W3C WoT in a nutshell

Yongjing Zhang (zhangyongjing@Huawei.com)
W3C WoT IG Co-chair
24 May 2017, oneM2M Industry Day @ TP#29
- Web of Things (WoT) overview
- WoT Thing Description (TD)
- WoT Scripting API
- WoT work organization & collaboration
- oneM2M-WoT Interworking
W3C WoT Mission

Interconnect the silos = de-silo

“enable easy integration across IoT platforms and application domains”
“complementing available standards”
# The Role of W3C in IoT/WoT – Play to the Strengths

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application</strong></td>
<td>Define thing behaviour in terms of their properties, actions and events, using APIs for control of sensor and actuator hardware.</td>
</tr>
<tr>
<td><strong>Things</strong></td>
<td>Software objects representing abstract or physical devices and state. Abstract thing to thing interaction. Semantics and Metadata, Data models and Data.</td>
</tr>
<tr>
<td><strong>Transfer</strong></td>
<td>Bindings of abstract messages to mechanisms provided by each protocol, including choice of communication pattern, e.g. pull, push, pub-sub, peer to peer, etc.</td>
</tr>
<tr>
<td><strong>Transport</strong></td>
<td>REST based protocols, e.g. HTTP, CoAP. Pub-Sub protocols, e.g. MQTT, XMPP. Others, including non IP transports, e.g. Bluetooth.</td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td>Underlying communication technology with support for exchange of simple messages (packets). Many technologies designed for different requirements</td>
</tr>
</tbody>
</table>
Semantic Metadata is the Key

**Metadata enables interoperability**
- Describe the interfaces exposed to applications
- Describe the communication and security requirements for accessing things
- Describe the data models, semantics, and domain constraints

**Metadata simplifies application development**
- Decouples underlying protocols
- Enables automated tooling
W3C WoT Framework

- Cloud Mirrors
  - Semantic metadata (Thing Description)
  - Application
    - Device Shadow
      - Scripting API
      - Interaction Model
      - Binding Templates
  - Remote access
  - Web integration
    - Web Browser
      - Scripting API
      - Interaction Model
      - Binding Templates
  - Direct Thing-to-Thing
    - Existing Device
      - (Thing Description, Binding Templates)
    - Thing
      - Powerfull or constrained Things

- Local Hubs
  - Application
    - Virtual Thing
      - Scripting API
      - Interaction Model
      - Binding Templates
  - Standardized APIs for portable application logic (Scripting API)
  - Cloud Mirrors
    - Binding Templates
  - Local Hubs
    - Binding Templates

Complementing existing devices and platforms

Web integration

- Existing Device
  - +
  - Thing Description, Binding Templates
4 Key Components: W3C WoT Building Blocks

**WoT Scripting API:**
A standardized API to simplify IoT application development and enable portable scripts across vendors and device, gateway, and cloud platforms.

**WoT Thing Description (TD):**
Provides metadata of the interactions, data model, communication, as well as security mechanisms of the Thing.

**WoT Binding Templates:**
The TD also describes the usage of protocols. A vanilla protocol stack can be configured at runtime to produce messages that will be understood by the targeted.

**Security & Privacy:**
W3C WoT does not invent new mechanisms, but ensures that all building blocks provide means to describe the security and privacy mechanisms used in a specific platform and provides adversary testing of Things.
WoT Thing Description
Describe Thing, communication, and security metadata
https://w3c.github.io/wot-thing-description/
```json
{
  "@context": [
    "http://w3c.github.io/wot/w3c-wot-td-context.jsonld",
    {
      "domain": "http://example.org/actuator#"
    }
  ],
  "@type": "Thing",
  "name": "MyLEDThing",
  "base": "coap://myled.example.com:5683/",
  "security": {
    "cat": "token:jwt",
    "alg": "HS256",
    "as": "https://authority-issuing.example.org"
  },
  "interactions": [
    {
      "@type": ["Property", "domain:onOffStatus"],
      "name": "status",
      "outputData": {
        "valueType": {
          "type": "boolean"
        }
      },
      "writable": true,
      "links": [
        "href": "pwr"
      ]
    }
  ]
}
```
"interactions": [
  {
    "@type": ["Property", "domain:onOffStatus"],
    "name": "status",
    "outputData": {"valueType": {"type": "boolean"}},
    "writable": true,
    "links": [
      {
        "href": "pwr",
        "mediaType": "application/exi"
      },
      {
        "href": "http://mytemp.example.com:8080/status",
        "mediaType": "application/json"
      }
    ]
  },
  {
    "@type": ["Action", "domain:fadeIn"],
    "name": "fadeIn",
    "inputData": {
      "valueType": {"type": "integer"},
      "domain:unit": "domain:ms"
    },
    "links": [
      {
        "href": "in",
        "mediaType": "application/exi"
      }
    ]
  }
]
inputData: {
    "valueType": {"type": "integer"},
    "domain:unit": "domain:ms"
},
"links": [
    {
        "href": "out",
        "mediaType": "application/exi"
    },
    {
        "href": "http://mytemp.example.com:8080/out",
        "mediaType": "application/json"
    }
],
"@type": ["Event", "domain:alert"],
"name": "criticalCondition",
"outputData": {"valueType": {"type": "string"}},
"links": [
    {
        "href": "ev",
        "mediaType": "application/exi"
    }
]
WoT Thing Description

