

# OCF Fundamentals

Ravi Subramaniam

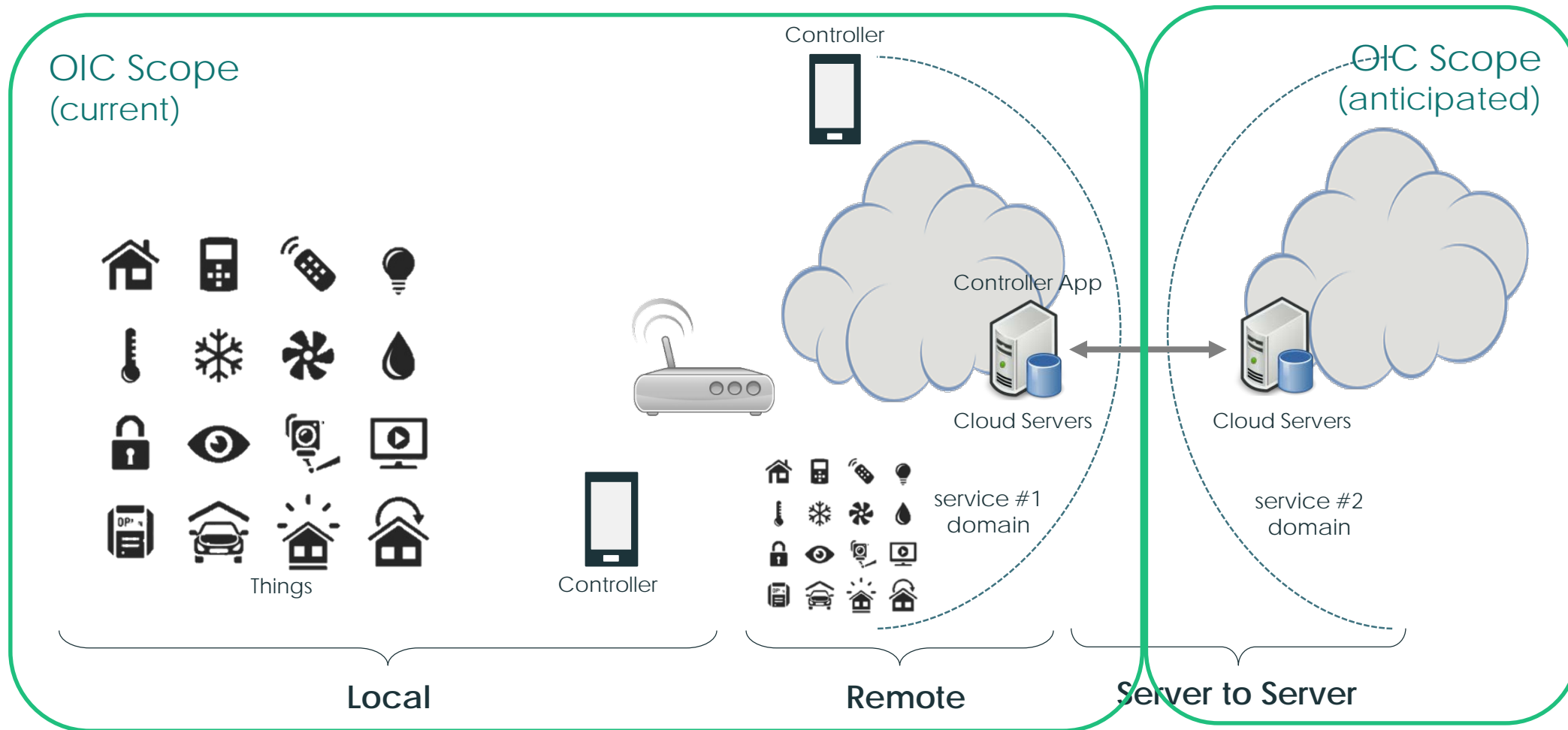
Principal Engineer, Intel Corporation

Fundamental Fundamentals

OCF Resource Model

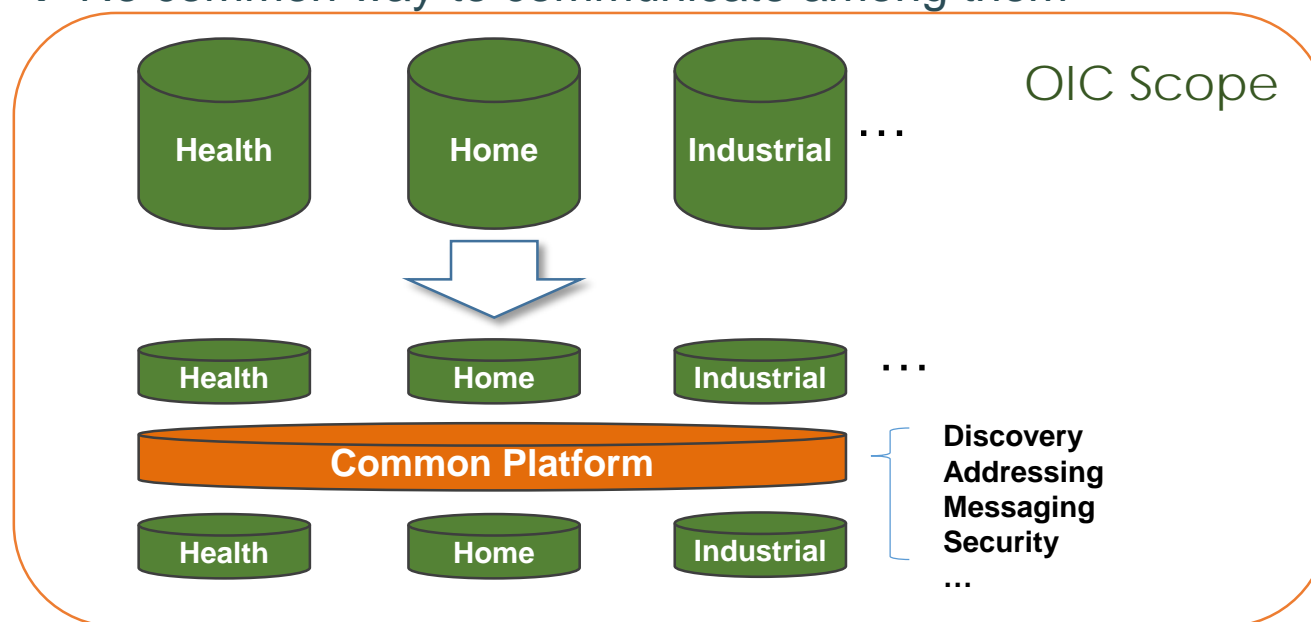
Mapping onto Comms Transports

# Many kinds of devices ...



# Many vertical/domains ...

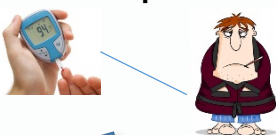
- Legacy vertical services usually designed as silos  
→ No common way to communicate among them



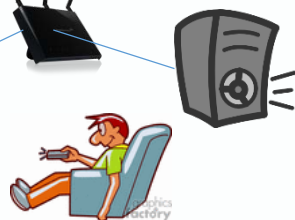
- A common platform provides a foundation for vertical services to collaborate and interwork by providing common services and data models

Home Health Domain

Insulin level low!  
Need Help!



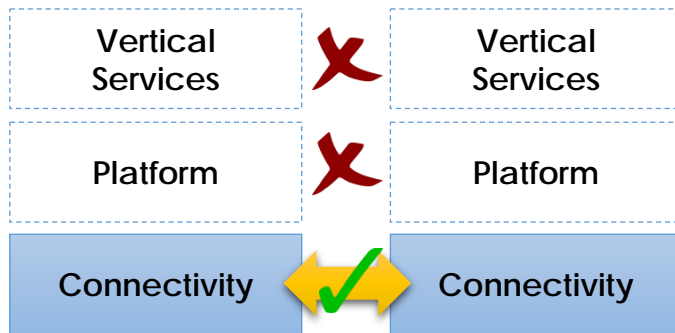
Smart Home Domain



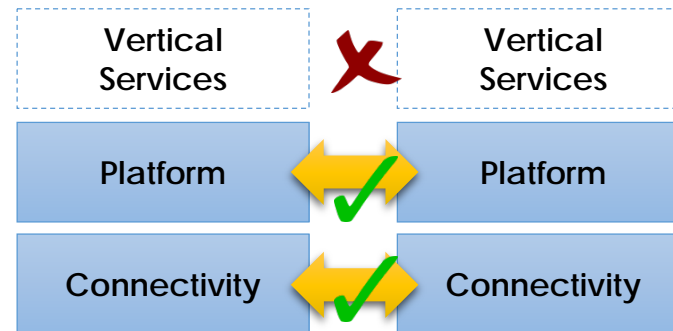
# Interoperability ...

- **Full interoperability** from the connectivity layer up to the service layer is the only way to truly guarantee a satisfactory UX
- Interoperability at the Connectivity and/or Platform layer only provides partial interoperability which can ultimately lead to fragmentation

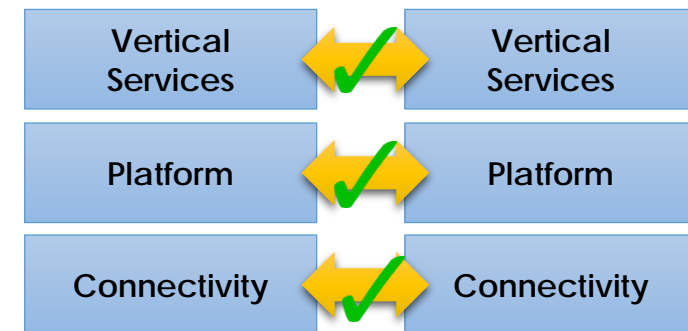
## ① Connectivity Level Interoperability



