KIT Karlsruhe Institute of Technology

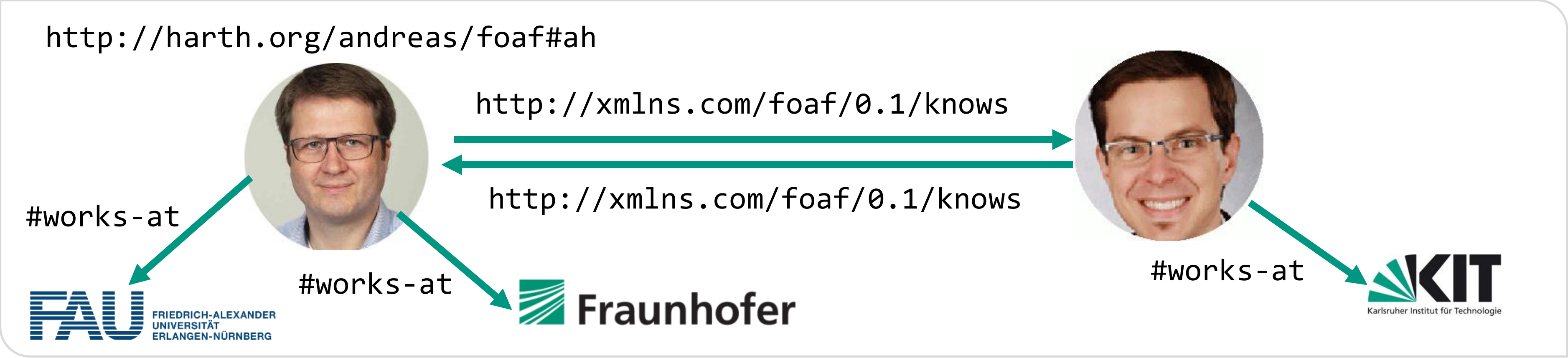
Fraunhofer

FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Tutorial on Distributed Knowledge Graphs for the Web of Things, Part V: Web of Things

Tobias Käfer (KIT) and Andreas Harth (FAU)

Tutorial @ 10th International Conference on the Internet of Things (IoT), 2020



http://harth.org/andreas/foaf#ah

http://xmlns.com/foaf/0.1/knows

http://xmlns.com/foaf/0.1/knows

#works-at

#works-at

#works-at

FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
Fraunhofer
KIT Karlsruher Institut für Technologie

Karlsruhe Institute of Technology (KIT); Fraunhofer IIS-SCS; Friedrich-Alexander University Erlangen-Nuremberg (FAU)

http://go.wiwi.kit.edu/togwot

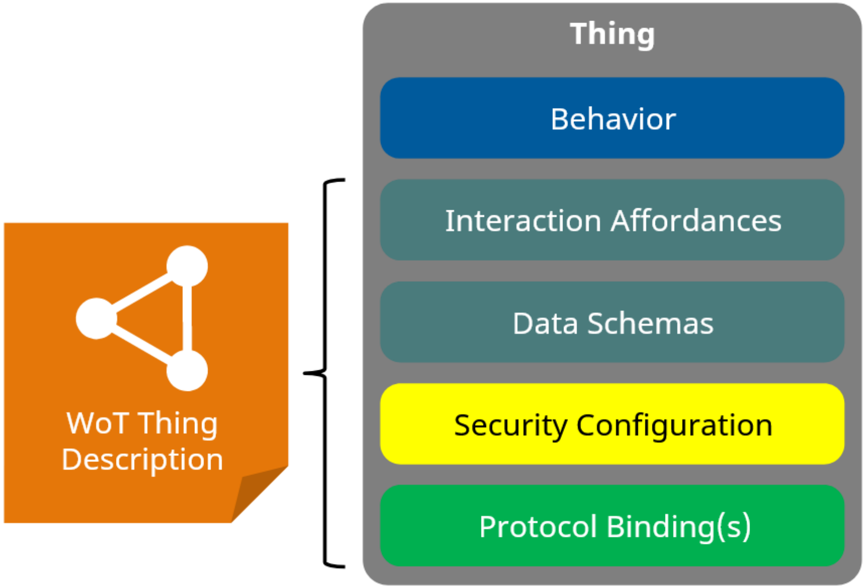
Tagline of W3C‘s WoT Interest Group [1] (until 9/2019)

The **Internet of Things (IoT)** suffers from a lack of interoperability across platforms. As a result developers are faced with data silos, high costs and limited market potential. This can be likened to the situation before the Internet when there were competing non-interoperable networking technologies. The Internet makes it easy to develop networked applications independently of those technologies. W3C is seeking to do the same for the Internet of Things.

To achieve this goal, we need platform independent APIs for application developers, and a means for different platforms to discover how to inter-operate with one another. The approach we are taking is based upon rich metadata that describes the data and interaction models exposed to applications, and the communications and security requirements for platforms to communicate effectively. A further aspect is the need to enable platforms to share the same meaning when they exchange data. We are therefore seeking to enable expression of the semantics of things and the domain constraints associated with them, building upon W3C's extensive work on RDF and **Linked Data**.

[1] http://www.w3.org/WoT/IG/

W3C Recommendation 04/2020
Web of Things Architecture [1]



→ WoT Scripting API

JSON Schema, …

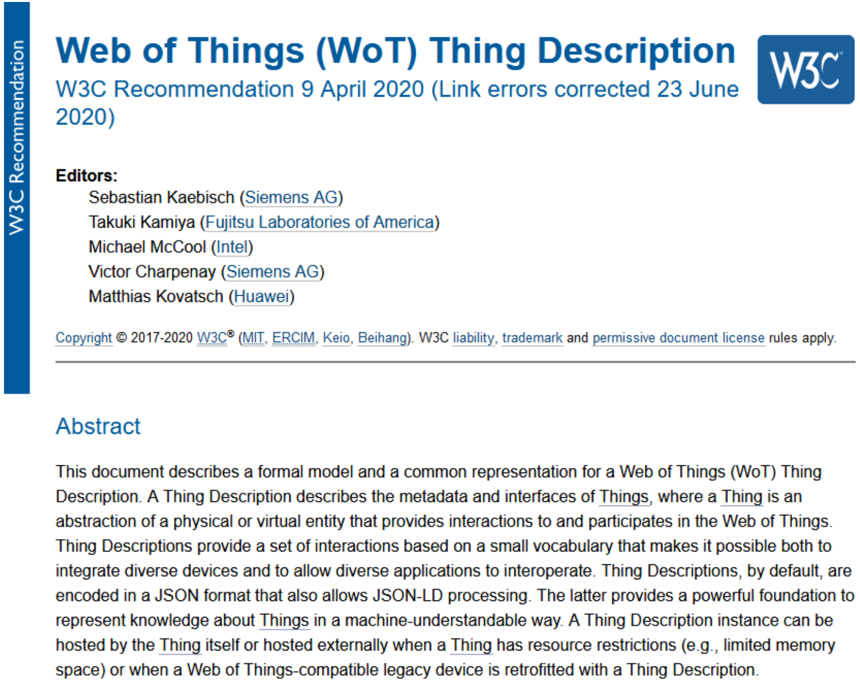
TLS, DTLS, OAuth, …

HTTP, CoAP, MQTT, …

Figure 19 Architectural Aspects of a Thing

[1] https://www.w3.org/TR/wot-architecture/

W3C Recommendation 04/2020
Web of Things Description [1]



Web of Things (WoT) Thing Description

W3C Recommendation 9 April 2020 (Link errors corrected 23 June 2020)

Editors:

Sebastian Kaebisch (Siemens AG)
Takuki Kamiya (Fujitsu Laboratories of America)
Michael McCool (Intel)
Victor Charpenay (Siemens AG)
Matthias Kovatsch (Huawei)

Copyright © 2017-2020 W3C® (MIT, ERCIM, Keio, Beihang). W3C liability, trademark and permissive document license rules apply.

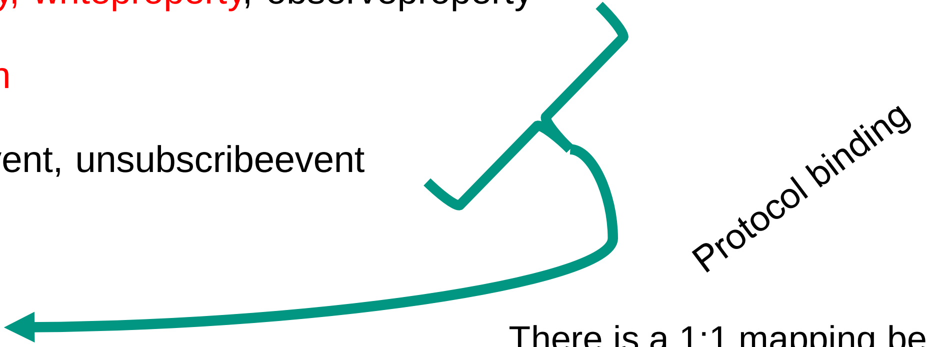
Abstract

This document describes a formal model and a common representation for a Web of Things (WoT) Thing Description. A Thing Description describes the metadata and interfaces of Things, where a Thing is an abstraction of a physical or virtual entity that provides interactions to and participates in the Web of Things. Thing Descriptions provide a set of interactions based on a small vocabulary that makes it possible both to integrate diverse devices and to allow diverse applications to interoperate. Thing Descriptions, by default, are encoded in a JSON format that also allows JSON-LD processing. The latter provides a powerful foundation to represent knowledge about Things in a machine-understandable way. A Thing Description instance can be hosted by the Thing itself or hosted externally when a Thing has resource restrictions (e.g., limited memory space) or when a Web of Things-compatible legacy device is retrofitted with a Thing Description.

[1] https://www.w3.org/TR/wot-thing-description/

WoT Thing Descriptions

- Description in RDF (JSON-LD):
  - What a Thing can do
  - How to do it (send JSON/… on this channel…)
- Functional descriptions targeted to developers
  - Interaction affordances and their interaction verbs
    - Property
      - readproperty, writeproperty, observeproperty
    - Action
      - invokeaction
    - Event
      - subscribeevent, unsubscribeevent
  - Protocols
    - HTTP
    - CoAP
    - MQTT
    - …



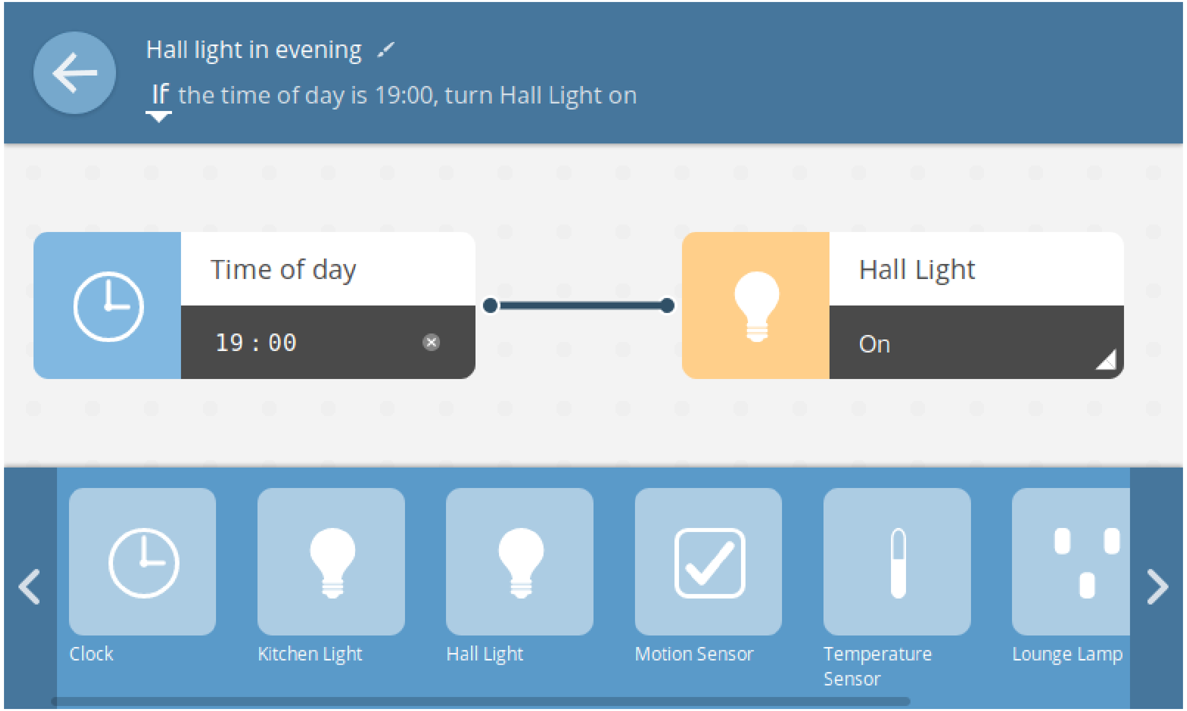
There is a 1:1 mapping between the red interaction verbs and HTTP methods GET, PUT, POST

A Sample Thing Description [1]

```
{
    "@context": "https://www.w3.org/2019/wot/td/v1",
    "id": "urn:dev:ops:32473-WoTLamp-1234",
    "title": "MyLampThing",
    "securityDefinitions": {
        "basic_sc": {"scheme": "basic", "in":"header"}
    },
    "security": ["basic_sc"],
    "properties": {
        "status" : {
            "type": "string",
            "forms": [{"href": "https://mylamp.example.com/status"}]
        }
    },
    "actions": {
        "toggle" : {
            "forms": [{"href": "https://mylamp.example.com/toggle"}]
        }
    },
    "events":{
        "overheating":{
            "data": {"type": "string"},
            "forms": [{ … … …
```

[1] https://www.w3.org/TR/wot-thing-description/

Rule-based Automation using http://webthings.io/ (related to W3C WoT; Mozilla Project until 09/2020)



Hall light in evening

If the time of day is 19:00, turn Hall Light on

Time of day: 19 : 00

Hall Light: On

*Scripting UI on your **local** gateway https://iot.mozilla.org/gateway/*

Read-Write Linked Data and the WoT Scripting API

Web of Things (WoT) Scripting API

…

2. Use Cases

- *This section is non-normative.*
- The following scripting use cases are supported in this specification:

2.1 Discovery

- …

2.2 Consuming a Thing

- …Fetch a TD and consume the thing (read the descriptions about the low level access APIs)…
- On a consumed Thing,
  - Read the value of a Property or set of properties.
  - Set the value of a Property or a set of properties.
  - Observe value changes of a Property.
  - Invoke an Action.
  - Observe Events emitted by the Thing.
  - Observe changes to the Thing Description of the Thing.
  - Get the Thing Description.
  - Get the list of linked resources based on the Thing Description.

2.3 Exposing a Thing

+ Property value/action payload in RDF
≈ Read-Write Linked Data = HTTP access [RFC7230seqq]
+ RDF data