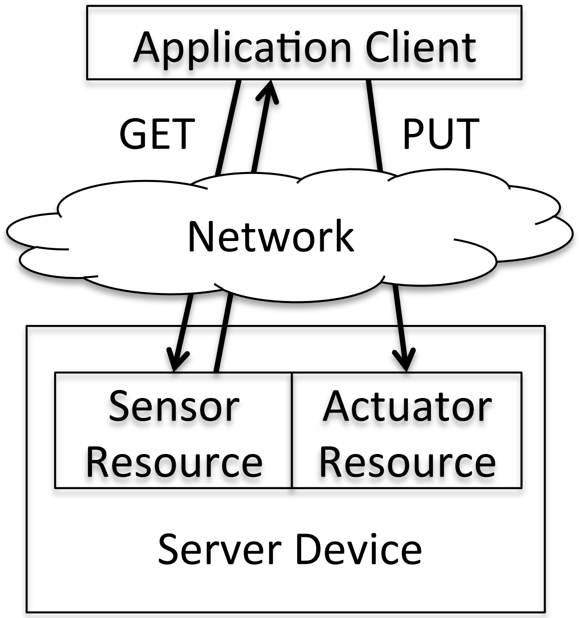
The Event-State Duality REST and Pub-Sub

Michael Koster
ARM

The REST model

- Representational State Transfer
- The server is the origin of system state
- Clients and Servers are protocol endpoints
- The pattern is Request-Response
- Client makes requests, Server encapsulates resource state in responses
- State changes are mapped to REST messages
- Clients may read state (GET) or update state (PUT, POST, DELETE) on the server

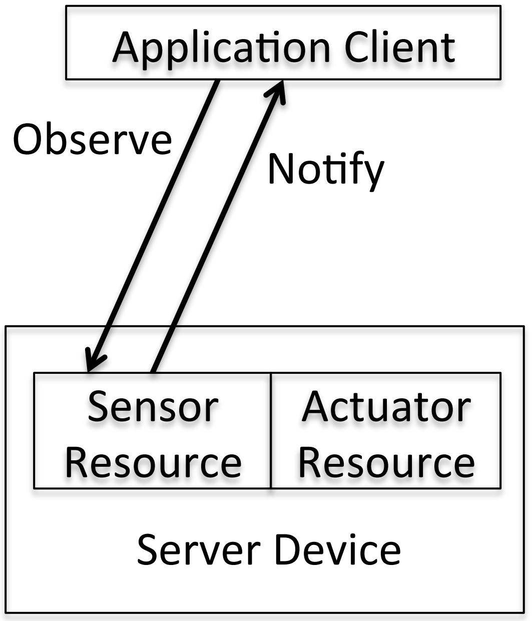
REST between device and application



REST + Events

- Events are state changes in REST resources
- Events can be created and sent when state changes, asynchronous to requests
- Events may be sent using asynchronous responses to GET operations (Observe) to update system state
- Events may also be sent asynchronously to server endpoints using PUT and POST (Push)
- Resource "hooks" are used to detect state changes and initiate event generation
- CoRE Interfaces Bindings (draft-ietf-core-interfaces) are explicit resource hooks with defined actions
- Push bindings send PUT or POST to a defined endpoint when resource state changes
- Observe bindings update the state of a resource when the observed resource changes

