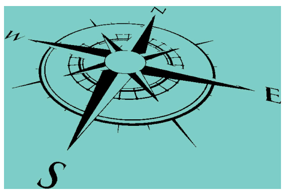
Web of Things (WoT) Architecture - Use Cases & Requirements

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, both given in this document. A set of modular building blocks are also 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.

Contents



World Wide Web Consortium (W3C)

Stakeholder(s):

WoT Users :
Use Cases — This section is non-normative. This section presents the application domains and use cases targeted by the W3C WoT and which are used to derive the abstract architecture discussed in § 7. WoT Building Blocks. The Web of Things architecture does not put any limitations on use cases and application domains. Various application domains have been considered to collect common patterns that have to be satisfied by the abstract architecture. The following sections are not exhaustive. Rather they serve as illustrations, where connected things can provide additional benefit or enable new scenarios.

Consumers :
In the consumer space there are multiple assets that benefit from being connected. Lights and air conditioners can be turned off based on room occupancy. Window blinds can be closed automatically based on weather conditions and presence. Energy and other resource consumption can be optimized based on usage patterns and predictions.

Industries :
The industrial use cases in this section are applicable to different industry verticals. Due to the nature of overlaps in the application scenarios, different verticals have similar use cases.

Vehicle Fleet Managers :
Transportation & Logistics — Monitoring of vehicles, fuel costs, maintenance needs and assignments helps to optimize the full utilization of the vehicle fleet. Shipments can be tracked to be en-route to ensure consistent quality and condition of the transported goods. This is especially useful to assert the integrity of the cold-chain from warehouses to refrigerated trucks to delivery. Centralized monitoring and management of stock in warehouses and yards can prevent out of stock and excessive stock situations.

Utilities :
Automated reading of residential and C&I (Commercial and Industrial) meters, and billing offers continuous insights into resource consumption and potential bottlenecks. Monitoring the condition and output of distributed renewable energy generation equipment enables optimization of distributed energy resources. Monitoring and remote-controlling of distribution equipment helps to automate the distribution process. Continuous monitoring of generation and distribution infrastructure is improving safety of utilities crew in the field.

Oil & Gas Providers :
Offshore platform monitoring, leakage detection and prediction of pipelines as well as monitoring and controlling the levels in tanks and reservoirs helps to improve the industrial safety for the workforce as well as for the environment. Automated calculation of a distributed stock through various storage tanks and delivery pipes/trucks allows for improved planning and resource optimization.

Insurance Providers :
Proactive Asset Monitoring of high value assets such as connected structures, fleet vehicles, etc. mitigates the risk of severe damage and high costs due to predictions and early detection of incidents. Usage based insurance can be offered with usage tracking and customized insurance policies. Predictive weather monitoring and re-routing fleet vehicles to covered garages can limit loss due to hail damage, tree damage.

Engineers :
Engineering and Construction — Monitoring for industrial safety reduces the risks of security hazards. Monitoring of assets at construction site can prevent damage and loss.

Builders

Farmers :
Agriculture — Soil condition monitoring and creating optimal plans for watering, fertilizing as well as monitoring the produce conditions optimize the quality and output of agricultural produce.

Healthcare Providers :
Data collection and analytics of clinical trial data helps to gain insights into new areas. Remote patient monitoring mitigates the risk of undetected critical situations for elderly people and patients after hospitalization.

Environmental Stewards :
Environment Monitoring — Environment monitoring typically relies on a lot of distributed sensors that send their measurement data to common gateways, edge devices and cloud services. Monitoring of air pollution, water pollution and other environmental risk factors such as fine dust, ozone, volatile organic compound, radioactivity, temperature, humidity to detect critical environment conditions can prevent unrecoverable health or environment damages.

Smart Cities :
Monitoring of Bridges, Dams, Levees, Canals for material condition, deterioration, vibrations discovers maintenance repair work and prevents significant damage. Monitoring of highways and providing appropriate signage ensures optimized traffic flow. Smart Parking is optimizing and tracking the usage and availability of parking spaces and automates billing/reservations. Smart control of street lights based on presence detec-

— continued next page

Stakeholders (continued)

tion, weather predictions, etc. reduces cost. Garbage containers can be monitored to optimize the waste management and the trash collection route.

