OPEN CONNECTIVITY FOUNDATION™

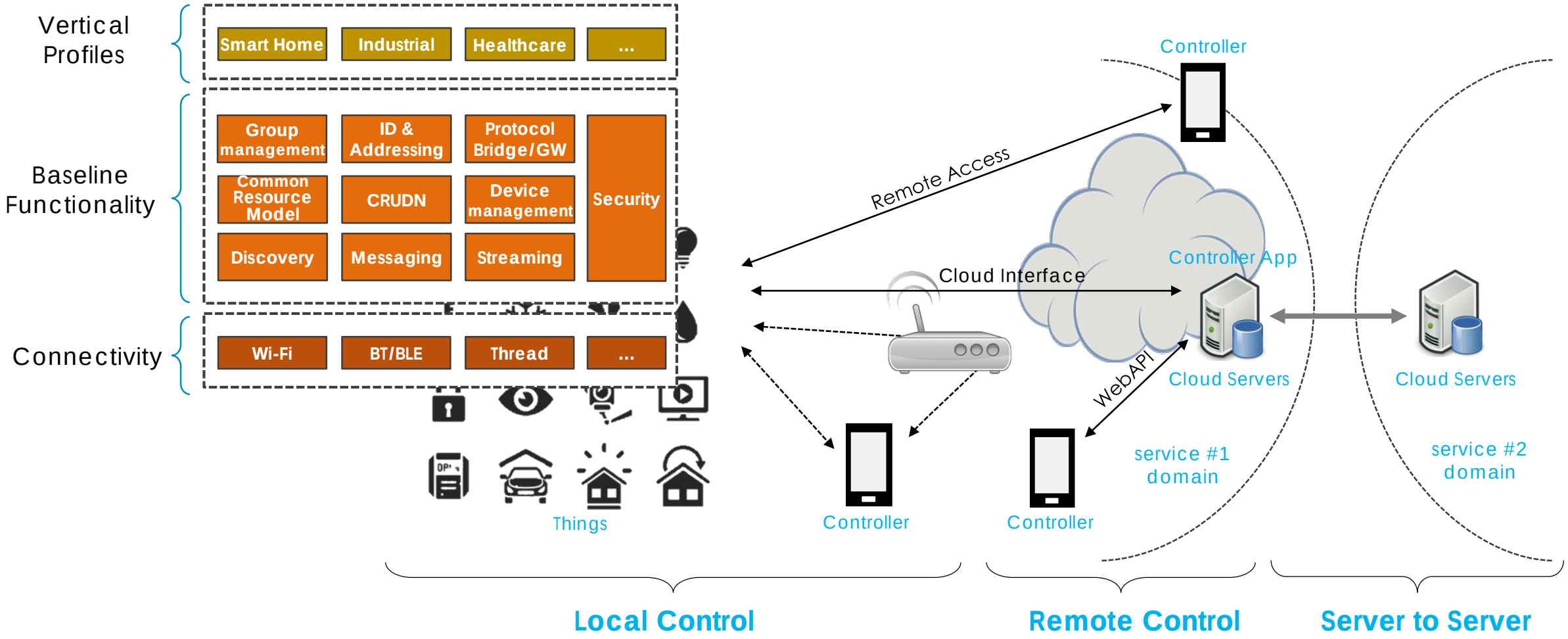
OCF Specification Introduction and Overview

July 2017

Table of Contents

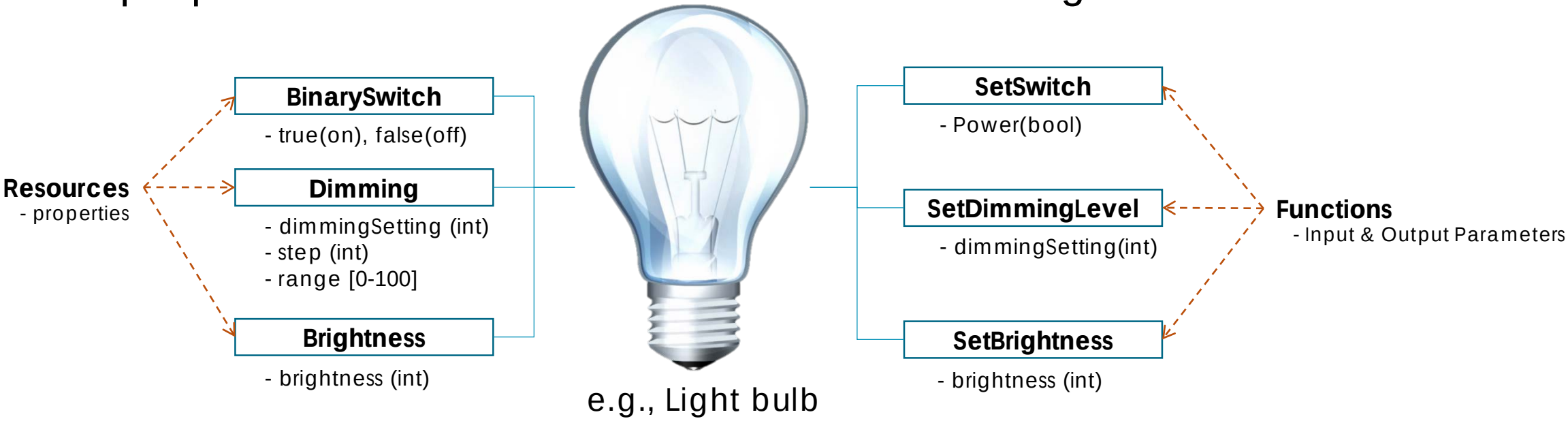
Technical Principles for an Internet of Things Ecosystem

Scope of IoT



Approaches to definition of various Things

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



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

- (Verbs + Objects)
- RPC model

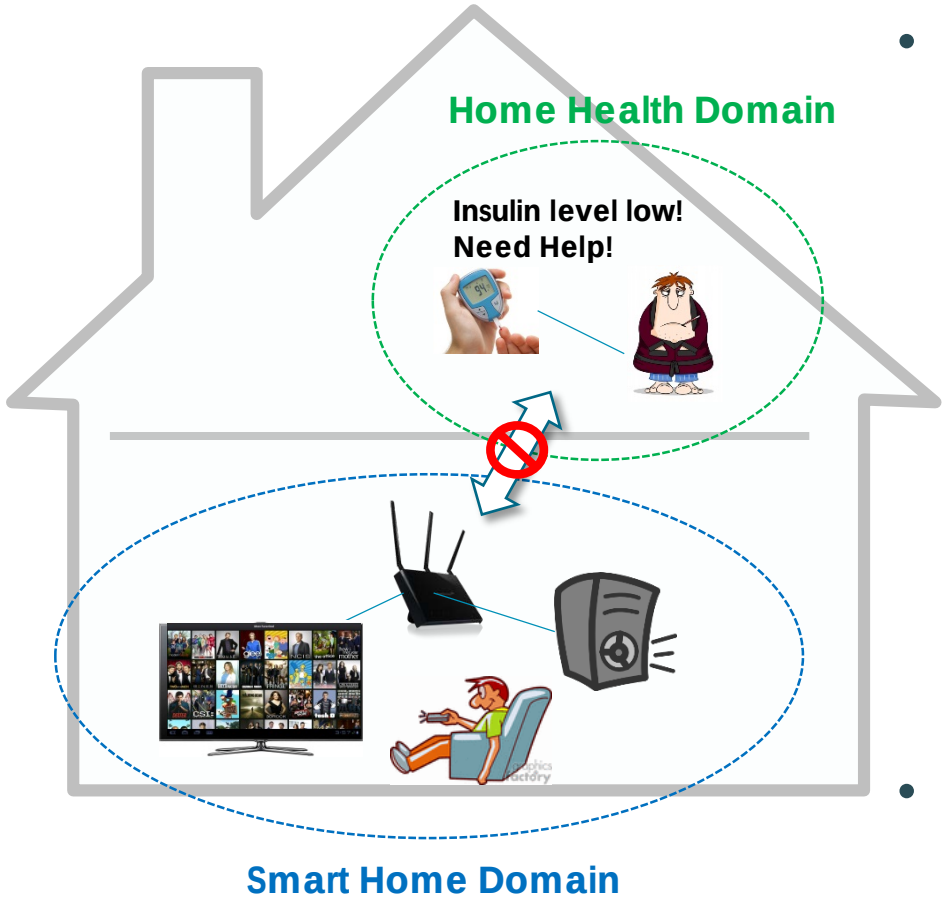
Support of Constrained Things

Class 2 Devices as Defined by RFC 7228

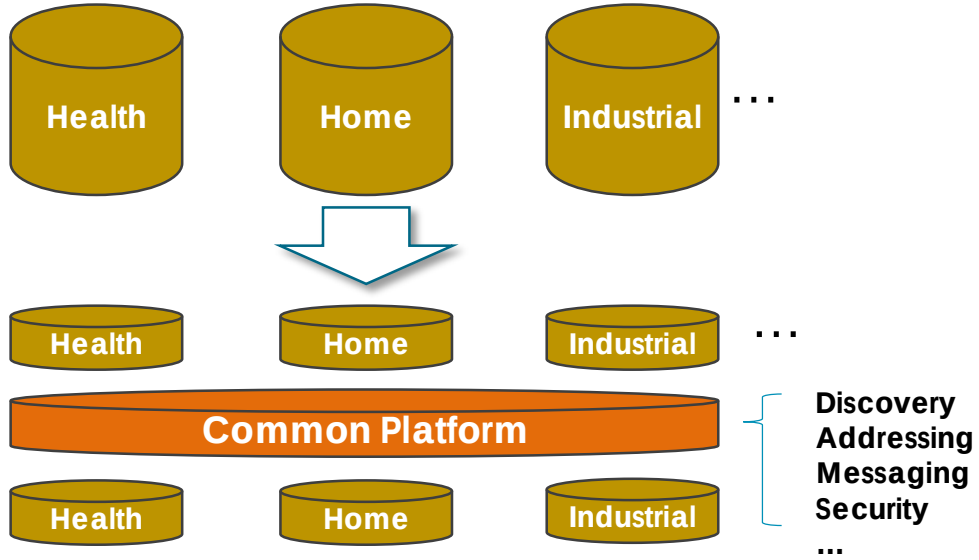
- Less overhead/ Less Traffic
  - Minimize CPU Load, Memory impacts, Traffic and Bandwidth
    - Compact header
    - Binary protocol
    - Compressed encoding of payload

