



OPEN AQUARIUM PROJECT SPECIFICATION

Matter Mapping

PRE-DRAFT

Revision	0.3
Status	Pre-Draft. A candidate document submitted for consideration.
Date	2026-10-18
Identifier	matter-mapping
Authors	Trent Scholl, Project Lead, Open Aquarium Project
Editors	Trent Scholl
License	Community Specification License 1.0

Pre-Draft. This is a proposed draft. The Working Group has not approved it, and it will change. Comment by opening an issue with the `matter-mapping` label.

1 License

This Specification is subject to the Community Specification License 1.0 (the “Specification License”), available at <https://github.com/CommunitySpecification/1.0> and in `LICENSES/Community-Spec-1.0.md`. Contributions to this Specification are made under the Community Specification Contributor License Agreement 1.0.

The mapping file, test vectors, and example files published with this Specification are licensed under the Apache License 2.0 (`LICENSES/Apache-2.0.txt`).

The Scope that bounds the patent commitments made under the Specification License is set out in `Scope.md`. Exclusion notices, license acceptances, and withdrawals are recorded in `Notices.md`.

Notes:

1. Appendices are informative (non-normative) unless an appendix is explicitly marked normative.
2. Matter is a standard of the Connectivity Standards Alliance. This Specification cites it, and lists the identifiers and names of the Matter device types, clusters, attributes, and commands it uses; it copies no other part of it.

THESE MATERIALS ARE PROVIDED “AS IS.” The Contributors and Licensees expressly disclaim any warranties (express, implied, or otherwise), including implied warranties of merchantability, non-infringement, fitness for a particular purpose, or title, related to the materials. The entire risk as to implementing or otherwise using the materials is assumed by the implementer and user. IN NO EVENT WILL THE CONTRIBUTORS OR LICENSEES BE LIABLE TO ANY OTHER PARTY FOR LOST PROFITS OR ANY FORM OF INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS DELIVERABLE OR ITS GOVERNING AGREEMENT, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND WHETHER OR NOT THE OTHER MEMBER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

2 Compliance with Open Aquarium Tenets

2.1 Openness

The Specification is published under the Community Specification License 1.0, and its mapping file, test vectors, and examples under the Apache License 2.0. The mapping is a machine-readable file, `mapping.json`, from which the tables of Sections 7 to 14 are generated, and against which implementations can be tested. A bridge can be built with the open-source Matter SDK, which is licensed under the Apache License 2.0, without membership of any organization. The Matter specification is free of charge and needs no membership, but it is not openly licensed, which the Project Lead accepted for this Specification (Appendix A), and selling a product as certified for Matter needs paid membership and certification. This Specification requires neither, and defines no certification of its own: whether to certify a bridge is each maker’s choice.

2.2 Interoperability

Every conforming controller presents the same equipment in the same way: with the same Matter device types and clusters, the same values in the same units, the same names, and the same limits on what home apps may do. An owner who moves to a controller from another maker finds the same equipment in their home apps, behaving the same way. The Specification reuses Matter’s own device types wherever one fits, and invents none. It reads everything from the Device Classes, Automation, and Data Model specifications, so nothing is described twice.

2.3 Local Control

Matter works on the local network, and the bridge needs no internet connection, cloud account, or maker's service. Nothing in control depends on the bridge: the controller runs its automation whether the bridge works or not (Section 7.1). The owner sees and removes every home ecosystem that can reach the bridge, on the local network (Section 7.3). Home ecosystems may add their own cloud services; the bridge is off until the owner turns it on, after being told what such services can see and do (Section 7.2).

2.4 Safety

Home apps reach aquarium equipment only through the controller, as overrides that end by themselves, under the controller's interlocks and its emergency and power-loss modes (Section 12). The Specification limits what home apps may do to each kind of equipment, from the Fail-Safe Behavior recommendations: "turn everything off" never stops a return pump, unless its owner allows pauses of up to 15 minutes, and "turn everything on" never starts a heater, a doser, or an ozone generator (Section 12.5). A home app that stops equipment again and again, directly or through modes, scenes, and virtual outputs, cannot keep it stopped for longer than one stop allows, nor for more than half the time, unless stopping is that equipment's safe side (Sections 12.6 and 13.6). Modes, scenes, and virtual outputs are offered to home apps only when what they do to every output, through triggers and conditions too, stays within the same limits (Section 13.5). A mode that a person started is protected from home apps, as a person's override is (Section 12.4.4). A home app that stops a water change's pump aborts the water change, and only an Open Aquarium app resumes it; such a stop is never refused for the pump's minimum run time (Section 12.11). Home apps never change fail-safe settings, power-up behavior, or the owner's limits (Section 12.9). A sensor whose reading cannot be used shows no value rather than a wrong one (Section 14). Matter is never a path for a safety function [OAP-FS Section 13.4].

2.5 Sustainability

Owners can see and use their aquarium in the home apps they already have, without another hub or another app. Power and energy readings are shown where equipment measures them, so owners can see what their equipment uses.

Revision History

Revision	Date	Description	Authors
0.1	2026-10-15	Initial Pre-Draft.	Trent Scholl
0.2	2026-10-16	Modes and virtual outputs offered as switches only while what they do to every output stays within Table 10, so that a mode that feeds, starts a heater, or stops a return pump for an hour, and a virtual output that a return pump follows, are not offered; a home app's repeated stops of an output never end later than the first stop allowed, and the output then runs for a minimum time, a new column of Table 10, before the next; a home app's <code>On</code> never extends a mode, and starts one it started again only after that time; a home app's stop of a water change's pump aborts the water change, which only an Open Aquarium app resumes; commissioning windows opened through Matter's own mechanism by any fabric's administrator, and each fabric added or removed recorded in the audit log as <code>fabric-added</code> or <code>fabric-removed</code> and notified to every owner; a write to <code>NodeLabel</code> sets the name it is drawn from; <code>DefaultOpenDuration</code> kept by the bridge, never the output's duration; <code>BUSY</code> among the statuses; the running bits and <code>LocalTemperature</code> of a <code>device-setpoint</code> function defined directly; the bridge on another host provisional until a controller API exists; the claim about return pumps qualified; a test procedure, mapped to its requirements, for every requirement; references cited by tag; one sentence per line; the decided issues resolved, and every other issue given a status; the Project Lead's decisions of 16 October 2026 applied: minimum run times that follow each category's longest duration, at least an hour, lights included, and switches that do more to an output than its own policy allows offered only after the owner confirms.	Trent Scholl
0.3	2026-10-18	The Project Lead's decisions of 17 October 2026 applied: what an offered mode does through triggers and conditions counted, so that a mode is offered only while every output that depends on it is one that home apps may run, for no longer than its longest duration, and never through a latch; scenes offered only within Table 10, with the	Trent Scholl

Revision	Date	Description	Authors
		owner's confirmation; the bound on repeated stops held per output, across direct stops, modes, scenes, and virtual outputs, each of which causes a Matter stop of every output it stops or that depends on it; a mode that a person started protected from home apps as a person's override is; no alarms held by a maintenance mode that a home app started; a stop of an output that a water change is using never refused for its minimum run time, and no minimum run time for categories that stop on loss of contact and may be stopped for a day; Matter a pinned normative reference, which the approval checklist records; the policies of Table 10, the proposed role <code>override</code> , and the last value with <code>SensorFault</code> no longer provisional; no certification of its own; a bridge on another host that reaches the controller through the Controller API, and gives it the fabric of each operation; audit records of the bridge with <code>via matter</code> and the fabric in <code>details</code> ; <code>manage-users</code> for the bridge's commissioning window and fabrics. Also: the Power Source cluster for devices that can run on a battery, and <code>IdentifyType</code> from each endpoint's list of optional elements; what the Matter SDK's data model confirms of the details that wait on the Matter specification's text; "A pass criterion that checks what a SHOULD asks applies to a product that does it", and the test of floods of commands reworded to it; a test for each new requirement, and a check that every MUST has one; references to the final revisions of the other specifications, with the Flow Programs and Controller API specifications added; issues 6, 9, 11 to 13, and 16 to 20 deferred beyond revision 1.0.	

Contents

1	License	2
2	Compliance with Open Aquarium Tenets	2
2.1	Openness	2
2.2	Interoperability	2
2.3	Local Control	3
2.4	Safety	3
2.5	Sustainability	3
	Revision History	3
3	Scope	7
4	References	7
4.1	Normative References	7
4.2	Informative References	8
5	Terms, Definitions, and Conventions	8
5.1	Requirements Language	8
5.2	Terms and Definitions	8
5.3	Notation	9
6	Introduction	9
6.1	Purpose	9
6.2	Design Principles	9
6.3	Overview	10
6.4	Example	10
7	The Bridge	11
7.1	Where It Runs	11
7.2	Turning It On	11
7.3	Fabrics	11
7.4	Endpoints	11
7.5	Unique IDs	12
7.6	Reachability	12
7.7	Matter Releases	12
7.8	Device Types	13
8	What Is Exposed	13
8.1	Endpoints	13
8.2	The Owner's Choices	13
8.3	What Is Not Exposed	14
9	Classes	15
9.1	Switch (switch)	15
9.2	Level Output (level-output)	16
9.3	Pump (pump)	17
9.4	Light (light)	18
9.5	Heater (heater)	19
9.6	Chiller (chiller)	20
9.7	Doser (doser)	20
9.8	Valve (valve)	20
9.9	Feeder (feeder)	21
9.10	Auto Top-Off (auto-top-off)	21
9.11	Filter Roller (filter-roller)	21
9.12	Analyzer (analyzer)	21
9.13	Sensor (sensor)	21
9.14	Level Switch (level-switch)	23
9.15	Leak Sensor (leak-sensor)	23
9.16	Contact Input (contact-input)	23
9.17	Analog Input (analog-input)	24
10	Common Attributes	25
10.1	Device Attributes	25
10.2	Endpoint Attributes	26
10.3	Output Attributes	26
10.4	Faults	26

10.5	Supervised Inputs	27
11	Values from Automation	27
11.1	Known Setpoints	27
11.2	Thermostats	27
11.3	Pumps	27
11.4	Valves	28
11.5	On/Off and Level Control	28
11.6	Power-Up Behavior	28
12	Controls	28
12.1	Operations and Roles	28
12.2	Matter Commands	29
12.3	Effects	29
12.4	Policies	30
12.5	Policies by Equipment	31
12.6	Durations	32
12.7	Setpoint Changes	33
12.8	Deciding a Command	33
12.9	What Home Apps Never Change	34
12.10	Groups, Scenes, and Bulk Commands	34
12.11	Water Changes	34
13	Virtual Switches	35
13.1	Modes	35
13.2	Scenes	35
13.3	Virtual Outputs	35
13.4	Presentation	35
13.5	Staying within Table 10	36
13.6	Stops through Switches	37
14	Unit Conversions	37
14.1	Rules	37
14.2	Limits and Qualifiers	37
14.3	Conversions	37
15	Naming	38
16	Conformance	39
16.1	Bridges	39
16.2	Controllers	39
16.3	Machine-Readable Definitions and Tools	39
16.4	Test Procedures	40
17	Security Considerations	43
Appendix A — Approval Checklist		44
Appendix B — Example (Informative)		45
Appendix C — Home Apps (Informative)		46
Appendix D — Open Issues (Informative)		46

3 Scope

This Specification defines:

- how a controller, or a host that serves it, acts as a Matter bridge;
- which endpoints, modes, scenes, and virtual outputs the bridge presents, and as which Matter device types and clusters;
- how Device Classes attributes, and values from the controller's automation, become Matter attributes, with unit conversions;
- what home apps may do to each kind of equipment, and how Matter commands become the controller's operations;
- names and identifiers in Matter;
- what is not presented, with reasons; and
- test vectors and test procedures.

This Specification does not define:

- Matter itself, its security, commissioning, or certification, which the Matter specification and the Connectivity Standards Alliance define;
- how a controller uses Matter devices as its own inputs or outputs;
- integrations other than Matter, such as an MQTT gateway; or
- the user interface of controllers or home apps.

4 References

4.1 Normative References

- **[RFC 2119]** Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- **[RFC 8174]** Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017.
- **[RFC 8259]** Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, December 2017.
- **[RFC 3629]** Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.
- **[RFC 2104]** Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.
- **[FIPS 180-4]** National Institute of Standards and Technology, "Secure Hash Standard (SHS)", FIPS PUB 180-4, August 2015, <https://doi.org/10.6028/NIST.FIPS.180-4>.
- **[Matter]** Connectivity Standards Alliance, "Matter Specification", Version 1.6.1, September 2026; "Matter Application Cluster Specification", Version 1.6.1; and "Matter Device Library Specification", Version 1.6.1. Available free of charge, without membership, from <https://csa-iot.org/developer-resource/specifications-download-request/>. Not openly licensed, which the Project Lead accepted for this Specification (Appendix A; Appendix D, issue 1).
- **[OAP-DM]** Open Aquarium Project, "Open Aquarium Data Model", Revision 0.4, [specs/data-model/spec.md](#).
- **[OAP-DC]** Open Aquarium Project, "Open Aquarium Device Classes", Revision 0.5, [specs/device-classes/spec.md](#).
- **[OAP-FS]** Open Aquarium Project, "Open Aquarium Fail-Safe Behavior", Revision 0.4, [specs/fail-safe-behavior/spec.md](#).
- **[OAP-AU]** Open Aquarium Project, "Open Aquarium Automation", Revision 0.3, [specs/automation/spec.md](#).
- **[OAP-SA]** Open Aquarium Project, "Open Aquarium Security and Access", Revision 0.3, [specs/security-and-access/spec.md](#).
- **[OAP-DP]** Open Aquarium Project, "Open Aquarium Device Protocol", Revision 0.3, [specs/device-protocol/spec.md](#).
- **[OAP-CA]** Open Aquarium Project, "Open Aquarium Controller API", Revision 0.1, [specs/controller-api/spec.md](#).

- **[OAP-LP]** Open Aquarium Project, “Open Aquarium Lighting Programs”, Revision 0.3, [specs/lighting-programs/spec.md](#).
- **[OAP-FP]** Open Aquarium Project, “Open Aquarium Flow Programs”, Revision 0.3, [specs/flow-programs/spec.md](#).
- **[OAP-AN]** Open Aquarium Project, “Open Aquarium Alarms and Notifications”, Revision 0.3, [specs/alarms-and-notifications/spec.md](#).
- **[OAP-WL]** Open Aquarium Project, “Open Aquarium Water Level and Water Change”, Revision 0.3, [specs/water-level-and-water-change/spec.md](#).

4.2 Informative References

- **[P0001]** Open Aquarium Project, “Proposal 0001: Reference Controller Architecture”, [proposals/0001-reference-controller-architecture.md](#).
- **[SDK]** Project CHIP, “Matter (connectedhomeip)”, tag `v1.6.1.0`, Apache License 2.0, <https://github.com/project-chip/connectedhomeip>. Its `data_model` directory holds the Matter data model in machine-readable form, under the Connectivity Standards Alliance’s license, which allows only internal use.
- **[HA]** Home Assistant, “Matter integration”, <https://github.com/home-assistant/core/tree/dev/homeassistant/components/matter>.

5 Terms, Definitions, and Conventions

5.1 Requirements Language

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “NOT RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in BCP 14 [RFC 2119] [RFC 8174] when, and only when, they appear in all capitals, as shown here.

5.2 Terms and Definitions

Terms defined in the Device Classes specification [OAP-DC], such as device, endpoint, class, feature, and controller, in the Fail-Safe Behavior specification [OAP-FS], such as fail-safe behavior and inactive state, in the Automation specification [OAP-AU], such as output, input, virtual output, mode, override, interlock, and configuration, and in the Security and Access specification [OAP-SA], such as role, operation, user, and service, have the same meaning here.

Matter: The smart-home standard of the Connectivity Standards Alliance [Matter].

Node: A device or program that takes part in Matter, as Matter defines it.

Matter endpoint: One of the numbered functions of a node, as Matter defines it. “Endpoint” alone means a Device Classes endpoint.

Device type, cluster: As Matter defines them: a device type says which clusters a Matter endpoint has, and a cluster is a group of attributes, commands, and events.

Fabric: As Matter defines it: the nodes that one administrator has commissioned, with their credentials and access control.

Ecosystem: The products and services of one company or project that manage a fabric, such as a home hub, its app, and its cloud service.

Home app: An app, voice assistant, or hub of an ecosystem that people use to see and control their home.

Bridge: A node that presents equipment that is not itself on Matter, here the equipment of an Open Aquarium controller.

Bridged endpoint: A Matter endpoint of the bridge that presents one endpoint, mode, scene, or virtual output.

Exposed: Presented by the bridge as a bridged endpoint.

Policy: What home apps may do to an exposed output: `none`, `stop`, or `full` (Section 12.4).

Effect: What a Matter command or attribute write asks for, such as `stop` or `level` (Section 12.3).

Matter override: An override that the bridge set.

Duration: How long a Matter override lasts when the command gives no time.

Longest duration: The longest a Matter override of an output can last (`maxDuration`).

Matter stop: A stop of an output that a home app causes, directly or through a mode, a scene, or a virtual output (Sections 12.6.7 and 13.6).

Minimum run time: How long an output goes without a Matter stop, after one ends, before a home app may stop it again (`minRun`; Section 12.6).

Bridge key: A secret that the bridge uses to make Unique IDs (Section 7.5).

Virtual switch: A bridged endpoint that presents a mode, a scene, or a virtual output as an On/Off Plug-in Unit (Section 13).

Depends on: An output depends on a mode or a virtual output when its demand can change with the mode's or the virtual output's state (Section 13.5.2).

Known setpoint: A setpoint that the controller knows for a heater or chiller, which makes it a Thermostat (Section 11.1).

5.3 Notation

Matter identifiers are written in hexadecimal, as `0x0006`, and Matter names in `monospace`, as the `OnOff` attribute of the On/Off cluster. Device Classes, Automation, and Security and Access names are also in `monospace`. Units are Data Model unit codes [OAP-DM], except where a table gives Matter's own. References to other Open Aquarium specifications name them, as in "Fail-Safe Behavior Section 13.1". A section number with a third part refers to a numbered item of that section.

