

W3C WoT architecture
Jan. 16, 2019
Ryuichi Matsukura, Fujitsu

Contents (new proposal)

- 1. Introduction
- 2. Terminology
- 3. Use cases
- 4. Functional requirements
- 5. WoT servient architecture
- 6. Deployment
- 7. Security

Matthias proposal in Lyon

1. Introduction
2. Terminology
3. Use Cases

4. Functional Requirements
5. Abstract Architecture
(Terminology and concepts)
Message flows

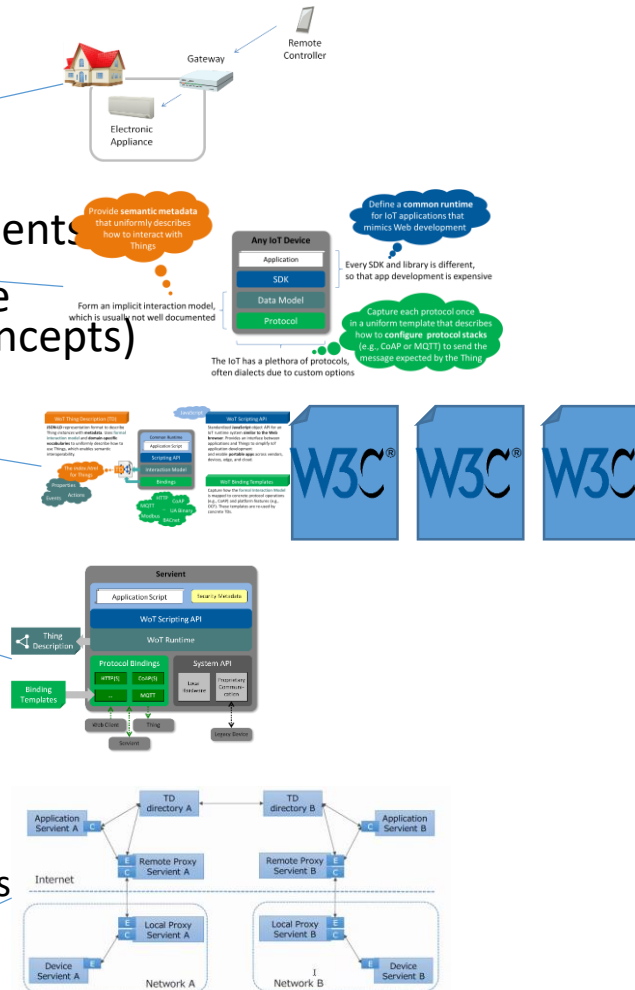
6. WoT Building Blocks
(Specifications)

7. WoT Servients
(Implementation)
Lifecycle

8. WoT Deployments

1. Device Servients
2. Proxy Servients
3. Application Servients
4. Directories
5. Integrations

9. Security and Privacy Considerations



1. Introduction

- 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.
- This document describes the abstract architecture for the W3C Web of Things. It is derived from a set of use cases and can be mapped onto a variety of concrete deployment scenarios, several examples of which are given. This document is focused on the standardization scope of W3C WoT.

2. Terminology

Note: terms described here are in the current draft and will be fixed after the all documents finalized.

