



Cisco Community Expert series Webcast

Introduction to the Web of Things: What is it and why is it useful?

Ines Robles- IoT Consultant and IT Researcher

October 16th 2018

News & Upcoming events



Ask the Expert following the Webcast

Now through Friday October 19th 2018

With
Ines Robles

<http://bit.ly/Ask-Expert-WoT>



Ines Robles
IoT Consultant & IT Reseracher



Cisco Community – Ask the Expert



MPLS Best Practices- Configuration & Troubleshooting

Till Friday
26th, October 2018

With
Cisco VIP
David Peñaloza

The banner features a blue header with a photo of David Peñaloza and the text "Ask the Expert". Below this, the main title "Solve your questions about MPLS Configuration & Troubleshooting" is displayed in white over a background image of two men sitting on a yellow sofa in a modern office setting, looking at a laptop. A large laptop screen in the foreground shows network configuration data. The bottom of the banner is a blue bar with the "VIP 2018" logo, the dates "15 -26 OCTOBER 2018", and the Cisco logo.

<http://bit.ly/Ask-MPLS>

Cisco Community – Ask the Expert in Spanish

VPN on FTD- Configuration & Troubleshooting

Till Friday 19th, October 2018

-For Customers & Partners-

With David Rivera
& Román Hernández

<http://bit.ly/Pregunte-VPN-FTD>

Pregunte al Experto

Aclare sus dudas de la Configuración y troubleshooting de VPN en FTD

David & Roman

CISCO

8 - 19 OCTUBRE, 2018
-Evento abierto para Clientes y Partners-

¡Haga Preguntas!

Become an event Top Contributor!

Participate in Live Interactive Technical Events and much more

<http://bit.ly/EventTopContributors>



A screenshot of the Cisco Community website's "Events Top Contributors" page. The page header includes the Cisco logo and "Cisco Community" text. Below the header is a search bar and a navigation menu with icons for Technology & Support, For Partners, Customer Connection, Events, and Members & Recognition. The main content area shows the breadcrumb "Cisco Community / Events Top Contributors" and the title "Events Top Contributors". A paragraph explains the program: "This program recognizes Cisco experts in the Cisco Community (CSC) that host technical events (Webcasts, Ask the Experts, Tech Talks, and Facebook Forums.) With this program, Cisco recognizes the positive, valuable influence that our top Cisco experts exert on the communities. To learn more, please visit our FAQs". Below this is a "2014 2013" filter and two contributor profiles: Julio Carvajal and Ryota Takao. A small video thumbnail is visible on the right side of the page.

Cisco Designated VIPs

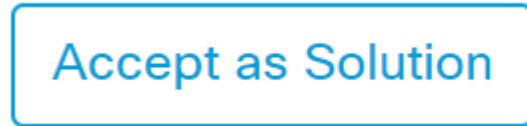
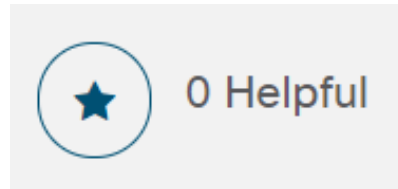


The Cisco Designated VIP program recognizes the top external individual contributors in Cisco's online communities, including the Cisco Support Community (CSC), Cisco Learning Network (CLN) and the Cisco Developers Network (CDN). Cisco Designated VIPs are recognized by their peers for their expertise and tireless contributions, and their abundant participation is vital to community success. With this program, Cisco formally recognizes the positive, valuable influence our top individual members exert on the communities overall. [FAQs](#)

Rate content at the Cisco Community

Help us to recognize the quality content in the community

Rate documents,
Videos & blogs!



Encourage and acknowledge people who
generously share their
time and expertise



Cisco Community Expert



Ines Robles
IoT Consultant & IT Researcher

Thank You For
Joining Us Today!



Download Today's Presentation
<http://bit.ly/WebcastSlides-WoT>

Submit Your Questions Now!

Use the **Q&A** panel to submit your questions and the panel of experts will respond.

They will be answered eventually



Please take a moment to complete the survey at the end of the webcast

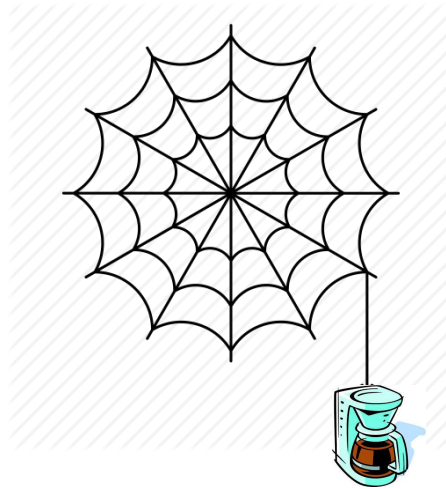
New Webex!

The screenshot shows a Cisco Webex Events window with a menu bar (File, Edit, Share, View, Communicate, Participant, Event, Help) and a toolbar. A sharing menu is open, with 'Screen /presso' highlighted in a red box. The main content area displays a presentation slide titled 'VxLAN Overview' with a network diagram. The diagram shows a central 'IP Network (Underlay)' cloud connected to four 'Edge Device' nodes. Each edge device is connected to a 'Local LAN Segment', which in turn connects to either a 'Physical Host' or 'Virtual Hosts' (via a 'Virtual Switch'). An 'IP Interface' is also shown within the IP Network cloud. A URL is visible at the bottom of the slide: <http://opendata.labs.lacnic.net/ipv6stats/graphs/ipv6evo.html>. The bottom of the window features a control bar with icons for call, video, share, stop, mute, chat, and close.

Please make sure you follow up the presentation in the right screen

WoT: Web of Things

“Turning smart objects into first-class citizens of the Web”



Maria Ines Robles
maria.robles@aalto.fi
2018

Polling Question 1

What is the Web of Thing (WoT) for you?

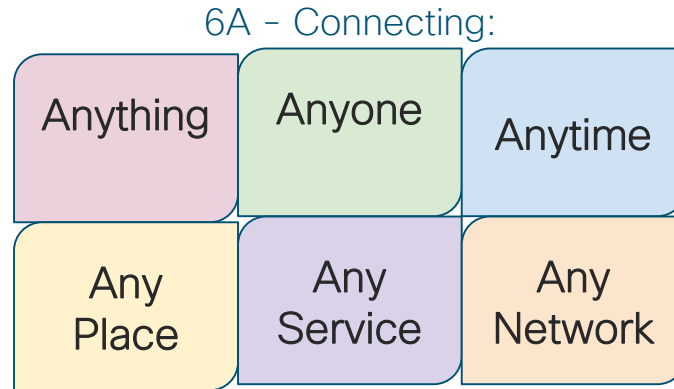
- A. A web page showing connected things on it.
- B. The WoT is an architecture that enables interoperability across IoT Platforms.
- C. Is Internet of Things at Network layer
- D. Is a description of a HTML web page.

