# Web to the Edge

REST and Hypermedia Design for Machine APIs

# Michael Koster – Current Work

- ARM
  - IoT Architecture and Standards
  - Application Level Interoperability
  - Developer Enablement
- IETF
  - Resource Directory
  - CoRE Interfaces
  - CoAP Pub-Sub
  - T2TRG
- OMA LWM2M
  - Peer-to-peer, firewall traversal, and M2M interface
- IPSO
  - Smart Object Committee, Data Model and Schema

# IoT Architecture – Problems to Solve

- Interoperability
  - In the way in which software interacts with physical resources
  - Decouple IoT devices from the software that manages them
  - Discovery, Management and Reporting, Security, Authorization
- Scalability
  - Large number of devices, users, interactions, connections
  - Scale-less interaction
- Technology Reuse and Modularity
  - Software, networks, protocols, data models
  - Across vendors in a vertical application segment
  - Across diverse vertical application segments
- Low Barrier to Innovation
  - Anyone can participate and innovate

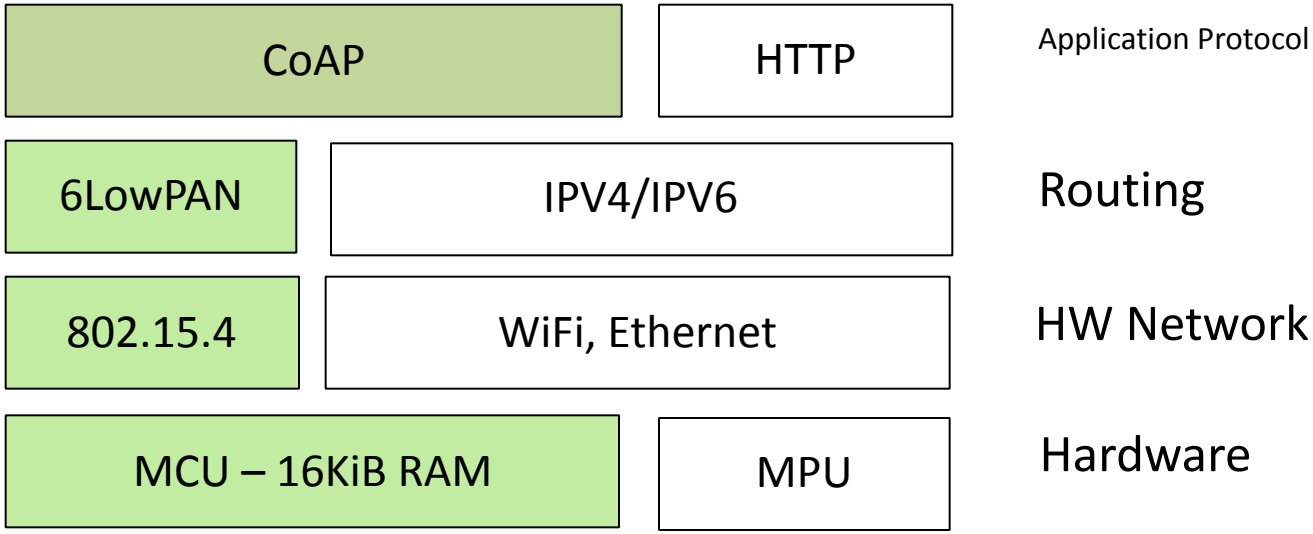
# Internet/WWW Design Patterns

- Narrow Waist, endpoint oriented
  - Innovation happens at the endpoints, enabled by common, openly available network protocols (the narrow waist)
- Layered Protocols
  - Common set of IP protocols (TCP, UDP) abstract the lower communication layers
  - Common Application Protocols (HTTP, REST) abstract resources
- Uniform Addressing
  - URIs and Hyperlinks point to resources
  - IP Addresses, DNS names are globally unique
- Stateless Interaction
  - Client-Server pattern
  - Hypermedia As The Engine Of Application State

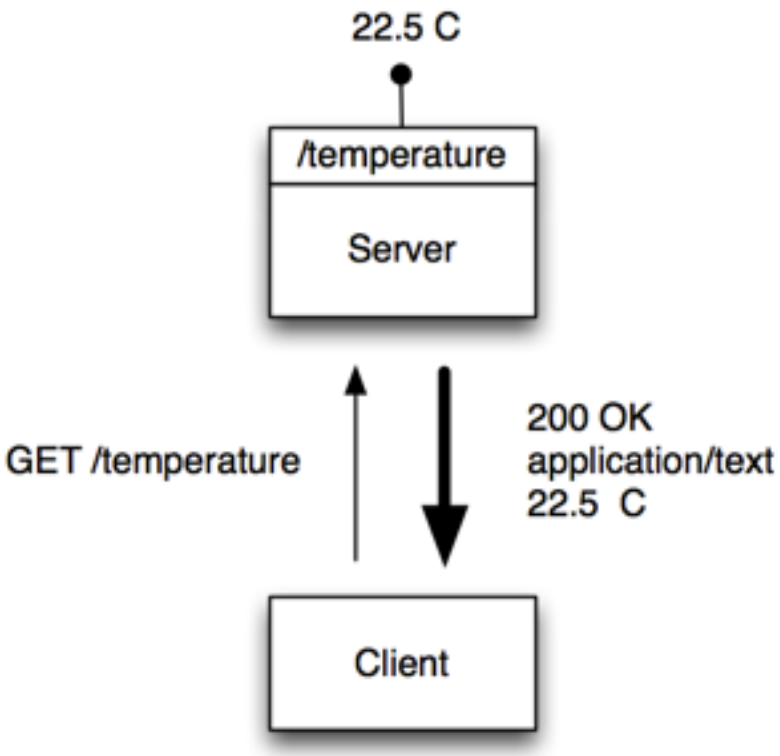
# How Does it Apply to IoT?

- Internet Protocol (IP) on Constrained Devices
- Machine to Machine (M2M) Application Protocols
- Standard Object Models and Data Models
- Hypermedia for Machine APIs

# IP for Constrained Environments



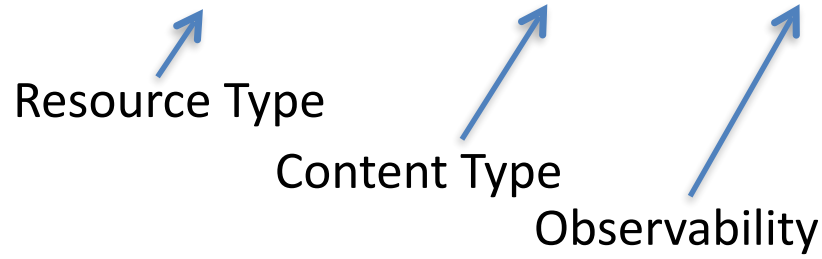
# CoAP Protocol




- Makes each device a lightweight origin server that exposes a REST API
- A CoAP endpoint may also contain application client logic
- A CoAP endpoint can be both client and server
- Peer to Peer interaction is based on a duplex client-server pattern
- Also supports asynchronous notification, Pub-Sub pattern