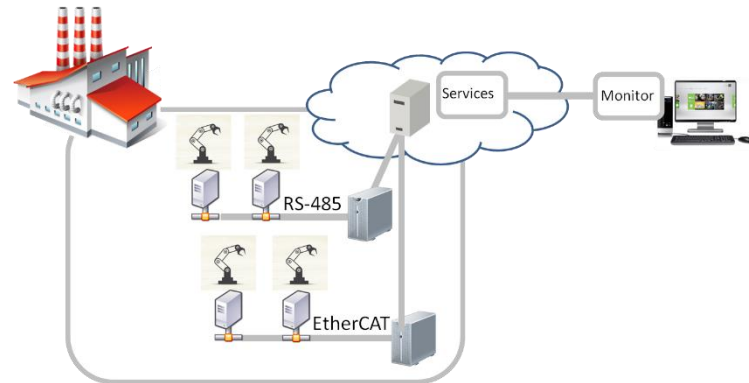
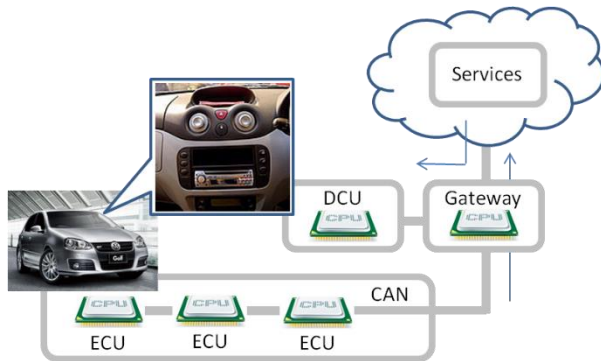
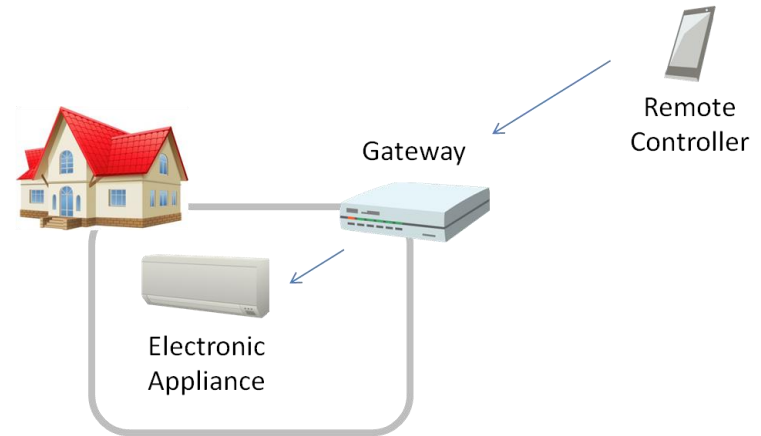
- Action
- Application
- Binding templates
- Client API
- To consume a thing
- Discovery API
- Domain-specific vocabulary
- Event
- Execution environment
- To expose a thing
- Interaction
- Interaction model
- Interaction pattern
- IoT platform
- Local discovery
- Manual discovery
- Nearby discovery
- Network discovery
- Property
- Protocol binding
- Remote discovery
- Scripting API
- Server API
- Servient
- TD
- TD vocabulary
- Thing
- Thing Description (TD)
- Thing directory
- WoT Client
- WoT Interface
- WoT Object
- WoT Runtime
- WoT Server
- CoAP
- CWT
- JSON-LD
- JWT
- RDF

3. Use cases

- This section presents the use cases targeted by the W3C WoT and which are used to derive the abstract architecture discussed in the following sections “Functional Requirements and Abstrat architecture”.
- 2 types use case will be described here
- Collection of user cases in a various fields.
- Collection of combinations and diagrams of IoT components.

