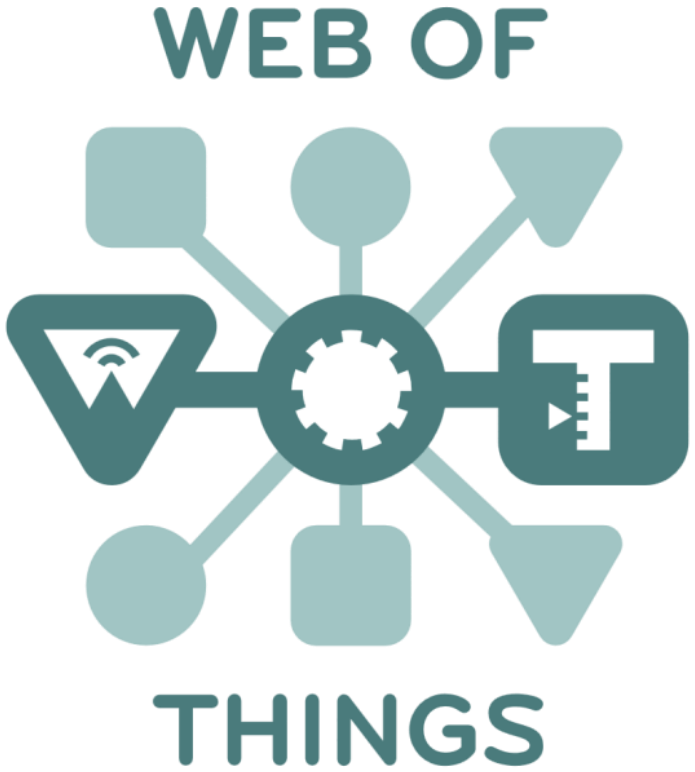


WoT Status Update and Next Steps

TPAC 2019, Sept 2019

Michael McCool: Intel Principal Engineer / W3C WoT WG Co-chair

W3C Web of Things

Goal: Support IoT Interoperability via Open Standards

- **W3C WoT Interest Group (IG)**
  https://www.w3.org/2016/07/wot-ig-charter.html
  - Started spring 2015
  - ~200 participants
  - Informal work and outreach
  - "PlugFest" validation with running code
  - Exploration of new building blocks
  - "OpenDays" with external speakers
  - Liaisons and collaborations with other organizations and SDOs
  - ***Second Workshop on Web of Things held 3-5 June 2019 in Munich***
  - ***Charter renewal submitted Sept 2019***

- **W3C WoT Working Group (WG)**
  https://www.w3.org/2016/12/wot-wg-2016.html
  - Started end of 2016 (effectively Feb 2017)
  - ~100 participants
  - Normative work on specific deliverables
  - W3C Patent Policy for royalty-free standards
  - Only W3C Members and Invited Experts
  - ***Architecture and Thing Description were published as Candidate Recommendations on 16 May 2019***
  - ***Notes published on Protocol Bindings, Security, and Scripting API***
  - ***Charter renewal in progress; work items and deliverables under discussion***