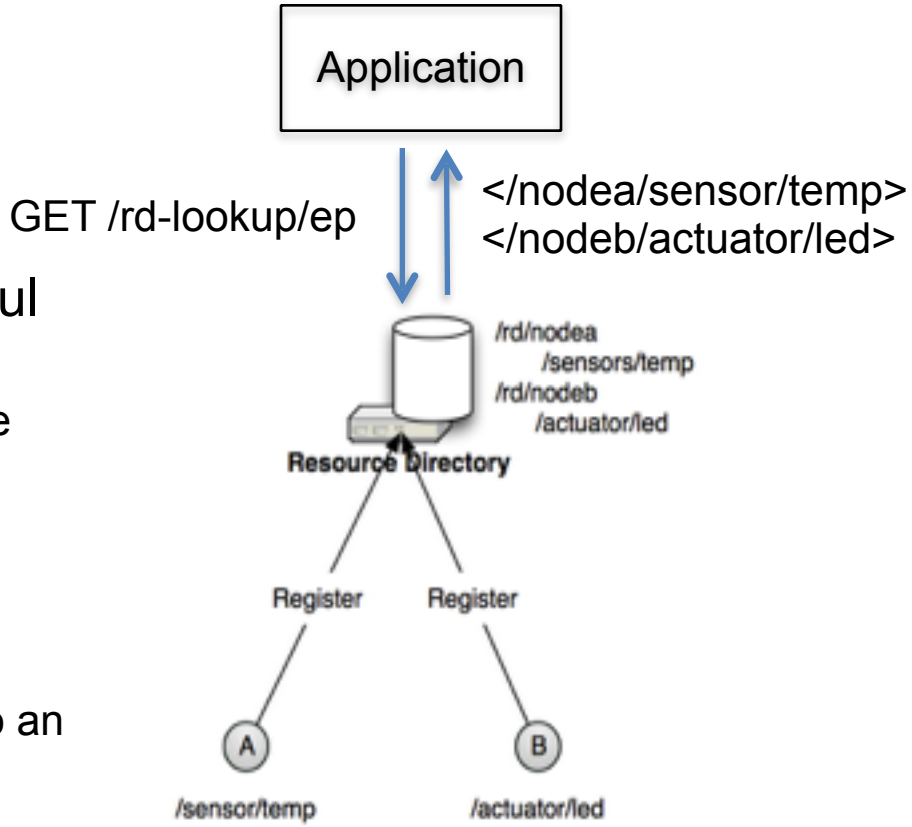
# RFC 6690 CoRE Link-Format

```
<4001/0/9002>;rt="oma.lwm2m";ct=50;obs=1
```