6 Introduction

6.1 Purpose

Owners want to see their aquarium where they see the rest of their home: the water temperature on a phone's home screen, a leak under the stand as an alert from a home hub, the tank lights switched off with the room's. Matter is the standard that the large home ecosystems share, and Proposal 0001 makes it the controller's bridge to them, while aquarium functions stay in the Device Protocol [P0001].

Without a common mapping, each controller would present the same heater differently: as a plug in one, a thermostat in another, and not at all in a third. Worse, a home app treats every plug alike. "Turn everything off" would stop a return pump as readily as a lamp, and a routine that switches a "plug" on each morning could switch on a heater that the controller is holding off.

This Specification defines one mapping for every controller, and the limits that keep home apps from harming livestock.

6.2 Design Principles

1. **Through the controller.** Home apps act only through the controller's operations, as overrides that end by themselves, below interlocks and the emergency and power-loss modes. The bridge never commands equipment itself.
2. **Safe whatever the command.** Every command is judged on what it would do to the equipment, wherever it came from: a person, a routine, a group, or a scene.
3. **Matter's meaning, or nothing.** Equipment is presented as the Matter device type that describes it. Where none does, it is not presented, rather than presented as something it is not.
4. **One way for everyone.** The device types, names, units, identifiers, and limits follow from the controller's configuration by fixed rules, so every controller presents the same system the same way.
5. **The owner decides.** The bridge is off until the owner turns it on, and the owner chooses what it presents and, within limits, what home apps may do.

6.3 Overview

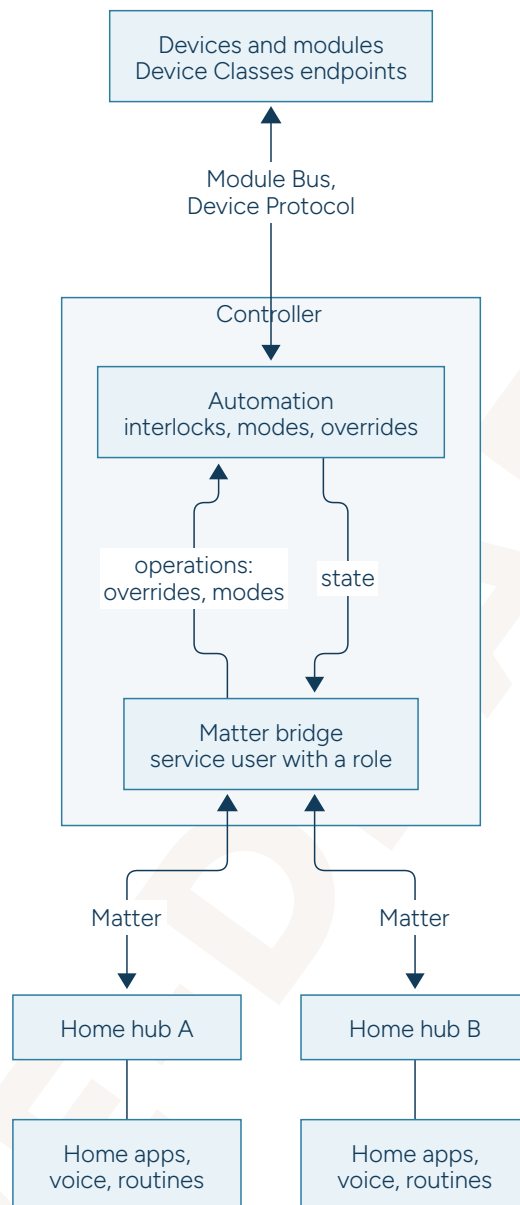


Figure 1 — The bridge and the controller

The bridge reads the controller’s configuration and the state of its endpoints, and presents each endpoint it can map as one bridged endpoint, under an Aggregator (Section 7.4). The class of the endpoint, its features, and the equipment it serves decide the device types (Section 9). Matter commands become effects, such as `stop` or `level` (Section 12.3); the output’s policy decides which effects are allowed (Section 12.4); and allowed effects become the controller’s operations, chiefly overrides with an end (Section 12.6).

6.4 Example

A heater on an outlet, driven by a setpoint function at 25.5 °C with a 1 °C dead band, appears as a heating Thermostat at 25.0 °C, showing the higher of two tank probes. Its policy is `stop`: a home app can switch it off for up to an hour, and “turn on” returns it to the controller’s thermostat without forcing it on. After a home app’s stop ends, the heater follows its thermostat for at least an hour before a home app can stop it again. The return pump appears as a Pump whose policy is `none`: home apps see that it runs, at what speed, and how much it draws, but cannot stop it. The pH probe does not appear, because Matter has no cluster for pH. Appendix B shows a whole system.

7 The Bridge

7.1 Where It Runs

1. A bridge MAY run on the controller, or on another host that serves it, such as the optional host of Proposal 0001. Which the reference controller uses is the reference design's choice, not this Specification's [P0001, Open Question 10].
2. A bridge on another host MUST reach the controller through the Controller API [OAP-CA], in a session as a party of kind `service` [OAP-DP Section 15.2], and MUST read state and carry out operations only through the controller. It MUST NOT connect to devices to command them.
3. A bridge MUST NOT be the controller of any device, and MUST NOT send commands to endpoints. The controller carries out the operations the bridge asks for, as Section 12 defines.
4. Losing the bridge, or its host, MUST NOT affect control. A bridge is never part of a safety function [P0001, design principle 8].

7.2 Turning It On

1. A bridge MUST be off until an owner turns it on, on the local network. Turning it on needs the `manage-users` and `manage-consent` operations [OAP-SA Table 6], because it gives access to a service, and can send the owner's data off the owner's network.
2. Before the bridge is first turned on, the controller MUST tell the owner that a home ecosystem may send what the bridge presents to its own services, and that those services may then control the presented equipment remotely, within the bridge's role and the policies of Section 12.
3. Turning the bridge on creates a user of kind `service` for it [OAP-SA Section 13.4], with the role the owner chooses. The role MUST NOT be `owner`. The owner chooses it, and the controller SHOULD propose `override`, with which home apps can do what Table 10 allows but cannot change the configuration (Appendix D, issue 3).
4. The bridge MUST open a commissioning window of its own only when an owner asks for one in an Open Aquarium app, which needs the `manage-users` operation [OAP-SA Section 13.4.4], or on physical presence at the controller [OAP-SA Section 9.10]. It MUST use a passcode chosen at random for each window, valid under Matter's rules, and shown only to the person who asked for it. It MUST NOT use a passcode that is printed on the unit or shared with other units. The administrator of a fabric the bridge already belongs to MAY also open a window, through Matter's own mechanism (Section 7.3.3).
5. An owner MUST be able to turn the bridge off on the local network at any time. Within 5 s (provisional), the bridge MUST then stop answering every fabric.

7.3 Fabrics

1. A bridge MUST support as many fabrics as Matter requires of a node, and every fabric has the same access: that of the bridge's role. A bridge MUST NOT let any fabric do more than the role allows, whatever privilege Matter's access control gives the fabric [OAP-SA Section 13.4.4].
2. The controller MUST show owners every fabric the bridge belongs to, with its label, vendor, and when it was added, and MUST let an owner remove any of them, on the local network, with the `manage-users` operation [OAP-SA Section 13.4.4]. A removed fabric MUST lose access within 5 s (provisional).
3. Matter lets the administrator of any fabric open a commissioning window for another ecosystem, and the bridge allows it (Section 7.2.4). The controller MUST record each fabric added to or removed from the bridge in its audit log, however it was added or removed, with the action `fabric-added` or `fabric-removed` [OAP-SA Table 8], `via_matter`, and the fabric's index, label, and vendor ID in `details`, as `fabricIndex`, `label`, and `vendorId` [OAP-SA Section 12.1.6]. It MUST also notify every owner of each, with the notices `fabric-added` and `fabric-removed`, whatever the owners' rules say [OAP-AN Section 13.4.4].

7.4 Endpoints

1. Matter endpoint 0 is the bridge's Root Node, and Matter endpoint 1 its Aggregator.

- Each exposed endpoint, mode, scene, or virtual output (Section 8) is one bridged endpoint, with the Bridged Node device type, the device types of Section 9 or 13, and the Bridged Device Basic Information and Descriptor clusters. The Aggregator's `PartsList` lists every bridged endpoint.
- Each bridged endpoint has the clusters that its device types make mandatory (Table 2), and each further cluster that an attribute of its class and declared features maps to (Sections 9 and 10).
- When the bridge is turned on, it numbers bridged endpoints from 2 in this order: outputs in the order of the configuration, then endpoint inputs in the order of the configuration, then virtual switches in the order the owner chose them.
- After that, the bridge MUST keep each bridged endpoint's number, across restarts and changes to the configuration, for as long as the same Unique ID reference (Table 1) is exposed. It gives the next unused number to anything newly exposed, and MUST NOT give a number to a different reference while unused numbers remain (provisional; Appendix D, issue 14).
- When what is exposed, or an endpoint's device types, change, the bridge MUST add and remove bridged endpoints while it runs, as Matter allows bridges to do.

Keeping numbers keeps the automations owners build in home apps working.

7.5 Unique IDs

- When it is first turned on, a bridge MUST make a bridge key of 32 bytes chosen at random, and keep it in storage that survives power loss. It MUST NOT send the key anywhere, and MUST make a new one after a factory reset.
- The `UniqueID` of each bridged endpoint is the first 16 bytes of HMAC-SHA-256 [RFC 2104] [FIPS 180-4], with the bridge key as the key and the UTF-8 [RFC 3629] encoding of its reference (Table 1) as the message, written as 32 lowercase hexadecimal digits.

Table 1 — Unique ID references

Bridged endpoint	Reference	Example
An endpoint	The device, as in Automation Section 8.5.3, a slash, and the endpoint's <code>id</code>	<code>module-bus:4f41503a000000010000a1b2c3d4e5f6/temp_pt1000</code>
A mode	<code>mode:</code> and the mode's <code>id</code>	<code>mode:feed-a</code>
A scene	<code>scene:</code> , the lighting function's <code>id</code> , a slash, and the scene's <code>id</code>	<code>scene:reef-lights/photo</code>
A virtual output	<code>output:</code> and the output's <code>id</code>	<code>output:night</code>

For example, with the bridge key of the test vectors, the first reference of Table 1 gives `96a6930f5ffebaa44324c6d452e3a830` (`test-vectors.json`). The same equipment keeps its Unique ID while the bridge keeps its key, and Unique IDs from two bridges cannot be linked, nor traced to serial numbers.

7.6 Reachability

- A bridged endpoint's `Reachable` MUST be `false` while the controller treats the endpoint's device as lost [OAP-FS Section 7.6], while an adapter reports the device unreachable [OAP-DP Section 16.3], and while a bridge on another host has lost the controller; and `true` otherwise. The bridge sends `ReachableChanged` when it changes.
- A virtual switch is reachable while the bridge reaches the controller.

7.7 Matter Releases

- This revision is written against Matter 1.6.1 [Matter]. A bridge SHOULD implement Matter 1.6.1 or a later release, and MUST implement a release that defines every device type it presents.
- The identifiers and names of `mapping.json` MUST match those of the release named in it; the tool of Section 16.3 checks them.

7.8 Device Types

Table 2 gives the device types that this Specification uses, with the clusters each needs and the features it requires of them. Every bridged endpoint also has the Bridged Node device type (0x0013), with the Bridged Device Basic Information cluster (0x0039), and the Descriptor cluster (0x001D). A bridged endpoint for an endpoint of a device that can run on a battery also has the Power Source cluster (0x002F) of the Bridged Node device type (Section 10.1).

Table 2 — Device types and their clusters

Device type	Clusters, besides Descriptor and those of Bridged Node	Features
On/Off Plug-in Unit (0x010A)	Identify (0x0003), Groups (0x0004), On/Off (0x0006), Scenes Management (0x0062)	On/Off: LT
Dimmable Plug-In Unit (0x010B)	Identify (0x0003), Groups (0x0004), On/Off (0x0006), Level Control (0x0008), Scenes Management (0x0062)	On/Off: LT; Level Control: 00, LT
On/Off Light (0x0100)	Identify (0x0003), Groups (0x0004), On/Off (0x0006), Scenes Management (0x0062)	On/Off: LT
Dimmable Light (0x0101)	Identify (0x0003), Groups (0x0004), On/Off (0x0006), Level Control (0x0008), Scenes Management (0x0062)	On/Off: LT; Level Control: 00, LT
Pump (0x0303)	Identify (0x0003), On/Off (0x0006), Level Control (0x0008), Pump Configuration and Control (0x0200)	Pump Configuration and Control: SPD
Thermostat (0x0301)	Identify (0x0003), Groups (0x0004), Thermostat (0x0201)	Thermostat: HEAT or COOL
Water Valve (0x0042)	Identify (0x0003), Valve Configuration and Control (0x0081)	—
Temperature Sensor (0x0302)	Identify (0x0003), Temperature Measurement (0x0402)	—
Humidity Sensor (0x0307)	Identify (0x0003), Relative Humidity Measurement (0x0405)	—
Flow Sensor (0x0306)	Identify (0x0003), Flow Measurement (0x0404)	—
Water Leak Detector (0x0043)	Identify (0x0003), Boolean State (0x0045)	Boolean State: CHGEVENT
Contact Sensor (0x0015)	Identify (0x0003), Boolean State (0x0045)	—
Generic Switch (0x000F)	Identify (0x0003), Switch (0x003B)	Switch: MS, MSR
Electrical Sensor (0x0510)	Power Topology (0x009C), Electrical Power Measurement (0x0090)	Power Topology: TREE; Electrical Power Measurement: ALTC or DIRC

8 What Is Exposed

8.1 Endpoints

1. An endpoint is exposed when all of these hold:
 - an output of the controller's configuration, other than a virtual output, is bound to it, or an endpoint input is [OAP-AU Sections 8.5 and 9.2];
 - Section 9 maps it; and
 - the owner has not hidden it.
2. An endpoint that several inputs use is exposed once, for the first of them in the configuration.
3. Endpoints that no output or input uses are not exposed, because a home app could reach them only by going around the controller's automation.
4. Modes, scenes, and virtual outputs are exposed only as Section 13 allows: as virtual switches that the owner chooses, and only while what they do stays within Table 10 (Section 13.5).

8.2 The Owner's Choices

1. An owner, or a user with the `full-control` role, MAY:
 - hide any exposed endpoint or virtual switch;

- choose an output's policy among those Table 10 allows for it, and its duration, from 60 s to its longest duration;
 - allow setpoint changes for a Thermostat (Section 12.7); and
 - offer modes, scenes, and virtual outputs as virtual switches, where Section 13 allows them.
2. Unless the owner chooses otherwise, everything Section 8.1 allows is exposed, with the policies and durations of Table 10, no setpoint changes, and no virtual switches.
 3. The controller MUST show the owner, for each exposed output, its policy and durations, and SHOULD say when they differ from Table 10.
 4. Choices are part of the controller's configuration of the bridge, and changing them needs the `change-configuration` operation [OAP-SA Table 6].

8.3 What Is Not Exposed

Table 3 — What is not exposed

What	Why
Modes of kind <code>emergency</code>	Only the owner ends an emergency mode, after checking the system (Automation Section 14.7); a home app's "turn everything off" would end it.
Modes of kind <code>power-loss</code>	Owners cannot start or end it (Automation Section 14.6).
Modes of kind <code>water-change</code>	A water change drains the tank and needs new water ready; a home app's "turn everything on" must never start one.
<code>stop-all</code>	As a switch, "turn everything on" would stop every output for an hour (Security and Access Section 11.8). An owner who wants a home app to stop the system in an emergency starts an emergency mode by a condition, such as a leak sensor.
Modes, scenes, and virtual outputs that do more to an output than Table 10 lets home apps do, or for longer	A home app's "turn everything on" would start them, and so do what no home app may do to that output directly (Section 13.5).
Step programs	Starting one needs the <code>full-control</code> role (<code>start-program</code>), and a program changes setpoints for days.
Outputs not bound to an endpoint, other than virtual outputs	Nothing would carry out a home app's command.
Endpoints that no output or input of the configuration uses	A home app's command could reach the equipment only by bypassing the controller's automation, interlocks, and modes.
Doser endpoints (<code>doser</code>)	Matter has no device type for this equipment, and presenting it as another kind would mislead home apps and the automations owners build in them. Dosing also adds chemicals to the water, which needs the <code>full-control</code> role (<code>dose-now</code>), and a doser's container levels have no Matter device type to carry them.
Feeder endpoints (<code>feeder</code>)	Matter has no device type for this equipment, and presenting it as another kind would mislead home apps and the automations owners build in them. No home app starts a feeder, so a feed mode that feeds is not offered either (Section 13.5); one that only pauses the pumps can be.
Auto Top-Off endpoints (<code>auto-top-off</code>)	Matter has no device type for this equipment, and presenting it as another kind would mislead home apps and the automations owners build in them. Stopping top-off for maintenance belongs to the maintenance mode, which can be offered as a virtual switch while it stays within Table 10 (Section 13.5).
Filter Roller endpoints (<code>filter-roller</code>)	Matter has no device type for this equipment, and presenting it as another kind would mislead home apps and the automations owners build in them.

What	Why
Analyzer endpoints (analyzer)	Matter has no clusters for water chemistry. Its concentration clusters measure air.
sensor endpoints with parameter salinity, specific-gravity, conductivity, ph, orp, alkalinity, general-hardness, calcium, magnesium, potassium, strontium, iodine, iron, copper, nitrate, nitrite, ammonia, phosphate, silicate, carbon-dioxide, dissolved-oxygen, dissolved-oxygen-saturation, or total-dissolved-solids	Matter has no clusters for water chemistry. Its concentration clusters measure air.
Other sensor endpoints	No Matter attribute carries this value. Every other parameter: ppfd, which does not convert to lux without the light's spectrum; water-level and pump-speed; the totals of elements; and electrical quantities, whose Electrical Sensor device type Matter uses only alongside another.
Level Switch endpoints (level-switch)	Matter has no device type for this equipment, and presenting it as another kind would mislead home apps and the automations owners build in them. A Contact Sensor or Water Leak Detector would mislead: an automation that shuts a house's water valve on a leak would act on a normal sump level.
Other analog-input endpoints	No Matter attribute carries this value. A voltage, or a value of any other parameter, as for sensor.

Each class's table in Section 9 also lists the attributes, commands, and faults that are not shown, with their reasons.