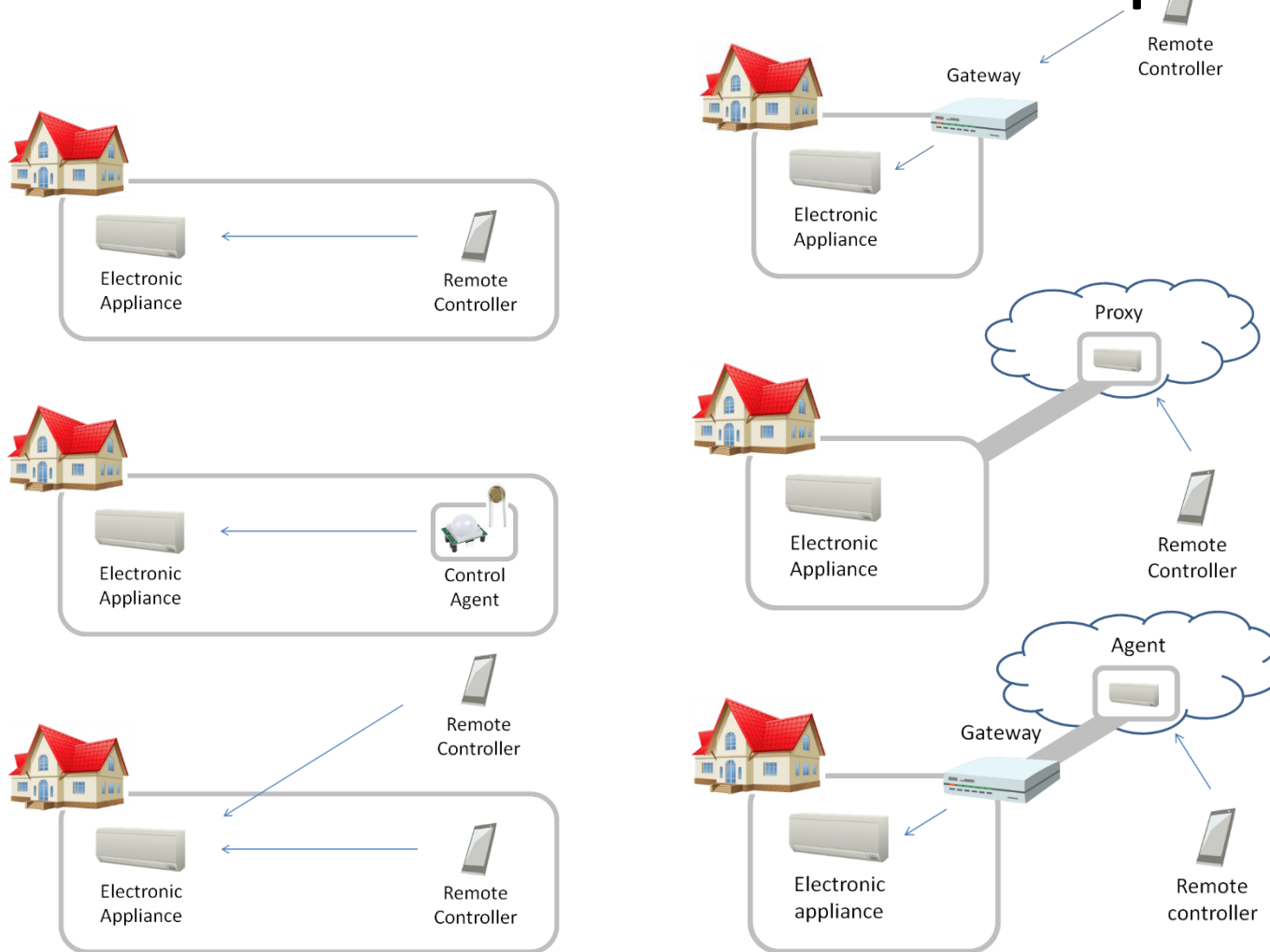
3.1 Use cases in a various fields

- Smart home
- Smart factory
- Connected car
- ...



A variety of devices that support diverse communication protocols have been connected in many fields. Common way to manage and handle devices are required for the application developers.

