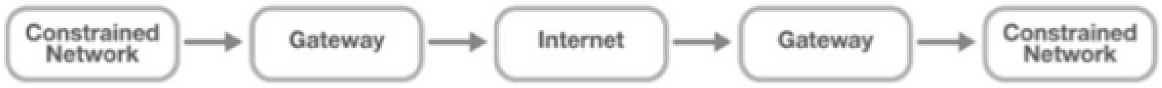
RESTful interfaces for Internet of Things gateways

Michael J. Koster
October 7th, 2012

Previous posts in this series have introduced the idea of standard data models for the Internet of Things and proposed the Smart Object API as one such data model. The Smart Object API provides a RESTful web object encapsulation of self describing resources including Observable Properties, Software Agents, and publish-on-change Subscriptions, also known as Observers.

In the last post, I showed a channel model for the Internet of Things which consists of constrained networks connecting to the Internet through gateways.

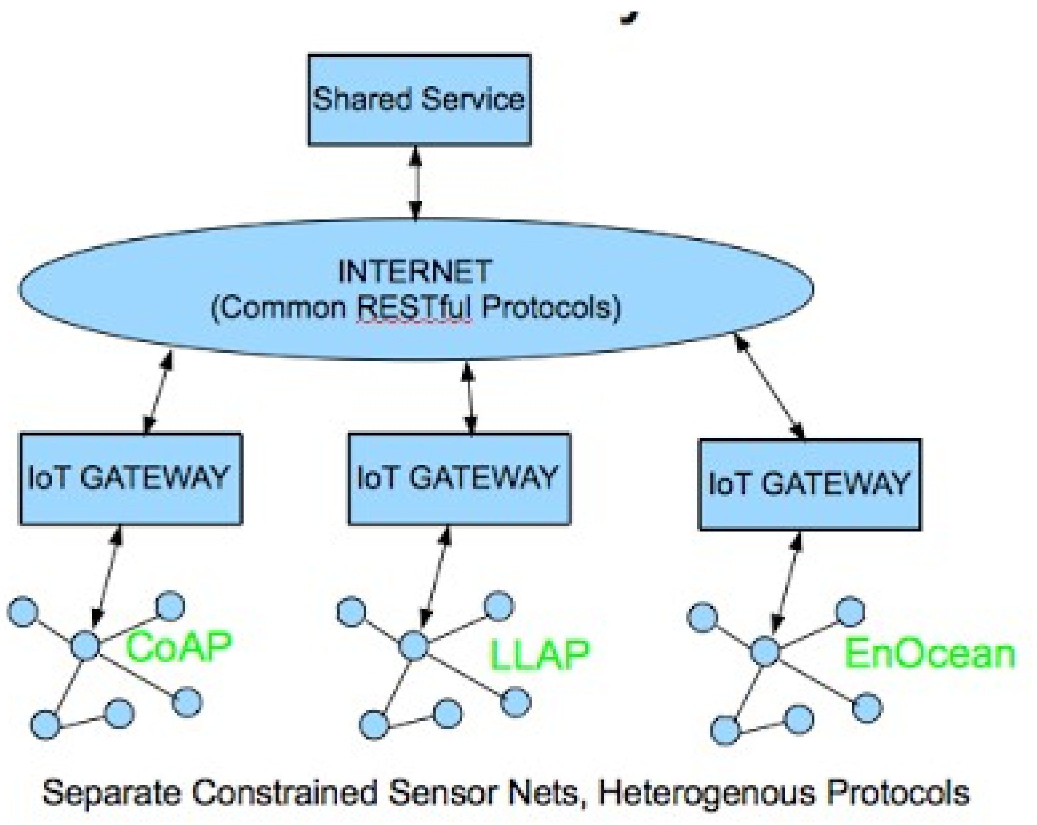


In this post I'd like to explore the IoT gateway, it's architecture, and it's internal resources and operation.

Gateway model for the IoT

In the current IoT deployment, the channel model shown above is very popular. Typical IoT deployments consist of constrained sensor nets connected to cloud services through gateways.

I'd like to broaden the model out a bit, to include a number of, possibly heterogenous, constrained sensor nets connected to the internet, connected to each other through the internet, and connected to shared services through a common protocol using the internet.


