WoT Transfer Layer Abstraction

Architecture Notes and Design Patterns

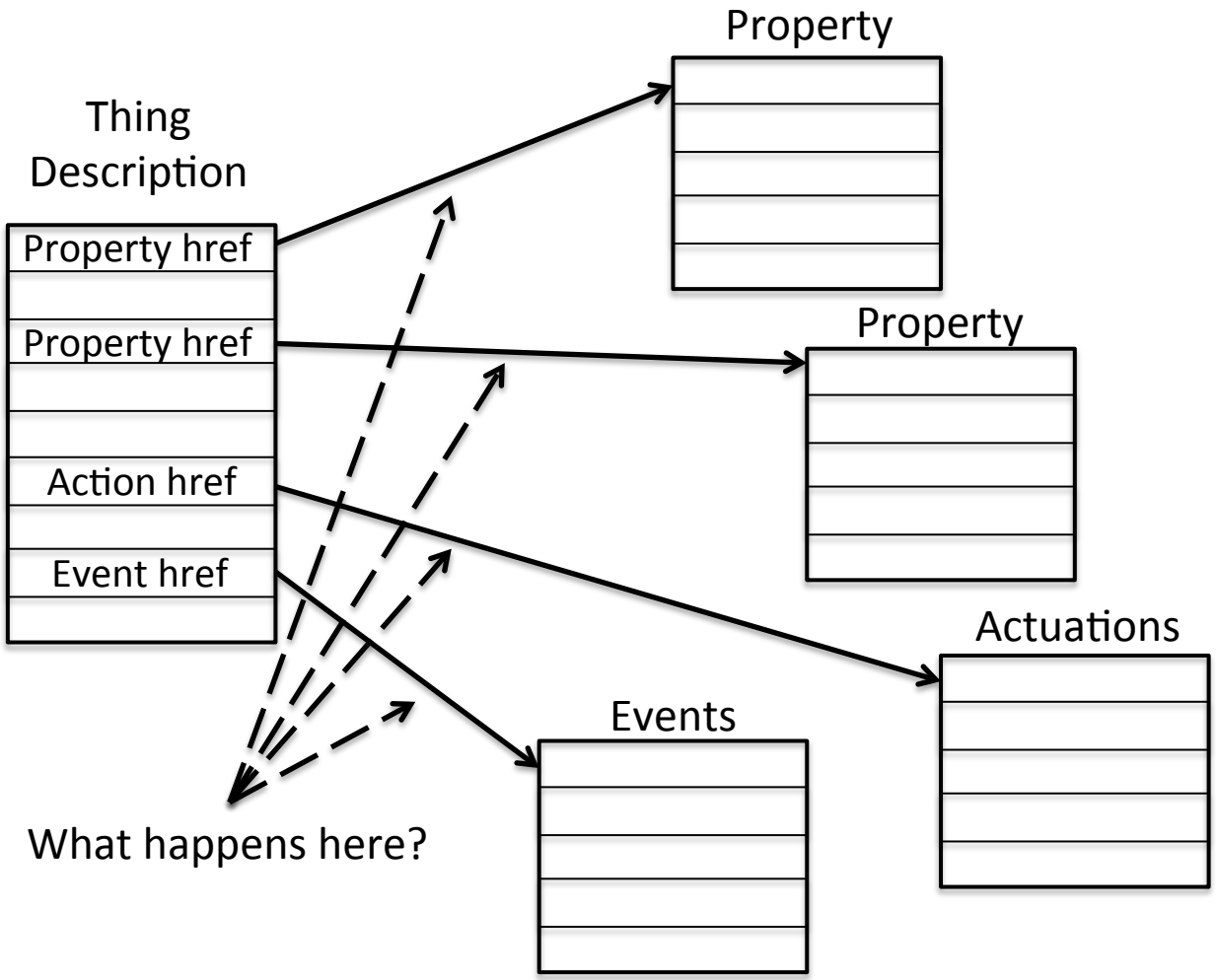
Scope

- How to implement the WoT servient pattern using TD for resource construction, discovery, and interaction
- Define the transfer layer mapping to common protocols CoAP, HTTP, MQTT, Websockets using protocol bindings
- Approach using a consistent definition for the low level interaction model exposed by the "hrefs" of an event, action, or property

