

TECHNICAL SPECIFICATION



**Electricity metering data exchange – The DLMS/COSEM suite –
Part 9-1: Communication profile using web-services to access a DLMS/COSEM
server via a COSEM Access Service (CAS)**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2016 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing 20 000 terms and definitions in English and French, with equivalent terms in 15 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

65 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

TECHNICAL SPECIFICATION



**Electricity metering data exchange – The DLMS/COSEM suite –
Part 9-1: Communication profile using web-services to access a DLMS/COSEM
server via a COSEM Access Service (CAS)**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 17.220.20; 35.100.01; 91.140.50

ISBN 978-2-8322-3329-0

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

FOREWORD.....	4
1 Scope.....	6
2 Normative references.....	6
3 Terms, definitions and abbreviations	7
4 Relation to the IEC 62056 smart metering architecture	8
4.1 Overview.....	8
4.2 Example: Using the S-FSK profile according to IEC 62056-8-3	8
5 Use cases	8
5.1 General.....	8
5.2 Use case: Device Access	9
5.2.1 Overview	9
5.2.2 On-demand Device Access	9
5.2.3 Scheduled Device Access	11
5.2.4 Device Access with Data-Notification.....	14
5.3 Use case: Device groups.....	15
6 Web services interface model	15
7 Message organisation for the WS interface.....	15
7.1 Overview.....	15
7.2 IEC 62056 messages	15
7.2.1 Overview	15
7.2.2 General	15
7.2.3 Verbs.....	16
7.2.4 Nouns.....	17
7.2.5 Payloads.....	17
7.2.6 Payload for noun Device Access	17
7.3 Common Message Envelope	23
7.3.1 Overview	23
7.3.2 General	23
7.3.3 Message header structure	23
7.3.4 Request message structure	25
7.3.5 Response message structure	27
7.3.6 Event message structure.....	30
7.3.7 Fault message structure	31
8 Interface specification.....	32
8.1 Overview.....	32
8.2 Interface using SOAP	32
8.2.1 General	32
8.2.2 WSDL structure.....	33
8.2.3 SOAP envelope	33
Annex A (informative) XML schema for the COSEM XML representation	35
A.1 COSEMpdu.....	35
A.2 XML schema for the payload definitions.....	56
A.2.1 DeviceAccess	56
A.2.2 DeviceGroup.....	57
A.3 XML Schema for common message envelope	57

A.4 WSDL for interface using SOAP	58
Bibliography	59

Figure 1 – Reference model for the COSEM Access Client to DLMS/COSEM server connection via a COSEM Access Service	6
Figure 2 – The smart metering architecture of IEC 62056.....	8
Figure 3 – Use case: On-demand Device Access.....	10
Figure 4 – Use case: Scheduled Device Access	12
Figure 5 – Use case: Scheduled Device Access with Events	13
Figure 6 – Use case: Device Access with Data-Notification	14
Figure 7 – IEC 62056 messages between CAS Client and COSEM Access Service (CAS)	16
Figure 8 – DeviceAccess overview structure.....	17
Figure 9 – DeviceGroups structure	18
Figure 10 – CosemAccessDescriptor sub-structure	19
Figure 11 – ACSEDescriptor sub-structure	19
Figure 12 – XDLMSDescriptor substructure	20
Figure 13 – cosemAccessResult substructure.....	21
Figure 14 – errorResult substructure	21
Figure 15 – ACSEResult substructure	21
Figure 16 – XDLMSResult substructure	22
Figure 17 – Common Message Envelope.....	23
Figure 18 – Common Message header structure.....	24
Figure 19 – RequestMessage structure	26
Figure 20 – XML for RequestMessage to create DeviceAccess	27
Figure 21 – ResponseMessage structure.....	28
Figure 22 – XML for ResponseMessage on create DeviceAccess.....	29
Figure 23 – EventMessage structure	30
Figure 24 – XML for EventMessage on change of DeviceAccess.....	31
Figure 25 – FaultMessage structure	32
Figure 26 – XML for FaultMessage on create of DeviceAccess	32
Figure 27 – SOAP message with RequestMessage.....	34

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**ELECTRICITY METERING DATA EXCHANGE –
THE DLMS/COSEM SUITE –****Part 9-1: Communication profile using web-services to access
a DLMS/COSEM server via a COSEM Access Service (CAS)**

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

The main task of IEC technical committees is to prepare International Standards. In exceptional circumstances, a technical committee may propose the publication of a technical specification when

- the required support cannot be obtained for the publication of an International Standard, despite repeated efforts, or
- the subject is still under technical development or where, for any other reason, there is the future but no immediate possibility of an agreement on an International Standard.

Technical specifications are subject to review within three years of publication to decide whether they can be transformed into International Standards.

IEC TS 62056-9-1, which is a technical specification, has been prepared by IEC technical committee 13: Electrical energy measurement and control.

The text of this technical specification is based on the following documents:

Enquiry draft	Report on voting
13/1641/DTS	13/1662/RVC

Full information on the voting for the approval of this technical specification can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 62056 series, published under the general title *Electricity metering data exchange – The DLMS/COSEM suite*, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC website under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- transformed into an International standard,
- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

A bilingual version of this publication may be issued at a later date.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

ELECTRICITY METERING DATA EXCHANGE – THE DLMS/COSEM SUITE –

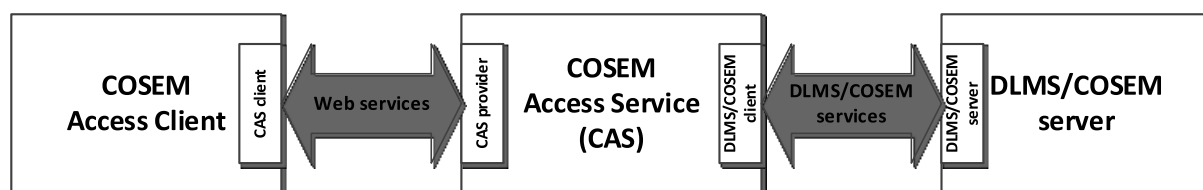
Part 9-1: Communication profile using web-services to access a DLMS/COSEM server via a COSEM Access Service (CAS)

1 Scope

This part of IEC 62056, which is a Technical Specification, defines how DLMS/COSEM servers can be accessed from a COSEM Access Client via an intermediate COSEM Access Service (CAS) providing Web services. The DLMS/COSEM server contains an application server supporting the data model of IEC 62056-6-1 / IEC 62056-6-2 and the application layer of IEC 62056-5-3. The underlying communication layers between the CAS and the DLMS/COSEM server are not covered by this specification. However, it is assumed that a profile standard exists describing how the underlying communication technology is used in conjunction with IEC 62056-6-1/ IEC 62056-6-2 and IEC 62056-5-3.

The Web services defined in this Technical Specification concern the G2 interface according to the architecture defined in IEC 62056-1-0.

The contents of this document define the Web services between the COSEM Access Client and the COSEM Access Service (CAS) as shown in Figure 1.



IEC

Figure 1 – Reference model for the COSEM Access Client to DLMS/COSEM server connection via a COSEM Access Service

The COSEM Access Client identifies the DLMS/COSEM server by its system title (see IEC 62056-5-3).

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61968-1:2012, *Application integration at electric utilities – System interfaces for distribution management – Part 1: Interface architecture and general recommendations*

IEC 61968-100:2013, *Application integration at electric utilities – System interfaces for distribution management – Part 100: Implementation profiles*

IEC 62056-1-0, *Electricity metering data exchange – Part 1-0: Smart metering standardization framework*

IEC 62056-5-3, *Electricity metering data exchange – The DLMS/COSEM suite – Part 5-3: DLMS/COSEM application layer*

IEC 62056-6-1, *Electricity metering data exchange – The DLMS/COSEM suite – Part 6-1: Object Identification System (OBIS)*

IEC 62056-6-2, *Electricity metering data exchange – The DLMS/COSEM suite – Part 6-2: COSEM interface classes*

IEC 62056-8-3, *Electricity metering data exchange – The DLMS/COSEM suite – Part 8-3: Communication profile for PLC S-FSK neighbourhood networks*

3 Terms, definitions and abbreviations

For the purposes of this document, the following terms, definitions and abbreviations apply.

ACSE	Association Control Service Element
APDU	Application Layer Protocol Data Unit
CAC	COSEM Access Client
CAS	COSEM Access Service
CAS Provider	Interface providing Web services to access the CAS
CME	Common Message Envelope
COSEM	Companion Specification for Energy Metering
COSEM Access Client	Client accessing the DLMS/COSEM server via the COSEM Access Service CAS
COSEM Access Service	Service entity providing Web service based COSEM access towards the COSEM Access Client and acting as a DLMS/COSEM client towards the DLMS/COSEM server
DLMS	Device Language Message Specification
DLMS/COSEM server	DLMS/COSEM server supporting the data models of IEC 62056-6-1 / IEC 62056-6-2 and the application layer of IEC 62056-5-3
HES	Head End System, also known as Data Collection System
IP	Internet Protocol
ISO	International Organization for Standardization
mRID	CIM master resource identifier
NN	Neighbourhood Network
NNAP	Neighbourhood Network Access Point
REST	REpresentational State Transfer
SOAP	Simple Object Access Protocol
UUID	Universal Unique Identifier
WAN	Wide area network
WS	Web services
WSDL	Web Services Definition Language
xDLMS	Extended DLMS
XML	eXtensible Markup Language
XSD	XML Schema

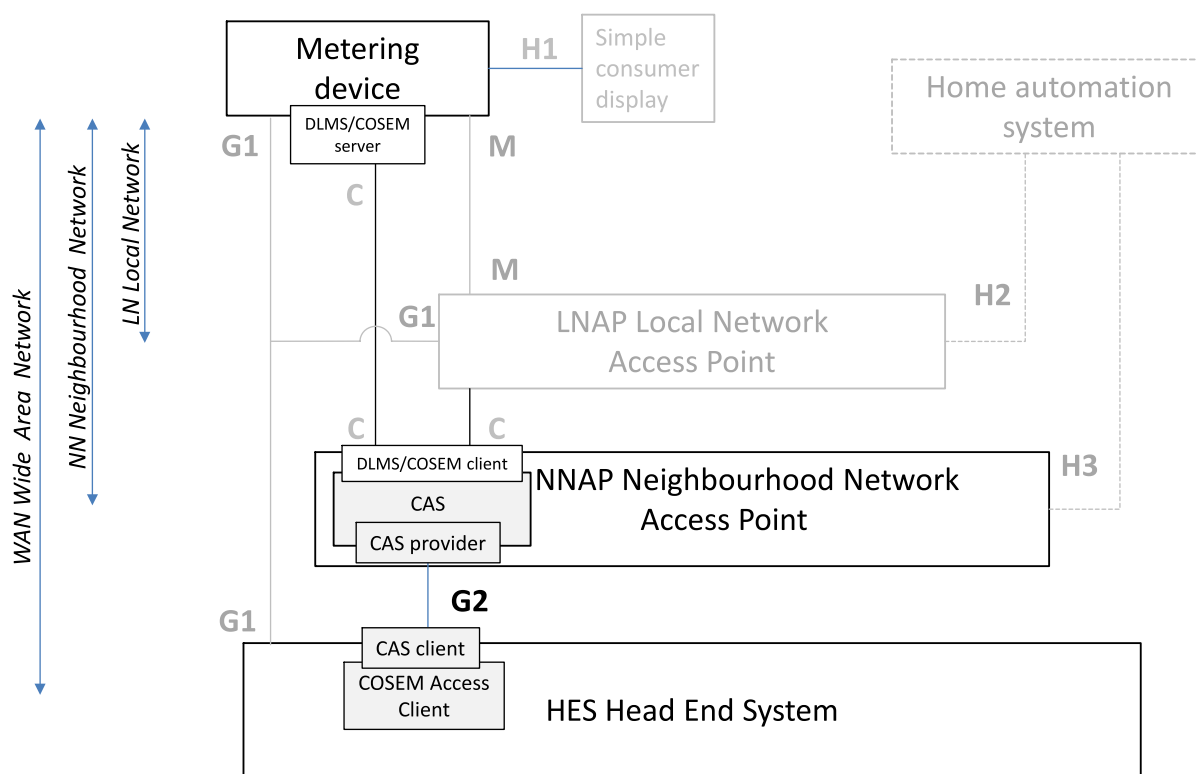
4 Relation to the IEC 62056 smart metering architecture

4.1 Overview

Considering the smart metering architecture introduced in IEC 62056-1-0 the abstract model of Figure 2 covers the communication between the HES and the NNAP as shown in Figure 2. Where the HES acts as COSEM Access Client, the NNAP provides the COSEM Access Service (CAS) and the DLMS/COSEM server corresponds to the metering device. The Web services defined in this TS concern the G2 interface. Typically, the communication technology used in the NN is not the same as the communication technology used in the WAN; i.e. the NN does not necessarily support IP based communication.

4.2 Example: Using the S-FSK profile according to IEC 62056-8-3

When the S-FSK PLC profile is used between the NNAP (CAS) and the metering device (DLMS/COSEM server), the system titles of the NNAP and the metering device are exchanged during the registration process using the CIASE protocol. In this case the information of the system titles shall be made available to the HES (COSEM Access Client) prior to the establishment of any communication with the metering device.



IEC

Figure 2 – The smart metering architecture of IEC 62056

5 Use cases

5.1 General

The use cases defined in the following subclauses are based on the reference model of Figure 1; i.e. the use cases concern the messaging for the data exchange between a COSEM Access Client and a DLMS/COSEM server via a CAS; the use cases cover the data exchange processes between the actors involved. The use cases for the applications to support the smart metering business processes are not considered.

The following actors are considered (see Figure 1):

- COSEM Access Client;
- COSEM Access Service;
- DLMS/COSEM server.

The use cases are supported by 3 generic types of messages (supporting the data exchange processes) exchanged between the COSEM Access Client and the COSEM Access Service:

- WS-Request();
- WS-Response();
- WS-Event().

The COSEM Access Client issues a request by sending a WS-Request message to the COSEM Access Service. The COSEM Access Service in return sends WS-Response messages or WS-Event messages to the COSEM Access Client. The COSEM Access Client expects one of the following results from the issued request:

- the request is successfully processed and a WS-Response message is returned in a timely manner;
- the request is accepted, but results in a WS-Response message that returns an application level error code or a fault;
- the request is accepted, the WS-Response message is returned and the results are sent asynchronously to COSEM Access Client by means of WS-Event messages.

5.2 Use case: Device Access

5.2.1 Overview

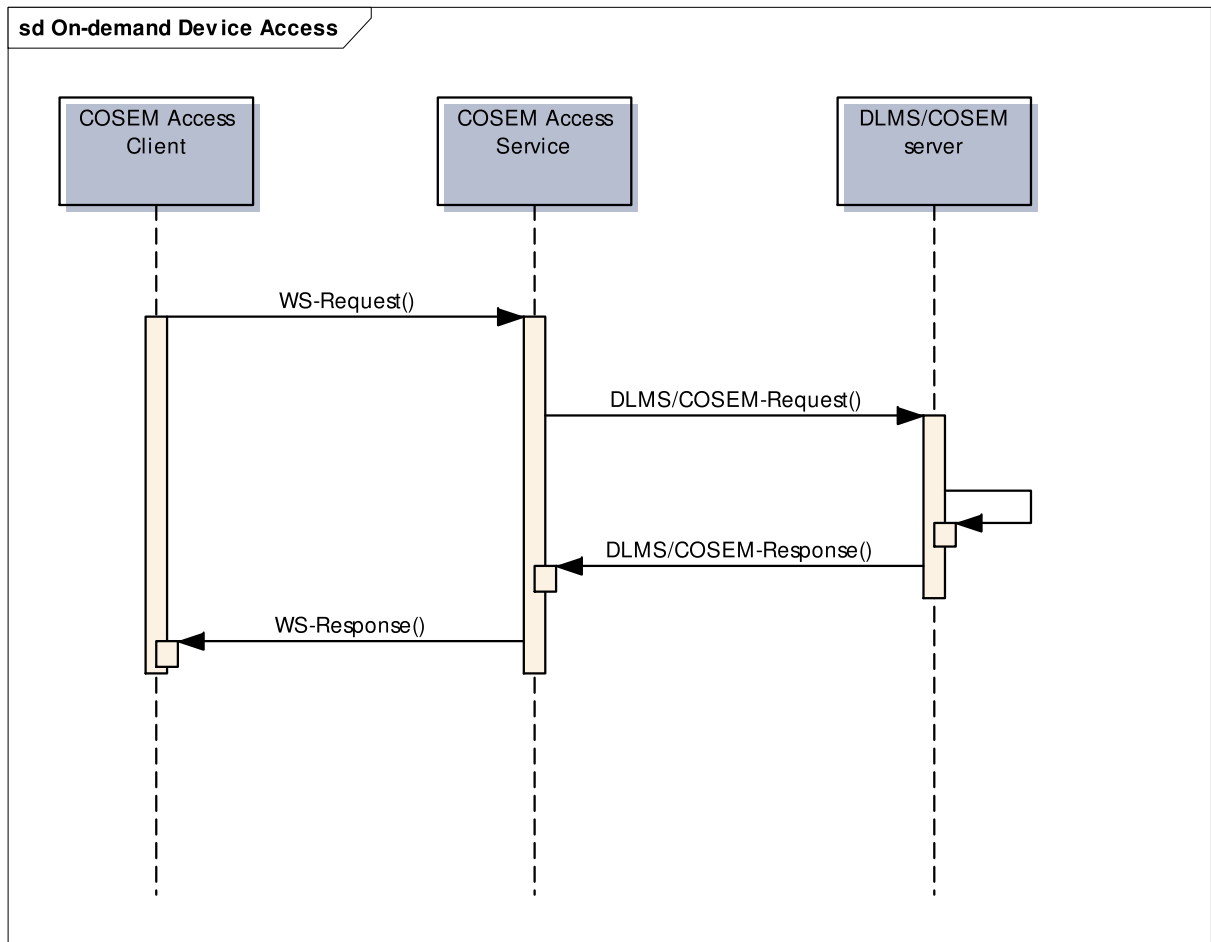
The COSEM Access Client requests to access a set of COSEM objects in a set of DLMS/COSEM servers.

The data exchange is initiated and managed by the COSEM Access Client using the standardised services and processes defined in IEC 62056-5-3. In particular, the COSEM Access Client assembles the DeviceAccess messages which encapsulate the corresponding xDLMS APDUs. The DeviceAccess messages are transmitted to the COSEM Access Service for processing and transmission to the DLMS/COSEM servers. Processing involves on-demand or scheduled activation of exchanges with DLMS/COSEM servers and providing the results of this exchange. For that purpose the COSEM Access Client uses the Web services provided by the WS Provider in the COSEM Access Service.

5.2.2 On-demand Device Access

5.2.2.1 Overview

The requests are executed immediately by the CAS. Upon availability, the results are made available by the CAS via Web services. The corresponding message sequence diagram is shown in Figure 3.



