

W3C Web of Things Interest/Working Group

FUJITSU

shaping tomorrow with you

Synchronization of WoT Servients

May 17th, 2017

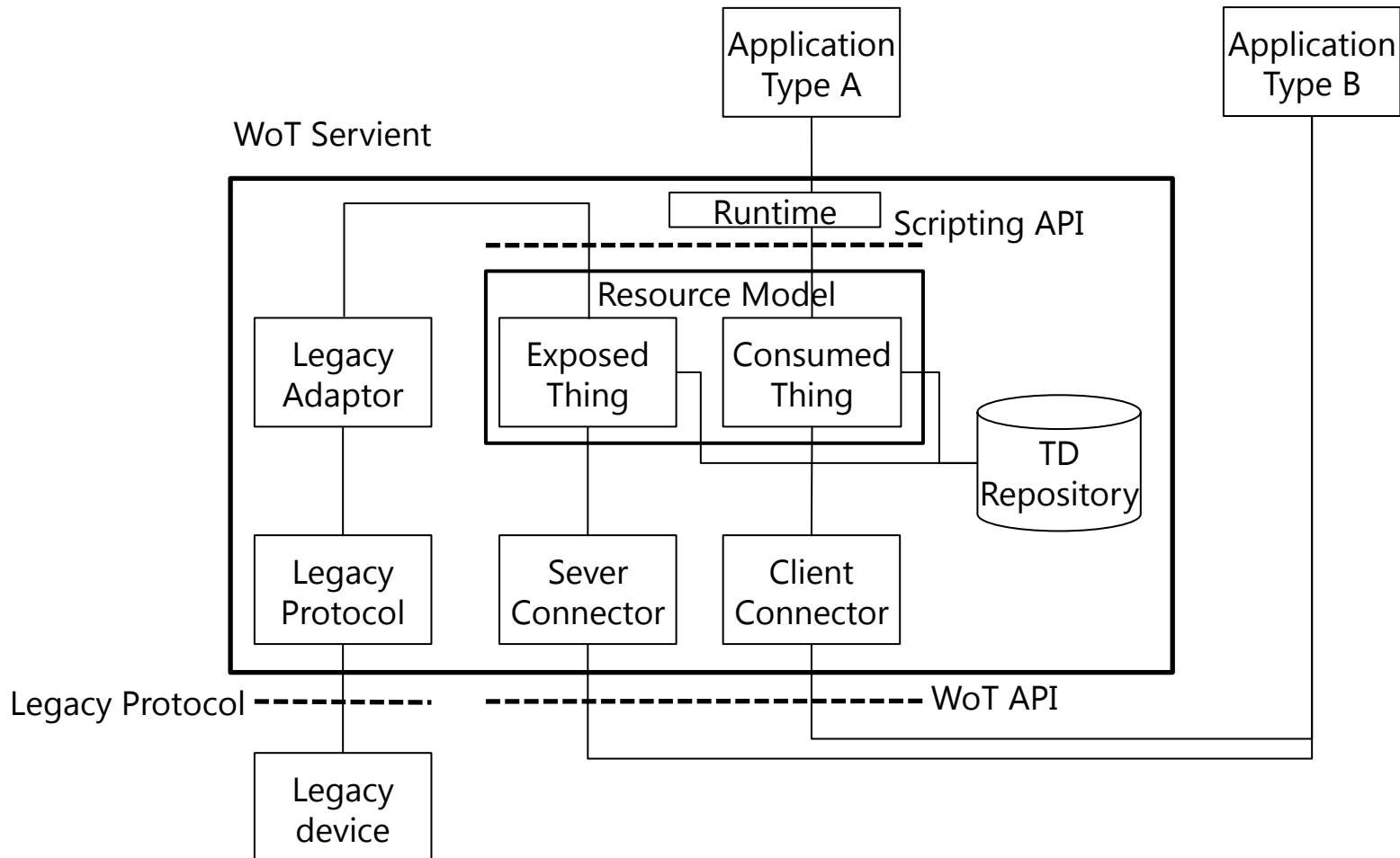
Ryuichi Matsukura

Fujitsu Laboratories Ltd. / Fujitsu Ltd.

- Example for basic processing flow inside WoT Servient
- Synchronization of WoT Servients
- Conclusion and proposals

WoT Servient architecture

- 2 types application are described in the below
 - Type A: uses Scripting API to execute
 - Type B: uses WoT API to execute