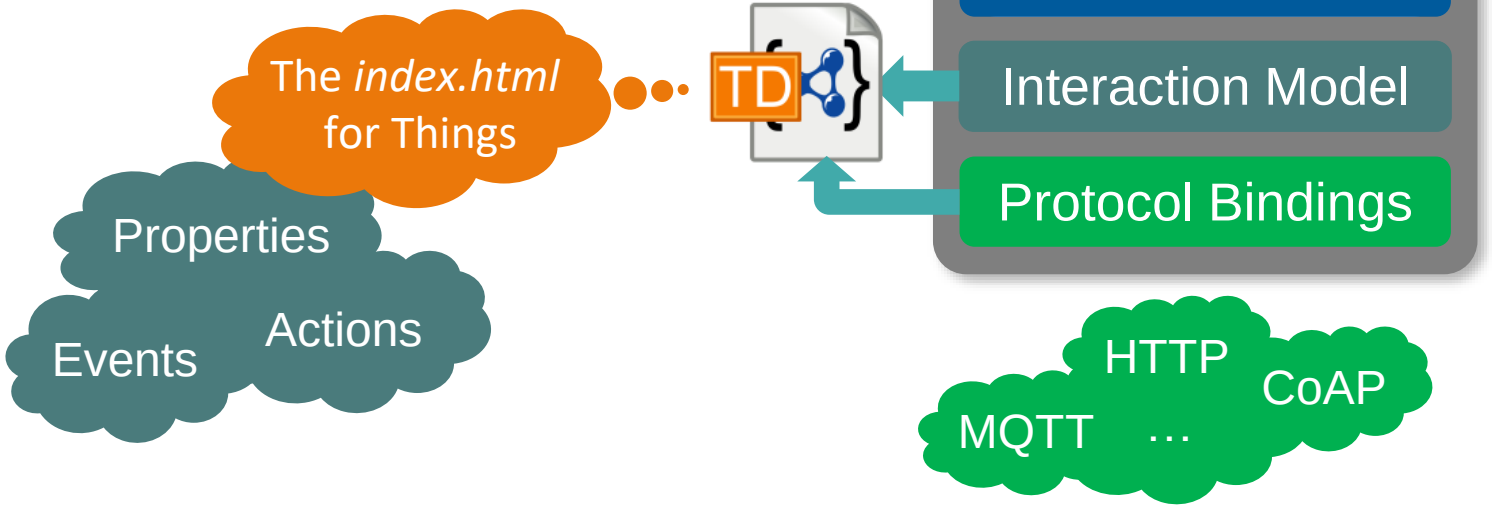
W3C Web of Things – Building Blocks

WoT Architecture

Overarching umbrella with architectural constraints and guidance on how to use and combine building blocks.

WoT Thing Description (TD)

JSON-LD representation format to describe Thing *instances* with **metadata**. Uses **formal interaction model** and **domain-specific vocabularies** to uniformly describe how to use Things, which enables semantic interoperability.



Security Guidelines

WoT Scripting API

Standardized **JavaScript** object API for an IoT runtime system **similar to the Web browser**. Provides an interface between applications and Things to simplify IoT application development and enable **portable apps** across vendors, devices, edge, and cloud.

WoT Binding Templates

Capture how the **formal Interaction Model** is mapped to concrete protocol operations (e.g., CoAP) and platform features (e.g., OCF). These templates are re-used by concrete TDs.

W3C Web of Things – Building Blocks

WoT Architecture

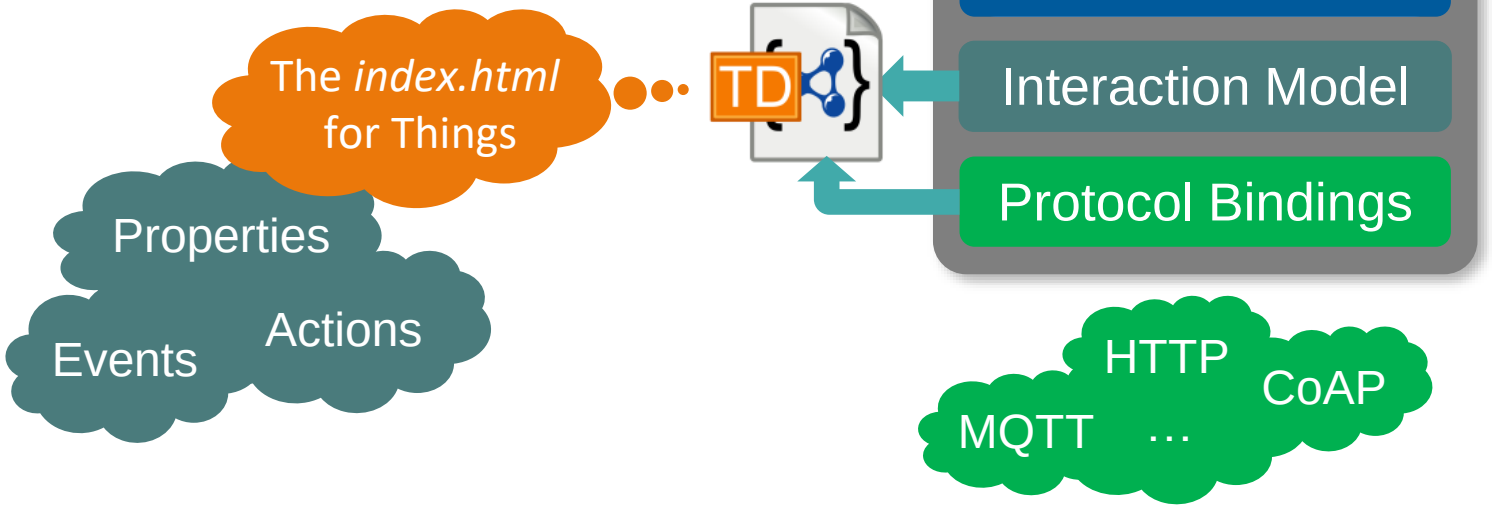
REC Track

Overarching umbrella with architec[illegible]nstraints and guidance on how to use and combine building blocks.

