Introduction of OIC standard

January, 2016

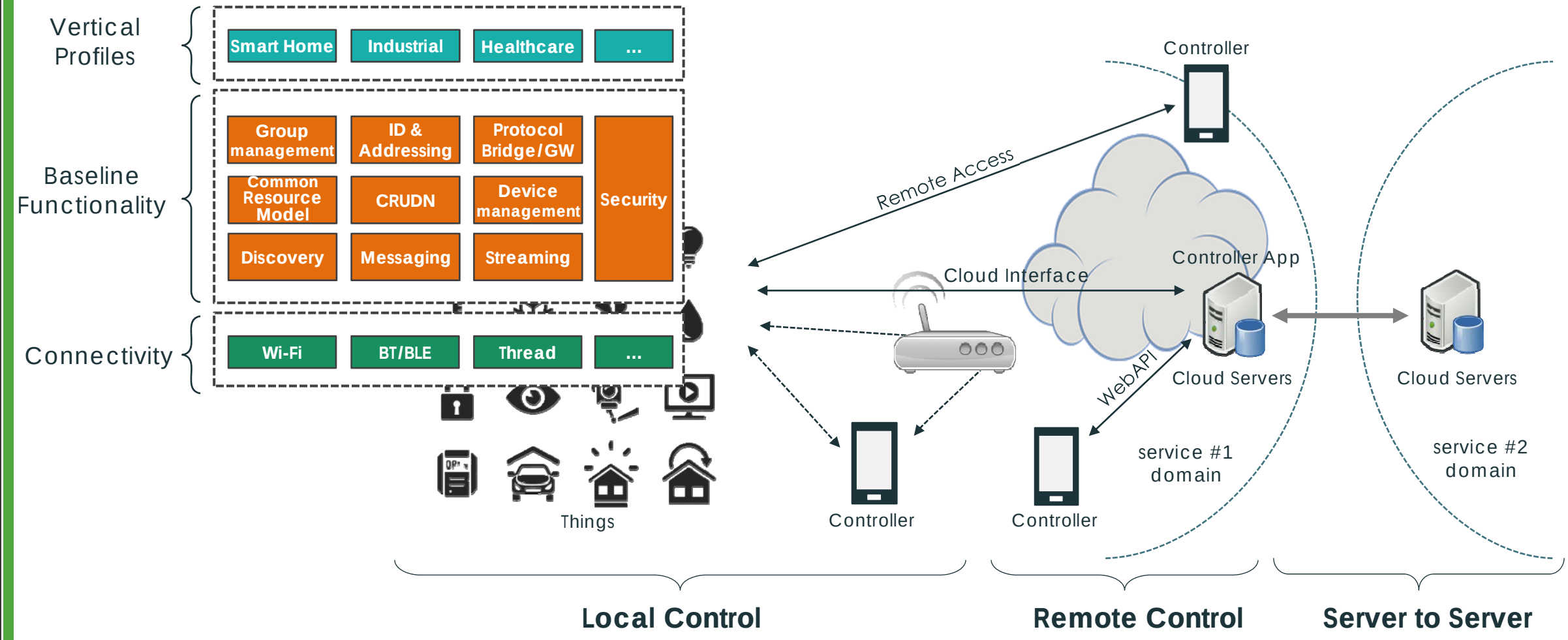
Standard Working Group
Open Interconnect Consortium

Table of Contents

Technical Principles for an Internet of Things Ecosystem

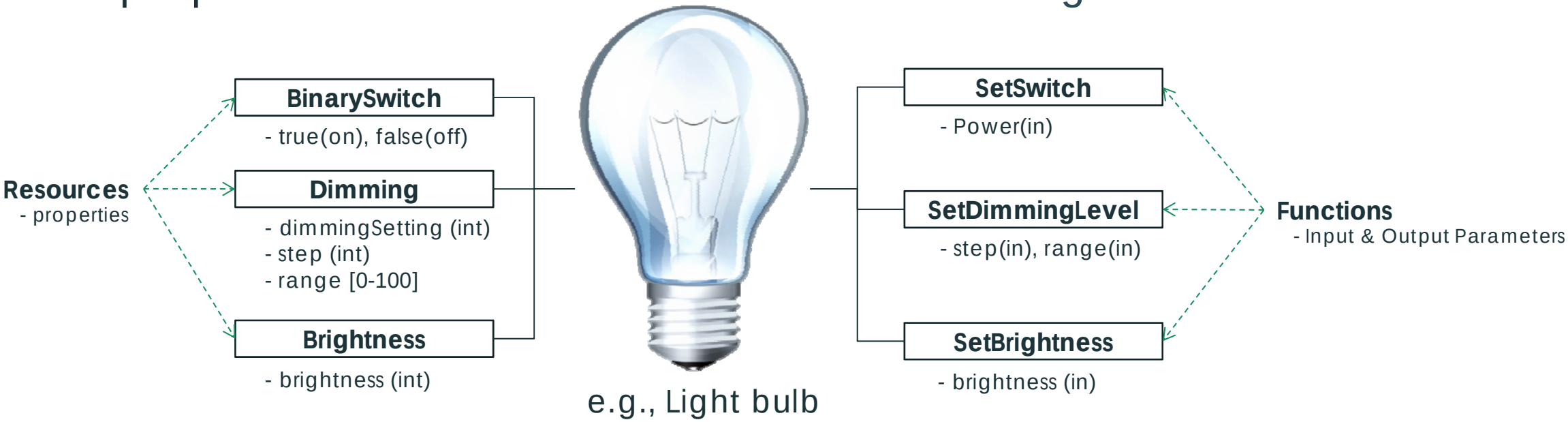
OPEN INTERCONNECT CONSORTIUM℠

Scope of IoT


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

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

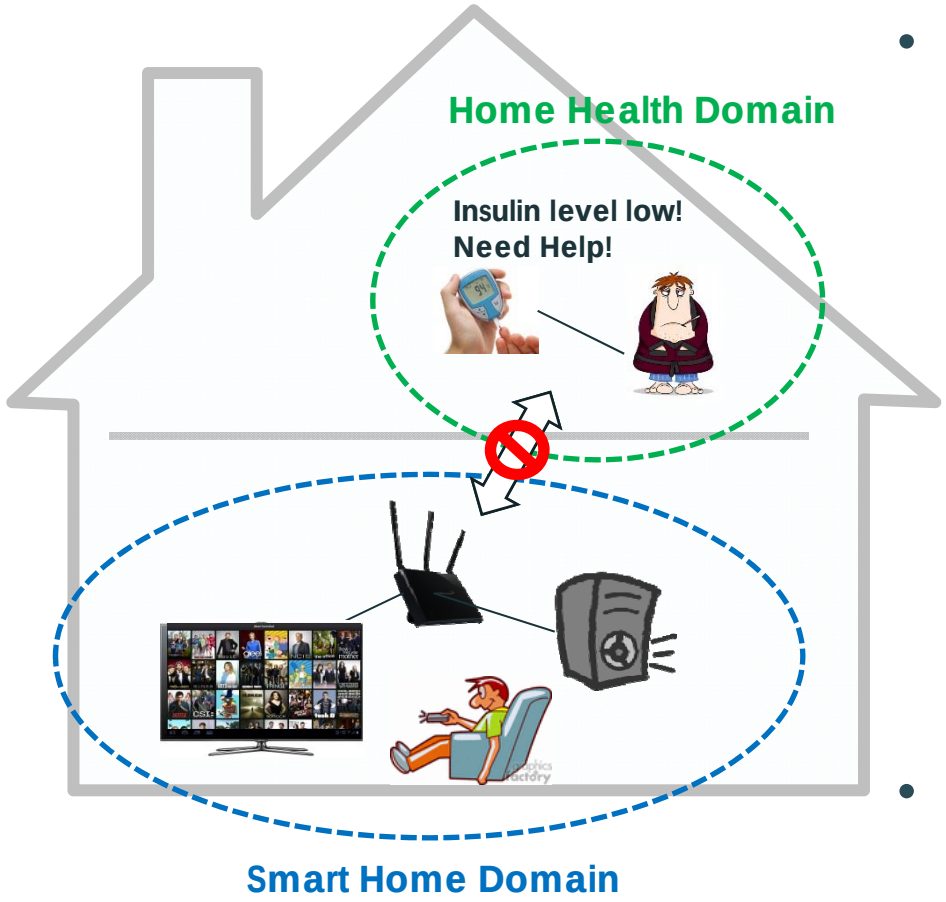
Support of Constrained Things

*RAM <10KB, Flash <100KB (RFC 7228)

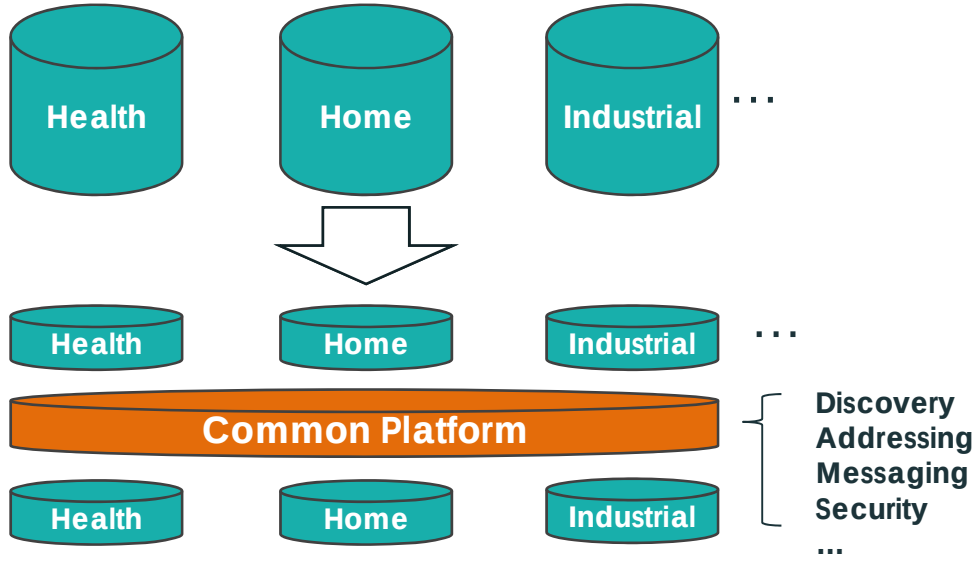
- Less overhead/ Less Traffic
  - Minimize CPU Load, Memory impacts, Traffic and Bandwidth
    - Compact header
    - Binary protocol
    - Compressed encoding of payload

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

Support of Multiple Verticals



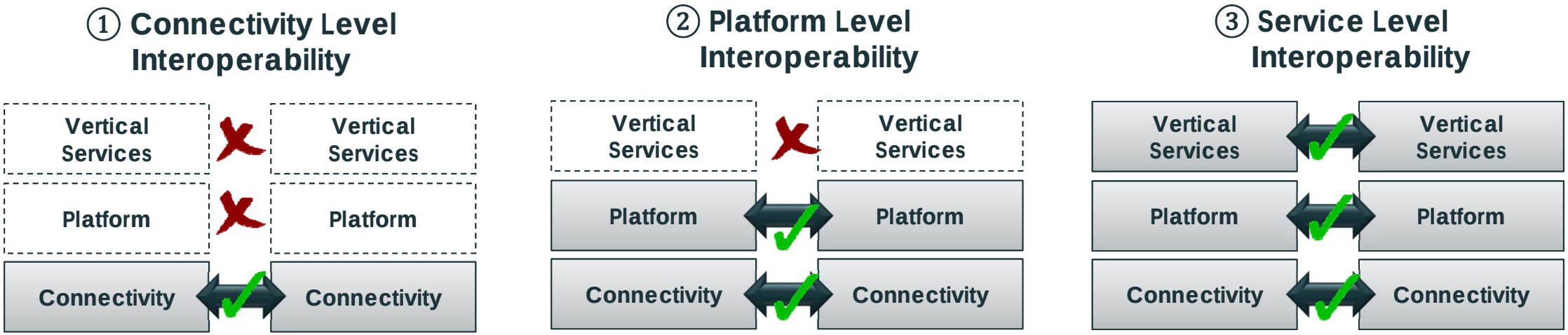
- Legacy vertical services usually designed as silos
  → No common way to communicate among them



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

Interoperability

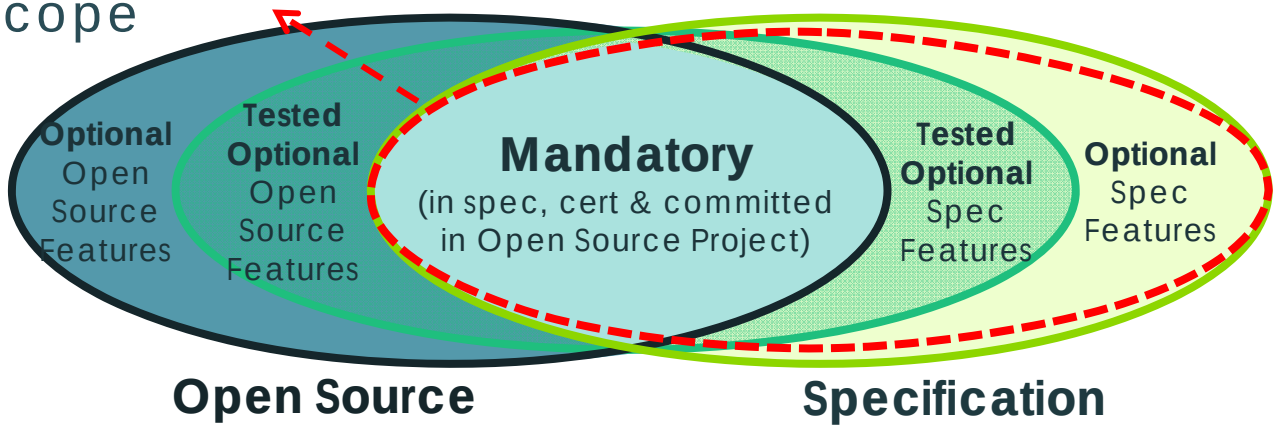
- **Full interoperability** from the connectivity layer up to the service layer is the only way to truly guarantee a satisfactory UX
- Interoperability at the Connectivity and/or Platform layer only provides partial interoperability which can ultimately lead to fragmentation



