DCC User Gateway Interface Design Specification

Annex - Service Request Definitions

18 – Parse Output

Author: DCC
Version: v5.2a
Date: June 2023

Contents

18 Parse Output

18.1 Introduction

This document contains description of the XML schema used for the DCC Service User Parse Output, which is the format in which the Parse software returns the interpretation of GBCS payload to DCC Service Users. There is an accompanying XML schema, the DCC Service User Parse Output XML Schema, referred to more briefly as the MMC XML Schema. Please note that this is a separate XML Schema to the DUIS XML Schema used for Service Requests and responses, as defined in Appendix 2 of the main document. They are referred to as follows:

- The DUIS XML Schema XSD (document 3 of this documentation set);
- the MMC XML Schema XSD (document 4 of this documentation set).

The DCC Service User Parse Output is used in the interaction between DCC Service User systems and the Parse software, after responses have been sent by the DSP to the DCC Service User. Messages sent by the DSP which were not originated by a Device will not use the DCC Service User Parse Output.

SMETS1 Service Responses and Device Alerts do not use the DCC Service Parse Output, but do use the MMC XML Schema. Because of this, some changes have been made to the MMC XML Schema v3.0 and to this document where common parts are affected.

This Annex section 18 describe the use of MMC XML for data from SMETS2 or later Devices. See Annex section 19 for the use of MMC XML for data from SMETS1 Devices.

This section and its contents shall only be used by DCC Service Users who choose to use the Parse Software as provided by the DCC. If a DCC Service User chooses not to use the Parse Software then the contents of this Annex 18 will not be applicable and can be ignored.

18.2 Context

The Parse Output is used to enable an XML representation of data in the GBCS payload XML element of the Service Response from a Device. It is used in the interaction between DCC Service User systems and the Parse software, after Service Responses and Device Alerts containing GBCS payload have been sent by the DSP to the DCC Service User.

The main document section 2.2 describes the context in which the Parse software is used.

The Device data returned by the Parse Output should be in conformance with SMETS definitions.

18.3 XML Schema

The XML Schema used for the Parse output format, known as the MMC XML Schema, defines the format of XML responses and Device Alerts which will be returned by Parse software. The MMC XML Schema imports data types from the DUIS XML schema.

The XML representation of the GBCS payload in response to each Service Request is normally based on the same name as the corresponding Service Request in the DUIS XML schema, with the addition of the suffix “Rsp” at the end, for example UpdateMeterBalance and UpdateMeterBalanceRsp.

Service Requests for which there is no GBCS payload response from a Device, e.g. DCC-only Service Requests, will not have a corresponding response in the MMC XML Schema.

Device Alerts are also passed from Parse software to the users in conformance to the MMC XML Schema, as described in section 18.4.3.

The namespace defined and used within the MMC XML Schema is “ra”. From MMC v3.0, the MMC XML Schema XSD no longer imports XML types from the DUIS XML Schema XSD, and the XML types which were formerly imported in that way have been duplicated within the MMC

XML Schema XSD. This means that for elements where the namespace was previously defined as “sr”, from MMC v3.0 they will have the namespace “ra”, but will be otherwise identical.

Where those XML types duplicated within MMC are featured in SRV descriptions, for references to further details the DUGIDS annex documents continue to refer to sections within the DUGIDS document set that describe the equivalent DUIS XML definitions.

18.4 XML High-Level Response Structure

The XML which carries the data extracted from the GBCS payload sent by a Device consists of a header which is in common between Service Responses and Device Alerts, and a body which will be either Service Response data or Device Alert data, structured by different types according to Service Request Variant or Device Alert Code.

DCC-only Service Responses and DCC Alerts are not sent as GBCS and therefore will not be returned by Parse software.

There is no MAC or signature information included in these headers. MAC and/or signature information appears in Service Responses for use where needed, e.g. for the DCC Service User to carry out integrity checks on Service Responses sent by the DCC, but do not need to be in the MMC XML Schema.

The top-level structure is shown in the following diagram.

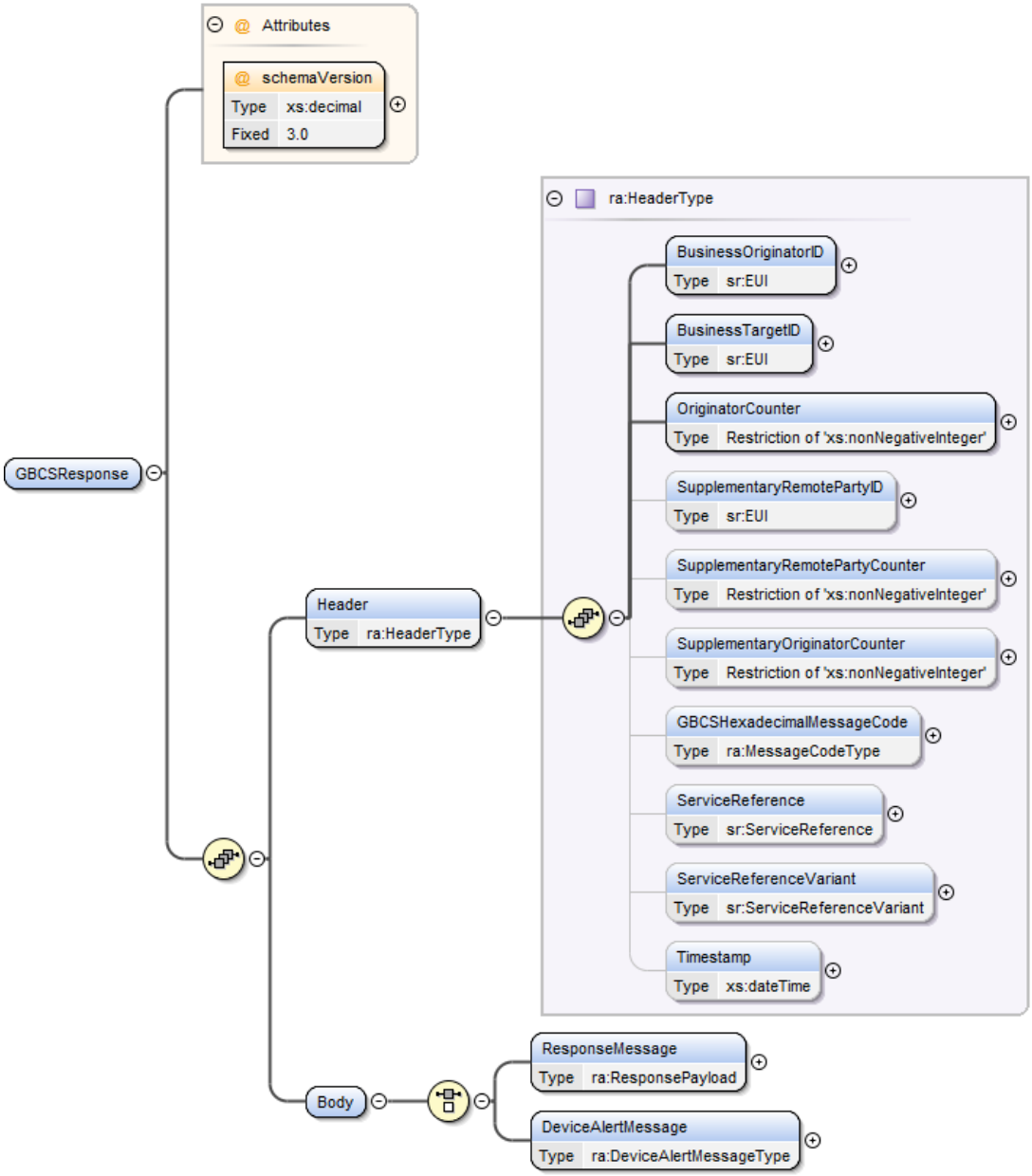

Figure 1 High-level response structure

As can be seen from the diagram above, the top-level element is of XML type GBCSResponse and contains a header and a body. The body is a choice between elements of the one of the following XML types:

- ResponseMessage. This is used to hold an XML representation of data returned in the GBCSPayload of a Service Response, as sent to the DCC by a Device, and which has then been passed to the Parse Software by the DCC Service User.
- DeviceAlertMessage. This is used to hold an XML representation of data which has been sent by a Device to the DCC as a Device Alert and which has then been passed to the Parse Software by the DCC Service User.

The header also contains information extracted from the GBCS payload, which in some cases will be a duplication of information in the Service Response that contained the GBCS payload.

Thee header and the body are described in the following sections.

18.4.1 Header

When a message is sent to a DCC Service User by the DSP, based on a GBCS message sent by a Device containing data intended for a DCC Service User, there will be header information in the Service Response, and additional header information will be contained within the GBCS payload. The header information described here is the header data within the GBCS payload sent by the Device, unpacked into an XML document conforming to the MMC XML Schema.

The Originator/Target relationship of the GBCS message will be between the DCC Service User and Device in most cases, except for cases where the DCC Service User is an Unknown Remote Party to the Device, or the request is DSP-scheduled, in which case it will be between the DSP and the Device.

See main document section 4 for illustrations of which Business Originator ID, Business Target ID and Originator Counter are used inside the GBCS payload in different circumstances. The ID of the DCC Service User for which the Service Response is intended will be in the SupplementaryRemotePartyID field in cases as listed in GBCS section 4.3.1.4, and in the BusinessTargetID field otherwise.