Outline

- Internet of Things
- Web of Things:
 - Introduction
 - Use Cases
 - WoT Building Blocks:
 - Thing
 - WoT Thing Description
 - WoT Binding Template
 - WoT Scripting API
- WoT Servient
- Open Source
- Takeaways

"Internet of Things" (IoT):

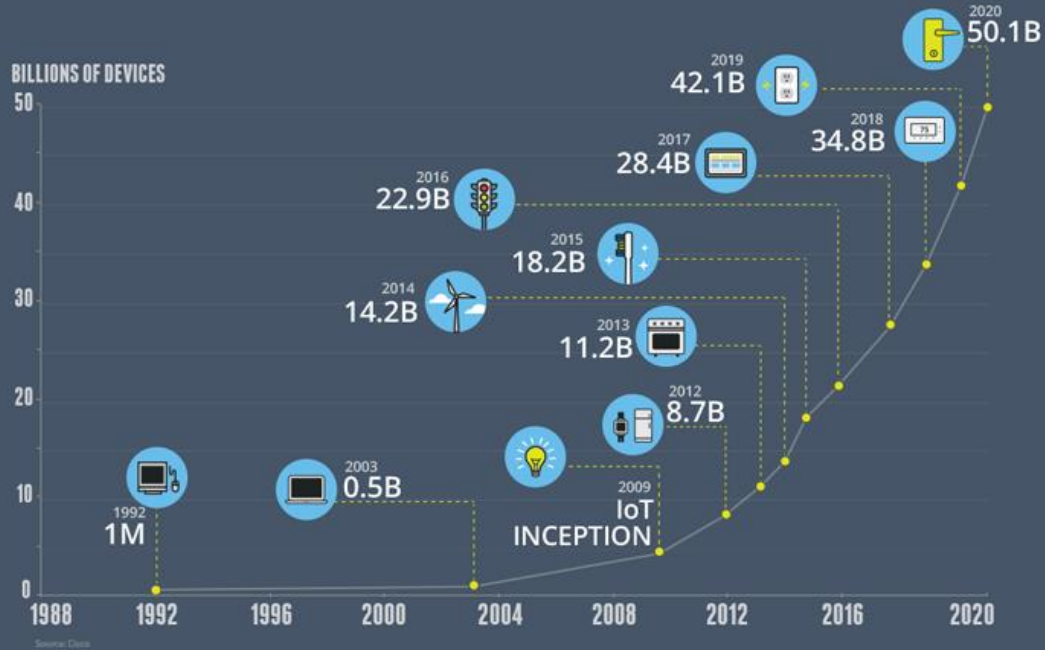
- "Everything that can be connected will be connected to Internet".



http://4.bp.blogspot.com/-QK1xQB5VOs4/VDV9vID7IjI/AAAAAAAAAD6Y/7VOnwsUy7Go/s1600/IoT_Connecting_6.png

GROWTH IN THE INTERNET OF THINGS

THE NUMBER OF CONNECTED DEVICES WILL EXCEED **50 BILLION** BY 2020



2

<http://theconnectivist-img.s3.amazonaws.com/wp-content/uploads/2014/05/Unknown.png>

Web of Things

What is the goal of the Web of Things?

One of them is to avoid this:



Source: <https://imgur.com/gallery/cqrPQHd>

What is the goal of the Web of Things?

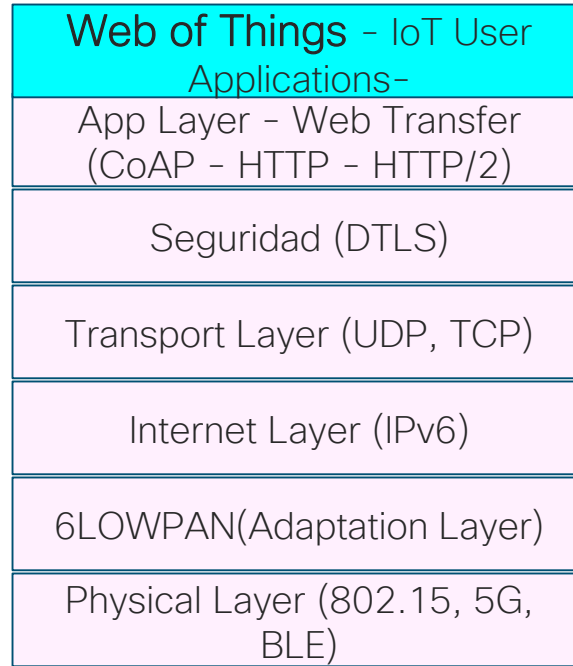
The W3C Web of Things (WoT) is intended to enable **interoperability** across IoT Platforms and application domains. Primarily, it provides mechanisms to formally describe IoT interfaces to allow IoT devices and services to communicate with each other, **independent of their underlying implementation**, and across multiple networking protocols. Secondly, it provides a standardized way to define and program IoT behavior.

Source: <https://w3c.github.io/wot-architecture/>

What is the WoT?

The WoT is an architecture to enable Internet of Things devices to interact through Web standards.

Web of Things location in an IoT protocol stack



Who develops the Web of Things?

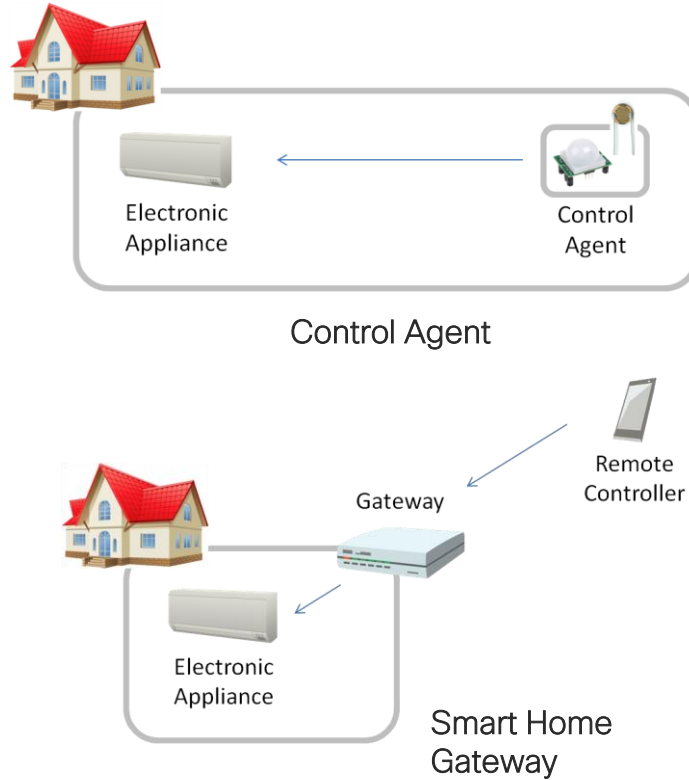
The World Wide Web Consortium (W3C) is shared by 2 working groups:

The WoT Interest Group (WoTIG): provides a forum for discussion of use cases and requirements, for investigation of areas that need further study before transfer to the standardization track, and for outreach and collaboration with industry alliances and standards development organizations. The Interest Group will take the lead on work on testing, e.g., organization of PlugFest events, test frameworks, and joint work with other organizations on interoperability testing.

The WoT Working Group (WoTWG): This working group is tasked with the standardization on those aspects that the WoTIG believes are mature enough to progress to W3C Recommendations

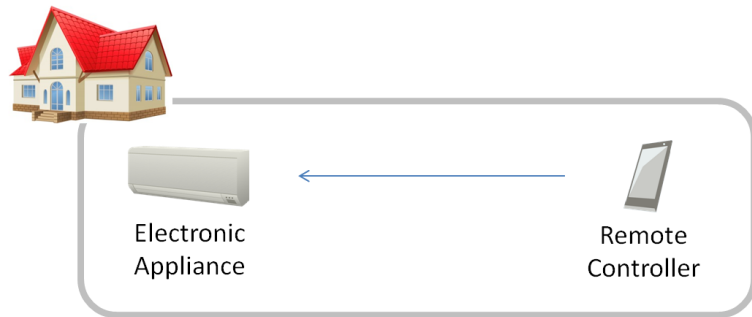
Source: <https://www.w3.org/2016/12/wot-wg-2016.html#coordination>