- JSON-LD is just one possible representation
  - Good for discussion, accepted by Web people

- TD is a semantic model
  - Backed by RDF and Linked Data vocabularies
  - Yet complexity of Semantic Web can be ignored

- Other formats possible
  - EXI, CBOR, ... for machines
  - Custom `application/wot-td+json` for developers
  - Just serializations of the semantic model
W3C WoT work organization
W3C WoT

Interest Group (IG)
https://www.w3.org/2016/07/wot-ig-charter.html
- Started spring 2015
- 218 participants
- Informal work, outreach
- Use cases, explorative work
- Liaisons and collaborations with other organizations and SDOs
- PlugFests with running code

Working Group (WG)
https://www.w3.org/2016/12/wot-wg-2016.html
- Started December 2016
- 71 participants
- Normative work
- Standardization of four initial building blocks identified by the IG
W3C WoT Task Forces

**IG**
- Current Practices (has deliverable)
- Testing (PlugFest scenarios)
- Thing Lifecycle
- Synchronization of Servients
- Linked Data and Semantic Processing
- Demonstrators
- Liaison with OCF
- Liaison with oneM2M (tbc, You’re wanted 😊)

**WG**
- Architecture (has deliverable)
- Thing Description (has deliverable)
  - Type System (JSON Schema Extensions)
  - Hypermedia (Actions, error handling, ...)
- Scripting API (has deliverable)
- Binding Templates (has deliverable)
- Security & Privacy

Liaison with OCF
Liaison with oneM2M (tbc, You’re wanted 😊)
W3C WoT Liaisons

- IETF / IRTF
  - Established, joint meetings since Nov 2015

- Open Connectivity Foundation (OCF)
  - Established, active alignment and joint PlugFest coming up

- oneM2M
  - Established, commonality identified and preparing input

- OPC Foundation
  - Established, need to agree on strategy etc.

- Plattform Industrie 4.0
  - Initial conference calls

- OpenFog
  - Initial outreach
WG Roadmap

2017

Feb (Santa Clara F2F)
• Create GitHub repos

May (Osaka F2F)
• Graphical and RDF model of TD
• Editor’s Drafts
• RC for WoT Arch. FPWD

Jul (Düsseldorf F2F)
• Draft for test suite
• Release FPWD of WoT Arch.

Nov (TPAC, Burlingame)
• RCs for FPWD
• RC Implementations and Test Suite

2018

Mar (London? F2F)
• Finish security review
• Release FPWDs

Jul (China? F2F)

Nov (TPAC, Asia?)
• Finish security review
• Start release process

May (Santa Clara?)
• Aggressive testing / adverse security testing?

Oct
• RCs for Candidate Recommendations

Dec (end of current charter)
• Release Candidate Recommendations (CRs)
W3C WoT Online Resources

- **W3C WoT Interest Group**
  - [https://www.w3.org/WoT/IG/](https://www.w3.org/WoT/IG/) (blog)
  - [https://www.w3.org/2016/07/wot-ig-charter.html](https://www.w3.org/2016/07/wot-ig-charter.html) (charter)
  - [https://lists.w3.org/Archives/Public/public-wot-ig/](https://lists.w3.org/Archives/Public/public-wot-ig/) (subscribe to mailing list)