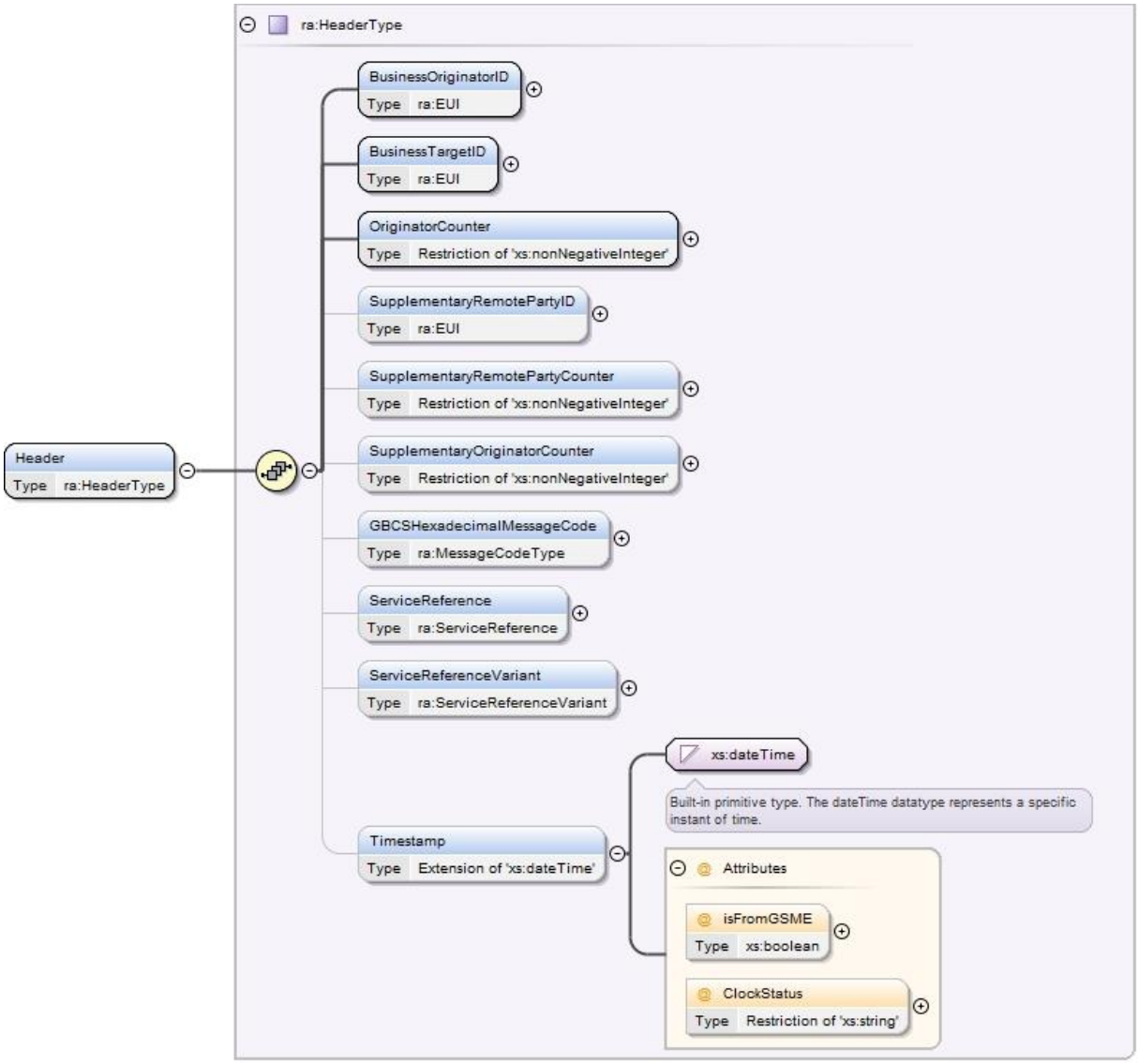

Figure 2 Header response structure

The XML header structure includes the following fields:

Data Item	Description	Type	Mandatory	Valid Values
BusinessOriginatorID	The Device ID of the Device which sent the GBCS payload. See main document section 4 for usage	ra:EUI	Yes	EUI-64 value, formatted in 8 octets (an octet is two hex digits) with a "-" as a separator, for example "AA-22-33-44-55-66-77-88". It is case insensitive
BusinessTargetID	In some cases as listed in GBCS 4.3.1.4 this will be the ID of the DSP Access Control Broker, and in most cases it will be the ID of the DCC Service User for which the GBCS payload is intended. See main document section 4 for usage	ra:EUI	Yes	EUI-64 value, formatted in 8 octets (an octet is two hex digits) with a "-" as a separator, for example "AA-BB-CC-DD-EE-FF-11-22". It is case insensitive
OriginatorCounter	For Service Responses the Originator Counter for this message from the DCC Service User which sent the GBCS payload (or the DSP Access Control Broker in cases where the DCC Service User is an Unknown Remote Party to the Device). See main document section 4 for usage. For Device Alerts it will be the Device's Originator Counter.	xs:nonNegativeInteger	Yes	>= 0 and < 2^{64}, e.g. 1234
SupplementaryRemotePartyID	For responses this will refer to the DCC Service User in cases where the GBCS command to the Device was originated by the DSP. It is not used otherwise. See GBCS 4.3.1.4. For Device Alerts it will be the DCC Service User ID of the second party to the Device Alert if there is one.	ra:EUI	No	EUI-64 value, formatted in 8 octets (an octet is two hex digits) with a "-" as a separator, for example "AA-22-33-44-55-66-77-88". It is case insensitive
SupplementaryRemotePartyCounter	For responses this will refer to the DCC Service User in cases where the GBCS command to the Device was originated by the DSP. It is not used otherwise. See GBCS 4.3.1.4.	xs:nonNegativeInteger	No	>= 0 and < 2^{64}, e.g. 1234
SupplementaryOriginatorCounter	This will be used for particular GBCS use cases in the circumstances listed in GBCS 4.3.1.4. It is not used otherwise. See GBCS 4.3.1.4.	xs:nonNegativeInteger	No	>= 0 and < 2^{64}, e.g. 1234

Data Item	Description	Type	Mandatory	Valid Values
GBCSHexadecimalMessageCode	The Message Code corresponding to the GBCS use case, e.g. 0021 (ECS10 Send Message to ESME) or 0071 (GCS07 Send Message to GSME). For Device Alerts the Message Code will be populated with generic codes used for Device Alerts unless there is a specific GBCS Use Case for it; see annex 15 section 15.3.1 for more details. Message Codes will be represented in XML by 4 hexadecimal characters e.g. “0021”, without the leading “0x” convention used in GBCS documentation.	xs:hexBinary	Yes	Values in 16 bit hexadecimal from 0001, as defined in GBCS Table 15.
ServiceReference	Identifier that signals the particular Service Request for which the response has been generated.	sr:ServiceReference (see Annex section 17)	No (required for a Service Response, not for a Device Alert)	See Main Document Table 33 Service Reference column
ServiceReferenceVariant	Identifier that signals the particular Service Request for which the response has been generated.	sr:ServiceReferenceVariant (see Annex section 17)	No (required for a Service Response, not for a Device Alert)	See Main Document Table 33 Service Reference Variant column
Timestamp	The time as sent by the Device, in UTC time. SMETS2: If the IsFromGSME attribute of the Timestamp in the Response is set to true, then this indicates that the value of Timestamp is set by the GSME, not the GPF. Additionally the ClockStatus attribute provides information about the timestamp. Valid set: • reliable • unreliable • invalid.	Extension of xs:dateTime (contains the optional attributes a. ‘IsFromGSME’ [1] of type xs:boolean b. ‘ClockStatus’ [2], which is a restriction of type xs:string)	No	UTC Date-Time

Table 1 Service Request Response Header Data Items

[1] This can only be present where indicated in Annex 4 that the Timestamp parameter can include it. See Table 1.1.

[2] This can only be present where indicated in Annex 4 that the Timestamp parameter can include it. See Table 1.2.

The following table shows how the IsFromGSME attribute of Timestamp will be populated by Parse for responses returned using MMC v5.1 or later. The attributes will not be present where responses are returned using an MMC version prior to MMC v5.1.

	GSME's GBCS version is v4.2 or later	GSME's GBCS version is prior to v4.2
GPF's GBCS version is v4.2 or later	True: The date-time source is GSME's clock; bit 2 is set in attribute 4 of the Clock object False: The date-time source is GPF's clock; bit 2 is unset in attribute 4 of the Clock object	False
GPF's GBCS version is prior to v4.2	False	False

Table 1.1 Derivation of the IsFromGSME attribute of Timestamp

The following table shows how the ClockStatus attribute of Timestamp will be populated by Parse for responses returned using MMC v5.1 or later. The attributes will not be present where responses are returned using an MMC version prior to MMC v5.1.

	GSME's GBCS version is v4.2 or later	GSME's GBCS version is prior to v4.2
GPF's GBCS version is v4.2 or later	• Where IsFromGSME is True: ClockStatus is the clock status of the GSME • Where IsFromGSME is False: ClockStatus is the clock status of the GPF	ClockStatus will always be the clock status of the GPF
GPF's GBCS version is prior to v4.2	ClockStatus will always be the clock status of the GPF	ClockStatus will always be the clock status of the GPF

Table 1.2 Derivation of the ClockStatus attribute of Timestamp

18.4.2 Body for a Service Response

Within the XML message containing data extracted from a GBCS payload XML element sent from the Device to DCC Service User, as well as a header XML structure there will be a body XML structure which contains data corresponding to either a Service Response or a Device Alert.

The response body features, at its top level, a sequence of three optional XML structures;

- A SMETSData group, which will hold the translated data from the message. In most cases this will be the only XML structure in the response body. It contains the overall status of the message, and successful data from GBCS responses if the GBCSPayload contains meaningful successful data. This structure will always be present except for cases where the message contains sensitive data which must be decrypted first, in which case SMETSData won't be included until the decrypted data has been returned by the DCC Service User calling a separate Parse function as defined in the Parse and Correlate Software.

- A GBCSData group, which can contain plain or encrypted raw GBCSData for an intermediate stage in Parse processing, where there is encrypted data in the GBCS payload included as part of the response.

- A DebugInfo group featuring any status information returned as part of an unsuccessful DLMS/COSEM or ZigBee message (this applies to most of them, the exception being the relatively small number coded in ASN.1 directly, which are mainly security-related). ASN.1 unsuccessful responses do not have an equivalent DebugInfo structure, instead status messages are embedded in the response in the SMETSData structure. The GBCS Use Cases which use ASN.1 data structures are listed and defined in GBCS section 13.

This is illustrated in the following diagram.

