

IoT Interoperability with W3C Web of Things

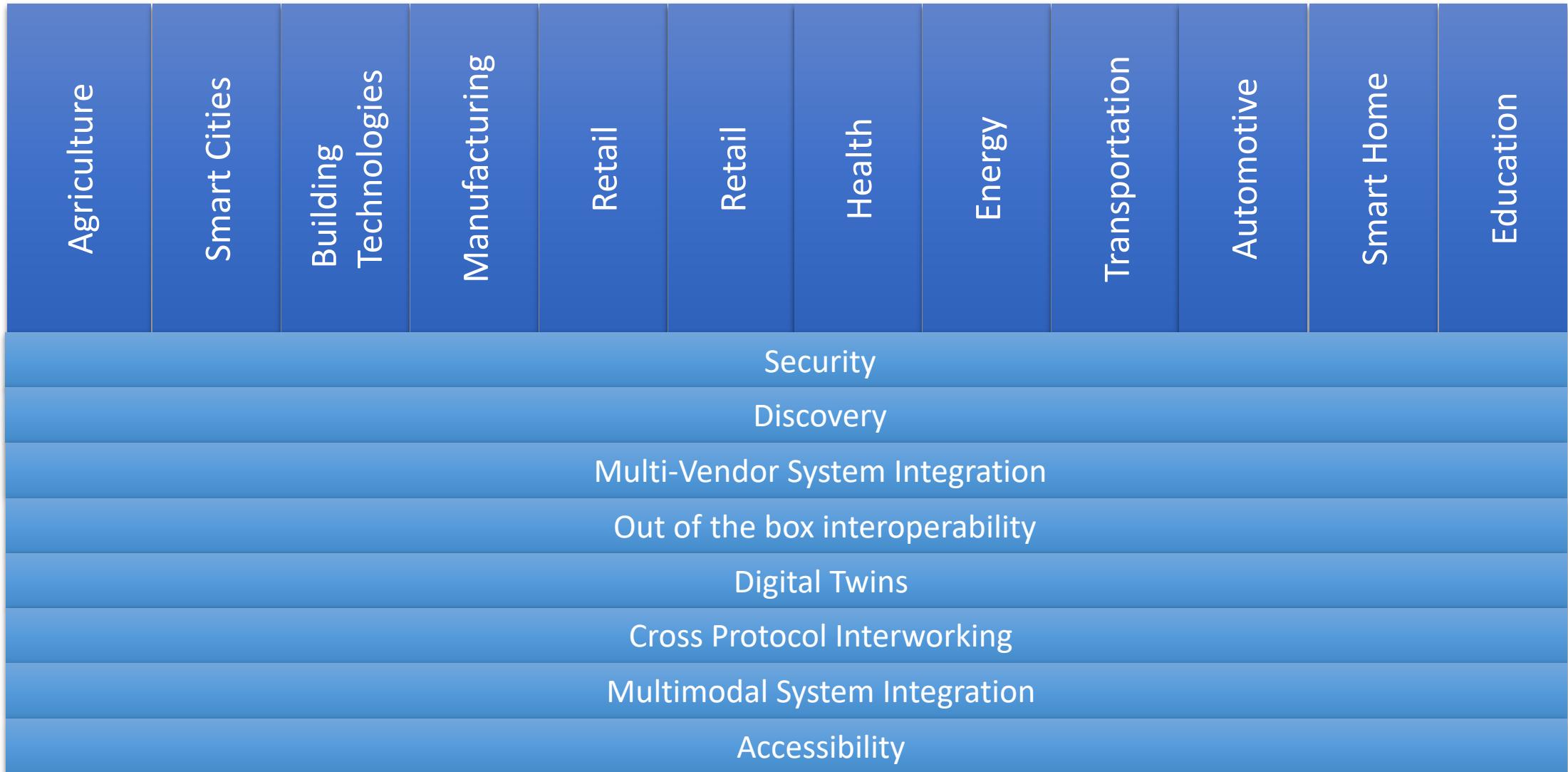
Michael Lagally

Jan 2022

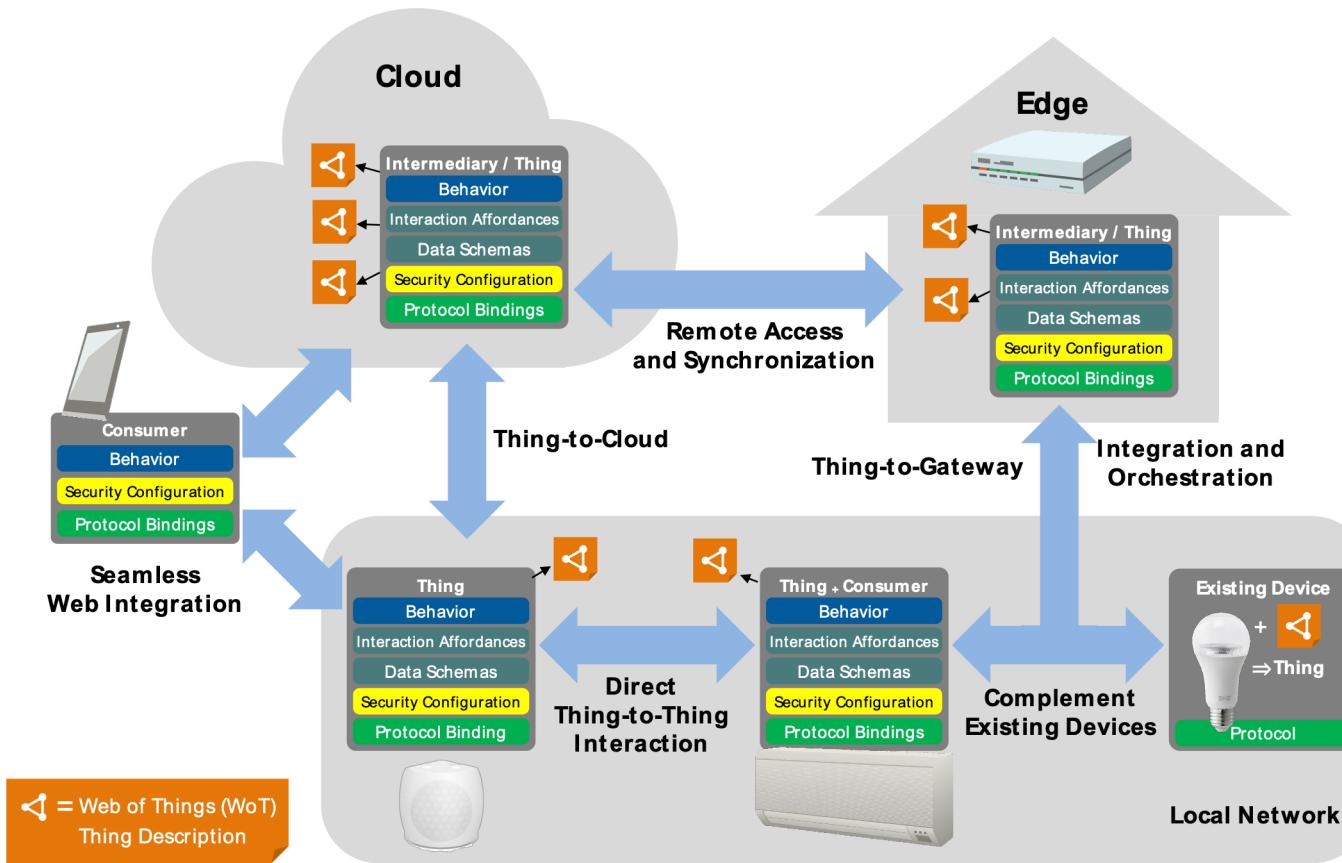
W3C Web of Things

Use Cases and Concepts

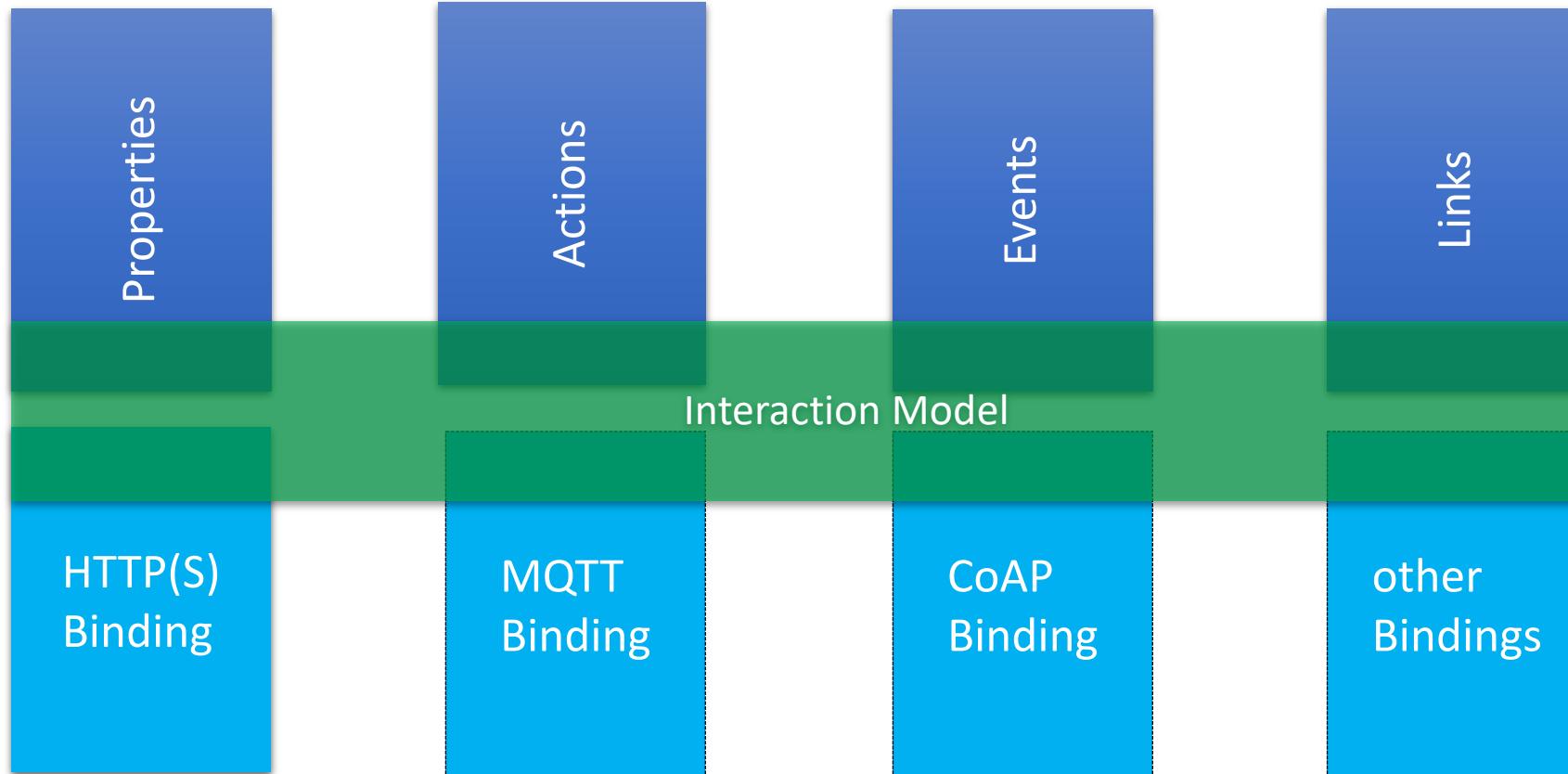
Web of Things: Horizontal and Vertical Use Cases



W3C Web of Things – Abstract Architecture



Web Things: A common interaction model across protocols



WoT Thing Description – A common description language

WoT Thing Descriptions define a protocol agnostic information model which can be augmented with semantic annotations to add context knowledge.

The interaction affordances of that model can be bound to different protocols, and a single WoT Thing Description can contain bindings for several protocols.