IEC

Figure 3 – Use case: On-demand Device Access

5.2.2.2 Parameters

On demand access is defined by the following parameters:

DevicesReferenceList (mandatory element):

Specifies the list of DLMS/COSEM servers which need to be accessed via the CAS. Besides the identification of the physical device the *DeviceReferenceList* also contains the identification of the logical device of the server.

CosemAccessList (mandatory element):

Specifies the list of COSEM services to be invoked in the DLMS/COSEM servers defined in *DevicesReferenceList*.

Activates (optional element):

For on-demand access the element *Activates* (optional element) is omitted, meaning that this service is immediately activated.

NotOlderThan (optional element):

Any result from the DLMS/COSEM servers invoked by the CAS which is older than “*NotOlderThan*” will no more be available at the CAS Provider.

Priority (optional element):

Defines the priority with which the CAS has to invoke the COSEM services listed in the COSEMAccessList.

Expires (optional element):

Specifies the date and time when the CAS has to stop invoking the services listed in the CosemAccessList.

5.2.2.3 Result

The result from the DLMS/COSEM server shall be delivered to the CAS Provider interface upon availability. The result may be notified, directly transmitted to the COSEM Access Client or queried by the COSEM Access Client. Only data which is “NotOlderThan” can be successfully queried. Once delivered to the COSEM Access Client the result may not be available in the CAS anymore.

The result of an “on-demand access” has the following form:

OnDemandDeviceAccess/CosemAccessList/CosemAccess[0..n]/CosemAccessResult[0...n]

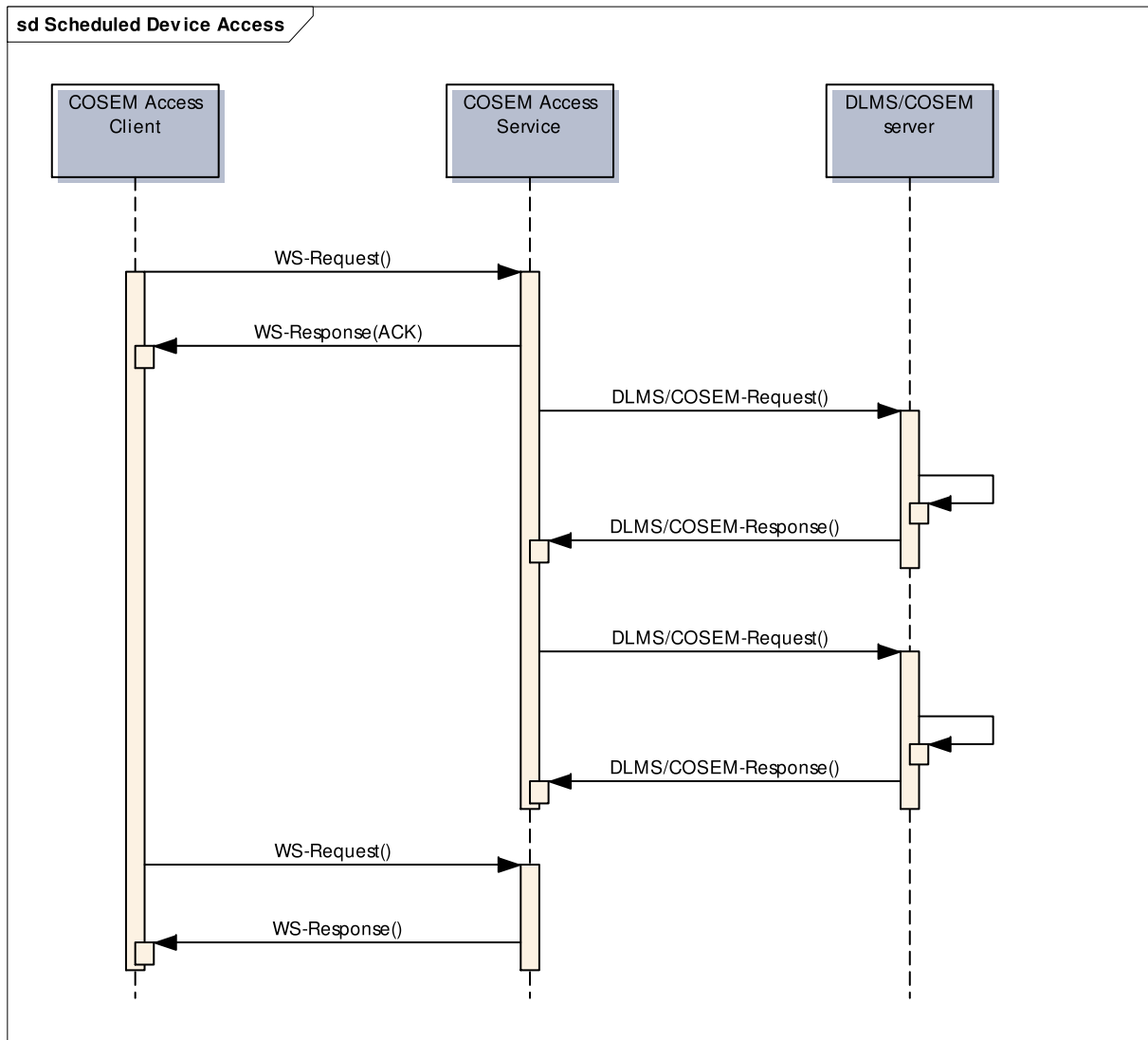
Any error message shall contain sufficient information to locate the source of the error. In particular three error locations shall be distinguished:

- to report errors on the DLMS/COSEM server level the CosemAccessResult is according to COSEM APDU definitions in IEC 62056-5-3;
- to report errors on the communication between the DLMS/COSEM client interface of the CAS and the DLMS/COSEM server the CosemAccessResult is extended according to 7.2.6.1.4;
- to report errors on the CAS level the CosemAccessResult is extended according to 7.2.6.1.4.

5.2.3 Scheduled Device Access

5.2.3.1 Overview

The execution time of the access to the DLMS/COSEM server by the CAS can be defined by means of the “*Activates*” parameter. In particular, the execution may be performed once at a specific point of time or the execution may be repeated several times. The corresponding message sequence diagrams are shown in Figure 4 and Figure 5.

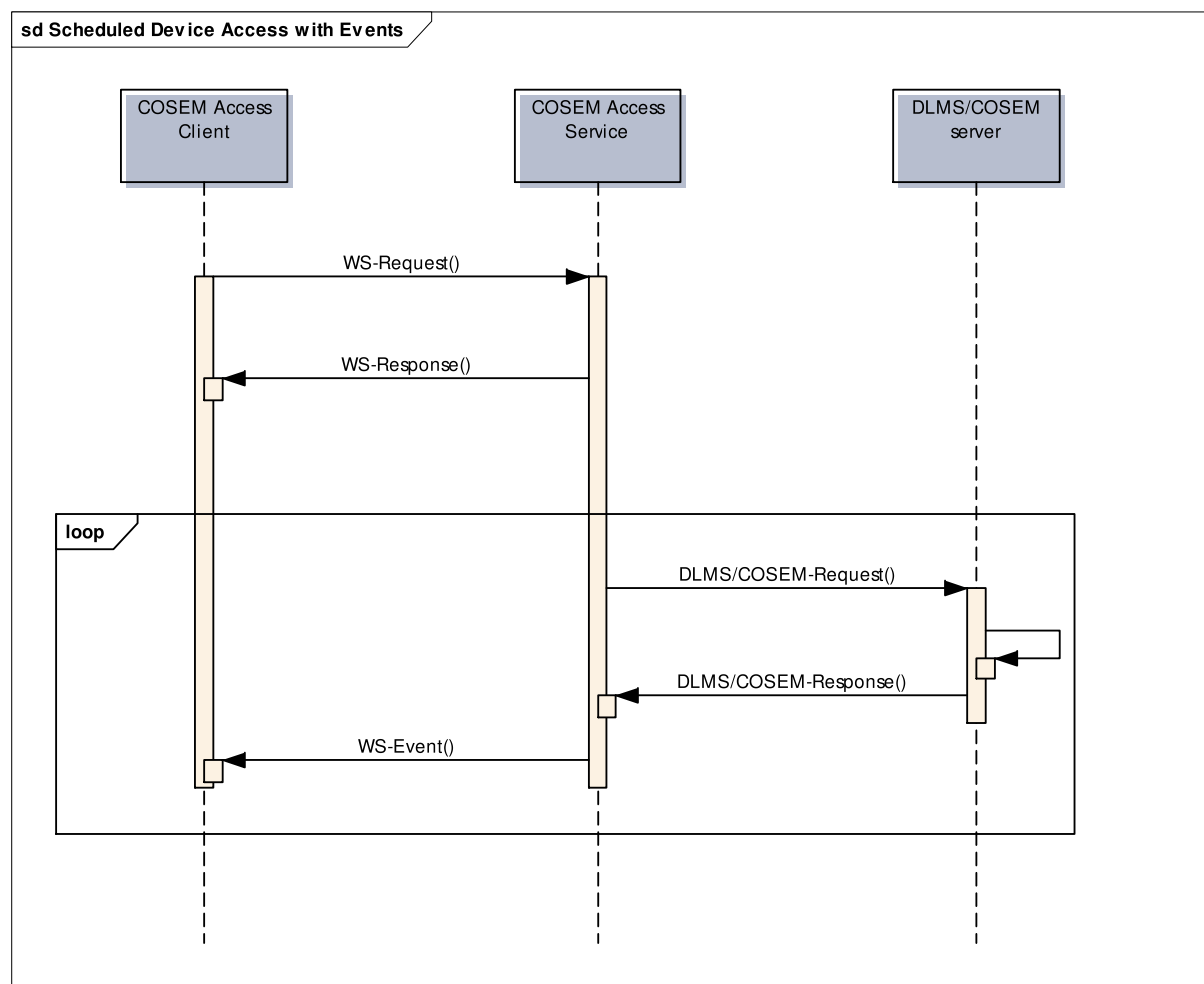


IEC

Figure 4 – Use case: Scheduled Device Access

The use case Scheduled Device Access message exchange consists of the following process:

- the COSEM Access Client issues a request by sending a WS-Request message to COSEM Access Service;
- the COSEM Access Service in return sends a WS-Response message acknowledging the request to COSEM Access Client;
- the COSEM Access Service issues DLMS/COSEM-Request messages to the DLMS/COSEM server according to the defined schedule and then processes the DLMS/COSEM-Response messages received from COSEM server;
- the COSEM Access Client issues a request by sending a WS-Request message to the COSEM Access Service requesting the results of the scheduled DLMS/COSEM requests;
- the COSEM Access Service sends a WS-Response message with the results of the scheduled DLMS/COSEM requests to the COSEM Access Client.



IEC

Figure 5 – Use case: Scheduled Device Access with Events

The use case Scheduled Device Access with Events message exchange consists of the following process:

- the COSEM Access Client issues a request by sending a WS-Request message to the COSEM Access Service;
- the COSEM Access Service in return sends a WS-Response message acknowledging the request to the COSEM Access Client;
- the COSEM Access Service issues DLMS/COSEM-Request messages to the DLMS/COSEM server according to defined schedule;
- the COSEM Access Service processes the DLMS/COSEM-Response messages and sends WS-Event messages to the COSEM Access Client.

5.2.3.2 Parameters

Scheduled access uses the parameters listed in 5.2.2.2. For scheduled access the parameter “Activates” becomes a mandatory element defining the details of the access schedule.

Activates (mandatory element)

Specifies the date and time when the CAS has to invoke the services listed in the CosemAccessList. Start time, period, end time, validity window, according to the Schedule (ISO 8601).

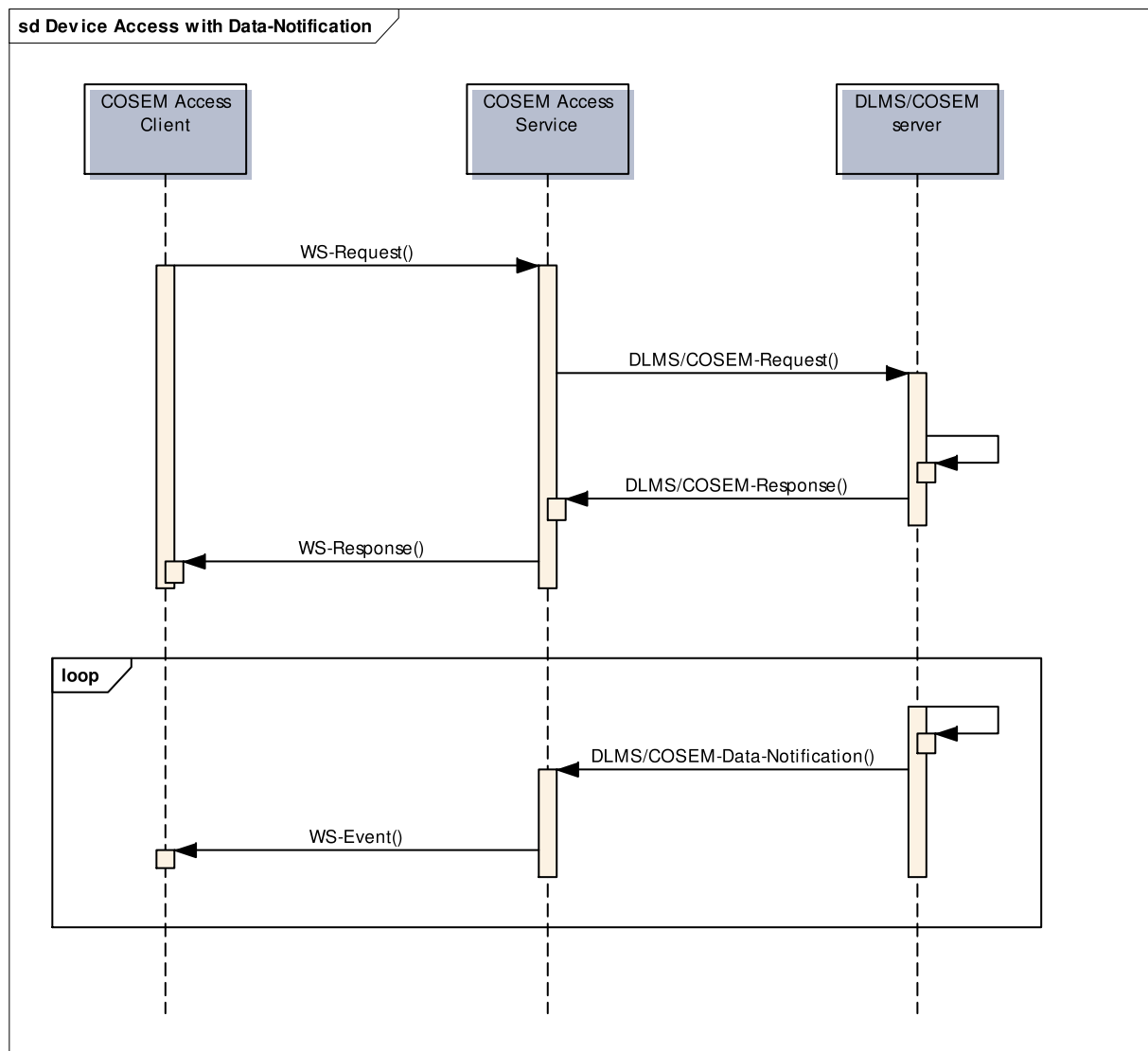
5.2.3.3 Result

The result from scheduled access follows the definitions in 5.2.2.3.

5.2.4 Device Access with Data-Notification

5.2.4.1 Overview

The requests from the COSEM Access Client are used to set-up the Push operation in the DLMS/COSEM server. With the response from CAS to the CAC the request (to set-up the Push operation) is acknowledged. The actual Push operation in the DLMS/COSEM server consists of sending Data-Notification APDUs to the CAS. The CAS then sends the received Data-Notification APDUs via events to the COSEM Access Client. The corresponding message sequence diagram is shown in Figure 6.



IEC

Figure 6 – Use case: Device Access with Data-Notification

The use case Scheduled Device Access with Data-Notification message exchange consists of the following process:

- the COSEM Access Client issues a request by sending a WS-Request message to the COSEM Access Service;

- the COSEM Access Service in return sends a WS-Response message acknowledging the execution of the request to the COSEM Access Client;
- the COSEM Access Service receives messages with DLMS/COSEM-Data-Notification APDUs from the DLMS/COSEM server;
- the COSEM Access Service processes the DLMS/COSEM-Data-Notification APDUs and sends the corresponding WS-Event messages to COSEM Access Client.

5.2.4.2 Parameters

Device Access with Data-Notification uses the parameters listed in 5.2.3.2.

5.2.4.3 Result

The result from Device Access with Data-Notification follows the definitions in 5.2.2.3.

5.3 Use case: Device groups

The COSEM Access Client requests the CAS to define a group of DLMS/COSEM servers.

6 Web services interface model

The interface follows the basic request/response and publish/subscribe patterns using Web services as shown in Figure 3, in Figure 4 and in Figure 6. The Web service client issues a request to the Web service provider in the CAS. The details of the interface are defined in Clause 8.

Depending on the nature of the request and the availability of the communication medium between the CAS and the DLMS/COSEM server, the Cosem Access Client may expect the result as a response to the request immediately (synchronous operation), or delayed (scheduled operation), or as an event triggered by the reception of a Data-Notification APDU from the DLMS/COSEM server. The use case “On Demand Device Access” uses synchronous operation whereas the use case “Scheduled Device Access” is based on scheduled operation. The use case “Device Access with Data-Notification” is using events on Data-Notification.

7 Message organisation for the WS interface

7.1 Overview

The generic Request/Reply services are supported by different types of messages at the Web service interfaces. The WS interface is constructed to accept messages consisting of a *Verb*, a *Noun* and a *Payload*. The Noun identifies the type of the Payload that may be part of the Request and of the Reply.