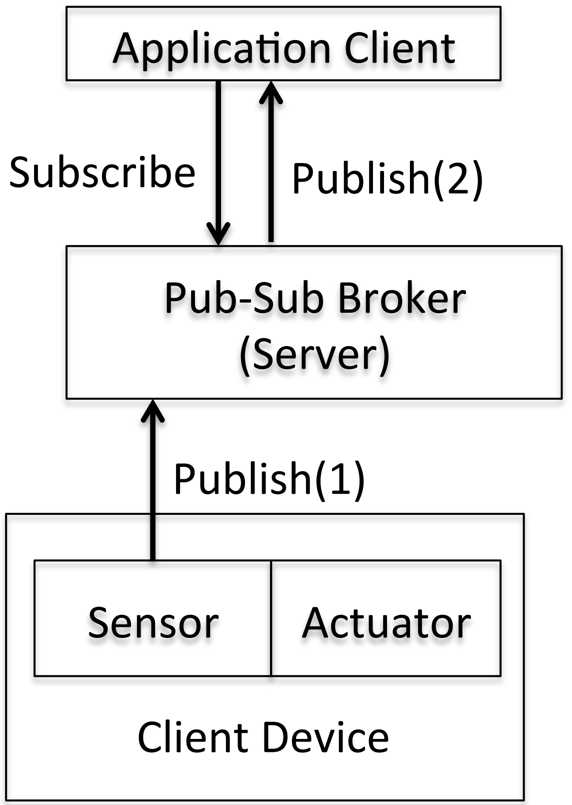
REST with Asynchronous Events



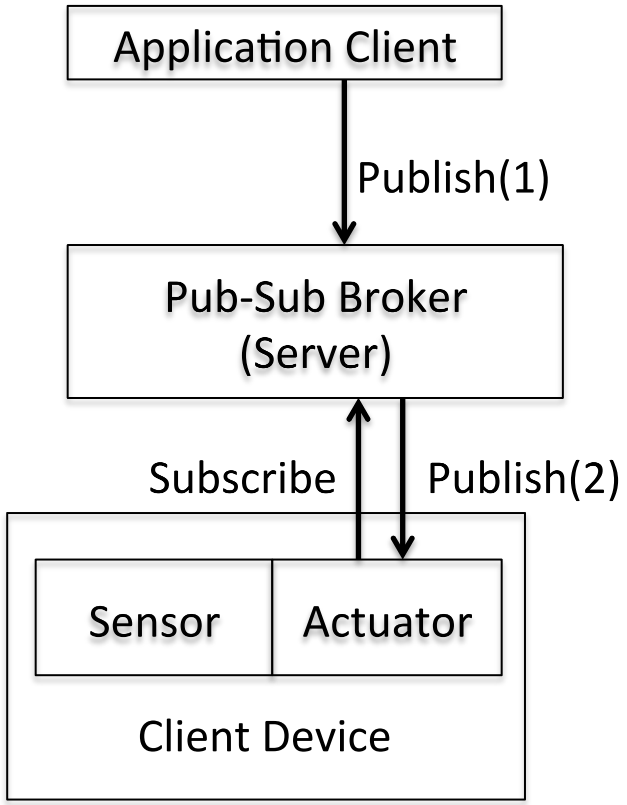
Publish-Subscribe (PubSub)

- Client-Server pattern but not request-response
- Clients publish state change messages to a centralized server called a broker
- Other clients subscribe to the broker and receive the published messages
- The broker receives published messages and re-publishes them to subscribed clients
- PubSub Topics identify particular data items to be exchanged through the broker
- Topics are like resource paths in a REST model; clients publish and subscribe on the same topic in order to communicate state updates of particular data items