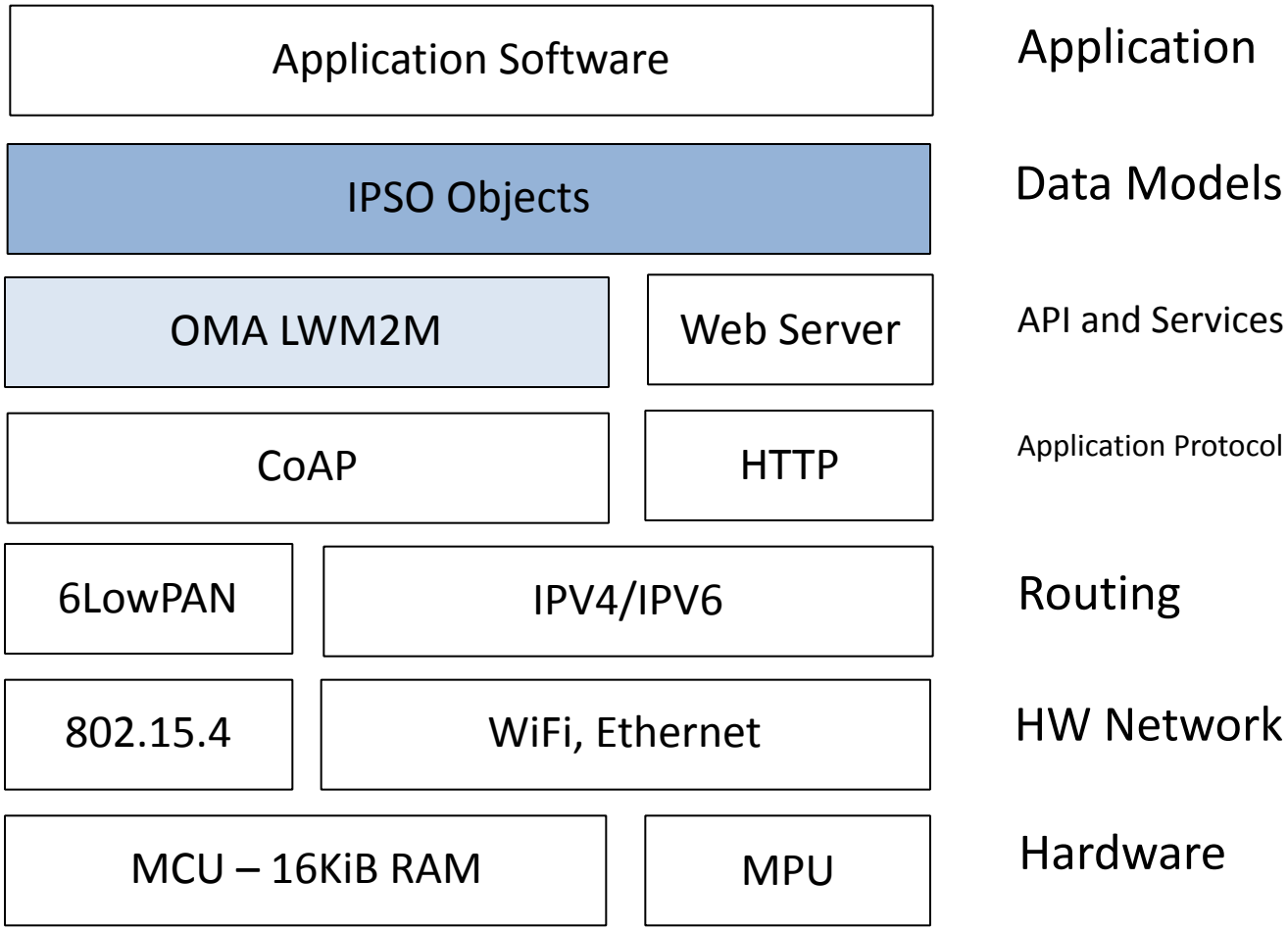
# Resource Discovery

- 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

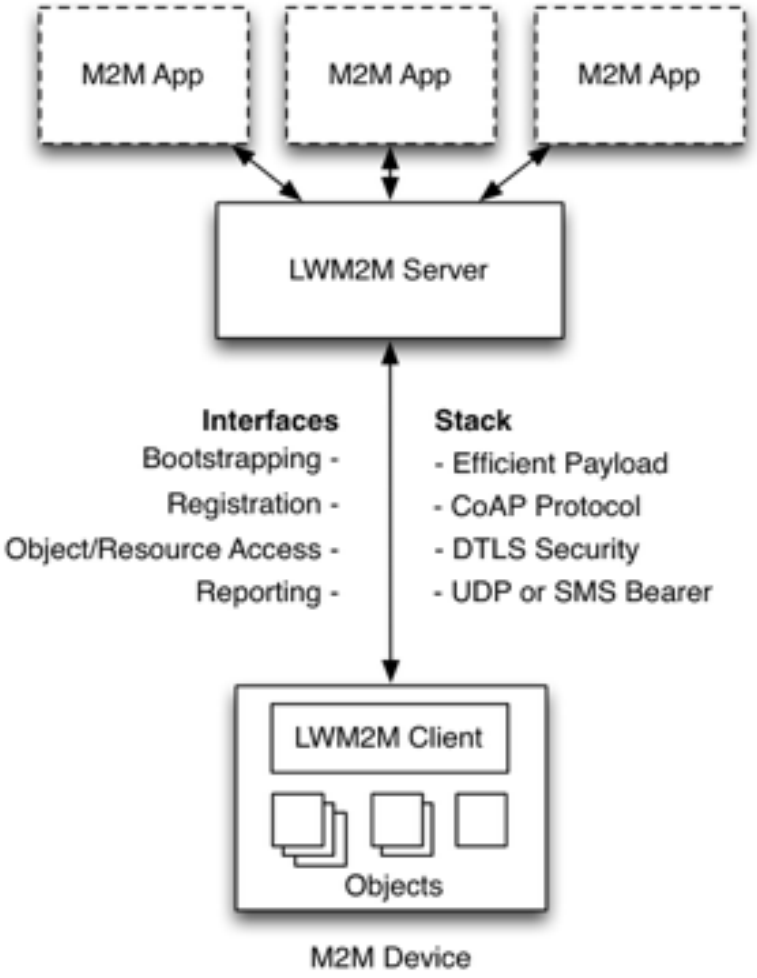



See draft-ietf-core-resource-directory

# Device Management and Application Support



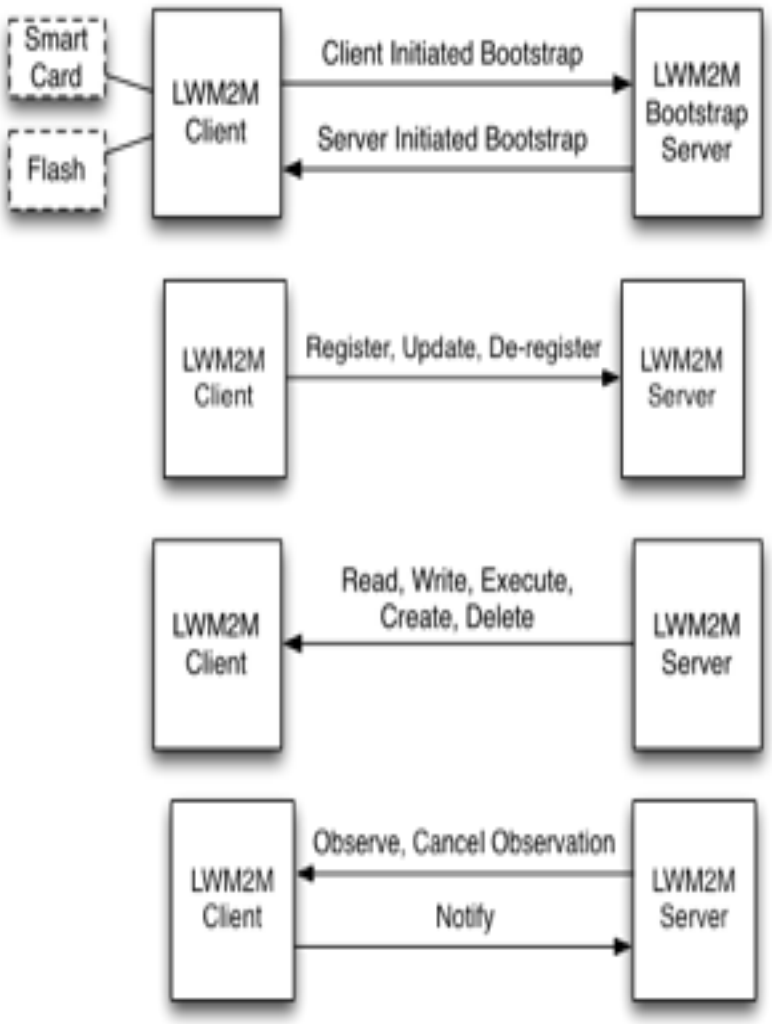
# OMA LWM2M Reference Architecture




- M2M Applications
  - Application abstraction through REST API
  - Resource Discovery and Linking
- LWM2M Server
  - CoAP Protocol persistent endpoint
  - Supports 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

# OMA 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 and Application Interface Using Objects
  - Device Management Objects and Resources
  - CoAP REST API
- Reporting Interface
  - Object Instances and Resources Report
  - Asynchronous notification using CoAP Observe



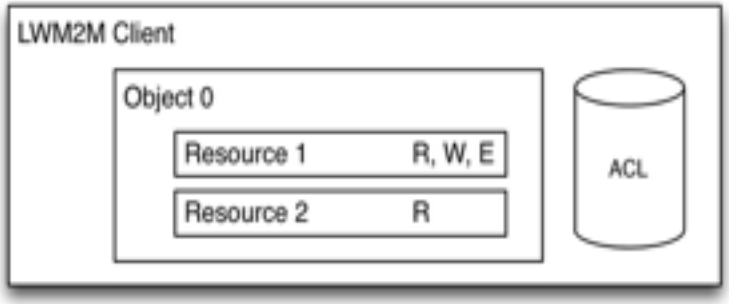
# OMA LWM2M Object Model

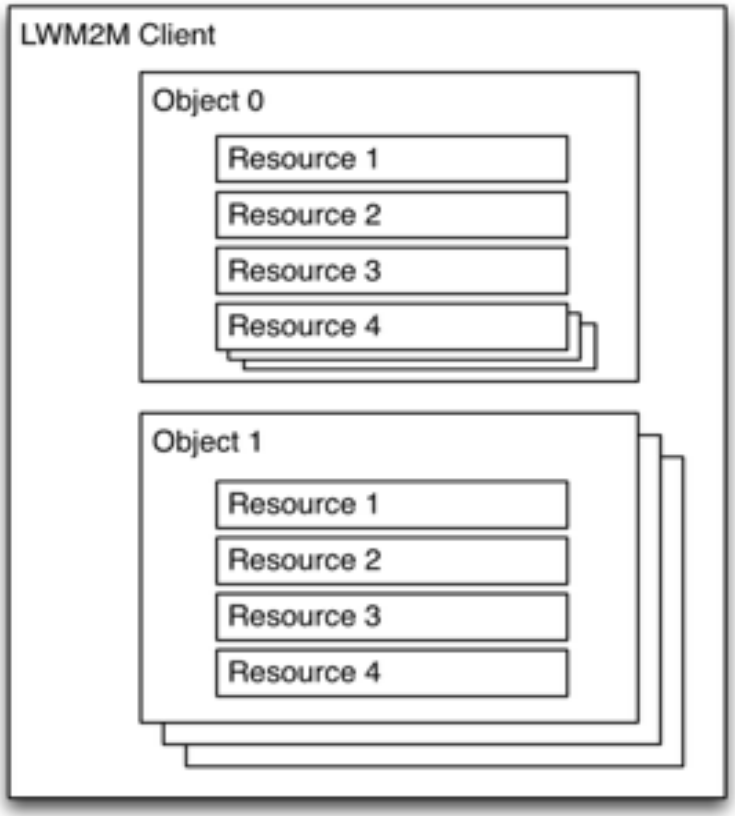
- A Client has one or more Object Instances
- An Object is a collection of Resources
- A Resource is an atomic piece of information that can be Read, Written or Executed
- Objects can have multiple instances
- Objects and Resources are identified by 16-bit Integers
- Objects/Resources are accessed with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

