INTERNATIONAL STANDARD

ISO/IEC 30118-6

Second edition
2021-10

Information technology — Open Connectivity Foundation (OCF) Specification —

Part 6: Resource to AllJoyn interface mapping specification

Technologies de l'information — Specification de la Fondation pour la connectivité ouverte (Fondation OCF) —

Partie 6: Spécification du mapping entre les ressources et l'interface AllJoyn

Reference number
ISO/IEC 30118-6:2021(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by the Open Connectivity Foundation (OCF) (as OCF Resource to AllJoyn Interface Mapping Specification, version 2.2.0) and drafted in accordance with its editorial rules. It was adopted, under the JTC 1 PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

This second edition cancels and replaces the first edition (ISO/IEC 30118-6:2018), which has been technically revised.

The main changes compared to the previous edition are as follows:

— AllJoyn text moved from the bridging specification to this document;

— addition of clarifications throughout.

A list of all parts in the ISO/IEC 30118 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

This document, and all the other parts associated with this document, were developed in response to worldwide demand for smart home focused Internet of Things (IoT) devices, such as appliances, door locks, security cameras, sensors, and actuators; these to be modelled and securely controlled, locally and remotely, over an IP network.

While some inter-device communication existed, no universal language had been developed for the IoT. Device makers instead had to choose between disparate frameworks, limiting their market share, or developing across multiple ecosystems, increasing their costs. The burden then falls on end users to determine whether the products they want are compatible with the ecosystem they bought into, or find ways to integrate their devices into their network, and try to solve interoperability issues on their own.

In addition to the smart home, IoT deployments in commercial environments are hampered by a lack of security. This issue can be avoided by having a secure IoT communication framework, which this standard solves.

The goal of these documents is then to connect the next 25 billion devices for the IoT, providing secure and reliable device discovery and connectivity across multiple OSs and platforms. There are multiple proposals and forums driving different approaches, but no single solution addresses the majority of key requirements. This document and the associated parts enable industry consolidation around a common, secure, interoperable approach.

ISO/IEC 30118 consists of eighteen parts, under the general title Information technology — Open Connectivity Foundation (OCF) Specification. The parts fall into logical groupings as described herein:

- Core framework
  - Part 1: Core Specification
  - Part 2: Security Specification
  - Part 13: Onboarding Tool Specification
- Bridging framework and bridges
  - Part 3: Bridging Specification
  - Part 6: Resource to Alljoyn Interface Mapping Specification
  - Part 8: OCF Resource to oneM2M Resource Mapping Specification
  - Part 14: OCF Resource to BLE Mapping Specification
  - Part 15: OCF Resource to EnOcean Mapping Specification
  - Part 16: OCF Resource to UPlus Mapping Specification
  - Part 17: OCF Resource to Zigbee Cluster Mapping Specification
  - Part 18: OCF Resource to Z-Wave Mapping Specification
- Resource and Device models
  - Part 4: Resource Type Specification
  - Part 5: Device Specification

- Core framework extensions
  - Part 7: Wi-Fi Easy Setup Specification
  - Part 9: Core Optional Specification
- OCF Cloud
  - Part 10: Cloud API for Cloud Services Specification
  - Part 11: Device to Cloud Services Specification
  - Part 12: Cloud Security Specification

Information technology — Open Connectivity Foundation (OCF) —

Part 6:
Resource to AllJoyn interface mapping specification

1 Scope

This document provides detailed mapping information to provide equivalency between AllJoyn defined Interfaces and OCF defined Resources.

This document provides mapping for Device Types (AllJoyn to/from OCF), identifies equivalent OCF Resources for both mandatory and optional AllJoyn interfaces and for each interface defines the detailed Property by Property mapping using OCF defined extensions to JSON schema to programmatically define the mappings.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 30118-1 *Information technology -- Open Connectivity Foundation (OCF) Specification -- Part 1: Core specification*
https://www.iso.org/standard/53238.html
Latest version available at: https://openconnectivity.org/specs/OCF_Core_Specification.pdf

ISO/IEC 30118-2 *Information technology -- Open Connectivity Foundation (OCF) Specification -- Part 2: Security specification*
https://www.iso.org/standard/74239.html
Latest version available at: https://openconnectivity.org/specs/OCF_Security_Specification.pdf

ISO/IEC 30118-4 *Information technology -- Open Connectivity Foundation (OCF) Specification -- Part 4: Resource type specification*
https://www.iso.org/standard/74241.html
Latest version available at:
https://openconnectivity.org/specs/OCF_Resource_Type_Specification.pdf

ISO/IEC 30118-5 *Information technology – Open Connectivity Foundation (OCF) Specification – Part 5: Smart home device specification*
https://www.iso.org/standard/74242.html
Latest version available at: https://openconnectivity.org/specs/OCF_Device_Specification.pdf

JSON Hyper-Schema, *JSON Hyper-Schema: A Vocabulary for Hypermedia Annotation of JSON*, October 2016
http://json-schema.org/latest/json-schema-hypermedia.html

Derived Models for Interoperability between IoT Ecosystems, Stevens & Merriam, March 2016
https://www.iab.org/wp-content/IAB-uploads/2016/03/OCF-Derived-Models-for-Interoperability-Between-IoT-Ecosystems_v2-examples.pdf

IETF RFC 4122, *A Universally Unique Identifier (UUID) URN Namespace*, July 2005
https://www.rfc-editor.org/info/rfc4122

IETF RF 4648, *The Base16, Base32 and Base64 Data Encodings*, October 2006
https://www.rfc-editor.org/info/rfc4648

IETF RFC 6973, *Privacy Considerations for Internet Protocols*, July 2013
https://www.rfc-editor.org/info/rfc6973

IETF RFC 7159, *The JavaScript Object Notation (JSON) Data Interchange Format*, March 2014
https://www.rfc-editor.org/info/rfc7159

AllJoyn Common Data Model Interface Definitions
https://wiki.alljoyn.org/cdm

AllJoyn About Interface Specification, *About Feature Interface Definitions*, Version 14.12
https://allseenalliance.org/framework/documentation/learn/core/about-announcement/interface

AllJoyn Configuration Interface Specification, *Configuration Interface Definition*, Version 14.12
https://allseenalliance.org/framework/documentation/learn/core/configuration/interface

D-Bus Specification, *D-Bus Specification*
https://dbus.freedesktop.org/doc/dbus-specification.html

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 30118-1 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

– ISO Online browsing platform: available at https://www.iso.org/obp

– IEC Electropedia: available at http://www.electropedia.org/

4 Document conventions and organization

4.1 Conventions

In this document a number of terms, conditions, mechanisms, sequences, parameters, events, states, or similar terms are printed with the first letter of each word in uppercase and the rest lowercase (e.g., Network Architecture). Any lowercase uses of these words have the normal technical English meaning.

In this document, to be consistent with the IETF usages for RESTful operations, the RESTful operation words CRUDN, CREATE, RETRIVE, UPDATE, DELETE, and NOTIFY will have all letters capitalized. Any lowercase uses of these words have the normal technical English meaning.

4.2 Notation

In this document, features are described as required, recommended, allowed or DEPRECATED as follows:

Required (or shall or mandatory).

These basic features shall be implemented to comply with the Mapping Specification. The phrases "shall not", and "PROHIBITED" indicate behaviour that is prohibited, i.e. that if performed means the implementation is not in compliance.

Recommended (or should).

These features add functionality supported by the Mapping Specification and should be implemented. Recommended features take advantage of the capabilities the Mapping Specification, usually without imposing major increase of complexity. Notice that for compliance testing, if a recommended feature is implemented, it shall meet the specified requirements to be in compliance with these guidelines. Some recommended features could become requirements in the future. The phrase "should not" indicates behaviour that is permitted but not recommended.

Allowed (or allowed).

These features are neither required nor recommended by the Mapping Specification, but if the feature is implemented, it shall meet the specified requirements to be in compliance with these guidelines.

Conditionally allowed (CA)

The definition or behaviour depends on a condition. If the specified condition is met, then the definition or behaviour is allowed, otherwise it is not allowed.

Conditionally required (CR)

The definition or behaviour depends on a condition. If the specified condition is met, then the definition or behaviour is required. Otherwise the definition or behaviour is allowed as default unless specifically defined as not allowed.

DEPRECATED

Although these features are still described in this document, they should not be implemented except for backward compatibility. The occurrence of a deprecated feature during operation of an implementation compliant with the current document has no effect on the implementation's operation and does not produce any error conditions. Backward compatibility may require that a feature is implemented and functions as specified but it shall never be used by implementations compliant with this document.

Strings that are to be taken literally are enclosed in "double quotes".

Words that are emphasized are printed in *italic*.

5 Theory of operation

5.1 Interworking approach

The interworking between AllJoyn defined interfaces and OCF defined Resource Types is modelled using the derived model syntax described in Derived Models for Interoperability between IoT Ecosystems. Determination of the minimum set of AllJoyn interfaces for which equivalency is required within the OCF data model was done by listing the set of interfaces required for each of the device

types defined by the CDM Project inside of AllJoyn. Where the AllJoyn interface supports methods then an actuation design pattern is applied.

5.2 Mapping syntax

5.2.1 Introduction

Within the defined syntax for derived modelling used by this document there are two blocks that define the actual Property-Property equivalence or mapping. These blocks are identified by the keywords "x-to-ocf" and "x-from-ocf". Derived Models for Interoperability between IoT Ecosystems does not define a rigid syntax for these blocks; they are free form string arrays that contain pseudo-coded mapping logic.

Within this document we apply the rules defined in clause 5.2 to these blocks to ensure consistency and re-usability and extensibility of the mapping logic that is defined.

5.2.2 General

All statements are terminated with a carriage return.

5.2.3 Value assignment

The equals sign (=) is used to assign one value to another. The assignee is on the left of the operator; the value being assigned on the right.

5.2.4 Property naming

All Property names are identical to the name used by the original model; for example, from the OCF Temperature Resource the Property name "temperature" is used whereas when referred to the derived ecosystem then the semantically equivalent Property name is used.

When the same name is used by both OCF and the derived ecosystem for semantically equivalent values then the name of the OCF defined Property is prepended by the ecosystem designator "ocf" to avoid ambiguity (e.g. "ocf.step").

5.2.5 Arrays

An array element is indicated by the use of square brackets "[]" with the index of the element contained therein, e.g. range[1]. All arrays start at an index of 0. If an entire array is being referenced then no index is included, e.g. selectablehumiditylevels[].

5.2.6 Default mapping

There are cases where the specified mapping is not possible as one or more of the Properties being mapped is optional in the source model. In all such instances a default mapping is provided. The default map is indicated by the prepending of an "otherwise:" modifier to the assignment. (e.g. "otherwise: step = 1").

5.2.7 Conditional mapping

When a mapping is dependent on the meeting of other conditions then the syntax:

if "condition", "mapping".

Is applied.

E.g. if step >0, ocf.step = step.

5.2.8 Loops

When a mapping can be represented by a repeated loop governed by some condition then the syntax:

for "initialize", "condition", "increment": "mapping"

Where:

"initialize" is an initial local loop control variable setting.

"condition" is the loop controller, the loop repeats until the condition evaluates to "false".

"increment" allows for update of the control variable, if omitted an increment of "1" is assumed.

Is applied.

E.g. for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] = modearray[supportedmodes[x]]

5.2.9 Method invocation

The invocation of a method or remote procedure call (RPC) from the derived ecosystem as part of the mapping from an OCF Resource is indicated by the use if a double colon "::" delimiter between the applicable resource, service, interface or other construct identifier and the method or RPC name. The method name always includes trailing parentheses which would include any parameters should they be passed.

For example, when dealing with the switchon() method from AllJoyn this gives a complete method invocation as: operation.oncontrol::switchon().

6 AllJoyn translation

6.1 Operational scenarios

The overall goals are to:

1) make Bridged Servers appear to OCF clients as if they were native OCF servers, and

2) make OCF servers appear to Bridged Clients as if they were native non-OCF servers.

6.2 Requirements specific to an AllJoyn bridging function

6.2.1 Introduction

The Bridge Platform shall be an AllJoyn Router Node. (This is a requirement so that users can expect that a certified Bridge will be able to talk to any AllJoyn device, without the user having to buy some other device.)

The requirements in clause 6.2 apply when using algorithmic translation, and by default apply to deep translation unless the relevant clause for such deep translation specifies otherwise.

6.2.2 Use of introspection

Whenever possible, the translation code should make use of metadata available that indicates what the sender and recipient of the message in question are expecting. For example, devices that are AllJoyn Certified are required to carry the introspection data for each object and interface they expose. When the metadata is available, Bridging Functions should convert the incoming payload to exactly

the format expected by the recipient and should use information when translating replies to form a more useful message.

For example, for an AllJoyn specific Bridging Function, the expected interaction list is presented in Table 1.

Table 1 – AllJoyn Bridging Function Interaction List

Message Type	Sender	Receiver	Metadata
Request	AllJoyn 16.10	OCF 1.0	Available
Request	OCF 1.0	AllJoyn 16.10	Available
Response	AllJoyn 16.10	OCF 1.0	Available
Response	OCF 1.0	AllJoyn 16.10	Available

6.2.3 Stability and loss of data

Round-tripping through the translation process specified in this document is not expected to reproduce the same original message. The process is, however, designed not to lose data or precision in messages, though it should be noted that both OCF and AllJoyn payload formats allow for future extensions not considered in this document.

However, a third round of translation should produce the same identical message as was previously produced, provided the same information is available. That is, in the chain shown in Figure 1, payloads 2 and 4 as well as 3 and 5 should be identical.