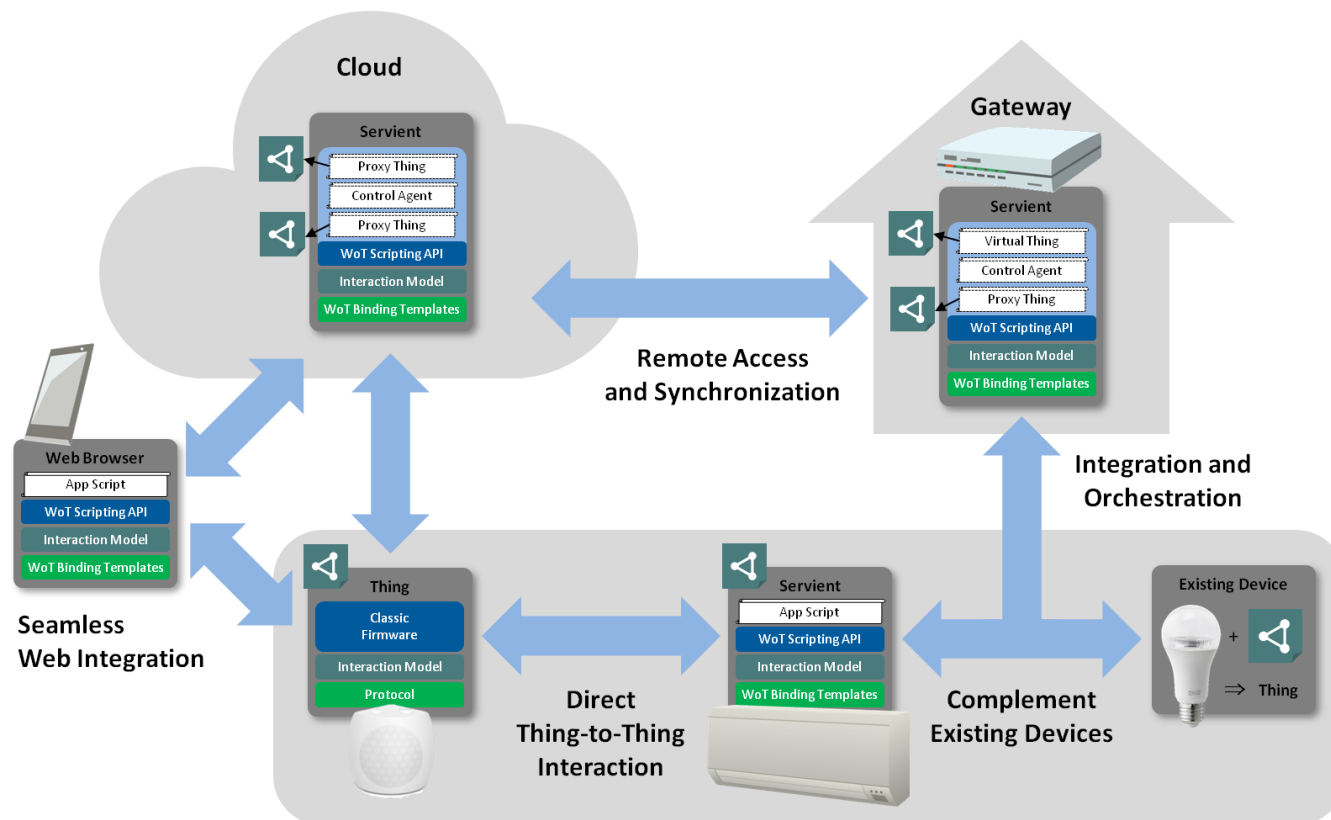
3.2 Combinations of IoT components



Various way to connect applications and devices are required for the real situation. Direct connections, gateways supporting their connection, connecting via cloud, ...

3.3 Summary of use cases

- As described above, various ways to connect with applications, devices, cloud and gateway are aggregated into the same architecture regardless of the fields.



4. Functional requirements

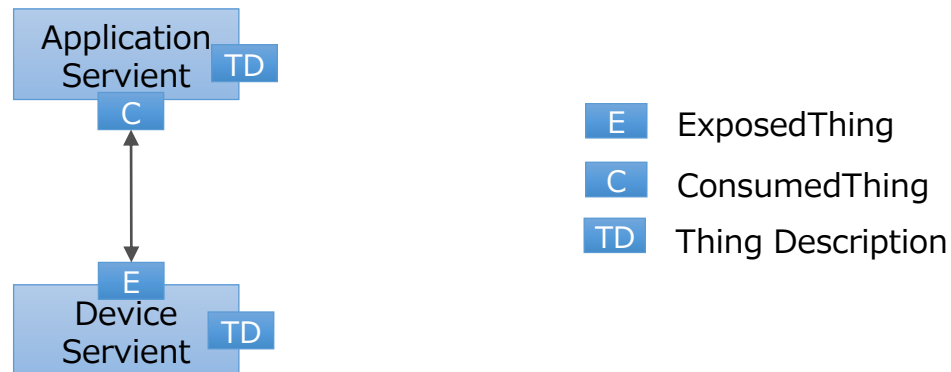
- The applications can access the devices that have different interfaces specified in different fields with the common interface. Devices need to expose their functions and attributes based on the same information model. (Thing description)
- It is necessary to convert domain specific standards into common interfaces. (Protocol binding)
- To set up the complicated architecture, it is necessary to define basic elements of applications, devices, and proxies as a common architecture (Servient architecture and deployment scenarios)
- Applications and devices can be connected beyond multiple networks. (Proxies architecture)