9 Classes

For each Device Classes class, this section gives the device types an endpoint is presented as, and what becomes of each of its attributes, commands, events, and faults.

- Presented as.** The first row whose conditions hold applies. "Equipment" is the Data Model category of the output's equipment [OAP-AU Section 8.1], else the endpoint's `loadCategory`, else `other`. "Parameter" is the endpoint's `parameter`, else the input's. "Known setpoint" is defined in Section 11.1. A row with "Also" adds a device type on the same Matter endpoint when the endpoint declares the feature.
- Attributes.** Each attribute is shown as the named Matter attribute, with the conversion of Table 13, or is not exposed, for the reason given. An attribute that names device types is shown only on those.
- Commands.** No class command is sent by the bridge; home apps act only through Section 12.

9.1 Switch (switch)

Presented as

When	Device types
Equipment heater, with a known setpoint (Section 11.1)	Thermostat (0x0301), heating
Equipment chiller, with a known setpoint (Section 11.1)	Thermostat (0x0301), cooling
Equipment light or algae-scrubber	On/Off Light (0x0100)
Otherwise	On/Off Plug-in Unit (0x010A)
Also, with power-monitoring	Electrical Sensor (0x0510), on the same endpoint

Attributes

Attribute	Feature	In Matter
on	—	On On/Off Plug-in Unit and On/Off Light: On/Off OnOff (boolean); on Thermostat: Thermostat ThermostatRunningState (running-bit)
voltage	power-monitoring	Electrical Power Measurement Voltage (milli)
current	power-monitoring	Electrical Power Measurement ActiveCurrent (milli)
power	power-monitoring	Electrical Power Measurement ActivePower (milli)
powerFactor	power-monitoring	Electrical Power Measurement PowerFactor (power-factor)

Attribute	Feature	In Matter
energy	power-monitoring	Electrical Energy Measurement CumulativeEnergyImported (energy)
currentLimit	overload-protection	Not exposed. A detail of the hardware, which home apps do not show.
tripped	overload-protection	Not exposed. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says.
ratedCurrent	—	Not exposed. A detail of the hardware, which home apps do not show.
switchingDevice	—	Not exposed. A detail of the hardware, which home apps do not show.
unpoweredState	—	Not exposed. A detail of the hardware, which home apps do not show.
endurance	—	Not exposed. A detail of the hardware, which home apps do not show.
switchCount	—	Not exposed. No Matter attribute carries this value.
backupThermostat	backup-thermostat	Not exposed. A fail-safe setting. Only the owner, confirming it in an Open Aquarium app, changes it (Fail-Safe Behavior Section 13.1).
cutoffContact	backup-thermostat	Not exposed. A detail of the hardware, which home apps do not show.
programRunning	backup-thermostat	Not exposed. Programs are defined by Open Aquarium specifications, and Matter cannot carry them.

Commands

Command	Feature	From home apps
set	—	Reached through overrides (Section 12), not sent by the bridge itself.
pulse	—	Home apps make a timed run with OnWithTimedOff, which becomes an override with an end (Section 12.6).
resetEnergy	power-monitoring	Changes recorded history, which needs the full-control role.
resetTrip	overload-protection	A recovery step that needs a person to check the equipment first.
runProgram	backup-thermostat	Programs are defined by Open Aquarium specifications, and Matter cannot carry them.

Faults

Fault	In Matter
overcurrent	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says
overtemperature	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says
stuck-on	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says
load-disconnected	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says
hardware-failure	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says
cutoff	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says

9.2 Level Output (level-output)

Presented as

When	Device types
Equipment light or algae-scrubber	Dimmable Light (0x0101)
Equipment return-pump or circulation-pump	Pump (0x0303)
Otherwise	Dimmable Plug-In Unit (0x010B)

Attributes

Attribute	Feature	In Matter
level	—	On/Off OnOff (level-on); on Dimmable Plug-In Unit and Dimmable Light: Level Control CurrentLevel (level-254); on Pump: Level Control CurrentLevel (pump-level); on Pump: Pump Configuration and Control PumpStatus (running-bit)

Attribute	Feature	In Matter
signal	—	Not exposed. A detail of the hardware, which home apps do not show.
minLevel	—	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
maxLevel	—	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
minVoltage	analog-voltage	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
maxVoltage	analog-voltage	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
frequency	pwm-frequency	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).

Commands

Command	Feature	From home apps
set	—	Reached through overrides (Section 12), not sent by the bridge itself.

Faults

Fault	In Matter
overcurrent	PumpStatus DeviceFault on a Pump; otherwise not shown
hardware-failure	PumpStatus DeviceFault on a Pump; otherwise not shown

9.3 Pump (pump)

Presented as

When	Device types
Always	Pump (0x0303)
Also, with telemetry	Electrical Sensor (0x0510), on the same endpoint

Attributes

Attribute	Feature	In Matter
running	—	On/Off 0n0ff (boolean); Pump Configuration and Control PumpStatus (running-bit)
speed	—	Level Control CurrentLevel (pump-level)
minSpeed	—	Not exposed. A detail of the hardware, which home apps do not show.
speedLimitLow	—	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
speedLimitHigh	—	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
ratedFlowRate	—	Pump Configuration and Control MaxFlow (flow)
flowCurve	—	Not exposed. A detail of the hardware, which home apps do not show.
direction	reverse	Not exposed. No Matter attribute carries this value.
reversePause	reverse	Not exposed. A detail of the hardware, which home apps do not show.
program	wave	Not exposed. Programs are defined by Open Aquarium specifications, and Matter cannot carry them.
syncGroup	wave	Not exposed. Programs are defined by Open Aquarium specifications, and Matter cannot carry them.
programRunning	wave	Not exposed. Programs are defined by Open Aquarium specifications, and Matter cannot carry them.
rpm	telemetry	Pump Configuration and Control Speed (rpm)
power	telemetry	Electrical Power Measurement ActivePower (milli)
temperature	telemetry	Not exposed. The only Matter attribute that could carry it means something else, so home apps would show it wrongly. The motor's or driver's temperature; a Temperature Measurement cluster on a Pump would be taken for the water's.
flowRate	telemetry	Flow Measurement MeasuredValue (flow)

Attribute	Feature	In Matter
flowRateSource	telemetry	Not exposed. A detail of the hardware, which home apps do not show.

Commands

Command	Feature	From home apps
set	—	Reached through overrides (Section 12), not sent by the bridge itself.
runProgram	wave	Programs are defined by Open Aquarium specifications, and Matter cannot carry them.

Faults

Fault	In Matter
stall	PumpStatus DeviceFault; PumpBlocked event
dry-run	PumpStatus DeviceFault; DryRunning event
overtemperature	PumpStatus DeviceFault; MotorTemperatureHigh event
overcurrent	PumpStatus DeviceFault; GeneralFault event
undervoltage	PumpStatus SupplyFault; SupplyVoltageLow event
overvoltage	PumpStatus SupplyFault; SupplyVoltageHigh event
load-disconnected	PumpStatus DeviceFault; GeneralFault event
hardware-failure	PumpStatus DeviceFault; ElectronicFatalFailure event

9.4 Light (light)

Presented as

When	Device types
Always	On/Off Light (0x0100)
Also, with power-monitoring	Electrical Sensor (0x0510), on the same endpoint

Attributes

Attribute	Feature	In Matter
on	—	On/Off OnOff (boolean)
channels	—	Not exposed. Overrides and modes can only run or stop a light (Lighting Programs Section 7.6), so a light is an On/Off Light; its scenes can be offered as virtual switches (Section 13).
levels	—	Not exposed. Overrides and modes can only run or stop a light (Lighting Programs Section 7.6), so a light is an On/Off Light; its scenes can be offered as virtual switches (Section 13).
colorTemperature	color-temperature	Not exposed. Overrides and modes can only run or stop a light (Lighting Programs Section 7.6), so a light is an On/Off Light; its scenes can be offered as virtual switches (Section 13).
intensity	color-temperature	Not exposed. Overrides and modes can only run or stop a light (Lighting Programs Section 7.6), so a light is an On/Off Light; its scenes can be offered as virtual switches (Section 13).
minColorTemperature	color-temperature	Not exposed. Overrides and modes can only run or stop a light (Lighting Programs Section 7.6), so a light is an On/Off Light; its scenes can be offered as virtual switches (Section 13).
maxColorTemperature	color-temperature	Not exposed. Overrides and modes can only run or stop a light (Lighting Programs Section 7.6), so a light is an On/Off Light; its scenes can be offered as virtual switches (Section 13).
program	local-program	Not exposed. Programs are defined by Open Aquarium specifications, and Matter cannot carry them.
programRunning	local-program	Not exposed. Programs are defined by Open Aquarium specifications, and Matter cannot carry them.
programMaxLevel	local-program	Not exposed. Programs are defined by Open Aquarium specifications, and Matter cannot carry them.

Attribute	Feature	In Matter
temperature	fixture-temperature	Not exposed. The only Matter attribute that could carry it means something else, so home apps would show it wrongly. The fixture's temperature, which would be taken for the room's.
thermalDimming	fixture-temperature	Not exposed. No Matter attribute carries this value.
thermalScale	fixture-temperature	Not exposed. No Matter attribute carries this value.
power	power-monitoring	Electrical Power Measurement <code>ActivePower</code> (milli)

Commands

Command	Feature	From home apps
set	—	Reached through overrides (Section 12), not sent by the bridge itself.
setColorTemperature	color-temperature	Overrides and modes can only run or stop a light (Lighting Programs Section 7.6), so a light is an On/Off Light; its scenes can be offered as virtual switches (Section 13).
runProgram	local-program	Programs are defined by Open Aquarium specifications, and Matter cannot carry them.

Faults

Fault	In Matter
overtemperature	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says
hardware-failure	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says

9.5 Heater (heater)

Presented as

When	Device types
With a known setpoint (Section 11.1)	Thermostat (0x0301), heating
Otherwise	On/Off Plug-in Unit (0x010A)
Also, with power-monitoring	Electrical Sensor (0x0510), on the same endpoint

Attributes

Attribute	Feature	In Matter
enabled	—	On On/Off Plug-in Unit: On/Off 0n0ff (boolean). Otherwise not exposed. Decides how the endpoint is presented, or is converted into what is shown, rather than being shown itself. On a Thermostat, <code>SystemMode</code> comes from the automation (Section 11.2).
heating	—	On Thermostat: Thermostat <code>ThermostatRunningState</code> (running-bit). Otherwise not exposed. No Matter attribute carries this value.
setpoint	thermostat	On Thermostat: Thermostat <code>OccupiedHeatingSetpoint</code> (temperature). Otherwise not exposed. No Matter attribute carries this value.
hysteresis	thermostat	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
temperature	thermostat	On Thermostat: Thermostat <code>LocalTemperature</code> (temperature). Otherwise not exposed. No Matter attribute carries this value.
cutoffTemperature	thermostat	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
power	power-monitoring	Electrical Power Measurement <code>ActivePower</code> (milli)

Commands

Command	Feature	From home apps
set	—	Reached through overrides (Section 12), not sent by the bridge itself.

Faults

Fault	In Matter
overtemperature	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says
sensor-disconnected	LocalTemperature is null
stuck-on	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says
load-disconnected	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says
hardware-failure	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says

9.6 Chiller (chiller)

Presented as

When	Device types
With a known setpoint (Section 11.1)	Thermostat (0x0301), cooling
Otherwise	On/Off Plug-in Unit (0x010A)
Also, with power-monitoring	Electrical Sensor (0x0510), on the same endpoint

Attributes

Attribute	Feature	In Matter
enabled	—	On On/Off Plug-in Unit: On/Off 0n0ff (boolean). Otherwise not exposed. Decides how the endpoint is presented, or is converted into what is shown, rather than being shown itself. On a Thermostat, SystemMode comes from the automation (Section 11.2).
cooling	—	On Thermostat: Thermostat ThermostatRunningState (running-bit). Otherwise not exposed. No Matter attribute carries this value.
setpoint	thermostat	On Thermostat: Thermostat 0ccupiedCoolingSetpoint (temperature). Otherwise not exposed. No Matter attribute carries this value.
hysteresis	thermostat	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
temperature	thermostat	On Thermostat: Thermostat LocalTemperature (temperature). Otherwise not exposed. No Matter attribute carries this value.
power	power-monitoring	Electrical Power Measurement ActivePower (milli)

Commands

Command	Feature	From home apps
set	—	Reached through overrides (Section 12), not sent by the bridge itself.

Faults

Fault	In Matter
overtemperature	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says
sensor-disconnected	LocalTemperature is null
hardware-failure	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says

9.7 Doser (doser)

Not exposed. Matter has no device type for this equipment, and presenting it as another kind would mislead home apps and the automations owners build in them. Dosing also adds chemicals to the water, which needs the full-control role (dose-now), and a doser's container levels have no Matter device type to carry them.

9.8 Valve (valve)

Presented as

When	Device types
Always	Water Valve (0x0042)

Attributes

Attribute	Feature	In Matter
open	—	Valve Configuration and Control <code>currentState</code> , <code>targetState</code> (<code>valve-state</code>)
normallyOpen	—	Not exposed. A detail of the hardware, which home apps do not show.

Commands

Command	Feature	From home apps
set	—	Reached through overrides (Section 12), not sent by the bridge itself.

Faults

Fault	In Matter
stuck-on	ValveFault Blocked
load-disconnected	ValveFault NotConnected
hardware-failure	ValveFault GeneralFault

9.9 Feeder (feeder)

Not exposed. Matter has no device type for this equipment, and presenting it as another kind would mislead home apps and the automations owners build in them. No home app starts a feeder, so a feed mode that feeds is not offered either (Section 13.5); one that only pauses the pumps can be.

9.10 Auto Top-Off (auto-top-off)

Not exposed. Matter has no device type for this equipment, and presenting it as another kind would mislead home apps and the automations owners build in them. Stopping top-off for maintenance belongs to the maintenance mode, which can be offered as a virtual switch while it stays within Table 10 (Section 13.5).

9.11 Filter Roller (filter-roller)

Not exposed. Matter has no device type for this equipment, and presenting it as another kind would mislead home apps and the automations owners build in them.

9.12 Analyzer (analyzer)

Not exposed. Matter has no clusters for water chemistry. Its concentration clusters measure air.

9.13 Sensor (sensor)

Presented as

When	Device types
Parameter <code>temperature</code> or <code>air-temperature</code>	Temperature Sensor (0x0302)
Parameter <code>relative-humidity</code>	Humidity Sensor (0x0307)
Parameter <code>flow-rate</code>	Flow Sensor (0x0306)
Parameter <code>salinity</code> , <code>specific-gravity</code> , <code>conductivity</code> , <code>ph</code> , <code>orp</code> , <code>alkalinity</code> , <code>general-hardness</code> , <code>calcium</code> , <code>magnesium</code> , <code>potassium</code> , <code>strontium</code> , <code>iodine</code> , <code>iron</code> , <code>copper</code> , <code>nitrate</code> , <code>nitrite</code> , <code>ammonia</code> , <code>phosphate</code> , <code>silicate</code> , <code>carbon-dioxide</code> , <code>dissolved-oxygen</code> , <code>dissolved-oxygen-saturation</code> , or <code>total-dissolved-solids</code>	Not exposed. Matter has no clusters for water chemistry. Its concentration clusters measure air.
Otherwise	Not exposed. No Matter attribute carries this value. Every other parameter: <code>ppfd</code> , which does not convert to lux without the light's spectrum; <code>water-level</code> and <code>pump-speed</code> ; the totals of elements; and electrical quantities, whose Electrical Sensor device type Matter uses only alongside another.

Attributes

Attribute	Feature	In Matter
parameter	—	Not exposed. Decides how the endpoint is presented, or is converted into what is shown, rather than being shown itself.
unit	—	Not exposed. Decides how the endpoint is presented, or is converted into what is shown, rather than being shown itself.
value	—	On Temperature Sensor: Temperature Measurement <code>MeasuredValue (temperature)</code> ; on Humidity Sensor: Relative Humidity Measurement <code>MeasuredValue (humidity)</code> ; on Flow Sensor: Flow Measurement <code>MeasuredValue (flow)</code>
qualifier	—	Not exposed. Matter cannot mark a value as a limit. The bridge reports the limit itself (Section 14.2).
quality	—	On Temperature Sensor: Temperature Measurement <code>MeasuredValue (availability)</code> ; on Humidity Sensor: Relative Humidity Measurement <code>MeasuredValue (availability)</code> ; on Flow Sensor: Flow Measurement <code>MeasuredValue (availability)</code>
updated	—	Not exposed. Matter reports a value when it changes, without the time it was measured.
measurementInterval	—	Not exposed. No Matter attribute carries this value.
rangeMin	—	On Temperature Sensor: Temperature Measurement <code>MinMeasuredValue (range)</code> ; on Humidity Sensor: Relative Humidity Measurement <code>MinMeasuredValue (range)</code> ; on Flow Sensor: Flow Measurement <code>MinMeasuredValue (range)</code>
rangeMax	—	On Temperature Sensor: Temperature Measurement <code>MaxMeasuredValue (range)</code> ; on Humidity Sensor: Relative Humidity Measurement <code>MaxMeasuredValue (range)</code> ; on Flow Sensor: Flow Measurement <code>MaxMeasuredValue (range)</code>
resolution	—	Not exposed. A detail of the hardware, which home apps do not show.
port	—	Not exposed. Decides how the endpoint is presented, or is converted into what is shown, rather than being shown itself.
probeNominal	—	Not exposed. Calibration needs a person at the equipment and the <code>calibrate</code> operation.
tdsFactor	—	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
lastCalibrated	calibration	Not exposed. Calibration needs a person at the equipment and the <code>calibrate</code> operation.
calibrationDue	calibration	Not exposed. Calibration needs a person at the equipment and the <code>calibrate</code> operation.
lastCalibration	calibration	Not exposed. Calibration needs a person at the equipment and the <code>calibrate</code> operation.
compensationSource	temperature-compensation	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
compensationTemperature	temperature-compensation	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
compensationMethod	temperature-compensation	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
compensationCoefficient	temperature-compensation	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
compensationSalinity	temperature-compensation	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
probeSerialNumber	smart-probe	Not exposed. A detail of the hardware, which home apps do not show.
probeInstalled	smart-probe	Not exposed. A detail of the hardware, which home apps do not show.
probeEndOfLife	smart-probe	Not exposed. A detail of the hardware, which home apps do not show.
pulsesPerLitre	pulse	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).

