

ETSI TS 118 121 V2.0.1 (2020-03)



TECHNICAL SPECIFICATION

oneM2M; oneM2M and AllJoyn® Interworking (oneM2M TS-0021 version 2.0.1 Release 2A)

*iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard available at
<https://standards.iteh.ai/catalog/standards/sis/23173d632-d330-4c7a-a35b-0a265888d562/etsi-ts-118-121-v2-0-1-2020-03>*



ReferenceRTS/oneM2M-000021v2A

Keywordsinterworking, 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

Important notice

The present document can be downloaded from:

<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status.

Information on the current status of this and other ETSI documents is available at

<https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:

<https://portal.etsi.org/People/CommiteeSupportStaff.aspx>

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2020.

All rights reserved.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.

3GPP™ and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

oneM2M™ logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners.

GSM® and the GSM logo are trademarks registered and owned by the GSM Association.

Contents

Intellectual Property Rights	5
Foreword.....	5
1 Scope	6
2 References	6
2.1 Normative references	6
2.2 Informative references.....	6
3 Definition of terms, symbols and abbreviations.....	7
3.1 Terms.....	7
3.2 Symbols.....	7
3.3 Abbreviations	7
4 Conventions.....	7
5 Architecture Model.....	7
5.1 Reference model.....	7
5.2 Composition of the IPE	8
6 Architecture Aspect.....	9
6.1 Introduction	9
6.2 oneM2M resource mapping structure.....	9
6.3 AllJoyn IPE registration	13
6.4 AllJoyn service discovery	14
6.5 AllJoyn service mapping	14
6.5.1 AllJoyn service categorization.....	14
6.5.2 AllJoyn base service mapping	14
6.5.3 Other framework service mapping.....	15
6.5.4 AllJoyn app-specific service mapping.....	15
6.6 AllJoyn access control mapping.....	15
7 Interworking Procedures	15
7.1 Introduction	15
7.2 IPE initial configuration	15
7.2.1 Introduction.....	15
7.2.2 IPE registration and AllJoyn service discovery	15
7.2.3 AllJoyn Service Resource creation	16
7.2.4 Subscription creation	18
7.2.4.1 Subscription creation by AE/CSE(s) acting as service producers.....	18
7.2.4.2 Subscription creation by entities acting as service consumers	19
7.3 IPE interworking	20
7.3.1 Introduction.....	20
7.3.2 AllJoyn services consumed by AE/CSE(s).....	21
7.3.3 AllJoyn services produced by oneM2M AE/CSE(s) via the IPE.....	21
7.3.4 AllJoyn services addition and deletion	22
Annex A (informative): oneM2M and AllJoyn technical comparison.....	23
A.1 API styles	23
A.2 Service Discovery/Advertisement	26
Annex B (normative): <flexContainer> resource specializations	27
B.1 Introduction	27
B.2 Resource Type <i>svcObjWrapper</i>	27
B.3 Resource Type <i>svcFwWrapper</i>	28
B.4 Resource Type <i>allJoynApp</i>	29

B.5 Resource Type *allJoynSvcObject*32

B.6 Resource Type *allJoynInterface*34

B.7 Resource Type *allJoynMethod*36

B.8 Resource Type *allJoynMethodCall*37

B.9 Resource Type *allJoynProperty*39

Annex C (normative): AllJoyn interface mapping to oneM2M resource structure.....42

C.1 Notification service42

C.1.1 Notification interface42

C.1.2 Notification Producer interface43

C.1.3 Dismissal interface44

C.2 Control Panel service.....45

C.2.1 Container interface45

C.2.2 Property interface46

C.2.3 LabelProperty interface48

C.2.4 Action interface49

C.2.5 NotificationAction interface51

C.2.6 Dialog interface52

C.2.7 ListProperty interface55

C.3 Configuration service59

C.3.1 Config interface59

History62

iTeh STANDARD PREVIEW
 (standards.iteh.ai)
 Full standard:
<https://standards.iteh.ai/catalog/standards/sist/3173d632-d330-4c7a-a35b-0a265888d562/etsi-ts-118-121-v2.0.1-2020-03>

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables 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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

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

ETSI STANDARD REVIEW
(standards.etsi.org)
Full standard:
<https://standards.etsi.org/catalog/standards/sist/3373d632-d330-4c7a-a35b-0a265888d562/etsi-ts-118-121-v2.0.1-2020-03>

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.