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

Example: /3/0/1

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

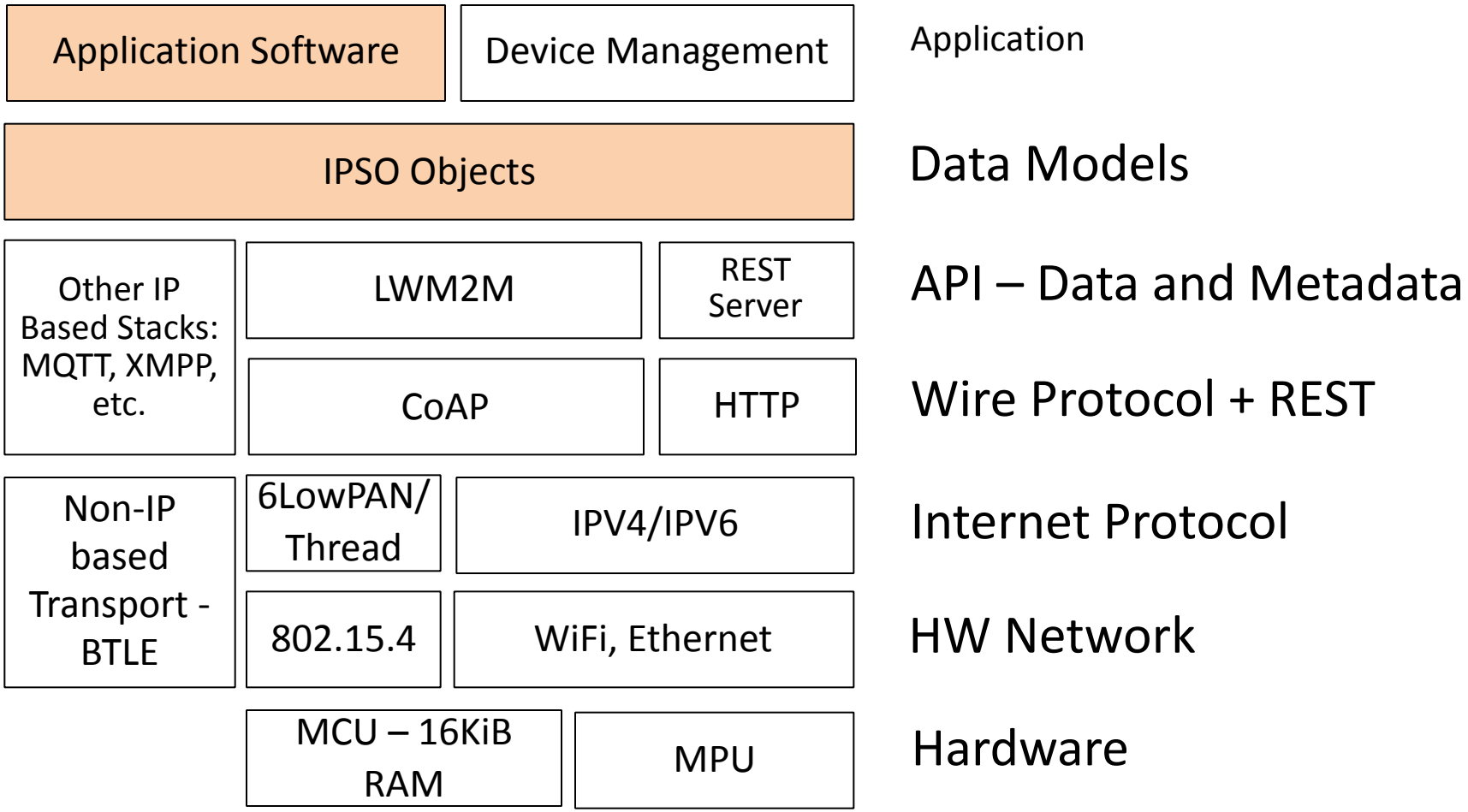


# IPSO Smart Objects

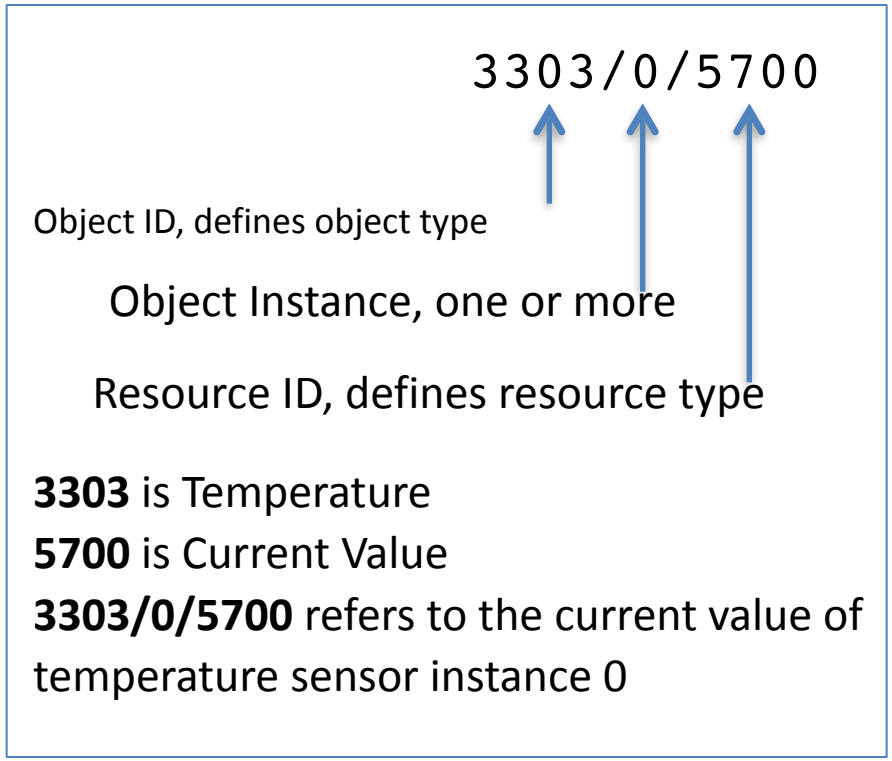
- Plug and play data models between applications and data sources
- Create a set of reusable objects and resources that represent single points of interest and fundamental concepts
  - Temperature Sensor, On-Off Control Switch
  - Current Value, Max Value, Min Value
- Make Objects and Resources Reusable across application domains
- Build up models of complex things by composing simple objects
  - For example, an electric motor with temperature sensors, rotation speed, power supply current, voltage, frequency, vibration sensors

# IPSO Smart Objects Support Modular Internet Protocols



# IPSO Smart Objects are based on LWM2M Data Model

- REST API with a URI template
  - 2 level Class Hierarchy and schema
  - Objects, one or more instances
  - Resources
- Reusable resource and object IDs
  - Object ID determines gross concept and object structure
  - Resource ID determines what aspect of the object is represented
  - Semantic Consistency e.g. Object=Temperature, Resource=Current Value
  - IDs are registered with the OMNA
- Data Types and Operations
  - String, Decimal, Boolean, Time, ObjLink
  - Read, Write, Create, Delete, Execute, Discover, Observe, Notify



