W3C Web of Things

POC Planning – Intel Contributions

Michael McCool, Principal Engineer
SSG/DPD/Technology Pathfinding and Innovation

Proof of Concept (POC) Development: Goals

***Overcome** a specific technical obstacle*

***Demonstrate** business value of a new technology or standard*

***Advance** engagements with key ecosystem players*

1. **Metadata bridges** (value: increase adoption and applicability)
   - Support *all* existing IoT devices from *multiple* ecosystems
2. **Voice control** (value: demonstrate utility, engage key ecosystem player)
   - Support *any* IoT device with adaptive semantic voice controls
3. **Fog Integration** (value: develop support for ubiquitous localized services)
   - Deploy using local compute resources for proxies, translators, and applications

Metadata Bridge

Goal: Increase number of devices accessible via WoT standard

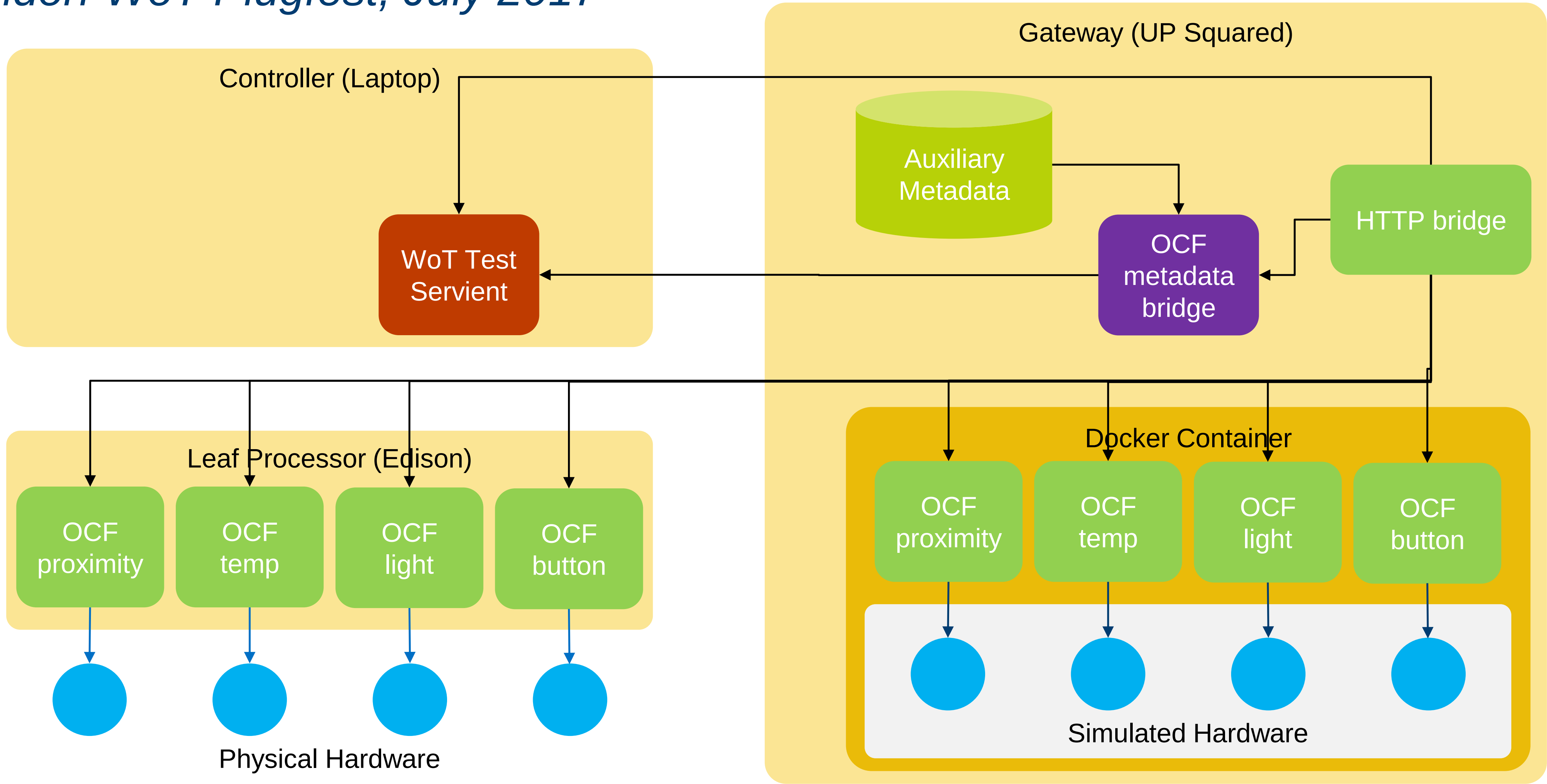
- Read external metadata and translate into WoT TDs
- Infer and add extra information not available in native metadata
- Make IoT devices available to any system that can process WoT TDs
- Does *not* need specific changes to target IoT devices.