WoT Thing Description (TD)

REC Track

JSON-LD represent[illegible]ion format to describe Thin[illegible]s with **metadata**. Uses **forma[illegible] model** and **domain-s[illegible]ularies** to uniformly d[illegible]w to use Things, which enables semantic interoperability.



Security Guidelines

WoT Scripting API

WG Note

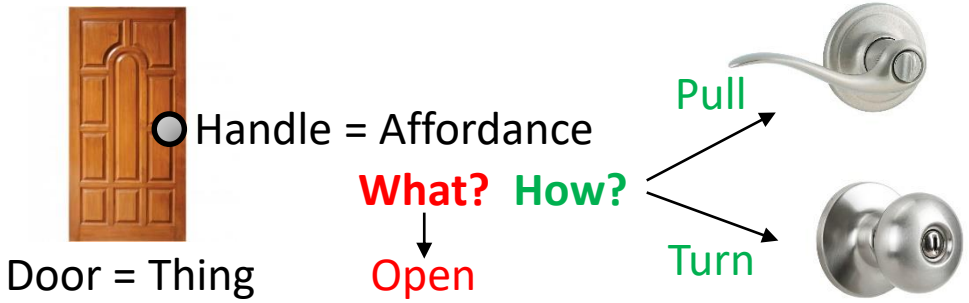
Standardized **JavaScri[illegible]** object API for an IoT runti[illegible] sy[illegible] **milar to the Web bro[illegible]** nterface between [illegible]ings to simplify Io[illegible]evelopment and enable **portable apps** across vendors, devices, edge, and cloud.

WoT Binding Templates

Capture how the **formal Interaction Model** is mapped to concrete protocol operations (e.g., CoAP) and platform features (e.g., OCF). These templates are re-used by concrete TDs.

Published Candidate Recommendations

- **WoT Architecture**
  - Constraints
    - Things must have TD (W3C WoT)
    - Must use hypermedia controls (general WoT)
      - URIs
      - Standard set of methods
      - Media Types
  - Interaction Affordances
    - Metadata of a Thing that shows and describes the possible choices (what) to Consumers, thereby suggesting how Consumers may interact with the Thing



- **WoT Thing Description (TD)**

```
{
  "@context": [
    "https://www.w3.org/2019/wot/td/v1",
    { "iot": "http://iotschema.org/" }
  ],
  "id": "urn:dev:org:32473:1234567890",
  "title": "MyLEDThing",
  "description": "RGB LED torchiere",
  "@type": ["Thing", "iot:Light"],
  "securityDefinitions": ["default": {
    "scheme": "bearer"
  }],
  "security": ["default"],
  "properties": {
    "brightness": {
      "@type": ["iot:Brightness"],
      "type": "integer",
      "minimum": 0,
      "maximum": 100,
      "forms": [ ... ]
    }
  },
  "actions": {
    "fadeIn": {
      ...
```

Plugfest, Use Cases, and Demos

WG Charter Proposal: Work Items

https://cdn.statically.io/gh/mmccool/wot/wg-charter-draft/charters/wot-wg-charter-draft-2019.html?env=dev

Architectural Requirements, Use Cases, and Vocabulary

- Understand and state requirements for new use cases, architectural patterns, and concepts.

Link Relation Types:

- Definition of specific link relation types for specific relationships.

Observe Defaults:

- For protocols such as HTTP where multiple ways to implement "observe" is possible, define a default.

Implementation View Spec:

- More fully define details of implementations.

Interoperability Profiles:

- Support plug-and-play interoperabilty via a profile mechanism
- Define profiles for specific application domains and use cases.

Thing Description Templates:

- Define how Thing Descriptions can defined in a modular way.

Complex Interactions:

- Document how complex interactions can be supported via hypermedia controls.