Publish – Subscribe with Broker



Sensor Publishes

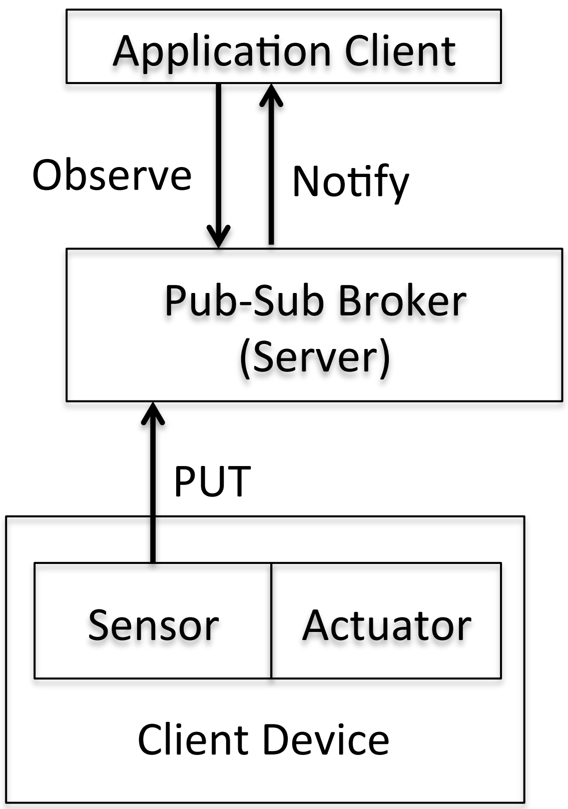


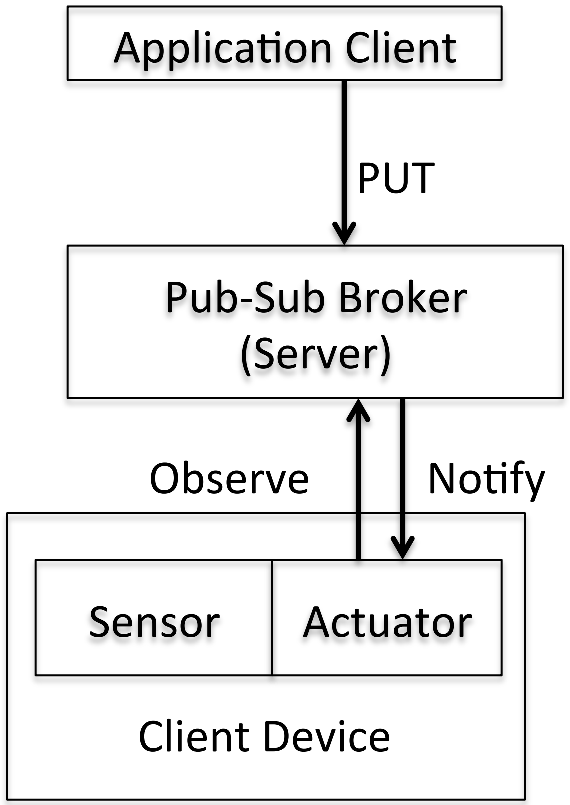
Actuator Subscribes

Mapping PubSub to REST: CoAP PubSub

- PUT and POST operations may be used by a PubSub client to publish state updates to to a broker
- CoAP Observe may be used by a client to subscribe to a broker
- The broker may re-publish to observing clients (subscribers) by using asynchronous response to Observe
- Clients may also read (GET) the last published value from the broker