Thing Descriptions are typically encoded in a JSON format that also allows JSON-LD (semantic linked data) processing.

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "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": [
        {"href": "https://mylamp.example.com/oh",
         "subprotocol": "longpoll"}
      ]
    }
  }
}
```

What is out of the box
interoperability OOTBI?

User Scenarios

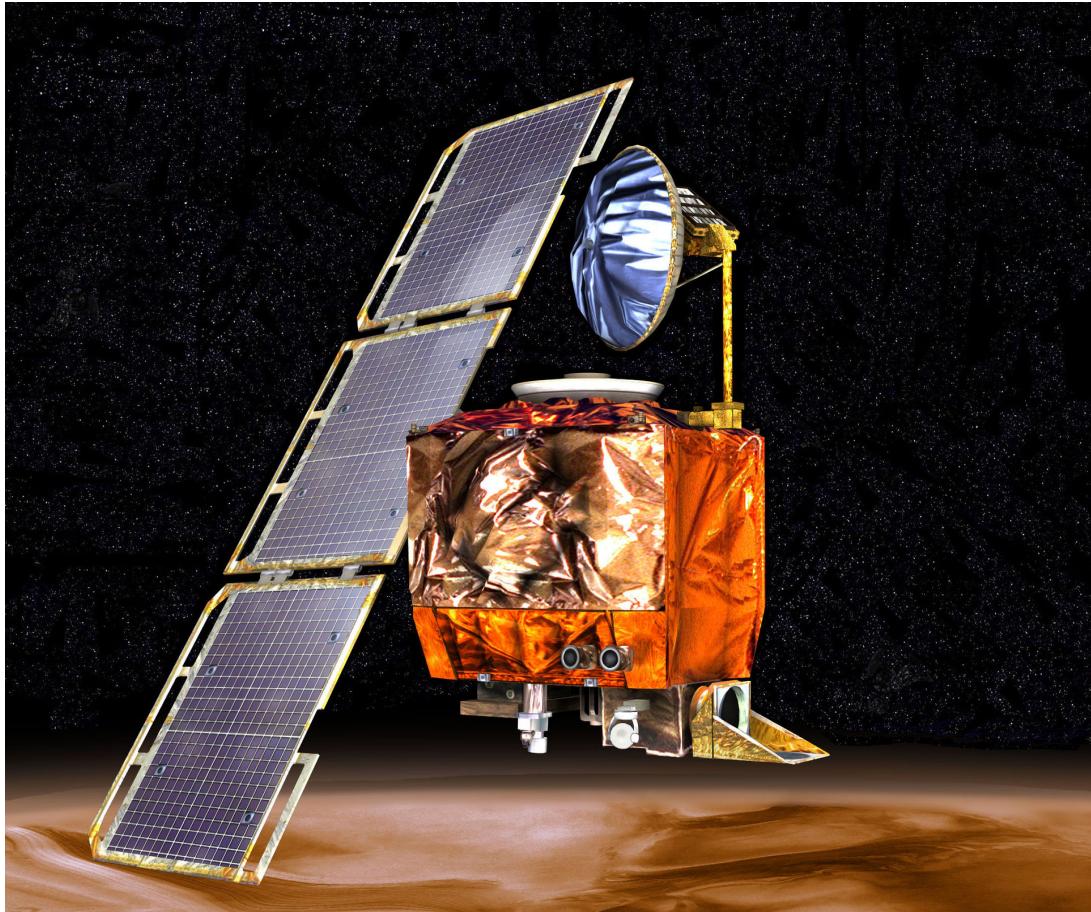
- As an end user, I want to know whether a device will work with my system before I purchase it to avoid wasting money.
- As a developer, I want TDs to be as simple as possible so that I can efficiently develop them.
- As a developer, I want to be able to validate that a Thing will be compatible with a Consumer without having to test against every possible consumer.
- As a System Integrator, I want to integrate devices from multiple vendors without deep customisation.

Out-of-the-box interoperability

- Devices from multiple vendors can be integrated together 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.

Why do we need OOTBI?

Mars Climate Orbiter (1999)



- The primary cause of this discrepancy was that one piece of ground software produced results in a **United States customary unit, contrary to its Software Interface Specification (SIS)**, while a second system, supplied by NASA, expected those results **to be in SI units**, in accordance with the SIS. Specifically, software that calculated the total impulse produced by thruster firings produced results in pound-force seconds. The trajectory calculation software then used these results – expected to be in newton seconds – to update the predicted position of the spacecraft.

Source: https://en.wikipedia.org/wiki/Mars_Climate_Orbiter

How could W3C WoT have helped to prevent that?

- There was a clear system description describing the measurement system (SI units).
 - One company was not using the right metric system
 - They made implicit assumptions
-
- WoT Thing Descriptions include a **unit**, which implies that people think about the interpretation of the associated value.
 - A Profile can enforce using units for measurable values.

Example OOTBI Scenario: Telemetry

A worldwide climate monitoring system obtains data from sensors and gateways around the world to indicate the current weather conditions and to be able to predict critical conditions.

The system displays a world map with all sensors, where the user can zoom in to individual regions.

