WEB OF

THINGS

W3C Standards for the IoT

RIOT Summit, Berlin, Germany, 2016

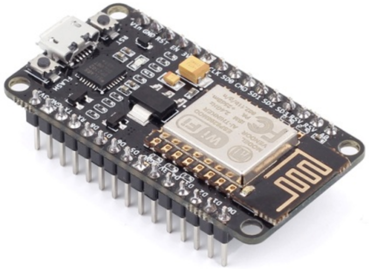
What is the Web of Things?

Application Layer

Internet of Things: **Connectivity**

IEEE 802.15.4

Ethernet



Wi-Fi

Bluetooth

LoRa

…

What is the Web of Things?

OCF

oneM2M

openHAB
empowering the smart home

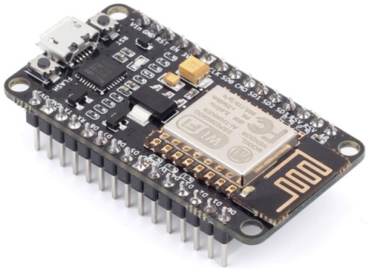
oma
Open Mobile Alliance
For a Connected World

OPC
FOUNDATION

Internet of Things: **Connectivity**

IEEE 802.15.4

Ethernet



Wi-Fi

Bluetooth

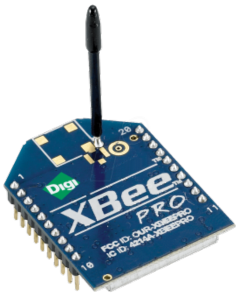
LoRa

…

What is the Web of Things?

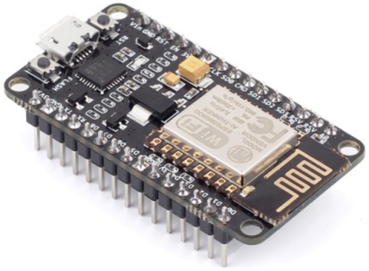
Web of Things: **Applications**

Internet of Things: **Connectivity**



IEEE 802.15.4

Ethernet



Wi-Fi

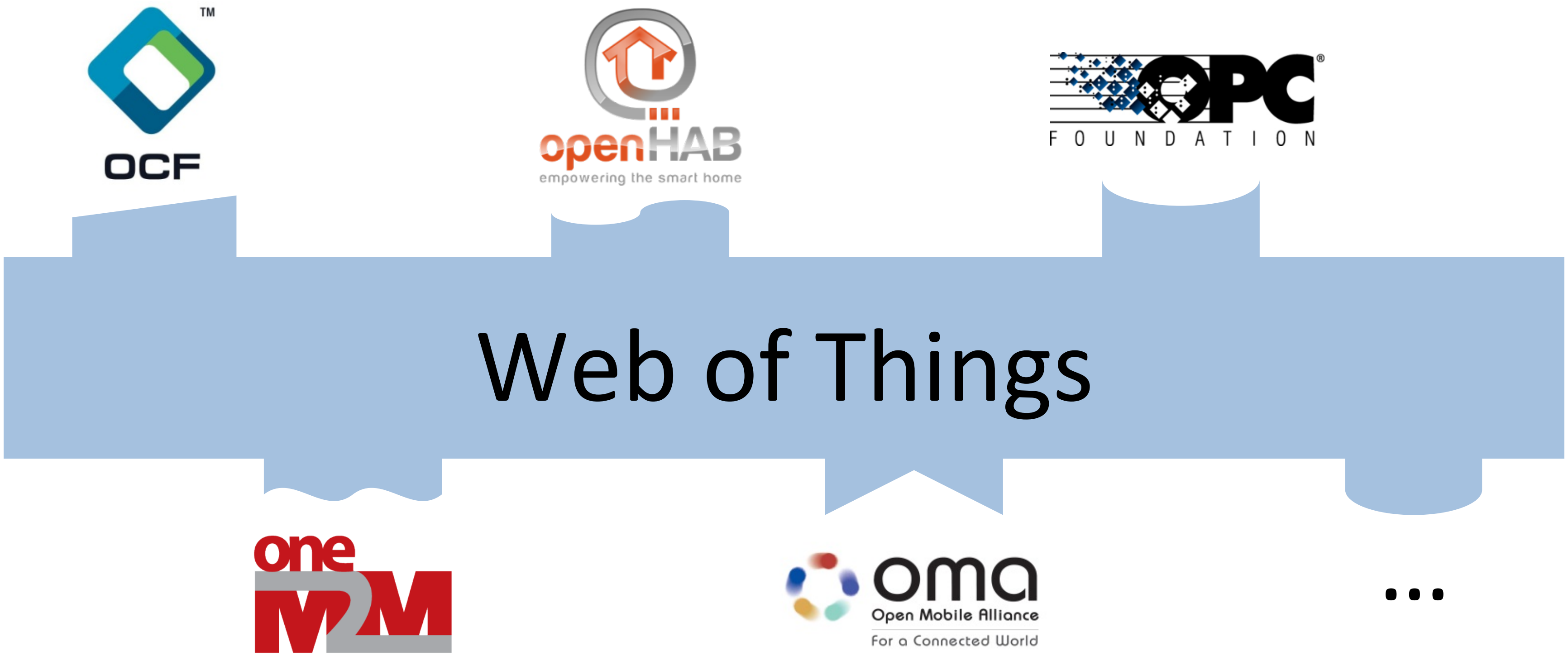
Bluetooth

LoRa

…

W3C WoT Mission

Not to be yet another standard



Web of Things

…

"interconnecting existing Internet of Things platforms and complementing available standards"

Need to Describe How to Interact with Things of Different IoT Platforms

http://w3c.github.io/wot/current-practices/wot-practices.html#sec-concepts

BUILDING BLOCKS

Goals and Challenges

- Goal: Web of Things Application Layer
  - need to program and deploy IoT applications like Web applications (outside embedded domain)
- Goal: Complement existing standards
  - need to describe heterogeneous Thing, communication, and security metadata
- Goal: Enable interoperability across platforms
  - need to support different protocols and communication patterns

Let’s start bottom-up …

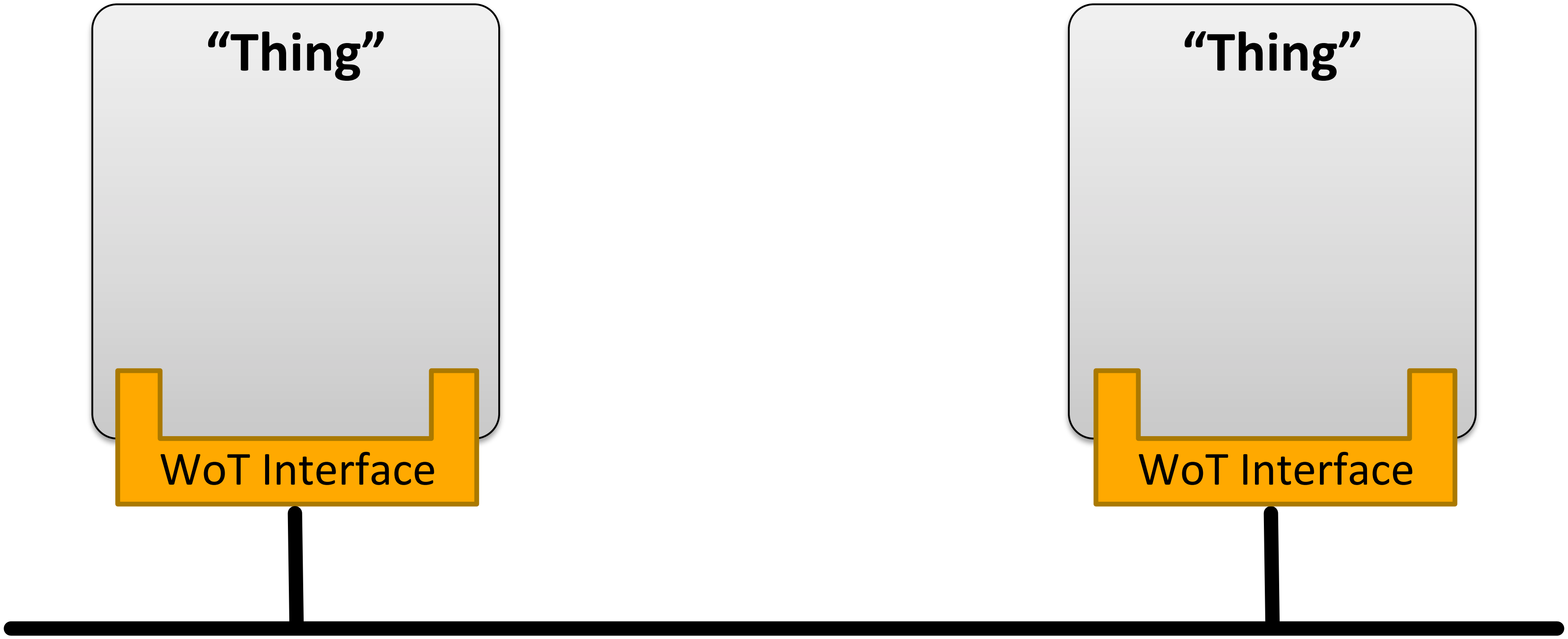
Need to Support Different Protocols and Communication Patterns

