WoT Virtual F2F
Architecture + Profiles

Michael Lagally

26 Oct 2021

Architecture + Profiles

Agenda:

- Architecture Task Force
  - People and Deliverables
  - Current work
  - Next steps
  - Publication schedule
- Profiles
  - Introduction to WoT profiles
  - Use Cases and Core Requirements
  - Current work
  - Open issues
  - Proposal for a “Push” event model
  - Next steps
  - Publication schedule

WoT Architecture Task Force People

The WoT Architecture task force is responsible for the abstract architecture and interoperability profiles for the Web of Things.

People:

WoT Architecture Co-Editors:

Michael Lagally (Oracle Corp.), Ryuichi Matsukura (Fujitsu Ltd.), Toru Kawaguchi (Panasonic Corp.), and Kunihiko Toumura (Hitachi, Ltd.)

WoT Profile Co-Editors:

Michael Lagally (Oracle Corp.), Michael McCool (Intel Corp.), Ryuichi Matsukura (Fujitsu Ltd.), Sebastian Kaebisch (Siemens AG), and Tomoaki Mizushima (Internet Research Institute, Inc.)

WoT Architecture Task Force Deliverables

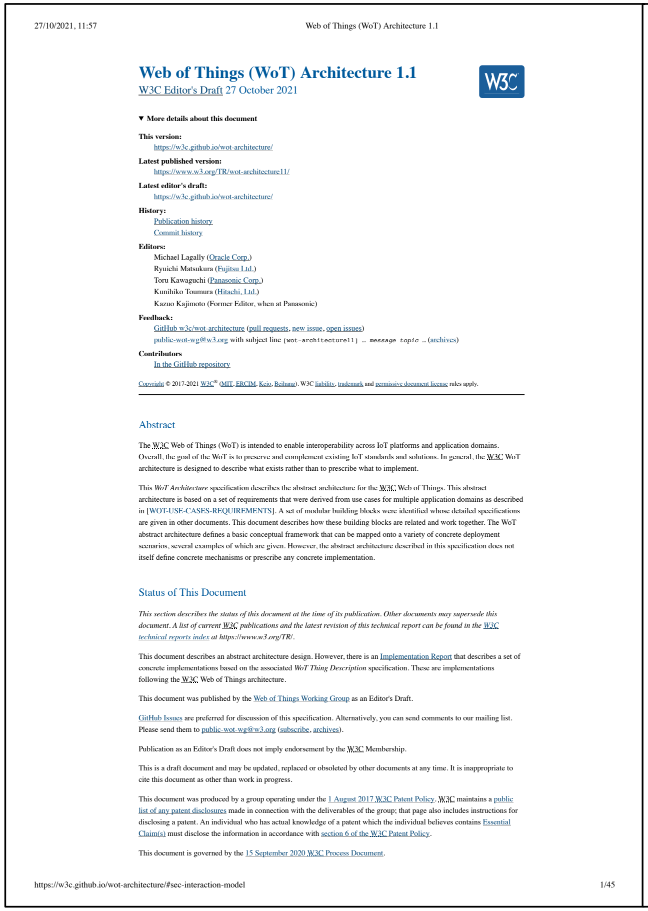
WoT Architecture (W3C Recommendation)

- The WoT Architecture specification describes the abstract architecture for the W3C Web of Things. It defines terminology that is used by all other WoT building blocks
- Conceptual framework that can be mapped onto a variety of concrete deployment scenarios.

WoT Profile (W3C Recommendation)

- Profiling Mechanism and a WoT Core Profile which enables out of the box interoperability among things and devices.
- Out of the box interoperability implies that devices can be integrated into various application scenarios without deep level adaptations. Integration can be done by anyone without specific training.

WoT Architecture 1.1



27/10/2021, 11:57 Web of Things (WoT) Architecture 1.1

Web of Things (WoT) Architecture 1.1

W3C Editor's Draft 27 October 2021

▼ **More details about this document**

This version:
https://w3c.github.io/wot-architecture/

Latest published version:
https://www.w3.org/TR/wot-architecture11/

Latest editor's draft:
https://w3c.github.io/wot-architecture/

History:
Publication history
Commit history

Editors:
Michael Lagally (Oracle Corp.)
Ryuichi Matsukura (Fujitsu Ltd.)
Toru Kawaguchi (Panasonic Corp.)
Kunihiko Toumura (Hitachi, Ltd.)
Kazuo Kajimoto (Former Editor, when at Panasonic)

Feedback:
GitHub w3c/wot-architecture (pull requests, new issue, open issues)
public-wot-wg@w3.org with subject line `[wot-architecture11] … message topic …` (archives)

Contributors
In the GitHub repository

Abstract

The W3C Web of Things (WoT) is intended to enable interoperability across IoT platforms and application domains. Overall, the goal of the WoT is to preserve and complement existing IoT standards and solutions. In general, the W3C WoT architecture is designed to describe what exists rather than to prescribe what to implement.