Interoperability & Certification

- Conformance test - Each device proves conformance to specifications
- Interoperability test - Each device proves interoperability with other devices

- Certification Scope



Licensing

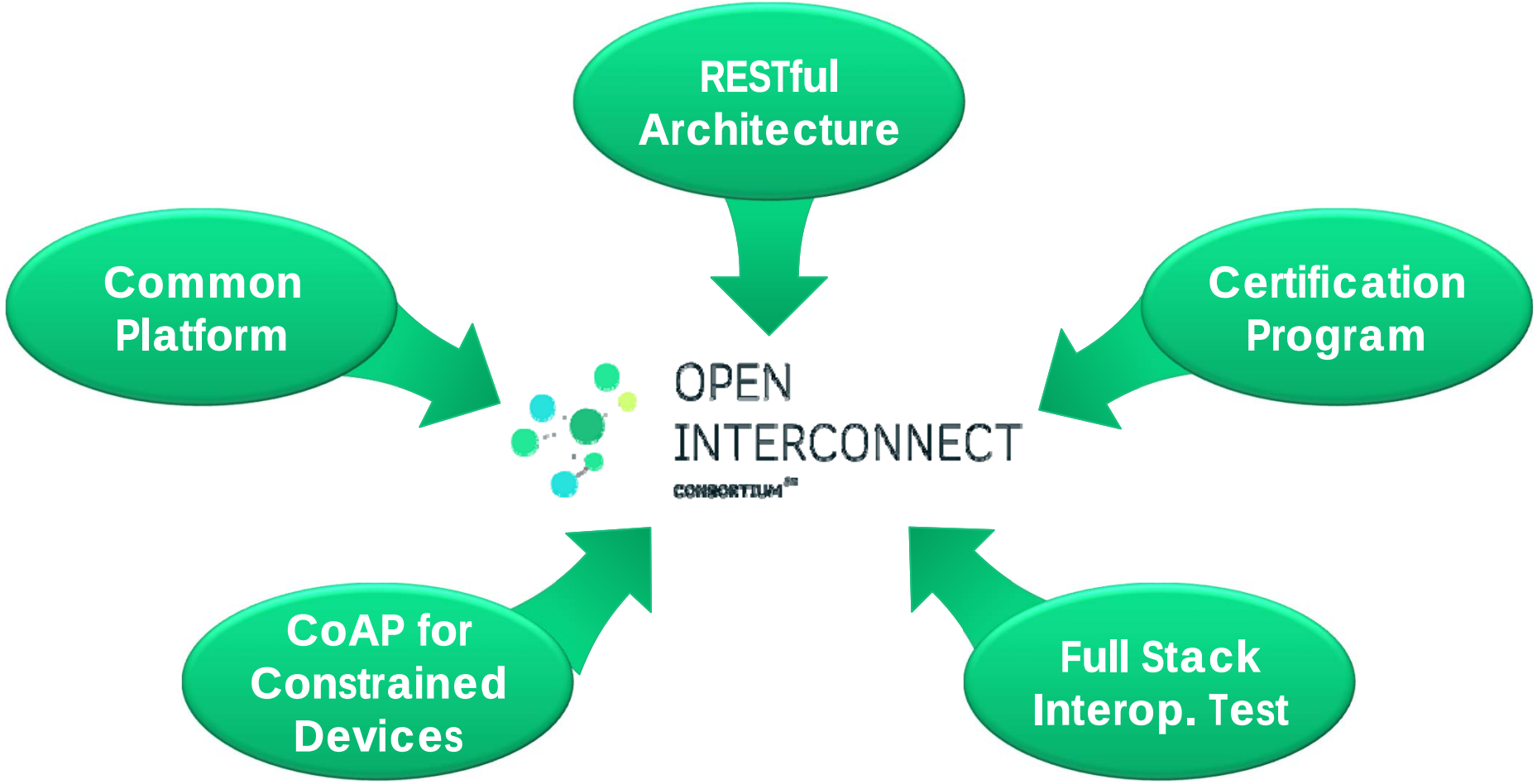
- For IPR Policy : RAND-Z > RAND >> no IPR policy
- For Open Source : Apache 2.0 > ISC

- 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 of Open Interconnect Consortium

Introduction to OIC – Optimized for IoT



OIC Key Concepts (1/2)

- **Free IPR License** (Code: Apache 2.0 & Spec: RAND-Z)
  - License covers both code, standards and related IPR
  - License applies to members and affiliates of members
- **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

OIC Key Concepts (2/2)

- **Full stack definition for maximum interoperability**
  - Connectivity, Platform and Vertical Services defined
  - License applies to members and affiliates of members
- **Certification and Logo program**
  - Guarantees all devices work together
  - Consistent user awareness for interoperability

Sample of Current Members

Diamond

ARRIS, CableLabs, Cisco, GE Software, intel, SAMSUNG

Platinum

Atmel, UOX, CAICT 中国信息通信研究院, DELL, Honeywell, IBM, MEDIATEK, ZTE

Gold

accenture, acer, AEPONA, ALLION, alticast, Astoria networks, asurion, ASUS

avempace wireless, Axstone TECHNOLOGY, B&O, BigJungle, BONTOM, BUFFALO, cadence, ESI

cryptosoft everything secured, ConnectingYourThings, D+M GROUP PERFORMANCE IS EVERYTHING, DigiOn, D-Link, oppo, DSP group

Dt&C, DTT TECHNOLOGY GROUP, ECS, ENCORED, ETRI, FIH 富智康 FIH Mobile Limited, fine point technologies, IB firstbuild

fuse, GCR, GoPro Be a HERO., GRL GRANITE RIVER LABS, GrowSafe SYSTEMS, HANSONG, Hisense, hitron

Sample of Current Members

Gold (continued)

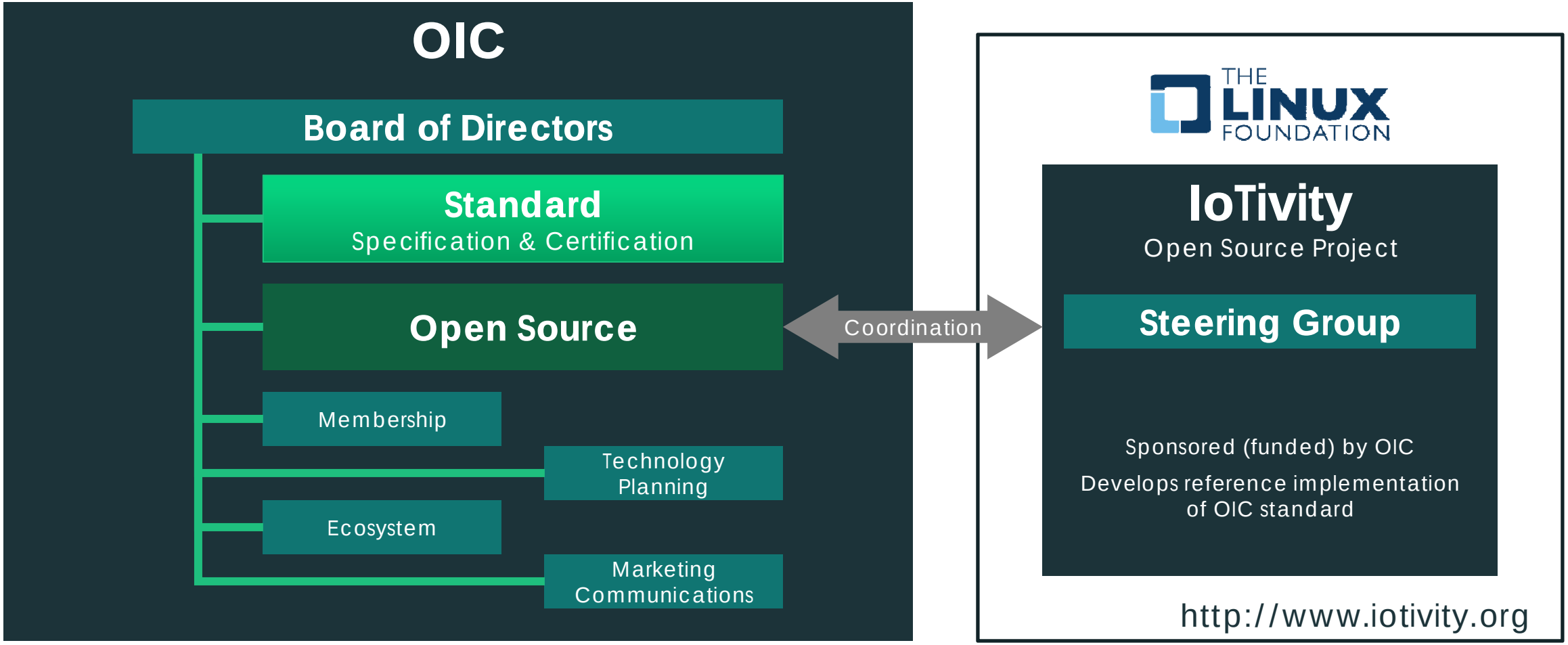
HUAWEI
iBASE
Imagination
InFocus
inside SECURE
thinnect
INTEGRATED SERVICE TECHNOLOGY
JASMIN INFOTECH
JetHead Development, Inc.
L&T Technology Services
FLEXTRONICS LAB IX
Lantiq
lenovo FOR THOSE WHO DO.
LIBRATONE
e-Navigation Solutions
MARVELL
MASHERY
McAfee An Intel Company
MERA Software Services, Inc.
METAGO
MICOSA
MODACOM
Mstar semiconductor
NATIONAL INSTRUMENTS
NewAer
openmind
OSS Nokalva
Parrot AUTOMOTIVE
REALTEK
roost
rti
SAMSUNG ELECTRO-MECHANICS
SAMSUNG SDS
SERCOM
signove
SK telecom
SmartThings
StreamUnlimited
SweetLabs
Swiftlet
SYSTECH
Systematic Consulting Group
同耀实验室 TA Test & certification Laboratories Group
tangor guild
TelHoc
TTA
Ubiquitous
UL
Valens
VIA we connect
VMC
OVU
WIND RIVER
WORKS SYSTEMS
ZyXEL

Sample of Current Members

Non-profit

Liaisons

OIC Structure



OIC Specification Overview

Core Framework Specification

Specification Structure

Infrastructure

- Core Framework
- Security
- Remote Access
- Certification Test Plans and Test Cases

Resource Model

- Resource Specification (Domain agnostic)

Per Vertical Domain

- Device Specification
- Domain Specific Resource Specification

Core Framework Specification

Overview

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
- Architect a core framework that is scalable from resource constrained devices to resource rich devices
- Evaluate technical specification(s) for maximum testability and interoperability
- Ensure alignment with OIC open source releases

OIC Roles

- OIC Client
  - i) Initiate an transaction (send a request) & ii) access an OIC Server to get a service

- OIC Server
  - i) host OIC Resource & ii) send a response & provide service

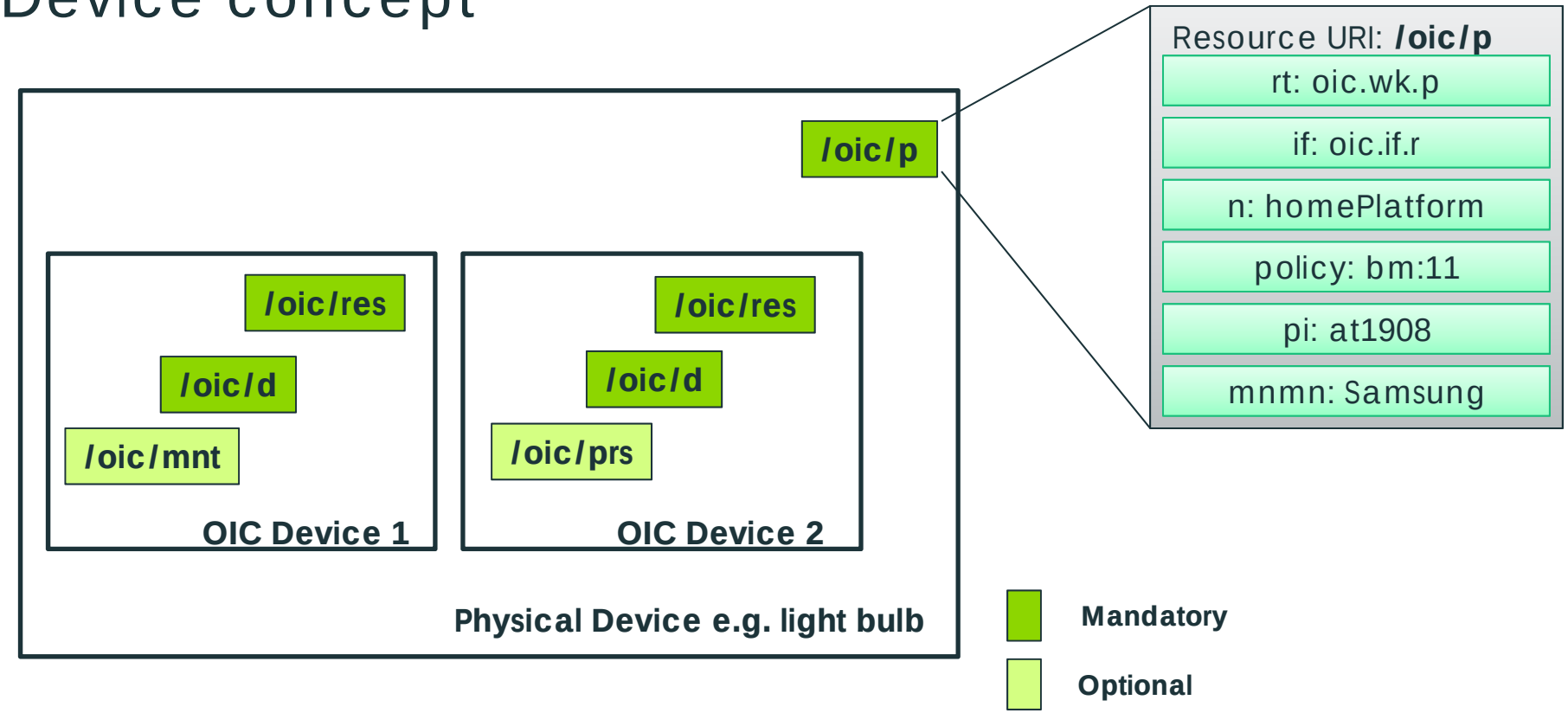
OIC Architecture

- OIC adopted RESTful Architecture
- Current OIC Architecture defines 2 logical roles that devices can take
  - OIC Server : A logical entity that exposes hosted resources
  - OIC Client : A logical entity that accesses resources on an OIC Server

