W3C WoT Current Practices

IRTF T2TRG Meeting, San Jose, CA, USA, 2016

Matthias Kovatsch (kovatsch@inf.ethz.ch)

Current Practices Document

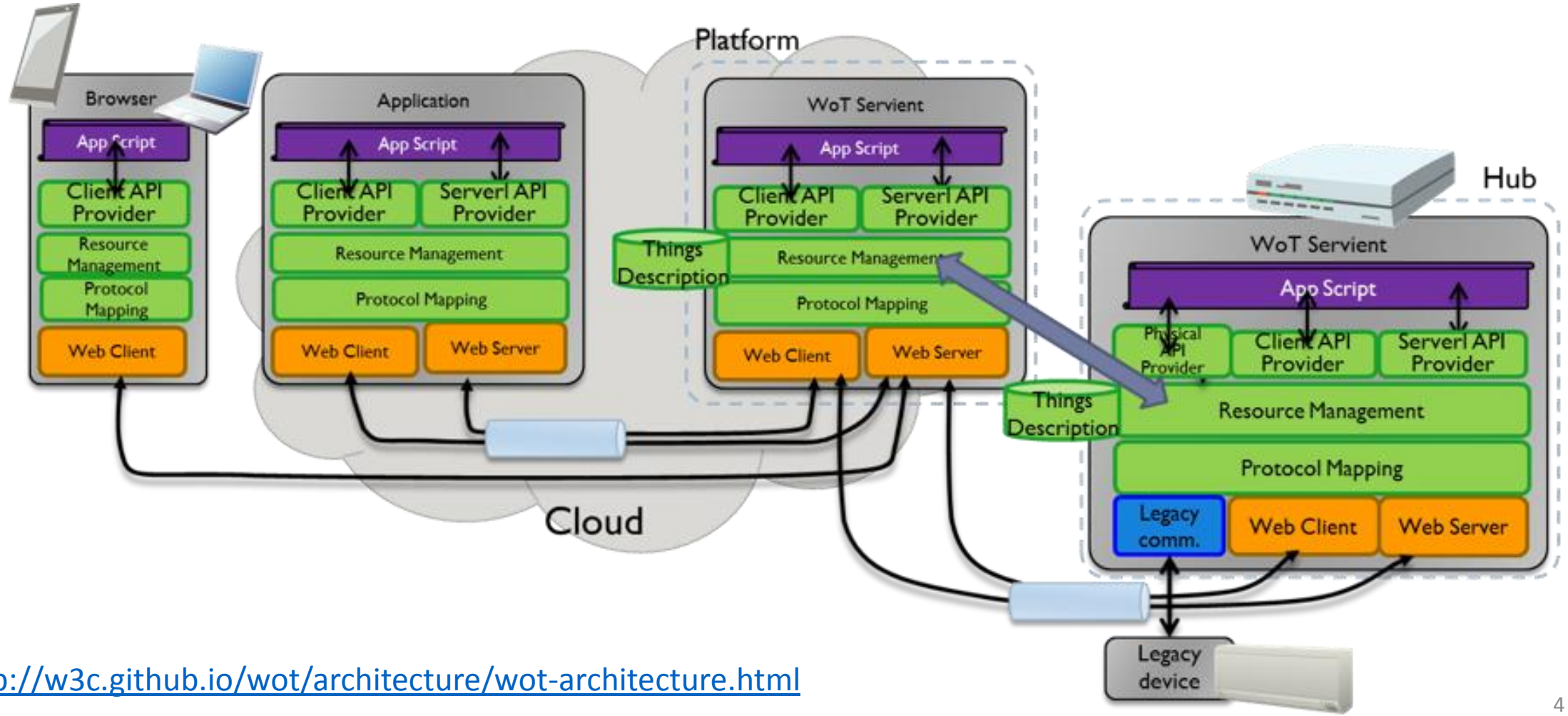
http://w3c.github.io/wot/current-practices/wot-practices.html

- Collect W3C WoT building blocks in a single document
- Try to reflect latest discussion and Plugfest results
- Getting started guide for prototype implementations
- Possible starting point for W3C WoT Working Group
- Note
  - Work in progress
  - Not a normative specification

Current Practices

- Assumptions
  - Web resource model with URIs, but multiple protocol bindings
  - Things are represented by “Internet application endpoints”/“Web components”
  - Hosted directly on physical devices or gateways/hubs or the cloud
  - Can be client, server, or both (“servient”)
- Thing Description (TD)
- WoT API
- Scripting API
- Plugfests

WoT Architecture is Flexible



http://w3c.github.io/wot/architecture/wot-architecture.html

Thing Description (TD)

- Semantic metadata
  - Human readable name, descriptions, location, etc.
  - Can change in different stages of the life cycle
- Communication metadata
  - Protocols (scheme part of base URIs)
  - Encodings (serialization formats)
- Security metadata
  - Prerequisites to access things/resources
  - Protection of the TD itself
- Interactions
  - Properties
  - Actions
  - Events

TD Sample

```
{ "@context": ["http://w3c.github.io/wot/w3c-wot-td-context.jsonld",
               {"ex":"http://example.org/application-specific#"}]
  "@type": "Thing",
  "name": "MyLED",
  "uri": ["coap://www.myled.com:5683/", "http://www.myled.com:8080/myled/"],
  "encodings": ["JSON"],
  "security": {"cat":"token:jwt", "alg":"HS256", "as":"https://authority.example.org"},
  "properties": [
    { "@type": "ex:StatusProperty",
      "name": "My status",
      "href": ["status", "properties/status"],
      "valueType": "xsd:boolean",
      "writable": false
    } ],
  "actions": [
    { "@type": "ex:FadeAction",
      "name": "Fade In",
      "href": ["in", "actions/fadein"]
    } ],
  "events": [
    { "@type": "ex:CriticalEvent",
      "name": "Critical Condition",
      "href": ["ev", "events/critical"]
    } ]
}
```

TD Discovery

- Used at Plugfests
  - Manual: Provide TD by user
  - Repository: Register and look up with SPARQL

- Planned
  - In proximity
  - On network
  - Directories

WoT API

- Web API that is describable by a TD (resource model)
- Client and server connectors
- Various protocol mappings
  - HTTP (commonly used)
  - CoAP (commonly used)
  - MQTT (envisioned, requires shim layer)
  - BLE (envisioned, requires URI definition)
  - …

Scripting API

- API provided by a common scripting runtime for portable apps (cf. Web browser API for the normal Web)
- Defined in WebIDL and intended for various scripting languages
  - JavaScript (common on the Web)
  - Lua (common in embedded systems)
  - …
- Different aspects
  - Physical API (directly attached hardware;)
  - Discovery API (find and filter things)
  - Client API (access other things)
  - Server API (provide interactions for other things)

WebIDL Discovery API: ThingRequest

```
[Constructor(ThingFilter filter)] interface ThingRequest {
  Promise<sequence<Thing>> start(); };

dictionary ThingFilter {
    DOMString? type;
    ThingProximity? proximity;
    DOMString? id;
    DOMString? server;
};
```