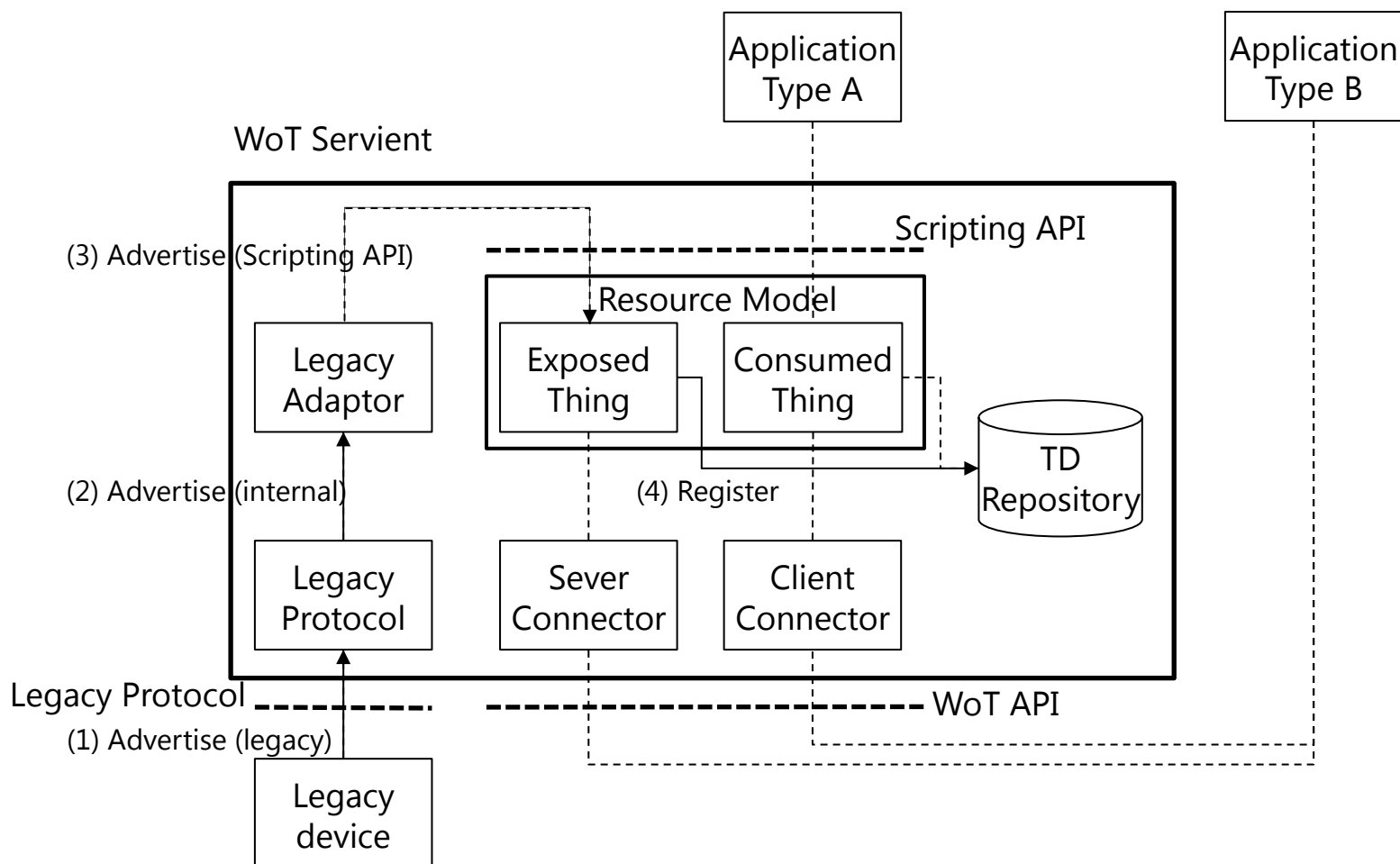


Overview of processing flow

- Discovery and provisioning of devices
 - Devices or applications initiate this process.
- Application handling devices
 - Application searches and recognizes TDs of the devices
 - Application gets from and sets to the devices
- Shutdown of the devices (omitted in this slides)

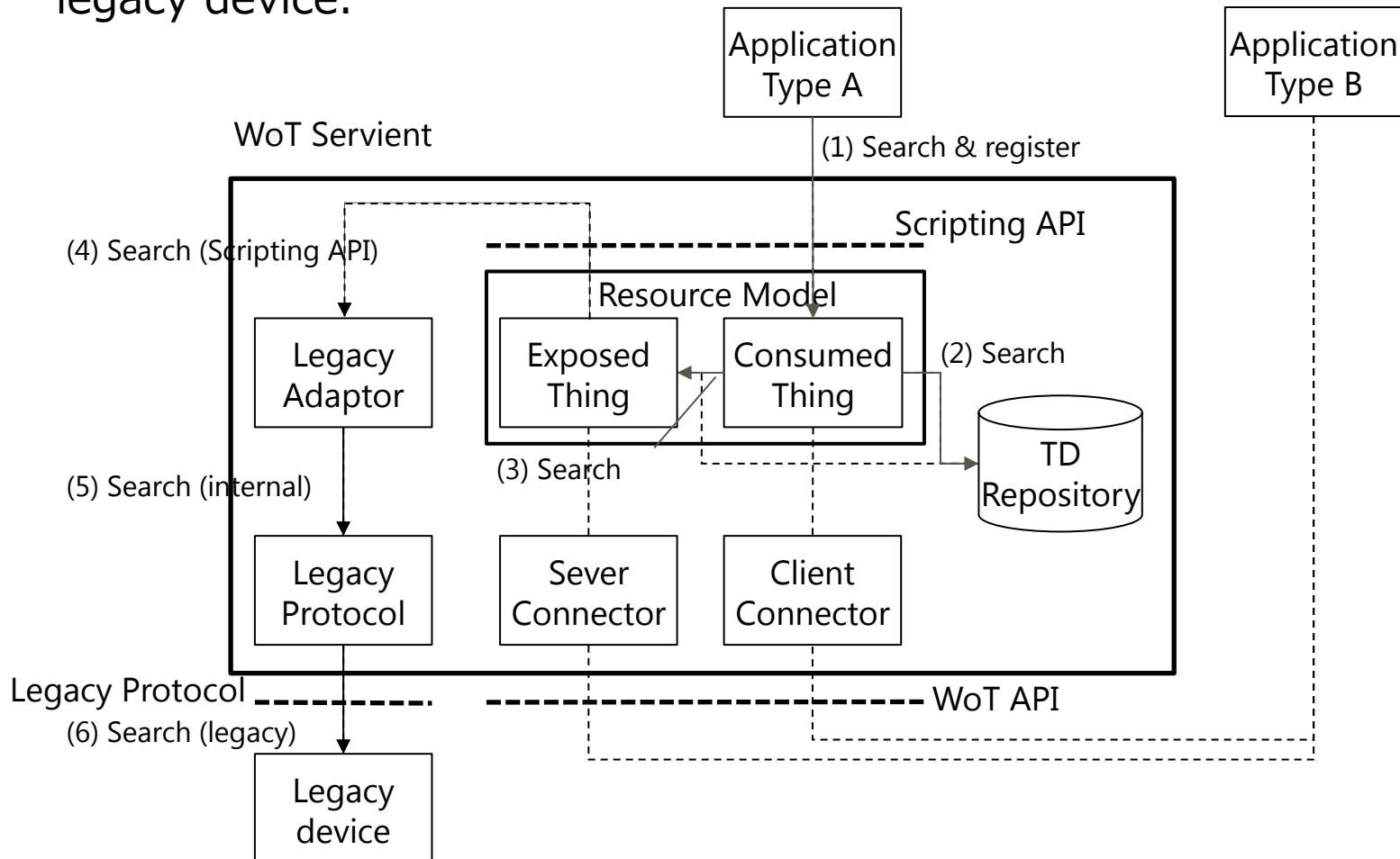
Discovery and Provisioning from device

- Legacy devices connect to Exposed Thing(ET) via legacy device adaptor.
- Legacy device adaptor converts from legacy protocol to Scripting API.



