ETSI TR 118 567 V4.0.0 (2021-11)

TECHNICAL REPORT

oneM2M: Study on Management Object migration to SDT (oneM2M TR-0067 version 4.0.0)

Reference

DTR/oneM2M-000067

Keywords

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 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

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

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

Contents

Intellectual Property Rights 4

Foreword 4

1 Scope 5

2 References 5

2.1 Normative references 5

2.2 Informative references 5

3 Definition of terms, symbols and abbreviations 6

3.1 Terms 6

3.2 Symbols 6

3.3 Abbreviations 6

4 Conventions 7

5 Introduction 7

5.0 Background 7

5.1 Analysis Background 8

5.2 DM Architecture 9

5.3 Conclusion and Proposal 10

5.3.0 Summary 10

5.3.1 Conclusion 11

Annex A: Proposal for update of ETSI TS 118 101 12

A.1 Introduction 12

Annex B: Proposal for update of ETSI TS 118 104 24

B.1 Introduction 24

Annex C: Proposal for update of ETSI TS 118 123 27

C.1 Introduction 27

History 37

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are 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 Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, 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.

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.

Foreword

This Technical Report (TR) has been produced by ETSI Partnership Project oneM2M (oneM2M).

1 Scope

The present document studies the completion of SDT (Smart Device Template) using <flexContainer> resource specializations and the possible migration of the existing device management model using Management Object (<mgmtObj>).

The present document is initiated in the context of the Management Object Migration having the following objectives (extract):

In Release 4, SDT (Smart Device Template) has been extended to include device management functions in addition to the existing services. This justified extension creates a new way to perform device management compared to the existing Device Management (DM) Common Services Function (CSF) model using <mgmtObj>.

In order not to live with 2 solutions for the same purpose, the present document proposes to work on a transition phase allowing implementation using DM <mgmtObj> to migrate towards SDT model.

The scope of the present document is to study the Device Management model based on Management Object (<mgmtObj>) model migration towards SDT model.

This action plan includes to:

1) Provide a temporary TR with a mapping between <mgmtObj> and the SDT DM <flexContainer>.

2) List in the present document all the specifications and sections that will have to be updated when <mgmtObj> will be replaced by SDT.

3) List in the present document the issues to be resolved by removing the <mgmtObj> after migration and the proposed solutions.

4) Depending on the present document outcomes, decide whether <mgmtObj> should be removed or not in oneM2M Release 4.

New tasks may be added during the process if necessary.

The present document intends to provide the study as part of the action plan above.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

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 https://www.onem2m.org/images/files/oneM2M-Drafting-Rules.pdf.

[i.2] ETSI TS 118 123: "oneM2M; Home Appliances Information Model and Mapping (oneM2M TS-0023)".

[i.3] ETSI TS 118 101: "oneM2M; Functional Architecture (oneM2M TS-0001)".

[i.4] ETSI TS 118 104: "oneM2M; Service Layer Core Protocol (oneM2M TS-0004)".

[i.5] ETSI TS 118 114: "oneM2M; LWM2M Interworking (oneM2M TS-0014)".

[i.6] ETSI TS 118 122: "oneM2M; Field Device Configuration (oneM2M TS-0022)".

[i.7] ETSI TS 118 103: "oneM2M; Security solutions (oneM2M TS-0003)".

[i.8] Work Item 0096: "Effective_IoT_Communication_to_Protect_3GPP_Networks-V0_0_2".

NOTE: Available at https://ftp.onem2m.org/Work%20Programme/.

[i.9] Broadband Forum TR-069: "CPE WAN Management Protocol Issue": 1 Amendment 5, November 2013.

[i.10] OMA-DM: "OMA Device Management Protocol", Version 1.3, Open Mobile Alliance.

[i.11] LWM2M: "OMA LightweightM2M", Version 1.0, Open Mobile Alliance.

[i.12] JNI 6.0 API Specification: "Java Native Interface 6.0 Specification".

[i.13] IEEE 802.11™-2020: "IEEE Standard for Information Technology--Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks--Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications".

[i.14] ETSI TS 118 116: "oneM2M; Secure Environment Abstraction (oneM2M TS-0016)".

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ADN	Application Dedicated Node
AE	Application Entity
API	Application Program Interface
ASN	Application Service Node
BBF	BroadBand Forum
CMDH	Communication Management and Delivery Handling
CR	Create Retrieve
CSE	Common Services Entity
CSF	Common Services Function
DM	Device Management
FFS	For Further Study
IN	Infrastructure Node
IN/MN	Infrastructure Node/Middle Node
IN-CSE	Infrastructure Node - Common Services Entity
IPE	Interworking Proxy application Entity
JNI	Java Native Interface

LWM2M	LightWeight M2M
MA	Mandatory Announced
MAF	M2M Authentication Function
MEF	M2M Enrolment Function
MN	Middle Node
MN/ASN	Middle Node/Application Service Node
MN-CSE	Middle Node - Common Services Entity
NA	Not Announced
NP	Not Present
OA	Optionally Announced
OMA	Open Mobile Alliance
RAM	Random Access Memory
RO	Read Only
RW	Read Write
SDT	Smart Device Template
TBC	To Be Completed
TBD	To Be Determined
TP	Traffic Patterns
TR	Technical Report
TS	Technical Specification
URI	Uniform Resource Identifier
WO	Write Once
XSD	XML Schema Definition

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 Introduction

5.0 Background

In the Release 4 of the ETSI TS 118 123 [i.2] specification (SDT-based Information Model and Mapping for Vertical Industries) was introduced a new approach for Device Management, based on Smart Device Template ModuleClasses, a concept that is mapped in oneM2M as <flexContainer> resource specializations.

The benefits of this approach are:

- Unified Device and Service Management of all nodes, including ADN or NoDN devices: it is possible to use the same type of resources for handling both the *functional behaviour* of devices and their *remote management* in the usual DM meaning (reboot, firmware update, configuration, log, etc.). Before this, the DM aspect was performed through <mgmtObj> resources, and the functional aspect through <container> or <flexContainer> resources.

- Enhanced expressivity. The SDT design allows a powerful information model to describe devices, with concepts such as Devices, SubDevices, ModuleClasses, Actions, DataPoints and Properties. The <mgmtObj> has a poorer semantics.

- Standardized, flexible, extensible and incremental Data Model. Adding new Devices, ModuleClasses or DataPoints is an easy process, compared with <mgmtObj> resources.

- Automated support for the generation of XML/XSD templates for DM ModuleClasses.

- Possible historization of the updates on DM resources, through the <flexContainerInstance> mechanism.

- Standardized ontological model: the SDT Information Model is aligned with the oneM2M Base Ontology (see clause 8 in ETSI TS 118 123 [i.2]).

- Ease of use. For oneM2M application developers, handling <flexContainers> that map ModuleClasses or Actions is quite simple and natural. For example, triggering a [reboot] SDT Action is very similar to calling a method, with a *rebootType* parameter, and more satisfying / less ambiguous than the [reboot] <mgmtObj> with its 2 writable attributes *reboot* and *factoryReset*.

The present document proposes to extend this approach to all <mgmtObj> resources.

5.1 Analysis Background

Table 5.1-1 lists specializations of <mgmtObj> resources that are currently specified in oneM2M Release 4 (referenced in ETSI TS 118 101 [i.3], clause 9.6.1.2.1):

- The first group handles CMDH policy management (Communication Management and Delivery Handling), specified in ETSI TS 118 101 [i.3], clause D.12. As the Work Item WI-0096 [i.8] is currently working on an evolution of CMDH based on flexContainers, these resources will not be considered in the present document.

- The second group handles Device Configuration in the Field Domain, defined in ETSI TS 118 103 [i.7] Security Solutions and specified in ETSI TS 118 122 [i.6] Field Device Configuration.

- The third group handles Device Management in the "classical" meaning, i.e. the remote management of devices (reboot, firmware/software management, configuration, logging, etc.). These <mgmtObj> already have been mapped to SDT <flexContainer> specializations in ETSI TS 118 123 [i.2].

Table 5.1-1

Resource specialization	Short Description	Comment
CMDH CSF (ETSI TS 118 101 [i.3], clauses 6.2.2 and D.12)		
activeCmdhPolicy	Provides a link to the currently active set of CMDH policies	
cmdhBuffer	Defines CMDH buffer usage limits	
cmdhDefaults	Defines CMDH default values	
cmdhEcDefParamValues	Represent a specific set of default values for the CMDH related parameters	
cmdhDefEcValue	Defines a value for the ***Event Category*** parameter of an incoming request when it is not defined	
cmdhLimits	Defines limits for CMDH related parameter values	
cmdhNetworkAccessRules	Defines rules for the usage of underlying networks	
cmdhNwAccessRule	Defines a rule for the usage of underlying networks	
cmdhPolicy	A set of rules defining which CMDH parameters will be used by default	
Field Device Configuration (ETSI TS 118 122 [i.6])		
registration	To convey the service layer configuration information	
dataCollection	To convey the application configuration information	
authenticationProfile	To convey the configuration information regarding establishing mutually-authenticated secure communications	
myCertFileCred	To configure a certificate or certificate chain	
trustAnchorCred	To identify a trust anchor certificate for validation of certificates	
MAFClientRegCfg	To convey instructions regarding the MAF Client Registration procedure	
MEFClientRegCfg	To convey instructions regarding the MEF Client Registration procedure	
wifiClient	To set up configuration of IEEE 802.11™-2020 [i.13] connection on the client device.	
DM CSF (ETSI TS 118 101 [i.3], clauses 6.2.4, D.1 to D.11 and D.13)		
areaNwkDeviceInfo	Provides information about the Node in the M2M Area Network	ETSI TS 118 123 [i.2], clause 5.8.11 "dmAreaNwkDeviceInfo"
areaNwkInfo	Describes the list of Nodes attached behind the MN node and its physical or underlying relation among the nodes in the M2M Area Network	ETSI TS 118 123 [i.2], clause 5.8.10 "dmAreaNwkInfo"