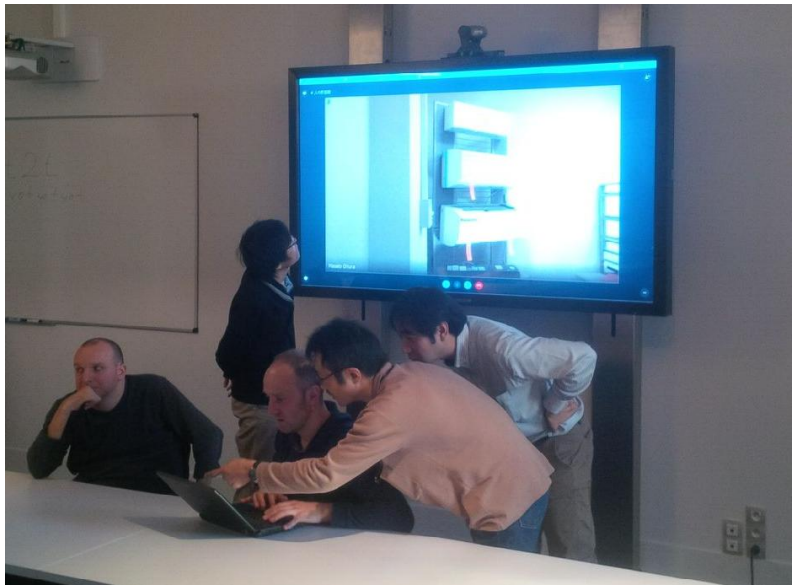
WebIDL Client API: ConsumedThing

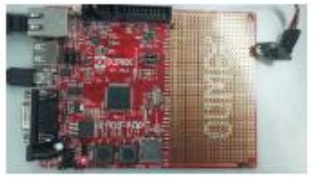
```
[Constructor(ThingDescription td)]
interface ConsumedThing {
    readonly attribute DOMString id;
    readonly attribute DOMString type;
    readonly attribute DOMString name;
    readonly attribute boolean reachable;
    attribute EventHandler onreachabilitychange;
    Promise<any> callAction(DOMString actionName, any parameter);
    Promise<any> setProperty(DOMString propertyName, any newValue);
    Promise<any> getProperty(DOMString propertyName);
    void addListener(DOMString eventName, ThingEventListener listener);
    void removeListener(DOMString eventName, ThingEventListener listener);
    void removeAllListeners(DOMString eventName);
};
callback ThingEventListener = void (ThingEvent event);
```

WebIDL Server API: ExposedThing

```
[Constructor(ThingDescription td), Constructor(DOMString name)]
interface ExposedThing {
    readonly attribute ThingDescription description;
    readonly attribute DOMString name;

    ExposedThing addAction(DOMString actionName, DOMString paraType, DOMString resType);
    ExposedThing addProperty(DOMString propertyName, DOMString propertyType);

    void onCall(DOMString actionName, ActionEventListener listener);
    void onChange(DOMString actionName, ChangeEventListener listener);

    void addListener(DOMString eventName, ThingEventListener listener);
    void removeListener(DOMString eventName, ThingEventListener listener);
    void removeAllListeners(DOMString eventName);
};
callback ThingEventListener = void (ThingEvent event);
callback ChangeEventListener = void (any newValue, any OldValue, ThingEvent event);
callback ActionEventListener = any (any param);
```