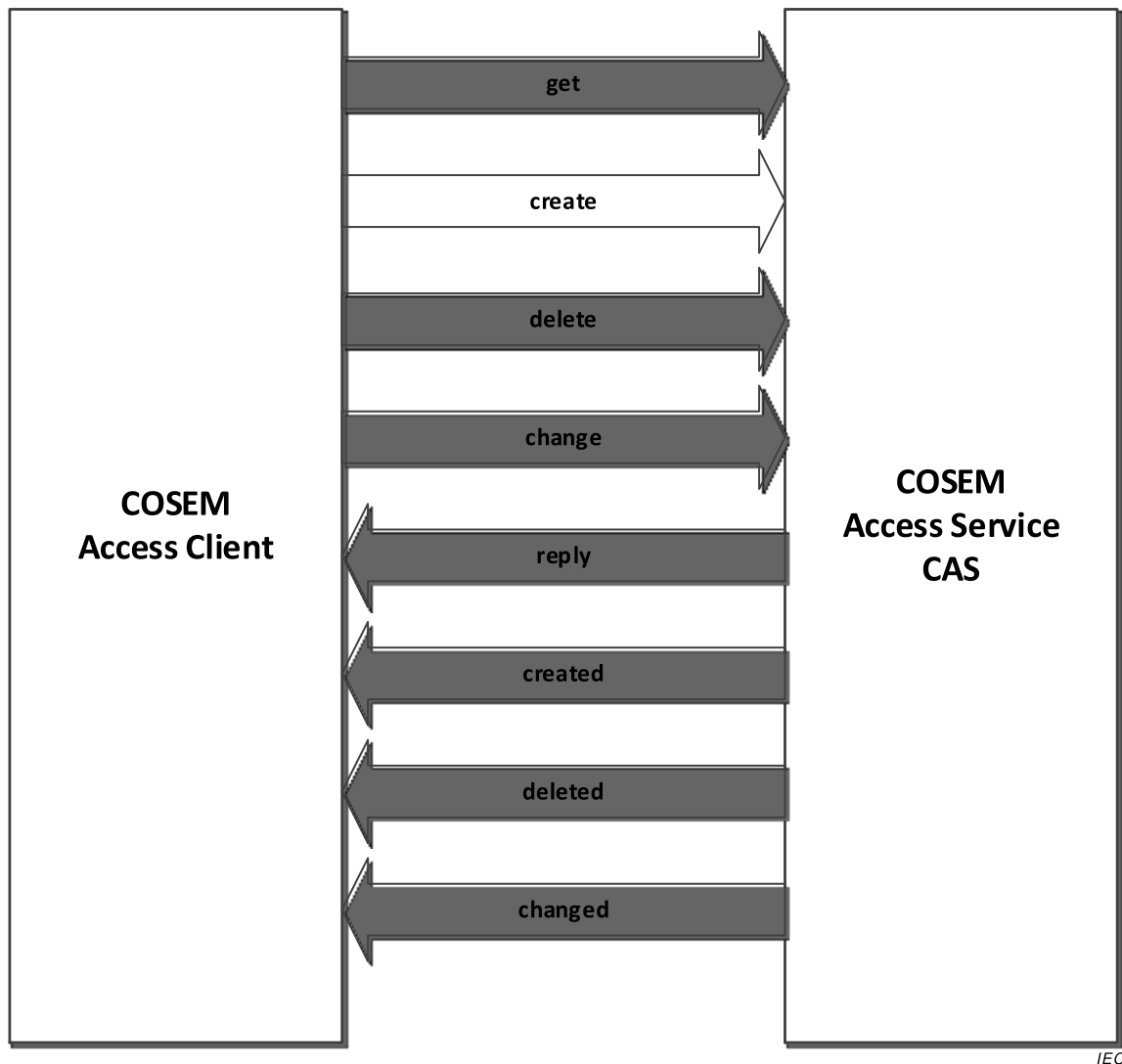
7.2 IEC 62056 messages

7.2.1 Overview

In the following the IEC 62056 information exchange messages in terms of a verb, a noun and the payload are described. IEC 62056 messages are based on IEC 61968-1 and IEC 61968-100 where only selected verbs are used and new nouns and payloads are defined.

7.2.2 General

IEC 61968-1 describes the information exchanges in terms of a verb, noun and payload. Figure 7 shows the IEC 62056 messages between the COSEM Access Client and the COSEM Access Service (CAS) described with the corresponding the verbs.



IEC

Figure 7 – IEC 62056 messages between CAS Client and COSEM Access Service (CAS)

7.2.3 Verbs

The verbs used in the Request are also used to identify the Reply and the publication of an Event which may be the consequence of the successful transaction initiated by the Request.

The following verbs are used in the IEC 62056 messages:

- 'get' is used in requests to query for objects of the type specified by the noun;
- 'create' is used in requests to create objects of the type specified by the noun;
- 'delete' is used in requests to delete objects of the type specified by the noun;
- 'change' is used in requests to change objects of the type specified by the noun;
- 'reply' is used in responses to requests;
- 'created' is used in events indicating that the objects specified by the noun have been created;
- 'deleted' is used in events indicating that the objects specified by the noun have been deleted;
- 'changed' is used in events indicating that the objects specified by the noun are have been changed.

7.2.4 Nouns

Nouns are used to identify the type of information being exchanged. Each noun has a corresponding XML Schema uniquely assigned to it. Nouns identify the use case supported by the exchanged information.

The following nouns are used in the IEC 62056 messages:

- Device Access;
- Device Groups.

7.2.5 Payloads

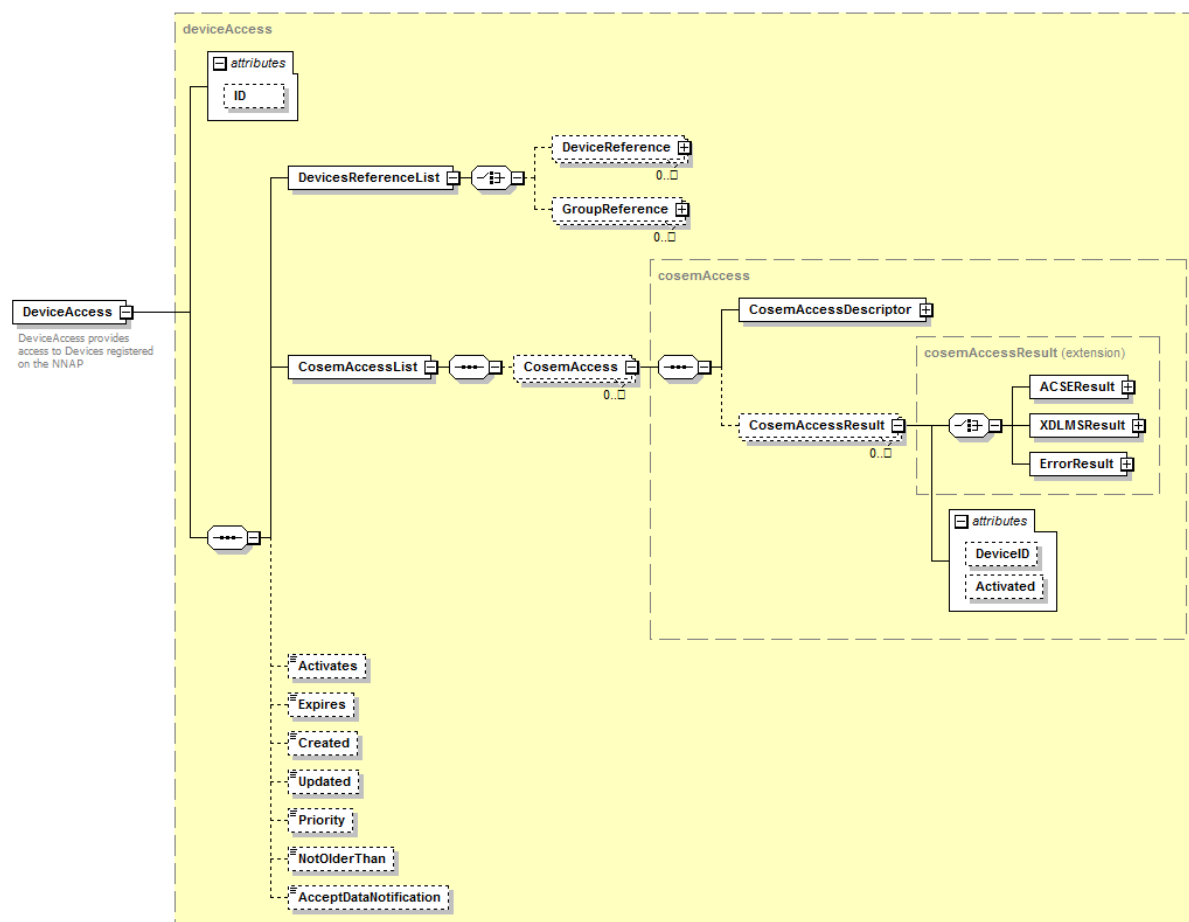
Each noun identifies the payload structure which is specified with the corresponding XML Schema. Depending on the situation, a payload may or may not be required in a message.

7.2.6 Payload for noun Device Access

7.2.6.1 Overview

7.2.6.1.1 General

The detailed XML schema for the Device Access payload is presented in A.2.1. The Figure 8 shows the structure of the payload for the noun Device Access.



IEC

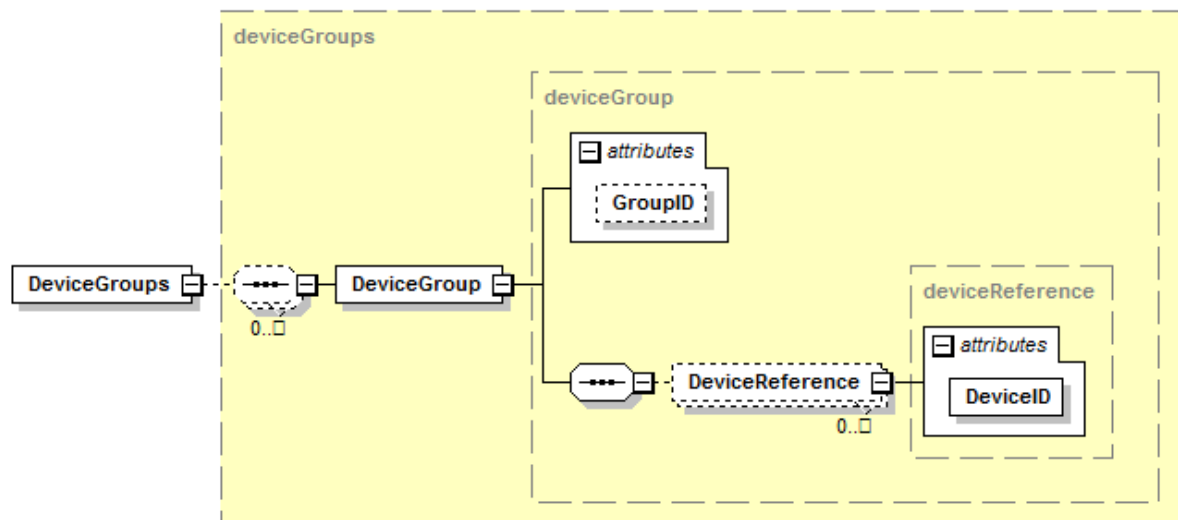
Figure 8 – DeviceAccess overview structure

Fields that can be optionally supplied include the following:

- **DeviceReferenceList:** Specifies the list of DLMS/COSEM servers which need to be accessed via the CAS. Besides the identification of the physical device the DeviceReferenceList also contains the identification of the logical device of the server;
- **CosemAccessList:** Specifies the list of DLMS/COSEM services to be invoked with the DLMS/COSEM servers defined with DevicesReferenceList;
- **Activates:** Specifies the date and time when the CAS has to invoke the DLMS/COSEM services listed in the CosemAccessList. Start time, period, end time, validity window, according to the “Schedule” definitions of ISO 8601;
- **Expires:** Specifies the date and time when the CAS has to stop invoking the DLMS/COSEM services listed in the CosemAccessList;
- **Created:** Contains the date and time when the DeviceAccess has been created;
- **Updated:** Contains the date and time when the DeviceAccess has been updated;
- **Priority:** Defines the priority with which the CAS has to invoke the DLMS/COSEM services in the COSEMAccessList;
- **NotOlderThan:** Any result from the DLMS/COSEM servers invoked by the CAS which is older than “NotOlderThan” will no more be available at the WS Provider of the CAS;
- **AcceptDataNotification:** Specifies if Data-Notification APDUs from the DLMS/COSEM servers are accepted. When not defined then the Data-Notification APDUs are discarded.

7.2.6.1.2 DeviceGroups

The GroupReference element of DeviceAccess (see Figure 8) corresponds to the GroupID attribute of DeviceGroups (see Figure 9).



IEC

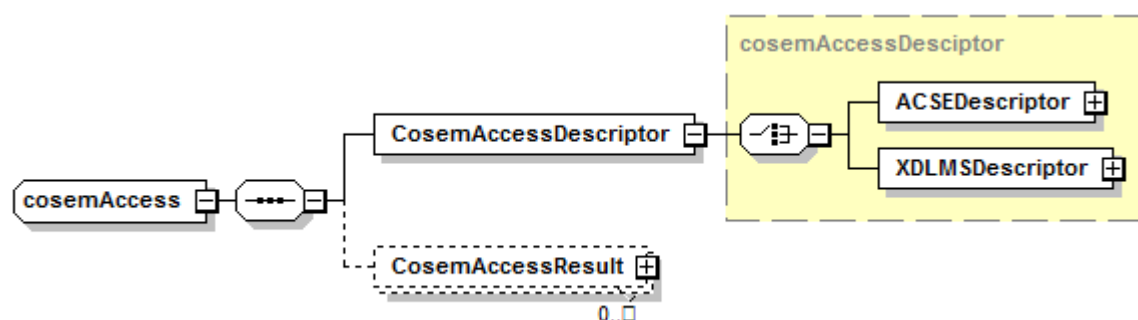
Figure 9 – DeviceGroups structure

DeviceGroups consist of a set of DeviceGroup elements where each element consists of the GroupID and a list of DeviceIDs.

7.2.6.1.3 CosemAccessDescriptor

7.2.6.1.3.1 General

The CosemAccessDescriptor element of the DeviceAccess (see Figure 8) is shown in Figure 10.



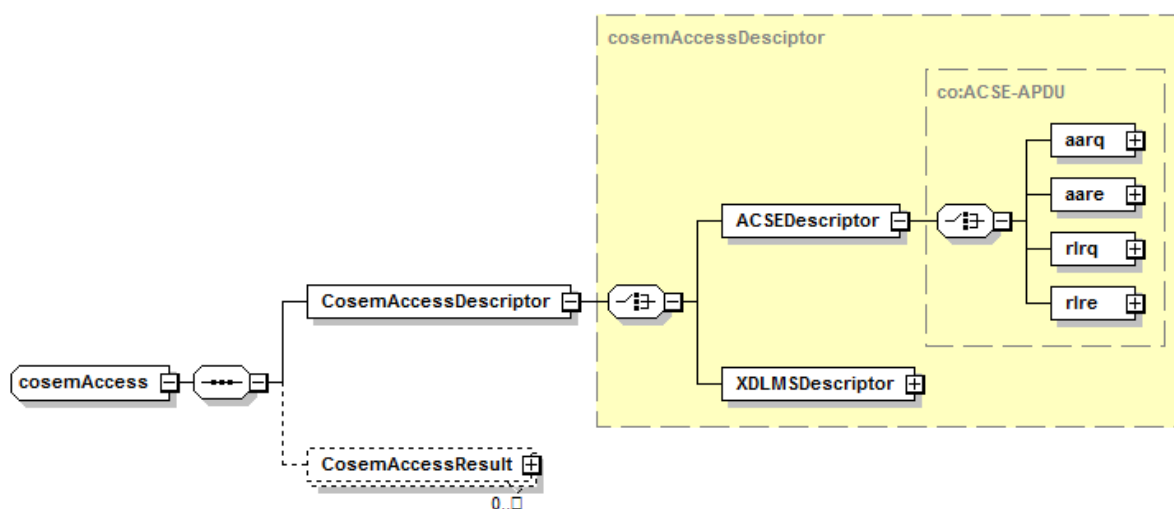
IEC

Figure 10 – CosemAccessDescriptor sub-structure

The CosemAccessDescriptor consists of the ACSEDescriptor or the XDLMSDescriptor element.

7.2.6.1.3.2 ACSEDescriptor

The ACSEDescriptor element of the DeviceAccess (see Figure 8) is shown in Figure 11.



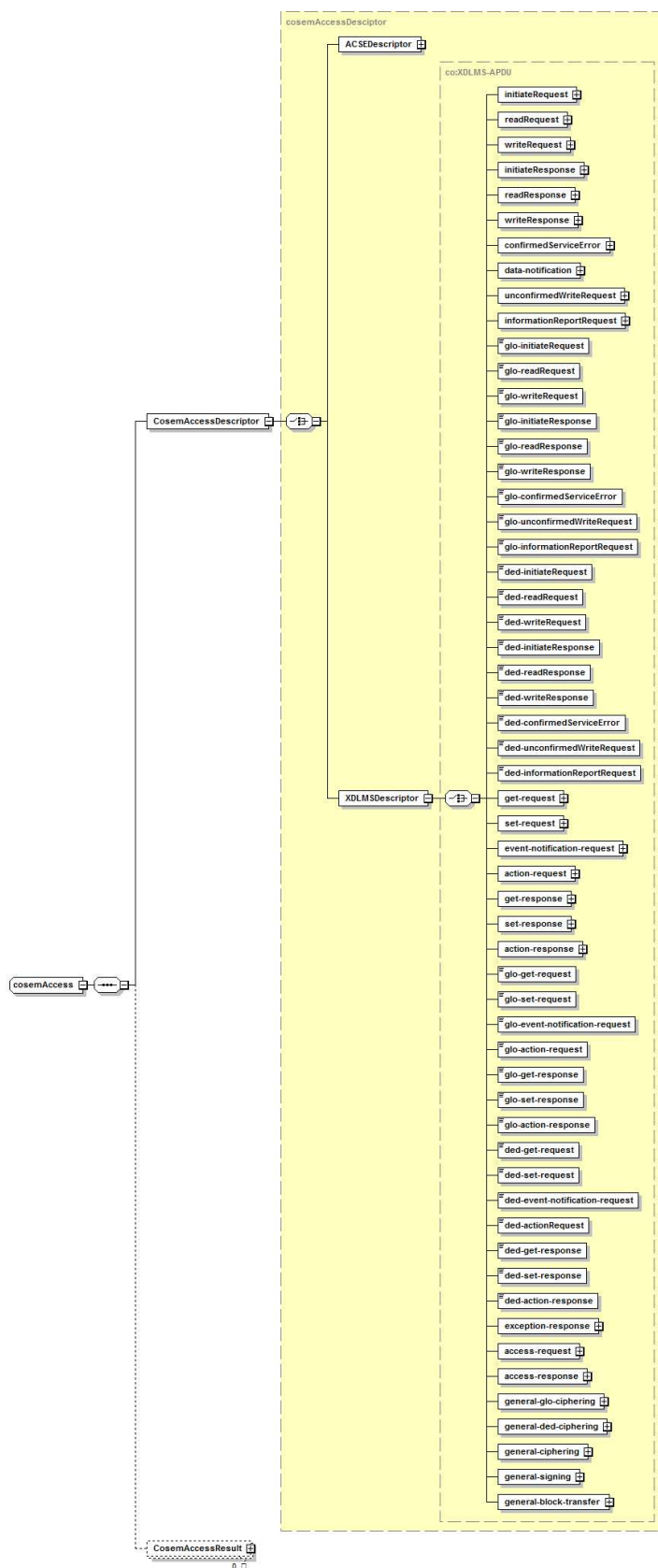
IEC

Figure 11 – ACSEDescriptor sub-structure

ACSEDescriptor consists of the aarq, aare, rlrq and rlre elements. These elements are representation of DLMS/COSEM ACSE APDUs. Only request DLMS/COSEM ACSE APDUs may be present in the ACSEDescriptor.