→ *Prototype built using OCF Smart Home Demo as test target.*

Metadata Bridge

Dusseldorf WoT Plugfest, July 2017

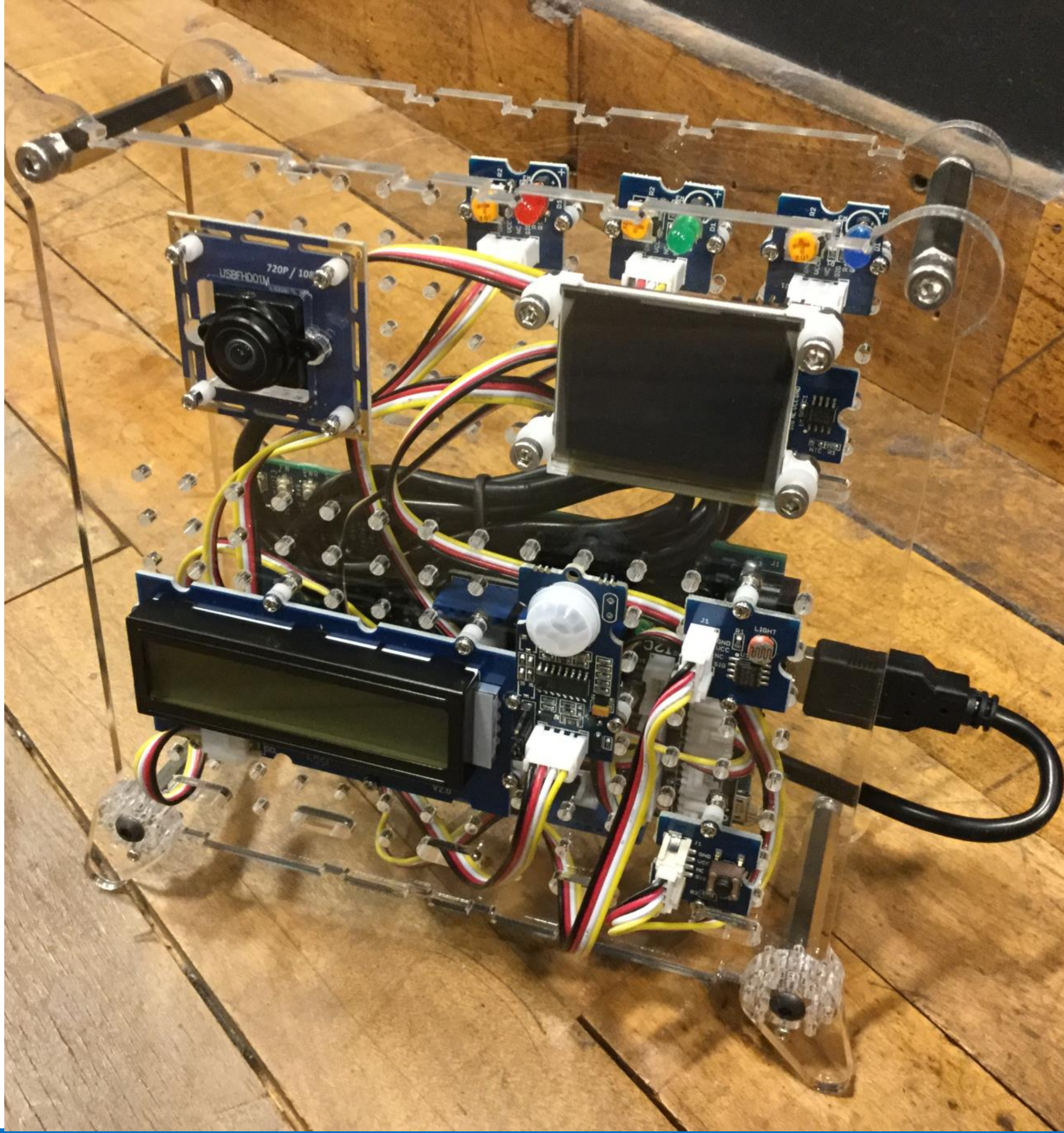
Goal: Translate OCF metadata and make it available to via WoT Thing Description