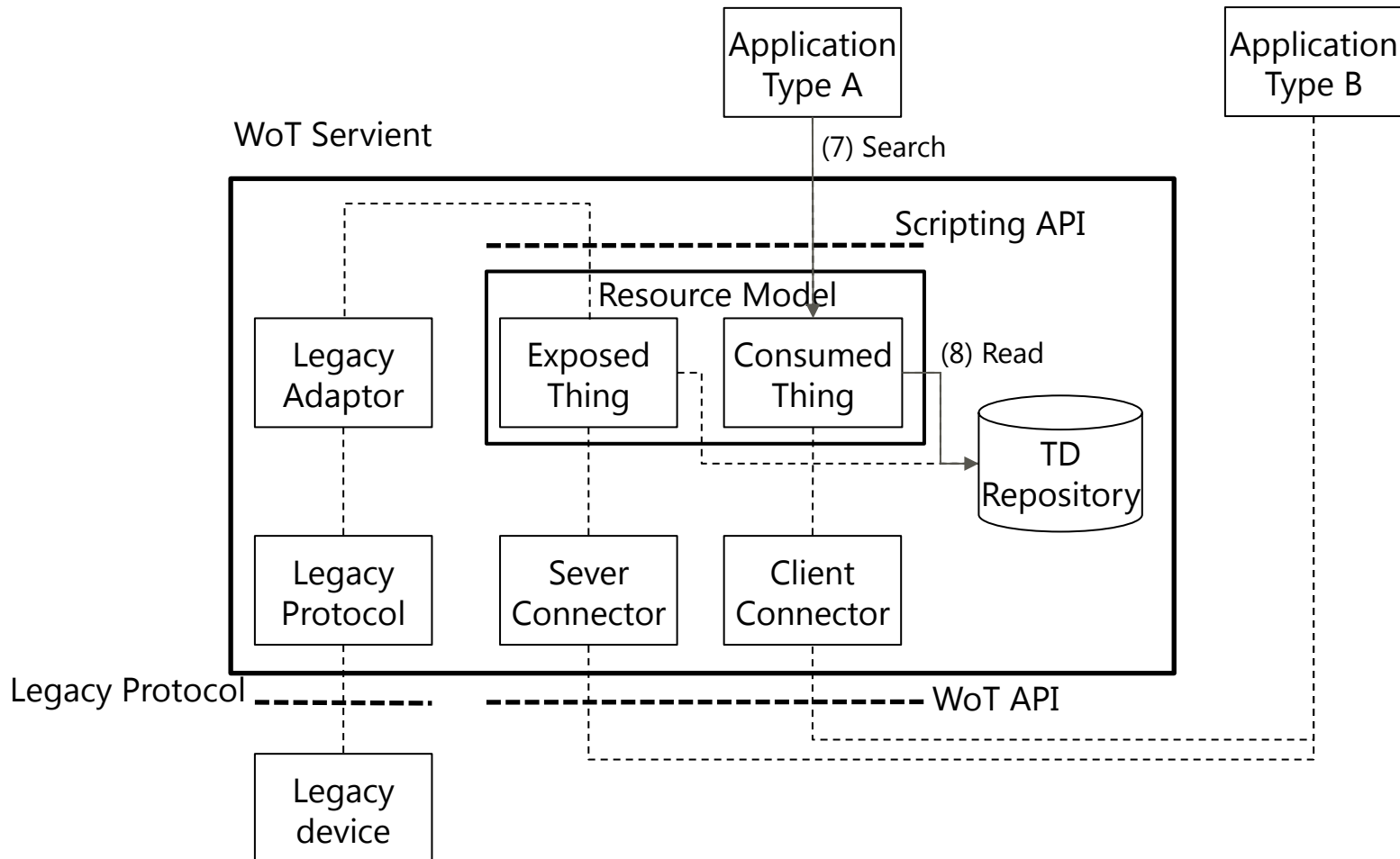
Discovery and Provisioning from App.

- Legacy devices is searched by Application Type A via legacy device adaptor.
 - Application Type A search and check to TD repository and search the legacy device.



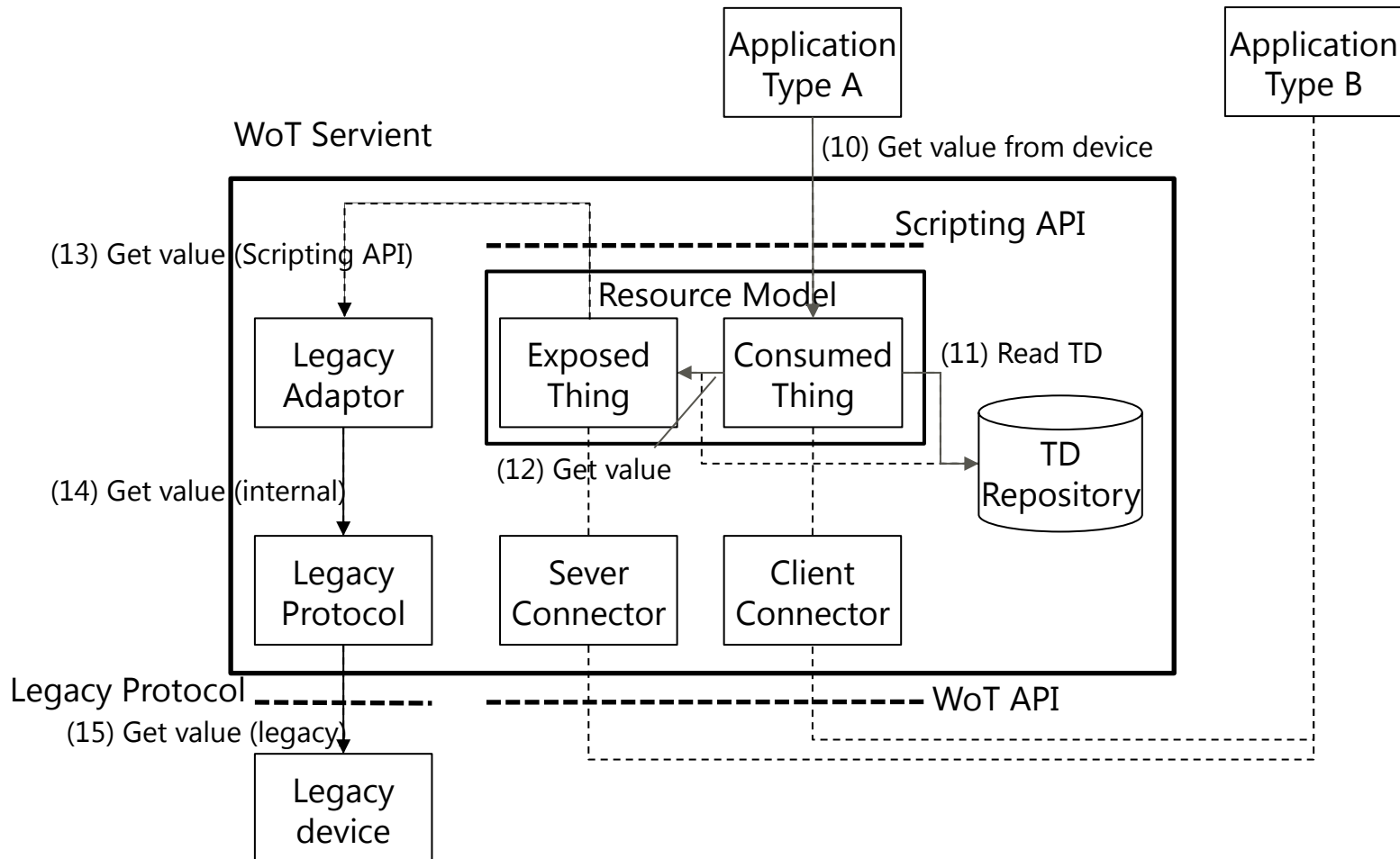
Application searches TD

- Application type A searches before beginning to handle the device.
 - Application asks consumed things if the devices is available now to search the TD corresponding to the device in the repository.