- The applications need a method to find a device. Therefore, it is possible to register the device information and search it via the network. (Discovery)

5. WoT servients architecture

- This section presents the initial WoT building blocks that build up the abstract architecture for the Web of Things. This architecture is derived from the use cases in Section 3. Use Cases and the requirements in Section 4. Functional Requirements.

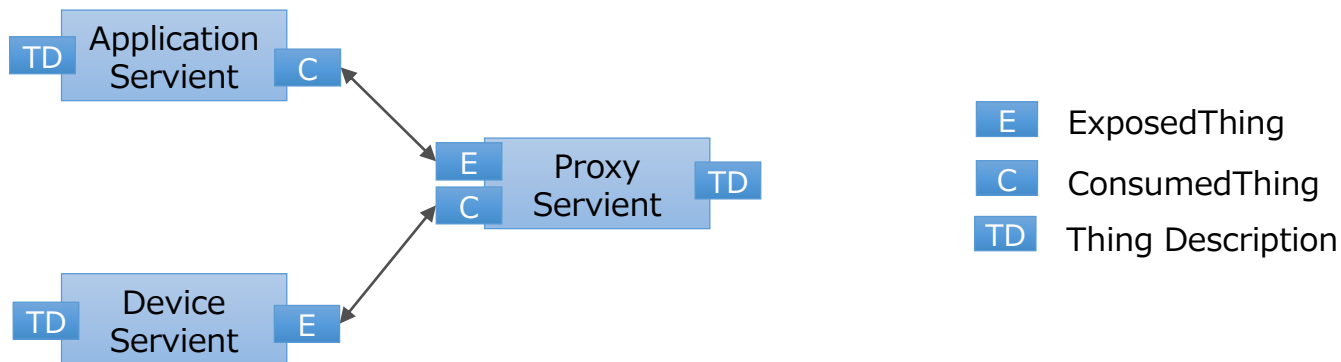
5.1 High-level architecture(1/2)

- Figure summarizes the high-level architectures in case of 2 basic unit such as an application and a device.
 - Thing Description(TD): describes the devices and their attributes.
 - ExposedThing: exposes the TD to the applications to get the status and to handle the function of the devices.
 - ConsumedThing: consumes the TD and generates the application interfaces to access and control the device corresponding to TD.
 - The device servient exposes TD that has been already described through the ExposedThing and then the application servient receives and consumes this TD to generate the ConsumedThing.