Discussion Items

- Architecture
- Transfer Layer
- Asynchronous Communication Patterns with Observable Resources
- Interaction Model Summary
- Content Type, Media Type Indicators for Base Transfer Mapping

Architecture



Transfer Layer

Layer	Description
Application	Scripts that expose and consume resources, execute the "business logic" of things
Things	Thing Description, Stateful Resources
Transfer	REST, Pub-Sub: HTTP, CoAP, MQTT
Transport	UDP, TCP
Network	IP, Ethernet, WiFi, 6LoWPAN, Thread

Common Transfer Semantics

- Define REST + Pubsub based transfer semantics that can serve as a common low level interaction model for the "hrefs" resources pointed to in TD
- One model to map to HTTP, CoAP, MQTT using protocol bindings
- Instances of Events, Actions, Properties, and other entity classes point to resources with well defined transfer semantics
- Messages constructed according to implied semantics rather than explicit forms

Example Common Transfer Model

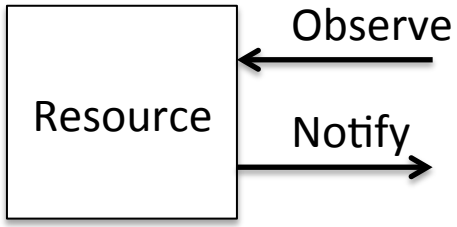
Abstract Transfer	Pubsub	CoAP	HTTP
Create	(Publish)	POST	POST
Retrieve	Subscribe (with retain)	GET	GET
Update	Publish	PUT	PUT
Delete	N/A	DELETE	DELETE
Observe	Subscribe	GET with OBS option	GET text/stream, TE=chunked (SSE)
Notify	Notify Client (onMessage)	Response with OBS option	SSE chunk Response

Observable Resources

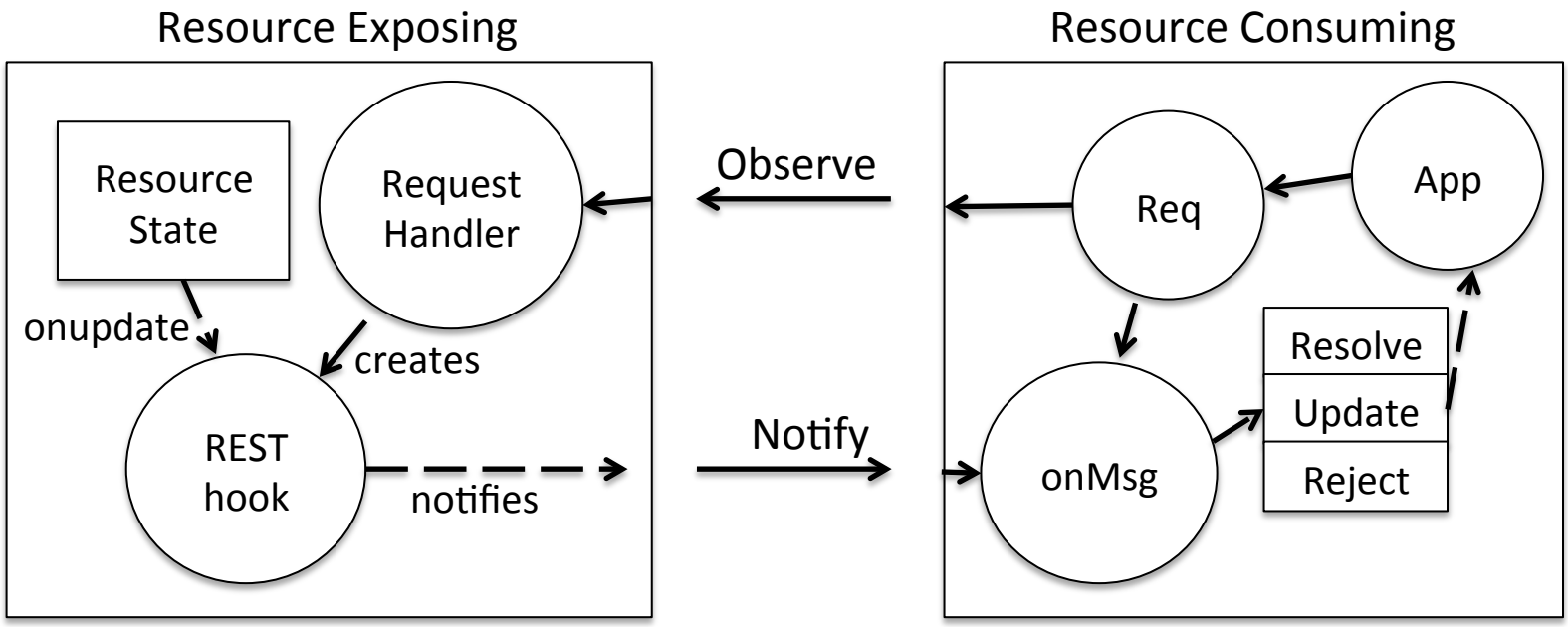
- Basis for asynchrounous events and resource state transfers
- Observable resource is a RESThook based pattern that enables two classes of asynchronous communication
  - Asynchronous callbacks to a software handler following from a request
  - State updates propagated from one resource to another

