

Web of Things

TD extended with iotschema.org

Darko Anicic, Michael Koster

WoT F2F Meeting Burlingame, USA

Motivation: Thing Discovery

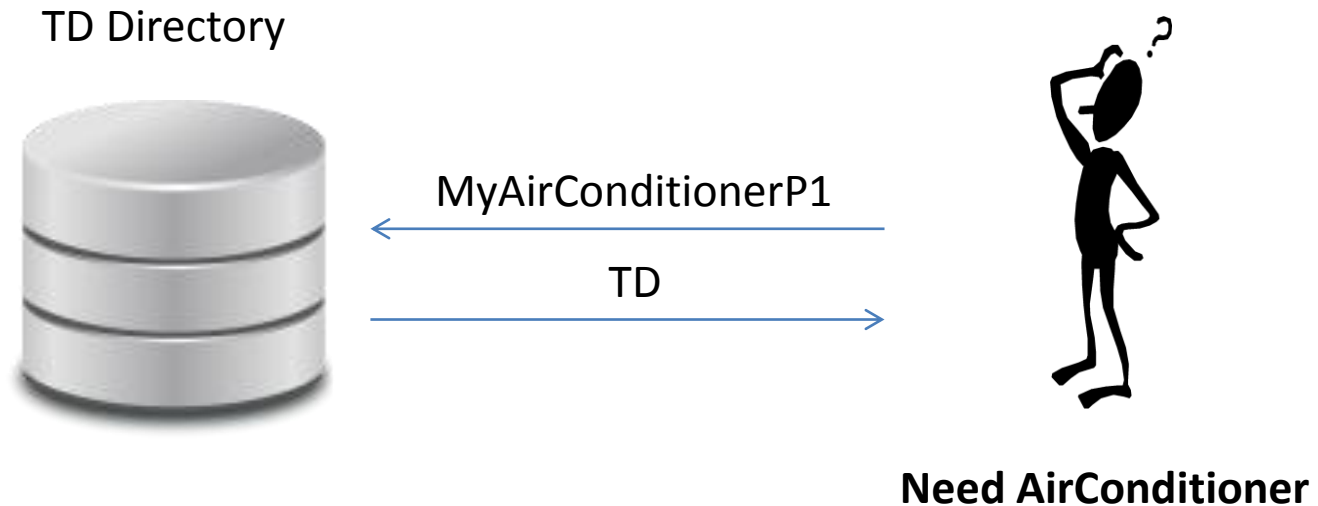
TD Directory



Need AirConditioner

- **Problem Statement**
 - Discovery of Things suitable for a WoT application;
 - Interop client can discover Things in order to dynamically adapt applications to changes, e.g., to replace malfunctioning sensor with an equivalent one;