# IPSO Smart Object Example

**Object Info:**

| Object | Object ID | Object URN | Multiple Instances? | Description |
|---|---|---|---|---|
| **IPSO Temperature** | 3303 | urn:oma:lwm2m:ext:3303 | Yes | Temperature sensor, example units = Cel |

**Resources:**

| Resource Name | Resource ID | Access Type | Multiple Instances? | Mandatory | Type | Range or Enumeration | Units | Descriptions |
|---|---|---|---|---|---|---|---|---|
| **Sensor Value** | 5700 | R | No | Mandatory | Float | | | Last or Current Measured Value from the Sensor |
| **Units** | 5701 | R | No | Optional | String | | | Measurement Units Definition e.g. "Cel" for Temperature in Celsius. |
| **Min Measured Value** | 5601 | R | No | Optional | Float | Same as Measured Value | Same as Measured Value | The minimum value measured by the sensor since power ON or reset |
| **Max Measured Value** | 5602 | R | No | Optional | Float | Same as Measured Value | Same as Measured Value | The maximum value measured by the sensor since power ON or reset |

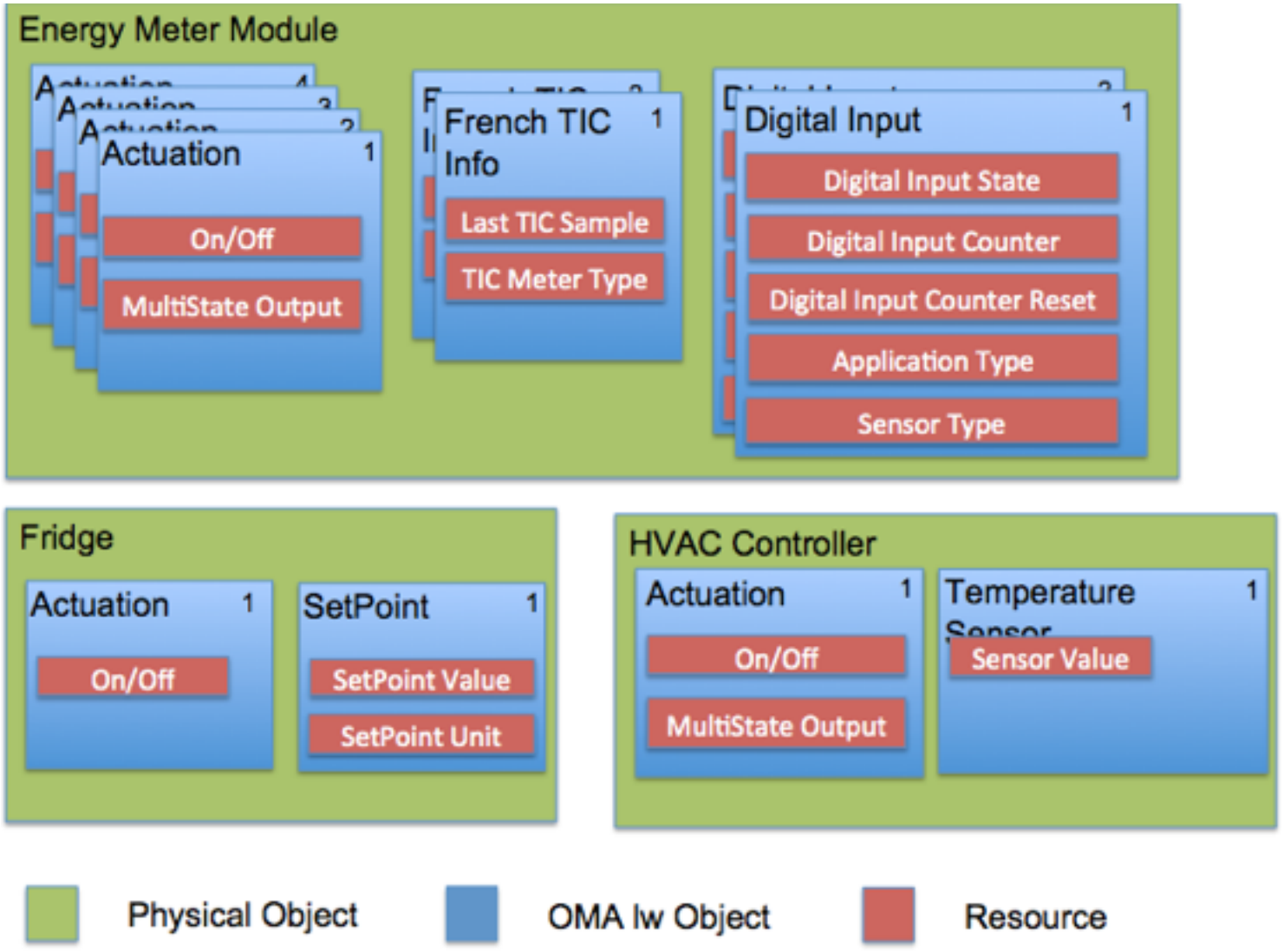
# Semantic Annotation of Smart Objects

- Object annotation uses RFC 6690 links for associating additional semantic descriptions with Smart Objects and Resources
- Can be used to add contextual metadata and dynamic link relations
- Semantic connectors to web scale common data models
- Discovery by relation and attribute
  - For example, using CoRE Resource Directory:

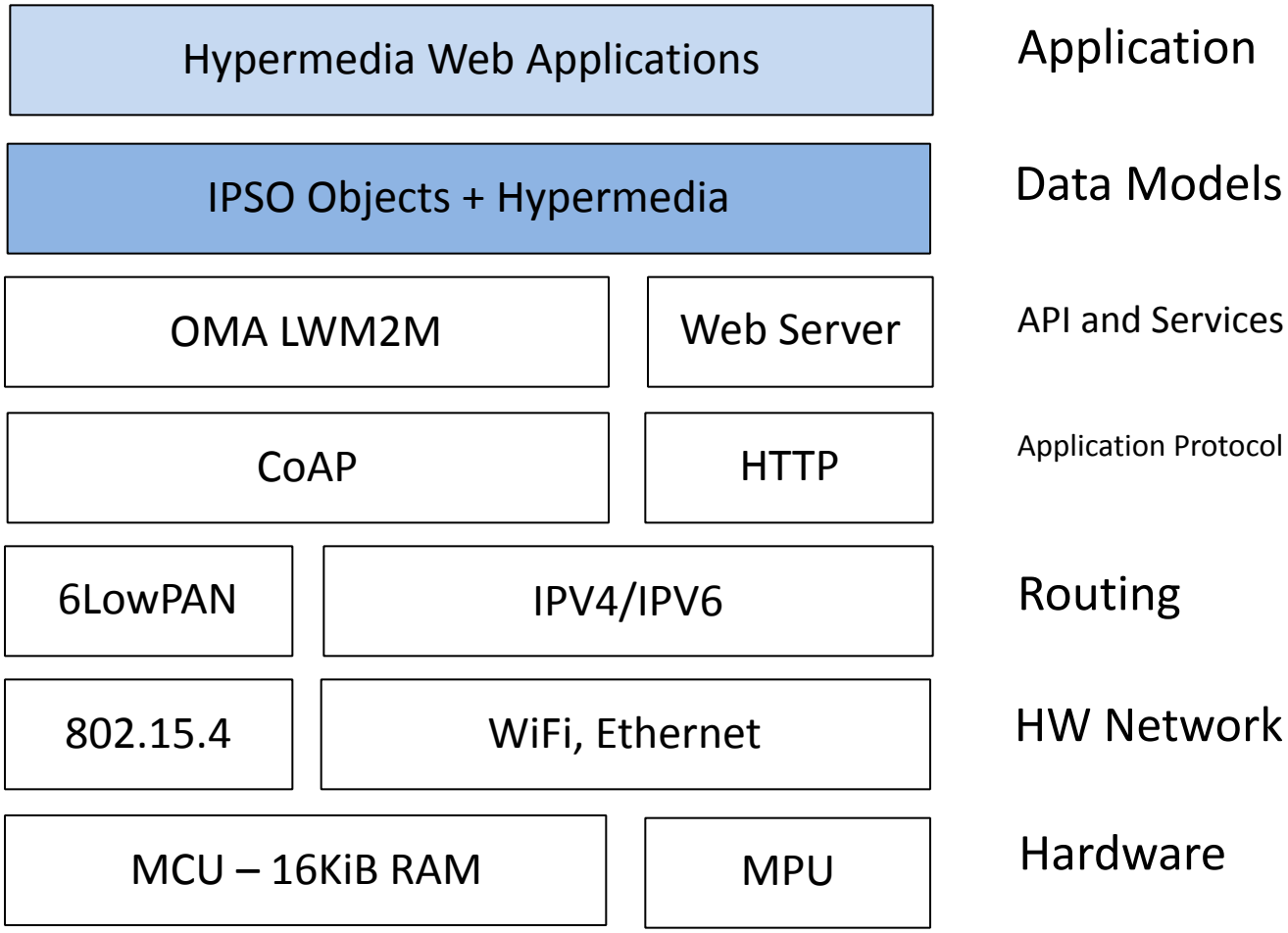
```
GET /rd-lookup?ep&rt="urn:X-ipso:temperature"

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

# Composite IPSO Smart Objects



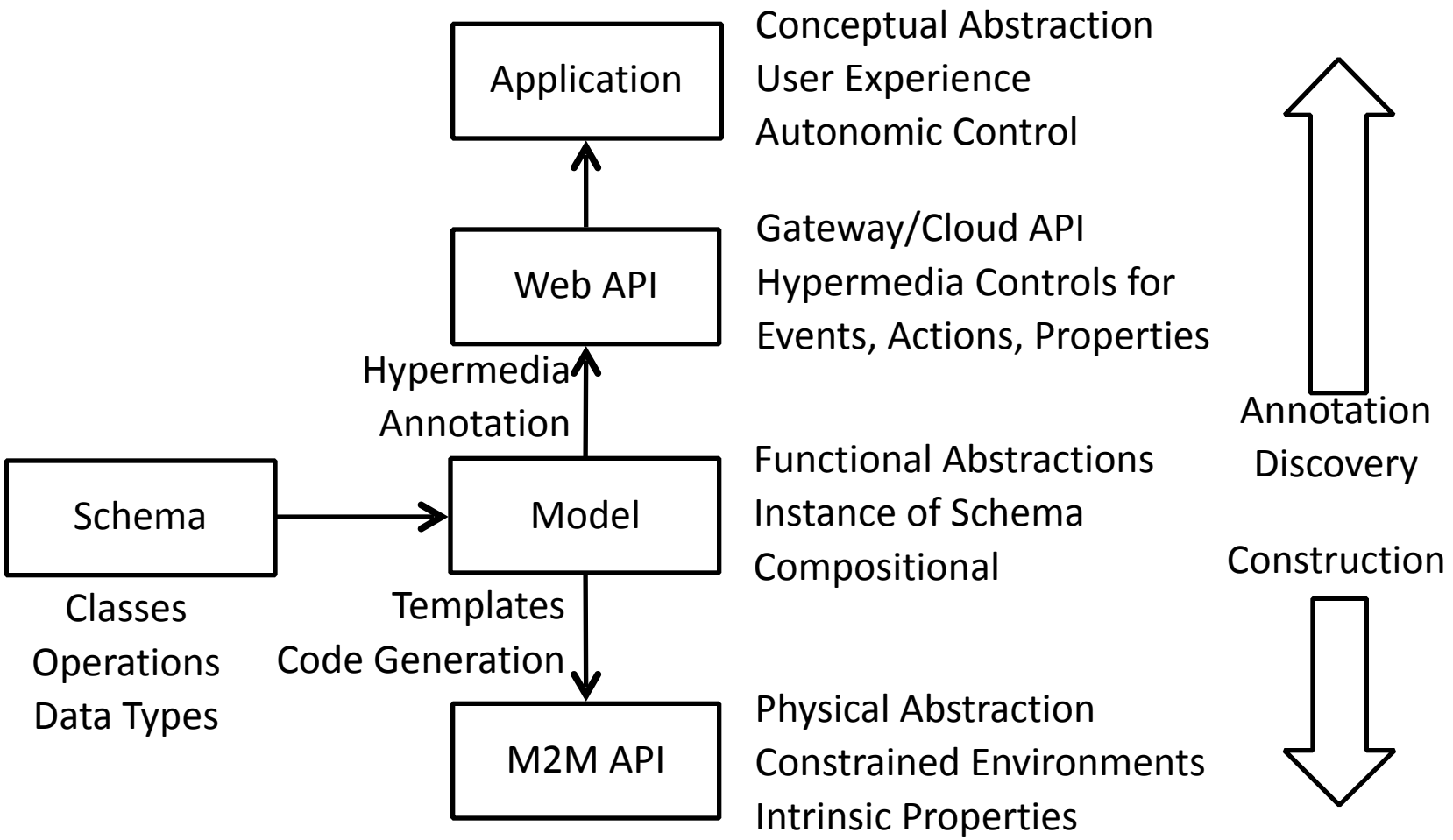
# Web of Things



# Hypermedia

- API State Machine Automation
  - Enables **Common Web Client** to interact with any device or virtual entity, regardless of schema or model structure
  - Hypermedia controls work like HTTP links and forms, but for functional abstractions of physical world items like on/off light switch controls
- Application Semantics and Discovery
  - Describes intrinsic resources and controls at the application level using physical concepts like temperature, light intensity
  - Enables applications to discover the proper resources to link to
- Context Integration
  - Contextual information like installed location, what is being measured and controlled, also used in application resource discovery
- Using Web Linking, Relations, and Attributes

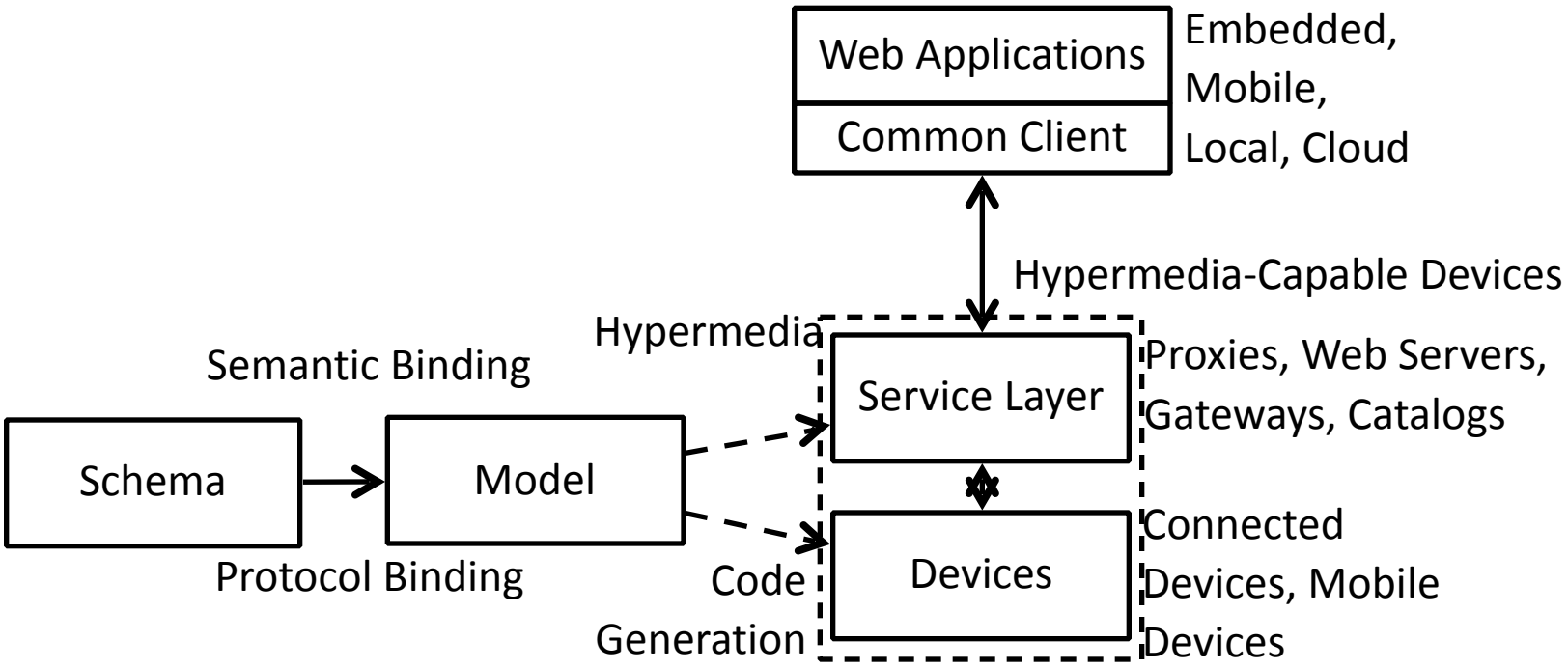
# Model-Based Information Architecture



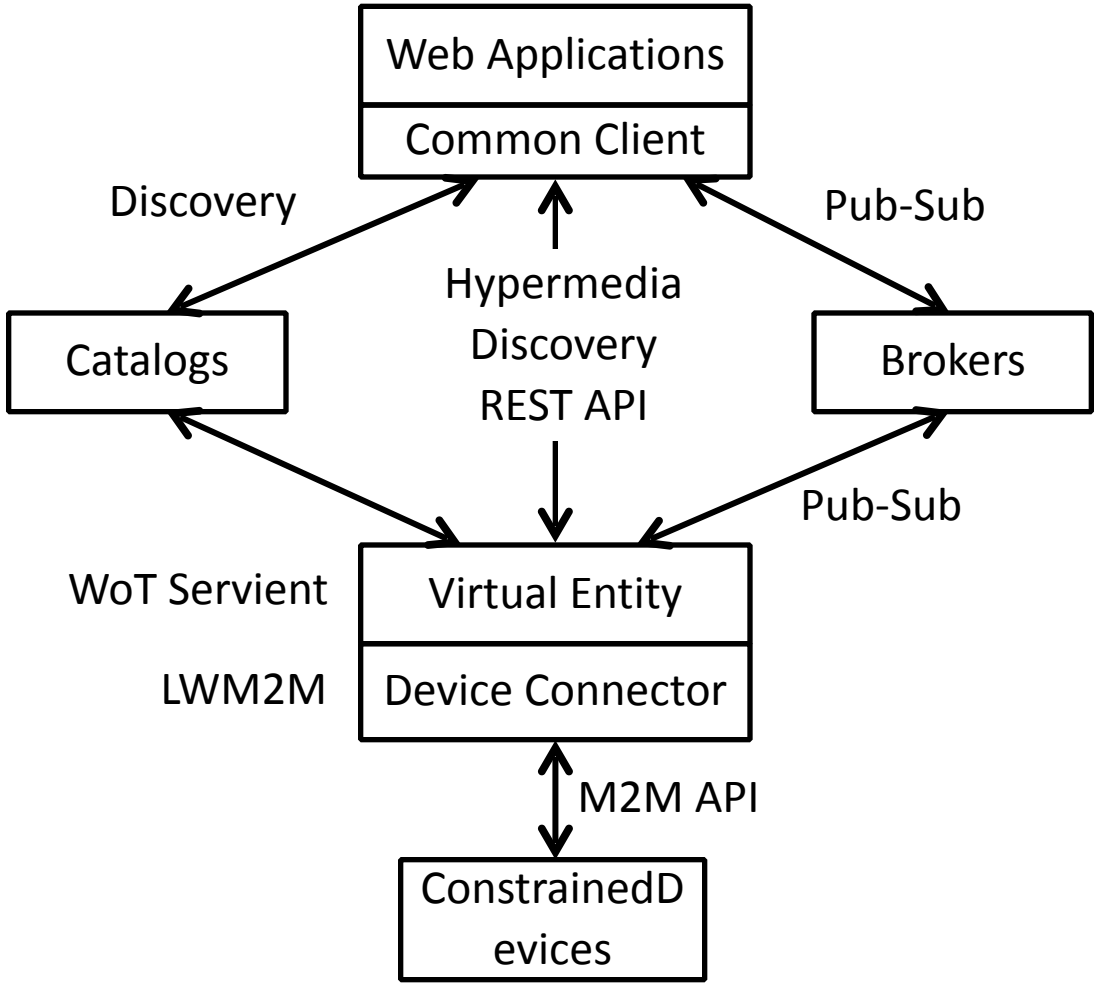