Resource specialization	Short Description	Comment
battery	Provides the power information of the node (e.g. remaining battery charge)	ETSI TS 118 123 [i.2], clauses 5.3.1.10 "battery" and 5.8.2 "flexNode"
deviceCapability	Contains information about the capability supported by the Node	ETSI TS 118 123 [i.2], clause 5.8.12 "dmDeviceCapability"
deviceInfo	Contains information about the identity, manufacturer and model number of the device	ETSI TS 118 123 [i.2], clause 5.8.4 "dmDeviceInfo"
eventLog	Contains information about the log of events of the Node	ETSI TS 118 123 [i.2], clause 5.8.8 "dmEventLog"
firmware	Provides information about the firmware of the Node (e.g. name, version)	ETSI TS 118 123 [i.2], clause 5.8.6 "dmFirmware"
memory	Provides the memory (typically RAM) information of the node (e.g. the amount of total volatile memory)	ETSI TS 118 123 [i.2], clause 5.8.3 "dmAgent"
reboot	Used to reboot or reset the Node	ETSI TS 118 123 [i.2], clause 5.8.3 "dmAgent"
software	Provides information about the software of the Node	ETSI TS 118 123 [i.2], clauses 5.8.7 "dmSoftware" and 5.8.9 "dmPackage"
storage	To manage available storage memory on the device	ETSI TS 118 123 [i.2], clause 5.8.13 "dmStorage"

5.2 DM Architecture

Currently, the DM Architecture is composed of two parts:

1) The <node> and <mgmtObj> model defined in ETSI TS 118 101 [i.3], clause 9.6.18.

2) The [flexNode] and [DM Module classes] model defined in ETSI TS 118 123 [i.2], clause 5.8.

These models coexisted, i.e. an IPE could create a device representation in SDT with a [flexNode] and/or a <node>. This choice was made in order not to modify the <node> resource, but it adds a lot of complexity.

It is proposed here to move the [flexNode] as child of the <node>, which supposes letting the <node> resource capable of having <flexContainer> children.

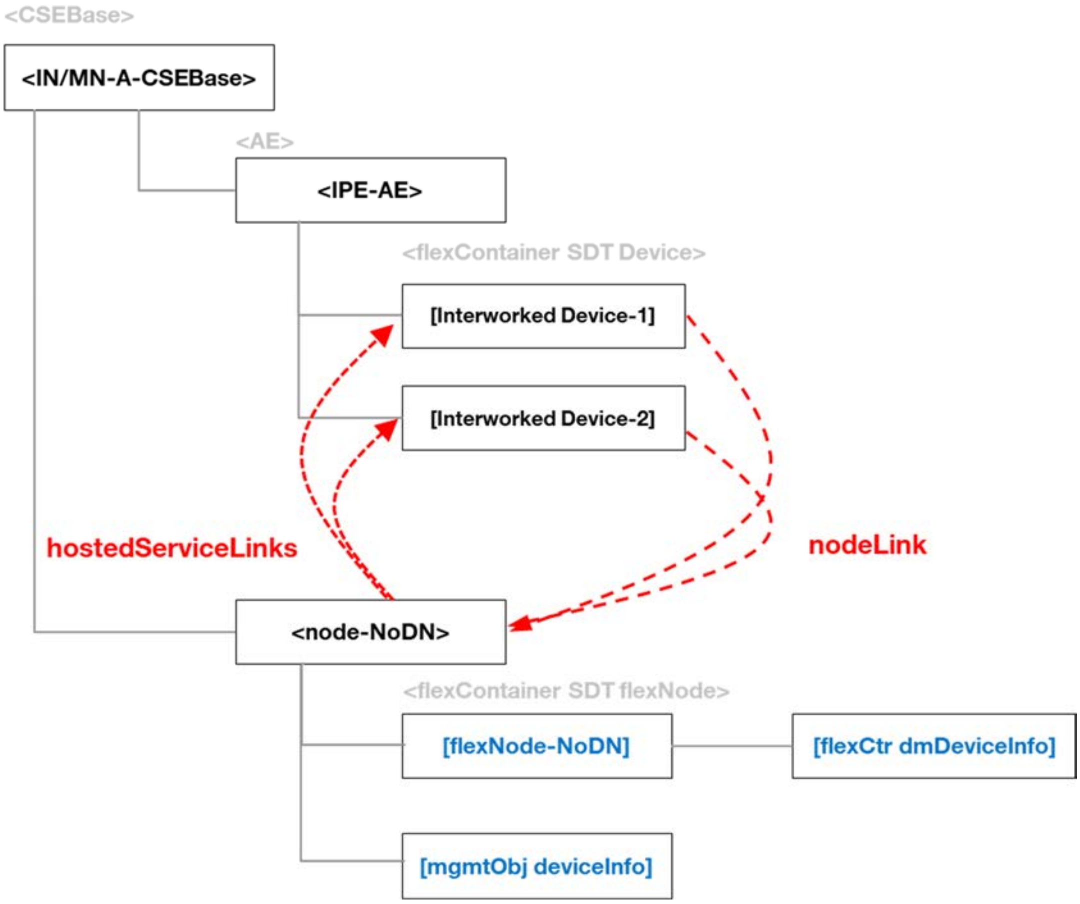

Figure 5.2-1: New DM architecture proposal

The resources in blue in the above figure are not supposed to coexist: either the <node> has <mgmtObj> children such as the [deviceInfo] specialization, or it has a [flexNode] child which is the root of DM <flexContainer> specializations such as [dmDeviceInfo]. It is defined a [dmBaseModule] SDT module class, mapped as a <flexContainer> specialization, and use the SDT inheritance mechanism to state that the <flexContainer> children of the [flexNode] (now referred to as DM <flexContainer>) extend / inherit from this [dmBaseModule] flexContainer.

This architecture presents three main benefits:

1) It forbids having <mgmtObj> resources and their equivalent <flexContainers> at the same level under the <node> resource, which would bring some confusion.

2) It maintains most of the current ETSI TS 118 123 [i.2] DM architecture (the emplacing of the [flexNode] was not forced), just removing the notion of *flexNodeLink* that linked an IPE to the associated [flexNode].

3) It maintains the current <node> / <mgmtObj> DM architecture, allowing a smoother transition from <mgmtObj>-based to <flexContainer>-based implementations.

5.3 Conclusion and Proposal

5.3.0 Summary

In the annexes of the present document it is presented an outline of the envisioned modifications that should be made to each oneM2M Technical Specification (TS) document that handles <mgmtObj> resources in order to take into account this proposed approach for Device Management.

Each annex is dedicated to an existing TS that deals with <mgmtObj> resources, and proposed a skeleton of the changes that could be made for using <flexContainers> instead.

5.3.1 Conclusion

In ETSI TS 118 101 [i.3], the Device Management architecture presented in clause 6.2.4 relies on Management Adapters, abstract components that communicate to external Management Servers of Management Clients through undefined interfaces ms/la. This is a very generic framework, that implies complex handling because it is the CSE itself that is responsible for managing so-called Management Sessions (in clause 10.2.8 and ETSI TS 118 104 [i.4], clause 7.3.4).

A possible implementation of this architecture relies on the IPE principles, and this is the approach that is used in ETSI TS 118 114 [i.5], the only existing specification of an implementation of the DM CSF for a specific DM protocol (here LwM2M). The ETSI TS 118 114 [i.5] does not explicitly refer to the Adapter/ms/la concepts, but it does correspond to an implementation of the ms interface (interface between a Management Adapter and an external Management Server). Nevertheless, DM CSF as specified in ETSI TS 118 101 [i.3] and ETSI TS 118 104 [i.4] is more general.

It is interesting for managing devices that can individually register as DM clients, and it is designed to fully map an existing DM data model into oneM2M resources.

It is therefore recommended that it is not desirable to engage in a full migration of the current DM CSF based on <mgmtObj> to a SDT-based DM because:

- <mgmtObj> mapping requires a one-to-one mapping with external DM technology data model, which is fine for DM technologies such as OMA DM [i.10], LwM2M [i.11] or BBF TR-069 [i.9], but not adapted to other IoT standards.
- <mgmtObj> are used in oneM2M security-related specifications (ETSI TS 118 103 [i.7], ETSI TS 118 116 [i.14], ETSI TS 118 122 [i.6] and ETSI TS 118 123 [i.2]), with specializations that are not purely DM constructs.
- It would complexify the SDT-based approach that should remain simple, restrained, abstract and IPE-based to handle all IoT networks. This is a CSE-based model, not IPE-based.

5.3.2 Proposal

Finally, the conclusion of this study is that the <mgmtObj> migration is a dead end due to some features (see above) that cannot be mapped in a useful way and thus, it is proposed that <mgmtObj> should not be removed from oneM2M Release 4.

Annex A: Proposal for update of ETSI TS 118 101

A.1 Introduction

In this annex, are presented the proposed changes to the ETSI TS 118 101 [i.3] specification for flexContainer introduction for device management operations.

---------------------- Start of change 1 ---

9.6.1.1 Resource Type Summary

TBD: Add <flexContainer> in <node>'s Child Resource Types and <node> in <flexContainer>'s Parent Resource Types.

---------------------- End of change 1 ---

---------------------- Start of change 2 ---

9.6.18 Resource Type *node*

The <*node*> resource represents specific information that provides properties of an M2M Node that can be utilized by other oneM2M operations. It contains child resources that represent the Node's context information (e.g. memory and battery), network topology, device information, device capability etc. These resources can be mapped in two distinct models:

1) Either the <*node*> resource has specialization of the <*mgmtObj*> as its child resources. The specialized <*mgmtObj*> resources are used to perform management of the Node.

2) Or the <*node*> resource has the [*flexNode*] specialization of a <*flexContainer*> as child resource, and this [*flexNode*] has specialization of the <*flexContainer*> as its child resources. These <*flexContainer*> resources are used to perform management of the Node.