http://w3c.github.io/wot/current-practices/wot-practices.html#wot-interface

WoT INTERFACE

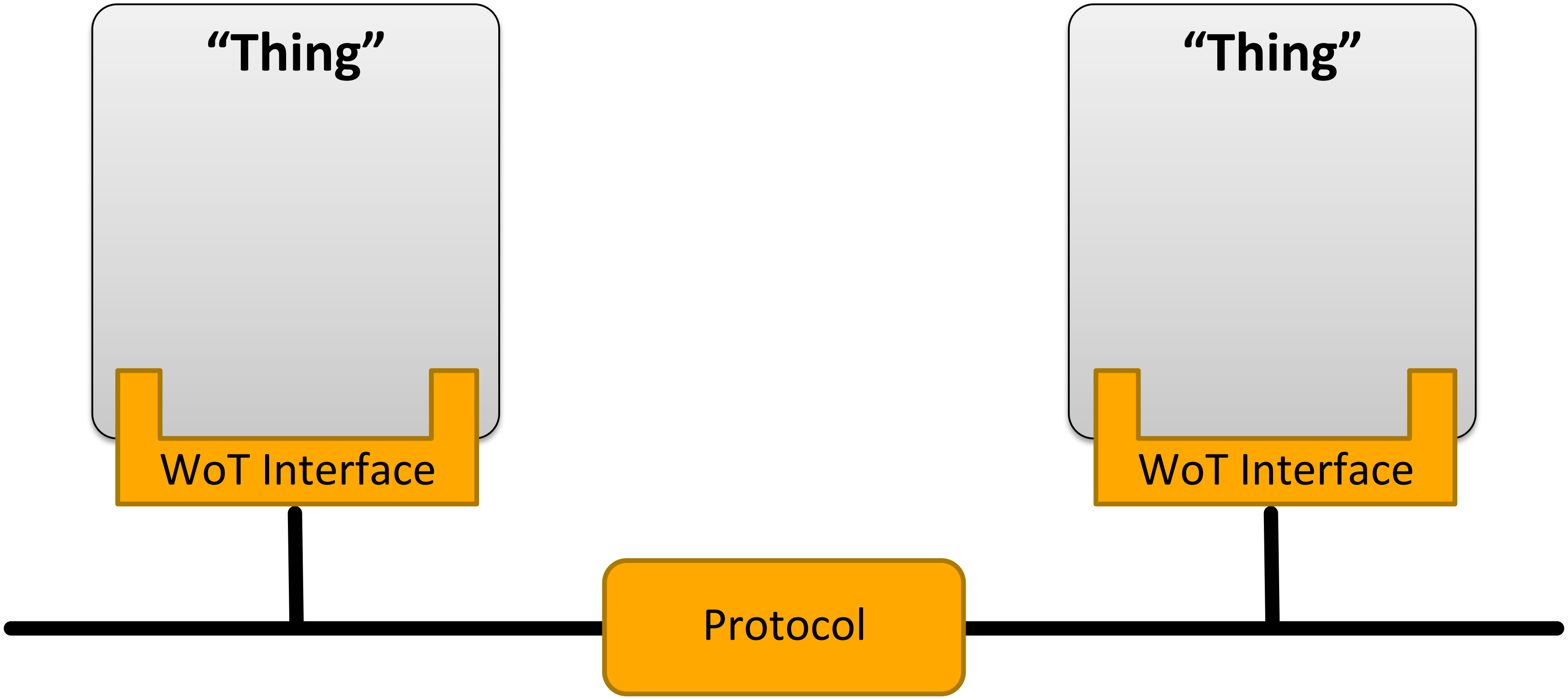
WoT Interface

- Interface of Things exposed to the network



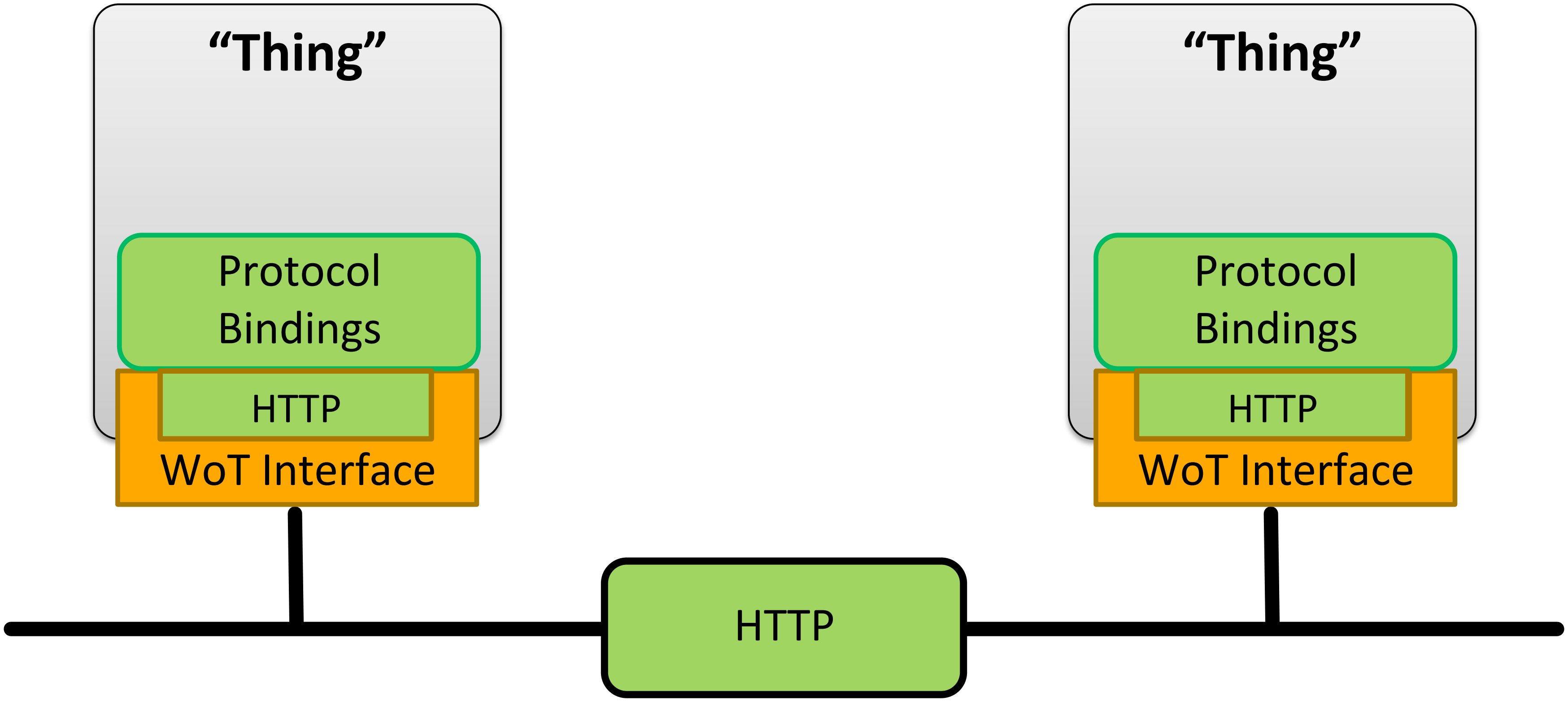
WoT Interface

- Interface of Things exposed to the network



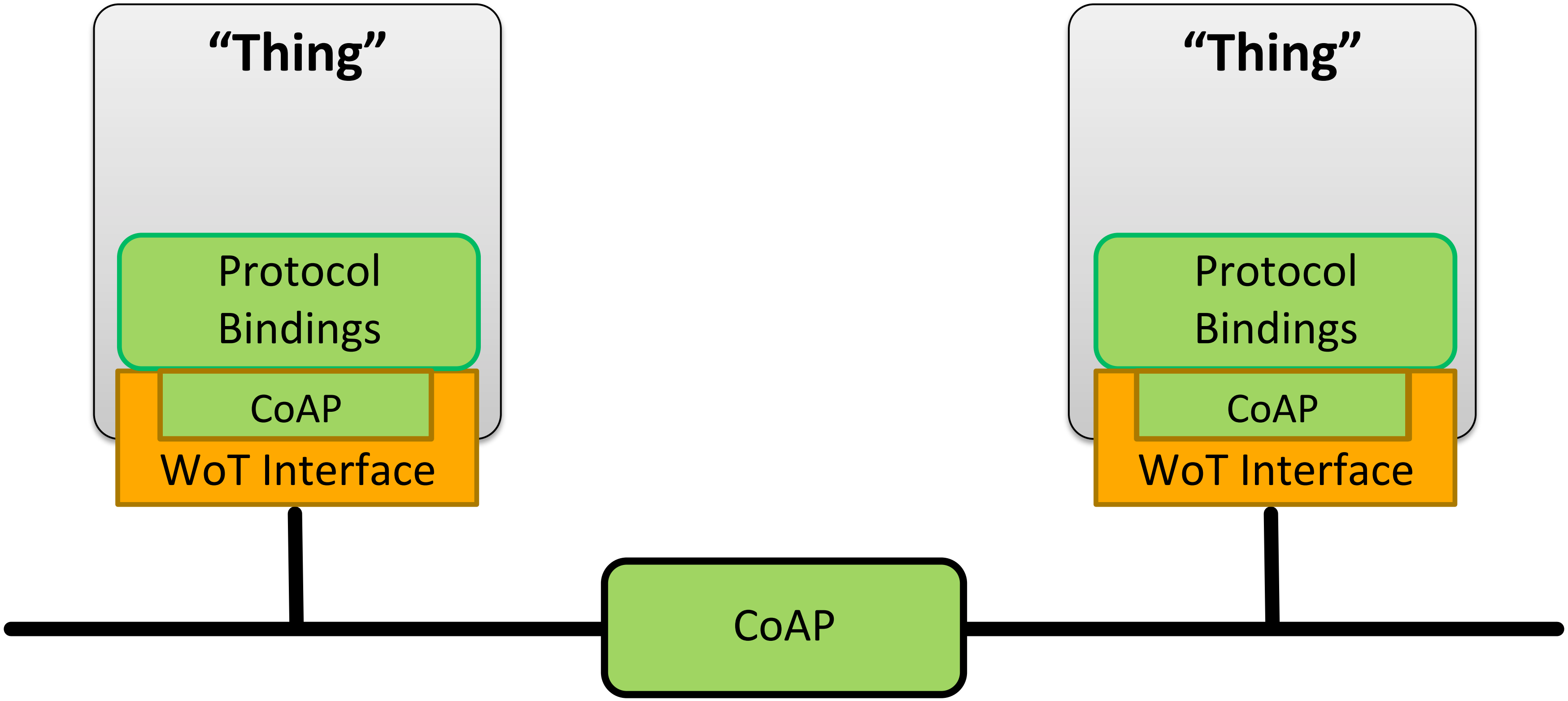
Protocol Bindings

- Various protocols can implement the interface



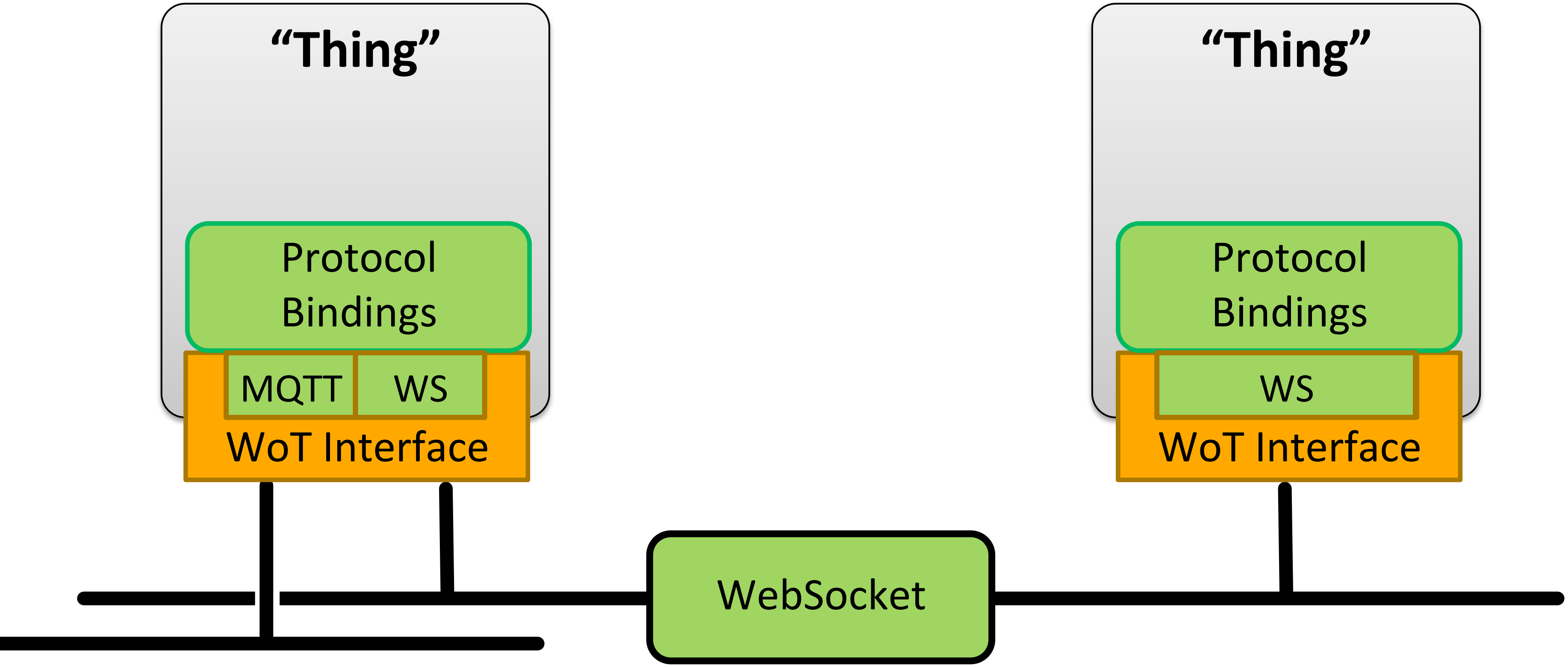
Protocol Bindings

- Various protocols can implement the interface



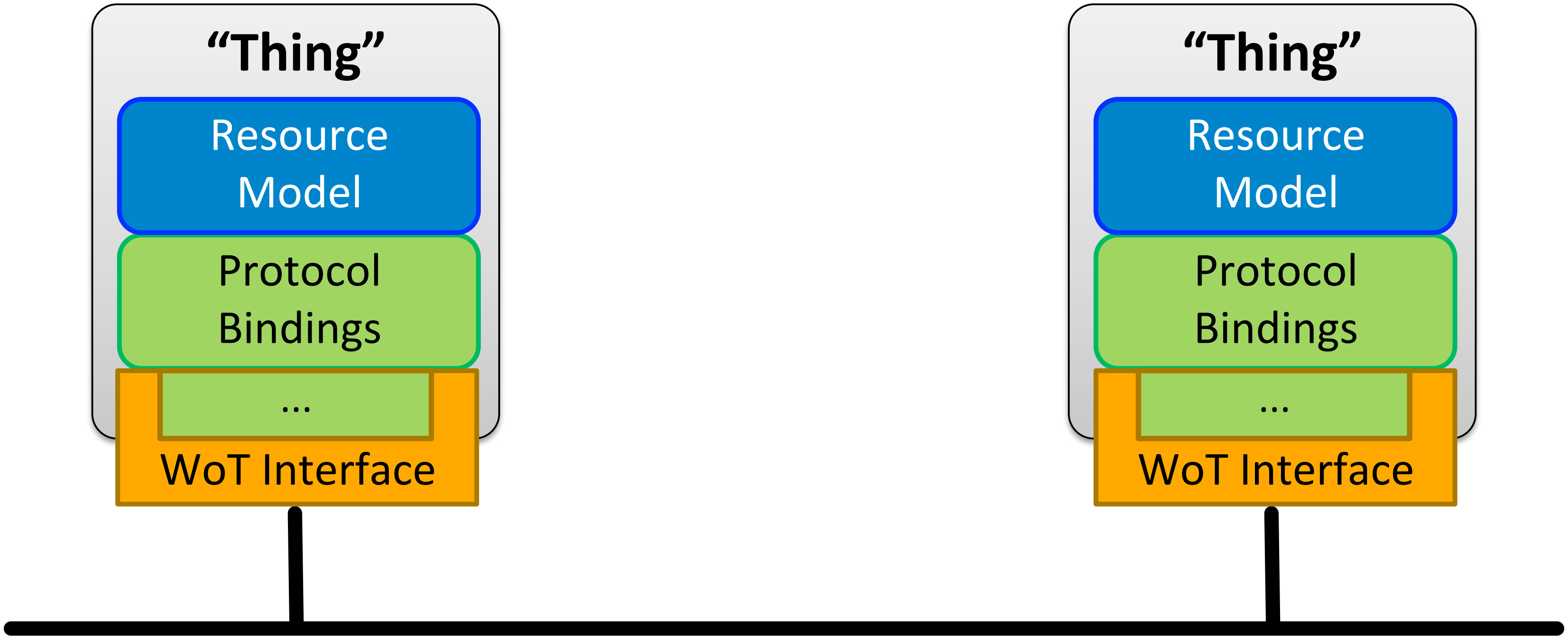
Protocol Bindings

- Multiple bindings per Thing possible