## ② Platform Level Interoperability



## ③ Service Level Interoperability

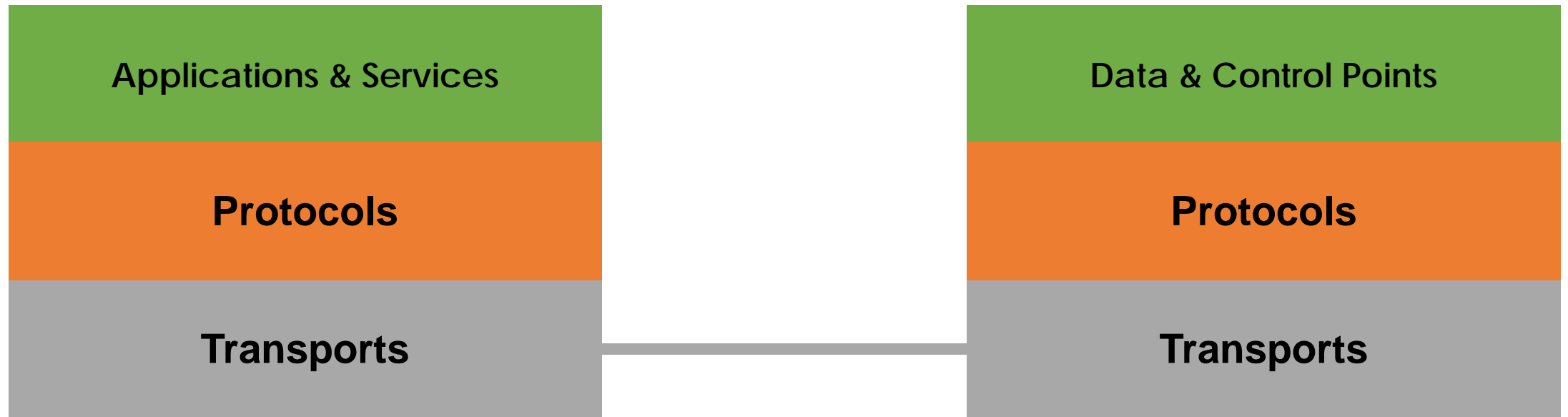


OIC Scope

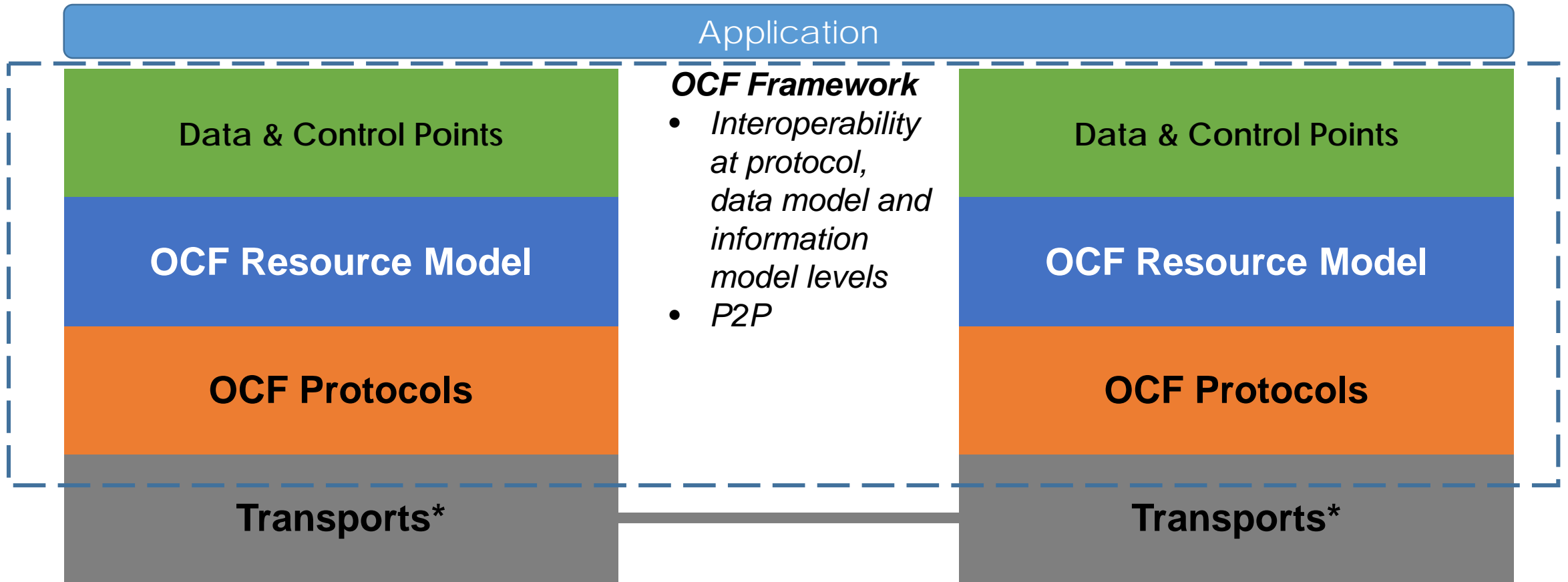
# Today ... Transport Oriented ...