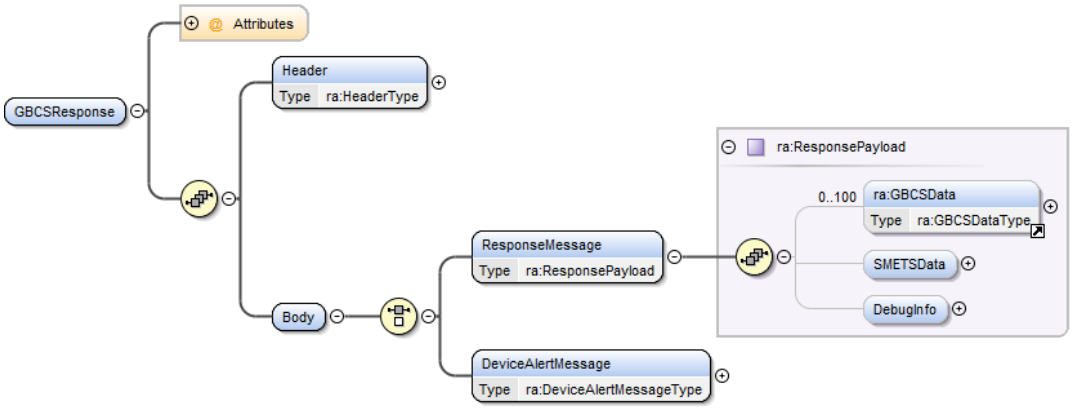

Figure 3 Response Body ResponseMessage XML type

The SMETSData group contains a choice of structures dependent on which service request it is responding to. Each one will include:

- A Boolean attribute called MessageSuccess, which is true for a message which was returned with no errors from the Device, and false if any error responses were returned;

- a set of elements corresponding to individual data items relevant to the Service Response.

The following diagram shows an illustration of some of the response types available in the SMETSData group. In the full list there is an XML type for the Service Response corresponding to each Service Request.

The full list is not shown because it would be too much detail for a diagram like this. The full list is available in the MMC XML Schema.

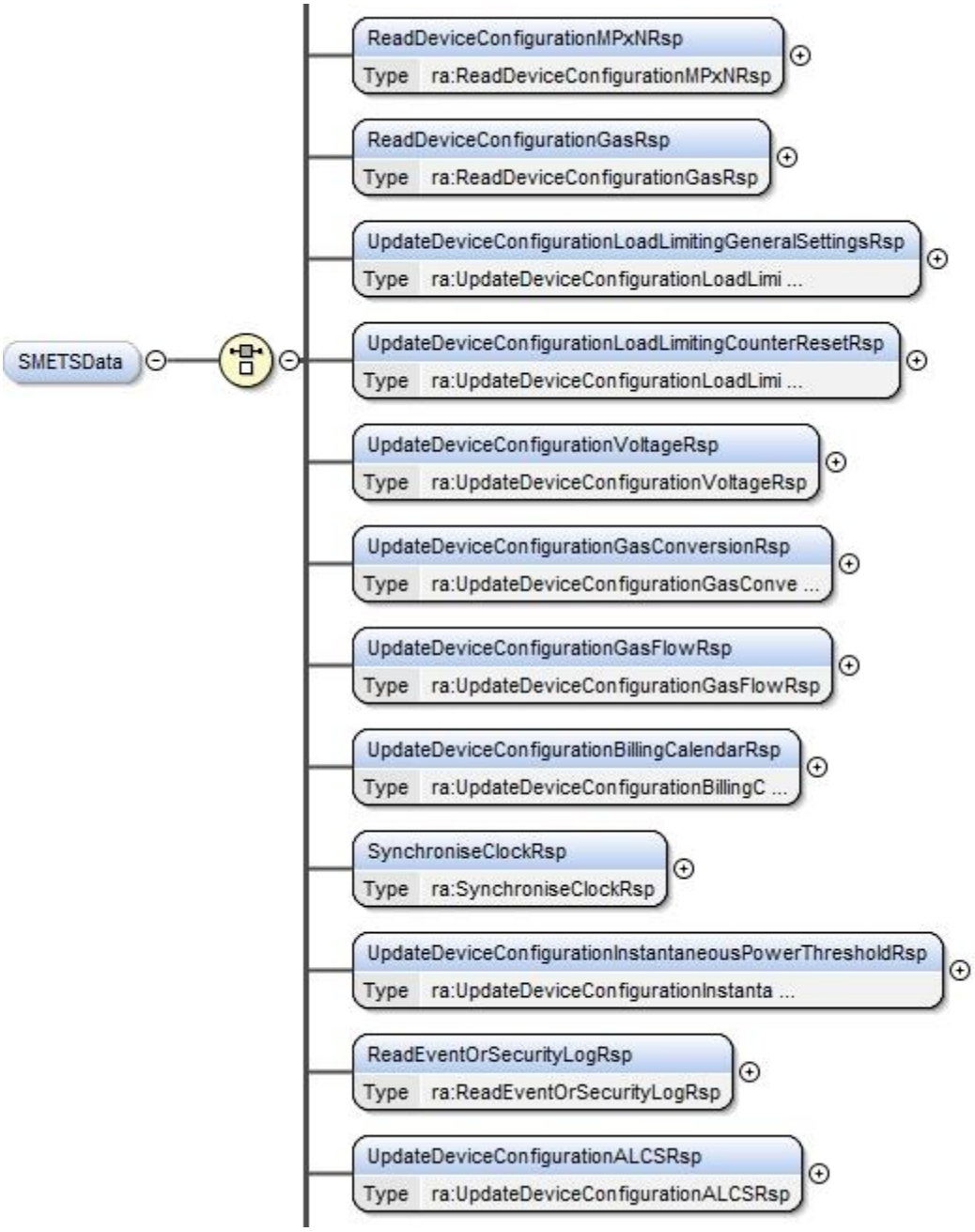

Figure 4 Response Body SMETSData with subset of response XML types (truncated for readability)

The structures corresponding to individual Service Response types are shown in the annexes corresponding to groups of Service Requests, e.g. Annex section 4 contains the Service Responses to read Service Requests such as 4.1.1.

Where applicable, common data returned by Electricity Smart Meters and Gas Smart Meters or Gas Proxy Functions are shared data items in the Service Responses to individual Service Requests. In cases where a response data item is applicable only to one of gas or electricity, this is found in a fuel-specific XML choice structure within the response message, and identified in data description tables for the response in the appropriate annex.

An example of Service Response with data differences between electricity and gas is shown below.

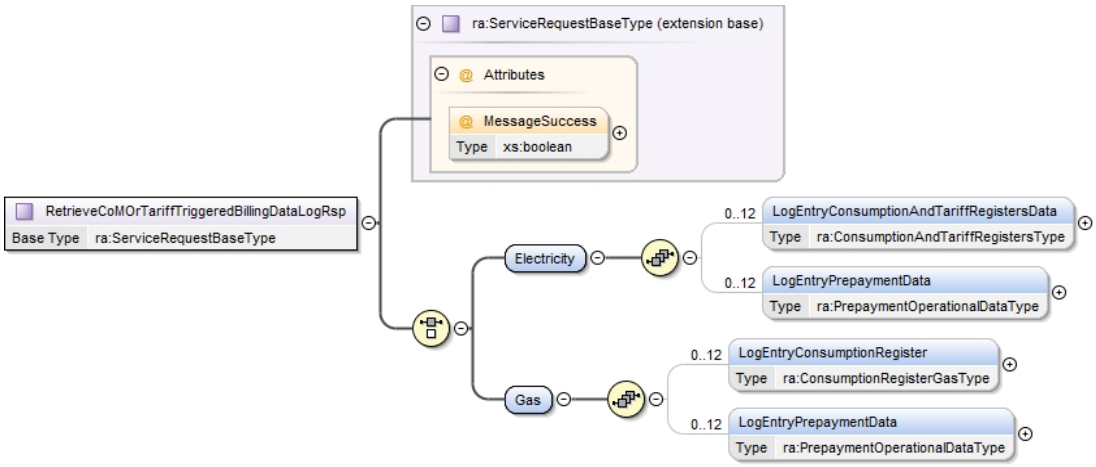

Figure 5 Response Body Example

18.4.3 Body for a Device Alert

Within the XML message containing data extracted from a Device Alert's GBCS payload, as well as a header there will be a body which contains data from the Device Alert. The majority of Device Alerts, as defined in GBCS, do not return any Device Alert specific additional information other than the identifier of the Device Alert and the time it was generated. Data common to all Device Alerts is shown in Table 2 Alert Data Items below. Each Device Alert will also have a timestamp, which is in the header of the XML message, as shown in Figure 2 in section 18.4.1.

Where Device Alerts have additional data sent by the Device there are specific XML types in the schema to represent the additional data.

In some cases, as defined by GBCS, Device Alerts can also contain encrypted data, which are handled in the same way as encrypted data in responses; see section 18.7.

See Annex section 15 for details of the DUIS XML Schema response wrapper for Device Alerts.

The structure is shown in the following diagram.