This *WoT Architecture* specification describes the abstract architecture for the W3C Web of Things. This abstract architecture is based on a set of requirements that were derived from use cases for multiple application domains as described in [WOT-USE-CASES-REQUIREMENTS]. A set of modular building blocks were identified whose detailed specifications are given in other documents. This document describes how these building blocks are related and work together. The WoT abstract architecture defines a basic conceptual framework that can be mapped onto a variety of concrete deployment scenarios, several examples of which are given. However, the abstract architecture described in this specification does not itself define concrete mechanisms or prescribe any concrete implementation.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at https://www.w3.org/TR/.

This document describes an abstract architecture design. However, there is an Implementation Report that describes a set of concrete implementations based on the associated *WoT Thing Description* specification. These are implementations following the W3C Web of Things architecture.

This document was published by the Web of Things Working Group as an Editor's Draft.

GitHub Issues are preferred for discussion of this specification. Alternatively, you can send comments to our mailing list. Please send them to public-wot-wg@w3.org (subscribe, archives).

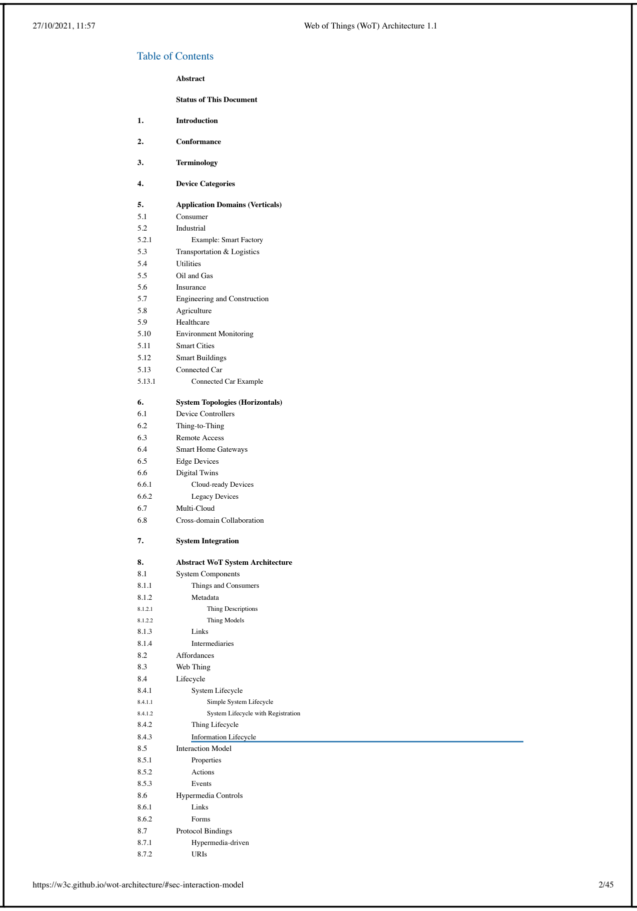
Publication as an Editor's Draft does not imply endorsement by the W3C Membership.

This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

This document was produced by a group operating under the 1 August 2017 W3C Patent Policy. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

This document is governed by the 15 September 2020 W3C Process Document.

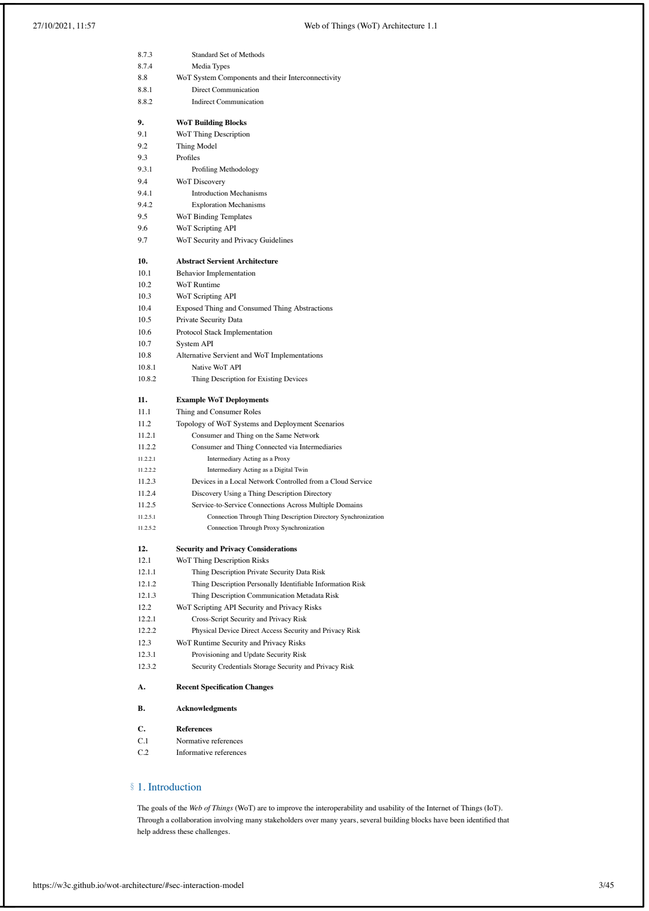
https://w3c.github.io/wot-architecture/#sec-interaction-model 1/45



27/10/2021, 11:57 Web of Things (WoT) Architecture 1.1

Table of Contents

https://w3c.github.io/wot-architecture/#sec-interaction-model 2/45



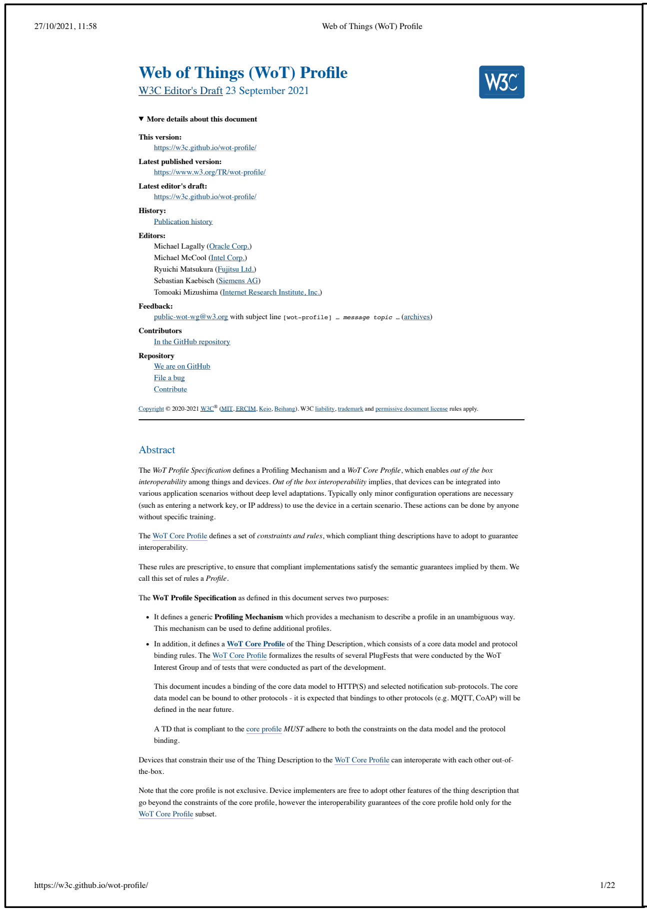
27/10/2021, 11:57 Web of Things (WoT) Architecture 1.1

§ 1. Introduction

The goals of the *Web of Things* (WoT) are to improve the interoperability and usability of the Internet of Things (IoT). Through a collaboration involving many stakeholders over many years, several building blocks have been identified that help address these challenges.

https://w3c.github.io/wot-architecture/#sec-interaction-model 3/45

WoT Profile



27/10/2021, 11:58 Web of Things (WoT) Profile

Web of Things (WoT) Profile

W3C Editor's Draft 23 September 2021

▼ **More details about this document**

This version:
https://w3c.github.io/wot-profile/

Latest published version:
https://www.w3.org/TR/wot-profile/

Latest editor's draft:
https://w3c.github.io/wot-profile/

History:
Publication history

Editors:
Michael Lagally (Oracle Corp.)
Michael McCool (Intel Corp.)
Ryuichi Matsukura (Fujitsu Ltd.)
Sebastian Kaebisch (Siemens AG)
Tomoaki Mizushima (Internet Research Institute, Inc.)

Feedback:
public-wot-wg@w3.org with subject line `[wot-profile] … message topic …` (archives)

Contributors
In the GitHub repository

Repository
We are on GitHub
File a bug
Contribute

Abstract

The *WoT Profile Specification* defines a Profiling Mechanism and a *WoT Core Profile*, which enables *out of the box interoperability* among things and devices. *Out of the box interoperability* implies, that devices can be integrated into various application scenarios without deep level adaptations. Typically only minor configuration operations are necessary (such as entering a network key, or IP address) to use the device in a certain scenario. These actions can be done by anyone without specific training.

The WoT Core Profile defines a set of *constraints and rules*, which compliant thing descriptions have to adopt to guarantee interoperability.

These rules are prescriptive, to ensure that compliant implementations satisfy the semantic guarantees implied by them. We call this set of rules a *Profile*.

The **WoT Profile Specification** as defined in this document serves two purposes:

- It defines a generic **Profiling Mechanism** which provides a mechanism to describe a profile in an unambiguous way. This mechanism can be used to define additional profiles.
- In addition, it defines a **WoT Core Profile** of the Thing Description, which consists of a core data model and protocol binding rules. The WoT Core Profile formalizes the results of several PlugFests that were conducted by the WoT Interest Group and of tests that were conducted as part of the development.

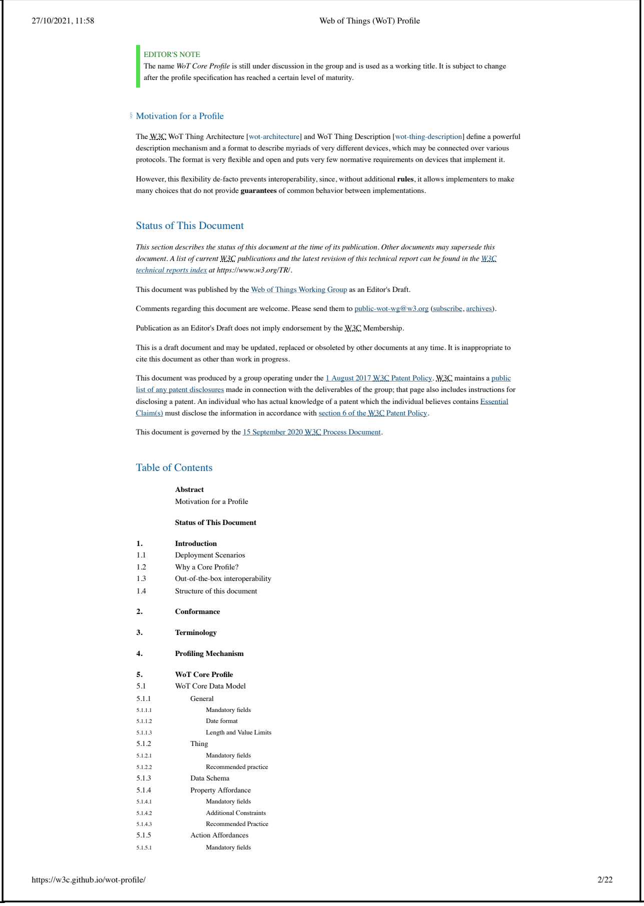
  This document incudes a binding of the core data model to HTTP(S) and selected notification sub-protocols. The core data model can be bound to other protocols - it is expected that bindings to other protocols (e.g. MQTT, CoAP) will be defined in the near future.

  A TD that is compliant to the core profile *MUST* adhere to both the constraints on the data model and the protocol binding.

Devices that constrain their use of the Thing Description to the WoT Core Profile can interoperate with each other out-of-the-box.

Note that the core profile is not exclusive. Device implementers are free to adopt other features of the thing description that go beyond the constraints of the core profile, however the interoperability guarantees of the core profile hold only for the WoT Core Profile subset.

https://w3c.github.io/wot-profile/ 1/22



27/10/2021, 11:58 Web of Things (WoT) Profile

EDITOR'S NOTE
The name *WoT Core Profile* is still under discussion in the group and is used as a working title. It is subject to change after the profile specification has reached a certain level of maturity.

§ Motivation for a Profile

The W3C WoT Thing Architecture [wot-architecture] and WoT Thing Description [wot-thing-description] define a powerful description mechanism and a format to describe myriads of very different devices, which may be connected over various protocols. The format is very flexible and open and puts very few normative requirements on devices that implement it.

However, this flexibility de-facto prevents interoperability, since, without additional **rules**, it allows implementers to make many choices that do not provide **guarantees** of common behavior between implementations.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at https://www.w3.org/TR/.

This document was published by the Web of Things Working Group as an Editor's Draft.

Comments regarding this document are welcome. Please send them to public-wot-wg@w3.org (subscribe, archives).

Publication as an Editor's Draft does not imply endorsement by the W3C Membership.

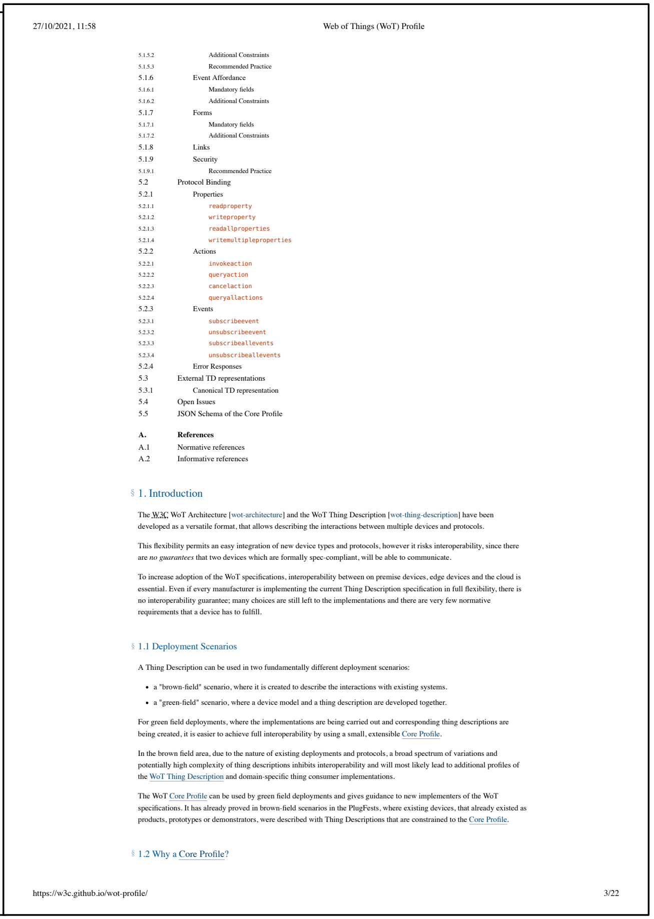
This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

This document was produced by a group operating under the 1 August 2017 W3C Patent Policy. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

This document is governed by the 15 September 2020 W3C Process Document.

Table of Contents

https://w3c.github.io/wot-profile/ 2/22



27/10/2021, 11:58 Web of Things (WoT) Profile

§ 1. Introduction

The W3C WoT Architecture [wot-architecture] and the WoT Thing Description [wot-thing-description] have been developed as a versatile format, that allows describing the interactions between multiple devices and protocols.

This flexibility permits an easy integration of new device types and protocols, however it risks interoperability, since there are *no guarantees* that two devices which are formally spec-compliant, will be able to communicate.

To increase adoption of the WoT specifications, interoperability between on premise devices, edge devices and the cloud is essential. Even if every manufacturer is implementing the current Thing Description specification in full flexibility, there is no interoperability guarantee; many choices are still left to the implementations and there are very few normative requirements that a device has to fulfill.

§ 1.1 Deployment Scenarios

A Thing Description can be used in two fundamentally different deployment scenarios:

- a "brown-field" scenario, where it is created to describe the interactions with existing systems.
- a "green-field" scenario, where a device model and a thing description are developed together.

For green field deployments, where the implementations are being carried out and corresponding thing descriptions are being created, it is easier to achieve full interoperability by using a small, extensible Core Profile.

In the brown field area, due to the nature of existing deployments and protocols, a broad spectrum of variations and potentially high complexity of thing descriptions inhibits interoperability and will most likely lead to additional profiles of the WoT Thing Description and domain-specific thing consumer implementations.

The WoT Core Profile can be used by green field deployments and gives guidance to new implementers of the WoT specifications. It has already proved in brown-field scenarios in the PlugFests, where existing devices, that already existed as products, prototypes or demonstrators, were described with Thing Descriptions that are constrained to the Core Profile.

§ 1.2 Why a Core Profile?

https://w3c.github.io/wot-profile/ 3/22

WoT Architecture

Current work: Terminology Alignment

The WoT Architecture REC defines the common terminology for all WoT specifications.

Other WoT specifications should contain a reference this terminology.

Some specifications have their own terminology section, need to align and move common definitions to architecture.

https://github.com/w3c/wot-architecture/labels/spec%20alignment

# 614 - Additional terminology entries for binding spec
# 613 - Move terminology section from discovery to architecture spec
# 612 - Move terminology from TD to architecture spec
# 611 - Update Terminology regarding Binding Templates

To avoid confusion and potential ambiguities, the terminology section should be normative.

Current work:
Protocol bindings

Recent restructuring of the WoT Binding Templates requires updates of the corresponding introduction section to this guideline.

Restructuring of the binding document goes on, some synchronisation is needed.

Initial discussions have started in the WoT editors call, will be continued in the Architecture call.

Next Steps

- Implementation Report
- Explainer Document

Help and contributions are very welcome.

Publication schedule

Tentative, based on the planned 6 months charter extension.

