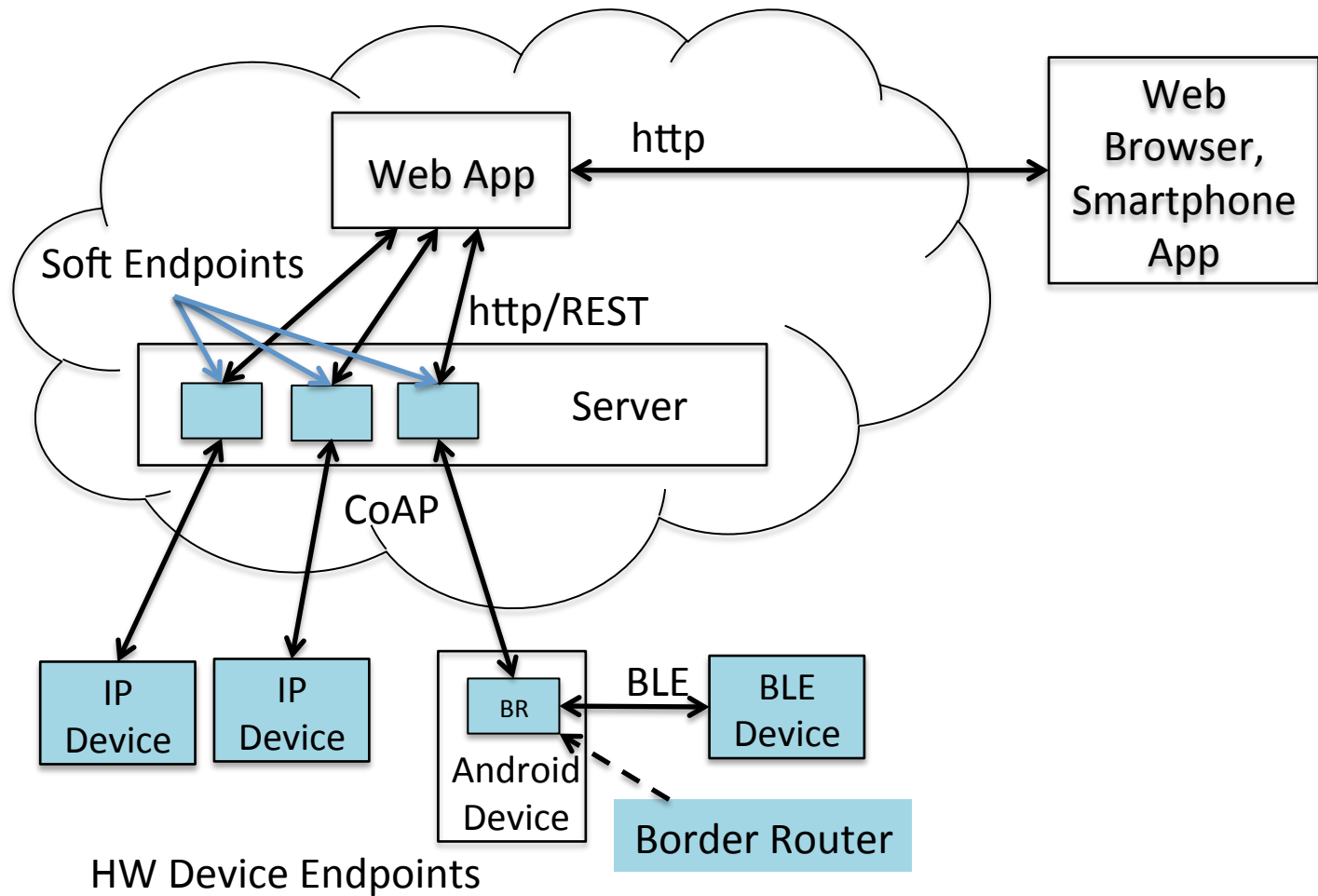
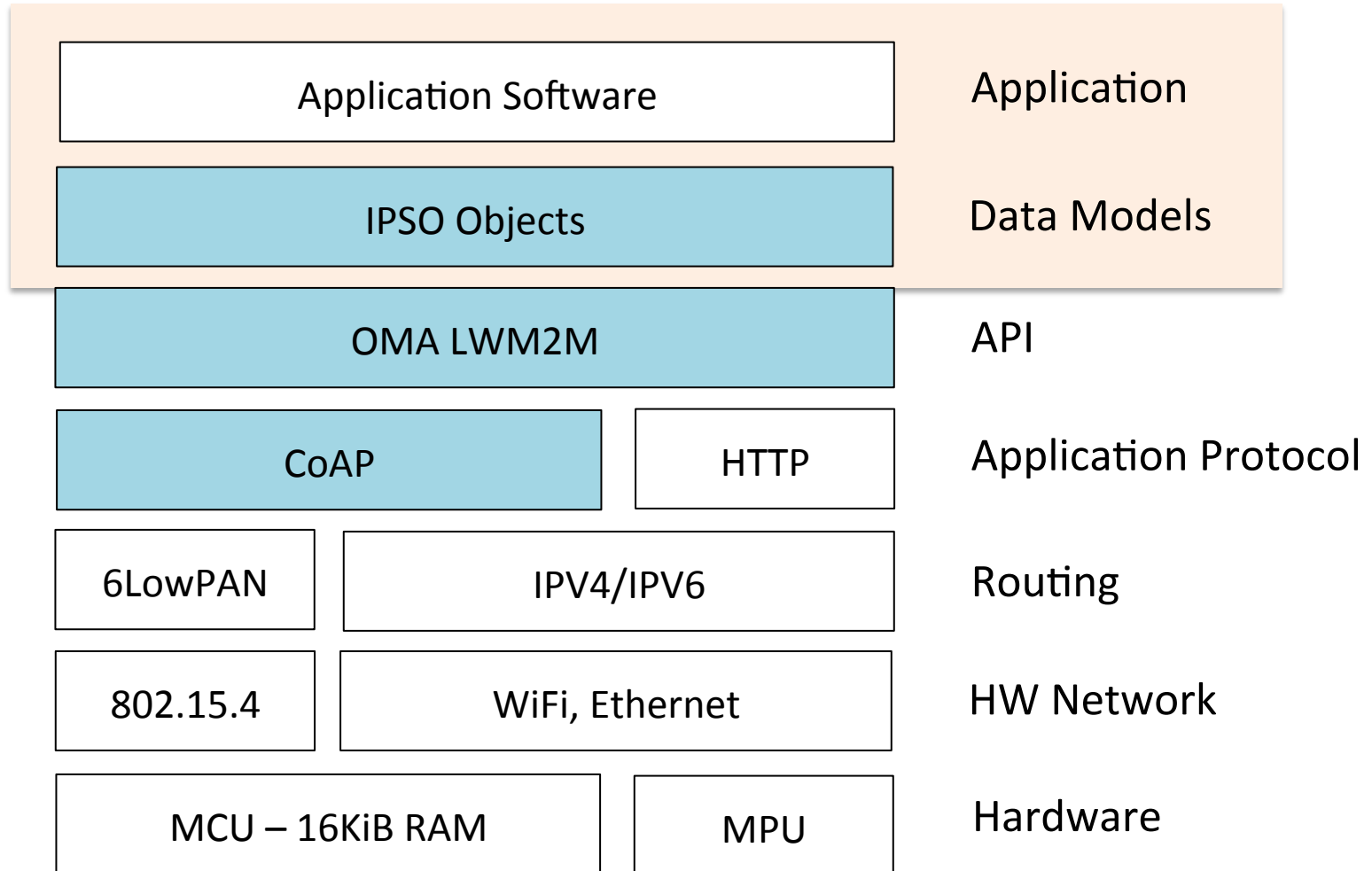