Shared Service
INTERNET
(Common RESTful Protocols)
IoT GATEWAY
IoT GATEWAY
IoT GATEWAY
CoAP
LLAP
EnOcean
Separate Constrained Sensor Nets, Heterogenous Protocols

In this model, the separate sensor net protocols are federated, using gateways, to a common protocol based on HTTP and RESTful interfaces. Common resource protocols and data models allow relatively easy sharing and federation of sensor data at a shared service. A single user app can be used to interact on all the different sensor networks.

Proxy Gateway patterns

The gateway operates on the local sensor network *on behalf of* a user device, service, or other sensor network, connected through the Internet or LAN.

The gateway operates on the Internet or LAN on behalf of devices on the sensor network.

Operation on behalf of another entity across a gateway can use some well-known patterns:

Forward Proxy
Proxy which is used by the client to access server resources across the gateway. Client "knows about" the proxy and interacts with proxy resources.

Reverse Proxy
Proxy which looks like a server to the client, i.e. the client interacts directly with the server resources across the gateway.

Interception Proxy
Proxy which transparently maps resources and protocols both directions, i.e. CoAP resources look like http resources from the http side and http resources look like CoAP resources from the CoAP side. This might also be called a *transparent bridge*.

A client-side proxy maps onto the client network, and a server-side proxy maps onto the server network.

A proxy may be caching, able to retain a copy of the return value of the last request which is used to satisfy subsequent requests for the same resource.

Gateway Capabilities

The gateway can add it's own resources as a proxy to provide additional capability to enable sensors to interact on the internet. For example, security and Contextual Semantic Web linkage can be "added" to the sensor by the gateway.

Another example is connecting CoAP sensor nets to the internet. CoAP has some useful capabilities, including semantic description of resources through the /.well_known/core interface (core-link-format) and data push capability. A gateway can provide HTTP emulation of the core-link format and /.well_known/core interface and perform HTTP POST operations to relay CoAP GET responses to a push-capable resource like a Cosm feed.

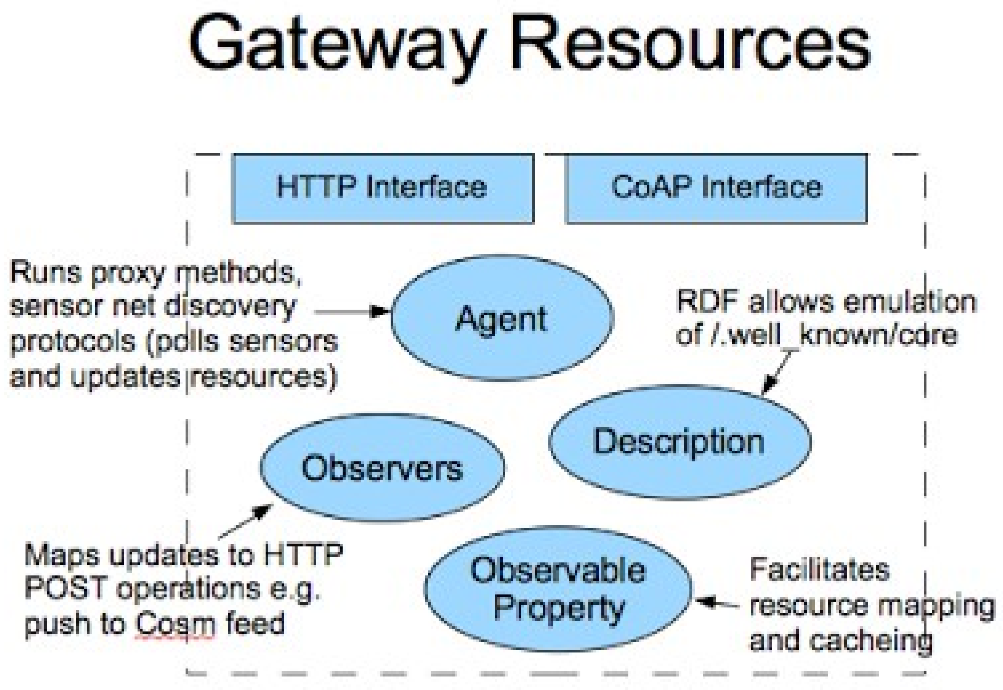
Sensor nets also may have unique discovery and location services that need to be mapped to special resources on the HTTP side or special data types.