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

Support of Multiple Verticals



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

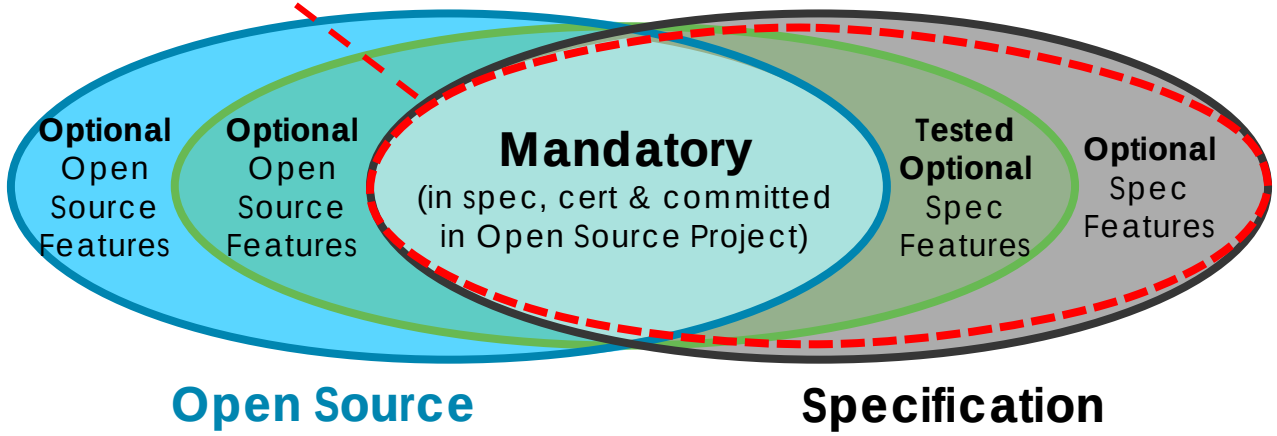


- A common platform provides a foundation for vertical services to collaborate and interwork by providing common services and data models

Conformance & Certification

- Conformance test - Each device proves conformance to specifications

- Certification Scope



Licensing

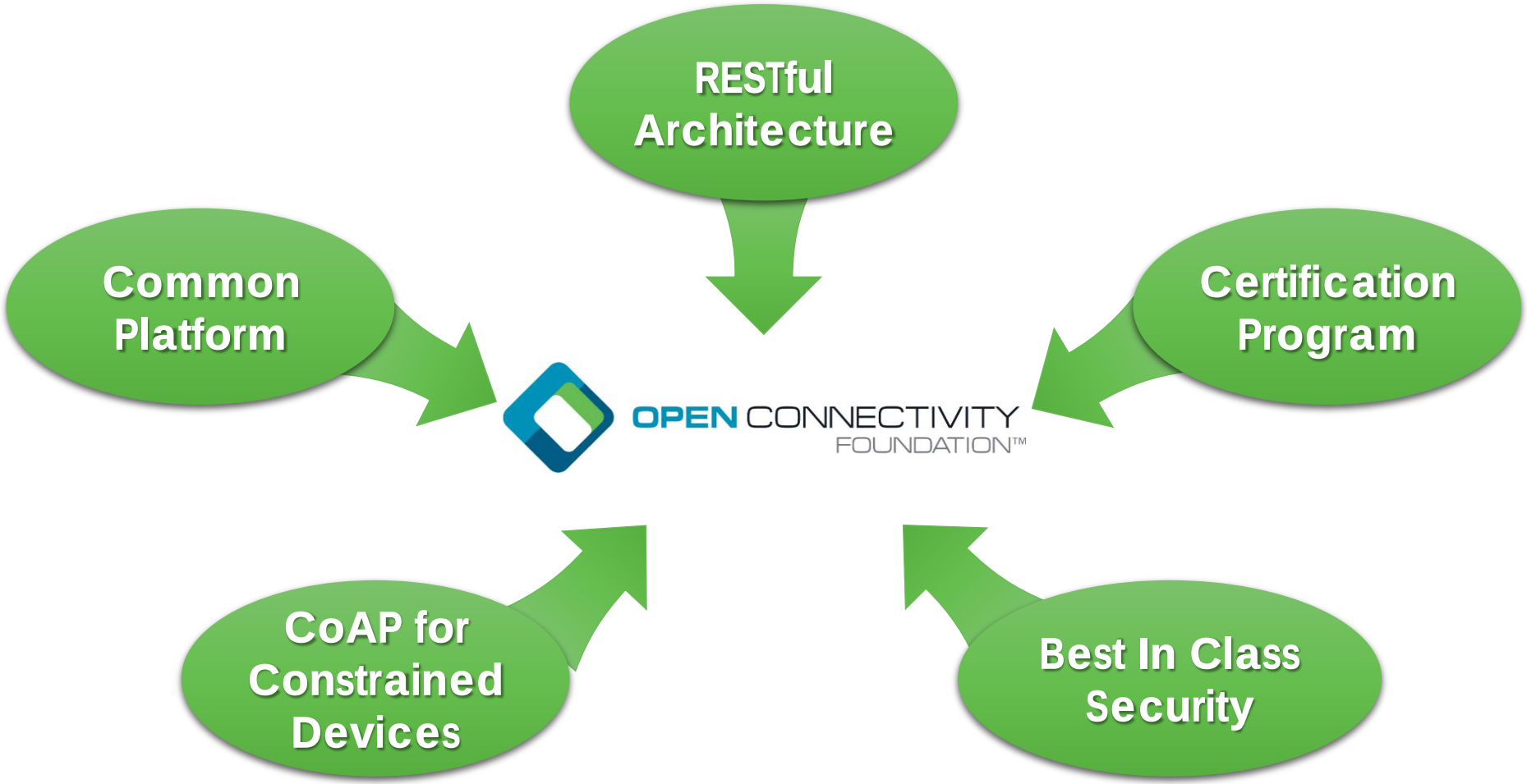
- For Intellectual Property Rights(IPR) Policy : RAND-Z > RAND >> no IPR policy
- For Open Source : Apache 2.0 > Internet Systems Consortium (ISC)

- 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



OCF Areas of Technology Development

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

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)

Specification Location

Where can I find the specifications and Resource Type definitions?