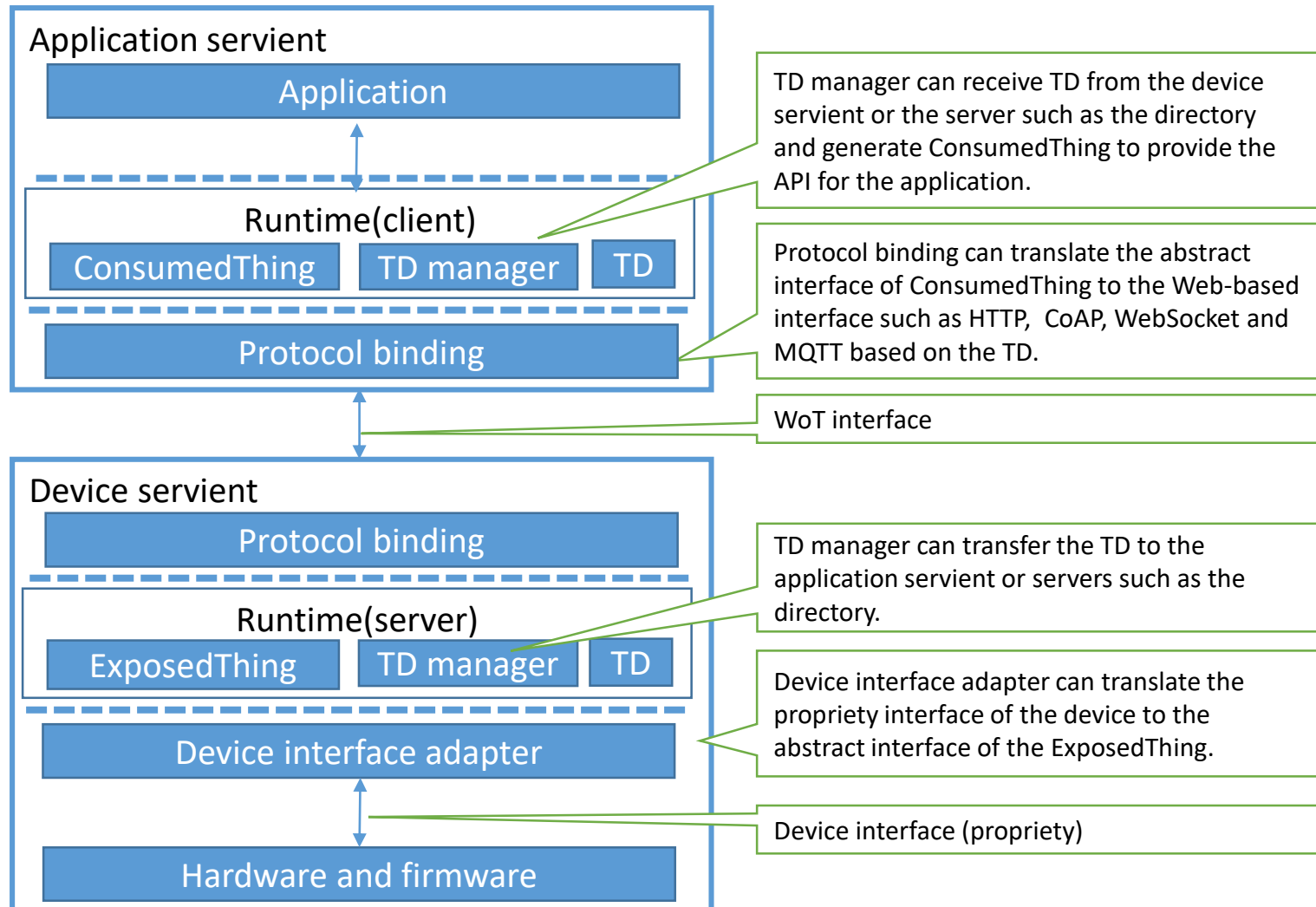


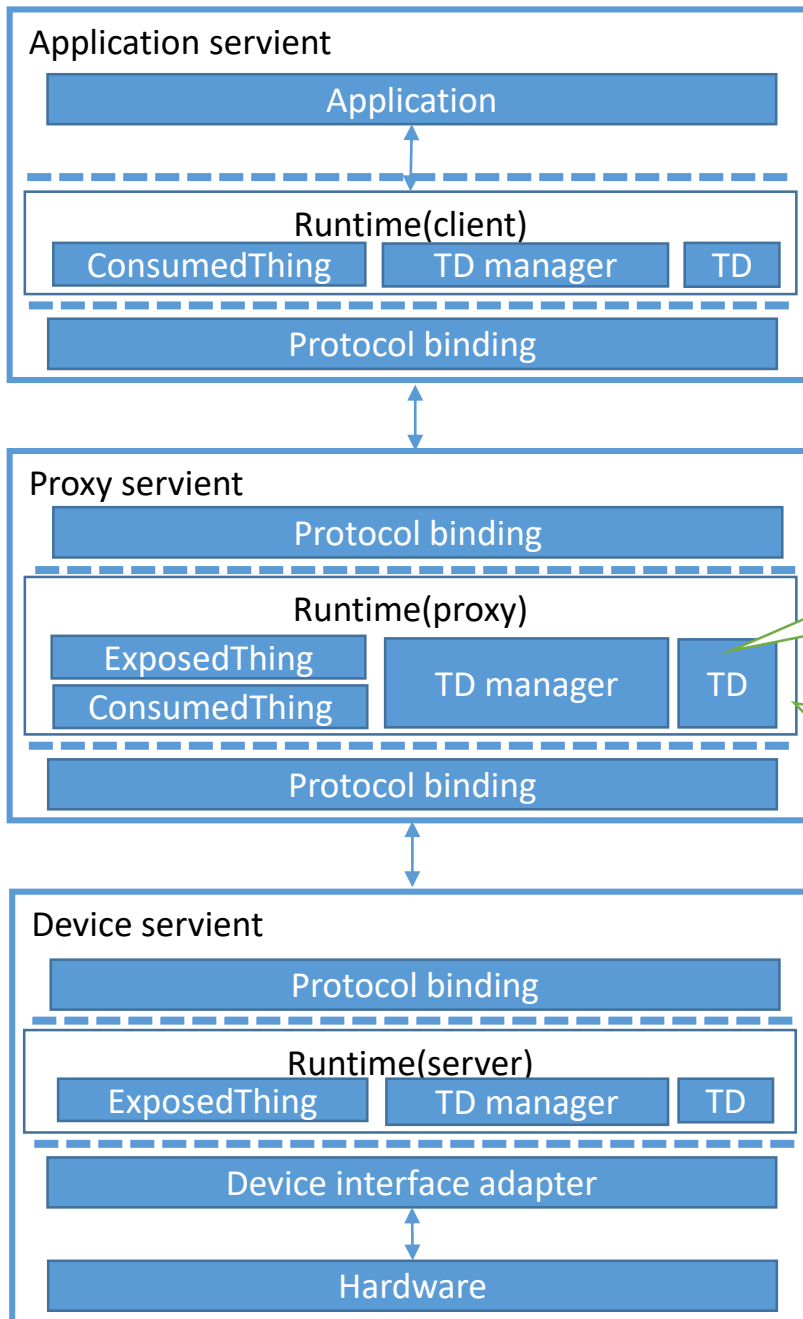
5.1 High-level architecture(2/2)

- Figure shows another high-level architectures in case of 3 basic units including a proxy in addition to the previous diagram.
 - Proxy servient: has both capabilities of the ExposedThing and the ConsumedThings to relay messages between the application and the device. The proxy servient can delegate the device servient to protect the application service to access directly the device servient.