Model 1

Organization of an OIC Device

- OIC Device concept

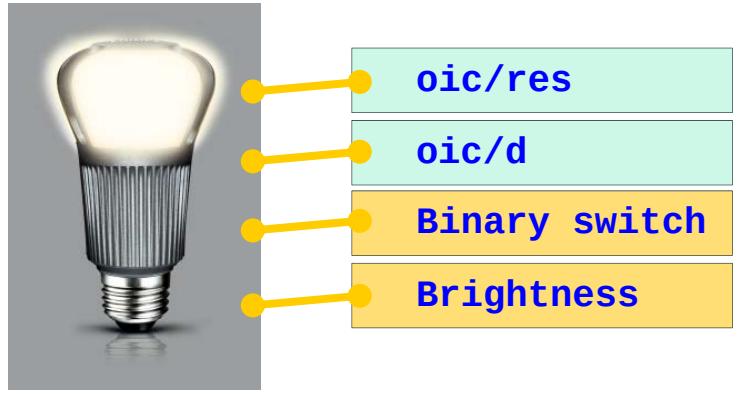


Device example: light device (oic.d.light)

- 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
  - Core resources: ① oic/res, ② oic/d
  - Device specific resources: ③ Binary switch (oic.r.switch.binary),
  - Other optional resources can be exposed, in this example ④ Brightness resource (oic.r.light.brightness)

Example: Smart light device with 4 resources

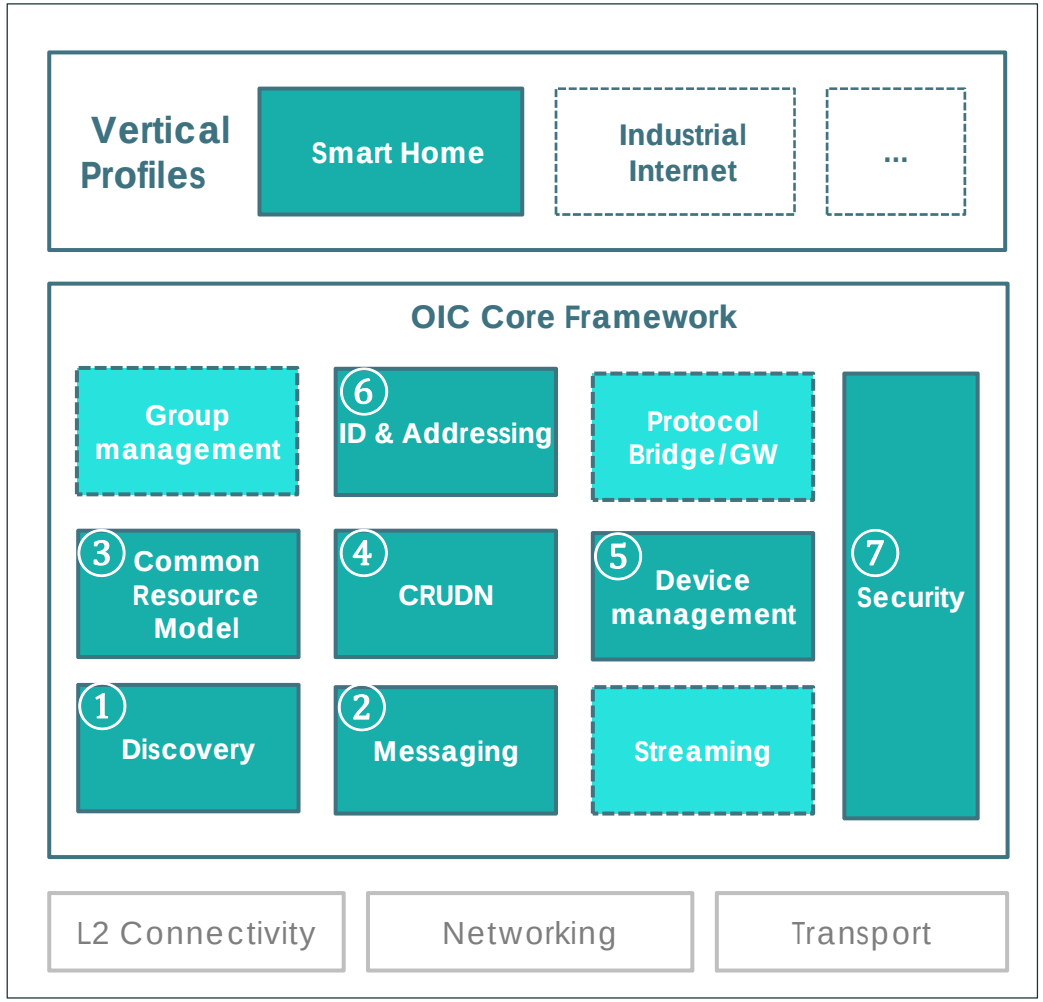
Device Title	Device Type	Associated Resource Type	M/O
Light	oic.d.light	oic/res (oic.wk.core)	M
		oic/d (oic.d.light)	M
		Binary switch (oic.r.switch.binary)	M
		Brightness (oic.r.light.brightness)	O



Core Framework Specification

Key Features

OIC 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 intermediaries
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. **Device Management:** Network connection settings and remote monitoring/reset/reboot functions
6. **ID & Addressing:** OIC IDs and addressing for OIC entities (Devices, Clients, Servers, Resources)
7. **Security:** Basic security for network, access control based on resources, key management etc

OIC Core Framework Basic Operation

Discovery → Operation → Goal

Discovery

- Discover access policies, device info and resources on the devices

---

Operation

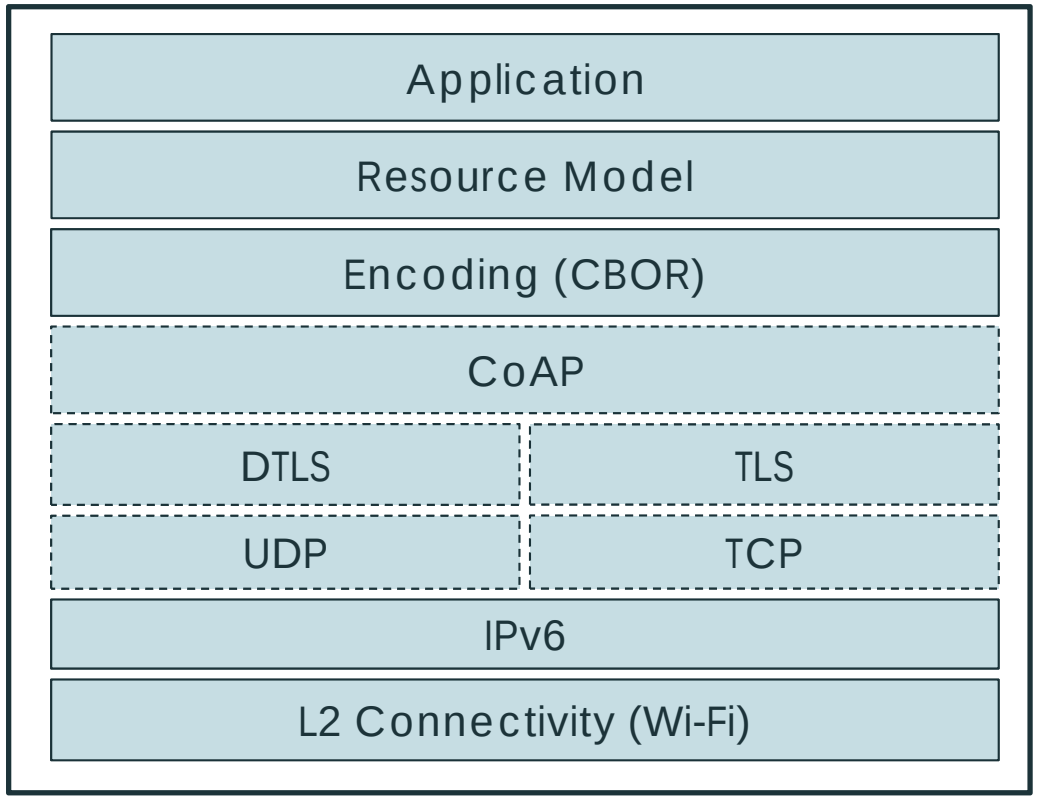
- Get device information by retrieving resources
- Control devices by changing resources
- Observe change on the properties of resources

Basic common services

- Device Monitoring
- Maintenance (e.g., reboot, factory reset, statistics collection, etc.)

Connectivity | Networking | Transport | Security

Protocol Stack



Project B OIC Stack

Alternatives

Encoding	JSON or XML/EXI can be negotiated
IP Version	v6 (v4 supported for legacy devices)

End point Discovery (CoAP Discovery)

- OIC devices make use of CoAP Discovery (defined by IETF RFC 7252)
  - Resource Discovery (Possible to discovery resource being hosted by device directly)
  - Low processing overhead on each node
  - High traffic efficiency (in terms of amount of data sent/received for discovery)

Encoding Schemes – JSON, XML/EXI, CBOR

- OIC resource is represented as sequence of bits by encoding schemes when to transfer it over the network
- OIC supports several encoding schemes and it will be negotiated and accepted by OIC Server when OIC Client requests
- OIC has mandated CBOR as the default encoding scheme

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

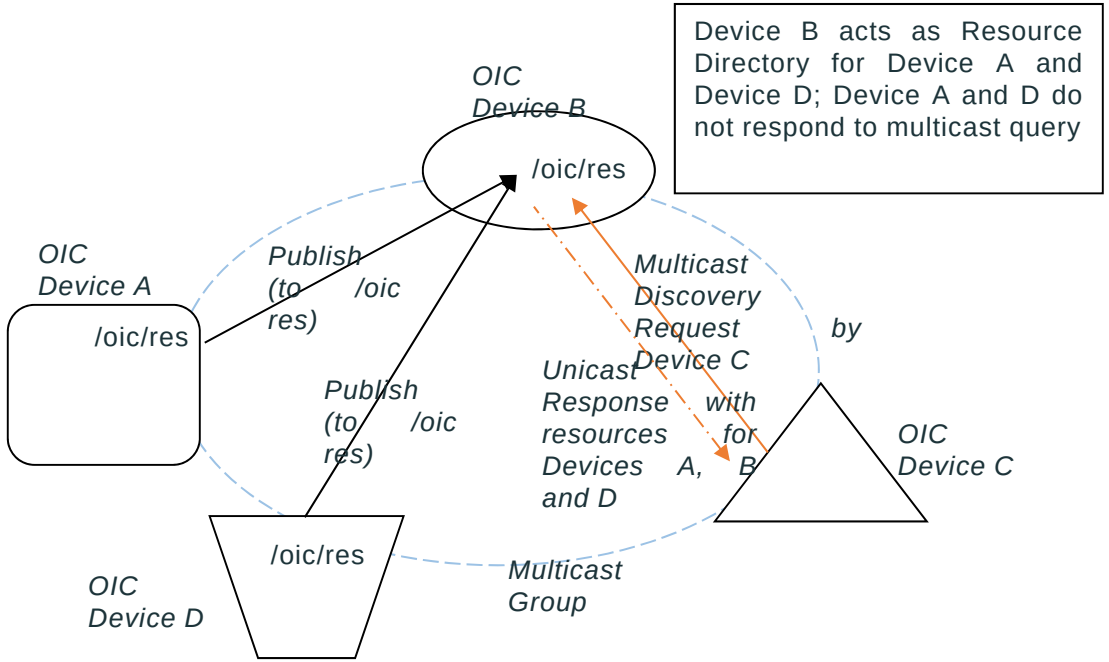
* JSON: JavaScript Object Notation, EXI: Efficient XML Interchange,
CBOR: Concise Binary Object Representation

Collection Resources

- A container is used to model complex structures
- An OIC Resource that contains one or more references (specified as OIC Links) to other OIC Resources is an OIC Collection
- An OIC Link embraces and extends typed “web links” as specified in RFC 5988

Resource Directory

- Offloads handling of discovery (response to multicast messages) to devices that are capable of doing so
- Key enabler for sleepy end nodes, enhances battery life.