This node specific information stored in these resources, <*mgmtObj*> specializations such as *[deviceInfo]* and *[firmware] or* <*flexContainer*> specializations such as [*dmDeviceInfo*] or [*dmFirmware*], can be obtained either by the existing device management technologies (OMA DM [i.10], BBF TR-069 [i.9]) or any other way (e.g. JNI [i.12]).

For the case when the <*node*> resource belongs to an ADN, please see figure 9.6.18-1 in conjunction with the description of *nodeLink* attribute in the <*AE*> resource (clause 9.6.5).

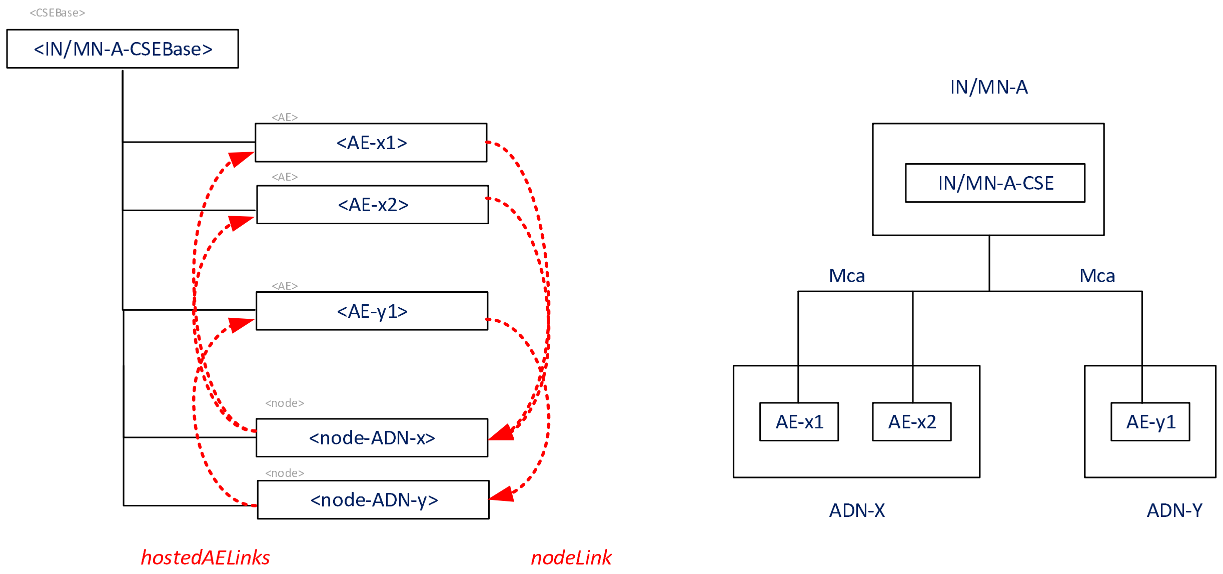

Figure 9.6.18-1: Relationship between IN/MN and ADN

For the case when the <*node*> resource belongs to an NoDN and the applications that correspond to interworked devices are represented by <*flexContainer*>*s* please see figure 9.6.18-2.

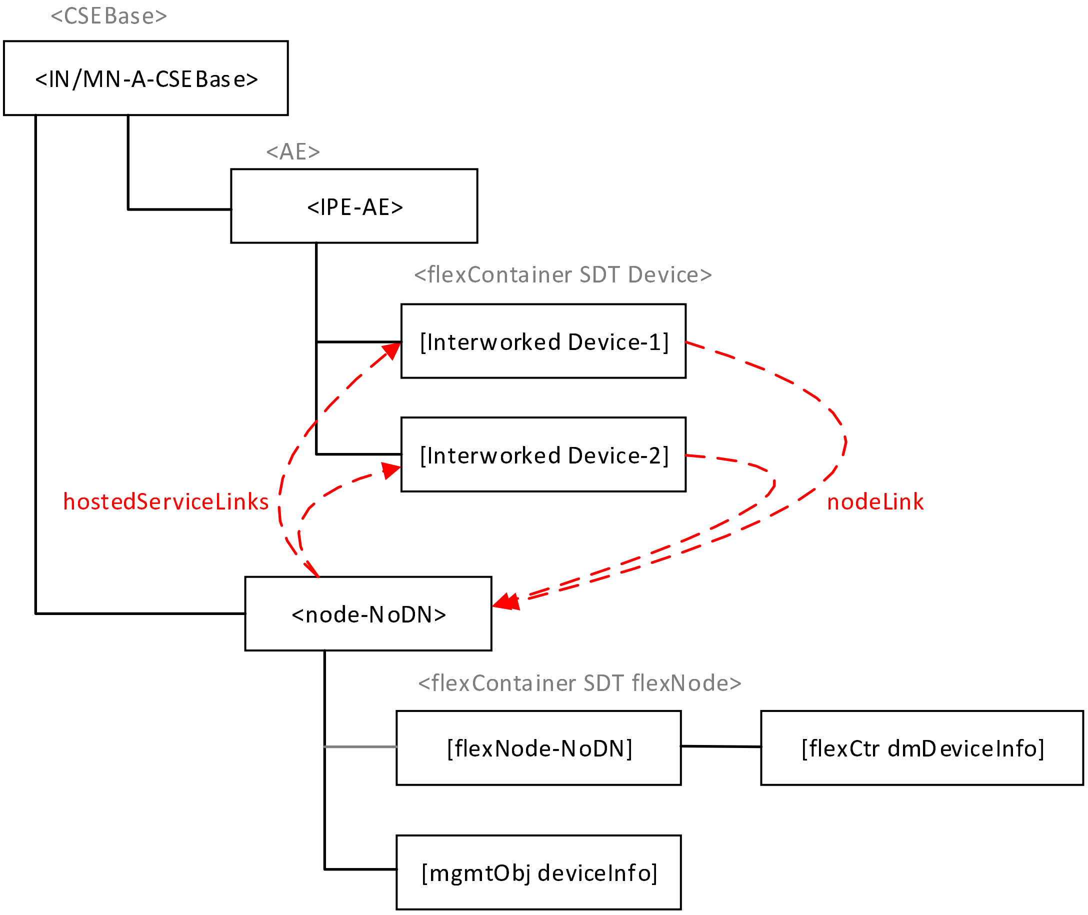

Figure 9.6.18-2: Relationship between IPE, interworked Services and NoDN

The *<node>* resource shall contain the child resources specified in table 9.6.18-1.

Table 9.6.18-1: Child resources of *<node>* resource

Child Resources of *<node>*	Child Resource Type	Multiplicity	Description	*<nodeAnnc>* Child Resource Type
[variable]	*<semanticDescriptor>*	0..n	See clause 9.6.30	*<semanticDescriptor>,* *<semanticDescriptorAnnc>*
[variable]	*<flexContainer> as defined in the specialization [flexNode]*	0..1	This resource provides the root for SDT-based <flexContainers> that correspond to Device Management related ModuleClasses (see ETSI TS 118 123 [i.2], clause 5.8). See note.	*<flexContainerAnnc>*
[variable]	*<mgmtObj>* as defined in the specialization [*memory]*	0..1	This resource provides the memory (typically RAM) information of the node. (E.g. the amount of total volatile memory), see clause D.4.	*<mgmtObjAnnc>*
[variable]	*<mgmtObj>* as defined in the specialization [*battery]*	0..n	The resource provides the power information of the node. (E.g. remaining battery charge). See clause D.7.	*<mgmtObjAnnc>*
[variable]	*<mgmtObj>* as defined in the specialization [*areaNwkInfo]*	0..n	This resource describes the list of Nodes attached behind the MN/ASN node and its physical or underlying relation among the nodes in the M2M Area Network. This attribute is defined in case the Node is MN/ASN. See clause D.5.	*<mgmtObjAnnc>*
[variable]	*<mgmtObj>* as defined in the specialization [*areaNwkDeviceInfo]*	0..n	This resource describes the information about the Node in the M2M Area Network. See clause D.6.	*<mgmtObjAnnc>*
[variable]	*<mgmtObj>* as defined in the specialization [*firmware]*	*0..n*	This resource describes the information about the firmware of the Node include name, version etc. See clause D.2.	*<mgmtObjAnnc>*
[variable]	*<mgmtObj>* as defined in the specialization [*software]*	0..n	This resource describes the information about the software of the Node. See clause D.3.	*<mgmtObjAnnc>*
[variable]	*<mgmtObj>* as defined in the specialization [*deviceInfo]*	0..n	The resource contains information about the identity, manufacturer and model number of the device. See clause D.8.	*<mgmtObjAnnc>*
[variable]	*<mgmtObj>* as defined in the specialization [*deviceCapability]*	0..n	The resource contains information about the capability supported by the Node. See clause D.9.	*<mgmtObjAnnc>*
[variable]	*<mgmtObj>* as defined in the specialization [*reboot]*	0..1	The resource is the place to reboot or reset the Node. See clause D.10.	*<mgmtObjAnnc>*
[variable]	*<mgmtObj>* as defined in the specialization [*eventLog]*	0..1	The resource contains the information about the log of events of the Node. See clause D.11.	*<mgmtObjAnnc>*
[variable]	*<mgmtObj>* as defined in the specialization *[cmdhPolicy]*	0..n	The resource(s) contain(s) information about CMDH policies that are applicable to the CMDH processing on the CSE hosted on the node represented by this <*node*> resource and identified by the *hostedCSELink* attribute of this <*node*> resource. See clause D.12.	NA

Child Resources of *<node>*	Child Resource Type	Multiplicity	Description	*<nodeAnnc>* Child Resource Type
[variable]	*<mgmtObj>* as defined in the specialization *[activeCmdhPolicy]*	0..1	This resource defines which of the present *[cmdhPolicy]* resource(s) shall be active for the CMDH processing on the CSE hosted on the node represented by this *<node>* resource and identified by the *hostedCSELink* attribute of this *<node>* resource. See clause D.12.	NA
[variable]	*<subscription>*	0..n	See clause 9.6.8.	<subscription>
[variable]	*<schedule>*	0..n	See clause 9.6.9.	<scheduleAnnc>
[variable]	*<transaction>*	0..n	See clause 9.6.48.	*<transaction>*
[variable]	*<action>*	0..n	See clause 9.6.61.	*None*
NOTE: if the [flexNode] child is present, then all <mgmtObj> children from [memory] to [eventLog] cannot be present.				