5.2 simple configuration of servient



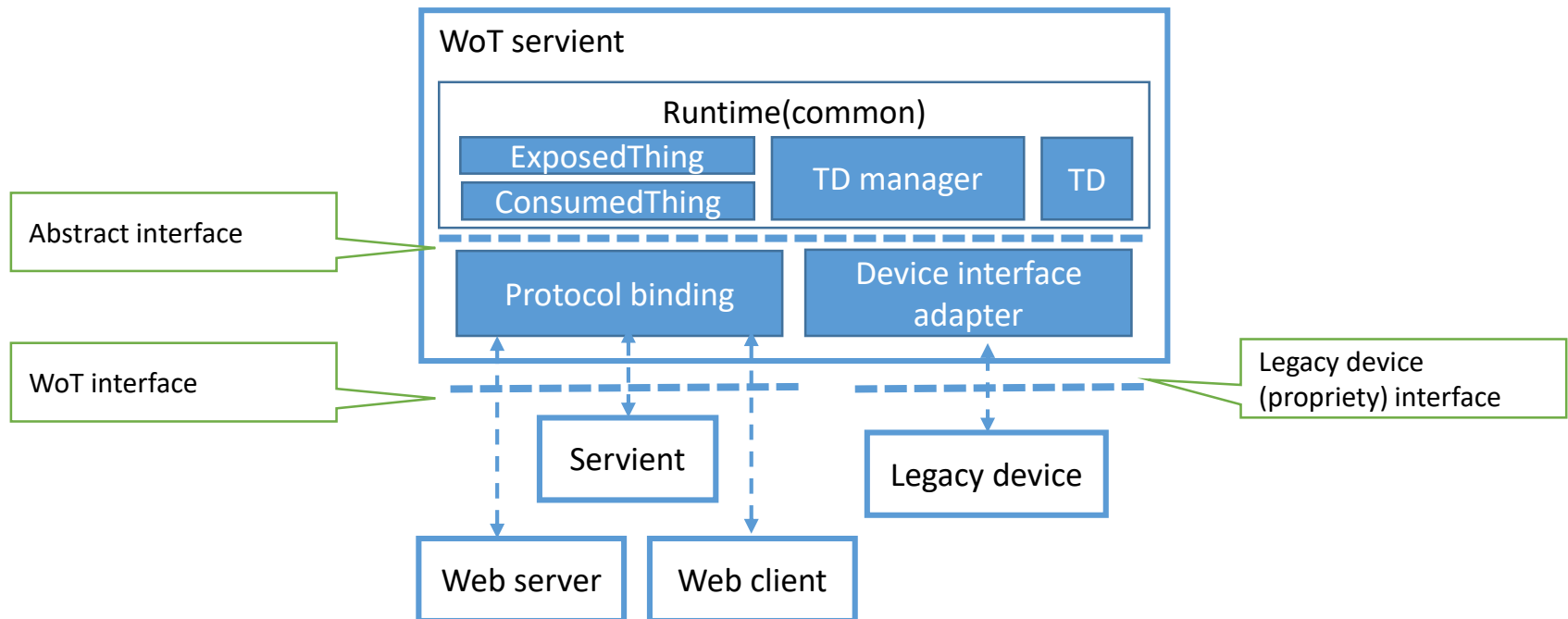


TDs can be collected from one or more device servients and exposed to one or more application servients.

Runtime of the proxy servient has both of the ExposedThing and ConsumedThing. 2 protocol binding can translate different or same Web-based protocol to the abstract interface. If the application and the device support different protocols, this proxy can connect each other.

