OPEN CONNECTIVITY FOUNDATION™

IoTivity Introduction

Martin Hsu

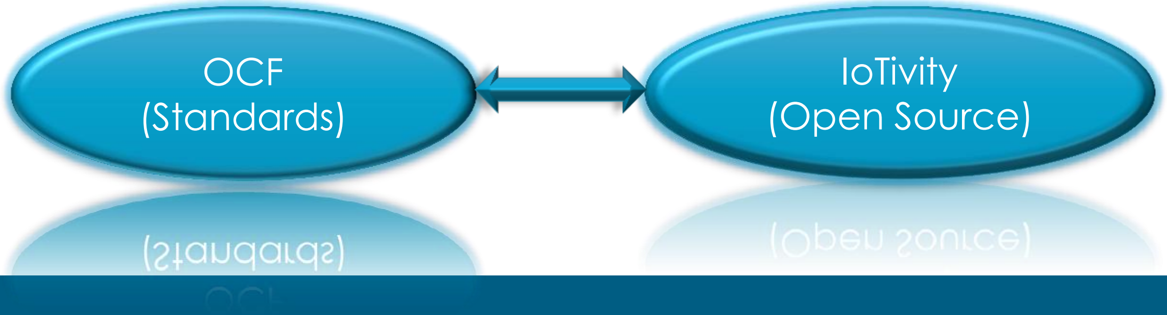
Intel Open Source Technology Center

Agenda

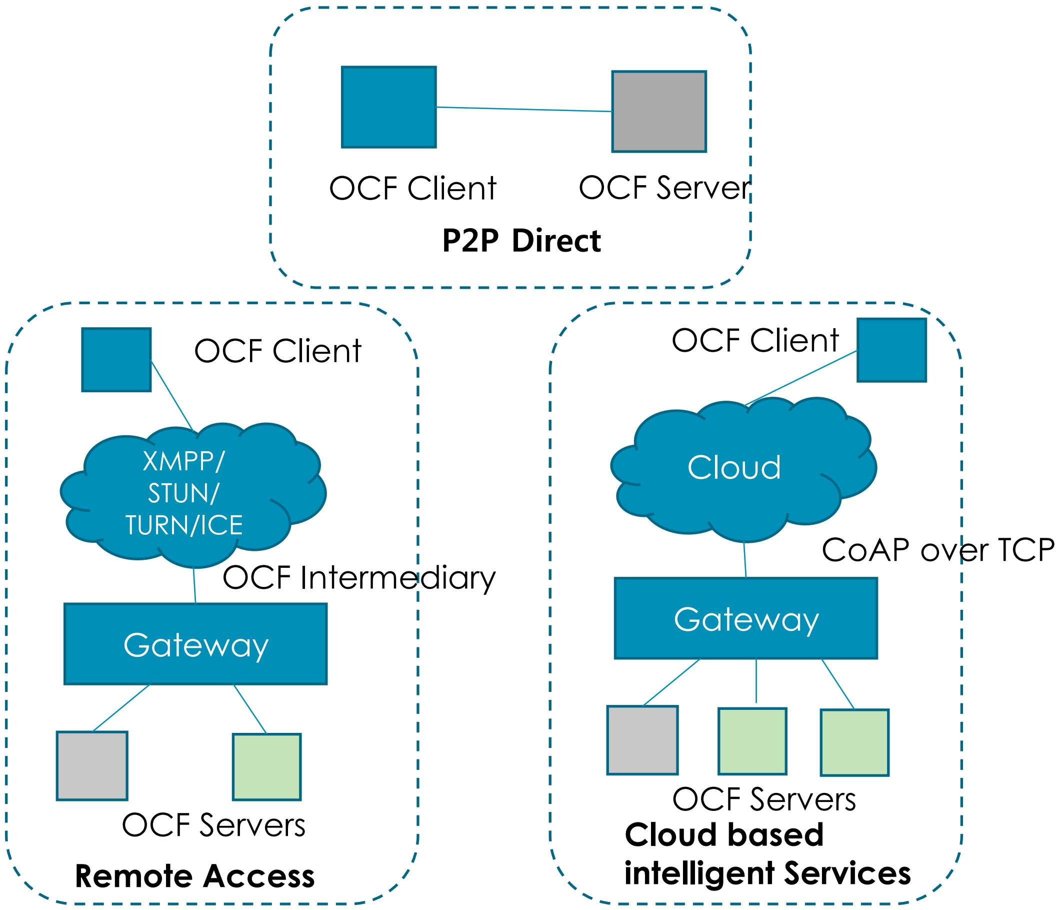
- Overview
- Architecture
- Base Layer & APIs
- Primitive Services & APIs

IoTivity Overview

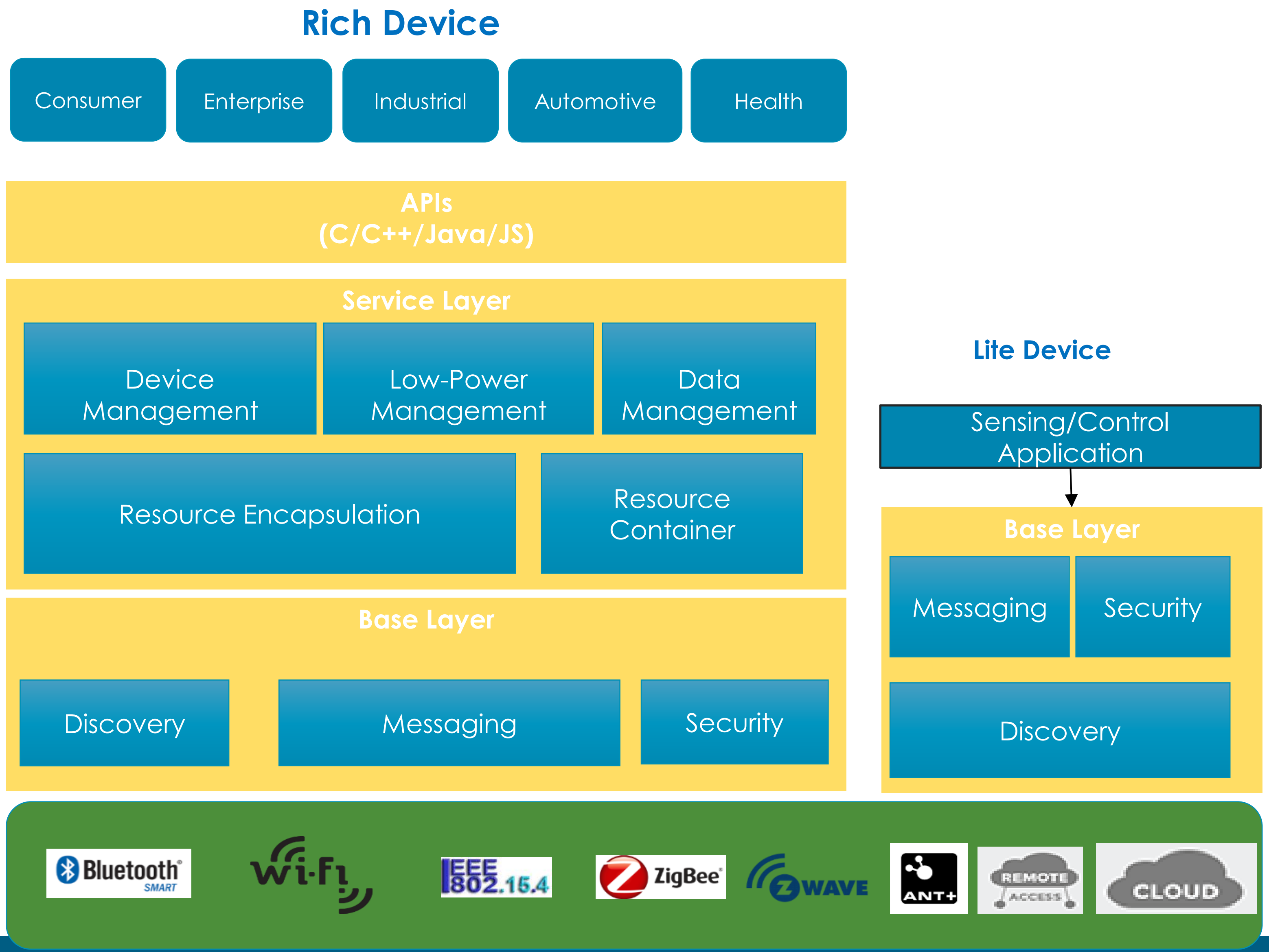
- An open source software framework implementing OCF Standards
- Ensures seamless device-to-device and device-to-cloud connectivity to address emerging needs of IoT
- Licensed under Apache License Version 2.0
- Available on TIZEN, Android, Arduino, Linux(Ubuntu) Platforms



OCF Topologies Supported



IoTivity – High Level Architecture



Rich Device

Consumer | Enterprise | Industrial | Automotive | Health

APIs (C/C++/Java/JS)