Attribute	Feature	In Matter
active	selectable	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9). An endpoint that is not active reports <code>quality stale</code> , so its <code>MeasuredValue</code> is null.

Commands

Command	Feature	From home apps
<code>beginCalibration</code>	calibration	Calibration needs a person at the equipment and the <code>calibrate</code> operation.
<code>calibratePoint</code>	calibration	Calibration needs a person at the equipment and the <code>calibrate</code> operation.
<code>completeCalibration</code>	calibration	Calibration needs a person at the equipment and the <code>calibrate</code> operation.
<code>cancelCalibration</code>	calibration	Calibration needs a person at the equipment and the <code>calibrate</code> operation.
<code>resetCalibration</code>	calibration	Calibration needs a person at the equipment and the <code>calibrate</code> operation.

Faults

Fault	In Matter
sensor-disconnected	<code>MeasuredValue</code> is null
out-of-range	<code>MeasuredValue</code> is null, unless the sensor reports a limit with <code>qualifier</code> (Section 14.2)
calibration-due	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says
calibration-failed	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says
expired	Not shown. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says
hardware-failure	<code>MeasuredValue</code> is null

9.14 Level Switch (`level-switch`)

Not exposed. Matter has no device type for this equipment, and presenting it as another kind would mislead home apps and the automations owners build in them. A Contact Sensor or Water Leak Detector would mislead: an automation that shuts a house's water valve on a leak would act on a normal sump level.

9.15 Leak Sensor (`leak-sensor`)

Presented as

When	Device types
Always	Water Leak Detector (0x0043)

Attributes

Attribute	Feature	In Matter
leak	—	Boolean State <code>StateValue</code> (boolean)
waterType	water-type	Not exposed. No Matter attribute carries this value.

Faults

Fault	In Matter
wiring-fault	<code>SensorFault</code> <code>GeneralFault</code> , and a <code>SensorFault</code> event, with the <code>supervised</code> feature (Section 10.5)
sensor-disconnected	<code>SensorFault</code> <code>GeneralFault</code> , and a <code>SensorFault</code> event, with the <code>supervised</code> feature (Section 10.5)
hardware-failure	<code>SensorFault</code> <code>GeneralFault</code> , and a <code>SensorFault</code> event, with the <code>supervised</code> feature (Section 10.5)

9.16 Contact Input (`contact-input`)

Presented as

When	Device types
With <code>momentary</code>	Generic Switch (0x000F)
Otherwise	Contact Sensor (0x0015)

Attributes

Attribute	Feature	In Matter
closed	—	On Contact Sensor: Boolean State <code>StateValue (boolean)</code> ; on Generic Switch: Switch <code>CurrentPosition (contact-position)</code>
debounce	—	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
invert	—	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).

Events

Event	Feature	In Matter
pressed	momentary	On Generic Switch: Switch <code>InitialPress, ShortRelease (press-events)</code>

Faults

Fault	In Matter
wiring-fault	On a Contact Sensor with the <code>supervised</code> feature, <code>SensorFault GeneralFault</code> (Section 10.5); otherwise not shown
hardware-failure	On a Contact Sensor with the <code>supervised</code> feature, <code>SensorFault GeneralFault</code> (Section 10.5); otherwise not shown

9.17 Analog Input (analog-input)**Presented as**

When	Device types
Parameter <code>temperature</code> or <code>air-temperature</code> , with <code>scaling</code>	Temperature Sensor (0x0302)
Parameter <code>relative-humidity</code> , with <code>scaling</code>	Humidity Sensor (0x0307)
Parameter <code>flow-rate</code> , with <code>scaling</code>	Flow Sensor (0x0306)
Otherwise	Not exposed. No Matter attribute carries this value. A voltage, or a value of any other parameter, as for <code>sensor</code> .

Attributes

Attribute	Feature	In Matter
voltage	—	Not exposed. No Matter attribute carries this value.
maxVoltage	—	Not exposed. A detail of the hardware, which home apps do not show.
updated	—	Not exposed. Matter reports a value when it changes, without the time it was measured.
parameter	scaling	Not exposed. Decides how the endpoint is presented, or is converted into what is shown, rather than being shown itself.
unit	scaling	Not exposed. Decides how the endpoint is presented, or is converted into what is shown, rather than being shown itself.
scale	scaling	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
offset	scaling	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
value	scaling	On Temperature Sensor: Temperature Measurement <code>MeasuredValue (temperature)</code> ; on Humidity Sensor: Relative Humidity Measurement <code>MeasuredValue (humidity)</code> ; on Flow Sensor: Flow Measurement <code>MeasuredValue (flow)</code>
current	current-loop	Not exposed. No Matter attribute carries this value.
loopVoltage	current-loop	Not exposed. A detail of the hardware, which home apps do not show.

Faults

Fault	In Matter
out-of-range	<code>MeasuredValue</code> is null
hardware-failure	<code>MeasuredValue</code> is null
wiring-fault	<code>MeasuredValue</code> is null

10 Common Attributes

10.1 Device Attributes

The Bridged Device Basic Information cluster of each bridged endpoint describes the device of its endpoint. Its `Reachable` follows Section 7.6, and its `UniqueID` Section 7.5. The bridge leaves out `VendorID` and `ProductID`, which identify products certified for Matter (provisional; Appendix D, issue 15).

Table 4 — Device attributes

Attribute	In Matter
<code>manufacturer</code>	Bridged Device Basic Information <code>VendorName</code> (text-32)
<code>model</code>	Bridged Device Basic Information <code>ProductName</code> (text-32)
<code>serialNumber</code>	Bridged Device Basic Information <code>SerialNumber</code> (serial)
<code>hardwareRevision</code>	Bridged Device Basic Information <code>HardwareVersionString</code> (text-64)
<code>firmwareVersion</code>	Bridged Device Basic Information <code>SoftwareVersionString</code> (text-64)
<code>specRevisions</code>	Not exposed. Decides how the endpoint is presented, or is converted into what is shown, rather than being shown itself.
<code>name</code>	Bridged Device Basic Information <code>NodeLabel</code> (name)
<code>faults</code>	Power Source <code>BatChargeLevel</code> (battery-charge). Otherwise not exposed. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says. Only <code>battery-weak</code> is shown, on a device that can run on a battery (Section 10.1).
<code>uptime</code>	Not exposed. No Matter attribute carries this value.
<code>supplyVoltage</code>	Not exposed. No Matter attribute carries this value.
<code>internalTemperature</code>	Not exposed. The only Matter attribute that could carry it means something else, so home apps would show it wrongly.
<code>mainsPresent</code>	Not exposed. The Power Source cluster that the bridge presents describes a device's battery (Section 10.1), not its mains supply; owners hear that mains power is lost through Alarms and Notifications, as Section 10.4 says.
<code>powerSource</code>	Power Source <code>Status</code> (power-status)
<code>batteryLevel</code>	Power Source <code>BatPercentRemaining</code> (battery-percent)
<code>batteryVoltage</code>	Power Source <code>BatVoltage</code> (milli)
<code>clockUncertainty</code>	Not exposed. No Matter attribute carries this value.
<code>updateChannel</code>	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
<code>securityVersion</code>	Not exposed. No Matter attribute carries this value.
<code>firmwareSigner</code>	Not exposed. No Matter attribute carries this value.
<code>updateFeed</code>	Not exposed. No Matter attribute carries this value.
<code>updateKeys</code>	Not exposed. No Matter attribute carries this value.
<code>supportEnds</code>	Not exposed. No Matter attribute carries this value.
<code>automaticInstall</code>	Not exposed. A setting that changes how the equipment works. Home apps must not change it (Section 12.9).
<code>lastUpdate</code>	Not exposed. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says.

A device that can run on a battery gives `powerSource` [OAP-DC Section 8.1], and so lists it among the optional elements of its device description [OAP-DC Section 12]. Each bridged endpoint for an endpoint of such a device has the Power Source cluster, with the `BAT` feature and no other, which describes the device's battery:

1. `Status` is `Active` while `powerSource` is `battery`, and `Standby` while it is `external`.
2. `Order` is 0, `Description` is `Battery`, and `EndpointList` lists the bridged endpoint itself.
3. `BatPercentRemaining` shows `batteryLevel`, and `BatVoltage` shows `batteryVoltage`, converted as Table 13 says; each is null while the device does not give its value.
4. `BatChargeLevel` is `Warning` while the device reports `battery-weak` [OAP-DC Section 8.4], and `OK` otherwise.
5. `BatReplacementNeeded` is `false`, and `BatReplaceability` is `Unspecified`, because a device reports only that its battery is weak, not how it is replaced.

So a home app warns of a weak battery in a leak sensor as it does for its own sensors.

10.2 Endpoint Attributes

Table 5 — Endpoint attributes

Attribute	In Matter
id	Not exposed. Used to make the endpoint's Matter identifiers (Section 7.5), not shown.
class	Not exposed. Decides how the endpoint is presented, or is converted into what is shown, rather than being shown itself.
features	Not exposed. Decides how the endpoint is presented, or is converted into what is shown, rather than being shown itself.
name	Bridged Device Basic Information <code>NodeLabel</code> (name)
faults	Not exposed. Shown, where Matter can, as each class's table in Section 9 says.
sounding	Not exposed. Matter has no attribute for it on this device type; owners hear of it through Alarms and Notifications, as Section 10.4 says.

The endpoint's `identify` command is reached through the Identify cluster (Section 12.2). Each endpoint lists the optional elements it implements in its device description [OAP-DC Sections 7.2.4 and 12]:

1. A bridged endpoint's `IdentifyType` is `VisibleIndicator` when its endpoint lists `identify`, and `None` when it does not.
2. The bridge asks the controller to send `identify` only to an endpoint that lists it. For another, an `Identify` command succeeds and changes nothing.
3. A device of Device Classes revision 0.1 lists no optional elements [OAP-DC Section 7.2.5]. For its endpoints, the bridge reports `IdentifyType None`, still asks the controller to send `identify`, and ignores a refusal.

10.3 Output Attributes

Table 6 — Output attributes

Attribute	In Matter
controlSource	On Pump: Pump Configuration and Control <code>EffectiveOperationMode</code> , <code>PumpStatus</code> (local-mode). Otherwise not exposed. No Matter attribute carries this value.
latched	Not exposed. No Matter attribute carries this value.
failSafeAction	Not exposed. A fail-safe setting. Only the owner, confirming it in an Open Aquarium app, changes it (Fail-Safe Behavior Section 13.1).
failSafeState	Not exposed. A fail-safe setting. Only the owner, confirming it in an Open Aquarium app, changes it (Fail-Safe Behavior Section 13.1).
failSafeTimeout	Not exposed. A fail-safe setting. Only the owner, confirming it in an Open Aquarium app, changes it (Fail-Safe Behavior Section 13.1).
powerOnBehavior	On On/Off Plug-in Unit, Dimmable Plug-In Unit, On/Off Light and Dimmable Light: <code>On/Off StartUpOnOff</code> (start-up). Otherwise not exposed. A fail-safe setting. Only the owner, confirming it in an Open Aquarium app, changes it (Fail-Safe Behavior Section 13.1). Shown, never written (Section 12.9).
powerOnDelay	Not exposed. A fail-safe setting. Only the owner, confirming it in an Open Aquarium app, changes it (Fail-Safe Behavior Section 13.1).
loadCategory	Not exposed. Decides how the endpoint is presented, or is converted into what is shown, rather than being shown itself.
operatingHours	On Pump: Pump Configuration and Control <code>LifetimeRunningHours</code> (hours). Otherwise not exposed. No Matter attribute carries this value.

10.4 Faults

1. Matter has no general attribute for faults. A fault is shown only where Section 9 says, for example as a `PumpStatus` bit, and otherwise reaches owners through Alarms and Notifications [OAP-AN].
2. A bridge MUST NOT show a fault as a reading or a state, for example a broken sensor as a leak, or a lost device as an outlet that is off.

10.5 Supervised Inputs

1. A Water Leak Detector or Contact Sensor for an endpoint with the `supervised` feature has the Boolean State Configuration cluster, with the `FAULTEV` feature.
2. While the endpoint reports `wiring-fault`, `sensor-disconnected`, or `hardware-failure`, its `SensorFault` MUST have `GeneralFault` set, and the bridge sends a `SensorFault` event when it is set.
3. Meanwhile, `StateValue` keeps its last value, because Matter cannot show that it is unknown, and a leak shown for a fault would make home automations act on it, such as by closing a house's water valve (Appendix D, issue 4).

11 Values from Automation

11.1 Known Setpoints

An output has a known setpoint when:

1. its endpoint has the `thermostat` feature; or
2. it is a `raise` or `lower` output of a `device-setpoint` function, or of a `setpoint` function whose input is a temperature: an endpoint input of the parameter `temperature` or `air-temperature`, or an `average`, `minimum`, or `maximum` of such inputs [OAP-AU Sections 9.7, 11.4, and 11.7].

A heater or chiller with a known setpoint is a Thermostat (Section 9), including one on an outlet whose equipment is a heater or chiller.

11.2 Thermostats

A Thermostat has the Thermostat cluster with the `HEAT` feature for a heater, or the `COOL` feature for a chiller, and no other features.

1. `ControlSequenceOfOperation` is `HeatingOnly` for a heater, and `CoolingOnly` for a chiller.
2. `LocalTemperature` is the value of the function's input, converted as Table 13 says, or, for a `device-setpoint` function, which has no input, and without a function, the endpoint's `temperature`. It is null while that value is unavailable [OAP-FS Section 10.1].
3. `OccupiedHeatingSetpoint`, or `OccupiedCoolingSetpoint`, is the function's effective setpoint S [OAP-AU Section 12.1], or, without a function, the endpoint's `setpoint`. For a function with both `raise` and `lower` outputs and a dead band d , a heater shows $S - d/2$ and a chiller $S + d/2$, the values at which each stops [OAP-AU Sections 11.5 and 11.7].
4. `MinHeatSetpointLimit` and `MaxHeatSetpointLimit`, or the cooling pair, are the function's `limits`, shifted as in item 3. Without `limits`, they are left out.
5. `SystemMode` is `Off` while an override, a mode, an interlock, or `stop-all` holds the output stopped; and `Heat`, or `Cool`, otherwise.
6. `ThermostatRunningState` has `Heat`, or `Cool`, set while the equipment heats or cools: while the outlet that drives it is on, while a heater reports `heating true`, or while a chiller reports `cooling true`.
7. Writes are handled as Section 12 says: `SystemMode` as the `system-mode` effect, and setpoints as the `setpoint` effect.

For example, a setpoint function at $S = 25.5\text{ }^{\circ}\text{C}$, with $d = 1.0\text{ }^{\circ}\text{C}$ and limits of 24.0 and $27.5\text{ }^{\circ}\text{C}$, shows a heater with `OccupiedHeatingSetpoint` 2500 and limits of 2350 and 2700, and a chiller with `OccupiedCoolingSetpoint` 2600 and limits of 2450 and 2800.

11.3 Pumps

A Pump has the Pump Configuration and Control cluster with the `SPD` feature.

1. `EffectiveControlMode` and `ControlMode` are `ConstantSpeed`.
2. `OperationMode` is `Normal`, and `EffectiveOperationMode` is `Local` while `controlSource` is `Local` and `Normal` otherwise.
3. `MaxPressure`, `MaxSpeed`, `MinConstSpeed`, `MaxConstSpeed`, and `Capacity` are null; `MaxFlow` is null unless the endpoint gives `ratedFlowRate`.

4. PumpStatus has Running set while the pump runs, LocalOverride while controlSource is local, and the fault bits of Section 9.
5. Level Control's CurrentLevel is the speed, with MinLevel 1 and MaxLevel 200.

A level-output presented as a Pump follows the same rules, with its level as its speed.

11.4 Valves

1. CurrentState and TargetState are Open while the valve is open, and Closed otherwise.
2. OpenDuration is the length, and RemainingDuration the time left, of a temporary open state with a known end, such as a Matter override [OAP-AU Section 17.5]; both are null otherwise.
3. DefaultOpenDuration is the output's duration until a home app writes it. A write is the default-duration effect: the bridge keeps the value for that Matter endpoint, as Section 11.5.4 keeps other settings, and uses it for later Open commands that give no duration. It never changes the output's duration, which only the owner chooses (Section 8.2).
4. The valve has neither the TS nor the LVL feature.

11.5 On/Off and Level Control

1. On/Off's GlobalSceneControl is true. OnTime is the time left, in tenths of a second, of a Matter override set with OnWithTimedOff, and 0 otherwise; OffWaitTime is 0.
2. Level Control's MinLevel is 1 and MaxLevel 254, except on a Pump (Section 11.3). While the level is 0 %, OnOff is false and CurrentLevel keeps its last value.
3. RemainingTime is 0, because fades are not reported.
4. Writes to OnLevel, OnTime, transition times, and Options are kept by the bridge for that Matter endpoint, and shape only later Matter commands.

11.6 Power-Up Behavior

On/Off's StartUpOnOff shows the endpoint's powerOnBehavior [OAP-FS Section 9.1]:

1. null when powerOnBehavior is last;
2. Off when it is fail-safe and the endpoint's failSafeAction is state with the inactive state;
3. On when it is fail-safe, failSafeAction is state, and failSafeState is a running state;
4. null when it is fail-safe, failSafeAction is continue, and the endpoint runs a local program (local-program or wave); and
5. Off otherwise.

Level Control's StartUpCurrentLevel is null. Writes to either are refused (Section 12.9).

12 Controls

12.1 Operations and Roles

1. The bridge MUST carry out each Matter command and attribute write that it accepts as the operation that Table 8 names for its effect, as its service user.
2. The controller MUST check the bridge's role for each operation, as for any user [OAP-SA Section 11.3], and MUST record audited operations in its audit log with the bridge's user as actor, via matter, and the fabric in details, as fabricIndex, label, and vendorId [OAP-SA Section 12.1.6]. A bridge on another host MUST give the controller the fabric of each operation it asks for, so that the controller can record it.
3. The set-override operation MUST create, change, or end an override of the output [OAP-AU Section 15], with by the bridge's user, so that the controller sends it with revertAfter and it ends on the device even if the controller fails [OAP-AU Section 17.5].
4. Apart from keeping the settings of Section 11.5.4, a bridge MUST NOT do anything else for a Matter command.

12.2 Matter Commands

Table 7 — Matter commands and writes

Cluster	Matter command or attribute	Effect
On/Off	Off, OffWithEffect	stop
On/Off	On, OnWithRecallGlobalScene	run
On/Off	Toggle	toggle
On/Off	OnWithTimedOff	run-for, for OnTime, in tenths of a second
On/Off	Writes to OnTime, OffWaitTime	bridge-setting
On/Off	Writes to StartUpOnOff	refuse
Level Control	MoveToLevel, Move, Step, Stop, MoveToLevelWithOnOff, MoveWithOnOff, StepWithOnOff, StopWithOnOff	level
Level Control	Writes to OnLevel, OnOffTransitionTime, OnTransitionTime, OffTransitionTime, DefaultMoveRate, Options	bridge-setting
Level Control	Writes to StartUpCurrentLevel	refuse
Scenes Management	RecallScene	scene
Scenes Management	AddScene, StoreScene, RemoveScene, RemoveAllScenes, CopyScene	bridge-setting
Groups	AddGroup, RemoveGroup, RemoveAllGroups, AddGroupIfIdentifying	bridge-setting
Pump Configuration and Control	Writes to OperationMode, ControlMode, LifetimeRunningHours, LifetimeEnergyConsumed	refuse
Thermostat	Writes to SystemMode	system-mode
Thermostat	Writes to OccupiedHeatingSetpoint, OccupiedCoolingSetpoint	setpoint
Thermostat	SetpointRaiseLower	setpoint
Thermostat	Writes to MinHeatSetpointLimit, MaxHeatSetpointLimit, MinCoolSetpointLimit, MaxCoolSetpointLimit, ControlSequenceOfOperation, LocalTemperatureCalibration, RemoteSensing	refuse
Valve Configuration and Control	Open	run-for, for OpenDuration, or else DefaultOpenDuration
Valve Configuration and Control	Close	stop
Valve Configuration and Control	Writes to DefaultOpenDuration	default-duration
Identify	Identify, TriggerEffect	identify
Identify	Writes to IdentifyTime	identify
Bridged Device Basic Information	Writes to NodeLabel	rename

Commands and writes not in Table 7, such as those of the Groups cluster that only query, are handled as Matter defines and change no output. A `SetpointRaiseLower` command asks for the current setpoint plus its `Amount`, in steps of 0.1 °C (provisional; Appendix D, issue 15). `Move` and `Step` commands are carried out as a series of `level` effects, no more often than once a second [OAP-AU Section 17.4].

12.3 Effects

The operation of `default-duration` and `bridge-setting` gives only the role they need; they set no override.

Table 8 — Effects

Effect	Operation	What the bridge does
stop	set-override	An override that stops the output, for the output's <code>duration</code> , within the limits on repeated stops of Section 12.6. Allowed by <code>stop</code> and <code>full</code> . It also aborts a water change that is using the output, and is then never refused for the output's minimum run time (Section 12.11).
run	set-override	With <code>full</code> , an override that runs the output, for its <code>duration</code> . With <code>stop</code> , it ends an override that the bridge set, which returns the output to its automation, and otherwise changes nothing.
toggle	set-override	<code>stop</code> while <code>OnOff</code> is <code>true</code> , and <code>run</code> otherwise.
run-for	set-override	As <code>run</code> , but for the duration the command gives, up to the output's <code>maxDuration</code> .
level	set-override	With <code>full</code> , an override with a level, for the output's <code>duration</code> . Refused otherwise.
scene	set-override	Each value the scene holds is applied as the matching command: <code>OnOff</code> as <code>stop</code> or <code>run</code> , and <code>CurrentLevel</code> as <code>level</code> .
system-mode	set-override	<code>Off</code> is <code>stop</code> . <code>Heat</code> on a heating Thermostat, or <code>Cool</code> on a cooling one, is <code>run</code> . Other values are refused with <code>CONSTRAINT_ERROR</code> .
setpoint	change-configuration	Refused unless the owner has allowed setpoint changes for the output; then an offset (Section 12.7).
identify	identify	The endpoint's <code>identify</code> command, with <code>duration</code> from <code>IdentifyTime</code> or the command, or 10 s.
rename	change-device-settings	Sets the name that <code>NodeLabel</code> is drawn from (Section 15.4): the endpoint's <code>name</code> , or, with <code>change-configuration</code> instead, the <code>name</code> of the output, input, or equipment.
default-duration	set-override	Kept by the bridge for the Matter endpoint, from 60 s to the output's <code>maxDuration</code> , as the duration of later <code>Open</code> commands without one; it never changes the output's <code>duration</code> (Section 11.4). Null is refused with <code>CONSTRAINT_ERROR</code> .
bridge-setting	set-override	Kept by the bridge for the Matter endpoint. It shapes later Matter commands, and never what the output's policy allows.
refuse	—	Always refused with <code>UNSUPPORTED_ACCESS</code> : a fail-safe setting, a setting of the equipment, or the owner's limits.

12.4 Policies

Table 9 — Policies

Policy	Home apps can
none	Read only. Every effect that would change the output is refused.
stop	Home apps can stop the output for a while, and end their own stop. They cannot start it or change its level.
full	Home apps can run, stop, and set the level of the output for a while.

1. Every exposed output has a policy, from Table 10 or the owner's choice (Section 8.2).
2. The bridge MUST apply the policy to every command, whatever sent it: a person, a routine, a group, a scene, or a binding between nodes.
3. An override that a person set in an Open Aquarium app takes precedence over home apps. While one is in effect, the bridge MUST NOT replace or end it: a `stop` while it stops the output changes nothing, and every other effect that would set an override is refused (Section 12.8). Under `stop`, `run` ends only a Matter override.
4. A mode that a person started, in an Open Aquarium app or at the controller, also takes precedence over home apps. While it, or an extended state of it [OAP-AU Section 14.4], acts on an output (Section 13.5.1), the bridge MUST NOT set an override of that output: a `stop` while the mode stops the output changes nothing, and every other effect that would set an override is refused (Section 12.8). The bridge MUST NOT end the mode, nor start another mode of its kind, which would end it [OAP-AU Section 14.3.3] (Section 13.1.7).

12.5 Policies by Equipment

Table 10 — Policies by equipment

Equipment category	On loss of contact	Policy	Owner may choose	Duration	Longest	Run before the next stop	Why
return-pump	run	none	none, stop	600 s	900 s	3600 s	Stopping the return pump stops filtration and gas exchange, and “turn everything off” in a home app must never do it. An owner who wants a pause from a home app gets at most 15 minutes, as a feed mode would.
circulation-pump	run	none	none, stop, full	600 s	900 s	3600 s	Flow keeps the water oxygenated. Running and changing speed are safe; a stop is limited to 15 minutes.
air-pump	run	none	none, stop, full	600 s	900 s	3600 s	Aeration keeps the water oxygenated.
protein-skimmer	run	none	none, stop, full	1800 s	3600 s	3600 s	Stopping a skimmer for an hour harms nothing, but restarting it with the sump too full can overflow its cup, so it is not stopped by default.
filter	run	none	none, stop	600 s	900 s	3600 s	A canister filter left off for hours loses its oxygen and releases waste when restarted.
uv-sterilizer	run	none	none, stop, full	3600 s	3600 s	3600 s	Fail-Safe Behavior keeps it running on loss of contact. Stopping it for an hour, for example while dosing a medication, is harmless; longer stops belong in the configuration.
heater	stop	stop	none, stop	3600 s	3600 s	3600 s	A heater forced on can kill livestock within hours. Stopping it is safe for an hour, but a nightly “good night” routine must not leave a tank unheated all night, so a stop lasts at most an hour.
chiller	stop	stop	none, stop	3600 s	3600 s	3600 s	As for heaters, in the other direction.
doser	stop	stop	none, stop	3600 s	86400 s	—	A missed dose is safer than an overdose. A home app never starts a doser.
reactor	stop	stop	none, stop	3600 s	86400 s	—	The parts of a reactor that add chemicals are never started from a home app.
co2-system	stop	stop	none, stop	3600 s	86400 s	—	CO2 without pH control can suffocate fish.
ozone-generator	stop	stop	none, stop	3600 s	86400 s	—	Ozone without ORP control can reach harmful levels.
feeder	stop	stop	none, stop	3600 s	86400 s	—	Uncontrolled feeding pollutes the water. A home app never starts a feeder, and a feed mode that feeds is not offered as a switch (Section 13.5).
ro-di	stop	stop	none, stop	3600 s	86400 s	—	An open supply valve can flood the room.
water-changer	stop	stop	none, stop	3600 s	86400 s	—	A person who sees a water change going wrong can stop it, which aborts the water change;

Equipment category	On loss of contact	Policy	Owner may choose	Duration	Longest	Run before the next stop	Why
							only an Open Aquarium app resumes it (Section 12.11).
light	continue	full	none, stop, full	3600 s	14400 s	14400 s	Turning a light on or off for a while is the most common home-app control. Four hours is the longest scene (Lighting Programs Section 13.2).
auto-top-off	continue	stop	none, stop	3600 s	14400 s	14400 s	A top-off pump forced on can flood the room; a top-off unit stopped for hours lets salinity rise.
algae-scrubber	continue	full	none, stop, full	3600 s	14400 s	14400 s	Its lights, like any light.
analyzer	continue	stop	none, stop, full	3600 s	86400 s	86400 s	Testing is harmless, but uses reagents.
fan	run	none	none, stop, full	600 s	900 s	3600 s	A fan stopped in a hot room lets the tank overheat, so a stop is limited to 15 minutes, as for air pumps; running it is harmless.
alarm-signal	run	none	none	600 s	900 s	3600 s	Home apps neither sound nor silence an alarm output, which follows only its alarm function (Alarms and Notifications Section 15). Its durations are never used, because its only policy is none.
other	stop	stop	none, stop	3600 s	86400 s	—	When in doubt, a home app may stop the equipment but not start it.

1. An output's default policy, durations, and minimum run time are those of its equipment category: the category of its Data Model equipment, else its endpoint's `loadCategory`, else `other`. A category that Table 10 does not list is treated as `other`.
2. When both are known and differ, the bridge MUST apply the stricter: the earlier policy in the order `none`, `stop`, `full`; only the policies both allow; the shorter of each duration; and the longer minimum run time.
3. Table 10 follows three rules, which the tools of Section 16.3 check:
 - equipment that Fail-Safe Behavior keeps running on loss of contact (`run`) has the default policy `none`, and no Matter override of it lasts more than an hour;
 - equipment that Fail-Safe Behavior stops on loss of contact (`stop`) never allows `full`, so that no home app can start it; and
 - every minimum run time is at least the longest duration, and at least an hour, so that repeated Matter stops never keep an output stopped for more than half the time (Section 12.6); but equipment that Fail-Safe Behavior stops on loss of contact, and whose longest duration is a day, has none, because stopping it is its safe side (Section 12.6.5).
4. The Project Lead decided the policies of Table 10 (Appendix D, issue 2), and the rule by which each minimum run time follows its category's longest duration (Appendix D, issue 28). The durations stay provisional until owners' experience of home apps confirms them (Appendix D, issue 21), and each minimum run time changes only with its category's longest duration. Changing any of them is a Project Lead decision.

12.6 Durations

1. Every Matter override MUST end: at its start plus the output's duration, or, for `run-for`, plus the time the command gives, but never later than the output's longest duration.

2. A new Matter command for an output replaces an earlier Matter override of it [OAP-AU Section 15, item 2], with a new end, but never an override that a person set (Section 12.4.3).
3. Matter overrides are overrides like any other: they are shown with the time left wherever the output is shown, interlocks and the emergency and power-loss modes take precedence over them, and they are kept across restarts [OAP-AU Sections 15 and 17.3].
4. Matter stops of an output that follow one another with no time between, each replacing the one before, form a chain. A Matter stop that continues a chain MUST end no later than the start of the chain's first stop plus the output's longest duration, whatever the command asks.
5. When a Matter stop of an output ends, for whatever reason, such as its end, a home app's `On`, or an owner's override, the bridge MUST refuse every Matter stop of the output, with `INVALID_IN_STATE`, until the output has gone without one for its minimum run time (Table 10). This item does not apply to an output whose categories have no minimum run time, which Fail-Safe Behavior stops on loss of contact and which may be stopped for a day (Section 12.5.3), nor to a stop of an output that an automatic water change is using (Section 12.11.4).
6. The bridge MUST keep what items 4 and 5 need, the start of each chain and the end of each output's last Matter stop, across restarts, as it keeps the overrides themselves.
7. Items 4 to 6 hold for each output across every Matter stop of it, whatever caused it: a Matter override, or a mode, a scene, or a virtual output that a home app switched (Section 13.6).

For example, a routine that switches a heater off every 59 minutes stops it for an hour, and is then refused until the heater has followed its thermostat for at least an hour, so the heater is never stopped for more than half the time. A routine that stops a return pump whose owner allows pauses every 5 minutes gets one pause of 15 minutes, and then none for an hour, whether it stops the pump directly or starts feed modes that pause it. A routine that stops an ozone generator every day keeps it stopped, which is its safe side.

12.7 Setpoint Changes

1. The bridge MUST refuse a setpoint change unless an owner has allowed setpoint changes for the output, and the bridge's role permits `change-configuration`. An owner can allow them only for an output driven by a `setpoint` or `device-setpoint` function with `limits`.
2. An allowed change to a value `v` adds an offset to the function's setpoint [OAP-AU Section 12.3], so that the Thermostat shows `v`, from now until the end of the output's duration. It replaces any offset that the bridge added before.
3. A value that would put the effective setpoint outside the function's `limits` is refused with `CONSTRAINT_ERROR`.

Offsets are part of the configuration in Automation revision 0.3, so a setpoint change needs the `full-control` role.

12.8 Deciding a Command

For each Matter command or write, the bridge decides in this order, and stops at the first that applies:

1. an effect of `refuse` is refused with `UNSUPPORTED_ACCESS`;
2. an effect whose operation the bridge's role does not permit is refused with `UNSUPPORTED_ACCESS`;
3. `toggle` and `system-mode` become `stop` or `run`, and a `SystemMode` the Thermostat does not offer is refused with `CONSTRAINT_ERROR`;
4. `identify`, `rename`, `default-duration`, and `bridge-setting` are carried out, or refused as Table 8 says;
5. an effect that the output's policy does not allow is refused with `UNSUPPORTED_ACCESS`, except that `run` and `run-for` under `stop` end a Matter override, if there is one, and otherwise change nothing;
6. a setpoint outside the limits is refused with `CONSTRAINT_ERROR`;
7. if the controller has lost the endpoint's device, the command is refused with `FAILURE`;
8. if an interlock, an emergency mode, power-loss mode, or `stop-all` decides the output, or the endpoint is under local control, the command is refused with `INVALID_IN_STATE`, and no override is set;
9. if an override that a person set is in effect, or a mode that a person started acts on the output (Section 12.4.4), a `stop` while it stops the output changes nothing, and any other effect is refused with `INVALID_IN_STATE`; and

- otherwise, the bridge carries out the effect, but refuses a `stop` with `INVALID_IN_STATE` before the output has gone without a Matter stop for its minimum run time, where Section 12.6.5 applies, ends a `stop` that continues a chain no later than Section 12.6.4 allows, and, for a `stop` of an output that a water change is using, also aborts the water change (Section 12.11).

A command sent to a group has no response, but is decided the same way for each Matter endpoint. The test vectors give the decision for typical commands (`test-vectors.json`).

Table 11 — Statuses

Status	Code	When
UNSUPPORTED_ACCESS	0x7E	The bridge's role does not permit the operation (Section 12.1), the output's policy does not allow the effect (Section 12.4), or the effect is <code>refuse</code> .
CONSTRAINT_ERROR	0x87	A value the output does not allow: a setpoint outside its limits, a <code>SystemMode</code> it does not offer, or a <code>DefaultOpenDuration</code> that is null or out of range.
INVALID_IN_STATE	0xCB	A layer above overrides decides the output now: an interlock, an emergency mode, power-loss mode, or <code>stop-all</code> ; the endpoint is under local control; a person's override is in effect (Section 12.4.3); or a home app stops an output, or starts a mode, before the minimum run time since its last Matter stop has passed (Sections 12.6 and 13.1).
FAILURE	0x01	The controller has lost the endpoint's device (Fail-Safe Behavior Section 7.6), or the bridge has lost the controller.
BUSY	0x9C	A command for an output, other than one that stops it, arrived less than 1 s after the previous one (Section 17.6).

12.9 What Home Apps Never Change

- A bridge MUST NOT change, and MUST refuse writes that would change, an endpoint's fail-safe settings or power-up behavior [OAP-FS Section 13], including through `StartupOnOff` and `StartupCurrentLevel`. Fail-Safe Behavior requires the owner's confirmation for such changes (its Section 13.1.3), which a home app cannot give.
- A bridge MUST NOT change the settings of equipment, such as a pump's operation mode, a thermostat's limits, or a calibration, nor the configuration, except for the setpoint offsets of Section 12.7 and the names of Section 15.4.

12.10 Groups, Scenes, and Bulk Commands

- A bridge MUST implement the Groups and Scenes Management clusters wherever its device types require them.
- A scene that a home app recalls MUST be applied as the commands of the values it holds, each decided as Section 12.8 says.
- Group commands, scene recalls, and commands such as "turn everything off" or "turn everything on" are therefore safe by Table 10: they never stop equipment that must keep running unless its owner allows it, and then only briefly, and not again until it has run for its minimum run time (Section 12.6), and they never start equipment that must stop when its controller is lost. "Turn everything on" also starts the modes, scenes, and virtual outputs offered as switches, which Section 13.5 keeps within the same limits.
- A routine can still stop equipment whose policy is `stop`, for example a heater for an hour each night. Controllers SHOULD tell owners so when they turn the bridge on.

12.11 Water Changes

An automatic water change drains water from the system and then refills it [OAP-WL Section 9]. A person who sees one going wrong may stop its pump from a home app, and must then be able to trust that nothing starts it again until someone has checked the system.

- A `stop` that the bridge carries out for an output that an automatic water change is using, an output of its `drain.outputs` or `fill.outputs` from the start of its drain to the end of its settling [OAP-WL Section 9.2], MUST also abort the water change [OAP-WL Section 9.11], rather than pause it as another override would [OAP-WL Section 9.9].