7.2.6.1.3.3 XDLMSDescriptor

The XDLMSDescriptor element of the DeviceAccess (see Figure 8) is shown in Figure 12.



IEC

Figure 12 – XDLMSDescriptor substructure

The XDLMSDescriptor consists of elements that are representation of DLMS/COSEM xDLMS APDUs. Only request DLMS/COSEM xDLMS APDUs may be present in the XDLMSDescriptor.

7.2.6.1.4 CosemAccessResult

7.2.6.1.4.1 General

The CosemAccessResult element of the DeviceAccess (see Figure 8) is shown in Figure 13.

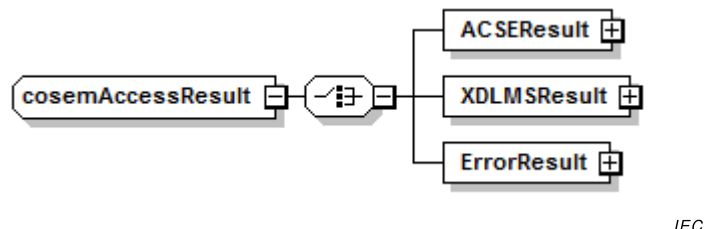


Figure 13 – cosemAccessResult substructure

CosemAccessResult consists of the ACSEResult, XDLMSError or ErrorResult element.

The errorResult element of the DeviceAccess (see Figure 8) is shown in Figure 14.

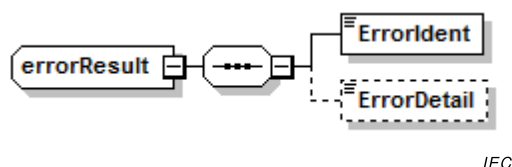


Figure 14 – errorResult substructure

In the case when the ACSEResult or XDLMSError cannot be returned the ErrorResult is used. The errorResult element consist of ErrorIdent and ErrorDetail.

- ErrorIdent: Contains the identification of the error that have occurred during the processing of the DeviceAccess message. The following values are defined:
 - Communication Error (1);
 - Internal CAS Error (2);
- ErrorDetail: Contains the text of the error detail.

7.2.6.1.4.2 ACSEResult

The ACSEResult element of the DeviceAccess element (see Figure 8) is shown in Figure 15.

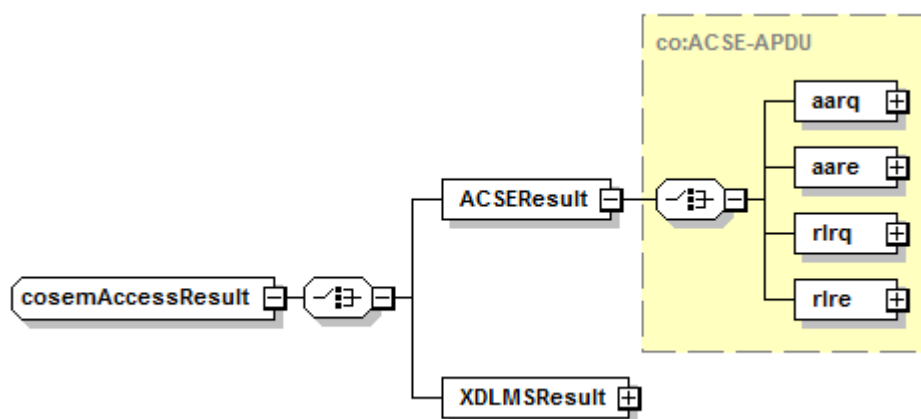
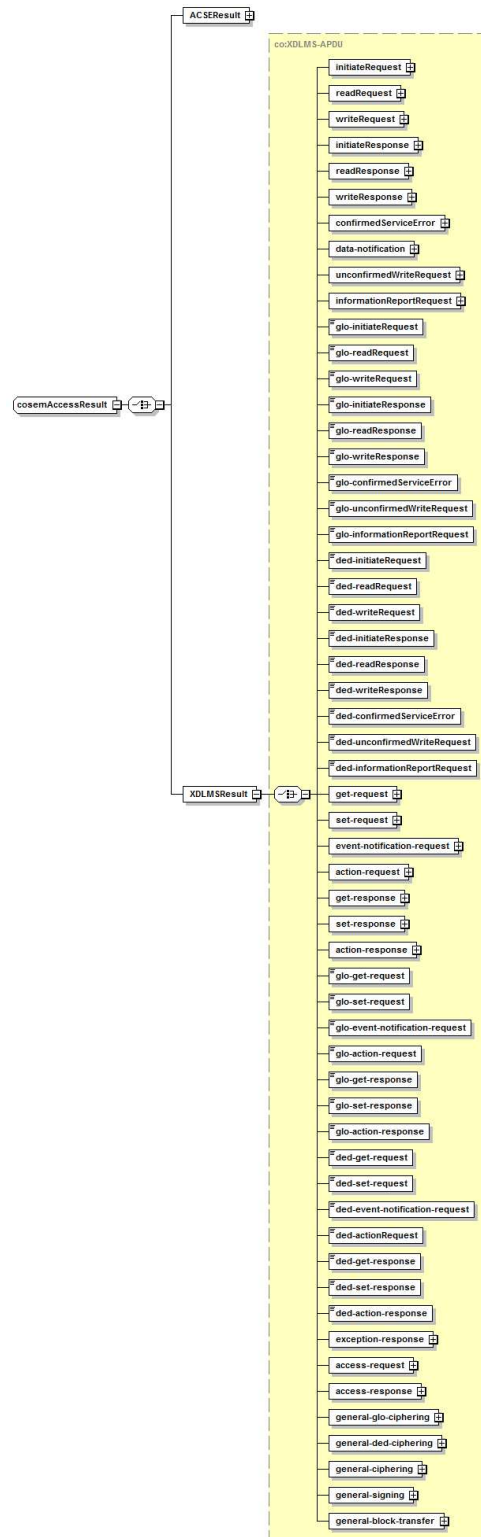


Figure 15 – ACSEResult substructure

ACSERResult consists of the aarg, aare, rlrq and rlre elements. These elements are representation of DLMS/COSEM ACSE APDUs. Only response DLMS/COSEM ACSE APDUs may be present in the ACSERResult.

7.2.6.1.4.3 XDLMSResult

The XDLMSResult element of the DeviceAccess (see Figure 8) is shown in Figure 16.



IEC

Figure 16 – XDLMSResult substructure

XDLMSResult consists of elements that are representation of DLMS/COSEM xDLMS APDUs. Only response DLMS/COSEM xDLMS APDUs may be present in the XDLMSResult.

7.3 Common Message Envelope

7.3.1 Overview

All messages use a common message envelope (CME) as defined in IEC 61968-100.

7.3.2 General

Messages are constructed with several sections, including:

- Header: Required for all messages (except for fault response messages). The header is providing a common structure for all service interfaces;
- Request: optional. The Request field defines commonly used parameters needed to qualify 'get' query requests, or identify specific objects for 'delete', 'cancel' or 'close' requests. Not used for event or response messages;
- Reply: Required only for response messages to indicate success, failure and error details. Not used for request or event messages;
- Payload: Used to convey message information as a consequence of the 'Verb' and 'Noun' combination in the message Header. Required for 'create', 'change' and 'execute' requests. Payload is also required for event messages. Optional in other cases as described in Annex B of IEC 61968-100:2013. The payload structure provides options for payload compression.

Figure 17 provides a generalized view of the high-level message structure:

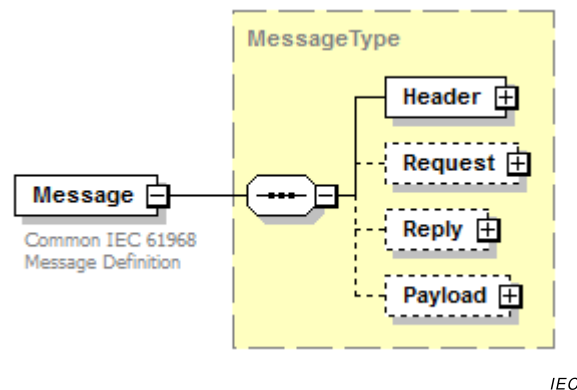


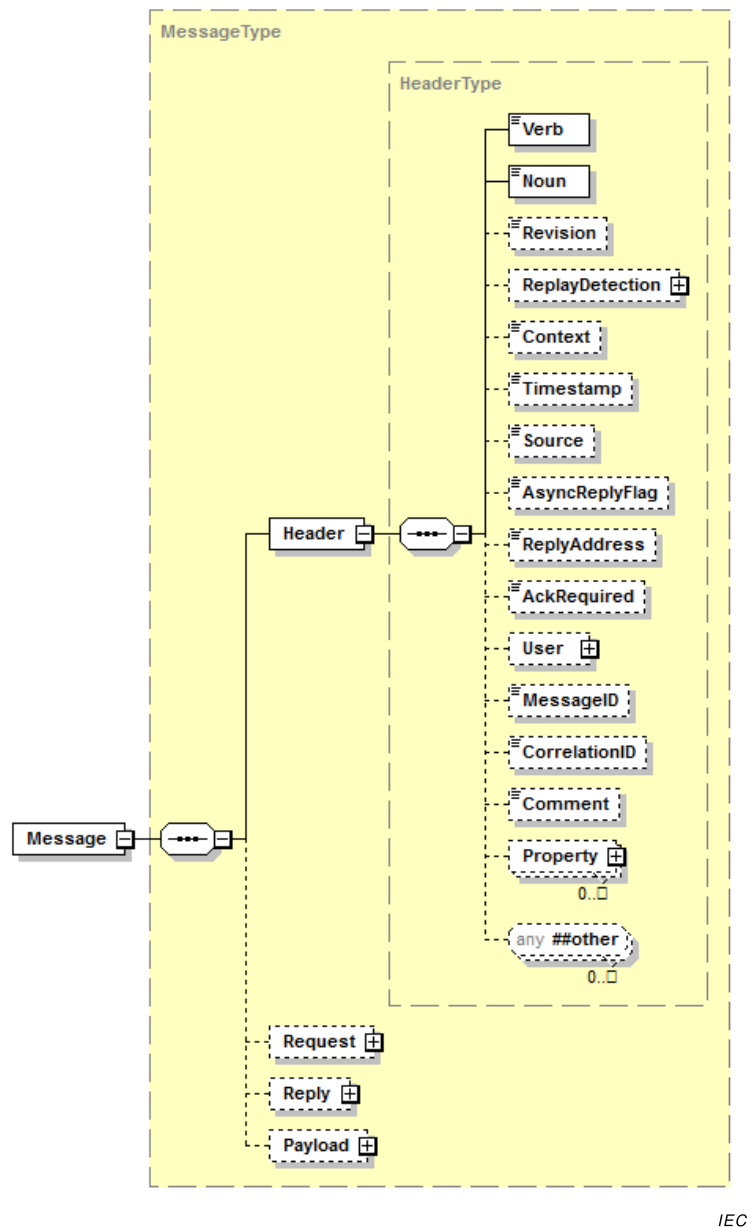
Figure 17 – Common Message Envelope

7.3.3 Message header structure

The header is common to request, response and event messages. The header has two required fields that shall be populated, these include:

- Verb, to identify a specific action to be taken. There is a defined set of valid verbs, where commonly used values include 'get', 'create', 'change', 'cancel', 'close', 'execute' and 'reply'. Within event notification messages 'past tense' verbs are used, which can include 'created', 'changed', 'cancelled', 'closed' and 'executed';
- Noun: to identify the subject of the action and/or the type of the payload, such as DeviceAccess, DeviceGroups, etc.

Figure 18 shows the common message header structure.



IEC

Figure 18 – Common Message header structure

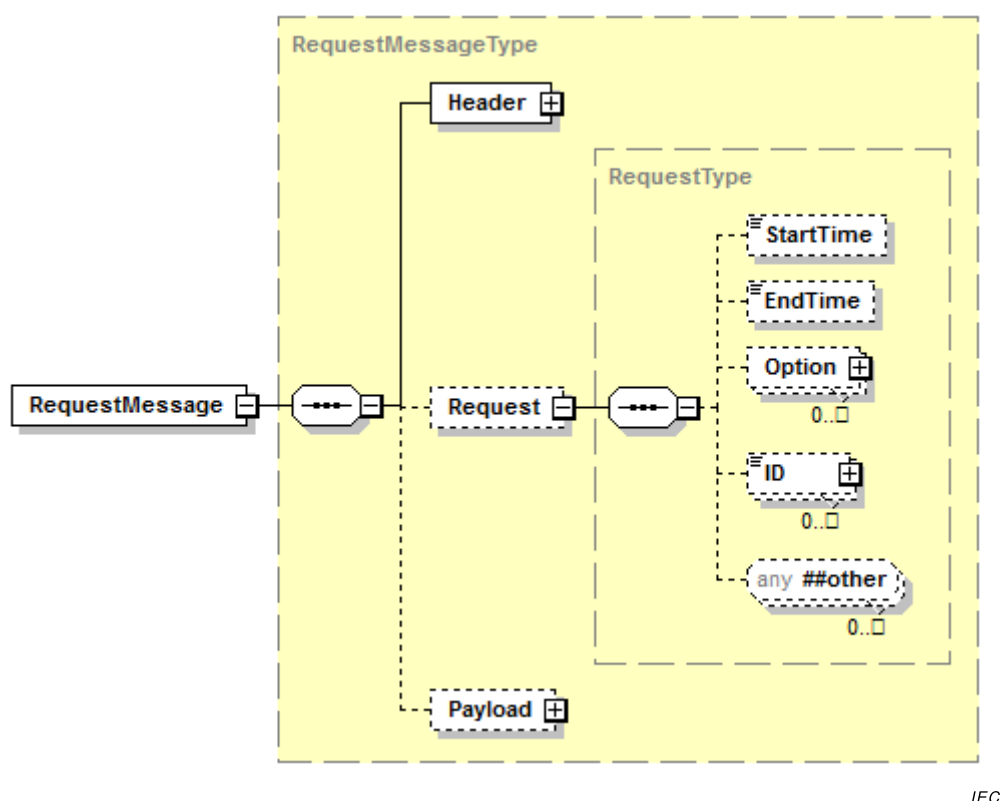
Fields that may be optionally supplied include the following:

- **Revision:** To indicate the revision of the message definition. This should be '1' by default. As long as there is no revision (i.e. there exists just one version) this field is not used;
- **ReplayDetection:** This is a complex element with a timestamp and a nonce used to guard against replay attacks. The timestamp is generated by the source system to indicate when the message was created. The nonce is a sequence number or randomly generated string (e.g. UUID) that will not be repeated by the source system for at least a day;
- **Context:** A string that may be used to identify the context of the message. This can help to provide an application level guard against incorrect message consumption in configurations where there may be multiple system environments running over the same messaging infrastructure. Some example values are PRODUCTION, TESTING, STUDY and TRAINING. This field is not used;
- **Timestamp:** An ISO 8601 compliant string that identifies the time the message was sent;
- **Source:** identifying the source of the message, which should be the name of the system or organization;

- AsyncReplyFlag: A Boolean ('true' or 'false') that indicates whether a reply message will be sent asynchronously. Replies are assumed to be sent synchronously by default. This field is not used;
- ReplyAddress: The address to which replies should be sent. This information is typically used for asynchronous replies. It is ignored when using one-way integration patterns (e.g. AckRequired=false). The reply address should be an URL beginning with 'http://' or 'https://'. In case of "Scheduled Access": when ReplyAddress is left empty then the events cannot be sent to the COSEM Access Client; i.e. the results have to be polled at the CAS;
- AckRequired: This is a Boolean ('true' or 'false') that indicates whether or not an acknowledgement is required. If false, it indicates that a one-way integration pattern is being used for communicating transactional messages. For the use cases foreseen in this document this field is not used;
- User: A complex structure that identifies the user and associated organization. The presence of this field may be required for some interfaces, depending upon underlying implementations. This field allows a UserID string and optional Organization string as subelements;
- MessageID: A string that uniquely identifies a message. The use of a UUID or sequence number is recommended. A process should not issue two messages using the same MessageID value;
- CorrelationID: This field is used to 'link' messages together. It shall be supplied as part of the request message to allow the client to link the corresponding reply message with the request. The server will place the incoming CorrelationID value as the CorrelationID on the outgoing reply. If not supplied on the request, the CorrelationID of the reply shall be set to the value of the MessageID that was used in the request, if present. Considering that the CorrelationID is used to 'link' messages together, the same CorrelationID may be used in more than one request message. The use of a UUID or sequence numbers as CorrelationIDs is recommended;
- Comment: Contains any descriptive text. The contents of this field are not be used for any processing logic;
- Property: A complex type that allows project-specific name/value pairs to be conveyed. The source and destinations need to agree upon the usage of this information;
- any: May be used for project-specific extensions.

7.3.4 Request message structure

Figure 19 shows the structure of RequestMessage:



IEC

Figure 19 – RequestMessage structure

Besides the Header and the Payload the RequestMessage may optionally contain the element “Request” consisting of additional parameters relevant for the request. The typical use of the “Request” element is to avoid the placement of application specific request parameters in the header or in the payload elements.

There are no required elements in the Request element. The usage of elements within the Request element is described as follows:

- **StartTime**: Used when a query needs to specify a start time as a filter, but no such parameter is provided in a ‘Get’ message (a message with the verb “get”). If both exist, this will be ignored;
- **EndTime**: Used when a query needs to specify an end time as a filter, but no such parameter is provided in a ‘Get’ message. If both exist, this will be ignored;
- **Option**: Used when name/value pairs are useful in filtering a query or to convey general or custom request options. Examples of usage are the specification of a transaction timeout value or specifying a response mode such as ‘Aggregated’ or ‘Streaming’. At the current time there are no normative enumerations for these values. This field is not used;
- **ID**: Used when the ID of one or more objects are needed to filter a query request. Can also be used to identify specific objects in the case of ‘delete’. Each ID can specify attributes, first to identify the kind of ID, which can be name, UUID, transaction or other. The default of UUID is used for mRID values;
- **any**: Can be used for other nonstandard extensions. In cases where a ‘Get’ message is used, the elements defined within the ‘Get’ message take precedence over the StartTime, EndTime and ID elements. This recognizes the asymmetry between the information needed to qualify a request from the information that is returned on a reply.