System Integration

Goal: Improve and Expand System Architecture

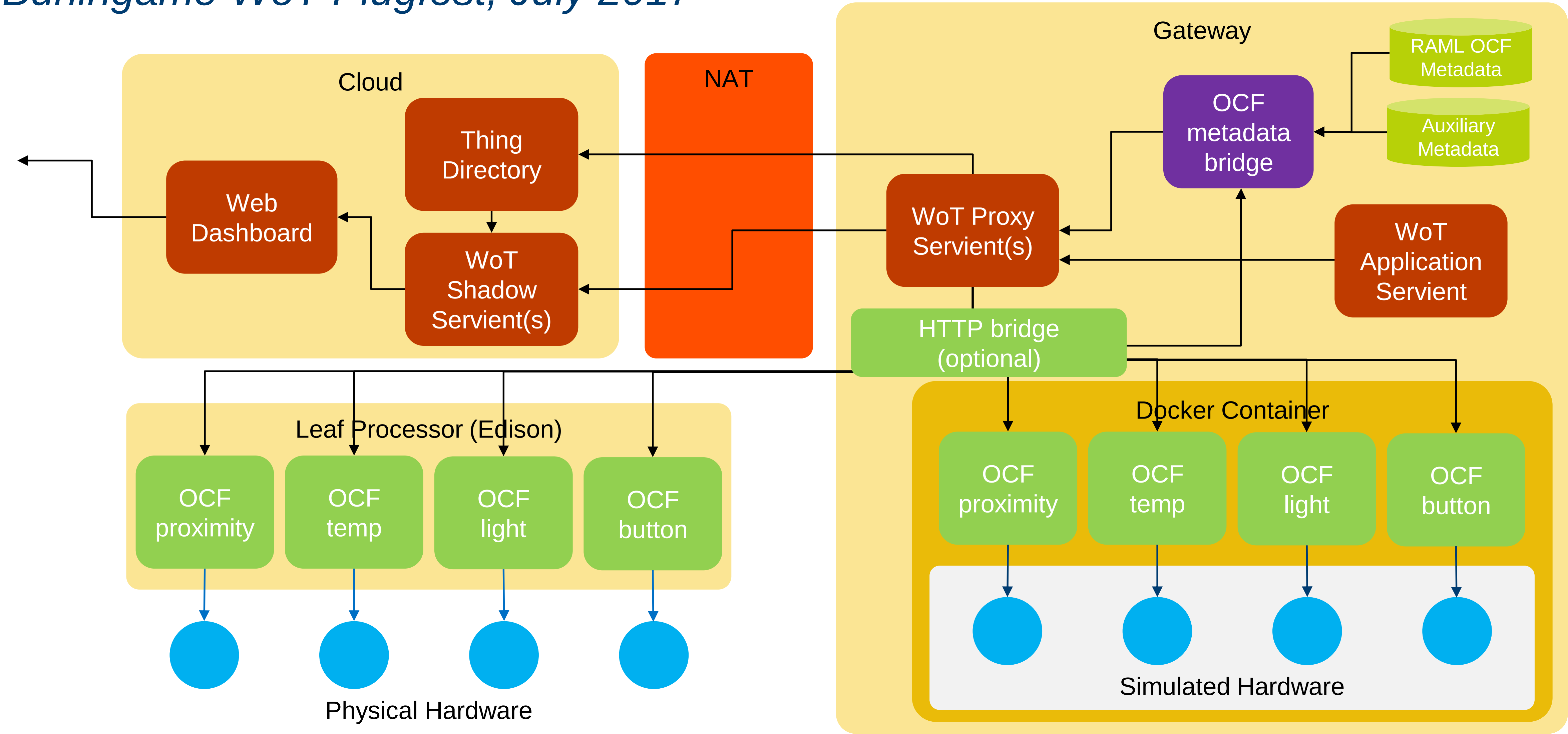
- Support NAT traversal, Thing Directory, Proxies, and Shadows
- Support CoAP
- Support OCF native metadata more completely
- Support non-OCF devices and services
- Support secure transport and authentication