# Models

- Instance(s) of Schema(s) with annotation
- Enable composition of complex entities from simple ones
- Connect functional abstractions to concrete APIs
- Constructors for hypermedia controls
- Contain relations and attributes
  - Provide a way to describe entities using semantic annotation
  - Provide a structured way to introduce contextual annotation
- Can function as instance constructors for constrained devices, e.g. server code generation
- Also provide for managed APIs and application code generation

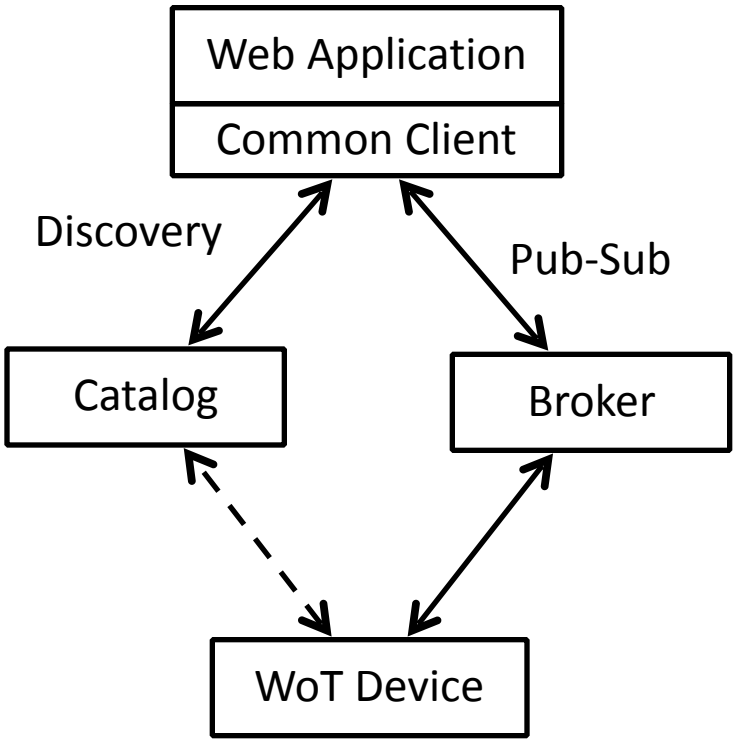
# Deployment From Models

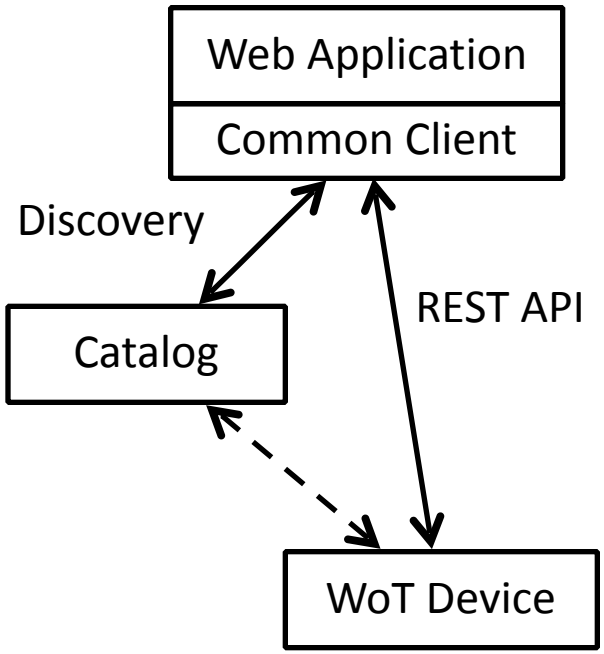


# Deployment with LWM2M

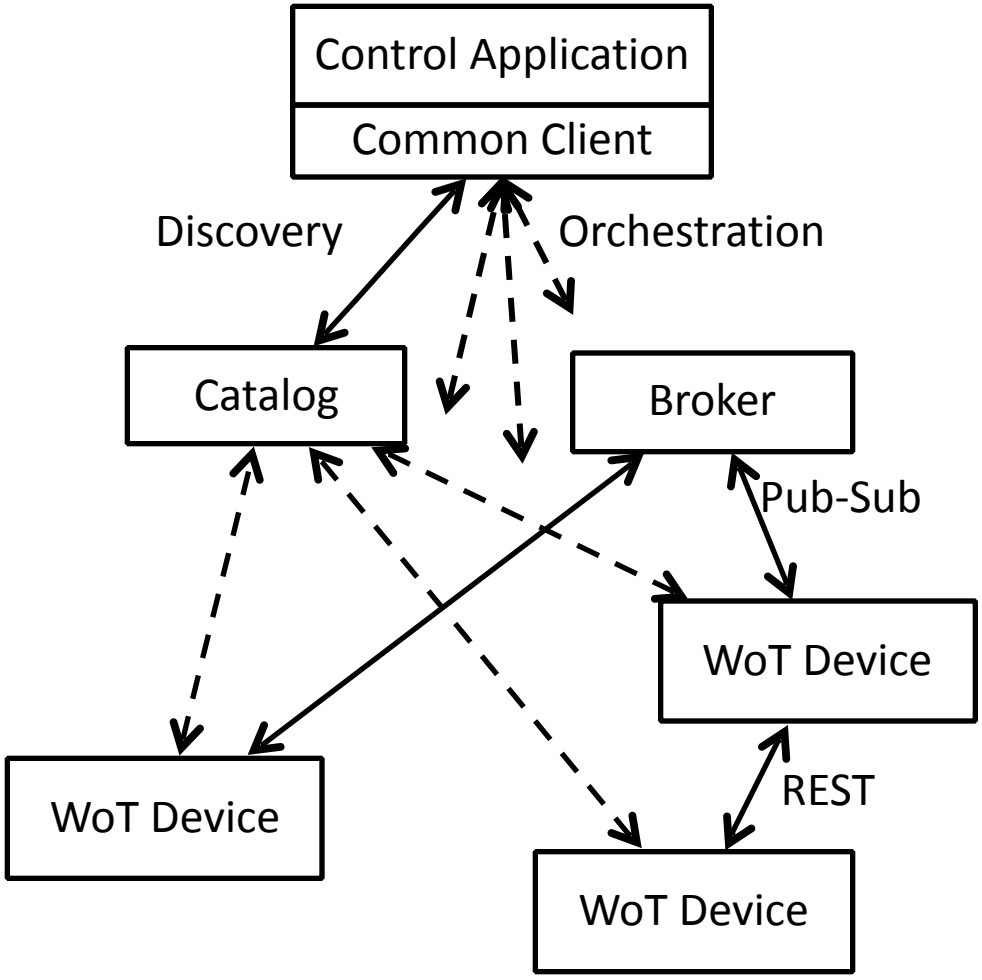


# Example Deployment



# Example Deployment



# Follow on work

- What do the hypermedia descriptions of events, actions, and properties look like?
- What are the functional abstractions that will be used to enable vertical specialization while still providing broad interoperability?
- How does discovery integrate with API automation?
- What are the metadata constructs for connecting constrained hypermedia to higher level abstractions?