Observable Resource

- Observe is a retrieve-like operation on a resource that results in an asynchronous sequence of messages rather than a single response

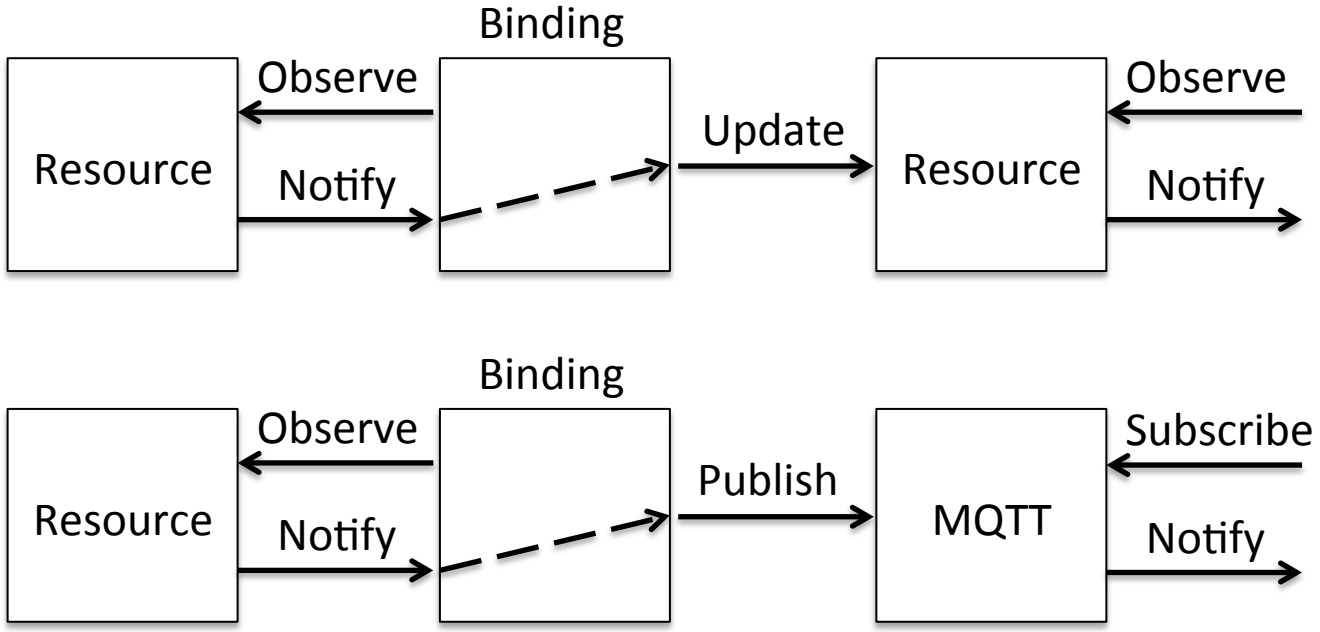