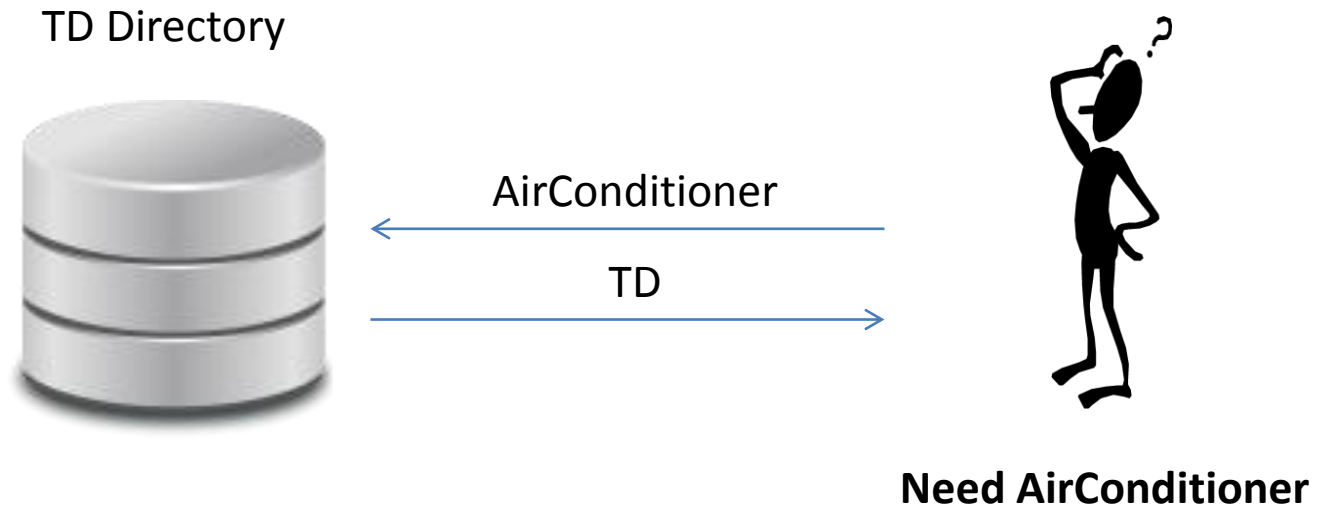
Motivation: Thing Discovery



- Problem Statement

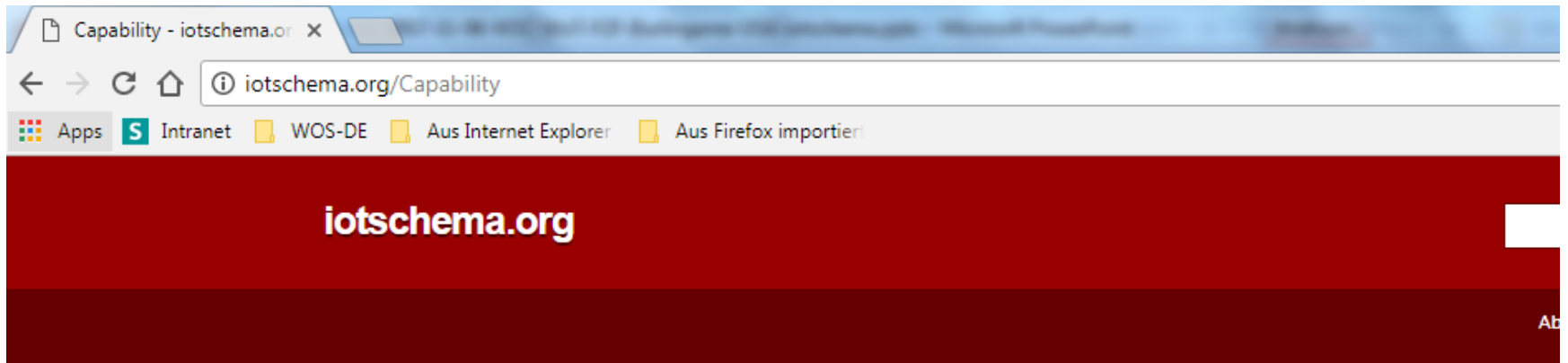
- Discovery of Things suitable for a WoT application;
- Interop client can discover Things in order to dynamically adapt applications to changes, e.g., to replace malfunctioning sensor with an equivalent one;

Motivation: Thing Discovery



- Problem Statement

- Discovery of Things suitable for a WoT application;
- Interop client can discover Things in order to dynamically adapt applications to changes, e.g., to replace malfunctioning sensor with an equivalent one;



Capability

Canonical URL: <http://iotschema.org/Capability>

Capability

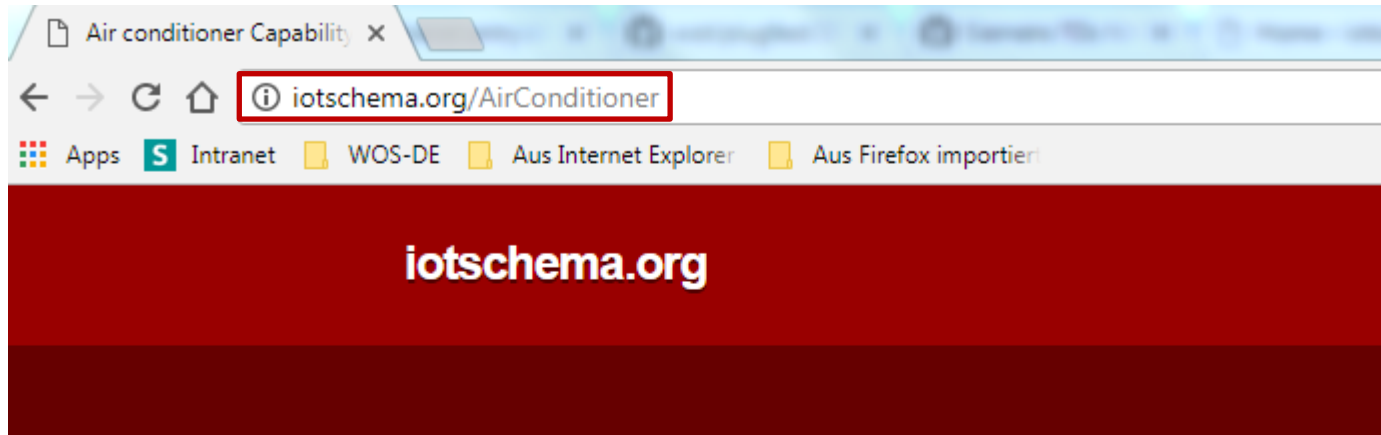
Capability Class

Property	Expected Type	Description
Properties from <u>Capability</u>		
<u>Property For InteractionPattern</u>	<u>Interaction Pattern</u>	A property that relates a capability with its interaction patterns.

More specific Types

- [Temperature Capability](#)
- [Air conditioner Capability](#)
- [Humidity Capability](#)
- [Illuminance Capability](#)
- [Light Control](#)
- [Motion Control](#)
- [Thermostat Capability](#)

Air Conditioner Uniquely Identifiable



Air conditioner Capability

Canonical URL: <http://iotschema.org/AirConditioner>

[Capability](#) > [Air conditioner Capability](#)

A capability for air conditioner

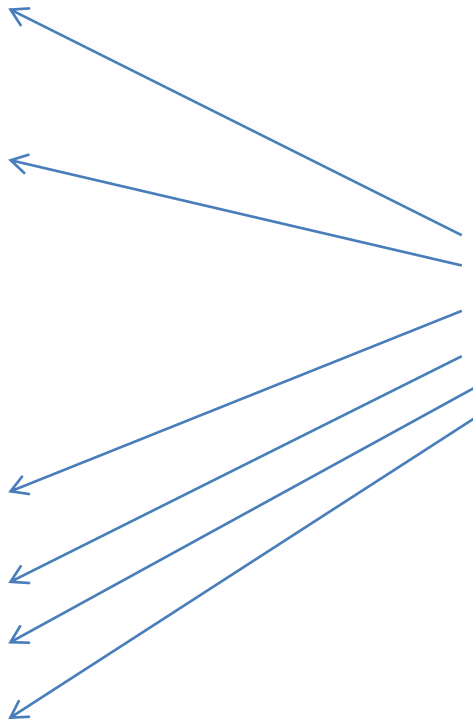
Instances of Property may appear as values for the following properties

Property	On Types	Description
<u>Observes</u>	Sensor	Relation between a Sensor and either a Property or an Event that it is capable of sensing.
<u>For Property</u>	Actuator	Relation between an Actuator and either a Property or an Action that it is capable of actuation.