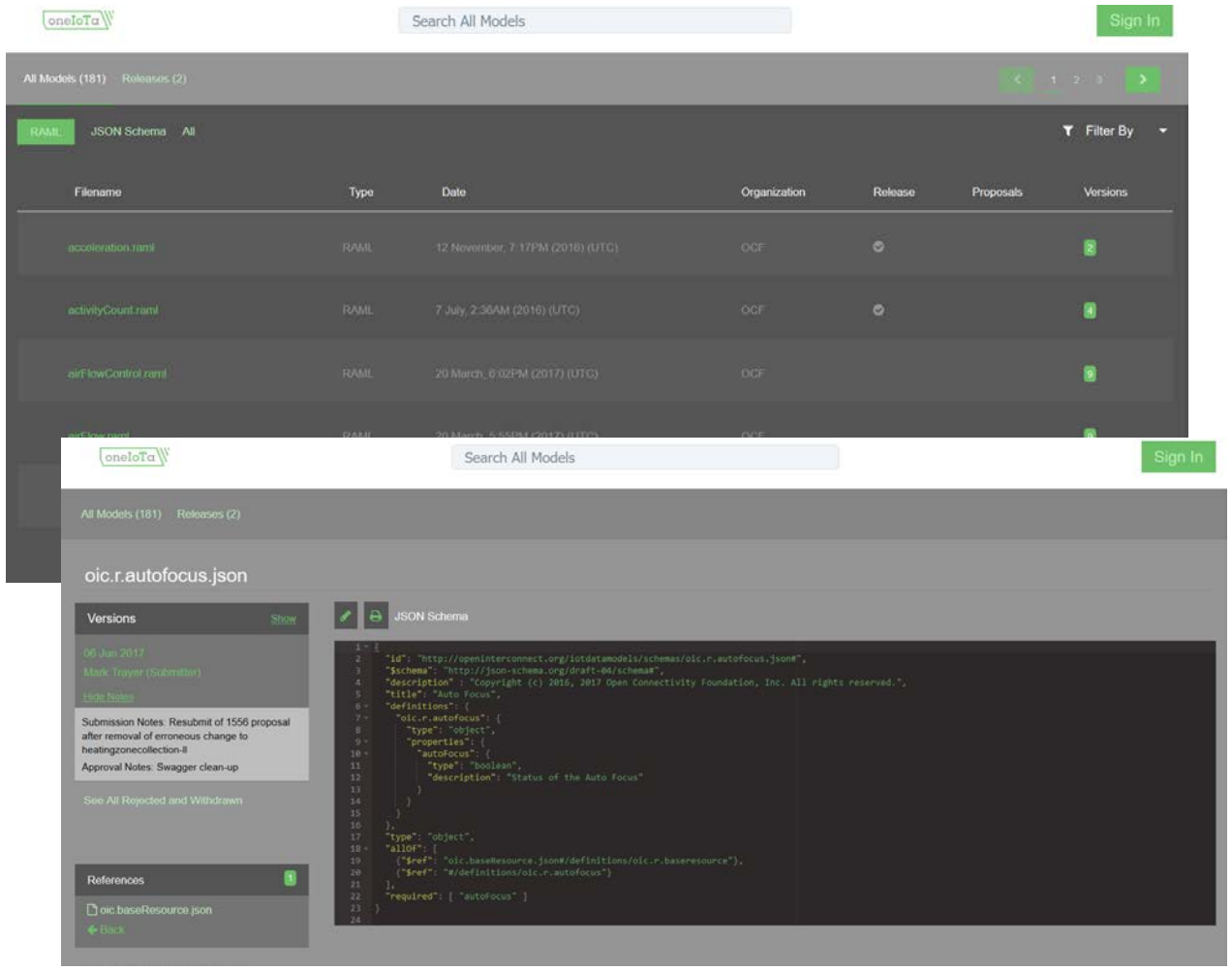
OCF Specifications:

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

Resource Type Definitions

- Core Resources: https://github.com/openconnectivityfoundation/core
- 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



- 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 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

Entity handler

Physical entity

Client

Server

REST Architecture Style

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

Request

Response

CRUD & N operation

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

Resource (representation)

Resource (representation)

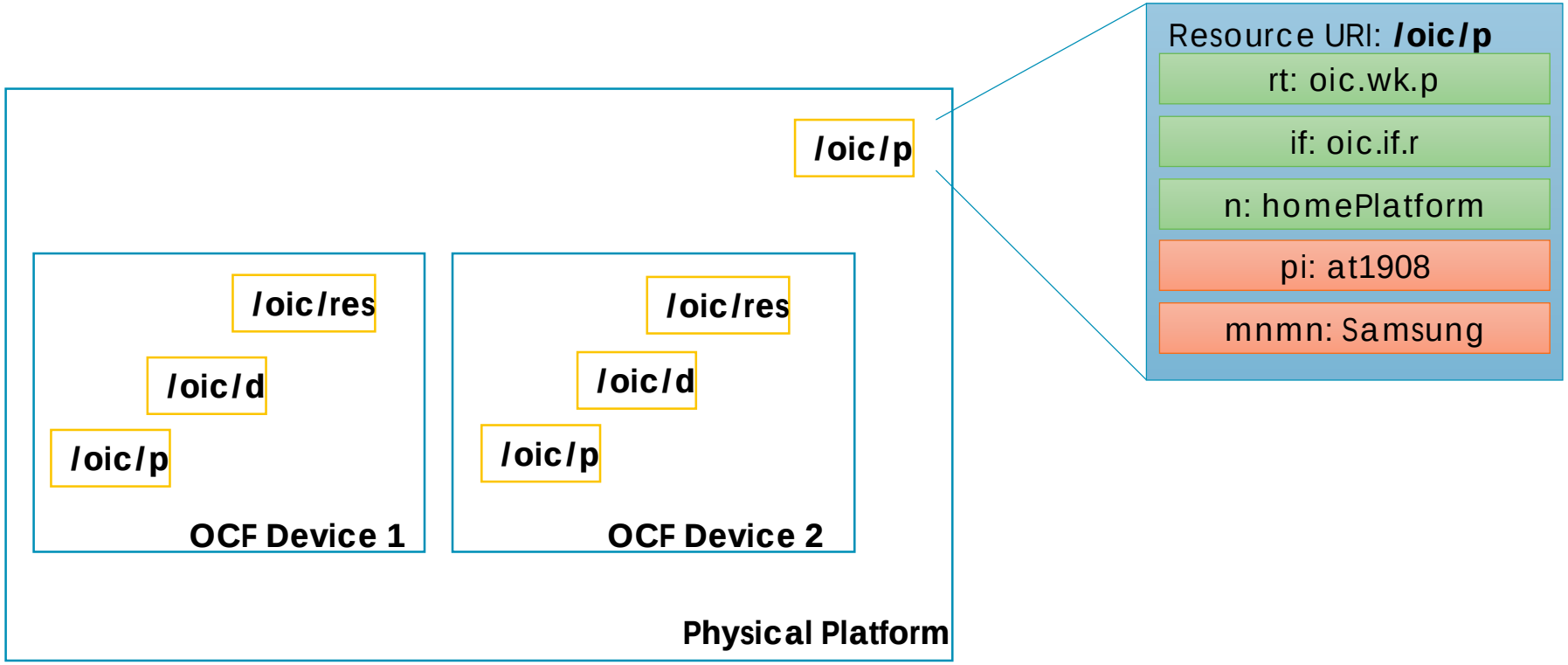
- RESTful Architecture (Representational State Transfer)
  - Resource based operation
    - Real world 'entity' is represented as 'Resource'
  - Resource manipulation via Request/ Response: CRUDN

OCF Roles

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

Organization of an OCF Device

- OCF Device concept

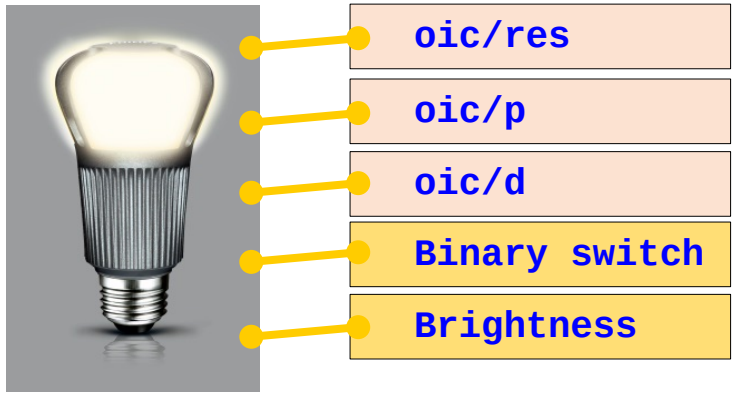


Device example: light device (oic.d.light)

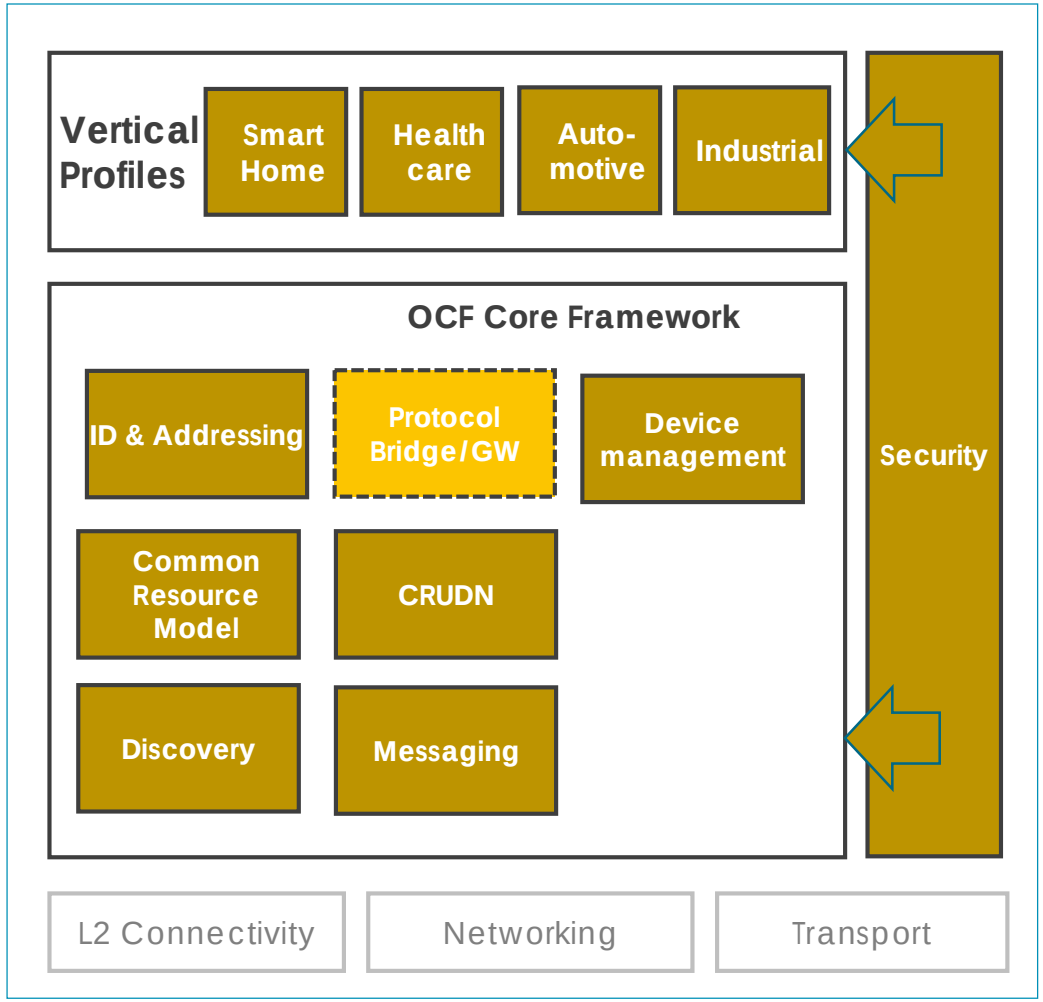
- Example overview
  - Smart light device with i) binary switch & ii) brightness resource
- Device type: Light device (oic.d.light) [Defined by the domain]
- Associated resources
  - Mandatory Core resources: oic/res, oic/p, oic/d
  - Mandatory Security Resources (not shown in the diagram)
  - Device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