# Protocol Abstraction



# OCF Framework



\*Transports not defined by OCF

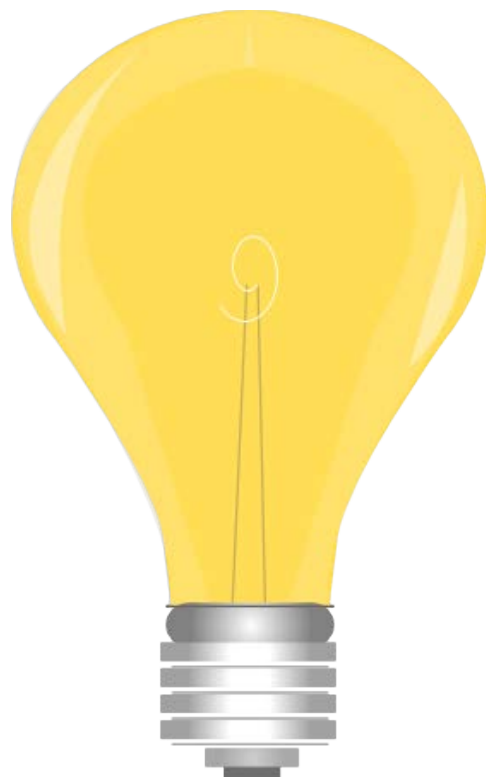


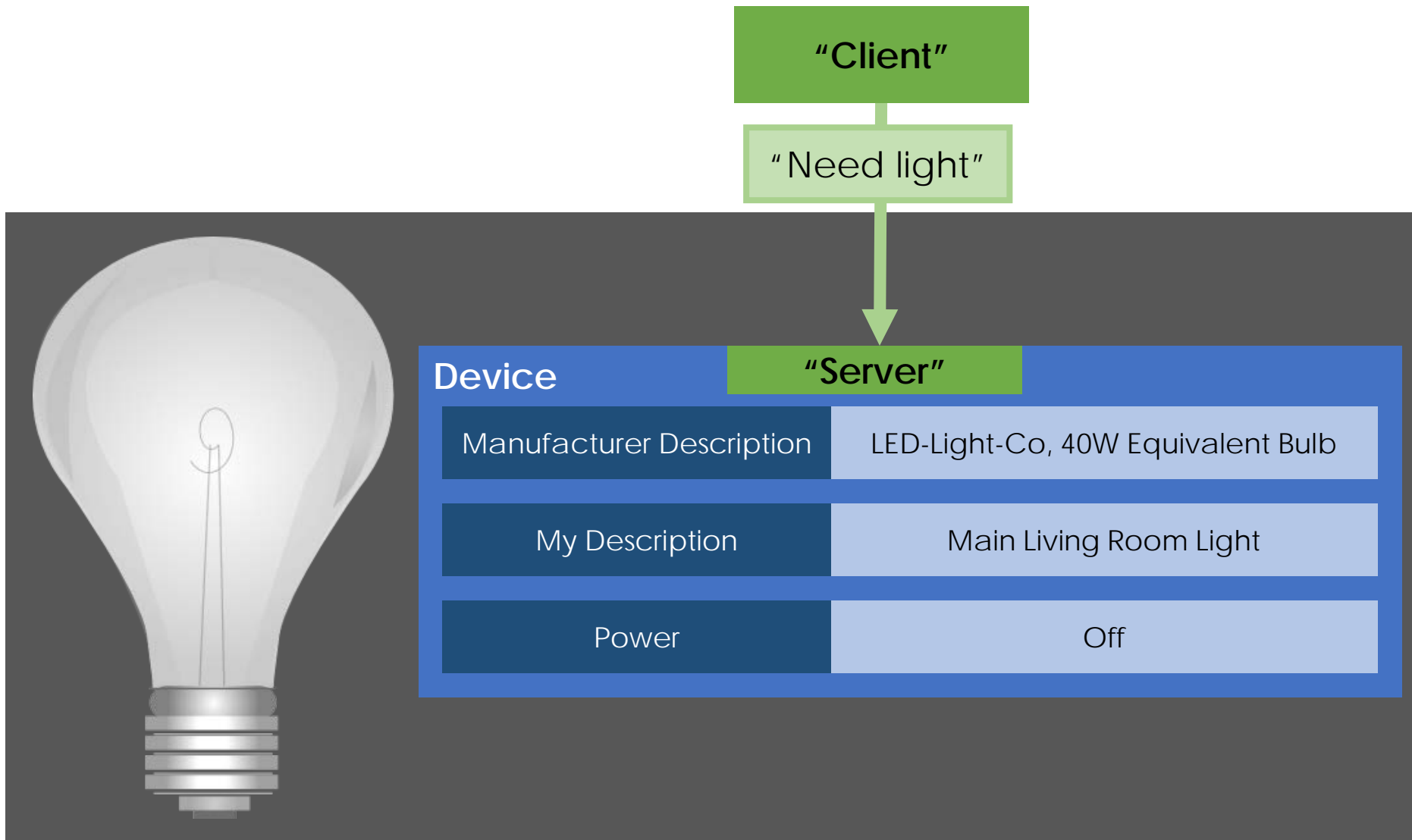
Apps &  
Services

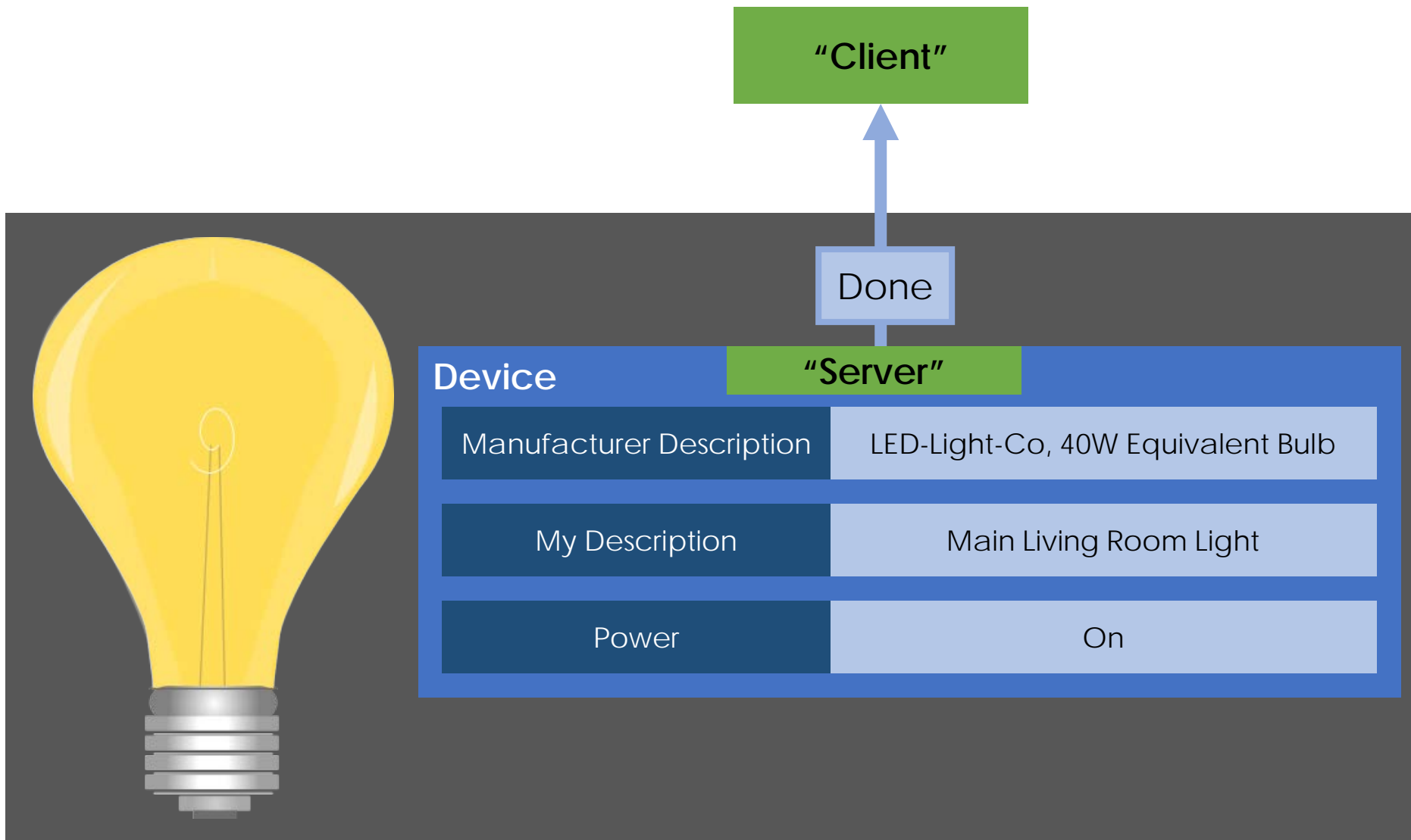
OCF  
Framework



Ah ... a light bulb !!

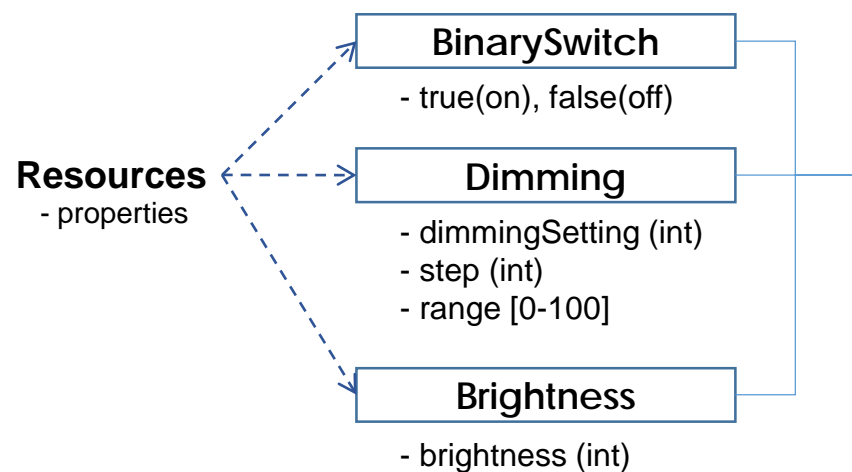






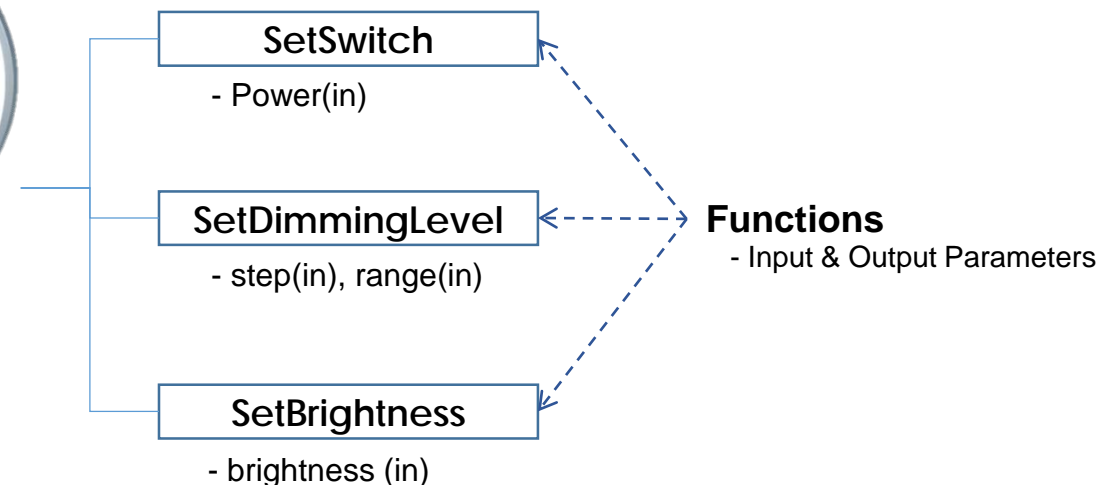
## OIC Scope

- **Declarative:** By declaring expected outcomes and intent on state i.e. **what**



e.g., Light bulb

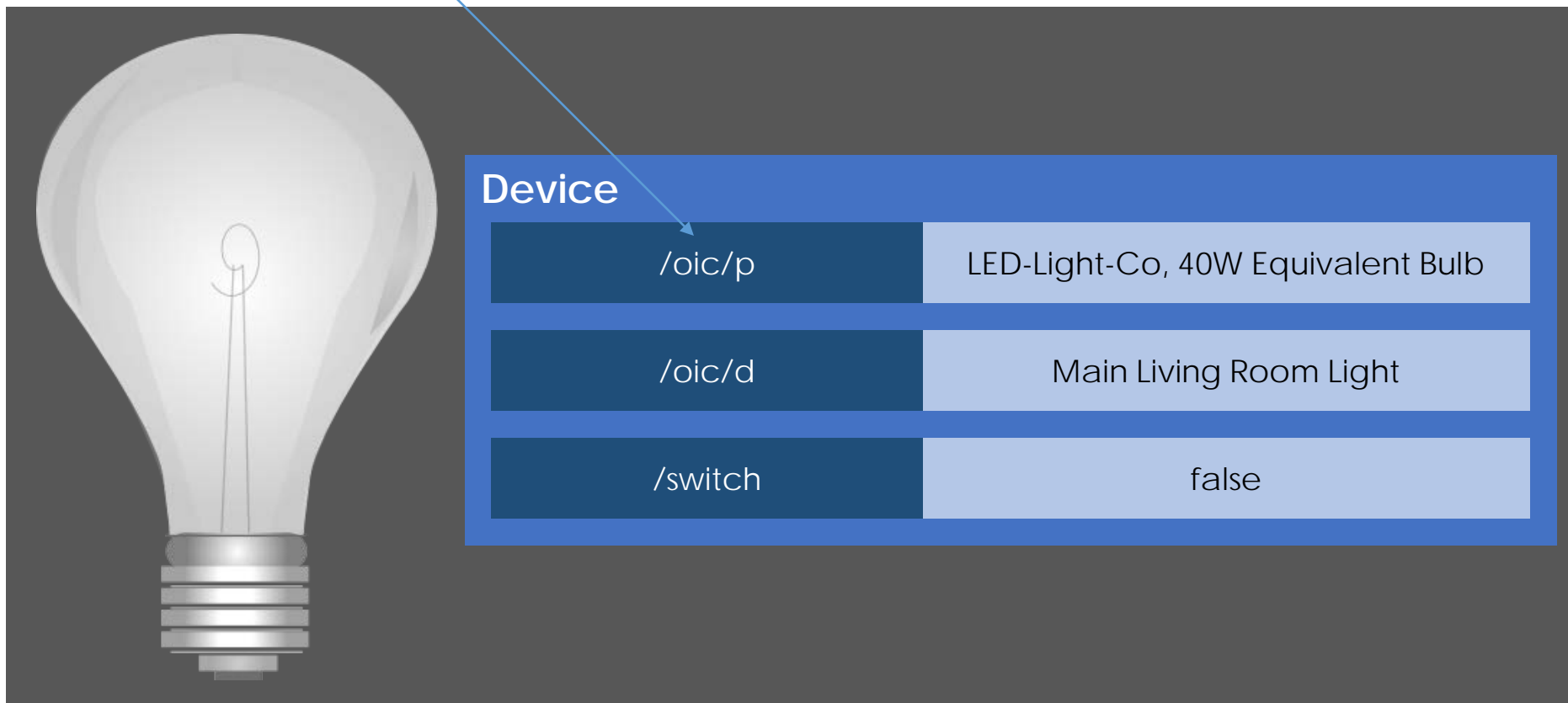
- **Imperative:** By defining functions of and operations on things i.e. **how**