The *<node>* resource shall contain the attributes specified in table 9.6.18-2.

Table 9.6.18-2: Attributes of *<node>* resource

Attributes of *<node>*	Multiplicity	RW/ RO/ WO	Description	*<nodeAnnc>* attributes
resourceType	1	RO	See clause 9.6.1.3.	NA
resourceID	1	RO	See clause 9.6.1.3.	NA
resourceName	1	WO	See clause 9.6.1.3.	NA
parentID	1	RO	See clause 9.6.1.3.	NA
expirationTime	1	RW	See clause 9.6.1.3.	MA
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1.3.	MA
creationTime	1	RO	See clause 9.6.1.3.	NA
lastModifiedTime	1	RO	See clause 9.6.1.3.	NA
labels	0..1 (L)	RW	See clause 9.6.1.3.	MA
announceTo	0..1 (L)	RW	See clause 9.6.1.3.	NA
announcedAttribute	0..1 (L)	RW	See clause 9.6.1.3.	NA
announceSyncType	0..1	RW	See clause 9.6.1.3.	MA
dynamicAuthorizationConsultationIDs	0..1 (L)	RW	See clause 9.6.1.3.	OA
owner	0..1	RW	See clause 9.6.1.3.	NA
nodeID	1	RW	The M2M-Node-ID of the node which is represented by this *<node>* resource.	MA
nodeType	0..1	RW	Indicates the type of node. It shall have one of the following values: • IN • MN • ASN • ADN • NoDN	OA
hostedCSELink	0..1	RW	This attribute allows to find the <CSEBase> or <remoteCSE> resource representing the CSE that is residing on the node that is represented by this *<node>* resource. The attribute contains the resource ID of a resource where all of the following applies: • The resource is a *<CSEBase>* resource or a *<remoteCSE>* resource. • The resource represents the CSE which resides on the specific node that is represented by the current *<node>* resource. In case the node that is represented by this <node> resource does not contain a CSE, this attribute shall not be present.	OA
hostedAELinks	0..1(L)	RW	This attribute allows to find the AEs hosted by the node that is represented by	OA

Attributes of *<node>*	Multiplicity	RW/ RO/ WO	Description	*<nodeAnnc>* attributes
			this <*node*> resource. The attribute shall contain a list of resource identifiers of <*AE*> resources representing the ADN-AEs residing on the node that is represented by the current <*node*> resource. In case the node that is represented by this <node> resource does not contain an AE, this attribute shall not be present.	
hostedServiceLinks	0..1(L)	RW	This attribute allows to find <*flexContainer*> *resources that have* been created by an IPE to represent services hosted on a NoDN, the NoDN being represented by this <*node*> resource. If the NoDN hosts a set of services represented by <*flexContainer*>*s,* then the attribute shall contain the list of resource identifiers of these <*flexContainer*> resources. In case the node that is represented by this <*node*> resource does not contain an service that is represented by a <*flexContainer*>*,* this attribute shall not be present.	OA
mgmtClientAddress	0..1	RW	Represents the physical address of management client of the node which is represented by this <node> resource. This attribute is absent if management server is able to acquire the physical address of the management client.	OA
roamingStatus	0..1	RO	Indicates if the M2M Node is currently roaming from the perspective of the underlying network. The allowed values are "Yes" or "No".	OA
networkID	0..1	RO	Configured with the identity of the underlying network which the M2M Node is currently attached to.	OA

----------------------- End of change 2 ---

----------------------- Start of change 3 ---

10.2.8 Device management

10.2.8.1 Introduction

This clause describes the procedures for managing device capabilities on MNs (e.g. M2M Gateways), ASNs and ADNs (e.g. M2M Devices), as well as devices that reside within an M2M Area Network.

Resources maintaining information and relationships that are specific to Device Management are termed Device Management Resources. This clause details the creation, retrieval, update and deletion of the information associated with the following Device Management Resources: <node> (clauses 10.2.8.2 to 10.2.8.6), <mgmtObj> (clauses 10.2.8.7 to 10.2.8.12), <mgmtCmd> and its child resource <execInstance> (clauses 10.2.8.13 to 10.2.8.21), [flexNode] and its child <flexContainer> child resources (clauses 10.2.8.22 to 10.8.24).

These operations are used in both Device Management options available in oneM2M: one utilizing existing technology protocols (e.g. BBF TR-069 [i.9], OMA-DM [i.10] and LWM2M [i.11]) and another utilizing the native oneM2M protocols. Clause 6.2.4 details the Device Management (DM) CSF supporting this functionality.

10.2.8.2 Node management

This clause describes node management procedures over Mca and Mcc reference points, using the <*node*> resource which represents information about M2M Nodes that can be utilized in Device Management and other operations.

M2M Nodes represented by the <node> resource are: MN-CSE, ASN-CSE, ADN and NoDN. Zero, one or more <*node*> resources may be used to represent each M2M Node, as follows:

- A <*node*> resource representing a MN-CSE or a ASN-CSE is hosted by the represented CSE or the registrar CSE. The *hostedCSELink* attribute of the resource allows to find the <CSEBase> or <remoteCSE> resource representing the MN-CSE or ASN-CSE represented by the <node> resource. All <*node*> resources hosted on M2M Node's CSE may be announced to associated IN-CSEs.
- A <*node*> resource representing an ADN is hosted by the registrar CSE. The *hostedAELink* attribute of the resource allows to find the <AE> resources representing the AEs residing on the node ADN.
- A <*node*> resource representing a NoDN is hosted by a CSE with DM capabilities used to perform Device Management operations on the NoDN. If the NoDN is an interworked device, the *hostedServiceLink* attribute of the resource allows to find the <*flexContainer*> resources representing the services hosted on the NoDN.

An entity co-located with a CSE on an ASN or MN which is managed using oneM2M Device Management shall be represented by the same <*node*> resource.

Device Management resources associated with a M2M Node that is represented by a <node> resource shall be created, either as:

- <flexContainer> specializations children of a [flexNode] child of the <node>; or
- <mgmtObj> direct children of the <node>; or
- <mgmtCmd> and <execInstance> resources: the <execInstance> are created as children of the <node> resource(s) referenced in the <mgmtCmd>'s *execTarget* attribute.

---------------------- End of change 3 --

---------------------- Start of change 4 [NEW SECTION] --

10.2.8.22 Device management using [flexNode] and <flexContainer> resources

This clause describes procedures for managing device capabilities, using DM <flexContainer> specializations. This is an alternative to the approach based on <mgmtObj> or <mgmtCmd> resources specified in clauses 10.2.8.7 to 10.2.8.21.

This clause describes the management procedures over Mca and Mcc reference points. If technology specific protocols are used for management, different operations addressing a DM <*flexContainer*> resource (or its attributes or child resources) shall be translated by IN-CSE or MN-CSE into technology specific requests performed on the mapped technology specific data model object on the managed entity. In this case, the DM <*flexContainer*> resources are hosted on the IN-CSE or MN-CSE. Although management requests by the AE are agnostic to the technology specific protocol, the DM <*flexContainer*> resource exposes information about the technology specific protocol. AEs have the capability to retrieve this information within the *objectIDs* attribute of the DM <*flexContainer*> resource.

In the scenario where the DM <*flexContainer*> resource does not utilize an external management technology but instead uses the M2M Service Layer to perform the management request, the DM <*flexContainer*> resource is hosted on the CSE of the managed entity when the managed entity is an ASN, MN or IN. If the managed entity is an ADN node or the managed entity is co-located on an ASN, MN or IN, the DM <*flexContainer*> resource is hosted on the registrar CSE of the managed entity. The DM <*flexContainer*>, its parent [flexNode] and its grand-parent <*node*> resources hosted on node's CSE may be announced to associated IN-CSEs.

In the scenario where the managed entity is a NoDN, the managed entities' DM <*flexContainer*> resources are hosted by a CSE with DM capabilities used to perform Device Management operations on the NoDN.

The Node management, as described in clauses 10.2.8.2 to 10.2.8.6, is unchanged, but in this case the only child of the <node> resource will be a [flexNode] specialization.

10.2.8.23 flexNode management

10.2.8.23.1 Create [flexNode]

This procedure shall be used for creating a *[flexNode]* resource.

NOTE: The creation of the *[flexNode]* resource is on discretion of the Originator.

Table 10.2.8.23.1-1: *[flexNode]* CREATE

***[flexNode]* CREATE**	
Information in Request message	All parameters defined in table 8.1.2-3 apply with the specific details for: ***Content:*** The representation of the [flexNode] resource described in clause 5.8.2 in ETSI TS 118 123 [i.2]
Processing at Originator before sending Request	According to clause 10.1.2
Processing at Receiver	According to clause 10.1.2
Information in Response message	All parameters defined in table 8.1.3-1 apply with the specific details for: • ***Content*****:** Address of the created *[flexNode]* resource, according to clause 10.1.2
Processing at Originator after receiving Response	According to clause 10.1.2
Exceptions	According to clause 10.1.2

10.2.8.23.2 Retrieve [flexNode]

This procedure shall be used for retrieving the attributes of a *[flexNode]* resource.

Table 10.2.8.23.2-1: *[flexNode]* RETRIEVE

***[flexNode]* RETRIEVE**	
Information in Request message	All parameters defined in table 8.1.2-3 apply with the specific details for: ***Content*****:** Void
Processing at Originator before sending Request	According to clause 10.1.3
Processing at Receiver	According to clause 10.1.3
Information in Response message	All parameters defined in table 8.1.3-1 apply with the specific details for: ***Content*****:** Attributes of the *[flexNode]* resource as defined in clause 5.8.2 in ETSI TS 118 123 [i.2]
Processing at Originator after receiving Response	According to clause 10.1.3
Exceptions	According to clause 10.1.3

10.2.8.23.3 Update [flexNode]

This procedure shall be used for updating the attributes and the actual data of a *[flexNode]* resource and its child resources.