Reference Architecture

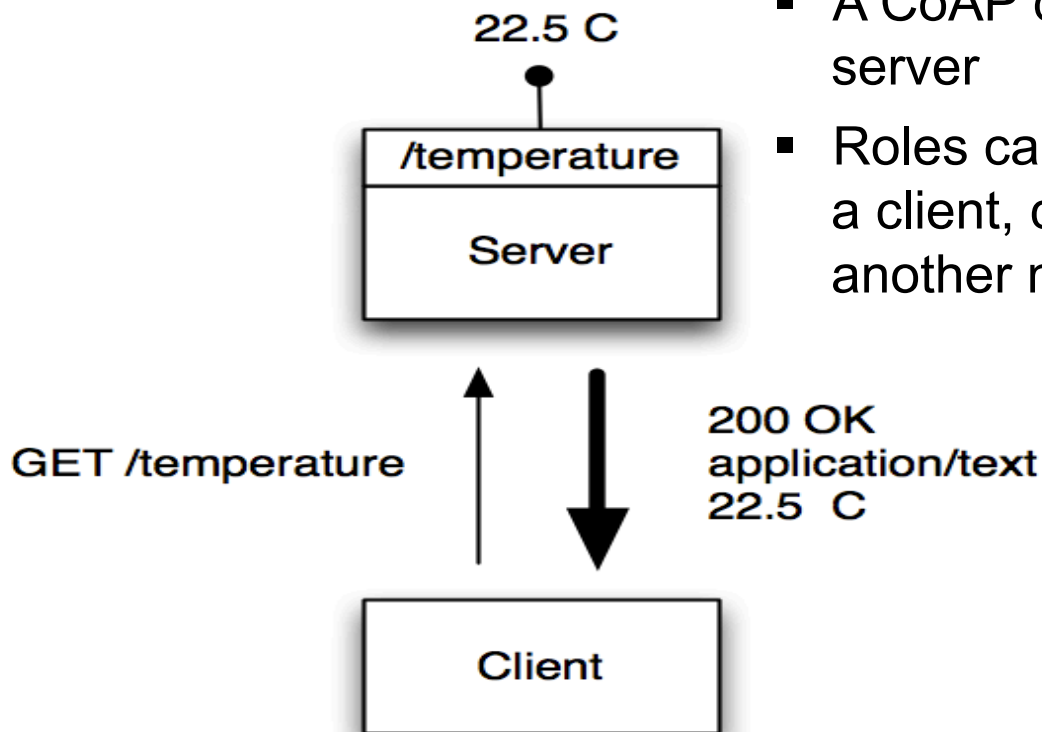


Protocol Layers and IoT Standards

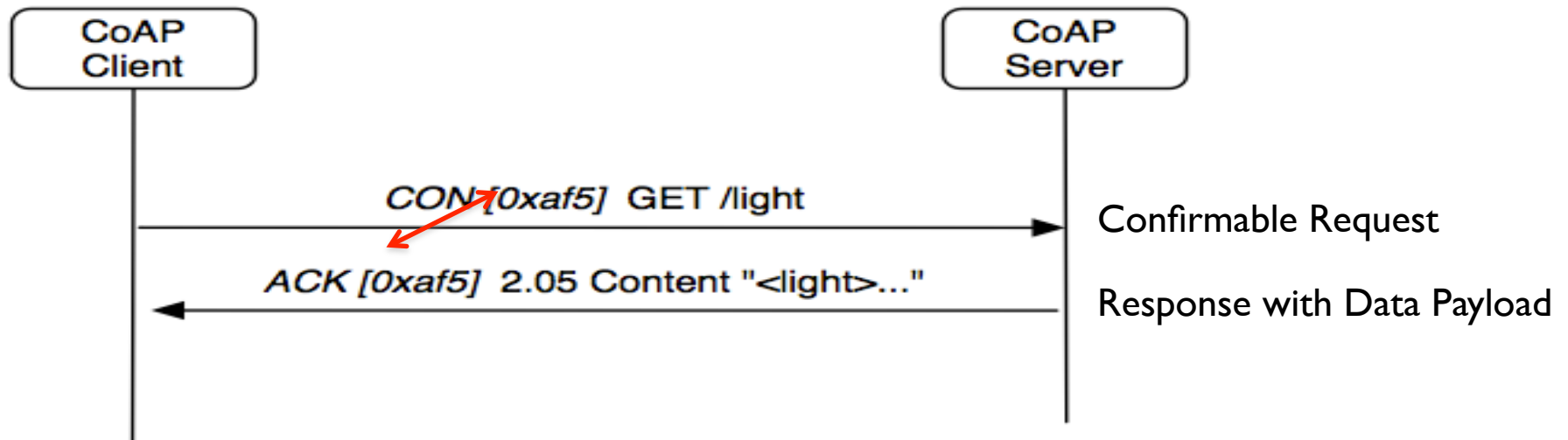


CoAP is REST for Constrained Devices

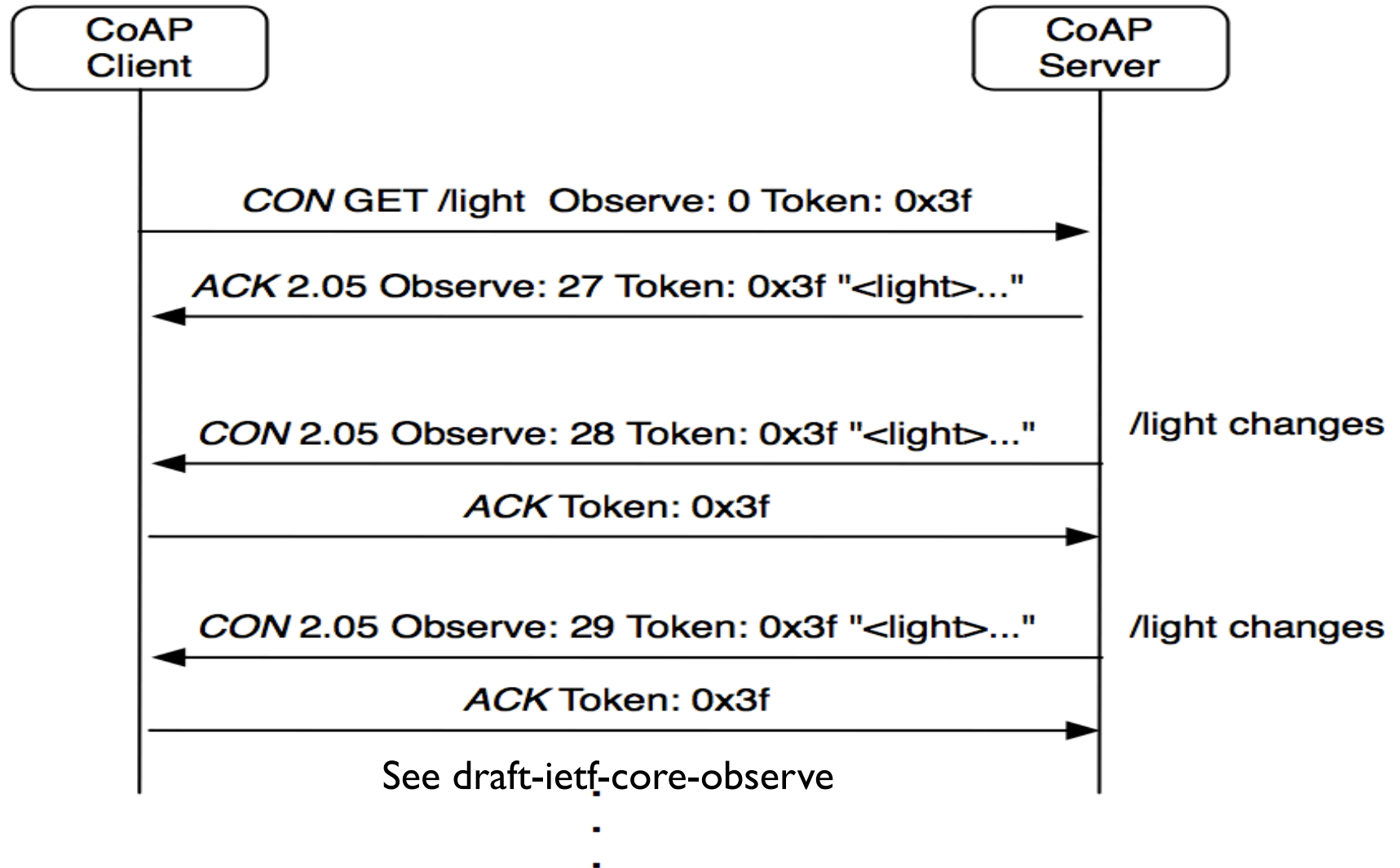
- Makes each device a lightweight server that exposes a REST API
- A CoAP device can be both client and server
- Roles can be reversed and the sensor, as a client, can update a REST API at another node, device or server