Web of Thing Use Case: Smart Home

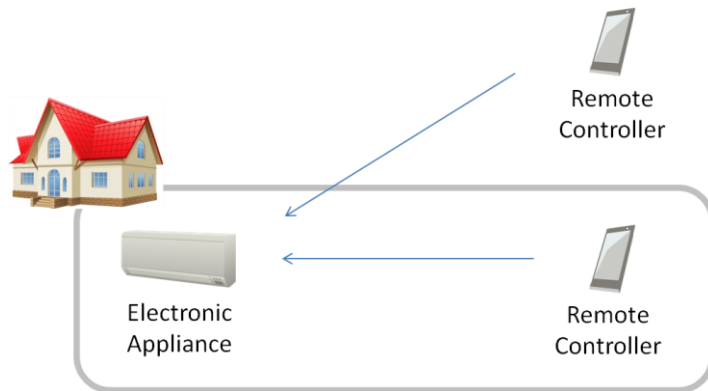


Source: <https://w3c.github.io/wot-architecture/>

Web of Thing Use Case: Smart Home



Device Control

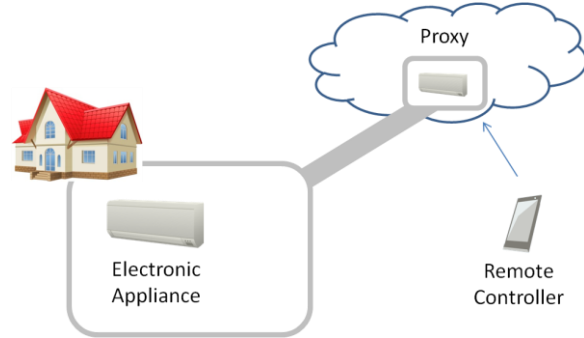


Multiple Network Interfaces

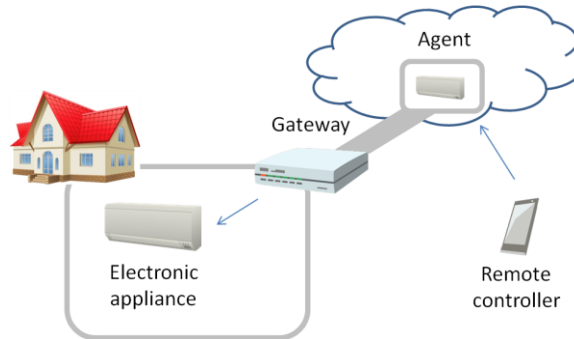
Source: <https://w3c.github.io/wot-architecture/>

© 2018 Cisco and/or its affiliates. All rights reserved.

Web of Thing Use Case: Smart Home



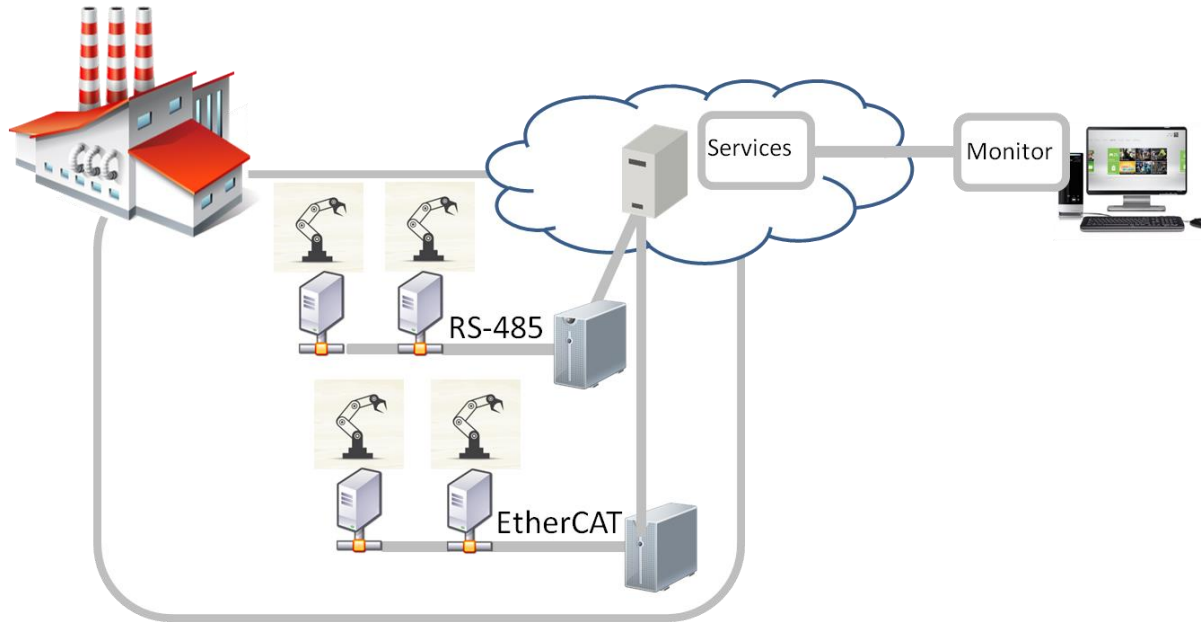
Proxies with Cloud-ready Devices



Proxies with Legacy Devices

Source: <https://w3c.github.io/wot-architecture/>

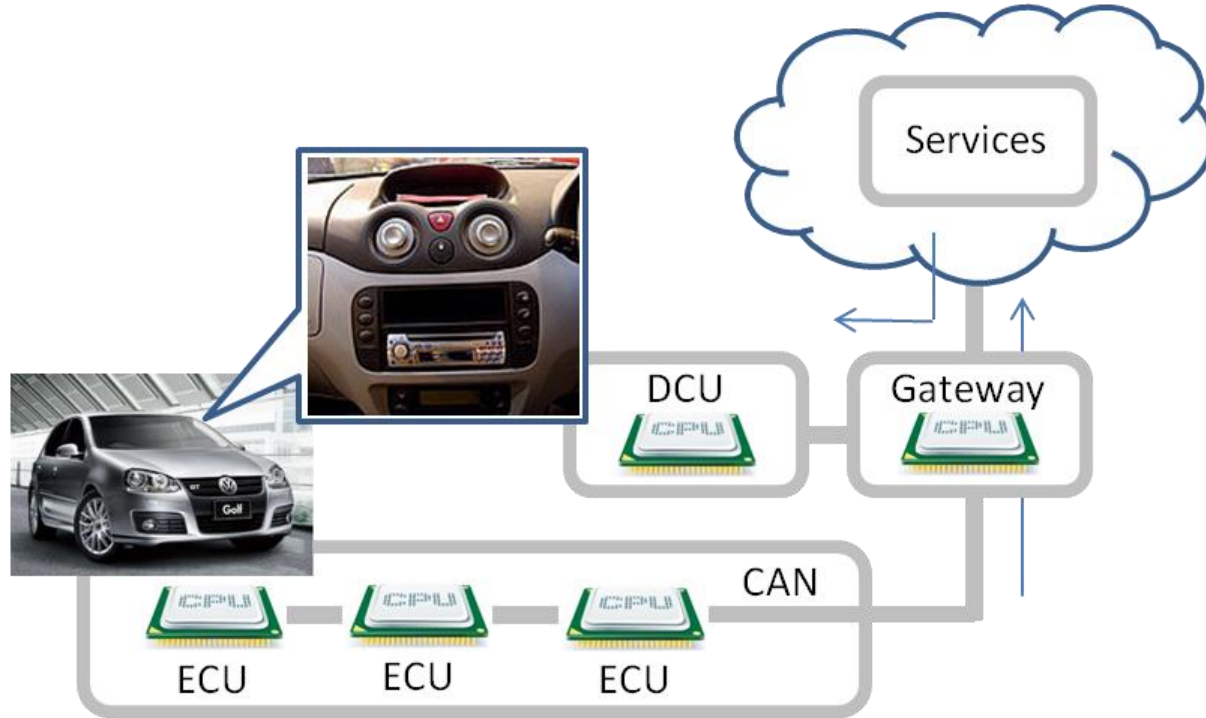
Web of Thing Use Case: Smart Factory



Source: <https://w3c.github.io/wot-architecture/>

© 2018 Cisco and/or its affiliates. All rights reserved.