Service Layer

Device Management | Low-Power Management | Data Management

Resource Encapsulation | Resource Container

Base Layer

Discovery | Messaging | Security

Lite Device

Sensing/Control Application

Base Layer

Messaging | Security

Discovery

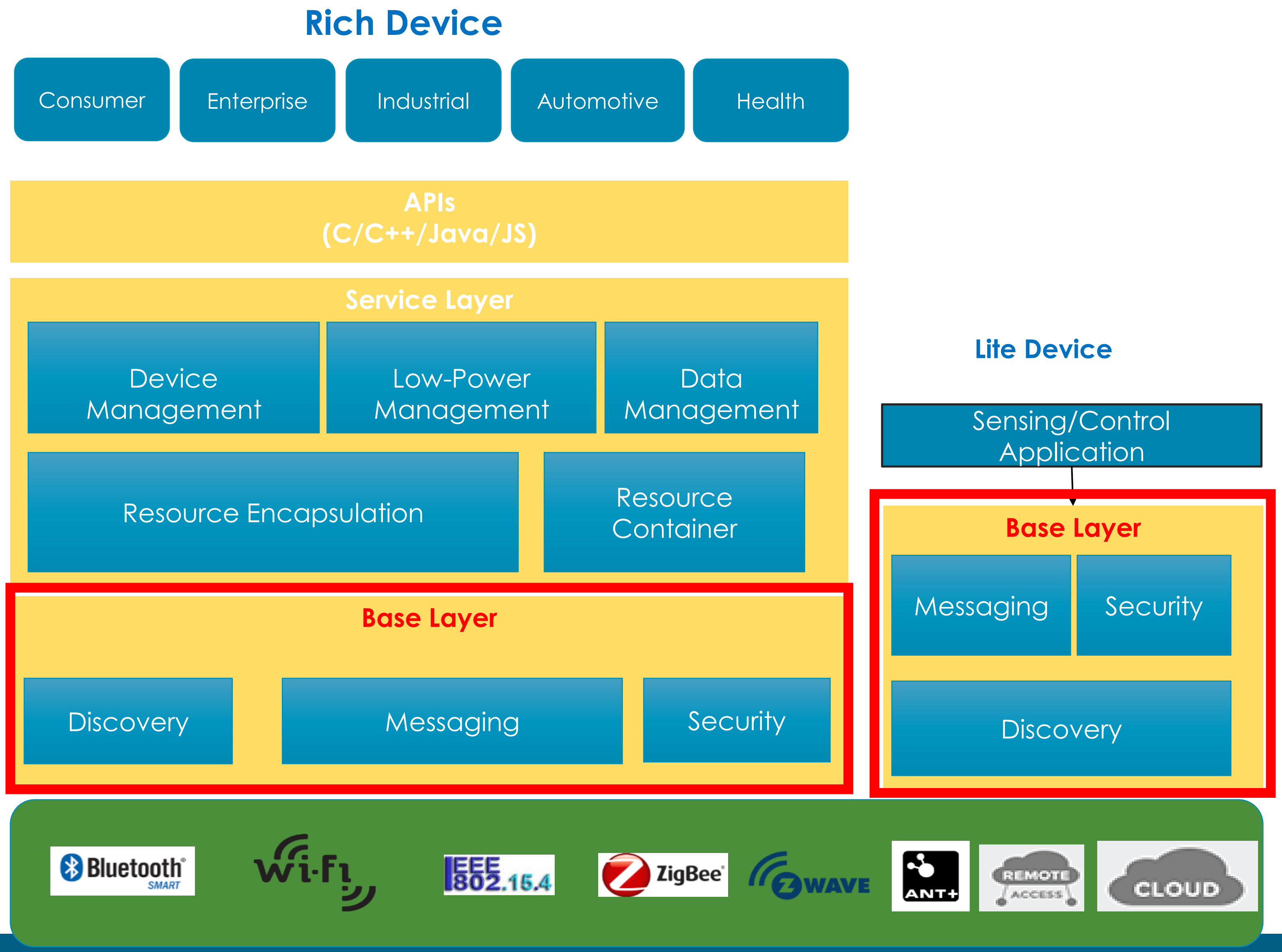
Key Goals

- ❖ Common Solution
- ❖ Established Protocols
- ❖ Security & Identity
- ❖ Standardized Profiles
- ❖ Interoperability
- ❖ Innovation Opportunities
- ❖ Necessary connectivity

🟦	IoTivity Profiles
🟨	IoTivity Framework
🟩	IoTivity Connectivities*

IoTivity Base Layer & APIs

IoTivity – High Level Architecture



Rich Device

Consumer | Enterprise | Industrial | Automotive | Health

APIs
(C/C++/Java/JS)

Service Layer

Device Management | Low-Power Management | Data Management

Resource Encapsulation | Resource Container

Base Layer

Discovery | Messaging | Security

Lite Device

Sensing/Control Application

Base Layer

Messaging | Security

Discovery

Key Goals

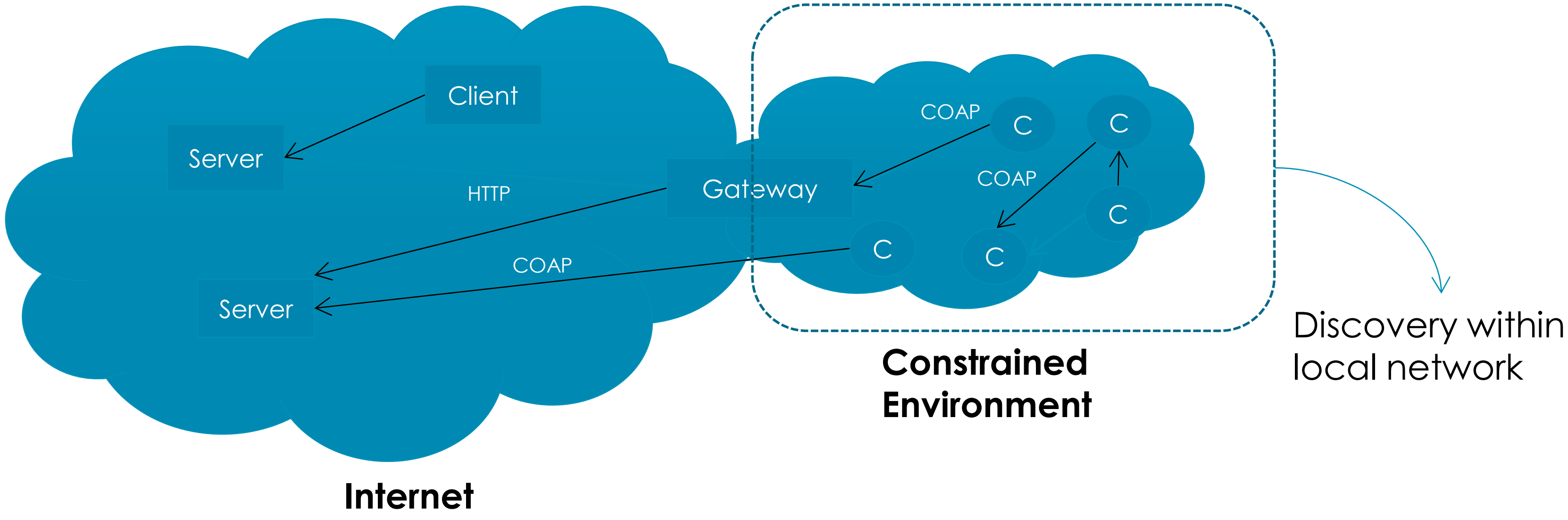
- ❖ Common Solution
- ❖ Established Protocols
- ❖ Security & Identity
- ❖ Standardized Profiles
- ❖ Interoperability
- ❖ Innovation Opportunities
- ❖ Necessary connectivity

🟦 IoTivity Profiles

🟨 IoTivity Framework

🟩 IoTivity Connectivities*

Discovery Subsystem



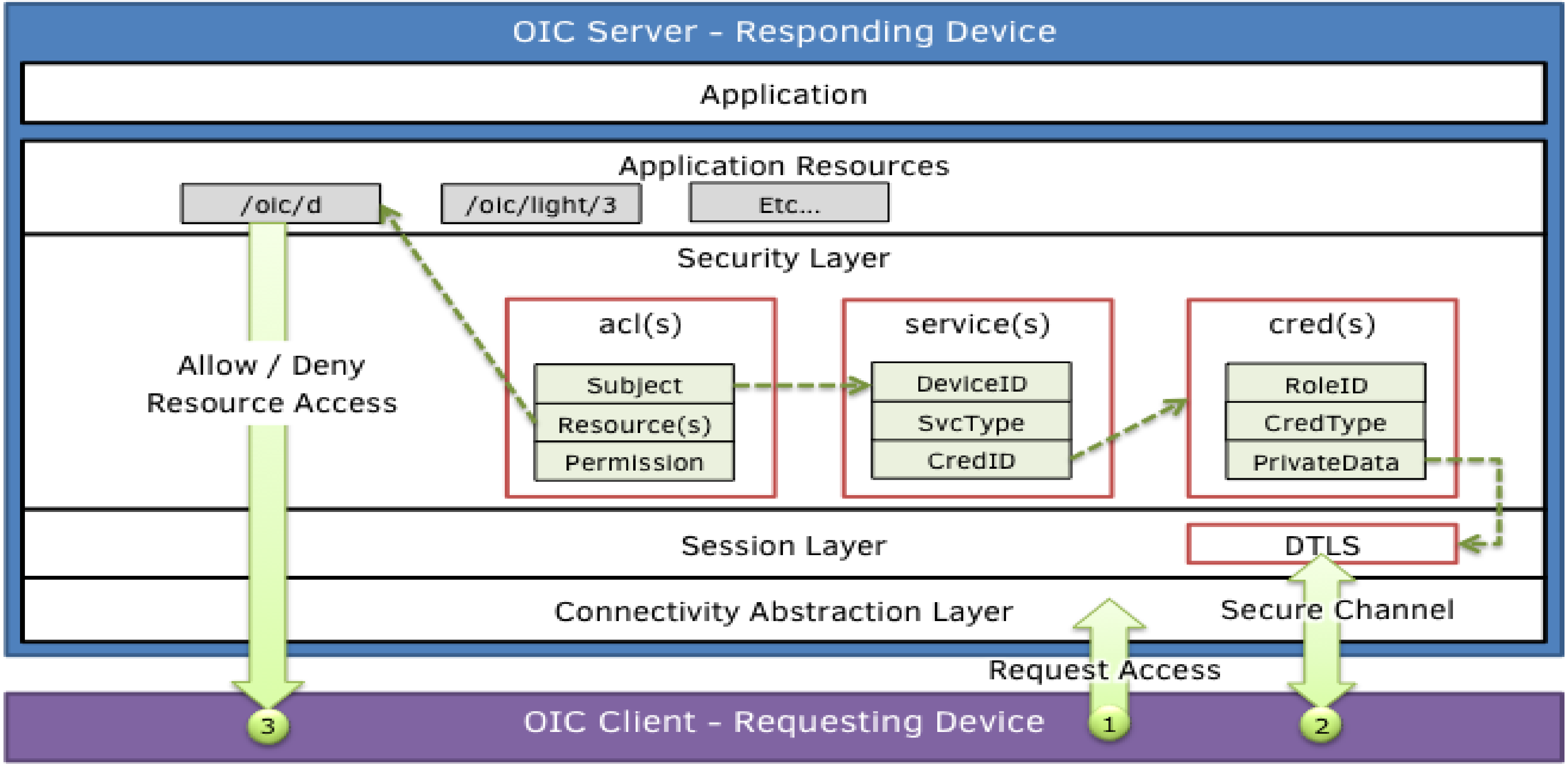
Connectivity	Discovery Mechanism	Description
WiFi & Ethernet (over IP)	IP Multicast	CoAP Multicast Port: 5683 (Assigned by IANA) CoAP Secure Port: 5684
	IP Unicast over UDP	Precondition: OIC Server Address & Port are known
Bluetooth (EDR & BLE)	Using Scan & Advertise	OCF Specific Service UUID