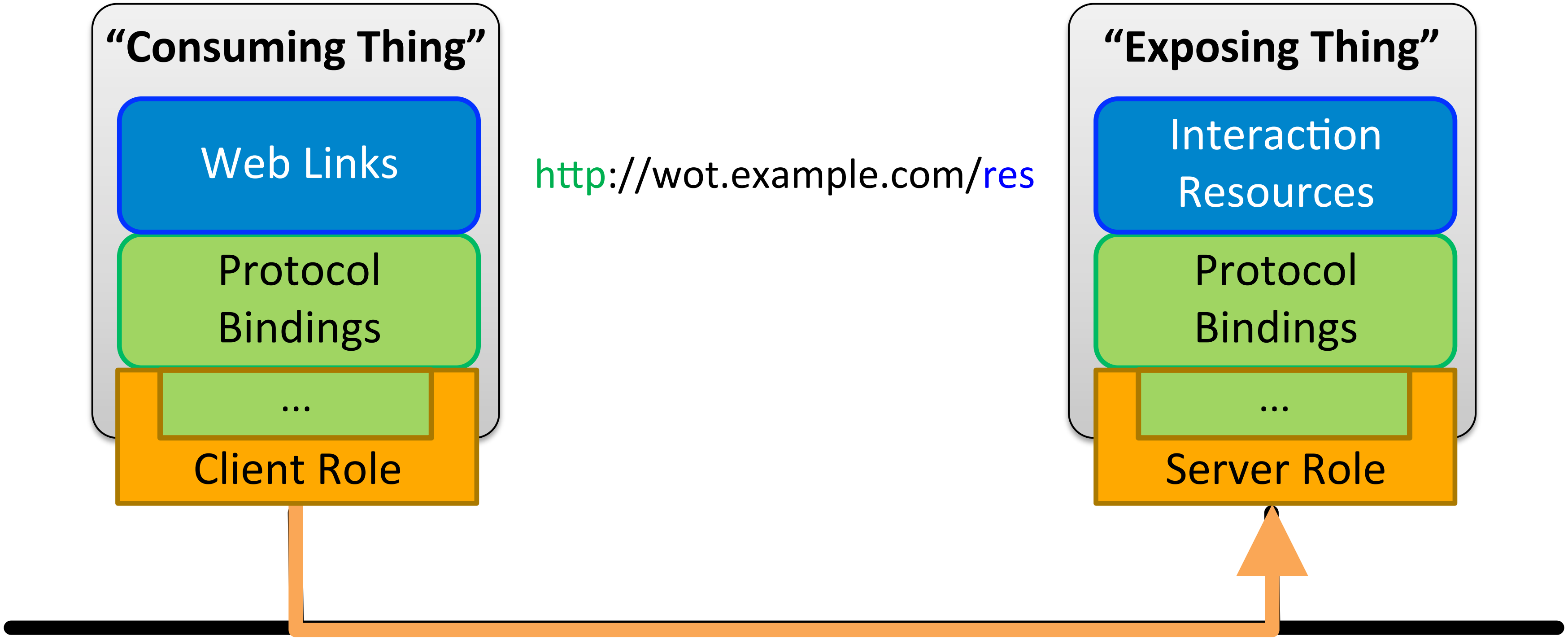
Resource Model

- Interaction points are Web resources



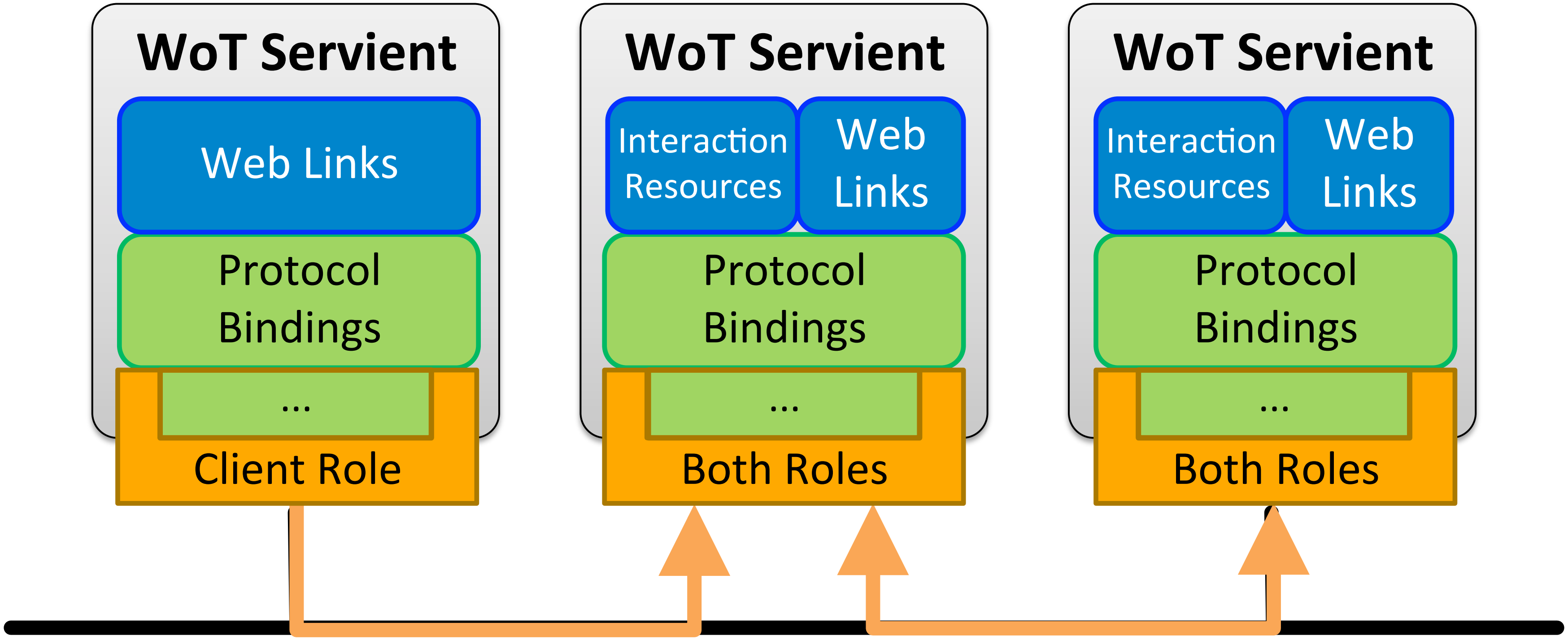
Interaction Role

- Consuming Things are in client role
- Exposing Things are in server role



Interaction Role

- Usually both roles at the same time → *Servient*



Need to Describe Heterogeneous Thing, Communication, and Security Metadata

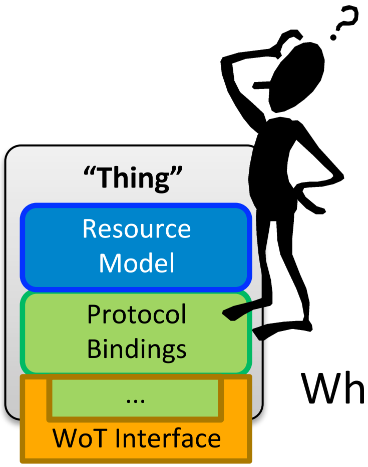
http://w3c.github.io/wot/current-practices/wot-practices.html#thing-description

THING DESCRIPTION

How to Interact with Things?

Who are you?

What kind of data do you serve?



How can I interact with you?