Scenes/Rules/Scripts (1 of 3)

- Overview
  - Mechanisms for automating certain operations
  - Rules, Scripts and Scenes can be grouped and reused
- Scenes
  - A static entity that stores a set of defined resource property values for a collection of resources.
  - Provide a mechanism to store a setting over multiple OIC Resources that may be hosted by multiple separate OIC Servers.
  - Once set up, can be used by multiple OIC Clients to recall a setup

Scenes/Rules/Scripts (2 of 3)

- Rules
  - A logical “if then” statement
  - Consists of a rule condition and a Rule Member (a script)
  - The rule condition is an evaluation criterion which can include evaluation of the value of a sensor on an OIC Server
  - When the evaluation criterion is evaluated true then the Rule Members are set to a specific determined value
  - A rule condition is evaluated when one of the observed resources in the rule condition changes

Scenes/Rules/Scripts (3 of 3)

- Scripts
  - A programmatic element that can be used to incorporate conditionals, delays, loops and other programmatic devices, including reading and writing scenes
  - Scripts can consist of a set of steps that are executed either upon meeting the conditions of a rule or as part of another script, in order to automate tasks
  - Scripts can also be used to set a scene to a specific value
  - A Script is realized as the set of Rule Members that are executed when a rule condition holds true
- Summary
  - Scenes are bundled user settings
  - Scripts are automated background tasks
  - Rules are conditional statements that execute scripts when the condition is true

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 draft-ietf-core-block-17* shall be used by all OIC Servers that receive a retrieve request for a content payload that would exceed the size of a CoAP datagram

Messaging Protocol Negotiation

- Supported messaging protocols are conveyed in the property (mpro) on the /oic/res (resource discovery)
- Omitting this property defaults to the messaging protocol as specified in the vertical specification (e.g., CoAP for Smart Home)
- After discovery, an OIC Client can use any of the supported messaging protocols supported by the OIC Server

CoAP Serialization over TCP

- Provides the ability for CoAP to run over TCP in environments where TCP is already available and where UDP may be blocked.
- If TCP is used then reliability is provided by TCP rather than the inherent reliability mechanisms within CoAP (confirmable messages).
- Use the new protocol negotiation feature to convey support during resource discovery (/oic/res)

OIC Specification Overview

Smart Home Device and Resource Specification

Smart Home Device and Resource Specification

Way of Working

Defining OIC Components (on top of CORE)

OIC Servers

- Defined by *device identifier*: standardized name of the device
- List of mandatory OIC resources per device
- Note that OIC Clients are implicitly specified as “opposite” side of an OIC Server.
  - Currently OIC does not impose interaction sequences.
  - All Resources are allowed to talk to/from any OIC Client at any point in time

OIC Resource

- Defined by *resource identifier*: standardized name of the resource
- List of mandatory properties per resource
- List of allowed actions (read/readwrite/..) per resource

Vendor extensions

Vendor is allowed to:

- Create own defined (none OIC standardized) resources
- Create own defined (none OIC standardized) device types
- Extend existing devices with additional (not mandated) resources
  - With standardized resource types
  - With vendor defined resource types

Tooling

- SHTG defines all resource schemas using JSON, all resource APIs using RAML
- SHTG developed Python based tool chain that auto-generates specification text based on the RAML and JSON that is defined per resource.
- Capabilities provided by the tooling include:
  - Auto validation of the RAML against RAML syntax rules
  - Auto validation of the JSON schemas against JSON Draft-04 rules
  - Auto validation of all example JSON against the applicable JSON schemas

High confidence level in the validity of the resource definitions

Ability to simulate all resources

Specifications

- Specifications are split in 2 documents:
  - Device specification
  - Resource specification

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

OIC SmartHome Device
Vendor Smart Home Extensions
Vendor Core Resources Extensions
Smart Home Device specification
Smart Home Resources
Core Resources
Smart Home Core Profiles

Smart Home Device and Resource Specification

Key Features

Resource Specification

- List of reusable resources that are used in an Smart Home Device
  - Contains generic list of error codes
  - Uses core definitions

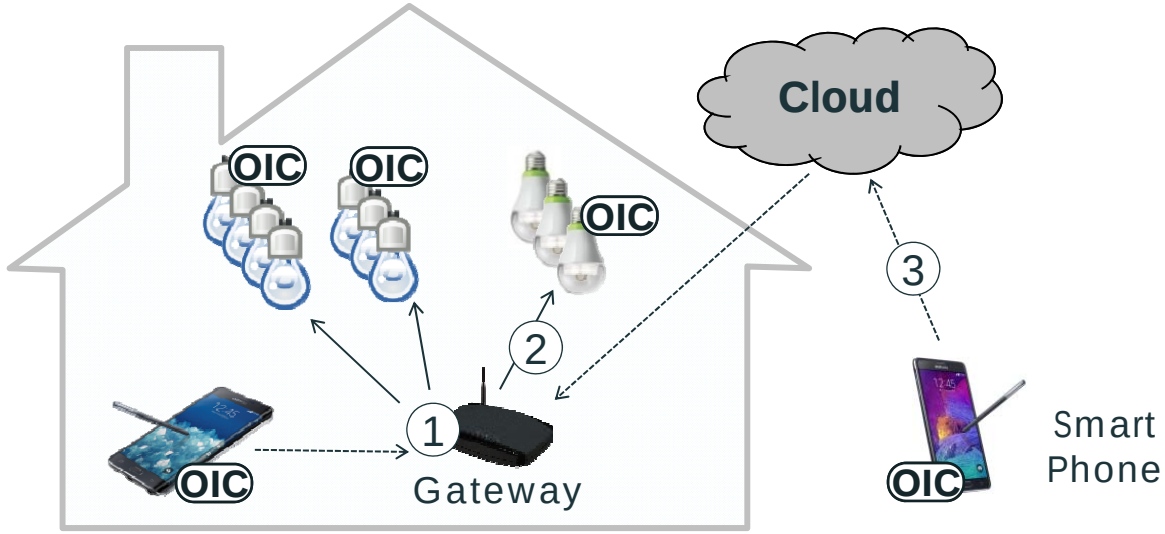
- Each Smart home resource definition contains:
  - unique identifier (rt)
  - Indication if the resource is an sensor or actuator
  - List supported methods
  - List per method the JSON schema for input and output

- ***Resources are specified in* RESTful API Modelling Language (*RAML*)**

Smart Home Use Cases

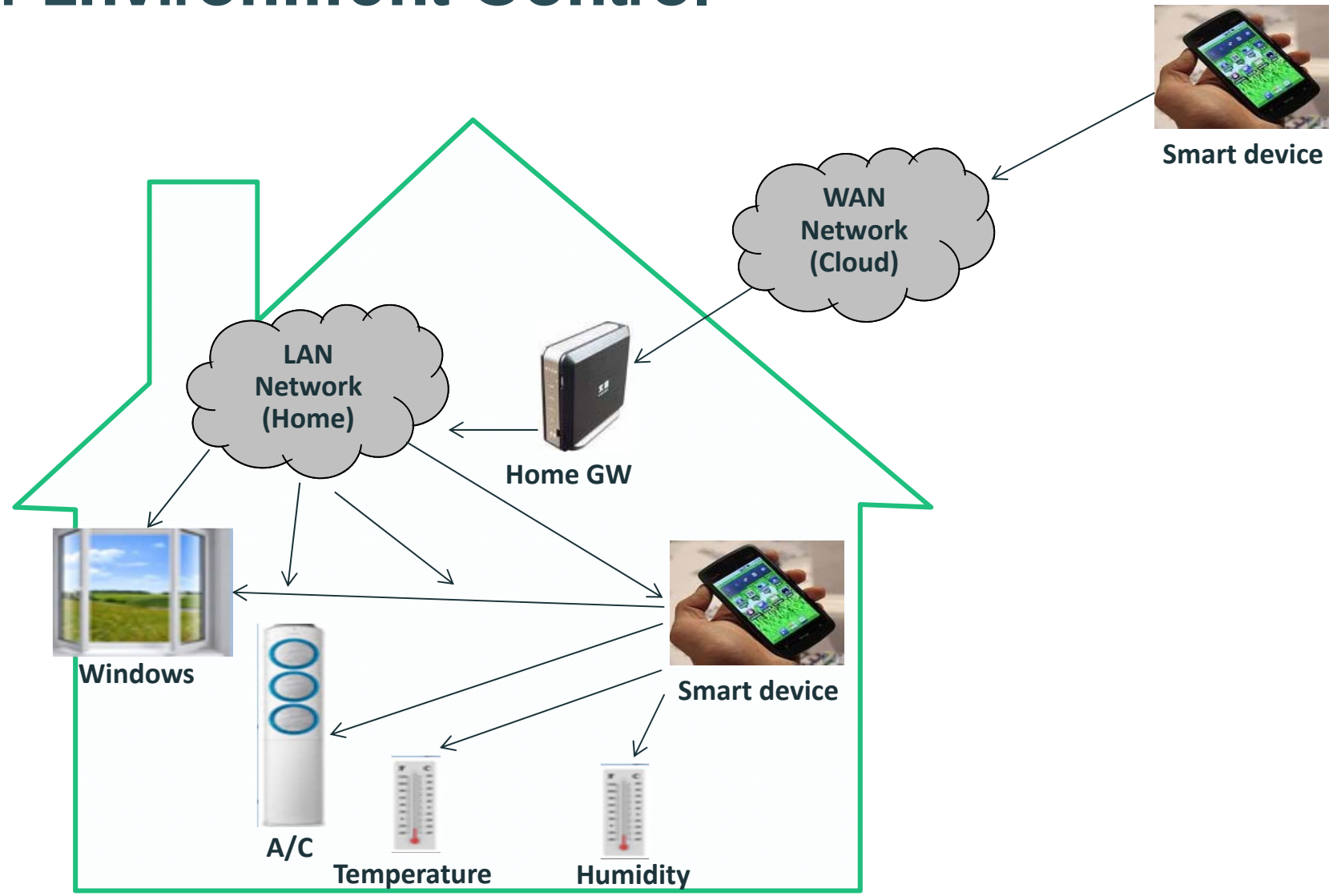
- Selected key enabling use cases to scope activity

Use Case	Priority
Indoor Environment Control	1
Lighting control	
Energy Saving Washer/Dryer	
Energy Management	
Remote Access for Device Control	
Smart watch notify and control	6
Smart Video Environment	3
Smart Home Office	
Smart Garage	
Device Grouping and Control	
Multi player gaming	7
Smart watch gaming on TV	
Fire safety monitor and Notify	4
Keyless Entry	2
Home Security	
Health Monitor and Notify	5

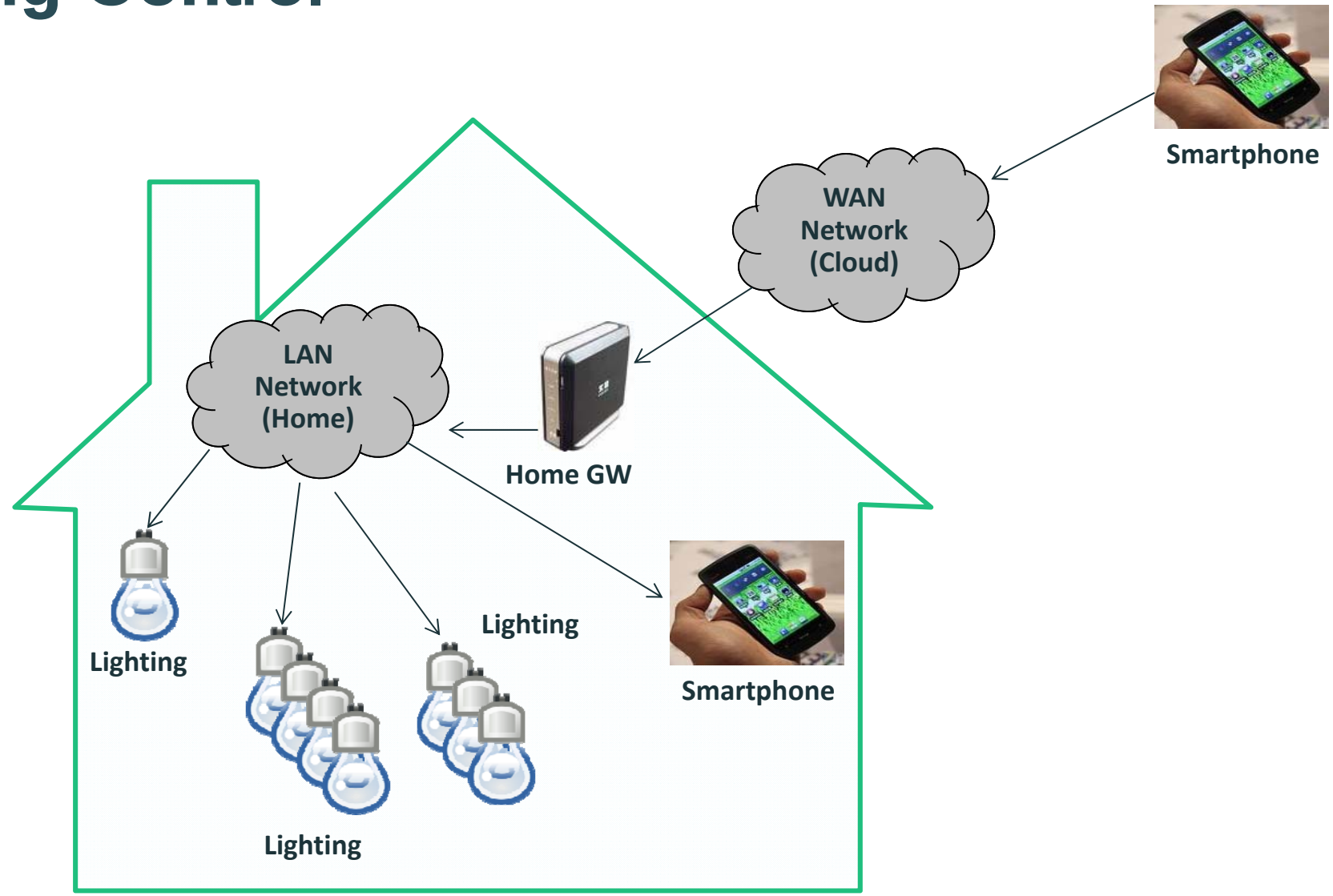


① Control proximal OIC Devices
② On board new Devices
③ Control remotely with an OIC Client