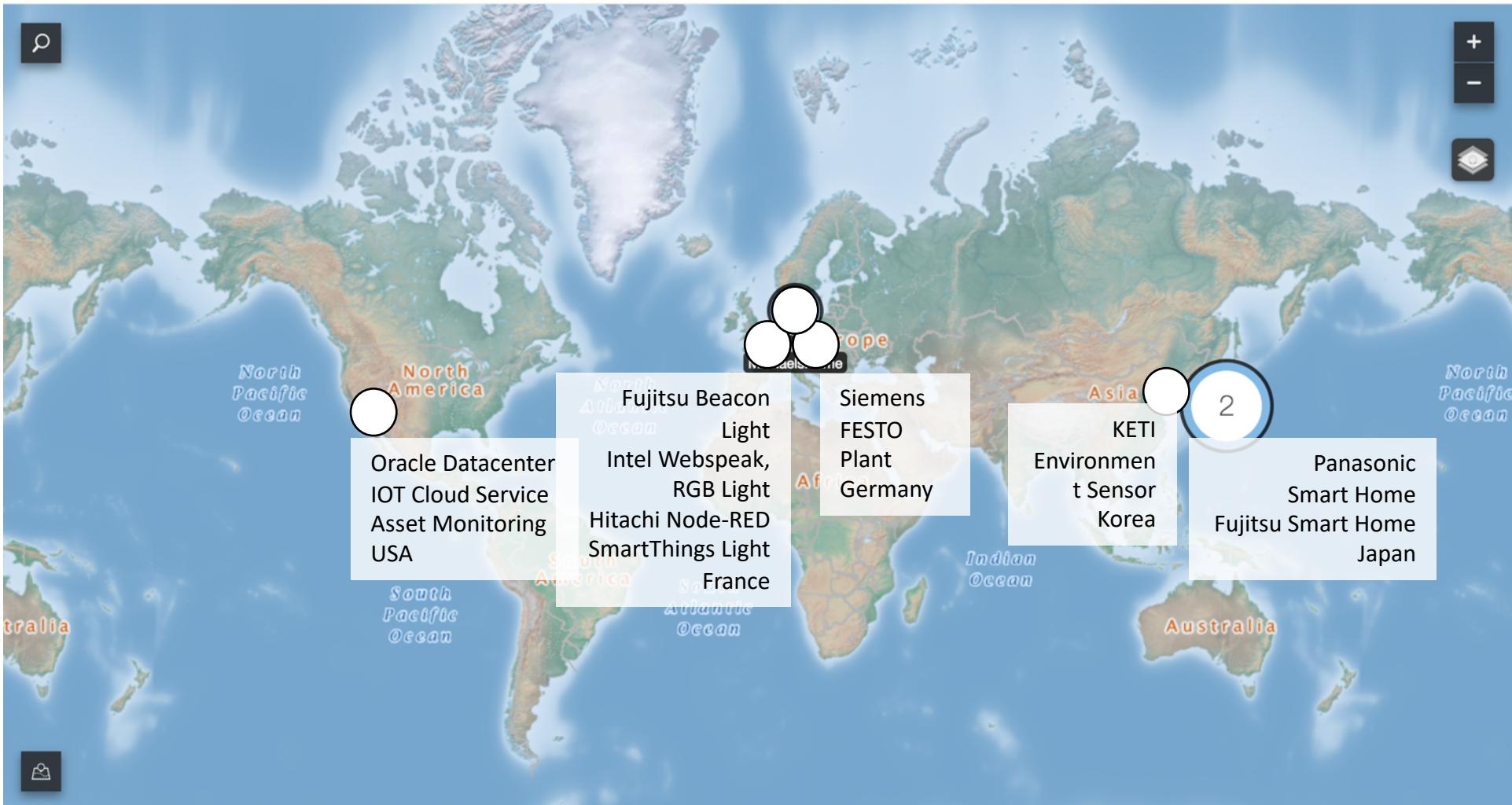
Temperature, humidity and other sensor readings etc. are provided to a common server, which aggregates the data and uses configurable rules to trigger alerts based on sensor data.

This example motivates to consider the following aspects:

- All sensors and gateways must use unit schemes that are known and can be interpreted by the consumer.
- All sensors and gateways must use an unambiguous time and date format.
- All sensors and gateways must provide a human readable name that can be displayed on a map.
- All sensors and gateways must provide their location in a format that is known to the consumer.
- If a sensor and gateways provides interactions, these must be displayed in a UI in a human readable form.

Sensor readings will be displayed in a UI, the names must be displayed in a UI in a human readable form.

WoT Plugfest Scenario



System Integration Challenges

For a system integrator all these four interoperability layers are required to ensure OOTBI:

- Technical Interoperability: OOTBI at protocol layer, use of HTTP(s), authentication mechanisms, self descriptive things (discovery)
- Syntactic Interoperability: well defined JSON format, validation, max lengths, common field names, guaranteed fields, common error response format, event payload, error codes, unambiguous mapping of HTTP protocol verbs (e.g. PUT vs. POST).
- Semantic Interoperability: consistent data model across things, e.g. unambiguous units, date + time representation, fields have the same "meaning", i.e. **common interpretation of values** across devices and vendors.
- Organisational Interoperability: things have the same **behavior**, if they are used in the same way. This is independent from the manufacturer, geographical location, and other situation context. Behavior is independent from the network topology or protocol stack.

See also: https://european-iot-pilots.eu/wp-content/uploads/2018/11/D06_02_WP06_H2020_CREATE-IoT_Final.pdf

WoT Profiles

Why, What, How?

WoT Interoperability: Why do we need a WoT Profile?

The WoT Architecture and Thing Description define a generic powerful description mechanism and a machine-readable format.

The format is very flexible and open and puts very few normative requirements on devices that implement it.

Without additional rules, WoT's flexibility allows implementers to make many choices that do not provide guarantees of common behavior between implementations.