Web of Thing Use Case: Connected Car



Source: <https://w3c.github.io/wot-architecture/>

Functional Requirements

- Flexibility
- Compatibility
- Security and Privacy

Source: <https://w3c.github.io/wot-architecture/>

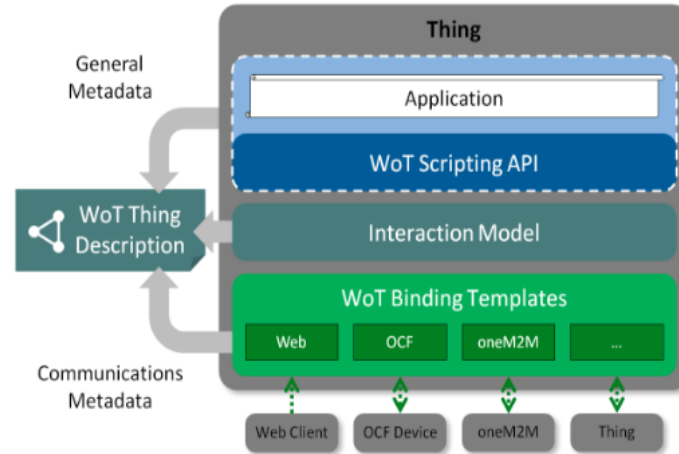
Conceptual Architecture of the WoT Building Blocks

Thing

The WoT Thing Description

The WoT Binding Templates

The WoT Scripting API.



Source: <https://w3c.github.io/wot-architecture/>

Thing

A Thing is the abstraction of a physical or virtual entity that needs to be represented in IoT applications. This entity can be a device, a logical component of a device, a local hardware component, or even a logical entity such as a location (e.g., room or building).



Source: <https://w3c.github.io/wot-architecture/>

Thing

Field Name	Description	Mandatory	Default value	Type
id	unique identifier of the Thing (URI, e.g. custom URN)	yes	.	<u>anyURI</u>
description	Provides additional (human) readable information.	no	.	<u>string</u>
name	Name of the Thing.	yes	.	<u>string</u>
support	Provides information about the TD maintainer (e.g., author, link or telephone number to get support, etc).	no	.	<u>string</u>
base	Define the base URI that is valid for all defined local interaction resources. All other URIs in the TD must then be resolved using the algorithm defined in [RFC3986] .	no	.	<u>anyURI</u>
properties	All Property-based interaction patterns of the Thing.	no	.	<u>Property</u>
actions	All Action-based interaction patterns of the Thing.	no	.	<u>Action</u>
events	All Event-based interaction patterns of the Thing.	no	.	<u>Event</u>
links	Provides Web links to arbitrary resources that relate to the specified Thing Description.	no	.	array of <u>Link</u>
security	Set of security configurations, provided as an array, that must all be satisfied for access to resources at or below the current level, if not overridden at a lower level.	no	.	array of <u>SecurityScheme</u>

Source: <https://w3c.github.io/wot-architecture/>

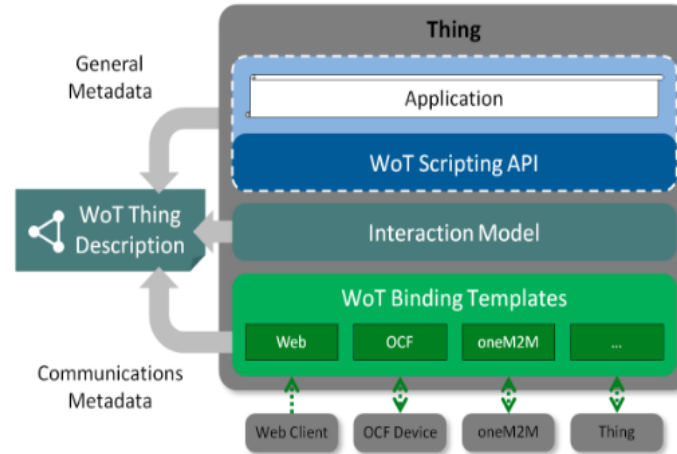
Conceptual Architecture of the WoT Building Blocks

Thing

The WoT Thing Description

The WoT Binding Templates

The WoT Scripting API.



Source: <https://w3c.github.io/wot-architecture/>

The WoT Thing Description (TD)

The Thing Description (TD) is a central building block in the W3C Web of Things (WoT) and can be considered as the entry point of a Thing

The TD consists of semantic metadata for the Thing itself, an interaction model based on WoT's Properties, Actions, and Events paradigm.

Source: <https://w3c.github.io/wot-architecture/>

The WoT Thing Description (TD)

The default Interaction Patterns are Property, Action, and Event.

Properties abstract data points that can be read and often written.

Actions abstract invokable processes that may run for a certain time.

Events abstract interactions where the remote endpoint pushes data asynchronously.

This semantic schema makes data models machine-understandable

Source: <https://w3c.github.io/wot-architecture/>

Simple Thing Description Sample with JSON serialization

```
{
  "id": "urn:dev:wot.com:example:servient:lamp",
  "name": "MyLampThing",
  "security": [{"scheme": "basic"}],
  "properties": {
    "status" : {
      "type": "string",
      "forms": [{"href": "https://mylamp.example.com/status"}]
    }
  },
  "actions": {
    "toggle" : {
      "forms": [{"href": "https://mylamp.example.com/toggle"}]
    }
  },
  "events":{
    "overheating":{
      "type": "string",
      "forms": [{
        "href": "https://mylamp.example.com/oh",
        "subProtocol": "LongPoll"
      }]
    }
  }
}
```

How can we indicate a specific protocol to a TD?



The WoT Binding Templates

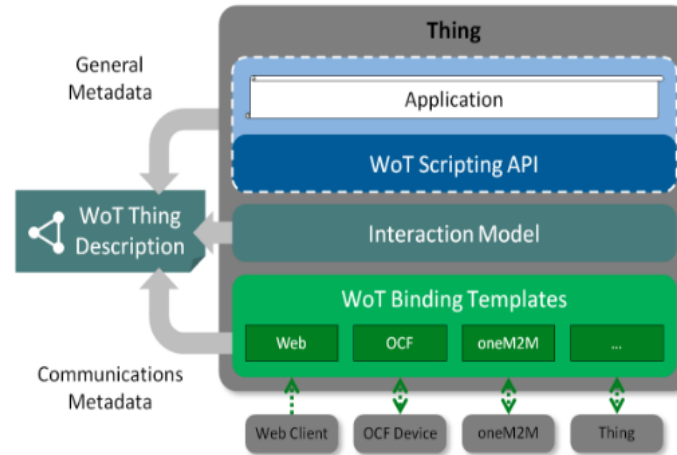
Conceptual Architecture of the WoT Building Blocks

Thing

The WoT Thing Description

The WoT Binding Templates

The WoT Scripting API.



Source: <https://w3c.github.io/wot-architecture/>

© 2018 Cisco and/or its affiliates. All rights reserved.

The WoT Binding Templates

Protocol Binding Templates enable a Thing Description to be adapted to the specific protocol usage across the different standards. This is done through additional descriptive vocabulary that is used in the Thing Description, through:

- Protocol Methods and Options
- Media Types
- Payload Structure
- Data Types and Value Constraints

The WoT Binding Templates

Protocol Methods and Options: set of methods that define the message type, the semantic intention of the message.

For example, REST and PubSub can cover most standard communication patterns. Common methods are GET, PUT, POST, DELETE, PUBLISH, and SUBSCRIBE.

The protocol methods are mapped to the abstract WoT Interaction verbs **readproperty**, **writeproperty**, **observeproperty**, **invokeaction**, **subscribeevent**, **unsubscribeevent**.