Plugfests





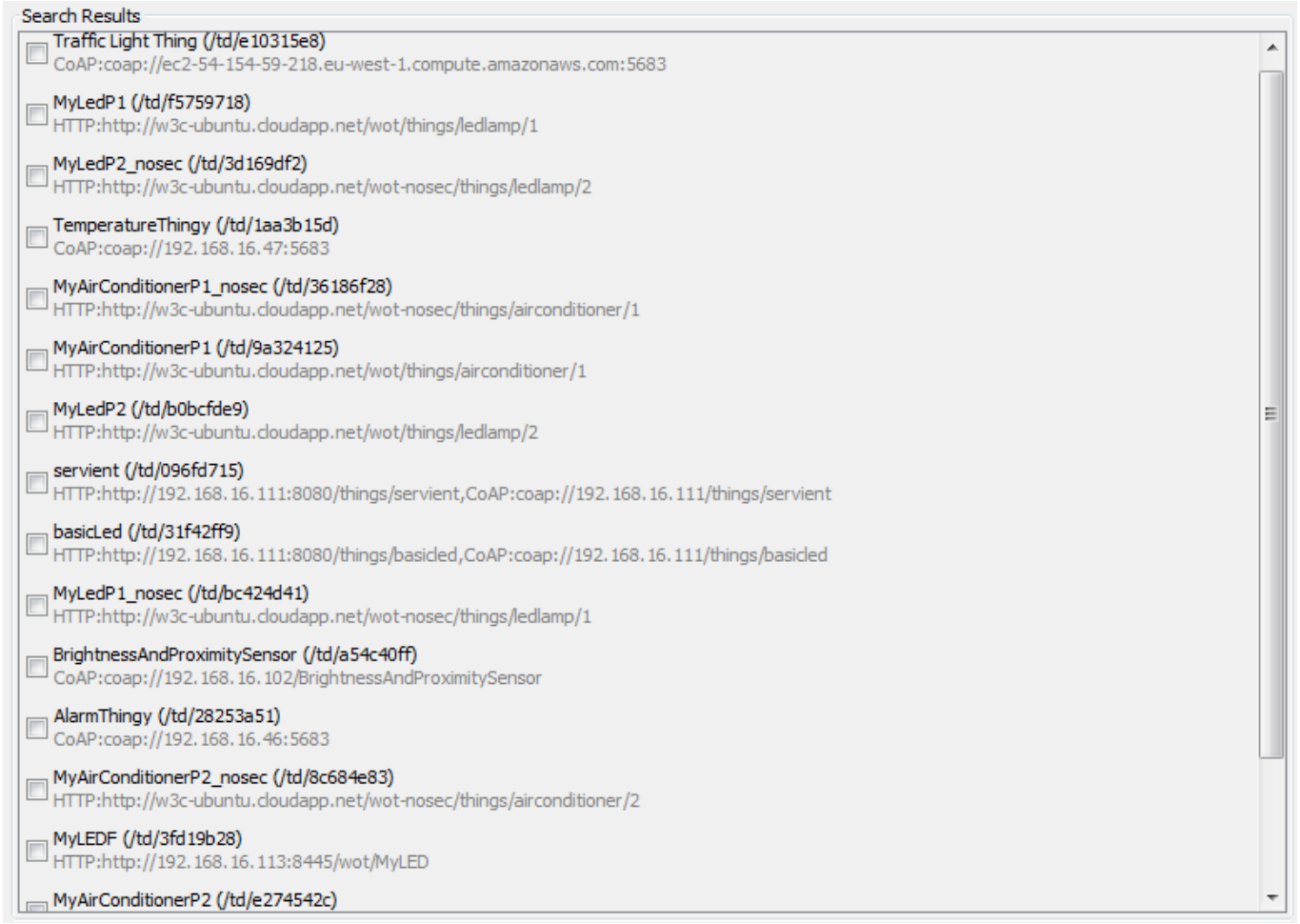
Sapporo Plugfest

- 15 TDs provided
- Mainly HTTP
- Few CoAP implementations
- Easy access
  (via browser, UI, cloud)
- Successful interaction

brightnessProximitySensor.jsonld	Darkos Thing Descriptions for Demo
car.jsonld	Fix json format.
door.jsonld	* Add 3 new jsonld files in the TD Samples folder from Fraunhofer FOKUS:
fan.jsonld	Darkos Thing Descriptions for Demo
fancy_led.jsonld	adjusted action to accept parameter
led.jsonld	(same as previous)
led_f.jsonld	Update led_f.jsonld
led_for_pi.jsonld	Darkos Thing Descriptions for Demo
led_v02.jsonld	update TD tutorial
outlet.jsonld	* Add 3 new jsonld files in the TD Samples folder from Fraunhofer FOKUS:
sensor_I2C.jsonld	Update sensor_I2C.jsonld
temperatureSensor.jsonld	Darkos Thing Descriptions for Demo
traffic light.jsonld.exi	rename
traffic_light.jsonld	traffic light thing descriptions in plain Json-LD and EXI format
weather.jsonld	* Add 3 new jsonld files in the TD Samples folder from Fraunhofer FOKUS:

Nice Plugfest

- 21 TDs provided
- Mainly HTTP
- Some CoAP
- Individual WS and MQTT
- TD repository
- Security
- Thing-2-Thing interaction



Montreal Plugfest

- Focus on updated mechanisms
- Scenario for more compex T2T interaction
- Participation
  - https://www.w3.org/WoT/IG/wiki/F2F_meeting_2016,_April,_11th_-_13th,_Montreal,_Canada
  - Sign up
  - Follow information in Current Practices document
- Provide information about your thing implementation → scenario
  - Descriptive name (e.g., “Color LED”, “Humidity Sensor”)
  - Properties, actions, and events

Links

- http://w3c.github.io/wot/current-practices/wot-practices.html
- http://w3c.github.io/wot/architecture/wot-architecture.html
- https://www.w3.org/WoT/IG/wiki/