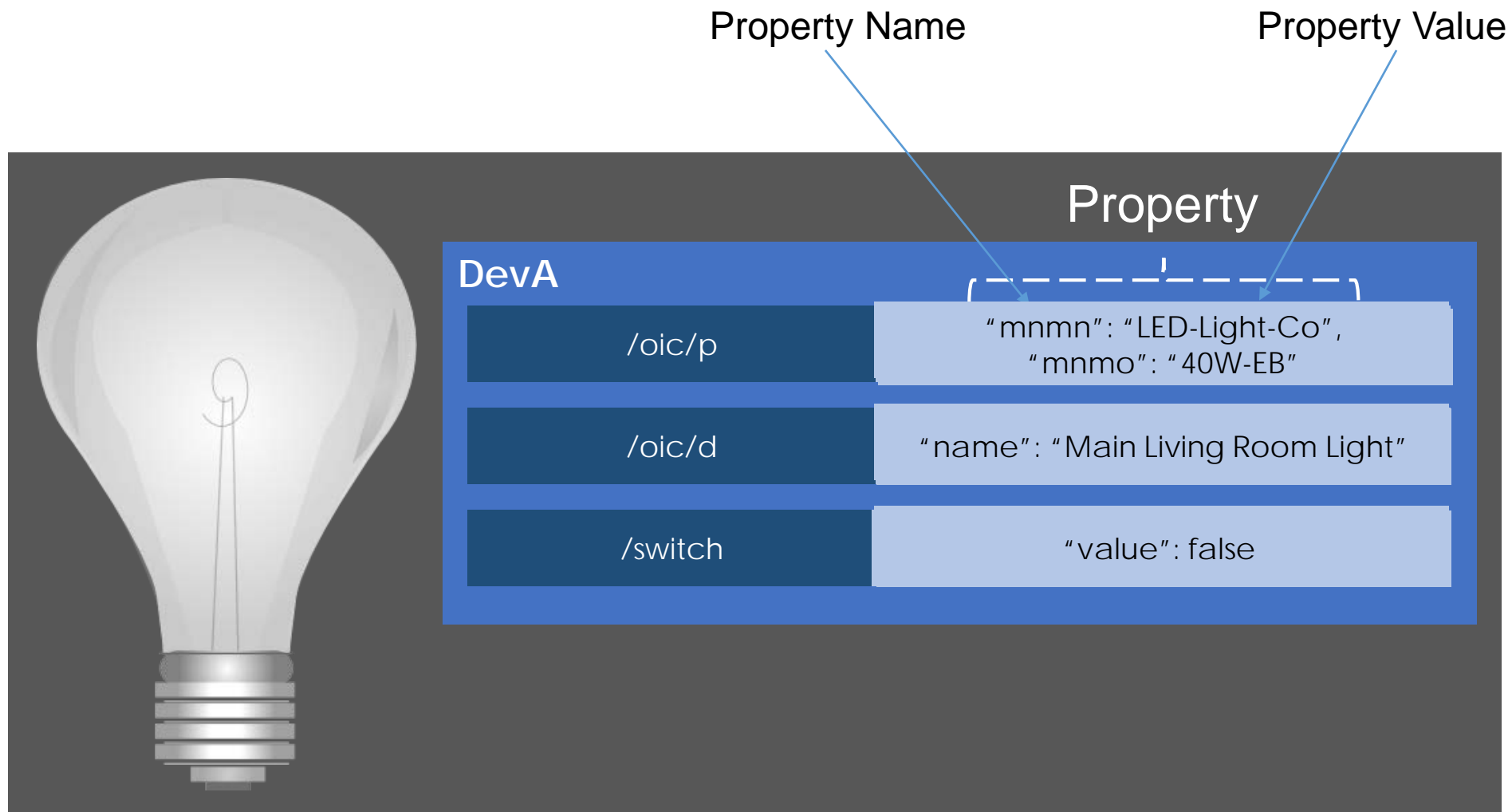
- State + Objects  
\*Fixed set of verbs (CRUDN)
- Resource oriented in RESTful Architecture

- (Verbs + Objects)
- RPC model

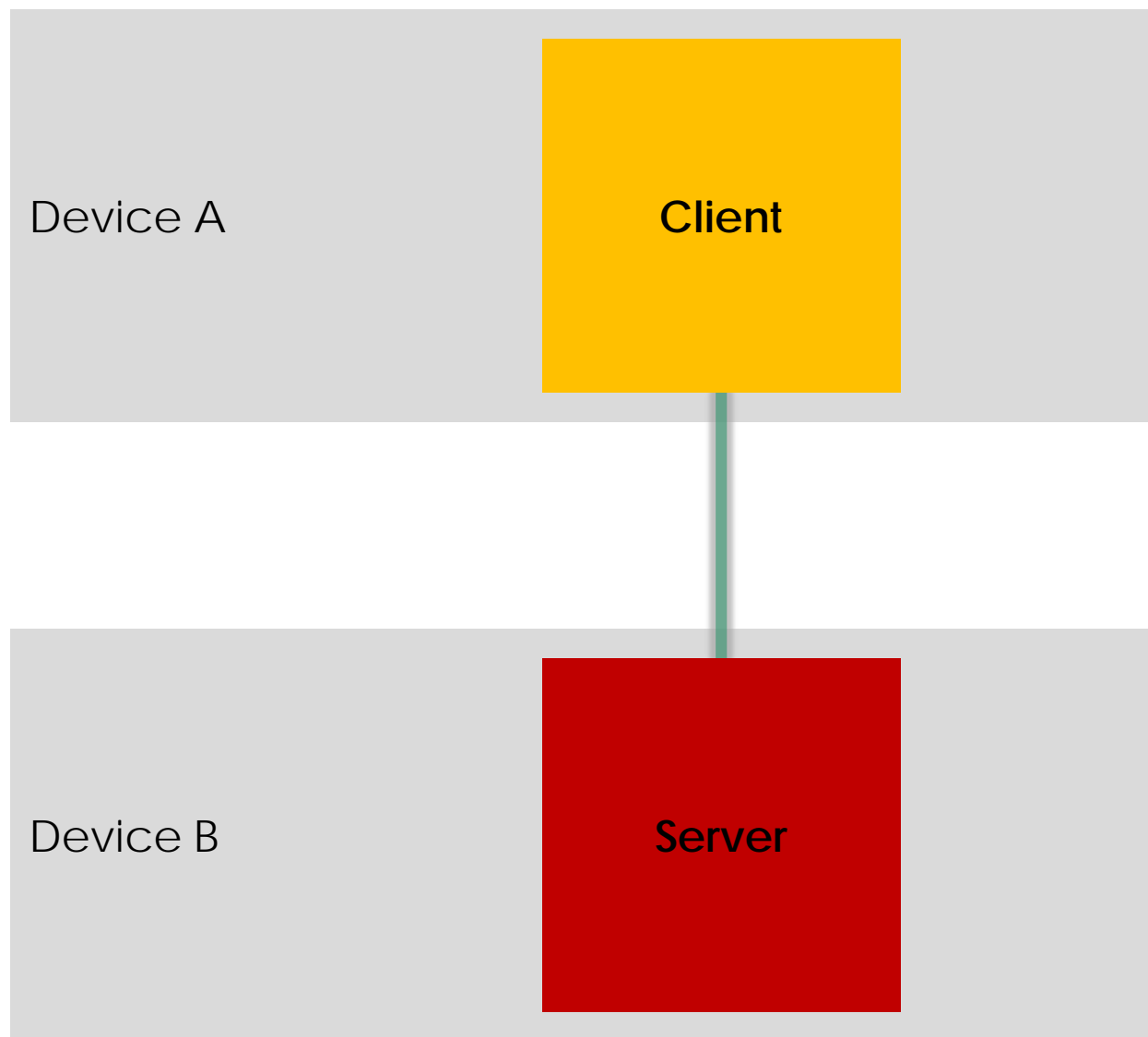
## Resource & Uniform Resource Identifier (URI)



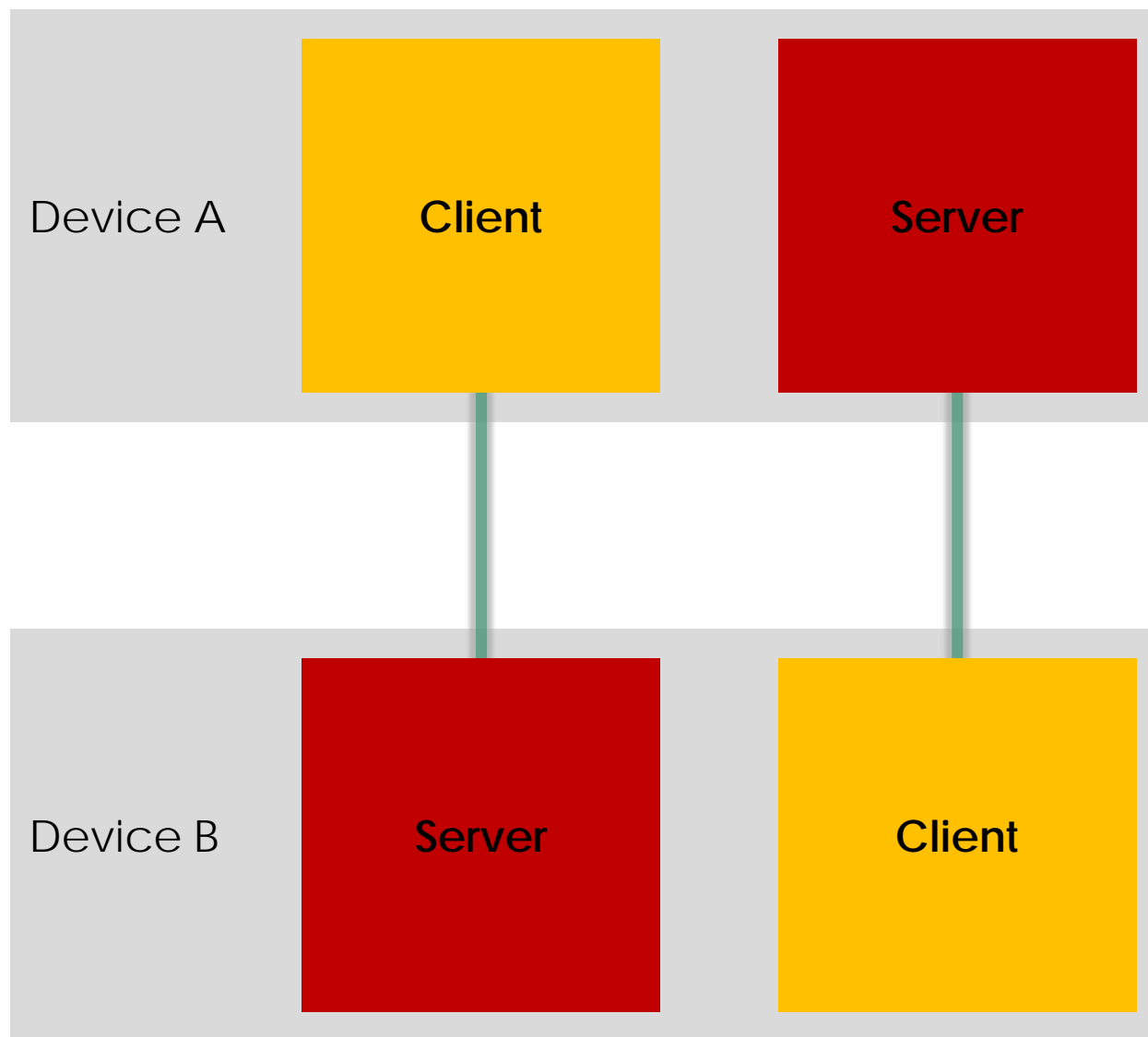
Device	
/oic/p	LED-Light-Co, 40W Equivalent Bulb
/oic/d	Main Living Room Light
/switch	false