Figure 20 is an XML example of the RequestMessage.

```

<?xml version="1.0" encoding="UTF-8"?>
<RequestMessage
  xmlns="http://iec.ch/TC57/2011/schema/message"
  xmlns:da="http://iec.ch/TC13/2014/schema/DeviceAccess"
  xmlns:co="http://www.dlms.com/COSEMpdu"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message IEC-61968-CME.xsd">
  <Header>
    <Verb>create</Verb>
    <Noun>DeviceAccess</Noun>
    <Timestamp>2014-01-12T10:02:00Z</Timestamp>
    <MessageID>78465521</MessageID>
    <CorrelationID>78465521</CorrelationID>
  </Header>
  <Payload>
    <da:DeviceAccess
      xsi:schemaLocation="http://iec.ch/TC13/2014/schema/DeviceAccess DeviceAccess.xsd">
      <da:DevicesReferenceList>
        <da:DeviceReference DeviceID="4D4D4D0000BC614E"/>
      </da:DevicesReferenceList>
      <da:CosemAccessList>
        <da:CosemAccess>
          <da:CosemAccessDescriptor>
            <da:XDLMSSDescriptor>
              <co:get-request>
                <co:get-request-normal>
                  <co:invoke-id-and-priority>64</co:invoke-id-and-priority>
                  <co:cosem-attribute-descriptor>
                    <co:class-id>3</co:class-id>
                    <co:instance-id>0100010801FF</co:instance-id>
                    <co:attribute-id>0</co:attribute-id>
                  </co:cosem-attribute-descriptor>
                </co:get-request-normal>
              </co:get-request>
            </da:XDLMSSDescriptor>
          </da:CosemAccessDescriptor>
        </da:CosemAccess>
      </da:CosemAccessList>
    </da:DeviceAccess>
  </Payload>
</RequestMessage>

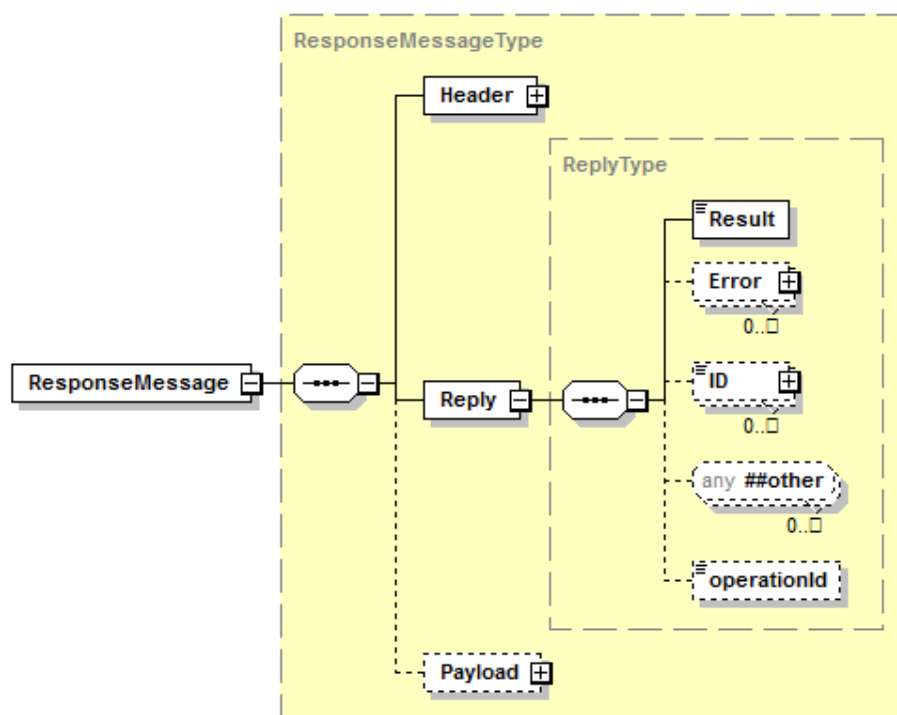
```

Figure 20 – XML for RequestMessage to create DeviceAccess

DeviceID “4D4D4D0000BC614E” represents DLMS/COSEM SystemTitle of the DLMS/COSEM server.

7.3.5 Response message structure

Figure 21 shows the structure of ResponseMessage:



IEC

Figure 21 – ResponseMessage structure

The Reply.result value is enumerated in CME XSD, and is populated in the following manner:

- "OK" if there are no errors and all results have been returned. There is no requirement that a Reply.Error element be present;
- "PARTIAL" if only a partial set of results has been returned, with or without errors. The existence of errors is indicated with one or more Reply.Error.code elements. PARTIAL is not used;
- "FAILED" if no result can be returned due to one or more errors, indicated with one or more Reply.Error elements, each with a mandatory application level 'code' '.

Figure 22 is an XML example of the ResponseMessage.

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseMessage
  xmlns="http://iec.ch/TC57/2011/schema/message"
  xmlns:da="http://iec.ch/TC13/2014/schema/DeviceAccess"
  xmlns:co="http://www.dlms.com/COSEMpdu"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message IEC-61968-CME.xsd">
  <Header>
    <Verb>reply</Verb>
    <Noun>DeviceAccess</Noun>
    <Timestamp>2014-01-12T10:02:10Z</Timestamp>
    <MessageID>78465522</MessageID>
    <CorrelationID>78465521</CorrelationID>
  </Header>
  <Reply>
    <Result>OK</Result>
    <ID>9BD5CA94-B181-476D-9BF1-8C6F5D9E69F8</ID>
  </Reply>
  <Payload>
    <da:DeviceAccess
      xsi:schemaLocation="http://iec.ch/TC13/2014/schema/DeviceAccess DeviceAccess.xsd"
      ID="9BD5CA94-B181-476D-9BF1-8C6F5D9E69F8">
      <da:DevicesReferenceList>
        <da:DeviceReference DeviceID="4D4D4D000BC614E"/>
      </da:DevicesReferenceList>
      <da:CosemAccessList>
        <da:CosemAccess>
          <da:CosemAccessDescriptor >
            <da:XDLMSSDescriptor>
              <co:get-request>
                <co:get-request-normal>
                  <co:invoke-id-and-priority>64</co:invoke-id-and-priority>
                  <co:cosem-attribute-descriptor>
                    <co:class-id>3</co:class-id>
                    <co:instance-id>0100010801FF</co:instance-id>
                    <co:attribute-id>0</co:attribute-id>
                  </co:cosem-attribute-descriptor>
                </co:get-request-normal>
              </co:get-request>
            </da:XDLMSSDescriptor>
          </da:CosemAccessDescriptor>
          <da:CosemAccessResult>
            <da:XDLMSSResult>
              <co:get-response>
                <co:get-response-normal>
                  <co:invoke-id-and-priority>64</co:invoke-id-and-priority>
                  <co:result>
                    <co:data>
                      <co:structure>
                        <co:octet-string>0100010801FF</co:octet-string>
                        <co:double-long-unsigned>34566678</co:double-long-
unsigned>
                      </co:structure>
                    </co:data>
                  </co:result>
                </co:get-response-normal>
              </co:get-response>
            </da:XDLMSSResult>
          </da:CosemAccessResult>
        </da:CosemAccess>
      </da:CosemAccessList>
    </da:DeviceAccess>
  </Payload>
</ResponseMessage>

```

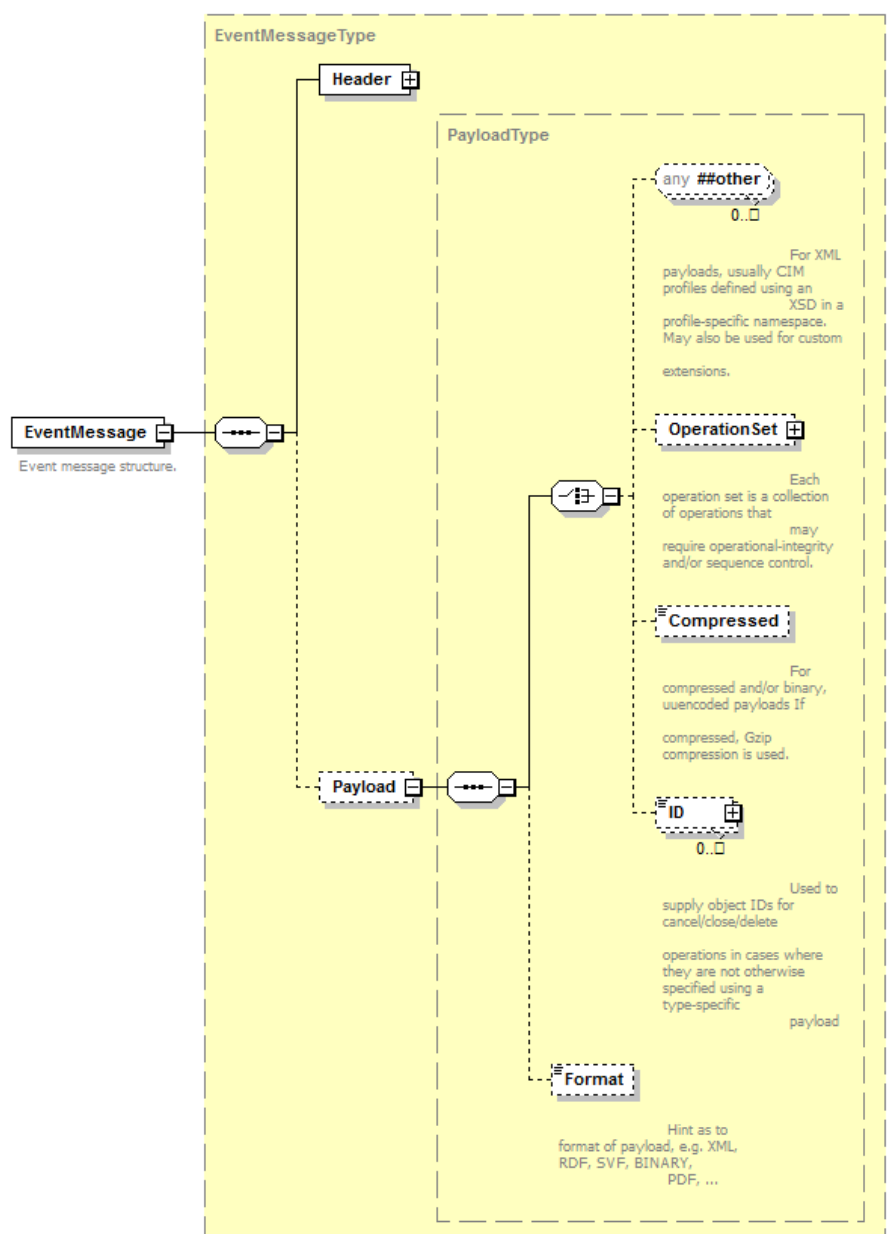
Figure 22 – XML for ResponseMessage on create DeviceAccess

The operation is synchronous and the result of the DLMS/COSEM get-request is provided in the payload.

7.3.6 Event message structure

An EventMessage is typically published to report a condition of potential interest. The verbs used in an event message are past tense, e.g. created, changed, cancelled, etc. An EventMessage will not include request or reply parameters, just a header and usually a payload.

Figure 23 shows the structure of an EventMessage.



IEC

Figure 23 – EventMessage structure

Figure 24 is an XML example of the EventMessage:

```

<?xml version="1.0" encoding="UTF-8"?>
<EventMessage
  xmlns="http://iec.ch/TC57/2011/schema/message"
  xmlns:da="http://iec.ch/TC13/2014/schema/DeviceAccess"
  xmlns:co="http://www.dlms.com/COSEmpdu"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message IEC-61968-CME.xsd">
  <Header>
    <Verb>changed</Verb>
    <Noun>DeviceAccess</Noun>
    <Timestamp>2014-01-12T10:02:10Z</Timestamp>
    <MessageID>78465523</MessageID>
    <CorrelationID>78465521</CorrelationID>
  </Header>
  <Payload>
    <da:DeviceAccess
      xsi:schemaLocation="http://iec.ch/TC13/2014/schema/DeviceAccess DeviceAccess.xsd"
      ID="9BD5CA94-B181-476D-9BF1-8C6F5D9E69F8">
      <da:DevicesReferenceList>
        <da:DeviceReference DeviceID="4D4D4D0000BC614E"/>
      </da:DevicesReferenceList>
      <da:CosemAccessList>
        <da:CosemAccess>
          <da:CosemAccessDescriptor>
            <da:XDLMSDescriptor>
              <co:get-request>
                <co:get-request-normal>
                  <co:invoke-id-and-priority>64</co:invoke-id-and-priority>
                  <co:cosem-attribute-descriptor>
                    <co:class-id>3</co:class-id>
                    <co:instance-id>0100010801FF</co:instance-id>
                    <co:attribute-id>0</co:attribute-id>
                  </co:cosem-attribute-descriptor>
                </co:get-request-normal>
              </co:get-request>
            </da:XDLMSDescriptor>
          </da:CosemAccessDescriptor>
          <da:CosemAccessResult>
            <da:XDLMSResult>
              <co:get-response>
                <co:get-response-normal>
                  <co:invoke-id-and-priority>64</co:invoke-id-and-priority>
                  <co:result>
                    <co:data>
                      <co:structure>
                        <co:octet-string>0100010801FF</co:octet-string>
                        <co:double-long-unsigned>34566678</co:double-long-
unsigned>
                      </co:structure>
                    </co:data>
                  </co:result>
                </co:get-response-normal>
              </co:get-response>
            </da:XDLMSResult>
          </da:CosemAccessResult>
        </da:CosemAccess>
      </da:CosemAccessList>
    </da:DeviceAccess>
  </Payload>
</EventMessage>

```

Figure 24 – XML for EventMessage on change of DeviceAccess

7.3.7 Fault message structure

A FaultMessage is typically used within the definition of a WSDL and implemented by a Web service to report a fault condition as a consequence of a failed attempt to process a RequestMessage (e.g. detection of a SOAP fault). It only uses a reply element (i.e. no header), as it may not have been able to interpret even the header of the RequestMessage.

The FaultMessage Structure is shown in Figure 25.

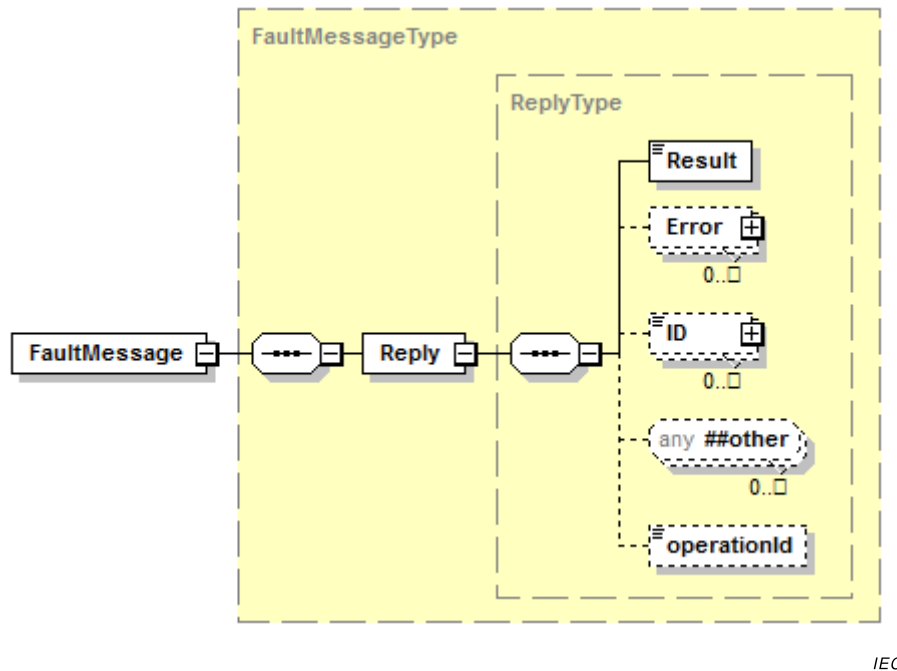


Figure 25 – FaultMessage structure

Figure 26 is an XML example of the FaultMessage.

```
<?xml version="1.0" encoding="UTF-8"?>
<FaultMessage
  xmlns="http://iec.ch/TC57/2011/schema/message"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message IEC-61968-CME.xsd">
  <Reply>
    <Result>FAILED</Result>
    <ID>9BD5CA94-B181-476D-9BF1-8C6F5D9E69F8</ID>
  </Reply>
</FaultMessage>
```

Figure 26 – XML for FaultMessage on create of DeviceAccess

8 Interface specification

8.1 Overview

Clause 8 contains the definitions of the Web service interface using SOAP. The Web service interface using SOAP describes how common message envelope (CME) XML messages are encapsulated into SOAP messages using the document-centric style.

Web services using REpresentational State Transfer (REST) are an alternative to the use of SOAP. Currently the interface using REST is not specified. However, the structure of this document allows future extensions to REST.

8.2 Interface using SOAP

8.2.1 General

This subclause describes the definition of the Web service interface using SOAP. The Web service interface is described with WSDL. WSDL describes the binding to the SOAP messages where the body of SOAP message contains common message envelope (CME) messages.

8.2.2 WSDL structure

Generic WSDL is specified in Clause A.4. WSDL is made of two parts, the abstract and the concrete part, with the following elements:

- Definitions;
- Types are defined with the CME XSDs;
- Messages are defined as generic Common Message Envelope messages:
 - RequestMessage;
 - ResponseMessage;
 - EventMessage;
- PortTypes define Operations:
 - Request;
 - Response;
 - PublishEvent;
- Bindings:
 - Standard SOAP binding;
- Services:
 - Service using SOAP.

8.2.3 SOAP envelope

SOAP (1.2) defines the “SOAP envelope” with a structure for composing messages for exchanging information using Web services. The envelope contains a header and a body.

The SOAP Body contains common message envelope (CME) messages as defined in 7.3.

Figure 27 is an XML example of the SOAP message with a RequestMessage:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:co="http://www.dlms.com/COSEMPdu"
  xmlns:da="http://iec.ch/TC13/2014/schema/DeviceAccess" >
  <soapenv:Header>
    <wsa:To>
      http://10.0.1.2/services/DeviceAccess
    </wsa:To>
  </soapenv:Header>
  <soapenv:Body>
    <m:RequestMessage xmlns:m="http://iec.ch/TC57/2011/schema/message">
      <m:Header>
        <m:Verb>create</m:Verb>
        <m:Noun>DeviceAccess</m:Noun>
        <m:Timestamp>2014-01-12T10:02:00Z</m:Timestamp>
        <m:MessageID>78465521</m:MessageID>
        <m:CorrelationID>78465521</m:CorrelationID>
      </m:Header>
      <m:Payload>
        <da:DeviceAccess>
          <da:DevicesReferenceList>
            <da:DeviceReference DeviceID="4D4D4D0000BC614E"/>
          </da:DevicesReferenceList>
          <da:CosemAccessList>
            <da:CosemAccess>
              <da:CosemAccessDescriptor>
                <da:XDLMSDescriptor>
                  <co:get-request>
                    <co:get-request-normal>
                      <co:invoke-id-and-priority>64</co:invoke-id-and-priority>
                      <co:cosem-attribute-descriptor>
                        <co:class-id>3</co:class-id>
                        <co:instance-id>0100010801FF</co:instance-id>
                        <co:attribute-id>0</co:attribute-id>
                      </co:cosem-attribute-descriptor>
                    </co:get-request-normal>
                  </co:get-request>
                </da:XDLMSDescriptor>
              </da:CosemAccessDescriptor>
            </da:CosemAccess>
          </da:CosemAccessList>
        </da:DeviceAccess>
      </m:Payload>
    </m:RequestMessage>
  </soapenv:Body>
</soapenv:Envelope>