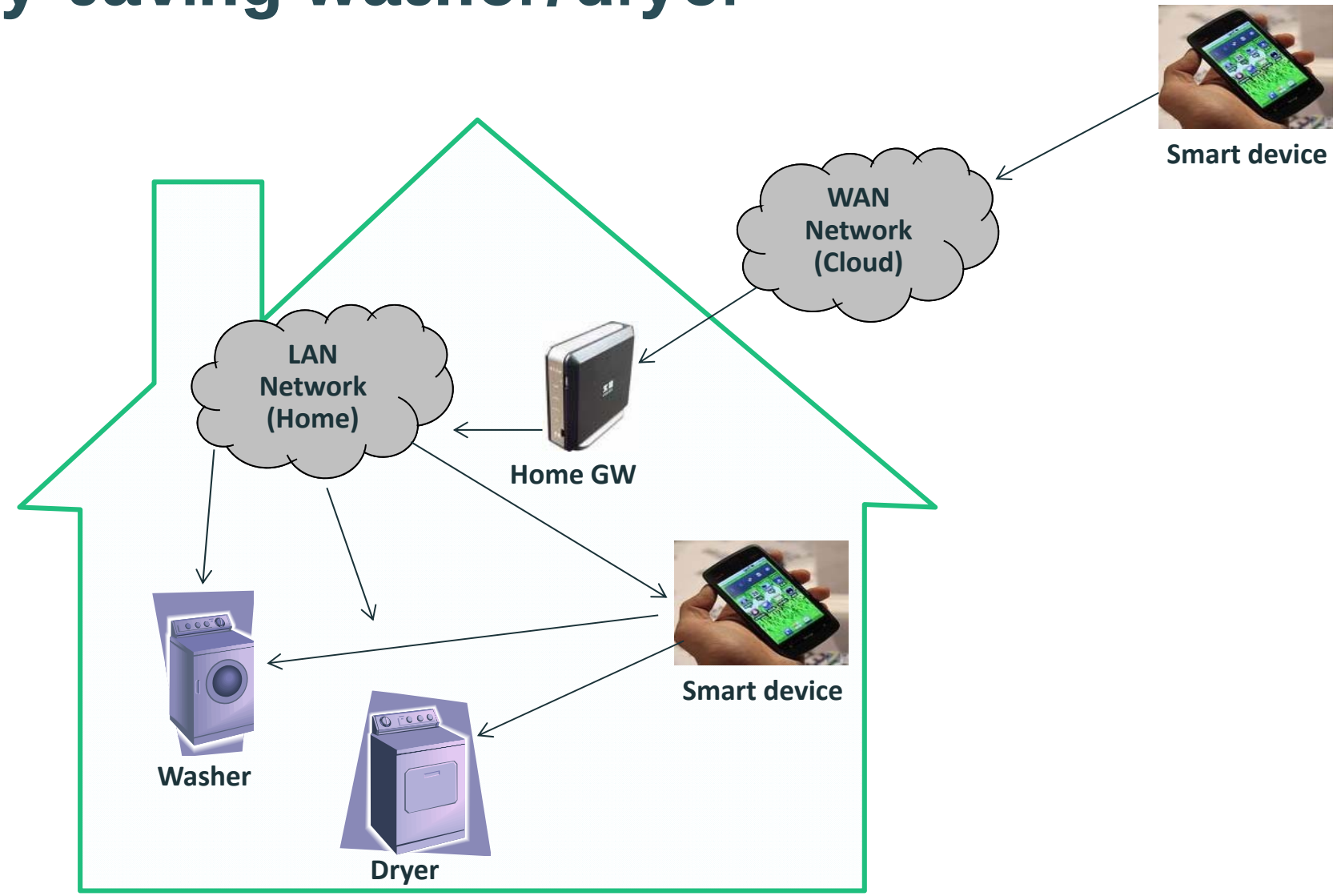
Indoor Environment Control



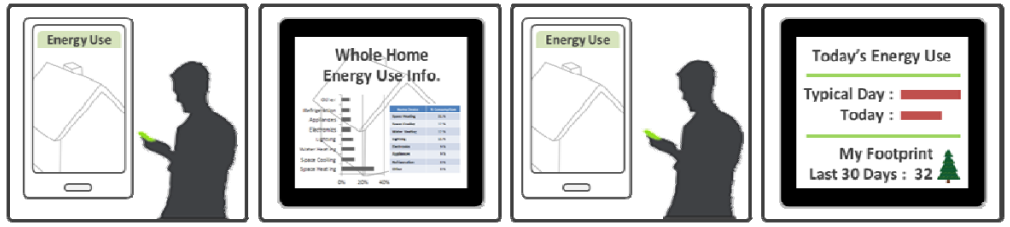
Lighting Control



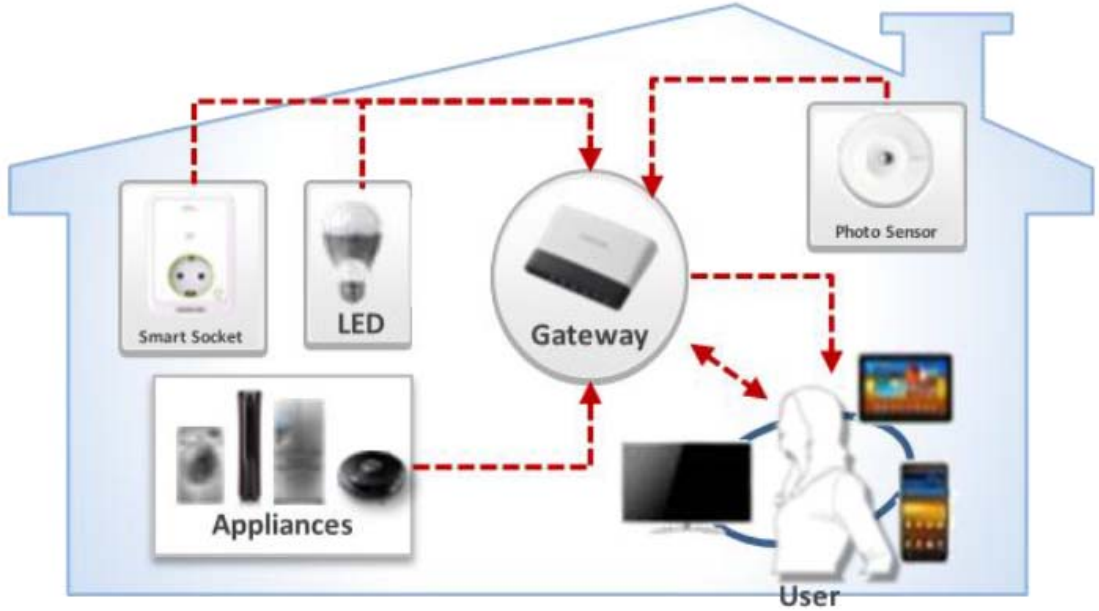
Energy-saving washer/dryer



Energy Management



Need to see my home energy use information | Whole-home energy use information | How much energy I consumed today? | Today, you consumed 20% less than usual

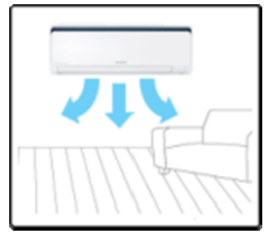


Remote Access Device Control

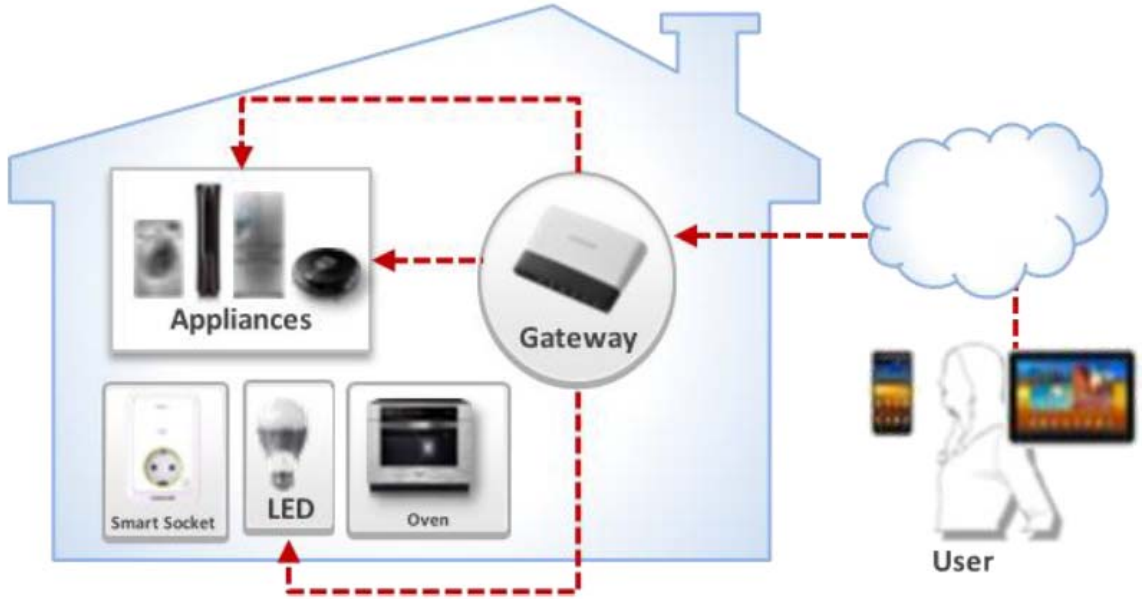
"Can I drop in your home, tonight?"

"Absolutely.."

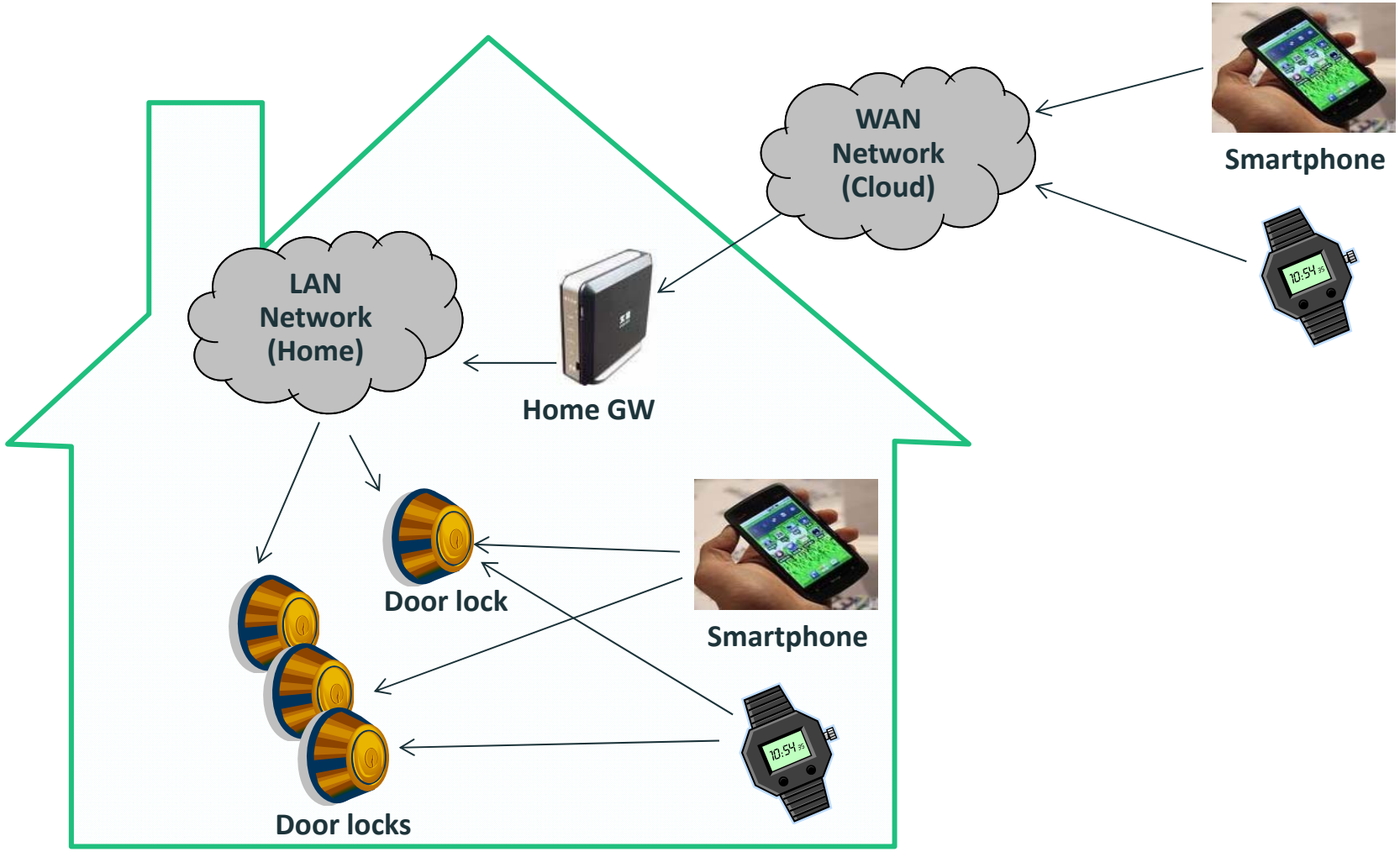
'I believe you, robot'