2. Only a person resumes or abandons the aborted water change, in an Open Aquarium app, with the `clear-latch` operation [OAP-WL Section 9.11.2]. A later Matter command, such as `On` or `Open`, ends the bridge's override, but MUST NOT resume or abandon the water change.
3. A Matter stop that changes nothing, because a person's override already stops the output (Section 12.4.3), does not abort the water change.
4. The bridge MUST NOT refuse a `stop` of such an output for its minimum run time (Section 12.6.5), so that a person can always stop a water change that is going wrong.

13 Virtual Switches

13.1 Modes

1. An owner MAY offer a mode of kind `feed` or `maintenance`, or of an extension kind, as a virtual switch, if the mode has a `duration` and what it does to every output stays within Table 10 (Section 13.5).
2. `On` starts the mode, and `Off` ends it, as if the owner had done so in an Open Aquarium app (operation `start-mode`); `OnOff` is `true` while it is active. The mode's `duration` [OAP-AU Section 14] bounds it, and its extended states follow as usual.
3. Modes of kind `emergency`, `power-loss`, and `water-change` MUST NOT be offered (Table 3).
4. Because "turn everything on" would start a mode offered as a switch, the controller SHOULD warn an owner who offers a maintenance mode.
5. A home app's `On` for a mode that is active MUST change nothing: it does not move the mode's end, as starting a mode again otherwise would [OAP-AU Section 14.3.4].
6. The bridge MUST refuse a home app's `On` for a mode, with `INVALID_IN_STATE`, while the mode would cause a Matter stop that Section 13.6.2 does not allow.
7. The bridge MUST refuse, with `INVALID_IN_STATE`, a home app's `Off` for a mode that a person started, and its `On` for a mode while a mode of the same kind that a person started is active (Section 12.4.4).
8. A maintenance mode that a home app started MUST NOT hold alarms: while it is active, the controller sounds and notifies alarms as if it were not [OAP-AN Section 12.8], because nobody need be at the tank.

So a routine that switches feed modes on every few minutes cannot keep the pumps they pause stopped, and a home app cannot end a maintenance mode that the owner started at the tank.

13.2 Scenes

An owner MAY offer a scene of a lighting function [OAP-LP Section 13] as a virtual switch, if what it does to every light stays within Table 10 (Section 13.5.6). `On` starts the scene for its `duration`, and `Off` ends it (operation `start-mode`). The bridge MUST refuse a home app's `On` for a scene, with `INVALID_IN_STATE`, while the scene would cause a Matter stop that Section 13.6.2 does not allow.

13.3 Virtual Outputs

An owner MAY offer a virtual output [OAP-AU Section 8.4] as a virtual switch, if every output that depends on it is one that Table 10 lets home apps run (Section 13.5). Its policy is `full`, its duration 3600 s, and its longest duration 86400 s, each no longer than the shortest longest duration of the outputs that depend on it, and its minimum run time is its longest duration, but at least 3600 s (Section 12.5.3). The bridge MUST refuse a home app's `On` or `Off` for it, with `INVALID_IN_STATE`, while its override would cause a Matter stop that Section 13.6.2 does not allow. Because a virtual output can change what several outputs do, the controller SHOULD show the owner which outputs depend on it.

13.4 Presentation

1. Every virtual switch is an On/Off Plug-in Unit (0x010A), named as Section 15 says.
2. Its `IdentifyType` is `None`, and identify commands change nothing.
3. Writes to its `NodeLabel` are refused with `UNSUPPORTED_ACCESS`.

Table 12 — Virtual switches

Kind	What	On	Off	Notes
mode	Modes of kind <code>feed</code> or <code>maintenance</code> , or of an extension kind, that have a <code>duration</code> and stay within Table 10 (Section 13.5)	Start the mode (<code>start-mode</code>)	End the mode (<code>start-mode</code>)	The mode's own <code>duration</code> bounds it. <code>On</code> never extends an active mode; it starts a mode only while the Matter stops it causes are allowed (Section 13.6), and neither command ends a mode that a person started (Section 13.1).
scene	Scenes of a lighting function (Lighting Programs Section 13) that stay within Table 10 (Section 13.5)	Start the scene (<code>start-mode</code>)	End the scene (<code>start-mode</code>)	The scene's own <code>duration</code> , at most 4 hours, bounds it. <code>On</code> starts it only while the Matter stops it causes are allowed (Section 13.6).
virtual-output	Virtual outputs (Automation Section 8.4) on which only outputs that Table 10 lets home apps run depend (Section 13.5)	An override that runs it (<code>set-override</code>)	An override that stops it (<code>set-override</code>)	Overrides last 3600 s unless the owner chooses otherwise, and no longer than the shortest longest duration of the outputs that depend on it (Section 13.3), and are set only while the Matter stops they cause are allowed (Section 13.6). At most 86400 s.

13.5 Staying within Table 10

A mode, a scene, or a virtual output offered as a switch can be started by “turn everything on”, like any other switch, so it MUST NOT do more to any output, or for longer, than Table 10 lets a home app do to that output directly.

1. A mode acts on each output it lists or sends a command to, and on each output that depends on it (item 2). It stays within Table 10 when it has a `duration`, and, with the categories of Section 12.5.1:
 - for each output it lists, a `stop` state, or a level of 0, is for an output whose categories let the owner choose `stop` or `full`;
 - for each output it lists, a `run` state or a level above 0, and for each output it sends a command to [OAP-AU Section 14.5], is for an output whose categories let the owner choose `full`;
 - for each output it lists, the mode's `duration`, plus the output's `extend` [OAP-AU Section 14.4], is no longer than the output's longest duration;
 - for a virtual output it lists, every output that depends on the virtual output is one whose categories let the owner choose `full`, and the mode's `duration` and `extend` are no longer than their longest durations; and
 - every output that depends on the mode is one whose categories let the owner choose `full`, because what a function or trigger does with the mode's state can be anything, and the mode's `duration` is no longer than its longest duration.
2. An output depends on a mode or a virtual output when its demand can change with the mode's or the virtual output's state: when a function, a mode, an interlock, a program, or a trigger that acts on it reads the mode, through a `mode` condition [OAP-AU Section 10.8], or the virtual output, through an `output` input [OAP-AU Section 9.3] or as the output a `requires` interlock names [OAP-AU Section 13.3]; or reads either through derived inputs, conditions, or other virtual outputs that depend on it. The other outputs of an `exclusive` interlock that includes a virtual output depend on it too. Triggers include the scene triggers of a lighting function [OAP-LP Section 13.2.2] and the triggers of a flow function [OAP-FP Section 12.2].
3. The controller MUST check this when an owner offers a mode, a scene, or a virtual output, and again whenever the configuration changes.
4. While a mode, a scene, or a virtual output does not stay within Table 10, the bridge MUST NOT offer it, and MUST remove its bridged endpoint if it was offered (Section 7.4.6). The controller MUST tell the owner which output it would do too much to, and why.
5. When an owner offers a mode, a scene, or a virtual output, the controller MUST show the owner each output that it would stop, run, or set to a level, and each output that depends on it, although that output's own policy (Section 12.4) does not let a home app do so directly, and MUST offer it only after the owner confirms. It MUST ask again when a change to the configuration, or to a policy, adds such an output.

6. A scene stays within Table 10 when, for each light of its lighting function, the light's categories let the owner choose `full`, and the scene's `duration` is no longer than the light's longest duration.
7. An output that depends on a mode or a virtual output through a latch [OAP-AU Section 10.4] never stays within Table 10, because the latch can keep acting after the mode or the override has ended. Through a timer [OAP-AU Section 10.5], the timer's `duration` adds to how long the mode or the virtual output acts on the output.

For example, a feed mode of 10 minutes that stops the return pump, a wave pump, and the skimmer, with the skimmer's extended state of 10 minutes, stays within Table 10: each may be stopped, the pumps for up to 15 minutes, and the skimmer for up to an hour. The same mode with a `feed` command to the feeder does not, because no home app may start a feeder; nor does a maintenance mode that stops the return pump for an hour; nor does a feed mode whose flow function slows the return pump through a trigger, because no home app may set a return pump's level.

13.6 Stops through Switches

A home app that switches a mode, a scene, or a virtual output can stop outputs as surely as one that stops them directly, so the limits of Section 12.6 count both.

1. A mode or a scene that a home app started, and a Matter override of a virtual output, each cause a Matter stop of every output that it stops (a `stop` state or a level of 0), and of every output that depends on it (Section 13.5.2), for as long as it acts on that output.
2. Such a Matter stop is allowed only when, for each of those outputs, Section 12.6.5 would not refuse a Matter stop of it, and the switch would stop acting on it no later than Section 12.6.4 allows the chain of Matter stops it continues.
3. A mode or a scene that a person started causes no Matter stop.

For example, a routine that starts an offered feed mode of 10 minutes every 20 minutes, and stops the circulation pump that the mode pauses in between, pauses the pump once, and then cannot pause it again, by either means, until it has run for an hour.

14 Unit Conversions

14.1 Rules

1. A value is first converted to the unit Table 13 starts from, with the Data Model registry [OAP-DM], and then by the rule of Table 13.
2. "round" rounds to the nearest whole number, with halves away from zero.
3. A result outside the range of the Matter attribute is kept at the nearest end of it.
4. A value that cannot be used is null, where the Matter attribute allows null, and otherwise not reported: for example a sensor's value while its `quality` is not `good`, or an input that is unavailable [OAP-FS Section 10.1].
5. A value from Matter is converted the other way: a temperature T is $T / 100$ °C, then converted to the unit of the setpoint; levels as Table 13 says.

The test vectors give examples of each conversion (`test-vectors.json`).

14.2 Limits and Qualifiers

A reading with a `qualifier` of `<` or `>` [OAP-DC Section 11.1] is shown as its limit, because Matter cannot mark a value as a limit. For example, a flow sensor that reports more than 9000 L/h shows 90.

14.3 Conversions

Table 13 — Conversions

Conversion	Matter	Rule
<code>boolean</code>	<code>bool</code>	The same value.
<code>level-on</code>	<code>bool</code>	<code>true</code> while the level is above 0 %.

Conversion	Matter	Rule
level-254	uint8, 1 to 254	$\max(1, \text{round}(p \times 254 / 100))$ while p is above 0 %; at 0 % the last value is kept and <code>on0ff</code> is <code>false</code> . From Matter: $p = \text{level} \times 100 / 254$.
pump-level	uint8, 1 to 200, 0.5 % steps	$\max(1, \text{round}(p \times 2))$ while p is above 0 %; at 0 % the last value is kept and <code>on0ff</code> is <code>false</code> . From Matter: $p = \min(\text{level}, 200) / 2$.
temperature	int16, 0.01 °C	Convert to <code>degC</code> with the Data Model registry, then $\text{round}(t \times 100)$, kept between -27315 and 32766. Null while the value is unavailable.
humidity	uint16, 0.01 %	$\text{round}(h \times 100)$, kept between 0 and 10000. Null while the value is unavailable.
flow	uint16, 0.1 m³/h	Convert to <code>L/h</code> with the Data Model registry, then $\text{round}(q / 100)$, kept between 0 and 65534. Null while the value is unavailable.
milli	int64, mV, mA, or mW; uint32, mV, for <code>BatVoltage</code>	$\text{round}(x \times 1000)$, from <code>V</code> , <code>A</code> , or <code>W</code> .
energy	int64, mWh	$\text{round}(e \times 1000000)$, from <code>kWh</code> .
power-factor	int64, -10000 to 10000	$\text{round}(f \times 10000)$ (provisional).
rpm	uint16, rpm	$\text{round}(n)$, kept between 0 and 65534.
hours	uint24, h	The whole number of hours, rounded down.
power-status	PowerSourceStatusEnum	<code>Active</code> (1) while <code>powerSource</code> is <code>battery</code> ; <code>Standby</code> (2) while it is <code>external</code> .
battery-percent	uint8, 0 to 200, 0.5 % steps	$\text{round}(p \times 2)$, kept between 0 and 200. Null while the device does not give it.
battery-charge	BatChargeLevelEnum	<code>Warning</code> (1) while the device reports <code>battery-weak</code> ; <code>OK</code> (0) otherwise.
valve-state	ValveStateEnum	<code>Open</code> (1) when open, <code>Closed</code> (0) otherwise.
running-bit	bitmap bit	Set while the attribute it is drawn from shows the equipment running: <code>on</code> or <code>running</code> <code>true</code> , a level above 0 %, or a heater's <code>heating</code> or a chiller's <code>cooling</code> <code>true</code> .
fault-bits	bitmap bits	Set while one of the listed faults is active (Section 10.4).
local-mode	OperationModeEnum and PumpStatus bit	<code>Local</code> and <code>LocalOverride</code> while <code>controlSource</code> is <code>local</code> ; otherwise <code>Normal</code> .
start-up	StartUpOnOffEnum, nullable	Section 11.6.
contact-position	uint8	1 while the contact is closed, 0 otherwise.
press-events	events	<code>InitialPress</code> with <code>NewPosition</code> 1, then <code>ShortRelease</code> with <code>PreviousPosition</code> 1.
availability	nullable value	The value is null unless <code>quality</code> is <code>good</code> and the input is available (Fail-Safe Behavior Section 10.1).
range	as the value	Converted as the value. Null when the endpoint does not give it.
text-32	string, 32 bytes at most	UTF-8, cut at the last whole character within 32 bytes.
text-64	string, 1 to 64 bytes	UTF-8, cut at the last whole character within 64 bytes; left out when empty.
serial	string, 32 bytes at most	Left out when longer than 32 bytes, because a shortened serial number is wrong.
name	string, 32 bytes at most	Section 15.
derived	as the attribute	From the controller's automation, as Section 11 describes.

15 Naming

1. A bridged endpoint's `NodeLabel` is the first of these that is not empty, without leading and trailing spaces, cut as `text-32` (Table 13):
 1. the `name` of the output or input [OAP-AU Section 7.3.5];
 2. the `name` of its Data Model equipment [OAP-DM Section 8.3];
 3. the endpoint's `name` [OAP-DC Section 8.2];
 4. the device's `name`, a space, and the endpoint's `id`; and
 5. the device's `model`, a space, and the endpoint's `id`.

2. A virtual switch's `NodeLabel` is the `name` of its mode, scene, or virtual output, or else its `id`.
3. The bridge's own `NodeLabel`, in its Basic Information cluster, is the configuration's `name`, or else the name of its first system.
4. A write to an endpoint's `NodeLabel` is the `rename` effect, which sets the name that `NodeLabel` is drawn from, so that the new name shows:
 - the `name` of the output or input, or of its equipment, when item 1.1 or 1.2 gives `NodeLabel`, with the `change-configuration` operation; and
 - otherwise the endpoint's `name`, with the `change-device-settings` operation.
5. The controller SHOULD warn an owner when two exposed endpoints have the same name, because voice assistants cannot tell them apart.
6. Names that owners choose are the owner's data [OAP-SA Section 16], and sharing them with an ecosystem is covered by the owner's consent (Section 7.2).

For example, an output named "Heater" shows as "Heater", and an input with no name, whose equipment is "Temperature, sump", shows as "Temperature, sump".

16 Conformance

16.1 Bridges

A conforming bridge:

1. runs, turns on, and handles fabrics as Sections 7.1 to 7.3 require;
2. presents what Section 8 exposes, with the endpoints, numbers, Unique IDs, and reachability of Sections 7.4 to 7.6;
3. presents each endpoint with the device types, clusters, and attributes of Sections 9 to 11, and the conversions of Section 14;
4. decides every command as Section 12 requires, with the policies of Table 10 or the owner's choices;
5. presents virtual switches only as Section 13 allows, and decides their commands as Sections 13.1 to 13.3 and 13.6 require;
6. names bridged endpoints as Section 15 requires; and
7. gives the results of `test-vectors.json`, and passes the test procedures of Section 16.4.

16.2 Controllers

A conforming controller that offers a bridge, on itself or through a host:

1. keeps the bridge off until an owner turns it on, after the notice of Section 7.2;
2. gives the bridge a service user, and checks its role for every operation (Section 12.1);
3. sets Matter overrides with an end, and sends them with `revertAfter` (Section 12.6);
4. shows owners the fabrics, policies, and durations, and lets them change them (Sections 7.3 and 8.2);
5. records the bridge's operations, and each fabric added or removed, in its audit log, and notifies every owner of each fabric (Sections 7.3 and 12.1);
6. checks which modes, scenes, and virtual outputs stay within Table 10, and tells owners of those that do not (Section 13.5);
7. aborts a water change that a home app stops (Section 12.11); and
8. holds no alarms for a maintenance mode that a home app started (Section 13.1.8).

16.3 Machine-Readable Definitions and Tools

1. `mapping.json`, a JSON document [RFC 8259], is normative, and contains the same definitions as Tables 2, 3, and 4 to 13 and Section 9, which are generated from it by `tools/render_spec_tables.py`. That tool also checks that every Device Classes class, attribute, command, event, and fault is either mapped or listed with a reason, and that Table 10 follows the rules of Section 12.5.
2. `test-vectors.json` is normative. The examples in `examples/` are informative.

3. The validator, `tools/validate_matter_mapping.py`, is informative. It holds a reference implementation of Sections 7.5, 9 to 12, 13.1, 13.5, 13.6, 14, and 15, checks the test vectors against it, and checks that each example presents what its `expected` member says.
4. `tools/check_matter_ids.py`, which is informative, checks every Matter identifier and name in the mapping file against the Matter SDK's data model [SDK], which it reads where it was downloaded and never copies.

16.4 Test Procedures

Tests use a controller with simulated devices, an Open Aquarium app, and home apps of at least two ecosystems. Each test names the requirements it checks. Every MUST and MUST NOT of Sections 7 to 17 is checked by at least one test. A pass criterion that checks what a SHOULD asks applies to a product that does it.