Pub-Sub Broker with REST API

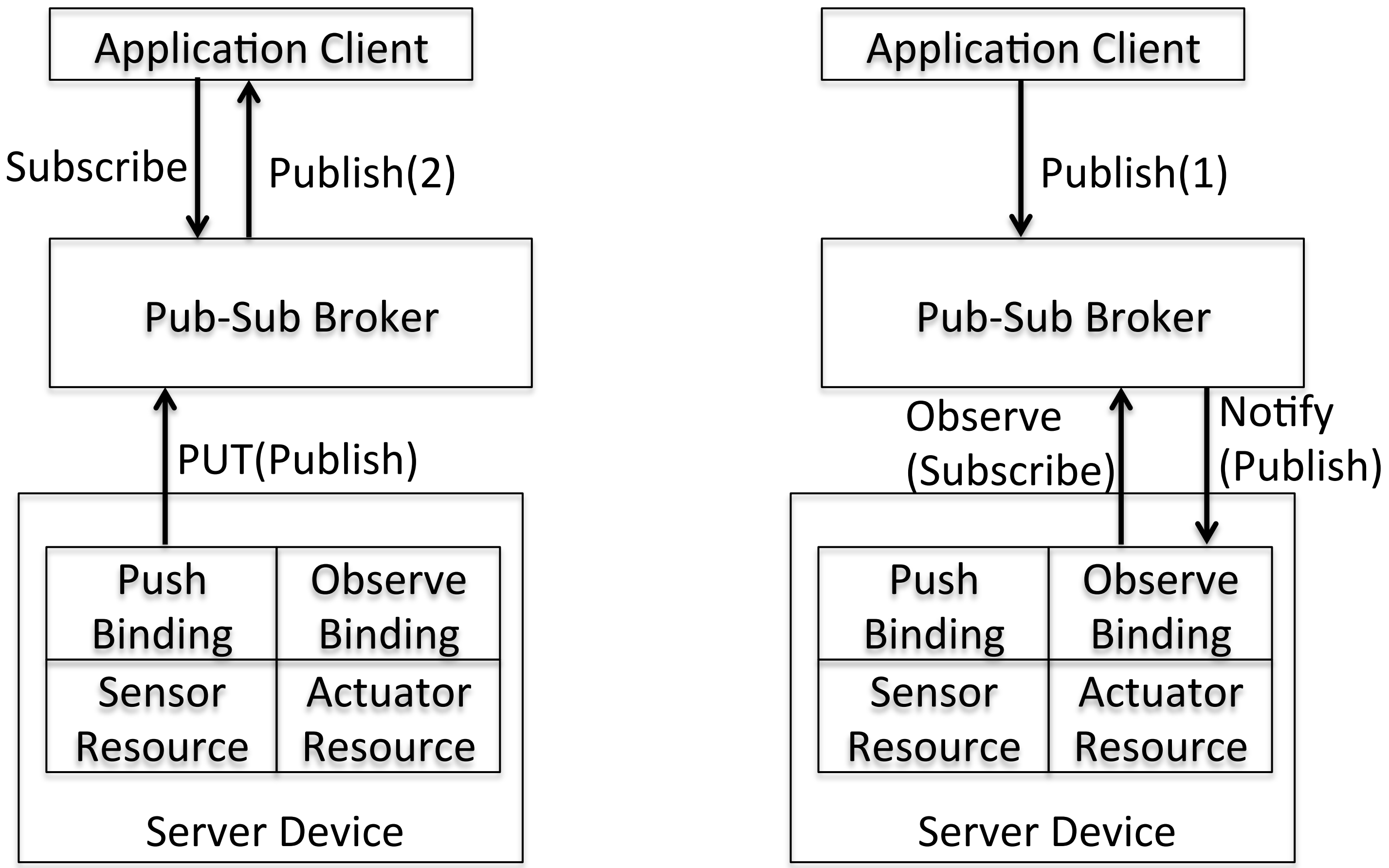




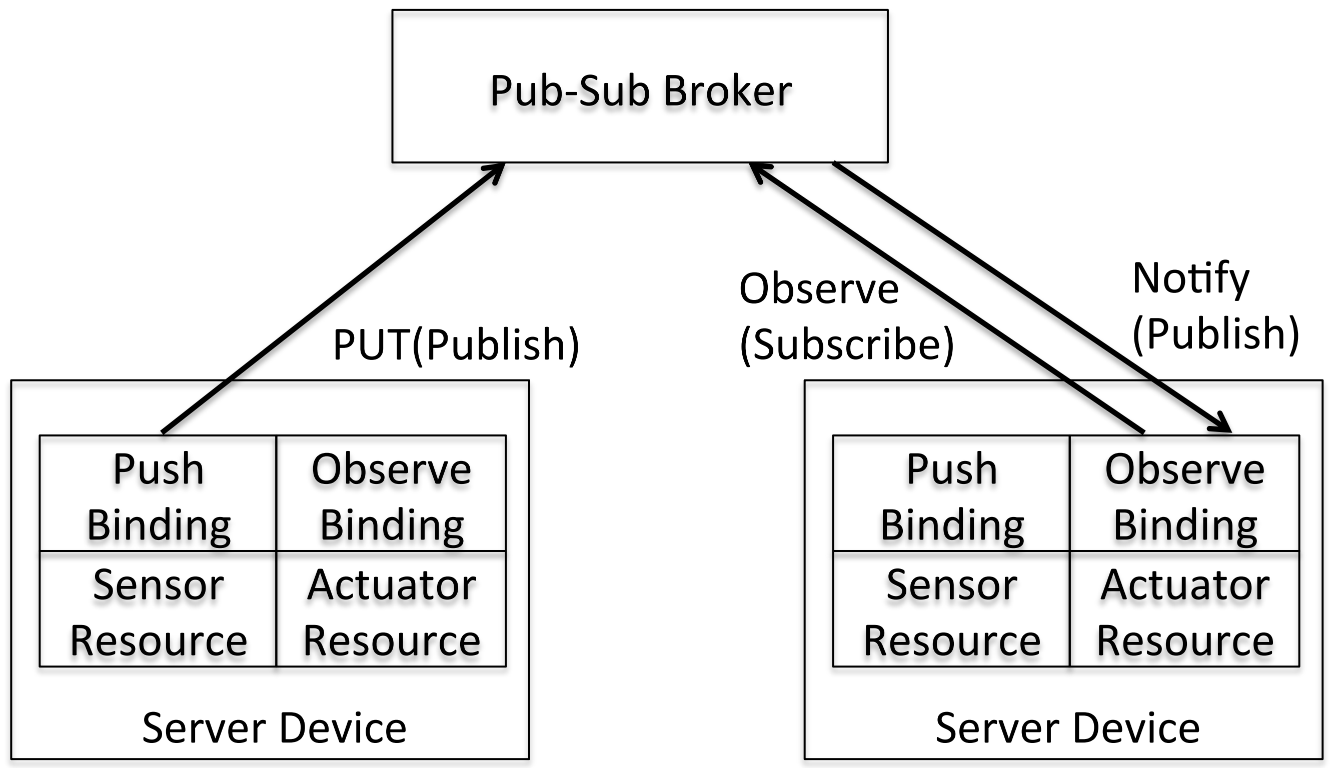
Mapping REST to PubSub: Event Hooks and Bindings

- State changes of REST resources may be published to a PubSub broker
- An Push binding from a REST resource to a broker will publish resource state changes to the broker
- An Observe binding from a REST resource to a broker will update the resource state when the broker publishes updates
- PubSub topics are unique mappings constructed from REST resource paths

REST mapped to Publish-Subscribe



REST mapped to REST through PubSub


Pub-Sub Broker
PUT(Publish)
Observe (Subscribe)
Notify (Publish)
Push Binding
Observe Binding
Sensor Resource
Actuator Resource
Server Device
Push Binding
Observe Binding
Sensor Resource
Actuator Resource
Server Device