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; Service Layer Core Protocol Specification (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.
- [i.3] oneM2M TR-0014: "oneM2M and AllJoyn Interworking".
- [i.4] D-Bus Tutorial.

NOTE: Available at <http://dbus.freedesktop.org/doc/dbus-tutorial.html#objects>.

- [i.5] AllJoyn Gateway Agent High Level Description(HLD).

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms 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

3.2 Symbols

Void.

3.3 Abbreviations

Void.

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 [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 describes the AllJoyn IPE that supports the following reference model.

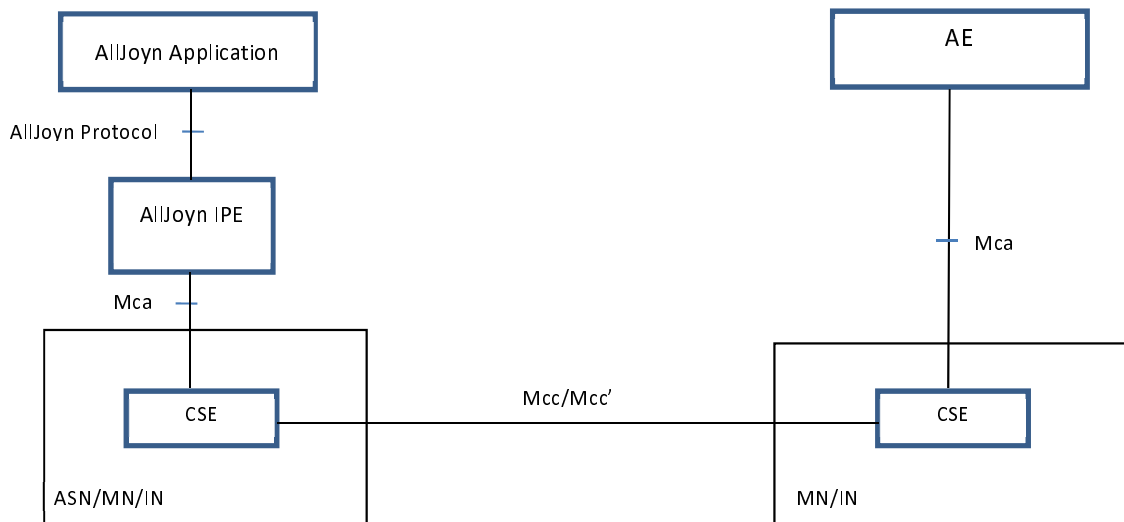


Figure 5.1-1: AllJoyn interworking reference model

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.

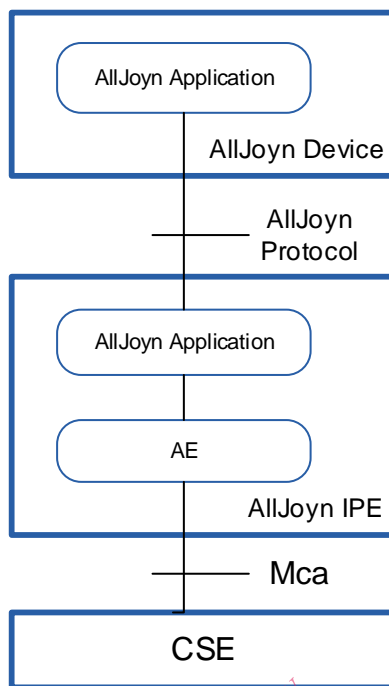


Figure 5.2-1: Composition of the AllJoyn IPE

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 introduced. 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.