```

Figure 27 – SOAP message with RequestMessage

Annex A (informative)

XML schema for the COSEM XML representation

A.1 COSEMpdu

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.dlms.com/COSEMpdu"
  targetNamespace="http://www.dlms.com/COSEMpdu"
  elementFormDefault="qualified">

  <!-- ASN.1 definitions -->
  <xsd:complexType name="NULL" final="#all" />

  <xsd:simpleType name="BitString">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[0-1]{0,}" />
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="ObjectIdentifier">
    <xsd:restriction base="xsd:token">
      <xsd:pattern value="[0-2](\[. [1-3]?[0-9]?(\. \d+)*\]" />
    </xsd:restriction>
  </xsd:simpleType>

  <!-- ACSE-APDU definition -->
  <xsd:element name="aCSE-APDU" type="ACSE-APDU"/>
  <xsd:complexType name="ACSE-APDU">
    <xsd:choice>
      <xsd:element name="aarg" type="AARQ-apdu"/>
      <xsd:element name="aare" type="AARE-apdu"/>
      <xsd:element name="rlrq" type="RLRQ-apdu"/>
      <xsd:element name="rlre" type="RLRE-apdu"/>
    </xsd:choice>
  </xsd:complexType>

  <!-- xDLMS-APDU definition -->
  <xsd:element name="xDLMS-APDU" type="XLMS-APDU"/>
  <xsd:complexType name="XLMS-APDU">
    <xsd:choice>
      <xsd:element name="initiateRequest" type="InitiateRequest"/>
      <xsd:element name="readRequest" type="ReadRequest"/>
      <xsd:element name="writeRequest" type="WriteRequest"/>
      <xsd:element name="initiateResponse" type="InitiateResponse"/>
      <xsd:element name="readResponse" type="ReadResponse"/>
      <xsd:element name="writeResponse" type="WriteResponse"/>
      <xsd:element name="confirmedServiceError" type="ConfirmedServiceError"/>
      <xsd:element name="data-notification" type="Data-Notification"/>
      <xsd:element name="unconfirmedWriteRequest" type="UnconfirmedWriteRequest"/>
      <xsd:element name="informationReportRequest" type="InformationReportRequest"/>
      <xsd:element name="glo-initiateRequest" type="xsd:hexBinary"/>
      <xsd:element name="glo-readRequest" type="xsd:hexBinary"/>
      <xsd:element name="glo-writeRequest" type="xsd:hexBinary"/>
      <xsd:element name="glo-initiateResponse" type="xsd:hexBinary"/>
      <xsd:element name="glo-readResponse" type="xsd:hexBinary"/>
      <xsd:element name="glo-writeResponse" type="xsd:hexBinary"/>
      <xsd:element name="glo-confirmedServiceError" type="xsd:hexBinary"/>
      <xsd:element name="glo-unconfirmedWriteRequest" type="xsd:hexBinary"/>
      <xsd:element name="glo-informationReportRequest" type="xsd:hexBinary"/>
      <xsd:element name="ded-initiateRequest" type="xsd:hexBinary"/>
      <xsd:element name="ded-readRequest" type="xsd:hexBinary"/>
      <xsd:element name="ded-writeRequest" type="xsd:hexBinary"/>
      <xsd:element name="ded-initiateResponse" type="xsd:hexBinary"/>
      <xsd:element name="ded-readResponse" type="xsd:hexBinary"/>
      <xsd:element name="ded-writeResponse" type="xsd:hexBinary"/>
      <xsd:element name="ded-confirmedServiceError" type="xsd:hexBinary"/>
      <xsd:element name="ded-unconfirmedWriteRequest" type="xsd:hexBinary"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:schema>
```

```

<xsd:element name="ded-informationReportRequest" type="xsd:hexBinary"/>
<xsd:element name="get-request" type="Get-Request"/>
<xsd:element name="set-request" type="Set-Request"/>
<xsd:element name="event-notification-request" type="EventNotificationRequest"/>
<xsd:element name="action-request" type="Action-Request"/>
<xsd:element name="get-response" type="Get-Response"/>
<xsd:element name="set-response" type="Set-Response"/>
<xsd:element name="action-response" type="Action-Response"/>
<xsd:element name="glo-get-request" type="xsd:hexBinary"/>
<xsd:element name="glo-set-request" type="xsd:hexBinary"/>
<xsd:element name="glo-event-notification-request" type="xsd:hexBinary"/>
<xsd:element name="glo-action-request" type="xsd:hexBinary"/>
<xsd:element name="glo-get-response" type="xsd:hexBinary"/>
<xsd:element name="glo-set-response" type="xsd:hexBinary"/>
<xsd:element name="glo-action-response" type="xsd:hexBinary"/>
<xsd:element name="ded-get-request" type="xsd:hexBinary"/>
<xsd:element name="ded-set-request" type="xsd:hexBinary"/>
<xsd:element name="ded-event-notification-request" type="xsd:hexBinary"/>
<xsd:element name="ded-actionRequest" type="xsd:hexBinary"/>
<xsd:element name="ded-get-response" type="xsd:hexBinary"/>
<xsd:element name="ded-set-response" type="xsd:hexBinary"/>
<xsd:element name="ded-action-response" type="xsd:hexBinary"/>
<xsd:element name="exception-response" type="ExceptionResponse"/>
<xsd:element name="access-request" type="Access-Request"/>
<xsd:element name="access-response" type="Access-Response"/>
<xsd:element name="general-glo-ciphering" type="General-Glo-Ciphering"/>
<xsd:element name="general-ded-ciphering" type="General-Ded-Ciphering"/>
<xsd:element name="general-ciphering" type="General-Ciphering"/>
<xsd:element name="general-signing" type="General-Signing"/>
<xsd:element name="general-block-transfer" type="General-Block-Transfer"/>
</xsd:choice>
</xsd:complexType>

<xsd:simpleType name="Application-context-name">
  <xsd:restriction base="ObjectIdentifier"/>
</xsd:simpleType>

<xsd:simpleType name="AP-title">
  <xsd:restriction base="xsd:hexBinary"/>
</xsd:simpleType>

<xsd:simpleType name="AE-qualifier">
  <xsd:restriction base="xsd:hexBinary"/>
</xsd:simpleType>

<xsd:simpleType name="AP-invocation-identifier">
  <xsd:restriction base="xsd:integer"/>
</xsd:simpleType>

<xsd:simpleType name="AE-invocation-identifier">
  <xsd:restriction base="xsd:integer"/>
</xsd:simpleType>

<xsd:simpleType name="ACSE-requirements">
  <xsd:union memberTypes="BitString">
    <xsd:simpleType>
      <xsd:list>
        <xsd:simpleType>
          <xsd:restriction base="xsd:token">
            <xsd:enumeration value="authentication"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:list>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>

<xsd:simpleType name="Mechanism-name">
  <xsd:restriction base="ObjectIdentifier"/>
</xsd:simpleType>

<xsd:simpleType name="Implementation-data">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>

<xsd:simpleType name="Association-information">

```

```
<xsd:restriction base="xsd:hexBinary"/>
</xsd:simpleType>

<xsd:simpleType name="Association-result">
  <xsd:union>
    <xsd:simpleType>
      <xsd:restriction base="xsd:token">
        <xsd:enumeration value="accepted"/>
        <xsd:enumeration value="rejected-permanent"/>
        <xsd:enumeration value="rejected-transient"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType>
      <xsd:restriction base="xsd:integer"/>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>

<xsd:simpleType name="Release-request-reason">
  <xsd:union>
    <xsd:simpleType>
      <xsd:restriction base="xsd:token">
        <xsd:enumeration value="normal"/>
        <xsd:enumeration value="urgent"/>
        <xsd:enumeration value="user-defined"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType>
      <xsd:restriction base="xsd:integer"/>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>

<xsd:simpleType name="Release-response-reason">
  <xsd:union>
    <xsd:simpleType>
      <xsd:restriction base="xsd:token">
        <xsd:enumeration value="normal"/>
        <xsd:enumeration value="not-finished"/>
        <xsd:enumeration value="user-defined"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType>
      <xsd:restriction base="xsd:integer"/>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>

<xsd:simpleType name="Integer8">
  <xsd:restriction base="xsd:byte"/>
</xsd:simpleType>

<xsd:simpleType name="Integer16">
  <xsd:restriction base="xsd:short"/>
</xsd:simpleType>

<xsd:simpleType name="Integer32">
  <xsd:restriction base="xsd:int"/>
</xsd:simpleType>

<xsd:simpleType name="Integer64">
  <xsd:restriction base="xsd:long"/>
</xsd:simpleType>

<xsd:simpleType name="Unsigned8">
  <xsd:restriction base="xsd:unsignedByte"/>
</xsd:simpleType>

<xsd:simpleType name="Unsigned16">
  <xsd:restriction base="xsd:unsignedShort"/>
</xsd:simpleType>

<xsd:simpleType name="Unsigned32">
  <xsd:restriction base="xsd:unsignedInt"/>
</xsd:simpleType>
```

```

<xsd:simpleType name="Unsigned64">
  <xsd:restriction base="xsd:unsignedLong"/>
</xsd:simpleType>

<xsd:simpleType name="Conformance">
  <xsd:union memberTypes="BitString">
    <xsd:simpleType>
      <xsd:list>
        <xsd:simpleType>
          <xsd:restriction base="xsd:token">
            <xsd:enumeration value="reserved-zero"/>
            <xsd:enumeration value="general-protection"/>
            <xsd:enumeration value="general-block-transfer"/>
            <xsd:enumeration value="read"/>
            <xsd:enumeration value="write"/>
            <xsd:enumeration value="unconfirmed-write"/>
            <xsd:enumeration value="reserved-six"/>
            <xsd:enumeration value="reserved-seven"/>
            <xsd:enumeration value="attribute0-supported-with-set"/>
            <xsd:enumeration value="priority-mgmt-supported"/>
            <xsd:enumeration value="attribute0-supported-with-get"/>
            <xsd:enumeration value="block-transfer-with-get-or-read"/>
            <xsd:enumeration value="block-transfer-with-set-or-write"/>
            <xsd:enumeration value="block-transfer-with-action"/>
            <xsd:enumeration value="multiple-references"/>
            <xsd:enumeration value="information-report"/>
            <xsd:enumeration value="data-notification"/>
            <xsd:enumeration value="access"/>
            <xsd:enumeration value="parameterized-access"/>
            <xsd:enumeration value="get"/>
            <xsd:enumeration value="set"/>
            <xsd:enumeration value="selective-access"/>
            <xsd:enumeration value="event-notification"/>
            <xsd:enumeration value="action"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:list>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>

<xsd:simpleType name="ObjectName">
  <xsd:restriction base="Integer16"/>
</xsd:simpleType>

<xsd:simpleType name="Data-Access-Result">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="success"/>
    <xsd:enumeration value="hardware-fault"/>
    <xsd:enumeration value="temporary-failure"/>
    <xsd:enumeration value="read-write-denied"/>
    <xsd:enumeration value="object-undefined"/>
    <xsd:enumeration value="object-class-inconsistent"/>
    <xsd:enumeration value="object-unavailable"/>
    <xsd:enumeration value="type-unmatched"/>
    <xsd:enumeration value="scope-of-access-violated"/>
    <xsd:enumeration value="data-block-unavailable"/>
    <xsd:enumeration value="long-get-aborted"/>
    <xsd:enumeration value="no-long-get-in-progress"/>
    <xsd:enumeration value="long-set-aborted"/>
    <xsd:enumeration value="no-long-set-in-progress"/>
    <xsd:enumeration value="data-block-number-invalid"/>
    <xsd:enumeration value="other-reason"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="Action-Result">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="success"/>
    <xsd:enumeration value="hardware-fault"/>
    <xsd:enumeration value="temporary-failure"/>
    <xsd:enumeration value="read-write-denied"/>
    <xsd:enumeration value="object-undefined"/>
    <xsd:enumeration value="object-class-inconsistent"/>
    <xsd:enumeration value="object-unavailable"/>
    <xsd:enumeration value="type-unmatched"/>
  </xsd:restriction>
</xsd:simpleType>

```

```

        <xsd:enumeration value="scope-of-access-violated"/>
        <xsd:enumeration value="data-block-unavailable"/>
        <xsd:enumeration value="long-action-aborted"/>
        <xsd:enumeration value="no-long-action-in-progress"/>
        <xsd:enumeration value="other-reason"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="Invoke-Id-And-Priority">
    <xsd:restriction base="Unsigned8"/>
</xsd:simpleType>

<xsd:simpleType name="Long-Invoke-Id-And-Priority">
    <xsd:restriction base="Unsigned32"/>
</xsd:simpleType>

<xsd:simpleType name="Cosem-Class-Id">
    <xsd:restriction base="Unsigned16"/>
</xsd:simpleType>

<xsd:simpleType name="Cosem-Object-Instance-Id">
    <xsd:restriction base="xsd:hexBinary">
        <xsd:length value="6"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="Cosem-Object-Attribute-Id">
    <xsd:restriction base="Integer8"/>
</xsd:simpleType>

<xsd:simpleType name="Cosem-Object-Method-Id">
    <xsd:restriction base="Integer8"/>
</xsd:simpleType>

<xsd:simpleType name="Key-Id">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="global-unicast-encryption-key"/>
        <xsd:enumeration value="global-broadcast-encryption-key"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="Kek-Id">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="master-key"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="Block-Control">
    <xsd:restriction base="Unsigned8"/>
</xsd:simpleType>

<xsd:complexType name="Authentication-value">
    <xsd:choice>
        <xsd:element name="charstring" type="xsd:string"/>
        <xsd:element name="bitstring" type="BitString"/>
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="AARQ-apdu">
    <xsd:sequence>
        <xsd:element name="protocol-version" minOccurs="0">
            <xsd:simpleType>
                <xsd:union memberTypes="BitString">
                    <xsd:simpleType>
                        <xsd:list>
                            <xsd:simpleType>
                                <xsd:restriction base="xsd:token">
                                    <xsd:enumeration value="version1"/>
                                </xsd:restriction>
                            </xsd:simpleType>
                        </xsd:list>
                    </xsd:simpleType>
                </xsd:union>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="application-context-name" type="Application-context-name"/>
    </xsd:sequence>
</xsd:complexType>

```

```

<xsd:element name="called-AP-title" minOccurs="0" type="AP-title"/>
<xsd:element name="called-AE-qualifier" minOccurs="0" type="AE-qualifier"/>
<xsd:element name="called-AP-invocation-id" minOccurs="0" type="AP-invocation-identifier"/>
<xsd:element name="called-AE-invocation-id" minOccurs="0" type="AE-invocation-identifier"/>
<xsd:element name="calling-AP-title" minOccurs="0" type="AP-title"/>
<xsd:element name="calling-AE-qualifier" minOccurs="0" type="AE-qualifier"/>
<xsd:element name="calling-AP-invocation-id" minOccurs="0" type="AP-invocation-identifier"/>
<xsd:element name="calling-AE-invocation-id" minOccurs="0" type="AE-invocation-identifier"/>
<xsd:element name="sender-acse-requirements" minOccurs="0" type="ACSE-requirements"/>
<xsd:element name="mechanism-name" minOccurs="0" type="Mechanism-name"/>
<xsd:element name="calling-authentication-value" minOccurs="0" type="Authentication-value"/>
<xsd:element name="implementation-information" minOccurs="0" type="Implementation-data"/>
<xsd:element name="user-information" minOccurs="0" type="Association-information"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Associate-source-diagnostic">
  <xsd:choice>
    <xsd:element name="acse-service-user">
      <xsd:simpleType>
        <xsd:union>
          <xsd:simpleType>
            <xsd:restriction base="xsd:token">
              <xsd:enumeration value="null"/>
              <xsd:enumeration value="no-reason-given"/>
              <xsd:enumeration value="application-context-name-not-supported"/>
              <xsd:enumeration value="calling-AP-title-not-recognized"/>
              <xsd:enumeration value="calling-AP-invocation-identifier-not-recognized"/>
              <xsd:enumeration value="calling-AE-qualifier-not-recognized"/>
              <xsd:enumeration value="calling-AE-invocation-identifier-not-recognized"/>
              <xsd:enumeration value="called-AP-title-not-recognized"/>
              <xsd:enumeration value="called-AP-invocation-identifier-not-recognized"/>
              <xsd:enumeration value="called-AE-qualifier-not-recognized"/>
              <xsd:enumeration value="called-AE-invocation-identifier-not-recognized"/>
              <xsd:enumeration value="authentication-mechanism-name-not-recognized"/>
              <xsd:enumeration value="authentication-mechanism-name-required"/>
              <xsd:enumeration value="authentication-failure"/>
              <xsd:enumeration value="authentication-required"/>
            </xsd:restriction>
          </xsd:simpleType>
          <xsd:simpleType>
            <xsd:restriction base="xsd:integer"/>
          </xsd:simpleType>
        </xsd:union>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="acse-service-provider">
      <xsd:simpleType>
        <xsd:union>
          <xsd:simpleType>
            <xsd:restriction base="xsd:token">
              <xsd:enumeration value="null"/>
              <xsd:enumeration value="no-reason-given"/>
              <xsd:enumeration value="no-common-acse-version"/>
            </xsd:restriction>
          </xsd:simpleType>
          <xsd:simpleType>
            <xsd:restriction base="xsd:integer"/>
          </xsd:simpleType>
        </xsd:union>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="AARE-apdu">
  <xsd:sequence>
    <xsd:element name="protocol-version" minOccurs="0">
      <xsd:simpleType>
        <xsd:union memberTypes="BitString">
          <xsd:simpleType>
            <xsd:list>
              <xsd:simpleType>
                <xsd:restriction base="xsd:token">
                  <xsd:enumeration value="version1"/>
                </xsd:restriction>
              </xsd:simpleType>
            </xsd:list>
          </xsd:simpleType>
        </xsd:union>
      </xsd:simpleType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```