Gateway Internal resources

The Gateway contains internal resources used for

- Resource mapping
- Protocol mapping
- Resource caching
- Security
- Semantic discovery and linkage

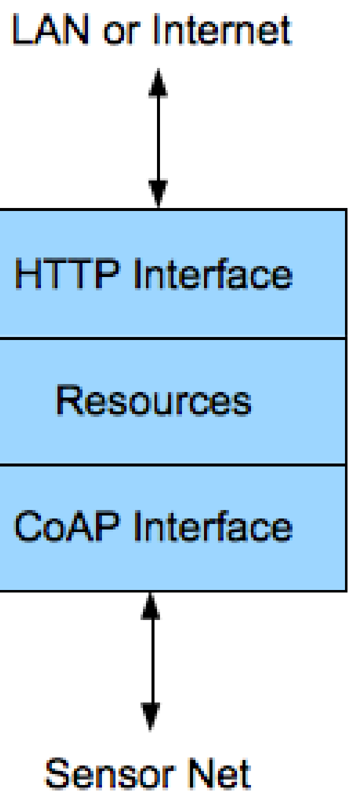
Here is an example of HTTP-to-CoAP gateway resources using the SmartObject API


Gateway Resources
HTTP Interface
CoAP Interface
Runs proxy methods, sensor net discovery protocols (polls sensors and updates resources)
Agent
RDF allows emulation of /.well_known/core
Description
Observers
Maps updates to HTTP POST operations e.g. push to Cosm feed
Observable Property
Facilitates resource mapping and cacheing

The SmartObject API provides resources useful for constructing gateways using any proxy pattern, with or without caching, while adding security and semantic web linkage. For a more thorough description of the SmartObject API resources, see the earlier posts in this blog.

Sensor Net Gateway

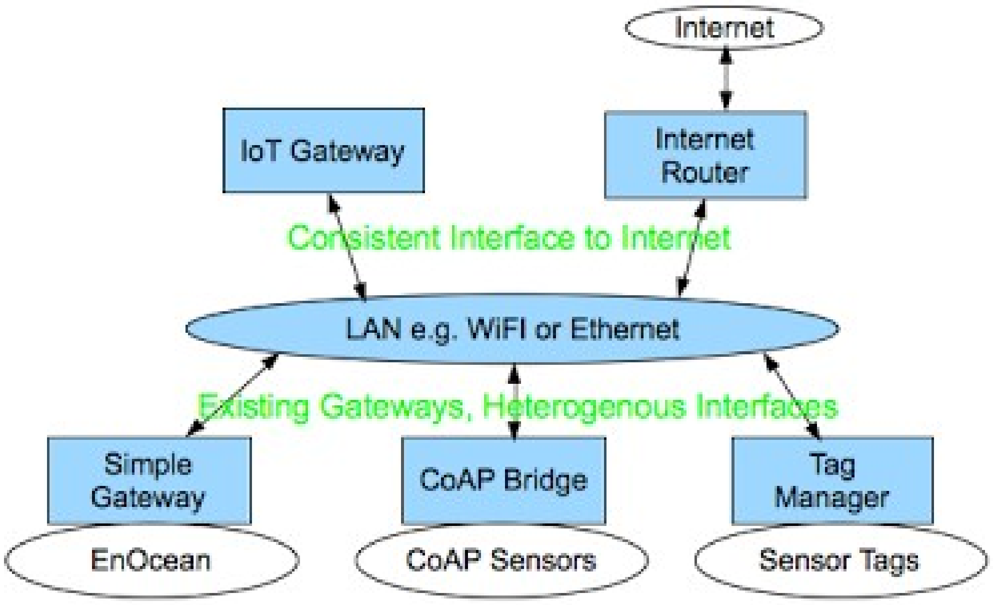
The SmartObject API may be implemented in a sensor net gateway that has multiple hardware interfaces, e.g. WiFi and some profile of 802.15.4 talking to smart sensors running CoAP.



