ETSI TS 118 121 V2.0.0 (2016-09)

oneM2M;
oneM2M and AllJoyn® Interworking
(oneM2M TS-0021 version 2.0.0 Release 2)

Reference
DTS/oneM2M-000021

Keywords
interworking, IoT, M2M

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Partnership Project oneM2M (oneM2M).

1 Scope

The present document specifies the oneM2M and AllJoyn® interworking technologies that enable AllJoyn® Applications and oneM2M entities produce/consume services.

Clause 5 defines the interworking architecture model that describes where the AllJoyn® IPE is hosted and how the IPE is composed with.

Clause 6 defines the architecture aspects that mainly describes AllJoyn® services to oneM2M resource mapping structure and rules. Furthermore, this explains the IPE registration and AllJoyn® service discovery by the IPE.

Clause 7 defines the IPE interworking procedures that consists of the initial setup and the service interworking procedures.

NOTE: AllJoyn® is the trade name of a product supplied by the Allseen Alliance, Inc. This information is given for the convenience of users of the present document and does not constitute an endorsement by ETSI of the product named. Equivalent products may be used if they can be shown to lead to the same results."

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 118 101: "oneM2M; Functional Architecture (oneM2M TS-0001)".

[2] ETSI TS 118 104: "oneM2M; Core Protocol (oneM2M TS-0004)".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] oneM2M Drafting Rules.

NOTE: Available at http://www.onem2m.org/images/files/oneM2M-Drafting-Rules.pdf.

[i.2] AllJoyn® System Description.

NOTE: Available at http://allseenalliance.org/framework/documentation.

[i.3] oneM2M TR-0014: "oneM2M and AllJoyn® Interworking".

[i.4] D-Bus Tutorial.

NOTE: Available at https://dbus.freedesktop.org/doc/dbus-tutorial.html.

3 Definitions

For the purposes of the present document, the following terms and definitions apply:

AllJoyn application: application that either produces an AllJoyn service or consumes an AllJoyn service

AllJoyn consumer application: role of an AllJoyn application that consumes AllJoyn services provided by other AllJoyn applications

AllJoyn producer application: role of an AllJoyn application that produces AllJoyn services consumed by other AllJoyn applications

AllJoyn service resources: resources created as children of an <AE> resource representing an AllJoyn IPE for the purpose of exposing services between AllJoyn Consumer/Producer Applications and AE/CSE entities

4 Conventions

The key words "Shall", "Shall not", "May", "Need not", "Should", "Should not" in the present document are to be interpreted as described in the oneM2M Drafting Rules [i.1].

5 Architecture Model

5.1 Reference model

The architecture model followed in the present document is based on the architecture model in annex F of ETSI TS 118 101 [1] that describes how interworking between oneM2M CSEs and non-oneM2M systems using specialized Interworking Proxy application Entities (IPE). The present document descibes the AllJoyn IPE that supports the following reference model.

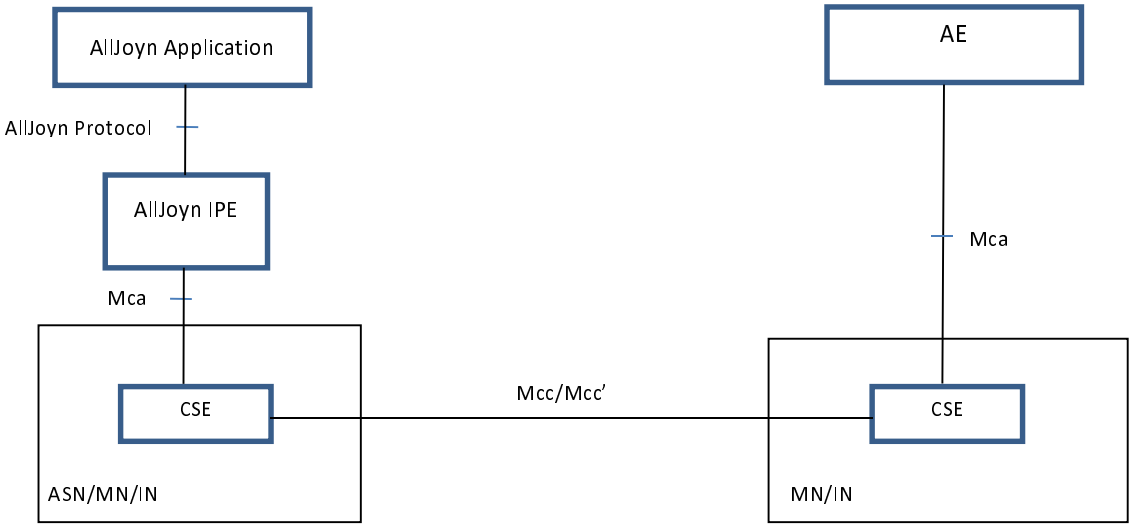

Figure 5.1-1: AllJoyn interworking reference model

This reference model describes how the oneM2M AEs can interwork with AllJoyn applications. By the IPE, AllJoyn services are exposed as oneM2M resources and stored in a CSE. AEs access the exposed AllJoyn service resources by the oneM2M APIs.

5.2 Composition of the IPE

The AllJoyn IPE consists of AE and AllJoyn application. To provide the interworking functions to other oneM2M entities, the IPE registers to a CSE and communicate with other AllJoyn applications using AllJoyn protocols. The IPE registration is mandatory in oneM2M systems which is not defined in AllJoyn system. AllJoyn discovery and session establishment are needed for the IPE to communicate with other AllJoyn applications.

In figure 5.2-1, AllJoyn Application on the AllJoyn device and AllJoyn IPE is either AllJoyn Consumer or Producer Application. Depending on the interworking service scenario, AllJoyn Consumer and/or Producer Application is hosted on AllJoyn devices and AllJoyn IPEs. For example, when a AllJoyn service is consumed by oneM2M entities, an AllJoyn devices hosts AllJoyn Producer Application and AllJoyn IPE hosts AllJoyn Consumer Application. In other case, if a oneM2M entity provides data to be consumed by AllJoyn Consumer Application on an AllJoyn device, then the AllJoyn IPE hosts AllJoyn Producer Application and the AllJoyn device hosts AllJoyn Consumer Application.

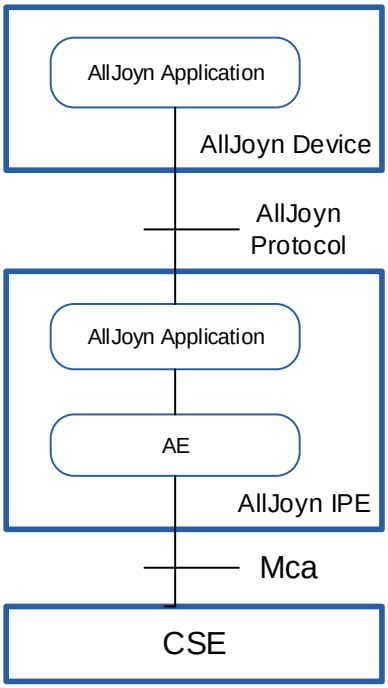

Figure 5.2-1: Composition of the AllJoyn IPE

6 Architecture Aspect

6.1 Introduction

oneM2M and AllJoyn have different architecture aspects (annex A oneM2M and AllJoyn technical comparison). The present document specifies the interworking functions for those differences in the following aspects:

- oneM2M resource mapping structure;
- AllJoyn IPE registration;
- AllJoyn service discovery;
- AllJoyn service mapping;
- AllJoyn access control mapping.

6.2 oneM2M resource mapping structure

In this clause, the overall resource mapping structure for exposing service between AllJoyn Applications(either Producer or Consumer) and oneM2M entities is intrtoduced. Since it is difficult to put the resource tree into a single figure that remains well-readable, the description is split up in multiple figures, In this description, not all specified attributes of the used resource types and not all possible child resources are depicted to keep the figures readable, For example <subscription> resources which may be used to subscribe to some of the depicted resources are not shown.

The resource mapping structure uses some specializations of the <flexContainer> resource. Each of this specialization is detailed in annex B.

The use of each of the resources within the resource mapping structure for consuming or producing services is defined in detail in clause 7.

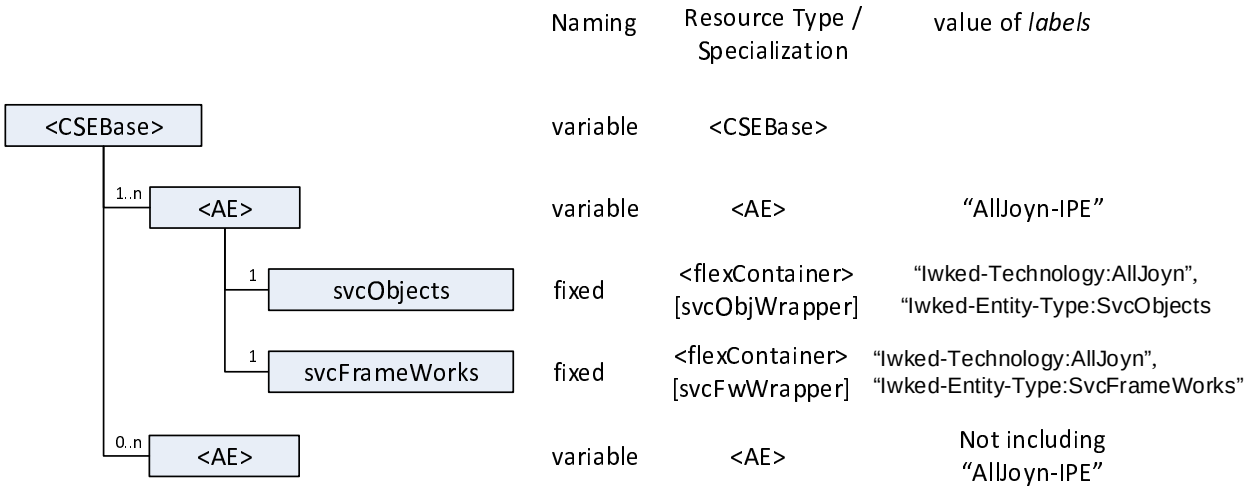

Figure 6.2-1: AllJoyn interworking reseource mapping, part 1

In figure 6.2-1, the first part of the resource mapping structure for AllJoyn interworking is depicted. The <CSEBase> resource of a CSE that is used for connecting to an AllJoyn proximal network is the root of the resource mapping structure.

The IPE responsible for AllJoyn interworking - see the reference model described in clause 5 - will register with that CSE and will create an <AE> resource which represents the IPE. This <AE> resource shall include the string "AllJoyn-IPE" in the *labels* attribute. All other relevant AllJoyn interworking resources will be hosted as children or descendants of this <AE> resource. For convenience, this <AE> is termed "IPE" in what follows. In figure 6.2-1 the multiplicity of the <AE> resource is defined to be equal to 1..n. Since it is assumed that the CSE represented by the <CSEBase> resource depicted in figure 6.2-1 supports AllJoyn interworking, atr least one <AE> respurce representing an AllJyon IPE shall exist. When the IPE registers with an ASN-CSE or an MN-CSE, the multiplicity shall be 1. This does not imply that no other application can register with that CSE. It is only meant to define that only one <AE> will act as an IPE for AllJoyn interworking on this CSE. As depicted in figure 6.2-1, other <AE> resources - not including the string "AllJoyn-IPE" in the value of the *labels* attribute can occur with multiplicity 0..n. On the other hand, when the IPE registers with IN-CSE, the multiplicity shall be 1..n to allow interworking with multiple different AllJoyn proximal networks. In figure 6.2-1, and in the following figures two specializations of <flexContainer> called "svcObjWrapper" and "svcFwWrapper" are used. These specializations of <flexContainer> do not contain any custom attributes. They serve only as wrappers for other resources in order to structure the resource tree allowing subscriptions to be made to that part of the tree. See Annex B for further details on the [svcObjwrapper] and [svcFwWrapper] specialization of <flexContainer>.

Two child resources of the <AE> resource representing the IPE are depicted in figure 6.2-1:

- A resource with a fixed name "svcObjects": It serves as a wrapper for hosting resources that are used to expose services based on the AllJoyn service object model, see also oneM2M TR-0014 [i.3] for details on AllJoyn service objects. Service Objects in AllJoyn contain one or more AllJoyn interfaces that are used to interact with services. The resource structure under this [svcObjWrapper] resource called "svcObjects" allows for direct interaction with AllJoyn interfaces via corresponding resources.

- A resource with a fixed name "svcFrameWorks": It serves as a wrapper for hosting resources that are used to expose services based on some higher level AllJoyn service frameworks, see also oneM2M TR-0014 [i.3] for details on AllJoyn service frameworks. AllJoyn service frameworks typically offer a simpler API to programmers for making use of services compared to operating directly on the AllJoyn interfaces within AllJoyn service objects. An example for such a service framework is the Lighting Services Framework (LSF) in AllJoyn. The resource structure under this [svcFwWrapper] resource called "svcFrameWorks" allows use of functions in selected AllJoyn service frameworks via corresponding resources.

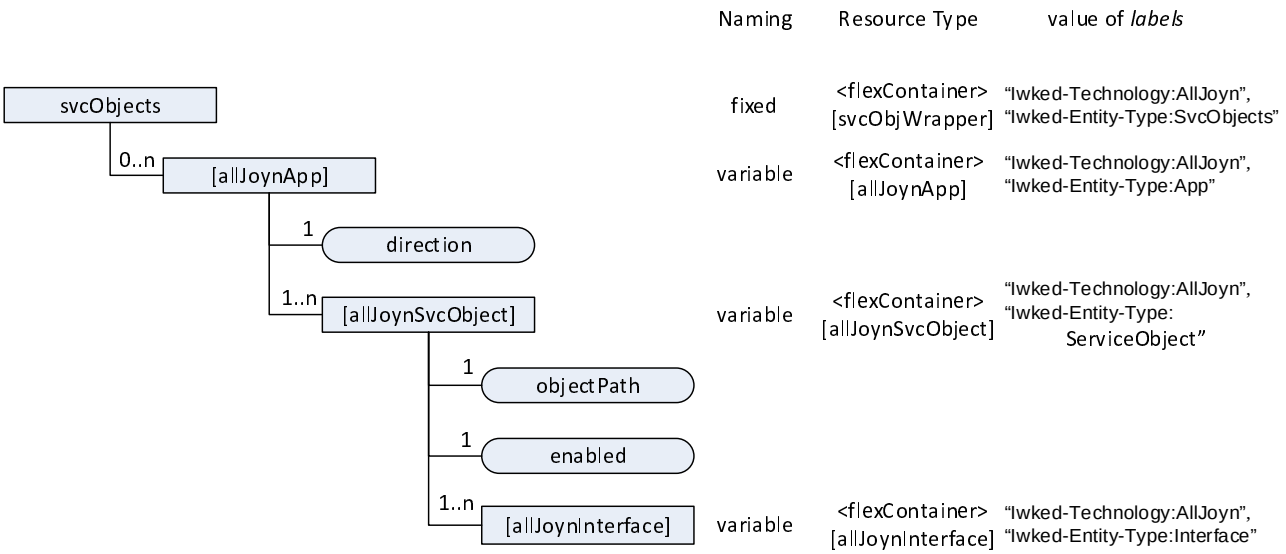

Figure 6.2-2: AllJoyn interworking reseource mapping, part 2

In figure 6.2-2, the second part of the AllJoyn interworking resource mapping is depicted. It describes the structure of resources hosted under the [svcObjWrapper] resource named "svcObjects".

For each AllJoyn application providing a service that is supposed to be exposed by the IPE via resources under the "svcObjects" [svcObjWrapper] resource and for all external services that shall be injected into the AllJoyn proximal network by the IPE, a separate child resource is created which is a [allJoynApp] specialization of <flexContainer>. The details of this [allJoynApp] specialization of <flexContainer> are defined in annex B.

The [allJoynApp] specialization of <flexContainer> contains a custom attribute *direction*, which indicates whether the service is exposed from the AllJoyn side to the oneM2M side (value equal to "allJoynToOneM2m") or vice versa (value equal to "oneM2mToAllJoyn"). At most one of the [allJoynApp] child resources under the "svcObjects" [svcObjWrapper] resource can have a direction set to "oneM2mToAllJoyn". For that particular [allJoynApp] - if present - the IPE acts as a service producer application at the AllJoyn side. It is the IPE's responsibility to create the [allJoynApp] resources under the "svcObjects" [svcObjWrapper] resource including:

- One [allJoynApp] specialization of <flexContainer> resource for each AllJoyn application providing a service that is supposed to be exposed by the IPE towards oneM2M entities. The name of each of these [allJoynApp] specialization shall be set to the unique Well-Known-Name (WKN) [i.2] of the corresponding AllJoyn applicaton. The value for the *direction* attribute shall be set to "allJoynToOneM2m".