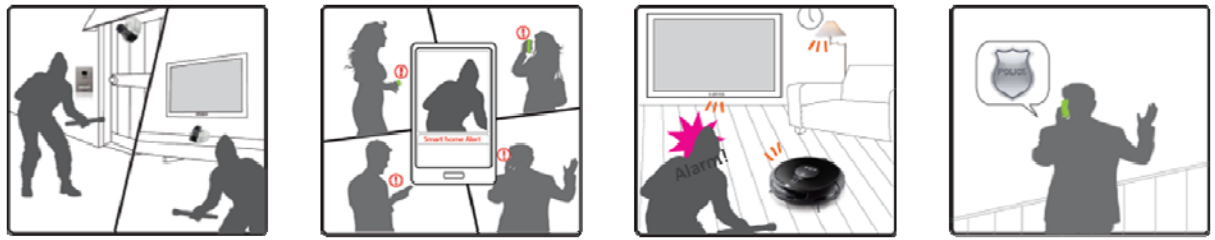
[ready for girls' visit : fresh and clean]

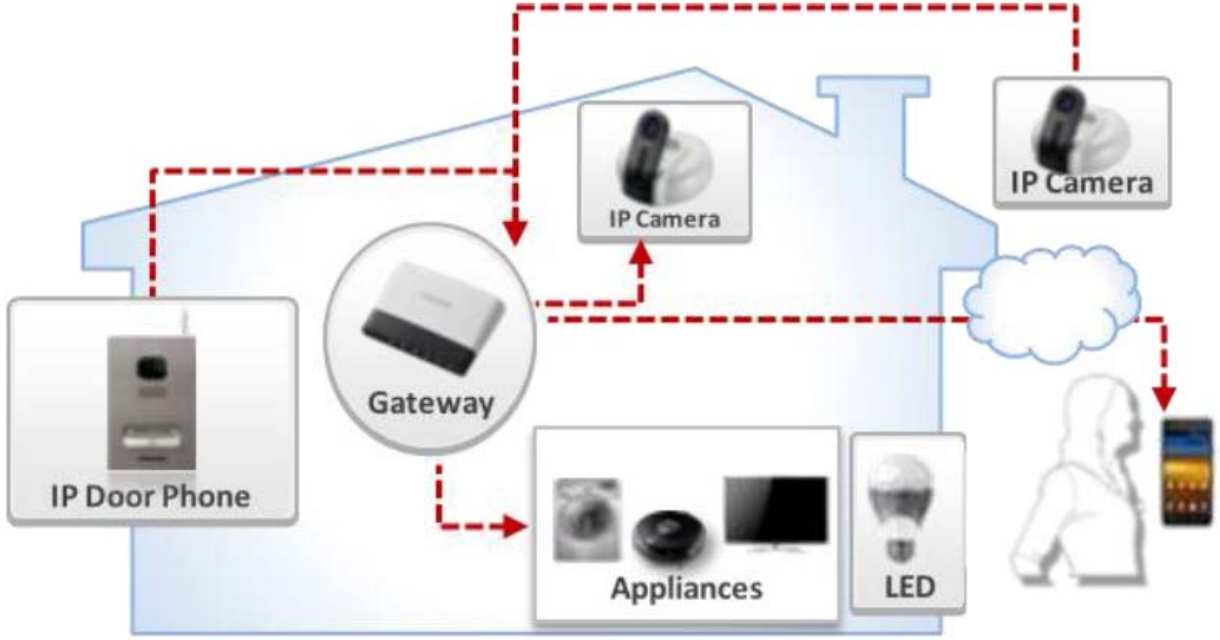


Keyless Entry

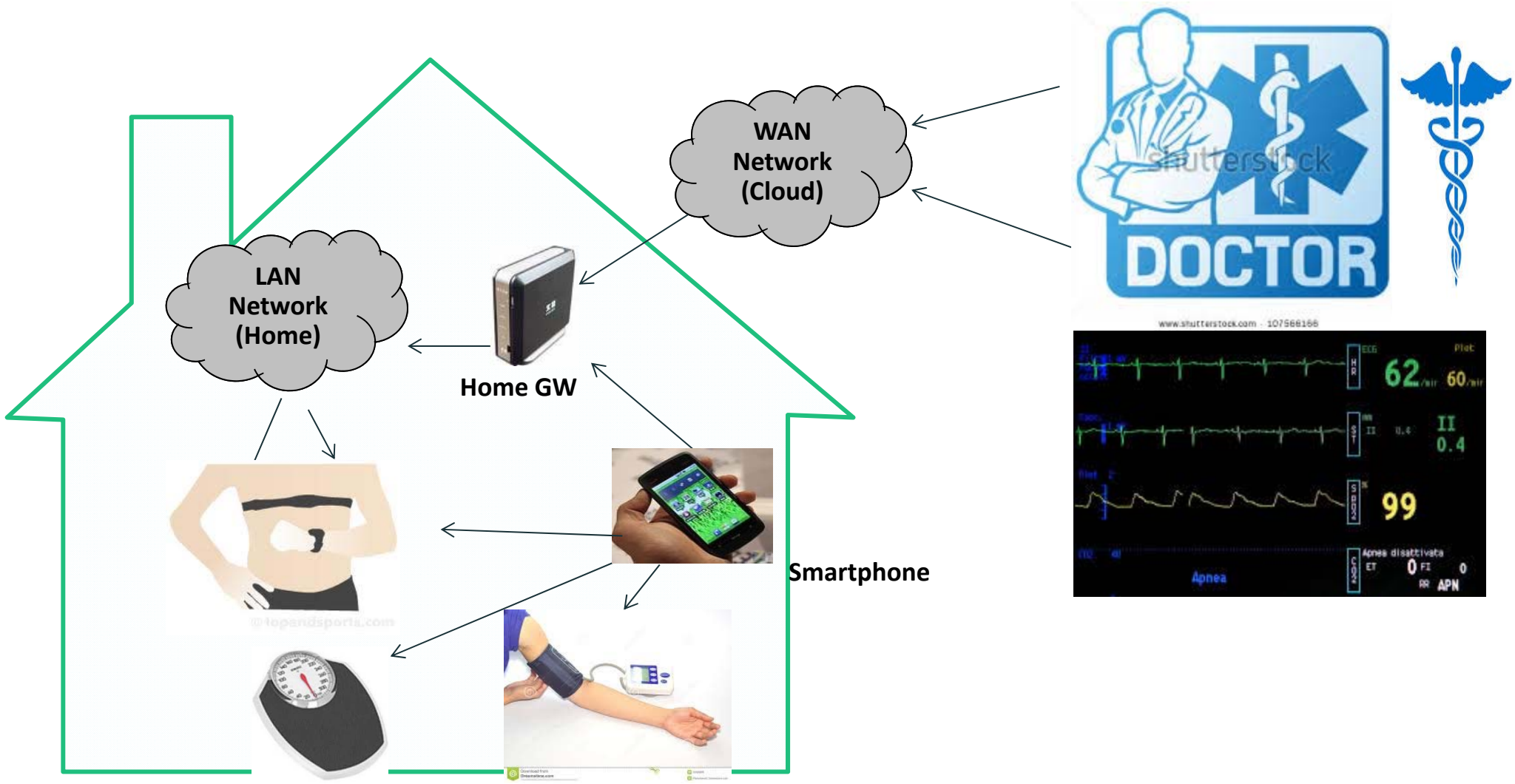


Home Security





Health Monitor & Notify



Smart Home Device Type

Device Type	Minimum Resource Set
Air Conditioner	Binary Switch, Temperature
Air Purifier	Binary Switch
Blind	Open Level
Dishwasher	Binary Switch, Mode
Door	Open Level
Clothes Dryer	Binary Switch, Mode
Clothes Washer	Binary Switch, Mode
Fan	Binary Switch
Garage Door	Door
Light	Binary Switch
Oven	Binary Switch, Temperature (2)
Printer	Binary Switch, Operational State

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

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

Defined Resource Types (1/2)

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

Defined Resource Types (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 Support Resources

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 OIC Server hosts an OIC known resource then it shall follow all normative requirements in the Resource Specification applicable to that Resource.

OIC Bridge - Background & technical need

- There are many different IoT standards out there
- There are many different vendor solutions out there

Hence it would be good for OIC if OIC could use these devices and create a (vendor defined) bridge to these non-OIC devices.

Goal:

- To represent non OIC devices by means of a bridge as an OIC server on the network.

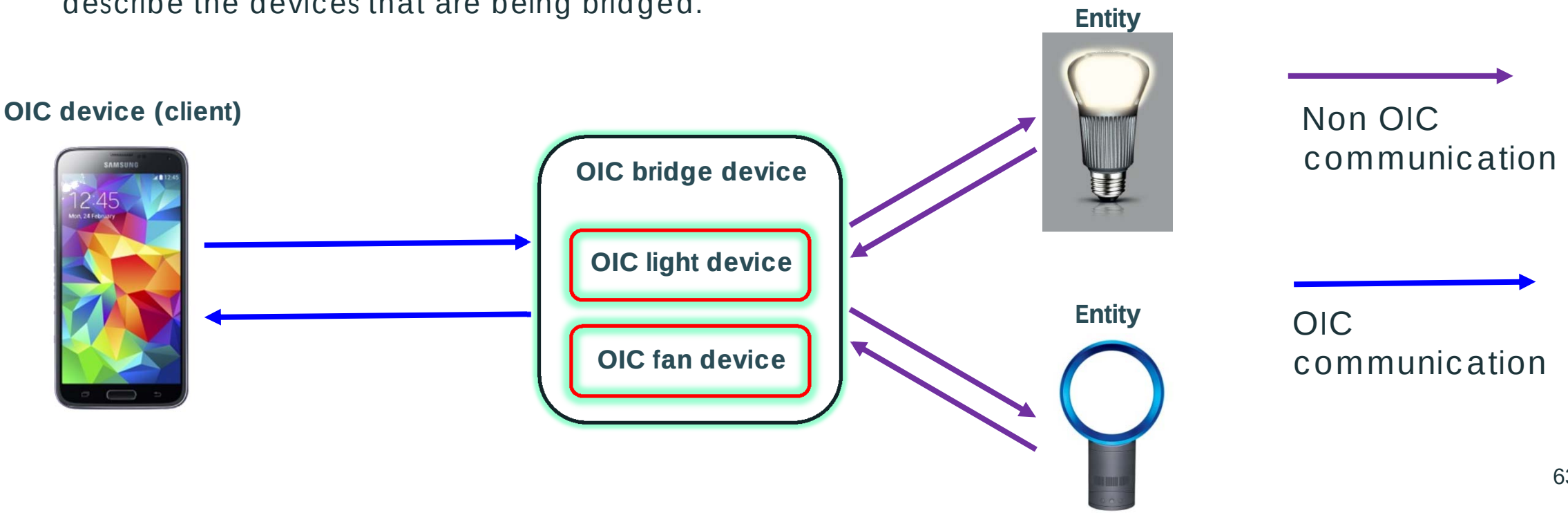
Conceptual:

- Bridge establishes an OIC standardized ***north bridge*** so that all OIC clients can use the bridged devices.
- The ***south bridge*** will be vendor/implementation specific: it uses the protocol defined by the bridged device.