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)

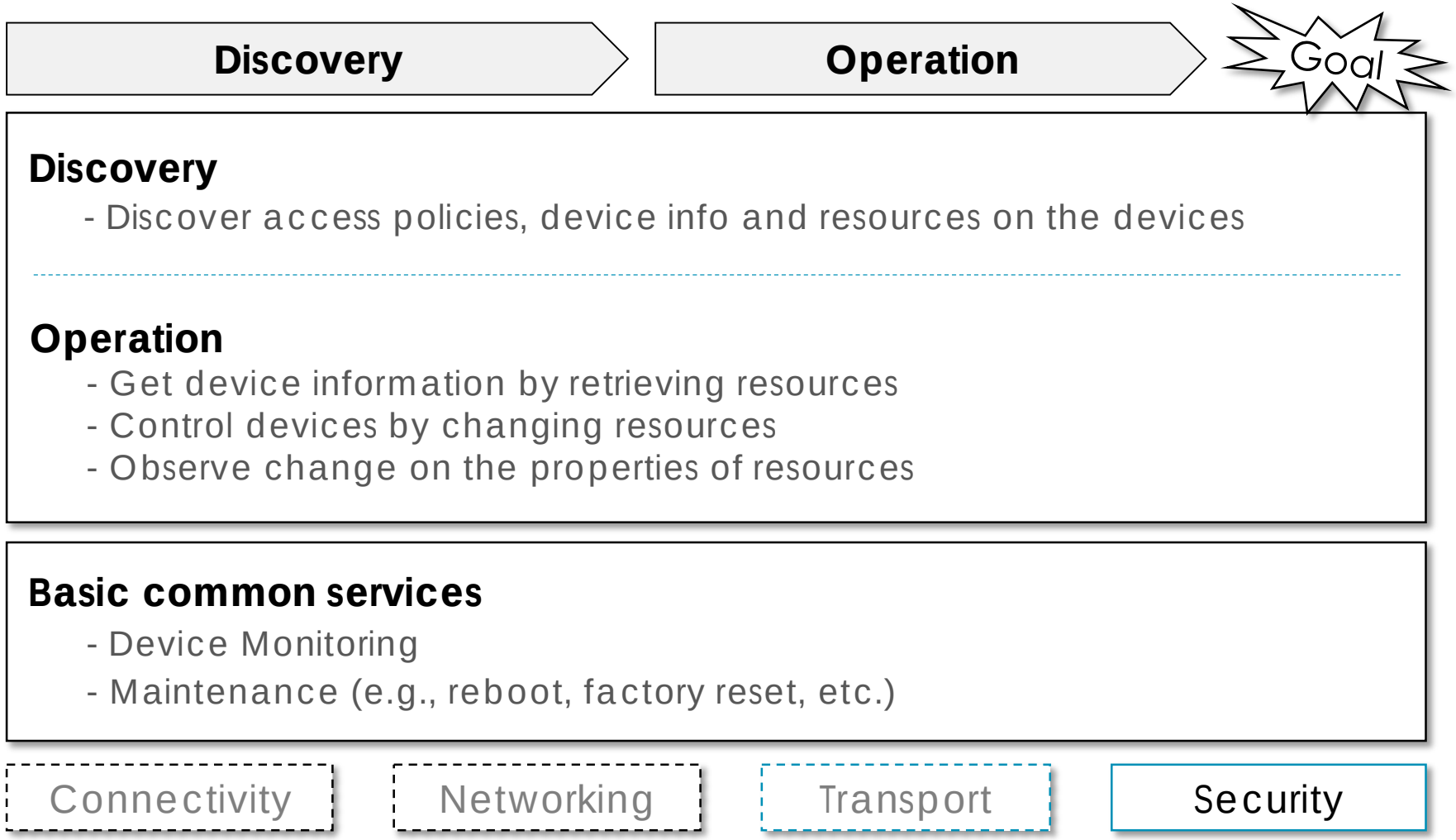
④ **CRUDN:** Simple Request/Response mechanism with Create, Retrieve, Update, Delete and Notify commands

⑤ **ID & Addressing:** OCF IDs and addressing for OCF entities (Devices, Clients, Servers, Resources)

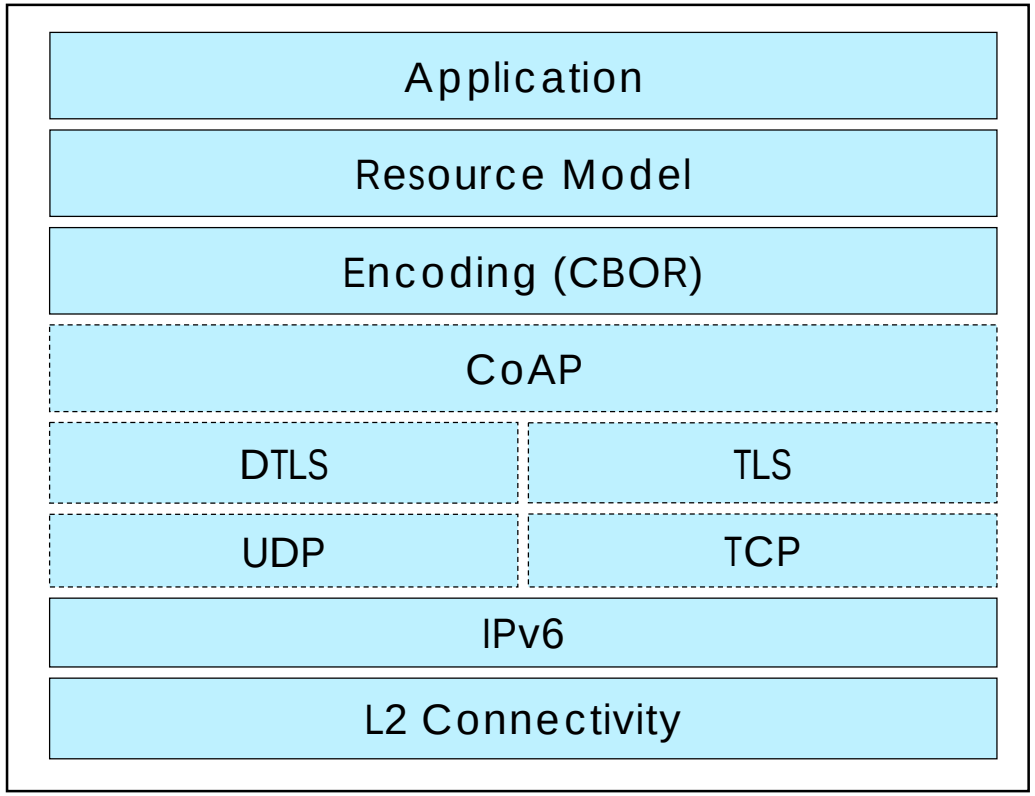
⑥ **Protocol Bridge/GW**: Handled by the Bridging Spec with some implications on the Core

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

OCF Core Framework Basic Operation



Protocol Stack



OCF Stack

End point 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

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

Collection Resources

- An OCF Resource that contains one or more references (specified as OCF Links) to other OCF Resources 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 collections

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: Goal

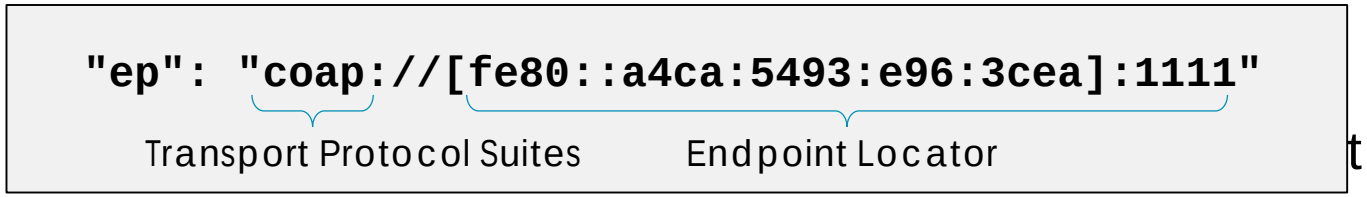
- Leave the current way of working intact: e.g. use RAML+JSON as is: use it as input for the swagger definition that will go on the wire.
- Same restrictions as already investigated and part of the:
  - 1 file to be transferred: e.g. definition should include
    - All end points, methods, query parameters, payload definitions
  - Same kind of negotiation to download the file
- Only this time it will be a swagger2.0 file.