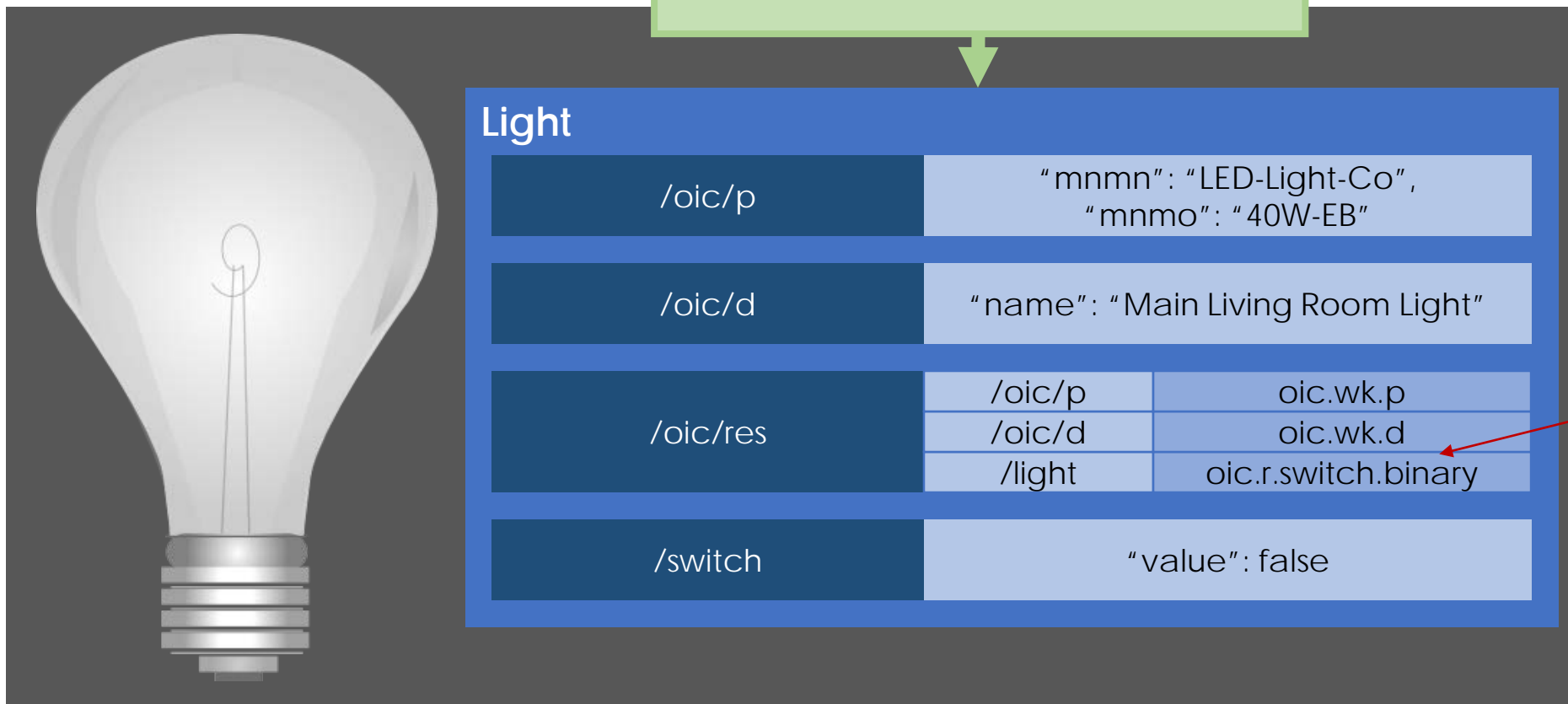
Client and  
Server as  
**Roles**



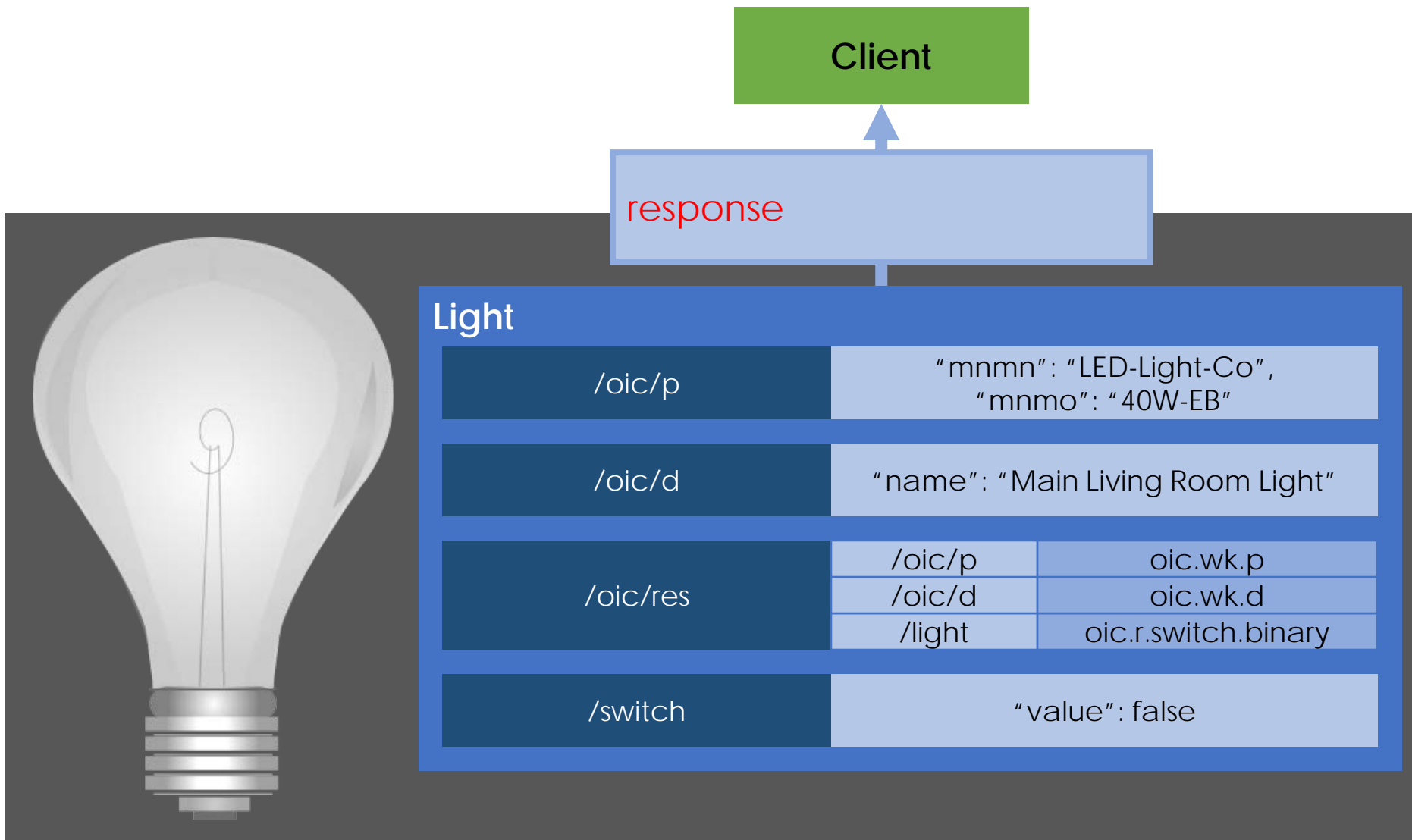
“Discovery”