RESThook and Application

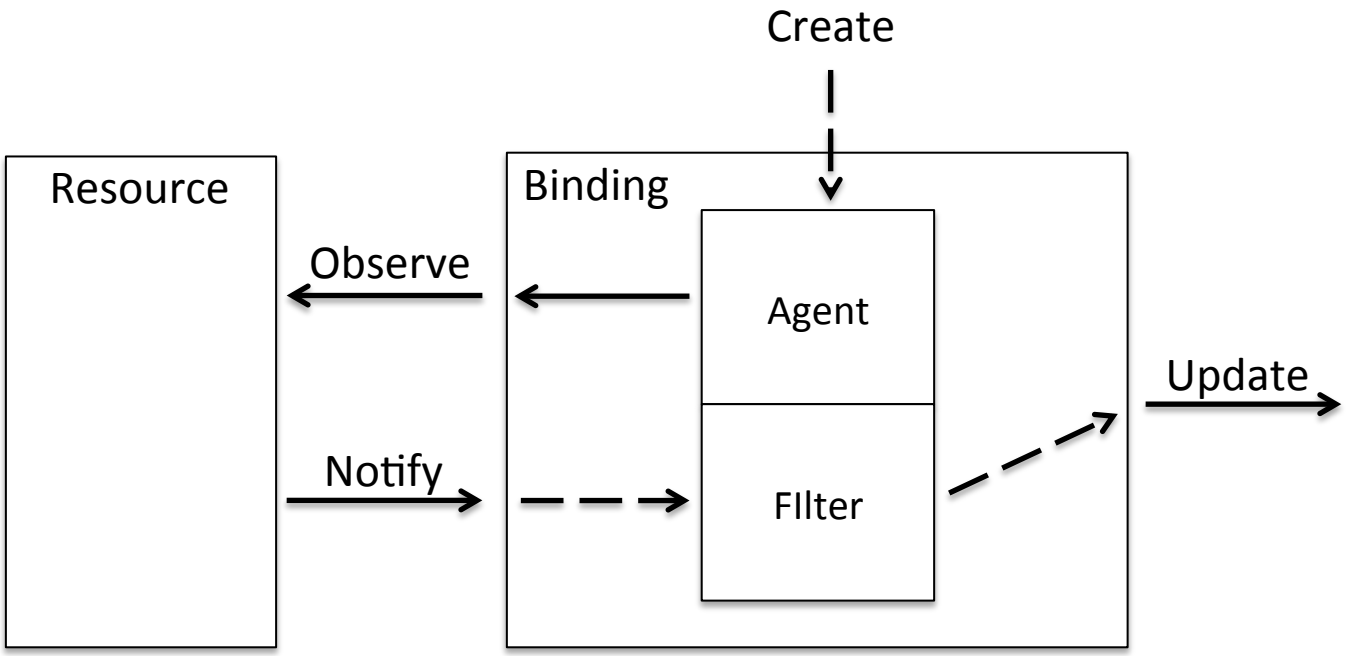


State-Message Binding

- Uses Observable Resource to update another resource or create + send a message
- AKA WoT Subscription