Table 10.2.8.23.3-1: *[flexNode]* UPDATE

[flexNode] **UPDATE**	
Information in Request message	All parameters defined in table 8.1.2-3 apply with the specific details for: ***Content***: attributes of the *[flexNode]* resource as defined in clause 5.8.2 in ETSI TS 118 123 [i.2] which need be updated, with the exception of the Read Only (RO) attributes cannot be modified
Processing at Originator before sending Request	According to clause 10.1.4
Processing at Receiver	According to clause 10.1.4 with the following: • The Receiver shall check whether the provided attributes of the [flexNode] resource represent a valid request for updating *[flexNode]* resource
Information in Response message	According to clause 10.1.4
Processing at Originator after receiving Response	According to clause 10.1.4
Exceptions	According to clause 10.1.4

10.2.8.23.4 Delete [flexNode]

This procedure shall be used for deleting an existing *[flexNode]* resource.

Table 10.2.8.23.4-1: *[flexNode]* DELETE

[flexNode] **DELETE**	
Information in Request message	All parameters defined in table 8.1.2-3 apply
Processing at Originator before sending Request	According to clause 10.1.5
Processing at Receiver	According to clause 10.1.5
Information in Response message	According to clause 10.1.5
Processing at Originator after receiving Response	According to clause 10.1.5
Exceptions	According to clause 10.1.5

10.2.8.24 DM <flexContainer> management

10.2.8.24.1 Introduction

DM <flexContainer> resources are <flexContainer> specializations that correspond to Smart Device Templates ModuleClasses specified in ETSI TS 118 123 [i.2], clause 5.8, that extend the abstract ModuleClass [dmBaseModule].

They are defined with two optional *custom attributes*, objectIDs and objectPaths, that have the same role as the corresponding *object attributes* in <mgmtObj> resources.

10.2.8.24.2 Create DM <*flexContainer*>

This procedure shall be used to create a specific *DM <flexContainer>* resource in the Hosting CSE to expose the corresponding management function of a managed entity (i.e. M2M Device/Gateway) over the Mca reference point. Depending on the data model being used, the created *DM <flexContainer>* resource may be a partial or complete mapping from the technology specific data model object on the managed entity. If such a technology specific data model object is missing from the managed entity, it shall be added to the managed entity. Further operations performed on the created *DM <flexContainer>* resource shall be converted by the Hosting CSE into a corresponding technology specific request performed on the mapped technology specific data model object on the managed entity using technology specific protocol (e.g. OMA-DM [i.10] or BBF TR-069 [i.9]).

Besides the generic create procedure defined in clause 10.1.2, the procedure in table 10.2.8.24.2-1 shall be used when management is performed using technology specific protocols.

If the management is performed by service layer entities, the procedure is the same as generic create procedure defined in clause 10.1.2. In this case, local APIs (drivers) on the managed entity is required to monitor the change of the DM <flexContainer> resource and reflect the change to the managed entity.

Table 10.2.8.24.2-1: *DM <flexContainer>* CREATE

***DM <flexContainer>* CREATE**	
Information in Request message	***From:*** Identifier of the AE or the CSE that initiates the Request ***To:*** The address of the *[flexNode]* where the *DM <flexContainer>* resource is intended to be Created ***Content:*** The representation of the *DM <flexContainer>* resource.
Processing at Originator before sending Request	The Originator shall be an AE, or a CSE: • The Originator is a CSE: In this case, the CSE first collects the original technology specific data model object (the management tree structure or also the value of the tree nodes if needed) of the local device and transforms the object into the *DM <flexContainer>* resource representation, then requests the Hosting CSE to create the corresponding *DM <flexContainer>* resource. • The Originator is an AE: In this case, the AE requests the Hosting CSE to add the corresponding technology specific data model object to the managed entity by creating an DM <flexContainer> resource in the Hosting CSE (See notes 1 and 2)
Processing at Receiver	For the CREATE operation, besides the common create operation defined in clause 10.1.2, the Receiver shall: • If the Originator is an AE: Check if there is existing management session between the management server and the managed entity. If not, request the management server to establish a management session towards the managed entity. Send the technology specific request to the managed entity or to the management server to add the corresponding technology specific data model object to the managed entity based on technology specific protocol • Maintain the mapping relationship between the created *DM <flexContainer>* resource and the technology specific data model object on the managed entity • Respond to the Originator with the appropriate responses based on the technology specific response. It shall also provide in the response the address of the created new resource
Information in Response message	Error code if the new technology specific data model object is not created
Processing at Originator after receiving Response	None
Exceptions	• The creation of the technology specific data model object is not allowed • The created technology specific data model object already exists • Corresponding technology specific data model object cannot be added to the managed entity for some reason (e.g. not reachable, memory shortage)
NOTE 1: The CSE can create the *DM <flexContainer>* resource locally by itself. The details are out of scope. In this case, the Hosting CSE first collects the original technology specific data model object on the managed entity via technology specific protocol (e.g. OMA DM [i.10], BBF TR-069 [i.9] or LWM2M [i.11]), then transforms the object into the *DM <flexContainer>* resource representation and create the *DM <flexContainer>* resource locally in the CSE. NOTE 2: The *DM <flexContainer>* resource can be created in the Hosting CSE by other offline provisioning means which are out of scope.	

10.2.8.24.3 Retrieve *DM <flexContainer>*

This procedure shall be used to retrieve information from an existing *DM <flexContainer>* resource. Besides the generic retrieve procedure defined in clause 10.1.3, the procedure in table 10.2.8.24.3-1 shall be used when management is performed using technology specific protocols. If the management is performed by service layer entities, the procedure is the same as generic retrieve procedure defined in clause 10.1.3.

Table 10.2.8.24.3-1: *DM <flexContainer>* RETRIEVE

***DM <flexContainer>* RETRIEVE**	
Information in Request message	***From:*** Identifier of the AE or the CSE that initiates the Request ***To:*** The address of the *DM <flexContainer>* resource
Processing at Originator before sending Request	None
Processing at Receiver	For the RETRIEVE operation, besides the common retrieve operation defined in clause 10.1.3, the Receiver shall: • If the Originator is an AE and if the requested information of the *DM <flexContainer>* resource is not available, identify the corresponding technology specific data object on the managed entity according to the mapping relationship that the CSE maintains. Check if there is an existing management session between the management server and the managed entity. If not, request the management server to establish a management session towards the managed entity. Send the technology specific request to get the corresponding technology specific data model object from the managed entity based on the external management technology, then return the result to the Originator based on the technology specific response
Information in Response message	Error code if the new technology specific data model object cannot be retrieved
Processing at Originator after receiving Response	None
Exceptions	• Corresponding technology specific data model object data cannot be retrieved from the managed entity (e.g. technology specific data model object not found)

10.2.8.24.4 Update DM <flexContainer>

This procedure shall be used to update information of an existing *DM <flexContainer>* resource. Besides the generic update procedure defined in clause 10.1.4, the procedure in table 10.2.8.24.4-1 shall be used when management is performed using technology specific protocol. If the management is performed by service layer entities, the procedure is the same as generic update procedure defined in clause 10.1.4. In this case, local APIs (drivers) on the managed entity is required to monitor the change of the DM <flexContainer> resource and reflect the change to the managed entity.

Table 10.2.8.24.4-1: *DM <flexContainer>* UPDATE

***DM <flexContainer>* UPDATE**	
Information in Request message	***From:*** Identifier of the AE or the CSE that initiates the Request ***To:*** The address of the *DM <flexContainer>* resource ***Content:*** The representation of the *DM <flexContainer>* resource.
Processing at Originator before sending Request	None
Processing at Receiver	For the UPDATE operation, besides the common update operation defined in clause 10.1.4, the Receiver shall: • If the Originator is an AE, identify the corresponding technology specific data model object on the managed entity according to the mapping relationship it maintains. Check if there is an existing management session between the management server and the managed entity. If not, request the management server to establish a management session towards the managed entity. Send the technology specific request to update the corresponding technology specific data model object in the managed entity accordingly based on technology specific protocol • Respond to the Originator with the appropriate response based on the technology specific response from the external management technology
Information in Response message	Error code if the technology specific data model object cannot be updated
Processing at Originator after receiving Response	None
Exceptions	• Corresponding technology specific data model object cannot be updated to managed entity (e.g. not reachable, technology specific data model object not found)

10.2.8.24.5 Delete DM <flexContainer>

This procedure shall be used to delete an existing *DM <flexContainer>* resource. An Originator uses this procedure to remove the corresponding technology specific data model object (e.g. an obsolete software package) from the managed entity. Besides the generic delete procedure defined in clause 10.1.5, the procedure in table 10.2.8.24.5-1 shall be used when management is performed using external management technologies. If the management is performed by service layer entities, the procedure is the same as generic delete procedure defined in clause 10.1.5. In this case, local APIs (drivers) on the managed entity is required to monitor the change of the DM <flexContainer> resource and reflect the change to the managed entity.

Table 10.2.8.24.5-1: *DM <flexContainer>* DELETE

***DM <flexContainer>* DELETE**	
Information in Request message	***From:*** Identifier of the AE, or the CSE that initiates the Request ***To:*** The address of the *DM <flexContainer>* resource
Processing at Originator before sending Request	The Originator shall be an AE or CSE: • The Originator is a CSE: In this case, the CSE issues the request to the Hosting CSE to hide the corresponding management function from being exposed by the *DM <flexContainer>* resource • The Originator is an AE: In this case, the AE requests the Hosting CSE to delete the *DM <flexContainer>* resource from the Hosting CSE and to remove the corresponding technology specific data model object from the managed entity (See notes 1 and 2)
Processing at Receiver	For the DELETE operation, besides the common create operation defined in clause 10.1.5, the Receiver shall: • If the Originator is an AE, identify the corresponding technology specific data model object on the managed entity according to the mapping relationship the CSE maintains. Check if there is an existing management session between the management server and the managed entity. If not, request the management server to establish a management session towards the managed entity. The CSE sends technology specific request to remove the corresponding technology specific data model object from the managed entity based on technology specific protocol • Respond to the Originator with the appropriate generic responses based on the technology specific response
Information in Response message	Error code if the technology specific data model object cannot be deleted
Processing at Originator after receiving Response	None
Exceptions	• Corresponding technology specific data model object cannot be deleted from managed entity (e.g. not reachable, technology specific data model object not found)
NOTE 1: The Hosting IN-CSE can delete the *DM <flexContainer>* resource locally by itself. This internal procedure is out of scope. NOTE 2: The *DM <flexContainer>* resource can be deleted in the Hosting CSE by offline provisioning means which are out of scope.	