What kind of functions do you have?

What kind of protocols/encodings do you support?

Are there some security constrains?

→ W3C Thing Description

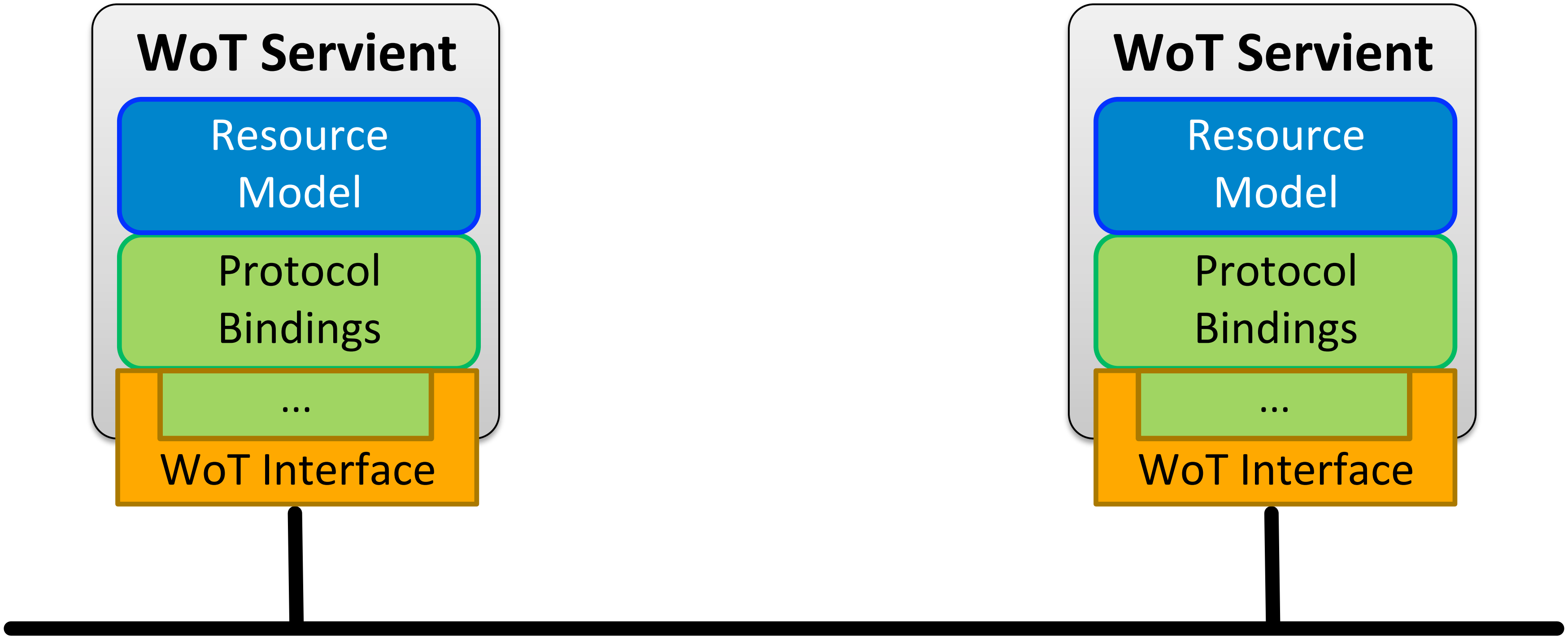
Semantic Description

- Reach semantic interoperability through Linked Data vocabularies
  - each individual Things is described by a set of triples
  - no fixed classes or types, but atomic vocabulary

- W3C Thing Description
  - vocabulary for thing, communication, and security metadata
  - extensible with domain-specific vocabulary
  - can augment existing Things (e.g., host externally on Web)

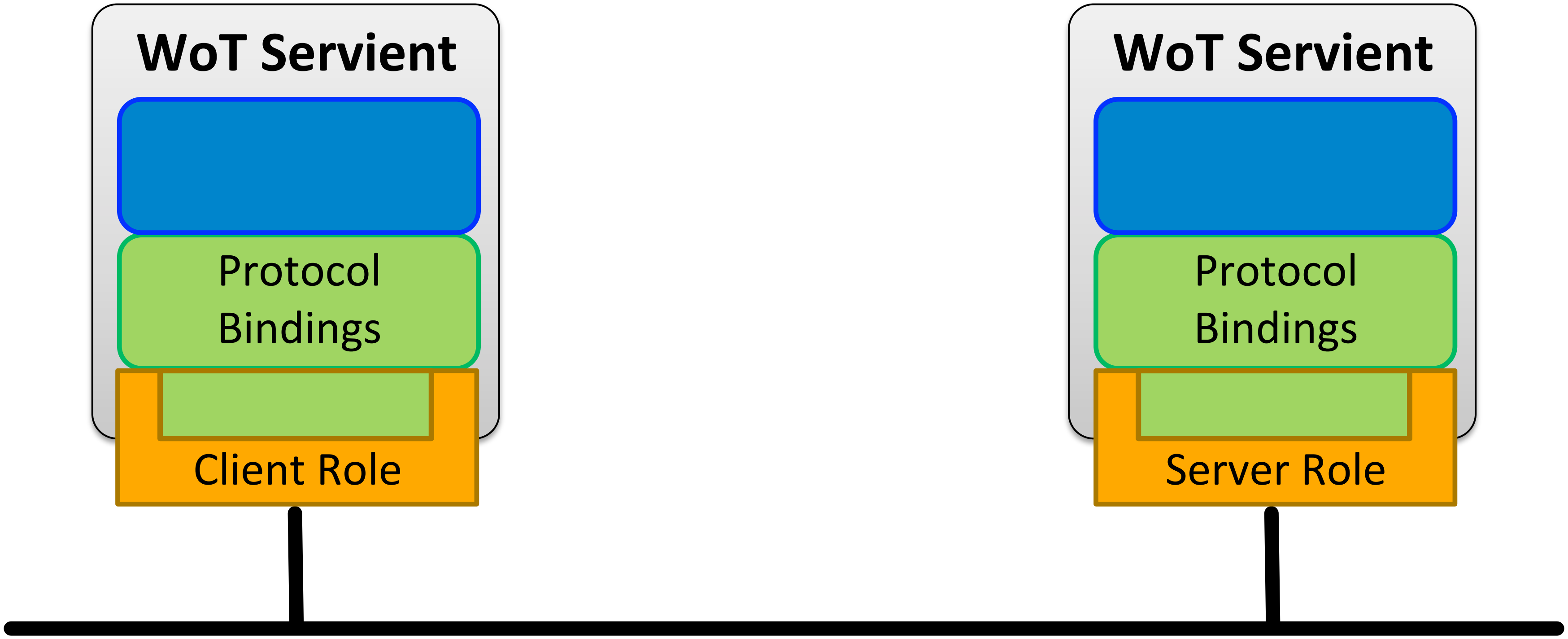
Thing Description (TD)