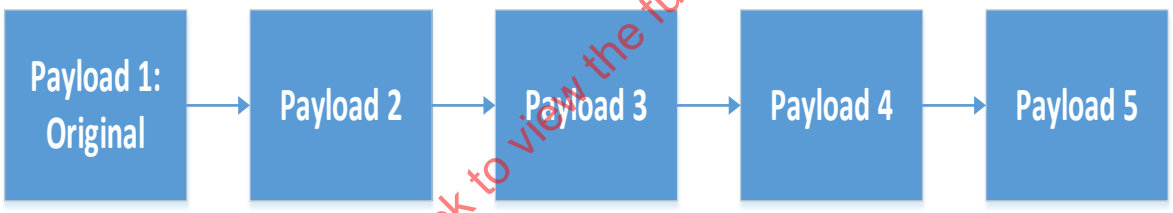

Figure 1 – Payload Chain

6.2.4 Exposing AllJoyn producer devices to OCF clients

6.2.4.1 Virtual OCF devices and resources

As specified in ISO/IEC 30118-2 the value of the "di" property of OCF Devices (including VODs) shall be established as part of Onboarding of that VOD.

Each AllJoyn object shall be mapped to one or more Virtual OCF Resources. If all AllJoyn interfaces can be translated to resource types on the same resource, there should be a single Virtual OCF Resource, and the path component of the URI of the Virtual OCF Resource shall be the AllJoyn object path, where each "_h" in the AllJoyn object path is transformed to "-" (hyphen), each "_d" in the AllJoyn object path is transformed to "." (dot), each "_t" in the AllJoyn object path is transformed to "~" (tilde), and each "_u" in the AllJoyn object path is transformed to "_" (underscore). Otherwise, a Resource with that path shall exist with a Resource Type of ["oic.wk.col", "oic.r.alljoynobject"] which is a Collection of links, where "oic.r.alljoynobject" is defined in clause 10.2 and the items in the collection are the Resources with the translated Resource Types.

The value of the "piid" property of "/oic/d" for each VOD shall be the value of the OCF-defined AllJoyn field "org.openconnectivity.piid" in the AllJoyn About Announce signal, if that field exists, else it shall be calculated by the Bridging Function as follows:

– If the AllJoyn device supports security, the value of the "piid" property value shall be the peer GUID.

– If the AllJoyn device does not support security but the device is being bridged anyway (see 10.2), the "piid" property value shall be derived from the DeviceId and AppId properties (in the About data), by concatenating the DeviceId value (not including any null termination) and the AppId bytes and using the result as the "name" to be used in the algorithm specified in IETF RFC 4122 clause 4.3, with SHA-1 as the hash algorithm, and 8f0e4e90-79e5-11e6-bdf4-0800200c9a66 as the name space ID. (This is to address the problem of being able to de-duplicate AllJoyn devices exposed via separate OCF Bridge Devices.)

A Bridging Function implementation is encouraged to listen for AllJoyn About Announce signals matching any AllJoyn interface name. It can maintain a cache of information it received from these signals, and use the cache to quickly handle "/oic/res" queries from OCF Clients (without having to wait for Announce signals while handling the queries).

A Bridging Function implementation is encouraged to listen for other signals (including EmitsChangedSignal of properties) only when there is a client subscribed to a corresponding resource on a Virtual AllJoyn Device.

There are multiple types of AllJoyn interfaces, which shall be handled as follows.

1) If the AllJoyn interface is in a well-defined set (defined in 6.2.4.2) of interfaces where standard forms exist on both the AllJoyn and OCF sides, the Bridging Function shall either:

   a) follow the requirements for translating that interface specially, or

   b) not translate the AllJoyn interface.

2) If the AllJoyn interface is not in the well-defined set, the Bridging Function shall either:

   a) not translate the AllJoyn interface, or

   b) algorithmically map the AllJoyn interface as specified in 6.3 to custom/vendor-defined Resource Types by converting the AllJoyn interface name to OCF resource type name(s).

An AllJoyn interface name shall be converted to a Device Type or a set of one or more OCF Resource Types as follows:

3) If the AllJoyn interface has any members, append a suffix ".<seeBelow>" where <seeBelow> is described in this clause.

4) For each upper-case letter present in the entire string, replace it with a hyphen followed by the lower-case version of that letter (e.g., convert "A" to "-a").

5) If an underscore appears followed by a (lower-case) letter or a hyphen, for each such occurrence, replace the underscore with two hyphens (e.g., convert "_a" to "--a", "_-a" to "---a").

6) For each underscore remaining, replace it with a hyphen (e.g., convert "_1" to "-1").

7) Prepend the "x." prefix.

Some examples are shown in Table 2. The first three are normal AllJoyn names converted to unusual OCF names. The last three are unusual AllJoyn names converted (perhaps back) to normal OCF names. ("xn--" is a normal domain name prefix for the Punycode-encoded form of an Internationalized Domain Name, and hence can appear in a normal vendor-specific OCF name.)

Table 2 – AllJoyn to OCF Name Examples

From AllJoyn name	To OCF name
example.Widget	x.example.-widget
example.my__widget	x.example.my----widget
example.My_Widget	x.example.-my---widget
xn_p1ai.example	x.xn--p1ai.example
xn__90ae.example	x.xn--90ae.example
example.myName_1	x.example.my-name-1

Each AllJoyn interface that has members and is using algorithmic mapping shall be mapped to one or more Resource Types as follows:

– AllJoyn Properties with the same EmitsChangedSignal value are mapped to the same Resource Type where the value of the <seeBelow> label is the value of EmitsChangedSignal. AllJoyn Properties with EmitsChangedSignal values of "const" or "false", are mapped to Resources that are not Observable, whereas AllJoyn Properties with EmitsChangedSignal values of "true" or "invalidates" result in Resources that are Observable. The Version property in an AllJoyn interface is always considered to have an EmitsChangedSignal value of "const", even if not specified in introspection XML. The name of each property on the Resource Type shall be "<ResourceType>.<AllJoynPropertyName>", where each "_d" in the <AllJoynPropertyName> is transformed to "." (dot), and each "_h" in the <AllJoynPropertyName> is transformed to "-" (hyphen).

– Resource Types mapping AllJoyn Properties with access "readwrite" shall support the "oic.if.rw" OCF Interface. Resource Types mapping AllJoyn Properties with access "read" shall support the "oic.if.r" OCF Interface. Resource Types supporting both the "oic.if.rw" and "oic.if.r" OCF Interfaces shall choose "oic.if.r" as the default Interface.

– Each AllJoyn Method is mapped to a separate Resource Type, where the value of the <seeBelow> label is the AllJoyn Method name. The Resource Type shall support the "oic.if.rw" OCF Interface. Each argument of the AllJoyn Method shall be mapped to a separate Property on the Resource Type, where the name of that Property is prefixed with "<ResourceType>arg<#>", where <#> is the 0-indexed position of the argument in the AllJoyn introspection xml, in order to help get uniqueness across all Resource Types on the same Resource. Therefore, when the AllJoyn argument name is not specified, the name of that property is "<ResourceType>arg<#>", where <#> is the 0-indexed position of the argument in the AllJoyn introspection XML. In addition, that Resource Type has an extra "<ResourceType>validity" property that indicates whether the rest of the properties have valid values. When the values are sent as part of an UPDATE response, the validity property is true, and any other properties have valid values. In a RETRIEVE (GET or equivalent in the relevant transport binding) response, the validity property is false, and any other properties can have meaningless values. If the validity property appears in an UPDATE request, its value shall be true (a value of false shall result in an error response).

– Each AllJoyn Signal (whether sessionless, sessioncast, or unicast) is mapped to a separate Resource Type on an Observable Resource, where the value of the <seeBelow> label is the AllJoyn Signal name. The Resource Type shall support the "oic.if.r" OCF Interface. Each argument of the AllJoyn Signal is mapped to a separate Property on the Resource Type, where the name of that Property is prefixed with "<ResourceType>arg<#>", where <#> is the 0-indexed position of the argument in the AllJoyn introspection xml, in order to help get uniqueness across all Resource Types on the same Resource. Therefore, when the AllJoyn argument name is not specified, the name of that property is "<ResourceType>arg<#>", where <#> is the 0-indexed position of the argument in the AllJoyn introspection XML. In addition, that Resource Type has an extra "<ResourceType>validity" property that indicates whether the rest of the properties have valid values. When the values are sent as part of a NOTIFY response, the validity property is true, and any other properties have valid values. In a RETRIEVE (GET or equivalent in the relevant transport binding) response, the validity property is false, and any other properties returned can have meaningless values. This is because in AllJoyn, the signals are instantaneous events, and the

values are not necessarily meaningful beyond the lifetime of that message. Note that AllJoyn does have a TTL field that allows store-and-forward signals, but such support is not required in OCF 1.0. We expect that in the future, the TTL may be used to allow valid values in response to a RETRIEVE that is within the TTL.

When an algorithmic mapping is used, AllJoyn data types shall be mapped to OCF property types according to 6.3.

If an AllJoyn operation fails, the Bridging Function shall send an appropriate OCF error response to the OCF client. If an AllJoyn error name is available and does not contain the "org.openconnectivity.Error.Code" prefix, it shall construct an appropriate OCF error message (e.g., diagnostic payload if using CoAP) from the AllJoyn error name and AllJoyn error message (if any), using the form "<error name>: <error message>", with the <error name> taken from the AllJoyn error name field and the <error message> taken from the AllJoyn error message, and the CoAP error code set to an appropriate value (if CoAP is used). If an AllJoyn error name is available and contains the "org.openconnectivity.Error.Code" prefix, the OCF error message (e.g., diagnostic payload if using CoAP) should be taken from the AllJoyn error message (if any), and the CoAP error code (if CoAP is used) set to a value derived as follows; remove the "org.openconnectivity.Error.Code" prefix, and if the resulting error name is of the form "<#>" where <#> is an error code without a decimal (e.g., "404"), the CoAP error code shall be the error code indicated by the "<#>". Example: "org.openconnectivity.Error.Code404" becomes "404", which shall result in an error 4.04 for a CoAP transport.

6.2.4.2 Exposing an AllJoyn producer application as a virtual OCF server

Table 3 shows how OCF Device properties, as specified in Table 27 in ISO/IEC 30118-1 shall be derived, typically from fields specified in the ISO/IEC 30118-1 and AllJoyn Configuration Interface Specification.

If the AllJoyn About or Config data field has a mapping rule defined (as in Table 3, Table 4, Table 5, and Table 6), the field name shall be translated based on that mapping rule; else if the AllJoyn About or Config data field has a fully qualified name (with a <domain> prefix (such as "com.example", "org.alljoyn"), the field name shall be translated based on the rules specified in 6.2.4 for mapping AllJoyn fields; else, the field shall not be translated as it may be incorrect (error) or it has no valid mapping (such as daemonRealm and passCode).

Table 3 – oic.wk.d resource type definition

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From AJ Field name	AJ Description	AJ Mandatory
(Device) Name	n	Human friendly name For example, "Bob's Thermostat"	Y	AppName (no exact equivalent exists)	Application name assigned by the app manufacturer (developer or the OEM).	Y
Spec Version	icv	Spec version of ISO/IEC 30118-1 this device is implemented to, the syntax is "core.major.minor"]	Y	(none)	Bridge Platform should return its own value	N
Device UUID	di	Unique identifier for Device. This value shall be as defined in ISO/IEC 30118-2 for DeviceID.	Y	(none)	Use as defined in ISO/IEC 30118-2	N

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From AJ Field name	AJ Description	AJ Mandatory
Protocol-Independent ID	piid	Unique identifier for OCF Device (UUID)	Y	org.openconnectivity.piid if it exists, else "Peer GUID" (not in About, but exposed by protocol) if authenticated, else Hash(DeviceId,AppId) where the Hash is done by concatenating the Device UUID (not including any null terminator) and the AppId and using the algorithm in IETF RFC 4122 clause 4.3, with SHA-1. This means that the value of di may change if the resource is read both before and after authentication, in order to mitigate privacy concerns discussed in RFC 6973.	Peer GUID: The peer GUID is the only persistent identity for a peer. Peer GUIDs are used by the authentication mechanisms to uniquely identify a remote application instance. The peer GUID for a remote peer is only available if the remote peer has been authenticated. DeviceId: Device identifier set by platform-specific means. AppId: A 128-bit globally unique identifier for the application. The AppId shall be a universally unique identifier as specified in **IETF RFC 4122**.	Per GUID: conditionally Y DeviceId: Y AppId: Y
Data Model Version	dmv	Spec version(s) of the vertical specifications this device data model is implemented to. The syntax is a comma separated list of "<vertical>.major.minor"]. <vertical> is the name of the vertical (i.e. sh for Smart Home)	Y	Comma separated list of the Version property values of each interface listed in the objectDescription argument of the Announce signal of About. In addition to the mandatory values specified in ISO/IEC 30118-1, additional values are formatted as "x.<interface name>.<Version property value>".	This document assumes that the value of the Version property is the same as the value of the "org.gtk.GDBus.Since" annotation of the interface in the AllJoyn introspection XML, and therefore the value of the Version property may be determined through introspection alone. Note that AllJoyn specifies that the default value is 1 if the "org.gtk.GDBus.Since" annotation is absent.	N, but required by IRB for all standard interfaces, and absence can be used to imply a constant (e.g., 0)

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From AJ Field name	AJ Description	AJ Mandatory
Localized Descriptions	ld	Detailed description of the Device, in one or more languages. This property is an array of objects where each object has a "language" field (containing an RFC 5646 language tag) and a "value" field containing the device description in the indicated language.	N	Description	Detailed description expressed in language tags as in RFC 5646.	Y
Software Version	sv	Version of the device software.	N	SoftwareVersion	Software version of the app.	Y
Manufacturer Name	dmn	Name of manufacturer of the Device, in one or more languages. This property is an array of objects where each object has a "language" field (containing an RFC 5646 language tag) and a "value" field containing the manufacturer name in the indicated language.	N	Manufacturer	The manufacturer's name of the app.	Y
Model Number	dmno	Model number as designated by manufacturer.	N	ModelNumber	The app model number.	Y

In addition, any additional vendor-defined fields in the AllJoyn About data shall be mapped to vendor-defined properties in the OCF Device resource "/oic/d" (which implements the "oic.wk.d" resource type), with a property name formed by prepending "x." to the AllJoyn field name.