CoAP

- Open IETF Standard (RFC 7252)
- Compact 4 Byte Header
- UDP (Default), SMS, TCP Support
- Strong DTLS Security
- Asynchronous Subscription
- Built-In Discovery

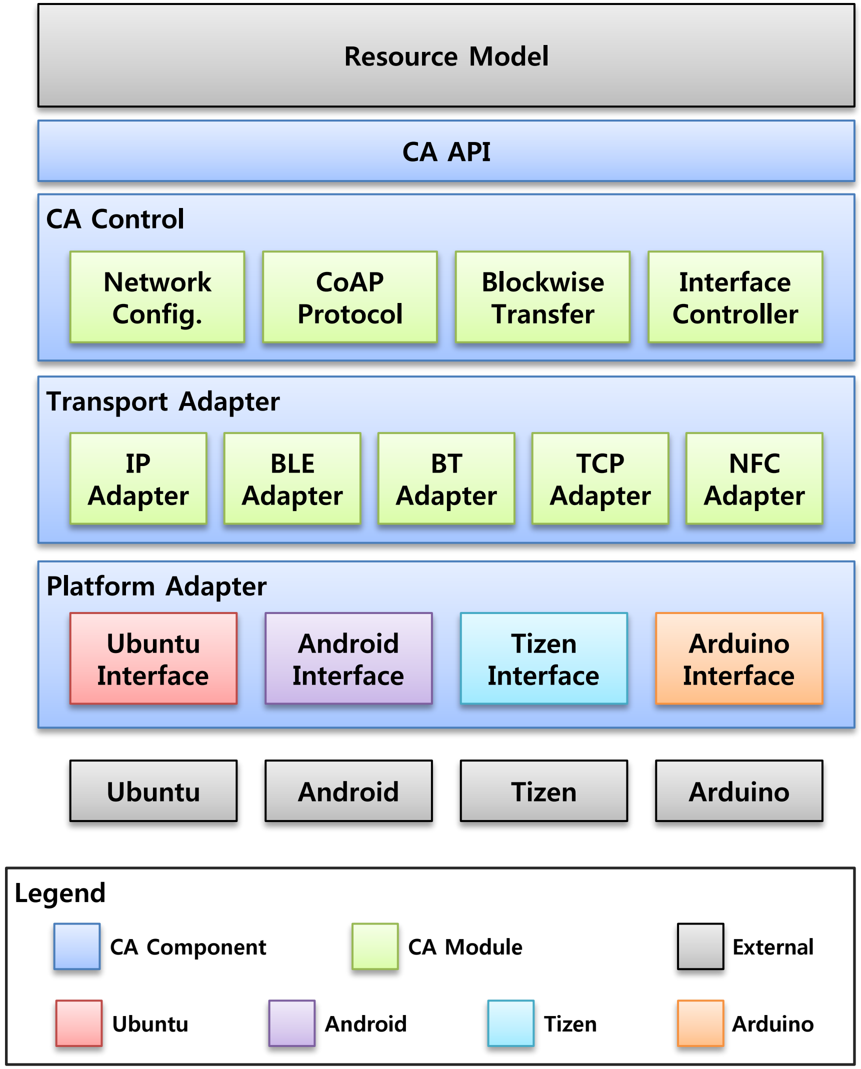
CoAP: Constrained Application Protocol
IANA: Internet Assigned Numbers Authority

Security Features & Architecture



Note platform hardening not part of the OCF Specs & IoTivity implementation.

Messaging - Connectivity Abstraction



CA Control Component

- Target network selection, interface control & monitoring
- CoAP message serialization & parsing
- Block-wise messaging flow control

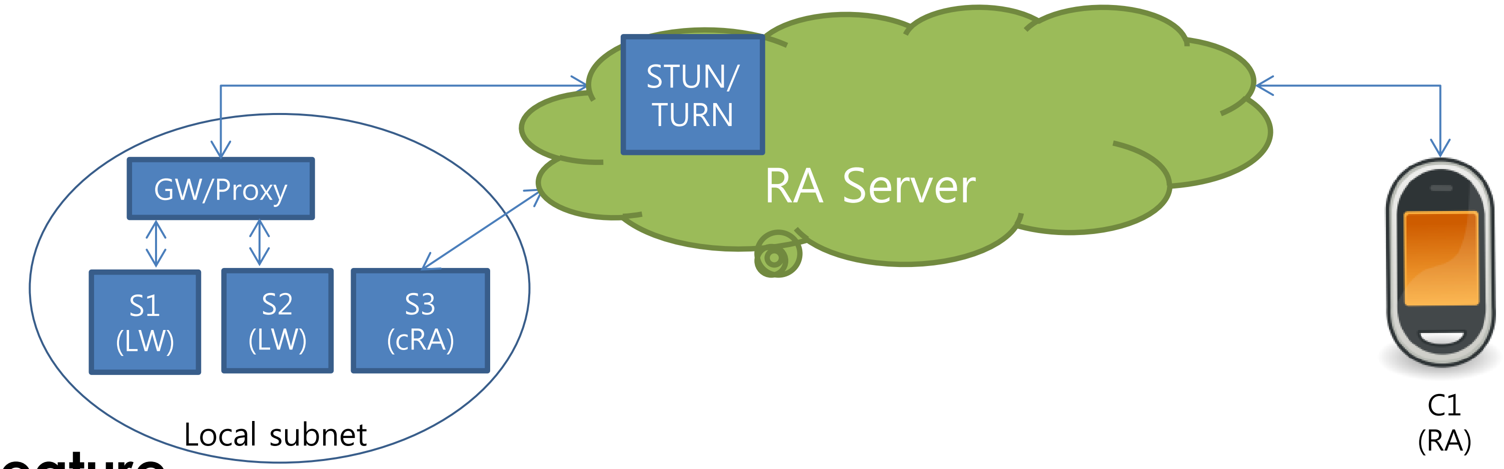
Transport Adapter Component

- Data transmission over UDP, TCP, BLE(GATT), BT(SPP) & NFC
- Secure data exchanging using DTLS

Platform Adapter Component

- Ubuntu, Wi-Fi, Ethernet and BLE
- Android Wi-Fi, BLE and BT
- Tizen Wi-Fi, BLE and BT
- Arduino Wi-Fi, Ethernet and BLE

