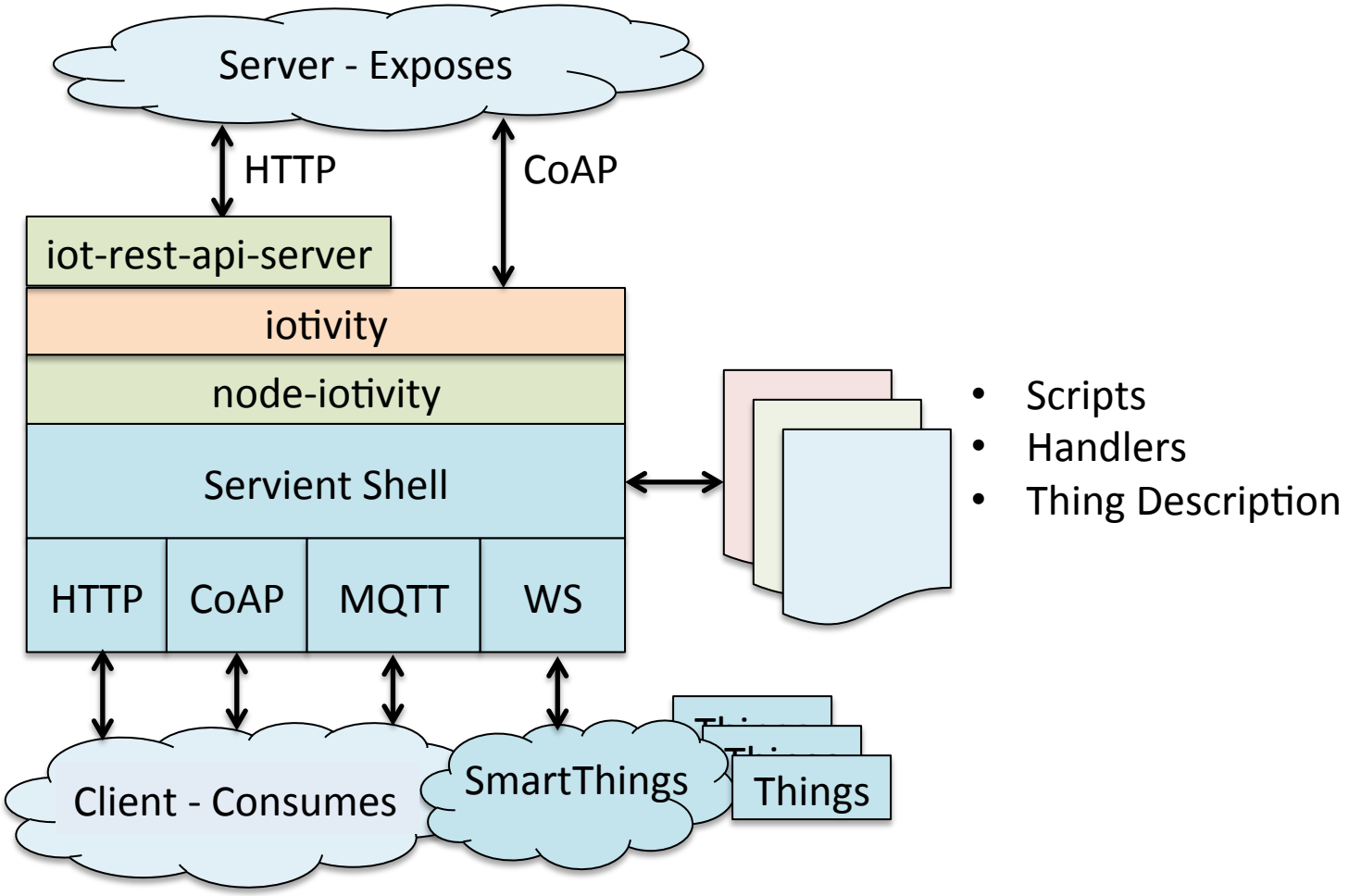
WoT Servient using SmartThings and Iotivity

May 23, 2016

WoT Servient using SmartThings and Iotivity

- Expose SmartThings capabilities as OCF resource types with WoT Thing Description
- HTTP, CoAP, and MQTT Protocol Bindings
- Uses iotivity as the resource layer
- node-iotivity: existing nodejs API and binding
- iot-rest-api-server: existing HTTP proxy for iotivity
- New nodejs based servient shell provides a client + server scripting API and abstract transfer layer

SmartThings + Iotivity WoT Servient



Some Questions

- What does a SmartThings TD look like?
- What does a SmartThings/OCF Resource Type look like?
- Can iotivity expose Thing Descriptions?
- How does the action collection pattern work?
- How does the subscription pattern work?

Mapping SmartThings Capabilities

- W3C WoT Thing Descriptions
  - WoT Thing models a SmartThings Thing, at the device level
  - Capability-level granularity would be better for applications and security
- OCF Resource Types and Collections
  - Attributes and Commands mapped using the WoT Interaction model and abstract transfer binding
  - https://github.com/connectIOT/iotivity-servient/blob/master/docs/abstract-transfer.pdf
  - Resource Directory + device bridge architecture

Use case for SmartThings Models

- IoTivity SmartThings Bridge
- A SmartThing is a collection of Capabilities
- Each Capability has a model
- Discover SmartThings Things and construct IoTivity resource instances from capabilities
- Map the W3C WoT Interaction model in RAML, customize the json-schema payloads for Capability types

Modeling SmartThings Capabilities

- Use oneiota to create a canonical model for mapping an abstract SmartThings Capability using the WoT Interaction Model
- Create application capability types by customizing payloads using json-schema and derived models
- Use WoT Thing Descriptions and RAML Capability Models to construct instances

Modeling Process

- Map ST Capabilities to Thing Descriptions
- Compose Thing TDs from the Capability TDs
- Capability TD hrefs point to action, event, and property instances exposed by IoTivity
- IoTivity exposes devices as collections of Capability models, each containing resources representing WoT properties, actions, and events
- Capability types differ only by payload schema and identifier values

OCF WoT Device Bridge

- Devices are registered in /oic/res/, with rich links to collections of Capabilities
- Capabilities are composed of OCF resource types that model WoT TD entities according to the proposed protocol binding and workflows
  - atl.wot.property
  - atl.wot.action
  - atl.wot.event
  - atl.wot.actioninstance
  - atl.wot.subscription

SmartThings OCF Device Bridge Mapping to Thing Description