What is a WoT Profile?

The Web of Things (WoT) Profile specification is being proposed to address these challenges and to enable *out-of-the-box interoperability*.

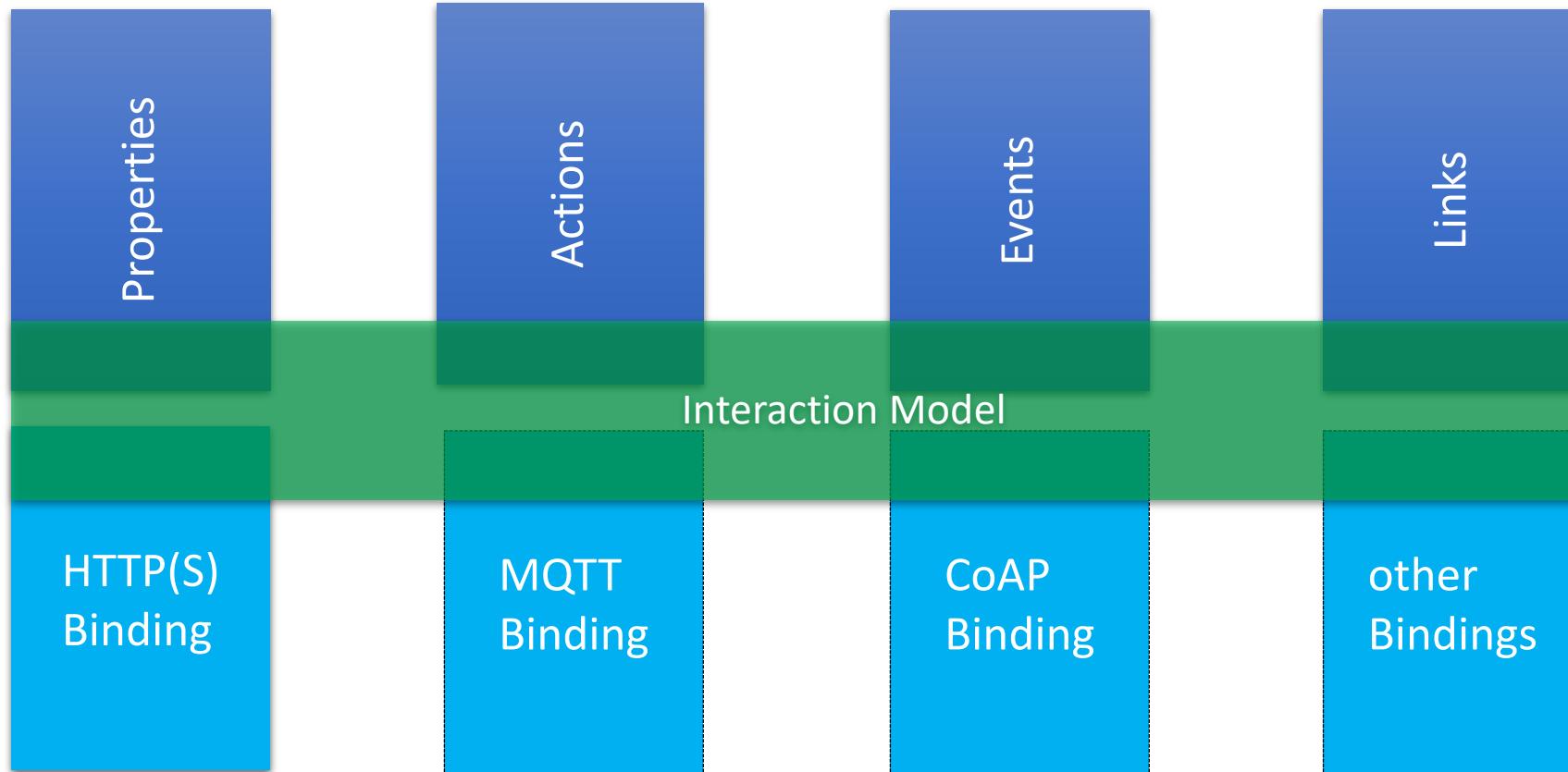
Things conforming to the same profile should work together without custom adaptation other than the usual security onboarding and network configuration.

The WoT Profile specification defines a set of common constraints on the information model, and rules which compliant Thing Descriptions must adopt to achieve interoperability.