Building Managers :
Smart Buildings — Monitoring the energy usage throughout the building helps to optimize resource consumption and reduce waste. Monitoring the equipment in the buildings such as HVAC, Elevators, etc. and fixing problems early improves the satisfaction of occupants.

Vehicle Drivers :
Connected Car — Monitoring of operation status, prediction of service needs optimizes maintenance needs and costs. Driver safety is enhanced with notifications of an early warning system for critical road and traffic conditions.

Web of Things (WoT) Architecture Editors

Matthias Kovatsch :
Huawei

Ryuichi Matsukura :
Fujitsu Ltd.

Michael Lagally :
Oracle Corp.

Toru Kawaguchi :
Panasonic Corp.

Kunihiko Toumura :
Hitachi, Ltd.

Kazuo Kajimoto :
Former Editor, when at Panasonic

Web of Things (WoT) Architecture Contributors :
Special thanks to Michael McCool, Takuki Kamiya, Kazuyuki Ashimura, Sebastian Käbisch, Zoltan Kis, Elena Reshetova, Klaus Hartke, Ari Keränen, Kazuaki Nimura, and Philippe Le Hegaret for their contributions to this document.

Michael McCool

Takuki Kamiya

Kazuyuki Ashimura

Sebastian Käbisch

Zoltan Kis

Elena Reshetova

Klaus Hartke

Ari Keränen

Kazuaki Nimura

Philippe Le Hegaret

W3C Staff :
Many thanks to the W3C staff and all other active Participants of the W3C Web of Things Interest Group (WoT IG) and Working Group (WoT WG) for their support, technical input and suggestions that led to improvements to this document.

W3C Web of Things Interest Group (WoT IG)

International Workshop on the Web of Things :
The WoT WG also would like to appreciate the pioneering efforts regarding the concept of "Web of Things" that started as an academic initiative in the form of publications such as [WOT-PIONEERS-1] [WOT-PIONEERS-2] [WOT-PIONEERS-3] [WOT-PIONEERS-4] and, starting in 2010, a yearly International Workshop on the Web of Things.

Joerg Heuer :
Finally, special thanks to Joerg Heuer for leading the WoT IG for 2 years from its inception and guiding the group to come up with the concept of WoT building blocks including the Thing Description.

Vision

An interoperable Web of Things (WoT)

Mission

To enable interoperability across IoT platforms and application domains.

Values

Principles: Common Principles -- * WoT architecture should enable mutual interworking of different eco-systems using web technology. * WoT architecture should be based on the web architecture using RESTful APIs. * WoT architecture should allow to use multiple payload formats which are commonly used in the web. * WoT architecture must enable different device architectures and must not force a client or server implementation of system components.

Flexibility: There are a wide variety of physical device configurations for WoT implementations. The WoT abstract architecture should be able to be mapped to and cover all of the variations.

Compatibility: There are already many existing IoT solutions and ongoing IoT standardization activities in many business fields. The WoT should provide a bridge between these existing and developing IoT solutions and Web technology based on WoT concepts. The WoT should be upwards compatible with existing IoT solutions and current standards.

Scalability: WoT must be able to scale for IoT solutions that incorporate thousands to millions of devices. These devices may offer the same capabilities even though they are created by different manufacturers.

Interoperability: WoT must provide interoperability across device and cloud manufacturers. It must be possible to take a WoT enabled device and connect it with a cloud service from different manufacturers out of the box.