Super Net Gateways

Some sensor nets provide integral gateways, e.g. "tag manager" with standard network interfaces but proprietary protocols. A SmartObject API gateway can be added to these networks for overall management and federation, and as a gateway to cloud services, e.g. Cosm, ThingWorx... In this case, only the LAN interface is needed.

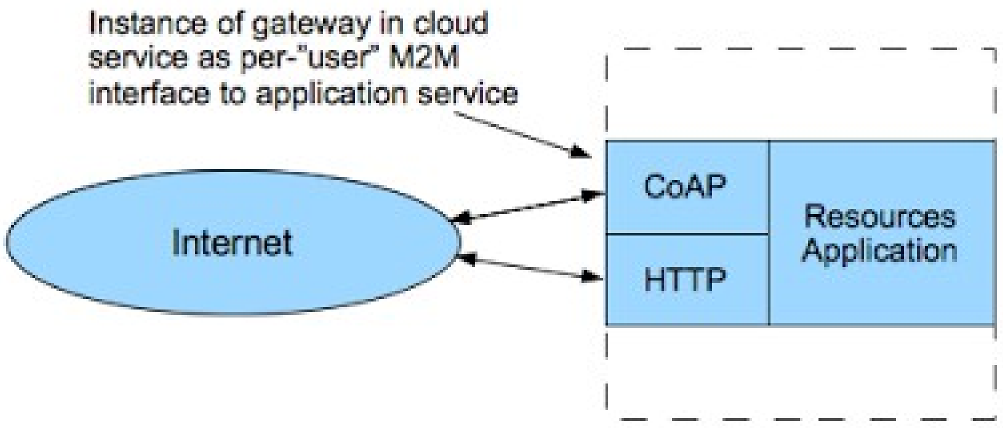
Another use case for the super-gateway is CoAP, which can be routed on LANs. The CoAP network may be connected by a simple CoAP router or bridge via CoAP-over-IP to the internet gateway.



Application agents and software can run in the IoT gateway, which can also host local smartphone and browser-based apps and controls. This allows the system to operate locally without an internet connection if needed.

Gateway-as-a-Service (GaaS)

Finally, the gateway can be used as a per-user resource provided by a cloud service to interact with gateways, other cloud services, and user devices. SmartObject gateway instances can be created by a cloud service to encapsulate per-user resources that can be load balanced, replicated, backed up, etc. Since SmartObject API objects include software agents, the gateway pattern can be used as a front end M2M interface for IoT cloud services. Note that CoAP can be routed over the Internet and is an alternate protocol for M2M interaction.


Instance of gateway in cloud service as per-"user" M2M interface to application service
Internet
CoAP
HTTP
Resources Application

Summary

The Internet of Things consists of a number of heterogenous sensor networks and disparate data sources. A gateway is used to connect these disparate protocols to a common RESTful M2M internet protocol.

The Smart Object API contains resources and methods to provide a framework for implementing IoT gateways. The reference implementation of the Smart Object API will provide a gateway with HTTP and CoAP interfaces, with caching proxy and core-link-format to Semantic Web link bridging.

The reference implementation of a Smart Object service consists of a set of Smart Object gateway instances, each having a separate virtual environment and user/owner encapsulation.

Other IoT gateway frameworks

The OSGi framework is of general interest as a standard way to plug in sensor net protocols, data models, software agents, and other modular software components. Many of the frameworks use OSGi.

There are a few IoT gateway framework projects underway in the EU IoT community.

ETSI - EU Telecommunication M2M Specification
FI-WARE - Future Internet Core Platform (http://www.fi-ware.eu/)
Eclipse M2M gateway (Eurotech)

Ericsson appear to be working on a gateway incorporating some or all of these specifications.

There seems to be a broad framework but no specific data models beyond the use of RESTful interfaces. The SmartObject API and gateway functions described above could be integrated into these frameworks and provide service level interoperability including semantic data discovery and linkage.

Further reading on CoRE and CoAP

Constrained Application Protocol
draft-ietf-core-coap-11

CoAp Feature Analysis
draft-shelby-6lowapp-coap-00

Observing Resources in CoAP

draft-ietf-core-observe-06

Best Practices for HTTP-CoAP Mapping Implementation
draft-castellani-core-http-mapping-05