensor Properties	Generic Sensor Modeling
User ID	Personal Health

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.

OPEN CONNECTIVITY FOUNDATION®

Device Specification

Overview

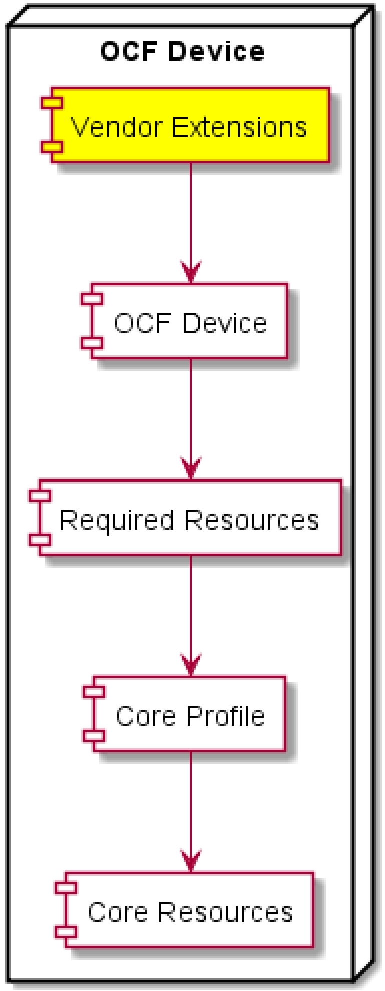
Higher Layer Specifications

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

The Device specification uses the resources defined in the resource specification

Device Specification

- Contains profiles of
  - Core specification
  - Security specification
- Contains list of OCF devices
- Each OCF device definition contains:
  - Human friendly name
  - unique identifier (rt) in the form oic.d.<thing>



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

Device Categories

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

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

Mandatory Resources per Device Type

- A vertical may specify a set of Resources that are mandatory to expose 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

Some Example Device Types

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

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

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

Friendly Name	Device Type
3D Printer	oic.d.3dprinter
Air Conditioner	oic.d.airconditioner
Air Purifier	oic.d.airpurifier
Air Quality Monitor	oic.d.airqualitymonitor
Battery	oic.d.battery
Blind	oic.d.blind
Blood Pressure Monitor	oic.d.bloodpressuremonitor
Body Scale	oic.d.bodyscale
Body Thermometer	oic.d.bodythermometer
Camera	oic.d.camera
Clothes Dryer	oic.d.dryer
Clothes Washer	oic.d.washer
Clothes Washer/Dryer	oic.d.washerdryer

Friendly Name	Device Type
Coffee Machine	oic.d.coffeemachine
Cooker Hood	oic.d.cookerhood
Cooktop	oic.d.cooktop
Dehumidifier	oic.d.dehumidifier
Dishwasher	oic.d.dishwasher
Door	oic.d.door
Electric Meter	oic.d.electricmeter
Electric Vehicle Charger	oic.d.electricvehiclecharger
Energy Generator	oic.d.energygenerator
Fan	oic.d.fan
Food Probe	oic.d.foodprobe
Freezer	oic.d.freezer
Garage Door	oic.d.garagedoor
Generic Sensor	oic.d.sensor

Friendly Name	Device Type
Glucose Meter	oic.d.glucosemeter
Grinder	oic.d.grinder
Humidifier	oic.d.humidifier
Humidifier	oic.d.humidifier
Kettle	oic.d.kettle
Light	oic.d.light
Microwave Oven	oic.d.microwave
Optical Augmented RFID Reader	oic.d.orfid
Oven	oic.d.oven
Printer	oic.d.printer
Printer (Multi-Function)	oic.d.multifunctionprinter
Receiver	oic.d.receiver

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

Friendly Name	Device Type
Refrigerator	oic.d.refrigerator
Robot Cleaner	oic.d.robotcleaner
Scanner	oic.d.scanner
Security Panel	oic.d.securitypanel
Set Top Box	oic.d.stb
Smart Lock	oic.d.lock
Smart Plug	oic.d.smartplug
Steam Closet	oic.d.steamcloset
Switch	oic.d.switch
Television	oic.d.tv
Thermostat	oic.d.thermostat
Water Heater	oic.d.waterheater
Water Valve	oic.d.watervalve

Friendly Name	Device Type
Window	oic.d.window

Items in red are new in OCF 2.0

OPEN CONNECTIVITY FOUNDATION®

References

Specification Location

Where can I find the specifications and Resource Type definitions?

OCF Specifications:

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

Resource Type Definitions

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

OPEN CONNECTIVITY
FOUNDATION®