----------------------- End of change 4 ---

----------------------- Start of change 5 ---

D.12 Resource *cmdhPolicy*

*TBD: Write DM <flexContainers> that correspond to the 8 [*cmdh*] <mgmtObj>.*

Waiting for new version of CMDH processing through flexContainers (WI-0096 [i.8]).

----------------------- End of change ---

Annex B: Proposal for update of ETSI TS 118 104

B.1 Introduction

In this annex, are presented the proposed changes to the ETSI TS 118 104 [i.4] specification for flexContainer introduction for device management operations.

---------------------- Start of change 1 --

7.3.4 Management common operations

7.3.4.0 Introduction

This clause describes common operations on Device Management resources that are associated with a M2M Node that is represented by a <node> resource. These resources are:

- either DM <flexContainer> specializations children of a [flexNode] child of the <node>;
- or <mgmtObj> direct children of the <node>;
- or <mgmtCmd> and <execInstance> resources: the <execInstance> are created as children of the <node> resource(s) referenced in the <mgmtCmd>'s *execTarget* attribute.

7.3.4.1 Identify the managed entity and the technology specific protocol

Where a managed entity is being addressed via a <mgmtObj> or DM <flexContainer> resource, the Hosting CSE shall identify the managed entity via the <node> resource that is the parent resource of the <mgmtObj>, or the parent of the [flexNode] parent of the DM <flexContainer> resource. In case of a <mgmtCmd> resource the entity to be managed is indicated by its *execTarget* attribute. This addresses either a <node> resource or a group of resources of type <node>. Hence, in all cases the managed entity is ultimately identified through a <node> resource, from which the identifier of the device can be retrieved.

The Hosting CSE shall determine the technology specific protocol to be used for communicating with the managed entity based on the *objectIDs* attribute of the addressed <mgmtObj> or DM <flexContainer> resource.

If the managed entity cannot be identified, the Hosting CSE shall reject the request with the ***Response Status Code*** indicating "EXTERNAL_OBJECT_NOT_REACHABLE" in the Response primitive.

7.3.4.2 Locate the technology specific data model objects to be managed on the managed entity

The Hosting CSE shall locate the technology specific data model object to be managed on the managed entity by the *objectPaths* attribute of the <mgmtObj> or DM <flexContainer> resource addressed by the URI provided in the ***To*** primitive parameter. In the case that the ***To*** addresses an [objectAttribute] (resp. [customAttribute]), the Hosting CSE shall locate the technology specific data model object on the managed entity through the *objectPaths* attribute of the <mgmtObj> (resp. DM <flexContainer>) resource of the addressed [objectAttribute] / [customAttribute], combined with their relative position in the technology specific data model object tree. If the technology specific data model object cannot be located, the Hosting CSE shall reject the request with the ***Response Status Code*** indicating "EXTERNAL_OBJECT_NOT_FOUND" in the Response primitive.

In the case that the management server is external to the Hosting CSE, the Hosting CSE shall identify the management server that is capable of performing the operation on the technology specific data model object. If the management server cannot be identified, the Hosting CSE shall reject the request with the ***Response Status Code*** indicating "EXTERNAL_OBJECT_NOT_REACHABLE" in the Response primitive.

---------------------- End of change 1 --

---------------------- Start of change 2 ---

7.3.4.4 Send the management request(s) to the managed entity corresponding to the received Request primitive

The Hosting CSE shall send the management request(s) to the managed entity or management server in the established management session in order to perform the management operation as requested by the received Request primitive. The management request shall address the technology-specific data model object on the managed entity as determined in clause 7.3.4 or in the primitive-specific clauses. The management request being used is specific to the technology specific protocol according to a pre-defined mapping relationship with the Request primitive. The internal data structure of the technology specific data model object addressed by the technology specific request shall be determined based on the mapping relationship of the <mgmtObj>, DM <flexContainer> or <mgmtCmd> resources and the technology specific data model objects or based on the generic mapping rule as specified in ETSI TS 118 101 [i.3], clauses 9.6.15, 9.6.16, 9.6.17 and 9.6.35. The Hosting CSE shall extract the management results received from the managed entity or management server in order to prepare a Response primitive to be sent to the originator later. Unless explicitly stated, if the management request cannot be performed successfully, the Hosting CSE shall reject the Request primitive with the management server in the Response primitive according to the mapping relationship with the technology specific protocol.

---------------------- End of change 2 ---

---------------------- Start of change 3 ---

7.4.18 Resource Type <node>

7.4.18.1 Introduction

The <node> resource represents specific information that provides properties of an oneM2M Node that can be utilized by other oneM2M operations. The <node> resource has either a [flexNode] <flexContainer> specialization or <mgmtObj> as its child resources.

Table 7.4.18.1-1: Data type definition of <node> resource

Data Type ID	File Name	Note
node	CDT-node-v4_2_0.xsd	

Table 7.4.18.1-2: Universal/Common Attributes of <node> resource

Attribute Name	Request Optionality	
	Create	Update
@resourceName	O	NP
resourceType	NP	NP
resourceID	NP	NP
parentID	NP	NP
accessControlPolicyIDs	O	O
creationTime	NP	NP
expirationTime	O	O
lastModifiedTime	NP	NP
labels	O	O
announceTo	O	O
announcedAttribute	O	O
dynamicAuthorizationConsultationIDs	O	O

Table 7.4.18.1-3: Resource Specific Attributes of <node> resource

Attribute Name	Request Optionality		Data Type	Default Value and Constraints
	Create	Update		
nodeID	M	O	m2m:nodeID	
hostedCSELink	O	O	m2m:ID	
hostedAELinks	O	O	m2m:listOfM2MID	
hostedServiceLinks	O	O	m2m:listOfM2MID	
mgmtClientAddress	O	O	xs:string	
roamingStatus	NP	NP	xs:boolean	No default. True means that the Node is currently roaming. When this attribute is not present, it indicates that no information is available.
networkID	NP	NP	xs:string	No default. When this attribute is not present, it indicates that no information is available.
nodeType	O	O	m2m:nodeType	Default is UNSPECIFIED.

Table 7.4.18.1-4: Child resources of <node> resource

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<mgmtObj>	[variable]	0..n	Clause 7.4.15 and annex D
<flexContainer>	[flexNode]	0..1	Clause 5.8.2 in ETSI TS 118 123 [i.2]
<subscription>	[variable]	0..n	Clause 7.4.8
<semanticDescriptor>	[variable]	0..n	Clause 7.4.34
<transaction>	[variable]	0..n	Clause 7.4.61
<schedule>	[variable]	0..n	Clause 7.4.9

---------------------- End of change 3 --

Annex C: Proposal for update of ETSI TS 118 123

C.1 Introduction

In this annex, are presented the proposed changes to the ETSI TS 118 123 [i.2] specification for flexContainer introduction for device management operations.

NOTE: Clause 5.8 of ETSI TS 118 123 [i.2] specifies the [flexNode] specialization and its DM-related child <flexContainer> resources.

---------------------- Start of change 1 ---

5.8.2 flexNode

This flexContainer specialization is the root for SDT-based Device Management modules.

The containerDefinition attribute of this specialization shall be "org.onem2m.management.device.flexNode".

It is a <flexContainer> child of the <node> resource targeted by the *nodeLink* attribute of *<flexContainer>* SDT devices (see in clause 6.2.2 the rules 1-7).

Table 5.8.2-1: Child resources of [*flexNode*] resource

Child Resources of [*flexNode*]	Child Resource Type	Multiplicity	Description
dmAreaNwkInfo_<i>	*[dmAreaNwkInfo]*	0..n	See clause 5.8.10
dmAgent	*[dmAgent]*	0..1	See clause 5.8.3
dmDeviceInfo	*[dmDeviceInfo]*	1	See clause 5.8.4
dmDataModelIO_<i>	*[dmDataModelIO]*	0..N	See clause 5.8.5
dmFirmware_<i>	*[dmFirmware]*	1..N	See clause 5.8.6
dmSoftware_<i>	*[dmSoftware]*	0..N	See clause 5.8.7
dmEventLog_<i>	*[dmEventLog]*	0..N	See clause 5.8.8
dmPackage_<i>	*[dmPackage]*	0..N	See clause 5.8.9
battery_<i>	*[battery]*	0..N	See clause 5.3.10
dmCapability_<i>	*[dmCapability]*	0..N	See clause 5.8.12
dmStorage_<i>	*[dmStorage]*	0..N	See clause 5.8.13

NOTES:

- The notation '_<i>' for child resources indicates that the resource name is the name of the child ModuleClass or SubDevice flexContainer, appended with an underscore '_' and an incrementing index so that it is unique in the [flexNode] children (e.g. "dmFirmware_0", "dmFirmware_1", etc.). The index shall not have leading 0's.
- The current list of modules for Device Management is not fixed and can evolve with new optional features.

All children of the [flexNode] are modules that extend the [dmBaseModule] <flexContainer> specialization defined in table 5.8.2-2.