System Integration

Burlingame WoT Plugfest, July 2017

Add: Ingest official OCF data models, traverse NAT, use Thing Directory, CoAP



Voice Control

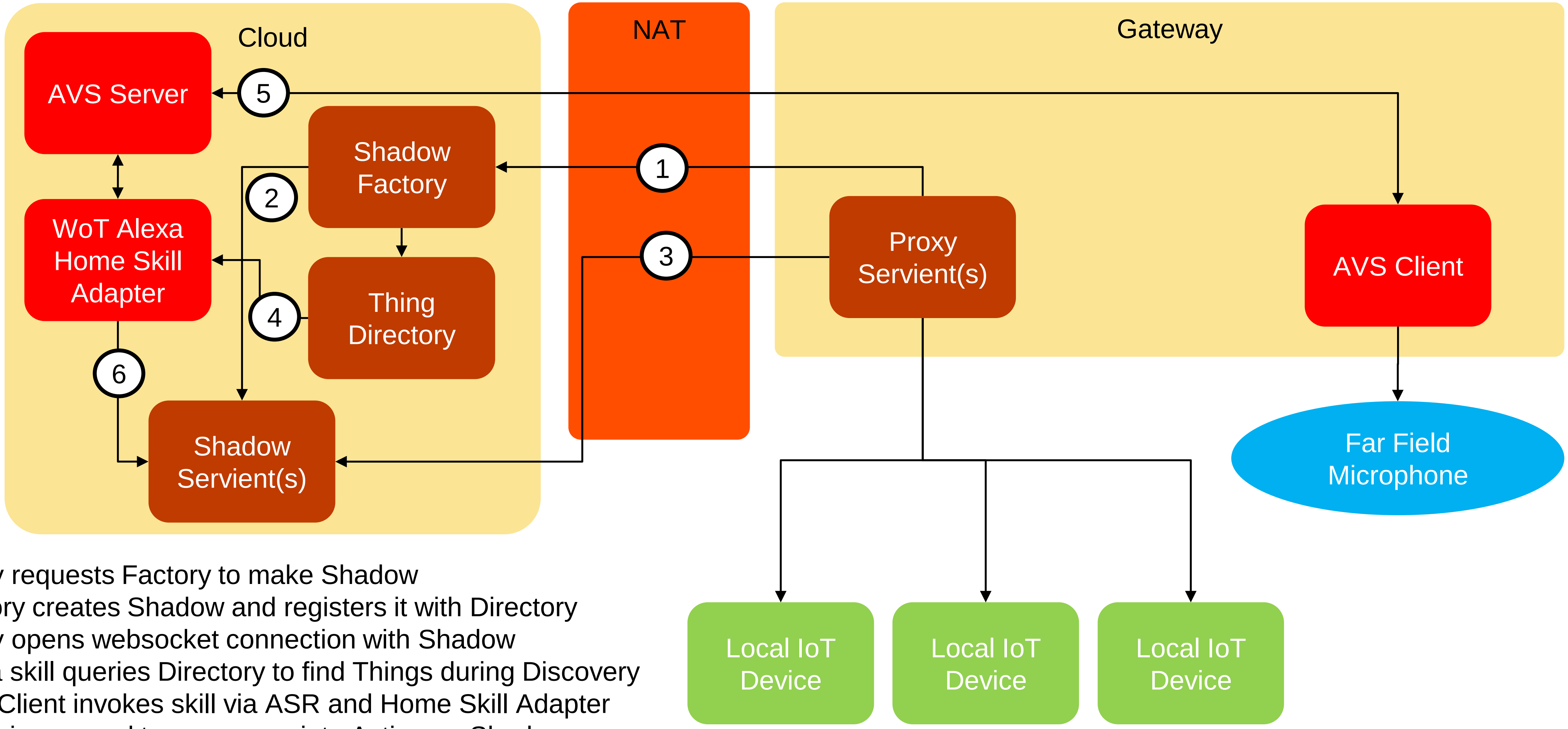
Goal: Enable *automatic* voice control of *any* WoT-enabled device