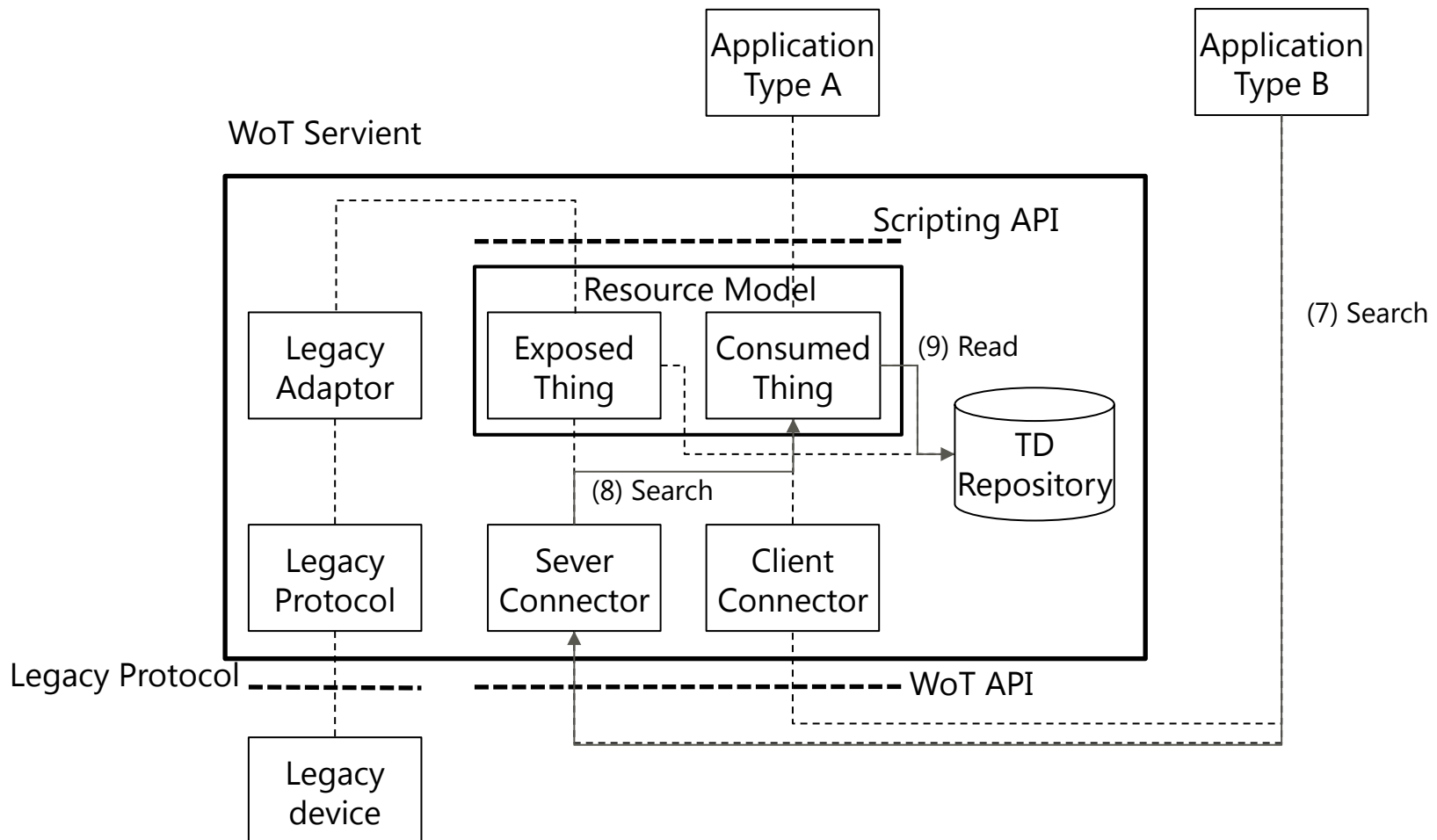
Application gets data from device

- Consumed Thing(CT) retrieve TD from the repository and send "get" to the device via legacy device adaptor.

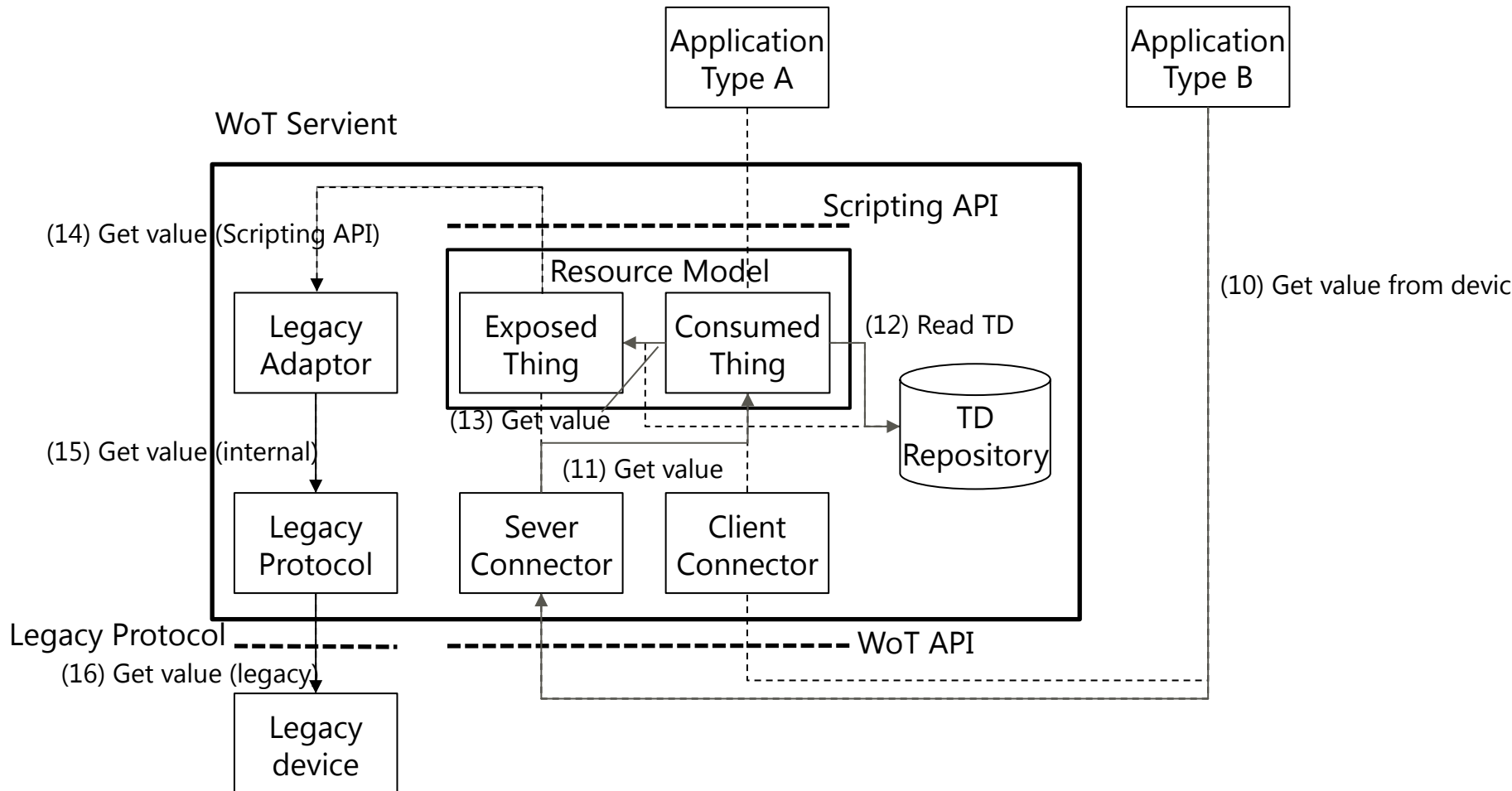


Application type B searches TD

- Application type B searches before beginning to handle the device.
 - Application asks consumed things if the devices is available now to search the TD corresponding to the device in the repository.



Application type B gets data from device

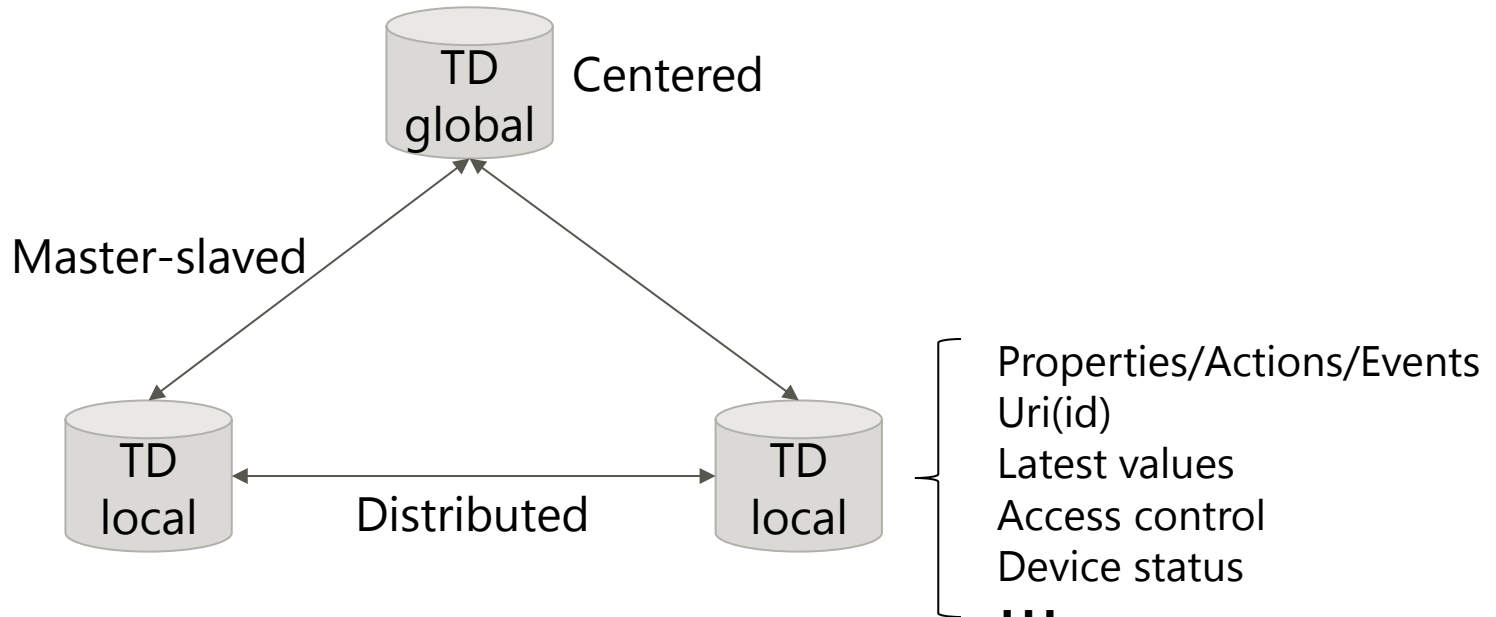


Why synchronization is required?

- Synchronization mechanism is required in some cases of management of multiple TD repositories on multiple WoT servients.
 - **Server-client/Master-slave**: master TD repository collects all devices connected to local WoT servients on gateways for instance.
 - **Distributed**: no master TD repository, all repository has all devices resources including devices connected to other servients.
 - **Redundant/Multiplexing**: redundant TD repositories for back-up purpose.

Some issues on multiple WoT servients

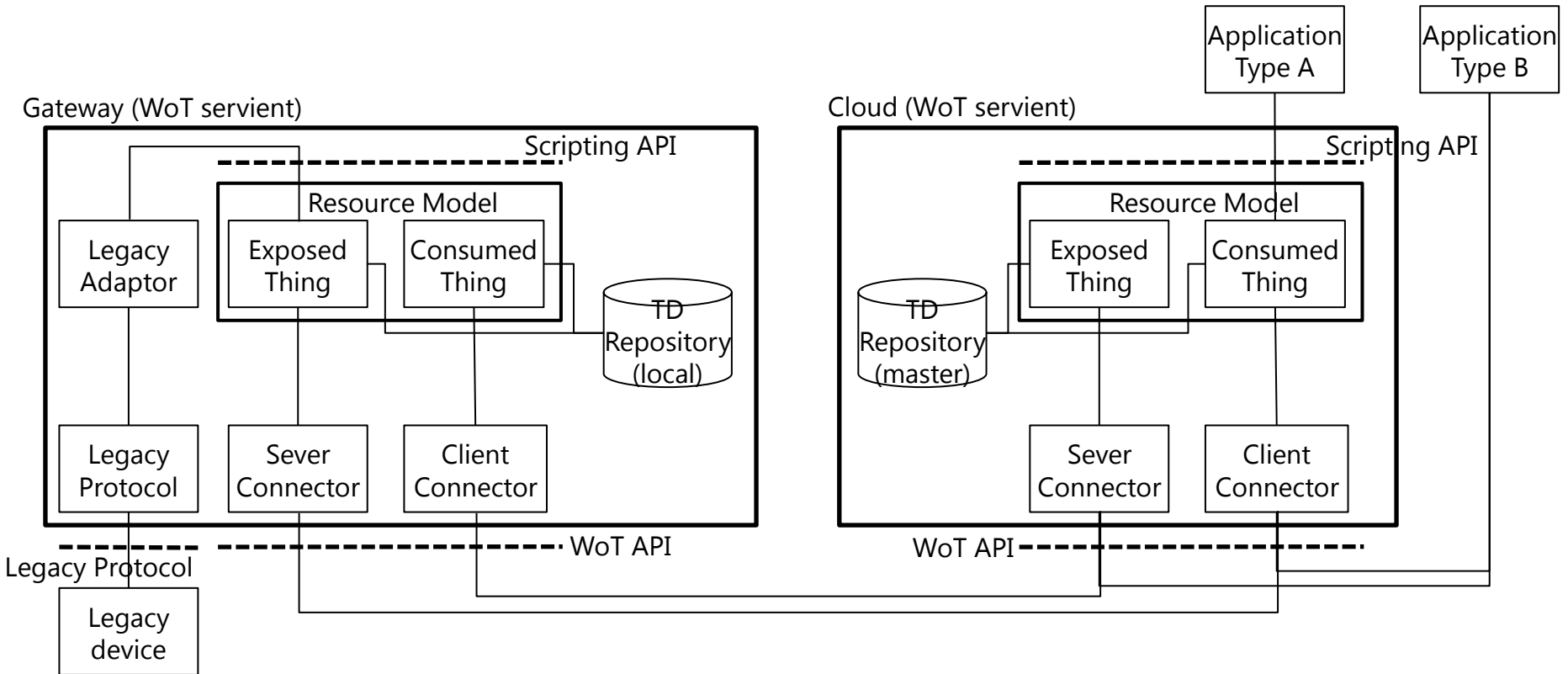
- Management of TD repositories on multiple WoT servients
 - Centered / Master-Slaved / Distributed
 - Data types for the repository: class definition / mapped physical devices / MPD with latest values for edge
- Access control to the distributed devices



Device status=Idle / connecting / connected / suspected/ disconnecting

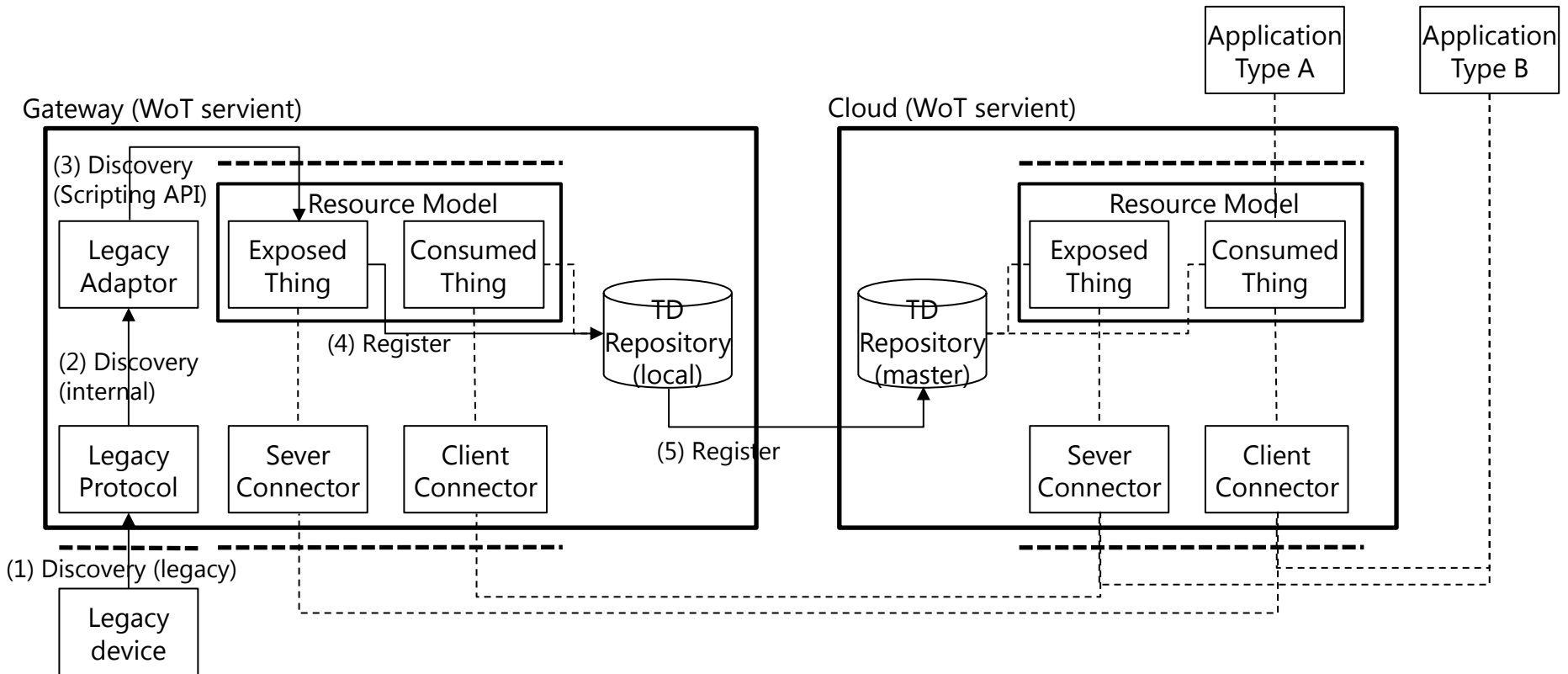
Example diagram for server-client

- This diagram is for discussion of synchronization of WoT Servients in case of server-client.
 - Gateway: WoT servient to connect local device
 - Cloud: WoT servient to connect applications
 - 2 WoT servients have TD repository. Repository on Cloud is master.



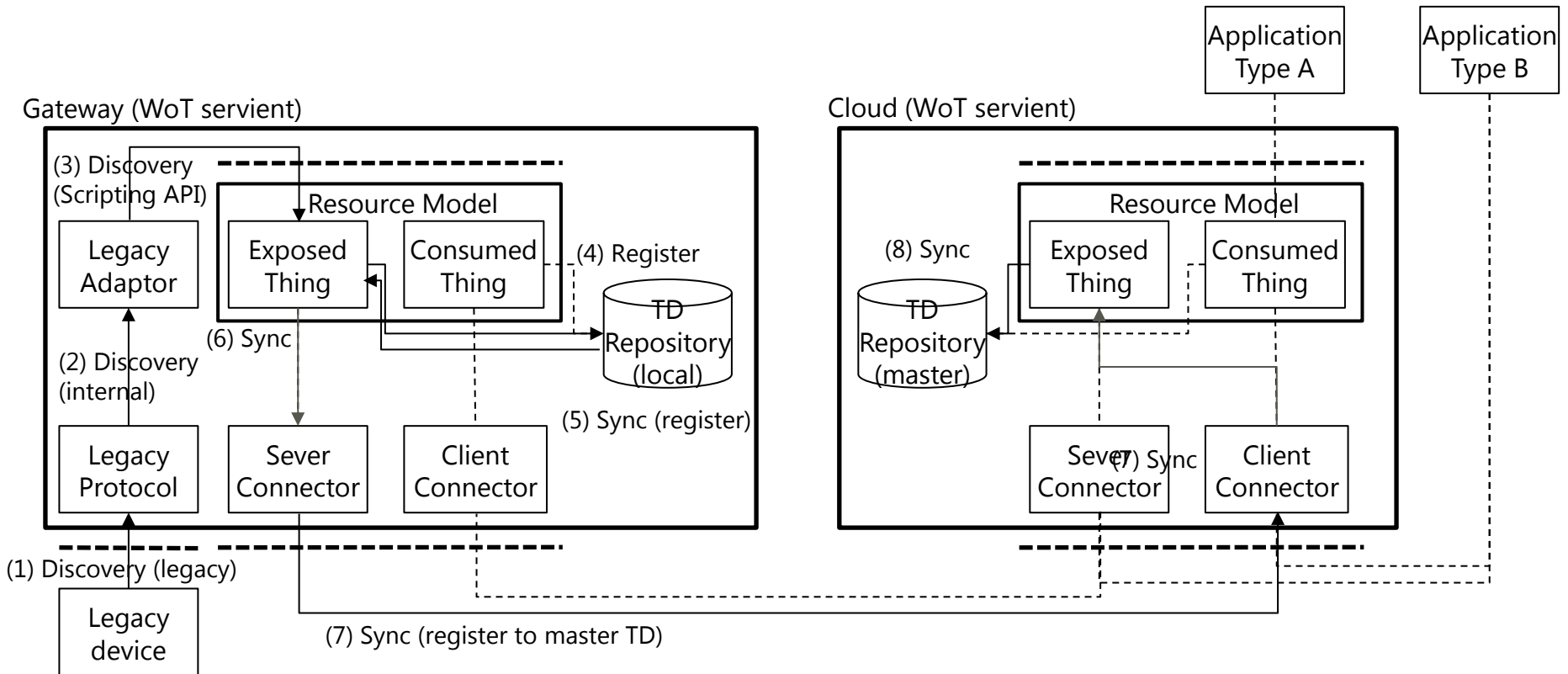
Device registered to master repository

- WoT servient on the gateway discover a device and register to local and master TD repository.
 - Local TD repository registers to master repository on WoT servient on cloud with synchronization protocol.
 - TD repositories synchronization protocol will be defined in this TF.