More specific Types

- Motion Detected
- Turn On
- Temperature
- Current Level
- Turn Off
- Humidity
- Binary Switch
- Light Colour
- Operation Status
- Motion Type
- Switch Status
- Ramp Time
- Illuminance
- Set Dimmer
- Target Humidity
- Target Temperature
- Transition Time
- Run Mode
- Set Dimmer
- Count Down
- Light Colour
- Wind Strength
- Motion Detected

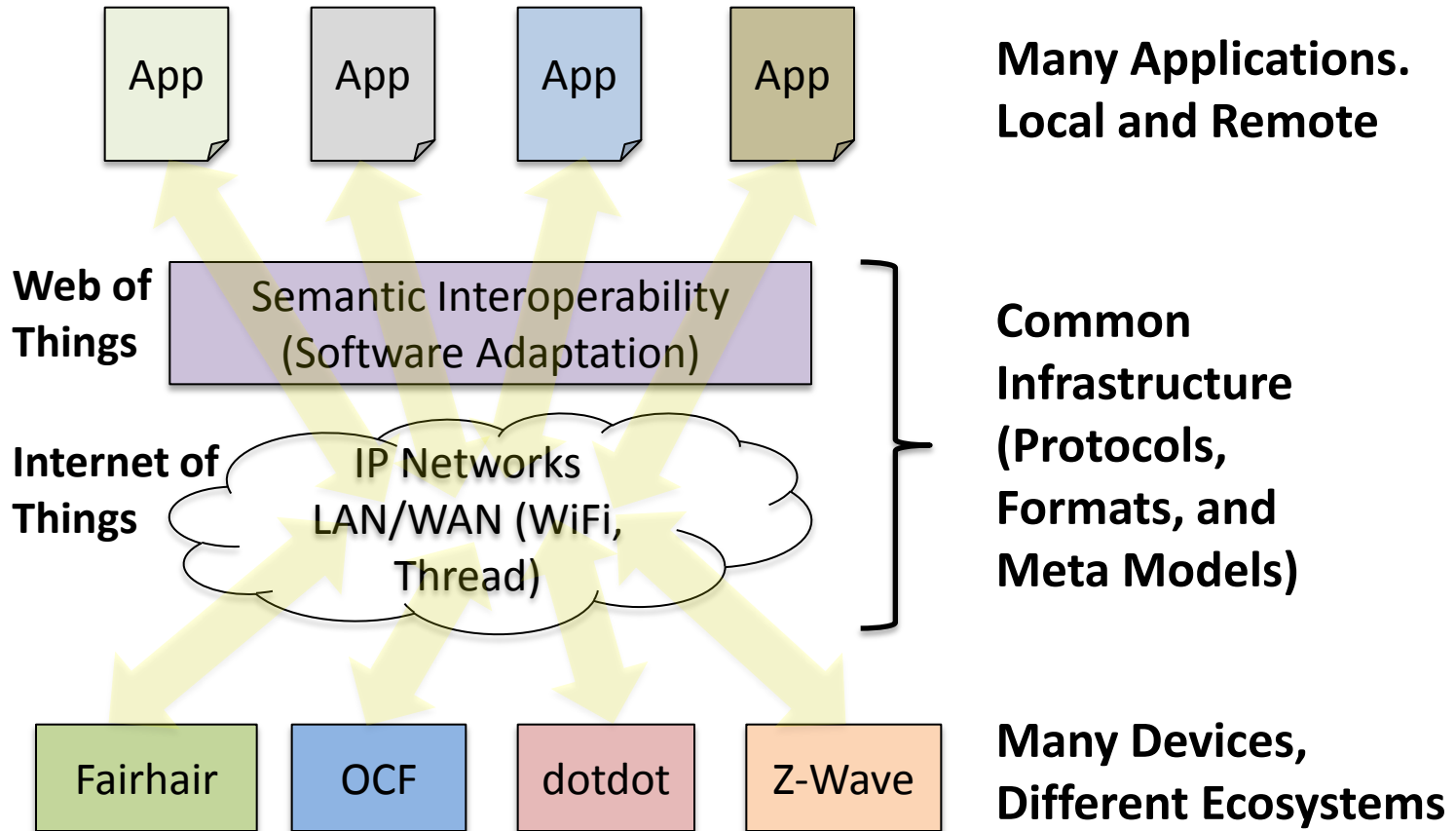
Interaction Patterns of
AirConditioner Capability