- Demonstrate use of semantic markup of Thing Description
  - Using iot.schema.org and SSNO ontologies and semantic inferencing tools
- Generate adaptive AVS Alexa skill, bridging with Alexa Home Skill
- Layer with WoT metadata bridges to control devices from multiple ecosystems (including OCF)

Voice Control

Burlingame WoT Plugfest, July 2017

Goal: Provide generic voice interface using WoT Thing Descriptions



1. Proxy requests Factory to make Shadow
2. Factory creates Shadow and registers it with Directory
3. Proxy opens websocket connection with Shadow
4. Alexa skill queries Directory to find Things during Discovery
5. AVS Client invokes skill via ASR and Home Skill Adapter
6. Intent is mapped to an appropriate Action on Shadow

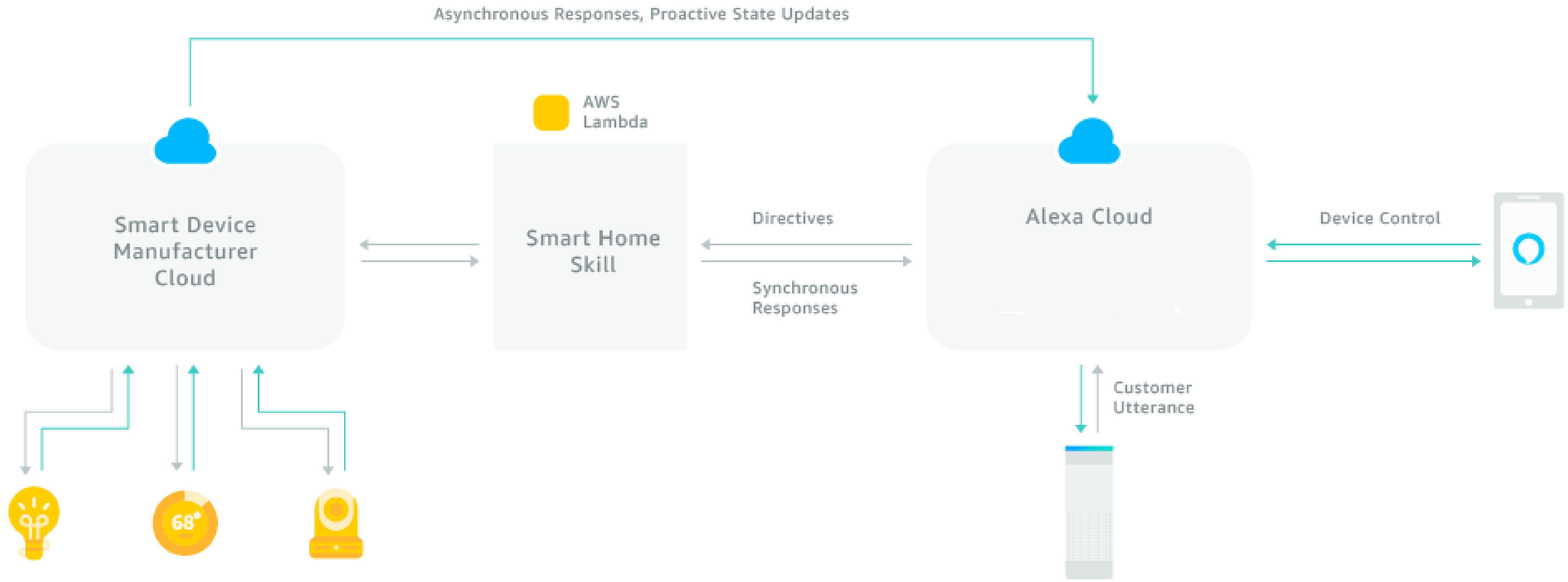
Alexa Smart Home Skill API - Summary

- Assumes devices controlled by “device cloud”: a web service with a public URL.
- Supports set of standard, built-in “intents” with predefined semantics
- When understands one of these intents, creates a “directive” which is sent to an “adapter”, which is typically an AWS lambda function. Lambda is called for discovery first, then later with directives.
- Directive contains: customer authentication, device identifier, and an action-specific payload.
- Adapter should check the authentication (OAuth2.0), then send the payload to the specified device via the device cloud. The “device identifier” includes uuid and cookies provided during discovery.
- Respond with an event that indicates success or failure. Can be synchronous (from the lambda) or asynchronous (from the device cloud).
- Devices can also have shadowed status information and this can be proactively updated.

→ *Note that all interactions MUST complete within 8 seconds, and ideally 5 seconds.*

For details, see https://developer.amazon.com/docs/smarthome/understand-the-smart-home-skill-api.html

Alexa Smart Home Skill API - Summary