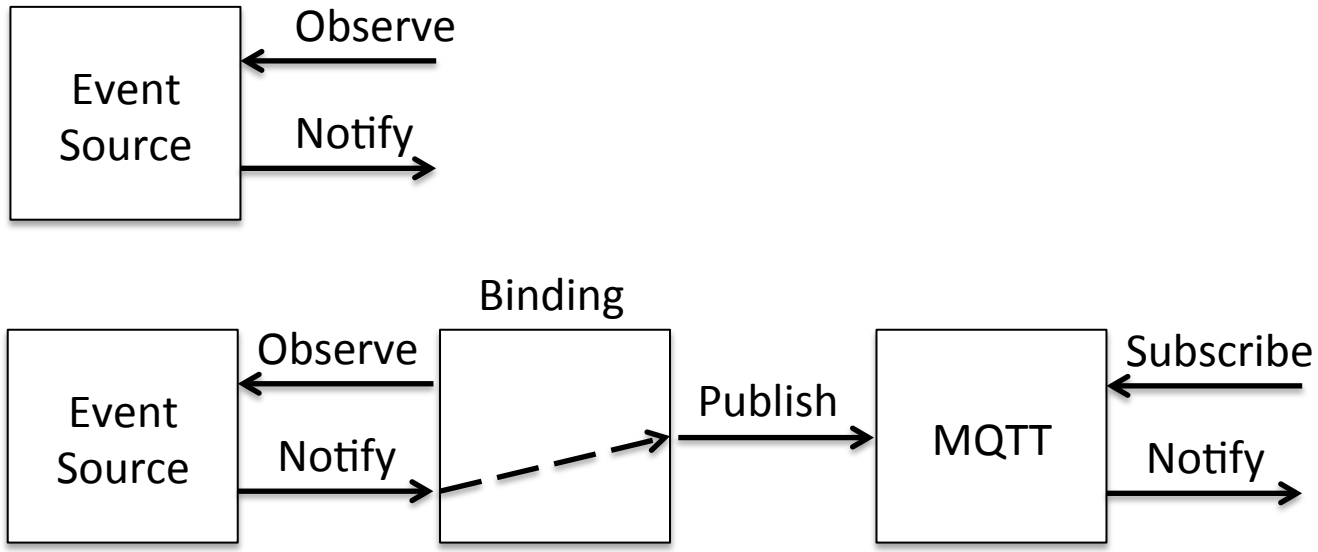


Binding with Filter



Events modeled as Observables

- Events can be modeled as observable resources which emit messages in some format and use the same communication patterns as observable resources



Interaction Model Binding

- Property
  - href points to a resource that can be retrieved, updated, or observed
  - Retrieve returns a representation of the current value or state
  - Update replaces the current value or state with the supplied value (and invokes RESThooks)
  - Emits state updates when Observed

Interaction Model Binding

- Action
  - href points to a resource on which to create instances of actuations
  - Retrieve returns a list of actuation instances
  - Create invokes an action specified by parameters in the payolad, and returns a handle to an observable resource that represents the current state of the actuation instance and emits status updates when observed
  - An actuation instance may be modified or deleted

Interaction Model Binding

- Event
  - href points to a resource which emits event representations when Observed
  - Create on this resource makes a binding to an observable resource, with optional filter parameters
  - Create returns a handle to the observable resource, which emits post-filter event representations when observed

Interaction Model Extension

- Group
  - href points to a group of resources (hrefs)
  - Doing something with a group repeats the same message to all members of the group
  - Separate resource is used for managing the membership of groups

Interaction Model Extension

- Group Configuration
  - href points to a resource which represents a collection of resource groups
  - Create on this resource adds a new group to the collection and returns a handle to the group
  - Retrieve on this resource returns the current list of group descriptors with hrefs
  - Each group exposes a configuration resource which is used to add, list, or remove resources

Content Format

- Use of base resource layer can be indicated with a content format identifier e.g. "application/wot+json" in the accept header of content-type header of the request made to the hrefs found in TD instances
- Defines common representation formats for the payloads of events, notifications, bindings, actuations, group constructors, etc.

WoT Servient – Another View