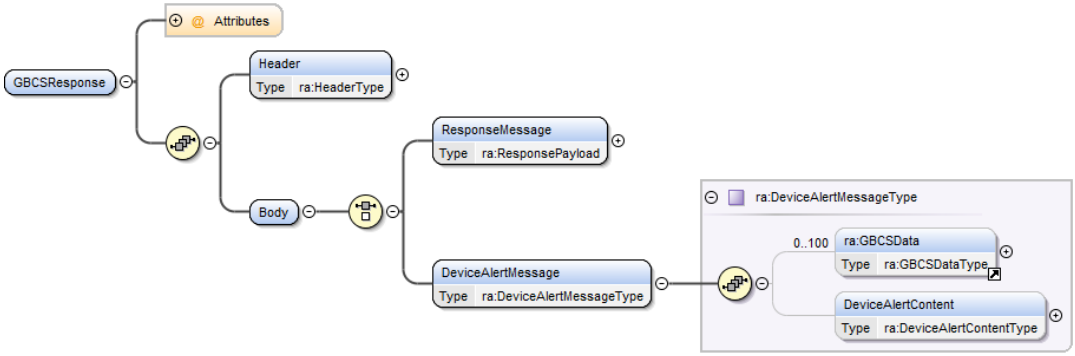

Figure 6 Alert Body DeviceAlertMessage XML type

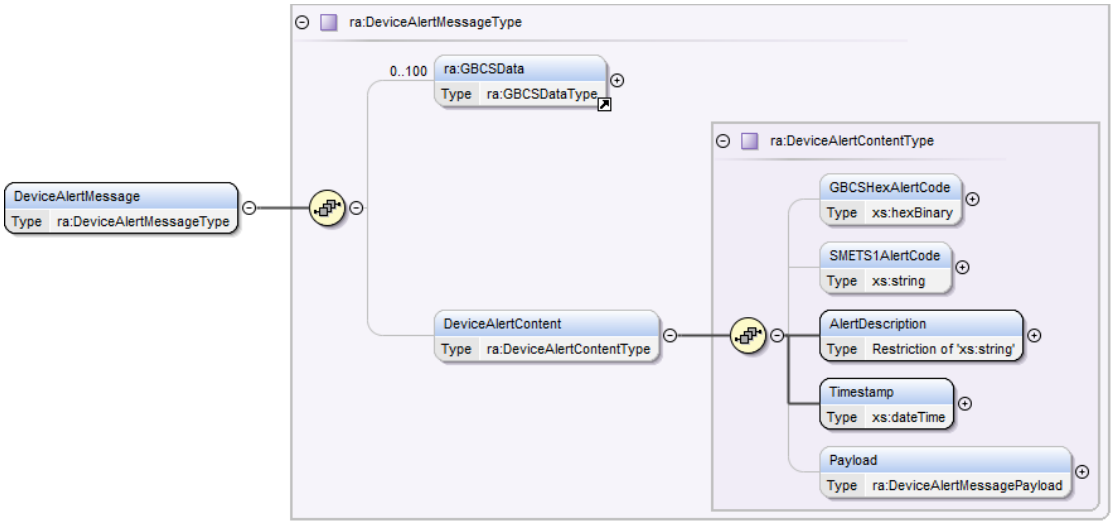

Figure 7 Alert Body DeviceAlertMessage with Alert content XML type

The Device Alert body XML structure includes the following fields:

Data Item	Description	Type	Mandatory	Valid Values
GBCSHexAlertCode	The Alert Code corresponding to the Device Alert defined in GBCS, e.g. 0x000C, or additional non-mandated codes as permitted by the GBCS definition. This is displayed in XML in format of 4 characters e.g. 000C. SMETS1 Alerts may contain a mandated GBCS code from the subset applicable to SMETS1 Devices, or a non-mandated code.	xs:hexBinary	Yes	Values in 16 bit hexadecimal from 0001, as defined in GBCS section 16.
AlertDescription	Description of the Device Alert as defined in GBCS e.g. “Clock adjustment greater than 10 seconds”, or additional information relevant to SMETS1.	xs:string (maxLength = 250)	Yes	See GBCS section 16 or SMETS1 Supporting Requirements Document
Timestamp	The Device Alert timestamp as sent by the Device, in UTC time.	xs:dateTime	Yes	UTC Date-Time
Payload	This is additional data specific to the GBCS Use Case, where there is data additional to the Alert Code. Most Device Alerts will not have additional data.	Ra:DeviceAlertMessagePayload	GBCS: No SMETS1: N/A	See Annex section 15.

Table 2 Alert Data Items

18.5 Sample Successful Responses

Three sample XML documents conforming to the MMC XML Schema are shown below, one for Electricity and two for Gas Smart Meters. These are shown as full XML documents conforming to the MMC XML Schema. The second gas sample shows the use of optional attributes of the Timestamp element.

In other annexes in this document set, header sections and the wrapping ra:Body, ra:ResponseMessage and ra:SMETSdata types are omitted from XML samples for specific Service Requests and corresponding responses.

```
<?xml version="1.0" encoding="UTF-8"?>
<ra:GBCSResponse xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
 xmlns:sr="http://www.dccinterface.co.uk/ServiceUserGateway"
 xmlns:ra="http://www.dccinterface.co.uk/ResponseAndAlert"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.dccinterface.co.uk/ResponseAndAlert" schemaVersion="1.0">
   <ra:Header>
      <ra:BusinessOriginatorID>00-00-00-00-00-00-00-00</ra:BusinessOriginatorID>
      <ra:BusinessTargetID>00-00-00-00-00-00-00-00</ra:BusinessTargetID>
      <ra:OriginatorCounter>50</ra:OriginatorCounter>
      <ra:GBCSHexadecimalMessageCode>002D</ra:GBCSHexadecimalMessageCode>
      <ra:ServiceReference>4.3</ra:ServiceReference>
      <ra:ServiceReferenceVariant>4.3</ra:ServiceReferenceVariant>
   </ra:Header>
   <ra:Body>
      <ra:ResponseMessage>
         <ra:SMETSData>
            <ra:ReadInstantaneousPrepayValuesRsp MessageSuccess="true">
               <ra:EmergencyCreditBalance>50</ra:EmergencyCreditBalance>
               <ra:AccumulatedDebtRegister>20</ra:AccumulatedDebtRegister>
               <ra:PaymentDebtRegister>10</ra:PaymentDebtRegister>
               <ra:TimeDebtRegister1>10</ra:TimeDebtRegister1>
               <ra:TimeDebtRegister2>10</ra:TimeDebtRegister2>
               <ra:MeterBalance>100</ra:MeterBalance>
            </ra:ReadInstantaneousPrepayValuesRsp>
         </ra:SMETSData>
      </ra:ResponseMessage>
   </ra:Body>
</ra:GBCSResponse>
```

Figure 8 Sample ReadInstantaneousPrepayRegistersRsp Parse Response Document for Electricity

```
<?xml version="1.0" encoding="UTF-8"?>
<ra:GBCSResponse xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
 xmlns:sr="http://www.dccinterface.co.uk/ServiceUserGateway"
 xmlns:ra="http://www.dccinterface.co.uk/ResponseAndAlert"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.dccinterface.co.uk/ResponseAndAlert" schemaVersion="1.0">
   <ra:Header>
      <ra:BusinessOriginatorID>00-00-00-00-00-00-00-00</ra:BusinessOriginatorID>
      <ra:BusinessTargetID>00-00-00-00-00-00-00-00</ra:BusinessTargetID>
      <ra:OriginatorCounter>50</ra:OriginatorCounter>
      <ra:GBCSHexadecimalMessageCode>0075</ra:GBCSHexadecimalMessageCode>
      <ra:ServiceReference>4.3</ra:ServiceReference>
      <ra:ServiceReferenceVariant>4.3</ra:ServiceReferenceVariant>
   </ra:Header>
   <ra:Body>
      <ra:ResponseMessage>
         <ra:SMETSData>
            <ra:ReadInstantaneousPrepayValuesRsp MessageSuccess="true">
               <ra:EmergencyCreditBalance>50</ra:EmergencyCreditBalance>
               <ra:AccumulatedDebtRegister>20</ra:AccumulatedDebtRegister>
               <ra:PaymentDebtRegister>10</ra:PaymentDebtRegister>
               <ra:TimeDebtRegister1>10</ra:TimeDebtRegister1>
               <ra:TimeDebtRegister2>10</ra:TimeDebtRegister2>
               <ra:MeterBalance>100</ra:MeterBalance>
            </ra:ReadInstantaneousPrepayValuesRsp>
         </ra:SMETSData>
      </ra:ResponseMessage>
   </ra:Body>
</ra:GBCSResponse>
```

Figure 9 Sample ReadInstantaneousPrepayRegisters Parse Response Document for Gas

```
<?xml version="1.0" encoding="UTF-8"?>
<ra:GBCSResponse xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:sr="http://www.dccinterface.co.uk/ServiceUserGateway"
 xmlns:ra="http://www.dccinterface.co.uk/ResponseAndAlert"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.dccinterface.co.uk/ResponseAndAlert" schemaVersion="1.0">
   <ra:Header>
      <ra:BusinessOriginatorID>00-00-00-00-00-00-00-00</ra:BusinessOriginatorID>
      <ra:BusinessTargetID>00-00-00-00-00-00-00-00</ra:BusinessTargetID>
      <ra:OriginatorCounter>50</ra:OriginatorCounter>
      <ra:GBCSHexadecimalMessageCode>0075</ra:GBCSHexadecimalMessageCode>
      <ra:ServiceReference>4.3</ra:ServiceReference>
      <ra:ServiceReferenceVariant>4.3</ra:ServiceReferenceVariant>
      <ra:Timestamp isFromGSME="true" ClockStatus="reliable">2017-08-25T03:04:05.00</ra:Timestamp>
   </ra:Header>
   <ra:Body>
      <ra:ResponseMessage>
         <ra:SMETSData>
            <ra:ReadInstantaneousPrepayValuesRsp MessageSuccess="true">
               <ra:EmergencyCreditBalance>50</ra:EmergencyCreditBalance>
               <ra:AccumulatedDebtRegister>20</ra:AccumulatedDebtRegister>
               <ra:PaymentDebtRegister>10</ra:PaymentDebtRegister>
               <ra:TimeDebtRegister1>10</ra:TimeDebtRegister1>
               <ra:TimeDebtRegister2>10</ra:TimeDebtRegister2>
               <ra:MeterBalance>100</ra:MeterBalance>
            </ra:ReadInstantaneousPrepayValuesRsp>
         </ra:SMETSData>
      </ra:ResponseMessage>
   </ra:Body>
</ra:GBCSResponse>
```

Figure 9.1 Sample Read Meter Balance Parse Response Document for Gas showing use of Timestamp attributes

18.6 Error Status in MMC XML Schema for Service Responses

18.6.1 Overview

This section applies to Service Responses. Device Alerts do not have a status of this sort since if a Device Alert has been received it must have been produced successfully.

For each Service Response, the SMETSData XML structure in the response Body has an overall Boolean status, called MessageSuccess, indicating the success (true) or failure (false) of the command requested by the Service User, as contained in the GBCS payload returned by the Device. If the Parse software has been unable to process the GBCS payload, this will be handled in a different way as indicated in the interface specification for Parse software.

Where there has been a failure in the execution of the command to a Device, the error statuses are handled in a different way according to the underlying specific protocol as defined by GBCS for the GBCS Use Case. In order to be able to return GBCS protocol-specific status responses there are different status types according to the underlying protocol used by the command. The underlying protocol could consist of DLMS/COSEM (e.g. for Electricity Smart Meters), ZigBee Smart Energy Protocol (e.g. for Gas Smart Meters) or ASN.1 (which could be for Gas or Electricity Smart Meters or other Devices).

- For GBCS commands based upon the DLMS/COSEM protocol, which means most commands to Electricity Smart Meters or Communications Hubs, the Device response status codes will be in a COSEMDebug structure within the DebugInfo group;
- for GBCS commands based upon the ZigBee protocol, which includes most commands to gas Devices, the Device response status codes will be in a ZIGBEEDebug structure within the DebugInfo group;
- a few security-related Device commands are implemented in a binary protocol represented in ASN.1. In these cases status codes are embedded within the response structure for the command and DebugInfo is not used.

The MessageSuccess Boolean in the SMETSData XML structure indicates that the GBCS commands to the Device which were initiated for the Service Request were successful overall if no errors were returned by the Device, so in successful cases there will be no need to read individual protocol-specific statuses. Protocol-specific statuses are provided for fault investigation in cases of unsuccessful responses from the Device.

Where errors have been returned by the Device, the MMC XML Schema provides additional status information in an XML structure called DebugInfo, as described in section18.4.2, except for cases where the underlying protocol is defined by GBCS as using ASN.1. Where a message has succeeded in its entirety only the overall MessageStatus and data are returned from Parse, in a SMETSData XML structure, with no DebugInfo structure. Where a message has been partially successful, Parse will return as much data as it was able to decode in a SMETSData structure, if any, along with the debug information in DebugInfo structure.

Responses in the MMC XML Schema may contain complex data structures and multiple data items, and status for these may be different from simply one result per data item, so in some cases a single status may be returned for a set of data. The correspondence of error status codes to individual or groups of attributes may be different in the different underlying protocols as defined in GBCS, namely DLMS/COSEM, ZigBee Smart Energy Protocol or using ASN.1.

In DLMS/COSEM there are different sets of error responses for action (update) commands and data access (read) commands. A DLMS/COSEM exchange with a Device uses break-on-error processing, meaning that if an individual component DLMS/COSEM command as part of a GBCS use case fails, then subsequent commands will not be executed; in this case the “other-reason” failure status will be returned for commands after the first failed command.

Some commands return no substantial information apart from status, e.g. many commands which update configuration on meters, and in these cases there are no specific examples in

the annex documents describing individual Parse Output responses for specific Service Requests, and the sections refer to the examples in this section.

The list of valid values for each underlying protocol within GBCS (DLMS/COSEM, ZigBee Smart Energy Protocol and ASN.1) is listed in this section and the MMC XML Schema as well as GBCS and underlying protocol documentation.

In the next few sections there are some examples of a command which returns no substantial data apart from status. In other annexes there will not be examples of commands which only return status.

18.6.2 ZIGBEEDebug Status Structure

This is the structure used to return debug information for a ZigBee message (also in this context the related terms ZSE or GBZ are sometimes used). It contains one or more ZIGBEEClusterResponse XML structures, the number depending on the GBCS Use Case and corresponding to the ZigBee clusters in the GBCS command.

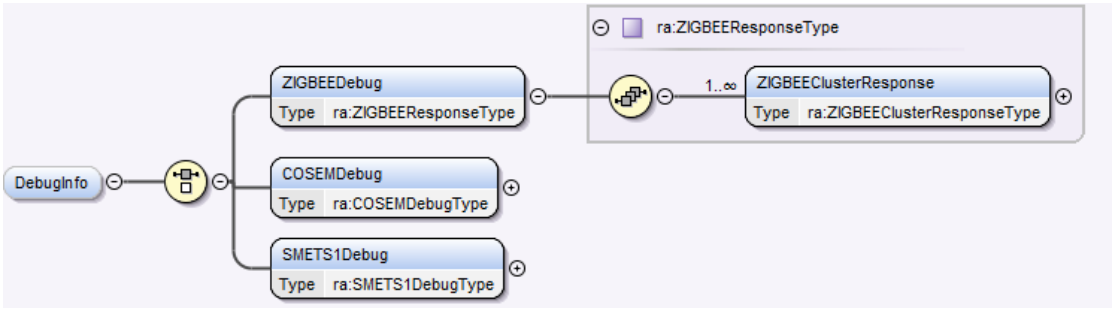

Figure 10 - ZIGBEEDebug Structure

The following diagram expands ZIGBEEClusterResponse.

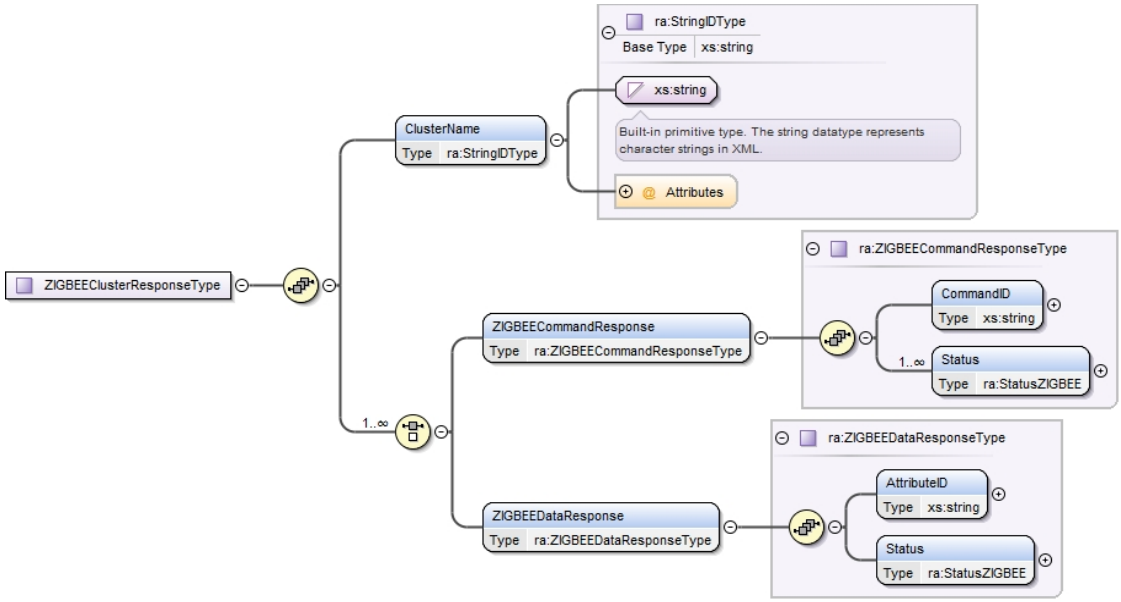

Figure 11 - ZIGBEEClusterResponse Structure

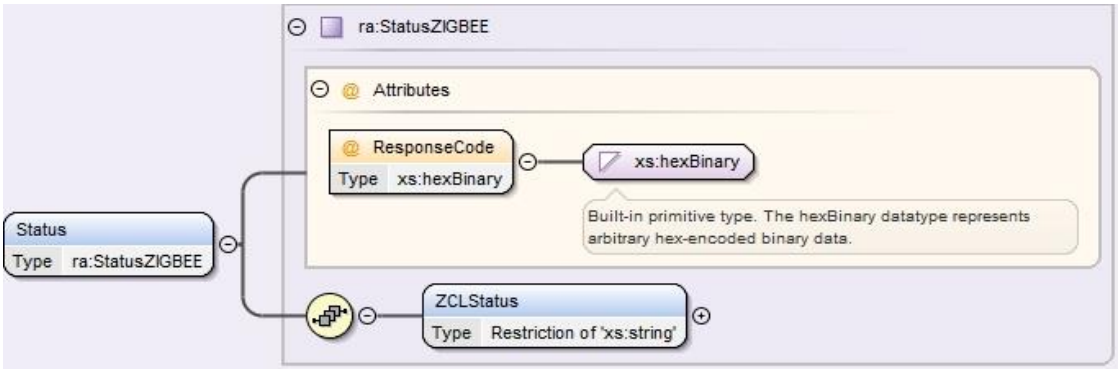

Figure 12 - StatusZIGBEE Structure

18.6.2.1 Data Items Definition

A command based on ZigBee, which means most GBCS commands to gas Devices (the remainder use ASN.1), will return information about one or more ZigBee Clusters, as defined in GBCS, each of which may be a ZIGBEEDataResponse structure (for status about attributes from the Device) or a ZIGBEECommandResponse structure (for status about commands). Note that in many cases data is read using a single command (rather than by attribute), and in these cases the status will be returned in a ZIGBEECommandResponse type.

A Zigbee failure response requires a level of expert interpretation. For some Zigbee failure responses the debug information will need to be augmented with the contextual detail in the Service Request in order to complete the analysis.

Data Item	Description / Valid Set	Type	Mandatory	Valid Values
ClusterName	The name of the ZSE cluster from which the response was received. The XML type also carries the ZigBee Cluster ID of the cluster, e.g. <ra:ClusterName id="0705">Select Available Emergency Credit</ra:ClusterName>	ra:StringIDType (maxLength=60) This type includes attribute numberID, which is type xs:hexBinary	Yes	Defined in GBCS Table 7.2.10b and "ZigBee Commands" tab of Table 20 Mapping Table in section 20
CommandID	ZigBee Smart Energy Protocol command identifier of an operation within the ZSE cluster that is used to update or read from a Device, e.g. 02. Part of a ZIGBEECommandResponse structure.	xs: hexBinary	No	Defined in GBCS Table 7.2.10b and "ZigBee Commands" tab of Table 20 Mapping Table in section 20
AttributeID	For ZSE read by attribute operations – the attribute ID or a value returned For ZSE update operations – the attribute or parameter updated. e.g.0100. Part of a ZIGBEEDataResponse structure.	xs: hexBinary	No	Defined in GBCS Table 7.2.10b and "ZigBee Commands" tab of Table 20 Mapping Table in section 20

Data Item	Description / Valid Set	Type	Mandatory	Valid Values
ZCLStatus	ZIGBEE status value, one of those defined in section 18.6.2.2 corresponding to the status result of using a CommandID (if part of a ZIGBEECommandResponse) or AttributeID (if part of a ZIGBEEDataResponse structure).	ra:StringIDType (maxLength=60) This type includes attribute numberID, which is type xs:hexBinary	Yes	Defined in GBCS. See section 18.6.2.2 for a summary.

18.6.2.2 ZigBee Smart Energy Response Codes (ZCLStatus Values)

The master reference for these codes is GBCS. A list is included here for convenience.

Response Code	Response Code Name
0x00	SUCCESS
0x01	FAILURE
0x7e	NOT_AUTHORIZED
0x7f	RESERVED_FIELD_NOT_ZERO
0x80	MALFORMED_COMMAND
0x81	UNSUP_CLUSTER_COMMAND
0x82	UNSUP_GENERAL_COMMAND
0x83	UNSUP_MANUF_CLUSTER_COMMAND
0x84	UNSUP_MANUF_GENERAL_COMMAND
0x85	INVALID_FIELD
0x86	UNSUPPORTED_ATTRIBUTE
0x87	INVALID_VALUE
0x88	READ_ONLY
0x89	INSUFFICIENT_SPACE
0x8a	DUPLICATE_EXISTS
0x8b	NOT_FOUND
0x8c	UNREPORTABLE_ATTRIBUTE
0x8d	INVALID_DATA_TYPE
0x8e	INVALID_SELECTOR
0x8f	WRITE_ONLY
0x90	INCONSISTENT_STARTUP_STATE
0x91	DEFINED_OUT_OF_BAND
0x92	INCONSISTENT
0x93	ACTION_DENIED

Response Code	Response Code Name
0x94	TIMEOUT
0x95	ABORT
0x96	INVALID_IMAGE
0x97	WAIT_FOR_DATA
0x98	NO_IMAGE_AVAILABLE
0x99	REQUIRE_MORE_IMAGE
0xc0	HARDWARE_FAILURE
0xc1	SOFTWARE_FAILURE
0xc2	CALIBRATION_ERROR
[any other]	RESPONSE_CODE_NOT_KNOWN

Table 3 ZigBee Smart Energy Response Codes

If a response code is issued by a device which is not part of this list then the number will be included in the response and the text will be “RESPONSE_CODE_NOT_KNOWN”.

18.6.2.3 Sample ZigBee Error Response

Below there is sample Service Response document Body showing ZigBee error status for both a Command response and a data response. Note that in some cases there may be partial data in SMETSData format, if some was sent by the Device which could be interpreted by Parse software, but in this example there is no data in SMETSData apart from the overall message status.

```
<ra:ResponseMessage>
    <ra:SMETSData>
        <ra:ReadDeviceConfigurationDataBillingCalendarRsp MessageSuccess="false"/>
    </ra:SMETSData>
  <ra:DebugInfo>
    <ra:ZIGBEEDebug>
      <ra:ZIGBEEClusterResponse>
        <ra:ClusterName id="0700">Price</ra:ClusterName>
        <ra:ZIGBEECommandResponse>
          <ra:CommandID>09</ra:CommandID>
          <ra:Status ResponseCode="7e">
            <ra:ZCLStatus>NOT_AUTHORIZED</ra:ZCLStatus>
          </ra:Status>
        </ra:ZIGBEECommandResponse>
      </ra:ZIGBEEClusterResponse>
    </ra:ZIGBEEDebug>
  </ra:DebugInfo>
</ra:ResponseMessage>
```

Figure 13 Sample ZigBee Error Response – Command Response

```
<ra:ResponseMessage>
    <ra:SMETSData>
        <ra:ReadDeviceConfigurationDataBillingCalendarRsp MessageSuccess="false"/>
    </ra:SMETSData>
  <ra:DebugInfo>
    <ra:ZIGBEEDebug>
      <ra:ZIGBEEClusterResponse>
        <ra:ClusterName id="0702">Metering</ra:ClusterName>
        <ra:ZIGBEEDataResponse>
          <ra:AttributeID>0205</ra:AttributeID>
          <ra:Status ResponseCode="00">
            <ra:ZCLStatus>SUCCESS</ra:ZCLStatus>
          </ra:Status>
        </ra:ZIGBEEDataResponse>
        <ra:ZIGBEEDataResponse>
          <ra:AttributeID>0014</ra:AttributeID>
          <ra:Status ResponseCode="c1">
            <ra:ZCLStatus>SOFTWARE_FAILURE</ra:ZCLStatus>
          </ra:Status>
        </ra:ZIGBEEDataResponse>
      </ra:ZIGBEEClusterResponse>
    </ra:ZIGBEEDebug>
  </ra:DebugInfo>
</ra:ResponseMessage>
```

Figure 14 Sample ZigBee Error Response – Data Response

18.6.3 COSEMDebug Status Structure

Debug information for DLMS/COSEM messages is returned using the structure below. The number of COSEMResponse structures depends on the GBCS Use Case and corresponds to the number of DLMS/COSEM instructions in the GBCS Command, in the order returned by the Device.

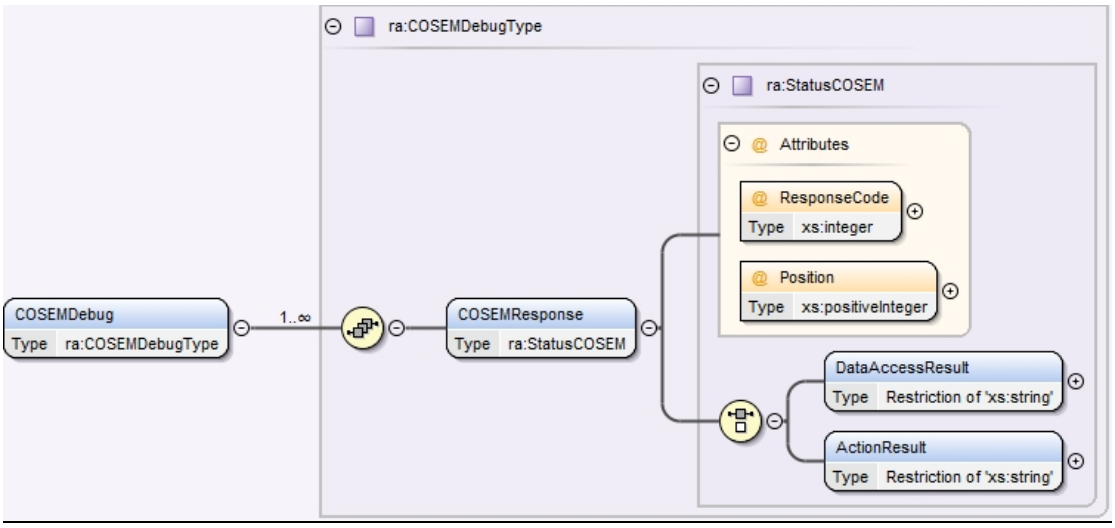

Figure 15 - COSEMDebug Structure

18.6.3.1 Data Items Definition

A command based on DLMS/COSEM, which is used for most of the commands to ESME and Communications Hub Devices, will return status using data access results and action results, the type and number depending on the type of operation. There may be just one data access result, one or more action results, or one of each.

Data Item	Description / Valid Set	Type	Mandatory	Valid Values
ResponseCode (attribute of COSEMResponse)	This contains the numerical code returned by the Device, which corresponds to the text string.	xs:integer	Yes	See sections 18.6.3.2 & 18.6.3.3
Position	This is an incrementing value showing the position of the response code in the order in which it was executed.	xs:positiveInteger	Yes	Positive integer starting from 1
DataAccessResult	Status string denoting the result of a get or set operation.	ra:StringIDType (maxLength=60) This type includes attribute numberID, which is type xs:hexBinary	Choice of this or DataActionResult	Defined in GBCS. See section 18.6.3.3 for a summary.
DataActionResult	Status string denoting the result of an action operation.	ra:StringIDType (maxLength=60) This type includes attribute numberID, which is type xs:hexBinary	Choice of this or DataAccessResult	Defined in GBCS. See section 18.6.3.2for a summary.

18.6.3.2 DLMS/COSEM Response Codes - Action

The master reference for these codes is GBCS. A list is included here for convenience.

These codes are a result of Action commands in DLMS/COSEM e.g. updating a configuration setting.

Response Code	Response Code Name	Response Code Type
0	success	Action
1	hardware-fault	Action
2	temporary-failure	Action
3	read-write-denied	Action
4	object-undefined	Action
9	object-class-inconsistent	Action
11	object-unavailable	Action
12	type-unmatched	Action
13	scope-of-access-violated	Action
14	data-block-unavailable	Action
15	long-action-aborted	Action
16	no-long-action-in-progress	Action

Response Code	Response Code Name	Response Code Type
250	other-reason	Action
[any other]	response-code-not-known	In case response code not known

Table 4 DLMS/COSEM Action Response Codes

18.6.3.3 DLMS/COSEM Response Codes – Data Access

The master reference for these codes is GBCS. A list is included here for convenience.

These codes are a result of data access commands in DLMS/COSEM e.g. reading a register.

Response Code	Response Code Name	Response Code Type
0	success	Data access
1	hardware-fault	Data access
2	temporary-failure	Data access
3	read-write-denied	Data access
4	object-undefined	Data access
9	object-class-inconsistent	Data access
11	object-unavailable	Data access
12	type-unmatched	Data access
13	scope-of-access-violated	Data access
14	data-block-unavailable	Data access
15	long-get-aborted	Data access
16	no-long-get-in-progress	Data access
17	long-set-aborted	Data access
18	no-long-set-in-progress	Data access
19	data-block-number-invalid	Data access
250	other-reason	Data access
[any other]	response-code-not-known	In case response code not known

Table 5 DLMS/COSEM Data Access Response Codes

18.6.3.4 Sample DLMS/COSEM Error Response

A sample Service Response document Body containing a DLMS/COSEM error status is as follows. Note that in some cases there may be partial data in SMETSData format, if some was sent by the Device which could be interpreted by Parse software, but in this example there is no data in SMETSData apart from the overall message status.

```
<ra:ResponseMessage>
    <ra:SMETSData>
        <ra:UpdatePricePrimaryElementRsp MessageSuccess="false"/>
    </ra:SMETSData>
  <ra:DebugInfo>
      <ra:COSEMDebug>
        <ra:COSEMResponse ResponseCode="0" Position="1">
            <ra:ActionResult>success</ra:ActionResult>
        </ra:COSEMResponse>
        <ra:COSEMResponse ResponseCode="0" Position="2">
            <ra:ActionResult>success</ra:ActionResult>
        </ra:COSEMResponse>
        <ra:COSEMResponse ResponseCode="3" Position="3">
            <ra:ActionResult>hardware-fault</ra:ActionResult>
        </ra:COSEMResponse>
        <ra:COSEMResponse ResponseCode="250" Position="4">
            <ra:ActionResult>other-reason</ra:ActionResult>
        </ra:COSEMResponse>
        <ra:COSEMResponse ResponseCode="250" Position="5">
            <ra:ActionResult>other-reason</ra:ActionResult>
        </ra:COSEMResponse>
        <ra:COSEMResponse ResponseCode="250" Position="6">
            <ra:ActionResult>other-reason</ra:ActionResult>
        </ra:COSEMResponse>
      </ra:COSEMDebug>
  </ra:DebugInfo>
</ra:ResponseMessage>
```

Figure 16 Sample DLMS/COSEM Error Response

18.6.4 ASN.1 Errors

A small number of GBCS Use Cases are encoded in ASN.1, as defined in GBCS section 13, and apply to both Electricity and Gas Smart Meter Devices as well as other Devices. Unlike GBCS Use Cases where the underlying specific protocol is DLMS/COSEM or ZigBee Smart Energy Protocol, MMC XML Schema responses based on GBCS payload responses which are encoded in ASN.1 have error statuses embedded in the structures, rather than using a separate DebugInfo structure as described in section 18.6.

18.6.4.1 ASN.1 Error Response Codes

The master reference for these codes is GBCS. A list is included here for convenience.

Note that in some cases the same response code numbers are used with different meanings in different ASN.1 Service Responses, as can be seen in the table below.

Service Request	Response Code Name	Response Code
All ASN.1 SRs	success	0
Potentially any ASN.1 SR	notKnown (for cases where the ASN.1 response code is not recognised by the Parse software)	As returned by device
6.11 (gas only), 8.1.1 (gas only)	reliable	0
6.11 (gas only) , 8.1.1 (gas only)	invalid	1
6.11 (gas only) , 8.1.1 (gas only)	unreliable	2
6.15.1, 6.21, 6.23, 8.5	badCertificate	5

Service Request	Response Code Name	Response Code
6.15.1, 6.21, 6.23, 8.5	noTrustAnchor	10
6.15.1, 6.21, 6.23, 8.5	insufficientMemory	17
6.15.1, 6.21, 6.23, 8.5	resourcesBusy	30
6.15.1, 6.21, 6.23, 8.5	other	127
6.15.2	invalidCertificate	1
6.15.2	wrongDeviceIdentity	2
6.15.2	invalidKeyUsage	3
6.15.2	noCorrespondingKeyPair	4
6.15.2	wrongPublicKey	5
6.15.2	certificateStorageFailed	6
6.15.2	privateKeyChangeFailed	7
6.17	invalidKeyUsage	1
6.17	keyPairGenerationFailed	2
6.17	cRProductionFailed	3
6.24.1	trustAnchorNotFound	25
6.24.1	other	127
6.24.2	invalidKeyUsage	1
6.24.2	noCertificateHeld	2
6.24.2	certificateRetrievalFailure	3
8.7.1, 8.7.2	invalidMessageCodeForJoinMethodAndRole	1
8.7.1, 8.7.2	invalidJoinMethodAndRole	2
8.7.1, 8.7.2	incompatibleWithExistingEntry	3
8.7.1, 8.7.2	deviceLogFull	4
8.7.1, 8.7.2	writeFailure	5
8.7.1, 8.7.2	keyAgreementNoResources	6
8.7.1, 8.7.2	keyAgreementUnknownIssuer	7
8.7.1, 8.7.2	keyAgreementUnsupportedSuite	8
8.7.1, 8.7.2	keyAgreementBadMessage	9
8.7.1, 8.7.2	keyAgreementBadKeyConfirm	10
8.7.1, 8.7.2	invalidOrMissingCertificate	11
8.7.1, 8.7.2	noPartnerLinkKeyReceived	12
8.7.1, 8.7.2	noCBKEResponse	13
8.8.1, 8.8.2	otherDeviceNotInDeviceLog	1

Service Request	Response Code Name	Response Code
8.8.1, 8.8.2	otherFailure	2
8.9	readFailure	1
8.12.2	incompatibleWithExistingEntry	3
8.12.2	deviceLogFull	4
8.12.2	writeFailure	5
11.3	noImageHeld	1
11.3	hashMismatch	2
11.3	activationFailure	3

Table 6 ASN.1 Response Codes

18.6.4.2 Sample ASN.1 Error Response

Unlike errors in response messages which use DLMS/COSEM or ZigBee, ASN.1 messages have error codes embedded in the message, so they do not use a separate DebugInfo structure.

18.7 Encrypted fields

In some cases encrypted data is returned by Devices, as defined in GBCS. This can be individual data items such as a single meter register, or a set of data which would require a complex XML structure to represent it.

There will need to be 2 calls from the DCC Service User to Parse software (as defined further in the Parse software Interface Specification) where the GBCS payload includes encrypted data, and the Parse Output XML Schema contains features to support this. There are several interactions, including 3 stages of XML data (in stages 2, 3 and 4 below):

1. The DCC Service User calls the Parse software with GBCS payload in which there is embedded encrypted data. The DCC Service User cannot decrypt this yet, since it is in GBCS format, so the data is passed to the Parse software to decode it.

2. The Parse software cannot decode the encrypted data, so it breaks the GBCS payload into fragments of GBCS, some plain and some encrypted. The XML it returns to the DCC Service User will be passed in a set of XML elements called GBCSData, with the “format” attribute set to “plain” or “encrypted”.

3. The DCC Service User decrypts the encrypted data, so the data is now plain text GBCS format, and writes the decrypted data back into the XML GBCSData element, changing the “format” attribute to “plain”. This could happen at several points in the XML message, depending on the Service Request. It passes the amended XML to the Parse software.

4. The Parse software is now able to decode the GBCS data and populates the XML message. The Parse software is then able to return the fully decrypted and decoded XML message back to the DCC Service User.

Note that Parse software will not partially decode GBCS data, it will return either a mixture of plain and encrypted GBCSData groups or fully decoded XML.

See section 18.8 for illustrations of this behaviour.

18.8 Interaction Diagrams

The following diagram illustrates the decoding of GBCS payload into XML in the case where there is no encrypted data.

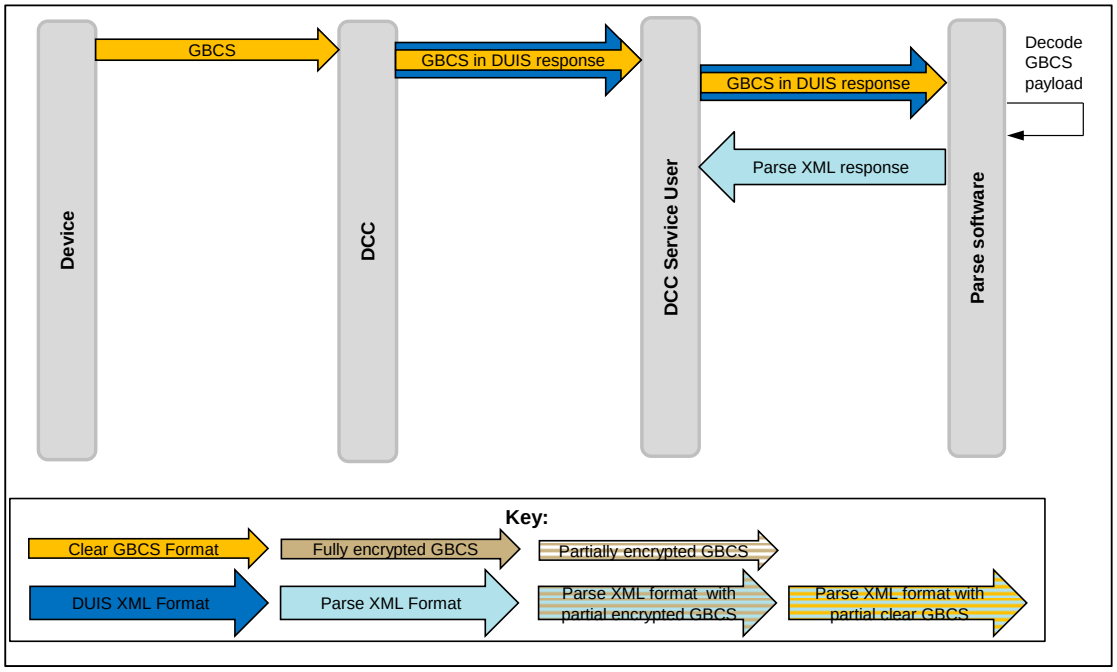

Figure 17 Parse of GBCS payload from Device with no encrypted data

The following diagram illustrates decryption in the interaction between DCC Service Users and the Parse software. For full details see the interface specification for the Parse software.

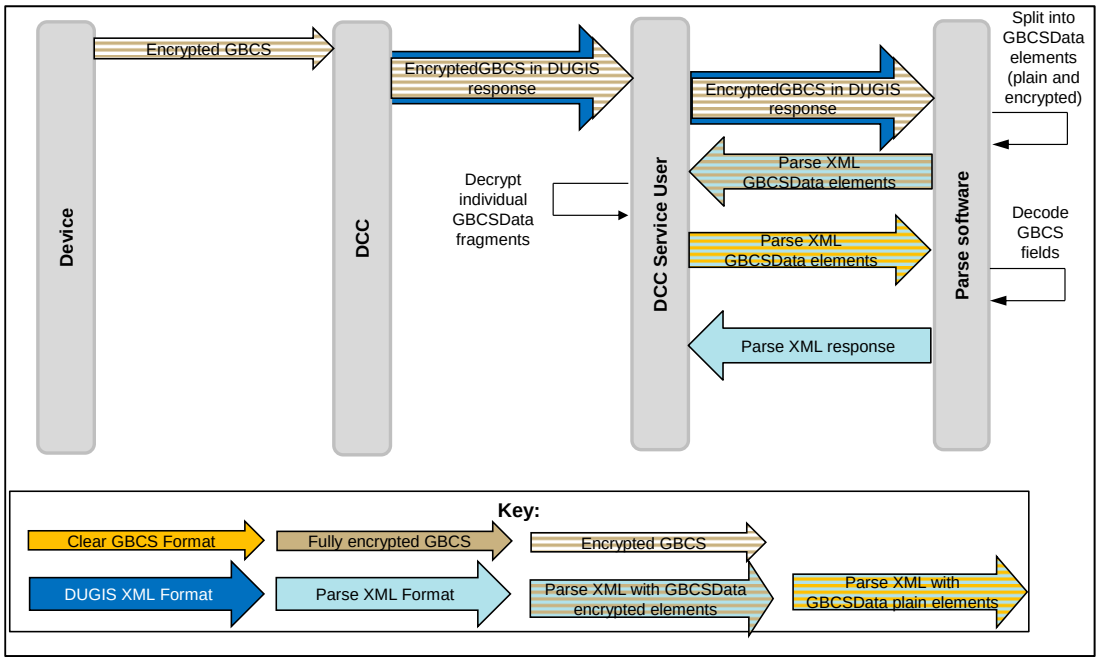

Figure 18 Parse of GBCS payload from Device with encrypted data

18.9 Status-Only Responses

Many responses from Devices in response to commands, e.g. commands which perform updates, contain no substantial payload, just status information. In successful cases these will

simply contain an overall success or failure, within an XML type. The name of the XML type which corresponds to the XML type of the Service Request in the DUIS XML Schema, with the suffix “Rsp”, e.g. ActivateEmergencyCreditRsp for Service Request to activate emergency credit on a meter.

Cases where an error message has been returned from the Device will follow the normal approach to unsuccessful responses, as described in section 18.5.

See the next section 18.9.1 in this document for examples of responses which contain no substantial data other than the status.

In general, annexes in this DUGIDS document set will not contain structure diagrams, data diagrams or XML samples for cases like this, as they all follow the pattern in the next section 18.9.1.

18.9.1 Sample Status-Only Responses

Sample status-only MMC XML Schema response documents are shown below, one successful and one unsuccessful for each of Electricity and Gas Smart Meter Devices.

```
<?xml version="1.0" encoding="UTF-8"?>
<ra:GBCSResponse xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
 xmlns:sr="http://www.dccinterface.co.uk/ServiceUserGateway"
 xmlns:ra="http://www.dccinterface.co.uk/ResponseAndAlert"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.dccinterface.co.uk/ResponseAndAlert" schemaVersion="1.0">
   <ra:Header>
      <ra:BusinessOriginatorID>10-00-00-00-00-00-00-00</ra:BusinessOriginatorID>
      <ra:BusinessTargetID>20-10-00-00-00-00-00-00</ra:BusinessTargetID>
      <ra:OriginatorCounter>50</ra:OriginatorCounter>
      <ra:GBCSHexadecimalMessageCode>0020</ra:GBCSHexadecimalMessageCode>
      <ra:ServiceReference>2.5</ra:ServiceReference>
      <ra:ServiceReferenceVariant>2.5</ra:ServiceReferenceVariant>
   </ra:Header>
   <ra:Body>
      <ra:ResponseMessage>
         <ra:SMETSData>
            <ra:ActivateEmergencyCreditRsp MessageSuccess="true"/>
         </ra:SMETSData>
      </ra:ResponseMessage>
   </ra:Body>
</ra:GBCSResponse>
```

Figure 19 Sample Activate Emergency Credit Parse Response Document for Electricity

```
<?xml version="1.0" encoding="UTF-8"?>
<ra:GBCSResponse xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
 xmlns:sr="http://www.dccinterface.co.uk/ServiceUserGateway"
 xmlns:ra="http://www.dccinterface.co.uk/ResponseAndAlert"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.dccinterface.co.uk/ResponseAndAlert" schemaVersion="1.0">
   <ra:Header>
      <ra:BusinessOriginatorID>10-00-00-00-00-00-00-00</ra:BusinessOriginatorID>
      <ra:BusinessTargetID>20-10-00-00-00-00-00-00</ra:BusinessTargetID>
      <ra:OriginatorCounter>50</ra:OriginatorCounter>
      <ra:GBCSHexadecimalMessageCode>0020</ra:GBCSHexadecimalMessageCode>
      <ra:ServiceReference>2.5</ra:ServiceReference>
      <ra:ServiceReferenceVariant>2.5</ra:ServiceReferenceVariant>
   </ra:Header>
   <ra:Body>
      <ra:ResponseMessage>
         <ra:SMETSData>
            <ra:ActivateEmergencyCreditRsp MessageSuccess="false"/>
         </ra:SMETSData>
         <ra:DebugInfo>
            <ra:COSEMDebug>
               <ra:COSEMResponse ResponseCode="1" Position="1">
                  <ra:ActionResult>hardware-fault</ra:ActionResult>
               </ra:COSEMResponse>
            </ra:COSEMDebug>
         </ra:DebugInfo>
      </ra:ResponseMessage>
   </ra:Body>
</ra:GBCSResponse>
```

Figure 20 Sample (Failed) Activate Emergency Credit Parse Response Document for Electricity

```
<?xml version="1.0" encoding="UTF-8"?>
<ra:GBCSResponse xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
 xmlns:sr="http://www.dccinterface.co.uk/ServiceUserGateway"
 xmlns:ra="http://www.dccinterface.co.uk/ResponseAndAlert"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.dccinterface.co.uk/ResponseAndAlert" schemaVersion="1.0">
   <ra:Header>
      <ra:BusinessOriginatorID>10-00-00-00-00-00-00-00</ra:BusinessOriginatorID>
      <ra:BusinessTargetID>20-10-00-00-00-00-00-00</ra:BusinessTargetID>
      <ra:OriginatorCounter>50</ra:OriginatorCounter>
      <ra:GBCSHexadecimalMessageCode>0070</ra:GBCSHexadecimalMessageCode>
      <ra:ServiceReference>2.5</ra:ServiceReference>
      <ra:ServiceReferenceVariant>2.5</ra:ServiceReferenceVariant>
   </ra:Header>
   <ra:Body>
      <ra:ResponseMessage>
         <ra:SMETSData>
            <ra:ActivateEmergencyCreditRsp MessageSuccess="true"/>
         </ra:SMETSData>
      </ra:ResponseMessage>
   </ra:Body>
</ra:GBCSResponse>
```

Figure 21 Sample Activate Emergency Credit Parse Response Document for Gas

```
<?xml version="1.0" encoding="UTF-8"?>
<ra:GBCSResponse xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
 xmlns:sr="http://www.dccinterface.co.uk/ServiceUserGateway"
 xmlns:ra="http://www.dccinterface.co.uk/ResponseAndAlert"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.dccinterface.co.uk/ResponseAndAlert" schemaVersion="1.0">
   <ra:Header>
      <ra:BusinessOriginatorID>10-00-00-00-00-00-00-00</ra:BusinessOriginatorID>
      <ra:BusinessTargetID>20-10-00-00-00-00-00-00</ra:BusinessTargetID>
      <ra:OriginatorCounter>50</ra:OriginatorCounter>
      <ra:GBCSHexadecimalMessageCode>0070</ra:GBCSHexadecimalMessageCode>
      <ra:ServiceReference>2.5</ra:ServiceReference>
      <ra:ServiceReferenceVariant>2.5</ra:ServiceReferenceVariant>
   </ra:Header>
   <ra:Body>
      <ra:ResponseMessage>
         <ra:SMETSData>
            <ra:ActivateEmergencyCreditRsp MessageSuccess="false"/>
         </ra:SMETSData>
         <ra:DebugInfo>
            <ra:ZIGBEEDebug>
               <ra:ZIGBEEClusterResponse>
                  <ra:ClusterName id="0705">Select Available Emergency Credit</ra:ClusterName>
                  <ra:ZIGBEECommandResponse>
                     <ra:CommandID>0705</ra:CommandID>
                     <ra:Status ResponseCode="c0">
                        <ra:ZCLStatus>HARDWARE_FAILURE</ra:ZCLStatus>
                     </ra:Status>
                  </ra:ZIGBEECommandResponse>
               </ra:ZIGBEEClusterResponse>
            </ra:ZIGBEEDebug>
         </ra:DebugInfo>
      </ra:ResponseMessage>
   </ra:Body>
</ra:GBCSResponse>
```

Figure 22 Sample (Failed) Activate Emergency Credit Parse Response Document for Gas

18.10 Mandatory Fields

The data which comes back from the Device in response to Service Requests and is represented in XML in the SMETSData group must all be regarded as non-mandatory, because in error cases there might not be any data to bring back, or partial data may be returned in an error case where the Device was able to return some of the data successfully. The only mandatory data item in SMETSData is the overall success status (true or false).

In the annex sections of this document set which describe responses conforming to the MMC XML Schema, specifically the “Specific Data Items” sub-sections within “Parse Output Format” sections, a convention has been adopted that data will be present in normal cases where commands completed without errors and data has been returned successfully by the Device, unless otherwise stated. Data items which are not always present in successful Service Responses, e.g. a secondary element of an Electricity Smart Meter, will be indicated in the “Description/Valid Set” columns.

18.11 Schema Version

The MMC XML Schema has a schema version as an attribute of the main element GBCSResponse, called schemaVersion. Please note that the samples included in this document have not been updated to reflect the change in schemaVersion number.

In later versions of the schema it is possible that Service Requests may change to use different GBCS Use Cases and thus the MMC XML Schema will need to maintain XML definitions that support all GBCS Use Cases for backwards compatibility. The Parse Software will use the appropriate MMC XML Schema definitions to represent the data returned according to the GBCS Use Case in the payload.

The schema version will be constructed of a major and minor version. In development of version 1.0, this schema version will always be 1.0 and a separate DUIS/MMC development version will be notified within the schema comments.

Once version 1.0 is in use in the Production environment, the XML schema version will be updated with minor version increments (eg 1.1, 1.2 etc) for minor updates to the current baseline, whilst major version updates (eg 2.0) will be used for significant changes to the baseline – for example a new version of GBCS.

It is expected that the latest version of Parse Software will always support the parsing of all GBCS Use Cases and Device Alerts that exist in any version of GBCS and that these will always be added to over time and never replaced. This ensures that the latest version of Parse and Correlate software supports all SMETS2 or later devices for any version of the technical specifications