- Describes Thing metadata and interactions



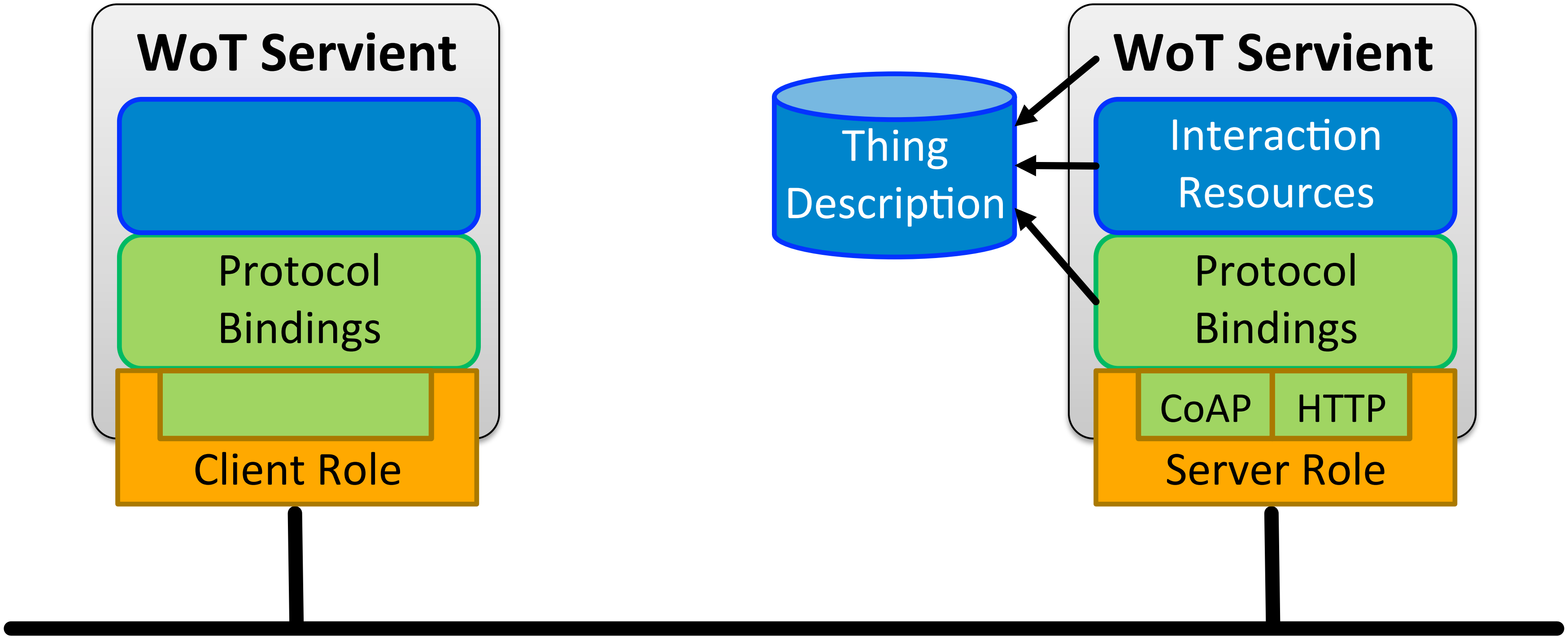
Thing Description (TD)

- Consuming Things are in client role
- Exposing Things are in server role



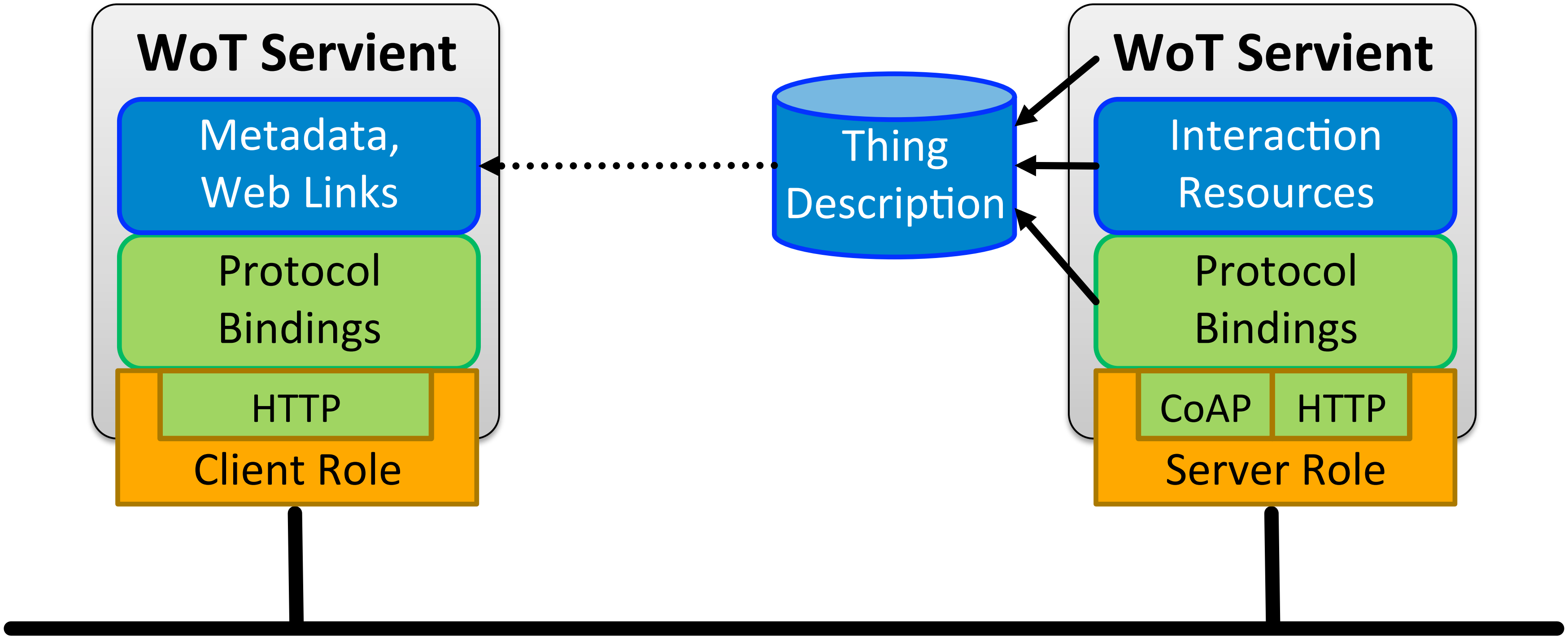
Thing Description (TD)

- Exposing Things provide Thing Description



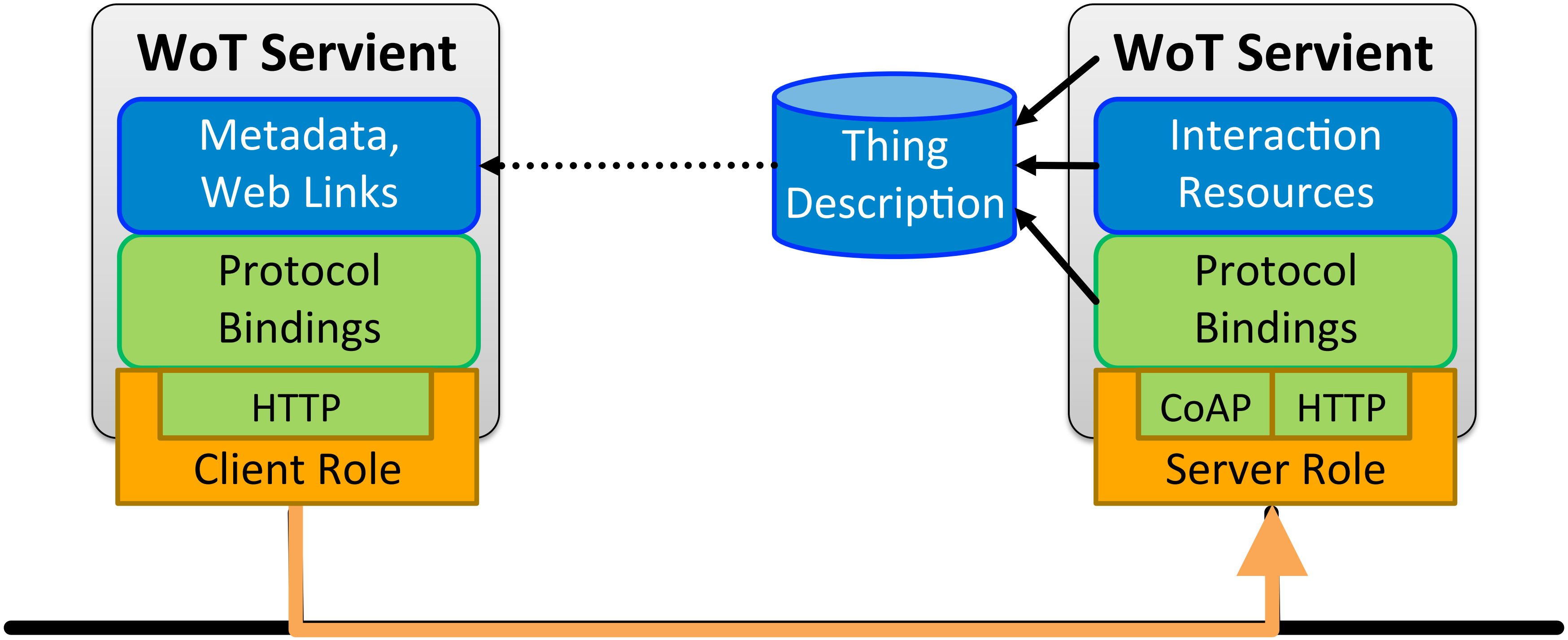
Thing Description (TD)

- Consuming Things learn WoT Interface from TD



Thing Description (TD)

- Bindings generate messages according to TD



Thing Description (TD)

- Default serialization is JSON-LD
  - convenient for development time
  - based on well established JSON format
  - `"@context"` field defines vocabularies
  - different implementations and tools available; http://json-ld.org/
- Other, also concise serializations possible
  - XML, EXI, CBOR, HDT, …

TD Example

```
{
  "@context": [
    "http://w3c.github.io/wot/w3c-wot-td-context.jsonld",
    { "actuator": "http://example.org/actuator#" }
  ],

  "@type": "Thing",
  "name": "MyLEDThing",

  "uris": [
    "coap://myled.example.com:5683/",
    "http://mything.example.com:8080/myled/"
  ],

  "encodings": ["JSON", "EXI"],
  "security": {
    "cat": "token:jwt",
    "alg": "HS256",
    "as": "https://authority-issuing.example.org"
  },

  "properties": [
"properties": [
  {
    "@type": "actuator:onOffStatus",
    "name": "status",
    "valueType": { "type": "boolean" },
    "writable": true,
    "hrefs": [ "pwr", "status" ]
  }
],
"actions": [
  {
    "@type": "actuator:fadeIn",
    "name": "fadeIn",
    "inputData": {
      "valueType": { "type": "integer" },
      "actuator:unit": "actuator:ms"
    },
    "hrefs": [ "in", "led/in" ]
  },
  {
    "@type": "actuator:fadeOut",
    "name": "fadeOut",
    "inputData": {
      "valueType": { "type": "integer" },
      "actuator:unit": "actuator:ms"
    },
    "hrefs": [ "out", "led/out" ]
  }
```

Property

Action

Action

Interaction resources

Events (work in progress)