Messaging - Remote Access over XMPP



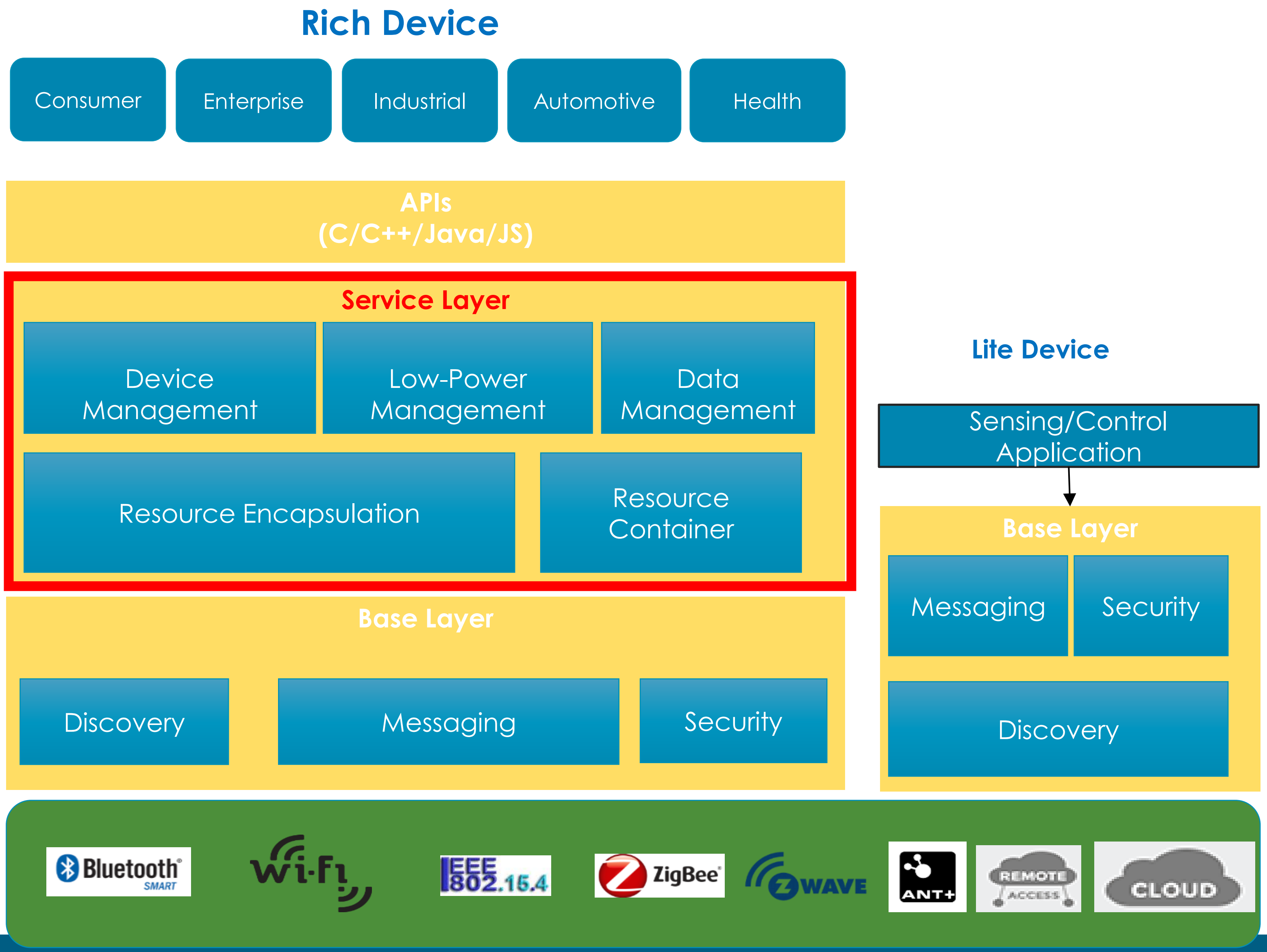
Feature

- Remote client discover & securely interface with resource servers when not on same subnet
- Adheres to access control policies
- End-to-End Secure

Device Type	Use Case
Light weight (LW) Device	Accessible within subnet. No RA, require GW/proxy device for access
Constrained RA (cRA) Endpoint	RA access for non latency-sensitive, low BW applications
RA Endpoint (RA)	Full RA access

IoTivity Primitive Services & APIs

IoTivity – High Level Architecture



Rich Device

Consumer | Enterprise | Industrial | Automotive | Health

APIs
(C/C++/Java/JS)

Service Layer

Device Management | Low-Power Management | Data Management

Resource Encapsulation | Resource Container

Base Layer

Discovery | Messaging | Security

Lite Device

Sensing/Control Application

Base Layer

Messaging | Security

Discovery

Key Goals

- ❖ Common Solution
- ❖ Established Protocols
- ❖ Security & Identity
- ❖ Standardized Profiles
- ❖ Interoperability
- ❖ Innovation Opportunities
- ❖ Necessary connectivity

🟦 IoTivity Profiles

🟨 IoTivity Framework

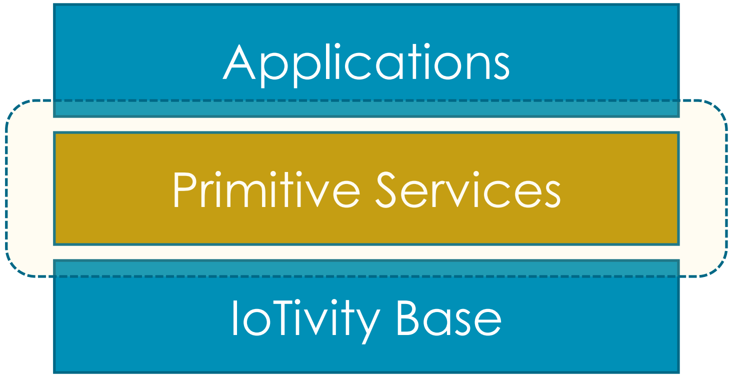
🟩 IoTivity Connectivities*

Purpose of Primitive Services

❖ Provides easier and simpler APIs for App developers (Heavy Lifting done by Framework)

❖ Mostly designed to run on Smart or Controller devices

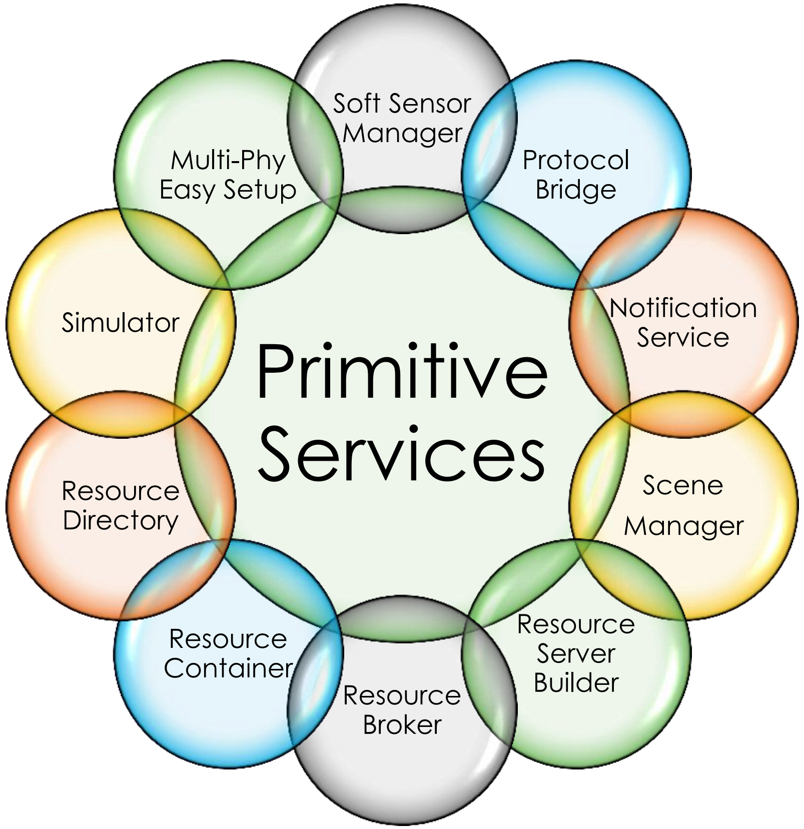
❖ Uses the Iotivity Base APIs



Applications

Primitive Services

IoTivity Base



Primitive Services

Soft Sensor Manager

Protocol Bridge

Notification Service

Scene Manager

Resource Server Builder

Resource Broker

Resource Container

Resource Directory

Simulator

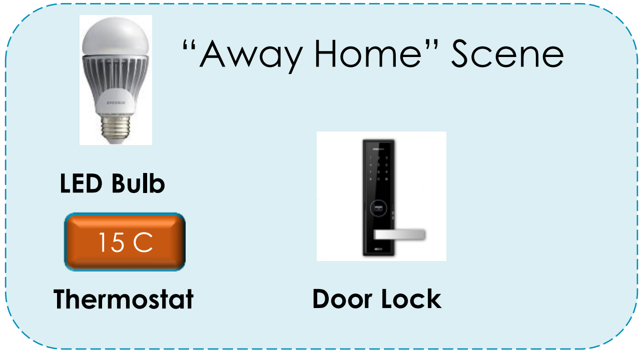
Multi-Phy Easy Setup

Scene Manager

Helps Users to create a Scenario or Scene for controlling Multiple IoT devices & their functionality

e.g.

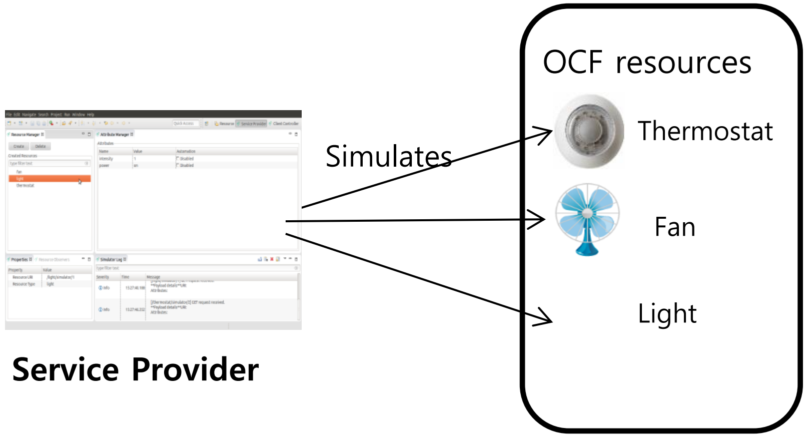
- Away Home – All Lights turned off, Doors locked
- Watching Movie – Living Room lights off, TV On, Speaker On