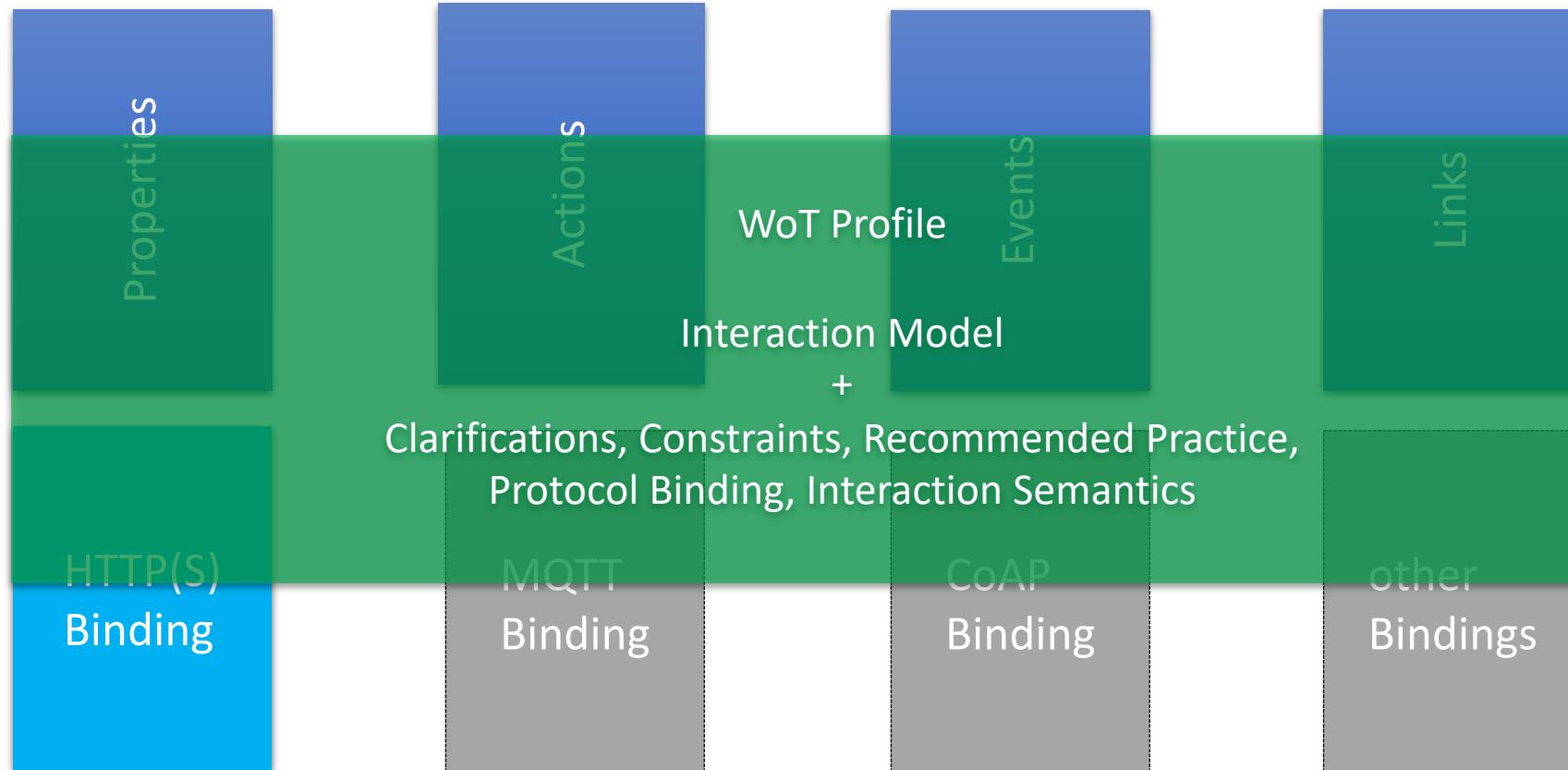
These rules are prescriptive, to ensure that compliant implementations satisfy the semantic guarantees implied by them.

This set of rules is called a Profile.

How to define a Profile?



How to define a Profile?



How to define a Profile

A set of rules which compliant Web Things must adopt to achieve interoperability.

These rules are prescriptive, to ensure that compliant implementations satisfy the semantic guarantees implied by them.