- One [allJoynApp] specialization of <flexContainer> resource representing the IPE itself as an AllJoyn Producer Application hosted by the IPE - in case the IPE supports exposure of services provided by oneM2M AEs to AllJoyn applications. The name of this [allJoynApp] specialization of <flexContainer> shall be set to the Well-Known-Name (WKN) of the IPE on the AllJoyn bus. The value for the *direction* attribute shall be set to "oneM2mToAllJoyn".

Each [allJoynApp] resource will contain one or more [allJoynSvcObject] child resources, which are specializations of <flexContainer> resources representing AllJoyn service objects. Details of the [allJoynSvcObject] specialization of <flexContainer> are defined in Annex B. The [allJoynSvcObject] specialization of <flexContainer> contain two custom attributes:

- *objectPath*: This is a string equal to the AllJoyn object path for the represented AllJoyn service object.

- *enabled*: This is a flag indicating whether the resource respresening the corresponding AllJoyn service object is currently enabled for service exposure. This is needed since the structure in this resource needs to be created in multiple steps (creation of children and children of children in this resource structure). During initialization of the resource structure under a specific [allJoynSvcObject] resrouce, the attribute *enabled* shall be set to False as it is not ready for use. Once the initialization is complete and the resource structure can be used for service interactions, the *enabled* flag shall be set to True.

The naming of [allJoynSvcObject] is up to the IPE. It would seem logical to derive names from the object path of the corresponding AllJoyn service object, but due to characters like a slash "/" appearing in object paths, it is not possible to use the exact same string as a name for the corresponding [allJoynSvcObject] resource.

Each AllJoyn service object contains one or more interfaces. This is reflected in the AllJoyn interworking resource structure by the presence of one or more [allJoynInterface] specializations of <flexContainer> under each [allJoynSvcObject] resource. Details of the [allJoynInterface] specializations of <flexContainer> resources are defined in annex B. The basic structure under each [allJoynInterface] resource is depicted in figure 6.2-3.

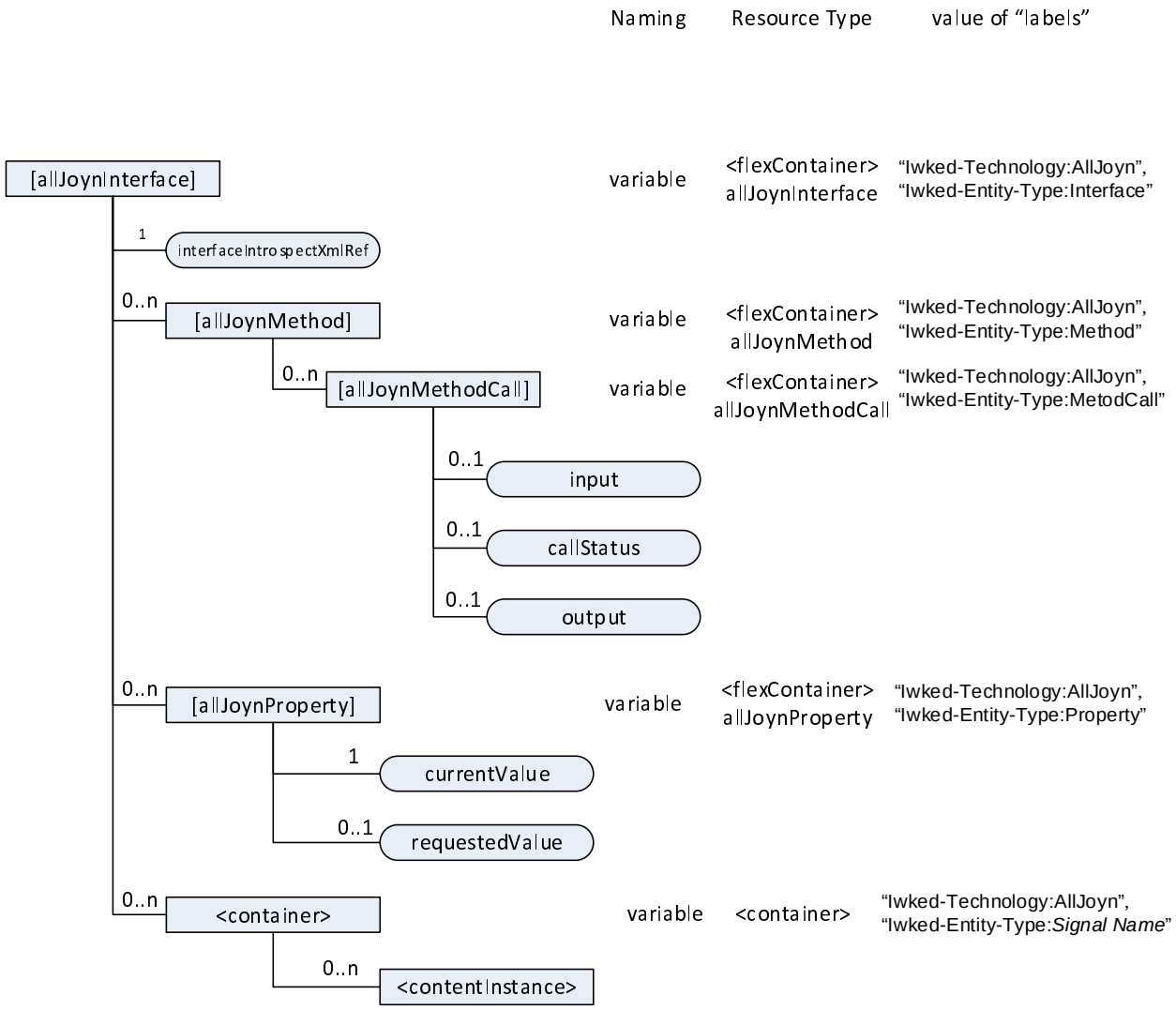

Figure 6.2-3: AllJoyn interworking reseource mapping, part 3

[allJoynInterface] specializations of <flexContainer> contain one custom attribute *interfaceIntrospectXmlRef* which refers to the AllJoyn interface definition in XML. AllJoyn interfaces consist of methods, properties and signals. In the AllJoyn interworking resource structure, methods are represented by [allJoynMethod] resources which are a specialization of <flexContainer>. Properties are represented by [allJoynProperty] resources which are a specialization of <flexContainer>. Signals are respresented by regular <container> resources. All specializations of <flexContainer> resources are defined in annex B. [allJoynMethod] resources shall be named with the same string as the method is named in the AllJoyn interface definition. [allJoynProperty] resources shall be named with the same string as the property is named in the AllJoyn interface definition.

Since different calling instances of AllJoyn interface methods need to be distinguishable in terms of input, output and call status, the interworking resource structure for methods includes zero or more [allJoynMethodCall] resources used to represent individual calling instances of a method. For each call of a method, the consumer of the service needs to create a new instance of a [allJoynMethodCall] resource. In cases where the corresponding AllJoyn method requires inputs, the *input* attribute needs to be present and contain a valid serialization of input parameters to the method call. The entity which acts on creation of a new [allJoynMethodCall] resource - i.e. the IPE in case of a service exposed from AllJoyn to oneM2M or another AE in case of exposing a service from oneM2M to AllJoyn - needs to perform the requested method call and populate the *callStatus* attribute with corresponding status information. Upon completion of the action taken, the *ouput* attribute shall get populated with the corresponding output of the method call - if any. Note that some AllJoyn methods do not return an output upon completion of a call.

AllJoyn properties are represented by [allJoynProperty] resources. The attribute *currentValue* is representing the current value of the property. When the property is capable of being changed, the attribute *requestedValue* shall be used to request a change to a new value. Upon completion of such a change request, the IPE or the service providing AE will update the new value of the property in the attribute *currentValue.*

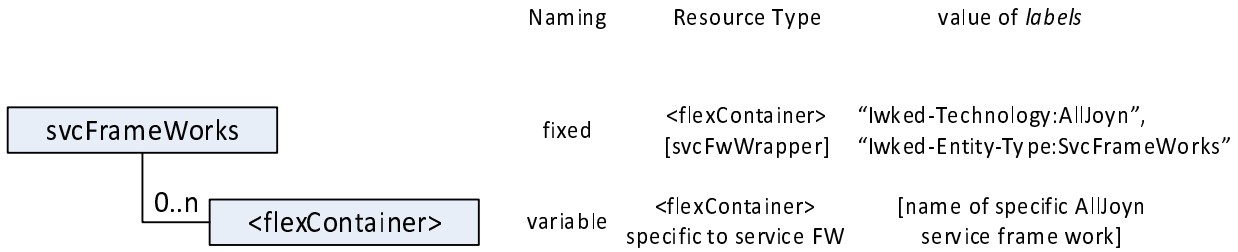

Figure 6.2-4: AllJoyn interworking reseource mapping, part 4

For exposure of AllJoyn service frameworks using higher level APIs that are not directly interacting with AllJoyn service objetcs and interfaces, separate resources are used under the [svcFwWrapper] resource named "svcFrameWorks". For each supported service framework with a higher level API, a separate specialization of a <flexContainer> resource shall be created as a child resource under "svcFrameWorks" in which the *labels* attribute includes a stringequal to the name of the service framework. An example for an AllJoyn service framework that could be exposed in this part of the resource structure is the "Lighting Services Framework" (LSF).

Since each service framework will have different services and methods associated with it, a separate definition of the resource structure for that service framework is needed. Supported service frameworks are FFS.

6.3 AllJoyn IPE registration

To start oneM2M-AllJoyn interworking, the IPE shall perform AE registration procedure to the a CSE specified in ETSI TS 118 101 [1]. To stop using the interworking function, IPE shall perform AE de-registration procedure.

The IPE shall be represented as the <AE> resource as follows.

Table 6.3-1: Attribute values of *IPE* resource

Attribute	Value	Description
resourceType	"2"	Enumeration value for the *AE* resource type
labels	"AllJoyn-IPE"	IPE for AllJoyn interworking

For each AllJoyn proximal network, only one IPE handles interworking between oneM2M entities and AllJoyn Applications. An ASN-CSE or MN-CSE shall accept at most one AllJoyn IPE. However, an IN-CSE may accept more than one AllJoyn IPE registration(s) to allow multiple AllJoyn proximity network interworking.

After the registration, the IPE may perform AllJoyn service discovery.

6.4 AllJoyn service discovery

After the IPE registration, the IPE performs AllJoyn service discovery. The IPE may provide the following AllJoyn service discovery modes:

- Advertisement-based discovery mode: This modes utilizes AllJoyn multicast service advertisement. When the IPE gets service advertisements from AllJoyn service provider applications in an AllJoyn proximal network, the IPE can expose those services in oneM2M system without oneM2M AE's request.
- Pre-configured (IPE initiated) discovery mode: This mode utilizes AllJoyn multicast service discovery. Interested AllJoyn services are pre-cofigured in the IPE (e.g. by M2M Service Provider), so the discovery is performed automatically once the IPE registration is complete.
- On-demand (AE initiated) discovery mode: This mode utilizes AllJoyn multicast service discovery. Interested Alljoyn services are requested by oneM2M AEs, so the discovery is performed upon AE's service discovery request.

Selection of supported mode(s) is implementation specific.

After the successful AllJoyn service discovery, the discovered services - if and only if allowed by the AllJoyn side (e.g. in line with provisioned policies on an AllJoyn Gateway Agent) - shall be exposed as oneM2M resources. When new services are discovered on the AllJoyn side after initial service discovery has already been completed by the IPE, the IPE can add those additional services to the set of exposed services by means of dynamically creating the corresponding resources on the CSE at which the IPE is registered. See clause 7 for more details.

6.5 AllJoyn service mapping

6.5.1 AllJoyn service categorization

AllJoyn services can be categorized into:

- AllJoyn framework services (pre-defined by AllJoyn service framework).
- App-specific services (implemented using application developer defined AllJoyn interfaces).

AllJoyn framework services are pre-defined and can be sub-categorized into two: base services (e.g. Notification, Control Pannel) and the other framework services (e.g. Lighting service framework). The base services have relatively simple interface member composition so they can be easily mapped with the resource structure defined in clause 6.2 under the [svcObjWrapper] specialization of <flecContainer>. For a subset of the AllJoyn base services, a normative resource mapping is specified in annex C. Selected other AllJoyn framework services - i.e. not the base services - shall be exposed using separate resource structures and this is FFS.

6.5.2 AllJoyn base service mapping

The list of AllJoyn base services [i.2]:

- OnBoarding.
- Configuration.
- Notification.
- ControlPanel.
- Audio Streaming.

The present document specifies base services mapping for Configuration, Notification, ControlPanel. See annex C for the AllJoyn base services mapping.

For Configuration, Notification or ControlPanel services which are discovered on the AllJoyn side and are intended to be exposed to the oneM2M side, the mapping specified in Annex C is normative.

OnBoarding service is intended to support AllJoyn network onboarding (Wi-Fi connection support) for the device that does not have user interface. This is out of scope for oneM2M interworking.

Audio Streaming is also out of scope since oneM2M system does not support in-band streaming feature.

6.5.3 Other framework service mapping

For example, AllJoyn Lighting Services Framework mapping can be specified as described in the figure 6.2-4. However, this is not specified in the present document.

6.5.4 AllJoyn app-specific service mapping

App-specific services implement AllJoyn interface(s) defined by 3rd party developers. In the AllJoyn system, the introspection feature can be used to deliver interface information of an AllJoyn Producer Application to AllJoyn Consumer Applications. Mapping for AllJoyn app-specific services shall follow the resource mapping structure in clause 6.2 and the <flexContainer> resource specializations in annex B.

6.6 AllJoyn access control mapping

AllJoyn security 1.0 feature supports authentication and encryption, but not authorization.

AllJoyn security 2.0 feature supports permission management (authorization), which is similar to oneM2M access control mechanism, as well as authentication and encryption.

NOTE: The present document does not specify AllJoyn security 2.0 interworking.

7 Interworking Procedures

7.1 Introduction

The interworking procedures describe the following:

- how does the IPE initially creates necessary resources for oneM2M-AllJoyn interworking (clause 7.2);
- how does the IPE performs interworking between the oneM2M system and AllJoyn networks depending on the service exposure direction (clause 7.3).

7.2 IPE initial configuration

7.2.1 Introduction

The initial configuration of the IPE consists of registration with its Registrar CSE and resource creation procedures. As part of registration, an <AE> resource is created to represent the AE part of the IPE. After that, AllJoyn-specific resources to support interaction between oneM2M AE/CSE(s) and AllJoyn Consumer and/or Producer Applications as well as necessary <subscription> resources are created by the IPE.

The IPE shall process incoming notifications from the oneM2M CSEs which have been triggered by the created <subscription> resources and react to them by executing the appropriate actions to the interworked AllJoyn Consumer and/or Producer Applications. Accordingly, the IPE shall react to messages received from interworked AllJoyn Consumer and/or Producer Applications that are relevant to the exposed services and do any necessary changes to the resources defined for AllJoyn interworking within the oneM2M CSE.

7.2.2 IPE registration and AllJoyn service discovery

For the interworking, firstly the IPE shall perform IPE registration (see clause 6.3). The '*labels*' attribute shall include the string "AllJoyn-IPE".

If the IPE is set to advertisement-based or pre-configured discovery mode (see clasue 6.4), the AE in the IPE shall create and expose resources corresponding to the exposed services (see clause 6.2) once the AllJoyn Application in the IPE completes AllJoyn service discovery and session establishment with AllJoyn Producer Applications referring its AllJoyn service objects.

If the IPE is set to on-demand discovery mode (see clasue 6.4), the IPE shall perform the same procedure above (i.e. advertisement-based or pre-configured discovery mode) when it receives a Notify request for on-demand discovery (see clause 10.2.6.2 [1]) by its Registrar CSE and then create and expose resources corresponding to the exposed services.

Which resources are created for service exposure from AllJoyn Producer Application to oneM2M AE/CSE(s) is described in more detail in the clause 7.2.3 and 7.2.4.

If the IPE supports exposure of services provided by other oneM2M AE/CSE(s) to AllJoyn Consumer Applications, the IPE shall create one instance of an [allJoynApp] specialization of the <flexContainer> resource representing the AllJoyn Producer Application inside the IPE itself (see clause 6.2). Other AE/CSE(s) providing services that are meant to be exposed to AllJoyn Consumer Applications on AllJoyn devices shall use that [allJoynApp] resource representing the AllJoyn Producer Application part inside the IPE itself as the parent of any [allJoynInterface] resources they may create for offering a service(s) to the AllJoyn Consumer Applications. The particular [allJoynApp] resource representing the AllJoyn Producer Application inside the IPE itself shall be identified by the value of the *direction* attribute which is set to 'oneM2MToAllJoyn'.