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

- Endpoint Information
  - Endpoint is identified with Endpoint Information which consists of
    - 1) **ep** for Transport Protocol Suite + Endpoint locator and 2) **pri** for priority.
  - The current



tes

Transport Protocol Suites	scheme	Endpoint Locator	"ep" Value example
coap + udp + ip	coap	IP address + port number	coap://[fe80::b1d6]:1111
coaps + udp + ip	coaps	IP address + port number	coaps://[fe80::b1d6]:1122
coap + tcp + ip	coap+tcp	IP address + port number	coap+tcp://[2001:db8:a::123]:2222
coaps + tcp + ip	coaps+tcp	IP address + port number	coaps+tcp://[2001:db8:a::123]:2233
http + tcp + ip	http	IP address + port number	http://[2001:db8:a::123]:1111
https + tcp + ip	https	IP address + port number	https://[2001:db8:a::123]:1122

eps Parameter for Endpoint Information

- a new Parameter "eps" to embed Endpoint Information in Link
  - "eps" has an array of items as its value and each item represents Endpoint information with two key-value pairs, "ep" and "pri", of which "ep" is mandatory and "pri" is optional.

```
{
  "anchor": "ocf://light_device_id",
  "href": "/myLightSwitch",
  "rt": ["oic.r.switch.binary"],
  "if": ["oic.if.a", "oic.if.baseline"],
  "p": {"bm": 3},
  "eps": [{"ep": "coap://[fe80::b1d6]:1111", "pri": 2}, {"ep": "coaps://[fe80::b1d6]:1122"}]
}
```

  - "anchor" represents the hosting OCF Device, "href", target Resource and "eps" the two Endpoints for the target Resource.
  - If the target Resource of a Link requires a secure connection (e.g. CoAPS), "eps" Parameter shall be used to indicate the necessary information (e.g. port number)

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.

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

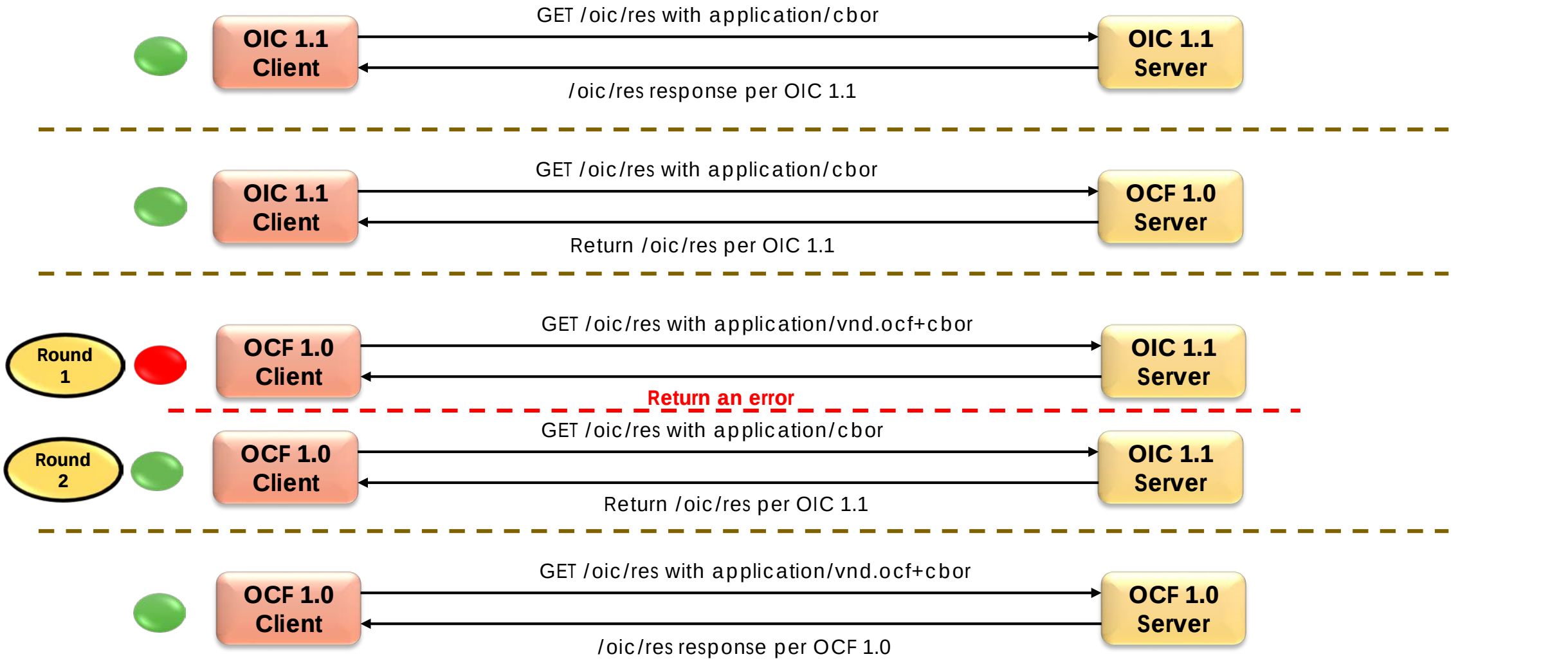
Version Representation

	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



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

Infrastructure Connectivity

- Current scope is very much addressing proximal network
- Active project activity to leverage native OCF capabilities for wider area (beyond the LAN, be it Cloud or other) connectivity:
  - Native CoAP
  - Resource Directory
  - Resource Host
  - Add Pub-Sub pattern to already supported Observe pattern

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

Infrastructure: Security Specification

Overview

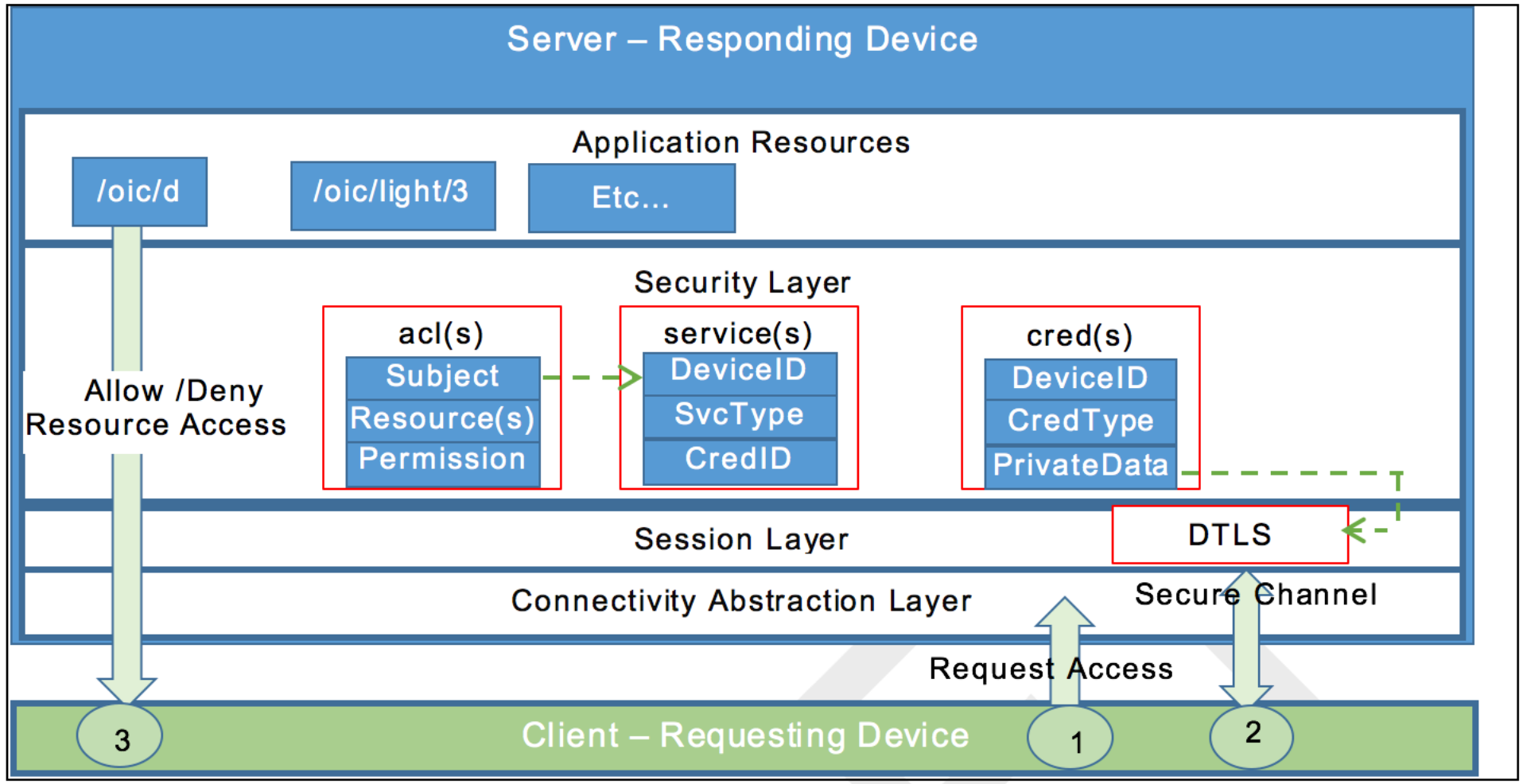
OCF Security Summary

- 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.
- **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.
- **Secure Virtual Resources (SVRs)** are special security resources with severely restricted permissions and access management.
- **Onboarding Tools (OBTs)** are trusted platforms that help bring OCF devices into the local network.