Table 4 shows how OCF Device Configuration properties, as specified in Table 22 in ISO/IEC 30118-1 shall be derived:

Table 4 – oic.wk.con resource type definition

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From AJ Field name	AJ Description	AJ Mandatory
(Device) Name	n	Human friendly name For example, "Bob's Thermostat"	Y	AppName (no exact equivalent exists)	Application name assigned by the app manufacturer (developer or the OEM).	Y
Location	loc	Provides location information where available.	N	org.openconnectivity.loc (if it exists, else property shall be absent)		N
Location Name	locn	Human friendly name for location For example, "Living Room".	N	org.openconnectivity.locn (if it exists, else property shall be absent)		N
Currency	c	Indicates the currency that is used for any monetary transactions	N	org.openconnectivity.c (if it exists, else property shall be absent)		N

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From AJ Field name	AJ Description	AJ Mandatory
Region	r	Free form text Indicating the current region in which the device is located geographically. The free form text shall not start with a quote (").	N	org.openconnectivity.r (if it exists, else property shall be absent)		N
Localized Names	ln	Human-friendly name of the Device, in one or more languages. This property is an array of objects where each object has a "language" field (containing an RFC 5646 language tag) and a "value" field containing the device name in the indicated language. If this property and the Device Name (n) property are both supported, the Device Name (n) value shall be included in this array.	N	AppName	Application name assigned by the app manufacturer (developer or the OEM).	Y
Default Language	dl	The default language supported by the Device, specified as an RFC 5646 language tag. By default, clients can treat any string property as being in this language unless the property specifies otherwise.	N	DefaultLanguage	The default language supported by the device. Specified as an IETF language tag listed in RFC 5646.	Y

In addition, any additional vendor-defined fields in the AllJoyn Configuration data shall be mapped to vendor-defined properties in the OCF Configuration resource (which implements the "oic.wk.con" resource type and optionally the "oic.wk.con.p" resource type), with a property name formed by prepending "x." to the AllJoyn field name.

Table 5 shows how OCF Platform properties, as specified in Table 28 in ISO/IEC 30118-1 shall be derived, typically from fields specified in the AllJoyn About Interface Specification and AllJoyn Configuration Interface Specification.

Table 5 – oic.wk.p resource type definition

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From AJ Field name	AJ Description	AJ Mandatory
Platform ID	pi	Unique identifier for the physical platform (UIUID); this shall be a UUID in accordance with IETF RFC 4122. It is recommended that the UUID be created using the random generation scheme (version 4 UUID) specific in the RFC.	Y	DeviceId if it is a UUID, else generate a name-based UUID from the DeviceId using the DeviceId value (not including any null termination) as the "name" to be used in the algorithm specified in **IETF RFC 4122** clause 4.3, with SHA-1 as the hash algorithm, and 8f0e4e90-79e5-11e6-bdf4-0800200c9a66 as the name space ID.	Name of the device set by platform-specific means (such as Linux and Android).	Y
Manufacturer Name	mnmn	Name of manufacturer (not to exceed 16 characters)	Y	Manufacturer (in DefaultLanguage, truncated to 16 characters)	The manufacturer's name of the app.	Y
Manufacturer Details Link (URL)	mnml	URL to manufacturer (not to exceed 32 characters)	N	org.openconnectivity.mnml (if it exists, else property shall be absent)		N
Model Number	mnmo	Model number as designated by manufacturer	N	ModelNumber	The app model number.	Y
Date of Manufacture	mndt	Manufacturing date of device	N	DateOfManufacture	Date of manufacture using format YYYY-MM-DD (known as XML DateTime format).	N
Platform Version	mnpv	Version of platform – string (defined by manufacturer)	N	org.openconnectivity.mnpv (if it exists, else property shall be absent)		N
OS Version	mnos	Version of platform resident OS – string (defined by manufacturer)	N	org.openconnectivity.mnos (if it exists, else property shall be absent)		N
Hardware Version	mnhw	Version of platform hardware	N	HardwareVersion	Hardware version of the device on which the app is running.	N
Firmware version	mnfv	Version of device firmware	N	org.openconnectivity.mnfv (if it exists, else property shall be absent)		N
Support URL	mnsl	URL that points to support information from manufacturer	N	SupportUrl	Support URL (populated by the manufacturer)	N

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From AJ Field name	AJ Description	AJ Mandatory
SystemTime	st	Reference time for the device	N	org.openconnectivity.st (if it exists, else property shall be absent)		N
Vendor ID	vid	Vendor defined string for the platform. The string is freeform and up to the vendor on what text to populate it.	N	DeviceId	Name of the device set by platform-specific means (such as Linux and Android).	Y

Table 6 shows how OCF Platform Configuration properties, as specified in Table 23 in the ISO/IEC 30118-1 shall be derived:

Table 6 – oic.wk.con.p resource type definition

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From AJ Field name	AJ Description	AJ Mandatory
Platform Names	mnpn	Platform Identifier	N	DeviceName	Name of the device set by platform-specific means (such as Linux and Android).	Device name assigned by the user. The device name appears on the UI as the friendly name of the device.

In addition, the "oic.wk.mnt" properties Factory_Reset ("fr") and Reboot ("rb") shall be mapped to AllJoyn Configuration methods FactoryReset and Restart, respectively.

6.2.5 Exposing OCF resources to AllJoyn consumer applications

6.2.5.1 Use of AllJoyn producer application

Unless specified otherwise, each OCF resource shall be mapped to a separate AllJoyn object.

Each OCF Server shall be exposed as a separate AllJoyn producer application, with its own About data. This allows platform-specific, device-specific, and resource-specific fields to all be preserved across translation. However, this requires that AllJoyn Claiming of such producer applications be solved in a way that does not require user interaction, but this is left as an implementation issue.

The AllJoyn producer application shall implement the "oic.d.virtual" AllJoyn interface. This allows Bridge Platforms to determine if a device is already being translated when multiple Bridge Platforms are present. The "oic.d.virtual" interface is defined as follows:

```
<interface name="oic.d.virtual"/>
```

The implementation may choose to implement this interface by the AllJoyn object at path "/oic/d".

The AllJoyn peer ID shall be the OCF Device UUID ("di").

Unless specified otherwise, the AllJoyn object path shall be the OCF URI path, where each "-" (hyphen) in the OCF URI path is transformed to "_h", each "." (dot) in the OCF URI path is transformed to "_d",

each "~" (tilde) in the OCF URI path is transformed to "_t", and each "_" (underscore) in the OCF URI path is transformed to "_u".

The AllJoyn About data shall be populated per Table 8.

A Bridging Function implementation is encouraged to maintain a cache of OCF resources to handle the implementation of queries from the AllJoyn side, and emit an Announce Signal for each OCF Server. Specifically, the implementation could always Observe "/oic/res" changes and only Observe other resources when there is a client with a session on a Virtual AllJoyn Device.

There are multiple types of resources, which shall be handled as follows.

1) If the Resource Type is in a well-defined set (defined in 6.2.5.2) of resource types where standard forms exist on both the AllJoyn and OCF sides, the Bridging Function shall either:

   a) follow the requirements for translating that resource type specially, or

   b) not translate the Resource Type.

2) If the Resource Type is not in the well-defined set (but is not a Device Type), the Bridging Function shall either:

   a) not translate the Resource Type, or

   b) algorithmically map the Resource Type as specified in 6.3 to a custom/vendor-defined AllJoyn interface by converting the OCF Resource Type name to an AllJoyn Interface name.

An OCF Resource Type or Device Type shall be converted to an AllJoyn interface name as follows:

1) Remove the "x." prefix if present

2) For each occurrence of a hyphen (in order from left to right in the string):

   a) If the hyphen is followed by a letter, replace both characters with a single upper-case version of that letter (e.g., convert "-a" to "A").

   b) Else, if the hyphen is followed by another hyphen followed by either a letter or a hyphen, replace two hyphens with a single underscore (e.g., convert "--a" to "_a", "---" to "_-").

   c) Else, convert the hyphen to an underscore (i.e., convert "-" to "_").

Some examples are shown in the Table 7. The first three are unusual OCF names converted (perhaps back) to normal AllJoyn names. The last three are normal OCF names converted to unusual AllJoyn names. ("xn--" is a normal domain name prefix for the Punycode-encoded form of an Internationalized Domain Name, and hence can appear in a normal vendor-specific OCF name.)

Table 7 – Example name mapping

From OCF name	To AllJoyn name
x.example.-widget	example.Widget
x.example.my----widget	example.my__widget
x.example.-my---widget	example.My_Widget
x.xn--p1ai.example	xn_p1ai.example
x.xn--90ae.example	xn__90ae.example
x.example.my-name-1	example.myName_1

An OCF Device Type is mapped to an AllJoyn interface with no members.

Unless specified otherwise, each OCF Resource Type shall be mapped to an AllJoyn interface as follows:

– Each OCF property is mapped to an AllJoyn property in that interface, where each "." (dot) in the OCF property is transformed to "_d", and each "-" (hyphen) in the OCF property is transformed to "_h".

– The EmitsChangedSignal value for each AllJoyn property shall be set to "true" if the resource supports NOTIFY, or "false" if it does not. (The value is never set to "const" or "invalidates" since those concepts cannot currently be expressed in OCF.)

– The "access" attribute for each AllJoyn property shall be "read" if the OCF property is read-only, or "readwrite" if the OCF property is read-write.

– If the resource supports DELETE, a Delete() method shall appear in the interface.

– If the resource supports CREATE, a Create() method shall appear in the interface, with input arguments of each property of the resource to create. (Such information is not available algorithmically can be determined via introspection.) If such information is not available, a CreateWithDefaultValues() method shall appear which takes no input arguments. In either case, the output argument shall be an OBJECT_PATH containing the path of the created resource.

– If the resource supports UPDATE (i.e., the "oic.if.rw" or "oic.if.a" OCF Interface) then an AllJoyn property set operation (i.e., an org.freedesktop.DBus.Properties.Set() method call) shall be mapped to a Partial UPDATE (e.g., POST in CoAP) with the corresponding OCF property.

– If a Resource has a Resource Type "oic.r.alljoynobject", then instead of separately translating each of the Resources in the collection to its own AllJoyn object, all Resources in the collection shall instead be translated to a single AllJoyn object whose object path is the OCF URI path of the collection.

OCF property types shall be mapped to AllJoyn data types according to 6.3.

If an OCF operation fails, the Bridging Function shall send an appropriate AllJoyn error response to the AllJoyn consumer. If an error message is present in the OCF response, and the error message (e.g., diagnostic payload if using CoAP) fits the pattern "<error name>: <error message>" where <error name> conforms to the AllJoyn error name syntax requirements, the AllJoyn error name and AllJoyn error message shall be extracted from the error message in the OCF response. Otherwise, the AllJoyn error name shall be "org.openconnectivity.Error.Code<#>" where <#> is the error code (e.g., CoAP error code) in the OCF response without a decimal (e.g., "404") and the AllJoyn error message is the error message in the OCF response.

6.2.5.2 Exposing an OCF server as a virtual AllJoyn producer

The object description returned in the About interface shall be formed as specified in the AllJoyn About Interface Specification, and Table 8 shows how AllJoyn About Interface fields shall be derived, based on properties in "oic.wk.d", "oic.wk.con", "oic.wk.p", and "oic.wk.con.p".

Table 8 – AllJoyn about data fields

To AJ Field name	AJ Description	AJ Mandatory	From OCF Property title	OCF Property name	OCF Description	OCF Mandatory
AppId	A 128-bit globally unique identifier for the application. The AppId shall be a universally unique identifier as specified in RFC 4122.	Y	Device UUID (no exact equivalent exists)	di	Unique identifier for OCF Device (UUID)	Y
DefaultLanguage	The default language supported by the device. Specified as an IETF language tag listed in RFC 5646.	Y	Default Language	dl	The default language supported by the Device, specified as an RFC 5646 language tag. By default, clients can treat any string property as being in this language unless the property specifies otherwise. If absent, the Bridge Platform shall return a constant, e.g., empty string	N
DeviceName (per supported language)	Name of the device set by platform-specific means (such as Linux and Android).	N	Platform Names	mnpn	Friendly name of the Platform. This property is an array of objects where each object has a "language" field (containing an RFC 5646 language tag) and a "value" field containing the platform friendly name in the indicated language. For example, [{"language":"en", "value":"Dave's Laptop"}]	N
DeviceId	Device identifier set by platform-specific means.	Y	Platform ID	pi	Platform Identifier	Y
AppName (per supported language)	Application name assigned by the app manufacturer (developer or the OEM).	Y	Localized Names, if it exists, else (Device) Name	ln or n	Human-friendly name of the Device, in one or more languages. This property is an array of objects where each object has a "language" field (containing an RFC 5646 language tag) and a "value" field containing the device name in the indicated language. If this property and the Device Name (n) property are both supported, the Device Name (n) value shall be included in this array.	N (ln), Y (n)

To AJ Field name	AJ Description	AJ Mandatory	From OCF Property title	OCF Property name	OCF Description	OCF Mandatory
Manufacturer (per supported language)	The manufacturer's name of the app.	Y	Manufacturer Name	dmn	Name of manufacturer of the Device, in one or more languages. This property is an array of objects where each object has a "language" field (containing an RFC 5646 language tag) and a "value" field containing the manufacturer name in the indicated language.	N
ModelNumber	The app model number.	Y	Model Number	dmno	Model number as designated by manufacturer	N
SupportedLanguages	List of supported languages.	Y	language fields of Localized Names	ln	If ln is supported, return the list of values of the language field of each array element, else return empty array	N
Description (per supported language)	Detailed description expressed in language tags as in RFC 5646.	Y	Localized Descriptions	ld	Detailed description of the Device, in one or more languages. This property is an array of objects where each object has a "language" field (containing an RFC 5646 language tag) and a "value" field containing the device description in the indicated language.	N
DateOfManufacture	Date of manufacture using format YYYY-MM-DD (known as XML DateTime format).	N	Date of Manufacture	mndt	Manufacturing date of device	N
SoftwareVersion	Software version of the app.	Y	Software Version	sv	Software version of the device.	N
AJSoftwareVersion	Current version of the AllJoyn SDK used by the application.	Y	(none)		Bridge Platform should return its own value	
HardwareVersion	Hardware version of the device on which the app is running.	N	Hardware Version	mnhw	Version of platform hardware	N
SupportUrl	Support URL (populated by the manufacturer).	N	Support URL	mnsl	URL that points to support information from manufacturer	N
org.openconnectivity.mnml		N	Manufacturer Details Link (URL)	mnml (if it exists, else field shall be absent)	URL to manufacturer (not to exceed 32 characters)	N

To AJ Field name	AJ Description	AJ Mandatory	From OCF Property title	OCF Property name	OCF Description	OCF Mandatory
org.openconnectivity.mnpv		N	Platform Version	mnpv (if it exists, else field shall be absent)	Version of platform – string (defined by manufacturer)	N
org.openconnectivity.mnos		N	OS Version	mnos (if it exists, else field shall be absent)	Version of platform resident OS – string (defined by manufacturer)	N
org.openconnectivity.mnfv		N	Firmware version	mnfv (if it exists, else field shall be absent)	Version of device firmware	N
org.openconnectivity.st		N	SystemTime	st (if it exists, else field shall be absent)	Reference time for the device	N
org.openconnectivity.piid		N	Protocol Independent ID	piid	A unique and immutable Device identifier. A Client can detect that a single Device supports multiple communication protocols if it discovers that the Device uses a single Protocol Independent ID value for all the protocols it supports.	Y

The AllJoyn field "org.openconnectivity.piid" shall be announced but shall not be localized and its D-Bus type signature shall be "s". All other AllJoyn field names listed in Table 5 which have the prefix "org.openconnectivity." shall be neither announced nor localized and their D-Bus type signature shall be "s".

In addition, any additional vendor-defined properties in the OCF Device resource "/oic/d" (which implements the "oic.wk.d" resource type) and the OCF Platform resource "/oic/p" (which implements the "oic.wk.p" resource type) shall be mapped to vendor-defined fields in the AllJoyn About data, with a field name formed by removing the leading "x." from the property name.

Table 9 shows how AllJoyn Configuration Interface fields shall be derived, based on properties in "oic.wk.con" and "oic.wk.con.p".

Table 9 – AllJoyn configuration data fields

To AJ Field name	AJ Description	AJ Mandatory	From OCF Property title	OCF Property name	OCF Description	OCF Mandatory
DefaultLanguage	Default language supported by the device.	N	Default Language	dl	The default language supported by the Device, specified as an RFC 5646 language tag. By default, clients can treat any string property as being in this language unless the property specifies otherwise.	N
DeviceName	Device name assigned by the user. The device name appears on the UI as the friendly name of the device.	N	PlatformNames	mnpn	Friendly name of the Platform. This property is an array of objects where each object has a "language" field (containing an RFC 5646 language tag) and a "value" field containing the platform friendly name in the indicated language. For example, [{"language":"en", "value":"Dave's Laptop"}]	N
org.openconnectivity.loc		N	Location	loc (if it exists, else field shall be absent)	Provides location information where available.	N
org.openconnectivity.locn		N	Location Name	locn (if it exists, else field shall be absent)	Human friendly name for location For example, "Living Room".	N
org.openconnectivity.c		N	Currency	c (if it exists, else field shall be absent)	Indicates the currency that is used for any monetary transactions	N
org.openconnectivity.r		N	Region	r (if it exists, else field shall be absent)	Free form text Indicating the current region in which the device is located geographically. The free form text shall not start with a quote (").	N

The AllJoyn field "org.openconnectivity.loc" shall be neither announced nor localized and its D-Bus type signature shall be "ad". All other AllJoyn field names listed in Table 5 which have the prefix "org.openconnectivity." shall be neither announced nor localized and their D-Bus type signature shall be "s".

In addition, the Configuration methods FactoryReset and Restart shall be mapped to "oic.wk.mnt" properties Factory_Reset ("fr") and Reboot ("rb"), respectively, and any additional vendor-defined properties in the OCF Configuration resource (which implements the "oic.wk.con" resource type and optionally the "oic.wk.con.p" resource type) shall be mapped to vendor-defined fields the AllJoyn Configuration data, with a field name formed by removing the leading "x." from the property name.

6.2.6 Security

For AllJoyn bridging, an OCF Onboarding Tool shall be able to block the communication of all OCF Devices with all Bridged Devices that don't communicate securely with the Bridge, by using the Bridge Device's "oic.r.securemode" Resource.

6.3 On-the-fly translation from d-bus and OCF payloads

6.3.1 Introduction

The "dbus1" payload format is specified in the D-Bus Specification and AllJoyn adopted the D-Bus protocol and made it distributed over the network. The modifications done by AllJoyn to the format are all in the header part of the packet, not in the data payload itself, which remains compatible with "dbus1". Other variants of the protocol that have been proposed by the Linux community ("GVariant" and "kdbus" payloads) contain slight incompatibilities and are not relevant for this discussion.

6.3.2 Translation without aid of introspection

6.3.2.1 Introduction

Clause 6.3.2 describes how Bridging Functions shall translate messages between the two payload formats in the absence of introspection metadata from the actual device. This situation arises in the when there is content not described by introspection, such as the inner payload of AllJoyn properties of type "D-Bus VARIANT".

Since introspection is not available, the Bridging Function cannot know the rich JSON sub-type, only the underlying CBOR type and from that it can infer the JSON generic type, and hence translation is specified in terms of those generic types.

6.3.2.2 Booleans

Boolean conversion is trivial since both sides support this type.

Table 10 – Boolean translation

D-Bus type	JSON type
"b" – BOOLEAN	boolean (true or false)

6.3.2.3 Numeric types

The translation of numeric types is lossy and that is unavoidable due to the limited expressiveness of the JSON generic types. This can only be solved with introspection.

The translation of numeric types is direction-specific.

Table 11 – Numeric type translation, D-Bus to JSON

From D-Bus type	To JSON type
"y" - BYTE (unsigned 8-bit)	Number
"n" - UINT16 (unsigned 16-bit)	
"u" - UINT32 (unsigned 32-bit)	
"t" - UINT64 (unsigned 64-bit)[a]	
"q" - INT16 (signed 16-bit)	
""" - INT32 (signed 32-bit)	
"x" - INT64 (signed 64-bit)[a]	
"d" - DOUBLE (IEEE 754 double precision)	
[a] D-Bus payloads of types "t" (UINT64) and "x" (INT64) can contain values that cannot be perfectly represented in IEEE 754 double-precision floating point. The RFCs governing JSON do not forbid such numbers but caution that many implementations may not be able to deal with them. Currently, OCF transports its payload using CBOR instead of JSON, which can represent those numbers with fidelity. However, it should be noted that ISO/IEC 30118-1 does not allow for integral numbers outside the range -2^53≤x≤2^53.	

Table 12 – Numeric type translation, JSON to D-Bus

From JSON type	To D-Bus type
number	"d" - DOUBLE[a]
[a] To provide the most predictable result, all translations from OCF to AllJoyn produce values of type "d" DOUBLE (IEEE 754 double precision).	

6.3.2.4 Text strings

Table 13 – Text string translation

D-Bus type	JSON type
"s" – STRING	string

Conversion between D-Bus and JSON strings is simple, as both require their content to be valid Unicode. For example, an implementation can typically do a direct byte copy, as both protocols specify UTF-8 as the encoding of the data, neither constrains the data to a given normalisation format nor specify whether private-use characters or non-characters should be disallowed.

Since the length of D-Bus strings is always known, it is recommended Bridging Functions not use CBOR indeterminate text strings (first byte 0x7f).

6.3.2.5 Byte arrays

The translation of a byte array is direction-specific.

Table 14 – Byte array translation

From D-Bus type	To JSON type
"ay" - ARRAY of BYTE	(base64-encoded) string

The base64url encoding is specified in IETF RF 4648 clause 5.

6.3.2.6 D-bus variants

Table 15 – D-Bus variant translation

D-Bus type	JSON type
"v" – VARIANT	see clause 6.3.2.6

D-Bus has a type called VARIANT ("v") that is a wrapper around any other D-Bus type. It's a way for the type system to perform type-erasure. JSON, on the other hand, is not type-safe, which means that all JSON values are, technically, variants. The conversion for a D-Bus variant to JSON is performed by entering that variant and encoding the type carried inside as per the rules in this document.

The algorithm must be recursive, as D-Bus variants are allowed to contain variants themselves.

6.3.2.7 D-bus object paths and signatures

The translation of D-Bus object paths and signatures is unidirectional (there is no mapping *to* them, only *from* them). This is shown in Table 16. In the reverse direction, clause 6.3.2.4 always converts to D-Bus STRING rather than OBJECT_PATH or SIGNATURE since it is assumed that "s" is the most common string type in use.

Table 16 – D-Bus object path translation

From D-Bus type	To JSON type
"o" - OBJECT_PATH	string
"g" – SIGNATURE	

Both D-Bus object paths and D-Bus type signatures are US-ASCII strings with specific formation rules, found in the D-Bus Specification. They are very seldom used and are not expected to be found in resources subject to translation without the aid of introspection.

6.3.2.8 D-bus structures

The translation of the types in Table 17 is direction-specific:

Table 17 – D-Bus structure translation

From D-Bus type	To JSON type
"r" – STRUCT	array, length > 0

D-Bus structures can be interpreted as a fixed-length array containing a pre-determined list of types for each member. This is how such a structure is mapped to JSON: as an array of heterogeneous content, which are the exact members of the D-Bus structure, in the order in which they appear in the structure.

6.3.2.9 Arrays

The translation of the types in Table 18 is bidirectional:

Table 18 – Byte array translation

D-Bus type	JSON type
"ay" - ARRAY of BYTE	(base64-encoded) string – see 6.3.2.5
"ae" - ARRAY of DICT_ENTRY	object – see 6.3.2.10

The translation of the types in Table 19 is direction-specific:

Table 19 – Other array translation

From D-Bus type	To JSON type
"a" – ARRAY of anything else not specified	array

Aside from arrays of bytes and arrays of dictionary entries, which are mapped to JSON strings and objects respectively, arrays in JSON cannot be constrained to a single type (i.e., heterogeneous arrays). For that reason, strictly speaking all D-Bus arrays excepting arrays of bytes and arrays of dictionary entries must first be converted to arrays of variant "av" and then that array can be converted to JSON. See Table 20.

Table 20 – JSON array translation

From JSON type	Condition	To D-Bus type
array	length=0	"av" – ARRAY of VARIANT
array	length>0, all elements of same type	"a" – ARRAY
array	length>0, elements of different types	"r" – STRUCT

Conversion of D-Bus arrays of variants uses the conversion of variants as specified, which simply eliminates the distinction between a variant containing a given value and that value outside a variant. In other words, the elements of a D-Bus array are extracted and sent as elements of the JSON array, as per the other rules of this document.

6.3.2.10 Dictionaries / objects

The choice of "dictionary of STRING to VARIANT" is made because that is the most common type of dictionary found in payloads and is an almost perfect superset of all possible dictionaries in D-Bus anyway. Moreover, it can represent JSON Objects with fidelity, which is the representation that OCF uses in its data models, which in turn means those D-Bus dictionaries will be able to carry with fidelity any OCF JSON Object in current use. See Table 21

Table 21 – D-Bus dictionary translation

D-Bus type	JSON type
"a{sv}" - dictionary of STRING to VARIANT	object

D-Bus dictionaries that are not mapping string to variant are first converted to those constraints and then encoded in CBOR.

6.3.2.11 Non-translatable types

The types in Table 22 are not translatable, and the Bridging Function should drop the incoming message. None of the types in Table 22 are in current use by either AllJoyn or OCF 1.0 devices, so the inability to translate them should not be a problem.

Table 22 – Non-translation types

Type Scope	Type Name	Description
D-Bus	"h"	UNIX_FD (Unix File Descriptor)
JSON	Null	
JSON	undefined	Not officially valid JSON, but some implementations permit it

6.3.2.12 Examples

Table 23 and Table 24 provide some translation examples.

Table 23 – D-Bus to JSON translation examples

Source D-Bus	JSON Result
BOOLEAN(FALSE)	false
BOOLEAN(TRUE)	true
VARIANT(BOOLEAN(FALSE))	false
VARIANT(BOOLEAN(TRUE))	true
BYTE(0)	0.0
BYTE(255)	255.0
INT16(0)	0.0
INT16(-1)	-1.0
INT16(-32768)	-32768.0
UINT16(0)	0.0
UINT16(65535)	65535.0
INT32(0)	0.0
INT32(-2147483648)	-2147483648.0
INT32(2147483647)	2147483647.0
UINT32(0)	0.0
UINT32(4294967295)	4294967295.0
INT64(0)	0.0
INT64(-1)	-1.0
UINT64(18446744073709551615)	18446744073709551615.0[1]
DOUBLE(0.0)	0.0
DOUBLE(0.5)	0.5
STRING("")	""
STRING("Hello")	"Hello"
ARRAY<BYTE>()	""
ARRAY<BYTE>(0x48, 0x65, 0x6c, 0x6c, 0x6f)	"SGVsbG8"

Source D-Bus	JSON Result
OBJECT_PATH("/")	"/"
SIGNATURE()	""
SIGNATURE("s")	"s"
VARIANT(INT32(0))	0
VARIANT(VARIANT(INT32(0)))	0
VARIANT(STRING("Hello"))	"Hello"

Table 24 – JSON to D-Bus translation examples

Source JSON	D-Bus Result
false	BOOLEAN(false)
true	BOOLEAN(true)
0	DOUBLE(0.0)
-1	DOUBLE(-1.0)
-2147483648	DOUBLE(-2147483648.0)
2147483647	DOUBLE(2147483647.0)
2147483648	DOUBLE(2147483648.0)
-2147483649	DOUBLE(-2147483649.0)
9223372036854775808[1]	DOUBLE(9223372036854775808.0)
0.0	DOUBLE(0.0)
0.5	DOUBLE(0.5)
0.0f	DOUBLE(0.0)
0.5f	DOUBLE(0.5)
""	STRING("")
"Hello"	STRING("Hello")
[]	ARRAY<VARIANT>()
[1]	ARRAY<DOUBLE>(DOUBLE(1.0))
[1, 2147483648, false, "Hello"]	STRUCT<DOUBLE, DOUBLE, BOOLEAN, STRING>(DOUBLE(1.0), DOUBLE(2147483648.0), BOOLEAN(false), STRING("Hello"))
{}	map<STRING, VARIANT>()
{1: 1}	map<STRING, VARIANT>("1" → VARIANT(DOUBLE(1.0)))
{"1": 1}	map<STRING, VARIANT>("1" → VARIANT(DOUBLE(1.0)))

{"rep":
{
"state": false,
"power": 1.0,
"name": "My Light"
}
}
	map<STRING, VARIANT>(
	{STRING("rep"), VARIANT(map<STRING, VARIANT>(
	{STRING("state") → VARIANT(BOOLEAN(FALSE))},
	{STRING("power") → VARIANT(DOUBLE(1.0))},
	{STRING("name") → VARIANT(STRING("My Light"))}
	))}
	)

NOTE This value cannot be represented with IEEE754 double-precision floating point without loss of information. It is also outside the currently-allowed range of integrals in OCF.

6.3.3 Translation with aid of introspection

6.3.3.1 Introduction to introspection metadata

When introspection is available, the Bridging Function can use the extra metadata provided by the side offering the service to expose a higher-quality reply to the other side. This chapter details modifications to the translation described in the previous chapter when the metadata is found.

Introspection metadata can be used for both translating requests to services and replies from those services. When used to translate requests, the introspection is "constraining", since the Bridging Function must conform exactly to what that service expects. When used to translate replies, the introspection is "relaxing", but may be used to inform the receiver what other possible values may be encountered in the future.

Note that OCF introspection uses JSON types, media attributes, and format attributes, not CBOR encoding. The actual encoding of each JSON type is discussed in clause 12.4 of ISO/IEC 30118-1 , JSON format attribute values are as defined in ISO/IEC 30118-1, and JSON media attribute values are as defined in JSON Hyper-Schema.

6.3.3.2 Translation of the introspection itself

Note that both OCF 1.0 and AllJoyn require all services exposed to include introspection metadata, which means the Bridging Function will need to translate the introspection information on-the-fly for each OCF resource or AllJoyn producer it finds. The Bridging Function shall preserve as much of the original information as can be represented in the translated format. This includes both the information used in machine interactions and the information used in user interactions, such as description and documentation text.

6.3.3.3 Variability of introspection data

Introspection data is not a constant and the Bridging Function may find, upon discovering further services, that the D-Bus interface or OCF Resource Type it had previously encountered is different than previously seen. The Bridging Function needs to take care about how the destination side will react to a change in introspection.

D-Bus interfaces used by AllJoyn services may be updated to newer versions, which means a given type of service may be offered by two distinct versions of the same interface. Updates to standardised interfaces must follow strict guidelines established by the AllSeen Interface Review Board, mapping each version to a different OCF Resource Type should be possible without much difficulty. However, there's no guarantee that vendor-specific extensions follow those requirements. Indeed, there's nothing preventing two revisions of a product to contain completely incompatible interfaces that have the same name and version number.

On the opposite direction, the rules are much laxer. Since OCF specifies optional properties to its Resource Types, a simple monotonically-increasing version number like AllJoyn consumer applications expect is not possible.

However, it should be noted that services created by the Bridging Function by "on-the-fly" translation will only be accessed by generic client applications. Dedicated applications will only use "deep binding" translation.

6.3.3.4 Numeric types

For numeric values, all D-Bus and JSON numeric types are treated equally as source and may all be translated into any of the other side's types. When translating a request to a service, the Bridging Function need only verify whether there would be loss of information when translating from source to destination. For example, when translating the number 1.5 to either a JSON integer or to one of the D-Bus integral types, there would be loss of information, in which case the Bridging Function should

refuse the incoming message. Similarly, the value 1,234,567 does not fit the range of a D-Bus byte, 16-bit signed or unsigned integer.

When translating the reply from the service, the Bridging Function shall use the following rules.

Table 25 indicates how to translate from a JSON type to the corresponding D-Bus type, where the first matching row shall be used. If the JSON schema does not indicate the minimum value of a JSON integer, 0 is the default. If the JSON schema does not indicate the maximum value of a JSON integer, $2^{32} - 1$ is the default. The resulting AllJoyn introspection XML shall contain "org.alljoyn.Bus.Type.Min" and "org.alljoyn.Bus.Type.Max" annotations whenever the minimum or maximum, respectively, of the JSON value is different from the natural minimum or maximum of the D-Bus type.

Table 25 – JSON type to D-Bus type translation

From JSON type	Condition	To D-Bus Type
integer	minimum ≥ 0 AND maximum < 2^8	"y" (BYTE)
	minimum ≥ 0 AND maximum < 2^{16}	"q" (UINT16)
	minimum ≥ -2^{15} AND maximum < 2^{15}	"n" (INT16)
	minimum ≥ 0 AND maximum < 2^{32}	"u" (UINT32)
	minimum ≥ -2^{31} AND maximum < 2^{31}	"i" (INT32)
	minimum ≥ 0	"t" (UINT64)
		"x" (INT64)
Number		"d" (DOUBLE)
String	pattern = "^0|([1-9][0-9]{0,19})$"	"t" (UINT64)
	pattern = "^0|(-?[1-9][0-9]{0,18})$"	"x" (INT64)

Table 26 indicates how to translate from a D-Bus type to the corresponding JSON type.

Table 26 – D-Bus type to JSON type translation

From D-Bus type	To JSON type	Note
"y" (BYTE) "n" (UINT16) "q" (INT16) "u" (UINT32) "i" (INT32)	integer	"minimum" and "maximum" in the JSON schema shall be set to the value of the "org.alljoyn.Bus.Type.Min" and "org.alljoyn.Bus.Type.Max" (respectively) annotations if present, or to the min and max values of the D-Bus type's range if such annotations are absent.
"t" (UINT64)	integer if org.alljoyn.Bus.Type.Max ≤ 2^{53}, else string with JSON pattern attribute "^0|([1-9][0-9]{0,19})$".	**IETF RFC 7159** clause 6 explains that higher JSON integers are not interoperable.
"x" (INT64)	integer (if org.alljoyn.Bus.Type.Min ≥ -2^{53} AND org.alljoyn.Bus.Type.Max ≤ 2^{53}), else string with JSON pattern attribute "^0|(-?[1-9][0-9]{0,18})$".	**IETF RFC 7159** clause 6 explains that other JSON integers are not interoperable.
"d" (double)	number	

6.3.3.5 Text string and byte arrays

There's no difference in the translation of text strings and byte arrays compared to clause 6.3.2. Clause 6.3.3 simply lists the JSON equivalent types for the generated OCF introspection. See Table 27.

Table 27 – Text string translation

D-Bus Type	JSON type	JSON media attribute, binaryEncoding property
"s" – STRING	string	(none)
"ay" - ARRAY of BYTE	string	base64

In addition, the mapping of the JSON Types in Table 28 is direction-specific:

Table 28 – JSON UUID string translation

From JSON type	Condition	To D-Bus Type
string	pattern = "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}$"	"ay" – ARRAY of BYTE

JSON strings with any other format value (e.g., date-time, uri, etc.) or pattern value not shown in Table 28 shall be treated the same as if the format and pattern attributes were absent, by simply mapping the value to a D-Bus string.

6.3.3.6 D-bus variants

If the introspection of an AllJoyn producer indicates a value in a request should be a D-Bus VARIANT, the Bridging Function should create such a variant and encode the incoming value as the variant's payload as per the rules in the rest of this document. See Table 29.

Table 29 – D-Bus variant translation

D-Bus Type	JSON Type
"v" – VARIANT	see clause 6.3.3.6

6.3.3.7 D-bus object paths and signatures

If the introspection of an AllJoyn producer indicates a value in a request should be a D-Bus Object Path or D-Bus Signature, the Bridging Function should perform a validity check in the incoming CBOR Text String. If the incoming data fails to pass this check, the message should be rejected. See Table 30.

Table 30 – D-Bus object path translation

From D-Bus Type	To JSON Type
"o" – OBJECT_PATH	string
"g" – SIGNATURE	

6.3.3.8 D-bus structures

D-Bus structure members are described in the introspection XML with the "org.alljoyn.Bus.Struct.*StructureName*.Field.*fieldName*.Type" annotation. The Bridging Function shall use the AJSoftwareVersion field of the About data obtained from a bridged AllJoyn producer as follows. When the version of AllJoyn implemented on the Bridged Device is v16.10.00 or greater and the member annotations are present, the Bridging Function shall use a JSON object to represent a structure, mapping each member to the entry with that name. The Bridging Function needs to be aware that the incoming CBOR payload may have changed the order of the fields, when compared to the D-Bus structure. When the version of AllJoyn implemented on the Bridged Device is less than v16.10.00, the Bridging Function shall follow the rule for translating D-Bus structures without the aid of introspection data.

6.3.3.9 Arrays and dictionaries

If the introspection of the AllJoyn interface indicates that the array is neither an ARRAY of BYTE ("ay") nor an ARRAY of VARIANT ("av") or that the dictionary is not mapping STRING to VARIANT ("a{sv}"), the Bridging Function shall apply the constraining or relaxing rules specified in other clauses.

Similarly, if the OCF introspection indicates a homogeneous array type, the information about the array's element type should be used as the D-Bus array type instead of VARIANT ("v").

6.3.3.10 Other JSON format attribute values

The JSON format attribute may include other custom attribute types. They are not known at this time, but it is expected that those types be handled by their type and representation alone.

6.3.3.11 Examples

Table 31 and Table 32 provide examples using introspection.

Table 31 – Mapping from AllJoyn using introspection

AllJoyn Source	AllJoyn Introspection Notes	Translated JSON Payload	OCF Introspection Notes
UINT32 (0)		0	JSON schema should indicate: "type": "integer", "minimum": 0, "maximum": 4294967295
INT64 (0)		0	Since no Min/Max annotations exist in AllJoyn, JSON schema should indicate: "type": "string", "pattern": "^0|(-?[1-9][0-9]{0,18})$"
UINT64 (0)		"0"	Since no Max annotation exists in AllJoyn, JSON schema should indicate: "type": "string", "pattern": ^0|([1-9][0-9]{0,19})$"
STRING("Hello")		"Hello"	JSON schema should indicate: "type": "string"
OBJECT_PATH("/")		"/"	JSON schema should indicate: "type": "string"

AllJoyn Source	AllJoyn Introspection Notes	Translated JSON Payload	OCF Introspection Notes
SIGNATURE("g")		"g"	JSON schema should indicate: "type": "string"
ARRAY<BYTE>(0x48, 0x65, 0x6c, 0x6c, 0x6f)		"SGVsbG8"	JSON schema should indicate: "type": "string", "media binaryEncoding": "base64"
VARIANT(*anything*)		*?*	JSON schema should indicate: "type": ["boolean", "object", "array", "number", "string", "integer"]
ARRAY<INT32>()		[]	JSON schema should indicate: "type": "array", "items": { "type": "integer" }
ARRAY<INT64>()		[]	JSON schema should indicate: "type": "array", "items": { "type": "string", "pattern": "^0|([1-9][0-9]{0,18})$" }
STRUCT< INT32, INT32>(0, 1)	AllJoyn introspection specifies the argument with the annotation: <struct name="Point"> <field name="x" type="i"/> <field name="y" type="i"/> </struct>	{"x": 0, "y": 1}	JSON schema should indicate: "type": "object", "properties": { "x": { "type": "integer" }, "y": { "type": "integer" } }

Table 32 – Mapping from CBOR using introspection

CBOR Payload	OCF Introspection Notes	Translated AllJoyn	AllJoyn Introspection Notes
0	"type": "integer"	INT32(0)	
0	"type": "integer", "minimum": -240, "maximum": 240	INT64(0)	org.alljoyn.Bus.Type.Min = -240 org.alljoyn.Bus.Type.Max = 240
0	"type": "integer", "minimum": 0, "maximum": 248	UINT64(0)	org.alljoyn.Bus.Type.Max = 248
0.0	"type": "number"	DOUBLE(0.0)	
[1]	JSON schema indicates: "type": "array", "items": { "type": "integer", "minimum": 0, "maximum": 246 }	ARRAY<UINT64>(1)	org.alljoyn.Bus.Type.Max = 246

7 Device type mapping

7.1 AllJoyn device types to OCF device types

Table 33 captures the equivalency mapping between AllJoyn defined Device Types (see AllJoyn Common Data Model Interface Definitions) and OCF defined Device Types (see Table 10-1 in ISO/IEC 30118-5). The minimum interface set for the AllJoyn definitions is provided in the HAE Theory of Operation; the minimum Resource sets for each OCF Device is provided in ISO/IEC 30118-5.

Table 33 – AllJoyn to OCF device type mapping

Classification	AllJoyn Device Type	AllJoyn ID	OCF Device Type
Air Care	Air Conditioner	5	oic.d.airconditioner
	Air Purifier	9	oic.d.airpurifier
	Air Quality Monitor	11	oic.d.airqualitymonitor
	Dehumidifier	8	oic.d.dehumidifier
	Humidifier	7	oic.d.humidifier
	Electric Fan	10	oic.d.fan
	Thermostat	6	oic.d.thermostat
Fabric Care	Clothes Washer	12	oic.d.washer
	Clothes Dryer	13	oic.d.dryer
	Clothes Washer-Dryer	14	oic.d.washerdryer
Food Preservation	Refrigerator	2	oic.d.refrigerator
	Ice-Maker	4	oic.r.icemaker (maps to Resource)
	Freezer	3	oic.d.freezer
Food Preparation	Oven	17	oic.d.oven
	Cooktop	18	oic.d.cooktop
	Cookerhood	19	oic.d.cookerhood
	Food probe	20	oic.d.foodprobe
Dish Care	Dishwasher	15	oic.d.dishwasher
Floor Care	Robot Cleaner	16	oic.d.robotcleaner
Entertainment	Television	21	oic.d.tv
	Set Top Box (STB)	22	oic.d.stb

7.2 OCF device types with no AllJoyn equivalent

Table 34 captures the Device Types defined by OCF have no direct equivalent in AllJoyn, they shall all be mapped to an AllJoyn Device Type of "Other" (Id of "1").

Table 34 – OCF device types with no AllJoyn equivalent

OCF Device Name	OCF Device Type
Receiver	oic.d.receiver
Blind	oic.d.blind
Door	oic.d.door
Garage Door	oic.d.garagedoor
Generic Sensor	oic.d.sensor
Light	oic.d.light
Smart Plug	oic.d.smartplug
Switch	oic.d.switch
Water Valve	oic.d.watervalve
Printer	oic.d.printer
Multi-Function Printer	oic.d.multifunctionprinter
Scanner	oic.r.scanner
Camera	oic.d.camera
Security Panel	oic.d.securitypanel
Smart Lock	oic.d.smartlock

8 Resource to interface equivalence

8.1 Introduction

Clause 8 captures the equivalency mapping between AllJoyn defined Interfaces (see AllJoyn Common Data Model Interface Definitions) and OCF defined Resource Types (see ISO/IEC 30118-4). Detailed Property by Property mappings are provided in clause 9.

Table 35 captures the mappings for Interfaces that are part of the minimum set for an AllJoyn Device.

Table 36 captures the mappings for Interfaces that are optional for an AllJoyn Device; deep translation for these interfaces via derived modelling is not within the scope of this release of the document.

Table 35 – AllJoyn interface to OCF resource type mapping – minimum interface set

AllJoyn Interface	OCF Resource Type Name	OCF Resource Type ID	OCF Interface(s)
Environment.CurrentAirQuality	Air Quality Collection	oic.r.airqualitycollection	oic.if.s
Environment.CurrentAirQualityLevel	Air Quality Collection	oic.r.airqualitycollection	oic.if.s
Environment.CurrentHumidity	Humidity	oic.r.humidity	oic.if.s
Environment.CurrentTemperature	Temperature	oic.r.temperature	oic.if.s
Environment.TargetHumidity	Humidity	oic.r.humidity, oic.r.selectablelevels	oic.if.a
Environment.TargetTemperature	Temperature	oic.r.temperature	oic.if.a
Operation.AudioVolume	Audio Controls	oic.r.audio	oic.if.a
Operation.Channel	Not mapped		
Operation.ClimateControlMode	Mode	oic.r.mode	oic.if.a
	Operational State	oic.r.operational.state	oic.if.s

AllJoyn Interface	OCF Resource Type Name	OCF Resource Type ID	OCF Interface(s)
Operation.ClosedStatus	Door	oic.r.door	oic.if.s
Operation.CycleControl	Operational State	oic.r.operational.state	oic.if.s
Operation.FanSpeedLevel	Air Flow	oic.r.airflow	oic.if.a
Operation.HeatingZone	Heating Zone Collection	oic.r.heatingzonecollection	oic.if.s
Operation.HvacFanMode	Mode	oic.r.mode	oic.if.a
Operation.OnOffStatus	Binary Switch	oic.r.switch.binary	oic.if.s
Operation.OvenCyclePhase	Operational State	oic.r.operationalstate	oic.if.s

Table 36 – AllJoyn interface to OCF resource type mapping – optional interface set

AllJoyn Interface	OCF Resource Type Name	OCF Resource Type ID	OCF Interface(s)
Environment.TargetTemperatureLevel	Mode	oic.r.mode	oic.if.a
Environment.WaterLevel	TBD	TBD	oic.if.s
Environment.WindDirection	Air Flow	oic.r.airflow	oic.if.a
Operation.AirRecirculationMode	Mode	oic.r.mode	oic.if.a
Operation.Alerts	TBD	TBD	TBD
Operation.AudioVideoInput	Media Source List	oic.r.media.input	oic.if.a
Operation.BatteryStatus	Battery	oic.r.energy.battery	oic.if.s
Operation.CurrentPower	Energy Usage	oic.r.energy.usage	oic.if.s
Operation.DishWashingCyclePhase	Operational State	oic.r.operationalstate	oic.if.s
Operation.EnergyUsage	Energy Usage	oic.r.energy.usage	oic.if.s
Operation.FilterStatus	TBD	TBD	TBD
Operation.LaundryCyclePhase	Mode	oic.r.mode	oic.if.s
Operation.MoistureOutputLevel	Mode	oic.r.mode	oic.if.a
Operation.PlugInUnits	TBD	TBD	TBD
Operation.RapidMode	Refrigeration	oic.r.refrigeration	oic.if.a
Operation.RemoteControllability	TBD	TBD	TBD
Operation.RepeatMode	Ecomode	oic.r.ecomode	oic.if.a
Operation.ResourceSaving	TBD	TBD	TBD
Operation.RobotCleaningCyclePhase	Mode	oic.r.mode	oic.if.s
Operation.SoilLevel	Mode	oic.r.mode	oic.if.a
Operation.SpinSpeedLevel	Mode	oic.r.mode	oic.if.a
Operation.Timer	Time Period	oic.r.time.period	oic.if.s

8.2 Environment.CurrentAirQuality mapping

If more than one instance of the AirQuality interface is exposed, then each instance maps to an instance of the OCF AirQuality Resource. The mapping defined in clause 9.2 describes the population of the OCF AirQuality Resource. Even if there is only a single instance of an OCF AirQuality Resource this shall be included in an instance of an OCF AirQualityCollection. The number of links in the collection equates to the number of instances of the AllJoyn CurrentAirQuality interface that are exposed. When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked only if this is set to "Measured".

8.3 Environment.CurrentAirQualityLevel mapping

If more than one instance of the AirQualityLevel interface is exposed, then each instance maps to an instance of the OCF AirQuality Resource. The mapping defined in clause 9.2 describes the population of the OCF AirQuality Resource. Even if there is only a single instance of an OCF AirQuality Resource then this shall be included in an instance of an OCF AirQualityCollection. The number of links in the collection equates to the number of instances of the AllJoyn CurrentAirQuality interface that are exposed. When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked only if this is set to "Qualitative".

8.4 Operation.ClimateControlMode mapping

ClimateControlMode has three Properties; these map as follows: mode and supportedmodes maps to the Mode Resource, operationalstate maps to the OperationalState Resource This can be represented in OCF either as two distinct Resource instances or a single instance with two Resource Types (oic.r.mode, oic.r.operationalstate).

8.5 Operation.FanSpeedLevel mapping

The setting of the FanSpeedLevel to "0x00" (off) is handled via the "OffControl" interface rather than writing directly to this interface. In such a case an instance of Binary Switch shall be exposed on the OCF side; this can be modelled as AirFlowControl which is then a collection of Binary Switch and AirFlow.

8.6 Operation.HeatingZone mapping

Each element in the array of heating zones within the AllJoyn HeatingZone interface maps to an instance of OCF HeatingZone, itself a link in an instance of an OCF HeatingZoneCollection. The mapping defined clause 9.13 describes the population of the OCF HeatingZone Resource that constitutes the Resources that are contained in the collection.

8.7 Operation.OnOffStatus, Operation.OnControl, and Operation.OffControl mapping

A discovered instance of a Binary Switch is always mapped to an Operation.OnOffStatus interface. A RETRIEVE on a Binary Switch maps to an action on an instance of an Operation.OnOffStatus Interface. An UPDATE on a Binary Switch maps to a method invocation on either Operation.OnControl or OffControl. value = true maps to Operation.OnControl value = false maps to Operation.OffControl.

8.8 Operation.OvenCyclePhase

OvenCyclePhase cyclephase Property pre-defines values 0x00-0x7F, 0x80-0xFF is for vendor specific values. The mapping defined in clause 9 covers only specification defined values. Any vendor defined value shall be represented in OCF using the x.<organization> syntax for a vendor defined Property.

9 Detailed mapping APIs

9.1 Introduction

This clause provides a mapping description (using JSON that aligns with the Derived Modelling syntax described in Derived Models for Interoperability between IoT Ecosystems for all Interfaces and Resources that are within scope.

The derived model definitions presented in clause 9 are formatted for readability, and so may appear to have extra line breaks.

Table 37 provides a reference and link to the per Interface clauses.

Table 37 – Interface to resource summary

AllJoyn Interface Name	Equivalent Resource(s)	Clause
Environment.CurrentAirQuality	oic.r.airqualitycollection	8.2
Environment.CurrentAirQualityLevel	oic.r.airqualitycollection	8.3
Environment.CurrentHumidity	oic.r.humidity	8.4
Environment.CurrentTemperature	oic.r.temperature	8.5
Environment.TargetHumidity	oic.r.humidity, oic.r.selectablelevels	8.6
Environment.TargetTemperature	oic.r.temperature	8.7
Operation.AudioVolume	oic.r.audio	8.8
Operation.ClimateControlMode	oic.r.mode, oic.r.operationalstate	8.9
Operation.ClosedStatus	oic.r.door	8.10
Operation.CycleControl	oic.r.operational.state	8.11
Operation.FanSpeedLevel	oic.r.airflow	8.12
Operation.HeatingZone	oic.r.heatingzonecollection	8.13
Operation.HvacFanMode	oic.r.mode	8.14
Operation.OnControl, Operation.OffControl	oic.r.switch.binary	8.15
Operation.OnOffStatus,	oic.r.switch.binary	8.16
Operation.OvenCyclePhase	oic.r.operationalstate	8.17

9.2 Current air quality

9.2.1 Derived model

The derived model: "asa.environment.currentairquality".

9.2.2 Property definition

Table 38 provides the detailed per Property mapping for "asa.environment.currentairquality".

Table 38 – The property mapping for "asa.environment.currentairquality"

AllJoyn Property name	OCF Resource	To OCF	From OCF
minvalue	oic.r.airquality	range[0] = minvalue	minvalue = range[0]
maxvalue	oic.r.airquality	range[1] = maxvalue	maxvalue = range[1]
contaminanttype	oic.r.airquality	valuetype = Measuredcontaminanttypearray = [CH2O,CO2,CO,PM2_5,PM10,VOC]ocf.contaminanttype = contaminanttypearray[contaminanttype]	contaminanttype = indexof contaminanttypearray[ocf.contaminanttype]
currentvalue	oic.r.airquality	contaminantvalue = currentvalue	currentvalue = contaminantvalue
updatemintime	oic.r.value.conditional	ocf.minnotifyperiod = updatemintime	updatemintime = ocf.minnotifyperiod
precision	oic.r.airquality	ocf.precision = precision	precision = ocf.precision

Table 39 provides the details of the Properties that are part of "asa.environment.currentairquality".

Table 39 – The properties of "asa.environment.currentairquality"

AllJoyn Property name	Type	Required	Description
minvalue	number	yes	
maxvalue	number	yes	
contaminanttype	integer	yes	The contaminant type
currentvalue	number	yes	
updatemintime	integer	yes	
precision	number	yes	

9.2.3 Derived model definition

```
{
  "id": "http://openinterconnect.org/asamapping/schemas/asa.environment.currentairquality.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Current Air Quality",
  "definitions": {
    "asa.environment.currentairquality": {
      "type": "object",
      "properties": {
        "contaminanttype": {
          "type": "integer",
          "description": "The contaminant type",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "valuetype = Measured",
              "contaminanttypearray = [CH2O,CO2,CO,PM2_5,PM10,VOC]",
              "ocf.contaminanttype = contaminanttypearray[contaminanttype]"
            ],
            "x-from-ocf": [
              "contaminanttype = indexof contaminanttypearray[ocf.contaminanttype]"
            ]
          }
        },
        "currentvalue": {
          "type": "number",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "contaminantvalue = currentvalue"
            ],
            "x-from-ocf": [
              "currentvalue = contaminantvalue"
            ]
          }
        },
        "minvalue": {
          "type": "number",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "range[0] = minvalue"
            ],
            "x-from-ocf": [
              "minvalue = range[0]"
            ]
          }
        },
        "maxvalue": {
          "type": "number",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "range[1] = maxvalue"
```

```
            ],
            "x-from-ocf": [
              "maxvalue = range[1]"
            ]
          }
        },
        "precision": {
          "type": "number",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "ocf.precision = precision"
            ],
            "x-from-ocf": [
              "precision = ocf.precision"
            ]
          }
        },
        "updatemintime": {
          "type": "integer",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.value.conditional",
            "x-to-ocf": [
              "ocf.minnotifyperiod = updatemintime"
            ],
            "x-from-ocf": [
              "updatemintime = ocf.minnotifyperiod"
            ]
          }
        }
      }
    }
  },
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/asa.environment.currentairquality"}
  ],
  "required": ["contaminanttype","currentvalue","minvalue","maxvalue","precision","updatemintime"]
}
```


9.3 Current air quality level

9.3.1 Derived model

The derived model: "asa.environment.currentairqualitylevel".

9.3.2 Property definition

Table 40 provides the detailed per Property mapping for "asa.environment.currentairqualitylevel".

Table 40 – The property mapping for "asa.environment.currentairqualitylevel"

AllJoyn Property name	OCF Resource	To OCF	From OCF
contaminanttype	oic.r.airquality	valuetype = Qualitativeif contaminanttype = 0, ocf.contaminanttype = CH2Oif contaminanttype = 1, ocf.contaminanttype = CO2if contaminanttype = 2, ocf.contaminanttype = COif contaminanttype = 3, ocf.contaminanttype = PM2_5if contaminanttype = 4, ocf.contaminanttype = PM10if contaminanttype = 5, ocf.contaminanttype = VOCif contaminanttype = 253, ocf.contaminanttype = Smokeif contaminanttype = 254, ocf.contaminanttype = Odorif contaminanttype = 255, ocf.contaminanttype = AirPollution	if ocf.contaminanttype = CH2O, contaminanttype = 0if ocf.contaminanttype = CO2, contaminanttype = 1if ocf.contaminanttype = CO, contaminanttype = 2if ocf.contaminanttype = PM2_5, contaminanttype = 3if ocf.contaminanttype = PM10, contaminanttype = 4if ocf.contaminanttype = VOC, contaminanttype = 5if ocf.contaminanttype = Smoke, contaminanttype = 253if ocf.contaminanttype = Odor, contaminanttype = 254if ocf.contaminanttype = AirPollution, contaminanttype = 255
maxlevel	oic.r.airquality	range[0] = 0range[1] = maxvalue	maxvalue = range[1]
currentlevel	oic.r.airquality	contaminantvalue = currentlevel	currentlevel = contaminantvalue

Table 41 provides the details of the Properties that are part of "asa.environment.currentairqualitylevel".

Table 41 – The properties of "asa.environment.currentairqualitylevel"

AllJoyn Property name	Type	Required	Description
contaminanttype	integer	yes	The contaminant type
maxlevel	integer	yes	
currentlevel	integer	yes	

9.3.3 Derived model definition

```
{
  "id": "http://openinterconnect.org/asamapping/schemas/asa.environment.currentairqualitylevel.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Current Air Quality Level",
  "definitions": {
    "asa.environment.currentairqualitylevel": {
      "type": "object",
      "properties": {
        "contaminanttype": {
          "type": "integer",
          "description": "The contaminant type",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "valuetype = Qualitative",
              "if contaminanttype = 0, ocf.contaminanttype = CH2O",
              "if contaminanttype = 1, ocf.contaminanttype = CO2",
              "if contaminanttype = 2, ocf.contaminanttype = CO",
              "if contaminanttype = 3, ocf.contaminanttype = PM2_5",
              "if contaminanttype = 4, ocf.contaminanttype = PM10",
              "if contaminanttype = 5, ocf.contaminanttype = VOC",
              "if contaminanttype = 253, ocf.contaminanttype = Smoke",
              "if contaminanttype = 254, ocf.contaminanttype = Odor",
              "if contaminanttype = 255, ocf.contaminanttype = AirPollution"
```

```
            ],
            "x-from-ocf": [
              "if ocf.contaminanttype = CH2O, contaminanttype = 0",
              "if ocf.contaminanttype = CO2, contaminanttype = 1",
              "if ocf.contaminanttype = CO, contaminanttype = 2",
              "if ocf.contaminanttype = PM2_5, contaminanttype = 3",
              "if ocf.contaminanttype = PM10, contaminanttype = 4",
              "if ocf.contaminanttype = VOC, contaminanttype = 5",
              "if ocf.contaminanttype = Smoke, contaminanttype = 253",
              "if ocf.contaminanttype = Odor, contaminanttype = 254",
              "if ocf.contaminanttype = AirPollution, contaminanttype = 255"
            ]
          }
        },
        "currentlevel": {
          "type": "integer",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "contaminantvalue = currentlevel"
            ],
            "x-from-ocf": [
              "currentlevel = contaminantvalue"
            ]
          }
        },
        "maxlevel": {
          "type": "integer",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "range[0] = 0",
              "range[1] = maxvalue"
            ],
            "x-from-ocf": [
              "maxvalue = range[1]"
            ]
          }
        }
      }
    }
  },
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/asa.environment.currentairqualitylevel"}
  ],
  "required": ["contaminanttype","currentlevel","maxlevel"]
}
```


9.4 Current humidity

9.4.1 Derived model

The derived model: "asa.environment.currenthumidity".

9.4.2 Property definition

Table 42 provides the detailed per Property mapping for "asa.environment.currenthumidity".

Table 42 – The property mapping for "asa.environment.currenthumidity"

AllJoyn Property name	OCF Resource	To OCF	From OCF
maxvalue	oic.r.humidity	range[0] = 0range[1] = maxvalue	maxvalue = range[1]
currentvalue	oic.r.humidity	humidity = currentValue	currentvalue = humidity

Table 43 provides the details of the Properties that are part of "asa.environment.currenthumidity".

Table 43 – The properties of "asa.environment.currenthumidity"

AllJoyn Property name	Type	Required	Description
maxvalue	number	yes	Max measured value for humidity
currentvalue	number	yes	Measured value

9.4.3 Derived model definition

```
{
  "id": "http://openinterconnect.org/asamapping/schemas/asa.environment.currenthumidity.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Current Humidity",
  "definitions": {
    "asa.environment.currenthumidity": {
      "type": "object",
      "properties": {
        "currentvalue": {
          "type": "number",
          "description": "Measured value",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.humidity",
            "x-to-ocf": [
              "humidity = currentValue"
            ],
            "x-from-ocf": [
              "currentvalue = humidity"
            ]
          }
        },
        "maxvalue": {
          "type": "number",
          "description": "Max measured value for humidty",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.humidity",
            "x-to-ocf": [
              "range[0] = 0",
              "range[1] = maxvalue"
            ],
            "x-from-ocf": [
              "maxvalue = range[1]"
            ]
          }
        }
      }
    }
  },
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/asa.environment.currenthumidity"}
  ],
  "required": [ "currentvalue","maxvalue"]
}
```

9.5 Current temperature

9.5.1 Derived model

The derived model: "asa.environment.currenttemperature".

9.5.2 Property definition

Table 44 provides the detailed per Property mapping for "asa.environment.currenttemperature".

Table 44 – The property mapping for "asa.environment.currenttemperature"

AllJoyn Property name	OCF Resource	To OCF	From OCF
precision	oic.r.temperature	ocf.precision = precision	precision = ocf.precision
currentvalue	oic.r.temperature	temperature = currentValueunits = C	oneOf
updatemintime	oic.r.value.conditional	ocf.minnotifyperiod = updatemintime	updatemintime = ocf.minnotifyperiod

Table 45 provides the details of the Properties that are part of "asa.environment.currenttemperature".

Table 45 – The properties of "asa.environment.currenttemperature"

AllJoyn Property name	Type	Required	Description
precision	number	yes	
currentvalue	number	yes	Measured value
updatemintime	integer	yes	

9.5.3 Derived model definition

```
{
  "id": "http://openinterconnect.org/asamapping/schemas/asa.environment.currenttemperature.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Current Temperature",
  "definitions": {
    "asa.environment.currenttemperature": {
      "type": "object",
      "properties": {
        "currentvalue": {
          "type": "number",
          "description": "Measured value",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.temperature",
            "x-to-ocf": [
              "temperature = currentValue",
              "units = C"
            ],
            "x-from-ocf": {
              "oneOf": [
                {
                  "properties": {
                    "units": "string",
                    "enum": ["C"]
                  },
                  "x-from-ocf": [
                    "currentvalue = temperature"
                  ]
                },
                {
                  "properties": {
                    "units": "string",
                    "enum": ["F"]
                  },
                  "x-from-ocf": [
                    "currentvalue = (temperature-32)*5/9"
                  ]
                },
                {
                  "properties": {
                    "units": "string",
                    "enum": ["K"]
                  },
                  "x-from-ocf": [
                    "currentvalue = temperature-273.15"
```

```
                    ]
                  }
                ]
              }
            }
          },
          "precision": {
            "type": "number",
            "x-ocf-conversion": {
              "x-ocf-alias": "oic.r.temperature",
              "x-to-ocf": [
                "ocf.precision = precision"
              ],
              "x-from-ocf": [
                "precision = ocf.precision"
              ]
            }
          },
          "updatemintime": {
            "type": "integer",
            "x-ocf-conversion": {
              "x-ocf-alias": "oic.r.value.conditional",
              "x-to-ocf": [
                "ocf.minnotifyperiod = updatemintime"
              ],
              "x-from-ocf": [
                "updatemintime = ocf.minnotifyperiod"
              ]
            }
          }
        }
      }
    },
    "type": "object",
    "allOf": [
      {"$ref": "#/definitions/asa.environment.currenttemperature"}
    ],
    "required": [ "currentvalue","precision","updatemintime" ]
}
```


9.6 Target humidity

9.6.1 Derived model

The derived model: "asa.environment.targethumidity".

9.6.2 Property definition

Table 46 provides the detailed per Property mapping for "asa.environment.targethumidity".

Table 46 – The property mapping for "asa.environment.targethumidity"

AllJoyn Property name	OCF Resource	To OCF	From OCF
minvalue	oic.r.humidity	range[0] = minvalue	minvalue = range[0]otherwise: minvalue = 0
targetvalue	oic.r.humidity,oic.r.selectablelevels	if minvalue != maxvalue, ocf.desiredhumidity = targetvalue;ocf.targetlevel = selectablehumiditylevels[0].if minvalue == maxvalue, ocf.targetlevel = targetvalue.	if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel.
maxvalue	oic.r.humidity	range[1] = maxvalue	maxvalue = range[1]otherwise: maxvalue = 100
stepvalue	oic.r.humidity	step = stepvalue	stepvalue = stepotherwise: stepvalue = 1
selectablehumiditylevels	oic.r.selectablelevels	availablelevels[] = selectablehumiditylevels[]	selectablehumiditylevels[] = availablelevels[]

Table 47 provides the details of the Properties that are part of "asa.environment.targethumidity".

Table 47 – The properties of "asa.environment.targethumidity"

AllJoyn Property name	Type	Required	Description
minvalue	number	yes	
targetvalue	number	yes	Measured value
maxvalue	number	yes	
stepvalue	number	yes	
selectablehumiditylevels	array	yes	

9.6.3 Derived model definition

```
{
  "id": "http://openinterconnect.org/asamapping/schemas/asa.environment.targethumidity.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Target Humidity",
  "definitions": {
    "asa.environment.targethumidity": {
      "type": "object",
      "properties": {
        "targetvalue": {
          "type": "number",
          "description": "Measured value",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
            "x-to-ocf": [
              "if minvalue != maxvalue, ocf.desiredhumidity = targetvalue;ocf.targetlevel =
selectablehumiditylevels[0].",
              "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
            ],
            "x-from-ocf": [
              "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
              "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
            ]
          }
        },
        "minvalue": {
```

```
          "type": "number",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.humidity",
            "x-to-ocf": [
              "range[0] = minvalue"
            ],
            "x-from-ocf": [
              "minvalue = range[0]",
              "otherwise: minvalue = 0"
            ]
          }
        },
        "maxvalue": {
          "type": "number",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.humidity",
            "x-to-ocf": [
              "range[1] = maxvalue"
            ],
            "x-from-ocf": [
              "maxvalue = range[1]",
              "otherwise: maxvalue = 100"
            ]
          }
        },
        "stepvalue": {
          "type": "number",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.humidity",
            "x-to-ocf": [
              "step = stepvalue"
            ],
            "x-from-ocf": [
              "stepvalue = step",
              "otherwise: stepvalue = 1"
            ]
          }
        },
        "selectablehumiditylevels": {
          "type": "array",
          "items": {
            "type": "number"
          },
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.selectablelevels",
            "x-to-ocf": [
              "availablelevels[] = selectablehumiditylevels[]"
            ],
            "x-from-ocf": [
              "selectablehumiditylevels[] = availablelevels[]"
            ]
          }
        }
      }
    }
  },
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/asa.environment.targethumidity"}
  ],
  "required": [ "targetvalue","minvalue","maxvalue","stepvalue","selectablehumiditylevels" ]
}
```


9.7 Target temperature

9.7.1 Derived model

The derived model: "asa.environment.targettemperature".

9.7.2 Property definition

Table 48 provides the detailed per Property mapping for "asa.environment.targettemperature".

Table 48 – The property mapping for "asa.environment.targettemperature"

AllJoyn Property name	OCF Resource	To OCF	From OCF
minvalue	oic.r.temperature	range[0] = minvalue	minvalue = range[0]otherwise: minvalue = -MAXINT
targetvalue	oic.r.temperature	temperature = targetvalueunits = C	oneOf
maxvalue	oic.r.temperature	range[1] = maxvalue	maxvalue = range[1]otherwise: maxvalue = MAXINT
step	oic.r.temperature	ocf.step = step	step = ocf.stepotherwise: step = undefined (0x00)

Table 49 provides the details of the Properties that are part of "asa.environment.targettemperature".

Table 49 – The properties of "asa.environment.targettemperature"

AllJoyn Property name	Type	Required	Description
minvalue	number	yes	
targetvalue	number	yes	Measured value
maxvalue	number	yes	
step	number	yes	

9.7.3 Derived model definition

```
{
  "id": "http://openinterconnect.org/asamapping/schemas/asa.environment.targettemperature.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Target Temperature",
  "definitions": {
    "asa.environment.targettemperature": {
      "type": "object",
      "properties": {
        "targetvalue": {
          "type": "number",
          "description": "Measured value",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.temperature",
            "x-to-ocf": [
              "temperature = targetvalue",
              "units = C"
            ],
            "x-from-ocf": {
              "oneOf": [
                {
                  "properties": {
                    "units": "string",
                    "enum": ["C"]
                  },
                  "x-from-ocf": [
                    "targetvalue = temperature"
                  ]
                },
                {
                  "properties": {
                    "units": "string",
                    "enum": ["F"]
                  },
                  "x-from-ocf": [
                    "targetvalue = (temperature-32)*5/9"
                  ]
                },
```

```
                {
                  "properties": {
                    "units": "string",
                    "enum": ["K"]
                  },
                  "x-from-ocf": [
                    "targetvalue = temperature-273.15"
                  ]
                }
              ]
            }
          }
        },
        "minvalue": {
          "type": "number",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.temperature",
            "x-to-ocf": [
              "range[0] = minvalue"
            ],
            "x-from-ocf": [
              "minvalue = range[0]",
              "otherwise: minvalue = -MAXINT"
            ]
          }
        },
        "maxvalue": {
          "type": "number",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.temperature",
            "x-to-ocf": [
              "range[1] = maxvalue"
            ],
            "x-from-ocf": [
              "maxvalue = range[1]",
              "otherwise: maxvalue = MAXINT"
            ]
          }
        },
        "step": {
          "type": "number",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.temperature",
            "x-to-ocf": [
              "ocf.step = step"
            ],
            "x-from-ocf": [
              "step = ocf.step",
              "otherwise: step = undefined (0x00)"
            ]
          }
        }
      }
    }
  },
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/asa.environment.targettemperature"}
  ],
  "required": [ "targetvalue","minvalue","maxvalue","step" ]
}
```


9.8 Audio volume

9.8.1 Derived model

The derived model: "asa.operation.audiovolume".

9.8.2 Property definition

Table 50 provides the detailed per Property mapping for "asa.operation.audiovolume".

Table 50 – The property mapping for "asa.operation.audiovolume"

AllJoyn Property name	OCF Resource	To OCF	From OCF
mute	oic.r.audio	ocf.mute = mute	mute = ocf.mute
maxvolume	oic.r.audio	range[0] = 0range[1] = maxvolume	maxvolume = range[1]otherwise: maxvalue = 100
volume	oic.r.audio	ocf.volume = volume	volume = ocf.volume

Table 51 provides the details of the Properties that are part of "asa.operation.audiovolume".

Table 51 – The properties of "asa.operation.audiovolume"

AllJoyn Property name	Type	Required	Description
mute	boolean	yes	
maxvolume	integer	yes	
volume	integer	yes	Speaker volume index

9.8.3 Derived model definition

```
{
  "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.audiovolume.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Audio Volume",
  "definitions": {
    "asa.operation.audiovolume": {
      "type": "object",
      "properties": {
        "volume": {
          "type": "integer",
          "description": "Speaker volume index",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.audio",
            "x-to-ocf": [
              "ocf.volume = volume"
            ],
            "x-from-ocf": [
              "volume = ocf.volume"
            ]
          }
        },
        "maxvolume": {
          "type": "integer",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.audio",
            "x-to-ocf": [
              "range[0] = 0",
              "range[1] = maxvolume"
            ],
            "x-from-ocf": [
              "maxvolume = range[1]",
              "otherwise: maxvalue = 100"
            ]
          }
        },
        "mute": {
          "type": "boolean",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.audio",
            "x-to-ocf": [
              "ocf.mute = mute"
            ],
            "x-from-ocf": [
              "mute = ocf.mute"
            ]
```

```
          }
        }
      }
    }
  },
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/asa.operation.audiovolume"}
  ],
  "required": [ "volume","maxvolume","mute" ]
}
```

9.9 Climate control mode

9.9.1 Derived model

The derived model: "asa.operation.climatecontrolmode".

9.9.2 Property definition

Table 52 provides the detailed per Property mapping for "asa.operation.climatecontrolmode".

Table 52 – The property mapping for "asa.operation.climatecontrolmode"

AllJoyn Property name	OCF Resource	To OCF	From OCF
operationalstate	oic.r.operationalstate	machinestates = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]currentmachinestate = machinestates[operationalstate]	statearray = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]operationalstate = indexof statearray[currentmachinestate[0]]
supportedmodes	oic.r.mode	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]for x=0, x <sizeof(supportedmodes): ocf.supportedmodes[x] = modearray[supportedmodes[x]]	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof modearray[ocf.supportedmodes[x]]
mode	oic.r.mode	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]ocf.mode[0] = modearray[mode]	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]mode = indexof modeArray[ocf.mode[0]]

Table 53 provides the details of the Properties that are part of "asa.operation.climatecontrolmode".

Table 53 – The properties of "asa.operation.climatecontrolmode"

AllJoyn Property name	Type	Required	Description
operationalstate	integer	yes	Current status of device
supportedmodes	array	yes	Array of supported modes
mode	integer	yes	Current mode of device.

9.9.3 Derived model definition

```
{
  "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.climatecontrolmode.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Climate Control Mode",
  "definitions": {
```

```
    "asa.operation.climatecontrolmode": {
      "type": "object",
      "properties": {
        "mode": {
          "type": "integer",
          "description": "Current mode of device.",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.mode",
            "x-to-ocf": [
              "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
              "ocf.mode[0] = modearray[mode]"
            ],
            "x-from-ocf": [
              "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
              "mode = indexof modeArray[ocf.mode[0]]"
            ]
          }
        },
        "supportedmodes": {
          "type": "array",
          "items": {
            "type": "integer"
          },
          "description": "Array of supported modes",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.mode",
            "x-to-ocf": [
              "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
              "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
modearray[supportedmodes[x]]"
            ],
            "x-from-ocf": [
              "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
              "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
modearray[ocf.supportedmodes[x]]"
            ]
          }
        },
        "operationalstate": {
          "type": "integer",
          "description": "Current status of device",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.operationalstate",
            "x-to-ocf": [
              "machinestates = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
              "currentmachinestate = machinestates[operationalstate]"
            ],
            "x-from-ocf": [
              "statearray = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
              "operationalstate = indexof statearray[currentmachinestate[0]]"
            ]
          }
        }
      }
    }
  },
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/asa.operation.climatecontrolmode"}
  ],
  "required": [ "mode","supportedmodes","operationalstate" ]
}
```

9.10 Closed status

9.10.1 Derived model

The derived model: "asa.operation.closedstatus".

9.10.2 Property definition

Table 54 provides the detailed per Property mapping for "asa.operation.closedstatus".

Table 54 – The property mapping for "asa.operation.closedstatus"

AllJoyn Property name	OCF Resource	To OCF	From OCF
isclosed	oic.r.door	if isClosed ocf.openState = Closed.if !isClosed ocf.openState = Open.	isClosed = (openState == Closed)

Table 55 provides the details of the Properties that are part of "asa.operation.closedstatus".

Table 55 – The properties of "asa.operation.closedstatus"

AllJoyn Property name	Type	Required	Description
isclosed	boolean	yes	Open/Closed status Indicator

9.10.3 Derived model definition

```
{
  "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.closedstatus.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Closed Status",
  "definitions": {
    "asa.operation.closedstatus": {
      "type": "object",
      "properties": {
        "isclosed": {
          "type": "boolean",
          "description": "Open/Closed status Indicator",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.door",
            "x-to-ocf": [
              "if isClosed ocf.openState = Closed.",
              "if !isClosed ocf.openState = Open."
            ],
            "x-from-ocf": [
              "isClosed = (openState == Closed)"
            ]
          }
        }
      }
    }
  },
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/asa.operation.closedstatus"}
  ],
  "required": [ "isclosed"]
}
```


9.11 Cycle control

9.11.1 Derived model

The derived model: "asa.operation.cyclecontrol".

9.11.2 Property definition

Table 56 provides the detailed per Property mapping for "asa.operation.cyclecontrol".

Table 56 – The property mapping for "asa.operation.cyclecontrol"

AllJoyn Property name	OCF Resource	To OCF	From OCF
operationalstate	oic.r.operationalstate	statearray = [Idle,Working,ReadyToStart,DelayedStart,Pause,EndOfCycle]currentmachinestate = statearray[operationalstate]	statearray = [Idle,Working,ReadyToStart,DelayedStart,Pause,EndOfCycle]operationalstate = indexof statearray[currentmachinestate[0]]
executeoperationalcomand	oic.r.action		
SupportedOperationalcommands	oic.r.action		
supportedoperationalstates	oic.r.operationalstate	statearray = [Idle,Working,ReadyToStart,DelayedStart,Pause,EndOfCycle]for x=0, x < sizeof(supportedoperationalstates): machinestates[x] = statearray[supportedoperationalstates[x]]	statearray = [Idle,Working,ReadyToStart,DelayedStart,Pause,EndOfCycle]for x=0, x < sizeof(machinestates): supportedoperationalstates[x] = indexof statearray[machinestates[x]]

Table 57 provides the details of the Properties that are part of "asa.operation.cyclecontrol".

Table 57 – The properties of "asa.operation.cyclecontrol"

AllJoyn Property name	Type	Required	Description
operationalstate	integer	yes	Current operational state of the appliance
executeoperationalcomand		no	Execute an operational command
SupportedOperationalcommands	array	no	Array of operational commands supported by the appliance
supportedoperationalstates	array	yes	Array of operational states supported by the Appliance.

9.11.3 Derived model definition

```
{
  "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.cyclecontrol.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Cycle Control",
  "definitions": {
    "asa.operation.cyclecontrol": {
      "type": "object",
      "properties": {
        "operationalstate": {
          "type": "integer",
          "description": "Current operational state of the appliance",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.operationalstate",
            "x-to-ocf": [
              "statearray = [Idle,Working,ReadyToStart,DelayedStart,Pause,EndOfCycle]",
              "currentmachinestate = statearray[operationalstate]"
            ],
            "x-from-ocf": [
              "statearray = [Idle,Working,ReadyToStart,DelayedStart,Pause,EndOfCycle]",
              "operationalstate = indexof statearray[currentmachinestate[0]]"
            ]
          }
        },
        "supportedoperationalstates": {
          "type": "array",
```

```
          "items": {
            "type": "integer"
          },
          "description": "Array of operational states supported by the Appliance.",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.operationalstate",
            "x-to-ocf": [
              "statearray = [Idle,Working,ReadyToStart,DelayedStart,Pause,EndOfCycle]",
              "for x=0, x < sizeof(supportedoperationalstates): machinestates[x] =
statearray[supportedoperationalstates[x]]"
            ],
            "x-from-ocf": [
              "statearray = [Idle,Working,ReadyToStart,DelayedStart,Pause,EndOfCycle]",
              "for x=0, x < sizeof(machinestates): supportedoperationalstates[x] = indexof
statearray[machinestates[x]]"
            ]
          }
        },
        "SupportedOperationalcommands": {
          "type": "array",
          "items": {
            "type": "integer"
          },
          "description": "Array of operatinal commands supported by the appliance",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.action"
          }
        },
        "executeoperationalcomand": {
          "x-ocf-type": "method",
          "description": "Execute an operational command",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.action"
          }
        }
      }
    }
  },
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/asa.operation.cyclecontrol"}
  ],
  "required": [ "operationalstate","supportedoperationalstates" ]
}
```


9.12 Fan speed level

9.12.1 Derived model

The derived model: "asa.operation.fanspeedlevel".

9.12.2 Property definition

Table 58 provides the detailed per Property mapping for "asa.operation.fanspeedlevel".

Table 58 – The property mapping for "asa.operation.fanspeedlevel"

AllJoyn Property name	OCF Resource	To OCF	From OCF
fanspeedlevel	oic.r.airflow	speed = fanspeedlevel	fanspeedlevel = speed
maxfanspeedlevel	oic.r.airflow	range[0] = 0range[1] = maxfanspeedlevel	maxfanspeedlevel = range[1]otherwise: maxfanspeedlevel = 100
automode	oic.r.airflow	if automode != NotSupported(0xFF) ocf.automode = automodeelse no mapping	automode = ocf.automodeotherwise: automode = NotSupported(0xFF)

Table 59 provides the details of the Properties that are part of "asa.operation.fanspeedlevel".

Table 59 – The properties of "asa.operation.fanspeedlevel"

AllJoyn Property name	Type	Required	Description
fanspeedlevel	integer	yes	Fan speed level. 0 = off.
maxfanspeedlevel	integer	yes	Max level allowed for fan speed
automode	integer	yes	Auto mode status.

9.12.3 Derived model definition

```
{
  "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.fanspeedlevel.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Fan Speed Level",
  "definitions": {
    "asa.operation.fanspeedlevel": {
      "type": "object",
      "properties": {
        "fanspeedlevel": {
          "type": "integer",
          "description": "Fan speed level. 0 = off.",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airflow",
            "x-to-ocf": [
              "speed = fanspeedlevel"
            ],
            "x-from-ocf": [
              "fanspeedlevel = speed"
            ]
          }
        },
        "maxfanspeedlevel": {
          "type": "integer",
          "description": "Max level allowed for fan speed",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airflow",
            "x-to-ocf": [
              "range[0] = 0",
              "range[1] = maxfanspeedlevel"
            ],
            "x-from-ocf": [
              "maxfanspeedlevel = range[1]",
              "otherwise: maxfanspeedlevel = 100"
            ]
          }
        },
        "automode": {
          "type": "integer",
          "description": "Auto mode status.",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airflow",
            "x-to-ocf": [
              "if automode != NotSupported(0xFF)",
              " ocf.automode = automode",
              "else no mapping"
            ],
            "x-from-ocf": [
              "automode = ocf.automode",
              "otherwise: automode = NotSupported(0xFF)"
            ]
          }
        }
      }
    }
  },
  "type": "object",
  "allOf": [
```