AVS Home Skill Semantics: Capabilities

Device functionality is modeled by specifying a set of standard capabilities.

- Each capability has a specific interaction model and payload
  …really more like an “interface” than just a semantic category
- Devices can support more than one capability
- “Device clouds” can have more than one device…

There are also a set of standard capabilities to get/set properties

Many capabilities support both absolute and relative mechanisms to adjust properties

Many “actions” also report the current state

AVS Home Skill Capabilities: Targets

Capability	Description
PowerController	TurnOn or TurnOff a device (has associated powerState)
PowerLevelController	SetPowerLevel (absolute) or AdjustPowerLevel (relative) on a device. Has an associated powerLevel property that can be set to a valuel between 0 and 100
PercentageController	Generic interface similar to PowerLevelController or BrightnessLevelController, but used when “power” or “brightness” is not a valid description of the controlled property. Assumes a value between 0 and 100 in an “percentage” property. Can also be used to read sensors.
TemperatureSensor	No controls, just use ReportState/StateReport to query “temperature” property.
BrightnessController	SetBrightness (absolute) or AdjustBrightness (relative) on a light. Has an associated brightness property that can be set to a value between 0 and 100
ColorTemperatureController	SetColorTemperature (absolute) or DecreaseColorTemperature, IncreaseColorTemperature (relative, no value) on a device. Has an associated colorTemperature property that can be set to a temperature in Kelvin (voice interface understands color names, “warm”, “daylight”, etc).
ColorController	SetColor of a light using “color” whose type is a hue, saturation, brightness (HSB) triple. Note that mapping from this to RGB involves a matrix transform and a nonlinearity. Voice interface uses color names.