Discovery:

- Define how Things are discovered in both local and global contexts and Thing Descriptions are distributed.

Identifier Management:

- Mitigate privacy risks by defining how identifiers are managed and updated.

Security Schemes:

- Vocabulary for new security schemes supporting targeted protocols and use cases.

Thing Description Vocabulary:

- **Extensions to Thing** Description vocabulary definitions.

Protocol Vocabulary and Bindings:

- Extensions to protocol vocabulary definitions and protocol bindings.

WG Charter Proposal: Work Items

https://cdn.statically.io/gh/mmccool/wot/wg-charter-draft/charters/wot-wg-charter-draft-2019.html?env=dev

Architectural Requirements, Use Cases, and Vocabulary

- Understand and state requirements for new use cases, architectural patterns, and concepts.

Link Relation Types:

- Definition of specific link relation types for specific relationships.

Observe Defaults:

- For protocols such as HTTP where multiple ways to implement "observe" is possible, define a default.

Implementation View Spec:

- More fully define details of implementations.

Interoperability Profiles:

- Support plug-and-play interoperabilty via a profile mechanism
- Define profiles for specific application domains and use cases.

Thing Description Templates:

- Define how Thing Descriptions can defined in a modular way.

Complex Interactions:

- Document how complex interactions can be supported via hypermedia controls.

Discovery:

- Define how Things are discovered in both local and global contexts and Thing Descriptions are distributed.

Identifier Management:

- Mitigate privacy risks by defining how identifiers are managed and updated.

Security Schemes:

- Vocabulary for new security schemes supporting targeted protocols and use cases.

Thing Description Vocabulary:

- **Extensions to Thing** Description vocabulary definitions.

Protocol Vocabulary and Bindings:

- Extensions to protocol vocabulary definitions and protocol bindings.

WG Charter Proposal: Deliverables

https://cdn.statically.io/gh/mmccool/wot/wg-charter-draft/charters/wot-wg-charter-draft-2019.html?env=dev

Normative:

- Architecture (Update)
- Thing Description (Update)
- Thing Description (Next)
- Interoperability Profiles
- Discovery

Key Collaborations:

- Privacy
- Distributed ID
- JSON-LD

Informative:

- Use case and requirement documents
- Test suites and implementation reports for the specifications
- Security and Privacy Guidelines (W3C Note)
- Security and Privacy Best Practices (W3C Note)
- Protocol Bindings (W3C Note)
- Scripting API (W3C Note)
- Management API Template (W3C Note)
- Script Packaging (W3C Note)
- Developer's Guide (W3C Note)
- Other Primer or Best Practice documents to support developers and users

W3C WoT Resources

- W3C WoT Wiki
  - https://www.w3.org/WoT/IG/wiki
    (IG/WG organizational information)
- W3C WoT Interest Group
  - https://www.w3.org/2016/07/wot-ig-charter.html
    (charter)
  - https://lists.w3.org/Archives/Public/public-wot-ig/
    (mailing list)
  - https://github.com/w3c/wot
    (technical proposals)
- W3C WoT Working Group
  - https://www.w3.org/2016/12/wot-wg-2016.html
    (charter)
  - https://www.w3.org/WoT/WG/
    (dashboard)
- W3C WoT Candidate Recommendations
  - https://www.w3.org/TR/wot-architecture/
  - https://www.w3.org/TR/wot-thing-description/
- W3C WoT Working Drafts / Group Notes
  - https://www.w3.org/TR/wot-binding-templates/
  - https://www.w3.org/TR/wot-scripting-api/
  - https://www.w3.org/TR/wot-security/
- W3C WoT Editors' Drafts and Issue Tracker
  - https://github.com/w3c/wot-architecture/
  - https://github.com/w3c/wot-thing-description/
  - https://github.com/w3c/wot-binding-templates/
  - https://github.com/w3c/wot-scripting-api/
  - https://github.com/w3c/wot-security/
- Reference Implementation: node-wot
  - https://github.com/eclipse/thingweb.node-wot

Contacts

https://www.w3.org/WoT/WG/

Dr. Michael McCool

Principal Engineer

Intel

Technology Pathfinding

michael.mccool@intel.com

Dr. Matthias Kovatsch

Principal Researcher

Huawei Technologies

Applied Network Technology Lab

matthias.kovatsch@huawei.com