Architecture Feature Freeze: Jan 31, 2022

CR transition: mid-March

PR transition: mid-April

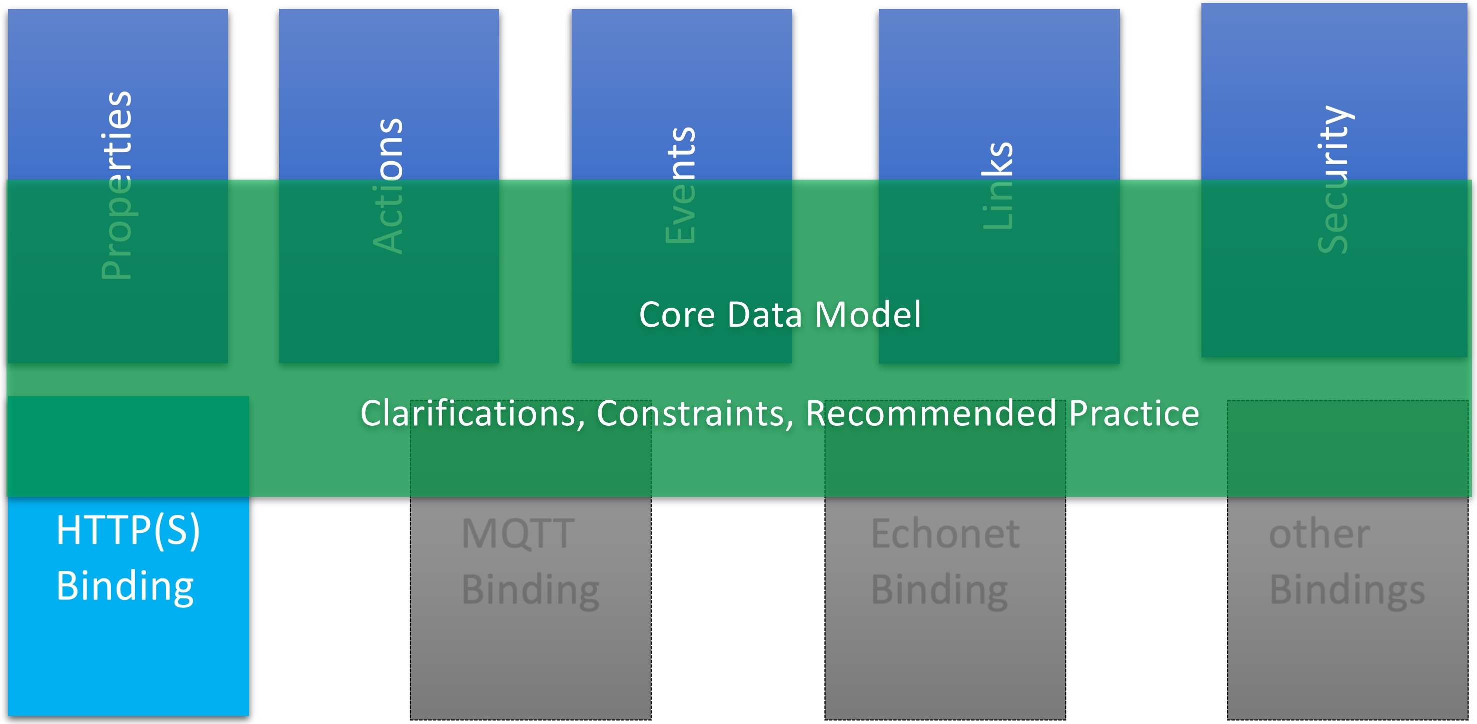
REC transition: before end of extended charter end of July