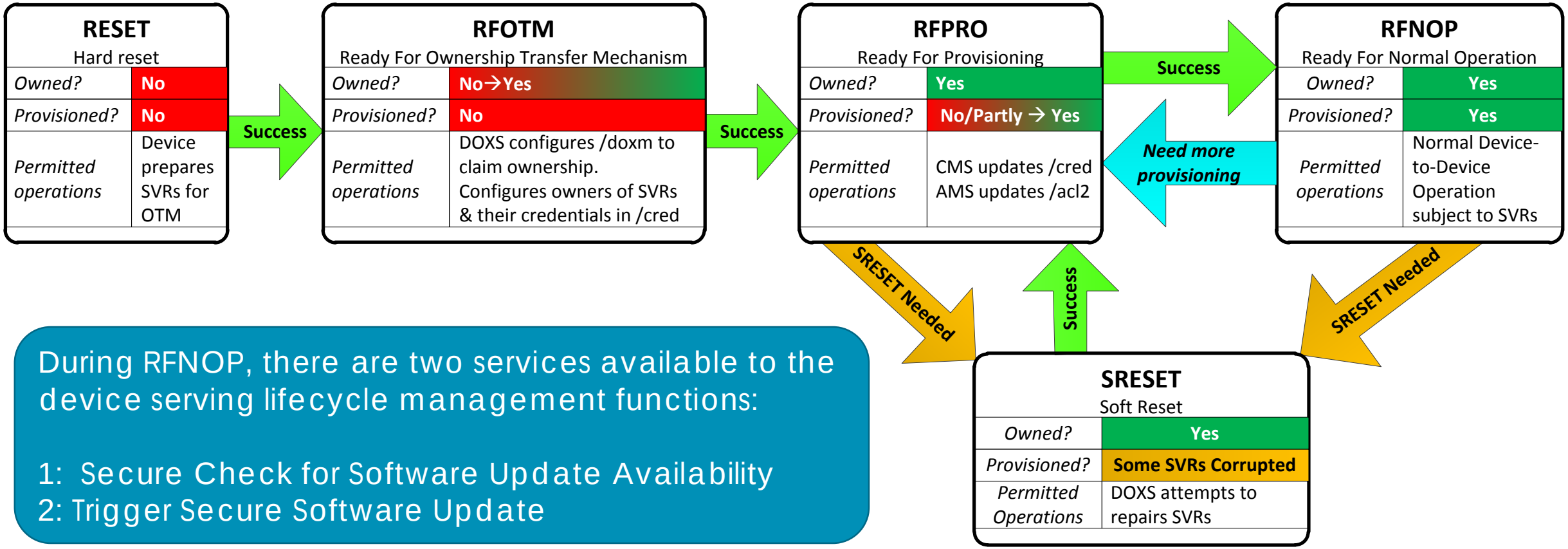
How OCF Security Protects Device Resources:



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



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

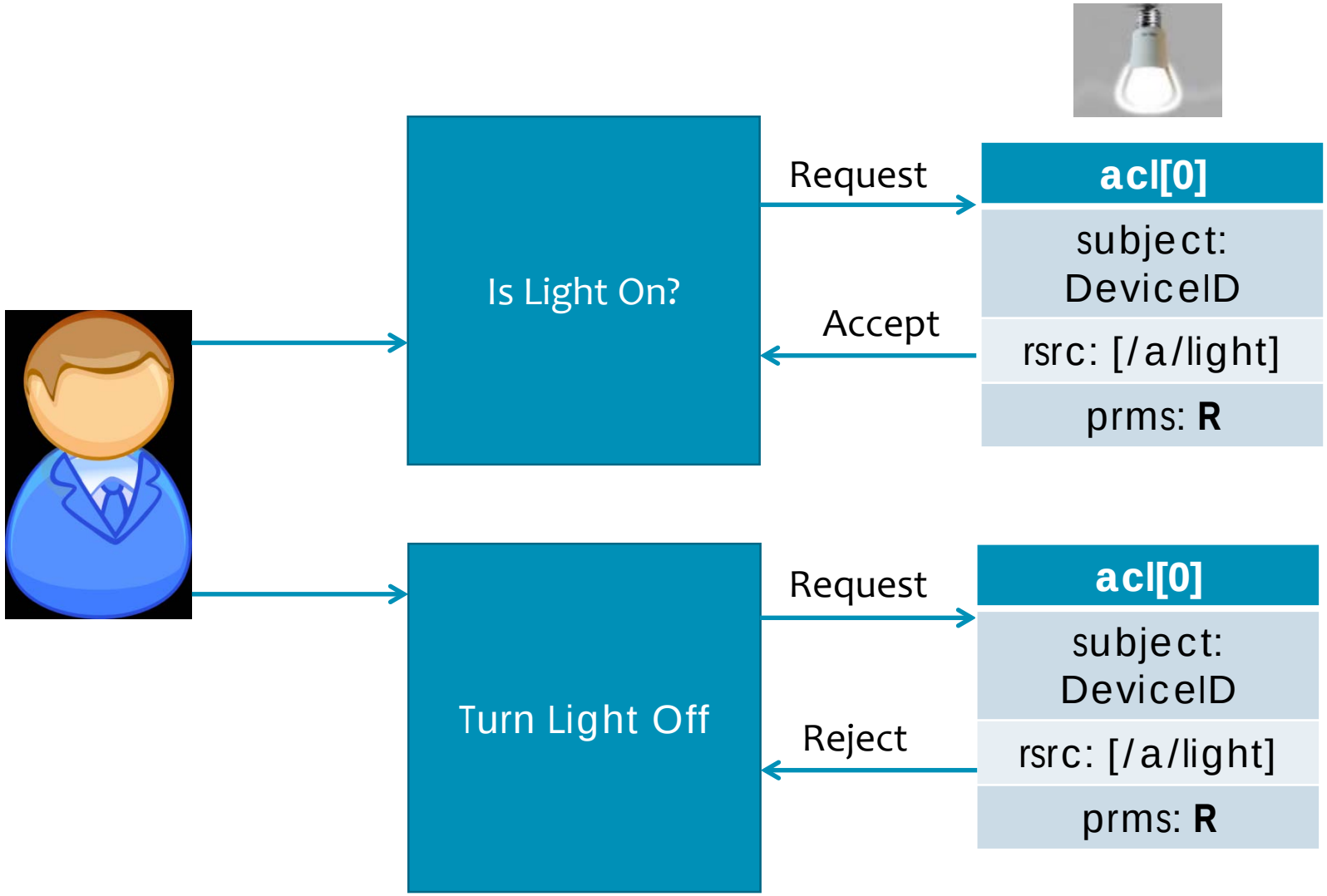
1: Secure Check for Software Update Availability
2: Trigger Secure Software Update

*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



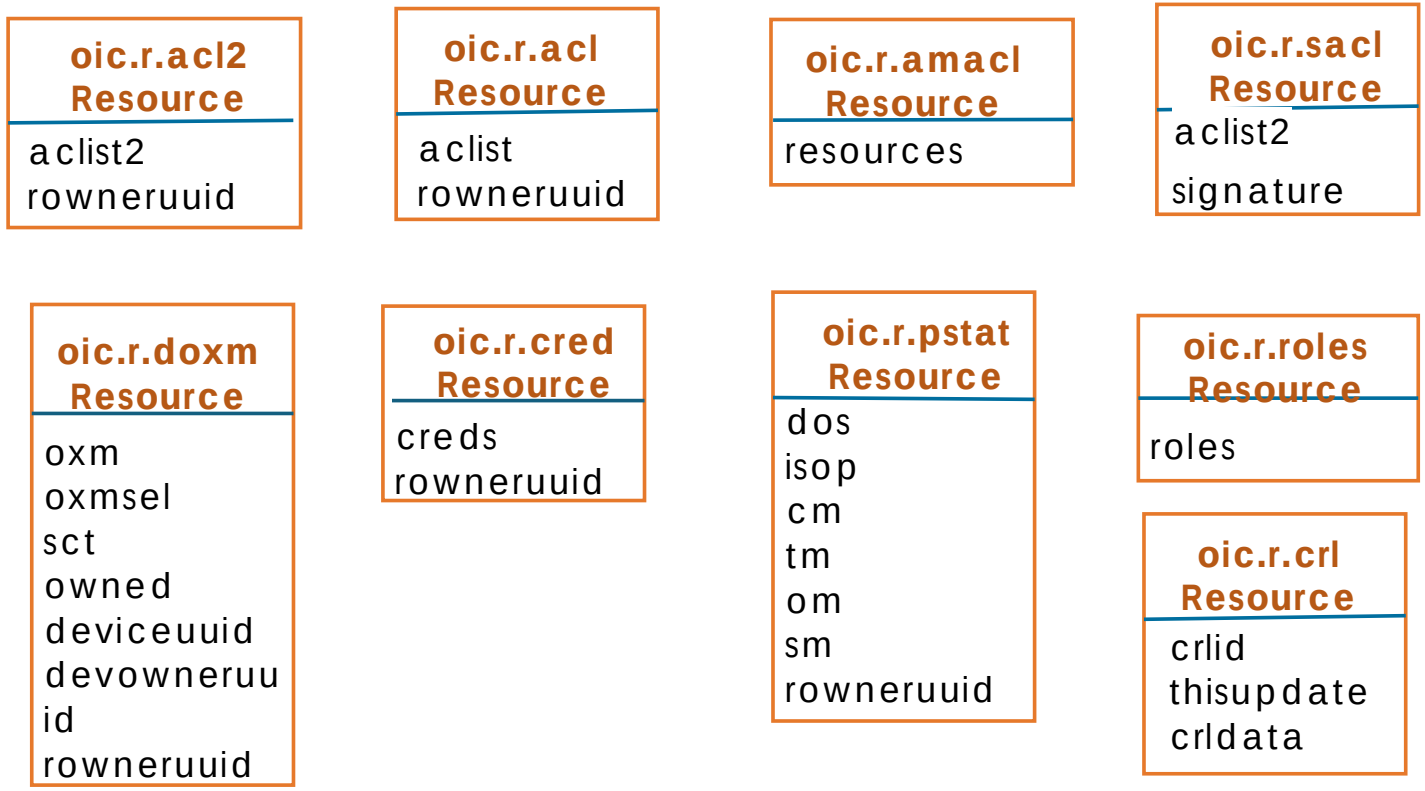
Access Control

- 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
- 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
- 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
  - Certificate Revocation List Resource(/oic/sec/crl) manage certificate revocation
  - Roles Resource (/oic/sec/roles) manage credentials based on the Role
  - Certificate Signing Request Resource (/oic/sec/csr) is used to signed certificate by DOXS
  - Security hardening applies to /oic/sec/cred Resource
- 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)



Message Integrity and Confidentiality

- Secured communications between clients and servers are protected against eavesdropping, tampering, and message replay.
- Unicast messages are secured using DTLS or TLS. Multicast messages are not secured.
- All secured 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.
- The sending device encrypts and authenticates messages as defined by the selected cipher suite and the receiving device verifies and decrypts the messages.
- Secured unicast messages use the specified cipher suites during device ownership transfer and normal operation (for symmetric keys and asymmetric credentials).

INFRASTRUCTURE: BRIDGING SPECIFICATION

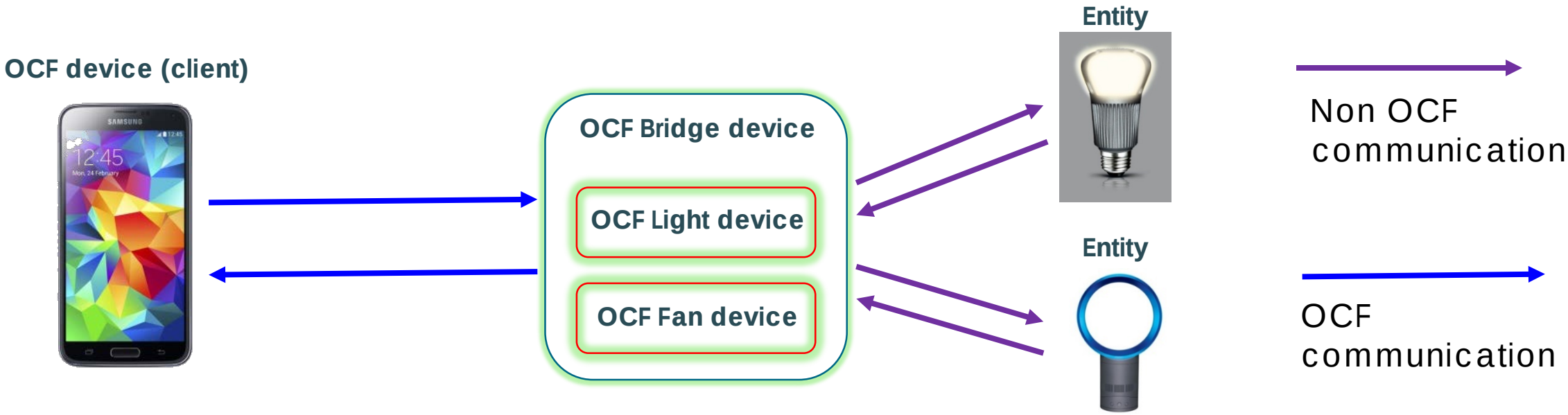
Overview

Bridging Specification

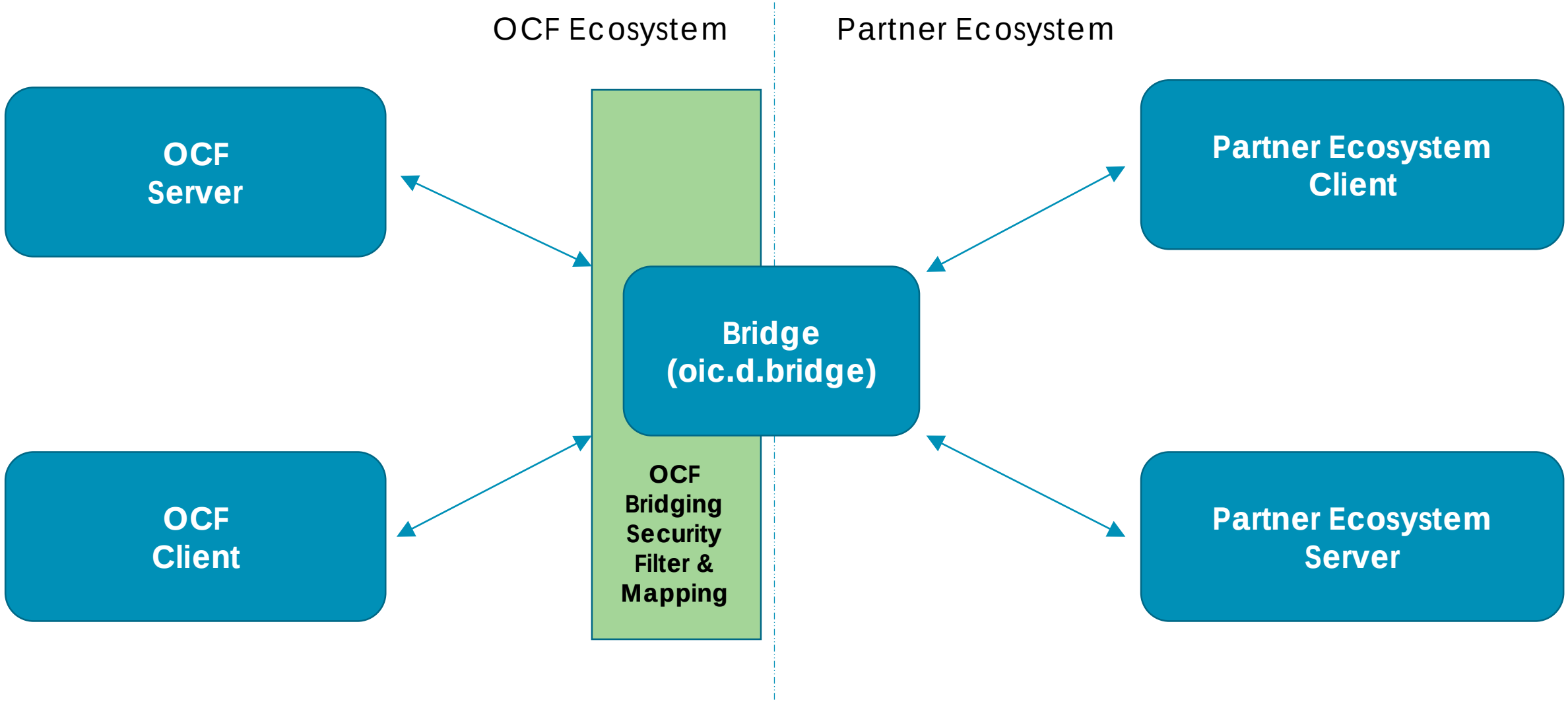
- Specifies a framework for bi-directional translation between devices in OCF and non-OCF ecosystems.
- Specifies general requirements for translation between OCF and non-OCF ecosystems
  - Requirements for resource discovery, message translation, security, and handling of multiple bridges.
- Specifies 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