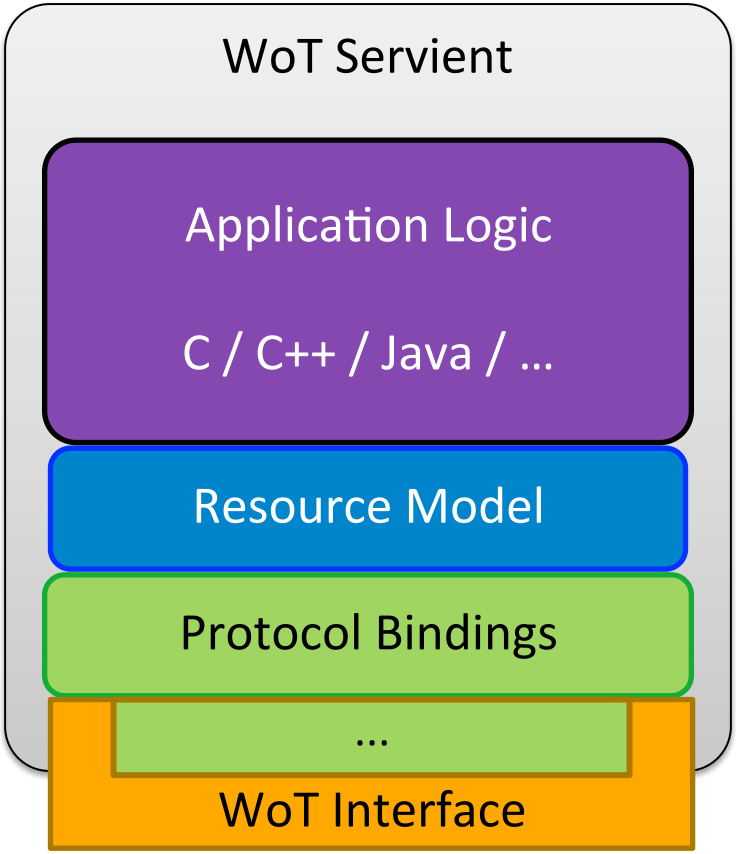
Need to Program and Deploy IoT Applications like Web Applications

http://w3c.github.io/wot/current-practices/wot-practices.html#scripting-api

SCRIPTING API

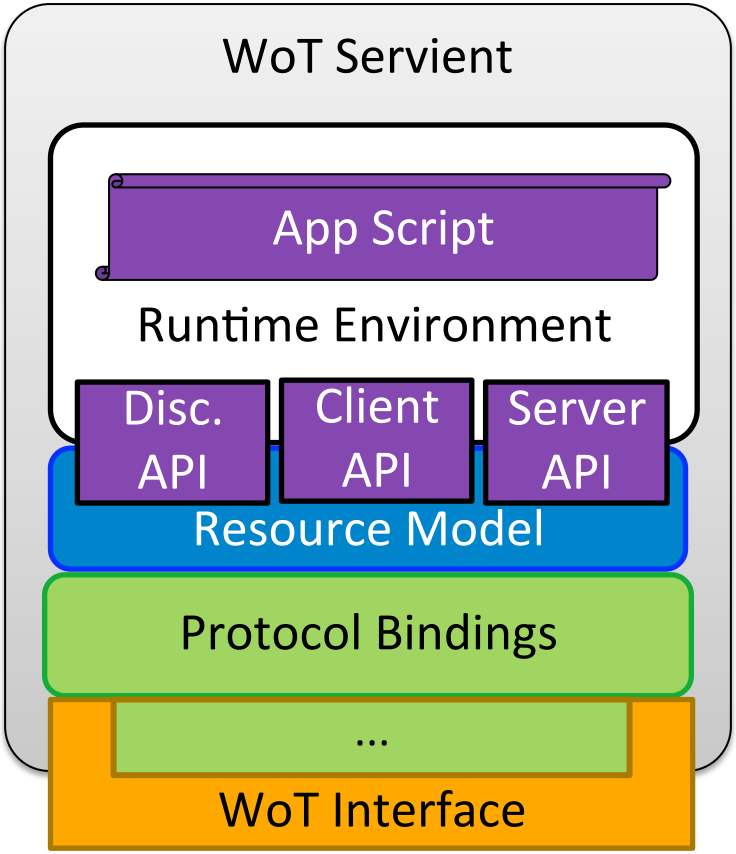
Without Scripting API

- Application logic often implemented natively



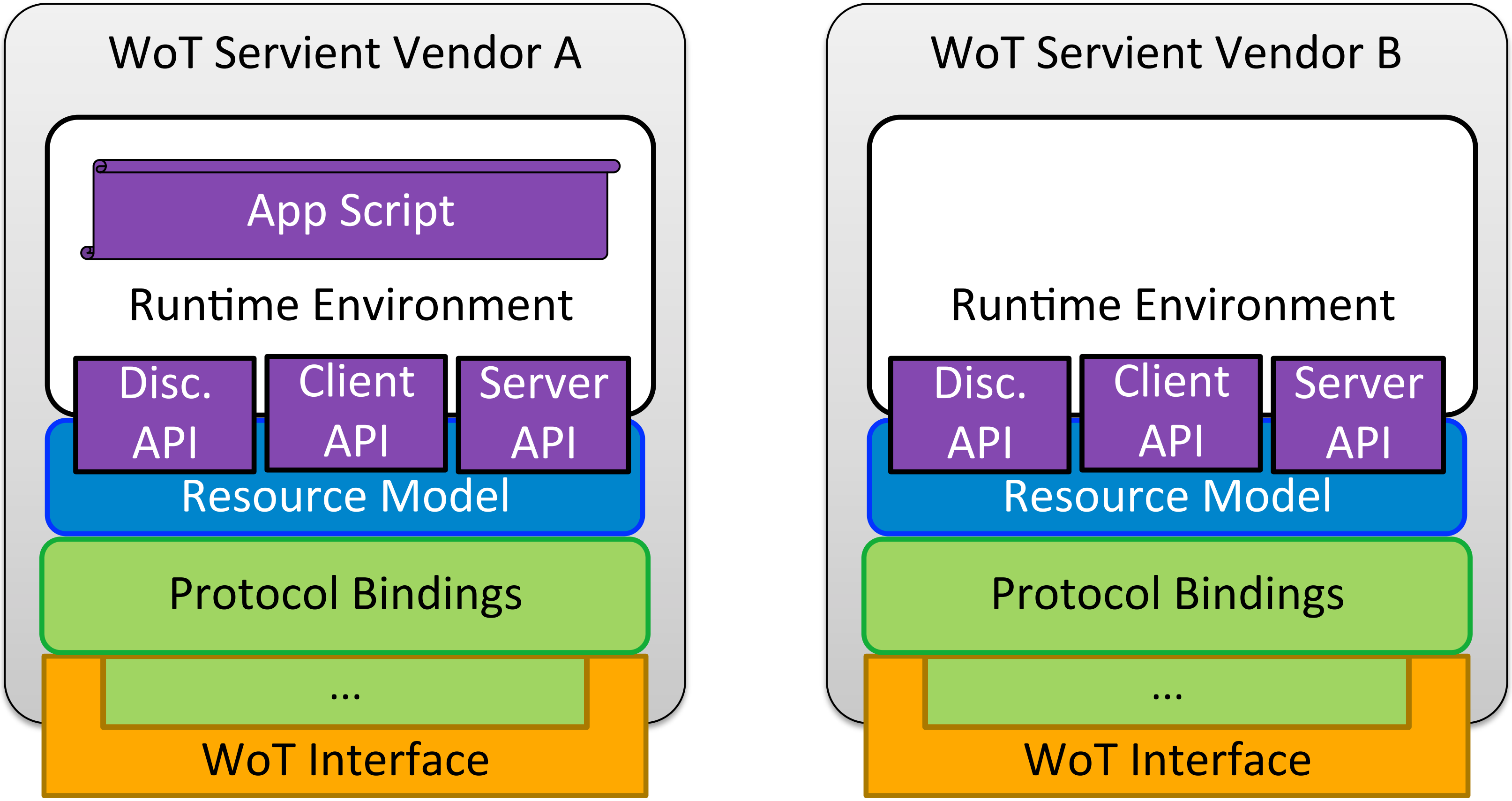
Scripting API

- Web-like development and deployment



Scripting API

- Common runtime enables portable apps



Script Example (Expose Thing)

```
// create software object to represent local Thing
WoT.newThing("counter")
    .then(function(thing) {
        thing
            // programmatically add interactions
            .addProperty("count", {"type": "integer"})
            .addAction("increment")
            .onInvokeAction("increment", function() {
                console.log("incrementing counter");
                // persistent state is managed by runtime environment
                var value = thing.getProperty("count") + 1;
                thing.setProperty("count", value);
                return value;
            })
        // initialize state (no builder pattern anymore)
        thing.setProperty("count", 0);
    })
    ._catch(console.err);
```