Overview

IOT SCHEMA

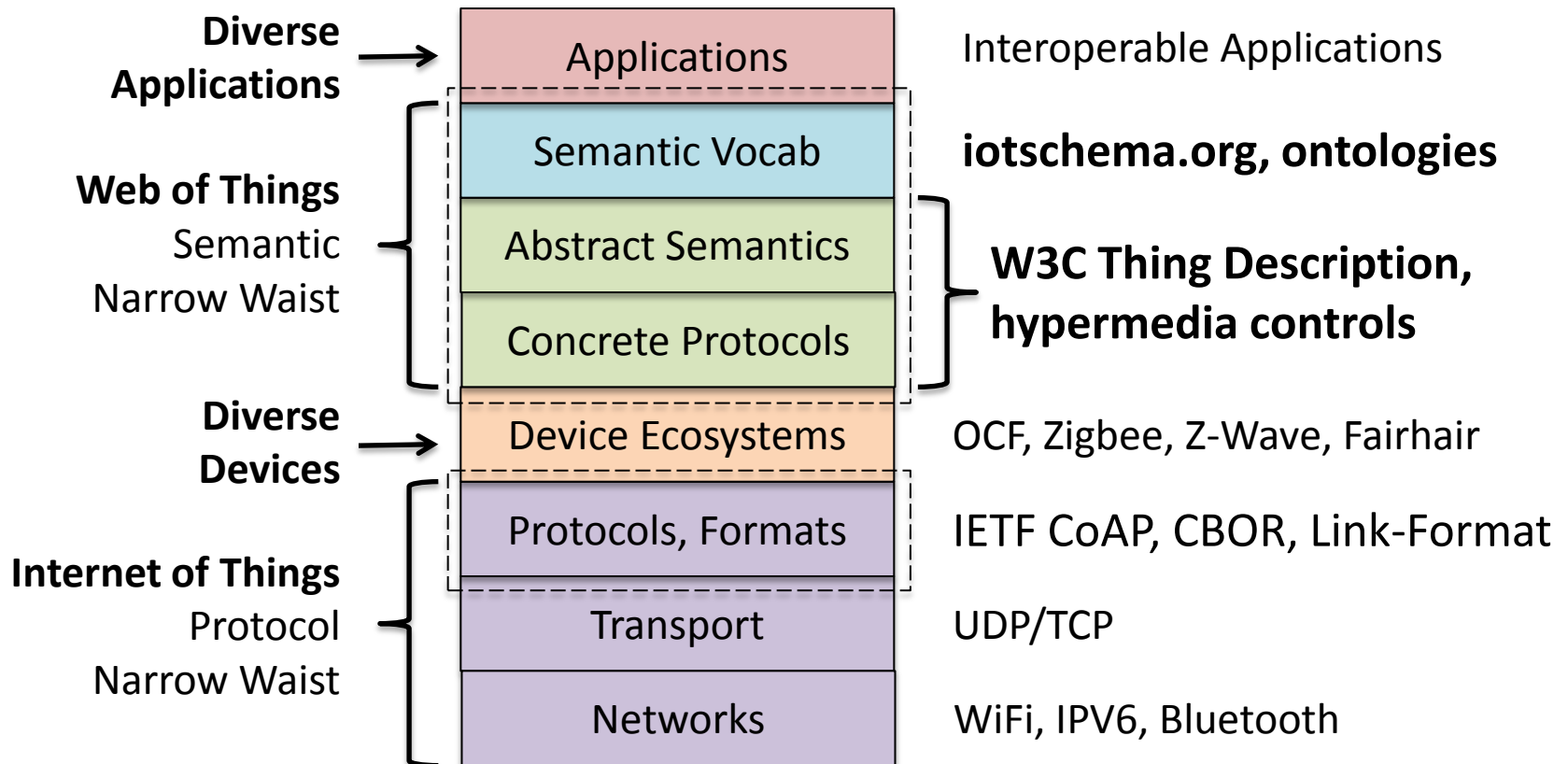
Narrow Waist in System Design



What needs to be built?

- Application level semantic interoperability
 - Well known formats to describe common affordances of connected things (What does it do? What can I control?)
 - A way to describe how to interact with connected things from different device ecosystems, which use similar protocols but diverse data models
 - Enable easy implementation of Bridges, Libraries, Translators, Mappings, Bindings, Proxies