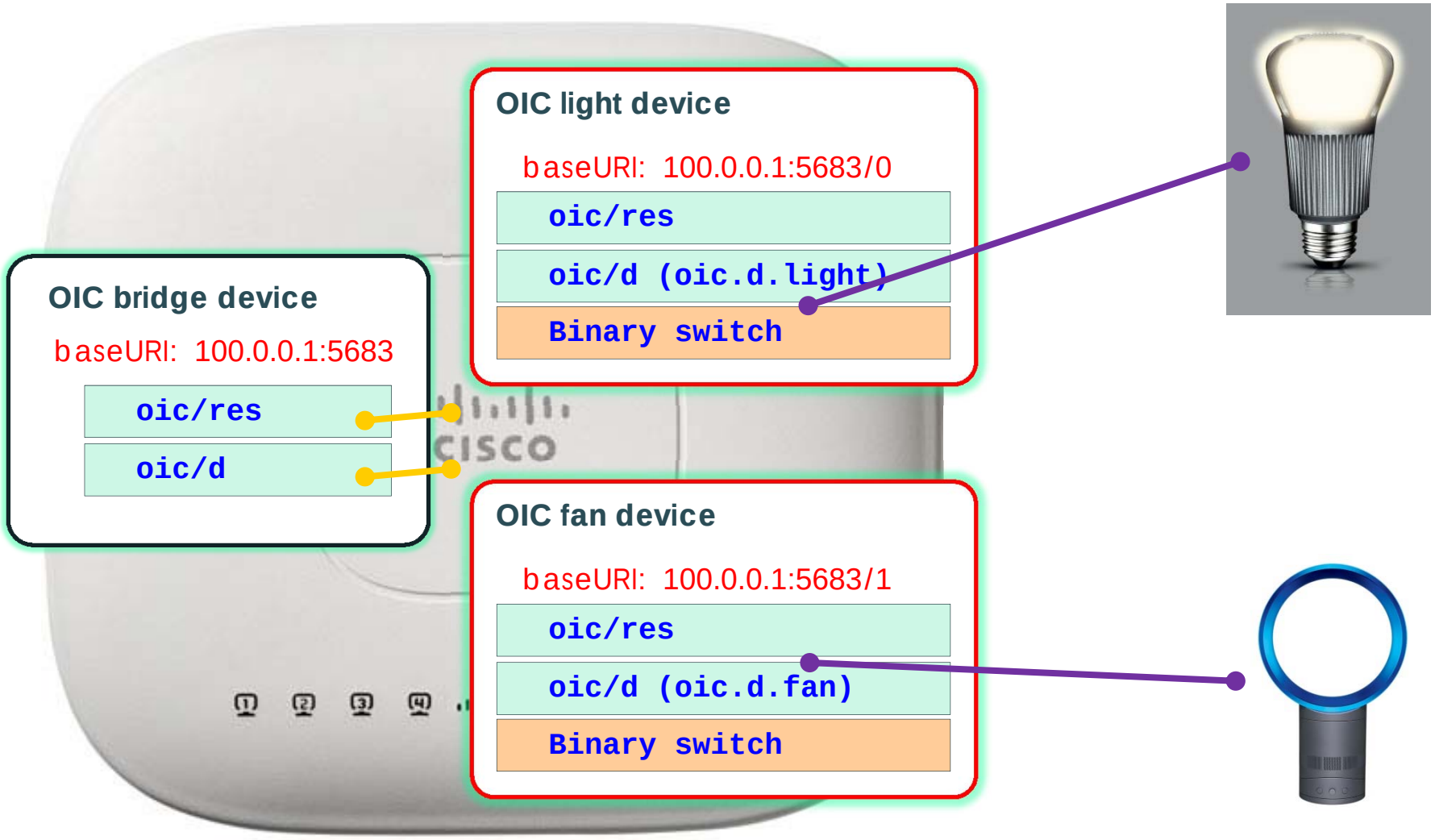
  (for example: it needs to realize Philips Hue APIs if a Hue light is bridged)

OIC Bridge - Definition

- An OIC smart home bridging device is a device that represents one or more other non-OIC devices as OIC Smart Home Devices on the network.
- The represented devices themselves are out of the scope of OIC. The bridging (that is, how the bridge communicates with the non-OIC devices) is implementation and vendor specific.
- The only difference between a 'regular' OIC Device and a bridged device is that the latter is encapsulated in an OIC Smart Home Bridge Device.
- An OIC Smart Home Bridge Device shall be indicated on the network with an "rt" of "oic.d.bridge". When such a device is discovered the exposed resources on the OIC Smart Home Bridge Device describe the devices that are being bridged.



Bridge Device example: bridge (oic.d.bridge)



Bridging relationship with oic/res

/oic/res

```
[
 {"di": "bridge_device_id",
  "links": [
    { "href": "/oic/d",
      "rt": "oic.d.bridge",
      "if": "oic.if.r",
      "rel": "hosts"}]},

 {"di": "light_device_id",
  "links": [
    { "href": "0/oic/d",
      "rt": "oic.d.light",
      "if": "oic.if.r",
      "rel": “contains external"},
    { "href": "1/myLightSwitch",
      "rt": "oic.r.switch.binary",
      "if": "oic.if.a",
      "rel": “contains external"}]},

 {"di": "fan_device_id",
  "links": [
    { "href": "1/oic/d",
      "rt": "oic.d.fan",
      "if": "oic.if.r",
      "rel": “contains external"},
    { "href": "1/myFanSwitch",
      "rt": "oic.r.switch.binary",
      "if": "oic.if.a",
      "rel": “contains external"}]}
]
```

/oic/d

```
{
  "n": "myRoomBridgeDevice",
  "rt": “oic.d.bridge",
  "if": "oic.if.r",
  “di": “bridge_device_id“,
  "icv": "oic.1.5“,
}
```

/oic/d

```
{
  "n": "myRoomLightDevice",
  "rt": “oic.d.light",
  "if": "oic.if.r",
  “di": “light_device_id“,
  "icv": "oic.1.5"
}
```

/oic/d

```
{
  "n": "myRoomFanDevice",
  "rt": “oic.d.light",
  "if": "oic.if.r",
  “di": “fan_device_id“,
  "icv": "oic.1.5"
}
```

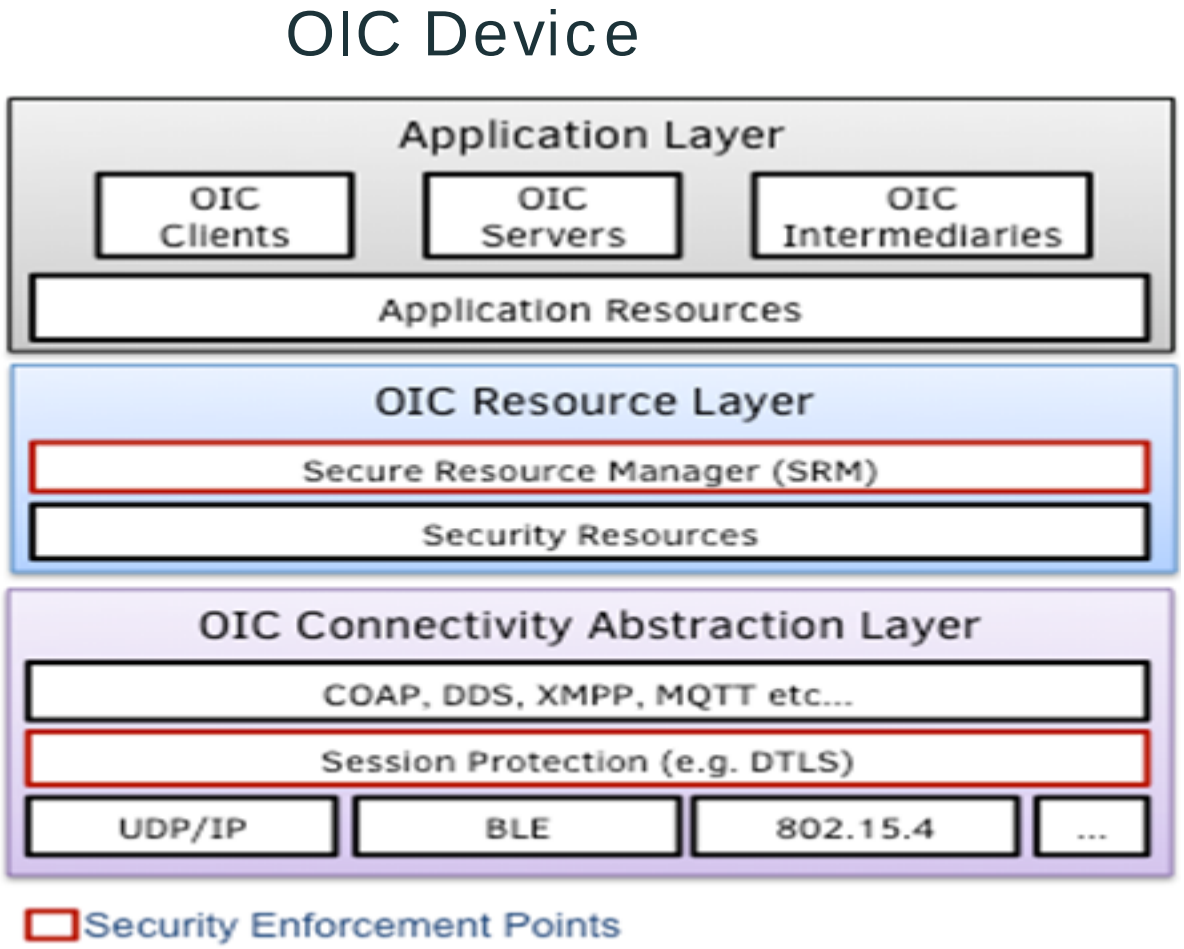
Security Specification

Open Interconnect Consortium, Inc.

OIC Security Summary

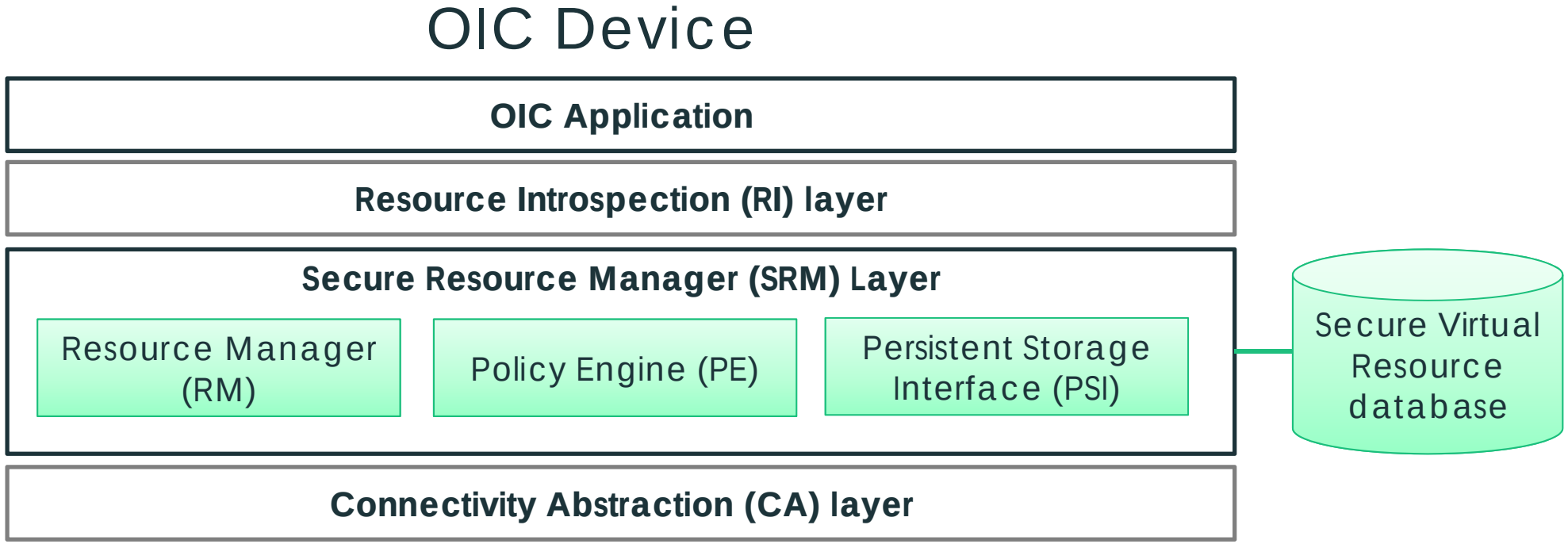
- OIC key management supports end-to-end device protection
- Resource layer ACLs allow intended interactions while preventing unintended interactions
- Secure device ownership helps prevent attacks when devices are added to the network

To Cross a Boundary We Must Define the Endpoint



- An OIC *device* is the endpoint
- ...more specifically it is the OIC resource layer
- OIC resources define how device capabilities are exposed to other OIC devices
- Resources are accessed securely through a secure channel such as DTLS
  - End-to-end message encryption, integrity and replay protection
- OIC does not define endpoint hardening techniques
  - Resource layer hardening is implied

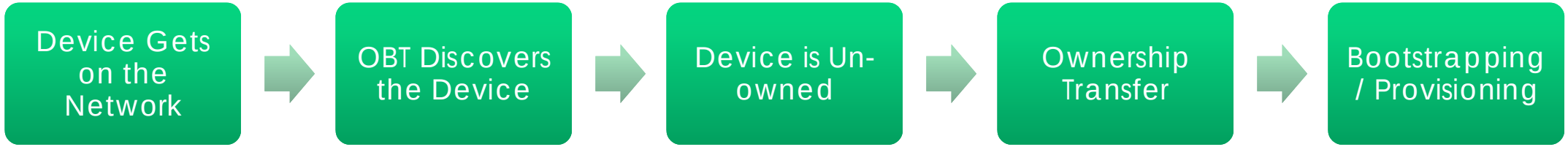
Secure Resource Manager (SRM)



- SRM Duties
  - Manage secure endpoint resources (Creds, ACLs, Device ID, Config status)
  - Enforce resource access and endpoint protection
  - Device ownership
  - Security provisioning
  - SVRD storage protection

Ownership Transfer and Bootstrapping

- Devices typically ship from a manufacturer in an “un-owned” state
- The user does some magic to affect taking ownership of the device, using an Onboarding Tool (OBT)
  - Take over responsibility of the device and relieve manufacturer of any liability due any actions the device may take under user’s ownership
- Ownership transfer creates a relationship between an OIC device and an OBT.
  - The relationship is defined through establishment of an Ownership Credential and a set of ownership-complete states



Ownership Transfer and Bootstrapping

- Security Spec Defines Several Ownership Transfer Methods (OTM):
  - Just-Works, DECAP, Random-PIN, Manufacturer Certificates
  - Also allows Vendor Specific Method
- All OTMs are optional for an OIC device to implement, but it is mandatory to support at least one among Just-Works, DECAP, Random-PIN or Manufacturer Certificates.
  - (We will need to be able to test all for certification ultimately)
  - Might change in the future spec
- OTMs differ in:
  - How a device establishes trust
  - How the physical owner's "intent" is proved
  - What cipher suites are used
- OTMs should bring the device to a well defined state

Secured vs. Un-secured

- OIC Servers support a secured and un-secured interface.
- Generally speaking, the un-secured interface is for discovery only. All other services should be visible on the secured interface only.
  - The un-secured interface has no message protection and no access control enforcement
  - Publicly visible unique IDs (device, platform, etc.) may present a privacy problem
- Discoverable resources are resources that can be delivered as part of a discovery request (secured interface or not)
  - At the time of creating, a resource is defined as “discoverable” or not.