7.2.3 AllJoyn Service Resource creation

The AllJoyn Service Resources listed in table 7.2.3-1 shall only be created by the AllJoyn IPE. The listed resources are created as children of the <AE> resource instance representing the AllJoyn IPE according to the tree structure and resource attribute values defined in clause 6.2.

Table 7.2.3-1: Resources created by IPE only

AllJoyn Service Aspect	Resource Type	
	Specialization	Resource Type
Grouping together service objects for exposure	[*svcObjWrapper*]	flexContainer
Grouping together service frameworks for exposure	[*svcFwWrapper*]	flexContainer
Representation of an individual AllJoyn Consumer or Producer Application	[*allJoynApp*]	flexContainer

The parent/child-relationship of these resources and their multiplicity are summarized in figure 7.2.3-1. After registration and service advertisement/discovery (see clause 7.2.2), an AllJoyn IPE shall create two children for the <AE> resource that represents this AllJoyn IPE:

- One resource instance named "svcOjects" of type <flexContainer> with specialization [*svcObjWrapper*] and including "Iwked-Technology:AllJoyn" and "Iwked-Entity-Type:SvcObjects" in the *labels* attribute.
- One resource instance named "svcFrameworks" of type <flexContainer> with specialization [*svcFwWrapper*] and including "Iwked-Technology:AllJoyn" and "Iwked-Entity-Type:SvcFrameWorks" in the *labels* attribute.

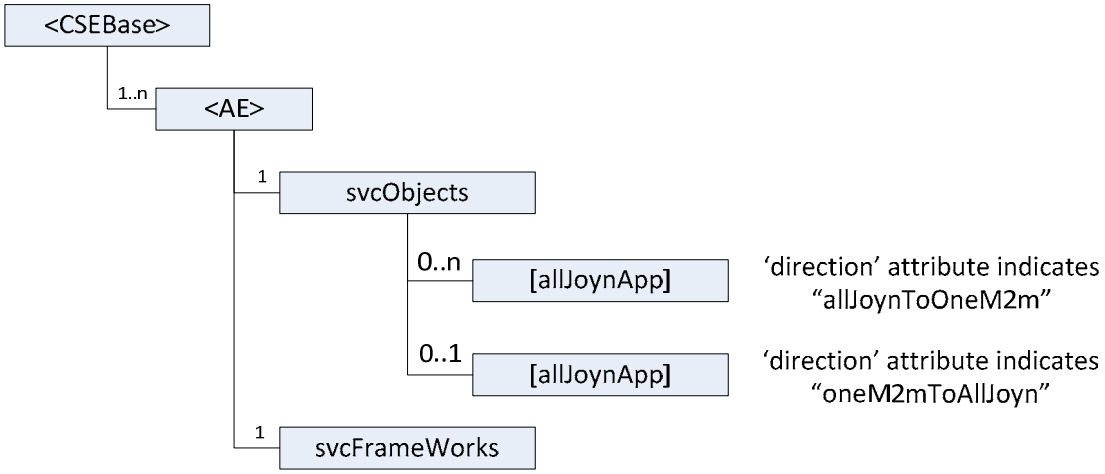

Figure 7.2.3-1: AllJoyn interworking reseource top level

For each discovered AllJoyn Producer Application that is supposed to be exposed to other oneM2M AE/CSE(s), the AllJoyn IPE shall create one instance of an [*allJoynApp*] specialization of a <flexContainer> resource with the *direction* attribute indicating "allJoynToOneM2M". The name of of each of these [allJoynApp] specialization instances shall be set to the unique Well-Known-Name (WKN) [i.2] of the corresponding AllJoyn Producer Application, see clause 6.2. All resources related to service exposure from AllJoyn Producer Applications toward AE/CSE(s) shall be children of these [*allJoynApp*] instances.

If the AllJoyn IPE supports exposure of services provided by other oneM2M AE/CSE(s) to AllJoyn Consumer Applications, it shall create one instance of an [*allJoynApp*] specialization of a <flexContainer> resource with the *direction* attribute indicating "oneM2MToAllJoyn" and named with the Well-Known-Name (WKN) [i.2] of the AllJoyn Producer Application part of the IPE.

The AllJoyn Service Resources listed in table 7.2.3-2 shall be created either:

- by the AllJoyn IPE for providing services from AllJoyn Producer Applications to other AE/CSE(s); or
- by other oneM2M AE/CSE(s) for providing services to AllJoyn Consumer Applications.

The listed resources are created as children of the instance of [*allJoynApp*] resources representing the corresponding AllJoyn Producer Applications inside the AllJoyn proximal network or the AllJoyn Producer Application part of the IPE in case it supports exposure of services provided by other oneM2M AE/CSE(s) into the AllJoyn proximal network.

Table 7.2.3-2: Resources created by AllJoyn IPE or by other oneM2M AE/CSE(s)

AllJoyn Service Aspect	Resource Type		Comments
	Specialization	Resource Type	
Representation of an individual service object for exposure	[*allJoynSvcObject*]	<flexContainer>	
Representation of an individual AllJoyn interface within an AllJoyn service Object	[*allJoynInterface*]	<flexContainer>	
Representation of an individual AllJoyn method within an AllJoyn interface	[*allJoynMethod*]	<flexContainer>	
Representation of an individual instance of calling a specific AllJoyn method	[*allJoynMethodCall*]	<flexContainer>	Instances of this <flexContainer> specialization are only created when a method call is attempted. Initially no instances of this <flexContainer> specializations are present in any instance of [*allJoynMethod*] specializations of <flexContainers>
Representation of an individual property within an AllJoyn interface	[*allJoynProperty*]	<flexContainer>	
Representation of an signal within an Alljoyn interface	N/A	<container>	Individual signal instances are represented by instances of <contentInstance> child resources within the <container> resource

Within each instance of an [*allJoynApp*] specialization of a <flexContainer> resource, the AllJoyn IPE (in case the *direction* attribute indicates "allJoynToOneM2M") or an oneM2M AE/CSE exposing a service to AllJoyn Conusmer Applications (in case *direction* attribute indicates "oneM2MToAllJoyn") shall create one instance of an [*allJoynSvcObject*] specialization of a <flexContainer> resource for each service to be exposed. The resource name of an [*allJoynSvcObject*] specialization of a <flexContainer> may be suggested by the Originator. It is recommended to derive it from the AllJoyn object path for the represented AllJoyn service object by properly replacing characters that are not allowed in resource names with adequate literals. Initially, when such an instance of an [*allJoynSvcObject*] specialization of a <flexContainer> is created, the attribute *enabled* shall be set to indicate "False" in order to make sure the resource is not used for interworking procedures until all other needed child resources have been created. The attribute *objectPath* of an [*allJoynSvcObject*] specialization of a <flexContainer> shall be set to a string equal to the AllJoyn object path for the represented AllJoyn service object. Once all needed child resources of an instance of an [*allJoynSvcObject*] specialization of a <flexContainer> are in place - see the following paragraphs and clause 7.2.4 - the *enabled* attribute shall be set to indicate "True". After that, the procedures for interworking between AllJoyn and oneM2M are supported for the exposed service as defined in clause 7.3.

The creator of an instance of an [*allJoynSvcObject*] resource - i.e. the AllJoyn IPE for cases of service exposure from AllJoyn to oneM2M or some other oneM2M AE/CSE for cases of service exposure from oneM2M to AllJoyn - is responsible for creating an instance of an [*allJoynInterface*] resource for each interface that the service is using, see also clause 6.2. Since interfaces in AllJoyn are comprised of methods, properties and signals, the creator of an [*allJoynInterface*] resource is also responsible for creating the necessary children:

- One instance of an [*allJoynMethod*] resource for each method of the respective AllJoyn interface. Instances of an [*allJoynMethod*] resource shall be named with the same string as the corresponding method is named in the AllJoyn interface definition. Instances of an [*allJoynMethod*] resource shall not contain any instances of an [*allJoynMethodCall*] resource before the *enabled* attribute of the [*allJoynSvcObject*] resource becomes "True".

- One instance of an [*allJoynProperty*] resource for each property of the respective AllJoyn interface. Instances of an [*allJoynProperty*] resource shall be named with the same string as the property is named in the AllJoyn interface definition. Instances of an [*allJoynProperty*] resource shall not contain the attribute '*requestedValue*' before the *enabled* attribute of the [*allJoynSvcObject*] resource becomes "True".

- One instance of a <container> resource for each signal defined in the AllJoyn interface definition. An instance of <container> which represents an AllJoyn signal shall be named with the same string as the signal is named in the AllJoyn interface definition.

7.2.4 Subscription creation

7.2.4.1 Subscription creation by AE/CSE(s) acting as service producers

In the context of AllJoyn interworking, the AllJoyn IPE and other AE/CSE(s) providing services to AllJoyn Consumer Applications are acting as producers of services to other entities:

- The AllJoyn IPE acts as a service producer towards other oneM2M AE/CSE(s) consuming the service. At the same time, the AllJoyn IPE acts as an AllJoyn Consumer Application inside the AllJoyn proximal network for the exposed service.
- Optional: Other oneM2M AE/CSE(s) providing services indirectly to AllJoyn Consumer Applications act as a service producer towards the AllJoyn IPE which is consuming the respective service. At the same time, the AllJoyn IPE acts as an AllJoyn Producer Application within the AllJoyn proximal network towards AllJoyn Consumer Applications.

Since these entities are acting as sevice producers on the oneM2M side, they need to monitor certain parts of the AllJoyn Service Resourcs in order to react to manipulation requests on these resources.

After the creation of AllJoyn Service Resources to expose a specific service, but before setting the *enable* attribute of the respective [*allJoynSvcObject*] resource to indicate "True" , the IPE and any AE/CSE(s) which are exposing services to AllJoyn Consumer Applications shall either create <subscription> resources to monitor attempts for AllJoyn service method call(s), property value change requests, signal generation by the IPE or other AE/CSE(s) manipulating the service resources or use frequent retrievals of those resource for the same purpose. Regarding signal generation the following subscription rules shall apply:

- A subscription to monitor signal generation by other oneM2M AE/CSE(s) within a <container> representing AllJoyn signals - i.e. to monitor the creation of instances of <contentInstance> resources - shall be created by the IPE if and only if the signal <container> resource is the great-grand child resource of [allJoynApp] resource with the *direction* attribute set to "oneM2MToAllJoyn".
- A subscription to monitor signal generation within a <container> representing AllJoyn signals - i.e. to monitor the creation of instances of <contentInstance> resources - shall be created by the AE/CSE providing the specific service to AllJoyn Consumer Applications if and only if the signal <container> resource is the great-grand child resource of [allJoynApp] resource with the *direction* attribute set to "allJoynToOneM2M".

When the IPE or an AE/CSE providing services to AllJoyn Consumer Applications create <subscription> resources, the <subscription> resources shall have the following attributes.

Table 7.2.4.1-1: Subscription for method call

Subscribed-to resource		**<subscription> resource**
resourceType	**Specialization**	**eventType of eventNotificationCriteria**
<flexContainer>	[allJoynMethod]	Creation of a direct child of the subscribed-to resource

Table 7.2.4.1-2: Subscription for property change

Subscribed-to resource		**<subscription> resource**
resourceType	**Specialization**	**eventType of eventNotificationCriteria**
<flexContainer>	[allJoynProperty]	Creation of a direct child of the subscribed-to resource

Table 7.2.4.1-3: Subscription for signal generation

Subscribed-to resource		**<subscription> resource**
resourceType	***labels* attribute include**	***eventType* of *eventNotificationCriteria***
<container>	"Iwked-Technology:AllJoyn", "Iwked-Entity-Type:Signal Name"	Creation of a direct child of the subscribed-to resource

When all required subscriptions for specific service represented by a specific [*allJoynSvcObject*] resource are created, the creator of the respective [*allJoynSvcObject*] resource shall set the *enabled* attribute to indicate "True". After that, the creator of the respective [*allJoynSvcObject*] resource shall act on reception of notifications triggered by the created subscriptions according to the procedures defined in clase 7.3.

7.2.4.2 Subscription creation by entities acting as service consumers

In the context of AllJoyn interworking, the AllJoyn IPE and other oneM2M AE/CSE(s) can act as consumers of services to other entities:

- In general, oneM2M AE/CSE(s) can act as service consumers towards the AllJoyn IPE which is producing services on the oneM2M side for services exposed from AllJoyn to oneM2M. At the same time the AllJoyn IPE acts as an AllJoyn Consumer Application for that service within the AllJoyn proximal network.

- The AllJoyn IPE acts as a service consumer towards other oneM2M AE/CSE(s) in case those other oneM2M AE/CSE(s) are providing services indirectly to AllJoyn Consumer Applications. At the same time the AllJoyn IPE acts as an AllJoyn Producer Application for these services on the AllJoyn proximal network.

For consuming services based on the AllJoyn service object model - which is represented by the structure of AllJoyn Service Resources in [*allJoynSvcObject*] resources - an oneM2M entity may need to subscribe to specific resources within the set of AllJoyn Service Resources instances in order to properly consume the respective service.

When an AllJoyn IPE supports exposure of services provided by other oneM2M AE/CSE(s) to AllJoyn Consumer Applications the following applies: In order for the AllJoyn IPE to be able to notice when a service provided by a oneM2M entity via the IPE to AllJoyn Consumer Applications gets initiated, it shall subscribe to the specific [*allJoynApp*] resource with the '*direction*' attribute indicating "oneM2MToAllJoyn" and named with the Well-Known-Name (WKN) [i.2] of the AllJoyn Producer Application part of the IPE.

The following cases for creating subscriptions in order to consume a service via a specific AllJoyn interface are relevant in context of AllJoyn interworking

- Method calls:

  - When a particular attempt to call a method is made (see clause 7.3 for details on how to initiate that), the consumer of the service is creates an instance of an [*allJoynMethodCall*] resource as a child of the respective [*allJoynMethod*] resource. In order to get notice of the status and outcome of the calling attempt, the service consumer should subscribe to the [*allJoynMethodCall*] resource. In particular the subscription should be limited to monitor creation/changes of the *callStatus* and *output* attributes. However, the same functionality can be achieved by retrieving that resource with sufficient frequency. In the case an AllJoyn IPE acts as a service consumer towards other oneM2M AE/CSE(s) in order to expose those services within the AllJoyn proximal network as an AllJoyn Producer Application, the AllJoyn IPE shall either subscribe to the [*allJoynMethodCall*] resource instances it has created in that role or retrieve with sufficient frequency that it gets to know about changes in the *callStatus* and *output* attributes of the [*allJoynMethodCall*] resources it created.

- Properties:

  - When a service consumer is interested in any changes of properties of an exposed AllJoyn interface, the consumer should subscribe to the respective [*allJoynProperty*] resource. Monitoring of properties is up to the consumer and is not required in general for consuming services based on the AllJoyn service object model. If an AllJoyn IPE is acting as a consumer of a service provided by a AE/CSE so that it can expose that service as an Alljoyn Producer Application inside the AllJoyn proximal network, it shall either subscribe to any [*allJoynProperty*] resource instances before it creates or modifies the *requestedValue* attribute of such a resource or frequently retrieve it in order to get to know about the outcome of such a request to change a property value.

- Signals:
  - When a service consumer is interested in receiving signals generated for a particular AllJoyn interface, the consumer should subscribe to the respective <container> resource representing that signal in the AllJoyn interface. However, the same functionality can be achieved by retrieving that resource with sufficient frequency. In the case an AllJoyn IPE acts as a service consumer towards other oneM2M AE/CSE(s) in order to expose those services within the AllJoyn proximal network as an AllJoyn Producer Application, the AllJoyn IPE shall either subscribe to all <container> resources representing signals in the exposed AllJoyn interface or retrieve them with sufficient frequency to know about the occurrence of new <contentInstance> resources representing new signal instances.

7.3 IPE interworking

7.3.1 Introduction

When the AllJoyn Service Resources are created by the IPE or an AE/CSE exposing services to the AllJoyn side, oneM2M AE/CSE(s) trying to consume those exposed services can manipulate certain parts of those AllJoyn Service Resources in order to initiate and process the service consumption.

Since an AllJoyn service is produced/consumed by an AllJoyn interface member access (i.e. by access to methods, properties, signals), the IPE delivers method call, property change and signal generation from/to oneM2M and AllJoyn systems.