The WoT Binding Templates

Media Types: The Media Types define the serialization details and other rules for processing the payloads. The media type is used to select a serializer/deserializer and to select an additional set of rules and constraints for the protocol driver.

For example, the mediaType "**application/ocf+cbor**" indicates that CBOR serialization is used, but also that OCF rules and namespaces apply to the processing of the representations.

The Media Type should be IANA-registered Media Types.

The WoT Binding Templates

Payload Structure:

The **DataSchema** elements describe the payload structure and data items that are passed between client and server during interactions.

Simple Payload Structure

```
{
  "level": 50,
  "time": 10
}
```

DataSchema for Simple JSON Object Payload

```
{
  "type": "object",
  "properties": {
    "level": {
      "@type": ["iot:LevelData"],
      "type": "integer",
      "minimum": 0,
      "maximum": 255
    },
    "time": {
      "@type": ["iot:TransitionTimeData"],
      "type": "integer",
      "minimum": 0,
      "maximum": 65535
    }
  }
}
```

Source: <https://w3c.github.io/wot-binding-templates/>

The WoT Binding Templates

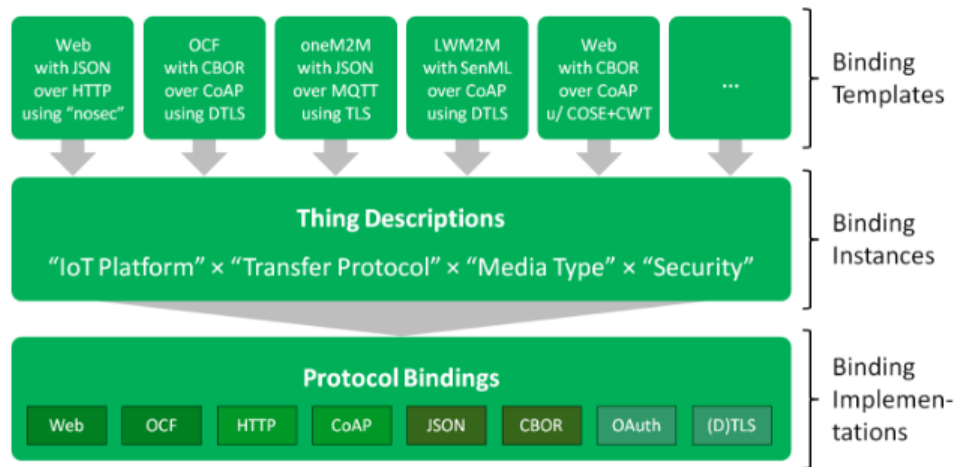
Data Types and Value Constraints: Simple data types and value constraints are currently used in WoT Thing Description. Additional forms of constraints are available to help adapt to the underlying data types. It is up to the client to adapt to the range and type as well as the engineering units provided.

DataSchema for Simple JSON Object Payload

```
{
  "type": "object",
  "properties": {
    "level": {
      "@type": ["iot:LevelData"],
      "type": "integer",
      "minimum": 0,
      "maximum": 255
    },
    "time": {
      "@type": ["iot:TransitionTimeData"],
      "type": "integer",
      "minimum": 0,
      "maximum": 65535
    }
  }
}
```

Source: <https://w3c.github.io/wot-binding-templates/>

The WoT Binding Templates



From Binding Templates to Protocol Bindings

Source: <https://w3c.github.io/wot-architecture/>

Polling Question 2

What is a Thing Description (TD)?

- A. Is the description of CoAP application Protocol
- B. Is a description of a HTML web page.
- C. The TD consists of semantic metadata for a Thing based on its Properties, Actions, and Events.
- D. The TD consists of semantic metadata for a Thing based only in its Properties but not in its Actions or Events.

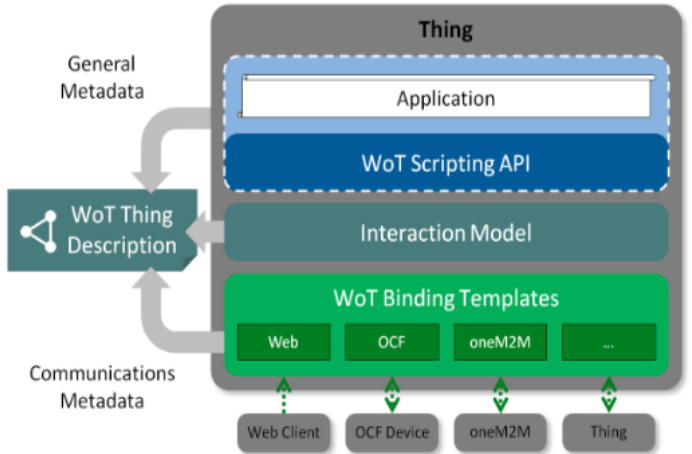
How can we manipulate a Thing Description?



The WoT Scripting API.

Conceptual Architecture of the WoT Building Blocks

- Thing
- The WoT Thing Description
- The WoT Binding Templates
- The WoT Scripting API.



Source: <https://w3c.github.io/wot-architecture/>

The WoT Scripting API

A [programming interface](#) representing the WoT Interface that allows scripts run on a Thing to discover and consume other Thing Descriptions and to expose Things characterized by WoT Interactions specified by a script.

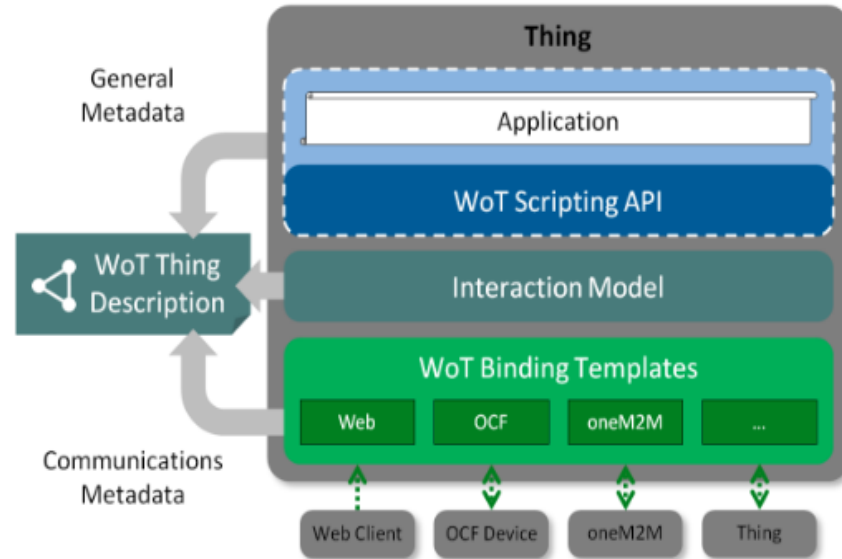
By consuming a TD, a client Thing creates a runtime resource model that allows accessing the Properties, Actions and Events exposed by the server Thing exposed on a remote device. Exposing a Thing requires defining a Thing Description (TD) and instantiating a software stack to serve requests for accessing the exposed Properties, Actions and Events