Table 5.8.2-2: DataPoints of dmBaseModule ModuleClass

Name	Type	R/W	Optional	Unit	Description
flexSchema	xs:anyURI	R	true		Contains a URI to the <*flexContainer*> schema definition which shall be used by the Hosting CSE to validate the syntax of incoming primitives targeting this <*flexContainer*> resource. This URI may refer to a oneM2M specified <*flexContainer*> definition as well as other <*flexContainer*> definitions.
objectIDs	xs:string	R	true		Contains the list URNs that uniquely identify the technology specific data model objects used for this <*flexContainer*> resource as well as the managed function and version it represents. This attribute shall be provided during the creation of the <*flexContainer*> resource and shall not be modifiable afterwards. If the <*flexContainer*> resource is mapped to multiple technology specific data model objects, this attribute shall list all URNs for each mapped technology specific data model objects. This is mandatory for the <*flexContainer*>, for which the data model is not specified by oneM2M but mapped from technology specific data model.
objectPaths	xs:string	R	true		Contains the list of local paths of the technology specific data model objects on the managed entity which is represented by the <*flexContainer*> resource in the Hosting CSE. This attribute shall be provided during the creation of the <*flexContainer*>, so that the Hosting CSE can correlate the created <*flexContainer*> with the technology specific data model object on the managed entity for further management operations. It shall not be modifiable after creation. The format of this attribute shall be a local technology specific data model object path in the form as specified by technology specific protocol. (e.g. "./anyPath/Fw1" in OMA DM [i.10], "Device.USBHosts.Host.3." in BBF TR-069 [i.9]). The combination of the *objectPaths* and the *objectIDs* attribute, allows to address the technology specific data model.
objectDesc	xs:string	R	true		Text format description of the resource.

The SDT modules that extend this [dmBaseModule] are mapped as specific oneM2M <flexContainer> resources that are called *DM <flexContainer>*.

The *DM <flexContainer>* resource contains management data which represents individual M2M management functions. It represents a general structure to map to technology specific data model, e.g. OMA DM [i.10], BBF TR-069 [i.9] or LWM2M [i.11]. Each instance of *DM <flexContainer>* resource shall be mapped to single technology specific protocol.

When mapping objects from technology specific protocol to a corresponding *DM <flexContainer>* resource, the following rules shall apply:

- The root object of technology specific data model objects maps to the *DM <flexContainer>* resource.
- For the child of the root of technology specific data model objects:
    - **Rule1:** If the child technology specific data model object cannot have another child technology specific data model object, the technology specific data model object maps to the *[customAttribute]* attribute of the *DM <flexContainer>* resource with the same resource name.
    - **Rule2:** If the child technology specific data model object can have another child technology specific data model object, the technology specific data model object maps to a new *DM <flexContainer>* resource that is a child of the *DM <flexContainer>* resource which is mapped from the parent technology specific data model object.

---------------------- End of change 1 --

---------------------- Start of change 2 --

6.2.2 Resource mapping for Device model

When the AE exposes a controlling interface for a home domain device which is specified as an information model in clause 5.5, a specialization of the <flexContainer> resource shall be created as the mapping of the model following conversion rules:

- Rule 1-1: Each Device model defined in clause 5.5 shall be mapped to a specialization of <flexContainer>. The *containerDefinition* attribute shall be set according to clause 6.4.2.

- Rule 1-2: Each entry in the 'Module' table shall be mapped to a child resource(s) which is mapped as a specialized <flexContainer> following the rule in clause 6.2.3.

- Rule 1-3: The specialized <flexContainer> resource of the Device model shall contain an attribute *nodeLink* (as defined in ETSI TS 118 101 [i.3] and in ETSI TS 118 104 [i.4]). The value of *nodeLink* shall be set to the resource identifier of a <node> resource described in Rule 1-5 below.

- Rule 1-4: XSD file for each Device model shall be named according to clause 6.5.2.

- Rule 1-5: A <node> resource shall be created on the same hosting CSE as the <flexContainer> representing this Device model. If the <node> resource does not contain a [flexNode] child resource (see Rule 1.7), then it contains all the management information as specialized <mgmtObj> resources (e.g. [firmware]) about the Device model instance for device management purposes.

- Void.

- Rule 1-7: The <node> resource targeted by the nodeLink attribute may contain a [flexNode] specialization of a <flexContainer> resource. This [flexNode] resource contains all the Device Management information as specialized <flexContainer> resources defined in clause 5.8 (e.g. [dmFirmware]) about the device model instance for Device Management purposes.

- Void.

- Rule 1-9: Each entry in the 'SubDevice' table shall be mapped to a child resource(s) which is mapped as a specialized <flexContainer> following the rule in clause 6.2.7.

---------------------- End of change 2 --

---------------------- Start of change 3 --

5.8.3 dmAgent

Extends: dmBaseModule

<…>

5.8.4 dmDeviceInfo

Extends: dmBaseModule

<…>

5.8.5 dmDataModelIO

Extends: dmBaseModule

<…>

5.8.6 dmFirmware

Extends: dmBaseModule

<…>

5.8.7 dmSoftware

Extends: dmBaseModule

<…>

5.8.8 dmEventLog

Extends: dmBaseModule

<…>

5.8.9 dmPackage

Extends: dmBaseModule

<…>

5.8.10 dmAreaNwkInfo

Extends: dmBaseModule

<…>

5.8.11 dmAreaNwkDeviceInfo

Extends: dmBaseModule

<…>

5.8.12 dmCapability

Extends: dmBaseModule

<…>

5.8.13 dmStorage

Extends: dmBaseModule

<…>

---------------------- End of change 3 --

---------------------- Start of change 4 --

6.2.2 Resource mapping for Device model

When the AE exposes a controlling interface for a home domain device which is specified as an information model in clause 5.5, a specialization of the <flexContainer> resource shall be created as the mapping of the model following conversion rules:

- Rule 1-1: Each Device model defined in clause 5.5 shall be mapped to a specialization of <flexContainer>. The *containerDefinition* attribute shall be set according to clause 6.4.2.
- Rule 1-2: Each entry in the 'Module' table shall be mapped to a child resource(s) which is mapped as a specialized <flexContainer> following the rule in clause 6.2.3.

- Rule 1-3: The specialized <flexContainer> resource of the Device model shall contain an attribute *nodeLink* (as defined in ETSI TS 118 101 [i.3] and in ETSI TS 118 104 [i.4]). The value of *nodeLink* shall be set to the resource identifier of a <node> resource described in Rule 1-5 below.

- Rule 1-4: XSD file for each Device model shall be named according to clause 6.5.2.

- Rule 1-5: A <node> resource shall be created on the same hosting CSE as the <flexContainer> representing this Device model. If the <node> resource does not contain a [flexNode] child resource (see Rule 1.7), then it contains all the management information as specialized <mgmtObj> resources (e.g. [firmware]) about the Device model instance for device management purposes.

- Rule 1-6: Void.

- Rule 1-7: The <node> resource targeted by the nodeLink attribute may contain a [flexNode] specialization of a <flexContainer> resource. This [flexNode] resource contains all the Device Management information as specialized <flexContainer> resources defined in clause 5.8 (e.g. [dmFirmware]) about the device model instance for Device Management purposes.

- Rule 1-8: Void.

- Rule 1-9: Each entry in the 'SubDevice' table shall be mapped to a child resource(s) which is mapped as a specialized <flexContainer> following the rule in clause 6.2.7.

---------------------- End of change 4 --

---------------------- Start of change 5 --

6.2.5 Resource mapping for Property

When the Device model (in clause 5.5) or the ModuleClass model (in clause 5.3) is mapped to the <flexContainer> resource, and if the device supports a Property, the following rules shall be applied:

- Rule 4-1: Each entry of 'Property' table in ModuleClass model, shall be mapped to the [customAttribute] of <flexContainer> resource which is mapped from associated ModuleClass model, with its Property name with prefix 'prop'.

- Rule 4-2: If the <node> resource targeted by the *nodeLink* attribute of a Device model does not contain a [flexNode] child, then each 'Property' of the Device model is mapped to a specialized [objectAttribute] of a [deviceInfo] <mgmtObj> resource child of this <node>, otherwise it is mapped to a [customAttribute] of a [dmDeviceInfo] <flexContainer> resource child of this [flexNode].

- Rule 4-3: Each entry of 'Property' table in SubDevice model, shall be mapped to the [customAttribute] of <flexContainer> resource which is mapped from associated SubDevice model, with its Property name with prefix 'prop'.

---------------------- End of change 5 --

---------------------- Start of change 6 --

A.2 Example for Device model 'deviceAirConditioner'

The present clause explains the creation process for the device typed 'deviceAirConditioner' (see clause 5.5.1.1 for device model definition of 'deviceAirConditioner').

Using the definition, 'deviceAirConditioner' model is mapped to [deviceAirConditioner] resource which is a specialization of <flexContainer> resource (see figure A.2-1).