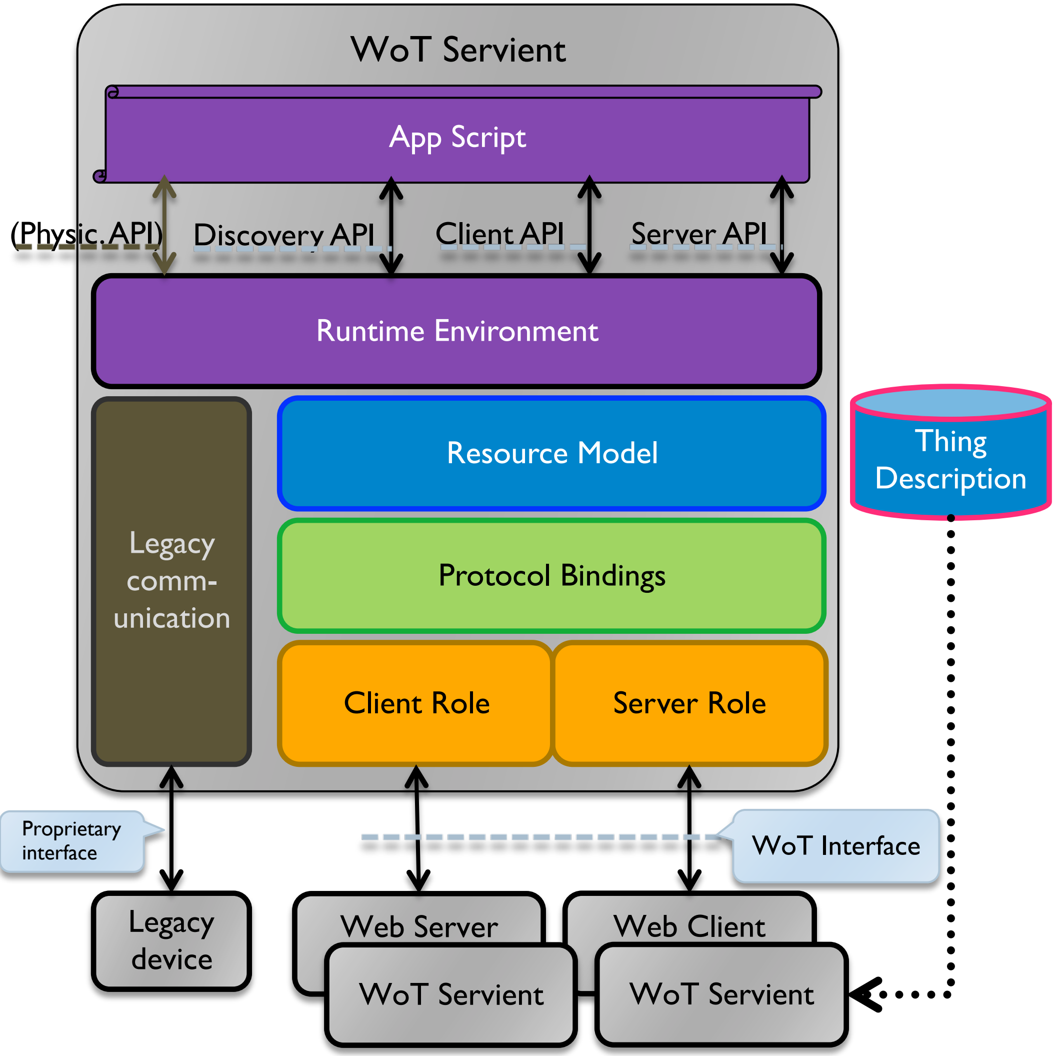
Script Example (Consume Thing)

```
// create software object to represent remote Thing based on TD URI
WoT.consumeDescriptionUri("http://servient.example.com/things/counter")
    // use promise to handle asynchronous creation
    .then(function(counter) {
        counter
            // invoke an Action without arguments
            .invokeAction("increment", {})
                // which is an asynchronous call -> promise
                .then(function() {
                    console.log("incremented");
                    counter
                        // read Property (async.) to confirm increment
                        .getProperty("count").then(function(count) {
                            console.log("new count state is " + count);
                        });
                })._catch(console.error);
    })
    ._catch(console.error);
```

W3C Web of Things

SUMMARY

Thing Implementation: **WoT Servient**



Application Logic:

Can consume remote Things through the Client API, local hardware and connected legacy devices through a Physical API (t.b.d.), and expose Things through the Server API. To allow portable app scripts, the Servient must povide a runtime environment.

Resource Model:

Provides a common abstraction with uniform interface across the different protocols. Like the Web, it allows to identify and address interaction points through URIs.

Thing Description (TD):

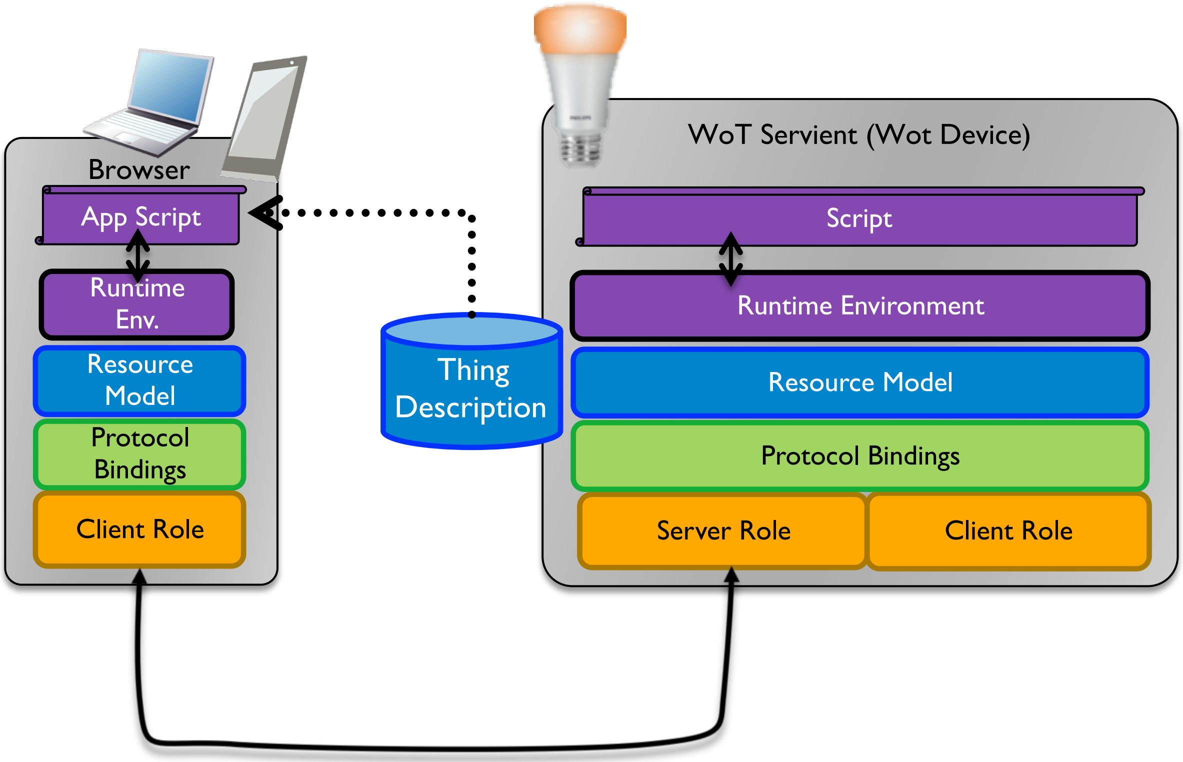
Declares WoT Interface for interaction and provides (semantic) metadata for the Thing. TD is used by WoT clients to instantiate local software object of the Thing.

Protocol Binding:

Converts abstract interactions with Things to different protocols using the information from TD.

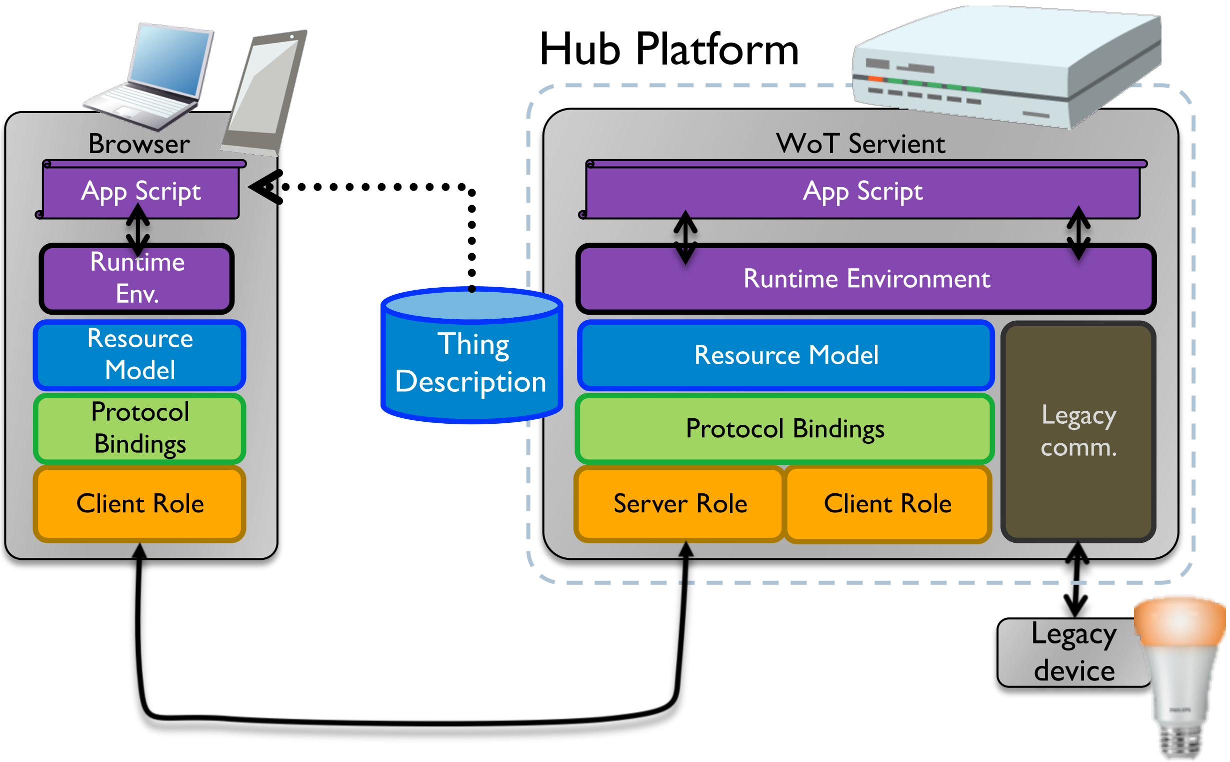
WoT Servient on Thing Itself

- Native WoT Things host a Servient directly
- TD is provided by Thing or supporting host on the Web