Source: <https://w3c.github.io/wot-scripting-api/>

The WoT Scripting API Use Cases

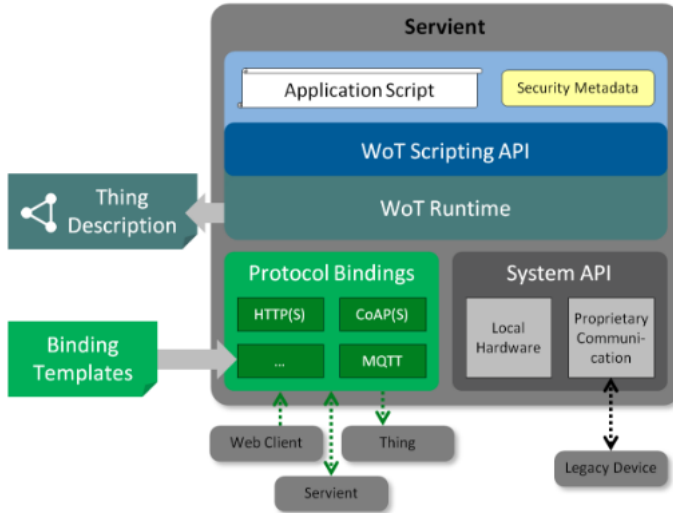
- **Discovery:**
 - Discover all Things in the WoT network by sending a broadcast request.
 - Discover Things running in the local WoT Runtime.
 - Discover nearby Things, for instance connected by NFC or Bluetooth.
- **Expose a Thing:**
 - Exposing the Thing includes generating the protocol bindings in order to access lower level functionality.
 - Create a local **ExposedThing** to be exposed, based on a Thing Description provided in string serialized format, or out of a template or an existing **ConsumedThing** object.
 - Add: a Property definition to the Thing, an Action definition to the Thing., an Event definition to the Thing. Attach semantic information to the Thing.
- **Consuming a Thing:**
 - Consume a TD of a Thing, including parsing the TD and generating the protocol bindings in order to access lower level functionality.
 - 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.

Conceptual Architecture of the WoT Building Blocks



WoT Servient Architecture

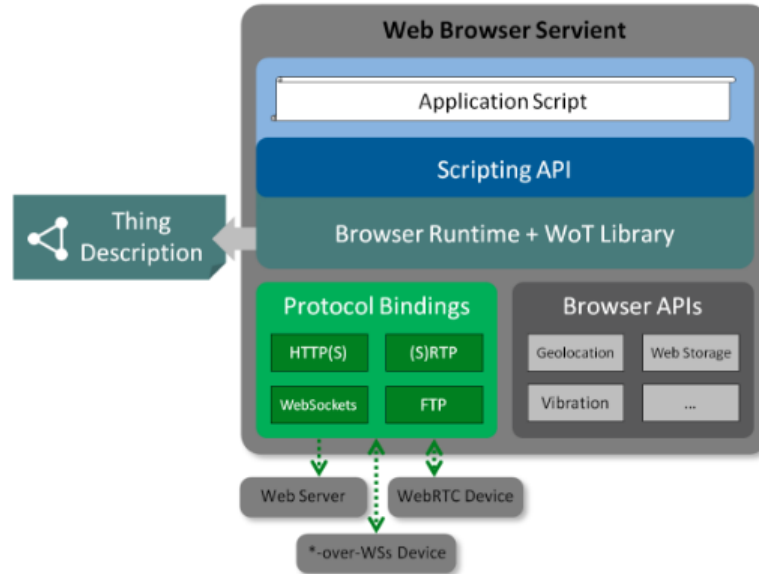
A Servient is a software stack that implements the WoT building blocks presented in the previous section. Servients can host and expose Things and/or consume Things. Thus, Servients can perform in both the server and client roles.