Client

RETRIEVE oic://<DevA>/oic/res



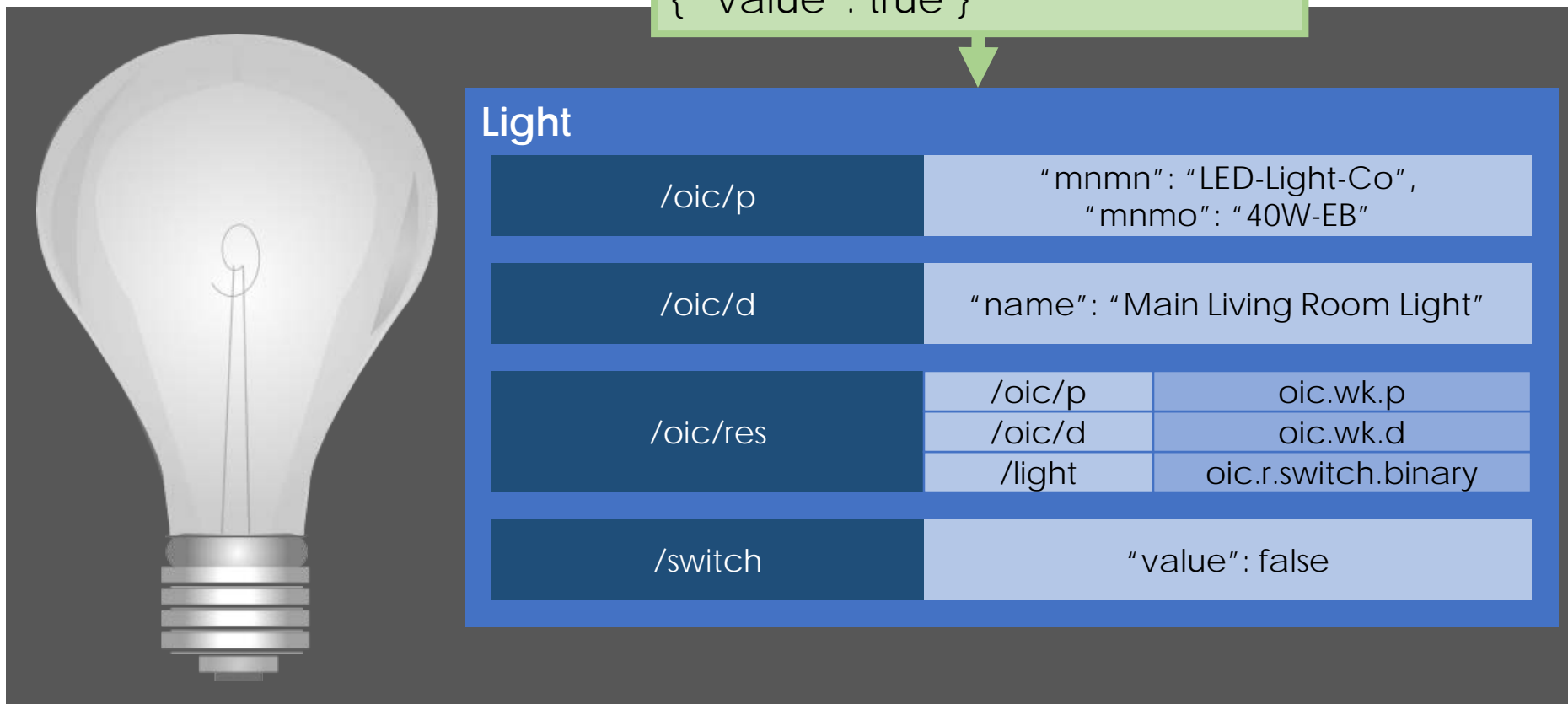
Resource Type  
• *Resource Template*

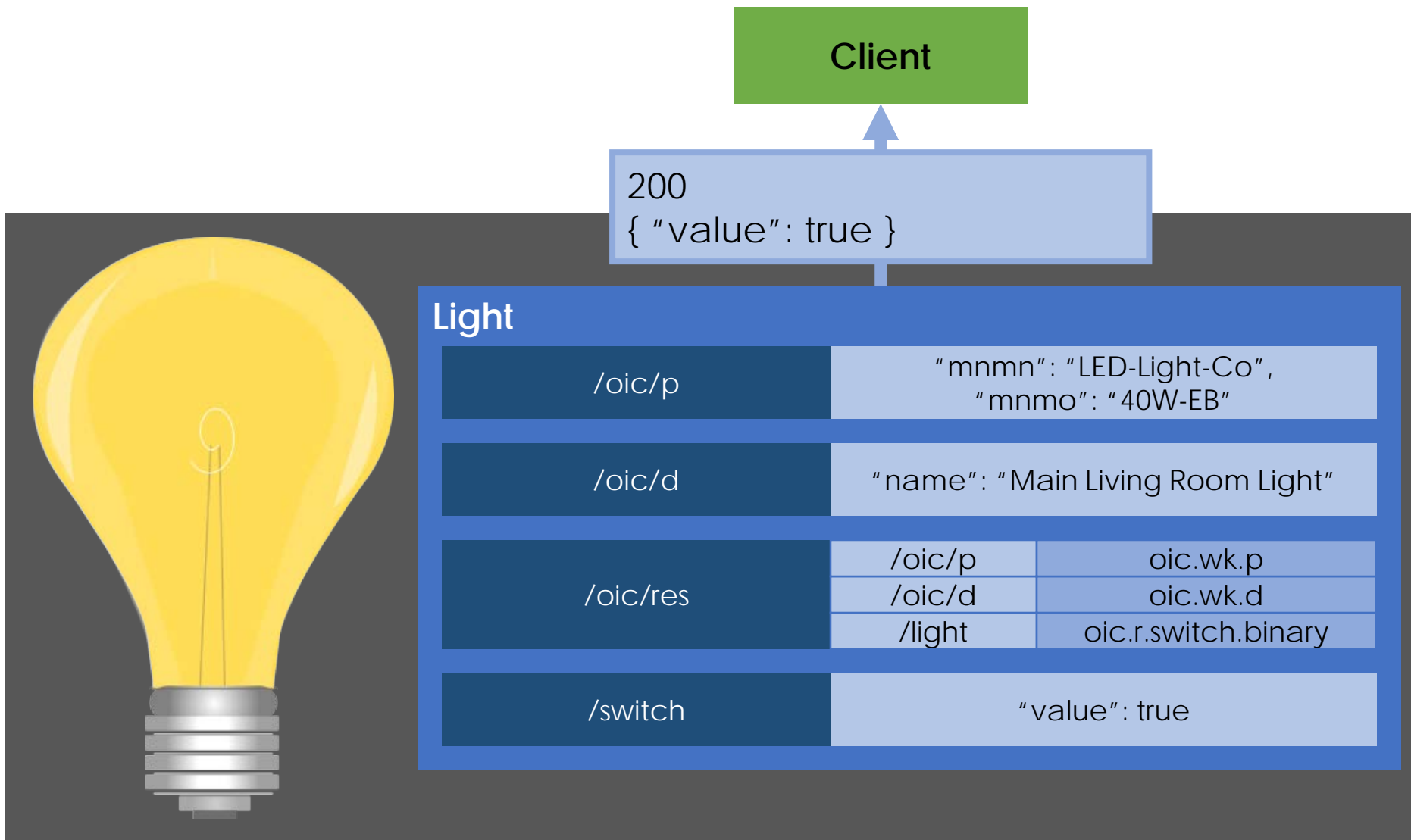


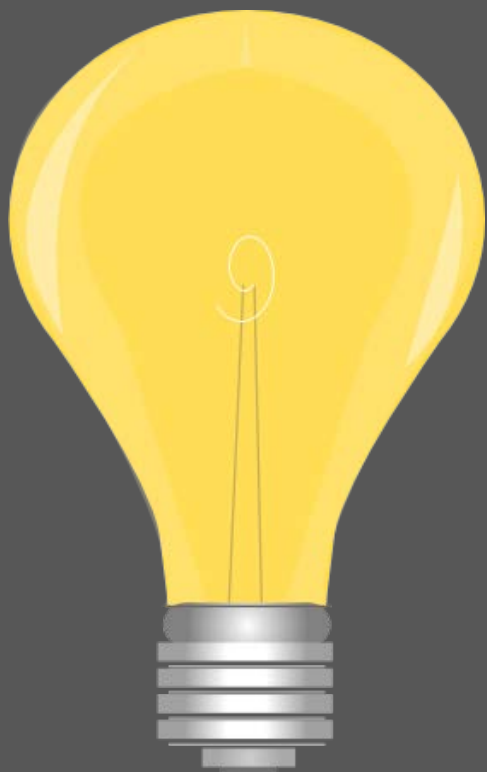
“Control”

Client

```
UPDATE oic://<DevA>/switch
{ "value": true }
```







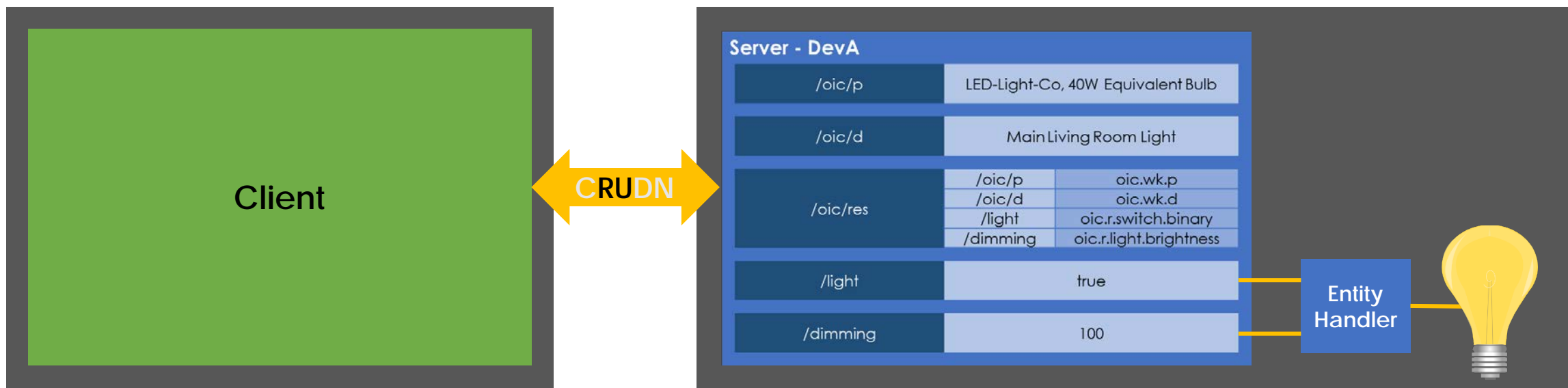
## Light

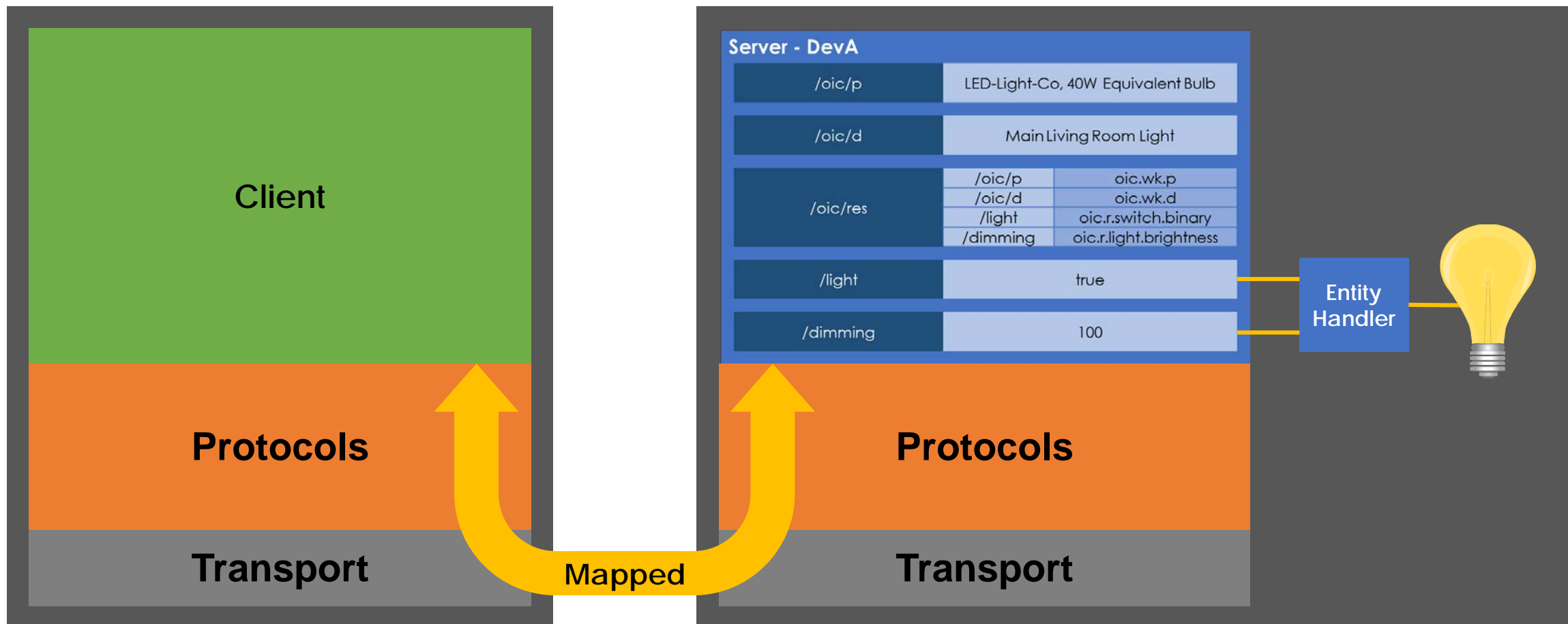
/oic/p	"mnmn": "LED-Light-Co", "mnmo": "40W-EB"	
/oic/d	"name": "Main Living Room Light"	
/oic/res	/oic/p	oic.wk.p
	/oic/d	oic.wk.d
	/light	oic.r.switch.binary
	/dimming	oic.r.light.brightness
/light	"value": true	
/dimming	"value": 100	