AVS Home Skill Capability: Additional Examples

Capability	Description
LockController	Lock or Unlock a lock device
ThermostatController	Supports thermostats with one, two, or three setpoints. Can also be queried to find the current temperature using a property and ReportState/StateReport. There is also a “mode” property: ECO, AUTO, COOL, HEAT, etc.
InputController	Select AV input to a TV using set of standard names (HDMI, etc)
ChannelController	Select Channel on a TV (both absolute and relative)
PlaybackController	Start and stop audio source
SceneController	Select a scene by name. Some limits on discovery based on device type.
Speaker	SetVolume, AdjustVolume, SetMute
StepSpeaker	AdjustVolume, SetMute (a subset of “Speaker” with no absolute volume, for when range of volume is not known).
CameraStreamController	Start and stop streaming video

AVS Home Skill Semantics: System Messages

Directives can also get or create messages with various types corresponding to "system capabilities".

Capability	Description
Discovery	Every device: reports initial identification and capabilities
Authorization	AcceptGrant, provides bearer token, currently only OAuth2.0 supported
Response	Respond to directive - success
ErrorResponse	Respond to directive – error/failure, gives type of error, message for logging. Errors can include system issues (eg connectivity) but also payload issues (value out of range).
ReportState	Request state report
StateReport	Reply to state report
DeferredResponse	Used to indicate that will respond asynchronously to request (typ. >8s needed)
EndPointHealth	Check if physical endpoint can be reached from cloud shadow. Has associated Boolean property, "connectivity"