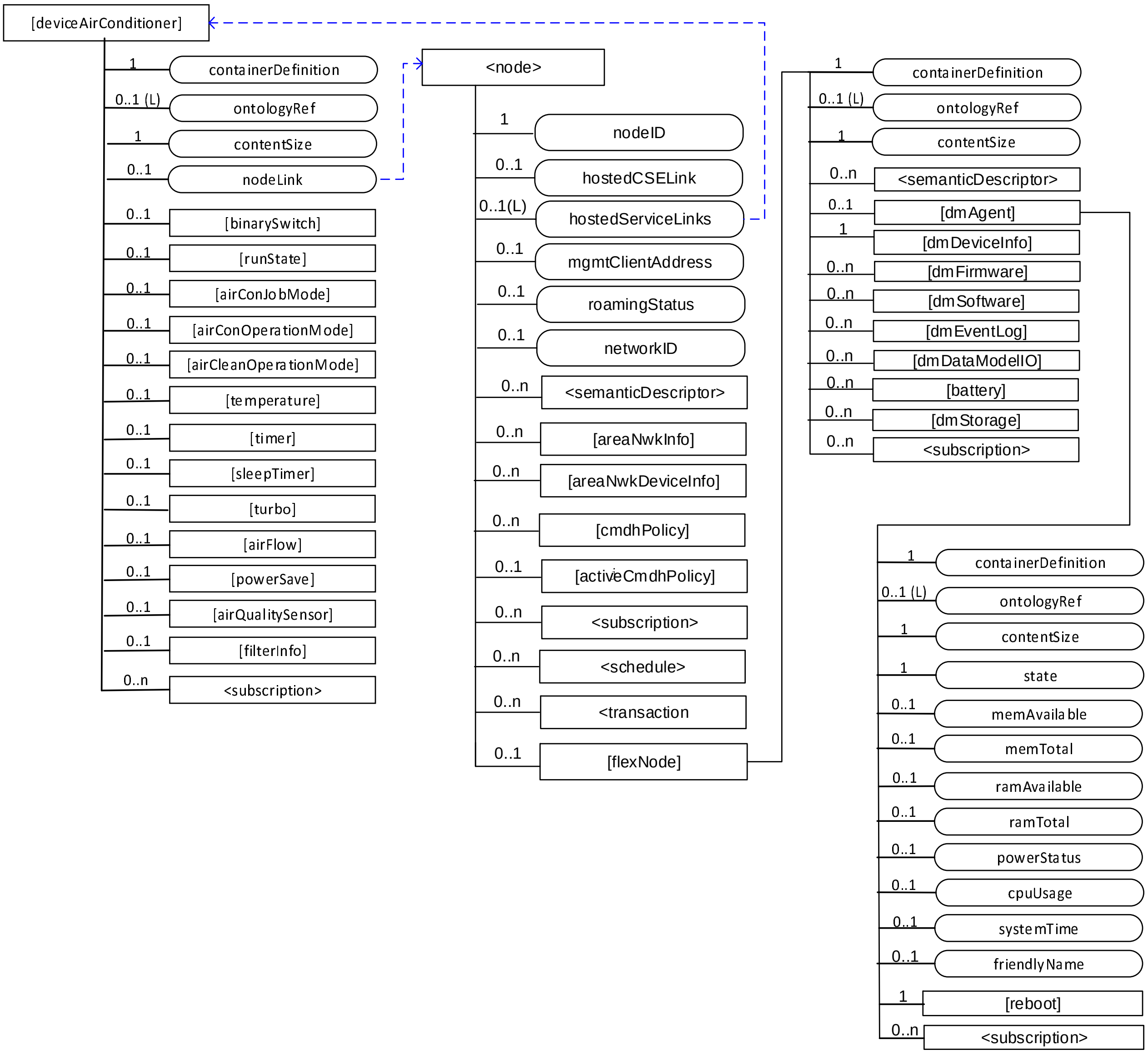

Figure A.2-1: Structure of [deviceAirConditioner] resource

The AE creates the [deviceAirConditioner] specialization of <flexContainer> resource for the Device model [deviceAirConditioner] resource.

The [deviceAirConditioner] resource contains the child resource specified in table A.2-2.

Table A.2-2: Child resources of *[deviceAirConditioner]* resource

Child Resources of *[deviceAirConditioner]*	Child Resource Type	Multiplicity	Description
[variable]	*<flexContainer> as defined in the specialization [binarySwitch]*	0..1	This resource is used to map 'binarySwith' ModuleClass defined in clause 5.3.1.12.
[variable]	*<flexContainer> as defined in the specialization [runState]*	0..1	This resource is used to map 'runState' ModuleClass defined in clause 5.3.1.75.
[variable]	*<flexContainer> as defined in the specialization [airConJobMode]*	0..1	This resource is used to map 'airConJobMode' ModuleClass defined in clause.
[variable]	*<flexContainer> as defined in the specialization [airConOperationMode]*	0..1	This resource is used to map 'airConOperationMode' ModuleClass defined in clause 5.3.1.57.
[variable]	*<flexContainer> as defined in the specialization [airCleanOperationMode]*	0..1	This resource is used to map 'airCleanOperationMode' ModuleClass defined in clause 5.3.1.57.
[variable]	*<flexContainer> as defined in the specialization [temperature]*	0..1	This resource is used to map 'temperature' ModuleClass defined in clause 5.3.1.87.
[variable]	*<flexContainer> as defined in the specialization [timer]*	0..1	This resource is used to map 'timer' ModuleClass defined in clause 5.3.1.90.
[variable]	*<flexContainer> as defined in the specialization [sleepTimer]*	0..1	This resource is used to map 'sleepTimer' ModuleClass defined in clause 5.3.1.90.
[variable]	*<flexContainer> as defined in the specialization [turbo]*	0..1	This resource is used to map 'turbo' ModuleClass defined in clause 5.3.1.91.
[variable]	*<flexContainer> as defined in the specialization [airFlow]*	0..1	This resource is used to map 'airFlow' ModuleClass defined in clause 5.3.1.4.
[variable]	*<flexContainer> as defined in the specialization [powerSave]*	0..1	This resource is used to map 'powerSave' ModuleClass defined in clause 5.3.1.66.
[variable]	*<flexContainer> as defined in the specialization [airQualitySensor]*	0..1	This resource is used to map 'airQualitySensor' ModuleClass defined in clause 5.3.1.6.
[variable]	*<flexContainer> as defined in the specialization [filterInfo]*	0..1	This resource is used to map 'filterInfo' ModuleClass defined in clause 5.3.1.35.
[variable]	*<subscription>*	0..n	See clause 9.6.8 in ETSI TS 118 101 [i.3].

The [deviceAirConditioner] resource contains the attributes specified in table A.2-3.

Table A.2-3: Attributes of *[deviceAirConditioner]* resource

Attributes of *[deviceAirConditioner]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
resourceID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
resourceName	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
parentID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
expirationTime	1	RW	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
creationTime	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
lastModifiedTime	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
labels	0..1	RW	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
dynamicAuthorizationConsultationIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
stateTag	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
creator	0..1	RW	See clause 9.6.35 in ETSI TS 118 101 [i.3]
containerDefinition	1	WO	The value is "org.onem2m.home.device.airconditioner"
ontologyRef	0..1	RW	See clause 9.6.35 in ETSI TS 118 101 [i.3]
contentSize	1	RO	See clause 9.6.35 in ETSI TS 118 101 [i.3]
nodeLink	1	RO	nodeLink attribute links to a <node> resource that is hosted on the same hosting CSE of the <flexContainer>. See clauses 6.2.2 and 6.2.5 for more details

---------------------- End of change 6 --

---------------------- Start of change 7 --

A.3 Example of ModuleClass 'binarySwitch'

The [*binarySwitch*] resource is used to share information regarding the modeled binary switch module as a ModuleClass. The [*binarySwitch*] resource is a specialization of the <*flexContainer*> resource.

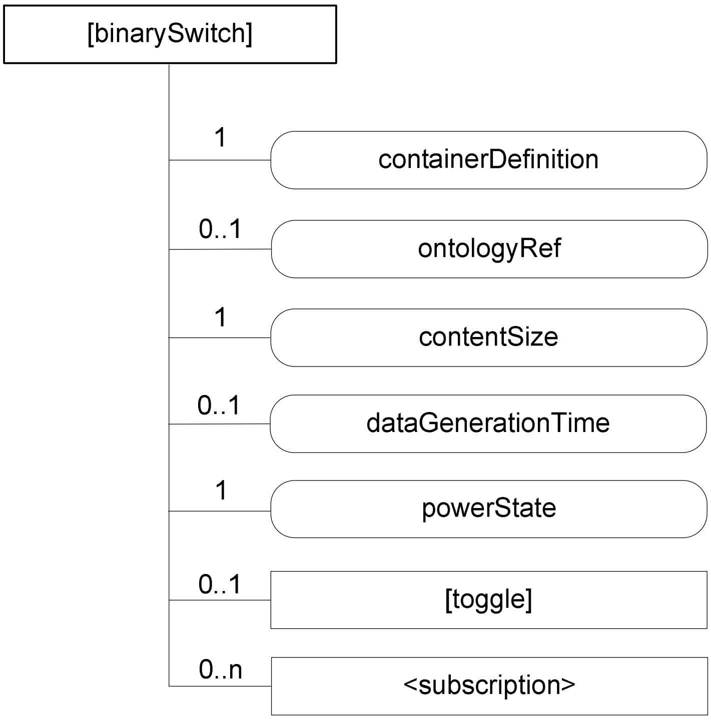

Figure A.3-1: Structure of *[binarySwitch]* resource

The *[binarySwitch]* resource contains the child resource specified in table A.3-1.

Table A.3-1: Child resources of *[binarySwitch]* resource

Child Resources of *[binarySwitch]*	Child Resource Type	Multiplicity	Description
[variable]	*<flexContainer> as defined in the specialization [toggle]*	0..1	This resource is used to map 'toggle' Action defined in clause 5.3.1.12.
[variable]	*<subscription>*	0..n	See clause 9.6.8 in ETSI TS 118 101 [i.3].

The *[binarySwitch]* resource contains the attributes specified in table A.3-2.

Table A.3-2: Attributes of *[binarySwitch]* resource

Attributes of *[binarySwitch]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
resourceID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
resourceName	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
parentID	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
expirationTime	1	RW	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
creationTime	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
lastModifiedTime	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
labels	0..1	RW	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
dynamicAuthorizationConsultationIDs	0..1 (L)	RW	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
stateTag	1	RO	See clause 9.6.1.3 in ETSI TS 118 101 [i.3]
creator	0..1	RW	See clause 9.6.35 in ETSI TS 118 101 [i.3]
containerDefinition	1	WO	The value is "org.onem2m.home.moduleclass.binaryswitch"
ontologyRef	0..1	RW	See clause 9.6.35 in ETSI TS 118 101 [i.3]
contentSize	1	RO	See clause 9.6.35 in ETSI TS 118 101 [i.3]
nodeLink	0..1	RW	Not applicable to a ModuleClass specialization. This attribute is not present in an instantiation of this resource
dataGenerationTime	0..1	RO	See clause 6.2.3
powerState	1	RW	See clause 5.3.1.12

---------------------- End of change 7 --

History

Document history		
V4.0.0	November 2021	Publication