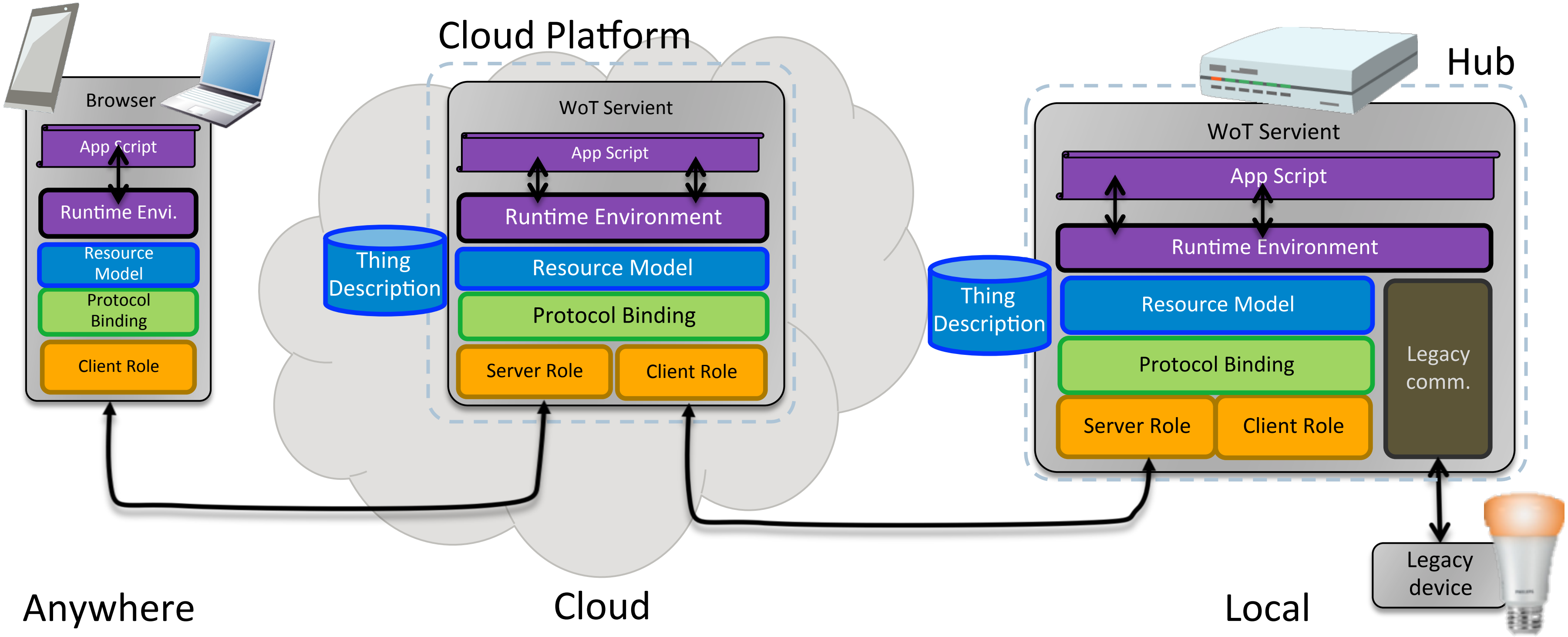
WoT Servient on Integration Hub

- WoT Servients can run on hubs (e.g., smartphone, gateway)
- Multiple Servients can be instantiated through sandboxed apps
- Apps can act as agents/proxies for legacy devices



WoT Servient in the Cloud

- A cloud mirror (device shadow) enables scalable remote access
- Is synchronized with local Servient
- Can forward interactions and cache data



Online Resources

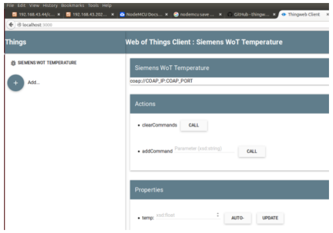
- Interest Group
  - https://www.w3.org/WoT/IG/
  - https://lists.w3.org/Archives/Public/public-wot-ig/ (subscribe to mailing list)
- Documents (for implementers)
  - http://w3c.github.io/wot/architecture/wot-architecture.html
  - http://w3c.github.io/wot/current-practices/wot-practices.html (living document)
    Beijing 2016 Release:
    http://w3c.github.io/wot/current-practices/wot-practices-beijing-2016.html
- GitHub (documents and proposals)
  - https://github.com/w3c/wot
- Wiki (organizational information: WebConf calls, Face-to-Face meetings, …)
  - https://www.w3.org/WoT/IG/wiki/Main_Page
- WoT Projects (implementing WoT Current Practices)
  - https://github.com/thingweb/
  - https://github.com/mkovatsc/wot-demo-devices
  - Please add yours!

W3C WoT F2F Beijing 2016

PLUGFEST

Scenario 1: Hello WoT

TD Web UI for human interaction



Things

Web of Things Client : Siemens WoT Temperature

SIEMENS WOT TEMPERATURE

Add...

Siemens WoT Temperature

coap://COAP_IP:COAP_PORT

Actions

- clearCommands CALL
- addCommand Parameter (xsd:string) CALL

Properties

- temp: xsd:float AUTO- UPDATE

/voteTooHot

Servient platform with scripted apps

/on

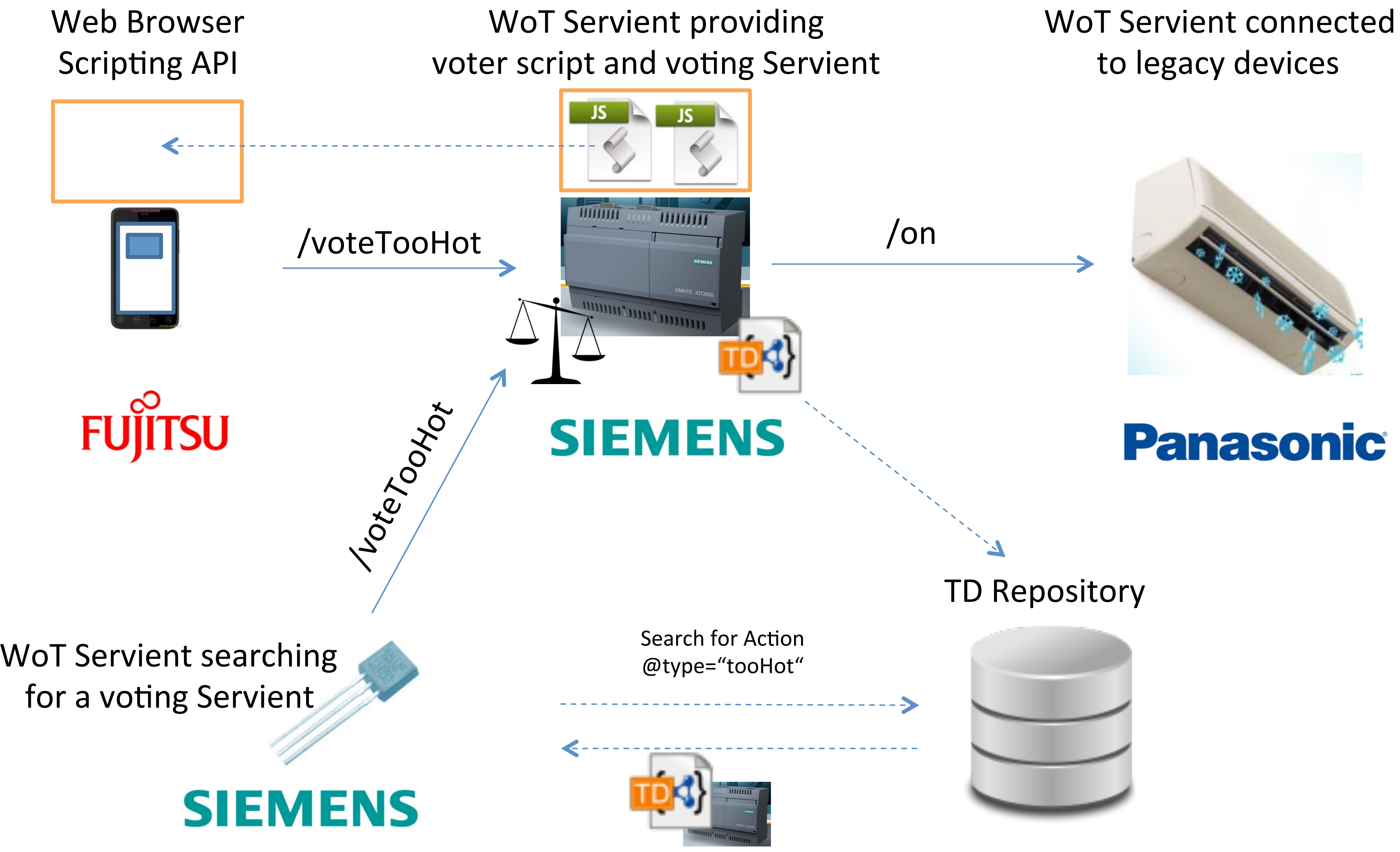
Servient connected to legacy devices

Open Source

SIEMENS

Panasonic

Scenario 2: Full WoT



Web Browser Scripting API

WoT Servient providing voter script and voting Servient

WoT Servient connected to legacy devices

/voteTooHot

/on

/voteTooHot

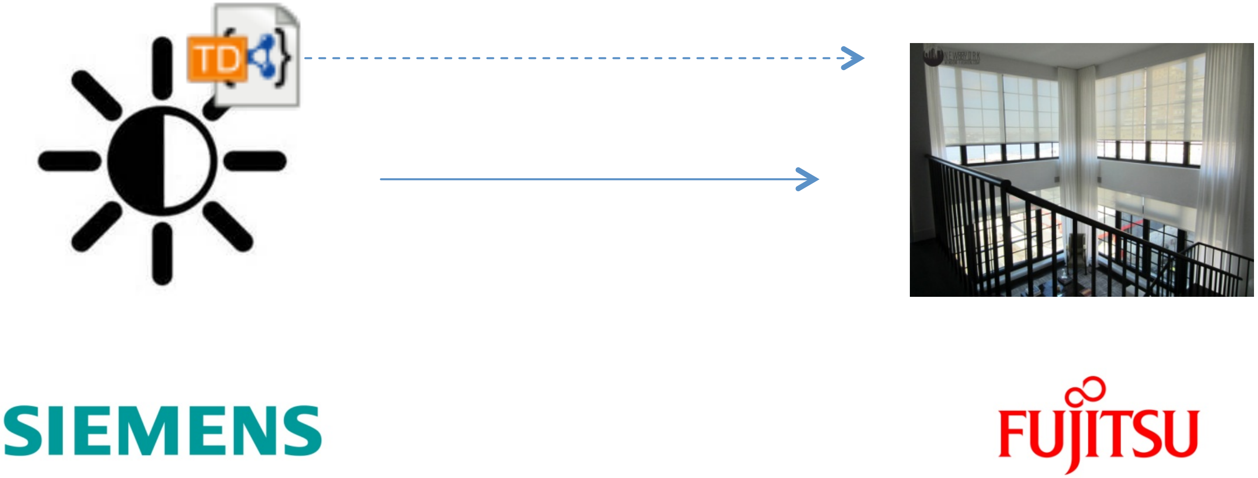
TD Repository

Search for Action @type=“tooHot“

WoT Servient searching for a voting Servient

Scenario 3: Mini Automation

Consume brightness sensor
to control curtain



PlugFest Online Resources

- Current Practices (Beijing Release)
  - http://w3c.github.io/wot/current-practices/wot-practices-beijing-2016.html
- Organization Wiki
  - https://www.w3.org/WoT/IG/wiki/F2F_meeting,_July_2016,_China,_Beijing#PlugFest
- Test Cases
  - https://github.com/w3c/wot/blob/master/plugfest/2016-beijing/plugfest-test-cases-beijing-2016.md
- Report Template
  - https://github.com/w3c/wot/blob/master/plugfest/2016-beijing/TestCaseCoverage.xlsx (t.b.d.)