Source: <https://w3c.github.io/wot-architecture/>

Implementation View of a Servient

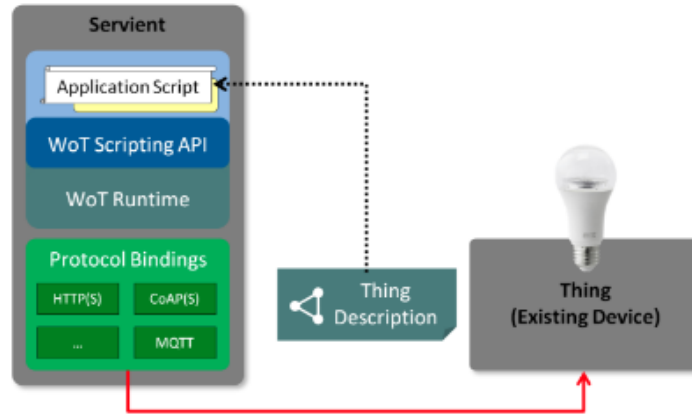
WoT Servient in the Web Browser



Implementing WoT in the Web Browser

Source: <https://w3c.github.io/wot-architecture/>

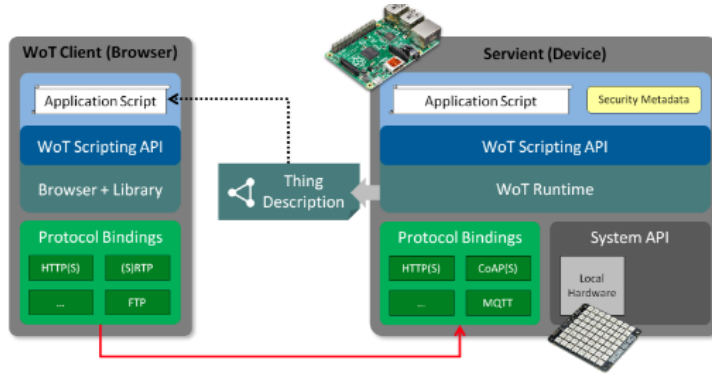
WoT Servient in client role



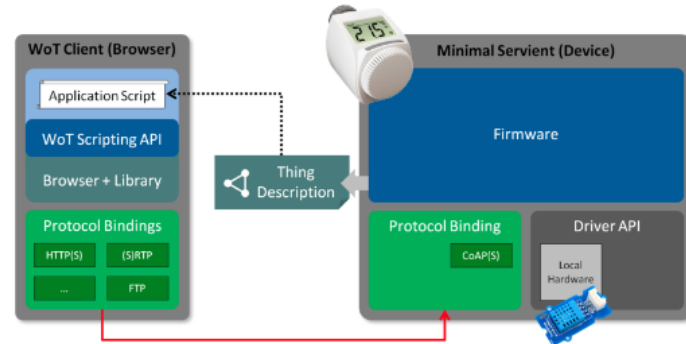
Servient Consuming Thing

Source: <https://w3c.github.io/wot-architecture/>

WoT Servient running on the device itself



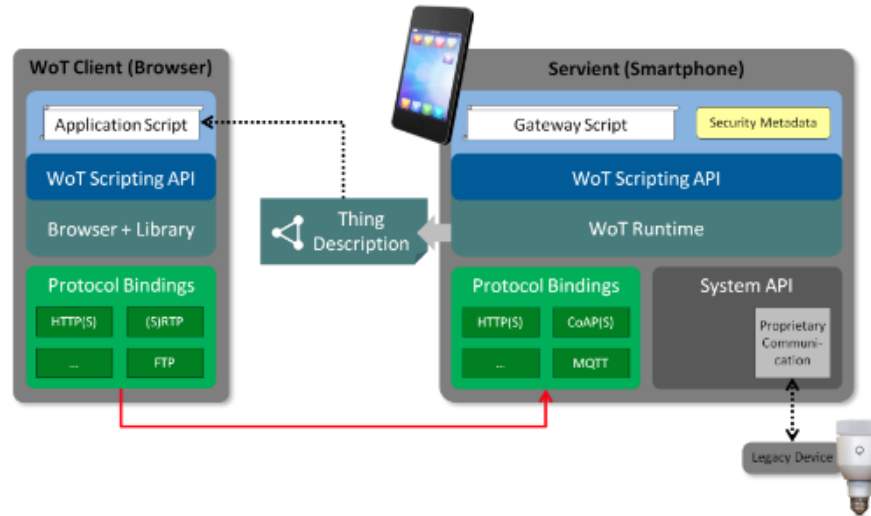
Servient on Device Itself



Resource-constrained Device as Thing

Source: <https://w3c.github.io/wot-architecture/>

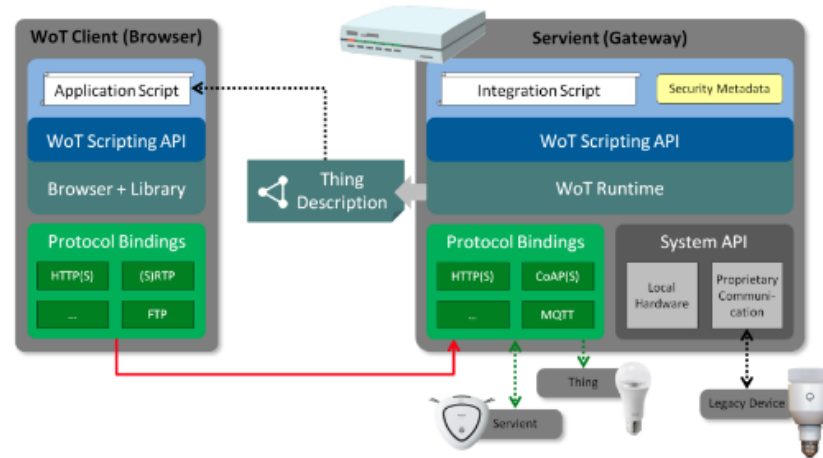
WoT Servient on Smartphone



Servient on Smartphone

Source: <https://w3c.github.io/wot-architecture/>

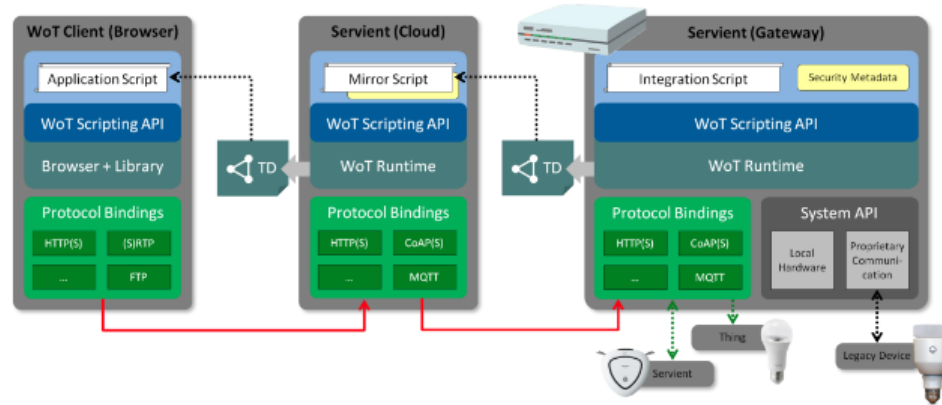
WoT Servient on gateway



Servient on Gateway

Source: <https://w3c.github.io/wot-architecture/>

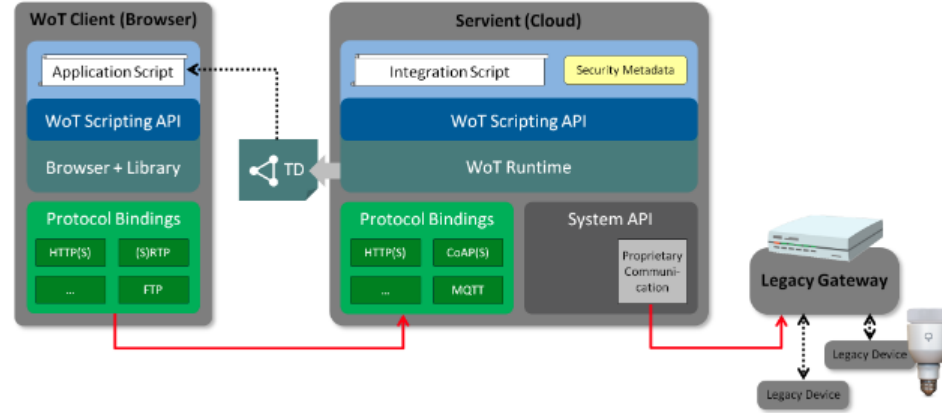
WoT Servient on Cloud and Gateway



Servient on Cloud Server and Gateway

Source: <https://w3c.github.io/wot-architecture/>

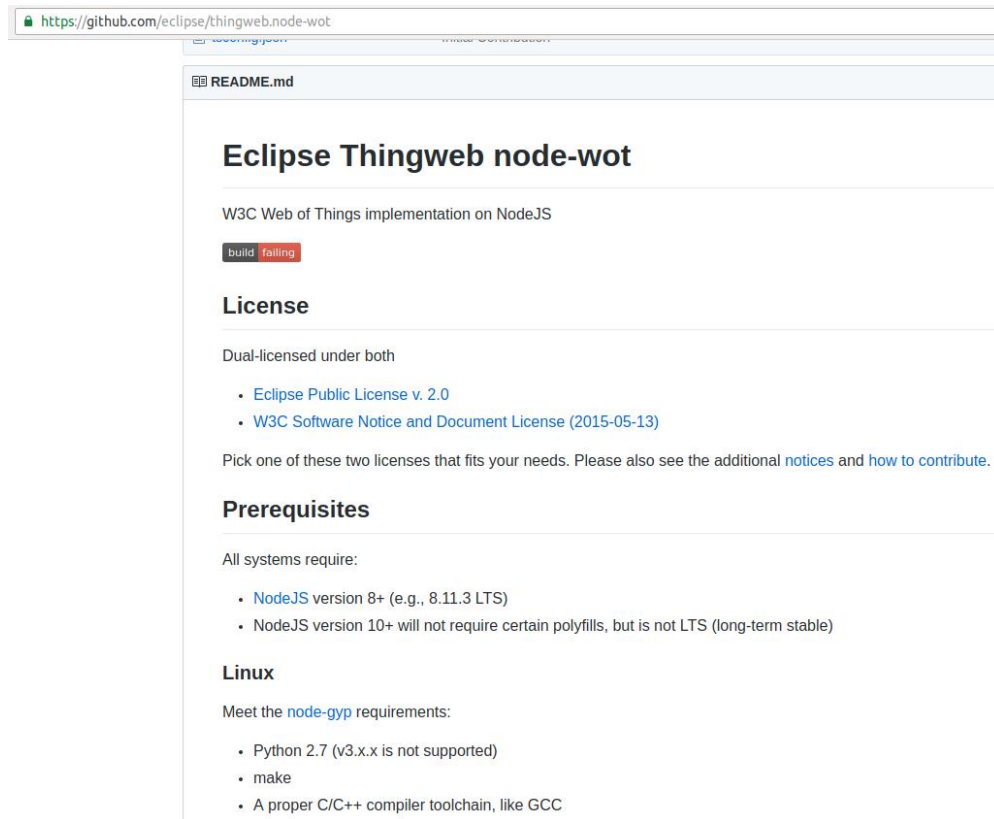
WoT Servient on Cloud Server Only



Implementing WoT on Cloud Server Only

Source: <https://w3c.github.io/wot-architecture/>

Open Source: node.js



The screenshot shows a web browser displaying the GitHub README for the repository `eclipse/thingweb.node-wot`. The page title is "Eclipse Thingweb node-wot". Below the title, it states "W3C Web of Things implementation on NodeJS" and includes a status badge that says "build failing". The "License" section indicates it is dual-licensed under both the Eclipse Public License v. 2.0 and the W3C Software Notice and Document License (2015-05-13). It asks users to pick one of these licenses and refers to additional notices and contribution guidelines. The "Prerequisites" section lists requirements for all systems, including NodeJS version 8+ (with a note that version 10+ does not require certain polyfills but is not LTS) and NodeJS version 10+ for long-term stability. The "Linux" section lists requirements for the `node-gyp` package, including Python 2.7 (noting that v3.x.x is not supported), the `make` command, and a proper C/C++ compiler toolchain like GCC.

<https://github.com/eclipse/thingweb.node-wot>

README.md

Eclipse Thingweb node-wot

W3C Web of Things implementation on NodeJS

build failing

License

Dual-licensed under both

- [Eclipse Public License v. 2.0](#)
- [W3C Software Notice and Document License \(2015-05-13\)](#)

Pick one of these two licenses that fits your needs. Please also see the additional [notices](#) and [how to contribute](#).

Prerequisites

All systems require:

- [NodeJS](#) version 8+ (e.g., 8.11.3 LTS)
- NodeJS version 10+ will not require certain polyfills, but is not LTS (long-term stable)

Linux

Meet the [node-gyp](#) requirements:

- Python 2.7 (v3.x.x is not supported)
- `make`
- A proper C/C++ compiler toolchain, like GCC

Open Source: python

<https://github.com/mozilla-iot/webthing-python>

README.rst

webthing

build passing | pypi v0.8.1 | license MPL-2.0

Implementation of an HTTP [Web Thing](#). This library is compatible with Python 3.5+.

Installation

`webthing` can be installed via `pip`, as such:

```
$ pip install webthing
```

Example

In this example we will set up a dimmable light and a humidity sensor (both using fake data, of course). Both working examples can be found in [here](#).

Dimmable Light

Imagine you have a dimmable light that you want to expose via the web of things API. The light can be turned on/off and the brightness can be set from 0% to 100%. Besides the name, description, and type, a `Light` is required to expose two properties:

- `on`: the state of the light, whether it is turned on or off
 - Setting this property via a `PUT {"on": true/false}` call to the REST API toggles the light.
- `brightness`: the brightness level of the light from 0-100%
 - Setting this property via a `PUT` call to the REST API sets the brightness level of this light.

First we create a new Thing:

Polling Question 3

What are the main building blocks in the WoT?

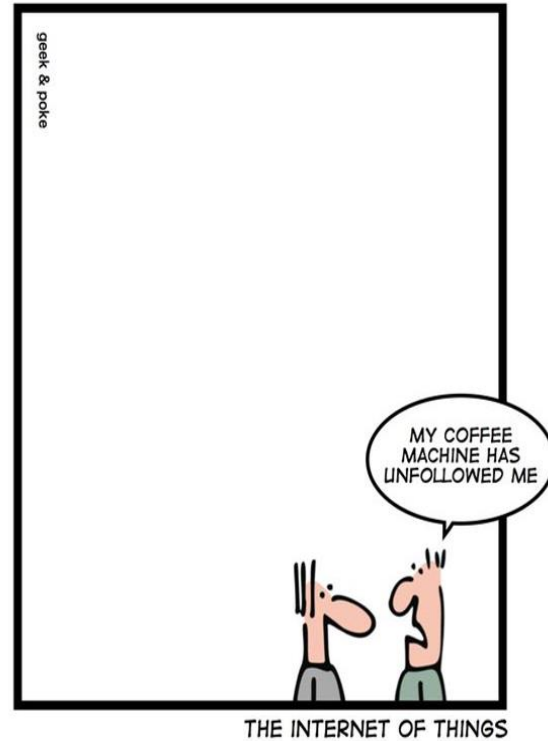
- A. The Thing, the WoT Thing Description and the WoT Scripting API.
- B. The Thing and the WoT Thing Description.
- C. The Thing, the WoT Thing Description and the WoT Binding Templates.
- D. The Thing, the WoT Thing Description, the WoT Binding Templates and the WoT Scripting API.

Takeaways

- The Web of Things aim to enable interoperability across IoT Platforms and application domains.
- The main building blocks of the WoT are:
 - Thing
 - The Thing Description
 - The Binding Template
 - The Scripting API
- A Servient can perform in both the server and client roles

Thank you! :-)

maria.robles@aalto.fi



<http://geek-and-poke.com/geekandpoke/2012/3/14/the-internet-of-things.html>

Thanks For Joining today!

Submit Your
Questions Now!



Use the Q&A panel to submit your
questions, our expert will respond

Ask the Expert following the Webcast

Now through Friday October 19th 2018

With
Ines Robles

<http://bit.ly/Ask-Expert-WoT>



Ines Robles
IoT Consultant & IT Reseracher



Collaborate within our Social Media



Twitter

- @Cisco_Support
- <http://bit.ly/csc-twitter>

Facebook

- Cisco Community
- <http://bit.ly/csc-facebook>

Learn About Upcoming Events

We invite you to review our Social Media Channels

YouTube

- Cisco Community
- <http://bit.ly/csc-youtube>



App

- Cisco Technical Support



LinkedIn

- Cisco Community
- <http://bit.ly/csc-linked-in>



Cisco has support communities in other languages!

If you speak Spanish, Portuguese, Japanese, Russian or Chinese we invite you to participate & collaborate



[Comunidad de Cisco](#)
Spanish

[Comunidade da Cisco](#)
Portuguese

[思科服务支持社区](#)
Chinese

[Сообщество Cisco](#)
Russian

[シスココミュニティ](#)
Japanese



More IT Training Videos and Technical Seminars on the Cisco Learning Network

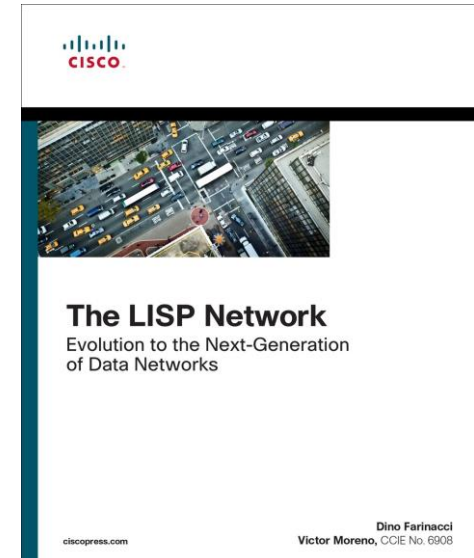
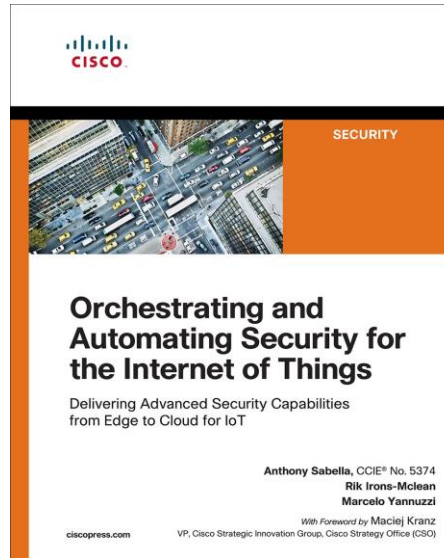
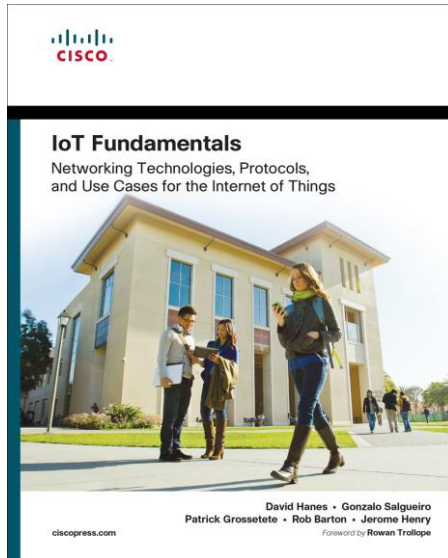
View Upcoming Sessions Schedule

<https://cisco.com/go/techseminars>

Thank you for participating, you earned a discount!

Redeem your 35% discount offer by entering code: CSC when checking out.

<http://bit.ly/Community-CiscoPress2018>



Thank you for Your
Time!

Please take a moment to complete the
survey



Thanks For Joining today!