“Away Home” Scene

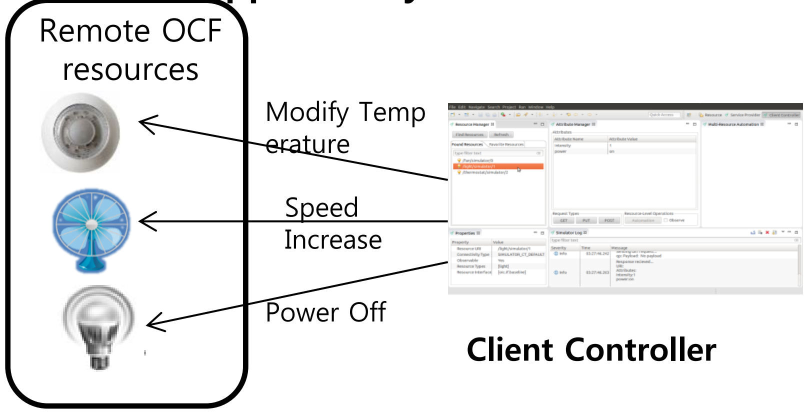
Simulator Service

Simulating different OCF resources



Service Provider

Sending different requests to verify features supported by OCF resources



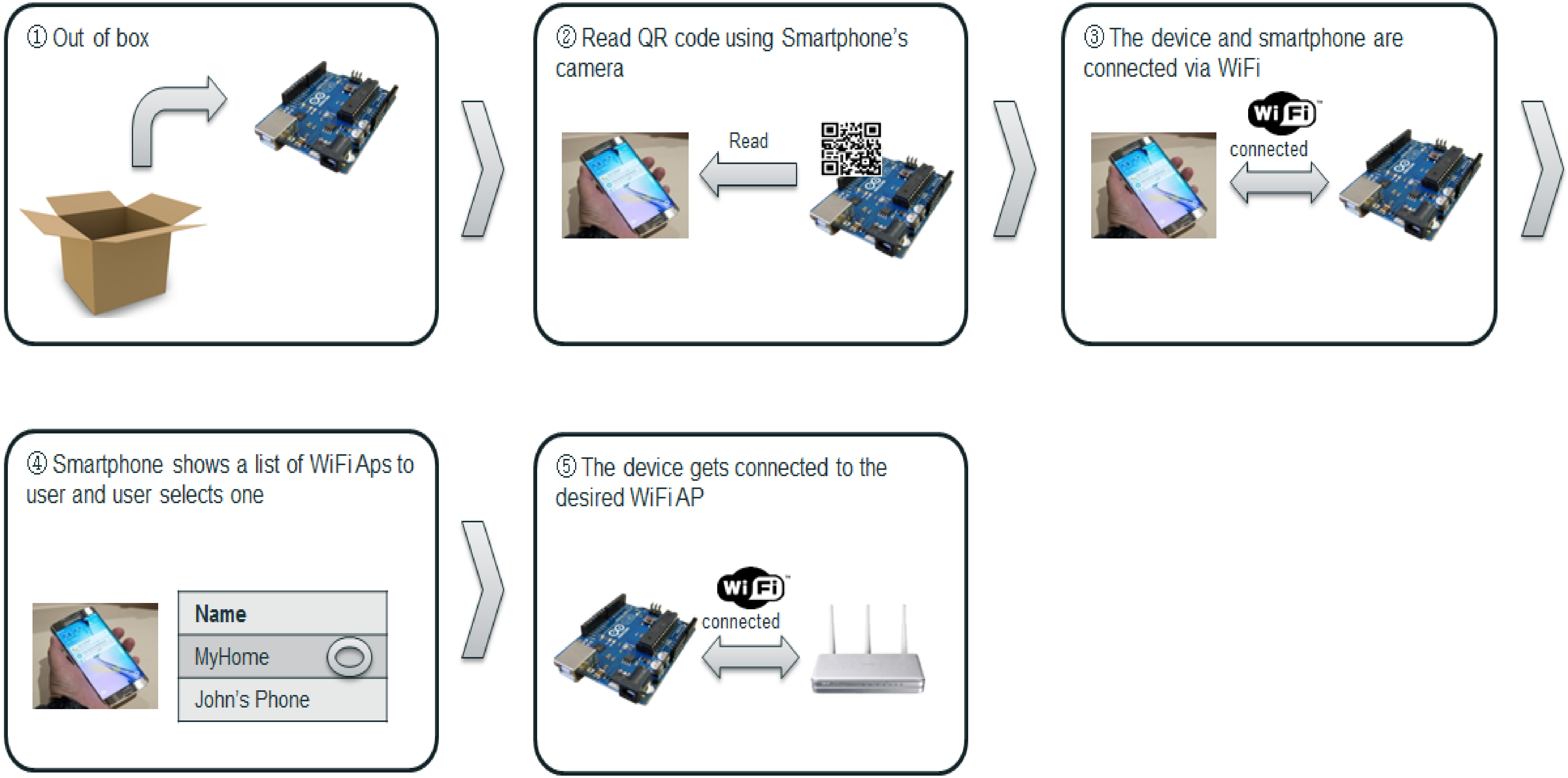
Client Controller

Feature

- **Server**
  - OCF resources can be simulated, Using resource model definition (RAML) files.
  - Manages creation, deletion, request handling and notifications for OCF resources.
- **Client**
  - Searching for different types of resources available in the network.
  - Sending different types of requests both manual and automatically and displays the response payload received.

MultiPhy Easy Setup

- Schematic illustrations



① Out of box

② Read QR code using Smartphone's camera

③ The device and smartphone are connected via WiFi

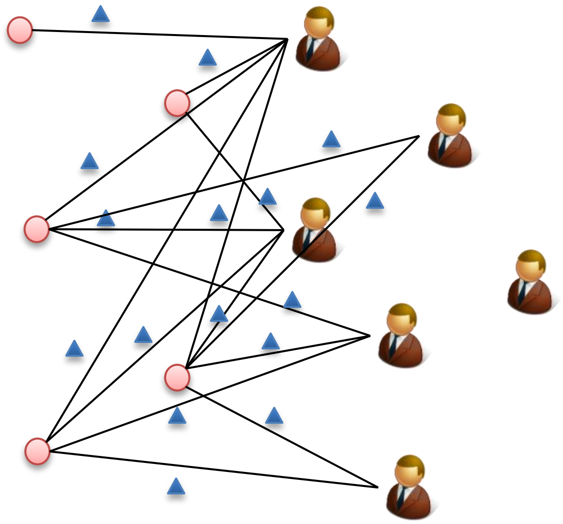
④ Smartphone shows a list of WiFi Aps to user and user selects one

Name
MyHome ○
John's Phone

⑤ The device gets connected to the desired WiFi AP

Low Power Management – Resource Hosting

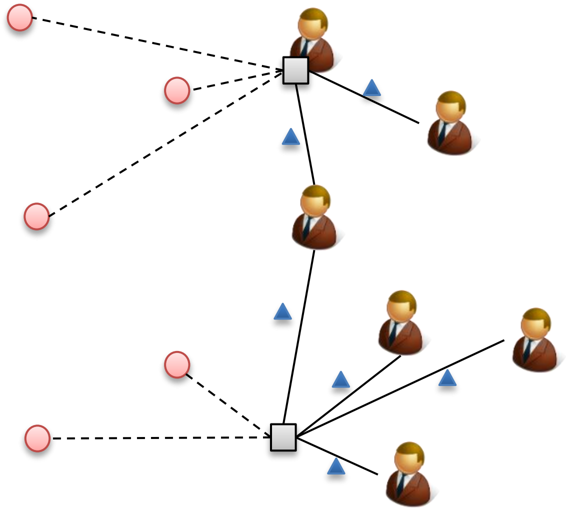
Problem



How many subscriptions thin device could support with its constrained system resource?

○ Thin(Light) device ▲ Subscription 👤 User/Consumer

Solution



Thin Device enhances its lifetime delegating its resource subscriber to richer hosting device

□ Hosting(Rich) device

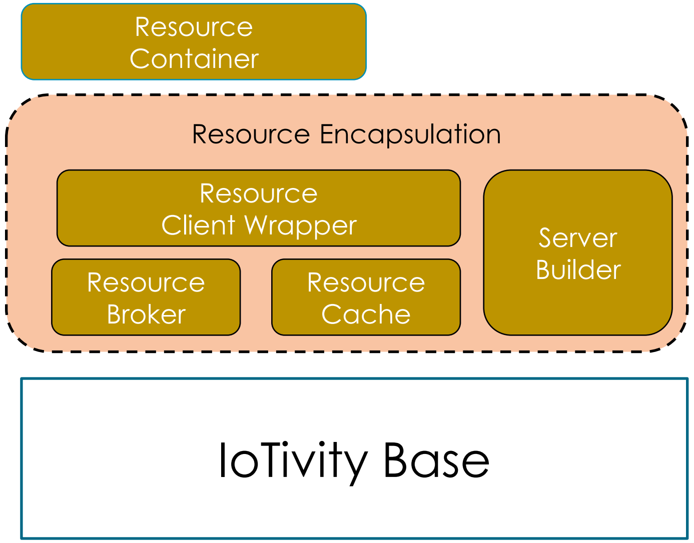
○ Thin(Light) device

▲ Subscription 👤 User/Consumer

- ❖Offloads request/data handling from remote clients
- ❖Reduces the power consumption of resource constraint device

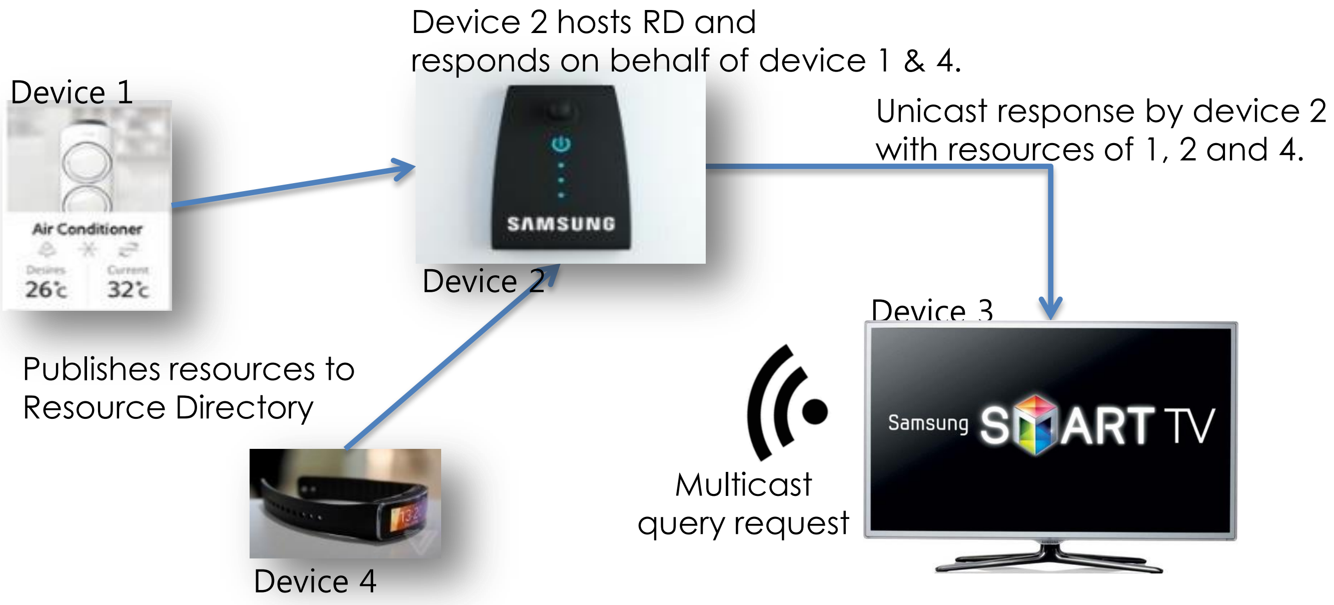
Supporting Material

Resource Encapsulation



Module	Description
Resource Broker	• Remote Resource Presence check (regardless of Remote Server supporting presence feature) • Provide consistent reachability management for discovered resource of interest
Resource Cache	• Maintains last information of Remote Resource (regardless of Remote Server is observable) • Data Centric API (Send/Recv Message Getter/Setter, Data Cache)
Server Builder	• Att. setter to provide easy way to create resource • Changes "msg Handling" to "Data Setting" for users • Monitors value of attributes so that notify-back for observation whenever attribute has changed

Low Power Management – Resource Directory



Device 2 hosts RD and responds on behalf of device 1 & 4.