Table 14 — Test procedures

Test	Requirements	Method	Pass criteria
Turned off by default	7.2.1, 7.2.2, 16.2.1	Start a new controller, and look for the bridge on the local network. Then turn the bridge on from an Open Aquarium app.	The bridge does not advertise or answer until an owner turns it on. Before it is turned on, the controller shows the notice of Section 7.2.2.
Turning on	7.2.1, 7.2.3, 16.2.2	Try to turn the bridge on as a user with the <code>full-control</code> role; then, as an owner, try to give the bridge the <code>owner</code> role; then give it <code>override</code> .	The first is refused, and so is the <code>owner</code> role. The bridge gets a user of kind <code>service</code> with the <code>override</code> role.
Commissioning windows	7.2.4, 7.3.3	Try to commission the bridge without any window open; have a user with the <code>full-control</code> role try to open a window; have an owner open two windows; then, from a home app of an ecosystem the bridge already belongs to, open a window through Matter's own mechanism, and commission a third ecosystem with it.	Commissioning fails without a window, and the <code>full-control</code> user cannot open one. Each window the bridge opens has its own passcode, printed nowhere, and neither works after its window closes. The third ecosystem is commissioned; <code>fabric-added</code> is recorded in the audit log, with <code>via matter</code> , and <code>fabricIndex</code> , <code>label</code> , and <code>vendorId</code> in <code>details</code> ; and every owner is notified of it.
Fabrics	7.3.1, 7.3.2, 7.3.3, 16.2.4, 16.2.5	Commission the bridge into as many ecosystems as Matter requires of a node, and give one home app Matter's highest privilege. From it, write a setting that the bridge's role does not permit. Then, with a rule that sends warnings to nobody, try to remove one ecosystem from an Open Aquarium app as a user with the <code>full-control</code> role, and remove it as an owner.	Every fabric is listed, with its label, vendor, and when it was added. The write is refused, as it is from every other fabric. The <code>full-control</code> user cannot remove the fabric. The removed fabric loses access within 5 s, <code>fabric-removed</code> is recorded in the audit log, and every owner is notified of each change.
Turning off	7.2.5	Turn the bridge off on the local network.	Within 5 s, no fabric can read or command it.
A bridge on a host	7.1.2, 7.1.3, 7.1.4, 7.6.1, 12.1.2	Run the bridge on a host, and capture its traffic while home apps of two ecosystems command outputs; then stop its connection to the controller; then stop the host.	The host connects only to the controller, through the Controller API, as a party of kind <code>service</code> , never to a device, and is never any device's controller. The controller records each command with <code>via matter</code> and the fabric that sent it. Every bridged endpoint becomes unreachable when the connection stops. Control goes on unchanged when the host stops, and Matter overrides end at their ends.
Device types	7.7.1, 7.7.2, 12.10.1, 16.1.3	Run <code>tools/check_matter_ids.py</code> against the data model of the release that <code>mapping.json</code> names. Read the bridge's Matter release, and the Descriptor of each bridged endpoint of the example of Appendix B.	Every identifier and name matches. The bridge's release defines every device type it presents. Each bridged endpoint has the device types and clusters of Table 2 and Section 9, the Groups and Scenes

Test	Requirements	Method	Pass criteria
			Management clusters included where Table 2 lists them.
Stable identity	7.4.5, 7.4.6, 7.5.1	Restart the bridge, rename equipment, and add an output; then give a heater's outlet a setpoint function, so that it becomes a Thermostat.	Existing bridged endpoints keep their numbers and Unique IDs, and the new output gets a new number. The outlet's bridged endpoint is replaced while the bridge runs, without a restart.
New key	7.5.1	Factory reset the controller, and turn the bridge on again.	Unique IDs differ from before.
Lost device	7.6.1, 12.8	Disconnect a device, and send a command to one of its endpoints. Then make an adapter report one of its bridged devices unreachable.	<code>Reachable</code> becomes <code>false</code> when the controller treats the device as lost, and when the adapter reports it, and the bridge sends <code>ReachableChanged</code> each time. The command is refused with <code>FAILURE</code> .
Owner's choices	8.2.3, 8.2.4, 16.2.4	Open the bridge's settings in an Open Aquarium app. As a user with the <code>override</code> role, try to change the skimmer's policy; then, as one with <code>full-control</code> , give it <code>stop</code> for 30 minutes.	Each exposed output's policy and durations are shown, and those that differ from Table 10 are marked. The first change is refused; the second is applied, and recorded.
What is exposed	8.1, 8.3	Configure a doser, a pH probe, a level switch, and an unbound outlet, and hide a sensor.	None of them is presented.
Batteries	10.1	Give a device that can run on a battery, such as a pump with a backup battery, external power; then run it on its battery until it reports <code>battery-weak</code> . Read the bridged endpoints of that device and of a device whose description does not list <code>powerSource</code> .	The battery device's bridged endpoints have the Power Source cluster with the BAT feature, and the other device's have none. <code>Status</code> is <code>Standby</code> on external power and <code>Active</code> on the battery; <code>BatPercentRemaining</code> is twice <code>batteryLevel</code> ; <code>BatChargeLevel</code> becomes <code>Warning</code> with <code>battery-weak</code> .
Identify	10.2	Send <code>Identify</code> to the bridged endpoint of an endpoint that lists <code>identify</code> , of one of Device Classes revision 0.2 or later that does not, and of one of revision 0.1 that refuses the command.	<code>IdentifyType</code> is <code>VisibleIndicator</code> , <code>None</code> , and <code>None</code> . The first endpoint identifies; nothing is sent to the second; the command is sent to the third, and its refusal is ignored. Every <code>Identify</code> succeeds.
Faults	10.4.2, 10.5.2	Open the wiring of a supervised leak sensor. Then make the device of an outlet lost.	<code>SensorFault</code> has <code>GeneralFault</code> , a <code>SensorFault</code> event is sent, and <code>StateValue</code> keeps its last value rather than showing a leak. The outlet shows no change of <code>OnOff</code> for the lost device.
Role	7.2.3, 12.1.1, 12.1.2, 12.8	Give the bridge the <code>view</code> role. Send <code>Off</code> , <code>On</code> , <code>MoveToLevel</code> , and <code>Identify</code> .	Every command but <code>Identify</code> is refused with <code>UNSUPPORTED_ACCESS</code> , and no override is set.
Audit	12.1.2, 16.2.5	Send commands that succeed and commands that are refused.	Each audited operation is recorded with the bridge's user as <code>actor</code> , via <code>matter</code> , and <code>fabricIndex</code> , <code>label</code> , and <code>vendorId</code> in <code>details</code> , whether it succeeded or was refused.
Bulk "off"	12.4.2, 12.10.2, 12.10.3	With the default policies, send <code>Off</code> to every bridged endpoint, singly and to a group, and recall a scene that holds <code>OnOff false</code> .	Outputs with <code>none</code> do not change, and unicast commands to them return <code>UNSUPPORTED_ACCESS</code> . Outputs with <code>stop</code> or <code>full stop</code> , each with an override that ends after its duration.
Bulk "on"	12.4.2, 12.5, 12.10.3	As above, with <code>On</code> .	No output whose equipment Fail-Safe Behavior stops on loss of contact starts. Outputs with <code>full</code> run for their duration, and outputs

Test	Requirements	Method	Pass criteria
			with <code>stop</code> that a home app had stopped return to their automation.
Stricter category	12.5.1, 12.5.2	Bind a heater to an outlet whose <code>loadCategory</code> is <code>Light</code> . Send <code>On</code> , then <code>Off</code> ; end the stop, and send <code>Off</code> again after 2 hours.	<code>On</code> starts nothing. The stop lasts no more than an hour, and the second <code>Off</code> is refused with <code>INVALID_IN_STATE</code> , because the light's minimum run time of 4 hours is the longer.
People's overrides	12.4.3, 12.4.4	Override a heater to stop for a day from an Open Aquarium app. Send <code>Off</code> , then <code>On</code> , from a home app; then override a light to run, and send <code>Off</code> . Start a maintenance mode that stops the skimmer and runs the moonlight from an Open Aquarium app, and send <code>Off</code> to both from a home app.	The heater's override is unchanged after both commands. The light's <code>Off</code> is refused with <code>INVALID_IN_STATE</code> , and its override is unchanged. The skimmer's <code>Off</code> changes nothing, the moonlight's is refused with <code>INVALID_IN_STATE</code> , and no override is set.
End on the device	12.1.3, 12.6.1, 16.2.3	Stop a heater from a home app, then stop the controller's control functions.	The stop is sent with <code>revertAfter</code> , and the heater resumes at the end of the override or applies its fail-safe behavior, whichever its configuration gives.
Timed runs	12.6.1, 12.6.2	Send <code>OnWithTimedOff</code> with an <code>OnTime</code> longer than a light's longest duration; then send <code>Off</code> .	The first override ends at the longest duration; <code>Off</code> replaces it.
Repeated stops	12.5.3, 12.6.4, 12.6.5, 12.6.6	Let home apps pause the return pump. Send <code>Off</code> to it every 5 minutes for 2 hours, and restart the bridge 30 minutes after its first pause ends. Then stop a heater from a home app, end the stop with <code>On</code> , and send <code>Off</code> 10 minutes later. Do the same with an ozone generator.	The pump pauses for 900 s once, from the first <code>Off</code> ; later <code>Off</code> s are refused with <code>INVALID_IN_STATE</code> , across the restart, until 3600 s after the pause ended, and the next one pauses it again. The heater's second <code>Off</code> is refused with <code>INVALID_IN_STATE</code> ; the ozone generator's is carried out.
Stops through switches	12.6.7, 13.2, 13.3, 13.6	Offer a feed mode of 10 minutes that pauses a circulation pump, a scene of a light, and a virtual output that the circulation pump follows. Start the feed mode from a home app; 5 minutes after it ends, send <code>Off</code> to the pump, <code>On</code> to the feed mode, and <code>On</code> to the virtual output; an hour after it ended, send <code>On</code> to the feed mode again. Stop the light from a home app, end the stop, and send <code>On</code> to the scene.	The <code>Off</code> , the feed mode's second <code>On</code> , and the virtual output's <code>On</code> are refused with <code>INVALID_IN_STATE</code> , and the last <code>On</code> starts the feed mode. The scene's <code>On</code> is refused with <code>INVALID_IN_STATE</code> until the light has gone 4 hours without a Matter stop.
Setpoints	12.7.1	Give the bridge the <code>full-control</code> role. Write a heater's setpoint with setpoint changes off; turn them on, and write values inside and outside the limits.	Refused with <code>UNSUPPORTED_ACCESS</code> ; then an offset that ends after the duration; then <code>CONSTRAINT_ERROR</code> .
Higher layers	12.8	Start an emergency mode, then send <code>On</code> to a light it stops.	<code>INVALID_IN_STATE</code> ; the light stays off, and no override is set.
Settings never written	12.1.4, 12.9.1, 12.9.2	Write <code>StartupOnOff</code> , <code>StartupCurrentLevel</code> , <code>OperationMode</code> , Thermostat setpoint limits, and <code>LocalTemperatureCalibration</code> , with the bridge's role <code>full-control</code> .	Each is refused with <code>UNSUPPORTED_ACCESS</code> , and nothing on the endpoint or in the configuration changes.
Water changes	12.11.1 to 12.11.4, 16.2.7	During the drain of an automatic water change, send <code>Off</code> from a home app to its drain pump; then <code>On</code> . Resume the water change from an Open Aquarium app. In another water change, have a person stop the drain pump with an override, and send <code>Off</code> from a home app. Then give the drain pump the category <code>circulation-pump</code> , stop it from a home app outside a water change, end the stop, and send <code>Off</code> to it during a water change 10 minutes later.	<code>Off</code> aborts the water change: it raises <code>water-change-aborted</code> and stops every one of its outputs. <code>On</code> ends the bridge's override, and the water change stays aborted until the person resumes it. In the second, the water change pauses, and does not abort. In the third, <code>Off</code> is carried out and aborts the water change, although the pump has not run for its minimum run time.

Test	Requirements	Method	Pass criteria
Mode switches	13.1.1 to 13.1.8, 16.1.5, 16.2.8	Offer a feed mode and a maintenance mode that stay within Table 10. Send <code>On</code> to the feed mode, <code>On</code> again 5 minutes later, then <code>Off</code> ; then <code>On</code> 10 minutes after it ended, and again an hour after it ended. Start the maintenance mode from a home app, and raise a warning during it. Start it from an Open Aquarium app, and send it <code>Off</code> , and <code>On</code> to another offered maintenance mode, from a home app. Try to offer an emergency mode.	The mode starts with its duration, the second <code>On</code> does not move its end, and <code>Off</code> ends it. The next <code>On</code> is refused with <code>INVALID_IN_STATE</code> , and the last starts it. The warning sounds and is notified at once. Both commands for the maintenance modes are refused with <code>INVALID_IN_STATE</code> , and the person's maintenance mode goes on. The emergency mode cannot be offered.
Staying within Table 10	13.5, 16.2.6	Try to offer a feed mode that sends <code>feed</code> to a feeder, a maintenance mode that stops the return pump for an hour, a virtual output that the return pump's function reads, a feed mode whose flow function slows the return pump through a trigger, a mode that sets a latch that stops a circulation pump, and a scene of 4 hours of a light on an outlet whose <code>loadCategory</code> is <code>other</code> . With the return pump's policy <code>none</code> , offer a feed mode that stops it for 10 minutes and stays within Table 10, then change its <code>duration</code> to 20 minutes.	None of the six is offered, and the controller names the output each would do too much to. The controller names the return pump, whose own policy does not allow a stop, and offers the feed mode only after the owner confirms; after the change, its bridged endpoint is removed and the owner is told why.
Names	15, 12.9.2	Name an output with more than 32 bytes. With the bridge's role <code>full-control</code> , write <code>NodeLabel</code> for an output named in the configuration, and for an endpoint known only by its device's model.	<code>NodeLabel</code> is cut at a whole character within 32 bytes. The first write renames the output in the configuration, and the second sets the endpoint's <code>name</code> ; both show as the new <code>NodeLabel</code> .
Floods	17.6	From one home app, send <code>MoveToLevel</code> to a fan 10 times a second for a minute, and <code>Off</code> among them.	Each <code>Off</code> is decided at once. A bridge that limits <code>level</code> effects, as Section 17.6 advises, carries out no more than one a second, and refuses the others with <code>BUSY</code> .
Unavailable readings	14.1	Unplug a temperature probe, and make another reading stale.	<code>MeasuredValue</code> and any Thermostat's <code>LocalTemperature</code> that use them become null.

17 Security Considerations

- Ecosystems can reach the bridge from the internet.** A home ecosystem that has commissioned the bridge can do what the bridge's role and policies allow, from wherever its own services reach, even with the controller's remote access turned off [OAP-SA Section 13]. This is why the bridge is off until an owner turns it on, after a notice, and why the role and the policies limit what any fabric can do (Sections 7.2 and 12).
- Matter's access control is not the limit.** Each fabric's administrator controls the fabric's access, and any administrator can add another fabric. The bridge applies its role and policies to every fabric, and the controller shows and records every fabric, and tells every owner when one is added or removed (Section 7.3).
- Group commands are not acknowledged.** Any node of a fabric that holds its group keys can send a group command, and gets no answer. Group commands are decided like any other (Section 12.8).
- Matter's own security.** The bridge's credentials, commissioning, and sessions are as Matter defines them. A bridge built with Matter's test credentials, as an uncertified bridge may be, gives no assurance of who made it.
- A bridge on a host.** Whoever controls the host can do what the bridge's service user can. Owners should give it no more than the role they need.

6. **Floods of commands.** A faulty or hostile ecosystem could send many commands. A bridge SHOULD carry out no more than one `Level` effect per output per second, and MAY refuse other commands for an output that arrive faster than one per second with `BUSY`, but MUST NOT delay a `stop` [OAP-AU Section 17.4].
7. **Privacy.** Names, readings, and states are the owner's data, and sharing them with an ecosystem is covered by the consent of Section 7.2. Unique IDs are keyed, so they cannot link two bridges, or a bridge to a serial number (Section 7.5).
8. **No safety function.** Nothing the bridge does is needed to keep livestock safe, and losing it changes nothing in control (Section 7.1).
9. **Repeated stops.** A routine, or a hostile ecosystem, could keep equipment stopped for good by stopping it again before each stop ended. Section 12.6 ends each chain of stops at the longest duration, and makes the equipment run for its minimum run time before the next, whether a home app stops it directly or through a mode, a scene, or a virtual output (Section 13.6), so that a home app never keeps equipment stopped for more than half the time, unless stopping is that equipment's safe side.
10. **Modes, scenes, and virtual outputs.** One switch can change what many outputs do, and "turn everything on" reaches it as it reaches any switch. Section 13.5 offers only those whose effect on every output, through triggers and conditions too, a home app could have directly.

Appendix A — Approval Checklist

Complete this checklist before requesting Working Group approval. Link each item to the evidence.

Item	Status	Details
Section 2 describes compliance with all five tenets.	Done	
A public review period of at least 30 days has been announced and completed.	Not started	
At least two independent implementations exist, at least one from an organization not represented among the Authors.	Not started	
The implementations have passed the test procedures in Section 16.4, with at least two home ecosystems.	Not started	
The two implementations present the same system the same way in the same home app.	Not started	
Every normative reference is publicly available.	Done	All are free to read. The Matter specification needs no membership, but is not openly licensed: it may not be republished, and the SDK's machine-readable data model may be used only internally. The Project Lead accepted it on 17 October 2026 as a pinned normative reference, because a bridge cannot be defined without Matter's own definitions, anyone can read them, and this Specification and its mapping file copy only identifiers and names (Appendix D, issue 1).
The mapping file, test vectors, and examples are published, and the examples and test vectors pass the validator.	Done	
Every Matter identifier in the mapping file matches the Matter data model of the release it names.	Done	<code>tools/check_matter_ids.py</code> against Matter 1.6.1.
The Project Lead has decided the default policies, minimum run times, and role.	Done	The rule for minimum run times on 16 October 2026, and the default policies and role on 17 October 2026 (Appendix D, issues 2, 3, and 28); the durations stay provisional (issue 21).

Item	Status	Details
The Open Issues appendix is empty, or each open issue is explicitly deferred.	In progress	Issue 5 waits on the lawyer's review of the project's scope, issues 14 and 15 on checks against the Matter specification's text, and issue 21 on owners' experience of home apps; issues 6, 9, 11 to 13, and 16 to 20 are deferred beyond revision 1.0.
No unresolved Exclusion Notices for this Specification are recorded in <code>Notices.md</code> .	Done	
The Maintainer has recorded consensus to approve.	Not started	

Appendix B — Example (Informative)

`examples/reef-system.json` presents the reef system of the Automation examples (`reef-system.json`), with device descriptions for its devices. The bridge has the role `override`. The owner has let home apps stop the skimmer for up to 30 minutes, and has asked to offer feed mode A, maintenance mode, and the `night` virtual output as switches.

Table 15 — The example's bridged endpoints

Matter endpoint	Name	Device types	Policy	What a home app sees and can do
2	Heater	Thermostat (heating), Electrical Sensor	<code>stop</code> , 1 h	25.31 °C from the higher of two probes; setpoint 25.0 °C, limits 23.5 to 27.0 °C; 300.5 W. It can switch the heater off for an hour.
3	Chiller	Thermostat (cooling), Electrical Sensor	<code>stop</code> , 1 h	Setpoint 26.0 °C. It can switch the chiller off for an hour.
4	Return pump	Pump, Electrical Sensor	<code>none</code>	Running at 100 %, 2850 rpm, 3.2 m³/h, 48.2 W. It cannot stop it.
5	Skimmer	On/Off Plug-in Unit, Electrical Sensor	<code>stop</code> , 30 min (owner's choice)	On, 18.2 W. It can stop it for 30 minutes.
6	Ozone	On/Off Plug-in Unit, Electrical Sensor	<code>stop</code> , 1 h	Off. It can stop it, but never start it.
7	Kalkwasser	On/Off Plug-in Unit	<code>stop</code> , 1 h	Off. It can stop it, but never start it.
8	Sump fan	Dimmable Plug-In Unit	<code>none</code>	At 45 % (level 114). It cannot stop it.
9	Wave pump	Pump	<code>none</code>	Running at 60 %, on mains power, with its backup battery full. It cannot stop it.
10	Moonlight	On/Off Light	<code>full</code> , 1 h	Off. It can switch it on or off for an hour, or up to 4 hours with <code>OnWithTimedOff</code> .
11, 12	Temperature, sump; Temperature, display	Temperature Sensor	—	25.12 °C and 25.31 °C.
13	Leak sensor under the stand	Water Leak Detector	—	No leak; a sensor fault if its wiring breaks.

The top-off unit, the feeder, the pH and ORP probes, and the two level switches are not exposed. Nor are the three switches the owner asked for, because none stays within Table 10 (Section 13.5):

- Feed A sends `feed` to the feeder, which no home app may start;
- the maintenance mode stops the return pump for an hour, four times as long as a home app may; and
- the return pump's function reads `night`, and slows the pump while it is on, which no home app may do to a return pump.

The controller tells the owner so. A feed mode that only paused the return pump, the wave pump, and the skimmer, for no more than 15 minutes, would be offered.

At bedtime, a “good night” routine turns everything off. The return pump, the wave pump, and the fan keep running, and the bridge refuses the commands sent to them alone. The heater, chiller, and kalkwasser stirrer stop for an hour, the skimmer for 30 minutes, and the moonlight for an hour, each then returning to its automation. The ozone generator stays stopped by the owner’s own override, which a home app neither shortens nor ends. If the routine runs again an hour later, the bridge refuses its stops until each output has run for its minimum run time: an hour for the heater, the chiller, and the skimmer, and 4 hours for the moonlight. The kalkwasser stirrer, whose equipment Fail-Safe Behavior stops on loss of contact, has no minimum run time, and stops again for an hour. If a leak starts the leak emergency mode meanwhile, it takes precedence over all of these overrides.

The wave pump’s device can run on a backup battery, so its bridged endpoint also has the Power Source cluster, which shows the battery on standby and full (Section 10.1). Its device lists `identify`, so a home app can make the pump show which one it is (Section 10.2); the devices of Device Classes revision 0.1 list nothing, and their endpoints report `IdentifyType None`.

Appendix C — Home Apps (Informative)

Home apps show different device types. A device type that an app does not show leaves that equipment out of that app; this Specification still uses the device type that describes the equipment, rather than a substitute that would mislead (Section 6.2). The maker of the first large home app in Table 16 publishes a list of the device types it supports, and “Listed” means that a device type is on that list. For the second, “reported” marks what others have reported.

Table 16 — Device types that large home apps showed in October 2026

Device type	First large home app	Second large home app	Home Assistant
Plug-in units, lights, Thermostat, Pump	Listed	Not checked	Supported
Temperature, Humidity, and Flow Sensor, Contact Sensor	Listed	Not checked	Supported
Water Leak Detector	Not listed	Supported (reported)	Supported
Water Valve	Not listed	Not accepted (reported, September 2026)	Supported
Generic Switch, Electrical Sensor	Not listed	Not checked	Supported

Appendix D — Open Issues (Informative)

Issues 1 to 21 were open in revision 0.1. The Project Lead decided issues 22 to 25 on 15 October 2026, accepting the recommendation in each. Issues 26 to 29 arose in revision 0.2, and the Project Lead decided issues 28 and 29 on 16 October 2026, accepting the recommendation in each. On 17 October 2026 the Project Lead froze the scope of revision 1.0, and decided issues 1 to 5, 7, 8, and 26; revision 0.3 applies the decisions, and resolves issue 10, which the Device Classes specification settled. Issue 30 records the choices that revision 0.3 made where those decisions left them open, which the Project Lead accepted on 18 October 2026. Issues 6, 9, 11 to 13, and 16 to 20 are deferred beyond revision 1.0, and are tracked outside this Specification. Issue 5 stays open until the lawyer’s review of the project’s scope, issues 14 and 15 wait on checks against the Matter specification’s text, and issue 21 on owners’ experience of home apps. Each issue says how a revision resolves it, or why it stays open. Issue numbers do not change, because other specifications refer to them.

Questions for the Project Lead

1. **Matter as a normative reference.** *Resolved in revision 0.3, as the Project Lead decided on 17 October 2026.* The Matter specification is free of charge and available without membership, which meets the project’s rule for normative references, but it is not openly licensed: it may not be republished [Matter], and the SDK’s machine-readable data model may be used only internally [SDK]. The alternatives were to cite

Matter informatively and define the bridge's behavior entirely here, which cannot be done without Matter's definitions, or to ask the Connectivity Standards Alliance for permission to publish the parts needed. The Project Lead decided to keep Matter 1.6.1 as a pinned normative reference, and Appendix A records why that is acceptable; the mapping file lists only identifiers and names.

2. **Which controls home apps may use by default.** *Resolved in revision 0.3, as the Project Lead decided on 17 October 2026: Table 10, as it stands.* Home apps cannot stop return pumps, circulation pumps, air pumps, filters, skimmers, UV sterilizers, or fans by default; they can stop heaters and chillers for at most an hour; they can stop, but never start, dosers, reactors, CO₂, ozone, feeders, RO/DI, water changers, and unknown equipment; and they have full control of lights and algae scrubbers for up to four hours. The alternative was a bridge that is read-only until the owner allows each control. Section 12.5.4 no longer calls the policies provisional; the durations stay provisional until owners' experience confirms them (issue 21).
3. **The bridge's default role.** *Resolved in revision 0.3, as the Project Lead decided on 17 October 2026.* The owner chooses the role, and the controller proposes `override` (Section 7.2.3). With `view`, home apps could only watch.
4. **Leak sensors with a wiring fault.** *Resolved in revision 0.3, as the Project Lead decided on 17 October 2026.* Matter cannot show a leak as unknown, so `StateValue` keeps its last value and `SensorFault` is set (Section 10.5.3). Showing a leak instead would make home automations act on a fault, for example by closing a house's water valve; keeping the last value hides the fault only from apps that do not show `SensorFault`.
5. **Matter's license.** *Open: the Project Lead decided on 17 October 2026 to keep the mapping file and CI's check as they are, and to include the question in the lawyer's review of the project's scope, an owner action; it stays open until that review.* Whether listing Matter identifiers and names in an Apache-2.0 file, and downloading the SDK's data model files to check them, are consistent with the Connectivity Standards Alliance's terms. Both are our reading of common practice, not legal advice.
6. **Where the bridge runs.** *Resolved in revision 0.3, as the Project Lead decided on 17 October 2026.* This Specification allows a bridge on the controller or on another host, which reaches the controller through the Controller API [OAP-CA] (Section 7.1). Which the reference controller uses is the reference design's choice, not this Specification's, and waits for measurements of a bridge prototype on the controller [P0001, Open Question 10].

For other specifications

8. **Security and Access: Matter.** *Resolved in revision 0.3, as the Project Lead decided on 17 October 2026, with Security and Access revision 0.3.* One service user per bridge, whose role caps every fabric; fabrics are not users, and have no roles of their own (Sections 7.3 and 12.1) [OAP-SA Section 13.4.4]. Records of what the bridge does carry `via matter`, with the fabric in `details` (Sections 7.3.3 and 12.1.2) [OAP-SA Section 12.1.6]. Turning the bridge on, opening its commissioning window, and removing a fabric need `manage-users`, with no operation of their own (Sections 7.2 and 7.3.2).
9. **Device Classes: batteries and identify.** *Resolved in revision 0.3.* Device Classes revision 0.2 added `powerSource`, `batteryLevel`, `batteryVoltage`, and the fault `battery-weak`, and lists each endpoint's optional elements, `identify` among them (its Appendix D, issues 11 and 23). The bridged endpoints of a device that can run on a battery now have the Power Source cluster (Section 10.1), and `IdentifyType` comes from each endpoint's list of optional elements (Section 10.2).

Details of this Specification

14. **Endpoint numbers.** *Open: to be checked against the Matter specification's text.* Section 7.4.5 never reuses a number, following the SDK's bridge example; what the Matter specification's text requires was not checked. The SDK's data model, which revision 0.3 checked, does not say.
15. **Unverified Matter details.** *Open, in part: to be checked against the Matter specification's text.* Revision 0.3 checked these against the SDK's machine-readable data model of Matter 1.6.1 [SDK], which follows the specification's text:
 - `PowerFactor` is an `int64` from -10000 to 10000, the range that a scale of 10000 to one gives;
 - the `TREE` feature of Power Topology covers an endpoint and its child endpoints, so it fits one outlet's bridged endpoint, which has none;

- `UniqueID` is a string of at most 32 characters, which the 32 hexadecimal digits of Section 7.5 fit;
- whether a bridged device gives `VendorID` and `ProductID` is described in the specification's text, not in the data model; and
- the Bridged Node device type has the Power Source cluster, with the attributes and the `BAT` feature that Section 10.1 uses, under a condition, `BridgedPowerSourceInfo`, whose meaning is in the text; Section 10.1 takes it to hold for a device that can run on a battery.

Still to check in the text: the Level Control semantics of the Pump device type (0.5 % steps), the unit of `SetpointRaiseLower's Amount` (0.1 °C), the scale of `PowerFactor`, whether Matter allows `UniqueID` to be a keyed hash, and when bridged devices should give `VendorID` and `ProductID`. The code of `BUSY, 0x9C`, matches the status codes of the SDK [SDK].

16. **Numbers.** *Open: to be confirmed with owners' experience of home apps.* The durations of Table 10, the 5 s limits of Sections 7.2 and 7.3, and the virtual outputs' durations are provisional. The minimum run times follow the longest durations, by the rule that issue 28 decided.

Decided by the Project Lead

22. **Modes and virtual outputs.** *Resolved in revision 0.2; extended in revision 0.3 by the Project Lead's decision of 17 October 2026.* The Project Lead decided on 15 October 2026 that a bridge offers a mode or a virtual output only if its effect on every output stays within Table 10. Section 13.5 says what that means: a mode with a `duration` that stops only what home apps may stop, runs or commands only what they may run, and holds each output no longer than its longest duration; and a virtual output on which only outputs that home apps may run depend, whose overrides last no longer than theirs. The bridge stops offering a switch that no longer qualifies after the configuration changes, and the controller tells the owner why. So the example's feed mode, which feeds, its maintenance mode, which stops the return pump for an hour, and its `night` virtual output, which slows the return pump, are no longer offered (Appendix B). On 17 October 2026 the Project Lead decided that a bridge also counts what a mode does through triggers and conditions; revision 0.3 counts every output that depends on a mode, as one may depend on a virtual output, and checks scenes the same way (Sections 13.5.1, 13.5.2, and 13.5.6).
23. **Repeated stops.** *Resolved in revision 0.2, with the minimum run times that issue 28 decided; extended in revision 0.3 by the Project Lead's decision of 17 October 2026.* The Project Lead decided on 15 October 2026 that a new Matter stop of an output never extends past the end that the first stop allowed, and that the output then runs for a minimum time before the next Matter stop. Section 12.6 says so: a chain of Matter stops ends no later than its first stop's start plus the output's longest duration, and, after a Matter stop ends, the bridge refuses another until the output has run for its minimum run time, a column of Table 10. On 17 October 2026 the Project Lead decided that this holds per output, across direct stops, modes, scenes, and virtual outputs; that a stop of an output that a water change is using is never refused; and that Section 12.6.5 does not apply to categories that stop on loss of contact and whose longest duration is a day. Revision 0.3 counts every stop that a home app causes through a switch as a Matter stop of each output it stops (Sections 12.6.7 and 13.6), and Table 10 gives those categories no minimum run time (Section 12.5.3).
24. **Water changes.** *Resolved in revision 0.2.* The Project Lead decided on 15 October 2026 that a Matter stop of an output that a water change is using aborts the water change, and that only an Open Aquarium app resumes it. Section 12.11 says so, for the outputs of a water change's `drain.outputs` and `fill.outputs` from the start of its drain to the end of its settling, and Table 10's reason for `water-changer` cites it. The Water Level and Water Change specification says the same in its Section 9.9 (its Appendix D, issue 28). Revision 0.3 adds that such a stop is never refused for the output's minimum run time (Section 12.11.4).
25. **Commissioning windows.** *Resolved in revision 0.2.* The Project Lead decided on 15 October 2026 that fabric administrators may open commissioning windows through Matter's own mechanism, and that Section 7.2.4 applies to the windows the bridge opens itself. Sections 7.2.4 and 7.3.3 say so. The controller records each fabric added or removed in its audit log, with the actions `fabric-added` and `fabric-removed` that Security and Access defines (its Table 8), and notifies every owner, with the notices of Alarms and Notifications (its Section 13.4.4).

The following issues arose in revision 0.2:

26. **A bridge on another host.** *Resolved in revision 0.3, as the Project Lead decided on 17 October 2026.* Section 7.1.2 had a bridge on another host act only through the controller, but no specification defined how a party reads a controller's automation, sets overrides, or starts modes and scenes through it. The Controller API specification now defines it [OAP-CA], and a bridge on another host uses it (Section 7.1.2).
27. **Alarm outputs.** *Resolved in revision 0.2, with Data Model revision 0.3.* The Project Lead decided on 15 October 2026 that alarm outputs have the equipment category `alarm-signal` (Fail-Safe Behavior Appendix C, issue 10) [OAP-AN Section 15.2]. Data Model revision 0.3 adds it, and Fail-Safe Behavior Table 3 recommends for it. Table 10 gives it the policy `none`, and lets the owner choose only `none`, so that home apps neither sound nor silence an alarm.
28. **Minimum run times.** *Decided by the Project Lead on 16 October 2026; resolved in revision 0.2.* Revision 0.2 proposed, for each category, a minimum run time equal to its longest duration, and at least an hour, so that repeated Matter stops never keep an output stopped for more than half the time: an hour for pumps, skimmers, filters, UV sterilizers, heaters, and chillers; 4 hours for lights, algae scrubbers, and top-off units; and a day for the rest. The hour for return pumps and other equipment that must keep running gives at most one pause of 15 minutes an hour. The 4 hours for lights may surprise an owner who switches a tank's lights off a second time after they have come back on; a shorter time for lights, such as an hour, would still keep them from staying dark. An alternative was a budget of Matter stops a day for each output, which is harder to explain to owners. The Project Lead decided on the rule of Section 12.5.3, with the values of Table 10, the same for every category, lights included. An owner who switches the lights off again before they come back on keeps them off for up to 4 hours without a refusal, because each stop continues the chain (Section 12.6.4); a refusal follows only a stop that has ended, and keeps a routine from holding a tank dark for more than half the time. On 17 October 2026 the Project Lead decided that categories that stop on loss of contact and whose longest duration is a day have no minimum run time, because stopping them is their safe side (issue 23). The longest durations stay provisional (issue 21), and each minimum run time changes with its category's.
29. **Modes and the owner's own policies.** *Decided by the Project Lead on 16 October 2026; resolved in revision 0.2.* A mode or virtual output stays within Table 10 when it does no more to each output than a policy that Table 10 lets the owner choose for that output would allow (Section 13.5.1), not only the policy the owner has chosen. So an owner can offer a feed mode that pauses the return pump for 10 minutes while the pump's own policy is `none`, so that "turn everything off" never stops it, though "turn everything on" starts the feed mode. The alternative was to require each output's own policy to allow what the mode does, which would make an owner who wants a feed switch also let "turn everything off" pause the return pump. The Project Lead decided to keep what Table 10 lets the owner choose, which still bounds every stop by the output's longest duration, and to have the controller show the owner each output that a mode or virtual output would do more to than its own policy allows, and offer the switch only after the owner confirms (Section 13.5.5).

The following issue records the choices of revision 0.3:

30. **Choices of revision 0.3.** *Resolved in revision 0.3: the Project Lead accepted each choice on 18 October 2026.* Revision 0.3 made these choices where the decisions it applies left them open:
- every output that depends on a mode, through a trigger or a condition, must be one that home apps may run, because what a function or trigger does with the mode's state can be anything, rather than the bridge reading what each trigger does (Section 13.5.1);
 - a latch between a mode or a virtual output and an output keeps the switch from being offered, because the latch can keep acting after the switch ends, and a timer adds its duration (Section 13.5.7);
 - a home app's `On` for a switch that would stop an output that may not take a Matter stop now is refused, rather than carried out without that output, which would change what the switch does (Section 13.6.2);
 - a stop of an output that a water change is using is never refused for the output's minimum run time, while the bridge's role, the output's policy, and the layers above overrides still apply (Section 12.11.4);
 - modes that a person started are protected from home apps, and scenes are not, as the decision covers modes (Section 12.4.4);

- a device's battery is one Power Source cluster with the BAT feature only, `battery-weak` sets `BatChargeLevel` to `Warning`, `BatReplacementNeeded` stays `false`, and mains presence is not shown (Section 10.1); and
- an endpoint that lists `identify` reports `IdentifyType VisibleIndicator` (Section 10.2).