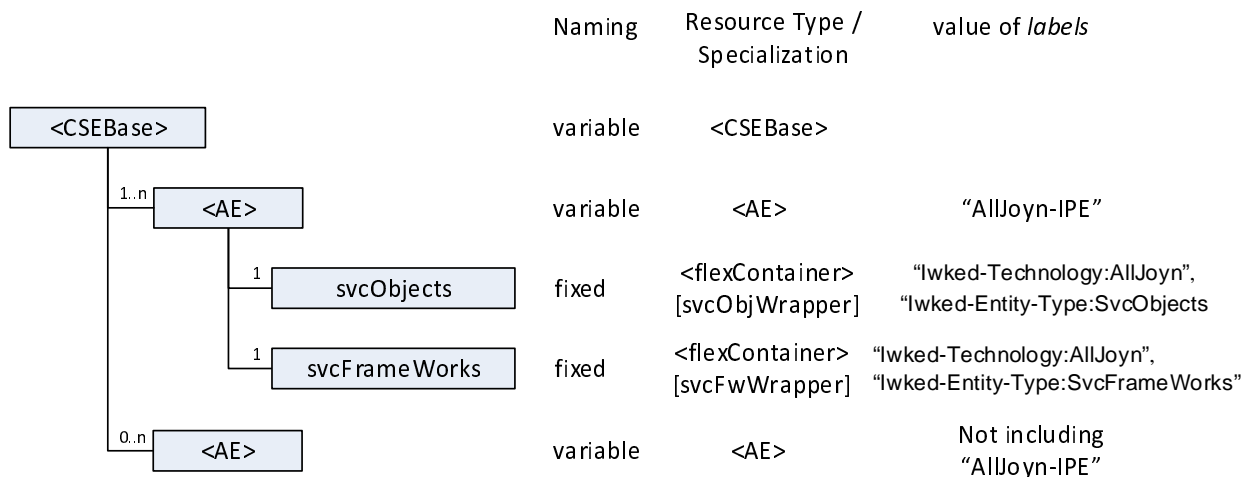


Figure 6.2-1: AllJoyn interworking resource mapping, part 1

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, at least one <AE> resource representing an AllJoyn 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.

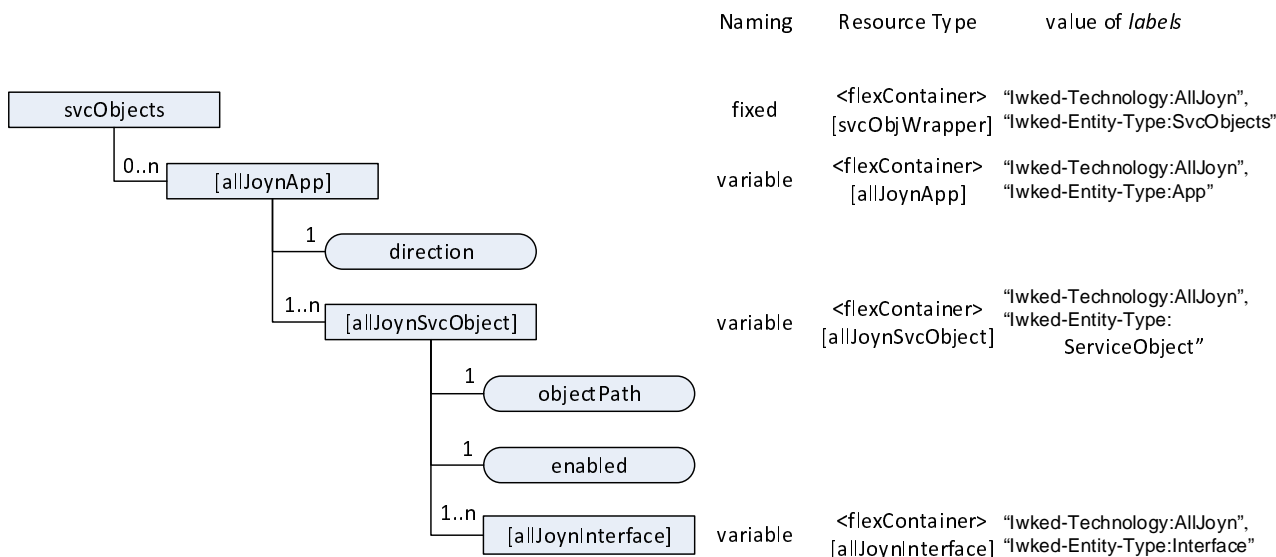


Figure 6.2-2: AllJoyn interworking resource mapping, part 2

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 application. 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 representing 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] resource, 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.