Message Integrity and Confidentiality

- DTLS only for now.
- The devices communicating need to have useable credentials to talk to each other. If they are missing, the devices could contact the CMS to get them.
- All secured communication is encrypted and signed.

Access Control

- Resources on the secured interface (that should be almost everything) are **only** accessible if there is a proper entry in the Access Control List
  - No ACL, No Service
- An ACL says “X can do Y on resource Z”
  - X can be a deviceID, a role, or a group (in the future)
  - Y can be any combination of CRUDN
- If no ACL is present, and the device has an AMS configured, it can ask the AMS what authorization X has on Z.

Access Control : example

/oic/sec/acl

Subject: *device/group or role*

Resource(s): one or more URN

Permission: bitmap of CRUDN

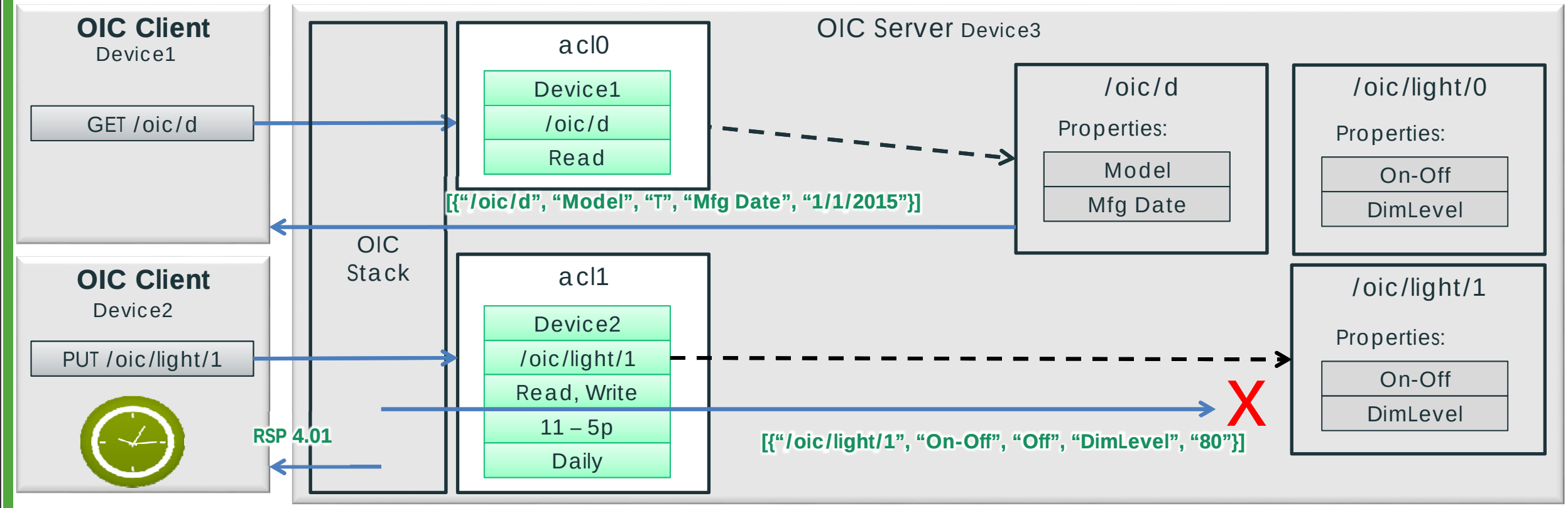
Period(s): validity periods

Recurrence(s): recurrence rule(s)

Rowner: the service that owns this acl

```
{
   "Subject": ”switch1",
   "Resource": "/light",
   "Permission": "00000100", <update>
   "Period": " ",
   "Recurrence": " ",
   "Rowner": "oic.sec.ams"
}
```

Resource Access Example



- Access is blocked if no ACL match is found
- Device1 request to get /oic/d is **accepted** due to ACL Read permission
- Device2 request to update /oic/light/1 is **denied** due to time-of-day policy
- An intermediary (Device4) may also enforce ACLs

Credential Management

- OIC devices can support the use of both symmetrical and asymmetrical credentials for establishing secure communication
  - Symmetric Key is mandatory
  - Local PKI mechanism is supported (Keys are issued in home domain and used only within that domain.)
- Missing credentials could be procured from a CMS
- Credentials may have an expiration period
  - Expired credentials can be refreshed

Credential Management : example

/oic/sec/cred

CredID: Local short ID

SubjectID: device or group

RoleID(s): roles this credential allows a subject to assert

CredType: sym/asym/cert/…

PublicData, PrivateData, OptionalData

Period: Expiration period of credential

Credential Refresh Method:

Rowner: service that can modify this resource

```
{
   ”CredID": ”1”,
   "SubjectID": ”device1”,
   ”RoleID”: ” ”,
   ”CredType": "1”, <symmetric pair-wise>
   ”PublicData”: “”,
   “PrivateData”: “ABCDEFGHIJKLMNP”,
   "Period": ”P1W ",
   "Recurrence": " ",
   "Rowner": "oic.sec.ams"
}
```

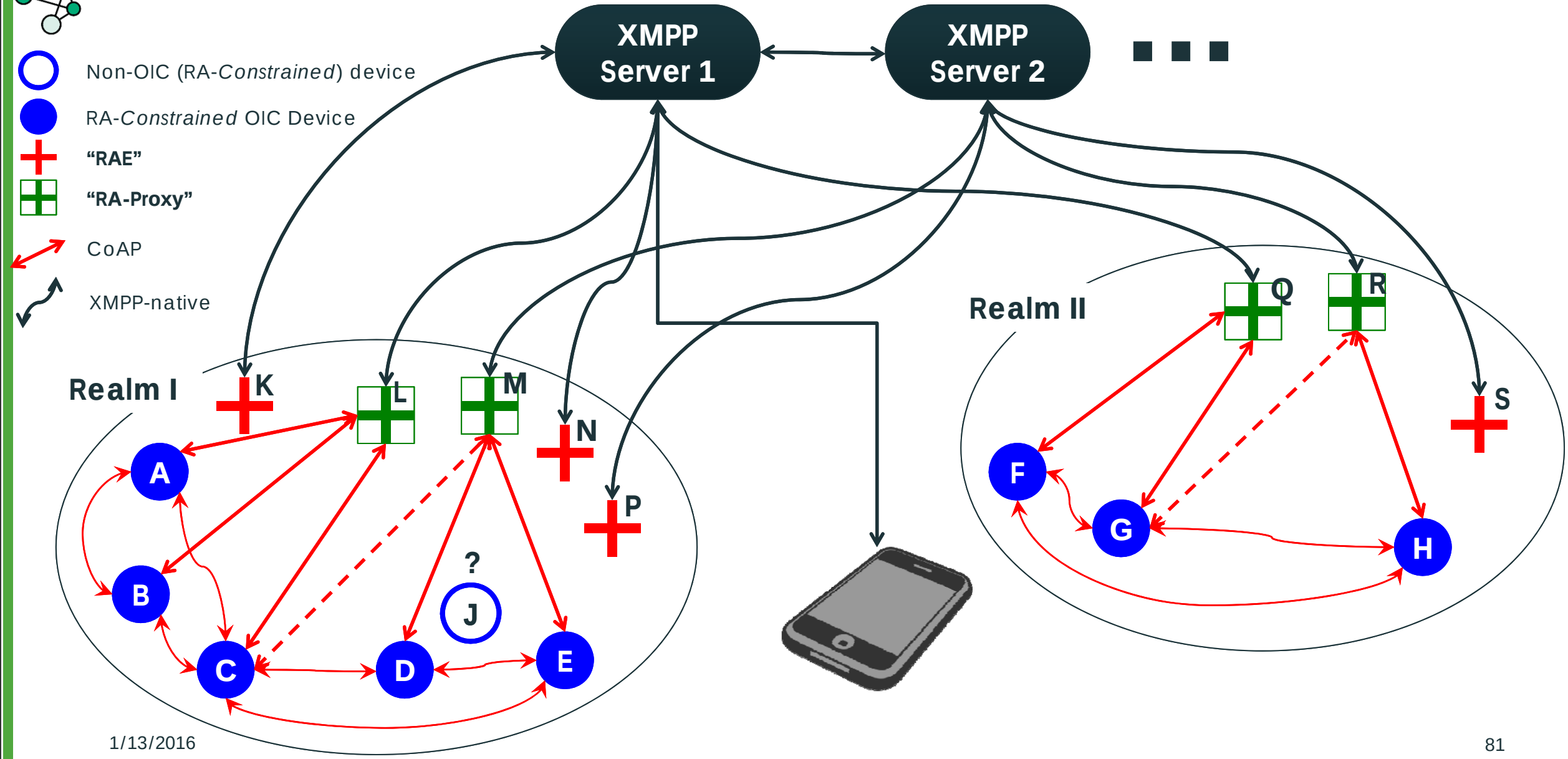
OIC Specification Overview

Remote Access

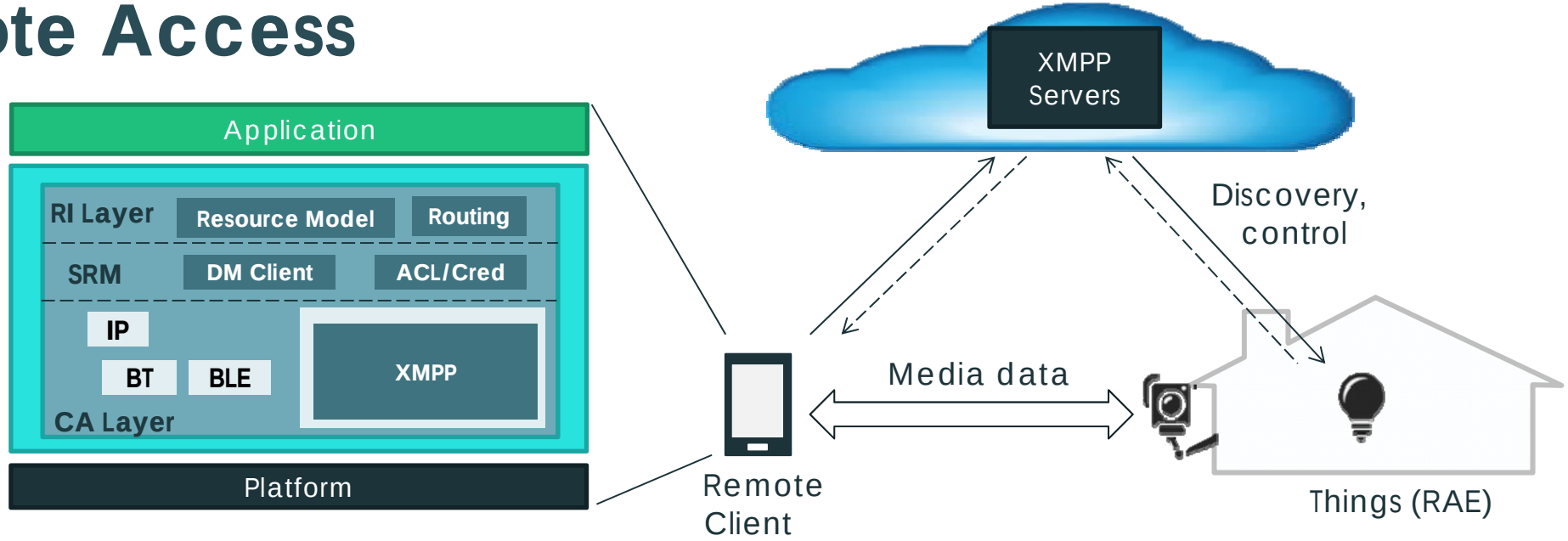
Remote Access (“RA”) in OIC (implementation plan)

- Remote Access endpoint Devices:
  - Remote Access Endpoints (“RAE”):
    - OIC Servers *also* capable of XMPP, optionally capable of ICE-client
  - Remote Access Proxies (“RA-Proxy”):
    - Superset of RAE – Capable of ‘representing’ “RA-constrained devices”
      - “RA-Constrained”: Devices incapable of *natively* supporting RA tech
- Cloud Components:
  - XMPP Server(s)

The OIC RA Model



Remote Access



- Server Components:
  - Device Management Server: Device/Capability Registration and Authorization
  - Signaling Server: Delivering candidate address to recipient, discovery, presence, low BW data, SDP control
- Client Components: RA Endpoint (RAE) & RA-Proxy
  - XMPP Client

RA as defined in Spec 1.0

- Format for bare-JIDs (owner) and full-JIDs for RAEs
  - Includes JID-Resource overloading for:
    - OIC Spec version
    - Device-type
    - UUID
- Mapping from Core/Smart-Home Resources to full-JID format
  - Allows for Presence, Remote Discovery, XMPP-Roster-based access
- Communication of CRUDN messages between the OIC clients and OIC servers that are in the same roster

RA-Roadmap – Post Spec 1.0 priorities

- Defining RA-Proxy functionality
  - Leverage XMPP PubSub (XEP-0060)
  - Extending full-JID overloading model & XMPP Presence
    - Adding RA-Proxy Device-type – avoid gratuitous remote device queries
- "App notes" for temporary remote access via XMPP Multi-User Chat (MUC – XEP-0045),
  - Family members, neighbors, etc.
- Adding Jingle (XEP-0166) for media signaling

Thank you!

- Access the OIC specifications http://openinterconnect.org/developer-resources/specs/
- Contact OIC at admin@openinterconnect.org