Layers in the Semantic Stack



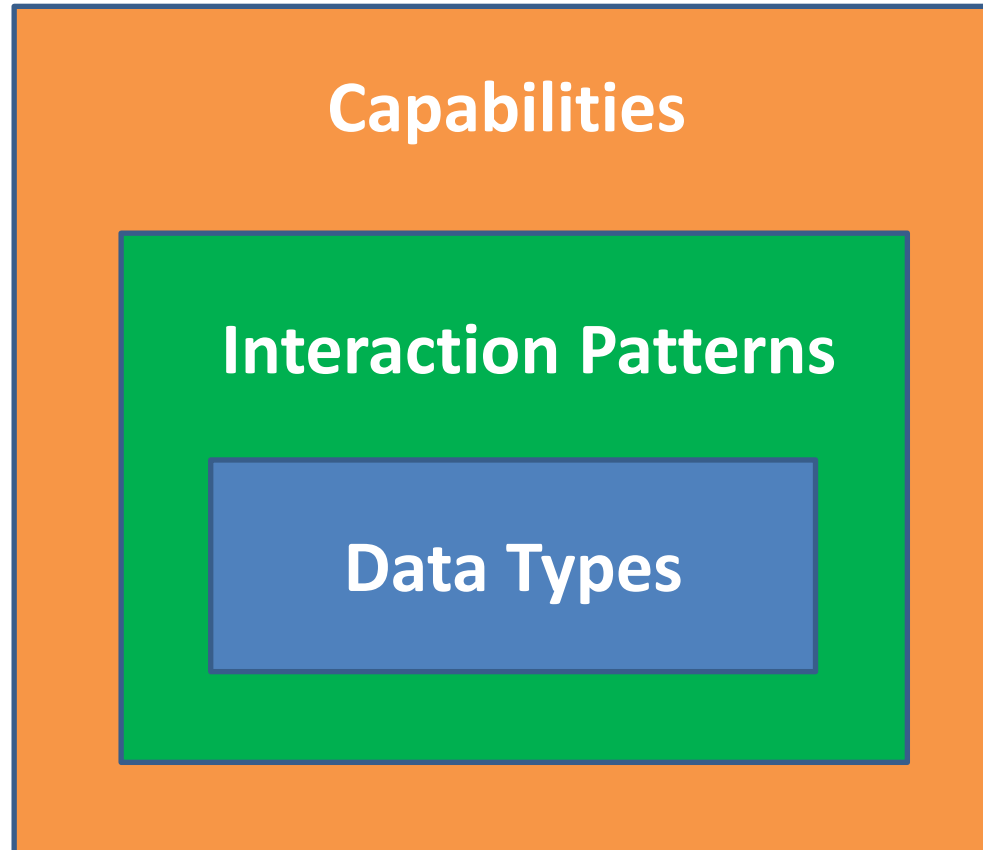
Capability Abstraction

- Connected things expose simple "traits" and functional affordances like REST resources, common command sets
- A capability is an abstraction of something a connected thing can do, like measure temperature or switch on and off
- Capability descriptions can be layered and composed for instances of connected things

Examples of Capabilities

Thing	Capabilities	Properties	Type	Actions
Motion Sensor	Motion Sensing	Motion	Boolean	(read)
Temperature	Temperature Sensing	Temperature	Number	(read)
Light	Switch	SwitchState	Boolean	TurnOn TurnOff
Light	Level Control	Level TransitionTime	Number Number	MoveToLevel MoveLevel StepLevel

Different Layers of iotschema.org



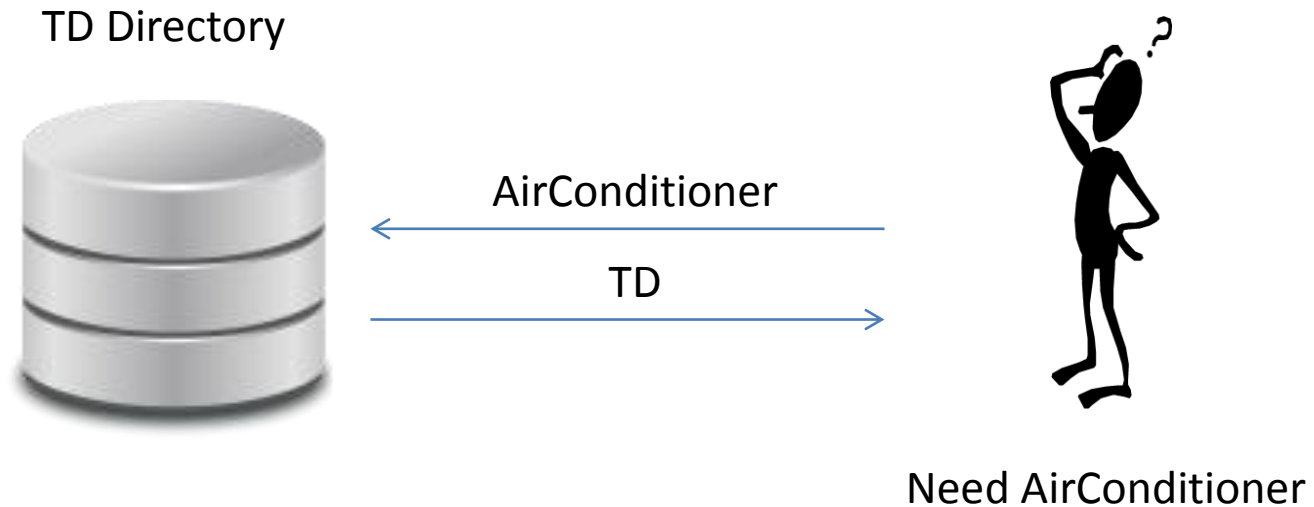
Available Resources for iotschema

- Current schema location: <http://iotschema.org/>
- A stable version should be moved to:
<http://iot.schema.org/>
- GitHub repo:
- <https://github.com/iot-schema-collab>
- Working document and notes:
- <https://docs.google.com/document/d/1p8KIUEcQYseoPzvjtKfvVCNAXx3OfyuTUEReB3H2B5M/edit>