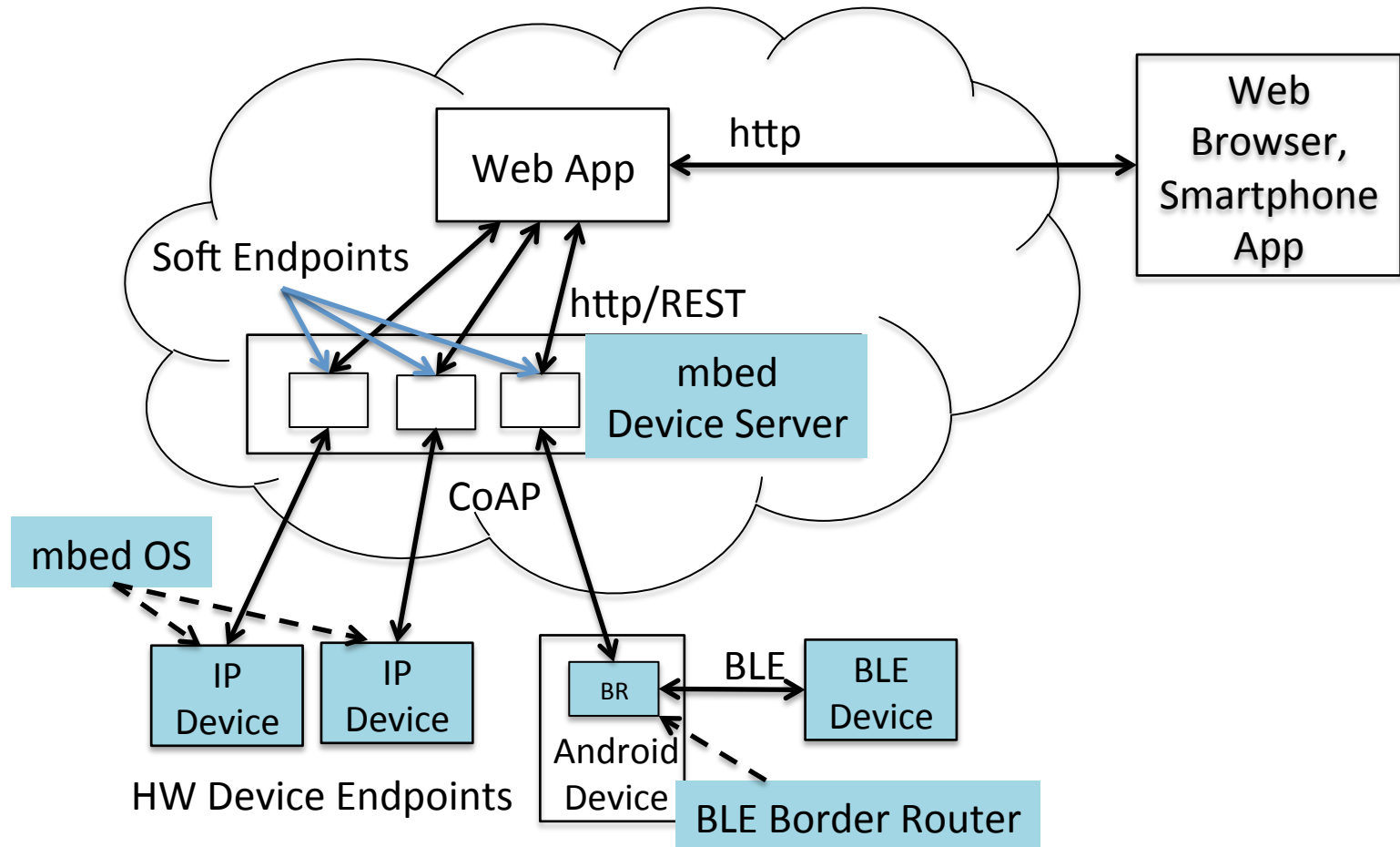
CoAP Request - Response



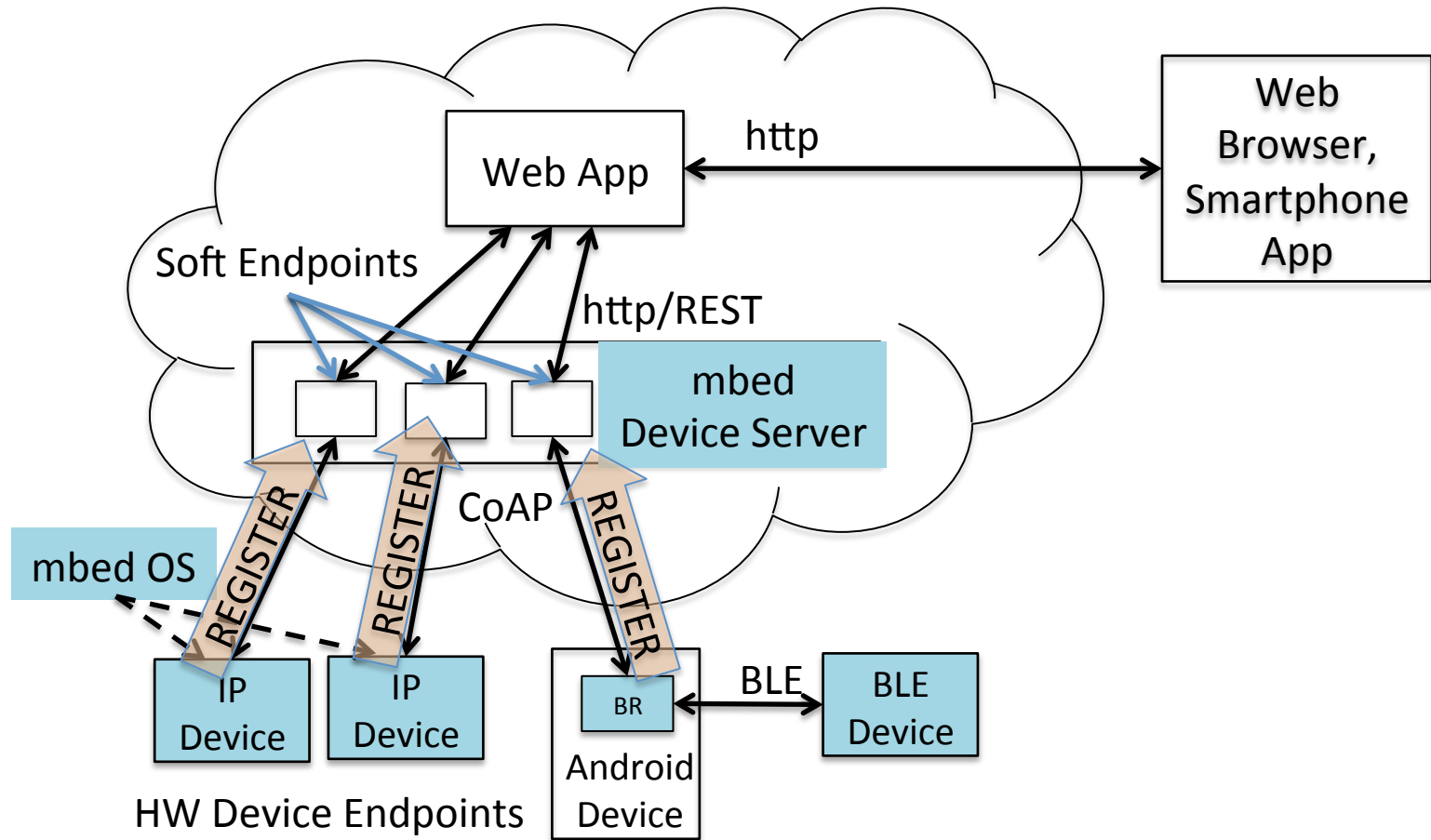
CoAP Observe – Asynchronous Notification



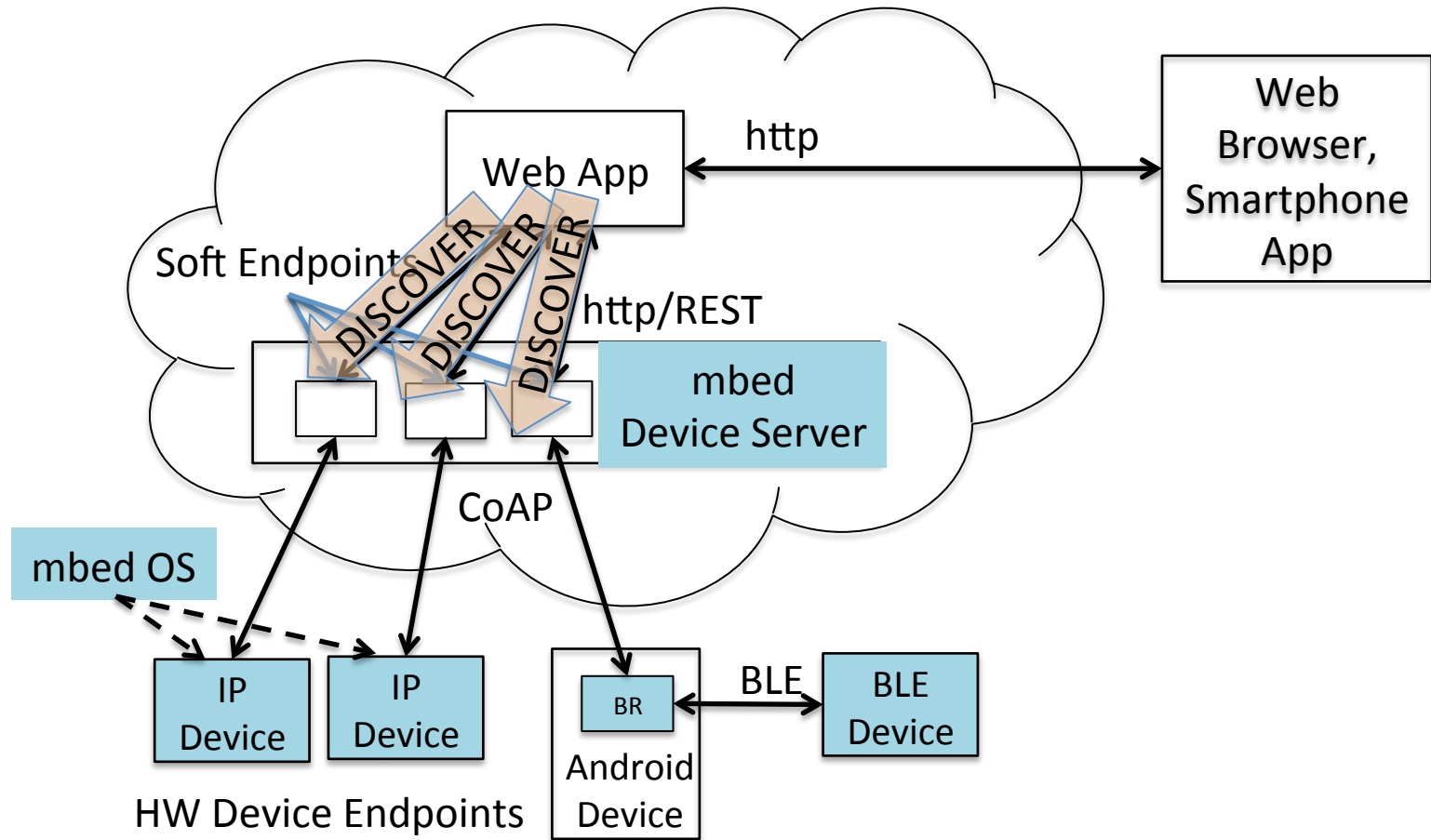
mbed Mapping to Reference Architecture



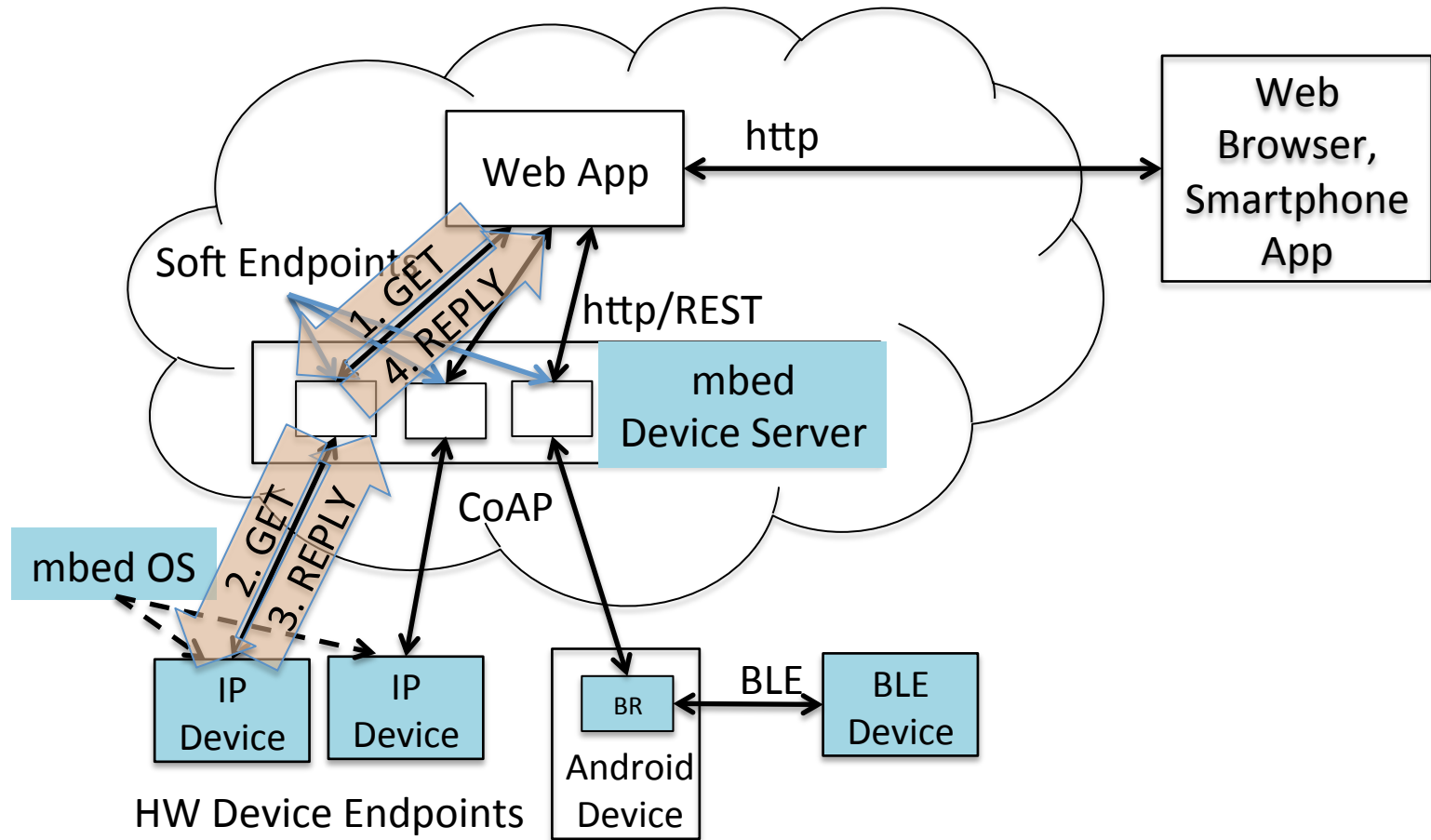
Registration



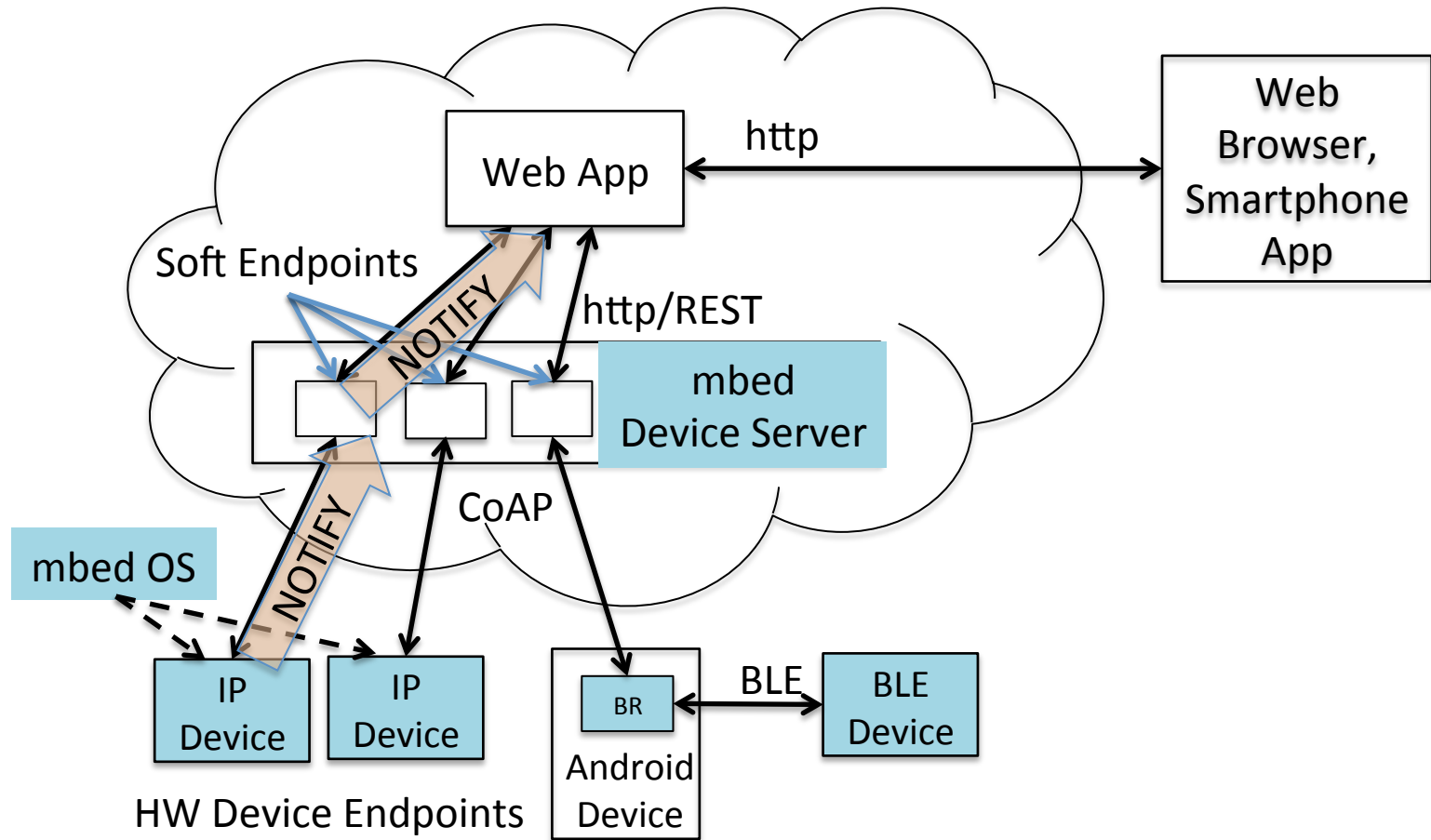
Discovery



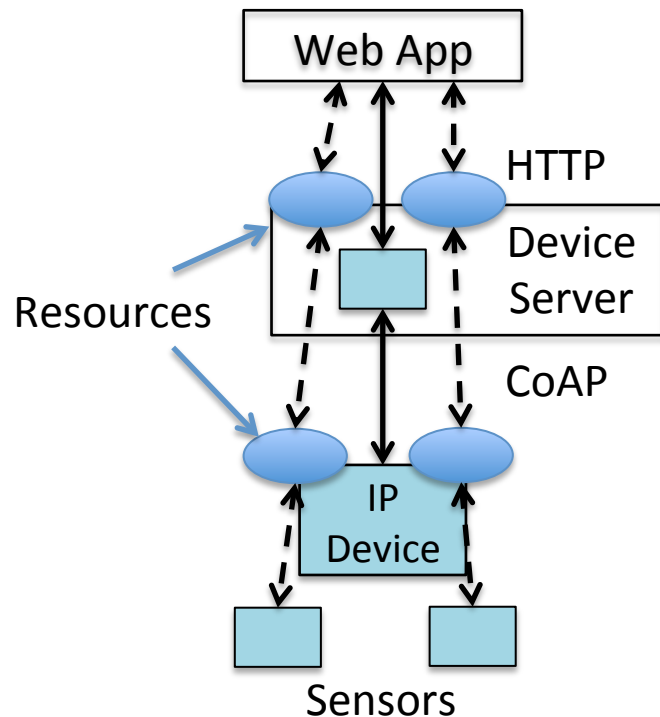
Example Transaction



Asynchronous Notification



Resources and Data Models



Resource:

- An addressable element of a REST API
- Has a path or URI
- May represent a sensor or actuator
- Has a set of operations e.g. GET, PUT
- Exposed by devices, registered with server
- Consumed by applications through server

The Data Model defines:

- How the path is constructed - URI template
- Which operations are allowed
- Other properties like data type

Smart Object Data Model - Temperature

Object info

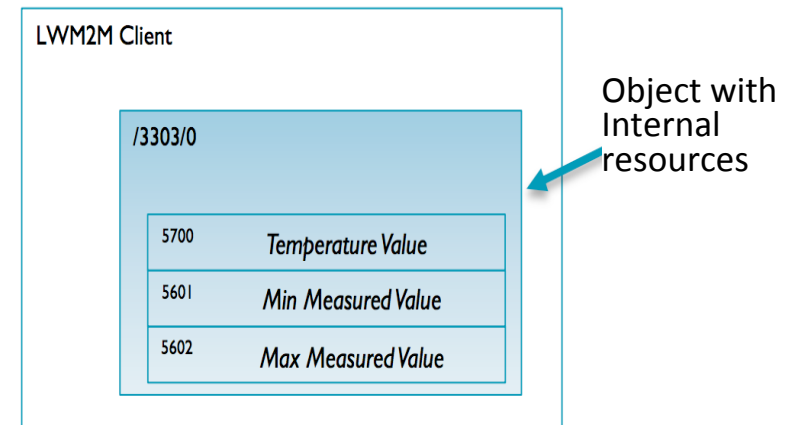
| Object | Object ID | Object URN | Multiple Instances? |
|------------------|-----------|------------------------|---------------------|
| IPSO Temperature | 3303 | urn:oma:lwm2m:ext:3303 | Yes |

Resource Info

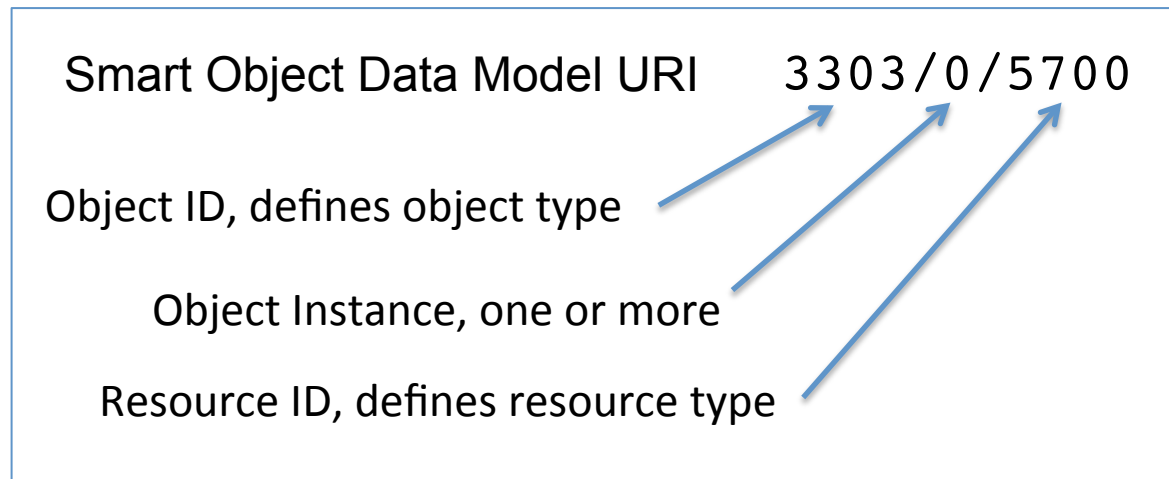
| Resource Name | Resource ID | Access Type | Multiple Instances? | Type | Units | Descriptions |
|--------------------|-------------|-------------|---------------------|---------|-------|---|
| Sensor Value | 5700 | R | No | Decimal | Cel | This resource type returns the Temperature Value in °C |
| Min Measured Value | 5601 | R | No | Decimal | Cel | The minimum value measured by the sensor since it is ON |
| Max Measured Value | 5602 | R | No | Decimal | Cel | The maximum value measured by the sensor since it is ON |

Accessing the Resources

- *Temperature Value* */3303/0/5700*
- *Min Measured Value* */3303/0/5601*
- *Max Measured Value* */3303/0/5602*



Smart Object Resource Design



- Objects represent single points of interest, for example a temperature sensor
- An object may have multiple instances
- Resources represent attributes of the object, for example the last measured value, the smallest measured value, the greatest measured value
- Object and resource IDs are meant to be reusable, representing common measurements and concepts
- Please see the IPSO Smart Object Guideline

Smart Object Starter Pack

Table 1 Smart Objects defined by this Technical Guideline

| Object | Object ID | Multiple Instances? |
|--------------------------------|-----------|---------------------|
| IPSO Digital Input | 3200 | Yes |
| IPSO Digital Output | 3201 | Yes |
| IPSO Analogue Input | 3202 | Yes |
| IPSO Analogue Output | 3203 | Yes |
| IPSO Generic Sensor | 3300 | Yes |
| IPSO Illuminance Sensor | 3301 | Yes |
| IPSO Presence Sensor | 3302 | Yes |
| IPSO Temperature Sensor | 3303 | Yes |
| IPSO Humidity Sensor | 3304 | Yes |
| IPSO Power Measurement | 3305 | Yes |
| IPSO Actuation | 3306 | Yes |
| IPSO Set Point | 3308 | Yes |
| IPSO Load Control | 3310 | Yes |
| IPSO Light Control | 3311 | Yes |
| IPSO Power Control | 3312 | Yes |
| IPSO Accelerometer | 3313 | Yes |
| IPSO Magnetometer | 3314 | Yes |
| IPSO Barometer | 3315 | Yes |

Device Programming for Resources

- The mbed library for LWM2M and Smart Objects provides resource classes (e.g. DynamicResource)
- **Two files** are involved in resource programming
- Resources are implemented in a **<resource>.h** file for each addressable resource, which is a wrapper for the generic class constructor and resource-specific code
- Resources are configured and created in the main program (e.g. **main.cpp**)
- Resources are registered with the Device Server when the endpoint start() is called, usually in main.cpp

Device Programming for Resources

- Refer to the lab note on resource programming for specific instructions
- Some attributes and parameters need to be customized for each device
 - Endpoint Name
 - Registration Domain
 - Device Server IP address
- And for each resource
 - Resource.h file
 - Resource path
 - Max-age to control cache lifetime - optional
 - Observation sample time - optional

Device Server API

- Devices register resources with a Device Server by uploading links
- Registrations have a lifetime and are refreshed periodically by the device
- The Device Server is a proxy for resources that are registered by devices
- Applications can access device resources by using the HTTP proxy, which exposes a web style REST API
- Each registered device resource has a corresponding web API resource
- Refer to the mbed Device Server User Guide

Device Server API – Endpoint Discovery

Device Server Endpoint Discovery URL

`http://10.10.10.10:8080/my_domain/endpoints`



Domain Name

- Endpoints can be discovered using the Device Server discovery interface
- Endpoint discovery returns a JSON array of endpoint objects

Device Server API – Endpoint Discovery

```
GET http://10.10.10.10:8080/my_domain/endpoints
```

```
0: {  
  name: "MBED-061590140030"  
  type: "LWM2M test client"  
  status: "ACTIVE"  
}-  
1: {  
  name: "mbed-6230600c000f"  
  type: "mbed_device"  
  status: "ACTIVE"  
}
```

Device Server API – Resource Discovery

Device Server Resource Discovery URL

`http://10.10.10.10:8080/my_domain/endpoints/my_ep_name`



- Resources can be discovered using the Device Server discovery interface
- Resource discovery returns a JSON array of resource objects

Device Server API – Resource Discovery

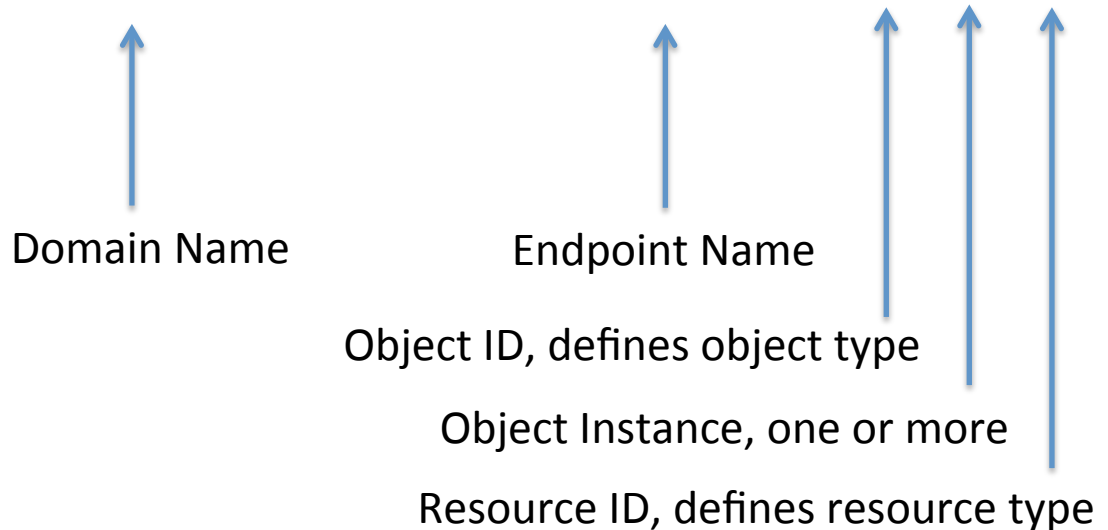
GET `http://10.10.10.10:8080/my_domain/endpoints/my_ep_name`

```
0: {  
  uri: "/3304/0/5700"  
  rt: "urn:X-ipso:humidity"  
  obs: false  
  type: ""  
}-  
1: {  
  uri: "/3303/0/5700"  
  rt: "urn:X-ipso:temperature"  
  obs: false  
  type: ""  
}-  
2: {  
  uri: "/3302/0/5500"  
  rt: "urn:X-ipso:presence"  
  obs: true  
  type: ""  
}-
```

Device Server API – Resource URL

Device Server Resource URL

`http://10.10.10.10:8080/my_domain/endpoints/my_ep_name/3303/0/5700`



- Web API URLs are constructed from the resource URIs registered by the device, the endpoint name of the registered device, and a preconfigured domain name

Device Server API – Resource URL

GET `http://10.10.10.10:8080/my_domain/endpoints/my_ep_name/3303/0/5700`

Returns -> 71.2

Standards References

IPSO Smart Object Guideline

<http://www.ipso-alliance.org/technical-information/ipso-guidelines>

OMA LWM2M Specification

<http://openmobilealliance.hs-sites.com/lightweight-m2m-specification-from-oma>

IETF CoAP and Related Specifications

CoAP (RFC 7252):

<http://tools.ietf.org/html/rfc7252>

CoRE Link-Format (RFC 6690):

<http://tools.ietf.org/html/rfc6690>

CoRE Resource Directory:

<http://tools.ietf.org/html/draft-ietf-core-resource-directory-01>

CoAP Community Site

<http://coap.technology/>