        </xsd:simpleType>
    </xsd:list>
    </xsd:simpleType>
</xsd:union>
</xsd:simpleType>
</xsd:element>
<xsd:element name="application-context-name" type="Application-context-name"/>
<xsd:element name="result" type="Association-result"/>
<xsd:element name="result-source-diagnostic" type="Associate-source-diagnostic"/>
<xsd:element name="responding-AP-title" minOccurs="0" type="AP-title"/>
<xsd:element name="responding-AE-qualifier" minOccurs="0" type="AE-qualifier"/>
<xsd:element name="responding-AP-invocation-id" minOccurs="0" type="AP-invocation-
identifier"/>
<xsd:element name="responding-AE-invocation-id" minOccurs="0" type="AE-invocation-
identifier"/>
<xsd:element name="responder-acse-requirements" minOccurs="0" type="ACSE-requirements"/>
<xsd:element name="mechanism-name" minOccurs="0" type="Mechanism-name"/>
<xsd:element name="responding-authentication-value" minOccurs="0" type="Authentication-
value"/>
<xsd:element name="implementation-information" minOccurs="0" type="Implementation-data"/>
<xsd:element name="user-information" minOccurs="0" type="Association-information"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="RLRQ-apdu">
    <xsd:sequence>
        <xsd:element name="reason" minOccurs="0" type="Release-request-reason"/>
        <xsd:element name="user-information" minOccurs="0" type="Association-information"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="RLRE-apdu">
    <xsd:sequence>
        <xsd:element name="reason" minOccurs="0" type="Release-response-reason"/>
        <xsd:element name="user-information" minOccurs="0" type="Association-information"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="InitiateRequest">
    <xsd:sequence>
        <xsd:element name="dedicated-key" minOccurs="0" type="xsd:hexBinary"/>
        <xsd:element name="response-allowed" default="true" type="xsd:boolean"/>
        <xsd:element name="proposed-quality-of-service" minOccurs="0" type="Integer8"/>
        <xsd:element name="proposed-dlms-version-number" type="Unsigned8"/>
        <xsd:element name="proposed-conformance" type="Conformance"/>
        <xsd:element name="client-max-receive-pdu-size" type="Unsigned16"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="TypeDescription">
    <xsd:choice>
        <xsd:element name="null-data" type="NULL"/>
        <xsd:element name="array">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="number-of-elements" type="Unsigned16"/>
                    <xsd:element name="type-description" type="TypeDescription"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="structure">
            <xsd:complexType>
                <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="TypeDescription" type="TypeDescription"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="boolean" type="NULL"/>
        <xsd:element name="bit-string" type="NULL"/>
        <xsd:element name="double-long" type="NULL"/>
        <xsd:element name="double-long-unsigned" type="NULL"/>
        <xsd:element name="octet-string" type="NULL"/>
        <xsd:element name="visible-string" type="NULL"/>
        <xsd:element name="utf8-string" type="NULL"/>
        <xsd:element name="bcd" type="NULL"/>
        <xsd:element name="integer" type="NULL"/>
    </xsd:choice>
</xsd:complexType>

```

```

<xsd:element name="long" type="NULL"/>
<xsd:element name="unsigned" type="NULL"/>
<xsd:element name="long-unsigned" type="NULL"/>
<xsd:element name="long64" type="NULL"/>
<xsd:element name="long64-unsigned" type="NULL"/>
<xsd:element name="enum" type="NULL"/>
<xsd:element name="float32" type="NULL"/>
<xsd:element name="float64" type="NULL"/>
<xsd:element name="date-time" type="NULL"/>
<xsd:element name="date" type="NULL"/>
<xsd:element name="time" type="NULL"/>
<xsd:element name="dont-care" type="NULL"/>
</xsd:choice>
</xsd:complexType>

<xsd:complexType name="SequenceOfData">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="null-data" type="NULL"/>
    <xsd:element name="array" type="SequenceOfData"/>
    <xsd:element name="structure" type="SequenceOfData"/>
    <xsd:element name="boolean" type="xsd:boolean"/>
    <xsd:element name="bit-string" type="BitString"/>
    <xsd:element name="double-long" type="Integer32"/>
    <xsd:element name="double-long-unsigned" type="Unsigned32"/>
    <xsd:element name="octet-string" type="xsd:hexBinary"/>
    <xsd:element name="visible-string" type="xsd:string"/>
    <xsd:element name="utf8-string" type="xsd:string"/>
    <xsd:element name="bcd" type="Integer8"/>
    <xsd:element name="integer" type="Integer8"/>
    <xsd:element name="long" type="Integer16"/>
    <xsd:element name="unsigned" type="Unsigned8"/>
    <xsd:element name="long-unsigned" type="Unsigned16"/>
    <xsd:element name="compact-array">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="contents-description" type="TypeDescription"/>
          <xsd:element name="array-contents" type="xsd:hexBinary"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="long64" type="Integer64"/>
    <xsd:element name="long64-unsigned" type="Unsigned64"/>
    <xsd:element name="enum" type="Unsigned8"/>
    <xsd:element name="float32" type="xsd:float"/>
    <xsd:element name="float64" type="xsd:double"/>
    <xsd:element name="date-time">
      <xsd:simpleType>
        <xsd:restriction base="xsd:hexBinary">
          <xsd:length value="12"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="date">
      <xsd:simpleType>
        <xsd:restriction base="xsd:hexBinary">
          <xsd:length value="5"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="time">
      <xsd:simpleType>
        <xsd:restriction base="xsd:hexBinary">
          <xsd:length value="4"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="dont-care" type="NULL"/>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="Data">
  <xsd:choice>
    <xsd:element name="null-data" type="NULL"/>
    <xsd:element name="array" type="SequenceOfData"/>
    <xsd:element name="structure" type="SequenceOfData"/>
    <xsd:element name="boolean" type="xsd:boolean"/>

```

```

<xsd:element name="bit-string" type="BitString"/>
<xsd:element name="double-long" type="Integer32"/>
<xsd:element name="double-long-unsigned" type="Unsigned32"/>
<xsd:element name="octet-string" type="xsd:hexBinary"/>
<xsd:element name="visible-string" type="xsd:string"/>
<xsd:element name="utf8-string" type="xsd:string"/>
<xsd:element name="bcd" type="Integer8"/>
<xsd:element name="integer" type="Integer8"/>
<xsd:element name="long" type="Integer16"/>
<xsd:element name="unsigned" type="Unsigned8"/>
<xsd:element name="long-unsigned" type="Unsigned16"/>
<xsd:element name="compact-array">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="contents-description" type="TypeDescription"/>
      <xsd:element name="array-contents" type="xsd:hexBinary"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="long64" type="Integer64"/>
<xsd:element name="long64-unsigned" type="Unsigned64"/>
<xsd:element name="enum" type="Unsigned8"/>
<xsd:element name="float32" type="xsd:float"/>
<xsd:element name="float64" type="xsd:double"/>
<xsd:element name="date-time">
  <xsd:simpleType>
    <xsd:restriction base="xsd:hexBinary">
      <xsd:length value="12"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="date">
  <xsd:simpleType>
    <xsd:restriction base="xsd:hexBinary">
      <xsd:length value="5"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="time">
  <xsd:simpleType>
    <xsd:restriction base="xsd:hexBinary">
      <xsd:length value="4"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="dont-care" type="NULL"/>
</xsd:choice>
</xsd:complexType>

<xsd:complexType name="Parameterized-Access">
  <xsd:sequence>
    <xsd:element name="variable-name" type="ObjectName"/>
    <xsd:element name="selector" type="Unsigned8"/>
    <xsd:element name="parameter" type="Data"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Block-Number-Access">
  <xsd:sequence>
    <xsd:element name="block-number" type="Unsigned16"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Read-Data-Block-Access">
  <xsd:sequence>
    <xsd:element name="last-block" type="xsd:boolean"/>
    <xsd:element name="block-number" type="Unsigned16"/>
    <xsd:element name="raw-data" type="xsd:hexBinary"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Write-Data-Block-Access">
  <xsd:sequence>
    <xsd:element name="last-block" type="xsd:boolean"/>
    <xsd:element name="block-number" type="Unsigned16"/>
  </xsd:sequence>

```

```

</xsd:complexType>

<xsd:complexType name="Variable-Access-Specification">
  <xsd:choice>
    <xsd:element name="variable-name" type="ObjectName"/>
    <xsd:element name="parameterized-access" type="Parameterized-Access"/>
    <xsd:element name="block-number-access" type="Block-Number-Access"/>
    <xsd:element name="read-data-block-access" type="Read-Data-Block-Access"/>
    <xsd:element name="write-data-block-access" type="Write-Data-Block-Access"/>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="ReadRequest">
  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Variable-Access-Specification" type="Variable-Access-Specification"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="WriteRequest">
  <xsd:sequence>
    <xsd:element name="variable-access-specification">
      <xsd:complexType>
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="Variable-Access-Specification" type="Variable-Access-
Specification"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="list-of-data">
      <xsd:complexType>
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="Data" type="Data"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="InitiateResponse">
  <xsd:sequence>
    <xsd:element name="negotiated-quality-of-service" minOccurs="0" type="Integer8"/>
    <xsd:element name="negotiated-dlms-version-number" type="Unsigned8"/>
    <xsd:element name="negotiated-conformance" type="Conformance"/>
    <xsd:element name="server-max-receive-pdu-size" type="Unsigned16"/>
    <xsd:element name="vaa-name" type="ObjectName"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Data-Block-Result">
  <xsd:sequence>
    <xsd:element name="last-block" type="xsd:boolean"/>
    <xsd:element name="block-number" type="Unsigned16"/>
    <xsd:element name="raw-data" type="xsd:hexBinary"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ReadResponse">
  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="CHOICE">
      <xsd:complexType>
        <xsd:choice>
          <xsd:element name="data" type="Data"/>
          <xsd:element name="data-access-error" type="Data-Access-Result"/>
          <xsd:element name="data-block-result" type="Data-Block-Result"/>
          <xsd:element name="block-number" type="Unsigned16"/>
        </xsd:choice>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="WriteResponse">
  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="CHOICE">
      <xsd:complexType>
        <xsd:choice>

```

```

        <xsd:element name="success" type="NULL"/>
        <xsd:element name="data-access-error" type="Data-Access-Result"/>
        <xsd:element name="block-number" type="Unsigned16"/>
    </xsd:choice>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ServiceError">
    <xsd:choice>
        <xsd:element name="application-reference">
            <xsd:simpleType>
                <xsd:restriction base="xsd:token">
                    <xsd:enumeration value="other"/>
                    <xsd:enumeration value="time-elapsed"/>
                    <xsd:enumeration value="application-unreachable"/>
                    <xsd:enumeration value="application-reference-invalid"/>
                    <xsd:enumeration value="application-context-unsupported"/>
                    <xsd:enumeration value="provider-communication-error"/>
                    <xsd:enumeration value="deciphering-error"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="hardware-resource">
            <xsd:simpleType>
                <xsd:restriction base="xsd:token">
                    <xsd:enumeration value="other"/>
                    <xsd:enumeration value="memory-unavailable"/>
                    <xsd:enumeration value="processor-resource-unavailable"/>
                    <xsd:enumeration value="mass-storage-unavailable"/>
                    <xsd:enumeration value="other-resource-unavailable"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="vde-state-error">
            <xsd:simpleType>
                <xsd:restriction base="xsd:token">
                    <xsd:enumeration value="other"/>
                    <xsd:enumeration value="no-dlms-context"/>
                    <xsd:enumeration value="loading-data-set"/>
                    <xsd:enumeration value="status-nochange"/>
                    <xsd:enumeration value="status-inoperable"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="service">
            <xsd:simpleType>
                <xsd:restriction base="xsd:token">
                    <xsd:enumeration value="other"/>
                    <xsd:enumeration value="pdu-size"/>
                    <xsd:enumeration value="service-unsupported"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="definition">
            <xsd:simpleType>
                <xsd:restriction base="xsd:token">
                    <xsd:enumeration value="other"/>
                    <xsd:enumeration value="object-undefined"/>
                    <xsd:enumeration value="object-class-inconsistent"/>
                    <xsd:enumeration value="object-attribute-inconsistent"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="access">
            <xsd:simpleType>
                <xsd:restriction base="xsd:token">
                    <xsd:enumeration value="other"/>
                    <xsd:enumeration value="scope-of-access-violated"/>
                    <xsd:enumeration value="object-access-violated"/>
                    <xsd:enumeration value="hardware-fault"/>
                    <xsd:enumeration value="object-unavailable"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
    </xsd:choice>
</xsd:complexType>

```

```

    <xsd:element name="initiate">
      <xsd:simpleType>
        <xsd:restriction base="xsd:token">
          <xsd:enumeration value="other"/>
          <xsd:enumeration value="dlms-version-too-low"/>
          <xsd:enumeration value="incompatible-conformance"/>
          <xsd:enumeration value="pdu-size-too-short"/>
          <xsd:enumeration value="refused-by-the-VDE-Handler"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="load-data-set">
      <xsd:simpleType>
        <xsd:restriction base="xsd:token">
          <xsd:enumeration value="other"/>
          <xsd:enumeration value="primitive-out-of-sequence"/>
          <xsd:enumeration value="not-loadable"/>
          <xsd:enumeration value="dataset-size-too-large"/>
          <xsd:enumeration value="not-awaited-segment"/>
          <xsd:enumeration value="interpretation-failure"/>
          <xsd:enumeration value="storage-failure"/>
          <xsd:enumeration value="data-set-not-ready"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="task">
      <xsd:simpleType>
        <xsd:restriction base="xsd:token">
          <xsd:enumeration value="other"/>
          <xsd:enumeration value="no-remote-control"/>
          <xsd:enumeration value="ti-stopped"/>
          <xsd:enumeration value="ti-running"/>
          <xsd:enumeration value="ti-unusable"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="ConfirmedServiceError">
  <xsd:choice>
    <xsd:element name="initiateError" type="ServiceError"/>
    <xsd:element name="getStatus" type="ServiceError"/>
    <xsd:element name="getNameList" type="ServiceError"/>
    <xsd:element name="getVariableAttribute" type="ServiceError"/>
    <xsd:element name="read" type="ServiceError"/>
    <xsd:element name="write" type="ServiceError"/>
    <xsd:element name="getDataSetAttribute" type="ServiceError"/>
    <xsd:element name="getTIAttribute" type="ServiceError"/>
    <xsd:element name="changeScope" type="ServiceError"/>
    <xsd:element name="start" type="ServiceError"/>
    <xsd:element name="stop" type="ServiceError"/>
    <xsd:element name="resume" type="ServiceError"/>
    <xsd:element name="makeUsable" type="ServiceError"/>
    <xsd:element name="initiateLoad" type="ServiceError"/>
    <xsd:element name="loadSegment" type="ServiceError"/>
    <xsd:element name="terminateLoad" type="ServiceError"/>
    <xsd:element name="initiateUpload" type="ServiceError"/>
    <xsd:element name="uploadSegment" type="ServiceError"/>
    <xsd:element name="terminateUpload" type="ServiceError"/>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="Notification-Body">
  <xsd:sequence>
    <xsd:element name="data-value" type="Data"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Data-Notification">
  <xsd:sequence>
    <xsd:element name="long-invoke-id-and-priority" type="Long-Invoke-Id-And-Priority"/>
    <xsd:element name="date-time" type="xsd:hexBinary"/>
    <xsd:element name="notification-body" type="Notification-Body"/>
  </xsd:sequence>
</xsd:complexType>

```

```

<xsd:complexType name="UnconfirmedWriteRequest">
  <xsd:sequence>
    <xsd:element name="variable-access-specification">
      <xsd:complexType>
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="Variable-Access-Specification" type="Variable-Access-
Specification"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="list-of-data">
      <xsd:complexType>
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="Data" type="Data"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="InformationReportRequest">
  <xsd:sequence>
    <xsd:element name="current-time" minOccurs="0" type="xsd:dateTime"/>
    <xsd:element name="variable-access-specification">
      <xsd:complexType>
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="Variable-Access-Specification" type="Variable-Access-
Specification"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="list-of-data">
      <xsd:complexType>
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="Data" type="Data"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Cosem-Attribute-Descriptor">
  <xsd:sequence>
    <xsd:element name="class-id" type="Cosem-Class-Id"/>
    <xsd:element name="instance-id" type="Cosem-Object-Instance-Id"/>
    <xsd:element name="attribute-id" type="Cosem-Object-Attribute-Id"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Selective-Access-Descriptor">
  <xsd:sequence>
    <xsd:element name="access-selector" type="Unsigned8"/>
    <xsd:element name="access-parameters" type="Data"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Get-Request-Normal">
  <xsd:sequence>
    <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
    <xsd:element name="cosem-attribute-descriptor" type="Cosem-Attribute-Descriptor"/>
    <xsd:element name="access-selection" minOccurs="0" type="Selective-Access-Descriptor"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Get-Request-Next">
  <xsd:sequence>
    <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
    <xsd:element name="block-number" type="Unsigned32"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Cosem-Attribute-Descriptor-With-Selection">
  <xsd:sequence>
    <xsd:element name="cosem-attribute-descriptor" type="Cosem-Attribute-Descriptor"/>
    <xsd:element name="access-selection" minOccurs="0" type="Selective-Access-Descriptor"/>
  </xsd:sequence>
</xsd:complexType>

```

```

    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Get-Request-With-List">
    <xsd:sequence>
      <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
      <xsd:element name="attribute-descriptor-list">
        <xsd:complexType>
          <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="Cosem-Attribute-Descriptor-With-Selection" type="Cosem-Attribute-Descriptor-With-Selection"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Get-Request">
    <xsd:choice>
      <xsd:element name="get-request-normal" type="Get-Request-Normal"/>
      <xsd:element name="get-request-next" type="Get-Request-Next"/>
      <xsd:element name="get-request-with-list" type="Get-Request-With-List"/>
    </xsd:choice>
  </xsd:complexType>

  <xsd:complexType name="Set-Request-Normal">
    <xsd:sequence>
      <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
      <xsd:element name="cosem-attribute-descriptor" type="Cosem-Attribute-Descriptor"/>
      <xsd:element name="access-selection" minOccurs="0" type="Selective-Access-Descriptor"/>
      <xsd:element name="value" type="Data"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="DataBlock-SA">
    <xsd:sequence>
      <xsd:element name="last-block" type="xsd:boolean"/>
      <xsd:element name="block-number" type="Unsigned32"/>
      <xsd:element name="raw-data" type="xsd:hexBinary"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Set-Request-With-First-Datablock">
    <xsd:sequence>
      <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
      <xsd:element name="cosem-attribute-descriptor" type="Cosem-Attribute-Descriptor"/>
      <xsd:element name="access-selection" minOccurs="0" type="Selective-Access-Descriptor"/>
      <xsd:element name="datablock" type="DataBlock-SA"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Set-Request-With-Datablock">
    <xsd:sequence>
      <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
      <xsd:element name="datablock" type="DataBlock-SA"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Set-Request-With-List">
    <xsd:sequence>
      <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
      <xsd:element name="attribute-descriptor-list">
        <xsd:complexType>
          <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="Cosem-Attribute-Descriptor-With-Selection" type="Cosem-Attribute-Descriptor-With-Selection"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="value-list">
        <xsd:complexType>
          <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="Data" type="Data"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

```

```

    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Set-Request-With-List-And-First-Datablock">
    <xsd:sequence>
      <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
      <xsd:element name="attribute-descriptor-list">
        <xsd:complexType>
          <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="Cosem-Attribute-Descriptor-With-Selection" type="Cosem-Attribute-Descriptor-With-Selection"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="datablock" type="DataBlock-SA"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Set-Request">
    <xsd:choice>
      <xsd:element name="set-request-normal" type="Set-Request-Normal"/>
      <xsd:element name="set-request-with-first-datablock" type="Set-Request-With-First-Datablock"/>
      <xsd:element name="set-request-with-datablock" type="Set-Request-With-Datablock"/>
      <xsd:element name="set-request-with-list" type="Set-Request-With-List"/>
      <xsd:element name="set-request-with-list-and-first-datablock" type="Set-Request-With-List-And-First-Datablock"/>
    </xsd:choice>
  </xsd:complexType>