Overview

TD RECIPES

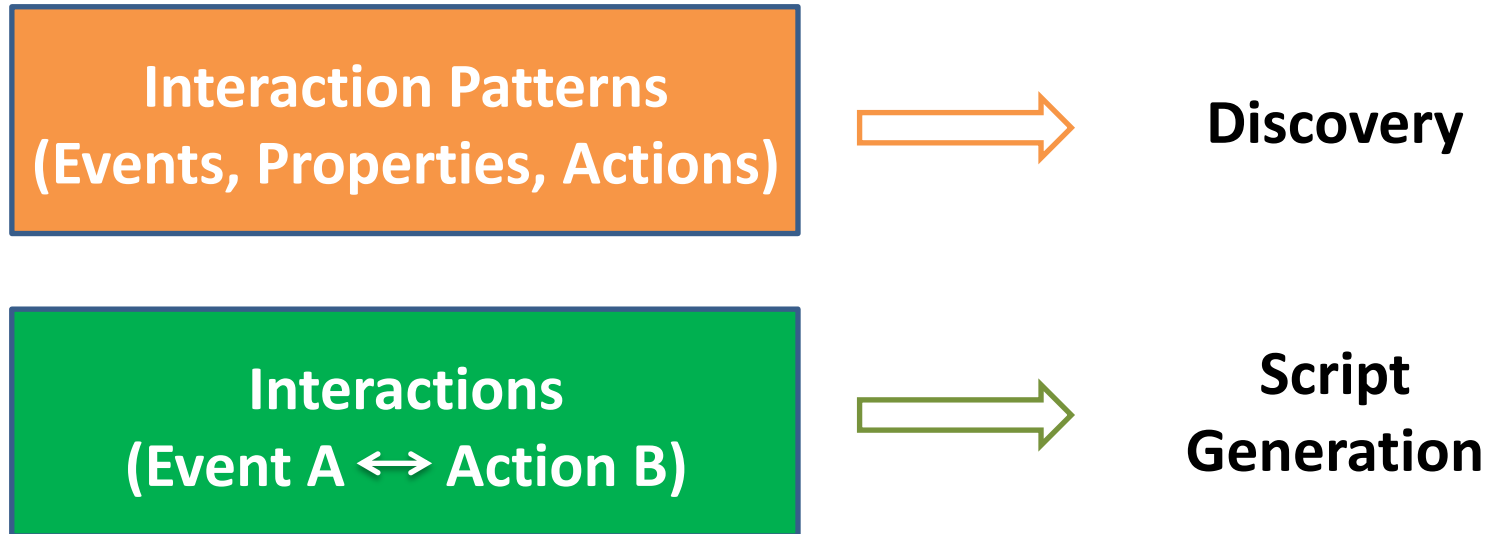
Motivation: Thing Discovery



- TD Directory:
 - Method: POST
 - URI: /td-lookup/sem
 - Payload: ?td a <http://iotschema.org/AirConditioner> .
- For users familiar with SPARQL and semantic models

Recipe

- How to easily create WoT applications based on application templates
- JSON-LD specification similar to Thing Description



Recipe Example

Recipe: Motion Detector Light Switch

Turn a light on when motion is detected in a room.

Ingredients

MotionStatus Property
TurnOn Action
TurnOff Action

Interactions

SUBSCRIBE MotionStatus
UPDATE TurnOn or
UPDATE TurnOff

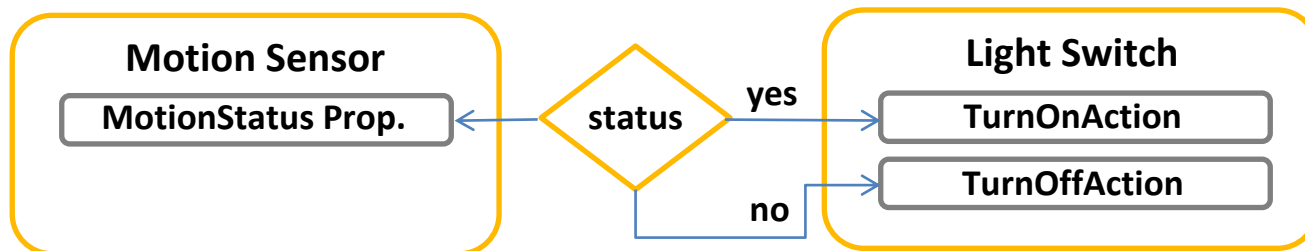



Recipe Example

Recipe: Motion Detector Light Switch

Turn a light on when motion is detected in a room.

Ingredients	Interactions
MotionStatus Property	SUBSCRIBE MotionStatus
TurnOn Action	UPDATE TurnOn or
TurnOff Action	UPDATE TurnOff