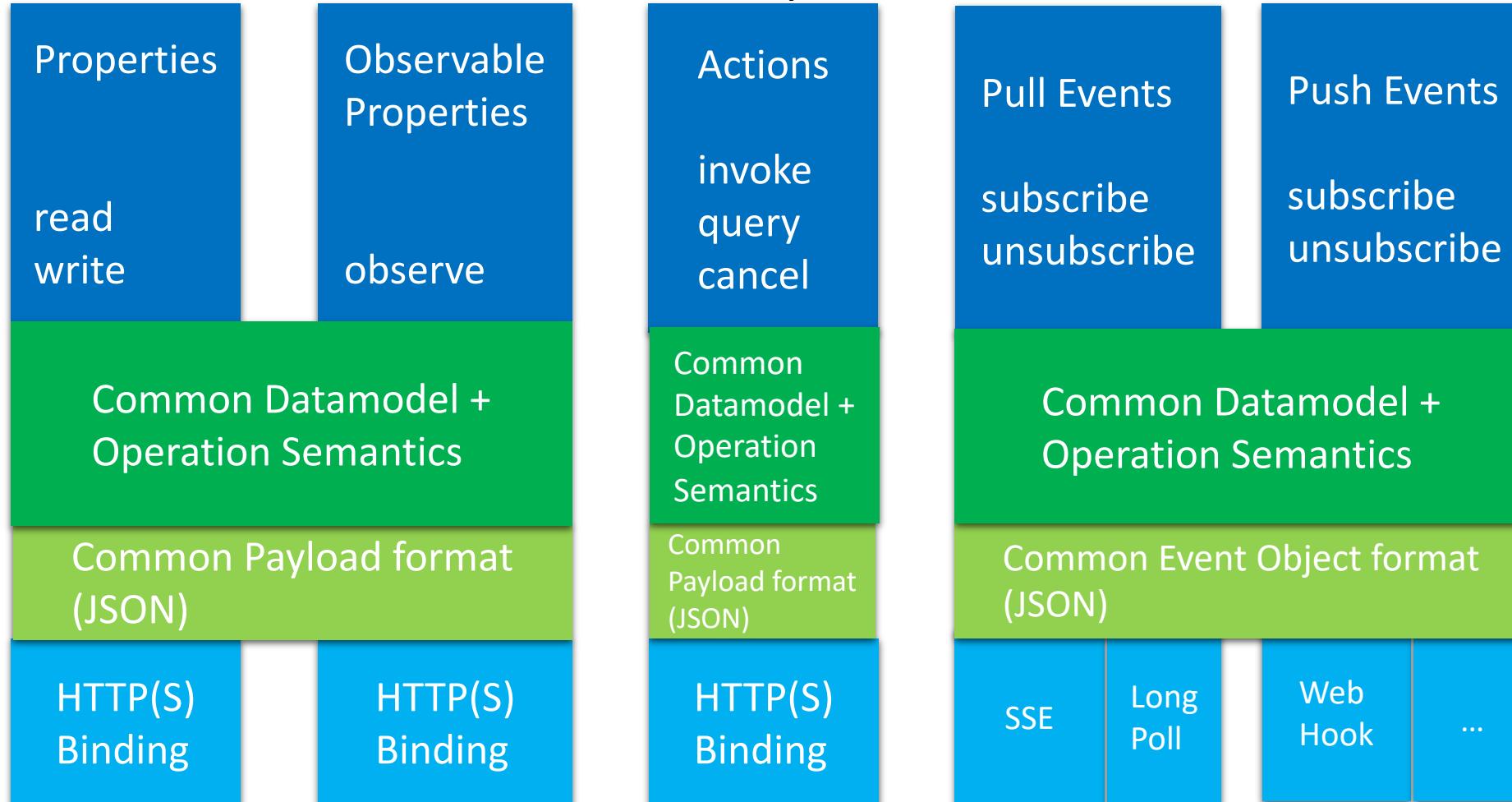
- Common constraints on the information model, e.g. nesting depth, mandatory metadata fields.
- Clarifications on data types (e.g. date and time) and unit systems.
- Unambiguous HTTP(s) protocol binding, common error behavior.
- Common well-defined behavior of interactions for properties, synchronous and asynchronous actions, events.
- Well-defined payload formats for error responses, action status and events.

The WoT HTTP + JSON profile

- HTTP(s) + JSON Protocol Binding
- Metadata requirements and type clarifications
- Readable and writable properties
- Synchronous and asynchronous actions
- Scalable event model
- Validation with JSON schema
- Guidelines and recommended practice

WoT HTTP + JSON Profile

Common Datamodel + Operation Semantics



Summary and Conclusion

- All WoT specification drafts are available on their respective github repositories.
- The architecture task force in the WoT working group is working towards completion of the HTTP + JSON profile specification.
- Future profiles are envisioned to be defined based on use cases and requirements for other application domains and protocols.
- If you would like to contribute to WoT use cases or participate in the profile work, please contact the author.

References

W3C Web of Things: <https://www.w3.org/WoT>

WoT Use Cases and Requirements: <http://w3c/wot-usecases>

WoT Architecture: <https://github.com/w3c/wot-architecture>

WoT Thing Description: <https://github.com/w3c/wot-thing-description>

WoT Profile: <https://github.com/w3c/wot-profile>