7.3.2 AllJoyn services consumed by AE/CSE(s)

For an AllJoyn service that is consumed by AE/CSE(s) (i.e. *direction* attribute of the respective [*allJoynApp*] resource is indicating "allJoynToOneM2m"), the IPE performs the following:

- When a new instance of an [*allJoynMethodCall*] resource is created, the IPE shall parse the *input* attribute of the [*allJoynMethodCall*] resource and convert it into the corresponding AllJoyn method call message and send the message to the corresponding AllJoyn Producer Application. The IPE should update the *callStatus* and *output* attributes of the [*allJoynMethodCall*] resource to represent the actual status of the calling attempt (e.g. failed, OK) and the returned output of the method call accordingly.

- When the *requestedValue* attribute of a [*allJoynProperty*] resource is created (i.e. added to the existing resource) , the IPE shall parse the attribute values and convert it into the corresponding AllJoyn property set message and send the message to the corresponding AllJoyn Producer Application. The IPE should delete the *requestedValue* attribute and update the *currentValue* attribute of the [*allJoynProperty*] resource to represent the outcome of the attempted change of property value accordingly.

- When the IPE receives an AllJoyn signal message, the IPE shall parse the message and convert it into a creation of a new instance of a <contentInstance> resource as a child of the signal <container> resource, whereby the *content* attribute of the <contentInstance> resource shall be set to a serialization of the signal's payload.

7.3.3 AllJoyn services produced by oneM2M AE/CSE(s) via the IPE

For an AllJoyn service that is produced by AE/CSE(s) and consumed by AllJoyn Consumer Applications on AllJoyn devices (i.e. *direction* attribute of [*allJoynApp*] resource is "oneM2MToAllJoyn"), the IPE shall perform the following:

- When the AllJoyn Producer Application on the IPE receives AllJoyn method call message, it shall parse the message and convert it into creation of a new instance of an [*allJoynMethodCall*] resource as a child resource of the corresponding [*allJoynMethod*] resource. The IPE shall either subscribe to the created new instance of an [*allJoynMethodCall*] resource so that it gets notified about any creation or changes of the *callStatus* and *output* attributes or retrieve the information with sufficient frequency. The entity which is actually providing the respective service shall either be subscribed to the corresponding [*allJoynMethod*] parent resource or use frequent polling of it - and therefore -gets to know about the attempt to call a method. Upon this notification or retrieval of the resource, the oneM2M entity providing the service shall carry out the requested method and update the *callStatus* and *output* attributes of the [*allJoynMethodCall*] resource to represent the actual status of the calling attempt (e.g. failed, OK) and the returned output of the method call accordingly. When a method calling attempt initiated by the IPE has completed (successfully or not) the IPE shall parse the *callStatus* and *output* attributes - if applicable - and convert that information into the appropriate AllJoyn method call response message. Note that this is not always required and depends on what particular method is called.

- When the AllJoyn Producer Application on the IPE receives an AllJoyn property set message, it shall parse and convert the message by the addition of the *requestedValue* attribute to the respective [*allJoynProperty*] resource. In order to find out what happens with the the respective [*allJoynProperty*] in the next steps, the IPE shall either subscribe to the corresponding [*allJoynProperty*] resource before addition of the *requestedValue* attribute or retrieve the corresponding [*allJoynProperty*] resource periodically. The oneM2M entity providing the service shall carry out the requested change of property value and update the value of *currentValue* attribute of the [*allJoynProperty*] accordingly. Finally the *requestedValue* attribute of the [*allJoynProperty*] shall be removed by the oneM2M entity providing the service. The IPE shall process the result of the property value change request which it will get to know - either by being notified due to the subscription made earlier or by periodic retrieval It is up to the IPE to choose polling or subscription/notification and possibly keep the subscription in place for later use or remove it.

- When a new <contentInstance> resource gets created as a child of a signal <container> resource, the IPE - which shall either be subscribed to the parent <container> resource according to clause 7.2.4.2 or use frequent retrieval - shall parse the *content* attribute of the <contentInstance> resource and convert it into the AllJoyn signal message and send the message to the corresponding AllJoyn Consumer Application(s).

7.3.4 AllJoyn services addition and deletion

After the initial configuration of the IPE, new AllJoyn services can be exposed to oneM2M and existing AllJoyn services interworking can be stopped. Also services produced by oneM2M AE/CSE(s) and exposed to AllJoyn via the AllJoyn IPE can be added and removed.

When a new AllJoyn service that is supposed to be exposed to oneM2M is discovered (see clause 7.3.2), the IPE shall create an [*allJoynApp*] resource as a child of the [*svcObjWrapper*] resource has and set the *direction* attribute set to indicate "allJoynToOneM2M" and create child/decendent resources as defined in clause 6.2 as well as clause 7.2.3.

When the IPE supports exposure of oneM2M-provided services to AllJoyn (i.e. the IPE can also act as an AllJoyn Producer Application on the AllJoyn side) and if an AE/CS initiates a service that shall be exposed by the IPE as an AllJoyn service that would be consumed by AllJoyn Consumer Applications on AllJoyn devices, the service providing oneM2M entity shall create a new instance of an [*allJoynSvcObject*] resource as a child of the [*allJoynApp*] resource that was already created by the IPE at time of initialization as a child of the [*svcObjWrapper*] resource which has the *direction* attribute set to "oneM2MToAllJoyn". The oneM2M entity producing the service for AllJoyn Consumer Applications via the IPE is also responsible to create child/decendent resources as defined in clause 6.2 and clasue 7.2.3. As already defined in clause 7.2.4.2, the IPE shall either be subscribed to the parent [*allJoynApp*] resource or retrieve it with sufficient frequency. Upon notification or retrieval about addition of a new [*allJoynSvcObject*] resource the IPE shall validate if the created child resource and its decendants are in line with the specified structure. When valid, the IPE shall initiate to offer the corresponding service as an AllJoyn Producer Application on the AllJoyn side, in that it shall start to carry out the adverertisement and discovery procedures defined in the AllJoyn system for this service.

When an already exposed AllJoyn service is stopped in the AllJoyn proximal network, then the IPE shall remove the corresponding [*allJoynApp*] resource as well as the child/decendent resources of this [*allJoynApp*] resource. Example usages of this removal are: AllJoyn Consumer Application on the IPE leaves the session established with an AllJoyn Producer Application in an AllJoyn proximal network, AllJoyn Producer Application on the IPE leaves the session established with AllJoyn Consumer Application(s) in an AllJoyn proximal network. Upon removal of the corresponding [*allJoynApp*] resource, oneM2M entites will not be able anymore to consume that service. For already exposed services produced on the oneM2M side, the service producer can remove the respective [*allJoynSvcObject*] resource that is a child of the [*allJoynApp*] resource that was already created by the IPE at time of initialization as a child of the [*svcObjWrapper*] resource which has the *direction* attribute set to "oneM2MToAllJoyn". The removal will either get notified to the IPE due to a previous established subscription or by periodic retrieval (polling). Upon getting information about the removal of a oneM2M-provided service that was exposed to AllJoyn, the IPE shall stop acting as an AllJoyn Producer Application and terminate any existing service sessions for that service on the AllJoyn side.

Annex A (informative): oneM2M and AllJoyn technical comparison

A.1 API styles

Table A.1-1 describes how API styles of oneM2M and AllJoyn system are different to expose its services that can provide some functionalties.

Table A.1-1: The Comparison of API Styles of oneM2M and AllJoyn System

	oneM2M	AllJoyn
API Style	**Resource based API** (RESTful Architecture). It assigns all procedures, instances of data a resource identified by a URI and leverage web-based protocol's operations (CRUD) to define service behaviors.	**D-Bus [i.4] based API.** It defines a protocol for exposing a typical object-oriented language-based service to other applications that consume the service.
Representation of a Service	An oneM2M Service is defined by one or more Resources specified in ETSI TS 118 101 [1]. An oneM2M resource defines a set of representations of data. Each resource has attributes that stores information regarding the resource itself, procedures or data.	An AllJoyn Service is defined by one or more AllJoyn Interfaces. An AllJoyn interface can include one or more of following types of members: • **Methods:** a function call that typically takes a set of inputs, performs and returns. • **Properties:** a variable that holds values. • **Signals:** an asynchronous notification that is generated by a service to notify remote peers of an event or change.

Tables A.1-2 and A.1-3 give examples how an oneM2M and AllJoyn service is described respectively. Both examples demonstrate a schema or introspection that describes location service to acquire the location information of a target information.

Table A.1-2: Resource Schema of <locationPolicy> (v2.4.0)

```
<xs:schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.onem2m.org/xml/protocols"
    xmlns:m2m="http://www.onem2m.org/xml/protocols"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    elementFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:include schemaLocation="CDT-commonTypes-v2_4_0.xsd" />
    <xs:include schemaLocation="CDT-subscription-v2_4_0.xsd" />

    <xs:element name="locationPolicy">
        <xs:complexType>
            <xs:complexContent>
                <!-- Inherit common attributes for announceable Resources -->
                <xs:extension base="m2m:announceableResource">
                    <xs:sequence>
                        <!-- Resource Specific Attributes -->
                        <xs:element name="locationSource"
type="m2m:locationSource" />
                        <xs:element name="locationUpdatePeriod"
type="m2m:listOfDuration" minOccurs="0" />
                        <xs:element name="locationTargetID" type="m2m:nodeID"
minOccurs="0" />
                        <xs:element name="locationServer" type="xs:anyURI"
minOccurs="0" />
                        <xs:element name="locationContainerID" type="xs:anyURI"
minOccurs="0" />
                        <xs:element name="locationContainerName" type="xs:string"
minOccurs="0" />
                        <xs:element name="locationStatus" type="xs:string" />

                        <!-- Child Resources -->
                        <xs:choice minOccurs="0" maxOccurs="1">
                            <xs:element name="childResource"
type="m2m:childResourceRef" minOccurs="1" maxOccurs="unbounded" />
                            <xs:choice minOccurs="1" maxOccurs="unbounded">
                                <xs:element ref="m2m:subscription"></xs:element>
                            </xs:choice>
                        </xs:choice>
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>

    <xs:element name="locationPolicyAnnc">
        <xs:complexType>
            <xs:complexContent>
                <!-- Inherit common attributes for announced Resources -->
                <xs:extension base="m2m:announcedResource">
                    <xs:sequence>
                        <!-- Resource Specific Attributes -->
                        <xs:element name="locationSource"
type="m2m:locationSource" minOccurs="0" />
                        <xs:element name="locationUpdatePeriod"
type="m2m:listOfDuration" minOccurs="0" />
                        <xs:element name="locationTargetID" type="m2m:nodeID"
minOccurs="0" />
                        <xs:element name="locationServer" type="xs:anyURI"
minOccurs="0" />
                        <xs:element name="locationContainerID" type="xs:anyURI"
minOccurs="0" />
```

```
                    <xs:element name="locationContainerName" type="xs:string"
minOccurs="0" />
                    <xs:element name="locationStatus" type="xs:string"
minOccurs="0" />

                    <!-- No Child Resources -->
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
</xs:element>
</xs:schema><xs:schema xmlns="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.onem2m.org/xml/protocols"
   xmlns:m2m="http://www.onem2m.org/xml/protocols"
    elementFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:include schemaLocation="CDT-commonTypes-v0_7_1.xsd" />

    <xs:element name="locationPolicy">
        <xs:complexType>
            <xs:complexContent>
              <!--  Inherit Announce able Attributes from
announceableResourceType     -->
                <xs:extension base="m2m:announceableResourceType">
                    <xs:sequence>
                        <!--          Resource Specific Attributes           -->
                    <xs:element name="locationSource"
type="m2m:locationSource" />
                    <xs:element name="locationUpdatePeriod"
type="xs:duration" minOccurs="0" />
                    <xs:element name="locationTargetID" type="m2m:nodeID"
minOccurs="0" />
                    <xs:element name="locationServer" type="xs:anyURI"
minOccurs="0" />
                    <xs:element name="locationContainerID" type="xs:anyURI"
minOccurs="0" />
                    <xs:element name="locationContainerName" type="xs:string"
minOccurs="0" />
                    <xs:element name="locationStatus" type="xs:string"
minOccurs="0" />

                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
</xs:element>
</xs:schema>
```

The resource schema described in the table above can be also graphically represented as figure A.1-1.

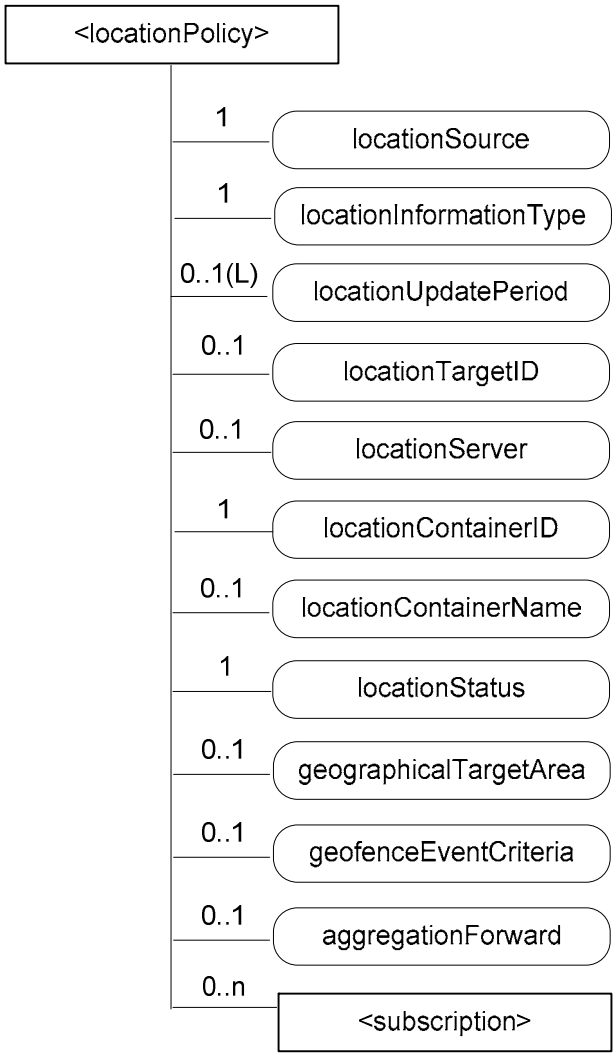

Figure A.1-1: Structure of <locationPolicy> resource

Table A.1-3: Introspection XML of an AllJoyn Location Service (example)

```
<node name="/com/alljoyn/LocationService">
   <interface name="com.alljoyn.LocationService">
      <method name="getLocation">
         <arg name="locationSource" type="s" direction="in"> </arg>
         <arg name="targetID" type="s" direction="in"> </arg>
      </method>

      <signal name="locationPoint" sessionless="true> </signal>

      <property name="locationStatus" type="y" access="read"> </property>

   </interface>
</node>
```

A.2 Service Discovery/Advertisement

One of the most significant features of service layer protocol for the IoT is to advertise or discover services to a remote entity or application. Both systems have specified significantly different advertisement and discovery mechanism.

Table A.2-1: The Comparison of API Styles of oneM2M and AllJoyn System

	oneM2M	AllJoyn
Service Consumer	Since oneM2M is a web-based architecture, the service consumer firstly discover a target service instantiated in a CSE using GET method. The discovery message is sent over unicast transport protocol. If the target service is found, the service consumer consumes the service subject to access control. This behavior is just like resource searching by using web-browser.	A service consumer in AllJoyn network send a DNS-SD query message over Multicast DNS (mDNS) network to discover a target service implemented in other applications/devices. Since a site-local IP multicast mechanism is used for mDNS, the DNS-SD query can be sent toward mDNS multicast members within proximal network. In order to enable secure access a service resides in the remote network, AllSeen Alliance has been specifying a Gateway Agent feature [i.4] (see note)
Service Provider	The notification for service advertisement which is sent toward unspecified applications causes network collapse due to the broadcast message flooding. Thus, oneM2M does not specify the proactive advertisement method acting by service provider. However, oneM2M has specified the service information forwarding (toward target node) method, "Announcement".	If the services specified in the received query is provided by the service provider, the service provider responds a DNS-SD response message over unicast to the service consumer (see note).
NOTE: This procedure supports after 14.06 version of AllJoyn system.		

Annex B (normative): <flexContainer> resource specializations

B.1 Introduction

In this annex, all AllJoyn specific resource types are defined. All resource types specific to AllJoyn interworking are specializations of the <*flexContainer*>.

B.2 Resource Type *svcObjWrapper*

This specialization of <*flexContainer*> is intended to be used to wrap a group of child resources related to AllJoyn service objects with minimal overhead. No custom attributes are needed for this specialization.

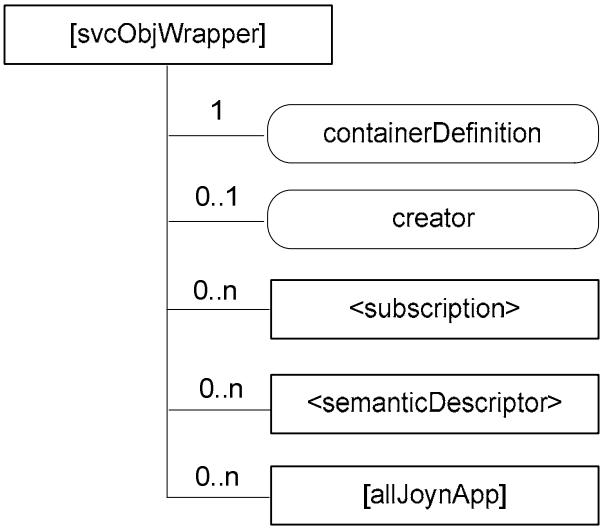

Figure B.2-1: Structure of [*svcObjWrapper*] resource

The [*svcObjWrapper*] resource shall contain the child resources specified in table B.2-1.

Table B.2-1: Child resources of [*svcObjWrapper*] resource

Child Resources of [*svcObjWrapper*]	Child Resource Type	Multiplicity	Description	[*svcObjWrapperAnnc*] Child Resource Type
[variable]	*<subscription>*	0..n	See clause 9.6.8 in ETSI TS 118 101 [1]	*<subscription>*
[variable]	*<semanticDescriptor>*	0..n	See clause 9.6.30 in ETSI TS 118 101 [1]	*<semanticDescriptor>, <semanticDescriptorAnnc>*
[variable]	*[allJoynApp]*	0..n	See clause B.4	*[allJoynAppAnnc]*

The [*svcObjWrapper*] resource shall contain the attributes specified in table B.2-2.

Table B.2-2: Attributes of [*svcObjWrapper*] resource

Attributes of [*svcObjWrapper*]	Multiplicity	RW/ RO/ WO	Description	[*svcObjWrapper*] Attributes
resourceType	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
resourceID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
resourceName	1	WO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
parentID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
expirationTime	1	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
labels	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
creationTime	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
lastModifiedTime	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
stateTag	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	OA
announceTo	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
announcedAttribute	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
dynamicAuthorizationConsultationIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	OA
containerDefinition	1	WO	This contains a reference (URN) to the definition which shall be used by the CSE to statically validate the syntax of the specialization of the [wrapper] resource (see note). This URN shall refer to AllJoyn Interworking, wrapper	MA
creator	0..1	RO	The AE-ID or CSE-ID of the entity which created the resource.	NA
ontologyRef	0..1	RW	A reference (URI) of the ontology used to represent the information that is stored in the present [svcObjWrapper] resource	OA
NOTE: The reference is a URN and not URL: the CSE does not retrieve the definition of the [*wrapper*] resource dynamically.				

B.3 Resource Type *svcFwWrapper*

This specialization of <*flexContainer*> is intended to be used to wrap a group of child resources related to AllJoyn framework services with minimal overhead. No custom attributes are needed for this specialization.

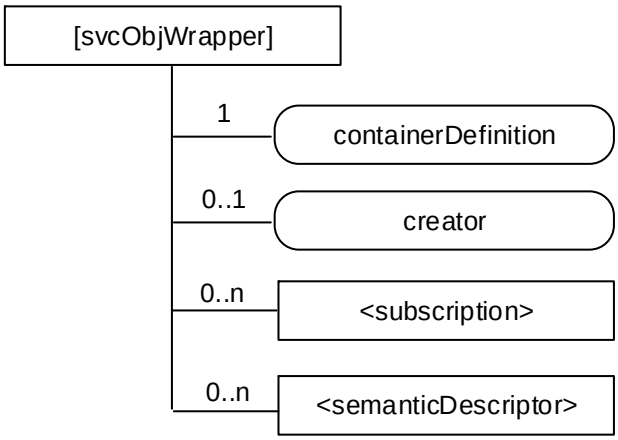

Figure B.3-1: Structure of [*svcFwWrapper*] resource

The [*svcFwWrapper*] resource shall contain the child resources specified in table B.3-1.

Table B.3-1: Child resources of [*svcFwWrapper*] resource

Child Resources of [*svcFwWrapper*]	Child Resource Type	Multiplicity	Description	[*svcFwWrapperAnnc*] Child Resource Type
[variable]	*<subscription>*	0..n	See clause 9.6.8 in ETSI TS 118 101 [1]	*<subscription>*
[variable]	*<semanticDescriptor>*	0..n	See clause 9.6.30 in ETSI TS 118 101 [1]	*<semanticDescriptor>, <semanticDescriptorAnnc>*

The [*svcFwWrapper*] resource shall contain the attributes specified in table B.3-2.

Table B.3-2: Attributes of [*svcFwWrapper*] resource

Attributes of [*svcFwWrapper*]	Multiplicity	RW/ RO/ WO	Description	[*svcFwWrapper*] Attributes
resourceType	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
resourceID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
resourceName	1	WO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
parentID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
expirationTime	1	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
labels	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
creationTime	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
lastModifiedTime	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
stateTag	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	OA
announceTo	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
announcedAttribute	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
dynamicAuthorizationConsultationIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	OA
containerDefinition	1	WO	This contains a reference (URN) to the definition which shall be used by the CSE to statically validate the syntax of the specialization of the [*wrapper*] resource (see note) This URN shall refer to: • AllJoyn Interworking, wrapper	MA
creator	0..1	RO	The AE-ID or CSE-ID of the entity which created the resource	NA
ontologyRef	0..1	RW	A reference (URI) of the ontology used to represent the information that is stored in the present [*svcObjWrapper*] resource	OA
NOTE: The reference is a URN and not URL: the CSE does not retrieve the definition of the [*wrapper*] resource dynamically.				

B.4 Resource Type *allJoynApp*

This specialization of *<flexContainer>* is used to represent a specific instance of an AllJoyn application. The name of an instance of this resource type shall be set to the Well-Known-Name of the represented AllJoyn application instance at the local AllJoyn bus for which interworking is enabled through this resource and its children.

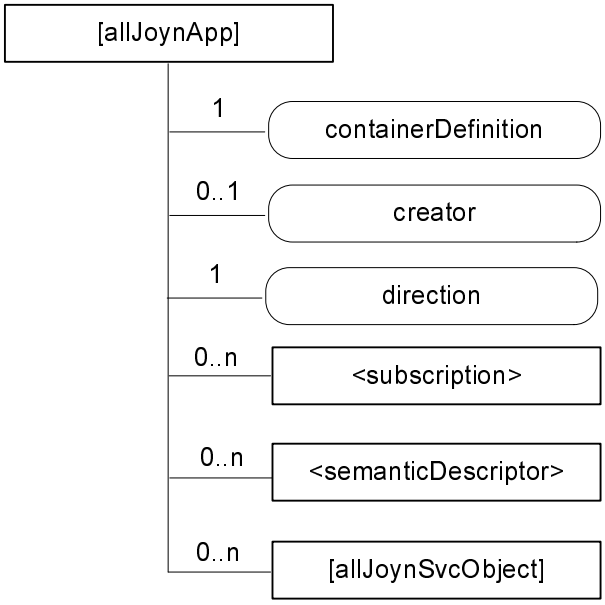

Figure B.4-1: Structure of [*allJoynApp*] resource

The [*allJoynApp*] resource shall contain the child resources specified in table B.4-1.

Table B.4-1: Child resources of [*allJoynApp*] resource

Child Resources of [*allJoynApp*]	Child Resource Type	Multiplicity	Description	[*allJoynAppAnnc*] Child Resource Type
[variable]	*<subscription>*	0..n	See clause 9.6.8 in ETSI TS 118 101 [1]	*<subscription>*
[variable]	*<semanticDescriptor>*	0..n	See clause 9.6.30 in ETSI TS 118 101 [1]	*<semanticDescriptor>*, *<semanticDescriptorAnnc>*
[variable]	*[allJoynSvcObject]*	0..n	See clause B.5 in the present document	[*allJoynSvcObject*] [*allJoynSvcObjectAnnc*]

The [*allJoynApp*] resource shall contain the attributes specified in table B.4-2.

Table B.4-2: Attributes of [*allJoynApp*] resource

Attributes of [*allJoynApp*]	Multiplicity	RW/ RO/ WO	Description	[*allJoynAppainerAnnc*] Attributes
resourceType	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
resourceID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
resourceName	1	WO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
parentID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
expirationTime	1	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
labels	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
creationTime	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
lastModifiedTime	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
stateTag	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	OA
announceTo	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
announcedAttribute	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
dynamicAuthorizationConsultationIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	OA
containerDefinition	1	WO	This contains a reference (URN) to the definition which shall be used by the CSE to statically validate the syntax of the specialization of the [*wrapper*] resource (see note) This URN shall refer to: • AllJoyn Interworking, allJoynApp	MA
creator	0..1	RO	The AE-ID or CSE-ID of the entity which created the resource.	NA
direction	1	WO	This attribute indicates whether the service is exposed from the AllJoyn side to the oneM2M side (value corresponding to "allJoynToOneM2m") or vice versa (value corresponding to "oneM2mToAllJoyn")	MA
ontologyRef	0..1	RW	A reference (URI) of the ontology used to represent the information that is stored in the present [*svcObjWrapper*] resource	OA
NOTE: The reference is a URN and not URL: the CSE does not retrieve the definition of the [*wrapper*] dynamically.				

B.5 Resource Type *allJoynSvcObject*

This specialization of <*flexContainer*> is used to represent a specific instance of an AllJoyn service object.

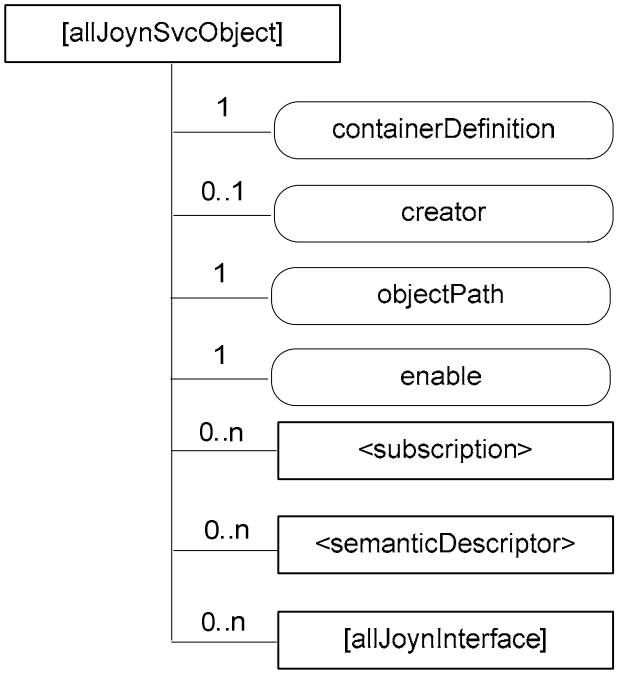

Figure B.5-1: Structure of [*allJoynSvcObject*] resource

The [*allJoynSvcObject*] resource shall contain the child resources specified in table B.5-1.

Table B.5-1: Child resources of [*allJoynSvcObject*] resource

Child Resources of [*allJoynSvcObject*]	Child Resource Type	Multiplicity	Description	[*allJoynSvcObjectAnnc*] Child Resource Type
[variable]	*<subscription>*	0..n	See clause 9.6.8 in ETSI TS 118 101 [1]	*<subscription>*
[variable]	*<semanticDescriptor>*	0..n	See clause 9.6.30 in ETSI TS 118 101 [1]	*<semanticDescriptor>, <semanticDescriptorAnnc>*
[variable]	*[allJoynInterface]*	0..n	See clause B.6 of the present document	*[allJoynInterface] [allJoynInterfaceAnnc]*

The [*allJoynSvcObject*] resource shall contain the attributes specified in table B.5-2.

Table B.5-2: Attributes of [*allJoynSvcObject*] resource

Attributes of [*allJoynSvcObject*]	Multiplicity	RW/ RO/ WO	Description	[*allJoynSvcObjectAnnc*] Attributes
resourceType	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
resourceID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
resourceName	1	WO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
parentID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
expirationTime	1	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
labels	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
creationTime	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
lastModifiedTime	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
stateTag	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	OA
announceTo	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
announcedAttribute	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
dynamicAuthorizationConsultationIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	OA
containerDefinition	1	WO	This contains a reference (URN) to the definition which shall be used by the CSE to statically validate the syntax of the specialization of the [*wrapper*] resource (see note) This URN shall refer to: • AllJoyn Interworking, allJoynSvcObject	MA
creator	0..1	RO	The AE-ID or CSE-ID of the entity which created the resource.	NA
objectPath	1	WO	This attribute contains a string equal to the AllJoyn object path for the represented AllJoyn service object	MA
enable	1	RW	This attribute contains a flag indicating whether the resource respresening the corresponding AllJoyn service object - and all its child resources - can currently be used for service exposure. When this attribute is set to False, no service exposure interactions are supported for the AllJoyn service object represented by this [allJoynSvcObject] resource. If this attribute is set to True, service exposure interactions are supported for the AllJoyn service object represented by this [allJoynSvcObject] resource.	OA
ontologyRef	0..1	RW	A reference (URI) of the ontology used to represent the information that is stored in the present [*svcObjWrapper*] resource	OA
NOTE: The reference is a URN and not URL: the CSE does not retrieve the definition of the [*wrapper*] dynamically.				

B.6 Resource Type *allJoynInterface*

This specialization of <*flexContainer*> is used to represent a specific implementation of an AllJoyn interface residing in an AllJoyn service object.

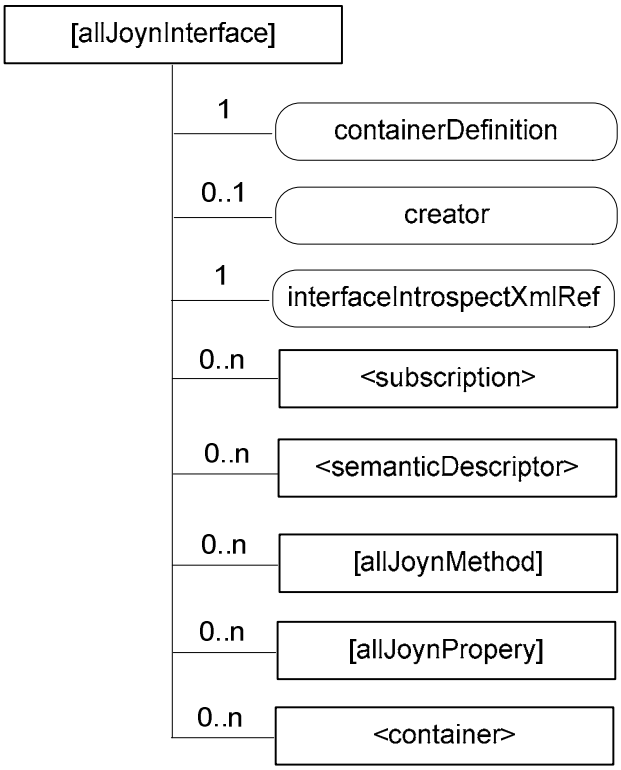

Figure B.6-1: Structure of [*allJoynInterface*] resource

The [*allJoynInterface*] resource shall contain the child resources specified in table B.6-1.

Table B.6-1: Child resources of [*allJoynInterface*] resource

Child Resources of [*allJoynInterface*]	Child Resource Type	Multiplicity	Description	[allJoynInterface *Annc*] Child Resource Type
[variable]	*<subscription>*	0..n	See clause 9.6.8 in ETSI TS 118 101 [1]	*<subscription>*
[variable]	*<semanticDescriptor>*	0..n	See clause 9.6.30 in ETSI TS 118 101 [1]	*<semanticDescriptor>, <semanticDescriptorAnnc>*
[variable]	*[allJoynMethod]*	0..n	See clause B.7 in the present document	*[allJoynMethod] [allJoynMethodAnnc]*
[variable]	*[allJoynProperty]*	0..n	See clause B.9 in the present document	*[allJoynProperty] [allJoynPropertyAnnc]*
[variable]	*<container>*	0..n	See clause 9.6.6 in ETSI TS 118 101 [1]	*<container> <containerAnnc>*

The [*allJoynInterface*] resource shall contain the attributes specified in table B.6-2.

Table B.6-2: Attributes of [*allJoynInterface*] resource

Attributes of [*allJoynInterface*]	Multiplicity	RW/ RO/ WO	Description	[*allJoynInterface Annc*] Attributes
resourceType	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
resourceID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
resourceName	1	WO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
parentID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
labels	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
stateTag	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	OA
announceTo	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
announcedAttribute	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
dynamicAuthorizationConsultationIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	OA
containerDefinition	1	WO	This contains a reference (URN) to the definition which shall be used by the CSE to statically validate the syntax of the specialization of the [*wrapper*] resource (see note) This URN shall refer to: • AllJoyn Interworking, allJoynInterface	MA
creator	0..1	RO	The AE-ID or CSE-ID of the entity which created the resource.	NA
interfaceIntrospectXmlRef	1	WO	This attribute contains a reference to the AllJoyn interface definition in XML. The reference can either be a reference to another resource containing the XML (e.g. a <contentInstance> resource) or a URI pointing to a publicly accessible location containing the XML definition.	OA
ontologyRef	0..1	RW	A reference (URI) of the ontology used to represent the information that is stored in the present [*svcObjWrapper*] resource	OA
NOTE: The reference is a URN and not URL: the CSE does not retrieve the definition of the [*wrapper*] dynamically.				

B.7 Resource Type *allJoynMethod*

This specialization of <*flexContainer*> is used to represent a specific method of an AllJoyn interface residing in an AllJoyn service object.The name of an instance of this resource type shall be set to the name of the represented method in the AllJoyn interface in which this method resides.

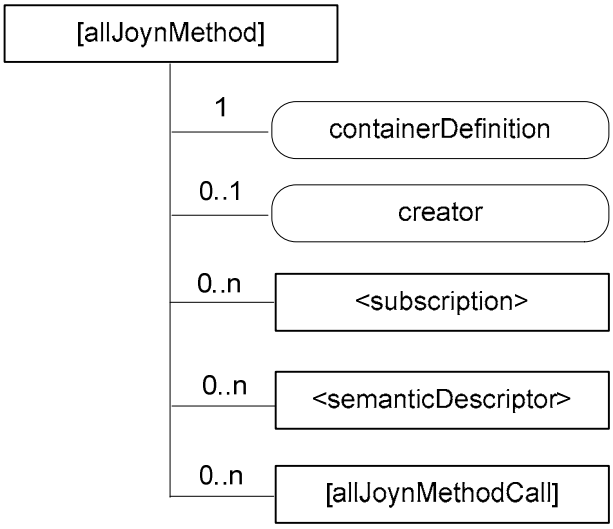

Figure B.7-1: Structure of [*allJoynMethod*] resource

The [*allJoynMethod*] resource shall contain the child resources specified in table B.7-1.

Table B.7-1: Child resources of [*allJoynMethod*] resource

Child Resources of [*allJoynMethod*]	Child Resource Type	Multiplicity	Description	[allJoynMethod*Annc*] Child Resource Type
[variable]	*<subscription>*	0..n	See clause 9.6.8 in ETSI TS 118 101 [1]	*<subscription>*
[variable]	*<semanticDescriptor>*	0..n	See clause 9.6.30 in ETSI TS 118 101 [1]	*<semanticDescriptor>, <semanticDescriptorAnnc>*
[variable]	*[allJoynMethodCall]*	0..n	See clause B.8 in the present document	*[allJoynMethodCall] [allJoynMethodCallAnnc]*

The [*allJoynMethod*] resource shall contain the attributes specified in table B.7-2.

Table B.7-2: Attributes of [*allJoynMethod*] resource

Attributes of [*allJoynMethod*]	Multiplicity	RW/ RO/ WO	Description	[allJoynMethodA*nnc*] Attributes
resourceType	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
resourceID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
resourceName	1	WO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
parentID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
labels	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
stateTag	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	OA
announceTo	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
announcedAttribute	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
dynamicAuthorizationConsultationIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	OA
containerDefinition	1	WO	This contains a reference (URN) to the definition which shall be used by the CSE to statically validate the syntax of the specialization of the [*wrapper*] resource (see note) This URN shall refer to: • AllJoyn Interworking, allJoynMethod	MA
creator	0..1	RO	The AE-ID or CSE-ID of the entity which created the resource.	NA
ontologyRef	0..1	RW	A reference (URI) of the ontology used to represent the information that is stored in the present [*svcObjWrapper*] resource	OA
NOTE: The reference is a URN and not URL: the CSE does not retrieve the definition of the [*wrapper*] dynamically.				

B.8 Resource Type *allJoynMethodCall*

This specialization of <*flexContainer*> is used to represent a specific calling instance of a method of an AllJoyn interface residing in an AllJoyn service object. For each call of the AllJoyn method represented by the parent [*allJoynMethodCall*] resource, the consumer of the service needs to create a new instance of a [allJoynMethodCall] resource. Upon successful creation of a [*allJoynMethodCall*], the corresponding AllJoyn method will be executed.

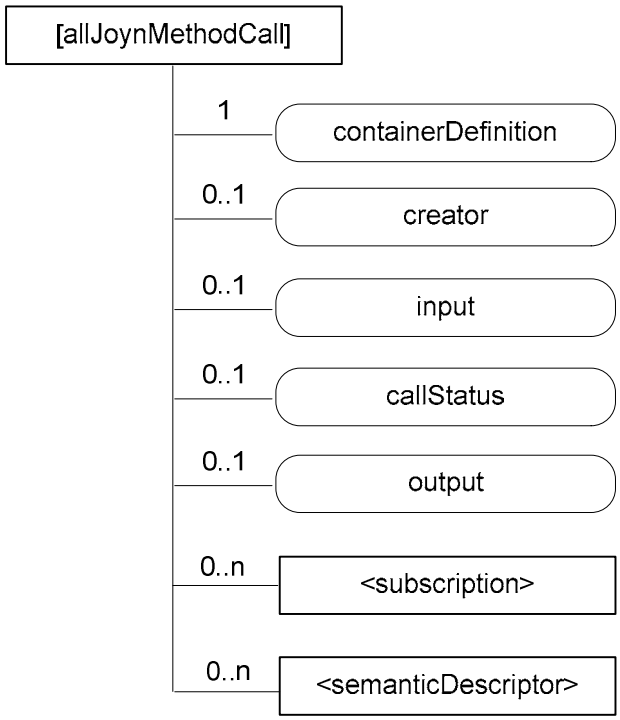

Figure B.8-1: Structure of [*allJoynMethodCall*] resource

The [*allJoynMethodCall*] resource shall contain the child resources specified in table B.8-1.

Table B.8-1: Child resources of [*allJoynMethodCall*] resource

Child Resources of [*allJoynMethodCall*]	Child Resource Type	Multiplicity	Description	[allJoynMethodCall*Annc*] Child Resource Type
[variable]	*<subscription>*	0..n	See clause 9.6.8 in ETSI TS 118 101 [1]	*<subscription>*
[variable]	*<semanticDescriptor>*	0..n	See clause 9.6.30 in ETSI TS 118 101 [**1**]	*<semanticDescriptor>, <semanticDescriptorAnnc>*

The [*allJoynMethodCall*] resource shall contain the attributes specified in table B.8-2.

Table B.8-2: Attributes of [*allJoynMethodCall*] resource

Attributes of [*allJoynMethodCall*]	Multiplicity	RW/ RO/ WO	Description	[allJoynMethodCall*Annc*] Attributes
resourceType	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
resourceID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
resourceName	1	WO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
parentID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
expirationTime	1	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
labels	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
creationTime	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
lastModifiedTime	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
stateTag	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	OA
announceTo	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
announcedAttribute	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
dynamicAuthorizationConsultationIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	OA

Attributes of [*allJoynMethodCall*]	Multiplicity	RW/ RO/ WO	Description	[allJoynMethodCall*Annc*] Attributes
containerDefinition	1	WO	This contains a reference (URN) to the definition which shall be used by the CSE to statically validate the syntax of the specialization of the [*wrapper*] resource (see note). This URN shall refer to: • AllJoyn Interworking, allJoynMethodCall	MA
creator	0..1	RO	The AE-ID or CSE-ID of the entity which created the resource.	NA
input	0..1	WO	In cases where the AllJoyn method represented by the parent [allJoynMethod] resource requires input parameters, the *input* attribute needs to be present and contain a valid serialization of input parameters to the method call.	NA
callStatus	0..1	RW	The entity which acts on creation of a new [allJoynMethodCall] resource - i.e. the IPE in case of a service exposed from AllJoyn to oneM2M or another AE in case of exposing a service from oneM2M to AllJoyn - needs to perform the requested method call and populate the *callStatus* attribute with corresponding status information.	NA
output	0..1	RW	Upon completion of an AllJoyn method call which produces output parameters, the *ouput* attribute shall get populated with a valid serialization of the corresponding output of the method call - if any. Note that some AllJoyn methods do not return any output parameters upon completion of a call.	NA
ontologyRef	0..1	RW	A reference (URI) of the ontology used to represent the information that is stored in the present [*svcObjWrapper*] resource	OA
NOTE: The reference is a URN and not URL: the CSE does not retrieve the definition of the [*wrapper*] dynamically.				

B.9 Resource Type *allJoynProperty*

This specialization of <*flexContainer*> is used to represent a specific property of an AllJoyn interface residing in an AllJoyn service object. The name of an instance of this resource type shall be set to the name of the represented property in the AllJoyn interface in which the represented property resides.

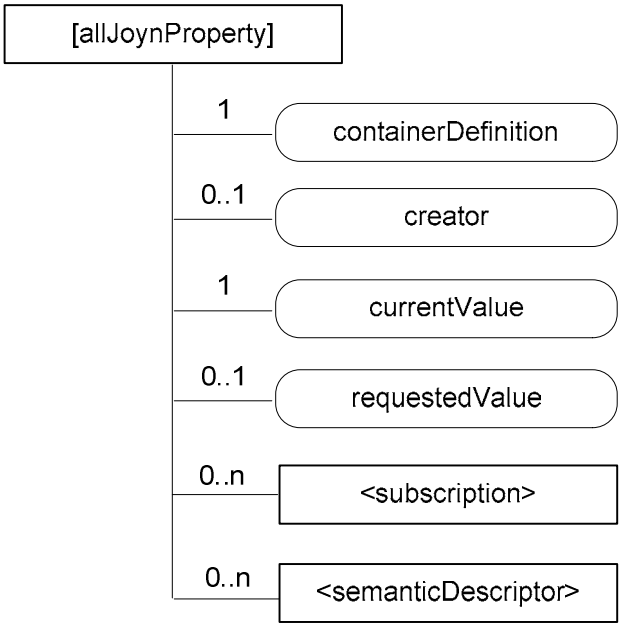

Figure B.9-1: Structure of [*allJoynProperty*] resource

The [*allJoynProperty*] resource shall contain the child resources specified in table B.9-1.

Table B.9-1: Child resources of [*allJoynProperty*] resource

Child Resources of [*allJoynProperty*]	Child Resource Type	Multiplicity	Description	[allJoynProperty*Annc*] Child Resource Type
[variable]	*<subscription>*	0..n	See clause 9.6.8 in ETSI TS 118 101 [1]	*<subscription>*
[variable]	*<semanticDescriptor>*	0..n	See clause 9.6.30 in ETSI TS 118 101 [1]	*<semanticDescriptor>*, *<semanticDescriptorAnnc>*

The [*allJoynProperty*] resource shall contain the attributes specified in table B.9-2.

Table B.9-2: Attributes of [*allJoynProperty*] resource

Attributes of [*allJoynProperty*]	Multiplicity	RW/ RO/ WO	Description	[allJoynPropertyAnnc] Attributes
resourceType	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
resourceID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
resourceName	1	WO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
parentID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
labels	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	MA
lastModifiedTime	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
stateTag	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [1]	OA
announceTo	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
announcedAttribute	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	NA
dynamicAuthorizationConsultationIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [1]	OA
containerDefinition	1	WO	This contains a reference (URN) to the definition which shall be used by the CSE to statically validate the syntax of the specialization of the [*wrapper*] resource (see note) This URN shall refer to: • AllJoyn Interworking, allJoynMethodCall	MA
creator	0..1	RO	The AE-ID or CSE-ID of the entity which created the resource.	NA
currentValue	1	RW	The attribute currentValue is representing the current value of the property	NA
requestedValue	0..1	RW	When the AllJoyn interface property represented by this resource is capable of being changed, the attribute *requestedValue* shall be created - if not already present - or updated - if already present - to request a change of the current value of the property to a new value. Upon completion of such a change request, the IPE or the service providing AE will update the new value of the property in the attribute *currentValue*.	NA
ontologyRef	0..1	RW	A reference (URI) of the ontology used to represent the information that is stored in the present [*svcObjWrapper*] resource	OA
NOTE: The reference is a URN and not URL: the CSE does not retrieve the definition of the [*wrapper*] dynamically.				

Annex C (normative): AllJoyn interface mapping to oneM2M resource structure

C.1 Notification service

C.1.1 Notification interface

AllJoyn Notification(*org.alljoyn.Notification*) interface and its members shall be mapped with the following structure.

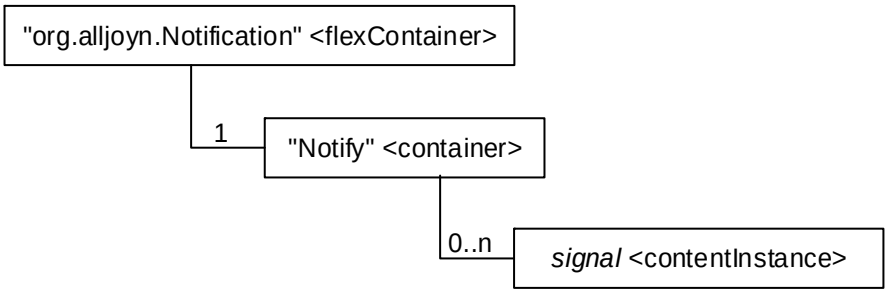

Figure C.1.1-1: Structure of Notification interface mapping

Each resource in the figure above shall have the attribute values as follows.

Table C.1.1-1: Attribute values of org.alljoyn.Notification resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"org.alljoyn. Notification"	AllJoyn interface name
containerDefinition	"alljoynInterface"	AllJoyn interface
version	"1"	Version property value is "1".

NOTE: The values for *containerDefinition* attribute is defined in ETSI TS 118 104 [2]. This applies in the other tables in this Annex C.

Table C.1.1-2: Attribute values of Notify resource

Attribute	Value	Description
resourceType	"3"	Enumeration value for the *container* resource type
resourceName	"Notify"	"Notify" signal
labels	"Iwked-Technology:AllJoyn", "Iwked-Entity-Type:Signal Name"	

Table C.1.1-3: Attribute values of signal resource

Attribute	Value	Description
resourceType	"4"	Enumeration value for the *contentInstance* resource type
resourceName	(variable)	
content	(variable)	Signal message

C.1.2 Notification Producer interface

AllJoyn Notification Producer(*org.alljoyn.Notification.Producer*) interface and its members shall be mapped with the following structure.

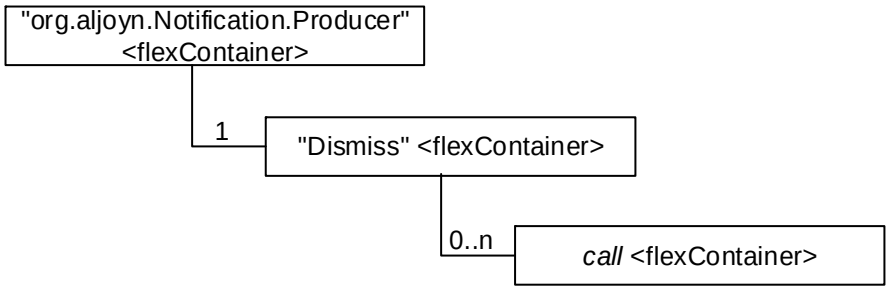

Figure C.1.2-1: Structure of Notification Producer interface mapping

Each resource in the figure above shall have the attribute values as follows.

Table C.1.2-1: Attribute values of org.alljoyn.Notification.Producer resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"org.alljoyn.Notification.Producer"	AllJoyn interface name
containerDefinition	Value indicating alljoynInterface	AllJoyn interface
version	"1"	Version property value is "1".

Table C.1.2-2: Attribute values of Dismiss resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"Dismiss"	"Dismiss" method
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.1.2-3: Attribute values of call resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
input	(variable)	key-value pair of "msgId"
callStatus	(variable)	

C.1.3 Dismisser interface

AllJoyn Dismisser(*org.alljoyn.Notification.Dismisser*) interface and its members shall be mapped with the following structure.

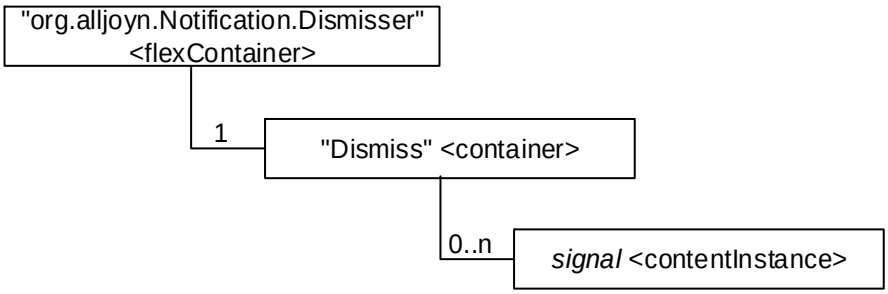

Figure C.1.3-1: Structure of Dismisser interface mapping

Each resource in the figure above shall have the attribute values as follows.

Table C.1.3-1: Attribute values of org.alljoyn.Notification.Dismisser resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"org.alljoyn.Notification.Dismisser"	AllJoyn interface name
containerDefinition	Value indicating alljoynInterface	AllJoyn interface
version	"1"	Version property value is "1"

Table C.1.3-2: Attribute values of Dismiss resource

Attribute	Value	Description
resourceType	"3"	Enumeration value for the *container* resource type
resourceName	"Dismiss"	"Dismiss" signal
labels	"Iwked-Technology:AllJoyn", "Iwked-Entity-Type:Signal Name"	

Table C.1.3-3: Attribute values of signal resource

Attribute	Value	Description
resourceType	"4"	Enumeration value for the *contentInstance* resource type
resourceName	(variable)	
content	(variable)	Signal message

C.2 Control Panel service

C.2.1 Container interface

AllJoyn Container(*org.alljoyn.ControlPanel.Container*) interface and its members shall be mapped with the following structure.

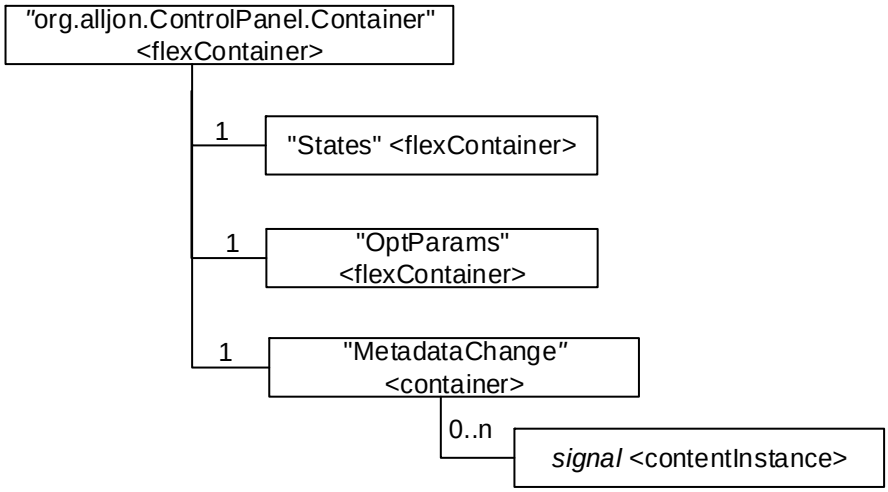

Figure C.2.1-1: Structure of Container interface mapping

Each resource in the figure above shall have the attribute values as follows.

Table C.2.1-1: Attribute values of org.alljoyn.ControlPanel.Container resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"org.alljoyn.ControlPanel.Container"	AllJoyn interface name
containerDefinition	Value indicating alljoynInterface	AllJoyn interface
version	"1"	Version property value is "1"

Table C.2.1-2: Attribute values of States resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"OptParams"	"States" property
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
currentValue	(variable)	Read-only state "enabled"

Table C.2.1-3: Attribute values of OptParams resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"OptParams"	"OptParams" property
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
currentValue	(variable)	Read-only "Container" widget metadata

Table C.2.1-4: Attribute values of MetadataChange resource

Attribute	Value	Description
resourceType	"3"	Enumeration value for the *container* resource type
resourceName	"MetadataChange"	"MetadataChange" signal
labels	"Iwked-Technology:AllJoyn", "Iwked-Entity-Type:Signal Name"	

Table C.2.1-5: Attribute values of signal resource

Attribute	Value	Description
resourceType	"4"	Enumeration value for the *contentInstance* resource type
resourceName	(variable)	
content	"changed"	Notificiation that the metadata is changed

C.2.2 Property interface

AllJoyn Property(*org.alljoyn.ControlPanel.Property*) interface and its members shall be mapped with the following structure.

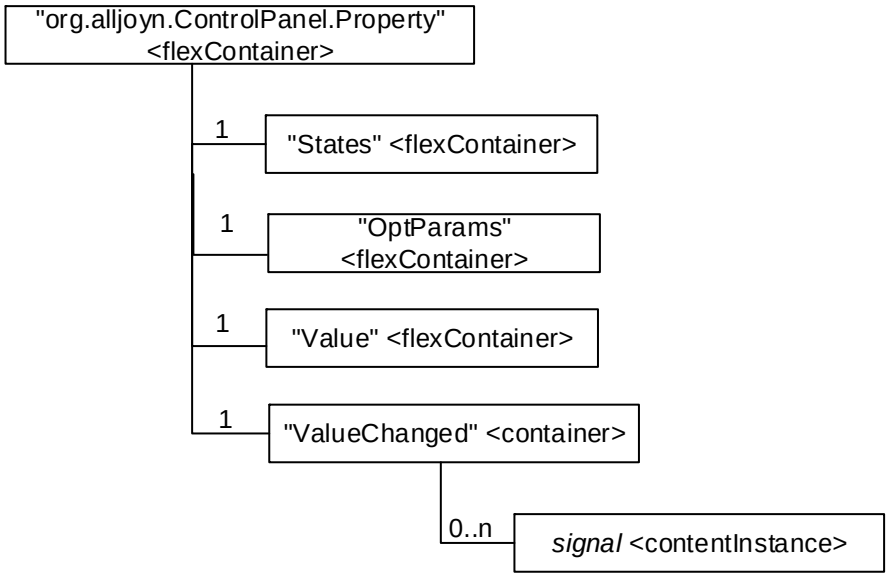

Figure C.2.2-1: Structure of Property interface mapping

Each resource in the figure above shall have the attribute values as follows.

Table C.2.2-1: Attribute values of org.alljoyn.ControlPanel.Property resource

Attribute	Value	Description
resourceType	"3"	Enumeration value for the *container* resource type
resourceName	"org.alljoyn.ControlPanel.Container"	AllJoyn interface name
containerDefinition	Value indicating alljoynInterface	AllJoyn interface
version	"1"	Version property value is "1"

Table C.2.2-2: Attribute values of States resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"States"	"States" property

Attribute	Value	Description
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
currentValue	(variable)	Read-only state "enabled" and "writable"

Table C.2.2-3: Attribute values of OptParams resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"OptParams"	"OptParams" property
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
currentValue	(variable)	Read-only "OptParams" widget metadata

Table C.2.2-4: Attribute values of Value resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"Value"	"Value" property
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
requestedValue	(variable)	Suggested Property value
currentValue	(variable)	Current Property value

Table C.2.2-5: Attribute values of *ValueChanged* resource

Attribute	Value	Description
resourceType	"3"	Enumeration value for the *container* resource type
resourceName	"ValueChanged"	"ValueChanged" signal
labels	"Iwked-Technology:AllJoyn", "Iwked-Entity-Type:Signal Name"	

Table C.2.2-6: Attribute values of signal resource

Attribute	Value	Description
resourceType	"4"	Enumeration value for the *contentInstance* resource type
resourceName	(variable)	
content	(variable)	Key-value pairs of property or metadata

C.2.3 LabelProperty interface

AllJoyn LabelProperty(*org.alljoyn.ControlPanel.LabelProperty*) interface and its members shall be mapped with the following structure.

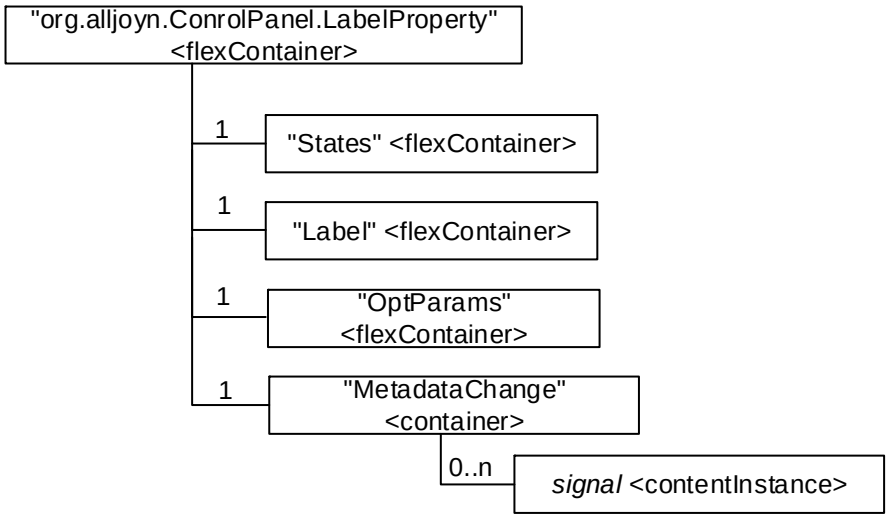

Figure C.2.3-1: Structure of *LabelProperty* interface mapping

Each resource in the figure above shall have the attribute values as follows.

Table C.2.3-1: Attribute values of org.alljoyn.ControlPanel.LabelProperty resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"org.alljoyn.ControlPanel.LabelProperty"	AllJoyn interface name
containerDefinition	Value indicating alljoynInterface	AllJoyn interface
version	"1"	Version property value is "1"

Table C.2.3-2: Attribute values of States resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"States"	"States" property
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
currentValue	(variable)	Read-only state "enabled"

Table C.2.3-3: Attribute values of Label resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"Label"	"Label" property
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
currentValue	(variable)	Read-only label

Table C.2.3-4: Attribute values of OptParams resource

Attribute	Value	Description
resourceType	"3"	Enumeration value for the *container* resource type
resourceName	"pOptParams"	"OptParams" property

Attribute	Value	Description
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
currentValue	(variable)	Read-only LabelProperty widget metadata

Table C.2.3-5: Attribute values of MetadataChange resource

Attribute	Value	Description
resourceType	"3"	Enumeration value for the *container* resource type
resourceName	"MetadataChange"	"MetadataChange" signal
labels	"lwked-Technology:AllJoyn", "lwked-Entity-Type:Signal Name"	

Table C.2.3-6: Attribute values of signal resource

Attribute	Value	Description
resourceType	"4"	Enumeration value for the *contentInstance* resource type
resourceName	(variable)	
content	"changed"	Notificiation that the metadata is changed

C.2.4 Action interface

AllJoyn Action(*org.alljoyn.ControlPanel.Action*) interface and its members shall be mapped with the following structure.

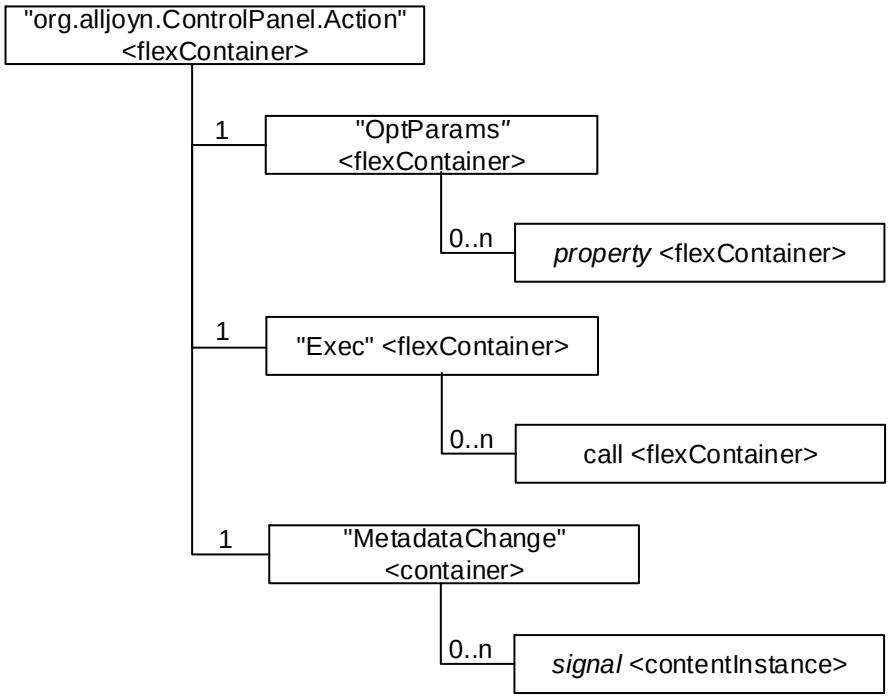

Figure C.2.4-1: Structure of Action interface mapping

Each resource in the figure above shall have the attribute values as follows.

Table C.2.4-1: Attribute values of org.alljoyn.ControlPanel.Action resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"org.alljoyn.ControlPanel.Container"	AllJoyn interface name

Attribute	Value	Description
containerDefinition	Value indicating alljoynInterface	AllJoyn interface
version	"1"	Version property value is "1"

Table C.2.4-2: Attribute values of States resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"States"	"States" property
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
currentValue	(variable)	Read-only state "enabled"

Table C.2.4-3: Attribute values of OptParams resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"OptParams"	"OptParams" property
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
currentValue	(variable)	Read-only "Action" widget metadata

Table C.2.4-4: Attribute values of *Exec* resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"Exec"	"Exec" method
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.2.4-5: Attribute values of call resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
callStatus	(variable)	

Table C.2.4-6: Attribute values of MetadataChange resource

Attribute	Value	Description
resourceType	"3"	Enumeration value for the *container* resource type
resourceName	"MetadataChange"	"MetadataChange" signal
labels	"lwked-Technology:AllJoyn", "lwked-Entity-Type:Signal Name"	

Table C.2.4-7: Attribute values of signal resource

Attribute	Value	Description
resourceType	"4"	Enumeration value for the *contentInstance* resource type
resourceName	(variable)	
content	"changed"	Notificiation that the metadata is changed

C.2.5 NotificationAction interface

AllJoyn NotificationAction(*org.alljoyn.ControlPanel.Container*) interface and its members shall be mapped with the following structure.

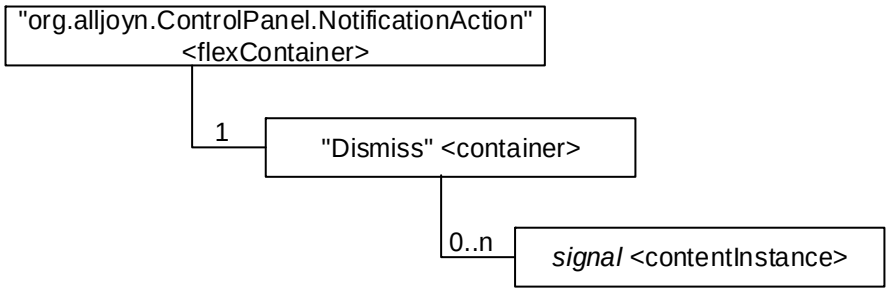

Figure C.2.5-1: Structure of *NotificationAction* interface mapping

Each resource in the figure above shall have the attribute values as follows.

Table C.2.5-1: Attribute values of org.alljoyn.ControlPanel.NotificationAction resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"org.alljoyn.ControlPanel.NotificationAction"	AllJoyn interface name
containerDefinition	Value indicating alljoynInterface	AllJoyn interface
version	"1"	Version property value is "1"

Table C.2.5-2: Attribute values of *Dismiss* resource

Attribute	Value	Description
resourceType	"3"	Enumeration value for the *container* resource type
resourceName	"Dismiss"	"Dismiss" signal
labels	"lwked-Technology:AllJoyn", "lwked-Entity-Type:Signal Name"	

Table C.2.5-3: Attribute values of signal resource

Attribute	Value	Description
resourceType	"4"	Enumeration value for the *contentInstance* resource type
resourceName	(variable)	
content	"dismiss"	Dismiss the notification panel

C.2.6 Dialog interface

AllJoyn Dialog(*org.alljoyn.ControlPanel.Dialog*) interface and its members shall be mapped with the following structure.

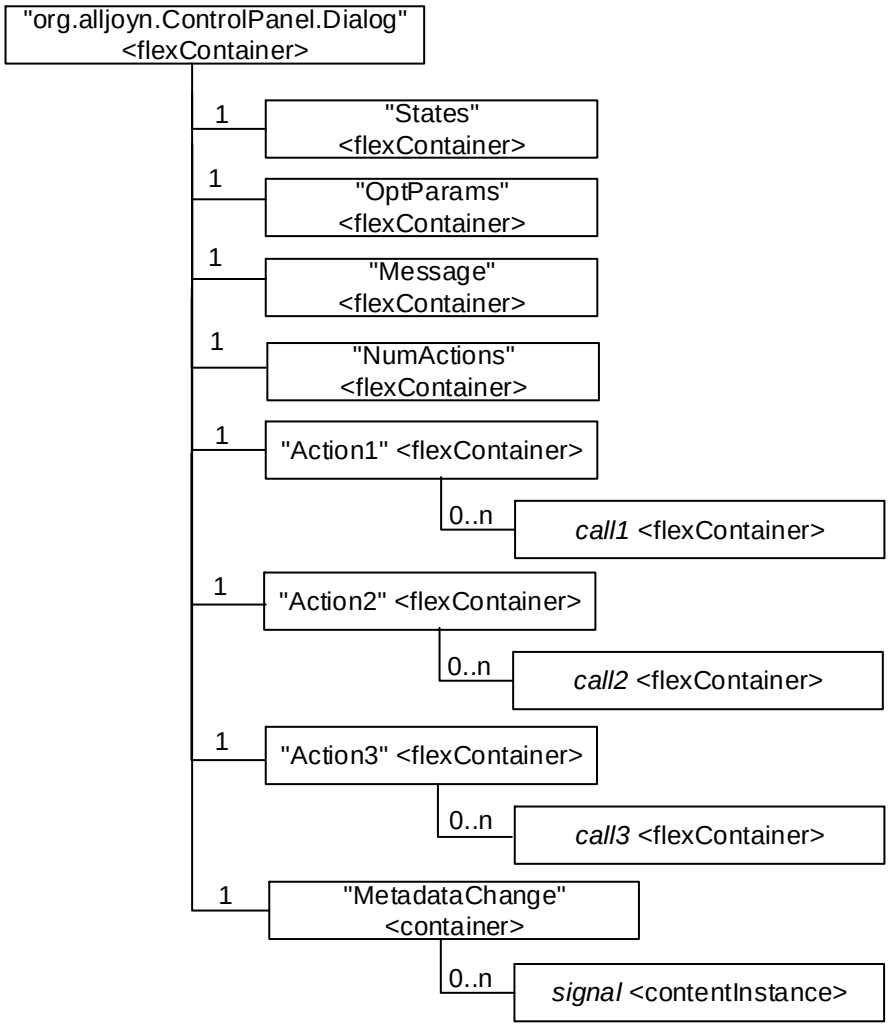

Figure C.2.6-1: Structure of Dialog interface mapping

Each resource in the figure above shall have the attribute values as follows.

Table C.2.6-1: Attribute values of org.alljoyn.ControlPanel.Dialog resource

Attribute	Value	Description
resourceType	"3"	Enumeration value for the *container* resource type
resourceName	"ControlPanel.Dialog"	*org.alljoyn.ControlPanel.Dialog* interface
containerDefinition	Value indicating alljoynInterface	AllJoyn interface
version	"1"	Version property value is "1"

Table C.2.6-2: Attribute values of States resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"States"	"States" property
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
currentValue	(variable)	Read-only state "enabled"

Table C.2.6-3: Attribute values of OptParams resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"OptParams"	"OptParams" property
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
currentValue	(variable)	Read-only "Dialog" widget metadata

Table C.2.4-4: Attribute values of Message resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"States"	"States" property
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
currentValue	(variable)	Read-only display message

Table C.2.4-5: Attribute values of NumActions resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"States"	"States" property
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
currentValue	(variable)	Read-only number of actions

Table C.2.6-6: Attribute values of Action1 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"Action1"	"Action1" method
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.2.6-7: Attribute values of call1 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
callStatus	(variable)	

Table C.2.6-8: Attribute values of Action2 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"Action2"	"Action2" method
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.2.6-9: Attribute values of call2 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	

Attribute	Value	Description
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
callStatus	(variable)	

Table C.2.6-10: Attribute values of Action3 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"Action3"	"Action3" method
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.2.6-11: Attribute values of call3 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
callStatus	(variable)	

Table C.2.6-12: Attribute values of MetadataChange resource

Attribute	Value	Description
resourceType	"3"	Enumeration value for the *container* resource type
resourceName	"MetadataChange"	"MetadataChange" signal
labels	"Iwked-Technology:AllJoyn", "Iwked-Entity-Type:Signal Name"	

Table C.2.6-13: Attribute values of signal resource

Attribute	Value	Description
resourceType	"4"	Enumeration value for the *contentInstance* resource type
resourceName	(variable)	
content	"changed"	Notificiation that the metadata is changed

C.2.7 ListProperty interface

AllJoyn ListProperty(*org.alljoyn.ControlPanel.ListProperty*) interface and its members shall be mapped with the following structure.

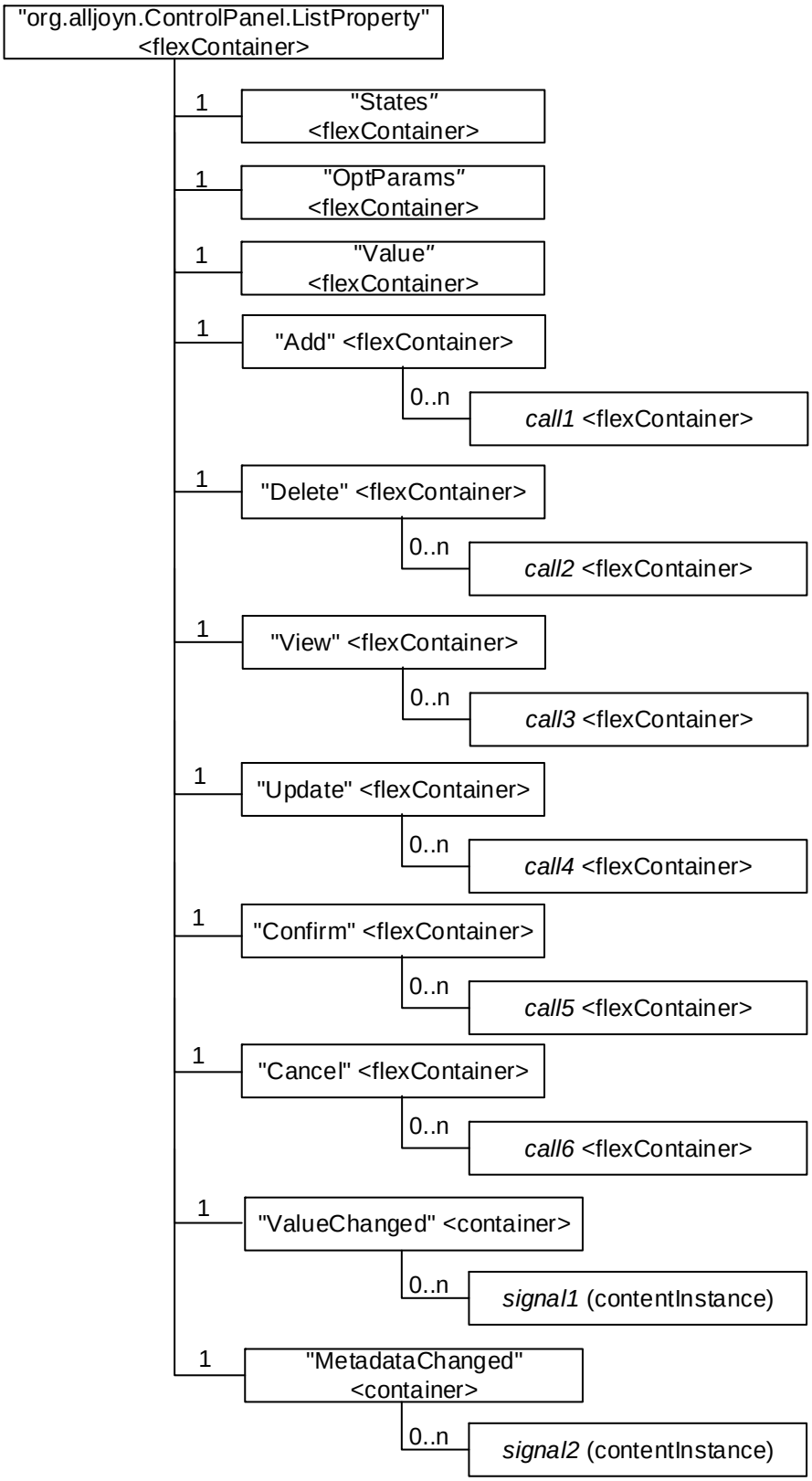

Figure C.2.7-1: Structure of *ListProperty* interface mapping

Each resource in the figure above shall have the attribute values as follows.

Table C.2.7-1: Attribute values of org.alljoyn.ControlPanel.ListProperty resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"org.alljoyn.ControlPanel.ListProperty"	AllJoyn interface name
containerDefinition	Value indicating alljoynInterface	AllJoyn interface
version	"1"	Version property value is "1"

Table C.2.7-2: Attribute values of States resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"States"	"States" property
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
currentValue	(variable)	Read-only state "enabled"

Table C.2.7-3: Attribute values of OptParams resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"OptParams"	"OptParams" property
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
currentValue	(variable)	Read-only "ListProperty" widget metadata

Table C.2.7-4: Attribute values of Value resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"Value"	"Value" property
containerDefinition	Value indicating alljoynProperty	AllJoyn interface property
currentValue	(variable)	Read-only list of records

Table C.2.7-5: Attribute values of Add resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"Add"	"Add" method
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.2.7-6: Attribute values of call1 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
callStatus	(variable)	

Table C.2.7-7: Attribute values of Delete resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"Delete"	"Delete" method
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.2.7-8: Attribute values of call2 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
input	(variable)	recordID
callStatus	(variable)	

Table C.2.7-9: Attribute values of View resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"View"	"View" method
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.2.7-10: Attribute values of call3 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
input	(variable)	recordID
callStatus	(variable)	

Table C.2.7-11: Attribute values of Update resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"Update"	"Update" method
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.2.7-12: Attribute values of call4 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
input	(variable)	recordID
callStatus	(variable)	

Table C.2.7-13: Attribute values of Confirm resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"Confirm"	"Confirm" method

Attribute	Value	Description
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.2.7-14: Attribute values of call5 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
callStatus	(variable)	

Table C.2.7-15: Attribute values of Cancel resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"Cancel"	"Cancel" method
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.2.7-16: Attribute values of call6 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
callStatus	(variable)	

Table C.2.7-17: Attribute values of ValueChanged resource

Attribute	Value	Description
resourceType	"3"	Enumeration value for the *container* resource type
resourceName	"Changed"	"ValueChanged" signal
labels	"Iwked-Technology:AllJoyn", "Iwked-Entity-Type:Signal Name"	

Table C.2.7-18: Attribute values of signal1 resource

Attribute	Value	Description
resourceType	"4"	Enumeration value for the *contentInstance* resource type
content	"ValueChanged""	Notificiation that the property value is changed

Table C.2.7-19: Attribute values of MetadataChanged resource

Attribute	Value	Description
resourceType	"3"	Enumeration value for the *container* resource type
resourceName	"Changed"	"MetadataChanged" signal
labels	"Iwked-Technology:AllJoyn", "Iwked-Entity-Type:Signal Name"	

Table C.2.7-20: Attribute values of signal2 resource

Attribute	Value	Description
resourceType	"4"	Enumeration value for the *contentInstance* resource type
content	"MetadataChanged"	Notificiation that the metadata is changed

C.3 Configuration service

C.3.1 Config interface

AllJoyn Config(*org.alljoyn.Config*) interface and its members shall be mapped with the following structure.

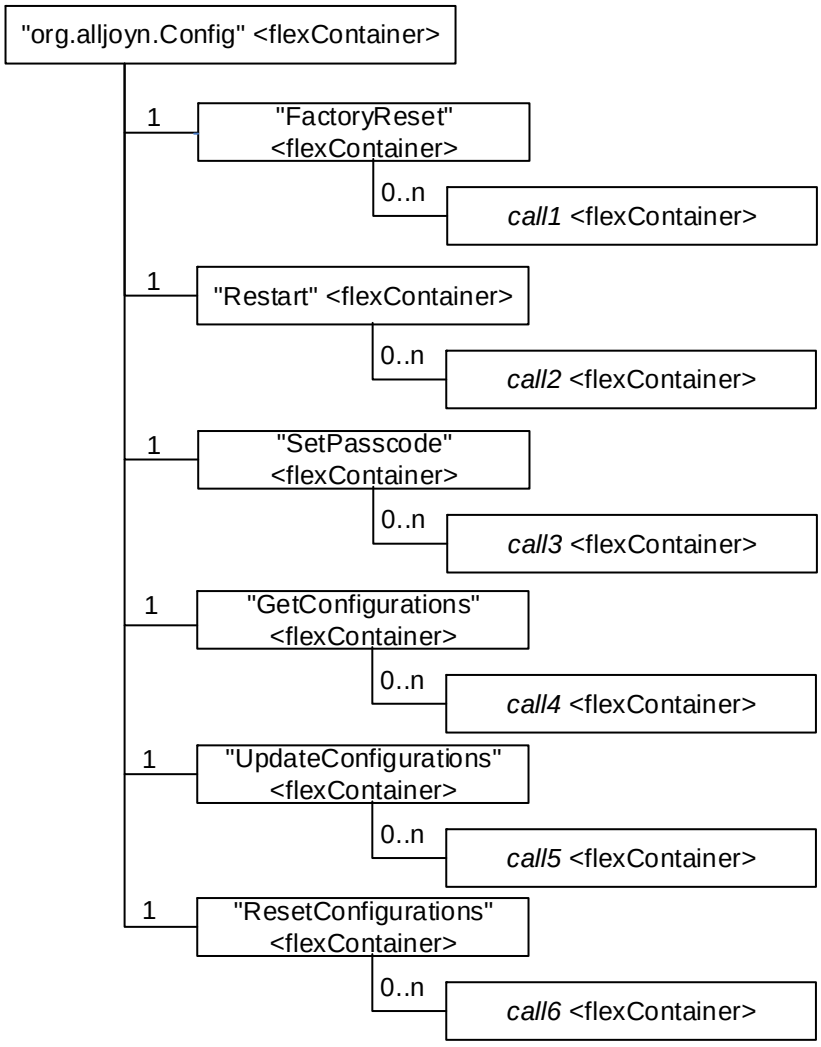

Figure C.3.1-1: Structure of Config interface mapping

Each resource in the figure above shall have the attribute values as follows.

Table C.3.1-1: Attribute values of Config resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"org.alljoyn.Config"	AllJoyn interface name
containerDefinition	Value indicating alljoynInterface	AllJoyn interface

Attribute	Value	Description
version	"1"	Version property value is "1"

Table C.2.3-2: Attribute values of FactoryReset resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"FactoryReset"	"FactoryReset" method
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.2.3-3: Attribute values of call1 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
callStatus	(variable)	

Table C.2.3-4: Attribute values of Restart resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"Restart"	"Restart" method
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.2.3-5: Attribute values of call2 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
callStatus	(variable)	

Table C.2.3-6: Attribute values of SetPasscode resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"SetPasscode"	"SetPasscode" method
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.2.3-7: Attribute values of call3 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
input	(variable)	"daeonRealm" and "newPasscode" parameter
callStatus	(variable)	

Table C.2.3-8: Attribute values of GetConfigurations resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"GetConfigurations"	"GetConfigurations" method
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.2.3-9: Attribute values of call4 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
input	(variable)	"languageTag" parameter
callStatus	(variable)	

Table C.2.3-10: Attribute values of UpdateConfigurations resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"UpdateConfigurations"	"UpdateConfigurations" method
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.2.3-11: Attribute values of call5 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
input	(variable)	"languageTag" and "configMap" parameter
callStatus	(variable)	

Table C.2.3-12: Attribute values of ResetConfigurations resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	"ResetConfigurations"	"ResetConfigurations" method
containerDefinition	Value indicating alljoynMethod	AllJoyn interface method

Table C.2.3-13: Attribute values of call6 resource

Attribute	Value	Description
resourceType	"28"	Enumeration value for the *flexContainer* resource type
resourceName	(variable)	
containerDefinition	Value indicating alljoynMethodCall	AllJoyn interface method call
input	(variable)	"languageTag" and "fieldList" parameter
callStatus	(variable)	

Annex D (informative): Bibliography

AllJoyn® Gateway Agent High Level Description (HLD).

NOTE: Available at http://allseenalliance.org/framework/documentation.

History

Document history		
V2.0.0	September 2016	Publication