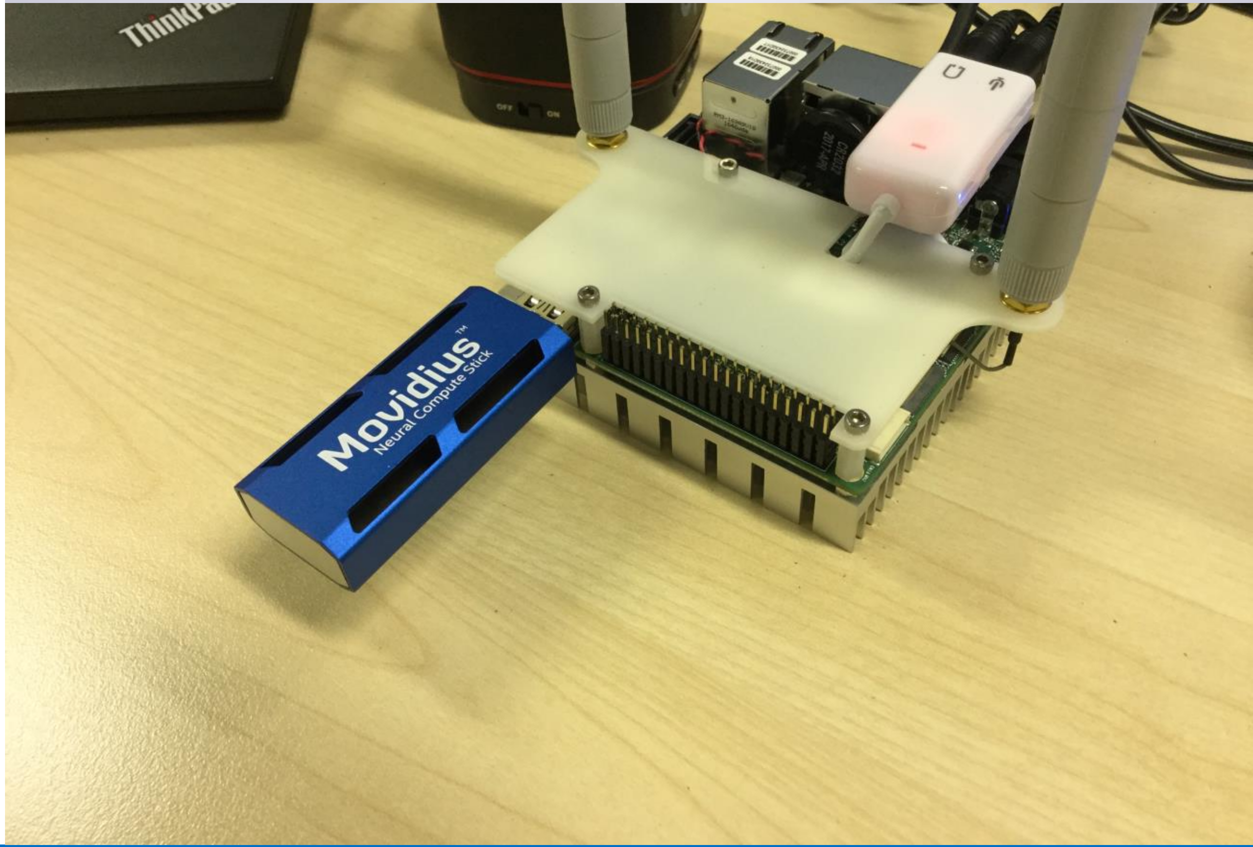
AVS Issues and Notes

- "Device Cloud" (Shadow) *must* support HTTPS.
- OAuth2.0 authentication *must* be supported.
- Responses *cannot* take longer than 8s.
  - Live querying of the device status, if it takes longer than that, needs to use a shadow. The shadow state then needs to be proactively or asynchronously updated.
  - Actions that take longer than 8s to complete should use an asynchronous response to indicate completion.
- Property status reporting
  - Not just the value, but also the timestamp, time uncertainty, and units
- Optional features:
  - Status query, proactive status updates, scenes and scene discovery, streaming video and audio

Fog Integration

Goal: Make services discoverable and locally available using WoT combined with fog capabilities.

- Run WoT services on fog (OpenFog) software stack
  - Thing directory (WoT discovery)
  - Metadata bridges (eg to OCF)
  - Voice skill services (eg AVS)
  - Application servients (IoT services)
- Connect to computational services
  - Recognition service (eg AlexNet)



TODO: Industrial POCs

- Voice control is primarily a *home* use case
- What do we want for industrial use cases?
- What is different about industrial requirements?
  - Set of standards and their requirements are different
  - Real-time, pub-sub architectures, TSN, functional safety, central management, asset management, complex access controls, energy management…
  - Less emphasis on privacy, more on safety
- Are municipal and building use cases more like home or industrial use cases?
- What about automotive? Transportation? Retail? Medical?
  - They all have their own ecosystems…

TODO: Scenarios

Step through some specific scenarios

- Voice Control for Consumer End Users
- Service Composition for System Integrators
- Compute Services (eg person recognition) for other Service Providers

Include not only home, but also industrial and municipal

- Long tail – translation to older protocols
- Adapt to different requirements, eg real time and functional safety

→ Need to talk to industrial users about requirements

Smart Security Scenario: Person Detection

This specific scenario can be demonstrated using OCF Smart Home capabilities, combined with one or more IP cameras. This application is of use in multiple verticals (home (eg security system), industrial (eg functional safety), municipal (eg subway track ingress).

Problem: Simple proximity detection systems, such as motion sensors, can be triggered by events other than human presence, leading to false alarms. However, constantly running AI-based human recognition is expensive.

Solution: Combine simple distributed motion sensors with AI-based person recognition. When a motion sensor is triggered, an image is captured in that location and sent to a person recognition service. Only if a person is detected is an alarm triggered. In addition, the AI service can be centralized so that only one recognition service is needed even if there are multiple cameras and motion sensors. As an extension, cameras could be mounted on mobile platforms (aka robots…).