Device 1

Unicast response by device 2 with resources of 1, 2 and 4.

Device 2

Device 3

Publishes resources to Resource Directory

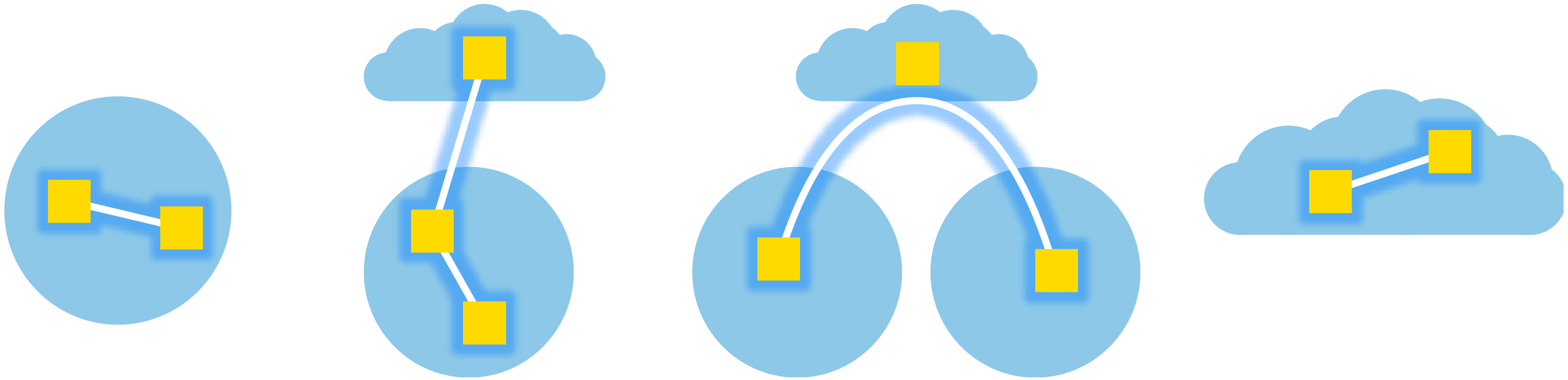
Multicast query request

Device 4

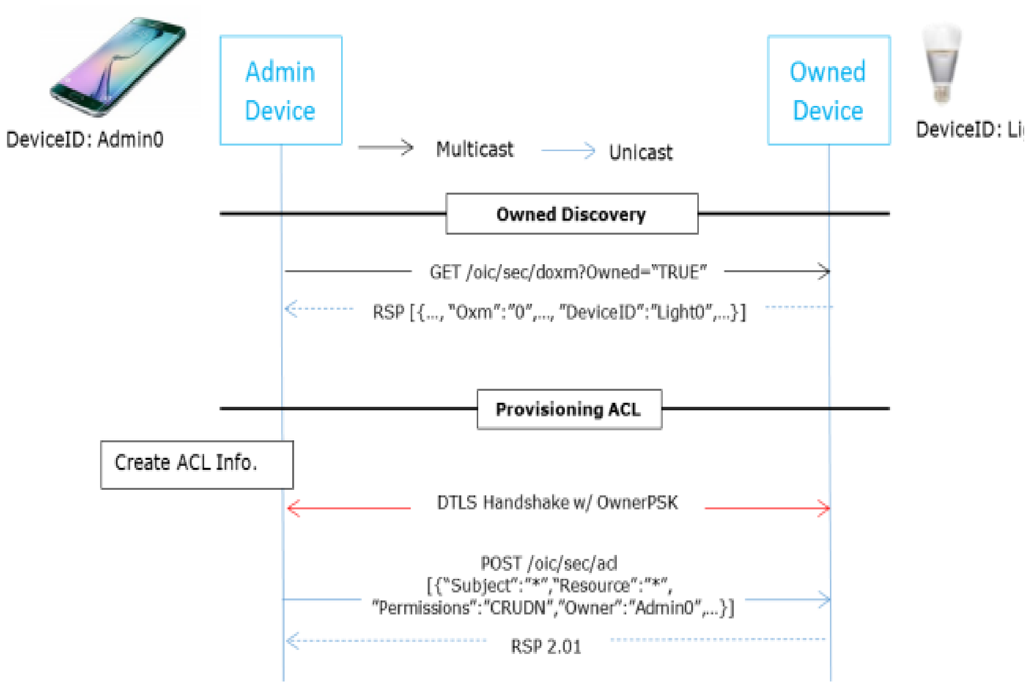
- Constrained device that needs to sleep and can not respond to multicast discovery queries
  - Discovery of RD server
  - Publish Resource to RD
  - Update / Delete Resource

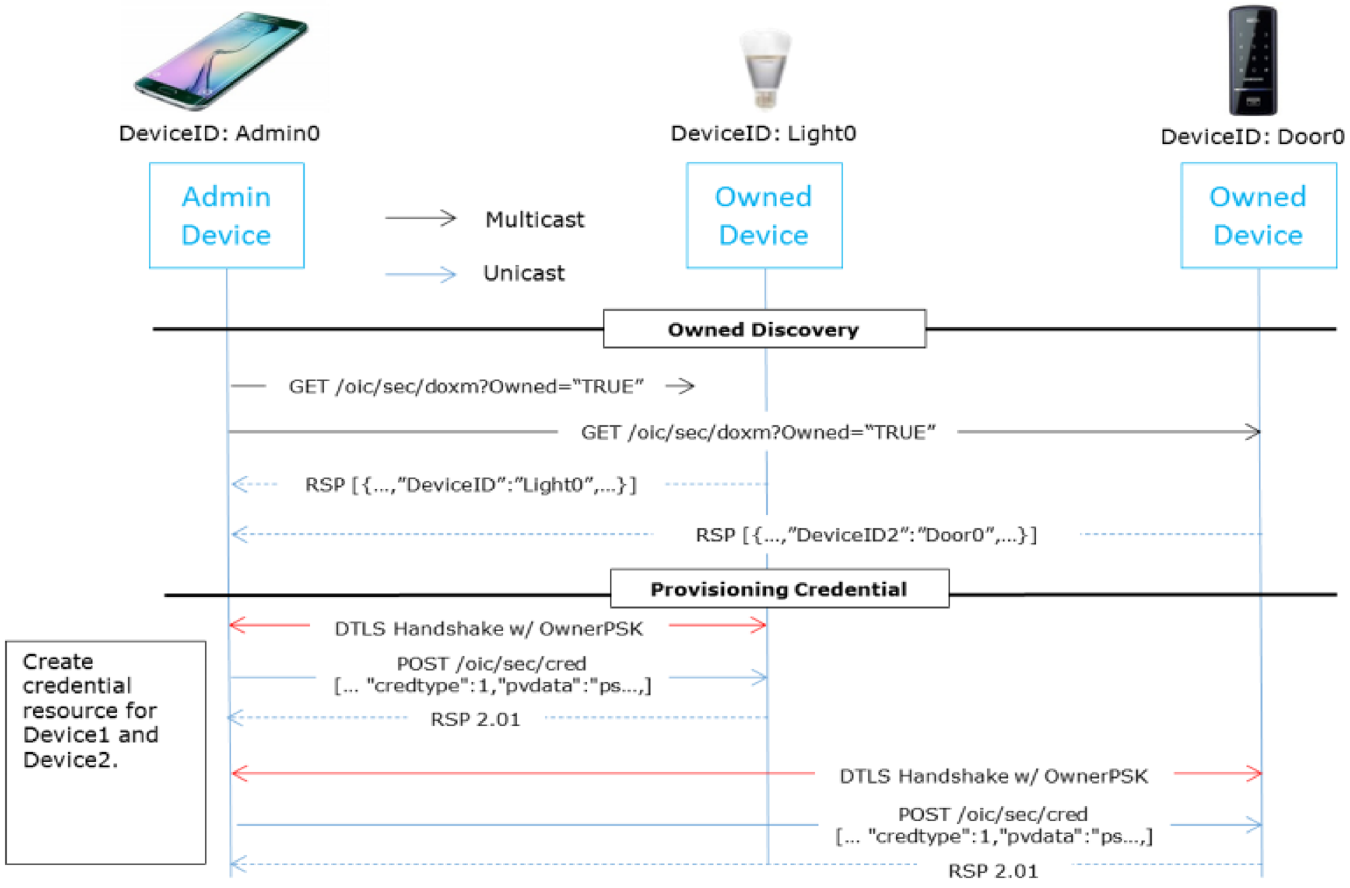
Cloud-Native Architecture for IoT

- Devices can reach the cloud directly.
  - Devices can self-organize if the cloud is not accessible.
- Architecture and protocols don't have to be replaced when device deployment changes from local-only to cloud-connected.
- Encourages end-to-end micro-services.