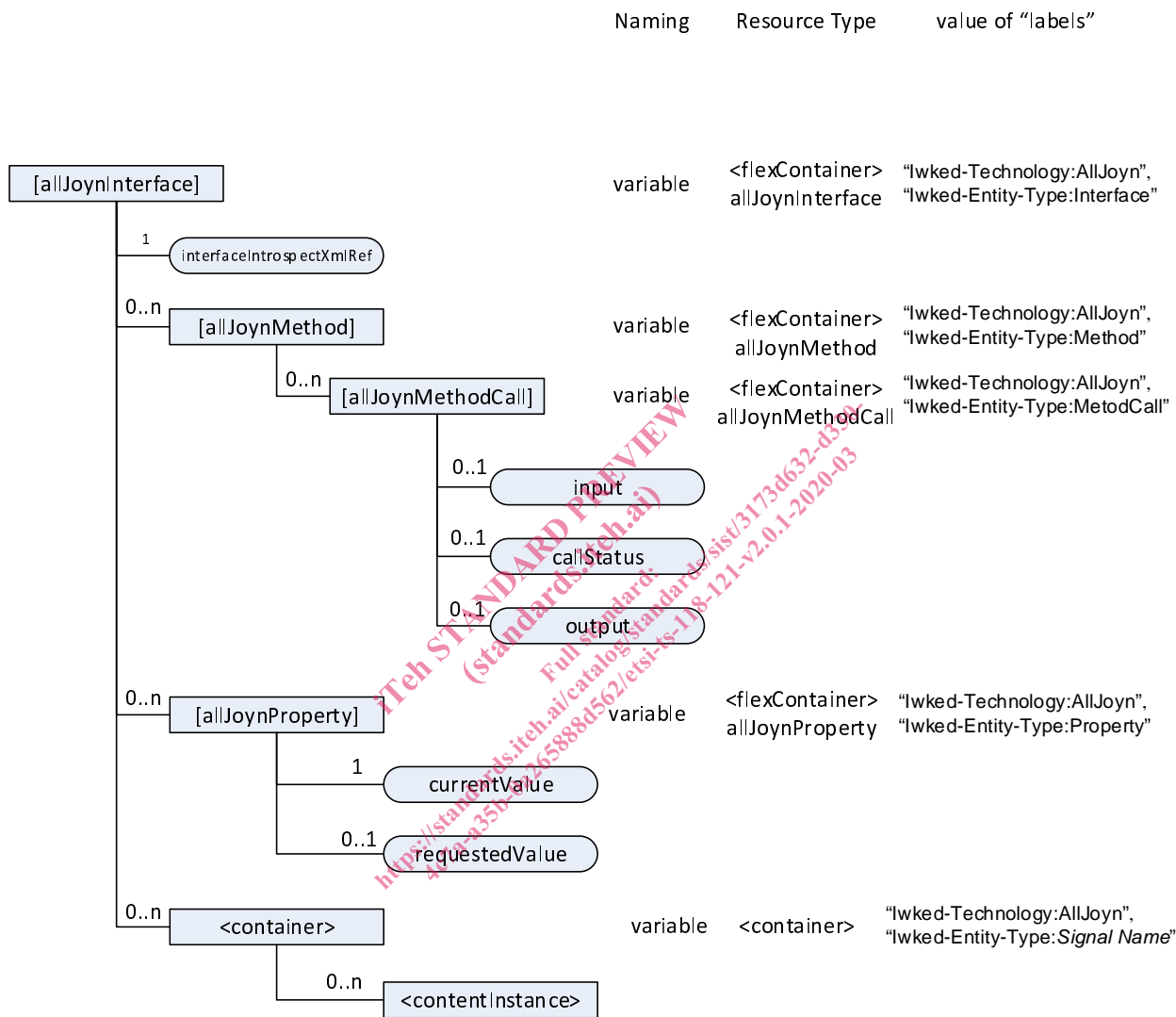


Figure 6.2-3: AllJoyn interworking resource mapping, part 3

[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 represented 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 *output* 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*.

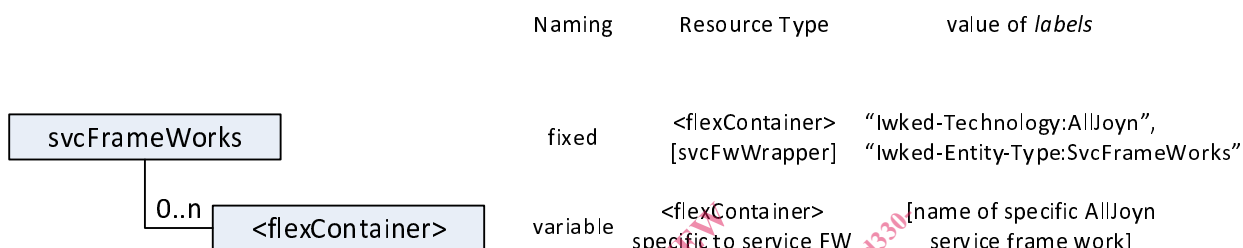


Figure 6.2-4: AllJoyn interworking resource mapping, part 4

For exposure of AllJoyn service frameworks using higher level APIs that are not directly interacting with AllJoyn service objects 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 string equal 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.