WoT Profiles

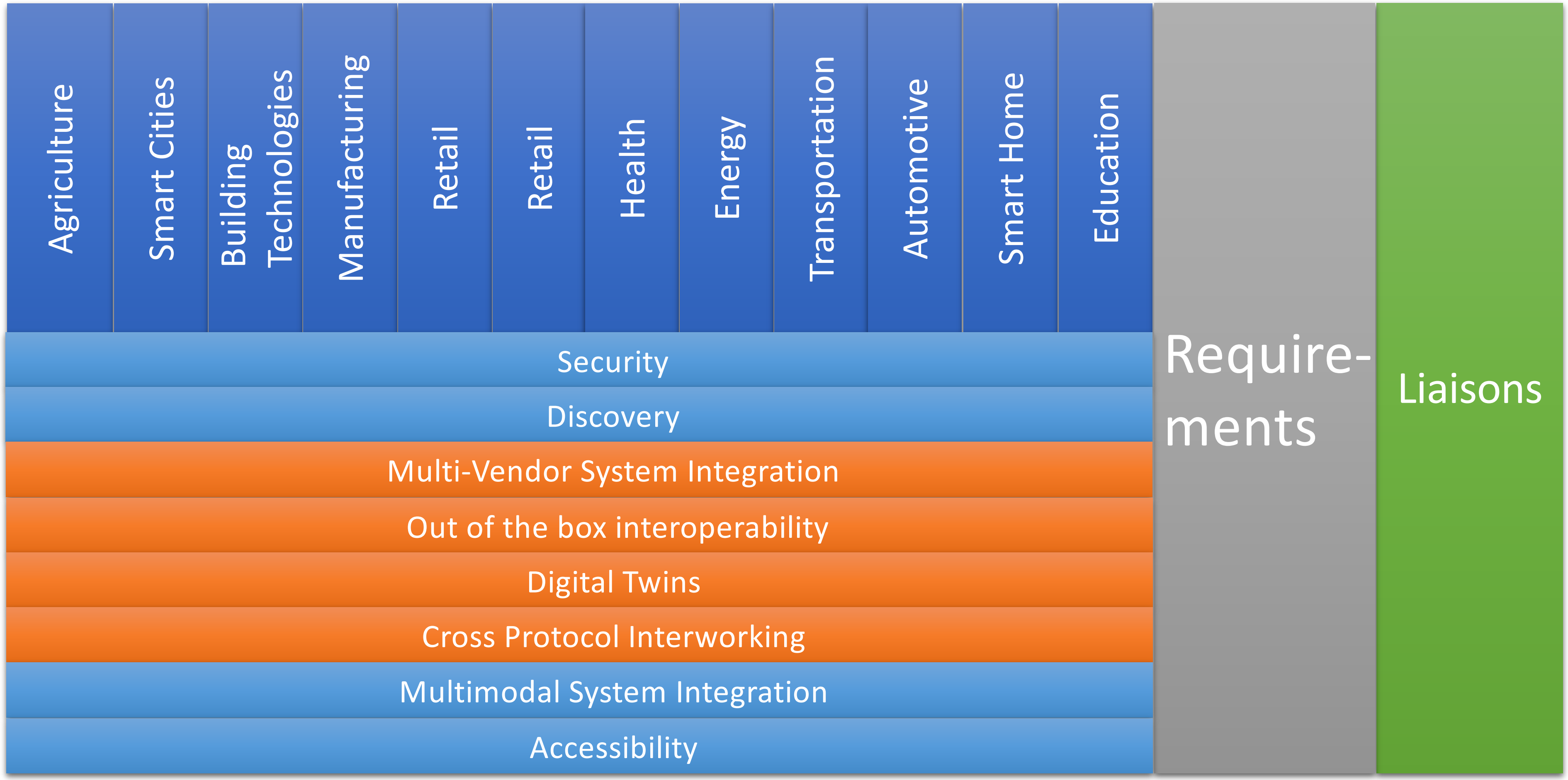
What is a WoT profile?

- A **WoT Profile** is a normative subset of a *WoT Thing Description* with a normative binding to a selected protocol.
- Profiles guarantee **interoperability** between compliant implementations, multiple profiles are possible.
- The **WoT Profile Specification** defines a **normative** set of *constraints and rules* on the **data model**, **representation format** and **protocol binding**.
- These constraints and rules set limitations and make decisions that reduce the complexity for implementers of the WoT standard.
- The rules are prescriptive, to ensure that compliant implementations satisfy the semantic guarantees implied by them.

Core Profile for HTTP



Use Cases for Profiles



Agriculture

Smart Cities

Building Technologies

Manufacturing

Retail

Retail

Health

Energy

Transportation

Automotive

Smart Home

Education

Security

Discovery

Multi-Vendor System Integration

Out of the box interoperability

Digital Twins

Cross Protocol Interworking

Multimodal System Integration

Accessibility

Require-ments

Liaisons

Use Case
Cross protocol interworking

In smart city, home and industrial scenarios various devices are connected to a common network. These devices implement different protocols.

To enable interoperability, an "agent" needs to communicate across different protocols. Platforms for this agent can be edge devices, gateways or cloud services.

Interoperability across protocols is a must for all user scenarios that integrate devices from more than one protocol.

Core requirements: Common data model across protocols,

Use Case
Out of the box interoperability

As a device owner, I want to know whether a device will work with my system before I purchase it to avoid wasting money.

As a developer, I want to be able to validate that a Thing will be compatible with a Consumer without having to test against every possible consumer.

As a cloud provider I want to onboard, manage and communicate with as many devices as possible out of the box. This should be possible without device specific customization.