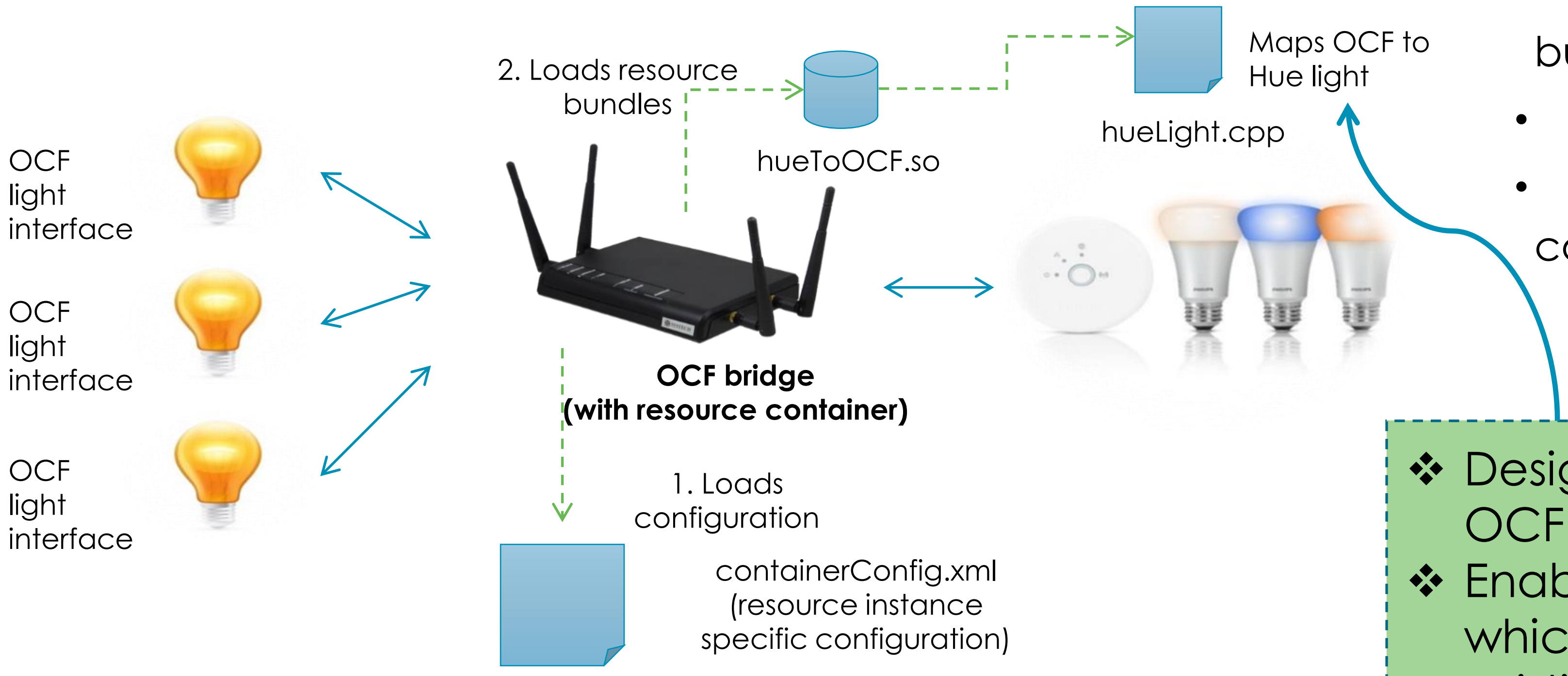


Onboarding & Provisioning Call Flow


DeviceID: Admin0
Admin Device
Owned Device
DeviceID: Li
Multicast
Unicast
Owned Discovery
GET /oic/sec/doxm?Owned="TRUE"
RSP [{..., "Oxm":"0",..., "DeviceID":"Light0",...}]
Provisioning ACL
Create ACL Info.
DTLS Handshake w/ OwnerPSK
POST /oic/sec/acl
[{"Subject":"*","Resource":"*",
"Permissions":"CRUDN","Owner":"Admin0",...}]
RSP 2.01


DeviceID: Admin0
DeviceID: Light0
DeviceID: Door0
Admin Device
Owned Device
Owned Device
Multicast
Unicast
Owned Discovery
GET /oic/sec/doxm?Owned="TRUE"
GET /oic/sec/doxm?Owned="TRUE"
RSP [{...,"DeviceID":"Light0",...}]
RSP [{...,"DeviceID2":"Door0",...}]
Provisioning Credential
Create credential resource for Device1 and Device2.
DTLS Handshake w/ OwnerPSK
POST /oic/sec/cred
[... "credtype":1,"pvdata":"ps...,]
RSP 2.01
DTLS Handshake w/ OwnerPSK
POST /oic/sec/cred
[... "credtype":1,"pvdata":"ps...,]
RSP 2.01

Protocol Bridge using Resource Container



OCF light interface

OCF light interface

OCF light interface

2. Loads resource bundles

hueToOCF.so

hueLight.cpp

Maps OCF to Hue light

OCF bridge (with resource container)

1. Loads configuration

containerConfig.xml (resource instance specific configuration)

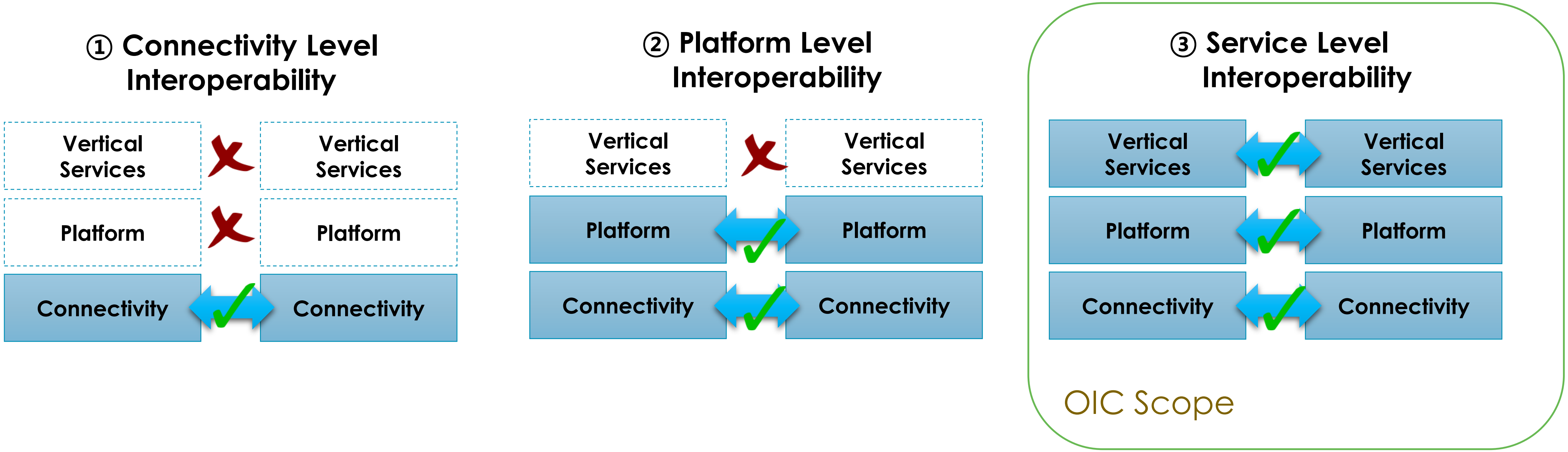
- Integrates non-OCF resources (Bundle)
- Handles dynamic loading of resource bundles & dynamic creation of resources
- Supports C++ .so files & Java .jar files
- Common configuration for bundles and configured resources

❖ Designed to work devices with non OCF devices

❖ Enables control of legacy devices which are already in market with existing APIs using a OIC complaint device

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

Vertical Services	✗	Vertical Services
Platform	✗	Platform
Connectivity	✓	Connectivity