- Prototype using constrained lighting objects with LWM2M and IoT Toolkit open source python framework
- Reference WoT client using Node-RED + extensions

# Modeling Tools and Frameworks

- RESTful API Modeling Language (RAML) uses JSON Schema
- JSON Schema, JSON Hyper-schema
- Web API Description Language (WADL)
- XML Schema Definition language (XSD)
- URI Templates (RFC 6570)
- IETF CoRE Interfaces Function Set
- Schema.org – Vocabulary for RDFa, JSON-LD
- RESTdesc.org – uses Turtle (RDF based)
- Hypertext Application Language (HAL)
- Eclipse Vorto
- Hydra
- ALPS

# Organizations and Initiatives

- W3C – WoT TF-TD, TF-DI
- OMA - GotAPI
- IETF – CoRE RD, Interfaces, link-format
- IPSO – Smart Object Guidelines
- IEEE – P2413
- OIC – SHP
- Allseen Alliance, Alljoyn
- Eclipse Foundation – Vorto, IoT Backend
- XMPP – XEP-323, XEP-325
- Hypercat – Hypercat 3.0
- OneM2M – MAS
- Zigbee Alliance – ZCL-over-IP
- HGI - SDT
- IIC – Use Cases and Testbeds
- UPnP – Common Data Model
- Cable Labs – Common Data Model

# 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

CoAP Community Site

http://coap.technology/