  <xsd:complexType name="EventNotificationRequest">
    <xsd:sequence>
      <xsd:element name="time" minOccurs="0" type="xsd:hexBinary"/>
      <xsd:element name="cosem-attribute-descriptor" type="Cosem-Attribute-Descriptor"/>
      <xsd:element name="attribute-value" type="Data"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Cosem-Method-Descriptor">
    <xsd:sequence>
      <xsd:element name="class-id" type="Cosem-Class-Id"/>
      <xsd:element name="instance-id" type="Cosem-Object-Instance-Id"/>
      <xsd:element name="method-id" type="Cosem-Object-Method-Id"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Action-Request-Normal">
    <xsd:sequence>
      <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
      <xsd:element name="cosem-method-descriptor" type="Cosem-Method-Descriptor"/>
      <xsd:element name="method-invocation-parameters" minOccurs="0" type="Data"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Action-Request-Next-Pblock">
    <xsd:sequence>
      <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
      <xsd:element name="block-number" type="Unsigned32"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Action-Request-With-List">
    <xsd:sequence>
      <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
      <xsd:element name="cosem-method-descriptor-list">
        <xsd:complexType>
          <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="Cosem-Method-Descriptor" type="Cosem-Method-Descriptor"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="method-invocation-parameters">
        <xsd:complexType>
          <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="Data" type="Data"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

```

```

        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Action-Request-With-First-Pblock">
      <xsd:sequence>
        <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
        <xsd:element name="cosem-method-descriptor" type="Cosem-Method-Descriptor"/>
        <xsd:element name="pblock" type="DataBlock-SA"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Action-Request-With-List-And-First-Pblock">
      <xsd:sequence>
        <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
        <xsd:element name="cosem-method-descriptor-list">
          <xsd:complexType>
            <xsd:sequence minOccurs="0" maxOccurs="unbounded">
              <xsd:element name="Cosem-Method-Descriptor" type="Cosem-Method-Descriptor"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="pblock" type="DataBlock-SA"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Action-Request-With-Pblock">
      <xsd:sequence>
        <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
        <xsd:element name="pblock" type="DataBlock-SA"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Action-Request">
      <xsd:choice>
        <xsd:element name="action-request-normal" type="Action-Request-Normal"/>
        <xsd:element name="action-request-next-pblock" type="Action-Request-Next-Pblock"/>
        <xsd:element name="action-request-with-list" type="Action-Request-With-List"/>
        <xsd:element name="action-request-with-first-pblock" type="Action-Request-With-First-Pblock"/>
        <xsd:element name="action-request-with-list-and-first-pblock" type="Action-Request-With-List-
And-First-Pblock"/>
        <xsd:element name="action-request-with-pblock" type="Action-Request-With-Pblock"/>
      </xsd:choice>
    </xsd:complexType>

    <xsd:complexType name="Get-Data-Result">
      <xsd:choice>
        <xsd:element name="data" type="Data"/>
        <xsd:element name="data-access-result" type="Data-Access-Result"/>
      </xsd:choice>
    </xsd:complexType>

    <xsd:complexType name="Get-Response-Normal">
      <xsd:sequence>
        <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
        <xsd:element name="result" type="Get-Data-Result"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="DataBlock-G">
      <xsd:sequence>
        <xsd:element name="last-block" type="xsd:boolean"/>
        <xsd:element name="block-number" type="Unsigned32"/>
        <xsd:element name="result">
          <xsd:complexType>
            <xsd:choice>
              <xsd:element name="raw-data" type="xsd:hexBinary"/>
              <xsd:element name="data-access-result" type="Data-Access-Result"/>
            </xsd:choice>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Get-Response-With-Datablock">
      <xsd:sequence>

```

```

        <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
        <xsd:element name="result" type="DataBlock-G"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Get-Response-With-List">
    <xsd:sequence>
        <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
        <xsd:element name="result">
            <xsd:complexType>
                <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="Get-Data-Result" type="Get-Data-Result"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Get-Response">
    <xsd:choice>
        <xsd:element name="get-response-normal" type="Get-Response-Normal"/>
        <xsd:element name="get-response-with-datablock" type="Get-Response-With-Datablock"/>
        <xsd:element name="get-response-with-list" type="Get-Response-With-List"/>
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="Set-Response-Normal">
    <xsd:sequence>
        <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
        <xsd:element name="result" type="Data-Access-Result"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Set-Response-Datablock">
    <xsd:sequence>
        <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
        <xsd:element name="block-number" type="Unsigned32"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Set-Response-Last-Datablock">
    <xsd:sequence>
        <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
        <xsd:element name="result" type="Data-Access-Result"/>
        <xsd:element name="block-number" type="Unsigned32"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Set-Response-Last-Datablock-With-List">
    <xsd:sequence>
        <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
        <xsd:element name="result">
            <xsd:simpleType>
                <xsd:list itemType="Data-Access-Result"/>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="block-number" type="Unsigned32"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Set-Response-With-List">
    <xsd:sequence>
        <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
        <xsd:element name="result">
            <xsd:simpleType>
                <xsd:list itemType="Data-Access-Result"/>
            </xsd:simpleType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Set-Response">
    <xsd:choice>
        <xsd:element name="set-response-normal" type="Set-Response-Normal"/>
        <xsd:element name="set-response-datablock" type="Set-Response-Datablock"/>
        <xsd:element name="set-response-last-datablock" type="Set-Response-Last-Datablock"/>
    </xsd:choice>

```

```

        <xsd:element name="set-response-last-datablock-with-list" type="Set-Response-Last-Datablock-
With-List"/>
        <xsd:element name="set-response-with-list" type="Set-Response-With-List"/>
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="Action-Response-With-Optional-Data">
    <xsd:sequence>
        <xsd:element name="result" type="Action-Result"/>
        <xsd:element name="return-parameters" minOccurs="0" type="Get-Data-Result"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Action-Response-Normal">
    <xsd:sequence>
        <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
        <xsd:element name="single-response" type="Action-Response-With-Optional-Data"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Action-Response-With-Pblock">
    <xsd:sequence>
        <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
        <xsd:element name="pblock" type="DataBlock-SA"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Action-Response-With-List">
    <xsd:sequence>
        <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
        <xsd:element name="list-of-responses">
            <xsd:complexType>
                <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="Action-Response-With-Optional-Data" type="Action-Response-With-
Optional-Data"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Action-Response-Next-Pblock">
    <xsd:sequence>
        <xsd:element name="invoke-id-and-priority" type="Invoke-Id-And-Priority"/>
        <xsd:element name="block-number" type="Unsigned32"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Action-Response">
    <xsd:choice>
        <xsd:element name="action-response-normal" type="Action-Response-Normal"/>
        <xsd:element name="action-response-with-pblock" type="Action-Response-With-Pblock"/>
        <xsd:element name="action-response-with-list" type="Action-Response-With-List"/>
        <xsd:element name="action-response-next-pblock" type="Action-Response-Next-Pblock"/>
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="ExceptionResponse">
    <xsd:sequence>
        <xsd:element name="state-error">
            <xsd:simpleType>
                <xsd:restriction base="xsd:token">
                    <xsd:enumeration value="service-not-allowed"/>
                    <xsd:enumeration value="service-unknown"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="service-error">
            <xsd:simpleType>
                <xsd:restriction base="xsd:token">
                    <xsd:enumeration value="operation-not-possible"/>
                    <xsd:enumeration value="service-not-supported"/>
                    <xsd:enumeration value="other-reason"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

```

```

    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Access-Request-Get">
    <xsd:sequence>
      <xsd:element name="cosem-attribute-descriptor" type="Cosem-Attribute-Descriptor"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Access-Request-Set">
    <xsd:sequence>
      <xsd:element name="cosem-attribute-descriptor" type="Cosem-Attribute-Descriptor"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Access-Request-Action">
    <xsd:sequence>
      <xsd:element name="cosem-method-descriptor" type="Cosem-Method-Descriptor"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Access-Request-Get-With-Selection">
    <xsd:sequence>
      <xsd:element name="cosem-attribute-descriptor" type="Cosem-Attribute-Descriptor"/>
      <xsd:element name="access-selection" type="Selective-Access-Descriptor"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Access-Request-Set-With-Selection">
    <xsd:sequence>
      <xsd:element name="cosem-attribute-descriptor" type="Cosem-Attribute-Descriptor"/>
      <xsd:element name="access-selection" type="Selective-Access-Descriptor"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Access-Request-Specification">
    <xsd:choice>
      <xsd:element name="access-request-get" type="Access-Request-Get"/>
      <xsd:element name="access-request-set" type="Access-Request-Set"/>
      <xsd:element name="access-request-action" type="Access-Request-Action"/>
      <xsd:element name="access-request-get-with-selection" type="Access-Request-Get-With-Selection"/>
      <xsd:element name="access-request-set-with-selection" type="Access-Request-Set-With-Selection"/>
    </xsd:choice>
  </xsd:complexType>

  <xsd:complexType name="List-Of-Access-Request-Specification">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="Access-Request-Specification" type="Access-Request-Specification"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="List-Of-Data">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="Data" type="Data"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Access-Request-Body">
    <xsd:sequence>
      <xsd:element name="access-request-specification" type="List-Of-Access-Request-Specification"/>
      <xsd:element name="access-request-list-of-data" type="List-Of-Data"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Access-Request">
    <xsd:sequence>
      <xsd:element name="long-invoke-id-and-priority" type="Long-Invoke-Id-And-Priority"/>
      <xsd:element name="date-time" type="xsd:hexBinary"/>
      <xsd:element name="access-request-body" type="Access-Request-Body"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Access-Response-Get">
    <xsd:sequence>

```

```

        <xsd:element name="result" type="Data-Access-Result"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Access-Response-Set">
      <xsd:sequence>
        <xsd:element name="result" type="Data-Access-Result"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Access-Response-Action">
      <xsd:sequence>
        <xsd:element name="result" type="Action-Result"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Access-Response-Specification">
      <xsd:choice>
        <xsd:element name="access-response-get" type="Access-Response-Get"/>
        <xsd:element name="access-response-set" type="Access-Response-Set"/>
        <xsd:element name="access-response-action" type="Access-Response-Action"/>
      </xsd:choice>
    </xsd:complexType>

    <xsd:complexType name="List-Of-Access-Response-Specification">
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Access-Response-Specification" type="Access-Response-Specification"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Access-Response-Body">
      <xsd:sequence>
        <xsd:element name="access-request-specification" minOccurs="0" type="List-Of-Access-Request-
Specification"/>
        <xsd:element name="access-response-list-of-data" type="List-Of-Data"/>
        <xsd:element name="access-response-specification" type="List-Of-Access-Response-
Specification"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Access-Response">
      <xsd:sequence>
        <xsd:element name="long-invoke-id-and-priority" type="Long-Invoke-Id-And-Priority"/>
        <xsd:element name="date-time" type="xsd:hexBinary"/>
        <xsd:element name="access-response-body" type="Access-Response-Body"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="General-Glo-Ciphering">
      <xsd:sequence>
        <xsd:element name="system-title" type="xsd:hexBinary"/>
        <xsd:element name="ciphered-content" type="xsd:hexBinary"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="General-Ded-Ciphering">
      <xsd:sequence>
        <xsd:element name="system-title" type="xsd:hexBinary"/>
        <xsd:element name="ciphered-content" type="xsd:hexBinary"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Identified-Key">
      <xsd:sequence>
        <xsd:element name="key-id" type="Key-Id"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Wrapped-Key">
      <xsd:sequence>
        <xsd:element name="kek-id" type="Kek-Id"/>
        <xsd:element name="key-ciphered-data" type="xsd:hexBinary"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Agreed-Key">

```

```
<xsd:sequence>
  <xsd:element name="key-parameters" type="xsd:hexBinary"/>
  <xsd:element name="key-ciphered-data" type="xsd:hexBinary"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Key-Info">
  <xsd:choice>
    <xsd:element name="identified-key" type="Identified-Key"/>
    <xsd:element name="wrapped-key" type="Wrapped-Key"/>
    <xsd:element name="agreed-key" type="Agreed-Key"/>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="General-Ciphering">
  <xsd:sequence>
    <xsd:element name="transaction-id" type="xsd:hexBinary"/>
    <xsd:element name="originator-system-title" type="xsd:hexBinary"/>
    <xsd:element name="recipient-system-title" type="xsd:hexBinary"/>
    <xsd:element name="date-time" type="xsd:hexBinary"/>
    <xsd:element name="other-information" type="xsd:hexBinary"/>
    <xsd:element name="key-info" minOccurs="0" type="Key-Info"/>
    <xsd:element name="ciphered-content" type="xsd:hexBinary"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="General-Signing">
  <xsd:sequence>
    <xsd:element name="transaction-id" type="xsd:hexBinary"/>
    <xsd:element name="originator-system-title" type="xsd:hexBinary"/>
    <xsd:element name="recipient-system-title" type="xsd:hexBinary"/>
    <xsd:element name="date-time" type="xsd:hexBinary"/>
    <xsd:element name="other-information" type="xsd:hexBinary"/>
    <xsd:element name="content" type="xsd:hexBinary"/>
    <xsd:element name="signature" type="xsd:hexBinary"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="General-Block-Transfer">
  <xsd:sequence>
    <xsd:element name="block-control" type="Block-Control"/>
    <xsd:element name="block-number" type="Unsigned16"/>
    <xsd:element name="block-number-ack" type="Unsigned16"/>
    <xsd:element name="block-data" type="xsd:hexBinary"/>
  </xsd:sequence>
</xsd:complexType>

</xsd:schema>
```

A.2 XML schema for the payload definitions

A.2.1 DeviceAccess

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns="http://iec.ch/TC13/2014/schema/DeviceAccess"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:co="http://www.dlms.com/COSEmpdu"
  xmlns:ns1="http://iec.ch/TC13/2014/schema/DeviceAccess"
  targetNamespace="http://iec.ch/TC13/2014/schema/DeviceAccess"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://www.dlms.com/COSEmpdu" schemaLocation="COSEmpdu-GB8.xsd"/>
  <xs:simpleType name="octetString">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:simpleType name="Unsigned8">
    <xs:restriction base="xs:unsignedByte"/>
  </xs:simpleType>
  <xs:simpleType name="Unsigned16">
    <xs:restriction base="xs:unsignedShort"/>
  </xs:simpleType>
  <xs:complexType name="errorResult">
    <xs:sequence>
      <xs:element name="ErrorIdent"/>
      <xs:element name="ErrorDetail" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="cosemAccessResult">
    <xs:choice>
      <xs:element name="ACSEResult" type="co:ACSE-APDU"/>
      <xs:element name="XDLMResult" type="co:XDLM-APDU"/>
      <xs:element name="ErrorResult" type="errorResult"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="cosemObjectReference">
    <xs:sequence>
      <xs:element name="ClassId" type="Unsigned16"/>
      <xs:element name="InstanceId" type="octetString"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="cosemAccessDescriptor">
    <xs:choice>
      <xs:element name="ACSEDescriptor" type="co:ACSE-APDU"/>
      <xs:element name="XDLMDescriptor" type="co:XDLM-APDU"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="cosemAccess">
    <xs:sequence>
      <xs:element name="CosemAccessDescriptor" type="cosemAccessDescriptor"/>
      <xs:element name="CosemAccessResult" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="cosemAccessResult">
              <xs:attribute name="DeviceID" type="xs:string"/>
              <xs:attribute name="Activated" type="xs:dateTime"/>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="deviceReference">
    <xs:attribute name="DeviceID" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="groupReference">
    <xs:attribute name="GroupID" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="deviceAccess">
    <xs:sequence>
      <xs:element name="DevicesReferenceList">
        <xs:complexType>
          <xs:choice>
            <xs:element name="DeviceReference" type="deviceReference" minOccurs="0"
              maxOccurs="unbounded"/>

```

```

        <xs:element name="GroupReference" type="groupReference" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:choice>
</xs:complexType>
</xs:element>
<xs:element name="CosemAccessList">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="CosemAccess" type="cosemAccess" minOccurs="0"
                maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Activates" type="xs:dateTime" minOccurs="0"/>
<xs:element name="Expires" type="xs:dateTime" minOccurs="0"/>
<xs:element name="Created" type="xs:dateTime" minOccurs="0"/>
<xs:element name="Updated" type="xs:dateTime" minOccurs="0"/>
<xs:element name="Priority" type="xs:unsignedByte" minOccurs="0"/>
<xs:element name="NotOlderThan" type="xs:dateTime" minOccurs="0"/>
<xs:element name="AcceptDataNotification" type="xs:boolean" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="ID" type="xs:string" use="optional"/>
</xs:complexType>
<xs:element name="DeviceAccess" type="deviceAccess">
    <xs:annotation>
        <xs:documentation>DeviceAccess provides access to Devices registered on the NNAP
        </xs:documentation>
    </xs:annotation>
</xs:element>
</xs:schema>

```

A.2.2 DeviceGroup

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
    <xs:complexType name="deviceReference">
        <xs:attribute name="DeviceID" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:complexType name="deviceGroup">
        <xs:sequence>
            <xs:element name="DeviceReference" type="deviceReference" minOccurs="0"
                maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="GroupID" type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="deviceGroups">
        <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:element name="DeviceGroup" type="deviceGroup"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="DeviceGroups" type="deviceGroups"/>
</xs:schema>

```

A.3 XML Schema for common message envelope

This XML schema is used to define a common message envelope (CME) for request, response, event and fault messages.

The XML Schema for a common message envelope is provided in Annex A of IEC 61968-100:2013.

A.4 WSDL for interface using SOAP

This WSDL is used to define a generic interface using SOAP. The WSDL is not strongly typed to a specific verb and noun of the common message envelope (CME) for request, response, event and fault messages.

The procedure for strongly typed WSDL generation is provided in Annex C of IEC 61968-100:2013.

The WSDL is provided by Annex D of IEC 61968-100:2013.

Bibliography

ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times*

ISO/IEC 9834-8:2014, *Information technology – Procedures for the operation of object identifier registration authorities – Part 8: Generation of universally unique identifiers (UUIDs) and their use in object identifiers*

IETF RFC 2616, *Hyper-Text Transport Protocol 1.1*

IETF RFC 4122, *A Universally Unique Identifier (UUID) URN Namespace*

Extensible Markup Language (XML): <http://www.w3.org/TR/REC-xml>

XML Schema: <http://www.w3.org/XML/Schema>

Simple Object Access Protocol (SOAP) 1.2: <http://www.w3.org/TR/soap12-part1/>

Web Services Description Language (WSDL) Version 2.0: <http://www.w3.org/TR/wsdl20>

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch