IPSO Smart Objects

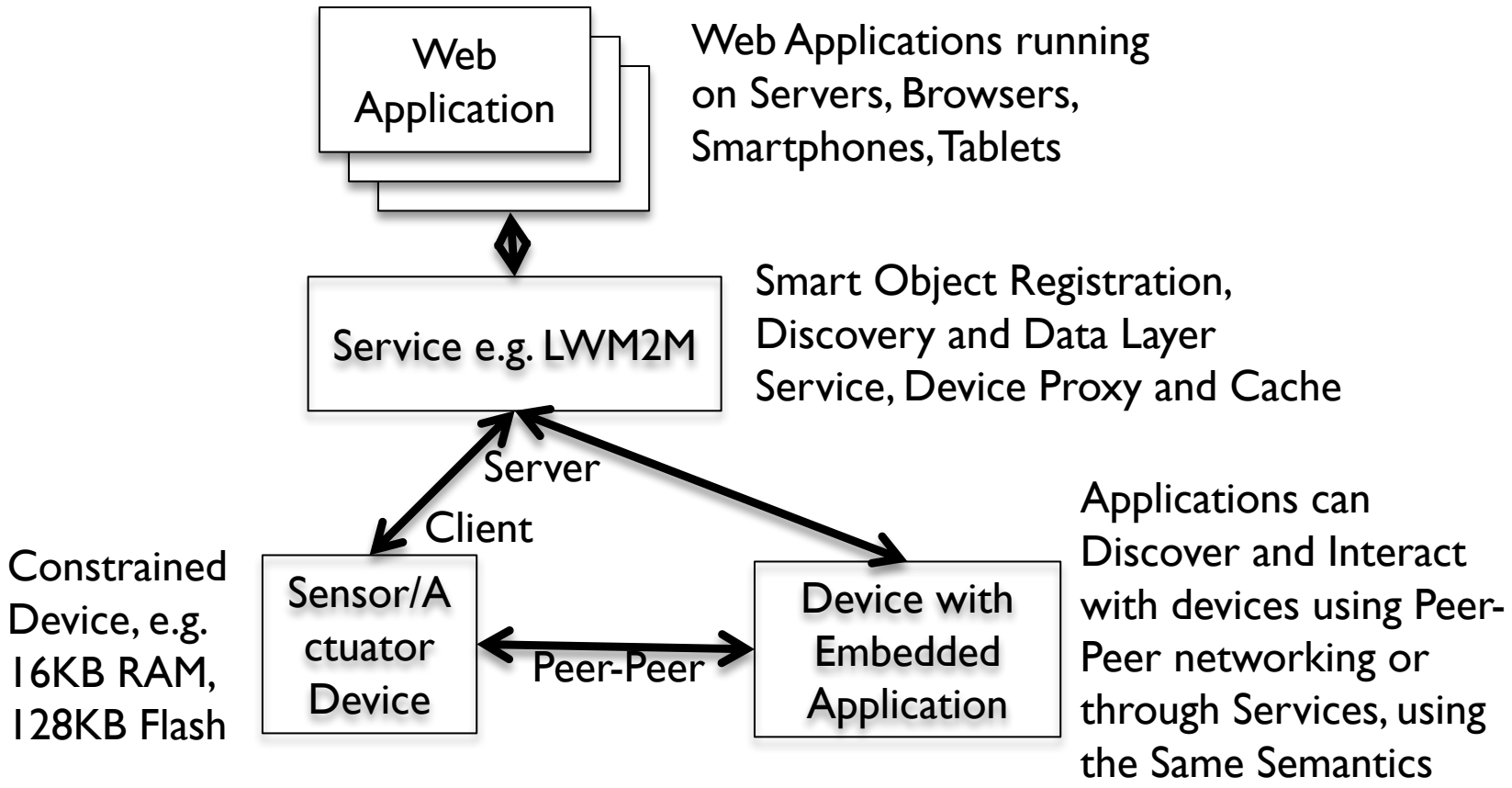
And Related Standards for IoT Interoperability

November18, 2014

IPSO Smart Objects

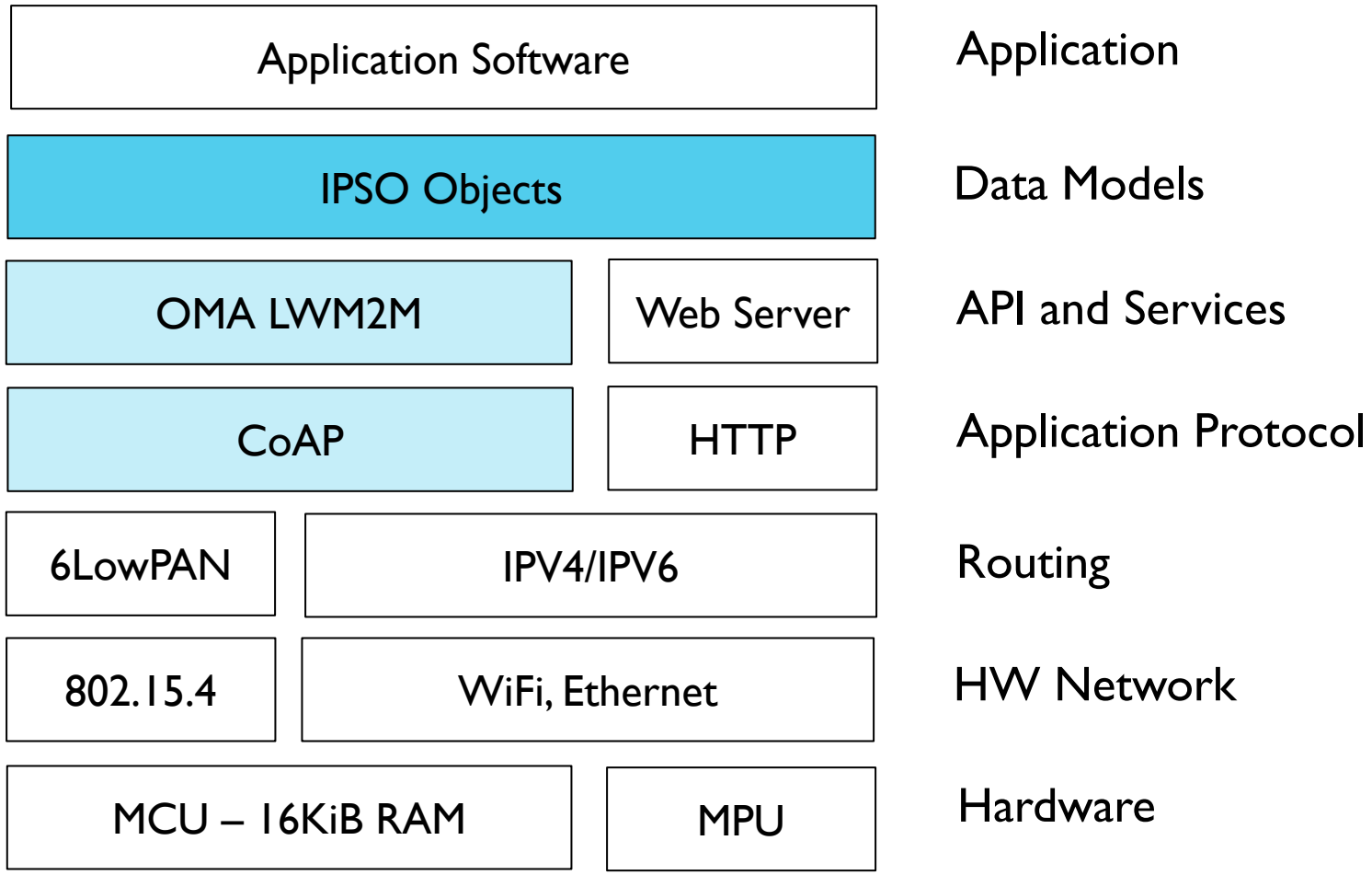
- A simple Data Model for
- Semantic Interoperability across IoT Devices
- Requires only simple URI addressing and a few data types
- Mapping to internet content-types
- Usable on many different transport protocols (CoAP, HTTP, MQTT)
- This presentation discusses the architecture and features of a rich implementation of IPSO Smart Objects on constrained devices and networks using CoAP and OMA LWM2M

IPSO Smart Object Architecture Use Cases



ARM

IPSO Smart Objects And Related IoT Standards



ARM

IoT Standards That Build On Each Other

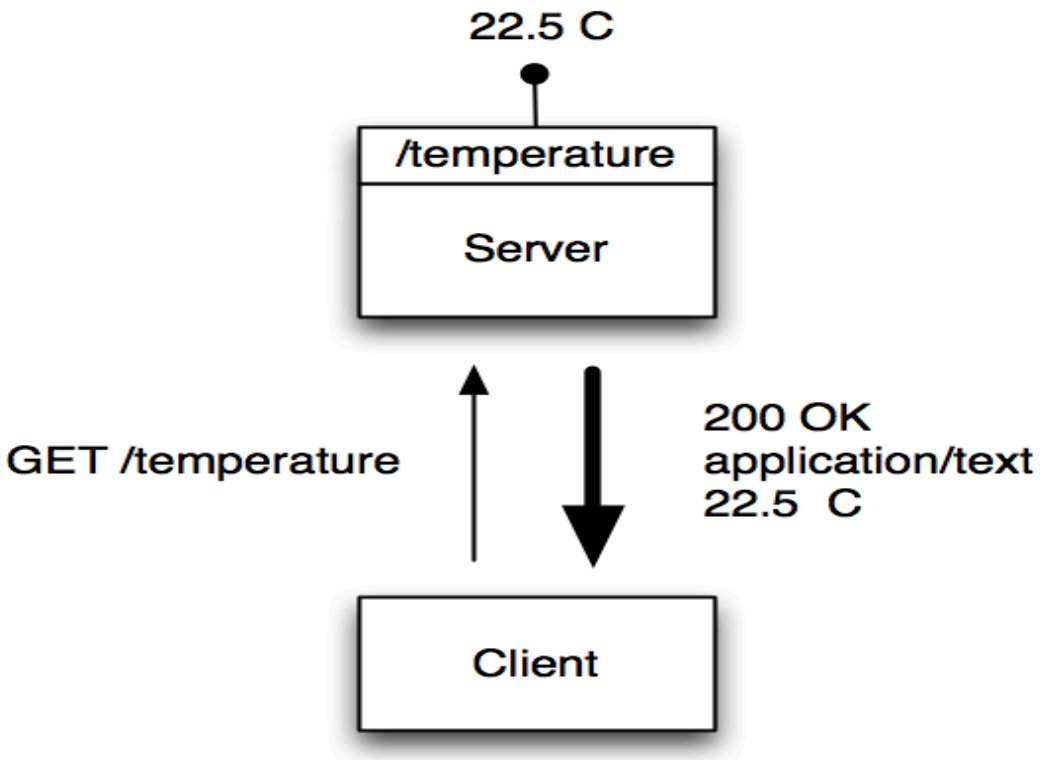
- CoAP and related standards from IETF
  - REST API for constrained networks and devices
  - HTTP Proxy provides abstraction through standard web APIs
  - Core-link-format (RFC 6690) provides semantic descriptors in the form of web links
  - Resource Directory provides an API for scalable discovery and linking using core link-format mediatype
- OMA LWM2M is based on CoAP
  - Provides a server profile for IoT middleware
  - Defines a simple reusable object model
  - Defines management objects and reuses REST API for onboarding and device life cycle management
- IPSO Smart Objects are based on OMA LWM2M
  - Defines application objects using the LWM2M Object Model
  - Complex objects can be composed from simple objects
  - Easy to add new resource and object types as needed

ARM

CoAP

ARM

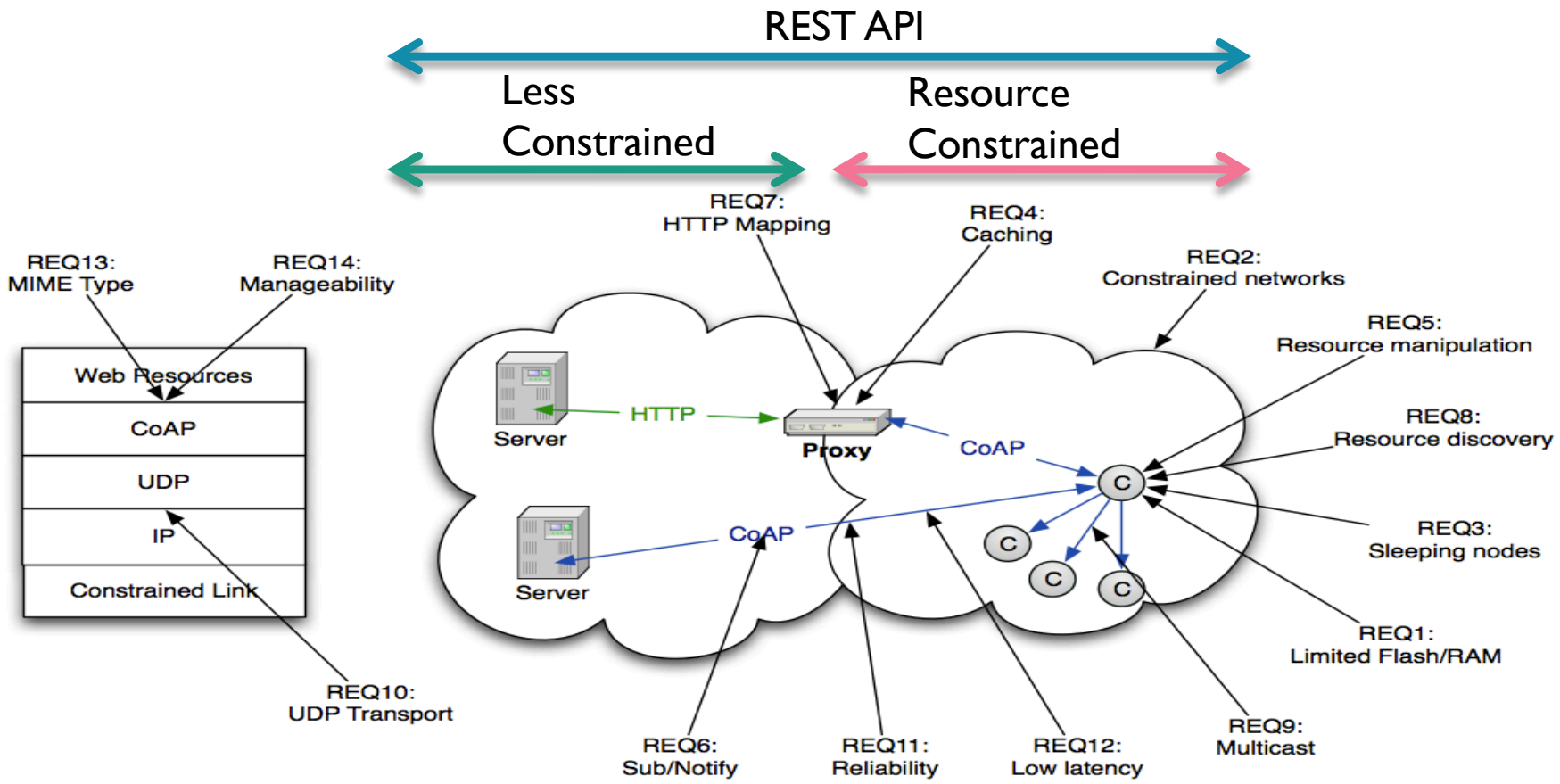
CoAP Protocol



- Makes each device a lightweight server that exposes a REST API
- A CoAP endpoint can be both client and server
- Roles can be reversed and the sensor, as a client, can also interact with a REST API at another endpoint or server node
- Peer to Peer interaction is based on a duplex client-server pattern

ARM

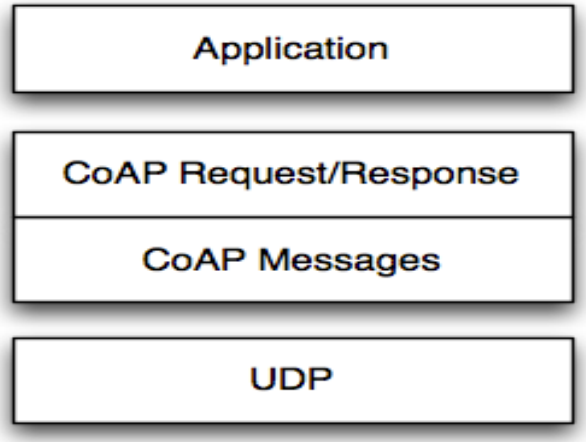
CoAP Use Case Requirements



ARM

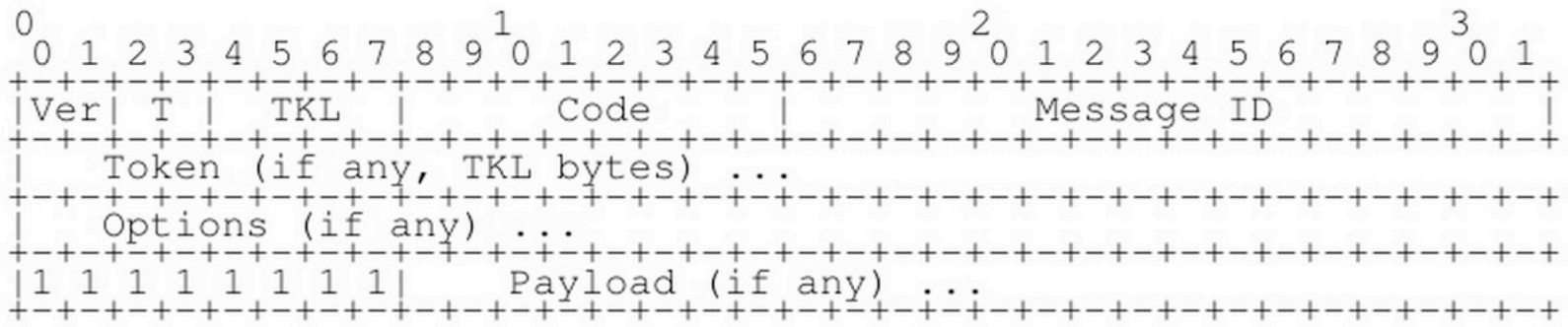
CoAP Scope

- Transport
  - CoAP currently defines:
  - UDP binding with DTLS security
  - CoAP over SMS or TCP possible
- Base Messaging
  - Simple message exchange between endpoints
  - Confirmable or Non-Confirmable Message
  - Answered by Acknowledgement or Reset Message
- REST Semantics
  - REST Request/Response mapped onto CoAP Messages
  - Method, Response Code and Options (URI, content-type etc.) define REST exchanges, very similar to HTTP (HTTP 404 response semantics (not found) mapped to CoAP 4.04 response code)
- Asynchronous Notifications
  - Observer option for GET allows asynchronous state update responses from a single request



ARM

CoAP Maps HTTP-like protocol to a binary format



Ver - Version (1)

T - Message Type (Confirmable, Non-Confirmable, Acknowledgement, Reset)

TKL- Token Length, if any, the number of Token bytes after this header

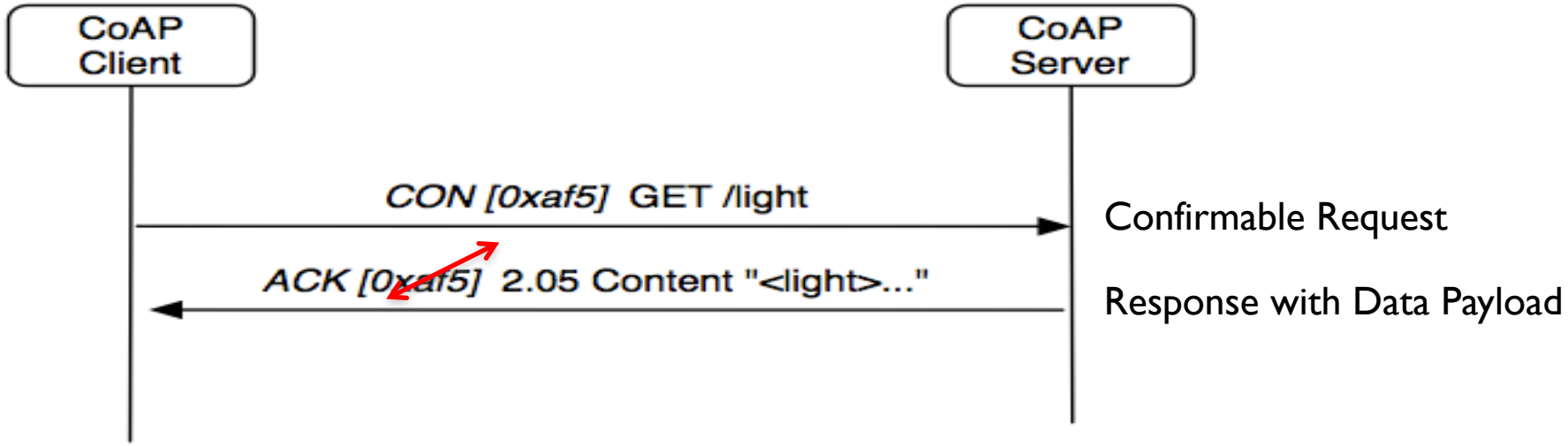
Code - Request Method (1-10) or Response Code (40-255)

Message ID - 16-bit identifier for matching responses

Token - Optional response matching token

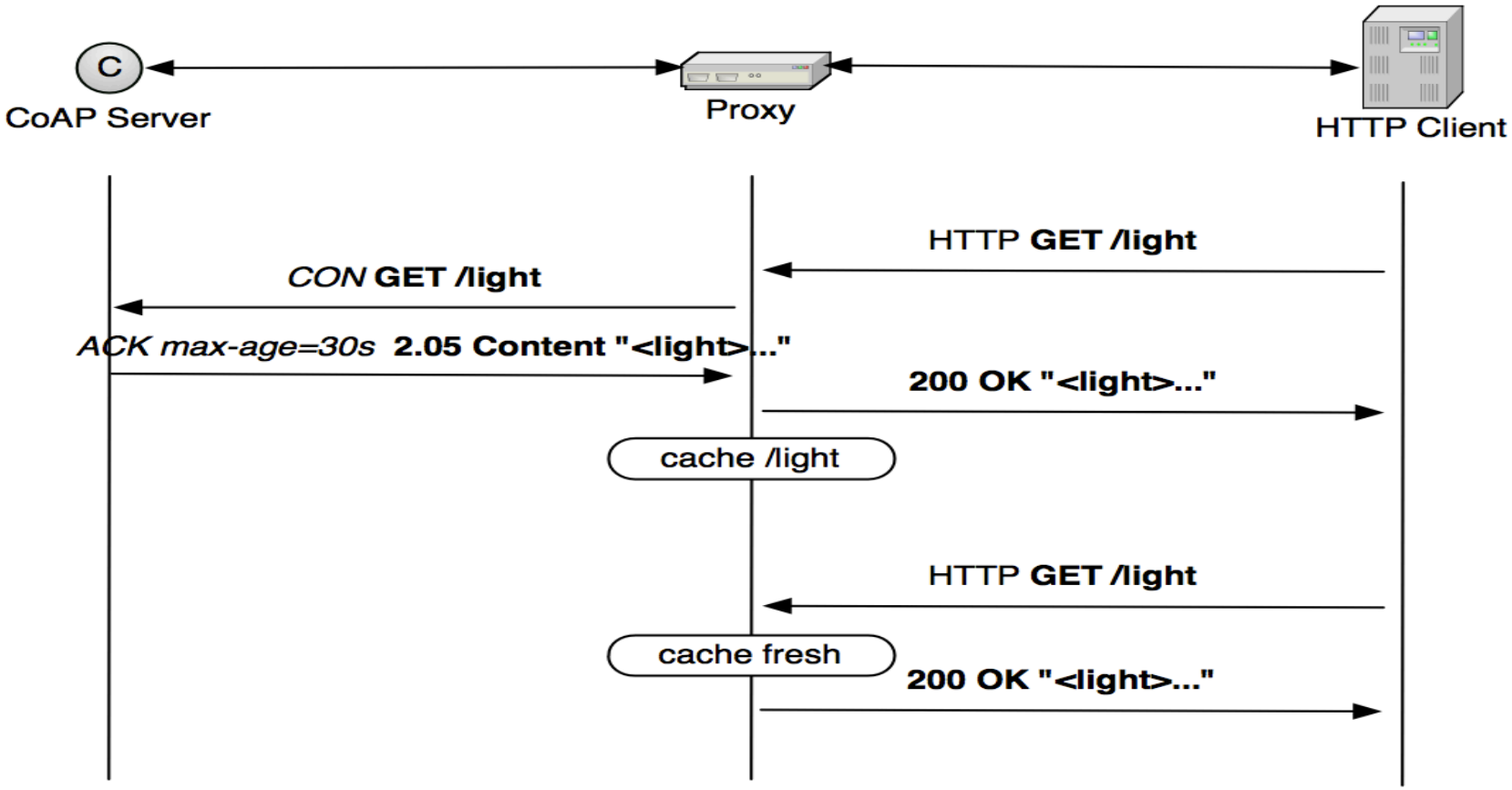
ARM

CoAP Example: GET Transaction



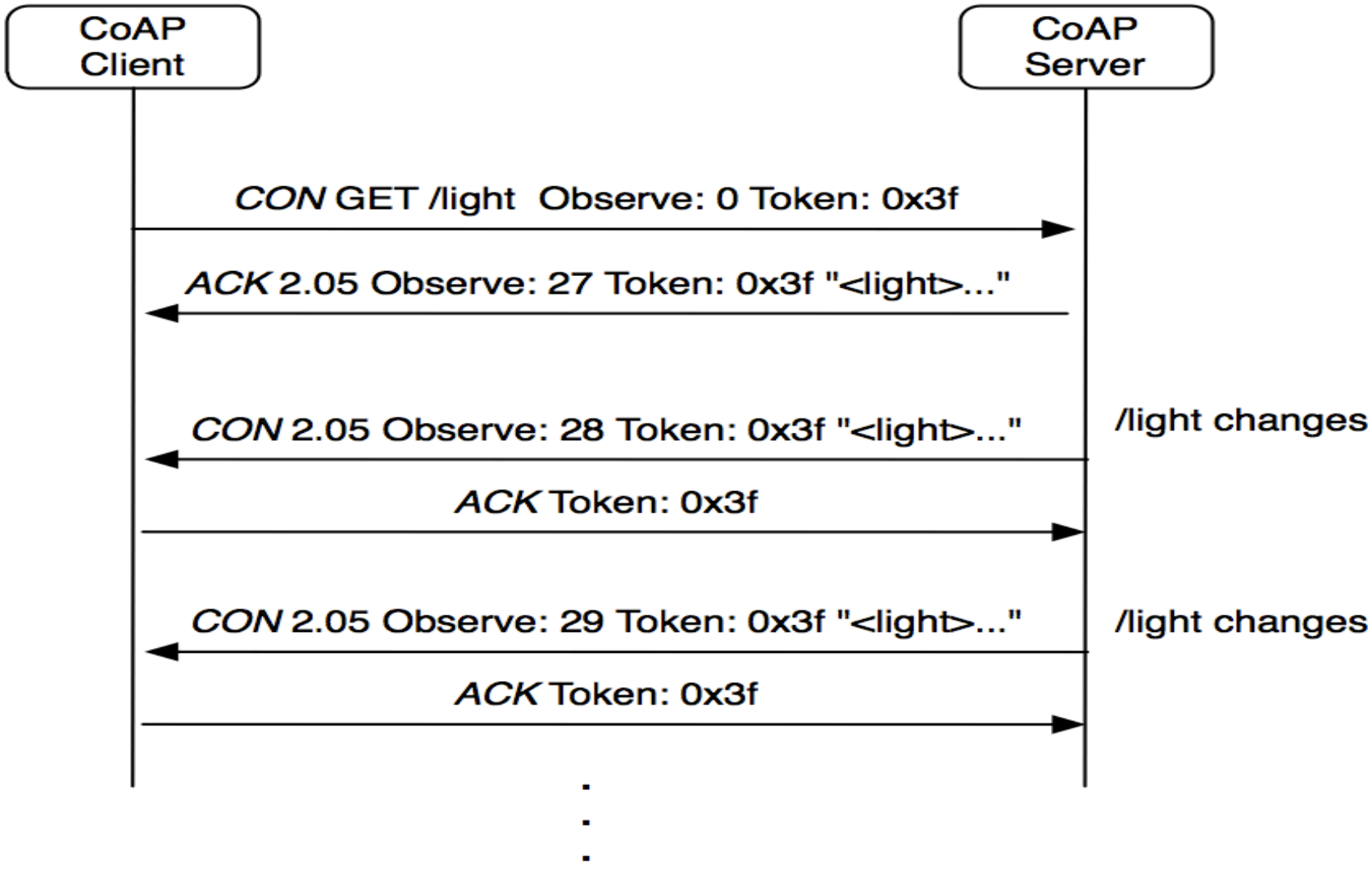
ARM

CoAP Proxy Caching



ARM

CoAP Asynchronous Notification

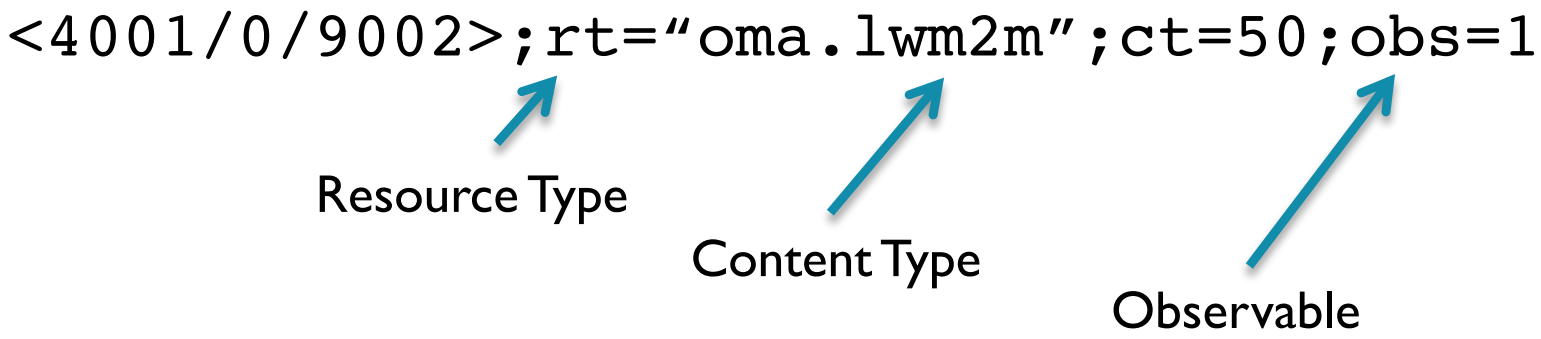


ARM

Web Linking for Machines

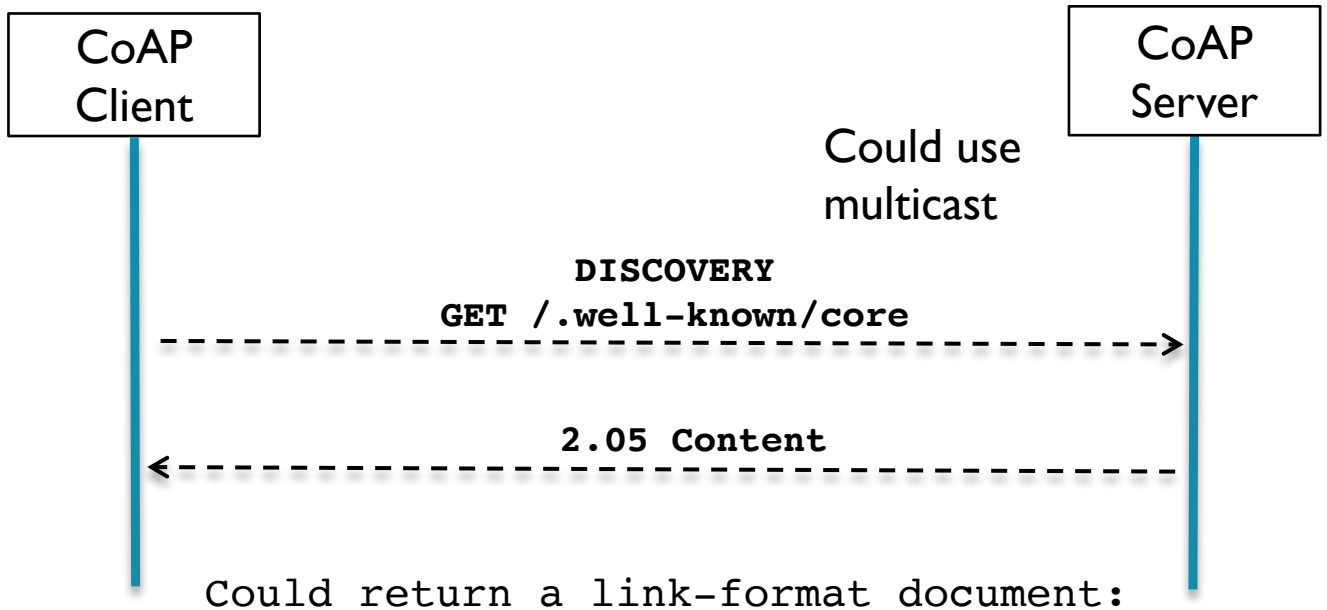
- RFC6690 is aimed at Resource Discovery and Linking for M2M
  - Defines semantic link serialization and content-types suitable for M2M
  - Defines a well-known resource where links are stored
  - Enables query string parameters for discovery by attribute and relation
  - Can be used with unicast or multicast (CoAP)
- Resource Discovery with RFC6690
  - Discovering the links hosted by CoAP (or HTTP) servers
  - GET /.well-known/core?optional_query_string
  - Returns a link-format document
  - URL, resource type, interface type, content-type, size are some basic relations

RFC 6690 CoRE Link-Format



ARM

Local Network Discovery



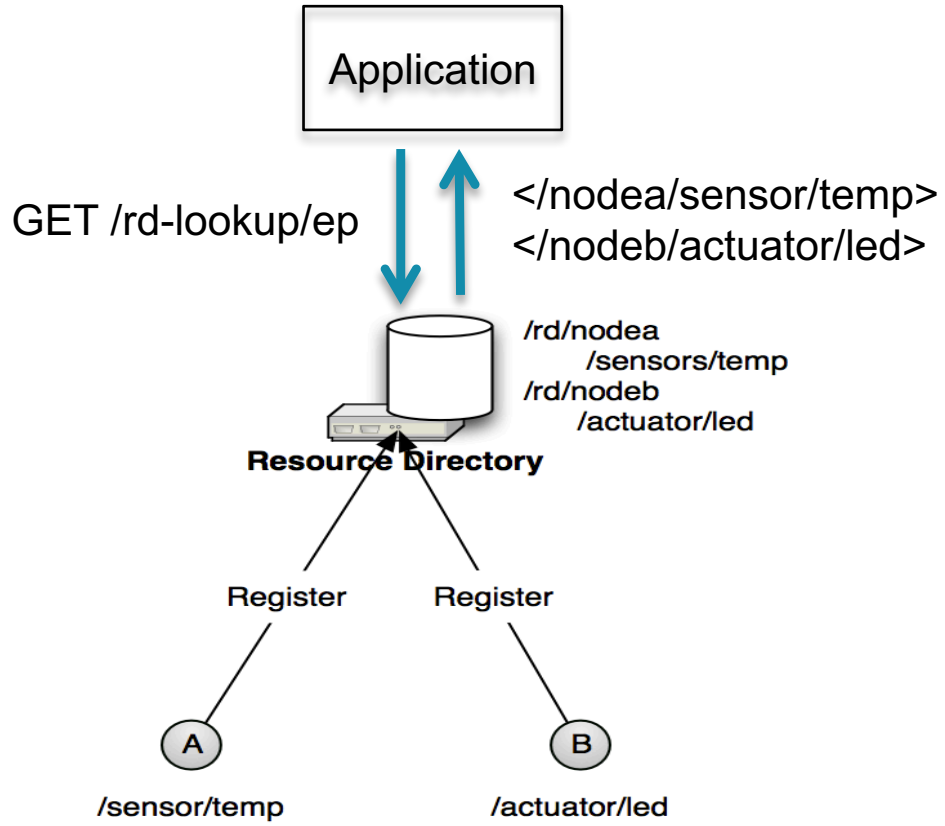
Could return a link-format document:

```
</3//9>;obs;rt="urn:X-ipso:batt-level";ct="50",
</3//0>;rt="urn:X-ipso:dev-mdl";ct="50",
</3//1>;rt="urn:X-ipso:dev-mfg";ct="50",
</3305/0/5800>;obs;rt="urn:X-ipso:pwr-w";ct="50",
</3305/0/5805>;obs;rt="urn:X-ipso:pwr-accum-wh";ct="50",
</3303/0/5700>;obs;rt="urn:X-ipso:temp-C";ct="50"
```


ARM

Resource Discovery

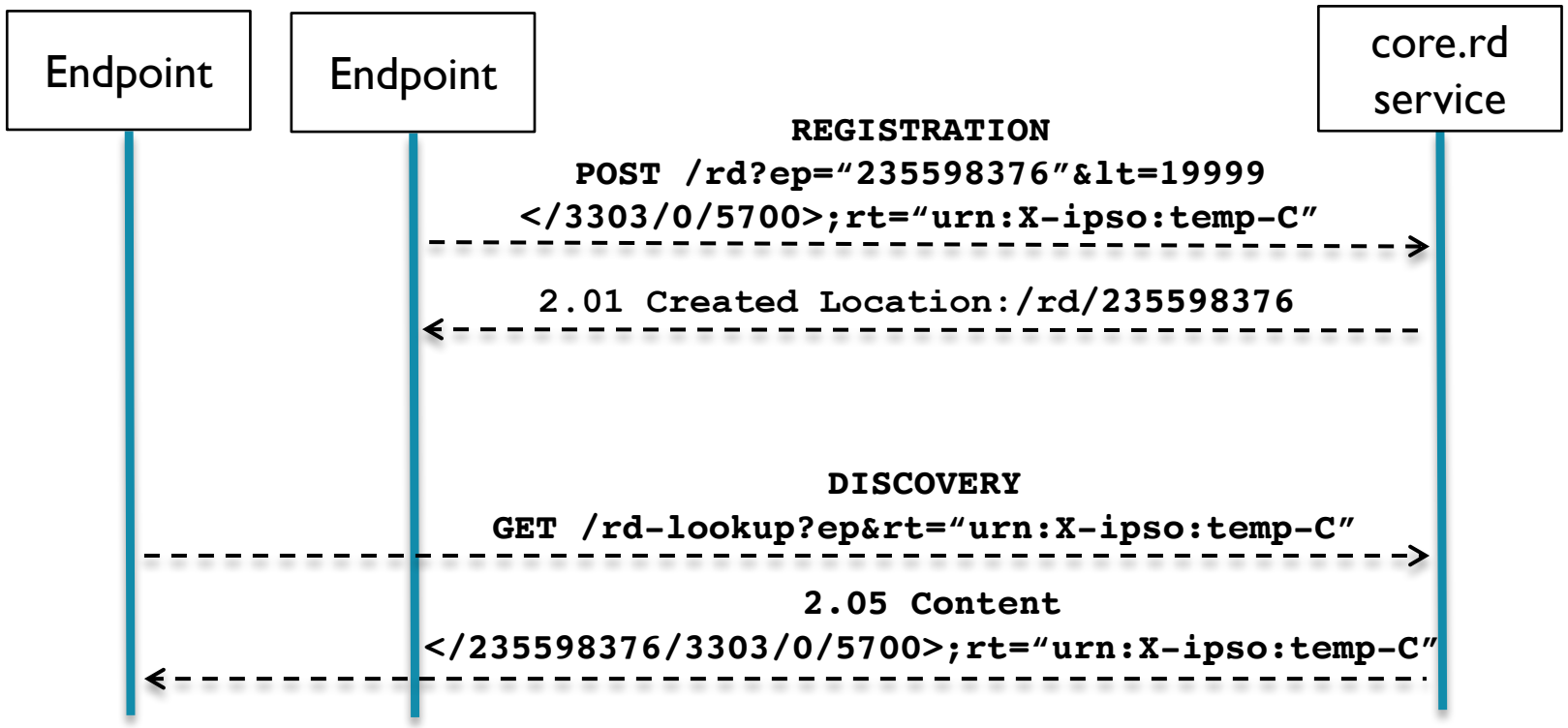
- RFC 6690 CoRE Link Format defines
  - The link format media type
  - Peer-to-peer discovery
- A directory approach is also useful
  - Supports sleeping nodes
  - No multicast traffic, longer battery life
  - Remote lookup, hierarchical and federated distribution
- CoRE Link Format is used in Resource Directories
  - Nodes register their resource links to an RD
  - Nodes refresh the RD periodically
  - Nodes may unregister (remove) their RD entry



See draft-ietf-core-resource-directory

ARM

Resource Discovery Example Flow



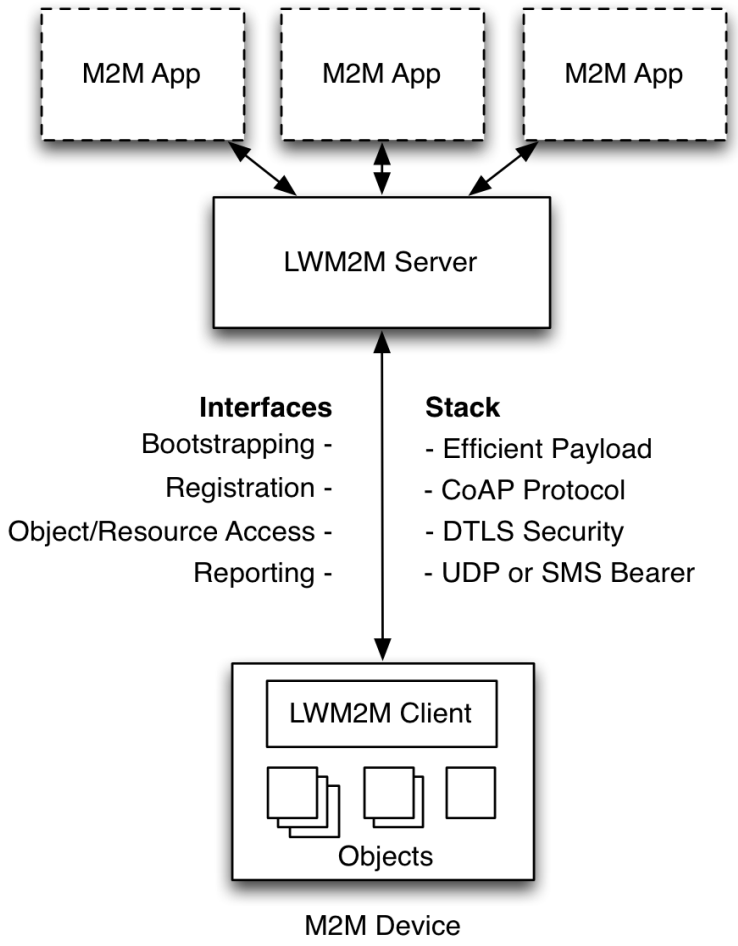
See draft-ietf-core-resource-directory

ARM

OMA LWM2M

ARM

OMA LWM2M Reference Architecture

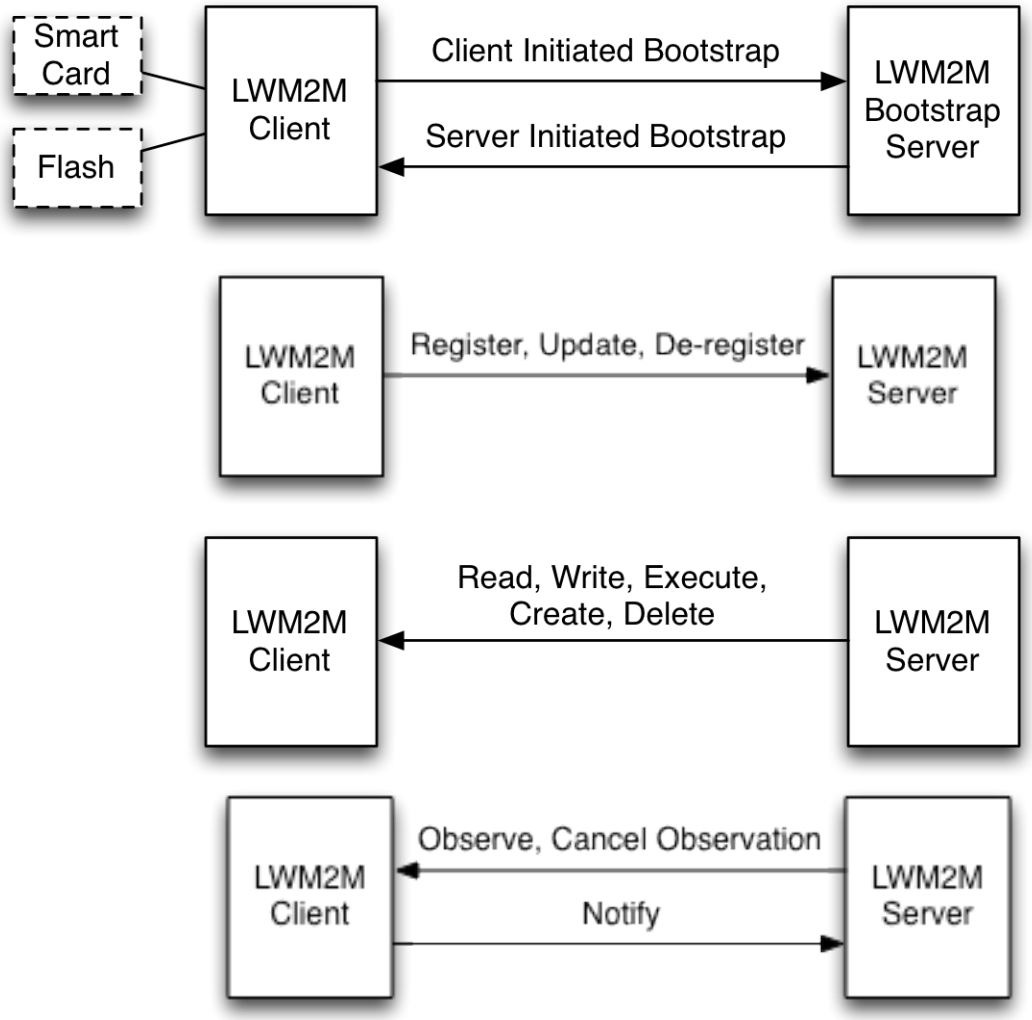


- M2M Applications
  - Application abstraction through REST API
  - Resource Discovery and Linking
- LWM2M Server
  - CoAP Protocol
  - Supports HTTP Caching Proxy
  - Resource Directory
  - Gateway and Cloud deployable
- LWM2M Clients are Devices
  - Device abstraction through CoAP
  - LWM2M Clients are CoAP Servers
  - Any IP network connection

ARM

LWM2M Interfaces

- Bootstrap Interface
  - Configure Servers & Keying
  - Pre-Configured, Smart Card, or Server Initiated Bootstrap
  - CoAP REST API
- Registration Interface
  - RFC6690 and Resource Directory
- Management Interface Using Objects
  - Management Objects and Resources
  - CoAP REST API
- Reporting Interface
  - Object Instances and Resources Report
  - Asynchronous notification using CoAP Observe



ARM

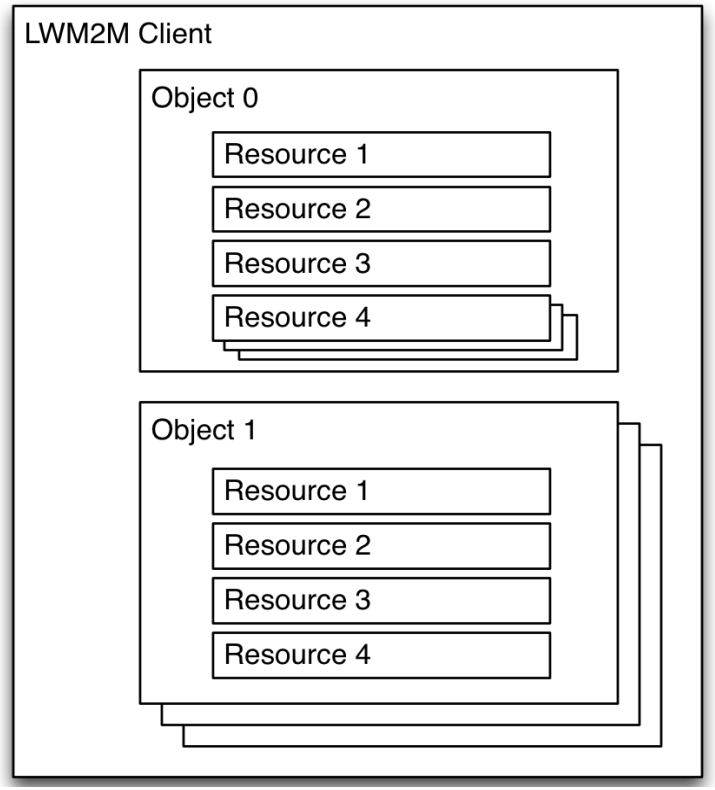
LWM2M Object Model

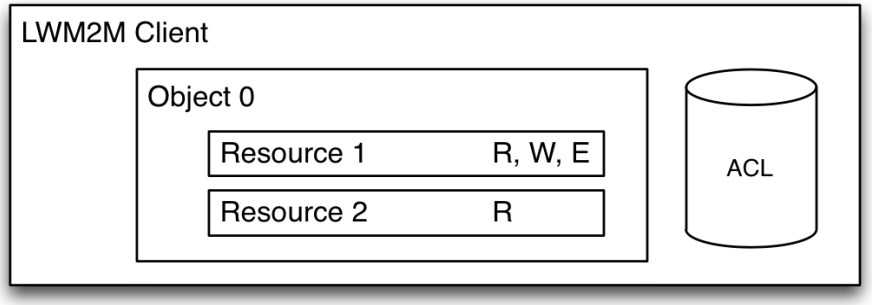
- A Client has one or more Object Instances
- An Object is a collection of Resources
- A Resource is an atomic piece of information that can be
  - Read, Written or Executed
- Objects can have multiple instances
- Objects and Resources are identified by a 16-bit Integer, Instances by an 8-bit Integer
- Objects/Resources are accessed with simple URIs:

  /{Object ID}/{Object Instance}/{Resource ID}

Example:

/3/0/1 - Object Type=3 (Device), Instance=0, Resource Type = 1 (Device Mfg.)





ARM

LWM2M Management Objects

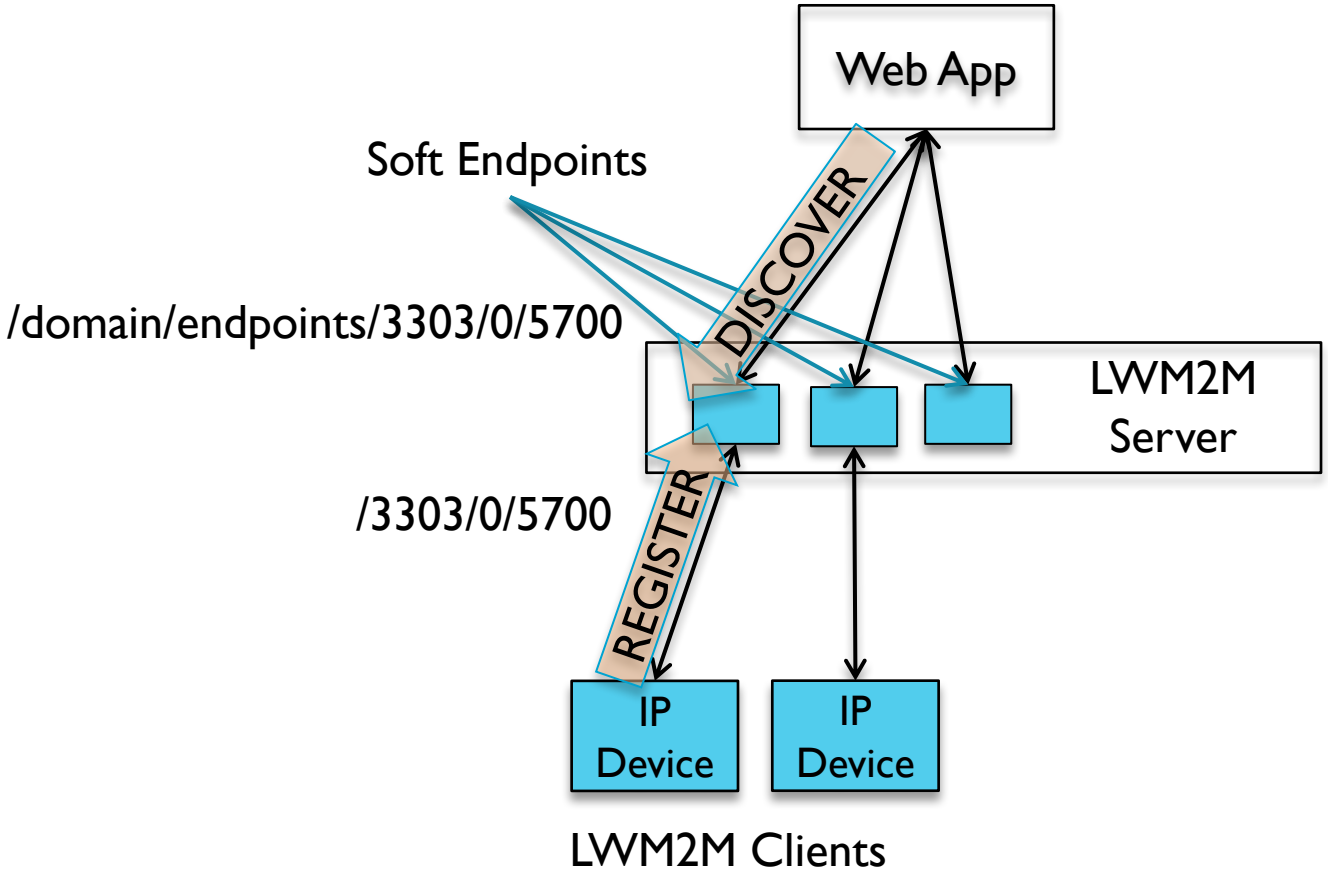
Object	Object ID
LWM2M Security	0
LWM2M Server	1
Access Control	2
Device	3
Connectivity Monitoring	4
Firmware	5
Location	6
Connectivity Statistics	7

LWM2M Position Object Example, OMA Template

Resource Name	ID	Access Type	Multiple Instances?	Type	Range	Units	Descriptions
Latitude	0	R	No	Decimal		Deg	The decimal notation of latitude, e.g. -43.5723 [World Geodetic System 1984]
Longitude	1	R	No	Decimal		Deg	The decimal notation of longitude, e.g. 153.21760 [World Geodetic System 1984]
Altitude	2	R	No	Decimal		m	The decimal notation of altitude in meters above sea level.
Uncertainty	3	R	No	Decimal		m	The accuracy of the position in meters.
Velocity	4	R	No	Refers to 3GPP GAD specs		Refers to 3GPP GAD specs	The velocity of the device as defined in 3GPP 23.032 GAD specification. This set of values may not be available if the device is static.
Timestamp	5	R	No	Time			The timestamp of when the location measurement was performed.