- **W3C WoT Working Group**
  - [https://www.w3.org/WoT/WG/](https://www.w3.org/WoT/WG/) (dashboard)
  - [https://www.w3.org/2016/12/wot-wg-2016.html](https://www.w3.org/2016/12/wot-wg-2016.html) (charter)

- **W3C WoT Wiki (IG+WG organizational information)**
  - [https://www.w3.org/WoT/IG/wiki/Main_Page](https://www.w3.org/WoT/IG/wiki/Main_Page)

- **W3C WoT GitHub (IG technical proposals)**
  - [https://github.com/w3c/wot](https://github.com/w3c/wot)

- **W3C WoT WG Documents**
  - [https://w3c.github.io/wot-architecture/](https://w3c.github.io/wot-architecture/)
  - [https://w3c.github.io/wot-thing-description/](https://w3c.github.io/wot-thing-description/)
  - [https://w3c.github.io/wot-scripting-api/](https://w3c.github.io/wot-scripting-api/)
  - [https://w3c.github.io/wot-binding-templates/](https://w3c.github.io/wot-binding-templates/)
Web of Things Participants
oneM2M-WoT Interworking

Preliminary thoughts for discussion
Interworking: WoT → oneM2M

- Exposing the WoT interface (described in TD) to oneM2M systems
  - Benefit: WoT services/data can be consumed by oneM2M applications
Interworking: oneM2M → WoT

- Exposing oneM2M interfaces to WoT systems
  - Benefit: oneM2M services/data can be consumed by WoT Servients

```
{
  "@context": [
    "http://w3c.github.io/iot/w3c-wot-td-context.jsonld",
    { "actor": "http://example.org/actor#" }
  ],
  "@type": "Thing",
  "name": "MyLEDThing",
  "@id": "http://example.org/myled/",
  "@context": ["http://w3c.github.io/iot/w3c-wot-td-context.jsonld", { "actor": "http://example.org/actor#" }],
  "description": "MyLEDThing",
  "properties": [
    { "@type": "actuator:OnOffStatus",
      "name": "status",
      "type": "boolean",
      "writable": true,
      "valueType": "boolean",
      "@context": ["http://w3c.github.io/iot/w3c-wot-td-context.jsonld", { "actor": "http://example.org/actor#" }]
    }
  ],
  "@id": "http://example.org/myled/"
}
```
oneM2M HAIM vs. WoT

Device Model

- `@type`
- `deviceAirConditioner`
- `contDefinition`
- `creator`
- `ontologyRef`
- `mgmtLink`
- `binarySwitch`
- `runMode`
- `temperature`
- `timer`
- `turbo`
- `wind`
- `subscription`
- `mgmtLink`

Device properties

- `deviceLabel`
- `deviceType`
- `deviceName`
- `model`
- `manufacturer`
- `subModel`
- `manufacturerDetailsLink`
- `manufacturingDate`
- `hwVersion`
- `swVersion`
- `osVersion`
- `hwVersion`
- `country`
- `location`
- `systemTime`
- `supportURL`
- `presentationURL`

Module Classes

- `@type`
- `binarySwitch`
- `@type` or `action`
- `toggle`
- `powerState`
- `event`
- `property`

well mapping based on large commonalities
oneM2M general Resource Model vs. WoT

Universal Attributes (mandatory)

<table>
<thead>
<tr>
<th>Attribute Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>resourceType</td>
</tr>
<tr>
<td>resourceID</td>
</tr>
<tr>
<td>resourceName</td>
</tr>
<tr>
<td>parentID</td>
</tr>
<tr>
<td>creationTime</td>
</tr>
<tr>
<td>lastModifiedTime</td>
</tr>
<tr>
<td>expirationTime</td>
</tr>
</tbody>
</table>

Common Attributes (optional)

<table>
<thead>
<tr>
<th>Attribute Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>accessControlPolicyIDs</td>
</tr>
<tr>
<td>stateTag</td>
</tr>
<tr>
<td>announceTo</td>
</tr>
<tr>
<td>announcedAttribute</td>
</tr>
<tr>
<td>labels</td>
</tr>
<tr>
<td>e2eSecInfo</td>
</tr>
<tr>
<td>dynamicAuthorizationConsultationIDs</td>
</tr>
<tr>
<td>creator</td>
</tr>
</tbody>
</table>

distributed vs centralized
Thanks You!

For more information on W3C see:

www.w3.org