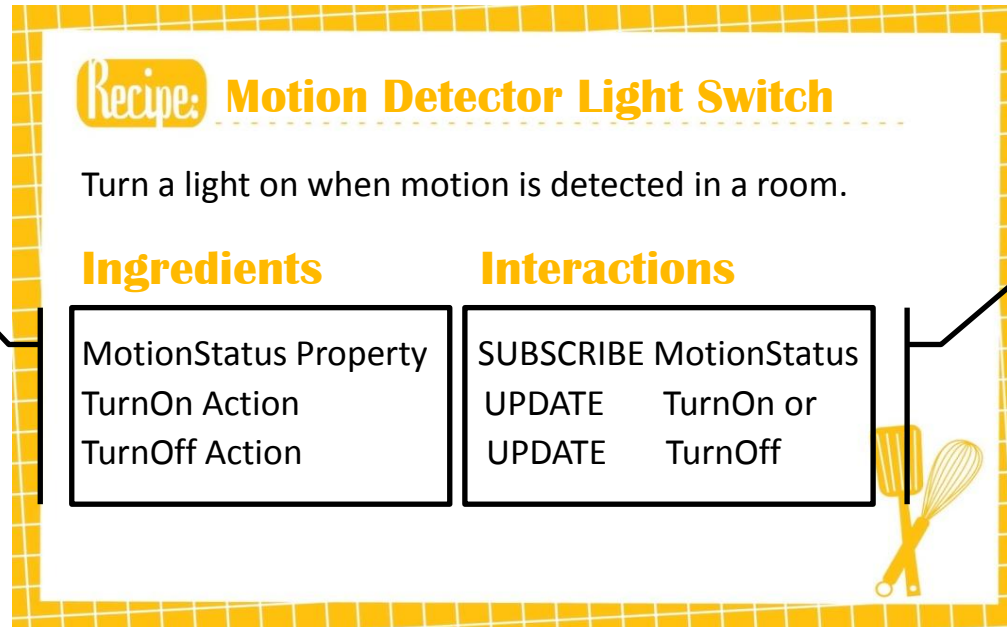


Recipe Example

Recipe: Motion Detector Light Switch

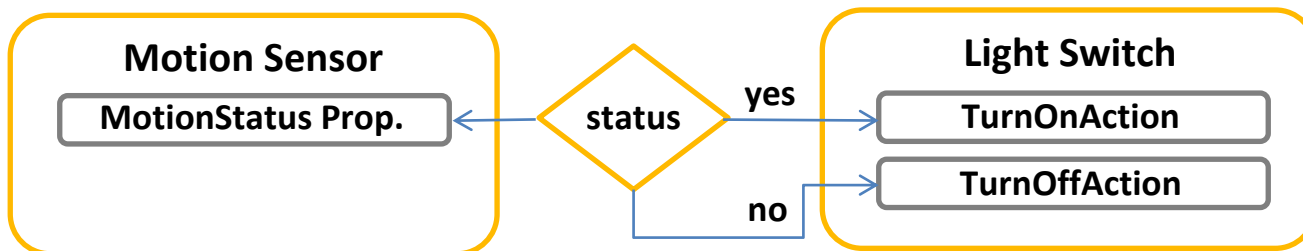
Turn a light on when motion is detected in a room.

Ingredients	Interactions
MotionStatus Property TurnOn Action TurnOff Action	SUBSCRIBE MotionStatus UPDATE TurnOn or UPDATE TurnOff



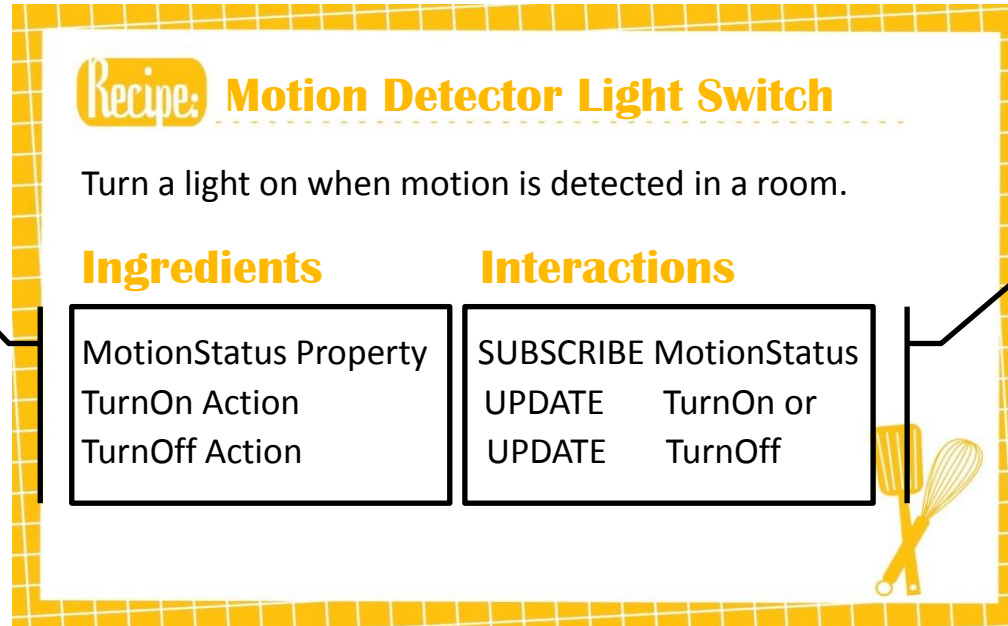
TD Interaction
Patterns &
iotschema.org
Capability