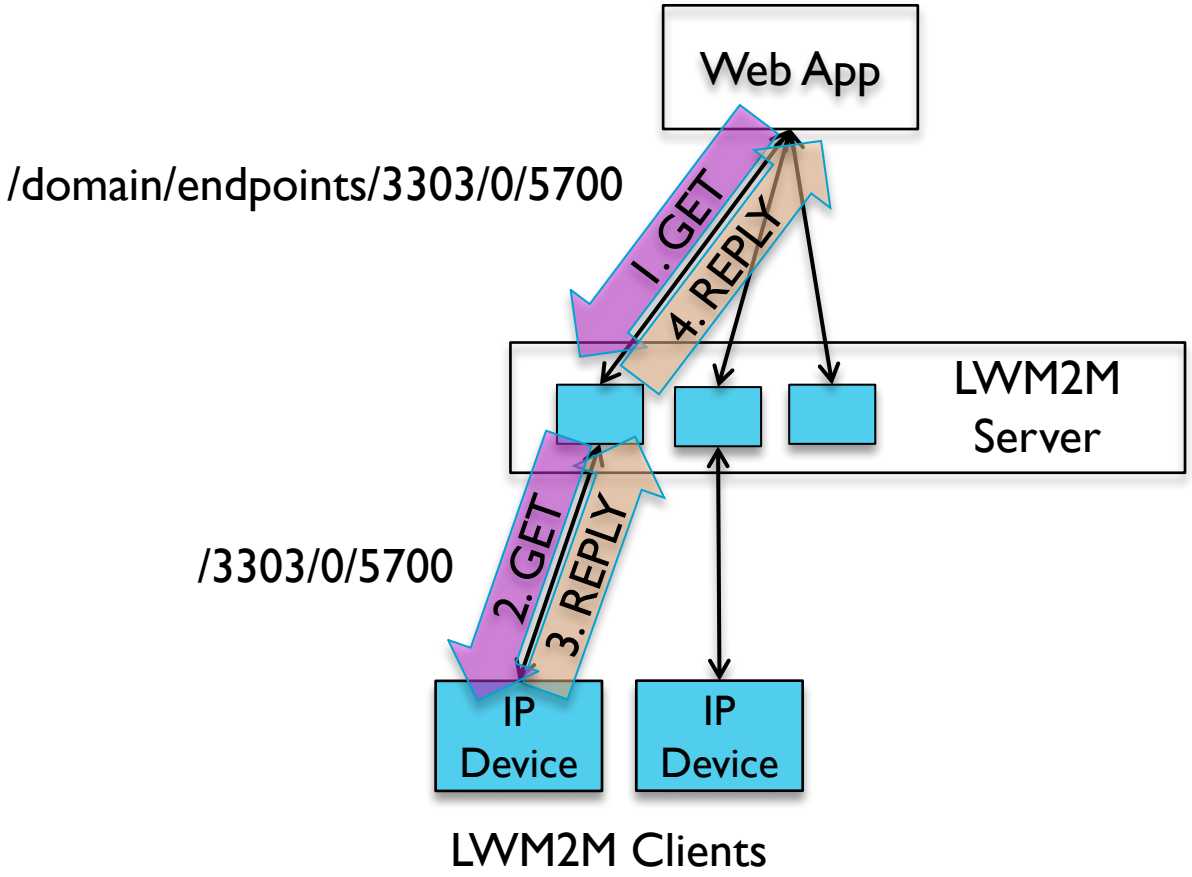
ARM

LWM2M Application Server



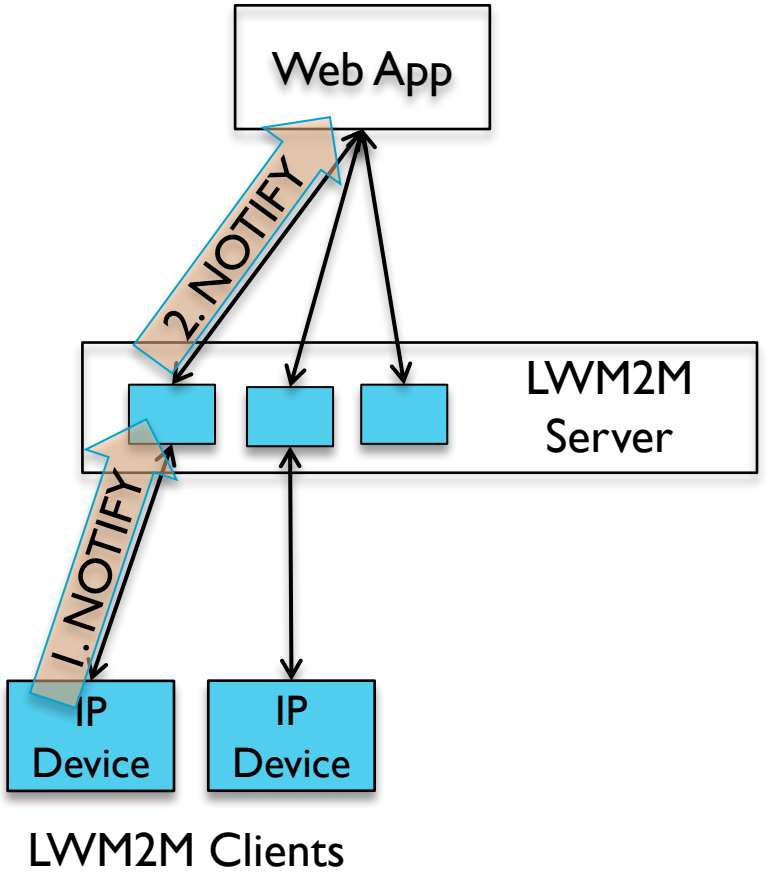
ARM

LWM2M Application Server



ARM

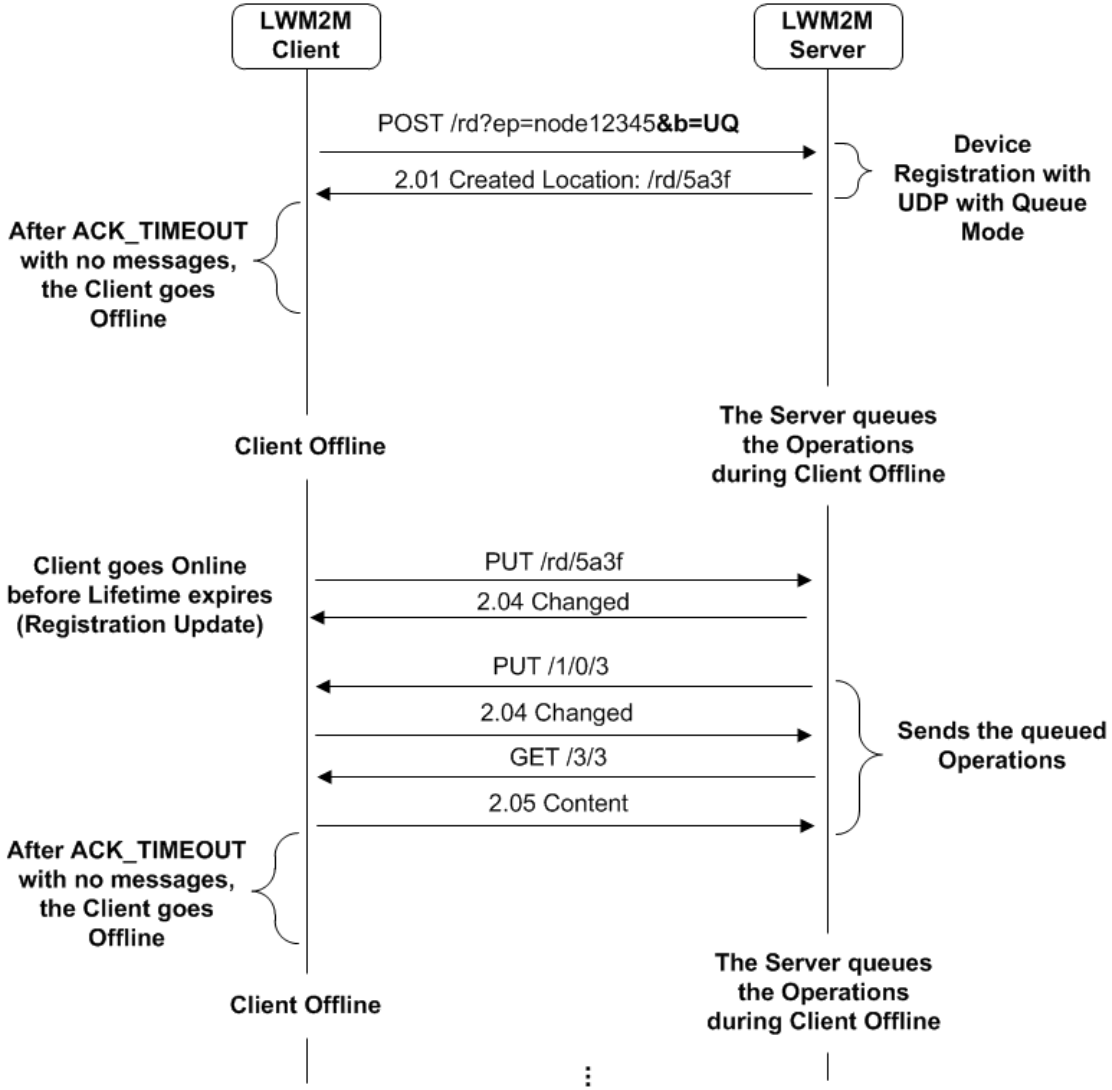
LWM2M Application Server



ARM

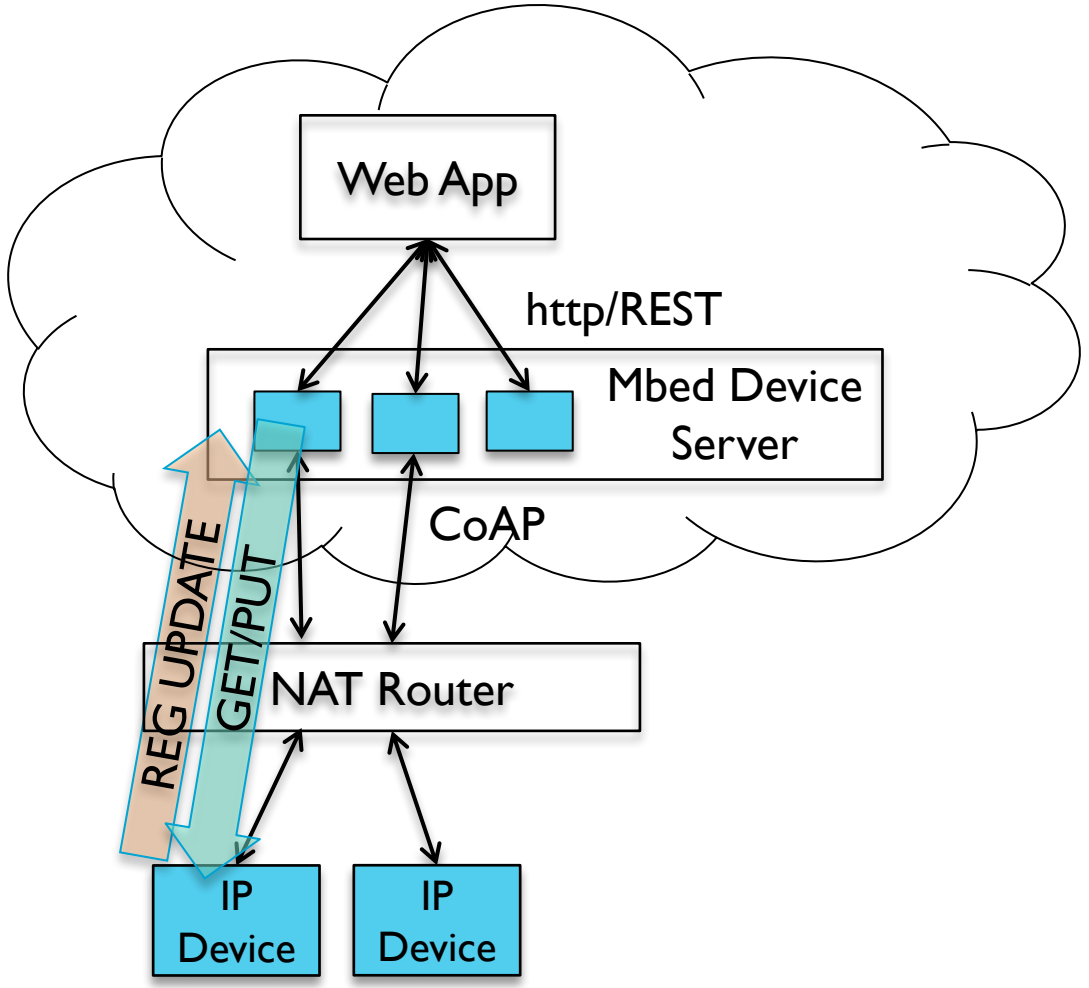
LWM2M Supports Sleeping Endpoints "b=uq"

- Client uses the registration refresh to inform LWM2M server that it is awake, and listens for any queued operations



ARM

LWM2M Communication Through NAT Router



HW Device Endpoints

ARM

LWM2M Observe Parameters

- LWM2M provides a mechanism to control Observation
- "Write Attributes" Interface using query parameters to set observe attributes:
- Pmin – minimum observation quiet period, to limit notification frequency
- Pmax – maximum observation quiet period, to guarantee notifications
- Lt – low limit measurement notification, like low alarm, in engineering units
- Gt – high limit measurement notification, like a high alarm, in engineering units
- Step – Minimum delta change required to notify, in engineering units

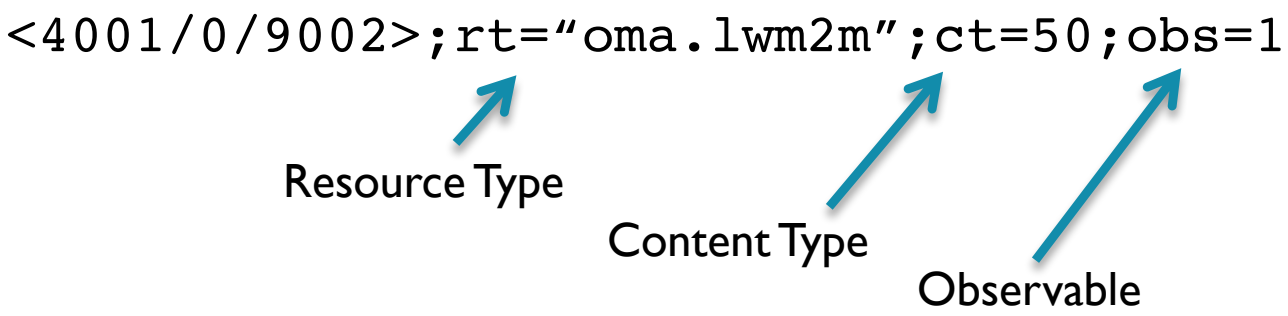
LWM2M Bulk Read

- Returns TLV or JSON based on requested content-format
- CBOR needs to be added
- Linked Objects are supported


```
{“e”:[
  {"n":"0","sv":"Open Mobile Alliance"},
  {"n":"1","sv":"Lightweight M2M Client"},
  {"n":"2","sv":"345000123"},
{"n":"3","sv":"1.0"},
{"n":"6/0","v":"1"},
  {"n":"6/1","v":"5"},