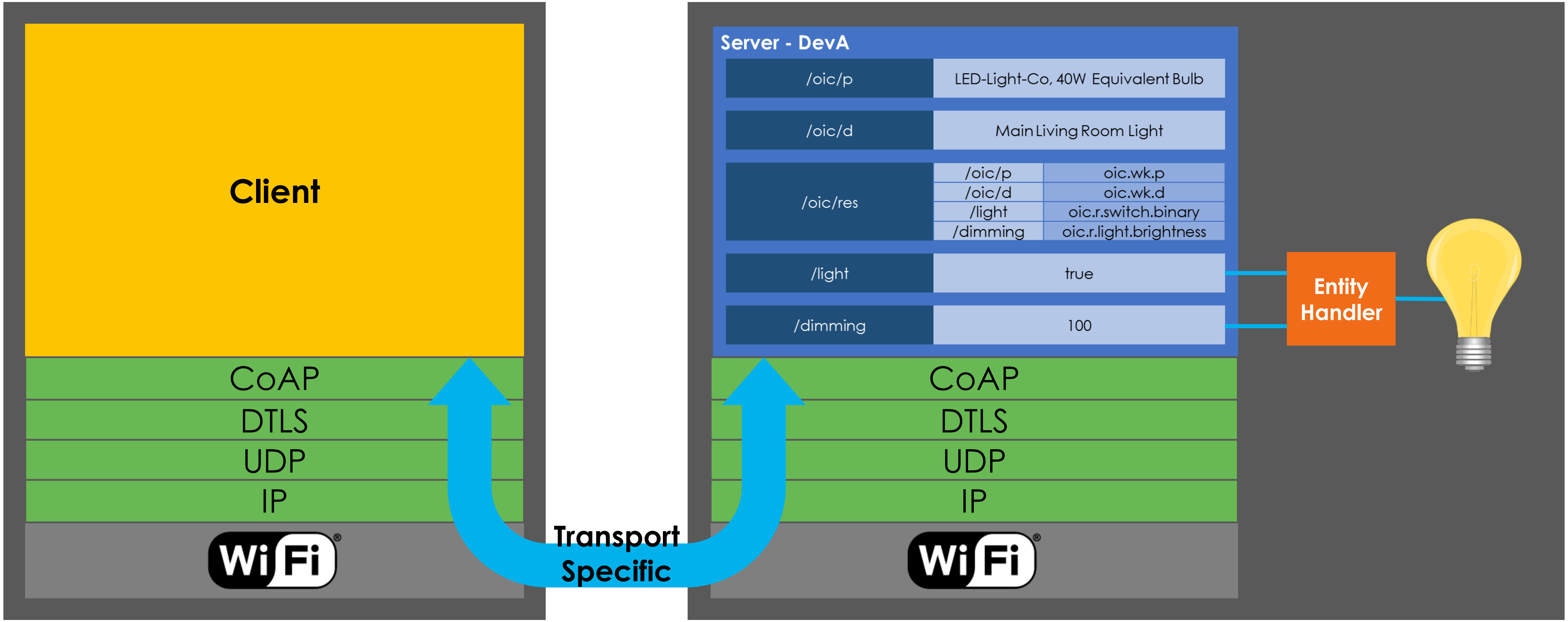
② Platform Level Interoperability

Vertical Services	✗	Vertical Services
Platform	✓	Platform
Connectivity	✓	Connectivity

③ Service Level Interoperability

Vertical Services	✓	Vertical Services
Platform	✓	Platform
Connectivity	✓	Connectivity

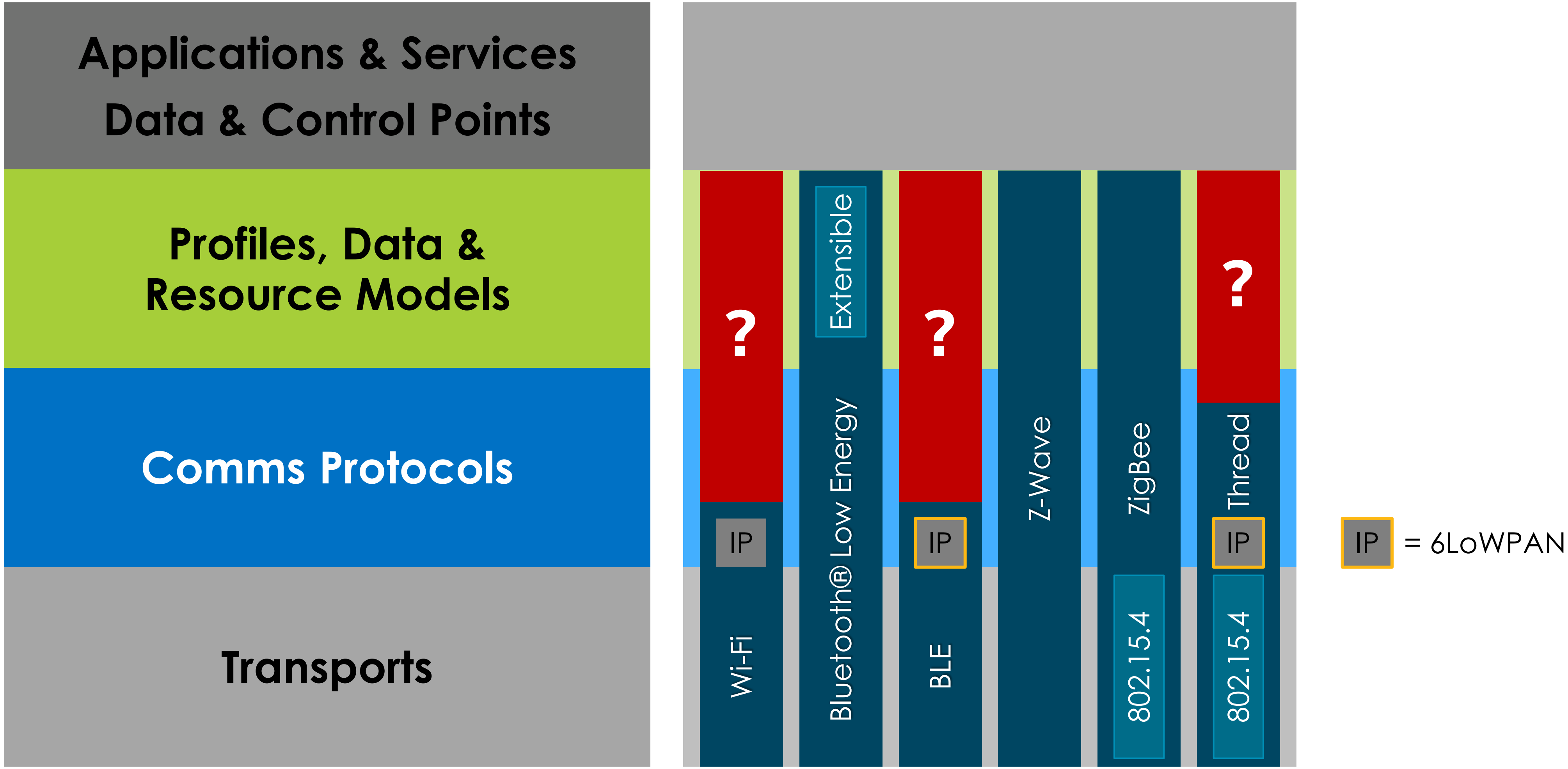
OIC Scope



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	

Consumer Radio-Based Standards



Applications & Services
Data & Control Points

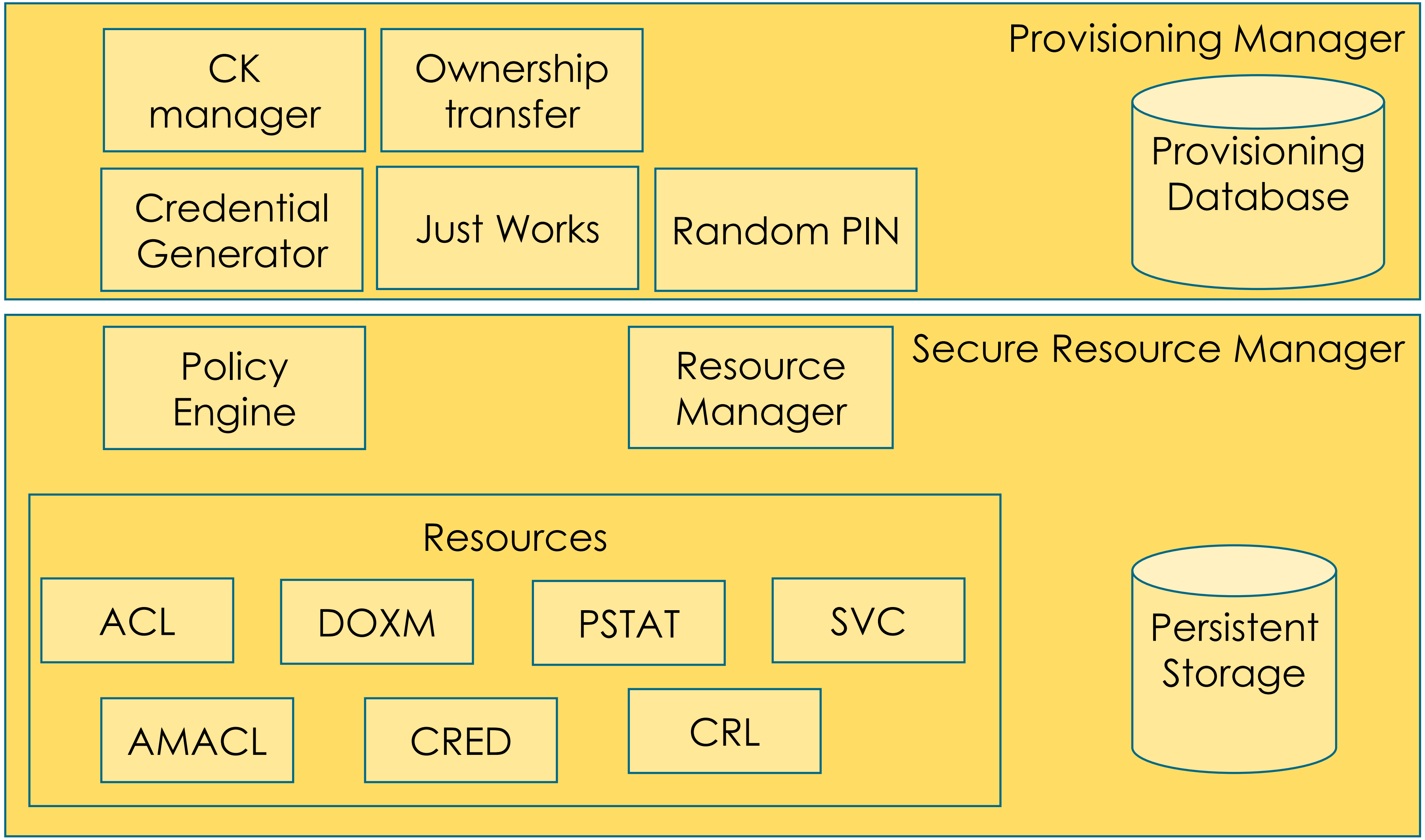
Profiles, Data & Resource Models

Comms Protocols

Transports

IP = 6LoWPAN

Security Building Blocks



Provisioning Manager

CK manager

Ownership transfer

Credential Generator

Just Works

Random PIN

Provisioning Database

Secure Resource Manager

Policy Engine

Resource Manager

Resources

ACL

DOXM

PSTAT

SVC

AMACL

CRED

CRL

Persistent Storage

OPEN CONNECTIVITY FOUNDATION™