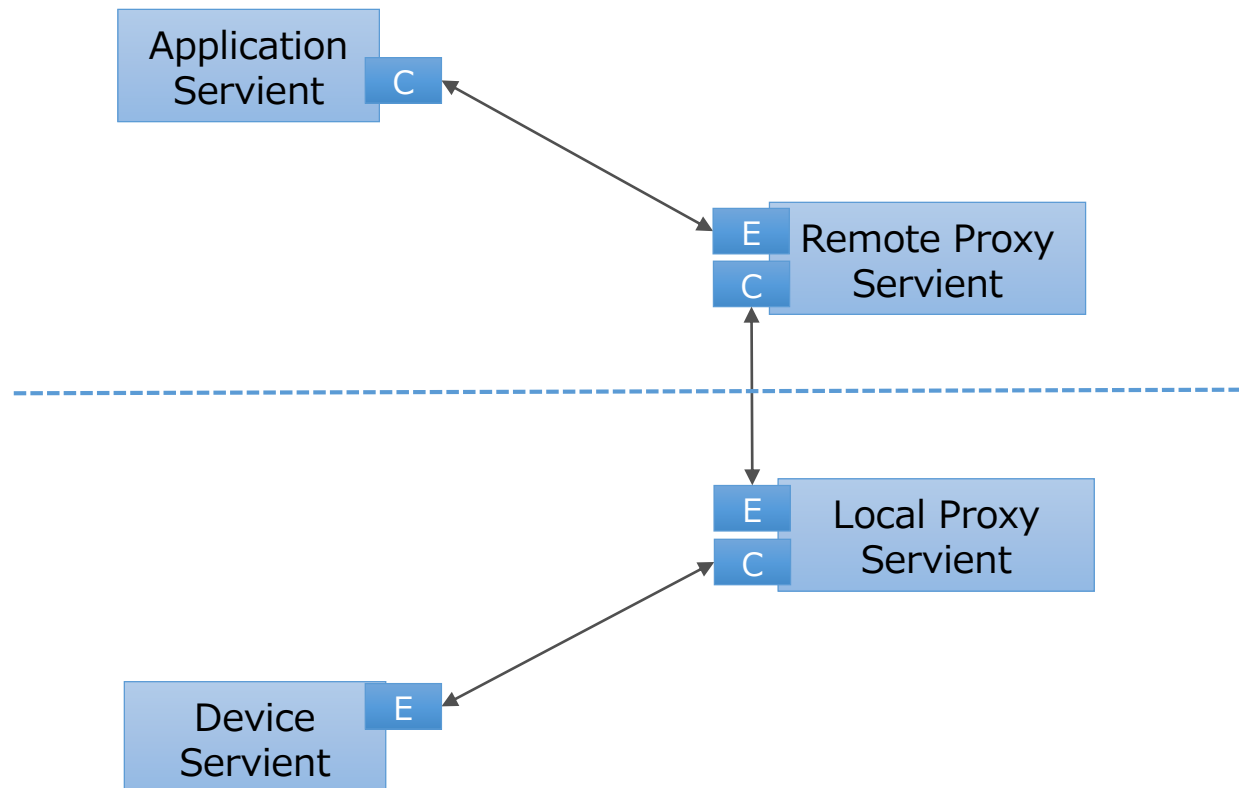
WoT Servient architecture

- Figure below shows WoT servient architecture.
 - Abstract and WoT interface can support methods to handle interaction model such as Read, Write, Invoke, Subscribe, and Event.



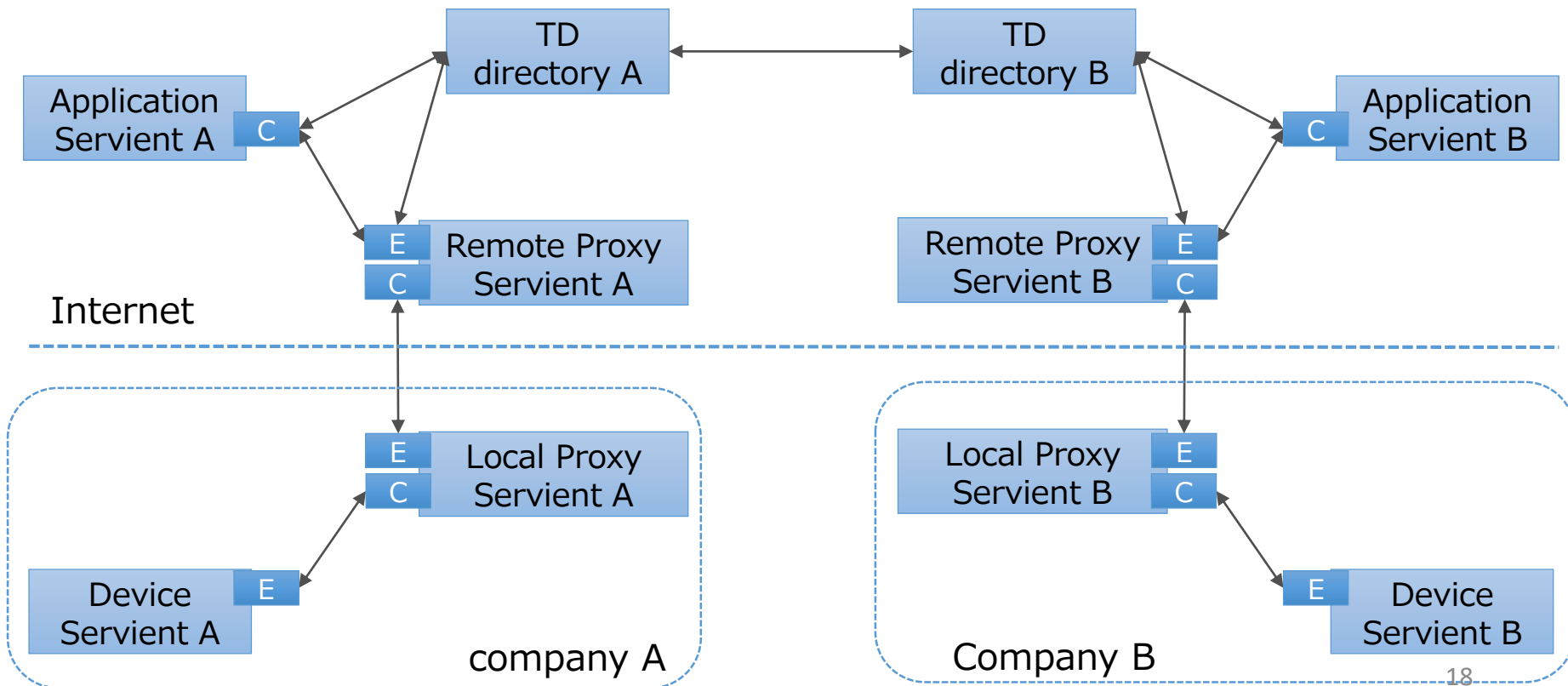
6. Deployment scenario

- To apply WoT architecture to the large scale IoT systems, some extension scenarios are shown in this section.
 - The application on the internet can access the device on the local network.



Deployment scenarios(2)

- 2 different WoT systems can be connected.



7. Security