  {"n":"7/0","v":"3800"},
  {"n":"7/1","v":"5000"},
  {"n":"8/0","v":"125"},
  {"n":"8/1","v":"900"},
  {"n":"9","v":"100"},
  {"n":"10","v":"15"},
  {"n":"11/0","v":"0"},
  {"n":"13","v":"1367491215"},
  {"n":"14","sv":"+02:00"},
{"n":"15","sv":"U"}]
}
```


ARM

LWM2M Discovery Returns RFC 6690 Links



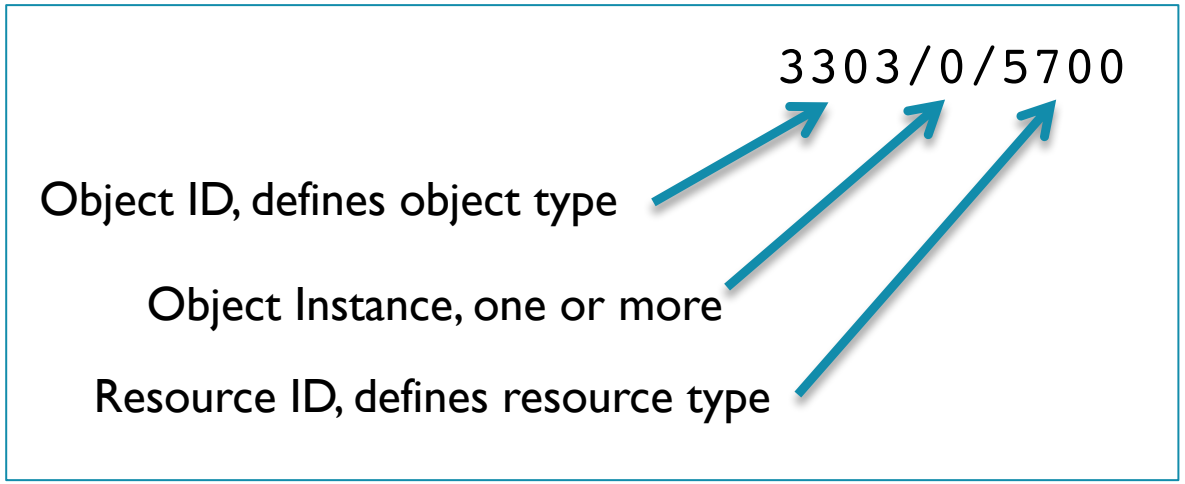
- Links are uploaded during registration to inform the LWM2M server about resources on the endpoint
- Links are discovered using GET with content type "application/link-format"
- JSON representation using content type "application/link-format+json"

ARM

IPSO Smart Objects

Smart Objects Use the LWM2M Object Model

- REST API with a URI template
  - Objects
  - Object Instances
  - Resources
  - (Resource Instances)
- Reusable resource and object IDs
  - Common definitions for concepts
  - Map to semantic terms e.g. temperature, currentValue
  - IDs are registered with the OMNA
- Can be embedded in a path hierarchy on the server
  - /home/weather/3303/0/5700



ARM

IPSO Smart Object Example

Object info

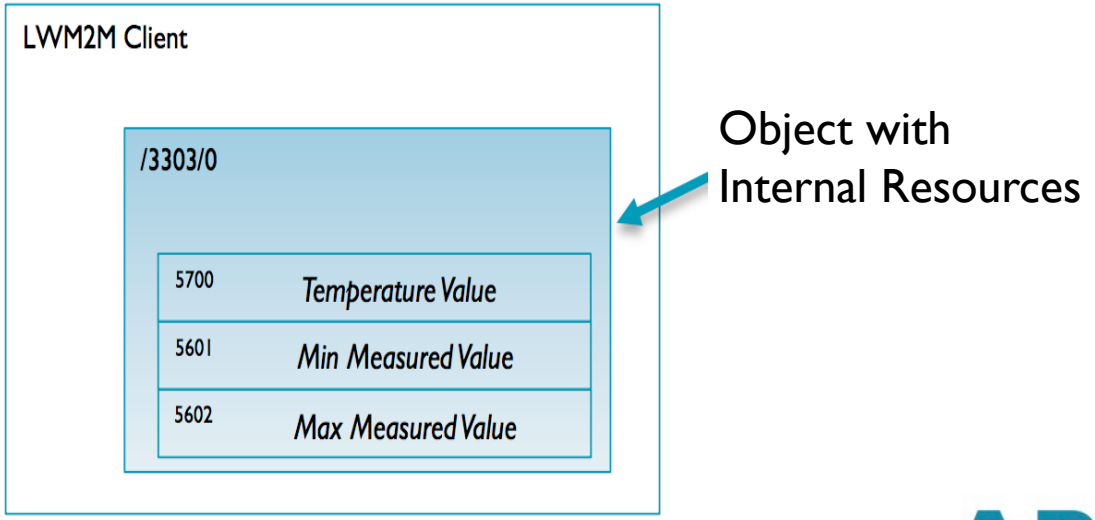
Object	Object ID	Object URN	Multiple Instances?
IPSO Temperature	3303	urn:oma:lwm2m:ext:3303	Yes

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*



ARM

IPSO 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

ARM

Ad-Hoc IPSO Smart Object – Smart Thermostat

Object info:

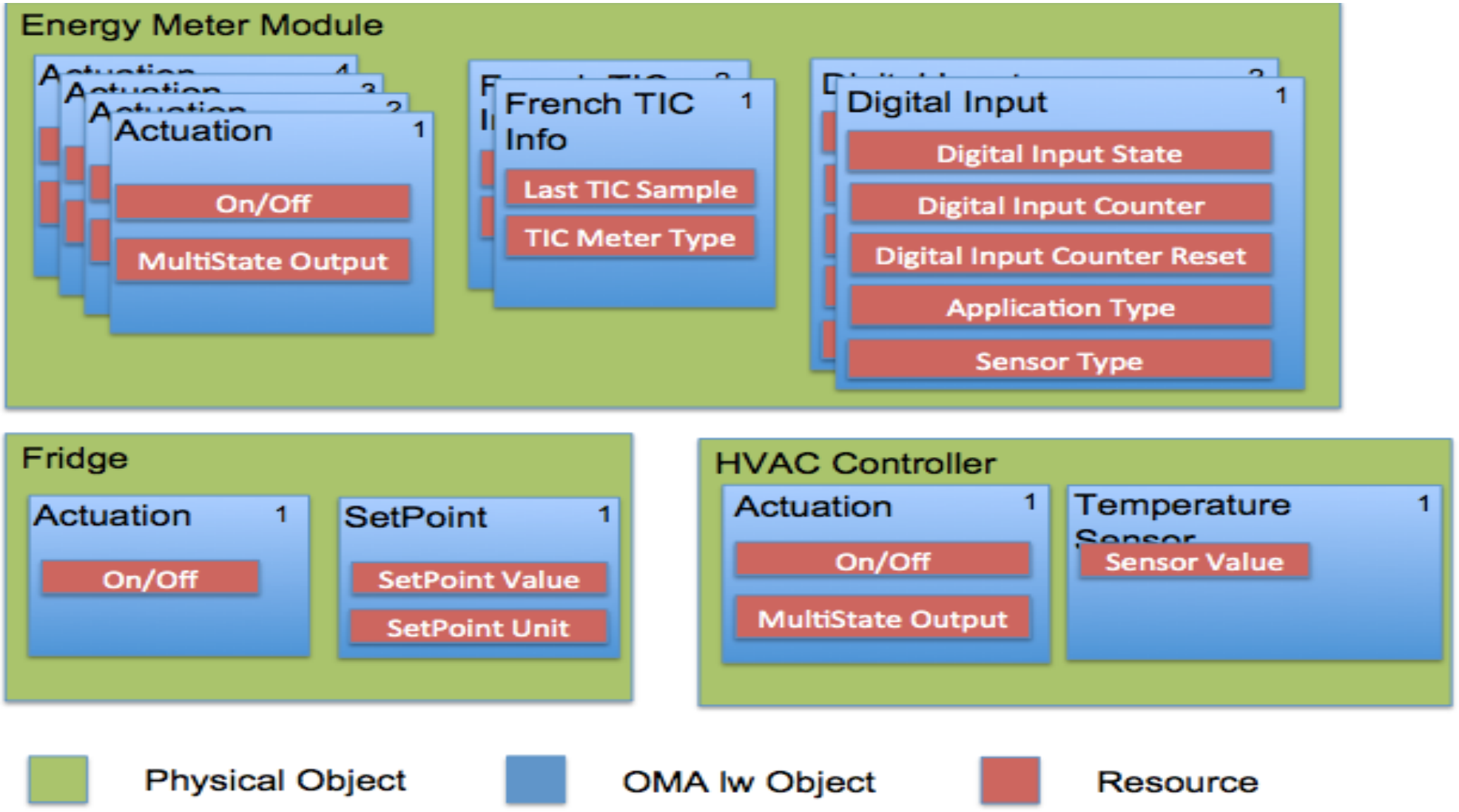
Object	Object ID	Object URN	Multiple Instances?	Description
Smart Thermostat	12300	urn:oma:lwm2m:x:12300	Yes	Smart Thermostat with multiple settings

Resource Info:

Resource Name	Resource ID	Access Type	Multiple Instances ?	Mandatory	Type	Range or Enumeration	Units	Descriptions
Sensor Value	5700	R	No	Mandatory	Float		Per Units resource	Temperature measurement
Units	5500	R,W	No	Mandatory	String	ucum:degF, ucum:degC		Units for 5700
Application Type	5750	R,W	No	Optional	String			Name, e.g. "Hall Thermostat"
Cooling	5200	R	No	Optional	Boolean			1=cooling
Heating	5201	R	No	Optional	Boolean			1=heating
Heat Source	5203	R	No	Optional	String	"Emergency", "Normal"		Indicates heat source
Fan Timer Active	5204	R,W	No	Optional	Boolean			1=running
Fan Timeout	5205	R,W	No	Optional	String		UTS	Time for fan to stop
Energy Save Mode	5206	R,W	No	Optional	Boolean			1= Energy Save mode
Away Mode	5207	R,W	No	Optional	Boolean			0=Home, 1=Away
Setpoint	5208	R	No	Optional	Float			Desired Temperature
HVAC Mode	5209	R,W	No	Optional	String	"Heat", "Cool", "Heat-Cool"		System Mode
High Setpoint	5210	R,W	No	Optional	Float			Highest desired temperature
Low Setpoint	5211	R,W	No	Optional	Float			Lowest desired temperature
High Away Setpoint	5212	R,W	No	Optional	Float			Highest away mode temperature
Low Away Setpoint	5213	R,W	No	Optional	Float			Lowest away mode temperature

ARM

Composite IPSO Smart Objects



ARM

IPSO Smart Object Development

- Smart Objects are Easy to Modify and Customize
  - Based on Consistent Design Patterns and Reusable Resource Definitions
  - Object Sets can be Forked and Modified
  - Expecting Domain-Specific Object Sets to be Created by Collaborative Vertical Working Groups
  - New Object Sets can be Released as new Smart Object Guidelines
  - Objects in Released Smart Object Guidelines are Registered with the OMA, Use Standard OMA DDF (XML) File Format Object Descriptors

IPSO Smart Objects Roadmap Candidates

- More Objects, Domain Specific
  - Gateway Management Objects – TR-069
  - Smart Home
  - HVAC
  - Smart Appliances
  - Mapping and Binding of Smart Objects to Zigbee Application Clusters
  - Mapping and Binding of Smart Objects to Bluetooth Application Profiles
  - Advanced Lighting Objects
  - Behavioral Objects – Timers, Controllers, Rules and Bindings
- Object Model
  - Linked Composite Objects
  - Semantic Annotation and Application Templates

Ad-Hoc IPSO Smart Object – BLE Heart Rate Sensor Profile

Object info:

Object	Object ID	Object URN	Multiple Instances?	Description
Heart Rate	12200	urn:oma:lwm2m:x:12200	Yes	Heart Rate Monitor

Resource Info:

Resource Name	Resource ID	Access Type	Multiple Instances?	Mandatory	Type	Range or Enumeration	Units	Descriptions
Sensor Value	5700	R	No	Mandatory	Float		BPM	Heart Rate Measurement Value
Digital Input State	5500	R	No	Optional	Boolean			Sensor contact status 0=no contact, 1= contact
Total Energy	5950	R	No	Optional	Float		kJ	Energy Expended
Reset Cumulative Energy	5822	E	No	Optional	Opaque			Reset 5950 Energy Expended to zero
Body Sensor Location	5951	R,W	No	Optional	String			Intended sensing location on the body
R-R Interval	5952	R	No	Optional	String			Sequence of R-wave intervals

ARM

Semantic Annotation of Smart Objects

- Object annotation could use RFC 6690 for associating additional semantic descriptions with Smart Objects and Resources
- Can be used to add contextual metadata and dynamic link relations
- Described in IETF CoRE Interfaces document (http://datatracker.ietf.org/doc/draft-ietf-core-interfaces), enables Discovery by relation and attribute
  - `For example, GET /rd-lookup?ep&rt="urn:X-ipso:temperature"`

```
Returns: </sensors/3303/0/5700>;obs;if="urn:X-ipso:sensor";rt="urn:X-ipso:temperature";ct=50;u="ucum:degC"
```

- Refers to qualified, resolvable namespaces and concepts
- Local discovery in .well-known/core or using Resource Directories
- GET/PUT can use semantic query `GET <URL>?rt="urn:X-ipso:temperature"`

ARM

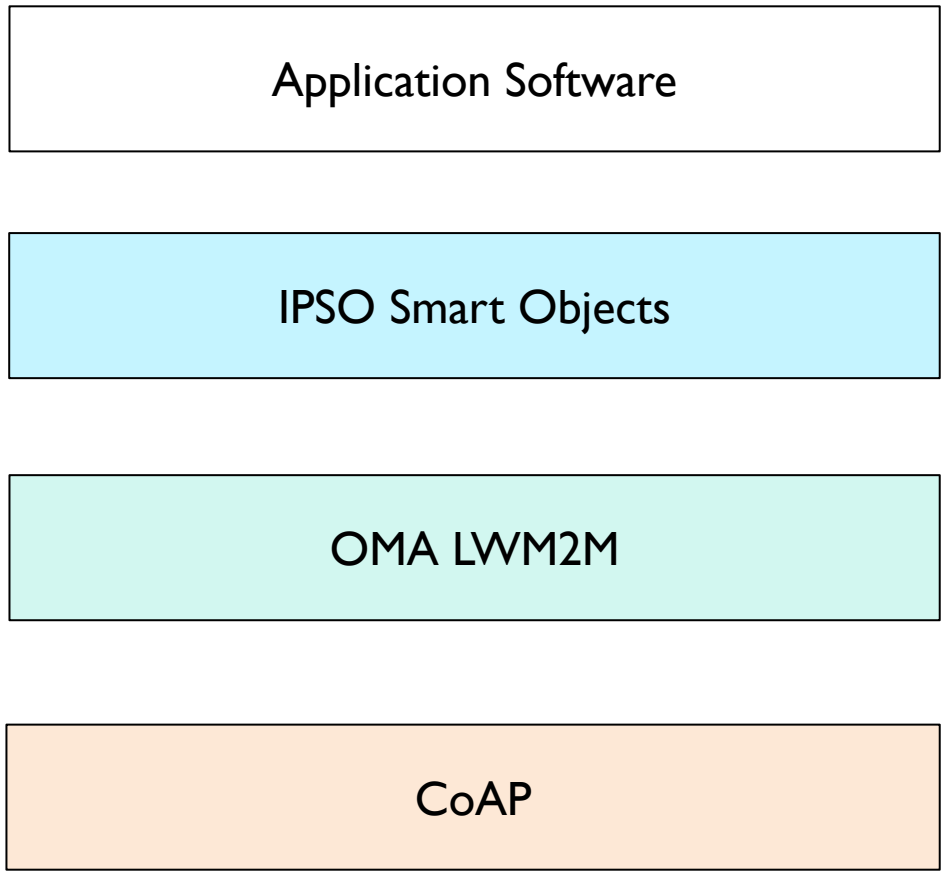
Application Templates

- JSON templates for instance constructor and application schema
- Interface to high level semantic models
- Example template fragment for OMA LWM2M Application
- Can carry Semantic Annotation as link attributes

```
"objects":{
  3303:{
    "description":"ipso temperature sensor",
    "attributes":{"pmin':60,"pmax":300,"max-age":360},
    "link-attributes":{"rt":["oma.lwm2m","urn:X-ipso:temperature"]},
    "instances":{
      0:{
        "attributes":{},
        "link-attributes":{"rt":"urn:oma:lwm2m:ext:3303"},
        "resources":{
          5700:{
            "description":"Current Measured Value"
            "attributes":{"pmin":10,"step":0.5},
            "link-attributes":{"rt":"ucum:temperature","obs","ct":50}
          },
          5701:{
            "description":"units",
            "value":"ucum:Cel",
            "operations":["r"]
          },
          5601:{"description":"Min Measured Value","value":100},
          5602:{"description":"Max Measured Value","value":0},
          5603:{"description":"Min Range Value","value":0},
          5604:{"description":"Max Range Value","value":100},
          5605:{"description":"Reset Min/Max"}
        }
```


ARM

Summary



Not tied to specific device or protocol
Any Programming Language
Runs on devices, gateways, and services

Application Level Interoperability
Reusable Device to Application API
Not tied to any specific protocol

Service Layer Specification
Device Management over CoAP
Object Model for DM and Applications

REST protocol for constrained devices
IETF Standard (RFC 7252)
Uses TCP or UDP, any IP connection
Discovery using IP Multicast or Directory

ARM

References

IPSO Smart Object Guideline

http://www.ipso-alliance.org/smart-object-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/

ARM