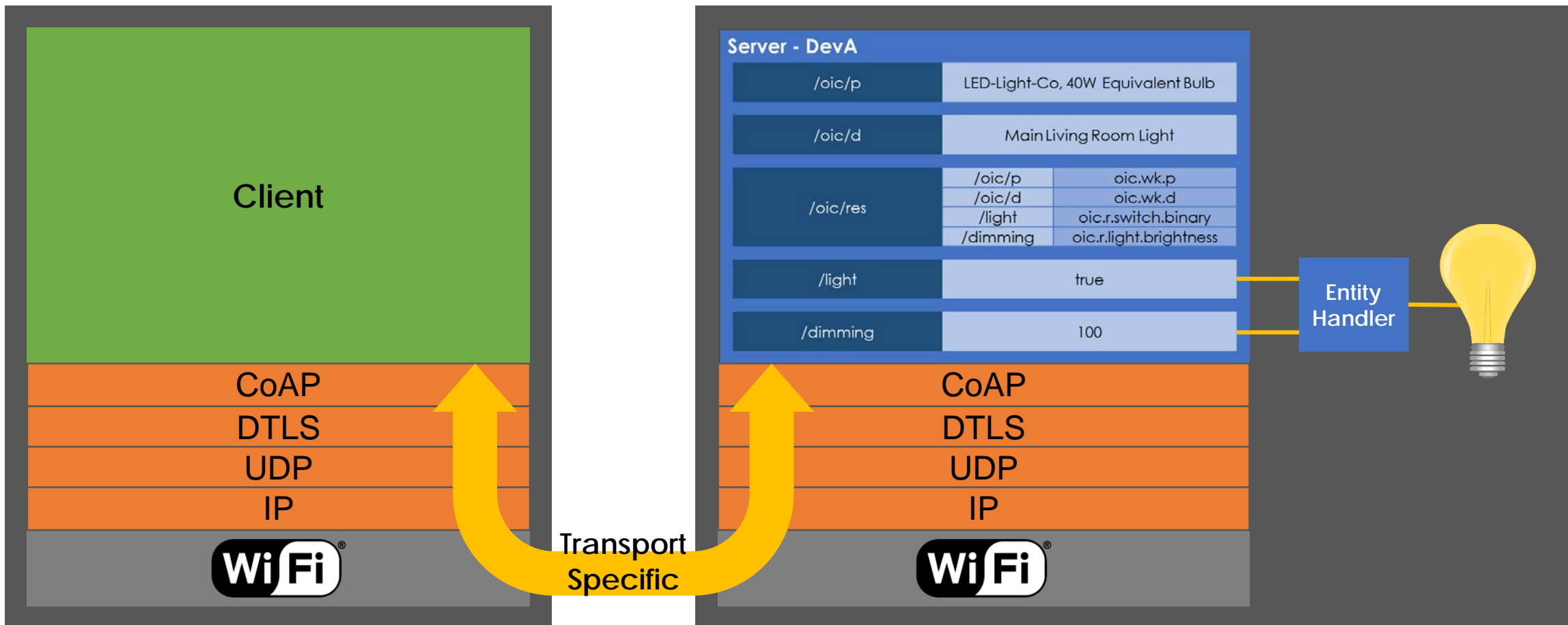
R  
e  
t  
r  
i  
e  
v  
e

U  
p  
d  
a  
t  
e

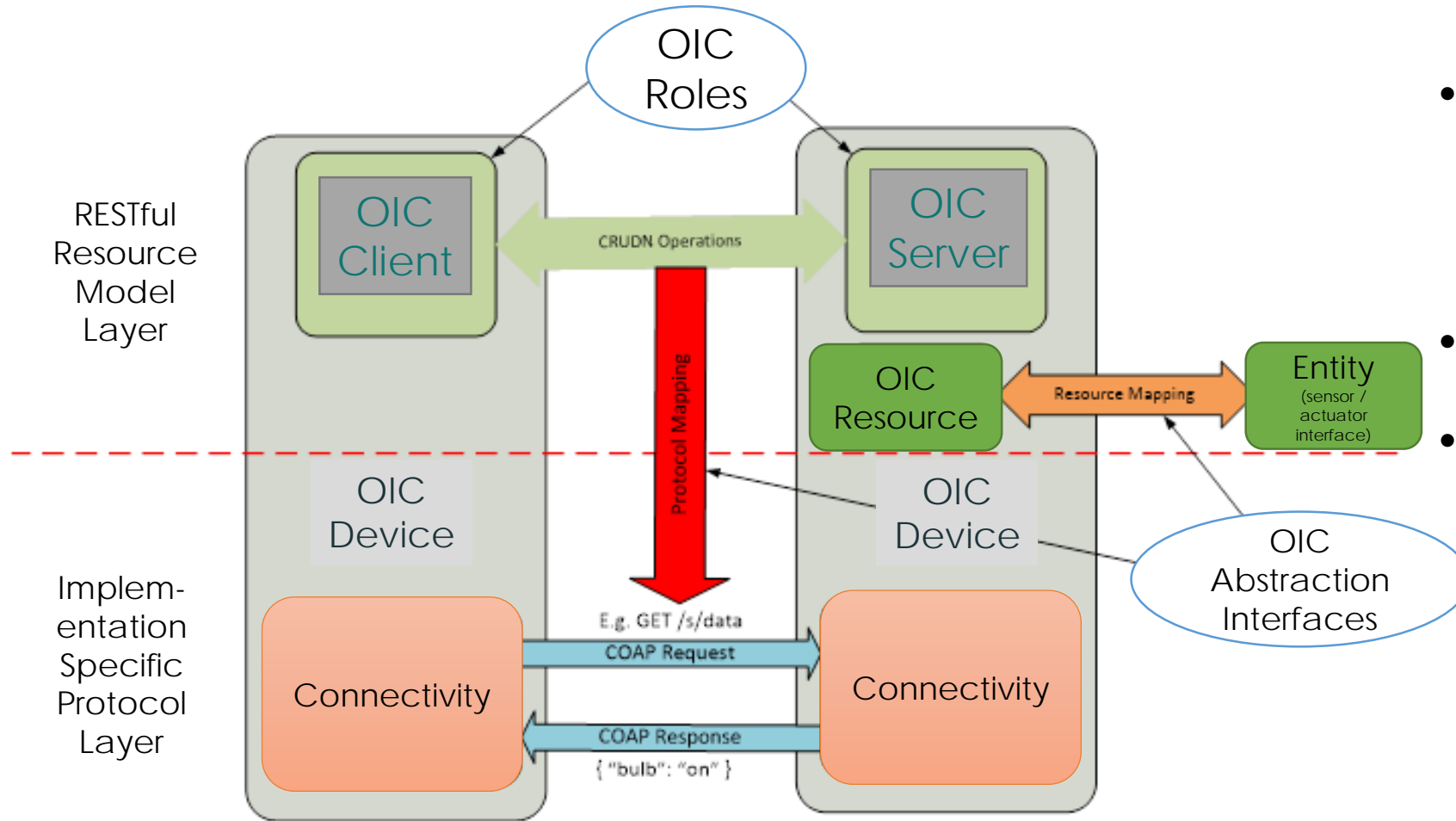








# Putting it all together ...



- **Information Model**
  - Resource oriented
  - RESTful architecture
  - Semantics
  - Physical abstraction
- **Data Model**
  - For vertical and device
- **Data connectivity abstraction**
  - Protocol and layer agnostic
  - Dynamic and late binding

Expanding on the preliminary examples ....