Implemented
with W3C WoT
Script API

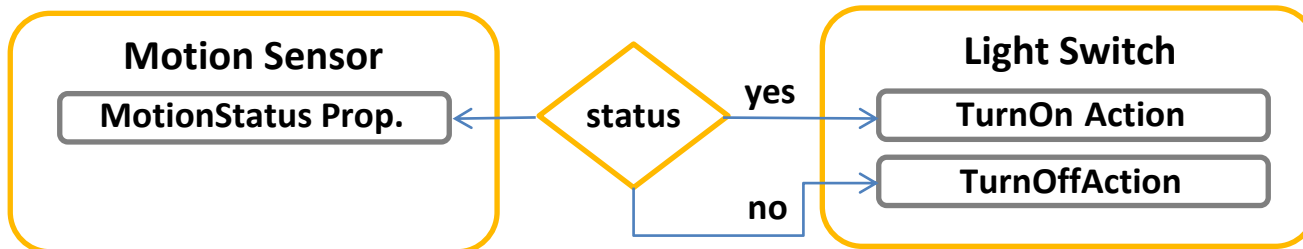


Recipe Example

TD Interaction
Patterns &
iotschema.org
Capability

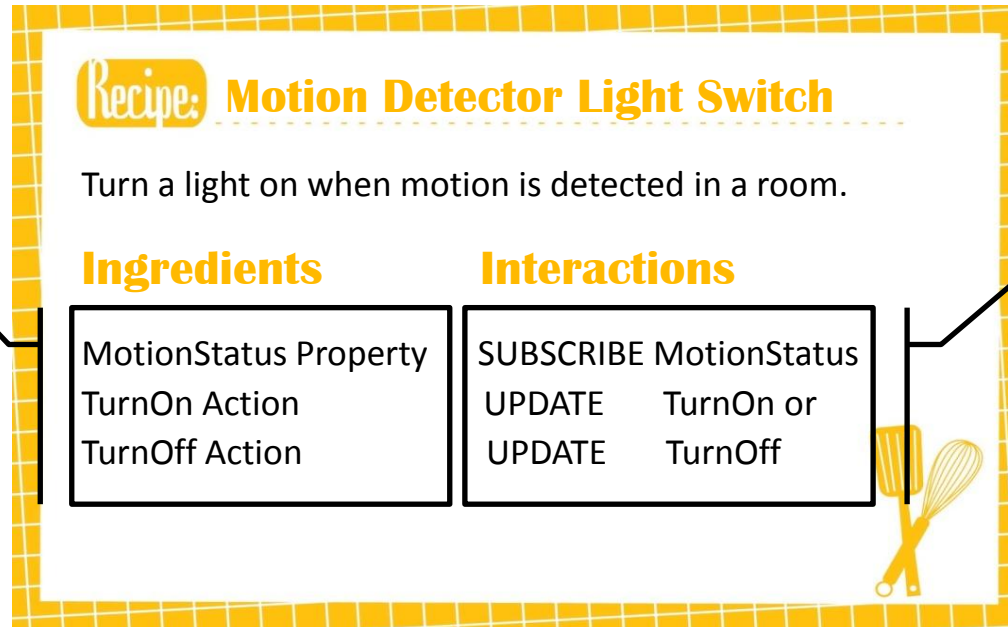


Discovery & Binding

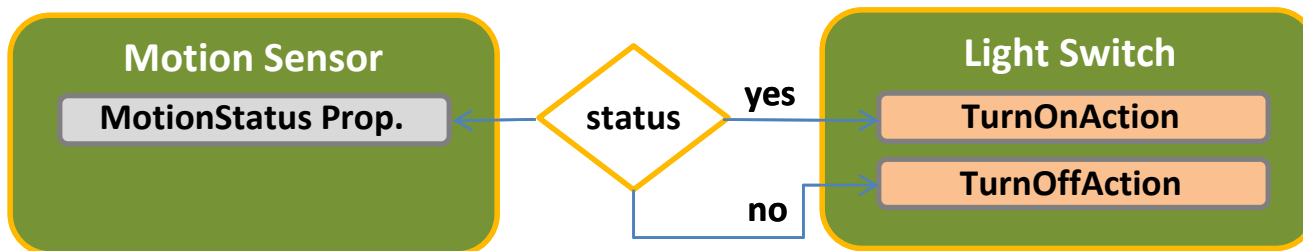


Recipe Example

TD Interaction
Patterns &
iotschema.org
Capability



Discovery & Binding



Example Recipe

```
["@context" :...
  "name" : "LightControlOnMotionDetection",
  "recipeId" : ["http://www.w3.org/ns/td-recipe/LightControlOnMotionDetection"],
  "comment": "The recipe for detecting the motion in a room and controls the lamp in a room.",
  "hasIngredient" : [
    {"@type" : "ThingIngredient",
      "thingIngredient" : [
        {"@type" : "http://iotschema.org/MotionDetected",
          "http://iotschema.org/Capability" :
            {"@type" : "http://iotschema.org/MotionControl"}}}],
        {"@type" : "ThingIngredient",
          "thingIngredient" : [
            {"@type" : "http://iotschema.org/BinarySwitch",
              "iot:Capability" :
                {"@type" : "http://iotschema.org/LightControl"}}
          ]}],
        "t2TInteraction":[
          {"@type" : "T2TInteraction",
            "t2TInteractsWith":[
              {"@type" : "http://iotschema.org/MotionDetected",
                "operation" : "Subscribe"},
              {"@type" : "http://iotschema.org/BinarySwitch",
                "operation" : "Update"}]]}]}
```

Interaction Patterns

iotschema Capability

Interactions

Tool: Recipe Workbench

The screenshot displays the SWAS IDE interface for a project named "Test_Project". The main workspace shows a recipe diagram with two components: "MotionSensor1" and "LYT8266".

MotionSensor1 (green box) contains the following services (from top to bottom):

- MotionDetected
- MotionType
- MotionDetectSubscribe - Update
- MotionType

LYT8266 (green box) contains the following services (from top to bottom):

- switch
- switch
- RGB
- Brightness
- RGB
- Brightness

A dashed line connects the "MotionDetectSubscribe - Update" service of "MotionSensor1" to the first "switch" service of "LYT8266".

The interface includes a "SWAS Navigator" on the left showing the project structure, a "Repository" pane with a search bar, and a "Palette" on the right with options like Select, Connect, Property, Action, Event, Script, Thing, and Service. The bottom status bar shows a red error message: "There is no connection to the repository. Check if the repository address is set properly". The taskbar at the bottom shows the system tray with the date 06.11.2017 and time 02:39.

Thank You!

Questions please...