Core requirements

Data Model:

- common metadata
- common unit/metrics system
- Common time formats
- Constrained data model across protocols
- Naming conventions

Protocol binding:

- Unambiguous protocol binding
- Clarifications and constraints
- Fully specified interaction semantics for properties, actions and events
- Fully specified error behaviour

Current work

- Defining a core/baseline profile with a HTTP binding.
- Identifying constraints and rules on the data model.
- Unambiguous interaction semantics for properties, actions and events.
- Constraints on payload formats.
- Protocol binding semantics, e.g. headers, response codes.
- Best practice security requirements.
- Compliance - TD Validation

Open issues

Core data model section

- Ongoing discussion in several PRs
- Current constraints need to be reworked
- One TF member proposed removal of the entire section, this would break cross-protocol interoperability use cases and impact OOTB interoperability

Event model limitations

- Current model is a “Consumer Pull model” using SSE
- Requires an open socket connection for each subscription
- Does not support firewalls
- Does not scale to cloud scenarios
- Works only in closed networks

Proposal for a “Push” event model (1)

In typical cloud scenarios devices are not reachable from the internet

- Devices send messages and events via firewalls
- Only on-demand network traffic, no open connection if no messages

Pull model wastes significant resources

- open connections to thousands of devices even if no traffic
- Network traffic overhead due to timeouts and reconnects

Webhooks over HTTP are industry standard for scalable event systems, see for example:

- Github: https://docs.github.com/en/developers/webhooks-and-events
- Paypal: https://developer.paypal.com/docs/api-basics/notifications/webhooks/
- Atlassian: https://developer.atlassian.com/server/jira/platform/webhooks/
- Wordpress: https://wordpress.com/support/webhooks/

Proposal for a “Push” event model (2)

In typical cloud scenarios devices are not reachable from the internet

- Devices send messages and events via firewalls
- Only on-demand network traffic, no open connection if no messages

Pull model wastes significant resources

- open connections to thousands of devices even if no traffic
- Network traffic overhead due to timeouts and reconnects

Webhooks over HTTP(S) are commonly used for scalable event systems, see for example:

- Github: https://docs.github.com/en/developers/webhooks-and-events
- Paypal: https://developer.paypal.com/docs/api-basics/notifications/webhooks/
- Atlassian: https://developer.atlassian.com/server/jira/platform/webhooks/
- Wordpress: https://wordpress.com/support/webhooks/

Push event model

Follow Industry best practice:

- Use Webhooks over HTTP(S) for push events
- Use cloud events as message payload format (https://github.com/cloudevents/spec/blob/v1.0.1/spec.md)
- Provide additional clarifications:
  - Retry and reconnect behaviour
  - Error behavior
  - Standard attributes

What are Cloud Events?

A specification for describing event data in a common way.

Spec has been developed by several companies, including Alibaba, Amazon, Google, Huawei, IBM, Intel, Microsoft, Oracle, PayPal, Red Hat, SAP, Serverless Inc, SolarWinds, VMWare.

It defines a type system, context attributes and a payload format.

Open source implementations for various languages, including Java, go, C#, javascript, python are available.

https://cloudevents.io

Cloud Event example (JSON)

```
{
"specversion" : "1.0",
"type" : "com.github.pull_request.opened",
"source" : "https://github.com/cloudevents/spec/pull",
"subject" : "123",
"id" : "A234-1234-1234",
"time" : "2018-04-05T17:31:00Z",
"comexampleextension1" : "value",
"comexampleothervalue" : 5,
"datacontenttype" : "text/xml",
"data" : "<much wow=\"xml\"/>"
}
```

Next steps

- “Push” event mechanism
- Rework data model section
- Implementation Report
- Compliance Section – JSON Schema for profile
- Authentication, Security
- Incorporate Plug Fest findings

Future Work – next charter period

Support for other protocol bindings

Candidates include:

- MQTT
- Echonet Lite
- OPC-UA

Include TD 1.1 features

- E.g. Thing Models

Include TD 2.0 features

Publication schedule

Tentative, based on the planned 6 months charter extension.

Profile Feature Freeze: Jan 31, 2022

CR transition: mid-March

PR transition: mid-April

REC transition: before end of extended charter end of July

References

WoT Architecture task force

- https://www.w3.org/WoT/activities/tf-architecture/

WoT Architecture repository / working draft

- https://github.com/w3c/wot-architecture

WoT Profile repository / working draft

- https://github.com/w3c/wot-profile

- Wot Use Cases – editors draft:
- http://w3c.github.io/wot-usecases

Backup

PRs

PRs

RFC 2119 markup: add css mark <span> into index.html

- https://github.com/w3c/wot-profile/pull/79

Discussion:

We noticed that some span markups are in editors notes – these need to be cleaned up in a further iteration.

We merged the PR, Mizushima-san will provide additional comments.

McCool will do another pass using the test tooling.

References

WoT Use Cases repository

- https://github.com/w3c/wot-usecases