# Uniform Resource Identifier (URI)

## Relative Reference

Form: /<path>?<query>

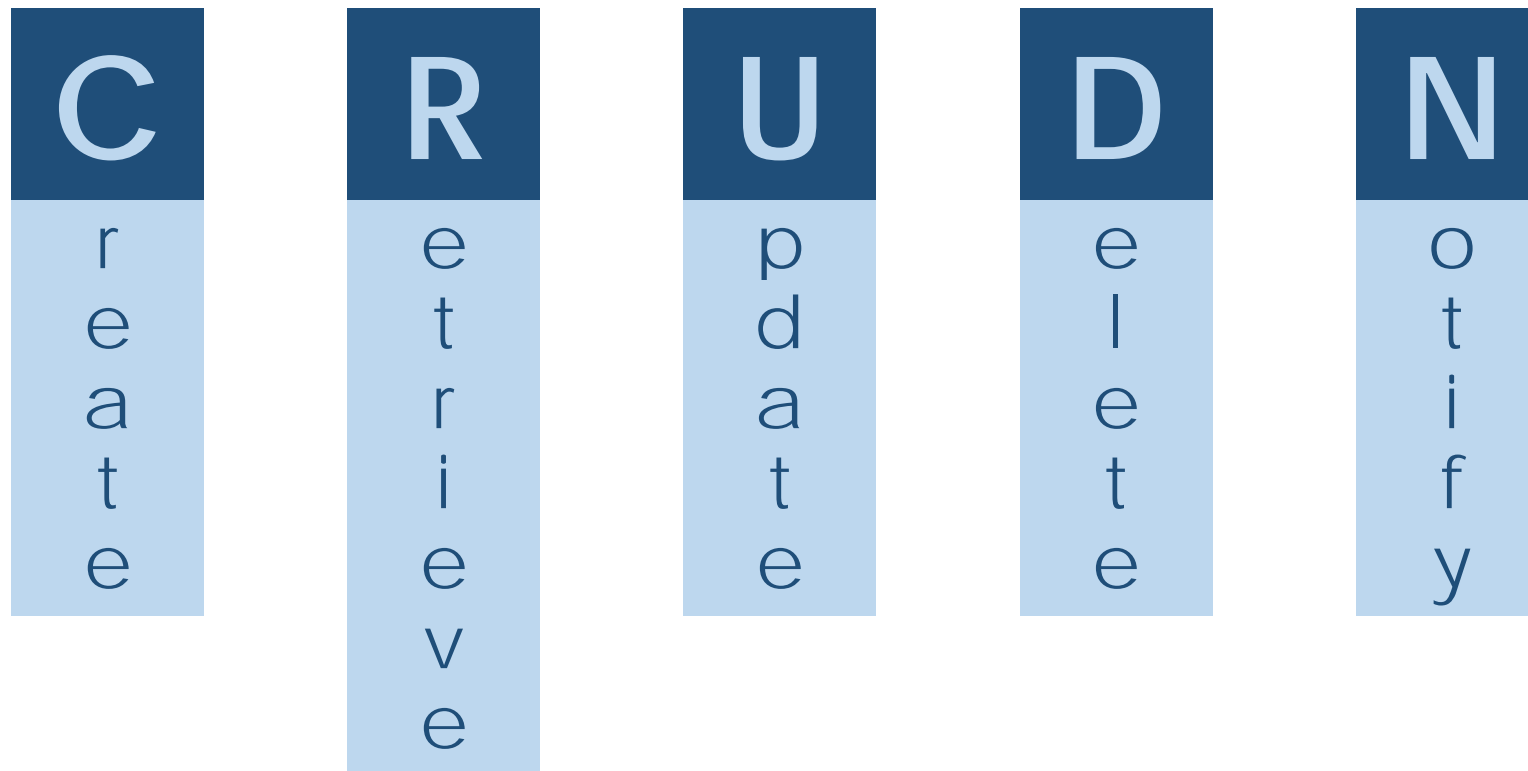
Example: /oic/p

## URI

Form: oic://<reg-name>/<path>?<query>

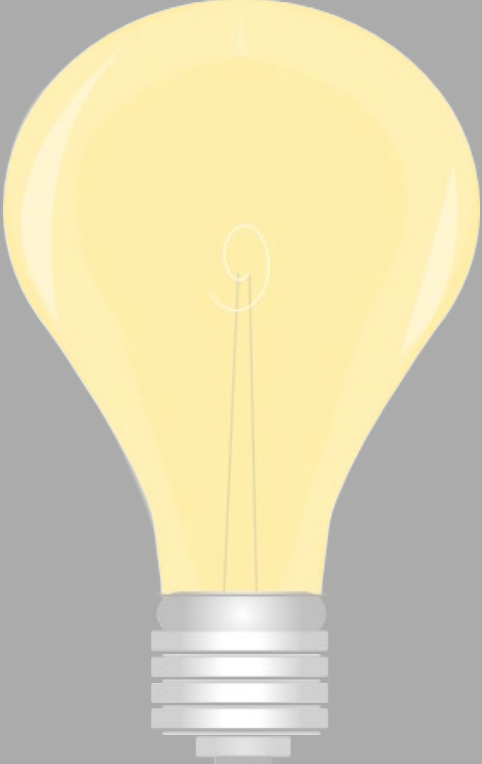
Example: oic://<UUID>/oic/p

- “oic” is the scheme for constructing and parsing the main parts
- Path: is a segmented name (used opaquely by recipient); not to be interpreted as hierarchy; case sensitive
- Query: Separator is ‘;’, order of query is significant (when URI is used as index in proxy, caches etc), ‘filter’ is a reserved key



# Five Methods

# Let's look deeper at a Resource ...



Server - DevA

/oic/p	LED-Light-Co, 40W Equivalent Bulb	
/oic/d	Main Living Room Light	
/oic/res	/oic/p	oic.wk.p
	/oic/d	oic.wk.d
	/switch	oic.r.switch.binary
	/dimming	oic.r.light.brightness
/switch	true	
/dimming	100	



# There can be more Properties ...

/oic/p	pi	GoGlow III - 40W
	mmn	Led-Light-Co
	mnml	<a href="http://ledlight.com/gg3-40">http://ledlight.com/gg3-40</a>
	mnmo	GG3-40

# Resource Type

```
"oic.r.switch.binary": {  
  "type": "object",  
  "properties": {  
    "value": {  
      "type": "boolean",  
      "description": "Status of the switch"  
    }  
  }  
}
```

```
get:  
  responses :  
    200:  
      body:  
        application/json:  
          schema: |  
            {  
            }  
          }
```

Defined by:  
(see *OIC\_Resource\_Type\_Specification*)

## JSON Schema\*

- **Describes the Properties and Representation**

## ReSTful API Modelling Language (RAML)\*

- **Describes the Request and Response**

Table 19. oic.wk.p resource type definition

OCF  
Specification  
defines  
“oic.wk.p”  
Resource  
Type for  
platform and  
its list of  
Properties

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Platform ID	pi	string			R	yes	Platform Identifier
Manufacturer Name	mnmn	string			R	yes	Name of manufacturer (not to exceed 16 characters)
Manufacturer Details Link (URL)	mnml	URL			R	no	URL to manufacturer (not to exceed 32 characters)
Model Number	mnmo	string			R	no	Model number as designated by manufacturer
Date of Manufacture	mndt	date		Time ( <i>show RFC</i> )	R	no	Manufacturing date of device
Platform Version	mnpv	string			R	no	Version of platform – string (defined by manufacturer)
OS Version	mnos	string			R	no	Version of platform resident OS – string (defined by manufacturer)
Hardware Version	mnhw	string			R	no	Version of platform hardware
Firmware version	mnfv	string			R	no	Version of device firmware
Support URL	mnsi	URL			R	no	URL that points to support information from manufacturer
SystemTime	st	datetime			R	no	Reference time for the device

## Table 12. List of OIC Core Resources

And ...

Additional  
core  
resources

CR:  
conditionally  
required (i.e.  
if capability is  
supported)

Fixed URI	Resource Type Title	Related Functional Interaction	Requirement (M/CR/O)
/oic/res	Default	Discovery	M
/oic/p	Platform	Discovery	M
/oic/d	Device	Discovery	M
/oic/rts	Resource Type	Discovery	CR
/oic/ifs	Interface	Discovery	CR
/oic/con	Configuration	Device Management	CR
/oic/mon	Monitoring	Device Management	CR
/oic/mnt	Maintenance	Device Management	CR

# Structure

## Complexity

## Context

## Behaviours

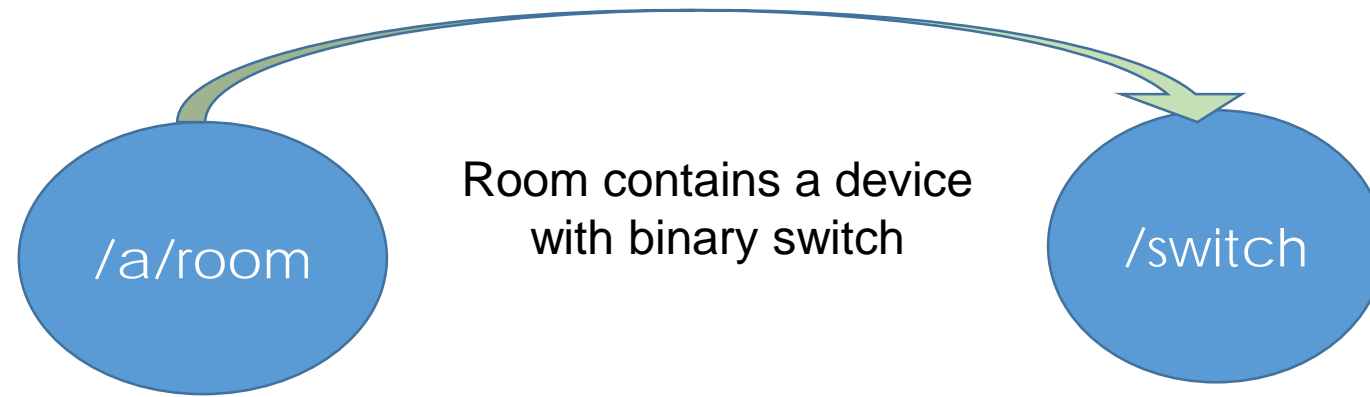
# How to represent ...

- Lights in a room
- All the devices in a room
- Rooms on a floor
- All floors in a building
  
- Devices in a group
- Members in a family
  
- A star topology



# Link

How to define the connection between two resources?



```

{
  "href": "/switch",
  "rel": "contains",
  "anchor": "/a/room",
  "rt": "oic.r.switch.binary",
  "if": "oic.if.a",
  "bif": "oic.if.baseline"
}

```

← Target

← Relation

← Context

Parameters

# Collections

- A Resource that also has Links
- Links declared in value of the “links” Property

DevD		
/room	“name”	“Living Room”
	“links”	“href”: “ oic://<DevA>/oic/d” “rel”: “contains”, “rt”: “oic.d.light” “href”: “ oic://<DevB>/oic/d” , “rel”: “contains”, “rt”: “ oic.d.light”

Server - DevA		
/oic/p	LED-Light-Co, 40W Equivalent Bulb	
/oic/d	Main Living Room Light	
/oic/res	/oic/p	oic.wk.p
	/oic/d	oic.wk.d
	/light	oic.r.switch.binary
	/dimming	oic.r.light.brightness
/light	true	
/dimming	100	

Server - DevB		
/oic/p	LED-Light-Co, 40W Equivalent Bulb	
/oic/d	Living Room Table Lamp	
/oic/res	/oic/p	oic.wk.p
	/oic/d	oic.wk.d
	/light	oic.r.switch.binary
	/dimming	oic.r.light.brightness
/light	true	
/dimming	100	



# Collections ... example

**“Room” collection – room has lights and fan**

```

/my/room/1
{
  "rt": "acme.room",
  "if": ["oic.if.r", "oic.if.rw"],
  "color": "blue",
  "dimension": "15bx15wx10h",
  "links": [
    { "href": "/the/light/1", "rel": "contains", "rt": "acme.light", "if": ["oic.if.a", "oic.if.baseline"] },
    { "href": "/the/light/2", "rel": "contains", "rt": "mycorp.light", "if": ["oic.if.s", "oic.if.baseline"] },
    { "href": "/the/fan/1", "rel": "contains", "rt": "hiscorp.fan", "if": ["oic.if.baseline"] }
  ]
}

```

# Interfaces

- Interface provides a “view” into a Resource or Collection
- Interface defines allowed methods and semantics on that “view”
- OCF has predefined Interfaces

## Interface views

## “Room” collection – room has lights and fan



# Other Interfaces

Batch – “*oic.if.b*”

- Request forwarded to Link targets
- Single Response with aggregated “target” responses

Link Modify – “*oic.if.lm*” (*under review*)

Factory – “*oic.if.f*” (*under review*)

Parameter – “*oic.if.p*” and “*oic.if.rp*” (*under review*)

# Complexity, Context & Behaviours

- Also have
  - Scenes – Pre-defined settings for a number of resources  
*(more details in subsequent presentation)*
  - Rules – If This Then That
  - Scripts – A piece of code

# Navigating the Specs

- Covered in this presentation
  - Core
  - Resource Type Specification (SmartHome)
  - SmartHome Device Specification
- Not covered in this presentation
  - Security
  - Remote Access

Thank You