1. Functionalities

Thing Functionalities — WoT architecture should allow things to have functionalities such as:

1.1. Status

Read thing's status information

1.2. Updating

Update thing's status information which might cause actuation

1.3. Status Notification

Enable subscription, receipt and unsubscription to notifications of changes of the thing's status information

1.4. Invocation

Invoke functions with input and output parameters which would cause certain actuation or calculation

1.5. Event Notification

Enable subscription, receipt and unsubscription to event notifications that are more general than just reports of state transitions

2. Search & Discovery

2.1. Attributes & Functionalities

Enable clients to know thing's attributes, functionalities and their access points

WoT architecture should allow clients to know thing's attributes, functionalities and their access points, prior to access to the thing itself.

2.2. Queries

Enable clients to search things by attributes and functionalities

WoT architecture should allow clients to search things by its attributes and functionalities.

2.3. Semantics

Support semantic search of things based on a unified vocabulary

WoT architecture should allow semantic search of things providing required functionalities based on a unified vocabulary, regardless of naming of the functionalities.

3. Description Mechanism

Support a common description mechanism

WoT architecture should support a common description mechanism which enables describing things and their functions.

3.1. Readability

Make descriptions both human- and machine-readable

Such descriptions should be not only human-readable, but also machine-readable.

3.2. Structure & Content

Support semantic annotation of the structure and content of descriptions

Such descriptions should allow semantic annotation of its structure and described contents.

3.3. Exchange

Enable exchange of descriptions using multiple common formats

Such description should be able to be exchanged using multiple formats which are commonly used in the web.

4. Descriptions

Describe the attributes of IoT things

Description of Attributes

4.1. Attributes

Support description of things' attributes

WoT architecture should allow describing thing's attributes such as

- name
- explanation
- version of spec, format and description itself
- links to other related things and metadata information

4.2. Internationalization

Support internationalization

Such descriptions should support internationalization.

5. Functionalities

Enable description of things' functionalities

Description of Functionalities — WoT architecture should allow describing thing's functionalities which is shown in § 5.1.2 Thing Functionalities

6. Network

6.1. Protocols

Support multiple web protocols

WoT architecture should support multiple web protocols which are commonly used. Such protocols include:

- protocols commonly used in the internet and
- protocols commonly used in the local area network

6.2. Access

Enable usage of multiple web protocols to access to the same functionality

WoT architecture should allow using multiple web protocols to access to the same functionality.

6.3. Combinations

Support usage of a combination of multiple protocols to the functionalities of the same thing

WoT architecture should allow using a combination of multiple protocols to the functionalities of the same thing (e.g., HTTP and WebSocket).

7. Deployment

7.1. Capabilities

Support a wide variety of thing capabilities

WoT architecture should support a wide variety of thing capabilities such as edge devices with resource restrictions and virtual things on the cloud, based on the same model.

7.2. Hierarchies

Support multiple levels of thing hierarchy

WoT architecture should support multiple levels of thing hierarchy with intermediate entities such as gateways and proxies.

7.3. Access

Support access to things in the local network from the outside of the local network

WoT architecture should support accessing things in the local network from the outside of the local network (the internet or another local network), considering network address translation.

8. Applications

Support description of applications for a wide variety of things based on the same model.

WoT architecture should allow describing applications for a wide variety of things such as edge device, gateway, cloud and UI/UX device, using web standard technology based on the same model.

9. Legacy Adoption

9.1. Protocol Mapping

Enable mapping of legacy IP and non-IP protocols to web protocols

WoT architecture should allow mapping of legacy IP and non-IP protocols to web protocols, supporting various topologies, where such legacy protocols are terminated and translated.

9.2. Transparency

Enable transparent use of existing IP protocols without translation

WoT architecture should allow transparent use of existing IP protocols without translation, which follow RESTful architecture.

9.3. Roles

Refrain from enforcing client or server roles on devices and services

WoT architecture must not enforce client or server roles on devices and services. An IoT device can be either a client or a server, or both, depending on the system architecture; the same is true of edge and cloud services.

Administrative Information

Start Date: 2020-04-09
End Date:

Publication Date: 2020-04-09
Source: https://www.w3.org/TR/wot-architecture/#sec-use-cases

Submitter:
Given Name: Owen
Surname: Ambur
Email: Owen.Ambur@verizon.net
Phone: