

# Examples of Application for the Matter ECHONET Lite Bridge

**First Edition**



## Revision record

- |                     |               |                                    |
|---------------------|---------------|------------------------------------|
| • 1st edition draft | May 29, 2024  | Disclosed to member companies only |
| • 1st edition       | June 26, 2024 | Published for the general public   |

- The specifications published by the ECHONET Consortium are established without regard to industrial property rights (e.g., patent and utility model rights). In no event will the ECHONET Consortium be responsible for industrial property rights to the contents of its specifications.
  - In no event will the publisher of these specifications be held liable for any damages arising out of their use.

# Contents

Chapter 1	Introduction .....	1- 1
1.1	Backgrounds .....	1- 1
1.2	What is ECHONET Lite Bridge? .....	1- 1
1.3	Target Devices .....	1- 2
1.3.1.	Assumed system configurations.....	1- 2
1.4	Comparison of Terms used in Matter and ECHONET Lite standards.....	1- 3
1.5	Definition of terms.....	1- 4
1.6	Bibliography.....	1- 4
Chapter 2	Additional uses of ECHONET Lite Bridge.....	2- 1
2.1.	Uses of ECHONET Lite devices .....	2- 1
2.2.	Use of Bridge .....	2- 1
2.3.	Interlocking operation with Matter devices .....	2- 2
2.4.	Assumed actions .....	2- 2
2.4.1.	Configuration of ECHONET Lite Bridge.....	2- 2
2.4.2.	Devices to be bridged .....	2- 3
Chapter 3	Configuration models for the Matter ECHONET Lite Bridge.....	3- 1
3.1.	Technical Terms for the Matter ECHONET Lite Bridge.....	3- 1
3.1.1.	Bridge.....	3- 1
3.1.2.	Fabric .....	3- 1
3.1.3.	Commissioning .....	3- 1
3.1.4.	Node.....	3- 1
3.1.5.	Endpoint .....	3- 1
3.1.6.	Device Type .....	3- 1
3.1.7.	Cluster.....	3- 2
3.1.8.	Attribute .....	3- 2
3.1.9.	Interaction .....	3- 2
3.2.	Exposing functions and metadata of Bridged devices .....	3- 3
3.3.	Discovery of Bridged devices .....	3- 3
3.4.	Configuration of Bridged devices .....	3- 3
3.5.	New features for Bridged devices .....	3- 4
3.6.	Changes to the set of Bridged Devices .....	3- 4
3.7.	Changes to device names and grouping of Bridged Devices.....	3- 4
3.8.	Setup flow for a Bridge (plus Bridged Devices).....	3- 4
3.9.	Access control.....	3- 4

---

3.10.	Software update (OTA) .....	3- 4
3.11.	Example of Endpoint configuration for Bridged devices .....	3- 5
Chapter 4	Example of Bridge device sequences .....	4- 1
4.1.	Concept .....	4- 1
4.2.	Registration and deletion of devices to be bridged .....	4- 1
4.2.1.	Detection of devices to be bridged .....	4- 1
4.2.2.	Registration of devices to be bridged .....	4- 2
4.2.3.	Deleting Bridged devices .....	4- 3
4.3.	Writing/reading values to and from Bridged devices .....	4- 3
4.3.1.	Writing values to Bridged devices .....	4- 4
4.3.2.	Reading values from Bridged devices .....	4- 6
Chapter 5	Application examples .....	5- 1
5.1.	Overview .....	5- 1
5.2.	General lighting class application example .....	5- 1
5.3.	Home Air Conditioner Class application examples .....	5- 2
Chapter 6	Conclusion .....	6- 1

## Chapter 1 Introduction

### 1.1 Backgrounds

In 2022, the Connectivity Standards Alliance established a common standard for smart homes called “Matter”. The “Matter” standard is another standard for smart homes different from the ECHONET Lite standard, requiring a function to interconnect the two protocols in order for Matter and ECHONET Lite devices to interwork. User convenience can be improved and further utilization of ECHONET Lite devices can be expected by controlling ECHONET Lite devices not only with conventional ECHONET Lite controllers but also with Matter controllers.

Note that ECHONET Lite defines a function that connects an ECHONET Lite domain to an external system (including other ECHONET Lite domains) as a Gateway for ECHONET Lite, while Matter defines the same as a Bridge. This document describes a function for operating an ECHONET Lite system, which is an external system from the perspective of “Matter Fabric”. We will therefore call the function “Matter ECHONET Lite Bridge”.

### 1.2 What is ECHONET Lite Bridge?

The Matter ECHONET Lite Bridge is a function intended to enable the operation of smart home appliances and smart energy devices supported by ECHONET Lite through a Matter controller.

The assumed use case is controlling ECHONET Lite home appliances with Matter controllers by adding Matter ECHONET Lite Bridge device to households using ECHONET Lite home appliances and Matter devices. Figure 1-1 shows an example of a network configuration assuming use of Matter ECHONET Lite Bridge.

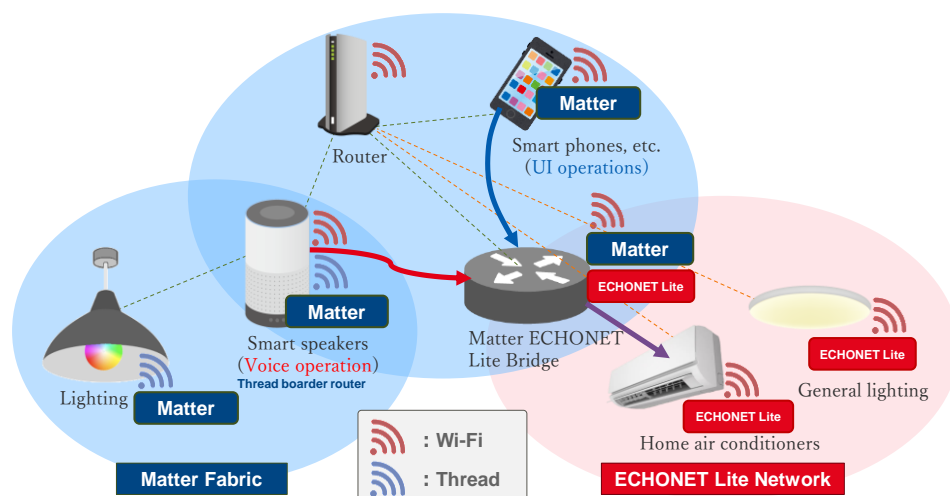


Figure 1-1 Example of a network configuration assuming use of Matter ECHONET Lite Bridge

## 1.3 Target Devices

Photovoltaic cells, fuel cells, smart electric energy meters, hot water supply systems (electricity/gas), EV chargers/chargers and dischargers, storage batteries, air conditioners, and lighting are defined as the “eight key devices” that comprise HEMS.

In this document, these “eight key devices” are considered Bridged devices that are controlled by Matter controllers via Matter ECHONET Lite Bridge.

### 1.3.1. Assumed system configurations

The system configuration in which the Matter ECHONET Lite Bridge device operates is shown in Figure 1-2.

#### (1) Networks

ECHONET Lite standards allow the use of multiple types of transmission media and protocols for network configuration, while Matter devices allow the use of either BLE (Bluetooth Low Energy) for device registration, Wi-Fi supporting IPv6 networks for device control, Thread, or Ethernet.

#### (2) Controllers

Matter controllers are assumed to be operated by voice from Matter smart home applications on a smartphone or from a smart speaker.

#### (3) Bridge configuration applications

This section assumes an application that detects ECHONET Lite devices that are non-Matter devices and selects devices to be bridged from the detected ECHONET Lite devices.

The Bridge configuration application environment depends on the system configuration; for example, in the following cases.

- Separately supplied as a smartphone application by a company that developed the Matter ECHONET Lite Bridge
- Implemented a Web server with Bridge configuration application functions on the Matter ECHONET Lite Bridge, and separately operated via a Web client (e.g. PC).

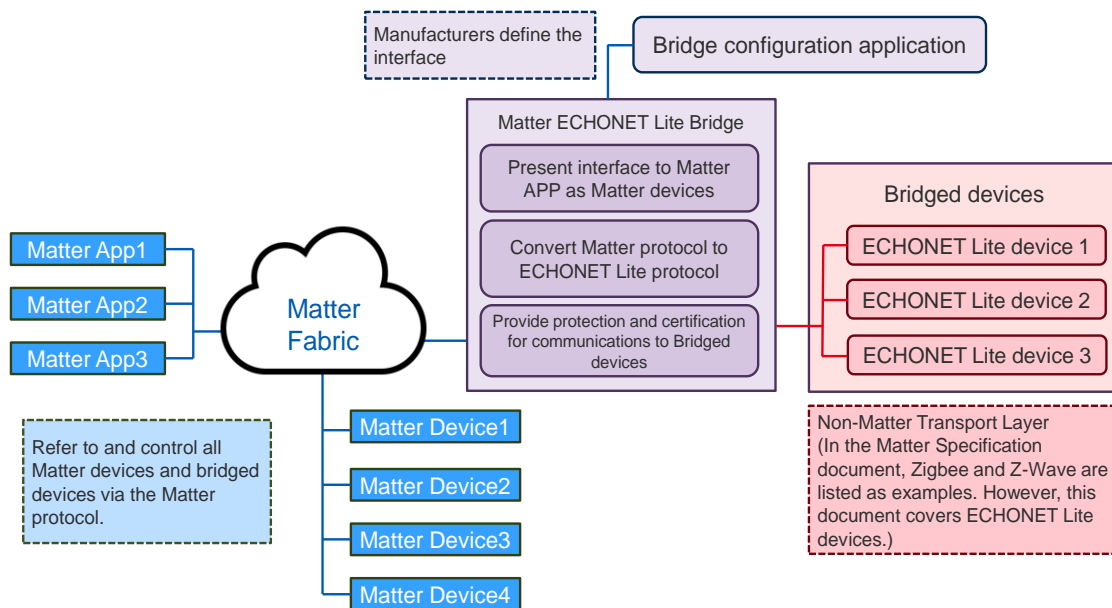


Figure 1-2 Matter ECHONET Lite Bridge schematic

#### 1.4 Comparison of Terms used in Matter and ECHONET Lite standards

The two standards use different terminologies whose meanings and concepts may not be same. For future reference, a mapping table for the terms used in the comparison of the two standards in preparing this document is shown in Table 1-1. For Matter standards, see reference [1] 1.4 Standards Terminology Mapping.

Table 1-1 Term Mapping Table

Matter	ECHONET Lite
Device	Device
Node	Node
Endpoint	—
DeviceType	Equivalent to ECHONET Lite object (EOJ) class group codes and class codes
—	ECHONET Lite Object (EOJ) instance codes
Cluster	—
Attribute	Equivalent to ECHONET Lite properties (EPC)
Command	Equivalent to ECHONET Lite Service (ESV), (Get, SetI (no response required), SetC (response required), etc.)
Event	—

## 1.5 Definition of terms

The terms used in the Matter standard are stated in Table 1-2.

Table 1-2 Definition of terms

Matter	Description
Attribute	A data entity which represents a physical quantity or state. This data is communicated to other Nodes using commands.
Bridge	A Node that represents one or more non-Matter devices on the Fabric.
Cluster	A specification defining one or more attributes, commands, behaviors and dependencies, that supports an independent utility or application function.
Command	Requests for action on a value with an expected response which may have parameters and a response with a status and parameters.
Commissioning	Sequence of operations to bring a Node into a Fabric by assigning an Operational Node ID and Node Operational credentials.
Controller	A Role of a Node that has permissions to enable it to control one or more Nodes.
Device	A piece of equipment containing one or more Nodes.
Endpoint	A particular component within a Node that is individually addressable.
Fabric	A logical collection of communicating Nodes, sharing a common root of trust, and a common distributed configuration state.
Node	An addressable entity which supports the Matter protocol stack and (once Commissioned) has its own Operational Node ID and Node Operational credentials. A Device MAY host multiple Nodes.

## 1.6 Bibliography

Table 1-3 List of Reference documents

Document number	Title	Website
[1]	Matter-1.2-Core-Specification, CSA, October 18, 2023	<a href="https://csa-iot.org/developer-resource/specifications-download-request/">https://csa-iot.org/developer-resource/specifications-download-request/</a>
[2]	Matter-1.2-Device-Library-Specification, CSA, October 18, 2023	
[3]	Matter-1.2-Application-Cluster-Specification, CSA, October 18, 2023	
[4]	Matter-1.2-Standard-Namespace-Specification, CSA, October 18, 2023	



## Chapter 2 Additional uses of ECHONET Lite Bridge

### 2.1. Uses of ECHONET Lite devices

ECHONET Lite Devices support the ECHONET Lite Standards that unified communication protocol specifications unique to each manufacturer. This allows control of devices from a single ECHONET Lite controller regardless of manufacturer or device (See Figure 2-1).

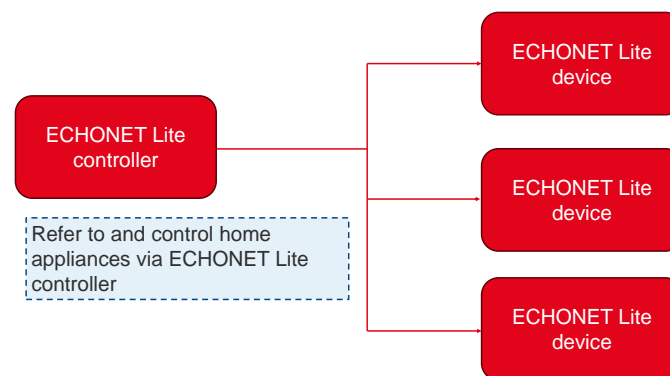


Figure 2-1 Operation of ECHONET Lite devices

However, note that devices supporting different smart home standards are not compatible with ECHONET Lite, and it is not possible to control Matter devices via ECHONET Lite controllers nor control ECHONET Lite devices via Matter controllers.

### 2.2. Use of Bridge

Figure 2-2As shown in Figure 2-2, Matter ECHONET Lite Bridge enables control of ECHONET Lite devices via a Matter controller.

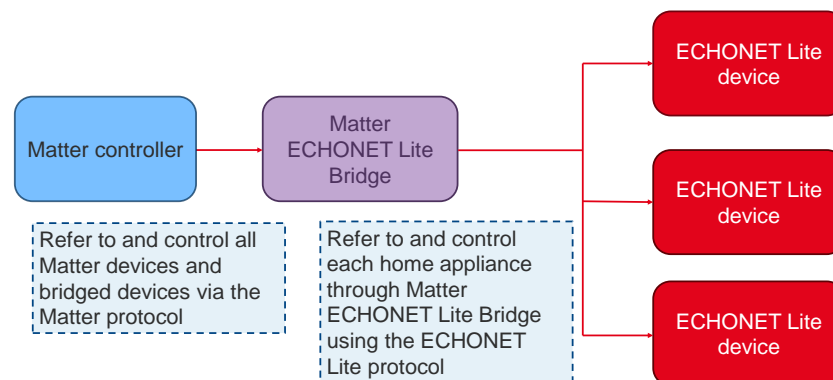


Figure 2-2 Operation using the Matter ECHONET Lite Bridge

The following use cases are possible for devices supporting the general lighting and home air conditioner classes of the ECHONET Lite standards using the Matter ECHONET Lite Bridge.

- (1) Operation of devices via Matter Controller
- (2) Operation using voice control of devices via smart speakers
- (3) Operation through interoperability of Matter devices and ECHONET Lite devices

### 2.3. Interlocking operation with Matter devices

With the Matter ECHONET Lite Bridge, it is possible to handle ECHONET Lite devices as Matter devices. Consequently, ECHONET Lite devices can be selected as control targets for Matter controller home automation functions, and Matter devices and ECHONET Lite devices can be interoperable via Matter controllers.

The following shows examples of interoperability:

Ex. 1) When a user returns home

Turning on Matter ceiling lights and ECHONET Lite air conditioners after unlocking the door with Matter smart locks.

Ex. 2) Measures against heat stroke

Turning on ECHONET Lite air conditioners when a Matter room temperature sensor detects a temperature of 28°C or higher.

Ex. 3) When a user goes to bed

Turning off ECHONET Lite lighting devices by voice operation via Matter smart speakers.

### 2.4. Assumed actions

#### 2.4.1. Configuration of ECHONET Lite Bridge

The following shows an example of procedures up to the point where a user controls ECHONET Lite devices from a Matter controller using a Matter ECHONET Lite Bridge.

- (1) Turn on the Matter ECHONET Lite Bridge
- (2) Commission Matter ECHONET Lite Bridge using a QR code or the like, and set the Wi-Fi SSID and a pass phrase in the course of commissioning
- (3) Search ECHONET Lite devices from the Matter ECHONET Lite Bridge settings screen (See Note 1)
- (4) Select and register the device(s) to be bridged from the list of ECHONET Lite devices detected on the Matter ECHONET Lite Bridge setting screen
- (5) Control ECHONET Lite devices using the Matter controller

Note 1: Searching for ECHONET Lite devices before commissioning the Matter ECHONET Lite Bridge is also possible, so the order of the procedure may be altered.

### 2.4.2. Devices to be bridged

Table. 2-1 shows definitions of DeviceType standardized by version up to the Matter Version 1.2.

Within this list, home air conditioners and general lighting are included in the device object classes defined in the ECHONET Lite standards as the “eight priority devices”.

Table. 2-1 DeviceType for Matter

Version	Device Type
Matter 1.0/1.1	<ul style="list-style-type: none"> <li>➤ Matter controllers</li> <li>➤ Bridge (to include devices that use Zigbee, Z-Wave, or the like)</li> <li>➤ Lighting (bulbs, tape lights, etc.)</li> <li>➤ Power supplies (smart plugs, etc.)</li> <li>➤ Safety and security sensors (burglar alarms, etc.)</li> <li>➤ Air-conditioning controllers (thermostats, air conditioners, etc.)</li> <li>➤ Window-related items and shades (smart blinds, smart shades, etc.)</li> <li>➤ Door locks (smart locks, etc.)</li> <li>➤ Media devices (TV, audio, etc.)</li> </ul>
Matter 1.2	<ul style="list-style-type: none"> <li>➤ Refrigerators</li> <li>➤ Room air conditioners (power on/off devices are added in addition to conventional air-conditioning controls)</li> <li>➤ Dishwashers</li> <li>➤ Washing machines</li> <li>➤ Robot vacuums</li> <li>➤ Smoke/carbon monoxide detectors</li> <li>➤ Air quality sensors</li> <li>➤ Air cleaners</li> <li>➤ Fans (electric fans, ventilation fans, etc.)</li> </ul>

In this document, “eight priority devices” not defined by DeviceType for Matter Version 1.2 (such as smart meters) are not bridged.

For DeviceType items to be added to Matter Version 1.2 or later, refer to the Matter specifications provided separately.

## Chapter 3 Configuration models for the Matter ECHONET Lite Bridge

### 3.1. Technical Terms for the Matter ECHONET Lite Bridge

This section lists technical terms needed for the development of Matter ECHONET Lite Bridge along with references.

#### 3.1.1. Bridge

A Bridge serves to allow the use of non-Matter IoT devices (e.g. devices on a ECHONET Lite, Zigbee or Z-Wave network, or any other non-Matter connectivity technology) in a Matter Fabric.

For details, refer to reference [1] “9.12. Bridge for non-Matter devices”.

#### 3.1.2. Fabric

A fabric is a set of nodes that interact by accessing data model elements as defined in the Interaction Model. A fabric is a security domain that allows a set of nodes to be identified and communicate within the context of the domain.

For details, refer to references [1] “2.5. Identifiers” and “7.5. Fabric”.

#### 3.1.3. Commissioning

Device commissioning is the process of joining a device to a Fabric. The device being commissioned is referred to as the Commissionee and the device administering commissioning is referred to as the Commissioner.

For details, refer to reference [1] “5. Commissioning”.

#### 3.1.4. Node

A Node encapsulates an addressable, unique resource on the network that has a set of functions and capabilities that a user recognizes distinctly as a functional whole. This distinction is usually physical, such as the physical device itself, or a logical instance of a physical device. A node is composed of one or more endpoints. For details, refer to reference [1] “7.8. Node”.

#### 3.1.5. Endpoint

An endpoint is an instance of something that could be a service or virtual device as indicated by a device type. For the services indicated by DeviceType, refer to Matter 1.2 “Device Library Specifications”.

For details, refer to reference [1] “7.9. Endpoint”.

#### 3.1.6. Device Type

A device type defines conformance for a set of one or more endpoints. A device type

defines a set of requirements for the node or endpoint in the market.  
For details, refer to reference [1] “7.15. Device Type”.

### 3.1.7. Cluster

Clusters are the functional building block elements of the data model. A cluster specification defines both a client and server side that correspond with each other through interactions. Each cluster is defined by a cluster specification that defines elements of a cluster including attributes, events, commands, as well as behavior associated with interactions with these elements.

The determination of the Client/Server is defined in detail by each DeviceType specification, while the actions and components at each Client/Server are defined by the Cluster specifications.

For details, refer to reference [1] “7.10. Cluster”.

Furthermore, Clusters are classified by Role. Clusters that are considered Matter Utilities, such as Descriptor Clusters and Identify Clusters, are called “Utility Clusters”. Clusters such as On/Off Clusters and Thermostat Clusters are called “Application Clusters”.

For details on the definition of each cluster, refer to reference [1] “Chapter 9. System Model Specification” or reference (3).

### 3.1.8. Attribute

An attribute is cluster data.

For details, refer to reference [1] “7.12. Attribute”.

### 3.1.9. Interaction

An interaction is a sequence of one or more transactions between nodes, that occurs in the context of an accessing fabric, or no fabric. Table. 3-1 shows the list.

As shown in Figure 3-1, each Interaction is composed of Transactions, and Transactions consist of Actions. Each Action can be transmitted by one or more Messages.

Table. 3-1 List of Matter Interactions

Interaction	Description
Read	This interaction is a request for cluster attributes and/or event data.
Write	This interaction modifies cluster attributes.
Invoke	This interaction invokes cluster commands.
Subscribe	This interaction subscribes to cluster attributes and/or event data. Subscriptions can be linked to Attributes and Events.

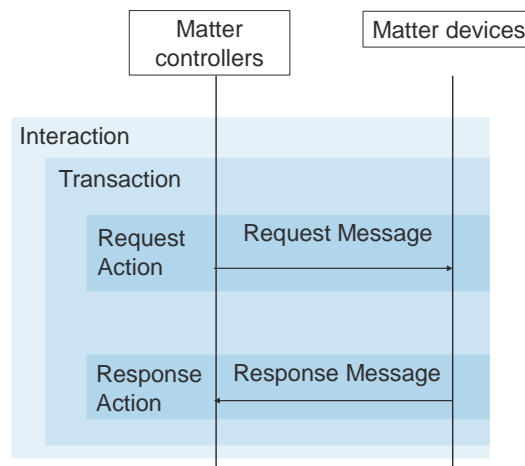


Figure 3-1 Overview of interaction sequence

### 3.2. Exposing functions and metadata of Bridged devices

After Commissioning, the Bridge SHALL expose (at least) one Node to the Fabric. The device implementing the Bridge MAY have more than one Node. The following Endpoints shall be exposed on this Node.

- Endpoints with a DeviceType supporting non-Matter devices
- Endpoints with an Aggregator, a DeviceType used to manage a group of Endpoints for Bridged devices.

For details on this and other metadata information, refer to reference [1] “9.12.2. Exposing functionality and metadata of Bridged Devices”.

### 3.3. Discovery of Bridged devices

A Node which discovers another Node with device type Aggregator on one of its endpoints SHOULD walk the entire tree of endpoints via the PartsList attributes and endpoints to discover the list of Bridged Devices, including their device types and other attributes, as well as any native Matter functionality potentially present on the Node.

Furthermore, the Bridge may expose a large number of Bridged devices. Therefore, they must allow users to easily identify and recognize the various Bridged devices.

For details, refer to reference [1] “9.12.3. Discovery of Bridged Devices”.

### 3.4. Configuration of Bridged devices

For the configuration of detected bridge devices, there are two basic archetypes. One is for actuators and the other is for sensors/switches.

For details, refer to reference [1] “9.12.4. Configuration of Bridged Devices”.

### 3.5. New features for Bridged devices

Bridged Devices can have their software updated independently of the Bridge, through Bridge Manufacturer-specific means.

For details, refer to reference [1] “9.12.5. New features for Bridged Devices”.

### 3.6. Changes to the set of Bridged Devices

Bridged Devices can be added to or removed from the Bridge through Bridge-specific means.

For details, refer to reference [1] “9.12.6. Changes to the set of Bridged Devices”.

### 3.7. Changes to device names and grouping of Bridged Devices

Typically, the user has some means (e.g. a Manufacturer-provided app) to assign names to the Bridged Devices, or names could be assigned automatically by the Bridge. The Bridge SHALL expose such names in the NodeLabel attribute of the Bridged Device Basic Information cluster on the applicable endpoint. Similarly, the user typically has some means to group the Bridged Devices (e.g. via a room/zone-concept) and provide names to such groups, or grouping could be applied automatically by the Bridge.

For details, refer to reference [1] “9.12.7. Changes to device names and grouping of Bridged Devices”.

### 3.8. Setup flow for a Bridge (plus Bridged Devices)

The Bridge together with its Bridged Devices is exposed as a single Node with a list of endpoints. Consequently, a single Node ID and a single Operational Certificate is assigned during Commissioning and a single pass through the commissioning flow is required to bring the Bridge (along with its Bridged Devices) onto a Fabric.

For details, refer to reference [1] “9.12.8. Setup flow for a Bridge (plus Bridged Devices)”.

### 3.9. Access control

The Bridge is a Matter node, therefore it has a single Access Control Cluster for the entire Node, like every other Matter Node. This cluster contains all Access Control Entries for each of its endpoints, including for all Bridged Devices and other native Matter functionality exposed by the Bridge Node. For details, refer to reference [1] “9.12.9. Access Control”.

### 3.10. Software update (OTA)

The Bridge is a Matter device and its Matter-related functionality MAY be updated using the mechanism described in reference [1] “11.20. Over-the-Air (OTA) Software Update”.

The Bridged Devices, on the other hand, are not native Matter devices, do not have a Product ID, and are not listed in the Distributed Compliance Ledger. They are typically updated using a mechanism defined and deployed by the Bridge Manufacturer.

For details, refer to references [1] “9.12.10. Software update (OTA)” and “11.19. Over-the-Air (OTA) Software Update”.

### 3.11. Example of Endpoint configuration for Bridged devices

Figure 3-2 below shows examples of Endpoint configurations in the case of Bridging ECHONET Lite general lighting devices and home air conditioners.

The Matter ECHONET Lite Bridge in this example has a Root Node (Endpoint 0 in the figure), which is a DeviceType that establishes the Bridge device itself as a Matter device, an Aggregator (Endpoint 1 in the figure), which is a DeviceType for managing the Bridged device in Matter, and a dynamic Endpoint (Endpoint 2 and Endpoint 3 in the figure), which is generated from the Endpoint information of each device when the Bridge is set up.

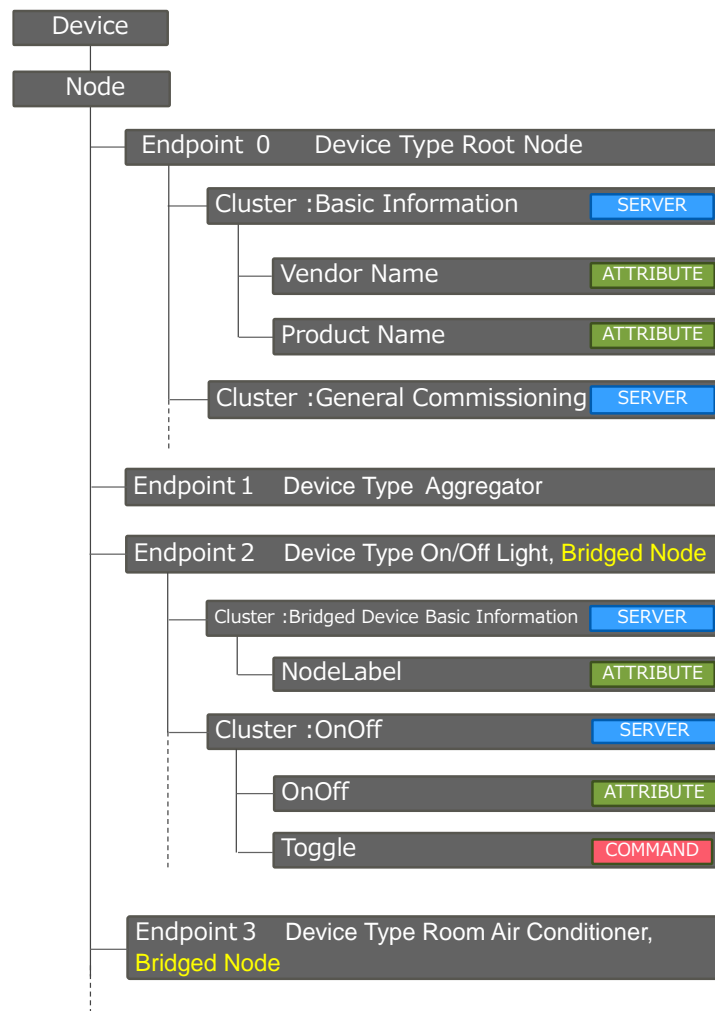


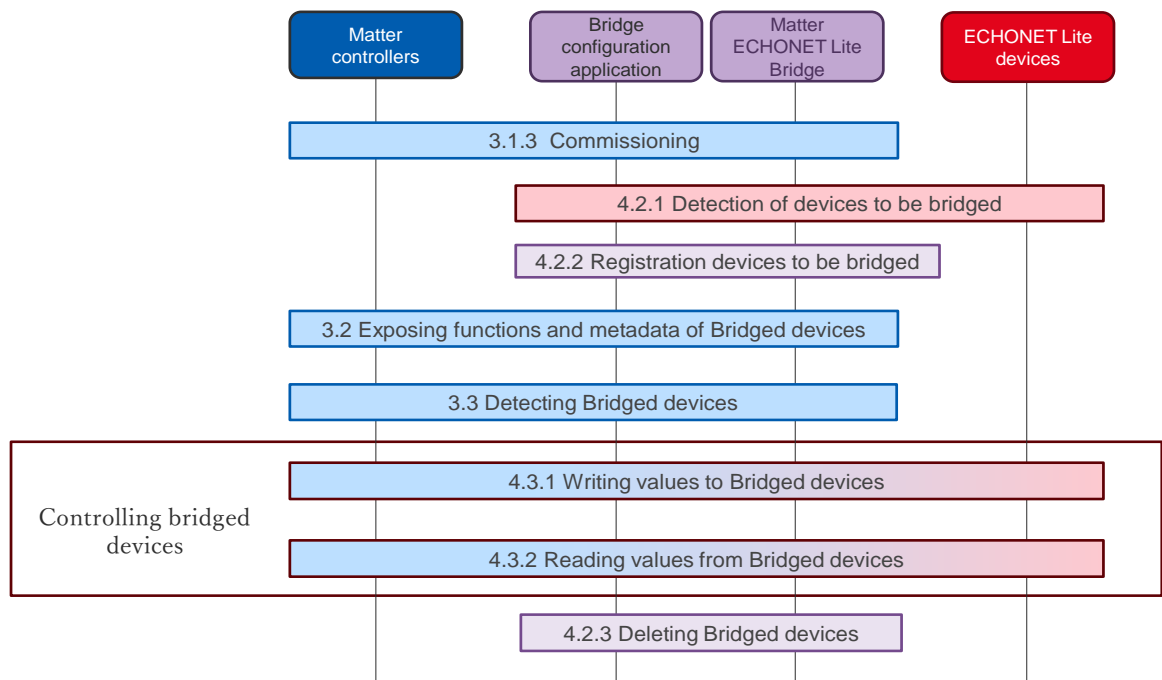
Figure 3-2 Example of Endpoints representing Bridge devices



## Chapter 4 Example of Bridge device sequences

### 4.1. Concept

This chapter describes an example sequence of basic Bridge operations based on the Bridge specifications in Matter described in the previous chapter. Figure 4-1 shows the process of enabling the use of Bridged devices using Matter ECHONET Lite Bridge.



(Note 1) It is possible to search ECHONET Lite devices before Commissioning, so the order of the procedure may be altered.

Figure 4-1 Overview of the Matter ECHONET Lite Bridge sequence

### 4.2. Registration and deletion of devices to be bridged

The following describes the procedure for registering or deleting ECHONET Lite device(s) to be bridged to the Matter ECHONET Lite Bridge.

#### 4.2.1. Detection of devices to be bridged

The Matter ECHONET Lite Bridge must detect ECHONET Lite devices by some means before registering them to the Bridge.

This section assumes, as an example, a Bridge configuration application running outside the Matter ECHONET Lite Bridge. Figure 4-2 describes a sequence assuming a scene in which the user operates the Bridge configuration application to detect ECHONET Lite devices. Note that the method of connecting to the network is implementation-dependent.

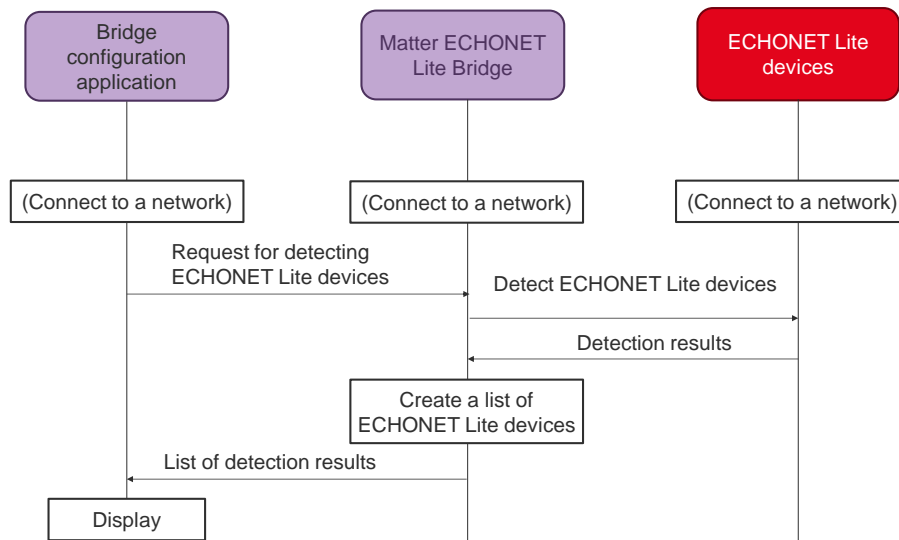


Figure 4-2 Detection of ECHONET Lite devices

#### 4.2.2.Registration of devices to be bridged

The user needs to select and register the device(s) to be actually bridged from the detected items displayed in the Bridge configuration application, for example. Figure 4-3 shows a sequence in which the user select and register devices to be bridged.

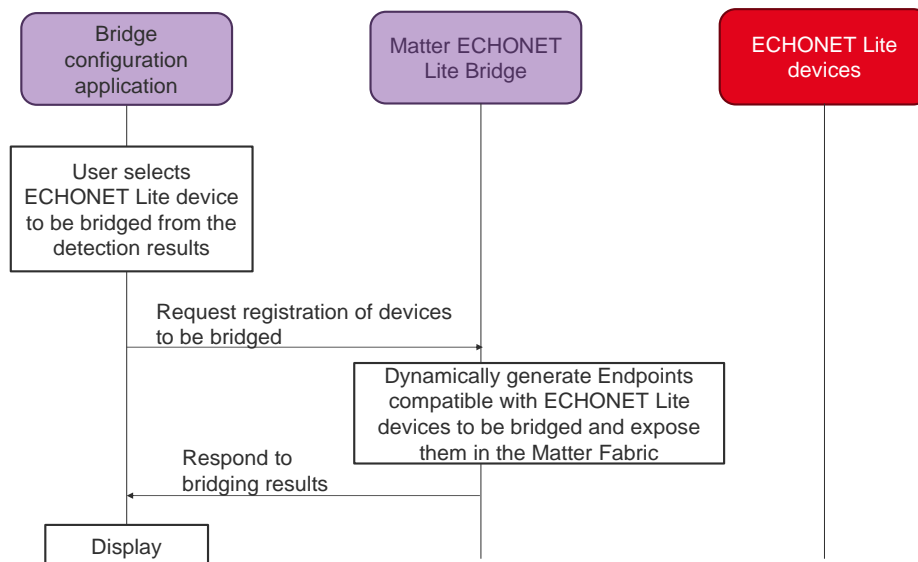


Figure 4-3 Registration of Bridge devices

### 4.2.3. Deleting Bridged devices

In this section, Figure 4-4 shows the sequence in which the user operates the Bridge configuration application to delete ECHONET Lite devices that are Bridged devices.

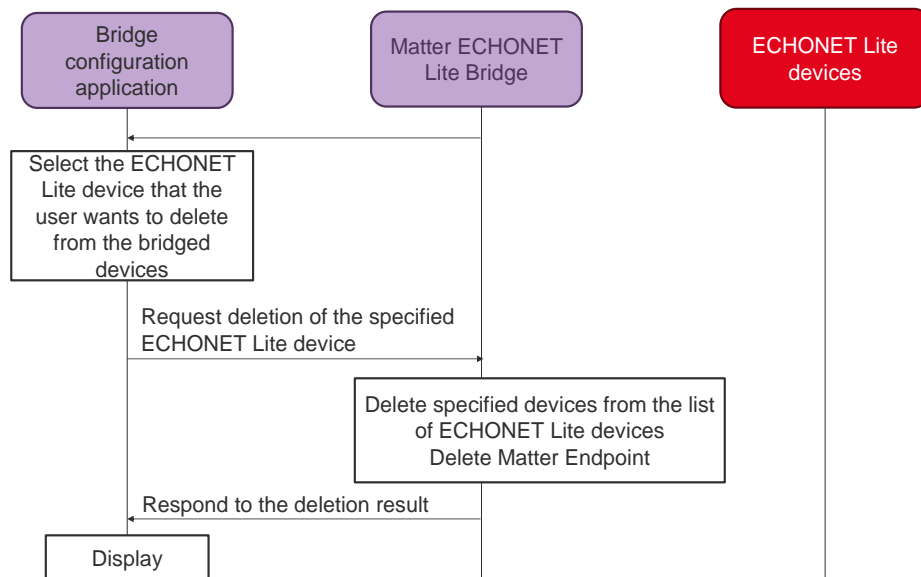


Figure 4-4 Deletion of Bridge devices

### 4.3. Writing/reading values to and from Bridged devices

After registering to the Matter ECHONET Lite Bridge, the user operates the Bridged devices through a Matter controller or similar.

This section describes the sequence of basic actions (read/write) in the operation. Note that this section only describes the control flow. Specific setting values for each class are discussed in Chapter 5 below.

A control request from the Matter controller to the Matter ECHONET Lite Bridge is sent to the Bridged devices as an ECHONET Lite service, after the protocol is converted. ECHONET Lite services include SetI, which does not require a response from the ECHONET Lite device, and SetC, which requires a response from the device. However, the ECHONET Lite services used by the Matter ECHONET Lite Bridge are implementation-dependent according to the specifications of the Bridged devices.

Note that a Matter controller different from the device that performed the Commissioning can also operate the Bridged device based on the Access Control settings.

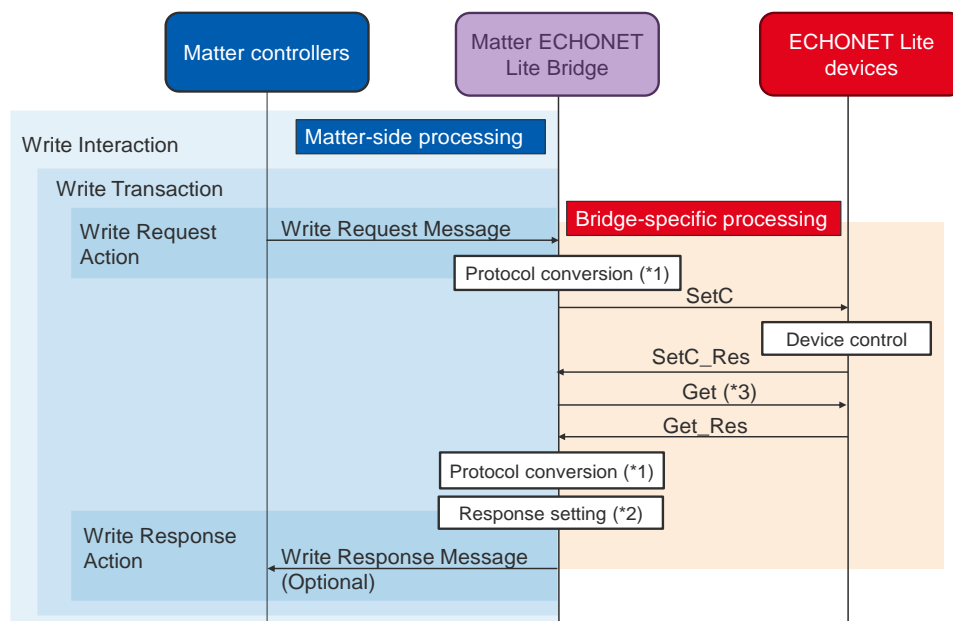
### 4.3.1. Writing values to Bridged devices

#### (1) Controlling a device using SetC

A request for control to be sent from the Matter controller to the Matter ECHONET Lite Bridge will write Attributes via Write Interaction as shown in Figure 4-5.

When a Command corresponding to the writing of an Attribute is invoked by the Invoke Interaction, the same sequence should be followed.

For details, refer to reference [1] “8. Interaction Model Specification”.



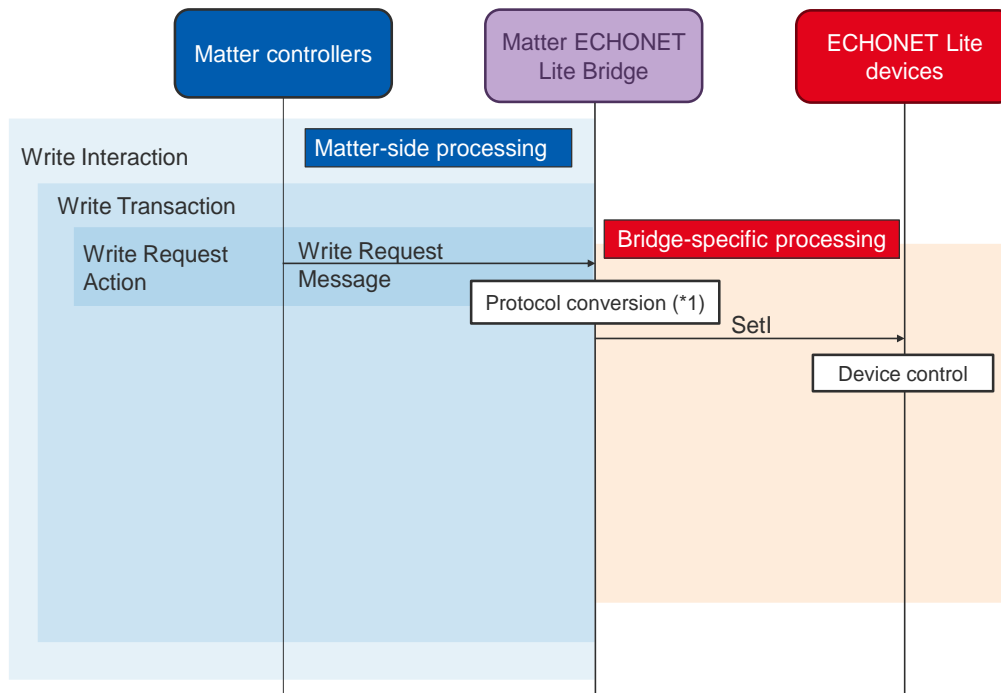
(\*1) For protocol conversions, refer to “5.2. General lighting class application example” and “5.3. Home Air Conditioner Class application examples” in this document.  
 (\*2) For details, refer to reference [1] “8.10. Status Codes”.  
 (\*3) SetC\_Res is basically an acceptance response, so this is an example of acquiring the status of an ECHONET Lite device through a Get request to confirm the process result of the SetC request.

Figure 4-5 Example of a request for SetC via Write Interaction

(2) Controlling devices using SetI

Reference [1] “8.7 Write Interaction” states that if the SuppressResponse flag of the Write Request Message is True, a Write Response is not required. Therefore, a sequence like the one shown in Figure 4-6 is possible.

For details, refer to reference [1] “8. Interaction Model Specification”.



(\*1) For protocol conversions, refer to “5.2. General lighting class application example” and “5.3. Home Air Conditioner Class application examples” in this document.

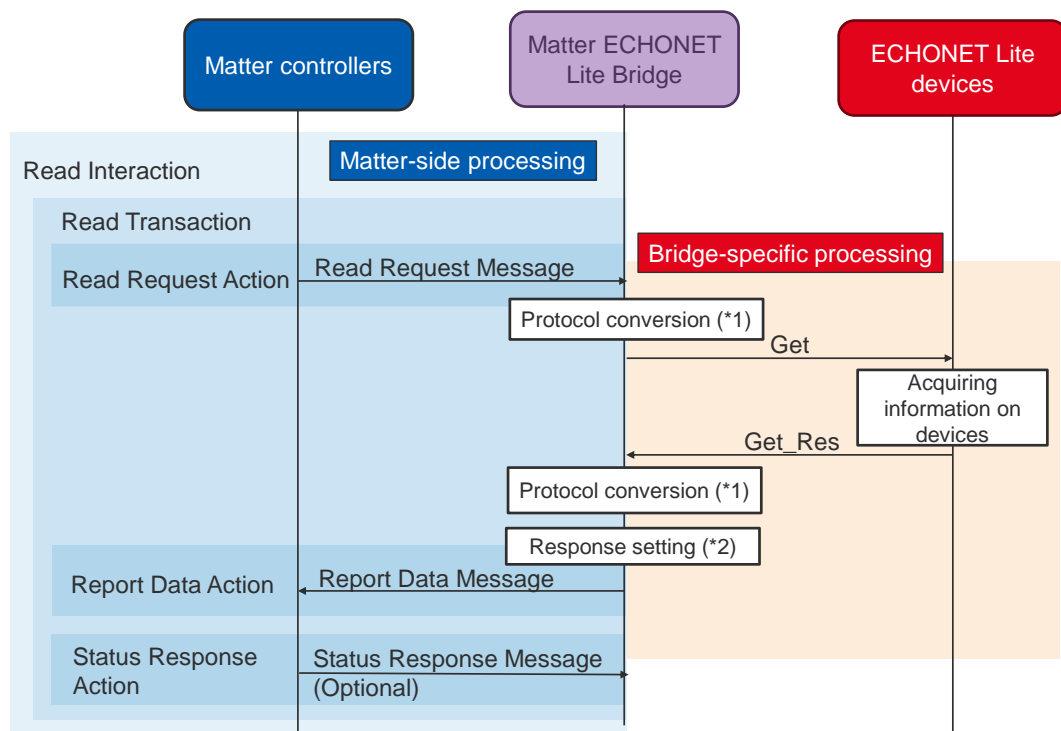
Figure 4-6 Example of a SetI request via Write Interaction

### 4.3.2. Reading values from Bridged devices

The Bridge configuration application and the Matter controller require reading of the current setting values of the Bridged devices, when updating the status display or checking the operation status of devices that currently exist in the Matter Fabric.

A request for control to be sent from the Matter controller to the Matter ECHONET Lite Bridge will read Attributes and Events via Read Interaction.

Figure 4-7 shows an example of a sequence in which the Matter ECHONET Lite Bridge reads Bridged device status using Get.



(\*1) For protocol conversions, refer to “5.2. General lighting class application example” and “5.3. Home Air Conditioner Class application examples” in this document.

(\*2) For details, refer to reference [1] “8.10. Status Codes”.

Figure 4-7 Example of a Get request via Read Interaction

## Chapter 5 Application examples

### 5.1. Overview

Matter ECHONET Lite Bridge must convert a request for control from the Matter to ECHONET Lite control requests.

This chapter describes application examples for mandatory properties in each ECHONET Lite class and mandatory Attribute in each DeviceType, for the general lighting class and the home air conditioner class.

Note that functions considered optional are not included in the scope of this study.

### 5.2. General lighting class application example

The Matter provides several DeviceTypes related to lighting, and when implementing any function in the general lighting class, it is necessary to select a DeviceType equipped with the appropriate function for the purpose.

For details on Matter’s DeviceType for lighting, refer to Reference [2] “Chapter 4. Lighting Device Types”.

This section assumes lighting with a dimming function. Also, Table 5-1 shows an example of the correspondence between Clusters and ECHONET Lite properties when considering the bridge function of DeviceType “Dimmable Light”.

Table 5-1 Application examples for lighting devices

Matter Dimmable Light Device Type (0x0101)						ECHONET Lite General lighting class (0x0290)			
Cluster		Attribute		Value		Property		Property setting values	
OnOff	0x0006	OnOff	0x0000	true	ON	Operation status	0x80	ON	0x30
				false	OFF			OFF	0x31
Not applicable						Lighting mode setting	0xB6	Automatic	0x41
								main lighting	0x42
								night lighting	0x43
								color lighting	0x45
Level Control	0x0008	Level	0x0000	INT8 U	0-254	Illuminance level	0xB0	illuminance level in %	0x00-0x64

As shown in the table above, there is no function directly applicable for lighting mode settings as of Matter Version 1.2. In order to set the lighting mode from the Matter controller, for example, an implementation such as linking a certain Level value with a lighting mode through the Matter ECHONET Lite Bridge using Level Control Cluster is required.

### 5.3. Home Air Conditioner Class application examples

In Matter, the DeviceType corresponding to the home air conditioner class is “Room Air Conditioner”.

For details, refer to reference [2] “13.3. Room Air Conditioner”.

Table 5-2 shows an example of correspondences between Cluster and ECHONET Lite properties when considering the bridge function of DeviceType “Room Air Conditioner” function.

Table 5-2 Application Examples of Air Conditioners

Matter Room Air Conditioner Device Type (0x0301)					ECHONET Lite Home air conditioner class (0x0130)								
Cluster		Attribute		Value		Property		Property Setting value					
OnOff	0x0006	OnOff	0x0000	true	ON	Operation status	0x80	ON	0x30				
				false	OFF			OFF	0x31				
Thermostat	0x0201	System mode	0x001c	Auto	1	Operation mode setting	0xB0	Automatic	0x41				
				Cool	3			Cooling	0x42				
				Heat	4			Heating	0x43				
				Dry	8			Dehumidification	0x44				
				Fan Only	7			Air circulation	0x45				
				Emergency Heat	5			Not applicable					
				Precooling	6			Not applicable					
				Off	0			Not applicable					
				Not applicable				Not applicable		Set temperature value	0xB3	0-50°C	0x00-0x32
				Occupied Cooling Setpoint				0x0011	Cooling mode temperature setpoint when someone is in the room [0.01°C]	MinCool SetpointLimit-MaxCool SetpointLimit	Set temperature value in cooling mode	0xB5	0-50°C
Occupied Heating Setpoint		0x0012	Heating mode temperature setpoint when someone is in the room [0.01°C]	MinHeat SetpointLimit-MaxHeat SetpointLimit	Set temperature value in heating mode	0xB6	0-50°C	0x00-0x32					



		Local Temperature	0x0000	Detected temperature [0.01°C]	-27315-32767	Measured value of room temperature	0xB B	-127-125°C	0x81-0x7D
Fan Control	0x0202	Fan Mode	0x0000	Auto	5	Air flow rate setting	0xA0	Automatic air flow rate control function used	0x41
				Low	1			Air flow rate	0x31-0x38
				Medium	2				
				High	3				
				On	4				
				Off	0				
Smart	6	Not applicable							
Not applicable						Power-saving operation setting	0x8F	Operating in power-saving mode	0x41
								Operating in normal mode	0x42

As shown in the table above, since DeviceType “Room Air Conditioner” is designed based on the Thermostat mechanism, there are some items that are not compatible with ECHONET Lite home air conditioners. Details are provided below.

(1) Set temperature value

Table 5-2 Table 5-2 shows that there is no item in Matter that corresponds to the ECHONET Lite temperature set value.

However, since both Matter and ECHONET Lite support functions to set temperature for cooling and heating modes optionally, "Set temperature value" can be substituted with those functions.

Note that in cases where the optional properties for "Set temperature value in cooling mode" and "Set temperature value in heating mode" for ECHONET Lite devices are not implemented, the mandatory property "Set temperature value" may be used.

(2) Air flow rate setting

Since Matter Version 1.2 states that Fan Control Cluster, which corresponds to "Air flow rate setting", is supported provisionally at the beginning, note that Fan Control Cluster is likely to be updated in the future.

Table 5-2 shows that there are at most three Matter setting values for air flow levels supported by Matter Version 1.2 (High, Midium, and Low). However, using the PercentSetting Attribute defined in the Fan Control Cluster, it is considered possible to support Levels 1 to 8, which are already defined by ECHONET Lite.

For On and Off functions, there are no corresponding ECHONET Lite items. However, since these items are mandatory for Matter, it is necessary to define these actions when they are set for bridge products.

Note that “Smart” is a set value to perform special actions rather than one that indicates a specific airflow, and its items are not found in ECHONET Lite. Therefore, “Smart” should be considered implementation-dependent. For details, refer to reference [3] “4.4. Fan Control Cluster”.

(3) Power-saving operation setting

This item cannot be set since it does not exist in Matter.

## Chapter 6 Conclusion

By implementing the Matter ECHONET Lite Bridge, general lighting and home air conditioners can now be operated in the same way as Matter devices. The convenience of ECHONET Lite devices for users is expected to improve through enabling multiple ECHONET Lite devices to be controlled by Matter controller application and voice control via smart speakers.

Considering the diffusion rate of smartphones and the standard installation of Matter controller apps in smartphones, the use of Matter ECHONET Lite Bridge will help to further spread the use of ECHONET Lite devices via the network, encouraging the development of smart home.

However, there are differences between the device object classes defined in the ECHONET Lite standards and the DeviceType items defined in the Matter standards. Because of this, and at this point, not all of the “eight priority devices” can be handled as bridged devices. The Matter standards are updated every six months and new DeviceType items will be added in the future. So it will be necessary to keep an eye on the latest trends in the Matter standards, while adding compatible devices to the Matter ECHONET Lite Bridge.

While the Matter standard supports device status acquisition on a regular basis through subscriptions, the ECHONET Lite standards have no such function. Therefore, the controller should acquire device status periodically. However, sequences related to subscriptions will be considered for addressing in the future if necessary.