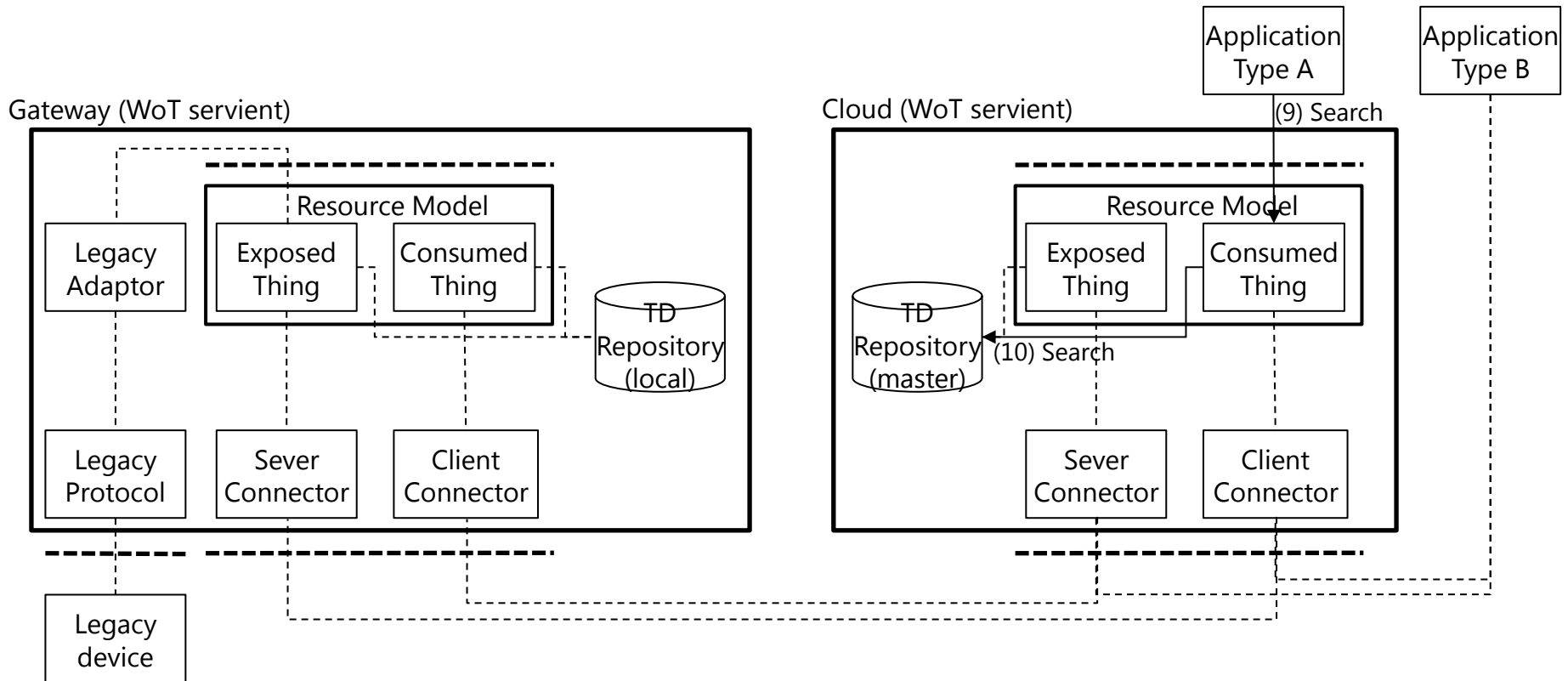
Another way on TD synchronization

- This example shows synchronization from local TD repository to master by WoT API with some extension.
 - Synchronization process starts after registering local device to the local repository.



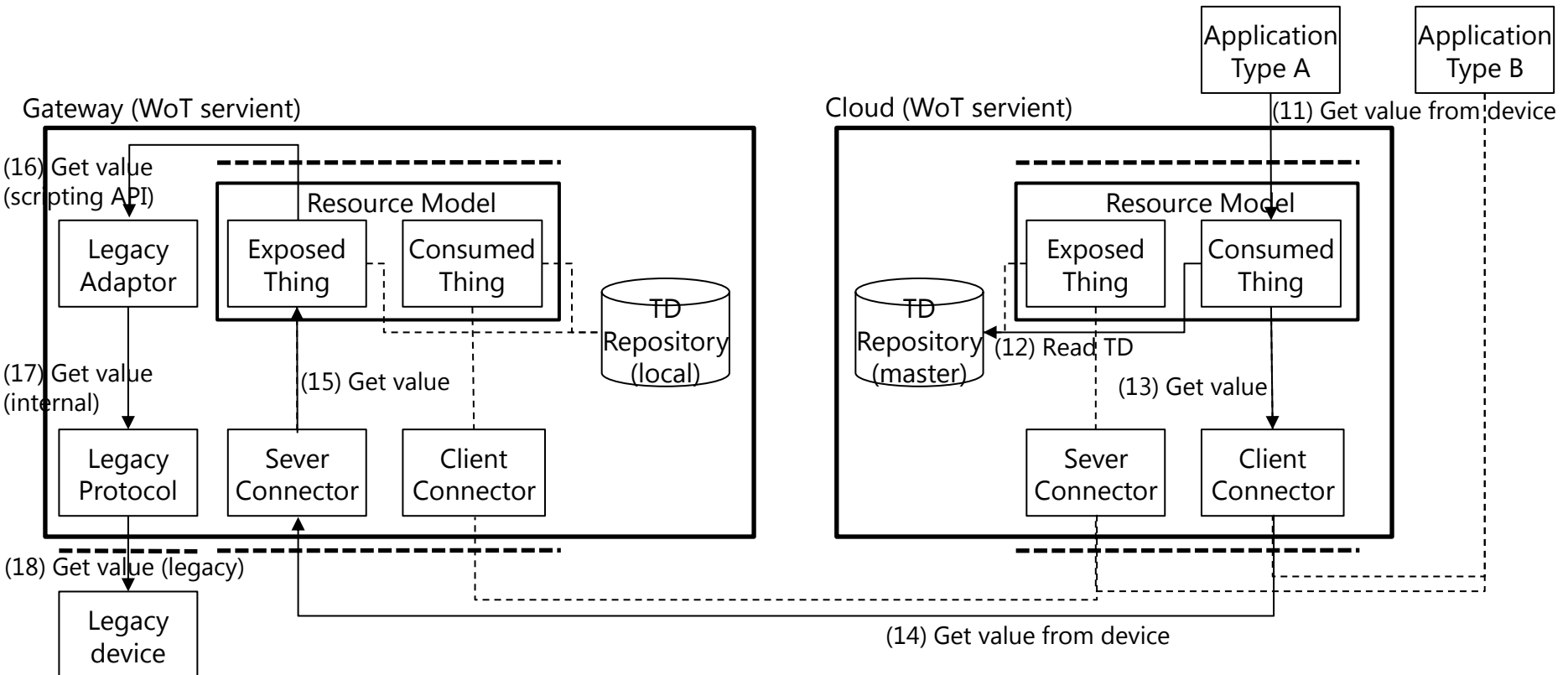
Application on cloud searches TD

- Application searches TD to TD repository, master, on the WoT servient on Cloud.
 - The master TD repository always has the latest TDs corresponding to TDs on each gateways.



App on cloud gets data from device

- CT on cloud retrieve TD from the repository and send “get” to the device on the gateway.
 - CT on cloud and ET on gateway communicate with WoT API.



■ Conclusions

- Synchronization mechanism is required for multiple WoT servients
- There are some methods to manage TD repositories on distributed servients.
- 2 types of application for the WoT servients.

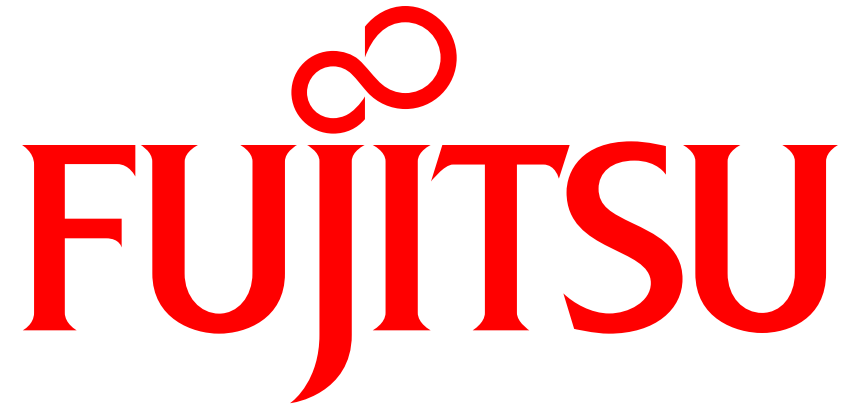
■ Issues for the next step

- Define protocols for synchronization between repositories.
- Define shadow device representation.
 - The shadow has uris for original and virtual(shadow), and the latest data/value(ex. current sensor value) on it.

■ Proposals

- Continue this discussion with Scripting TF if possible.
- Check consistency with Architecture and Scripting API document.
- Input the description to be discussed to the WG deliverables.

Thank you!



shaping tomorrow with you