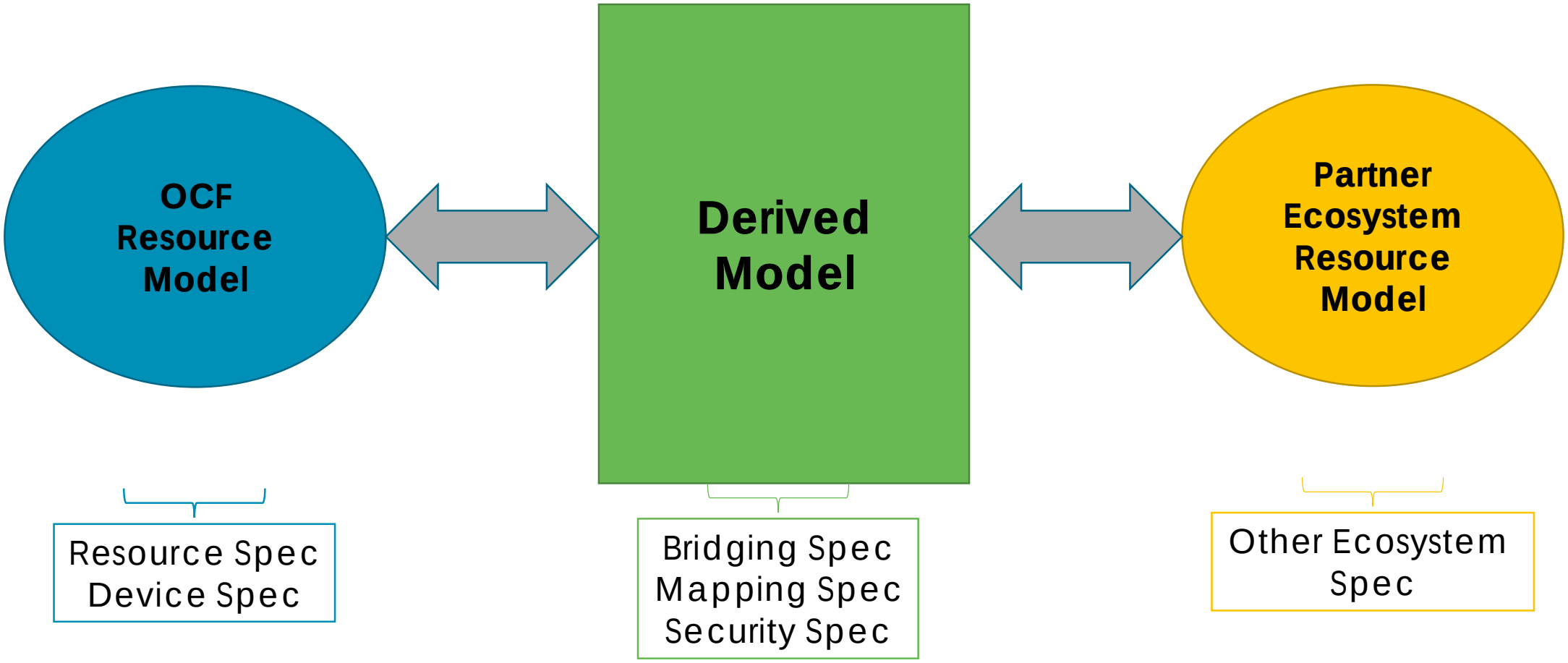
- An OCF Bridge is a device that represents one or more non-OCF devices (bridged devices) as virtual OCF devices on the OCF network.
- The bridged devices themselves are out of the scope of OCF.
- The only difference between a 'regular' OCF device and a virtual OCF device is that the latter is encapsulated 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.



Bridging Concept – Bidirectional Operation



Bridging Concept – Data Model & Security



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™

RESOURCE MODEL: RESOURCE TYPE SPECIFICATION

Overview

Resource Specification

- List of reusable resources that are used in an OCF Device
  - Total of 74 Resource Types defined as of OCF 1.0
  - Uses core definitions
- Each resource definition contains:
  - unique identifier (rt)
  - Identification of the default interface and other supported interfaces
  - List 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 Swagger2.0

Sample Set Defined Resource Types – OIC 1.1

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

Sample Set Defined Resource Types – OIC 1.1 (2/2)

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	

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

New Resource Types – OCF 1.0

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 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™

Resource Model: 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 = selectablehumiditylevels[0].",
            "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
          ],
          "x-from-ocf": [
            "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
            "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
          ]
        }
      }
    }
```

AllJoyn Interface Name

AllJoyn Property Name

OCF Equivalent Resource Type

To OCF 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™

PER VERTICAL:
SMART HOME DEVICE SPECIFICATION

Overview

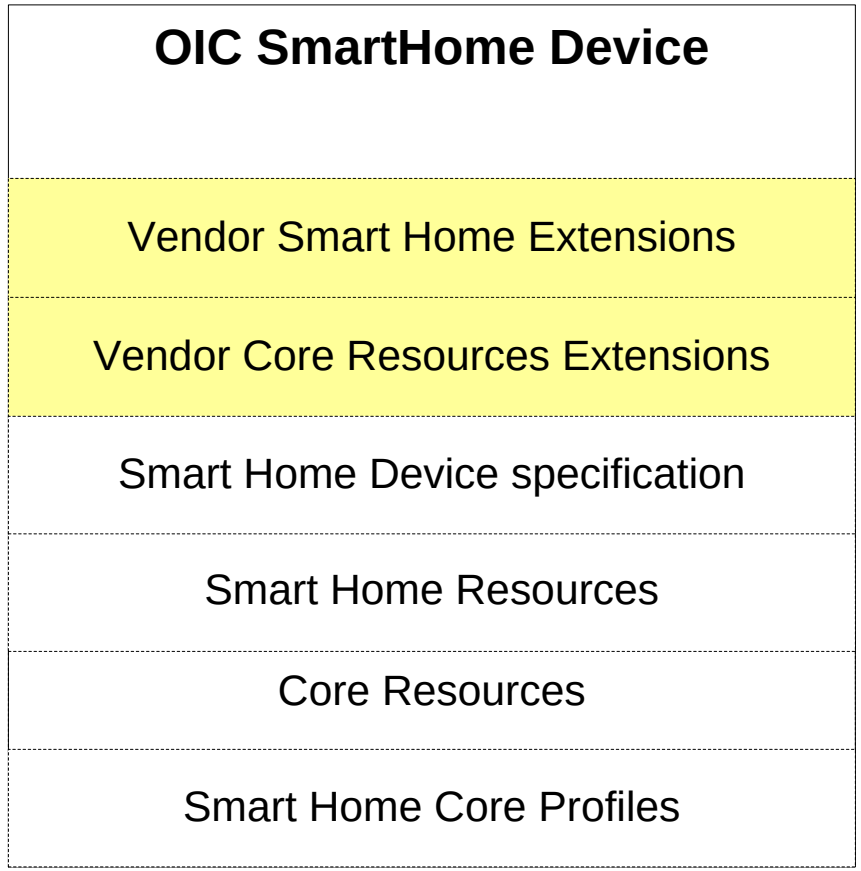
Higher Layer Specifications

- Specifications are split into 2 documents:
  - Device specification (per vertical 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 smart home devices
- Each Smart home device definition contains:
  - unique identifier (rt)
  - a list of mandatory resources



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.

Smart Home Device Type (1/2)

Device Type	Minimum Resource Set
Air Conditioner	Binary Switch, Temperature
Air Purifier	Binary Switch
Air Quality Monitor	Air Quality Collection
Blind	Open Level
Camera	Media
Clothes Dryer	Binary Switch, Mode
Clothes Washer	Binary Switch, Mode
Clothes Washer/Dryer	Binary Switch, Operational State
Cooker Hood	Airflow Control, Binary Switch, Mode
Cooktop	Heating Zone Collection
Dehumidifier	Binary Switch, Humidity
Dishwasher	Binary Switch, Mode
Door	Open Level
Fan	Binary Switch

Device Type	Minimum Resource Set
Food Probe	Temperature
Freezer	Temperature (2)
Garage Door	Door
Generic Sensor	Sensor
Humidifier	Binary Switch
Light	Binary Switch
Oven	Binary Switch, Temperature (2)
Printer	Binary Switch, Operational State
Printer (Multi-Function)	Binary Switch, Operational State (2), Automatic Document Feeder
Receiver	Binary Switch, Audio Media Source List (2)

Smart Home Device Type (2/2)

Device Type	Minimum Resource Set
Refrigerator	Binary Switch, Refrigeration, Temperature (2)
Robot Cleaner	Binary Switch, Mode
Scanner	Binary Switch, Operational State, Automatic Document Feeder
Security Panel	Mode
Set Top Box	Binary Switch
Smart Lock	Lock Status
Smart Plug	Binary Switch
Switch	Binary Switch
Television	Binary Switch, Audio, Media Source List
Thermostat	Temperature (2)
Water Valve	Open Level

Thank you!

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

OPEN CONNECTIVITY FOUNDATION™