Handbuch für Hersteller und Betreiber von Smart-Meter Gateway-Administrations-Software

Smart Meter Gateway

Stand: 19.07.2022
Dokumentversion: 4.8

Power Plus Communications AG • Dudenstrasse 6 • 68167 Mannheim • Deutschland • www.ppc-ag.de

Inhalt

1 Einführung 4
1.1 Hintergrund 4
1.2 Copyright 4
1.3 Dokumentenreferenz 4
1.4 Gliederung des Dokuments 5
2 Beschreibung des Smart Meter Gateways 7
2.1 Wirkungsweise des Smart Meter Gateways 7
2.2 Rollenmodel des Smart Meter Gateways 8
2.3 Anschlüsse des Smart Meter Gateways 9
2.4 Leuchtdioden des Smart Meter Gateways 10
2.5 Gehäuse des Smart Meter Gateways 11
2.6 Betriebszustände des Smart Meter Gateways 15
3 Information für Smart Meter Gateway-Administratoren 16
3.1 Bedingungen zum Einsatz des SMGW 16
3.1.1 Anforderungen bei der Entgegennahme des Smart Meter Gateways 16
3.1.2 Anforderungen an Einsatzumgebungen, Qualifikation und Betrieb 17
4 Informationen zur Integration des *Smart Meter Gateways* in eine GWA-Software 22
4.1 Allgemeines 22
4.2 TLS und CMS-Inhaltsdatensicherung 26
4.3 RESTful-Cosem-Webservice 28
4.4 WKS1: „Management“-Webservice-Schnittstelle 33
4.4.1 WAF1: Administration und Konfiguration 34
4.4.2 Schnittstellenkonfiguration des SMGW 34
4.4.2.1 Konfiguration der WAN-, HAN- oder CLS-Schnittstelle 35
4.4.2.2 Konfiguration der LMN-Schnittstelle 40
4.4.3 Profilverwaltung und -konfiguration des SMGW 40
4.4.3.1 Verwaltung der Zählerprofile 42
4.4.3.2 Verwaltung der WAN- oder HAN-Profile 63
4.4.3.3 Verwaltung der TAF-Profile 85
4.4.3.4 Verwaltung der CLS-Profile 162
4.4.3.5 Pseudonymisierung 167
4.4.4 GWA-Zugriff auf das SMGW-Sicherheitsmodul 168
4.4.4.1 Personalisierung 182
4.4.4.2 Zertifikatswechsel 182
4.4.4.3 Root-Zertifikatswechsel 183
4.4.4.4 Gateway-Administrator-Wechsel 183
4.4.5 Verwaltung der Logdaten 185
4.4.6 Funktionen des Smart Meter Gateways 195
4.5 WKS2: „ADMIN-SERVICE“-Webservice-Schnittstelle 200
4.5.1 WAF2: Zugriff auf Dienste des GWA 200
4.5.2 WAF3: Alarmierung und Benachrichtigung 205
4.5.3 WAF4: Auslieferung von Messwerten und Netzzustandsdaten 206
4.6 WKS3: „Info-Report“-Schnittstelle zu einem EMT 206

4.6.1 WAF5: Übertragung von Daten an externe Marktteilnehmer 207
4.6.2 Statuswort der übermittelten Messwerte 207
4.7 WKS6: CLS-Tunnel-Funktionalität (WAF6) 210
4.8 WKS5: Zeitsynchronisation („NTP-TLS“) 213
4.9 WKS7: Wake-Up-Paket (WAF7) 216
5 Update-Funktionalitäten 219
5.1 Sicherer Kommunikationskanal zwischen GWH und GWA 219
5.2 Information des GWA über ein neues Firmware Update 219
5.3 Download, Prüfung und Entfernung der äußeren Signatur durch den GWA 219
5.4 Aufruf des Update Prozesses beim SMGW 220
6 Logging und Auditing 224
6.1 System-Log 224
6.2 Eichtechnisches Log 226
6.3 Eichrechtlich relevante Fehlermeldungen 227
7 Glossar 229
8 XML- und Cosem-Datentypen samt Mapping auf IEC 62056-62 231
9 Verzeichnis der im SMGW verwendeten *COSEM* Klassen 232
10 Verzeichnis der im SMGW verwendeten Object Identifier (OID) 234
11 Verzeichnis der kryptografischen Kennungen 236
12 Tabellenverzeichnis 237
13 Abbildungsverzeichnis 239
14 Literaturverzeichnis 240

1 Einführung

1.1 Hintergrund

Die zunehmend dezentrale Einspeisung erneuerbarer Energien stellt künftige Energieversorgungssysteme vor eine sehr große Herausforderung. Zum einen erfolgt die Energieeinspeisung durch erneuerbare Energien zu unvorhersehbaren Zeitpunkten, zum anderen können Energieverbräuche zu bestimmten Tageszeiten erhebliche Spitzenlasten erzeugen. Im Zuge der Einrichtung von intelligenten Energienetzen (*Smart Grids*) kommen intelligente Messsysteme (*Smart Metering Systems*) nach dem „Gesetz über den Messstellenbetrieb und die Datenkommunikation in intelligenten Energienetzen“ (MsbG, siehe https://www.gesetze-im-internet.de/messbg/) zum Einsatz.
Durch die Nutzung dieser gesetzlich vorgeschriebenen, in ein Kommunikationsnetz eingebundenen Messsysteme, erhalten Verbraucher eine höhere Transparenz über den eigenen Energieverbrauch und die Möglichkeit, das eigene Verbrauchsverhalten zu analysieren, um entsprechend die Energiekosten über den laufenden Verbrauch zu senken. Mit Hilfe moderner Tarife, die über das Messsystem abgebildet und ermöglicht werden, können Verbraucher ihren Energieverbrauch intelligent gestalten.

1.2 Copyright

1.3 Dokumentenreferenz

Diese Dokumentation bezieht sich auf das Produkt **Smart Meter Gateway**. Jedes **Smart Meter Gateway** enthält ein **SMGW Version 1.2.2 oder Version 2.0**.

Das **SMGW Version 1.2.2** besteht aus einer **SMGW Hardware Generation 1A** oder **Generation 1B** und einer der folgenden Software Versionen:

- **SMGW Software Version 1.1.3**
- **SMGW Software Version 1.1.2**
- **SMGW Software Version 1.1.1**
- **SMGW Software Version 1.1**
- **SMGW Integrationsmodul Software Version 1.0**

Das **SMGW Version 2.0** besteht aus einer **SMGW Hardware Generation 2A** und einer der folgenden Software Versionen:

- **SMGW Software Version 2.1.3**

Die korrekten Versionsangaben können Sie den Sicherheitsvorgaben [Security_Target] zu dem Ihnen vorliegenden Produkt, die auf der Webseite des Bundesamts für Sicherheit in der Informationstechnik zur Verfügung stehen, entnehmen.

Die genannten Software Versionen unterscheiden sich hinsichtlich ihres Funktionsumfangs; dieses Handbuch bezieht sich auf den aktuellsten Stand. Ein zentrales funktionales Merkmal sind die in der Software umgesetzten Tarifanwendungsfälle (TAF).

In **SMGW Integrationsmodul Software Version 1.0** sind folgende Tarifanwendungsfälle umgesetzt:

- TAF-1 *„Datensparsamer Tarif“*
- TAF-2 *„Zeitvariabler Tarif“*
- TAF-6 *„Ablesung von Messwerten im Bedarfsfall“*
- TAF-7 *„Zählerstandsgangmessung“*

Ab **SMGW Software Version 1.1** sind zudem die Tarifanwendungsfälle

- TAF-9 *„Abruf der Ist-Einspeisung“*
- TAF-10 *„Abruf von Netzzustandsdaten“*
- TAF-14 *„Hochfrequente Messwertbereitstellung für Mehrwertdienste“*

realisiert.

Weitere Details zu den Unterschieden bei den Software Versionen können über smgw-info@ppc-ag.de abgefragt werden.

Die folgenden Produktkonfigurationen enthalten eine **SMGW Hardware Generation 1A, Generation 1B** oder **Generation 2A**, auf der alle sicherheitsrelevanten Funktionen ausgeführt werden:

- **BPL Smart Meter Gateway** (BPL-SMGW)
- **CDMA Smart Meter Gateway** (CDMA-SMGW)
- **ETH Smart Meter Gateway** (ETH-SMGW)
- **GPRS Smart Meter Gateway** (GPRS-SMGW)
- **LTE Smart Meter Gateway** (LTE-SMGW), drei Ausführungen
- **powerWAN-ETH Smart Meter Gateway** (pWE-SMGW)
- **G.hn Smart Meter Gateway** (G.hn-SMGW)
- **LTE450 Smart Meter Gateway** (LTE450-SMGW)

Die Konfigurationen des Smart Meter Gateways unterscheiden sich durch die eingebaute WAN Kommunikationstechnologie und werden nachfolgend gemeinsam als Smart Meter Gateway(s) oder SMGW(s) bezeichnet.

1.4 Gliederung des Dokuments

Diese Dokumentation gliedert sich in zwei Teile. Zum einen sind hierin Informationen enthalten, die der Hersteller einer *Smart- Meter-Gateway-Administrations*-Software („GWA-Software“) benötigt, um das Produkt *Smart Meter Gateway* in eine „GWA-Software“ zu integrieren. Die Integration umfasst dabei alle Schritte, die vom Hersteller einer GWA-Software durchzuführen

sind, um das Smart Meter Gateway zu administrieren und Messdaten vom Smart Meter Gateway zu erhalten.
Die folgenden Kapitel, einschließlich ihrer Unterkapitel, sind dafür relevant:

- Kapitel 4 Informationen zur Integration des *Smart Meter Gateways* in eine GWA-Software
- Kapitel 5 Update-Funktionalitäten
- Kapitel 6 Logging und Auditing
- Kapitel 7 Glossar
- Kapitel 8 XML- und Cosem-Datentypen samt Mapping auf IEC 62056-62
- Kapitel 9 Verzeichnis der im SMGW verwendeten *COSEM* Klassen
- Kapitel 10 Verzeichnis der im SMGW verwendeten Object Identifier (OID)
- Kapitel 11 Verzeichnis der kryptografischen Kennungen

Zum anderen enthält diese Dokumentation Informationen, die der Hersteller einer „GWA-Software“ an deren Benutzer (also an einen *Smart- Meter-Gateway- Administrator* GWA) weiterzugeben hat.
Die folgenden Kapitel, einschließlich ihrer Unterkapitel, sind dafür relevant:

- Kapitel 3 Information für Smart Meter Gateway-Administratoren
- Alle in Kapitel 3 referenzierten Kapitel und Informationen

2 Beschreibung des Smart Meter Gateways

2.1 Wirkungsweise des Smart Meter Gateways

Die Aufgabe des *Smart Meter Gateways* besteht in der Speicherung der Messwerte der Strom-, Gas-, Wärme- und Wasserzähler und Berechnung der verschiedenen Tarifstufen anhand des Vertrages zwischen Letztverbraucher und Energieversorger und deren Versand an Ihren Energieversorger zur Abrechnung.
Das *Smart Meter Gateway* stellt Datenschutz und Datensicherheit für den Verbraucher sicher, indem nur ausgesuchte zur Rechnungserstellung notwendige Daten verschlüsselt an die externen Marktteilnehmer geschickt werden.

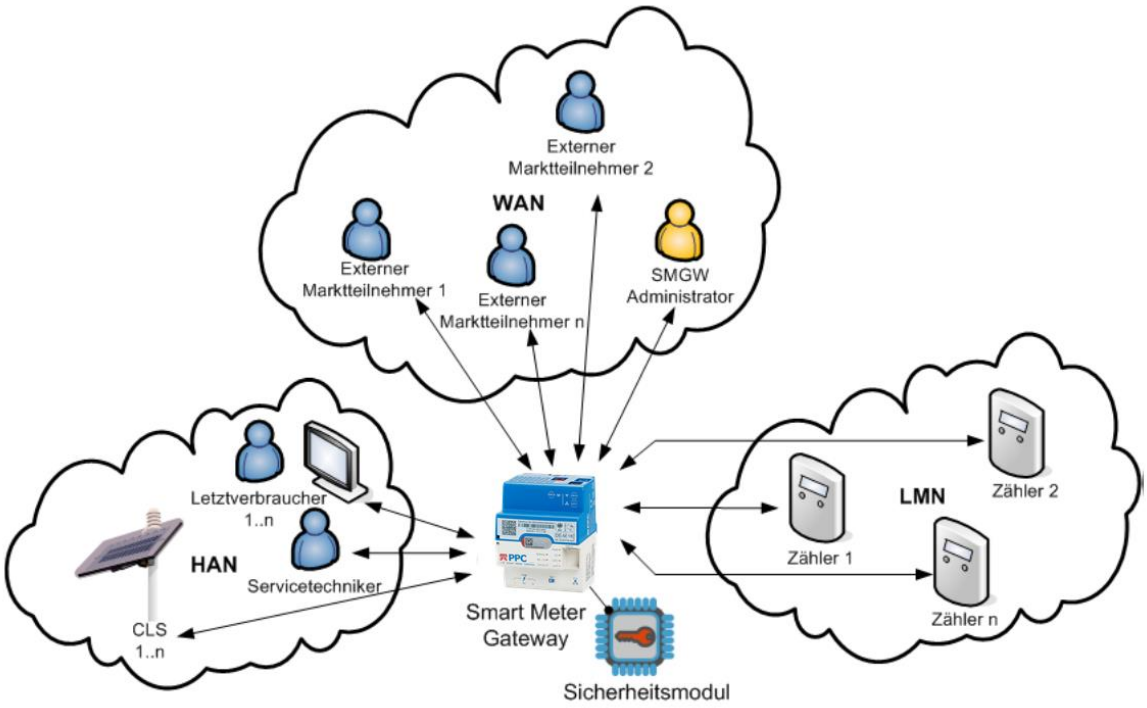


Abbildung 1: Einbettung des *Smart Meter Gateways* in seine Einsatzumgebung

Das *Smart Meter Gateway* bietet dabei u. a. Funktionen für Verbraucher, damit diese in ihrem Heimnetzwerk Verbrauchsdaten bzw. Systeminformationen abrufen können. Das *Smart Meter Gateway* kann mit steuerbaren Energieverbrauchern bzw. Energieerzeugern (sog. *Controllable Local Systems*, CLS) kommunizieren. Dabei handelt es sich zum Beispiel um intelligente Haushaltsgeräte oder Photovoltaik-Anlagen. So könnten zukünftig auch Elektrofahrzeuge als *Controllable Local Systems* dienen.

Abbildung 2: Das ETH *Smart Meter Gateway* (unterschiedliche Bedruckungsvarianten)

Abbildung 2 zeigt das Smart Meter Gateway mit seinen unterschiedlichen Schnittstellen. Im Detail werden diese Schnittstellen in Kapitel 2.3 beschrieben.

2.2 Rollenmodel des Smart Meter Gateways

Das *Smart Meter Gateway* arbeitet nach folgendem Rollenkonzept:

- *(Letzt-)Verbraucher* (*Consumer*): Der Verbraucher ist die natürliche oder juristische Person, die elektrische Energie, Gas, Wasser oder Wärme bezieht, bzw. mittels eines lokalen, dezentralen Erzeugers produziert. Der Verbraucher ist Eigentümer der im *Smart Meter Gateway* verarbeiteten und gespeicherten Messwerte. Unter die Rolle des Verbrauchers fallen somit der Anschlussnutzer, ggf. der Anschlussnehmer, bzw. der Anlagenbetreiber.
- *Externe (Markt-)Teilnehmer* (EMT, *Authorized External Entity*): Externe (Markt)-Teilnehmer sind aus Sicht des *Smart Meter Gateway* alle Teilnehmer mit Ausnahme des *Smart Meter Gateway Administrators* im WAN-Netz, mit denen das *Smart Meter Gateway* eine Kommunikation zum Austausch von Daten aufnehmen kann. Hierunter fallen also z.B. der Verteilnetzbetreiber (VNB), der Messstellenbetreiber (MSB), der Messdienstleister (MDL), der Lieferant (LF) und sonstige Dienstleister.
- *Smart Meter Gateway Administrator* (GWA): Das *Smart Meter Gateway* fällt in den Verantwortungsbereich des Messstellenbetreibers. Der GWA ist die vertrauenswürdige Instanz, die das *Smart Meter Gateway* installiert, konfiguriert, überwacht und steuert. Er erstellt und administriert die in das *Smart Meter Gateway* eingespielten Tarifprofile und führt bei Bedarf die Aktualisierung der Software des *Smart Meter Gateways* durch. Für jedes einzelne *Smart Meter Gateway* gibt es nur je einen GWA.
- *Service-Techniker*: Die Rolle des „Service-Technikers“ dient der Fehleranalyse am Aufstellungsort des *Smart Meter Gateways*. Hierzu erhält er die Möglichkeit, sich die Einträge des System-Logs sowie die Informationen zum Status des *Smart Meter Gateways* vor Ort anzeigen zu lassen. Hierbei ist ausschließlich ein lesender Zugriff auf die anzuzeigenden Daten möglich.

2.3 Anschlüsse des Smart Meter Gateways

Das SMGW liegt, wie in Kapitel 1.3 beschrieben, in verschiedenen Produkt-Konfigurationen vor. Diese enthalten jeweils unterschiedliche, integrierte WAN Kommunikationstechnologie und sind im Folgenden aufgelistet:

- die Produkt-Variante „*ETH Smart Meter Gateway*“ (kurz „*ETH-SMGW*“) mit WAN-seitiger Anbindung des SMGW per Ethernet,
- die Produkt-Variante „*powerWAN-ETH Smart Meter Gateway*“ (kurz „pWE-SMGW“) mit WAN-seitiger Anbindung des SMGW per Ethernet und zusätzlich Bereitstellung einer Versorgungsspannung über die Ethernet WAN-Schnittstelle zur Versorgung von angebundenen Peripheriegeräten (bspw. externe Kommunikationsmodems),
- die Produkt-Variante „*BPL Smart Meter Gateway*“ (kurz „*BPL-SMGW*“) mit WAN-seitiger Anbindung des SMGW per BPL[1] entsprechend des IEEE 1901[2] Standards,
- die Produkt-Variante „*G.hn Smart Meter Gateway*“ (kurz „*G.hn-SMGW*“) mit WAN-seitiger Anbindung des SMGW per BPL entsprechend des G.hn[3] Standards,
- die Produkt-Variante „*GPRS Smart Meter Gateway*“ (kurz „*GPRS-SMGW*“) mit WAN-seitiger Anbindung des SMGW per GPRS[4],
- die Produkt-Varianten „*LTE Smart Meter Gateway*“ (kurz „*LTE-SMGW*“) mit WAN-seitiger Anbindung des SMGW per LTE[5],
- die Produkt-Variante „*CDMA Smart Meter Gateway*“ (kurz „*CDMA-SMGW*“) mit WAN-seitiger Anbindung des SMGW per CDMA[6],
- die Produkt-Variante „*LTE450 Smart Meter Gateway*“ (kurz „*LTE450-SMGW*“) mit WAN-seitiger Anbindung des SMGW per LTE450[7].

In allen Produkt-Varianten des SMGW sind die folgenden physischen Schnittstellen vorhanden:

- physische Schnittstelle „PWR“ zum Anschluss der 230-V-Wechselstromversorgung mittels eines 3-poligen Stromkabels,
- physische Schnittstelle „HAN“ zum Anschluss von abgesetzten Anzeigeeinheiten für den Letztverbraucher oder Service-Techniker über Ethernet gemäß [IEEE 802.3i],
- physische Schnittstelle „CLS“ zur Kommunikation mit CLS-Geräten über Ethernet gemäß [IEEE 802.3i] unter Nutzung der Proxy-Funktionalität des SMGW,
- physische Schnittstelle „LMN-A“ zum Anschluss einer Antenne zwecks Kommunikation mit gemäß [EN 13757-4] („wM-Bus“) drahtlos arbeitenden intelligenten Zählern,
- physische Schnittstelle „LMN-1“ gemäß [EIA RS-485] zum Anschluss von gemäß [OMS-

1 Bzgl. des Begriffs „BPL“ s. https://en.wikipedia.org/wiki/Broadband_over_power_lines.

2 Bzgl. des Begriffs „IEEEE 1901“ s. https://de.wikipedia.org/wiki/IEEE_1901.

3 Bzgl. des Begriffs „G.hn“ s. https://de.wikipedia.org/wiki/G.hn.

4 Bzgl. des Begriffs „GPRS“ s. https://de.wikipedia.org/wiki/General_Packet_Radio_Service.

5 Bzgl. des Begriffs „LTE“ s. https://de.wikipedia.org/wiki/Long_Term_Evolution.

6 Bzgl. des Begriffs „CDMA“ s. https://de.wikipedia.org/wiki/CDMA2000

7 Bzgl. des Begriffs “LTE450” s. https://www.bundesnetzagentur.de/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Frequenzen/OeffentlicheNetze/450MHz/450MHz-node.html

2] drahtgebunden arbeitenden intelligenten Zählern.

Die an der „HAN“ und an der „CLS“ Schnittstelle bereitgestellten Dienste sind jeweils an beiden Schnittstellen verfügbar.
In der „ETH-Variante“ und der „powerWAN-ETH-Variante“ ist zusätzlich die physische Schnittstelle „WAN“ zum Anschluss an das WAN-Netz über Ethernet gemäß [IEEE 802.3i] vorhanden.
In der „GPRS-Variante“, in den „LTE-Varianten“ und in der „LTE450-Variante“ sind zusätzlich die folgenden physischen Schnittstellen vorhanden:

- physische Schnittstelle „WAN-A“ zum Anschluss einer Antenne zwecks Anbindung an das WAN-Netz per GPRS[8] oder LTE bzw. LTE450 und
- physische Schnittstelle „SIM“ zur Integration einer „Mini-SIM-Karte“[9] in das Produkt (falls eine Steck-SIM eingebracht werden kann).

In der „CDMA-Variante“ ist zusätzlich die folgende physische Schnittstelle vorhanden:

- physische Schnittstelle „WAN-A“ zum Anschluss einer Antenne zwecks Anbindung an das WAN-Netz per CDMA.

2.4 Leuchtdioden des Smart Meter Gateways

Alle Varianten des *Smart Meter Gateways* zeigen ihre Betriebszustände über drei grüne LEDs und eine blaue LED an. Abhängig von der Variante sind weitere LEDs vorhanden, die eine Aussage zum Zustand des WAN-Kommunikationsmoduls machen.
Wie in Abbildung 2 dargestellt, verfügt das *Smart Meter Gateway* in allen Varianten über folgende LEDs:

- die „PWR“-LED (*Power*) als allgemeine Betriebsanzeige (grün),
- die „TLS“-LED (*Transport Layer Security*) zur Anzeige einer gesicherten, verschlüsselten Verbindung (grün),
- die „LMC“-LED (*Local Meter Connect*) zur Anzeige des Datenverkehrs mit drahtgebundenen Zählern (grün),
- die „wMT“-LED (*wMBus-Traffic*) zur Anzeige des Datenverkehrs mit drahtlosen Funkzählern (blau).

Sobald das *Smart Meter Gateway* an das elektrische Netz angeschlossen oder die Energieversorgung nach einem Stromausfall wiederhergestellt wird, blinken alle vier LEDs für einen Zeitraum von drei Sekunden gleichzeitig kurz auf. Danach werden sie für die Dauer des weiteren Startvorganges ausgeschaltet. Sobald das Betriebssystem bereit ist, beginnt die LED mit der Bezeichnung „PWR“ mit einer Periode von 0,5 Sekunden zu blinken. Die „PWR“-LED leuchtet schließlich dauerhaft, wenn das *Smart Meter Gateway* vollständig betriebsbereit ist.

8 Bzgl. der GPRS-Spezifikation siehe unter http://www.3gpp.org/DynaReport/44060.htm.
9 Vgl. https://de.wikipedia.org/wiki/SIM-Karte.

Die „TLS“-LED leuchtet dauerhaft, solange eine verschlüsselte Verbindung zu einem externen Teilnehmer besteht.

Die „LMC“-LED leuchtet durchgängig, sobald mindestens ein drahtgebundener Zähler an das Smart Meter Gateway angeschlossen ist und vom *Smart Meter Gateway* eine HDLC (*High Level Data Link Control*) Adresse erhalten hat.

Die „wMT“-LED leuchtet auf, sobald Daten von einem drahtlosen Funkzähler empfangen werden. Sie leuchtet während der Verarbeitung des Datenpakets des Funkzählers für maximal 250 ms auf.

Während der Installation eines Software-Updates blinken die „PWR“-, „TLS“- und „wMT“-LED gleichzeitig. Um einen korrekten Installationsablauf zu gewährleisten, sollte das *Smart Meter Gateway* während dieser Phase nicht von der Versorgungsspannung getrennt werden.

Sollte eines der folgenden Blinkverhalten auftreten, ist eine sogenannte *Sicherheitsschutzverletzung* aufgetreten:

- Die „PWR“- und die „TLS“-LED blinken dauerhaft gleichzeitig.
- Die „PWR“- und die „LMC“-LED blinken dauerhaft gleichzeitig.
- Die „PWR“-, die „TLS“- und die „LMC“-LED blinken dauerhaft gleichzeitig.
- Die „PWR“- und die „wMT“-LED blinken dauerhaft gleichzeitig oder zeitlich versetzt.
- Alle vier LEDs blinken dauerhaft gleichzeitig.

Das *Smart Meter Gateway* ist nicht mehr vollständig einsatzfähig und muss gewartet bzw. ausgetauscht werden.

Falls keine LED mehr leuchtet liegt ein Defekt des Geräts oder der Spannungsversorgung vor. Unter Umständen konnten auch die grundliegenden Software-Komponenten auf dem Gerät nicht gestartet werden. In diesem Fall ist das *Smart Meter Gateway* nicht mehr einsatzfähig. Der Fehler muss vor Ort analysiert und behoben werden, ggf. muss das Gerät ausgetauscht werden.

2.5 Gehäuse des Smart Meter Gateways

Anhand eines Siegels auf dem Gehäuse des *Smart Meter Gateways* lassen sich physische Manipulationen erkennen, da sich das Gehäuse des Geräts nicht ohne Siegelbruch öffnen lässt. Eine Gesamtabbildung des Geräts, in der auch die Positionierung des Siegels erkennbar ist, ist in Abbildung 4 dargestellt.

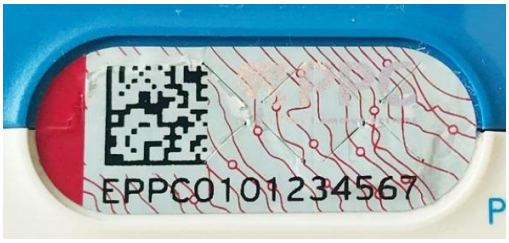
Das Siegel zeigt folgende Elemente auf graugemustertem Hintergrund mit roten Linien:

- ID des *Smart Meter Gateways*: Die ID beginnt stets mit „EPPC“ und ist eine, für jedes *Smart Meter Gateway* eindeutige Kennung.

- 2D-Barcode: Der 2D-Barcode enthält die ID des *Smart Meter Gateways.* Für das Gateway mit der ID „EPPC0210487462“ ergibt sich beispielsweise folgender 2D-Barcode:

Abbildung 3: Beispiel eines 2D-Barcodes

- Hologramm: Als Hologramm ist auf dem Siegel das Logo der Power Plus Communications AG aufgebracht.

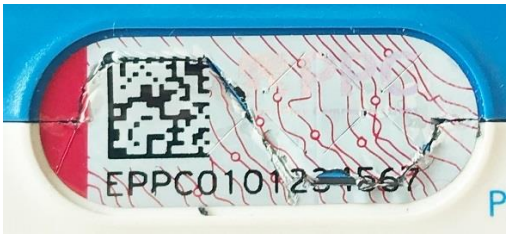


✓ Keine Beschädigung am Siegel zu erkennen. Rote Linien auf grau-gemustertem Hintergrund.

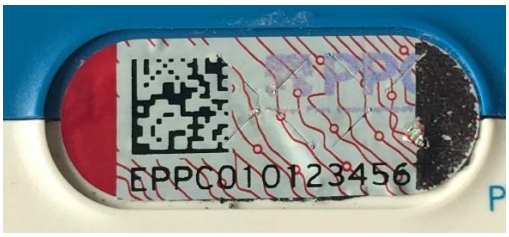
Bei Erwärmung verschwindet die rote Musterung.

✗ Das Siegel zeigt die schwarze Schriftzugmusterung „PPC“. Es wurde versucht, das Siegel abzuziehen. **Das Smart Meter Gateway darf nicht weiterverwendet werden!**

✗ Auf dem Siegel ist der 2D-Barcode und die ID des Smart Meter Gateways nicht mehr zu erkennen. Es wurde versucht, das Siegel abzuziehen. **Das Smart Meter Gateway darf nicht weiterverwendet werden!**



✗ Das Siegel wurde mit einem scharfen Gegenstand in der Mitte durchtrennt**. Das Smart Meter Gateway darf nicht weiterverwendet werden!**



Das Siegel zeigt eine schwarze Markierung an der rechten Seite. Es wurde versucht, das Siegel abzuziehen. **Das Smart Meter Gateway darf nicht weiterverwendet werden!**

Abbildung 4: Zustände des Siegels des Smart Meter Gateways

Das Siegel ist dabei derart angebracht, dass es direkt und ohne Werkzeug einer Sichtprüfung unterzogen werden kann. Damit können Sie die physische Unversehrtheit des *Smart Meter Gateways* prüfen.

Auf der Vorderseite des SMGWs oberhalb des Siegels (siehe Abbildung 2) können Sie den Typenschlüssel des SMGWs ablesen. Der Typenschlüssel enthält Informationen über die im *Smart Meter Gateway* verbauten Hardware Komponenten.

Bei einem zertifizierten SMGW ist abhängig von der Gerätevariante einer der folgenden Typenschlüssel aufgebracht:

- SMGW-B-1A-111-00
- SMGW-B-1B-111-00
- SMGW-B-2A-111-00
- SMGW-L-1A-111-30
- SMGW-L-1B-111-30
- SMGW-L-1A-111-10
- SMGW-L-1B-111-10
- SMGW-G-1A-111-30
- SMGW-C-1A-111-00
- SMGW-P-1B-111-00
- SMGW-E-1A-111-00
- SMGW-E-1B-111-00
- SMGW-E-2A-111-00
- SMGW-N-1B-111-00
- SMGW-N-2A-111-00
- SMGW-V-1B-111-20
- SMGW-V-1B-111-10
- SMGW-V-2A-111-20
- SMGW-K-1B-111-10
- SMGW-K-1B-111-20
- SMGW-K-1B-111-30
- SMGW-K-2A-111-10
- SMGW-K-2A-111-30
- SMGW-J-2A-111-30
- SMGW-J-2A-111-10

Ein zertifiziertes *Smart Meter Gateway* kann zusätzlich durch den Schriftzug „Smart Meter Gateway“ identifiziert werden, der auf Vorderseite des SMGWs aufgedruckt ist. Abhängig von der jeweiligen Produkt-Konfiguration ist entweder „BPL Smart Meter Gateway“, „G.hn Smart Meter Gateway“, „LTE Smart Meter Gateway“, „GPRS Smart Meter Gateway“, „CDMA Smart Meter Gateway“, „pWE Smart Meter Gateway“, „LTE450 Smart Meter Gateway“ oder „ETH Smart Meter Gateway“ aufgedruckt.

Bitte prüfen Sie den Typenschlüssel und den Schriftzug „Smart Meter Gateway“. Falls eines der beiden Beschriftungselemente nicht den beschriebenen Kennungen entspricht, darf das Gerät NICHT mehr verwendet werden und NICHT beim Letztverbraucher verbleiben.

2.6 Betriebszustände des Smart Meter Gateways

Das *Smart Meter Gateway* kennt folgende Betriebszustände:

Tabelle 1: Betriebszustände des SMGW

Betriebs-zustand	Ursache	Auswirkung
Regel-betrieb	• Erfolgreicher Start des Smart Meter Gateways	• Smart Meter Gateway funktioniert korrekt
Einge-schränkter Betrieb	• Wesentliche Komponenten der Software des Geräts starten nicht • Systemzeit ist ungültig • Speicherkapazität des Logbuchs für den Kunden ist erschöpft • Speicherplatz des Smart Meter Gateways ist erschöpft • Zertifikat zur verschlüsselten Kommunikation mit dem Zähler ist nicht mehr gültig	• Keine Messwerterfassung • Keine Messwertverarbeitung • Kein Messwertversand • Die Logmeldung „SMGW_LIMITEDOPERATION_ON“ gemäß [SMGW_Logging] wird geschrieben
Minimal-betrieb	• Speicherkapazität des Eichlogbuchs ist erschöpft • Integritätsverletzung der Datenbank wurde festgestellt	• Abschaltung der HAN Schnittstelle • Keine Messwerterfassung • Keine Messwertverarbeitung • Kein Messwertversand • Updatefähigkeit wird deaktiviert • Die Logmeldung „SMGW_MINIMALOPERATION_ON“ gemäß [SMGW_Logging] wird geschrieben
Eingestellter Betrieb	• Wesentliche Komponenten der Software des Geräts starten nicht und kann nicht neu gestartet werden	• Netzwerkschnittstellen werden deaktiviert • Alle Dienste des Smart Meter Gateways werden abgeschaltet

	• Verdacht auf eine Sicherheitsschutzverletzung liegt vor	• LEDs signalisieren eine Sicherheitsschutzverletzung (siehe Kapitel 2.4)

Sie können an den Logeinträgen des SMGWs erkennen, ob der Regelbetrieb, der eingeschränkte Betrieb oder der Minimalbetrieb eingetreten ist. Bei Eintritt des eingestellten Betriebs ist es nicht mehr möglich eine Kommunikationsverbindung zu dem SMGW zu etablieren, daher können Sie auch keine Logmeldungen mehr auslesen oder Benachrichtigungen von diesem SMGW erhalten. Falls das SMGW dauerhaft kommunikativ nicht erreichbar ist, ist das ein Hinweis auf den eingestellten Betrieb. Veranlassen Sie in diesem Fall einen Service-Techniker Einsatz zur Prüfung der LEDs vor Ort.

Falls der Minimal- oder der eingestellte Betrieb eingetreten ist, müssen Sie den Austausch des Geräts veranlassen. Falls der eingeschränkte Betrieb eingetreten ist, müssen Sie die Ursache für den Eintritt dieses Zustands beheben (Gültige Zeitinformation zur Synchronisation bereitstellen, Größe der Logbücher für den Letztverbraucher erweitern, neue LMN Zertifikate erzeugen) oder ebenfalls einen Austausch des Geräts veranlassen.

3 Information für Smart Meter Gateway-Administratoren

Die nachfolgenden Informationen sind vom Hersteller einer *Smart-Meter-Gateway-Administrator*-Software („GWA-Software“) an deren Benutzer (also an einen *Smart-Meter-Gateway-Administrator* GWA) weiterzugeben.

3.1 Bedingungen zum Einsatz des SMGW

Bitte stellen Sie sicher, dass für den zugelassenen Einsatz des SMGW die folgenden Bedingungen erfüllt sind.

3.1.1 Anforderungen bei der Entgegennahme des Smart Meter Gateways

Abgleich Lieferschein: Falls vom SMGW-Hersteller *Smart Meter Gateways* an Sie geliefert werden, müssen Sie die Lieferung hinsichtlich Stückzahl, Produkt-Variante und IDs der Gateways mit dem Lieferschein abgleichen. Falls die Angaben im Lieferschein nicht mit der Lieferung übereinstimmen, wenden Sie sich bitte an den SMGW-Hersteller. **Die *Smart Meter Gateways* aus dieser Lieferung dürfen in diesem Fall nicht verbaut werden.**

Beachtung der Maßnahmen zur sicheren Auslieferung: Falls vom SMGW-Hersteller *Smart Meter Gateways* an Sie geliefert werden, müssen Sie alle Maßnahmen beachten und durchführen, die im Anhang [Sichere_Auslieferung] definiert sind. Sie dürfen nur *Smart Meter Gateways* entgegennehmen, für die vom SMGW-Hersteller alle im Anhang [Sichere_Auslieferung] definierten Maßnahmen umgesetzt wurden.

Einsatz eines zertifizierten Geräts: Prüfen Sie, ob das erhaltene Gerät ein zertifiziertes Gerät darstellt. Zur Prüfung der Hardware kontrollieren Sie das Siegel und die Angabe des Typs entsprechend der Anweisung in Kapitel 2.5. **Falls Sie bei der Prüfung des Siegels oder der Angabe des Typs Auffälligkeiten feststellen, ist das Gerät NICHT mehr einsatzbereit.**

Falls Ihnen das Gerät nicht physisch vorliegt, prüfen Sie bitte bei der Kommunikation mit dem Gerät (TLS-Verbindungen, Signatur- und Verschlüsselung der CMS-Container) ob die Zertifikate des Geräts gegen das Root-Zertifikat der SM-PKI validiert werden können. Diese Prüfung ist vor allem vor der Personalisierung wichtig, solange die Zertifikate mit den Gütesiegelzertifikaten kommunizieren, die durch den Hersteller in der Vorpersonalisierung aufgebracht wurden. **Falls das Gerät mit Zertifikaten kommuniziert, die sich nicht gegen das Root-Zertifikat der SM-PKI validieren lassen, ist das Gerät NICHT einsatzbereit.** Die Kommunikation zu diesem Gerät muss eingestellt werden und der Austausch des Gerätes muss veranlasst werden.

Weiterhin können Sie über die COSEM-IC *smgw_info* die aktuelle Firmware Version Ihres *Smart Meter Gateways* auslesen. Bitte prüfen Sie, ob diese Firmware Version dem zertifizierten Stand entspricht. Die korrekte Firmware Version können Sie den Sicherheitsvorgaben / Security Target (siehe [Security_Target]) zu dem Ihnen vorliegenden Produkt, die auf der Webseite des Bundesamts für Sicherheit in der Informationstechnik zur Verfügung stehen, entnehmen. **Falls eine Firmware Version installiert ist, die nicht dem zertifizierten Stand entspricht, darf das Gerät nicht beim Letztverbraucher verbleiben und ist NICHT einsatzbereit.**

Alle Geräte, die in der Vorpersonalisierung mit den Kommunikationsparametern Ihres GWA-Systems ausgestattet wurden, beim Letztverbraucher eingebaut sind und kommunikativ (WAN-Schnittstelle) ausreichend angebunden sind, müssen eine Kommunikationsverbindung zu Ihrem GWA-System aufbauen. Falls in diesem Fall von einem Gerät keine Kommunikationsverbindung aufgebaut wird, muss eine Analyse vor Ort veranlasst werden. **Falls keine plausible Ursache festgestellt werden kann, darf das Gerät nicht beim Letztverbraucher verbleiben und ist NICHT einsatzbereit.**

Bitte informieren Sie bei jeglichen Auffälligkeiten, die einen Rückschluss auf einen sicherheitsrelevanten Fehler oder auf den Einsatz eines nicht zertifizierten Gerätes zulassen umgehend den Hersteller des SMGWs. Ein Gerät, bei dem ein derartiger Verdacht besteht, darf auf keinen Fall weiterhin eingesetzt werden.

3.1.2 Anforderungen an Einsatzumgebungen, Qualifikation und Betrieb

Durchführung des Geräte-Selbsttests:

Starten Sie den Selbsttest, indem Sie die entsprechende Methode über die WAN-Schnittstelle des SMGW aufrufen (s. *IC smgw_info*). Das SMGW schickt nach Abschluss des Selbsttests einen entsprechenden Event an den GWA. Falls das Event ausbleibt oder über einen fehlgeschlagenen Selbsttest informiert, ist das Gerät NICHT mehr einsatzbereit.
Der Selbsttest wird auch automatisch bei jedem Neustart des Gerätes durchgeführt. Auch in

diesem Fall wird der GWA über den Erfolg des Selbsttests mittels eines Events informiert. Der GWA kann den Neustart auch über einen Aufruf der entsprechenden Methode auslösen (s. IC *smgw_info*) und somit ebenfalls einen Selbsttest veranlassen.
Zudem wird ein Neustart verbunden mit einem Selbsttest zyklisch alle 28 Tage selbstständig durch das SMGW durchgeführt.

Qualifizierung des GWA:

Sie dürfen das Produkt nur betreiben und administrieren, wenn Sie die notwendigen Schulungen und Zertifizierungen absolviert haben und entsprechend vertrauenswürdig sind.

Konfiguration des *Smart Meter Gateways*:

Wenn Sie Konfigurationsdaten, beispielsweise in Form von Profilen in das Gerät einbringen, müssen Sie sicherstellen, dass diese Konfigurationsdaten aus vertrauenswürdigen und verlässlichen Quellen stammen.

Übermittlung von Messdaten:

Sie müssen bei der Konfiguration des Geräts gewährleisten, dass abrechnungsrelevante oder personenbezogene Daten nur an entsprechend autorisierte und vertrauenswürdige externe Marktteilnehmer übermittelt werden.

Sicherheitsmodul:

Sie müssen gewährleisten, dass das von Ihnen verbaute Gerät über ein zertifiziertes Sicherheitsmodul verfügt. Falls es sich bei dem vorliegenden Gerät um ein zertifiziertes Gerät handelt (entsprechende Möglichkeiten zur Prüfung sind ebenfalls in diesem Kapitel beschrieben), können Sie davon ausgehen, dass auch ein zertifiziertes Sicherheitsmodul im Gerät verbaut ist und genutzt wird. Sie können die direkte Kommunikation mit dem Sicherheitsmodul auch testen, indem Sie den AUTH-Zustand über den Aufruf der entsprechenden Methoden (s. IC *security_module*) herstellen und wieder terminieren.

Firmware Update:

Sie müssen sicherstellen, dass alle Firmware Updates die auf das Gerät ausgebracht werden, zertifiziert sind und von einer autorisierten und vertrauenswürdigen Bezugsquelle stammen. Genaue Anweisungen zur Übermittlung und Prüfung eines Firmware Updates können Sie der Beschreibung in Kapitel 5 entnehmen.

Schlüsselerzeugung zur Kommunikation im LMN:

Sie müssen sicherstellen, dass die Schlüsselpaare zur Kommunikation mit dem Zähler sicher im Gateway eingebracht werden bzw. dort erzeugt werden. Zur Erzeugung der Schlüssel des *Smart Meter Gateways* müssen Sie die entsprechenden Methoden der IC *security_module* aufrufen. Der initiale Schlüssel des Zählers muss als Attribut des Zählerprofils (s. IC *kaf_lmn_container*) in das *Smart Meter Gateway* eingebracht werden.

Beistellung Authentifizierungsinformationen an Letztverbraucher und Service-Techniker:

Letztverbrauchern und Service-Technikern, die über die HAN-Schnittstelle auf das *Smart Meter Gateway* zugreifen, müssen Sie die Zugangsinformationen (Zertifikatscontainer oder Username/Passwort) über einen sicheren Kommunikationskanal bereitstellen. Darüber hinaus müssen Sie bei entsprechender Nachfrage das Zertifikat des *Smart Meter Gateways* über diesen Kommunikationskanal bereitstellen, das zum Aufbau des TLS-Kanals verwendet wird. Service-Techniker und Letztverbraucher können dieses Zertifikat nutzen, um sicherzustellen, dass sie tatsächlich mit dem gewünschten *Smart Meter Gateway* kommunizieren.

Benachrichtigung des Herstellers bei bestimmten Logmeldungen und bei sicherheitsrelevanten Fehlern:

Benachrichtigen Sie den Hersteller immer, wenn

- Logmeldungen im System- oder Eichlog eingetragen sind, die Sie nicht nachvollziehen können und die auf ein Fehlverhalten des *Smart Meter Gateways* schließen lassen.
- Benachrichtigungen vom *Smart Meter Gateway* übermittelt werden, die Sie nicht nachvollziehen können und die auf ein Fehlverhalten des *Smart Meter Gateways* schließen lassen.
- vom Service-Techniker ein sicherheitsrelevantes Ereignis gemeldet wird.
- die Logmeldung SEC_UNALLOWEDACCESS (siehe [SMGW_Logging]) im Systemlog des SMGW verzeichnet ist.
- Auffälligkeiten feststellbar sind, die relevant für die Sicherheit des SMGWs sein könnten.

Bitte senden Sie zusätzlich dazu eine E-Mail an die folgende Adresse: security@ppc-ag.de. Die E-Mail sollte Informationen zur Version des verwendeten Gerätes enthalten. Die Version können Sie auf dem Gehäuse des Smart Meter Gateways oder über die WAN-Schnittstelle aus dem Objekt ***smgw_info*** auslesen. Bitte beschreiben Sie das festgestellte Problem in der E-Mail möglichst ausführlich und machen Sie alle notwendigen Angaben, damit der Hersteller des Smart Meter Gateways den Fehler nachstellen kann.

Einstellen des Messbetriebs:

Wenn Sie aufgrund einer entsprechenden Logmeldung oder eines Events feststellen, dass das *Smart Meter Gateway* den Messbetrieb eingestellt hat, prüfen Sie bitte umgehend Ihnen zugängliche Logbücher des *Smart Meter Gateways* und identifizieren Sie die Ursache. Sorgen Sie dafür, dass der Messbetrieb zeitnah wiederaufgenommen wird. Abhängig von der Ursache für dieses Verhalten können Sie das Problem remote lösen oder müssen einen Service-Techniker-Einsatz vor Ort veranlassen.

Gültigkeit der konfigurierten Zertifikate:

Achten Sie bei den konfigurierten Zertifikaten auf den Gültigkeitszeitraum und darauf, dass alle auf dem *Smart Meter Gateway* konfigurierten Zertifikate stets gültig sind. Falls das *Smart Meter Gateway* am konkreten Einsatzort über längere Zeit kommunikativ nicht erreichbar ist, haben Sie in diesem Zeitraum auch keine Möglichkeit die Zertifikate zu aktualisieren. Falls

der Gültigkeitszeitraum der Zertifikate des GWAs, die auf dem *Smart Meter Gateway* hinterlegt sind, abgelaufen ist, kann keine Kommunikationsverbindung zwischen *Smart Meter Gateway* und GWA aufgebaut werden. In diesem Fall kann das *Smart Meter Gateway* nicht mehr genutzt werden und muss ausgetauscht werden.

Signatur Wake-Up Paket:

Zur Bildung der Signatur im Wake-Up-Paket müssen Sie stets das aktuell gültige Signatur Zertifikat (Tabelle 103, GWADM_SIG_PRV) verwenden. Beachten Sie hierzu auch die Vorgaben aus Kapitel 4.9.

Bereitstellung Informationen an den EMT:

Sie müssen die EMTs, die Sie als Messwertempfänger im *Smart Meter Gateway* konfigurieren, in die Lage versetzen, die zugesendeten Messwerte zu verarbeiten. Übermitteln Sie den EMTs dazu die Informationen aus Kapitel 4.6. Falls der EMT keine Messwerte entgegennimmt, sondern über das *Smart Meter Gateway* mit CLS-Geräten kommuniziert, stellen Sie ihm bitte die Informationen aus 4.7 zur Verfügung.

Konfiguration und Bereitstellung von Diensten zum Betrieb des SMGWs:

Sie sind für den sicheren Betrieb des *Smart Meter Gateways* verantwortlich. Führen Sie daher alle Operationen, die zur Konfiguration des *Smart Meter Gateways* notwendig sind, wie in Kapitel 4.4 beschrieben, aus. Weiterhin sind Sie verpflichtet, folgende Dienste bereitzustellen:

- Dienst zur Zeitsynchronisation des Smart Meter Gateways, der den Vorgaben aus Kapitel 4.8 genügt
- Dienst zum Empfang und zur Verarbeitung von Benachrichtigungen, tarifierten Messwerten und Netzzustandsdaten vom *Smart Meter Gateway,* der den Vorgaben aus Kapitel 4.5 genügt
- Dienst zur Durchführung eines Firmware Updates, der den Vorgaben aus Kapitel 5 genügt.

Abruf von Logdaten:

Rufen Sie die in Kapitel 6 beschriebenen Logdaten regelmäßig ab und kontrollieren Sie die Logdaten auf Auffälligkeiten und sicherheitsrelevante Ereignisse. Beachten Sie hierbei auch den Anhang [SMGW Logging], der Ihnen mit diesem Handbuch übermittelt wurde und die Informationen zum Verhalten beim Auftreten bestimmter Logmeldungen, die während der Integrationstests vom Hersteller des *Smart Meter Gateways* geschult wurden. Kontaktieren Sie bei unklaren oder nicht plausiblen Logmeldungen den Hersteller des *Smart Meter Gateways*.

Personalisierung:

Sobald das *Smart Meter Gateway* die erste Kommunikationsverbindung zum GWA-System aufgebaut hat, muss die Personalisierung eingeleitet werden. Gehen Sie dazu wie in Kapitel 4.4.4.1 beschrieben vor. Smart Meter Gateways die nicht personalisiert werden können sind NICHT einsatzbereit und dürfen NICHT beim Letztverbraucher verbleiben.

Mess- und eichrechtkonforme Verwendung:

Für eine mess- und eichrechtkonforme Verwendung müssen die Angaben im Dokument "Betriebshinweise für eine mess- und eichrechtkonforme Verwendung" beachtet und umgesetzt werden. Sie können die Betriebshinweise unter https://service.ppc-ag.de abrufen oder unter smgw-info@ppc-ag.de erfragen.

4 Informationen zur Integration des *Smart Meter Gateways* in eine GWA-Software

4.1 Allgemeines

Abhängig von der entsprechenden Produktkonfiguration verwendet das SMGW zur Kommunikation ins WAN eine der in Kapitel 2.3 aufgeführten Technologien. Über den Kommunikationstechnologie-abhängigen unterlagerten Protokollschichten verwendet das SMGW das „*Ethernet Address Resolution Protocol*" (ARP) gemäß [RFC 826], das IPv4-Protokoll gemäß [RFC 791], alternativ das IPv6-Protokoll gemäß [RFC°2460] und das TCP-Protokoll gemäß [RFC 793].

Bereits in der Produktion wird festgelegt, ob das SMGW an der „WAN"-Schnittstelle das IPv4- oder das IPv6-Protokoll verwendet. Diese Konfiguration kann nach der Auslieferung nicht mehr geändert werden. Wenn das IPv4-Protokoll genutzt wird, können die Mechanismen zum Adressbezug gemäß Kapitel 4.4.2.1 konfiguriert werden. Wenn das IPv6-Protokoll genutzt wird, wird stets „*Stateless Address Autoconfiguration*" gemäß [RFC°4862] zum Adressbezug eingesetzt.

Das SMGW benutzt als „Basisprotokoll" der Anwendungsschicht das „*Hypertext Transfer Protocol*" HTTP/1.1 gemäß [RFC 2616] einschließlich aller in [BSI TR-03109-1] spezifizierten Einschränkungen und Erweiterungen.

Das *Smart Meter Gateway* verwendet dabei die HTTP-Header und HTTP-Statuscodes aus [RFC 2616], die in [BSI TR-03109-1] aufgeführt sind. Die HTTP-Header des *Smart Meter Gateways* sind dabei wie folgt:

Request-Header:

- *Content-Type:* Gibt den Typ des Request-Bodies an. Ist ein Request-Body nach dieser Spezifikation vorhanden, ist dieser Header verpflichtend.
  1. Der Content-Type „*application/octet-stream*" wird für Binärdaten verwendet.
  2. Der Content-Type „*application/vnd.de-dke-k461-cosem+xml{;parameters}*" wird für COSEM-XML-Dateien mit XML-Schema-Bindung ohne Klassentypisierung der COSEM-Objekte verwendet.
  3. Der Content-Type „*application/vnd.de-dke-k461-ic1+xml{;parameters}*" wird für COSEM-XML-Dateien mit Bindung an ein XML-Schema nach Klassensatz AK142 V1.0 („ic1") verwendet.

  Der *Content-Type*-Header kennt dabei für den Parameter „*encap*" folgende Werte:
    - den Standardwert „*cms-tr03109-zlib*" für CMS-Dateien mit ECKA-EG nach [BSI TR-03109] mit innerem Content-type „*id Compressed*" sowie
    - den Wert „*cms-tr03109*" für CMS-Dateien mit ECKA-EG nach [BSI TR-03109].
- *Host*: Identifiziert den Host, an den die Anfrage gerichtet ist. Dies entspricht der host-Identifikation der HTTP-URI. HTTP-Requests, die das SMGW empfängt, müssen im *host*-Header die kanonische Geräte-ID des SMGW enthalten. Das SMGW muss beim Empfang eines Requests prüfen, ob diese Bedingung erfüllt ist. Falls diese Prüfung fehlschlägt, wird die Anfrage mit dem HTTP-Fehler „*403 Forbidden*" abgelehnt.

- *Accept*: Teilt dem Server mit, welche Content-Types im Body einer Response erlaubt sind. Falls kein *Accept*-Header vorhanden ist, wird die Verwendung des Content-Type „*application/de-dke-k461-ic1+xml;encap=cms-tr03109-zlib*“ angenommen.
- *HTTP Kompression*: Das SMGW unterstützt unkomprimierte HTTP-Body. Bei Verwendung eines verschlüsselten Inhaltstyp im HTTP-Body, ist eine Kompression vor der Verschlüsselung sinnvoll, die durch den Content-Type „*application/de-dke-k461-ic1+xml;encap=cms-tr03109-zlib*“ angegeben wird.

Response-Header:

- *Content-Type*: (analog zum entsprechenden Request-Header)
- *Content-Range*: In der Antwort auf eine Anfrage mit *Range*-Request-Header liefert der Server den HTTP-Status-Code „*206 Partial Content*“ und gibt im *Content-Range*-Response-Header an, welchen Teil (in Bytes) einer Ressource der Content-Body enthält. Typischerweise verwendet der GWA diese Antwort bei der Übermittlung eines Firmware Updates.
- *Retry-after Header*: Der Server teilt in der Antwort nach einem (Request-Timeout-) Fehler mit, wann der Client frühestens einen neuen Request schicken soll.

Unbekannte HTTP-Header-Felder werden vom SMGW ignoriert, d.h. nicht ausgewertet. Der Vergleich der Header-Bezeichner wird dabei Case-insensitiv durchgeführt.

Ein Pipelineing von HTTP-Requests ist nicht erlaubt, d.h. bevor ein neuer Request innerhalb einer HTTP-Session gesendet wird, muss eine HTTP-Response vorliegen.

Ein HTTP-Blocktransfer ist nur für die idempotenten Operationen PUT und GET zugelassen. Die Operation gilt erst nach der Übertragung des letzten Blockes als abgeschlossen.

Der URI-Baum enthält „Listen-Ressourcen“. Der Name einer Listen-Ressource endet mit dem Buchstaben „*s*“. Der Name des Elementes einer Listen-Ressource entspricht dem Pfadnamen in der URI. Die Liste kann leer sein. Die Unterelemente einer Listen-Ressource besitzen den Namen der Listenressource ohne mit dem Buchstaben „**s“** zu enden. Listen-Ressourcen sind *objects* (wie z.B. *ldevs*), *attributes* und *methods*. Ein GET-Request auf eine Listen-Ressource liefert eine Liste der Elemente unter dieser Ressource.

Die zulässigen HTTP-Operationen sind in Tabelle 5 beschrieben.

TLS-Verbindungen:

Das SMGW sichert seine Kommunikationsverbindungen gemäß den Anforderungen des Schutzprofils [SMGW_PP] oberhalb der TCP-Transportschicht generell mittels TLS v.1.2 [RFC 5246] ab. Für die Kommunikation mit Teilnehmern im WAN befinden sich das SMGW immer in der Rolle des TLS-Client und die Gegenstelle immer in der Rolle des TLS-Servers. Die Authentisierung erfolgt dabei stets beidseitig zertifikatsbasiert. Die Zertifikate stammen aus der Smart-Metering-Public-Key-Infrastruktur („SM-PKI“) gemäß [BSI TR-03109-4].

Das SMGW akzeptiert keine TLS-Verbindungen, die von Teilnehmern aus dem WAN initiiert werden. Das SMGW kann jedoch über den Wake-Up-Dienst (s. Kap. 4.9) veranlasst werden, eine TLS-Verbindung zum GWA aufzubauen. Zu einem Zeitpunkt können jeweils mehrere TLS-Verbindungen zwischen SMGW und GWA gleichzeitig existieren, wie nachfolgend aufgeführt. Dabei nutzen die Verbindungen stets das gleiche Zertifikat und den zugehörigen privaten

Schlüssel auf ihrer jeweiligen Seite. Im Fall, dass die Anzahl gleichzeitiger WAN-Verbindungen unbeschränkt ist, wird diese Anzahl faktisch durch den Ressourcen-Monitor begrenzt. Bzgl. der einzelnen WAN-Kommunikationsszenarien vergleiche übernächste Seite:

- MANAGEMENT (WKS1): Verbindungsanzahl beschränkt auf 1
- ADMIN-SERVICE (WKS2): Verbindungsanzahl unbeschränkt
- INFO-REPORT (WKS3): Verbindungsanzahl unbeschränkt
- NTP-TLS (WKS5): Verbindungsanzahl beschränkt auf 1
- „Kommunikation EMT mit CLS über das SMGW“ (WAF6): Verbindungsanzahl unbeschränkt

Die TLS-Implementierung des SMGW erfolgt nach den Vorgaben aus [BSI TR-03109-3] gemäß [RFC 5246]. Es werden die TLS-Ciphersuiten gemäß Tabelle 2 und Tabelle 3 verwendet. Das SMGW setzt dabei die folgenden kryptographischen Funktionalitäten um:

- Generierung von SHA-256- bzw. SHA-384-Hashwerten gemäß [FIPS 180-4] für die Signaturerzeugung und -prüfung
- Symmetrische Ver- und Entschlüsselung mit AES-128 resp. AES-256 gemäß [FIPS 197] nach Aufbau des TLS-Kanals

Ein Benutzer dieser Schnittstelle kann ohne erfolgreiche Authentisierung (außer der Authentisierungsfunktionalität des SMGW selbst) keinerlei sicherheitsfunktionale Aktionen des SMGW ausführen. Auf Basis des an der Schnittstelle in Form des TLS-Zertifikats präsentierten Authentisierungsmaterials weist das SMGW dem Benutzer seine Rolle zu (Sicherheitsattribut „*role membership*“). Für die Zuweisung der Rolle „GWA“ muss ferner die im SMGW konfigurierte IP-Adresse des GWA mit der für den Authentisierungsvorgang benutzten IP-Adresse übereinstimmen.

Im Fall einer nicht erfolgreichen Authentisierung antwortet das SMGW mit einer sog. *Alert Message* des Niveaus „*fatal*“ gemäß [RFC 5246, Kap. 7.2], woraufhin das SMGW den Verbindungsaufbau abbricht bzw. die Verbindung wieder abbaut. Das SMGW schreibt dabei die Meldung "*Verbindungsaufbau zu [kommunikationsprofil-id] abgebrochen. Grund: [grund]*" ins System-Log. Eine neuerliche Verbindungsaufnahme sowie eine länger als 48 Stunden offen gehaltene TLS-Verbindung bedingen eine neue vollständige Authentisierung der Verbindung. In jedem Falle (d. h. sowohl einer nicht erfolgreichen, wie auch erfolgreichen Verbindungsaufnahme und Kommunikation) werden nach Abbau der Verbindung alle mit dieser Verbindung assoziierten Ressourcen freigegeben.

Das für den TLS-Kanal notwendige Zertifikat des WAN-Teilnehmers wurde zuvor durch den GWA in das SMGW eingebracht. Im Fehlerfall wird das empfangene Paket verworfen und ein Eintrag ins System-Log erstellt. Das Einbringen der Benutzerzertifikate erfolgt, indem der GWA diese Daten innerhalb des XML-Elements *cert_tls* der COSEM-IC *user_setup* für den betreffenden Benutzer via WKS1 überträgt.

Konfiguriert der GWA sein eigenes kryptografisches Schlüsselmaterial, muss der GWA vorher unter Benutzung des UC_PN_04_04 gemäß [BSI TR-03109-2-UC] und [BSI TR-03109-2] die entsprechenden Prüfschlüssel in das Sicherheitsmodul importieren, bevor das Update des Zertifikatsmaterials via COSEM-IC *user_setup* erfolgen kann.

In den höheren Schichten des OSI-Modells setzt das SMGW die folgenden Protokolle bzw. Mechanismen ein:

- Zurverfügungstellung eines RESTful-COSEM-Webservices seitens des SMGW (WKS1)
- Zugriff auf RESTful-COSEM-Webservices beim GWA (WKS2) oder EMT (WKS3)
- Zugriff auf einen vom GWA bereitgestellten NTP-Service (WKS5)
- Weiterleitung von Datenpaketen zwischen einem CLS und einem EMT im „Proxy-Modus“ (WKS6)
- Empfang eines sog. Wake-Up-Pakets seitens des SMGW (WKS7)
- Empfang eines proprietären Firmware-Updates seitens des SMGW

Das SMGW kennt demnach also die folgenden WAN-Kommunikationsszenarien (WKS) gemäß [BSI TR-03109-1]:

1. „MANAGEMENT“: Das WAN-Kommunikationsszenario WKS1 dient dem Zugriff des GWA auf den RESTful-COSEM-Webservice des SMGW, der vom SMGW unter der Adresse */smgw/cosem…* angeboten wird. Die vollständige Beschreibung der Ausprägung dieses Webservices erfolgt in Kapitel 4.4.
2. „ADMIN-SERVICE“: Das WAN-Kommunikationsszenario WKS2 dient dem Zugriff des SMGW auf den RESTful-COSEM-Webservice des GWA, der unter der Adresse */gwa/…<Service-PoC>* angeboten wird. Die vollständige Beschreibung der Ausprägung dieses Webservices erfolgt in Kapitel 4.5.
3. „INFO-REPORT“: Das WAN-Kommunikationsszenario WKS3 dient dem Zugriff des SMGW auf den RESTful-COSEM-Webservice eines (vom GWA konfigurierten) EMT zur Auslieferung von tarifierten Messwerten oder Netzzustandsdaten unter der Adresse */<emt>/…<Report-PoC>*. Die vollständige Beschreibung der Ausprägung dieses Webservices erfolgt in Kapitel 4.6.
4. „NTP-TLS“: Das WAN-Kommunikationsszenario WKS5 dient der Zeitsynchronisation des SMGW durch einen vom GWA bereitgestellten NTP-Service. Die vollständige Beschreibung der Ausprägung dieses Kommunikationsszenarios erfolgt in Kapitel 4.8. Das SMGW unterstützt **nicht** das WAN-Kommunikationsszenario WKS4 „NTP-HTTPS“.

Ferner kennt das SMGW die Anwendungsfälle

5. „Kommunikation EMT mit CLS über das SMGW“ (WAF6) gemäß Kapitel 4.7 und den
6. „Wake-Up-Service“ (WAF7) gemäß Kapitel 4.9.

Für die Kommunikation des SMGW über das WAN implementiert das SMGW demnach also den folgenden, in grün gekennzeichneten, obligatorischen Protokollstapel:

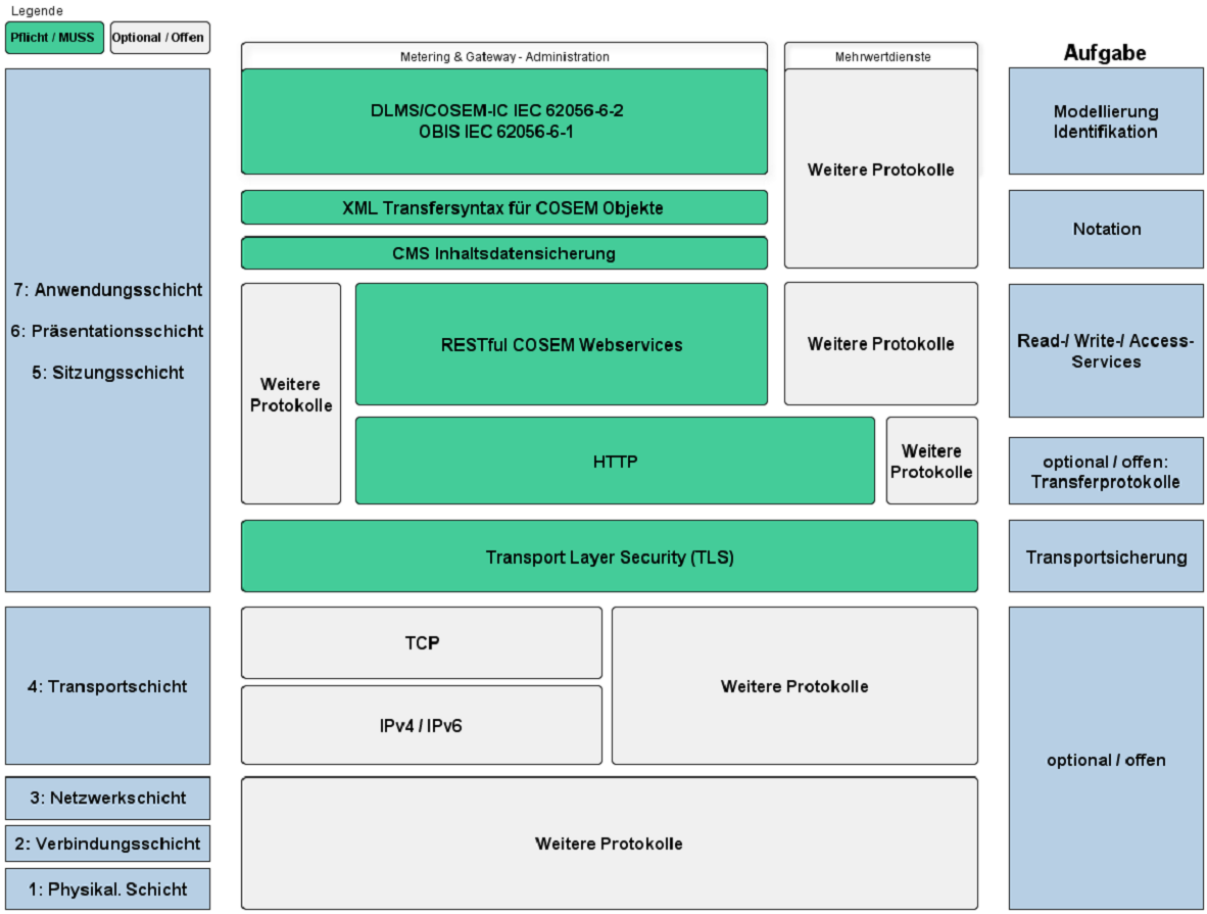


Abbildung 5: Protokollstapel für die WAN-Kommunikation gemäß [BSI TR-03109-1]

4.2 TLS und CMS-Inhaltsdatensicherung

Zur Sicherung der Inhaltsdaten im WAN verschlüsselt das SMGW gemäß [SMGW_PP] die COSEM-Objekte bzw. aggregierten Objekt-Container für den Endempfänger und signiert diese.

Zusätzlich werden die folgenden Vorgaben für die HTTP-Kommunikation erfüllt:

- Für die Kennzeichnung der COSEM-Daten mit XML-Transfersyntax und CMS-Inhaltsdatensicherung wird der Content-Type „*application/vnd.de-dke-k461-cosem+xml;encap=cms-tr03109*“ verwendet.
- Für die Kennzeichnung der CMS-Inhaltsdatenverschlüsselung ohne vorherige Kompression der XML-Daten ist KEIN „*Content-Encoding*“-Header-Feld vorhanden.
- Für die Kennzeichnung der CMS-Inhaltsdatenverschlüsselung mit vorheriger Kompression der XML-Daten wird das Content-Encoding „*deflate*“ verwendet.
- Das SMGW (wie auch seine Webservice-Gegenstelle) kann sowohl komprimierte als auch unkomprimierte CMS-Daten verarbeiten. Der ASN.1-ContentType des verschlüsselten Inhalts hat den ASN.1-Object-Identifier-Wert „*id-data*“ oder „*id-ct-compressedData*“.
- Requests/Responses ohne HTTP-Body werden NICHT mittels Inhaltsdatensicherung abgesichert, d.h. Status-Codes über den HTTP-Header werden durch TLS gesichert, aber nicht zusätzlich CMS-verpackt.

Für die zu übermittelnden Inhaltsdaten (Netzstatusdaten, Abrechnungsdaten, Administrationsdaten) führt das SMGW eine symmetrische Inhaltsdatenverschlüsselung gemäß CMS [RFC 5652] durch und sichert die verschlüsselten Inhaltsdaten mit einem MAC. Die Verschlüsselung und Integritätssicherung erfolgt hierbei nach den Vorgaben aus [BSI TR-03116-3, Kap. 8] nach dem AES-Standard mit symmetrischen 128-Bit-Schlüsseln im CBC-Modus gemäß [FIPS 197] und einer AES-CMAC-Sicherung gemäß [RFC 4493]. Eine Verschlüsselung und Integritätssicherung gemäß AES-CBS-CMAC wird vom SMGW **nicht** unterstützt.

Erfolgreich übermittelte Inhaltsdaten werden vom SMGW samt der unter Verwendung des kryptografischen Schlüsselmaterials des Endempfängers erfolgenden Inhaltsdatensignatur im Log des zugehörigen Letztverbrauchers protokolliert, wodurch eine per Signaturverifikation nachprüfbare Zuordnung der versendeten Daten zum Empfänger entsteht. Fehler bei der Verschlüsselung oder Übermittlung der Inhaltsdaten protokolliert das SMGW im System-Log.

Für eine solche TLS-Verbindung werden gemäß den Vorgaben aus [BSI TR-03109-3] resp. [BSI TR-03116-3] die folgenden Ciphersuiten gemäß [RFC 5289] und EC-Kurven gemäß [RFC 5639] bzw. [RFC 5114] eingesetzt:

Tabelle 2: Vom SMGW unterstützte TLS-Ciphersuiten

Ciphersuite	Verwendung von	Verwendung bis
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	2015	2028+
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	2015	2028+
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	2015	2028+
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	2015	2028+

Tabelle 3: Vom SMGW unterstützte EC-Kurven

EC-Kurven	Verwendung von	Verwendung bis
NIST P-256 gemäß [RFC 5114]	2015	2028+
NIST P-384 gemäß [RFC 5114]	2015	2028+
BrainpoolP256r1 gemäß [RFC 5639]	2015	2028+
BrainpoolP384r1 gemäß [RFC 5639]	2015	2028+
BrainpoolP512r1 gemäß [RFC 5639]	2015	2028+

Kommt zwischen den Kommunikationspartnern keine Verbindung nach TLS Version 1.2 und/oder unter Verwendung der notwendigen Ciphersuiten zustande, wird keinerlei anders geartete Verbindung aufgebaut, und das SMGW erstellt einen entsprechenden Eintrag im System-Log. In jedem Falle (d. h. sowohl einer nicht erfolgreichen, wie auch erfolgreichen Verbindungsaufnahme und Kommunikation) werden nach Abbau der Verbindung alle mit dieser Verbindung assoziierten Ressourcen freigegeben.

Das SMGW verwendet dabei nach den Vorgaben aus [BSI TR-03109-3, Kap. 4] die folgenden kryptographischen Algorithmen und Schlüssellängen (vgl. Tabelle 1 und Tabelle 2):

- Hash-Generierung nach SHA-256 und SHA-384 gemäß [FIPS 180-4]
- Erzeugung von Signaturen mit ECDSA mit den elliptischen Kurven
    - NIST P-256 und NIST P-384 gemäß [RFC 5114] und
    - BrainpoolP256r1, BrainpoolP384r1 und BrainpoolP512r1 gemäß [RFC 5639].

Die Abbildung 30 aus [BSI TR-03109-1] zeigt exemplarisch und lediglich informativ die Interaktion zwischen SMGW und seinem Sicherheitsmodul im Bereich der Inhaltsdatenverschlüsselung, Integritätssicherung und Signierung. Umgesetzt sind Maßnahmen zur Erfüllung der Anforderungen gemäß [BSI TR-03109-2].

4.3 RESTful-Cosem-Webservice

Der Zugriff auf Ressourcen findet in einem RESTful-API-Style mittels HTTP/1.1 gemäß [RFC 2616] statt. Der *Point-Of-Contact* <PoC> bildet dabei die Wurzel des Ressourcen-Baumes, d.h. der *Point-Of-Contact* stellt den URI-Path-Prefix dieser API dar. Der <PoC> wird dabei dem HTTP-Client vom HTTP-Server bekannt gemacht.

Die Modellierung der Datenstrukturen, die vom SMGW im WAN über RESTful-COSEM-Webservices übertragen werden, geschieht mit Hilfe der sog. „COSEM-Interface-Klassen“ gemäß [IEC-62056-6-2], ergänzt um weitere Festlegungen gemäß [DKE-AK 142]. „COSEM-Objekte“ werden dabei durch Instanziierung aus der ihnen jeweils zugeordneten COSEM-Interface-Klasse gebildet und eindeutig unter Verwendung der sog. „OBIS-Codes“ gemäß [IEC-62056-6-1], [EN 13757-1] und [DKE-AK 142] benannt. Existiert kein derlei standardisierter OBIS-Code einer COSEM-Interface-Klasse, spezifiziert der SMGW-Hersteller zusätzliche OBIS-Codes für die SMGW-proprietären COSEM-Interface-Klassen.

Zur Adressierung des SMGW und der „Logical Devices“ (d.h. der virtuellen Zähler) innerhalb des SMGW wird eine kanonische Geräte-ID verwendet. Jede Instanz des SMGW und jedes dort vorhandene „Logical Device“ erhält daher eine eindeutige herstellerübergreifende Identifikationsnummer nach [DIN 43863-5]. Diese Identifikationsnummer wird gemäß [BSI TR-03109-1] in folgender Form kanonisiert und dient dann als „Hostname“ bzw. als „Logical Device Name“ innerhalb einer URI:

- Großbuchstaben werden zu Kleinbuchstaben.
- Das Suffix „*.sm*“ wird angehängt.

Die resultierende Zeichenfolge enthält nur die Zeichen „*a-z*“, „*0-9*“ und „.“ (Punkt) und hat eine Länge von 1 - 63 Zeichen. Da die Kennung des SMGW aus der Sparte Kommunikation („*E*“) stammt, beginnt die kanonische Geräte-ID einer SMGW-Instanz stets mit „*e*“, wobei für jede SMGW-Instanz vom SMGW-Hersteller eine 10-stellige eindeutige Nummer <*nnnnnnnnn*>

vergeben wird, deren erste 2 Stellen den Fabrikationsblock und die weiteren 8 Stellen die Fabrikationsnummer des SMGW-Exemplars darstellen, wobei diese Nummer rechtsbündig mit führenden Nullen gebildet wird.

Um jede SMGW-Instanz eindeutig identifizieren zu können, werden die letzten (niedrigsten) drei Bytes der eindeutigen MAC-Adresse der HAN-Schnittstelle einer SMGW-Instanz verwendet, um die Fabrikationsnummer der Geräte-ID dieser SMGW-Instanz abzuleiten. Diese drei Bytes der hexadezimalen MAC-Adresse werden dabei gemäß [DIN 43863-5] dezimal dargestellt. So ergäbe sich z.B. aus einer MAC-Adresse „00:25:18:**ab:cd:ef**“ mit der Umrechnung abcdef $_{HEX}$ == 11259375 $_{DEZ}$ die Geräte-ID „eppc02**11259375**“.

Für Container-Objekte werden gemäß [BSI TR-03109-1] OBIS-IDs (z.B. aus einem Länder- oder Konsortium-zugewiesenen Bereich) verwendet. Damit sind die Container innerhalb eines *Logical Devices* eindeutig adressierbar.

Der Zugriff auf den RESTful-Webservice erfolgt gemäß folgender generischen Rechte-Matrix. Für den jeweils angegebenen WAN-Anwendungsfall WAF sind dem Aufrufer (Caller) die folgenden Rechte zum Anlegen (CREATE „C“), zum Lesen (READ „R“), zum Updaten (UPDATE „U“) und zum Löschen (DELETE „D“) zugeordnet.

Tabelle 4: Generische WAF-Rechtematrix

Anwendungsfall	Szenario	Caller / Rolle	Responder	Rechte
WAF1	MANAGEMENT	GWA	SMGW	CRUD
WAF2	ADMIN-SERVICE	SMGW	GWA	RU
WAF3	ADMIN-SERVICE	SMGW	GWA	CU
WAF4	ADMIN-SERVICE	SMGW	GWA	CU
WAF5	INFO-REPORT	SMGW	EMT	CU

Gemäß [DKE-AK 142] lässt das SMGW lediglich eine Adressierung von Objekten über das in COSEM verwendete sog. *Long-Name-Addressing* zu. Damit ergeben sich zum einen die Basis-Dienste *Get*, *Set* und *Action* sowie zusätzliche Dienste, die zum Anlegen und Löschen von Ressourcen zur Abbildung von dynamischen Datenmodellen definiert werden. In Abbildung 6 sind sowohl die Zugriffsdienste auf COSEM-Objekte als auch die für Webservice notwendigen Zugriffsdienste tabellarisch gelistet.

Tabelle 5: Allgmeine Zugriffs-Dienste des SMGW [10]

Zugriffs-Dienst	HTTP-Verb	Request-URI enthält	Request-Body	Response-Body
Logical-Device-Ebene				
Abfrage der *ldevids* aller vorhanden *Logical Devices*	GET	<PoC>/cosem/ldevs/	Leer	Liste der *ldevids*
Anlegen eines *Logical Devices*	POST	<PoC>/cosem/ldevs/	Logical-Device-Element	Leer oder Fehler
Löschen eines *Logical Devices* der ID *ldevid*	DELETE	<PoC>/cosem/ldevs/*{ldevid}*/	Leer	Leer oder Fehler
Objekt-Ebene				
Abfrage aller Objekte, die unter dem genannten *Logical Devices* angeordnet sind	GET	<PoC>/cosem/ldevs/*{ldevid}*/objects	Leer	Liste der OBIS-Codes, der Class IDs und der Class Version der Objekte
Abfrage des Objekts *logicalname* eines *Logical Devices* der ID *ldevid*	GET	<PoC>/cosem/ldevs/*{ldevid}*/objects/*{logicalname}*/	Leer	Objekt-Element oder Fehler
Update eines Objekts	PUT	<PoC>/cosem/ldevs/*{ldevid}*/objects/*{logicalname}*/	Objekt-Element	Leer oder Fehler

10 N.B.: Die in den Request-URIs enthaltenen (in kursiv gesetzten) Angaben *ldevid*, *classid-logicalname* resp. *attrid* sind mit den konkreten Werten der betreffenden COSEM-Objekt-Instanzen des sog. Logischen Geräts, der COSEM-Klasse bzw. des COSEM-Attributs zu ersetzen.

Ebenso muss die Angabe <PoC> mit der konkreten Angabe des Dienste-Zugriffspunkts gefüllt werden.

Zugriffs-Dienst	HTTP-Verb	Request-URI enthält	Request-Body	Response-Body
Anlegen oder Updaten eines Objekts	POST	<PoC>/cosem/ldevs/*{ldevid}*/objects/	Objekt-Element	Leer oder Fehler
Delete Object	DELETE	<PoC>/cosem/ldevs/*{ldevid}*/objects/*{logicalname}*/	Leer	Leer oder Fehler
Attribut-Ebene				
Abfrage der Attributliste des Objekts *logicalname* eines *Logical Devices* der ID *ldevid*	GET	<PoC>/cosem/ldevs/*{ldevid}*/objects/*{logicalname}* */?q.fromidx=0&q.count=1*	Leer	Attributliste oder Fehler
Methoden-Ebene				
Ausführen einer Methode	POST	<PoC>/cosem/ldevs/*{ldevid}*/objects/*{logicalname}/methods/{method-name}*	Methoden-Element	Methoden-Element oder Fehler

Die HTTP-Verben GET, PUT, POST und DELETE operieren genau auf der adressierten Ressource (und evtl. vorhandenen untergeordneten Ressourcen). Bei Verwendung der HTTP-Verben GET, PUT und DELETE muss die Ressource existieren. Mit dem HTTP-Verb POST wird die Ressource angelegt, falls sie nicht existiert bzw. überschrieben, falls sie bereits existiert.

Die Abfrage einer Liste („*Read-With-List*“, z. B. die Abfrage einer Attributliste) wird über GET mit dem Query-Parameter *q.fromidx und q.count* abgebildet. Die Startposition und Anzahl an Listenelementen wird über die Abfrageparameter *q.fromidx* und *q.count* bestimmt. Dieser Mechanismus wird vor allem bei der Abfrage der Logbücher des SMGWs verwendet.

Das SMGW bildet die zu übertragenden COSEM-Object/Attribut-Ressource-Strukturen gemäß den XML-Schema-Dateien aus der vom SMGW-Hersteller zur Verfügung gestellten Archivdatei *SMGW-XSD_v.1.0.zip* in XML ab. Die Archivdatei wird Ihnen zusammen mit dem vorliegenden Handbuch verschlüsselt zur Verfügung gestellt. Bei Bedarf kann die Prüfsumme des Archivs telefonisch mit dem SMGW-Hersteller abgeglichen werden.

Die folgende Darstellung veranschaulicht die Objektstruktur im SMGW.

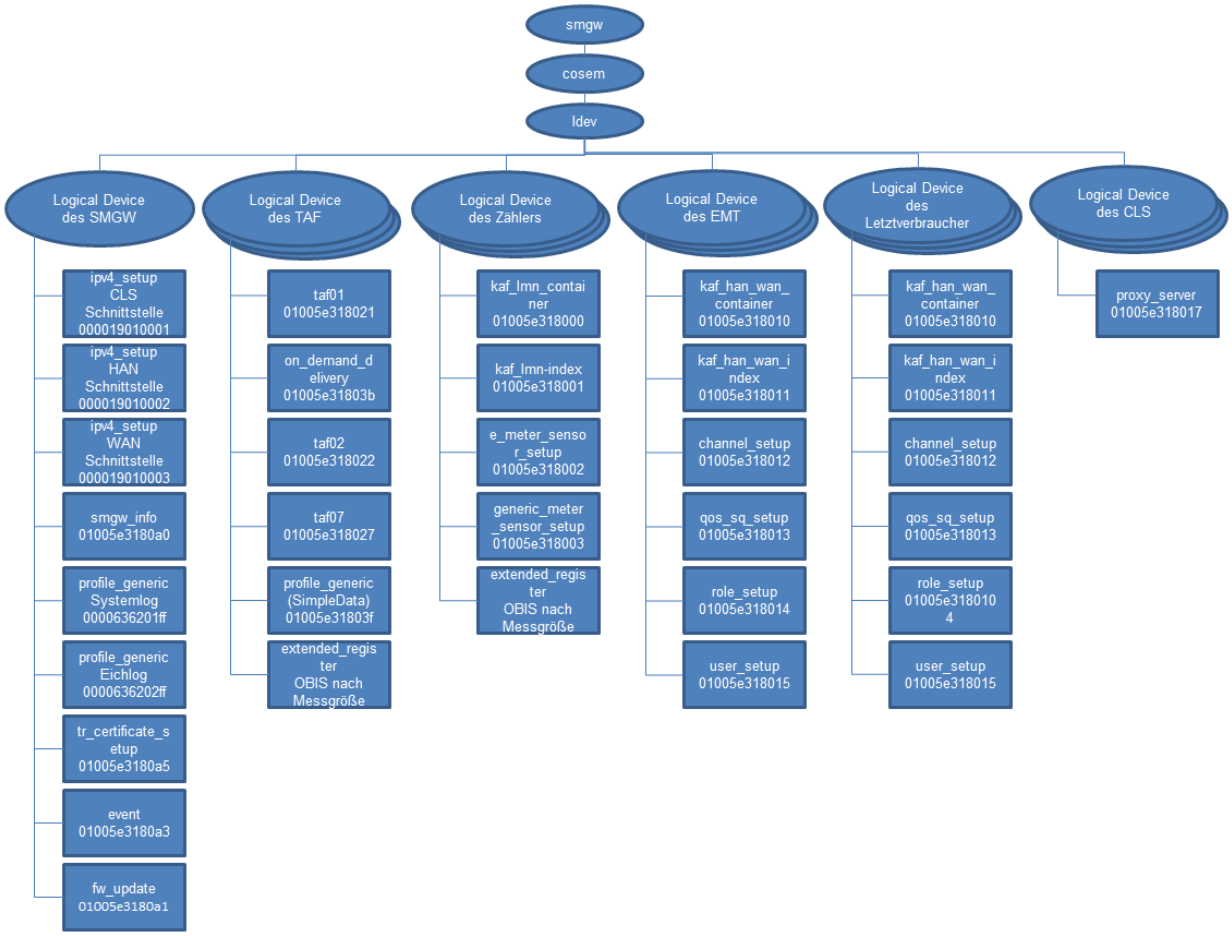


Abbildung 6: Objektmodell des SMGW

Der *Logical Device Name* (im Folgenden auch „logical_name“) jedes Objekts setzt sich aus der Logical Device ID (im Folgenden auch „ldevid“) und aus der OBIS-Kennzahl des Objekts zusammen. Jedes Objekt ist über den *Logical Device Name* eindeutig je Smart Meter Gateway adressierbar.

Der *Logical Device Name* des Objekts, das zur Konfiguration der IP-Adresse der HAN-Schnittstelle des SMGW „eppc0011259375“ verwendet wird, lautet daher „000019010002.eppc0011259375.sm“.

Um die IP-Adresse des Geräts zu modifizieren, muss gemäß Abbildung 6 ein Objekt der Klasse mittels eines PUT-Requests an die Request-URI „/smgw/cosem/ldevs/eppc0011259375.sm/objects/000019010002“ gesendet werden.

Die sog. COSEM-Attribute werden in XML in Form von XML-Elementen abgebildet. Jedes dieser XML-Elemente trägt dabei verpflichtend ein XML-Attribut namens „*id*“ vom XML-Datentyp „*byte*“. Alle XML-Attribute eines XML-Elements (= COSEM-Attribut) werden dabei fortlaufend durchnummeriert, beginnend mit „1“ für das XML-Attribut „*id*“.

Das Smart Meter Gateway beantwortet die HTTP-Requests des GWA auf die Zugriffsdienste im Erfolgsfall wie in folgender Tabelle dargestellt.

Tabelle 6: HTTP-Statuscodes des SMGW auf Requests des GWA

Zugriffs-Dienst	HTTP-Verb	HTTP-Response bei erfolgreicher Ausführung
Anlegen eines *Logical Devices oder Objekts*	POST	201 Created
Löschen eines *Logical Devices*	DELETE	200 OK
Abfrage eines Objekts	GET	200 OK
Update eines Logical Device oder eines Objekts	PUT/POST	200 OK
Ausführen einer Methode	POST	200 OK

Im Fehlerfall werden folgende http-Statuscodes verwendet:

Tabelle 7: HTTP-Statuscode im Fehlerfall

Fehlerfall	HTTP Response
Request-URI nicht bekannt	404 Not Found
Anfrage-Nachricht war fehlerhaft. Mögliche Ursachen sind dafür: • Syntax der Inhaltsdaten war nicht entsprechend der Vorgaben. • Werte der Inhaltsdaten entsprechen nicht den Datentypen. • Beim Einspielen von Zertifikaten konnte die Zertifikatskette nicht überprüft werden. • Die Übergabeparameter einer Methode entsprechen nicht den erwarteten Daten.	400 Bad Request
HTTP-Header passen nicht zu den im HTTP-Body übermittelten Daten.	406 Not Acceptable
Interner Fehler im SMGW	500 Internal Server Error

4.4 WKS1: „Management“-Webservice-Schnittstelle

Der vom SMGW betriebene RESTful-Webservice dient dem Empfang von Administrationsdaten durch den GWA. Dieser Webservice unterstützt ausschließlich die in diesem Kapitel genannten Administrationsbefehle. Diese Administrationsbefehle werden dabei durch die vom SMGW-Hersteller gelieferten XML-Schema-Dateien aus der Archivdatei *SMGW-XSD_v.1.0.zip* festgelegt. Andere Webservice-Aufrufe oder nicht XML-Schema-konforme Webservice-Aufrufe werden vom SMGW nicht unterstützt. Die WAN-Schnittstelle des SMGW ist die einzige Schnittstelle, über die der GWA auf das SMGW zugreifen und Administrationsbefehle zur Abfrage, Änderung oder Löschung von SMGW-Konfigurationsparametern ausführen kann.

4.4.1 WAF1: Administration und Konfiguration

Auf Basis des an der Schnittstelle in Form des TLS-Zertifikats präsentierten, vom GWA konfigurierten Authentisierungsmaterials weist das SMGW dem Benutzer seine Rolle zu (Sicherheitsattribut „*role membership*“). Für die Zuweisung der Rolle „GWA“ muss ferner die im SMGW konfigurierte IP-Adresse des GWA mit der für den Authentisierungsvorgang benutzten IP-Adresse übereinstimmen.

An der WAN-Schnittstelle wird es erst dem erfolgreich authentisierten GWA (Sicherheitsattribut „*status of identity*“ = „*authenticated*“) erlaubt, die Daten einzusehen, die im SMGW gemäß dem Rollenmodell genau diesem Benutzer mit dem gegebenen Authentisierungszustand an dieser Schnittstelle zugeordnet sind, in diesem Fall also, auf das System-Log sowie das eichtechnische Log lesend zuzugreifen. Andere Zugriffe, insbesondere ein Zugriff auf die Verbrauchs- oder Log-Daten des Letztverbrauchers, sind für den GWA an der WAN-Schnittstelle nicht möglich.

Die Befehle lassen sich in die folgenden thematischen Gruppen kategorisieren:

- Schnittstellenkonfiguration des SMGW zerfallend in die zwei Teile
  - Konfiguration der WAN- oder HAN-Schnittstelle gemäß Kap. 4.4.2.1
  - Konfiguration der LMN-Schnittstelle gemäß Kap. 4.4.2.2
- Profilverwaltung und -konfiguration des SMGW
  - Verwaltung der Zählerprofile gemäß Kap. 4.4.3.1
  - Verwaltung der WAN- oder HAN-Profile gemäß Kap. 4.4.3.2
  - Verwaltung der TAF-Profile gemäß Kap. 4.4.3.3
  - Verwaltung der CLS-Profile gemäß Kap. 4.4.3.4
- GWA-Zugriff auf das SMGW-Sicherheitsmodul gemäß Kap. 4.4.4
- Verwaltung der Logdaten gemäß Kap. 4.4.5

4.4.2 Schnittstellenkonfiguration des SMGW

Zur Schnittstellenverwaltung werden im SMGW folgende Klassen verwendet:

Tabelle 8: Überblick über die COSEM-ICs zur Schnittstellenkonfiguration des SMGW

Name der COSEM-IC	Aufgabe
ipv4_setup	Konfiguration der IPv4-Adresse der Netzwerkgeräte

4.4.2.1 Konfiguration der WAN-, HAN- oder CLS-Schnittstelle

IC *ipv4_setup*

Instanzen der Klasse *ipv4_setup* werden vom GWA zur Konfiguration der IPv4-Parameter Netzwerkschnittstellen des SMGW benutzt. Diese Klasse besitzt keine Methoden. Die Parameter sind wie folgt:

Tabelle 9: Attribute der IC *ipv4_setup*

Name		XML-Datentyp	Beschreibung
logical_name		string	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
dl_reference	*logical_name*	string	Referenz auf ein Setup-Objekt des Data-Link-Layer anhand ihres „logical_name" Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.
	class_id	unsignedShort	
	class_version	unsignedByte	
ip_address		string	Enthält den Wert der IPv4-Adresse des Netzwerkgeräts. Diese kann statisch sein oder dynamisch per DHCP vergeben werden. Wird die IP-Adresse dynamisch per DHCP vergeben (use_dhcp_flag ‚true'), ist der Wert ‚0.0.0.0'. Das Attribut muss angegeben werden. Der Wert muss einer gültigen IPv4 Adresse entsprechen (vier Blöcke, durch ‚.' getrennt, je 8 Bit, d.h. Wertebereich von 0 -255).
multicast_ip_addresses	*multicast_ip_address*	string	Unbegrenzte Liste von IP-Multi-Cast-Adressen für das SMGW. Eine solche IP-Adresse muss im „Multi-Cast-Adress-Bereich" (224.0.0.0 – 239.255.255.255) liegen. Das Attribut kann angegeben werden. Wenn es angegeben wird, muss der Wert einer gültigen IPv4 Adresse entsprechen (vier Blöcke, durch ‚.' getrennt, je 8 Bit, d.h. Wertebereich von 0 -255). Es wird im SMGW nicht ausgewertet.

ip_options	*ip_option*	*Type*	unsignedByte	Angaben zum „*options*“-Feld des IP-Pakets gemäß [RFC 791], wobei nur die dort aufgeführten ‚type‘-Werte verwendet werden dürfen, wie z.B. ‚68‘ für ‚Internet Timestamp‘. Die Kodierung erfolgt im TLV-Format (*Type-Length-Value*). Das Attribut kann angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.
		Length	unsignedByte	
		Data	hexBinary	
subnet_mask			string	Angabe der Subnetz-Maske Das Attribut muss angegeben werden. Der Wert muss einer gültigen IPv4 Adresse entsprechen (vier Blöcke, durch ‚.‘ getrennt, je 8 Bit, d.h. Wertebereich von 0 -255).
gateway_ip_address			string	Optionale Angabe der IP-Adresse des SMGW Das Attribut kann angegeben werden. Der Wert muss einer gültigen IPv4 Adresse entsprechen (vier Blöcke, durch ‚.‘ getrennt, je 8 Bit, d.h. Wertebereich von 0 -255).
use_dhcp_flag			boolean	Ist dieses Flag auf ‚true‘ gesetzt, benutzt das SMGW DHCP zur IP-Adress-Vergabe. Ist das Flag auf ‚false‘ gesetzt, müssen IP-Adresse, Subnetz-Maske und Gateway-Adresse statisch gesetzt sein. Das Attribut muss angegeben werden. Der Wert muss entweder auf ‚false‘ oder ‚true‘ gesetzt werden.
primary_dns_address			string	Angabe der IP-Adresse des 1. DNS-Servers Das Attribut kann angegeben werden. Wenn es angegeben wird, muss der Wert einer gültigen IPv4 Adresse entsprechen (vier Blöcke, durch ‚.‘ getrennt, je 8 Bit, d.h. Wertebereich von 0 -255). Es wird im SMGW nicht ausgewertet.

secondary_dns_address		string	Angabe der IP-Adresse des 2. DNS-Servers Das Attribut kann angegeben werden. Wenn es angegeben wird, muss der Wert einer gültigen IPv4 Adresse entsprechen (vier Blöcke, durch ‚.' getrennt, je 8 Bit, d.h. Wertebereich von 0 -255). Es wird im SMGW nicht ausgewertet.
register_tx_bytes	*logical_name*	string	Optional nutzbarer Verweis auf ein Objekt vom Typ IC Register (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version). Der von diesem Register kumulierte Zählerstand dient der Zählung der in Sende-Richtung (WAN an HAN) transportierten Bytes. Das Attribut darf in der vorliegenden Variante des SMGW nicht angegeben werden.
	class_id	unsignedShort	
	class_version	unsignedByte	
register_rx_bytes	*logical_name*	string	Optional nutzbarer Verweis auf ein Objekt vom Typ IC Register (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version). Der von diesem Register kumulierte Zählerstand dient der Zählung der in Empfangs-Richtung (HAN an WAN) transportierten Bytes. Das Attribut darf in der vorliegenden Variante des SMGW nicht angegeben werden.
	class_id	unsignedShort	
	class_version	unsignedByte	

Im SMGW sind drei Instanzen der Klasse *ipv4_setup* vorhanden. Die drei Instanzen sind über ihre OBIS Kennzahl je einer Schnittstelle zugeordnet.

Die drei OBIS-Kennzahlen lauten wie folgt:

- CLS-Schnittstelle: *000019010001*
- HAN-Schnittstelle: *000019010002*
- WAN-Schnittstelle *000019010003*

Um die Konfiguration der IPv4-Schnittstelle zu ändern, kann der Gateway Administrator mittels des HTTP-Verbs PUT ein Update auf jede Instanz der Klasse durchführen.

Die Instanzen der Klasse sind der Logical Device ID *ldevid* des SMGW zugeordnet.

Tabelle 10: Aktualisierung der IPv4-Konfiguration der HAN-Schnittstelle

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für Update der HAN-Schnittstelle	Request-Body	HTTP Response Statuscode	Response-Body
Update der bestehenden Klasseninstanzen	PUT	<PoC>/cosem/ldevs/eppc0110123456.sm/objects/000019010002	ipv4_setup	200	Leer

Tabelle 11: Aktualisierung der IPv4-Konfiguration der CLS-Schnittstelle

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für Update der CLS-Schnittstelle	Request-Body	HTTP Response Statuscode	Response-Body
Update der bestehenden Klasseninstanzen	PUT	<PoC>/cosem/ldevs/eppc0110123456.sm/objects/000019010001	ipv4_setup	200	Leer

Tabelle 12: Aktualisierung der IPv4-Konfiguration der WAN-Schnittstelle

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für Update der WAN-Schnittstelle	Request-Body	HTTP Response Statuscode	Response-Body
Update der bestehenden Klasseninstanzen	PUT	<PoC>/cosem/ldevs/eppc0110123456.sm/objects/000019010003	ipv4_setup	200	Leer

Bereits in der Produktion wird festgelegt, ob das SMGW das IPv4 oder das IPv6 Protokoll an der WAN-Schnittstelle unterstützt. Wenn festgelegt wurde, dass das SMGW das IPv6 Protokoll an der WAN-Schnittstelle unterstützt wird eine Instanz der Klasse *ipv4_setup* zur Modifikation der bestehenden Netzwerkkonfiguration (vgl. Tabelle 8) vom SMGW abgelehnt und mit einem http Response Statuscode 400 beantwortet.
Wenn die Verwendung von IPv6 in der Produktion festgelegt wurde, verwendet das SMGW zum Adressbezug immer „IPv6 Stateless Address Autoconfiguration“ gemäß [RFC 4862]. Diese Einstellung ist nach der Produktion nicht mehr änderbar.
Die HAN- und CLS-Schnittstelle des SMGWs unterstützen ausschließlich IPv4. Die Konfiguration dieser Schnittstellen durch eine Verwendung der Klasse *ipv4_setup* ist daher uneingeschränkt möglich.

4.4.2.2 Konfiguration der LMN-Schnittstelle

Die LMN-Schnittstelle ist von außen durch den Gateway Administrator nicht konfigurierbar. Die entsprechenden Parameter werden durch die Applikation des SMGW korrekt gesetzt. Dabei kommen folgende Parameter zum Einsatz:

Wireless-M-Bus-Schnittstelle Wireless Mode: T1

RS485-Schnittstelle Baudrate: 921,6 kBaud

Datenformat/Byte: 1 Start-Bit, 8 Datenbits, kein Paritätsbit, 1 Stop-Bit

CRC: Berechnung gemäß IEC 62056-46

4.4.3 Profilverwaltung und -konfiguration des SMGW

Zur Profilverwaltung und -konfiguration des SMGW können von einem erfolgreich an der WAN-Schnittstelle authentisierten GWA folgende Klassen verwendet:

Tabelle 13: Überblick über die COSEM-ICs zur Profilverwaltung und -konfiguration des SMGW

Name der COSEM-IC	Aufgabe
channel_setup	Klasse zur Konfiguration eines Kommunikationskanals
on_demand_delivery	Wird dazu verwendet, die Spontanauslesung gemäß Tarifanwendungsfalls TAF6 zu initiieren.
e_meter_sensor_setup	Konfiguration eines Strom-Zählers
kaf_han_wan_container	Containerklasse zum Einspielen eines HAN- oder WAN-Profils
kaf_han_wan_index	Hilfsklasse von COSEM-IC *kaf_han_wan_container*

Name der COSEM-IC	Aufgabe
kaf_lmn_container	Containerklasse zum Einspielen eines Zählerprofils
kaf_lmn_index	Hilfsklasse von COSEM-IC *kaf_lmn_container*
generic_meter_sensor_setup	Konfiguration eines Zählers eines anderen Mediums
proxy_server	Konfiguration der CLS-Proxy-Komponente
qos_sq_setup	Konfiguration des *Quality-Of-Service*
gateway_signed_extended_register	Abbildung eines Messwert-Registers im SMGW
role_setup	Konfiguration einer bestimmten Nutzerrolle
taf01	Konfiguration des Tarifanwendungsfalls TAF1 im SMGW
taf02	Konfiguration des Tarifanwendungsfalls TAF2 im SMGW
taf07	Konfiguration des Tarifanwendungsfalls TAF7 im SMGW
taf09	Konfiguration des Tarifanwendungsfalls TAF9 im SMGW
taf10	Konfiguration des Tarifanwendungsfalls TAF10 im SMGW
taf14	Konfiguration des Tarifanwendungsfalls TAF14 im SMGW
tr_certificate_setup	Konfiguration der Zertifikate des SMGW
user_setup	Beschreibt den konkreten Nutzer

Das Management des kryptografischen Schlüsselmaterials erfolgt durch einen erfolgreich an der WAN-Schnittstelle authentisierten GWA unter Benutzung der COSEM-Interface-Klasse *user_setup*. Das vom GWA dabei eingepflegte Schlüsselmaterial muss dabei unter Benutzung der COSEM-IC *role_setup* eindeutig einer Benutzerschnittstelle zugeordnet werden, damit es in einem Kommunikationsprofil (s. COSEM-IC *kaf_han_wan_container*) angewendet werden kann.

4.4.3.1 Verwaltung der Zählerprofile

IC *kaf_lmn_container*

Instanzen dieser Klasse fassen im Sinne eines Protokoll-Stapels alle Festlegungen zu einem LMN-Anwendungsfall zusammen. Diese Klasse kennt keine Methoden. Die Parameter dieser Klasse sind wie folgt:

Tabelle 14: Attribute der IC *kaf_lmn_container*

Name des XML-Elements	XML-Datentyp	Beschreibung
logical_name	String	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
kaf_object	*TYPE_object* von *kaf_han_wan_index*	Angabe des *kaf_han_wan_index* Das Objekt muss angegeben werden.
e_meter_device_object	*TYPE_object* von *e_meter_sensor_setup*	Angabe des *e_meter_sensor_setup*. Entweder dieses Objekt oder das Objekt ‚generic_device_object' muss angegeben werden.
generic_device_object	*TYPE_object* von *generic_sensor_setup*	Angabe des *generic_meter_sensor_setup*. Entweder dieses Objekt oder das Objekt ‚e_meter_device_object' muss angegeben werden.

IC *kaf_lmn_index*

Instanzen dieser Klasse fassen im Sinne eines Protokoll-Stapels alle Festlegungen zu einem LMN- Anwendungsfall zusammen. Sie enthält die Objekt-Referenzen auf die unten beschriebenen Objekte. Diese Klasse kennt keine Methoden. Die Attribute dieser Klasse sind wie folgt:

Tabelle 15: Attribute der IC *kaf_lmn_index*

Name des XML-Elements		XML-Datentyp	Beschreibung
logical_name		string	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
Interface		string	Textuelle Angabe der Schnittstelle, an der der Kanal verwendet werden soll. Das Attribut muss angegeben werden. Der Wert muss auf ‚IF_GW_MTR' gesetzt werden.
device_reference	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) des für diesen Anwendungsfall festgelegten Instanz einer Interface-Klasse vom Typ
	class_id	unsignedShort	

Name des XML-Elements		XML-Datentyp	Beschreibung
	class_version	unsignedByte	*abstract_device_setup* (d.h. jede davon abgeleitete COSEM-IC, wie *generic_meter_sensor_setup*, ist möglich). Das Attribut muss angegeben werden. Der Wert des Attributs muss dem referenzierten Objekt entsprechen. Im Fall einer Referenz auf ein Objekt der IC *e_meter_sensor_setup* muss das Attribut ‚logical_name' dem Schema ‚01005e318002.LogicalDevice. sm', das Attribut ‚class_id' dem Wert ‚32809' und das Attribut ‚class_version' dem Wert ‚0' entsprechen. Im Fall einer Referenz auf ein Objekt der IC *generic_meter_sensor_setup* muss das Attribut ‚logical_name' dem Schema ‚01005e318003.LogicalDevice. sm', das Attribut ‚class_id' dem Wert ‚32810' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
register_tx_bytes	*logical_name*	string	Optional nutzbarer Verweis auf ein Objekt vom Typ IC *Register* (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version). Der von diesem Register kumulierte Zählerstand dient der Zählung der in Empfangs-Richtung (LMN an SMGW) transportierten Bytes. In dieser Produktversion wird das Attribut nicht genutzt. Das Attribut darf in der vorliegenden Variante des SMGW nicht angegeben werden.
	class_id	unsignedShort	
	class_version	unsignedByte	
register_rx_bytes	*logical_name*	string	Optional nutzbarer Verweis auf ein Objekt vom Typ IC *Register* (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version). Der von diesem Register kumulierte Zählerstand dient der Zählung der in Empfangs-Richtung (SMGW an LMN) transportierten Bytes. In dieser Produktversion wird das Attribut nicht genutzt. Das Attribut darf in der vorliegenden Variante des SMGW nicht angegeben werden.
	class_id	unsignedShort	
	class_version	unsignedByte	

IC *e_meter_sensor_setup*

Instanzen der Klasse *e_meter_sensor_setup* werden zur Konfiguration von Stromzählern benutzt. Das Einbringen der Konfiguration eines Zählers erfolgt durch den erfolgreich authentisierten GWA, indem an der WAN-Schnittstelle die folgenden XML-Daten via WKS1 übermittelt werden. Diese Klasse besitzt keine Methoden. Die Parameter sind wie folgt:

Tabelle 16: Attribute der IC *e_meter_sensor_setup*

Name des XML-Elements	XML-Datentyp	Beschreibung
logical_name	String	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
device_identifier	String	Bezeichner des Geräts. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps und den Vorgaben der DIN 43863-5 entsprechen.
device_type	String	Bezeichner des Gerätetyps. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.

Name des XML-Elements		XML-Datentyp	Beschreibung
active		Boolean	Aktivitätsstatus des Geräts. Bei ‚*true*' können Attribut-Inhalte / -Messwerte vom physischen Zähler kommend übernommen, in der Sensorwerte-Liste abgelegt und zur Tarifierung verwendet werden. Bei ‚false' ist der Objekt-Zustand eingefroren. Das Attribut muss angegeben werden. Der Wert muss entweder auf ‚false' oder ‚true' gesetzt werden.
samplerate		unsignedInt	(read-only); Das Rohdaten-Abtast-Intervall wird in Sekunden angegeben. Der Zahlenwert ‚*0*' legt den Zähler-Push-Mode fest. In diesem Fall wird jeder neue, vom Zähler kommende Wert vom SMGW übernommen. Das Attribut ist ‚read-only'[11]; das SMGW legt den je nach konkret vorliegendem Tarifprofil den zu benutzenden Wert fest.
driver_reference	*logical_name*	String	Bezeichner der diesem Zähler zugeordneten Instanz der COSEM-Klasse „*driver*"; maximal 255 Zeichen gemäß dem XML-String-Muster "[0-9a-f]{12}(\.[a-z0-9\-]{1,63})*" unter Angabe der ID der COSEM-Klasse in *class_id* und der Version der COSEM-Klasse in *class_version.* Das Attribut muss angegeben werden. Die möglichen Werte werden in Tabelle 15 beschrieben.
	class_id	unsignedShort	
	class_version	unsignedByte	

11 Die Angabe "read-only" bezieht sich stets auf die Möglichkeit des GWA, diese Werte zu verändern, d.h. für als "read-only" gesetzte Elemente kann der GWA nicht schreibend zugreifen.

Name des XML-Elements	XML-Datentyp	Beschreibung
com_type	String	Bezeichner der Art der Verschlüsselung für die Geräte-Kommunikation. Das Attribut muss angegeben werden. Der Wert muss entweder auf ‚AES' oder ‚TLS' gesetzt werden.
com_scenario	string	Bezeichner der LMN-Kommunikationsszenarios mit dem Gerät. Das Attribut muss angegeben werden. ‚LKS1' muss dabei für eine bidirektionale, ‚LKS2' für eine unidirektionale Kommunikation zwischen SMGW und Zähler angegeben werden. Der Wert muss entweder auf ‚LKS1' oder LKS2' gesetzt werden.
sensor_key	hexBinary	Zählerindividueller symmetrischer Schlüssel *MK* des Messgeräts für den Aufbau des Kommunikationsszenarios LKS1 zur Abfrage eines Zählers oder für die Entschlüsselung bzw. CMAC-Prüfung von nach LKS2 vom Zähler gesendeten Messwerten. Es handelt sich bei *MK*, um einen zugelieferten symmetrischen Schlüssel, der vom Gateway Administrator einzubringen ist. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen und hat eine Länge von 32 Zeichen. Um eine Kommunikation zwischen SMGW und Zähler zu ermöglichen, muss dieser Schlüssel korrekt konfiguriert werden. Beim Auslesen der Klasse *e_meter_sensor_setup* wird als Wert des Attributs nicht der tatsächliche Schlüssel sondern stets ‚00000000000000000000000000000000' zurückgegeben. Das Attribut darf beim Einspielen nicht mit dem Wert ‚00000000000000000000000000000000' belegt werden.

Name des XML-Elements	XML-Datentyp	Beschreibung
key_id_prv_lmn_gw	hexBinary	(read only); Key-ID zum privaten Schlüssel des SMGW zum Zertifikat GW_LMN_TLS_CRT.
c_tls_lmn_gw	hexBinary	(read-only) Das TLS-Zertifikat GW_LMN_TLS_CRT des SMGW für den Aufbau des Kommunikationsszenarios LKS1 zur Abfrage eines drahtgebundenen Zählers
c_tls_lmn_sensor	hexBinary	(read-only) Das TLS-Zertifikat MTR_TLS_CRT des Messgerätes für den Aufbau des Kommunikationsszenarios LKS1 zur Abfrage eines drahtgebundenen Zählers.
tls_type	string	Bezeichner des Typs der TLS-Kommunikation aus Sicht des SMGW. Das Attribut muss angegeben werden. Der Wert muss auf ‚TLS-Client' gesetzt werden.
tls_keep_alive	boolean	Angabe, ob bei der Kommunikation zwischen SMGW und einem drahtgebundenen Zähler der TLS-Kanal aufgehalten werden muss (*Keepalive*). Per ‚*true*' wird festgelegt, den TLS-Kanal nach einer Applikations-Aktion bis zur maximalen Sitzungslänge geöffnet zu lassen. ‚*false*' definiert, den TLS-Kanal nach einer Applikations-Aktion wieder zu schließen. Unabhängig von dem Wert des Attributs wird der TLS Kanal zum Zähler unter Berücksichtigung der Einschränkung aus ‚tls_max_session_time' immer offen gehalten. Das Attribut muss angegeben werden. Der Wert muss entweder auf ‚true' oder auf ‚false' gesetzt werden. Es wird im SMGW nicht ausgewertet.

Name des XML-Elements	XML-Datentyp	Beschreibung
tls_max_idle_time	unsignedInt	Angabe der maximal zulässigen Zeit, in der der TLS-Kanal nicht benutzt wird (in Sekunden). Nach Verstreichen dieser Zeitspanne wird der TLS-Kanal geschlossen. Der TLS Kanal wird zum Zähler unter Berücksichtigung der Einschränkung aus ‚tls_max_session_time' immer offengehalten (s.a. ‚tls_keepalive') Das Attribut muss angegeben werden. Der Wert muss zwischen 1 und 3600 liegen. Es wird im SMGW nicht ausgewertet.
tls_max_session_time	unsignedInt	Angabe der maximal zulässigen Zeit für die Session des TLS-Kanals (in Sekunden). Nach Verstreichen dieser Zeitspanne wird der TLS-Kanal geschlossen. Unabhängig vom Wert des Attributs wird der TLS Kanal immer maximal 28 Tage oder bis 5 MB Datenvolumen über den Kanal transferiert wurden offen gehalten und anschließend wieder neu aufgebaut. Das Attribut muss angegeben werden. Der Wert muss zwischen 0 und 172800 liegen. Es wird im SMGW nicht ausgewertet.

Name des XML-Elements		XML-Datentyp	Beschreibung
values	*value*	hexBinary	Sequenz der übertragenen Messwerte des Zählers. das XML-Element ‚*value*' kann mehrfach auftauchen, die Anzahl der Messwerte wird in Form des Attributs ‚*count*' vom Typ *unsignedShort* des XML-Elements ‚*values*' angegeben. Diese Liste gibt in Form von OBIS-Kennzahlen jene vom Messgerät zu liefernden Inhalte wieder, die vom Messgerät erfasst werden und für die weiterführende Tarifierung verwendet werden können. Das Attribut muss angegeben werden. Der Wert muss einer gültigen OBIS Kennzahl entsprechen (6 Blöcke, Wertbereich jeweils hexadezimal 00 – ff, bspw. ‚010001080077').
transformer_ratio_u		unsignedLong	Angabe des Wandlerfaktors des Zählerprofils gemäß [BSI TR-03109, Kap. 4.4.2] Das Attribut kann angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.
transformer_ratio_i		unsignedLong	Angabe des Wandlerfaktors des Zählerprofils gemäß [BSI TR-03109, Kap. 4.4.2] Das Attribut kann angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.

Name des XML-Elements	XML-Datentyp	Beschreibung
balancing_mode	boolean	Bei ‚*true*' erfolgt die Messung als saldierende Messung, ansonsten nicht. Das Attribut kann angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.

Im Attribut ‚*driver_reference*' muss abhängig vom Hersteller und vom Typ des physischen Zählers der durch das Profil im SMGW repräsentiert wird, einer der folgenden Werte angegeben werden.

Tabelle 17: Mögliche Werte des Attributs *driver_reference* in IC *e_meter_sensor_setup*

Hersteller und Typ des angebundenen Zählers	logical_name	class_id	class_version
Elster AS300/AM200 (vom Zähler werden unidirektional, gemäß ‚LKS2', folgende Messgrößen übermittelt: ‚0101010800ff', ‚0101020800ff', 0101010700ff'). Die Messgröße ‚0101010800ff' wird bei dieser ‚driver_reference' in der Einheit kWh mit drei Nachkommastellen erwartet.	01005e31805f.LogicalDevice-SMGW.sm	32814	0
Elster AS300/AM200 (vom Zähler werden unidirektional, gemäß ‚LKS2', folgende Messgrößen übermittelt: ‚0101010800ff')	01005e31805e.LogicalDevice-SMGW.sm	32814	0
Elster AS300/AM200 (vom Zähler werden unidirektional, gemäß ‚LKS2', folgende Messgrößen übermittelt: ‚0101010800ff', ‚0101020800ff', 0101010700ff').	01005e31805d.LogicalDevice-SMGW.sm	32814	0

Hersteller und Typ des angebundenen Zählers	logical_name	class_id	class_version
Die Messgröße ‚0101010800ff' wird bei dieser ‚driver_reference' in der Einheit kWh ohne Nachkommastellen erwartet.			
Elster AS300/AM200 (vom Zähler werden unidirektional, gemäß ‚LKS2', folgende Messgrößen übermittelt: ‚0101010800ff', ‚0101020800ff', ‚0101050800ff', ‚0101060800ff', ‚0101070800ff', ‚0101080800ff')	01005e31805b.LogicalDevice-SMGW.sm	32814	0
Easymeter Q3D (vom Zähler werden unidirektional, gemäß ‚LKS2', folgende Messgrößen übermittelt: ‚0101010800ff', ‚0101010700ff', sowie vier weitere Messgrößen, die vom SMGW nicht ausgewertet werden)	01005e318059.LogicalDevice-SMGW.sm	32814	0
Easymeter Q3BA1000/1200 (vom Zähler werden unidirektional, gemäß ‚LKS2', folgende Messgrößen übermittelt: ‚0101010800ff', ‚0101010801ff', ‚0101010802ff', ‚0101010700ff', sowie vier weitere Messgrößen, die vom SMGW nicht ausgewertet werden)	01005e318058.LogicalDevice-SMGW.sm	32814	0
Easymeter Q3BA1020 (vom Zähler werden unidirektional, gemäß ‚LKS2', folgende Messgrößen übermittelt: ‚0101010800ff', ‚0101020800ff', ‚0101010801ff', ‚0101010802ff', zwei weitere Messwerte, die vom SMGW nicht ausgewertet werden, ‚0101010700ff', sowie vier weitere Messgrößen, die vom SMGW nicht ausgewertet werden)	01005e318057.LogicalDevice-SMGW.sm	32814	0
Easymeter Q3BA1030 (vom Zähler werden unidirektional, gemäß ‚LKS2', folgende Messgrößen übermittelt: ‚0101020800ff', sowie sieben weitere Messgrößen, die vom SMGW nicht ausgewertet werden)	01005e318056.LogicalDevice-SMGW.sm	32814	0

Hersteller und Typ des angebundenen Zählers	logical_name	class_id	class_version
Itron OpenWay 3.HZ (Messwerte werden bidirektional, gemäß ‚LKS1', vom Zähler abgefragt)	01005E31809f.LogicalDevice-SMGW.sm	32816	0
EMH eBZD (Messwerte werden bidirektional, gemäß ‚LKS1', vom Zähler abgefragt)	01005E31809f.LogicalDevice-SMGW.sm	32816	0
Weitere Zähler die gemäß ‚LKS1' über eine TLS Verschlüsselung mittels des Protokolls ‚SML' mit dem SMGW kommunizieren	01005E31809f.LogicalDevice-SMGW.sm	32816	0
Alternative Konfiguration für weitere Zähler die gemäß ‚LKS1' über eine TLS Verschlüsselung mittels des Protokolls ‚SML' mit dem SMGW kommunizieren	01005E318040.LogicalDevice-SMGW.sm	32813	0
Generische Angabe für Zähler, die Messwerte gemäß ‚LKS2' entsprechend der Vorgaben aus [OMS-2] versenden	01005E31805A. LogicalDevice-SMGW.sm	32814	0

IC *generic_meter_sensor_setup*

Instanzen der Klasse *generic_meter_sensor_setup* werden zur Konfiguration von Zählern aller anderen Medien außer Strom (also Gas, Wasser etc.) benutzt. Diese Klasse kennt keine Methoden. Die Attribute dieser Klasse sind wie folgt:

Tabelle 18: Attribute der IC *generic_meter_sensor_setup*

Name des XML-Elements	XML-Datentyp	Beschreibung
logical_name	string	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse.

Name des XML-Elements	XML-Datentyp	Beschreibung
		Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
device_identifier	string	Bezeichner des Geräts Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps und den Vorgaben der DIN 43863-5 entsprechen. Alternativ kann auch die Identifikationsnummer einer M-Bus Adresse nach OMS-2 angegeben werden.
device_type	string	Bezeichner des Gerätetyps Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
active	boolean	Aktivitätsstatus des Geräts. Bei ‚*true*' können Attribut-Inhalte / -Messwerte vom physischen Zähler kommend übernommen, in der Sensorwerte-Liste abgelegt und zur Tarifierung verwendet werden. Bei ‚false' ist der Objekt-Zustand eingefroren. Das Attribut muss angegeben werden. Der Wert muss entweder auf ‚false' oder ‚true' gesetzt werden.
samplerate	unsignedInt	(read-only); Das Rohdaten-Abtast-Intervall wird in Sekunden angegeben. Der Zahlenwert ‚*0*' legt den „Zähler-Push"-Modus fest. In diesem Fall wird jeder neue, vom Zähler kommende Wert vom

Name des XML-Elements		XML-Datentyp	Beschreibung
			SMGW übernommen. Das Attribut ist ‚read-only'; das SMGW legt den je nach konkret vorliegendem Tarifprofil den zu benutzenden Wert fest.
driver_reference	*logical_name*	string	Bezeichner der diesem Zähler zugeordneten Instanz der COSEM-Klasse „*driver*". Maximal 255 Zeichen gemäß dem XML-String-Muster "[0-9a-f]{12}(\.[a-z0-9\-]{1,63})*" unter Angabe der ID der COSEM-Klasse in *class_id* und der Version der COSEM-Klasse in *class_version.* Das Attribut muss angegeben werden. Die möglichen Werte werden in Tabelle 17 beschrieben.
	class_id	unsignedShort	
	class_version	unsignedByte	
com_type		string	Bezeichner der Art der Verschlüsselung für die Geräte-Kommunikation. Das Attribut muss angegeben werden. Der Wert muss entweder auf ‚AES' oder ‚TLS' gesetzt werden.
com_scenario		string	Bezeichner der LMN-Kommunikationsszenarios mit dem Gerät. Das Attribut muss angegeben werden. ‚LKS1' muss dabei für eine bidirektionale, ‚LKS2' für eine unidirektionale Kommunikation zwischen SMGW und Zähler angegeben werden. Der Wert muss entweder auf ‚LKS1' oder LKS2' gesetzt werden.
sensor_key		hexBinary	Zählerindividueller symmetrischer Schlüssel *MK* des Messgeräts für den Aufbau des Kommunikationsszenarios LKS1 zur Abfrage eines drahtgebundenen Zählers oder für die Entschlüsselung bzw. CMAC-Prüfung von nach LKS2 vom Zähler gepushten Telegrammen. Es

Name des XML-Elements	XML-Datentyp	Beschreibung
		handelt sich bei *MK* um einen zugelieferten symmetrischen Schlüssel, der vom Gateway Administrator einzubringen ist. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen und hat eine Länge von 32 Zeichen. Um eine Kommunikation zwischen SMGW und Zähler zu ermöglichen, muss dieser Schlüssel korrekt konfiguriert werden. Beim Auslesen der Klasse *generic_meter_sensor_setup* wird als Wert des Attributs nicht der tatsächliche Schlüssel sondern stets ‚00000000000000000000000000000000' zurückgegeben. Das Attribut darf beim Einspielen nicht mit dem Wert ‚00000000000000000000000000000000' belegt werden.
key_id_prv_lmn_gw	hexBinary	(read only); Key-ID des privaten Schlüssels des SMGW zum Zertifikat GW_LMN_TLS_CRT.
c_tls_lmn_gw	hexBinary	(read-only); Das TLS-Zertifikat GW_LMN_TLS_CRT des SMGW für den Aufbau des Kommunikationsszenarios LKS2 zur Abfrage eines drahtlosen Zählers.
c_tls_lmn_sensor	hexBinary	(read-only); Das TLS-Zertifikat MTR_TLS_CRT des Messgerätes für den Aufbau des Kommunikationsszenarios LKS2 zur Abfrage eines drahtlosen Zählers.
tls_type	String	Bezeichner des Typs der TLS-Kommunikation aus Sicht des SMGW. Das Attribut muss angegeben werden. Der Wert muss auf ‚TLS-Client' gesetzt werden.

Name des XML-Elements	XML-Datentyp	Beschreibung
tls_keepalive	Boolean	Angabe, ob bei der Kommunikation zwischen SMGW und Zähler der TLS-Kanal aufgehalten werden muss (*Keepalive*). Per ‚*true*' wird festgelegt, den TLS-Kanal nach einer Applikations-Aktion bis zur maximalen Sitzungslänge geöffnet zu lassen. ‚*false*' definiert, den TLS-Kanal nach einer Applikations-Aktion wieder zu schließen. Unabhängig von dem Wert des Attributs wird der TLS Kanal zum Zähler unter Berücksichtigung der Einschränkung aus ‚tls_max_session_time' immer offen gehalten. Das Attribut muss angegeben werden. Der Wert muss entweder auf ‚true' oder auf ‚false' gesetzt werden. Es wird im SMGW nicht ausgewertet.
tls_max_idle_time	unsignedInt	Angabe der maximal zulässigen Zeit, in der der TLS-Kanal nicht benutzt wird (in Sekunden). Nach Verstreichen dieser Zeitspanne wird der TLS-Kanal geschlossen. Der TLS Kanal wird zum Zähler unter Berücksichtigung der Einschränkung aus ‚tls_max_session_time' immer offengehalten (s.a. ‚tls_keepalive'). Das Attribut muss angegeben werden. Der Wert muss zwischen 1 und 3600 liegen. Es wird im SMGW nicht ausgewertet.
tls_max_session_time	unsignedInt	Angabe der maximal zulässigen Zeit für die Session des TLS-Kanals (in Sekunden). Nach Verstreichen dieser Zeitspanne wird der TLS-Kanal geschlossen.

Name des XML-Elements		XML-Datentyp	Beschreibung
			Unabhängig vom Wert des Attributs wird der TLS Kanal immer maximal 28 Tage oder bis 5 MB Datenvolumen über den Kanal transferiert wurden offen gehalten und anschließend wieder neu aufgebaut. Das Attribut muss angegeben werden. Der Wert muss zwischen 0 und 172800 liegen. Es wird im SMGW nicht ausgewertet.
values	*value*	hexBinary	Sequenz der übertragenen Messwerte des Zählers. das XML-Element ‚*value*' kann mehrfach auftauchen, die Anzahl der Messwerte wird in Form des Attributs ‚*count*' vom Typ *unsignedShort* des XML-Elements ‚*values*' angegeben. Diese Liste gibt in Form von OBIS-Kennzahlen jene vom Messgerät zu liefernden Inhalte, die vom Messgerät erfasst werden und für die weiterführende Tarifierung verwendet werden können. Das Attribut muss angegeben werden. Der Wert muss einer gültigen OBIS Kennzahl entsprechen (6 Blöcke, Wertbereich jeweils hexadezimal 00 – ff, bspw. ‚010001080077').

Im Attribut ‚driver_reference' muss abhängig vom Hersteller und vom Typ des physischen Zählers der durch das Profil im SMGW repräsentiert wird, einer der folgenden Werte angegeben werden.

Tabelle 19: Mögliche Werte des Attributs *driver_reference* in IC *generic_meter_sensor_setup*

Hersteller und Typ des angebundenen Zählers	logical_name	class_id	class_version
Landis+Gyr G350 (vom Zähler werden unidirektional, gemäß ‚LKS2', folgende Messgrößen übermittelt: ‚0700030100ff', sowie zwei weitere Messgrößen, die vom SMGW nicht ausgewertet werden)	01005e318055.LogicalDevice-SMGW.sm	32814	0
Honeywell / Elster BK-G6AT (vom Zähler wird unidirektional, gemäß ‚LKS2', folgende Messgröße übermittelt: ‚0700030100ff')	01005E318051. LogicalDevice-SMGW.sm	32814	0
Engelmann Sensus Star U (vom Zähler werden unidirektional, gemäß ‚LKS2', folgende Messgrößen übermittelt: ‚0600010000ff', ‚0600020000ff', ‚0600090000ff', ‚0600080000ff', ‚06000A0000ff', ‚06000B0000ff' sowie drei weitere Messgrößen, die vom SMGW nicht ausgewertet werden)	01005E318052. LogicalDevice-SMGW.sm	32814	0
Landis+Gyr T550a (vom Zähler werden unidirektional, gemäß ‚LKS2', folgende Messgrößen übermittelt: ‚0600010000ff', ‚0600020000ff', ‚0600080000ff', ‚0600090000ff', ‚06000A0000FF', ‚06000B0000FF', sowie drei weitere Messgrößen, die vom SMGW nicht ausgewertet werden)	01005E318053. LogicalDevice-SMGW.sm	32814	0
Landis+Gyr T550 (vom Zähler werden unidirektional, gemäß ‚LKS2', folgende Messgrößen übermittelt: ‚0600010000ff', 0600020000ff', ‚0600080000ff', ‚0600090000ff', ‚06000A0000FF', ‚06000B0000FF', sowie drei weitere Messgrößen, die vom SMGW nicht ausgewertet werden)	01005E318054. LogicalDevice-SMGW.sm	32814	0

Hersteller und Typ des angebundenen Zählers	logical_name	class_id	class_version
Generische Angabe für Zähler, die Messwerte gemäß ‚LKS2' entsprechend der Vorgaben aus [OMS-2] versenden	01005E31805A. LogicalDevice-SMGW.sm	32814	0

Bei der Verwaltung der Zählerprofile erfolgt der Zugriff jeweils auf dem kompletten Container. Bei Auslieferung ist im SMGW kein Zählerprofil vorhanden.

Dabei sind die folgenden OBIS Kennzahlen zu verwenden:

- *kaf_lmn_container:* *01005E318000*
- *kaf_lmn_index:* *01005E318001*
- *e_meter_sensor_setup:* *01005E318002*
- *generic_meter_sensor_setup:* *01005E318003*

Die Instanzen der Klasse sind der Logical Device ID ldevid des Zählers zugeordnet. Bei mehreren Zählerprofilen erfolgt die Unterscheidung also durch die Logical Device ID und nicht durch die OBIS Kennzahl. Der GWA kann Zählerprofile im SMGW anlegen. Profile werden mittels des HTTP-Verbs POST auf dem SMGW angelegt. Vorbedingung ist, dass das Zählerprofil das angelegt werden soll zuvor nicht bereits angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 20: Anlegen eines Zählerprofils

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Anlegen eines Zählerprofils	Request-Body	HTTP Response Statuscode	Response-Body
Anlegen einer neuen Instanz	POST	<PoC>/cosem/ldevs/	*kaf_lmn_container* und zugehörige Objekte	201	Leer

Der GWA kann Zählerprofile aus dem SMGW auslesen. Profile werden mittels des HTTP-Verbs GET ausgelesen. Vorbedingung ist, dass das Zählerprofil das ausgelesen werden soll zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 21: Auslesen eines Zählerprofils

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Auslesen eines Zählerprofils	Request-Body	HTTP Response Statuscode	Response-Body
Auslesen einer bestehenden Instanz	GET	<PoC>/cosem/ldevs/1mtr0110111111.sm/objects/01005e318000	---	200	*kaf_lmn_container* und zugehörige Objekte

Der GWA kann Zählerprofile aus dem SMGW löschen. Profile werden mittels des HTTP-Verbs DELETE gelöscht.

Vorbedingung ist, dass das Zählerprofil das gelöscht werden soll nicht durch ein Tarifprofil referenziert wird. Falls das Zählerprofil durch das Tarifprofil referenziert wird, ist ein Löschen nicht möglich. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 22: Löschen eines Zählerprofils

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Löschen eines Zählerprofils	Request-Body	HTTP Response Statuscode	Response-Body
Löschen einer bestehenden Instanz	DELETE	<PoC>/cosem/ldevs/1mtr0110111111.sm/	---	200	---

Der GWA kann Zählerprofile im SMGW aktualisieren. Profile werden mittels des HTTP-Verbs UPDATE aktualisiert. Die Aktualisierung erfolgt stets auf den gesamten Container.

Vorbedingung ist, dass das Zählerprofil das aktualisiert werden soll zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 23: Aktualisieren eines Zählerprofils

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Aktualisieren eines Zählerprofils	Request-Body	HTTP Response Statuscode	Response-Body
Aktualisieren einer bestehenden Instanz	PUT	<PoC>/cosem/ldevs/1mtr0110111111.sm/objects/01005e318000	kaf_lmn_container und zugehörige Objekte mit den gewünschten Änderungen	200	---

4.4.3.2 Verwaltung der WAN- oder HAN-Profile

IC *kaf_han_wan_container*

Instanzen dieser Klasse fassen im Sinne eines Protokoll-Stapels alle Festlegungen zu einem HAN- oder WAN-Kommunikations-Anwendungsfall zusammen. Diese Klasse kennt keine Methoden.

Mit einer Instanz dieser Klasse kann ein Letztverbraucher, ein EMT, ein Service-Techniker oder ein CLS-Gerät im SMGW angelegt oder modifiziert werden. Auch die GWA-Kommunikationsprofile können über Instanzen dieser Klasse angelegt oder modifiziert werden (siehe auch Kapitel 4.4.4.4).

Falls mit einer Instanz dieser Klasse Zugangsdaten für einen autorisierten Service-Techniker oder Letztverbraucher ins Smart Meter Gateway eingebracht werden, müssen diese dem Service-Techniker oder Letztverbraucher über einen sicheren Kommunikationsweg zur Verfügung gestellt werden. Die Attribute dieser Klasse sind wie folgt:

Tabelle 24: Attribute der IC *kaf_han_wan_container*

Name des XML-Elements	XML-Datentyp	Beschreibung
logical_name	string	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
channel_object	TYPE_object von *channel_setup*	Angabe des Objekts (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) des für diesen Anwendungsfall festgelegten Instanz einer Interface-Klasse vom Typ *channel_setup*. Das Objekt muss angegeben werden.

Name des XML-Elements	XML-Datentyp	Beschreibung
qos_object	TYPE_object von *qos_sq_setup*	Angabe des Objekts (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version) des für diesen Anwendungsfall festgelegten Instanz einer Interface-Klasse vom Typ *qos_sq_setup*. Bei der Nutzung des *kaf_han_wan_container* im Kontext eines WAN Anwendungsfalls muss das Objekt angegeben werden. Bei der Nutzung des *kaf_han_wan_container* im Kontext eines HAN Anwendungsfalls entfällt das Objekt angegeben werden.
role_object	TYPE_object von *role_setup*	Angabe des Objekts (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version) des für diesen Anwendungsfall festgelegten Instanz einer Interface-Klasse vom Typ *role_setup* (mit *role* = „*EMT*“, „*GWA*“, „*CON*“, „*SRV*“, „*SMGW*“ oder „CLS“). Das Objekt muss angegeben werden.
emt_object	TYPE_object von *user_setup*	Angabe des Objekts (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version) des für diesen Anwendungsfall festgelegten Instanz einer Interface-Klasse vom Typ *user_setup* für einen EMT / GWA. Das Objekt muss angegeben werden.

Name des XML-Elements	XML-Datentyp	Beschreibung
smgw_object	TYPE_object von *tr_certificate_setup*	Verweis auf ein Objekt (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version) der Interface-Klasse vom Typ *tr_certificate_setup*. Das Objekt enthält die Zertifikate des SMGW. Im Gegensatz zu den anderen Objekten, die in der IC *kaf_han_wan_container* enthalten sind, handelt bei diesem Objekt nur um einen Verweis. Bei der Nutzung des *kaf_han_wan_container* im Kontext eines WAN Anwendungsfalls, wird in hier immer das Objekt angegeben, das die WAN Zertifikate des SMGW enthält. Bei der Nutzung des *kaf_han_wan_container* im Kontext eines HAN-Anwendungsfalls entfällt dieses Attribut. Das Objekt muss angegeben werden.

IC *kaf_han_wan_index*

Instanzen dieser Klasse fassen im Sinne eines Protokoll-Stapels alle Festlegungen zu einem HAN- oder WAN-Kommunikations-Anwendungsfall zusammen. Sie enthält die Objekt-Referenzen auf die unten beschriebenen Objekte. Diese Klasse kennt keine Methoden. Die Attribute dieser Klasse sind wie folgt:

Tabelle 25: Attribute der IC *kaf_han_wan_index*

Name des XML-Elements	XML-Datentyp	Beschreibung
logical_name	string	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
kaf_priority	string	Textuelle Angabe der Priorität des Anwendungsfalls; Auswahl zwischen den Werten „STRICT" (für striktes *Priority-Queuing*), „HIGH" (für hohe Priorität), „MEDIUM" (für mittlere Priorität), „NORMAL" (für normale Priorität) und „LOW" (für niedrige Priorität) Das Attribut muss angegeben werden. Der Wert muss entweder auf ‚STRICT' oder ‚HIGH' oder ‚MEDIUM' oder ‚LOW' gesetzt werden. Es wird im SMGW nicht ausgewertet.

Name des XML-Elements		XML-Datentyp	Beschreibung
dest_addresses	*dest_address*	anyURI	Liste von URIs zur Festlegung der Zieladressen für diesen Anwendungsfall. Es können mehrere Einträge vorgenommen werden, um Rückfall-Szenarien für den Fehlerfall bieten zu können. Der maximale Wert beträgt „*10*“. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
channel_reference	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version) des für diesen Anwendungsfall festgelegten Instanz einer Interface-Klasse vom Typ *channel_setup*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name‘ muss dem Schema ‚01005e318012.LogicalDevice.sm“, das Attribut ‚class_id‘ dem Wert ‚32797‘ und das Attribut ‚class_version‘ dem Wert ‚0‘ entsprechen.
	class_id	unsignedShort	
	class_version	unsignedByte	
qos_sq_reference	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version) des für diesen Anwendungsfall festgelegten Instanz einer Interface-Klasse vom Typ *qos_sq_setup*. Bei der Nutzung des *kaf_han_wan_container* im Kontext eines WAN Anwendungsfalls muss das Attribut angegeben werden. Bei der Nutzung des *kaf_han_wan_container* im Kontext eines HAN Anwendungsfalls entfällt das Attribut angegeben werden. Der Wert des Attributs ‚logical_name‘ muss dem Schema ‚01005e318013.LogicalDevice.sm“, das Attribut ‚class_id‘ dem Wert ‚32799‘ und das Attribut ‚class_version‘ dem Wert ‚0‘ entsprechen.
	class_id	unsignedShort	
	class_version	unsignedByte	

Name des XML-Elements		XML-Datentyp	Beschreibung
role_reference	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version) des für diesen Anwendungsfall festgelegten Instanz einer Interface-Klasse vom Typ *role_setup*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name‘ muss dem Schema ‚01005e318014.LogicalDevice.sm“, das Attribut ‚class_id‘ dem Wert ‚32798‘ und das Attribut ‚class_version‘ dem Wert ‚0‘ entsprechen.
	class_id	unsignedShort	
	class_version	unsignedByte	
ext_user_reference	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version) des für diesen Anwendungsfall festgelegten Instanz einer Interface-Klasse vom Typ *user_setup*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name‘ muss dem Schema ‚01005e318015.LogicalDevice.sm“, das Attribut ‚class_id‘ dem Wert ‚32796‘ und das Attribut ‚class_version‘ dem Wert ‚0‘ entsprechen.
	class_id	unsignedShort	
	class_version	unsignedByte	
int_user_reference	*logical_name*	string	Verweis auf ein Objekt (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version) der Interface-Klasse vom Typ *tr_certificate_setup*. Das Objekt enthält die Zertifikate
	class_id	unsignedShort	

Name des XML-Elements		XML-Datentyp	Beschreibung
	class_version	unsignedByte	des SMGW. Im Gegensatz zu den anderen Objekten, die in der IC *kaf_han_wan_container* enthalten sind, handelt bei diesem Objekt nur um einen Verweis. Bei der Nutzung des *kaf_han_wan_container* im Kontext eines WAN Anwendungsfalls, muss immer das Objekt angegeben, das die WAN Zertifikate des SMGW enthält. In diesem Fall muss der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e3180a5.LogicalDevice. sm", das Attribut ‚class_id' dem Wert ‚32821' und das Attribut ‚class_version' dem Wert ‚0' entsprechen. Bei der Nutzung des *kaf_han_wan_container* im Kontext eines HAN-Anwendungsfalls entfällt dieses Attribut.

IC *channel_setup*

Instanzen dieser Klasse konfigurieren das Verhalten eines Kommunikations-Kanals. Diese Klasse besitzt keine Methoden. Die Parameter sind wie folgt:

Tabelle 26: Attribute der IC *channel_setup*

Name des XML-Elements	XML-Datentyp	Beschreibung
logical_name	String	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
channel_id	hexBinary	Bezeichner des jeweiligen Kommunikationskanals. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
channel_name	String	Textueller Bezeichner des Kommunikations-Kanals. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen und hat eine maximale Länge von 255 Zeichen.

Name des XML-Elements	XML-Datentyp	Beschreibung
channel_purpose	String	Textuelle Angabe des Kommunikationsszenarios, für das der Kanal verwendet werden soll. Das Attribut muss angegeben werden. Der Wert muss entweder auf ‚MANAGEMENT' oder ‚ADMIN-SERVICE' oder ‚INFO-REPORT' oder ‚WAN-CLS' oder ‚HKS1' oder ‚HKS2' oder ‚HKS3' oder ‚HKS4' oder ‚HKS5' oder ‚NTP-TLS' gesetzt werden.
channel_tos	unsignedByte	Definition des im IP-Layer zu benutzenden Type-Of-Service, wobei dies durch die unteren 6 Bits geschieht und die oberen beiden Bits immer auf ‚*0*' zu setzen sind. Gemäß gilt dabei: - Bits 0-2: Vorrang gemäß [RFC 791] - Bit 3: Verzögerung (delay) = Normal (0) oder Niedrig (1) - Bit 4: Durchsatz (throughput) = Normal (0) oder High (1) - Bit 5: Verlässlichkeit (reliability) = Normal (0) oder High (1) Das Attribut muss angegeben werden. Der Wert muss entweder auf ‚0' gesetzt werden. Es wird im SMGW nicht ausgewertet.
interface	String	Textuelle Angabe der Schnittstelle, an der der Kanal verwendet werden soll. Das Attribut kann angegeben werden. Der Wert muss entweder auf ‚IF_GW_WAN' oder ‚IF_GW_HAN1' oder ‚IF_GW_HAN2' gesetzt werden.

Name des XML-Elements	XML-Datentyp	Beschreibung
tls_keep_alive	boolean	Per ‚*true*' wird festgelegt, den TLS-Kanal nach einer Applikations-Aktion bis zur maximalen Sitzungslänge geöffnet zu lassen. ‚*false*' definiert, den TLS-Kanal nach einer Applikations-Aktion wieder zu schließen. Das Attribut muss angegeben werden. Der Wert muss entweder auf ‚false' oder ‚true' gesetzt werden.
tls_max_idle_time	unsignedInt	Maximal zulässige Inaktivitäts-Zeit einer TLS-Session (in Sekunden). Nach Verstreichen dieser Zeitspanne ist der TLS-Kanal zu schließen, wenn das Attribut ‚tls_keep_alive dem Wert ‚false' entspricht. Das Attribut muss angegeben werden. Falls bei Profilen für Nutzer an IF_GW_HAN oder IF_GW_CLS größere Werte als *‚600'* verwendet werden, wird vom SMGW stets der Wert *‚600'* angewendet. Der Wert muss den folgenden Vorgaben entsprechen: • HAN: Der Wert muss zwischen 1 und 86400 liegen • WAN: Der Wert muss zwischen 1 und 3600 liegen • CLS: Der Wert muss zwischen 1 und 3600 liegen

Name des XML-Elements		XML-Datentyp	Beschreibung
tls_max_session_time		unsignedInt	Maximal zulässige Sitzungslänge einer TLS-Session (in Sekunden). Wenn das Attribut ‚tls_keepalive' den Wert ‚true' und ‚tls_max_session_time' den Wert ‚0' hat, wird der Kanal bis zur maximal möglichen Sitzungslänge (172800 Sekunden) offen gehalten und nach einer Unterbrechung, beispielsweise wegen des Erreichens der maximal möglichen Sitzungslänge von 172800 Sekunden, unmittelbar wieder aufgebaut. Diese Konfiguration ist nur zulässig, wenn das Attribut ‚channel_purpose' den Wert ‚MANAGEMENT' enthält. Andernfalls muss ein Wert größer ‚0' gesetzt werden. In beiden Fällen liegt der maximal mögliche Wert bei ‚172800'. Das Attribut muss angegeben werden. Der Wert muss den folgenden Vorgaben entsprechen: • HAN: Der Wert muss zwischen 0 und 86400 liegen • WAN: Der Wert muss zwischen 0 und 172800 liegen • CLS: Der Wert muss zwischen 0 und 172800 liegen
register_tx_bytes	*logical_name*	string	Optional nutzbarer Verweis auf ein Objekt vom Typ IC *Register* (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version). Der von diesem Register kumulierte Zählerstand dient der Zählung der in Sende-Richtung (WAN an HAN) transportierten Bytes. Das Attribut darf in der vorliegenden Variante des SMGW nicht angegeben werden.
	class_id	unsignedShort	
	class_version	unsignedByte	

Name des XML-Elements		XML-Datentyp	Beschreibung
register_rx_bytes	*logical_name*	string	Optional nutzbarer Verweis auf ein Objekt vom Typ IC *Register* (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version). Der von diesem Register kumulierte Zählerstand dient der Zählung der in Empfangs-Richtung (HAN an WAN) transportierten Bytes. Das Attribut darf in der vorliegenden Variante des SMGW nicht angegeben werden.
	class_id	unsignedShort	
	class_version	unsignedByte	

IC *qos_sq_setup*

Instanzen der Klasse *qos_sq_setup* werden vom GWA zur Konfiguration des *Quality-Of-Service* des SMGW benutzt. Diese Klasse besitzt keine Methoden. Die Parameter sind wie folgt:

Tabelle 27: Attribute der IC *qos_sq_setup*

Name des XML-Elements	XML-Datentyp	Beschreibung
logical_name	string	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
max_loop_count	unsignedShort	Maximale Anzahl von Runden
log_loop_failures	boolean	Bei ‚true' erfolgt ein Eintrag in das Logbuch, wenn die zweite oder eine weitere Runde gestartet wird. Das Attribut muss angegeben werden. Der Wert muss entweder auf ‚false' oder ‚true' gesetzt werden. Es wird im SMGW nicht ausgewertet.

Name des XML-Elements		XML-Datentyp	Beschreibung
loop_times	*waitseconds*	unsignedInt	Diese Liste muss mindestens einen Eintrag enthalten. Jeder Listeneintrag legt eine Rundenwartezeit in Sekunden fest. Der erste Listen-Eintrag legt die erste Wartezeit fest. Falls ein zweiter Listen-Eintrag vorhanden ist, legt der zweite Listen-Eintrag die zweite Rundenwartezeit fest. Weitere Listen-Einträge sind entsprechend zu verwenden. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
max_retry_count		unsignedShort	Maximale Anzahl von Wiederholungen innerhalb einer Runde.
log_retry_failures		boolean	Bei ‚true' erfolgt ein Eintrag in das Logbuch, wenn die zweite oder eine weitere Wiederholung erneut gestartet wird. Das Attribut muss angegeben werden. Der Wert muss entweder auf ‚false' oder ‚true' gesetzt werden. Es wird im SMGW nicht ausgewertet.
retry_times	*waitseconds*	unsignedInt	Diese Liste muss mindestens einen Eintrag enthalten. Jeder Listeneintrag legt eine Wiederholungswartezeit in Sekunden fest. Der erste Listen-Eintrag legt die erste Wartezeit fest. Falls ein zweiter Listen-Eintrag vorhanden ist, legt der zweite Listen-Eintrag die zweite Wiederholungswartezeit fest. Weitere Listen-Einträge sind entsprechend zu verwenden. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.

Name des XML-Elements	XML-Datentyp	Beschreibung
fallback_time	unsignedShort	Diese Wartezeit, anzugeben in Sekunden, legt eine Pause fest, bevor der Vorgang vollständig als fehlerhaft markiert beendet wird. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
log_fallback_failures	boolean	Bei ‚*true*' erfolgt ein Eintrag in das System-Log, wenn der Fallback-Fall eingetreten ist, ansonsten nicht. Das Attribut muss angegeben werden. Der Wert muss entweder auf ‚false' oder ‚true' gesetzt werden. Es wird im SMGW nicht ausgewertet.

Falls alle Versandversuche, die mit der IC *qos_sq_setup* konfiguriert sind, durchgeführt wurden und der Versandauftrag dennoch nicht erfolgreich bereitgestellt werden konnte, wird täglich versucht den Versandauftrag bereitzustellen.

IC *role_setup*

Instanzen der Klasse *role_setup* werden vom GWA zur Modellierung der im SMGW vorhandenen Benutzerrollen benutzt. Diese Klasse kennt keine Methoden. Die Attribute dieser Klasse sind wie folgt:

Tabelle 28: Attribute der IC *role_setup*

Name des XML-Elements	XML-Datentyp	Beschreibung
logical_name	string	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
role	string	Angabe der Rollen-Zugehörigkeit. Das Attribut muss angegeben werden. Der Wert muss entweder auf ‚SMGW' oder ‚GWA' oder ‚EMT' oder ‚CON' oder ‚SRV' oder ‚CLS' gesetzt werden.

Bitte achten Sie darauf, dass der Wert des Attributs *role* korrekt gesetzt wird! Sie dürfen die Rolle „GWA" nur setzen, wenn es sich wirklich um ein Kommunikationsprofil für einen Gateway Administrator handelt und keinesfalls versehentlich für ein Kommunikationsprofil, das zur Kommunikation mit einer anderen Rolle verwendet wird.

IC *user_setup*

Instanzen dieser Klasse dienen der Modellierung der Benutzer des SMGW. Die vom GWA dabei eingepflegten Konfigurationsdaten müssen dabei unter Benutzung der COSEM-IC *role_setup* eindeutig einer Benutzerschnittstelle zugeordnet werden, damit es in einem Kommunikationsprofil (s. COSEM-IC *kaf_han_wan_container*) angewendet werden kann. Diese Klasse kennt keine Methoden. Die Attribute dieser Klasse sind wie folgt:

Tabelle 29: Attribute der IC *user_setup*

Name des XML-Elements	XML-Datentyp	Beschreibung
logical_name	String	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
cert_tls	String	TLS-Authentifizierungs-Zertifikat gemäß [BSI TR-03109-1] für die HAN- oder WAN-Schnittstelle. Die maximale Größe beträgt 1024 Byte. Das Attribut muss angegeben werden, falls das Attribut ‚channel_purpose' der zugehörigen Instanz der IC ‚*channel_setup*' den Wert „MANAGEMENT", „ADMIN-SERVICE", „INFO-REPORT", „WAN-CLS", „HKS1", „HKS3", „HKS4", „HKS5" oder „NTP-TLS" hat. Das Attribut darf nicht angegeben werden, falls das Attribut ‚channel_purpose' der zugehörigen Instanz der IC ‚*channel_setup* ' den Wert „HKS2" hat. Der Wert des Attributs muss einem gültigen Zertifikat im DER-Format entsprechen, das *hexbinary* codiert wurde.

Name des XML-Elements	XML-Datentyp	Beschreibung
key_id_tls	String	Key-Identifier des TLS-Authentifizierungs-Zertifikats. Die maximale Größe beträgt 4 Byte. Das Attribut kann angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.
cert_enc	String	Schlüsseltransport-Zertifikat gemäß [BSI TR-03109-1]. Die maximale Größe beträgt 1024 Byte. Das Attribut muss angegeben werden, falls das Attribut channel_purpose der zugehörigen Instanz der IC ‚*role_setup*' den Wert „MANAGEMENT", „ADMIN_SERVICE" oder „INFO-REPORT" hat. Das Attribut darf nicht angegeben werden, falls das Attribut ‚channel_purpose' der zugehörigen Instanz der IC ‚*role_setup*' einen anderen Wert hat.
key_id_enc	String	Key-Identifier des Schlüsseltransport-Zertifikats. Die maximale Größe beträgt 4 Byte. Das Attribut kann angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.
cert_sig	String	Inhaltsdaten-Signatur-Zertifikat gemäß [BSI TR-03109-1]. Die maximale Größe beträgt 1024 Byte. Das Attribut muss angegeben werden, falls das Attribut channel_purpose der zugehörigen Instanz der IC ‚*role_setup*' den Wert „MANAGEMENT", „ADMIN_SERVICE" oder „INFO-REPORT" hat. Das Attribut darf nicht angegeben werden, falls das Attribut ‚channel_purpose' der zugehörigen Instanz der IC ‚*role_setup*' einen anderen Wert hat.

Name des XML-Elements	XML-Datentyp	Beschreibung
key_id_sig	String	Key-Identifier des Inhaltsdaten-Signatur-Zertifikats. Die maximale Größe beträgt 4 Byte. Das Attribut kann angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.
username	String	Name des Benutzers gemäß [BSI TR-03109-1] für die Authentisierung per Benutzername / Passwort an IF_GW_CON. Das Attribut muss angegeben werden, falls das Attribut channel_purpose der zugehörigen Instanz der IC ‚*role_setup*' den Wert „HKS2" hat. Das Attribut darf nicht angegeben werden, falls das Attribut ‚channel_purpose' der zugehörigen Instanz der IC ‚*role_setup*' einen anderen Wert hat.

Name des XML-Elements	XML-Datentyp	Beschreibung
password	String	Passwort des Benutzers gemäß [BSI TR-03109-1] für die Authentisierung per Benutzername / Passwort an IF_GW_CON. Das Attribut muss angegeben werden, falls das Attribut channel_purpose der zugehörigen Instanz der IC ‚*role_setup*' den Wert „HKS2" hat. Das Attribut darf nicht angegeben werden, falls das Attribut ‚channel_purpose' der zugehörigen Instanz der IC ‚*role_setup*' einen anderen Wert hat. Der Wert muss folgenden Vorgaben an die Passwort Vergabe genügen: • Länge: mindestens 10 Zeichen • Es muss mindestens ein Zeichen folgender Zeichengruppe enthalten sein: ○ Großbuchstaben ("ABCDEFGHIJKLMNOPQRSTUVWXYZ") ○ Kleinbuchstaben ("abcdefghijklmnopqrstuvwxyz") ○ Ziffern ("1234567890") ○ Sonderzeichen (!"§$%&/()=?+*~#',;.:-_) • Sonstige Zeichen sind erlaubt

Falls der Letztverbraucher sich zu häufig mit falschen Login-Daten anmeldet, werden weitere Anmeldeversuche für einen bestimmten Zeitraum unterbunden.Der Letztverbraucher kann fünf fehlerhafte Login Versuche unternehmen, bis der Zugang für einen Zeitraum von fünf Minuten gesperrt wird.

Bei der Verwaltung der WAN- oder HAN-Profile erfolgt der Zugriff jeweils auf dem kompletten Container. Bei Auslieferung ist im SMGW kein WAN- oder HAN-Profil vorhanden.

Dabei sind die folgenden OBIS-Kennzahlen zu verwenden:

- *kaf_han_wan_container:* *01005e318010*
- *kaf_han_wan_index:* *01005e318011*
- *channel_setup:* *01005e318012*
- *qos_sq_setup:* *01005e318013*
- *role_setup:* *01005e318014*
- *user_setup (emt object):* *01005e318015*
- Referenz auf *tr_certificate_setup* *01005e3180a5*

Die Instanzen der Klasse sind der Logical Device ID *ldevid* des WAN- oder HAN-Nutzers zugeordnet. Bei mehreren WAN- oder HAN-Profilen erfolgt die Unterscheidung also durch die *Logical Device ID* und nicht durch die OBIS-Kennzahl. Der GWA kann WAN- oder HAN-Profile im SMGW anlegen. Profile werden mittels des HTTP-Verbs POST auf dem SMGW angelegt. Vorbedingung ist, dass das WAN- oder HAN-Profil, das angelegt werden soll, zuvor nicht bereits angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 30: Anlegen eines HAN-/WAN-Profils

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Anlegen eines HAN-/WAN-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Anlegen einer neuen Instanz	POST	<PoC>/cosem/ldevs/	*kaf_han_wan_container* und zugehörige Objekte	201	Leer

Der GWA kann WAN- oder HAN-Profile aus dem SMGW auslesen. Profile werden mittels des HTTP-Verbs GET ausgelesen. Vorbedingung ist, dass das WAN- oder HAN-Profil, das ausgelesen werden soll, zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 31: Auslesen eines HAN-/WAN-Profils

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Auslesen eines HAN-/WAN-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Auslesen einer bestehenden Instanz	GET	<PoC>/cosem/ldevs/emt12345678.sm/objects/01005e318010	---	200	*kaf_han_wan_container* und zugehörige Objekte

Der GWA kann WAN- oder HAN-Profile aus dem SMGW löschen. Profile werden mittels des HTTP-Verbs DELETE gelöscht.

Vorbedingung ist, dass das WAN- oder HAN-Profil, das gelöscht werden soll, nicht durch ein Tarifprofil referenziert wird. Falls das WAN- oder HAN-Profil durch das Tarifprofil referenziert wird, ist ein Löschen nicht möglich.

Beim Löschen von HAN-Profilen ist eine weitere Vorbedingung, dass für den zugehörigen Nutzer keine Logeinträge im Letztverbraucher Logbuch vorhanden sind.

Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 32: Löschen eines HAN-/WAN-Profils

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Löschen eines HAN-/WAN-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Löschen einer bestehenden Instanz	DELETE	<PoC>/cosem/ldevs/ emt12345678.sm/	---	200	---

Der GWA kann WAN- oder HAN-Profile im SMGW aktualisieren. Profile werden mittels des HTTP-Verbs PUT aktualisiert. Die Aktualisierung erfolgt stets auf den gesamten Container. Vorbedingung ist, dass das WAN- oder HAN-Profil, das aktualisiert werden soll, zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 33: Aktualisieren eines HAN-/WAN-Profils

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Aktualisieren eines HAN-/WAN-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Aktualisieren einer bestehenden Instanz	PUT	<PoC>/cosem/ldevs/emt12345678.sm/objects/01005e318010	*kaf_han_wan_container* und zugehörige Objekte mit den gewünschten Änderungen	200	---

Die Aktualisierung eines WAN-Profils ist vor allem beim Wechsel der EMT- oder GWA-Zertifikate relevant.

Bei der Verwendung von Objekten der Klasse *kaf_han_wan_container* als WAN-Kommunikationsprofile muss im Attribut „int_user_reference“ stets eine Referenz auf das Objekt der Klasse *tr_certificate_setup* angegeben sein, das die Zertifikate des SMGW enthält.

4.4.3.3 Verwaltung der TAF-Profile

Der Tarifanwendungsfall TAF6 ist unabhängig von dem speziell konfigurierten TAF bei jedem TAF-Profil zusätzlich aktiv und nicht deaktivierbar. Der Versand der Messwerte, die in TAF6 erzeugt werden, wird durch das Attribut *on_demand_delivery_ref* gesteuert.

Entsprechend BSI TR-03109-1 werden für TAF6 die Messwerte zu Beginn des abrechnungstechnischen Kalendertages rückwirkend für 42 Tage vorgehalten. Ein Löschen alter Messwerte für TAF1, TAF6 und TAF7 erfolgt durch das SMGW automatisch entsprechend der Konfiguration im TAF-Profil, wobei die Messwerte mindestens 15 Monate vorgehalten werden. Für TAF9, TAF10 und TAF14 werden keine Messwerte persistent auf dem SMGW vorgehalten.

Bei allen Tarifprofilen kann das Attribut ‚pseudonym‘ gesetzt werden. Wenn dieses Attribut gesetzt ist, werden alle Messwerte die aufgrund dieses Tarifprofils erfasst und übertragen werden pseudonymisiert. Die Funktion unterscheidet dabei nicht zwischen Netzzustandsdaten und Daten, die für Abrechnungszwecke erfasst werden. Wenn das Attribut nicht gesetzt wird, ist ein Rückschluss auf den konkreten Zähler (und damit ggf. auch auf das SMGW und den Letztverbraucher) möglich.

IC *taf01*

Instanzen der Klasse *taf01* werden vom GWA zur Konfiguration des Tarifanwendungsfalls TAF1 im SMGW benutzt. Diese Klasse kennt keine Methoden. Die Attribute dieser Klasse sind wie folgt:

Tabelle 34: Attribute der IC *taf01*

Name des XML-Elements	XML-Datentyp	Beschreibung
logical_name	string	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
taf_name	string	Freier Bezeichner des Auswertungsprofils gemäß [BSI TR-03109-1]. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
taf_identifier	string	Für das SMGW eindeutiger Bezeichner für das Auswertungsprofil. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
pseudonym	string	(optional) Mit diesem Attribut kann ein Pseudonym angegeben werden. Bei der Übertragung der Messwerte werden sämtlich Hinweise auf den Zähler (Logical Device ID des Zählers) durch das Pseudonym ersetzt. Somit ist

Name des XML-Elements			XML-Datentyp	Beschreibung
				ein Rückschluss auf den konkreten Zähler oder das konkrete Smart Meter Gateway aus dem die Messwerte stammen nicht mehr möglich.
input_references	*input_reference*	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version) des Typs *gateway_signed_extended_register*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name‘ muss auf den Wert des Attributs ‚value‘ einer auf dem SMGW vorhandenen Instanz der IC ‚*e_meter_sensor_setup*‘ oder der IC ‚*generic_meter_sensor_setup*‘ und somit dem Schema ‚value.LogicalDevice-Zähler.sm‘ entsprechen. Beispielsweise kann der Wert auf 0100010800ff.1emh0005288229.sm‘ gesetzt werden. Der Wert des Attributs ‚class_id‘ muss auf ‚3‘ gesetzt werden. Der Wert des Attributs ‚class_version‘ muss auf ‚0‘ gesetzt werden.
		class_id	unsignedShort	
		class_version	unsignedByte	
number_of_tariffs			byte	Anzahl der Tarifstufen. Das Attribut muss angegeben werden. Der Wert wird vom SMGW nicht ausgewertet.
active_tariff			byte	(read-only) Dieses Attribut spiegelt den jeweils aktuell in dem TAF geltenden Tarif wider.

Name des XML-Elements			XML-Datentyp	Beschreibung
end_of_billing_period	*datetime_var*	*day_of_month*	string	Schreibt die abgeleiteten Register in die turnusgemäße Abrechnungsliste, entspricht dem Abrechnungszeitraum aus [BSI TR-03109] und dem Turnus aus den dortigen Abbildungen. Die Datentypen ‚datetime_var' und ‚datetime_intervall' können alternativ zueinander verwendet werden. Das Attribut muss angegeben werden. Bei Verwendung des Datentyps ‚datetime_var' muss der Wert den Vorgaben des Datentyps entsprechen und zwischen ‚1' und ‚31' liegen. Bei Verwendung des Datentyps ‚datetime_intervall' wird ein Startzeitpunkt angegeben und davon ausgehend eine Periode, die zyklisch das Ende des Abrechnungszeitraums angibt.
	datetime_intervall	*start_time*	dateTime	
		period	unsignedInt	
capture_period			unsignedInt	Aktualisiert die Messwertliste, aktualisiert den Eingangswert des Zählers und die abgeleiteten Register. (Registrierperiode). Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen, in Sekunden angegeben werden und zwischen ‚1' und ‚86400' liegen.
capture_time	*hour*		unsignedByte	Uhrzeit zu der der abrechnungstechnische Kalendertag beginnt. Zu diesem Zeitpunkt wird der Wert für eine Spontanauslesung gemäß TAF-6 aufgezeichnet. Das Attribut muss angegeben werden.
	minute		unsignedByte	
	second		unsignedByte	

Name des XML-Elements				XML-Datentyp	Beschreibung
					Der Wert des Attributs ‚hour' muss zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen.
consumer_reference			*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *kaf_han_wan_container*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318011.LogicalDevice-Letztverbraucher.sm", das Attribut ‚class_id' dem Wert ‚32795' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
			class_id	unsignedShort	
			class_version	unsignedByte	
delivery_references	*delivery_reference*	*comm_profile_reference*	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *kaf_han_wan_container*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318011.LogicalDevice-EMT. sm", das Attribut ‚class_id' dem Wert ‚32795' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
			class_id	unsignedShort	
			class_version	unsignedByte	
		delivery_time	*datetime_var*	*year*	Uhrzeit des Messwert-Versands. Das Attribut muss angegeben werden. Die Datentypen ‚datetime_var' und ‚datetime_intervall' können alternativ zueinander verwendet werden.
				month	string
				day_of_month	string

Name des XML-Elements					XML-Datentyp	Beschreibung
				day_of_week	string	Bei Verwendung des Datentyps ‚datetime_var' muss der Wert des Attributs ‚year' zwischen ‚2000' und ‚2199', des Attributs ‚month' zwischen ‚1' und ‚12', des Attributs ‚day_of_month' zwischen ‚1' und ‚31', des Attributs ‚hour' zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen. Der Wert des Attributs ‚day_of_week' muss entweder auf ‚Mon' oder ‚Tue' oder ‚Wed' oder ‚Thu' oder ‚Fri' oder ‚Sat' oder ‚Sun' gesetzt werden. Die Zeitangaben, die nicht gesetzt werden, werden als Wildcard angenommen. Wird beispielsweise nur das Attribut ‚minute' mit dem Wert ‚0' und das Attribut ‚second' mit dem Wert ‚0' angegeben, erfolgt der Messwertversand an jedem Tag zu jeder vollen Stunde zur 0. Minute und zur 0. Sekunde. Bei Verwendung des Datentyps ‚datetime_intervall' wird ein Startzeitpunkt angegeben und davon ausgehend eine Periode, die zyklisch den Versandzeitpunkt angibt.
				hour	unsignedByte	
				minute	unsignedByte	
				second	unsignedByte	
			datetime_interval	*start_time*	dateTime	
				period	unsignedInt	
		qos_profile_reference		*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *qos_sq_setup*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318013.LogicalDevice-EMT.sm", das Attribut ‚class_id' dem Wert ‚32799' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
				class_id	unsignedShort	
				class_version	unsignedByte	
			data_ref	*logi*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *profile_generic*. Die Werte, die in diesem

Name des XML-Elements					XML-Datentyp		Beschreibung
		data_references	*erence*	*cal_name*			Attribut angegeben sind, werden an den in ‚comm_profile_reference' festgelegten EMT versendet. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚OBIS.LogicalDevice-Zähler. sm' oder dem Schema ‚OBIS.LogicalDevice-TAF.sm', das Attribut ‚class_id' dem Wert ‚4' oder dem Wert ‚32822' und das Attribut ‚class_version' dem Wert ‚0' entsprechen. Der Wert des Attributs ‚attr_reference' muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.
				class_id	unsignedShort		
				class_version	unsignedByte		
				attr_references	attr_reference	unsignedByte	
validity_window		*start_time*			dateTime		

Name des XML-Elements		XML-Datentyp	Beschreibung
	end_time		Zeitangabe des Gültigkeitszeitraums. Das Element *start_time* muss immer und ohne die Verwendung von Wildcards angegeben werden. Das Element *end_time* muss entweder ohne Wildcards angegeben werden oder vollständig auf leer gesetzt werden. Ist *end_time* auf leer gesetzt, wird der TAF nicht eigenständig durch das SMGW beendet. Der Wert des Elements *end_time* muss vor dem 01.01.2038 liegen. Das Attribut muss angegeben werden. Der Wert muss dem Schema „yyyy-mm-ddThh:mm:ssZ“ (mit 4-stelliger Abgabe des Jahres yyyy , 2-stelliger Angabe der Minuten mm, 2-stelliger Angabe der Stunden hh und 2-stelliger Angabe der Sekunden ss) entsprechen. Der Wert ist als UTC Zeit mit einem ‚Z‘ am Ende anzugeben.
metering_point_id		string	Zählpunktbezeichnung. Das Attribut kann angegeben werden.
auto_cleanup		Boolean	Legt fest, ob ein TAF und die damit verbundenen Register und Messwerte automatisch entsprechend der folgenden Parameter gelöscht werden. Das Attribut kann angegeben werden. Der Wert muss entweder auf ‚false‘ oder ‚true‘ gesetzt werden. Es wird im SMGW nicht ausgewertet.
archive_duration		unsignedInt	Legt fest, wie lange (in Tagen) für diesen TAF relevante Messwerte vorgehalten werden müssen.

Name des XML-Elements		XML-Datentyp	Beschreibung
			Nach dem Erreichen dieser Dauer werden die für diesen TAF relevanten Messwerte automatisch durch das SMGW gelöscht. Wenn weniger als 15 Monate (458 Tage) angegeben sind, werden die TAF relevanten Daten erst nach 458 Tagen gelöscht. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
taf_state		string	(read-only) Ein TAF durchläuft grundsätzlich drei Zustände *ready*, *running* und *archive*. Ist ein TAF im Zustand ‚archive' oder im Zustand ‚running', werden einmal täglich alle den Input-Registern zugeordneten originären Messwerte, die „archive_duration" Tage (aber mindestens 458 Tage) alt oder älter sind, vom SMGW eigenständig entfernt. Ein TAF kann im Zustand "*ready*" durch den GWA rückstandsfrei entfernt werden. Ein TAF kann zusätzlich im Zustand "*archive*" durch den GWA rückstandsfrei entfernt werden, wenn er mindestens "*archive_duration*" Tage im SMGW vorgehalten worden ist. Alle von einem TAF benötigten Laufzeit-Objekte (also alle Objekte, die nicht in der Domain des EMT angeordnet sind und nicht von anderen Objekten des SMGW referenziert werden) werden mit Wechsel in den Zustand "*Running*" instanziiert und mit Wechsel in den Zustand "*archive*" entfernt.
on_demand_delivery_ref	logical_name	string	Diese Referenz verweist auf ein Objekt des Typs on_demand_delivery.
	class_id	unsignedShort	

Name des XML-Elements		XML-Datentyp	Beschreibung
	class_version	unsignedByte	Wenn die Referenz beim Einspielen gesetzt ist, wird das Objekt beim erstmaligen Einspielen des TAFs leer erzeugt. Das referenzierte Objekt wird dazu benutzt, eine Spontanablesung wie folgt zu ermöglichen: - Solange das referenzierte Objekt vom GWA nicht geschrieben wurde, liegt keine Anforderung zu einer Spontan-Ablesung vor. Sobald das referenzierte Objekt vom GWA geschrieben wurde, soll die Spontan-Ablesung gemäß der Informationen, die ‚On Demand Delivery' liefert, ausgeführt werden. - Wurde der Spontan-Ablesungsauftrag erzeugt, wird das referenzierte Objekt wieder geleert. - Das SMGw darf nur das Beschreiben von leeren Objekten des Typs ‚On Demand Delivery' zulassen. Das Attribut kann angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e31803b.LogicalDevice-TAF.sm", das Attribut ‚class_id' dem Wert ‚32820' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
sensor_signature		Boolean	Der Wert des Attributs muss entweder auf ‚false' oder ‚true' gesetzt werden. Er wird im SMGW nicht ausgewertet.

Kennzahl zu verwenden:

- *taf01: 01005E318021*

Die Instanzen der Klasse sind der Logical Device ID ldevid des TAFs zugeordnet. Bei mehreren taf01 erfolgt die Unterscheidung also durch die Logical Device ID und nicht durch die OBIS Kennzahl.

Der GWA kann TAF-Profile im SMGW anlegen. Profile werden mittels des HTTP-Verbs POST auf dem SMGW angelegt. Vorbedingung ist, dass das TAF-Profil, das angelegt werden soll, zuvor nicht bereits angelegt wurde. Weiterhin müssen folgende Referenzen vor dem Einspielen eines TAFs auf dem Gateway vorhanden sein:

- *input_reference*
- *consumer_reference*
- *com_profile_reference*
- *qos_profile_reference*

Die entsprechenden Profile müssen durch den GWA also vor dem Einspielen eines TAFs in das SMGW eingebracht worden sein. Das folgende Beispiel verdeutlicht das Einbringen eines TAF-Profils mit dem HTTP-Verb POST:

Tabelle 35: Anlegen eines *taf01*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Anlegen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Anlegen einer neuen Instanz	POST	<PoC>/cosem/ldevs/	taf01	201	Leer

Der GWA kann TAF-Profile aus dem SMGW auslesen. Profile werden mittels des HTTP-Verbs GET ausgelesen. Vorbedingung ist, dass das TAF-Profil, das ausgelesen werden soll, zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 36: Auslesen eines *taf01*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Auslesen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Auslesen einer bestehenden Instanz	GET	<PoC>/cosem/ldevs/taf0112345679.sm/objects/01005E318021	---	200	taf01

Der GWA kann TAF-Profile aus dem SMGW löschen. Profile werden mittels des HTTP-Verbs DELETE gelöscht. Vorbedingung ist, dass das TAF-Profil im Zustand „*archive*“ oder im Zustand „*ready*“ im SMGW vorhanden ist. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 37: Löschen eines *taf01*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Löschen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Löschen einer bestehenden Instanz	DELETE	<PoC>/cosem/ldevs/taf0112345679.sm/	---	200	---

Der GWA kann TAF-Profile im SMGW aktualisieren. Profile werden mittels des HTTP-Verbs UPDATE aktualisiert. Die Aktualisierung erfolgt stets auf den gesamten Container. Der GWA kann im TAF-Profil nur das Attribut „*end_time*“ anpassen und den entsprechenden TAF so beenden. Vorbedingung ist, dass das TAF-Profil, das aktualisiert werden soll, zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 38: Aktualisieren eines *taf01*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Aktualisieren eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Aktualisieren einer bestehenden Instanz	PUT	<PoC>/cosem/ldevs/taf0112345679.sm/objects/01005E318021	*taf01* mit angepasster *end_time*	200	---

IC *on_demand_delivery*

Instanzen dieser Klasse werden verwendet um Spontanauslesungen von Messwerten gemäß TAF6 zu ermöglichen. Durch diese Klasse kann die Spontanauslesung getriggert werden. Diese Klasse besitzt keine Methoden. Die Parameter sind wie folgt:

Tabelle 39: Attribute der IC on_demand_delivery

Name des XML-Elements		XML-Datentyp	Beschreibung
logical_name		string	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
com_profile_reference	*logical_name*	String	Referenz auf das Kommunikationsprofil, an das die Messwerte versendet werden. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318011.LogicalDevice-EMT.sm", das Attribut ‚class_id' dem Wert ‚32795' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
	class_id	unsignedShort	
	class_version	unsignedByte	
qos_profile_reference	*logical_name*	String	Referenz auf das QoS-Profil das für den Versand genutzt wird. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318013.LogicalDevice-EMT.sm", das Attribut ‚class_id' dem Wert ‚32799' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
	class_id	unsignedShort	
	class_version	unsignedByte	

Name des XML-Elements	XML-Datentyp	Beschreibung
meter_reading_date	TYPE_DATE_FIX	Datum des Zählerprofils, für das die Messwerte ausgegeben werden. Das Attribut muss angegeben werden. Der Wert muss dem Schema „yyyy-mm-dd“ (mit 4-stelliger Abgabe des Jahres yyyy , 2-stelliger Angabe des Monats mm, 2-stelliger Angabe des Tags dd) entsprechen.

Die OBIS Kennzahl für das Objekt lautet wie folgt:

- *on_demand_delivery:* *01005e31803b*

Das entsprechende Objekt wird automatisch im Gateway angelegt, wenn die Referenz im TAF-Profil gesetzt wurde und ist damit der Logical Device ID des TAF-Profils zugeordnet. Das Objekt kann durch den GWA nur aktualisiert werden. Sobald die Spontanauslesung durchgeführt wurde, wird das Objekt wieder geleert. Um eine neue Spontanauslesung zu veranlassen führt der GWA eine erneute Aktualisierung des Objektes durch. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 40: Aktualisieren eines IC *on_demand_delivery*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für Update der HAN-Schnittstelle	Request-Body	HTTP Response Statuscode	Response-Body
Aktualisieren einer bestehenden Instanz	PUT	<PoC>/cosem/ldevs/taf0112345679.sm/objects/01005e31803b	*on_demand_delivery*	200	---

IC *taf02*

Instanzen der Klasse *taf02* werden vom GWA zur Konfiguration des Tarifanwendungsfalls TAF2 im SMGW benutzt. Diese Klasse kennt keine Methoden. Die Attribute dieser Klasse sind wie folgt:

Tabelle 41: Attribute der IC *taf02*

Name des XML-Elements			XML-Datentyp	Beschreibung
logical_name			string	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
taf_identifier			string	Für das SMGW eindeutiger Bezeichner für das Auswertungsprofil. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
pseudonym			string	(optional) Mit diesem Attribut kann ein Pseudonym angegeben werden. Bei der Übertragung der Messwerte werden sämtlich Hinweise auf den Zähler (Logical Device ID des Zählers) durch das Pseudonym ersetzt. Somit ist ein Rückschluss auf den konkreten Zähler oder das konkrete Smart Meter Gateway aus dem die Messwerte stammen nicht mehr möglich.
input_references	*input_reference*	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) des Typs *gateway_signed_extended_register*.

Name des XML-Elements			XML-Datentyp	Beschreibung
		class_id	unsignedShort	Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss auf den Wert des Attributs ‚value' einer auf dem SMGW vorhandenen Instanz der IC ‚*e_meter_sensor_setup*' oder der IC ‚*generic_meter_sensor_setup*' und somit dem Schema ‚value.LogicalDevice-Zähler.sm' entsprechen. Beispielsweise kann der Wert auf , 0100010800ff.1emh0005288229.sm' gesetzt werden. Der Wert des Attributs ‚class_id' muss auf ‚3' gesetzt werden. Der Wert des Attributs ‚class_version' muss auf ‚0' gesetzt werden.
		class_version	unsignedByte	
number_of_tariffs			Byte	Anzahl der Tarifstufen Das Attribut muss angegeben werden. Der Wert zwischen ‚1' und ‚8' liegen.
active_tariff			byte	(read-only) Dieses Attribut spiegelt den jeweils aktuell in dem TAF geltenden Tarif wider.
end_of_billing_period	*datetime_var*	*day_of_month*	string	Schreibt die abgeleiteten Register in die turnusgemäße Abrechnungsliste. entspricht dem Abrechnungszeitraum aus [BSI TR-03109] und dem Turnus aus den dortigen Abbildungen.
		hour	unsignedByte	

Name des XML-Elements			XML-Datentyp	Beschreibung
		minute	unsignedByte	Das Attribut muss angegeben werden. Wenn das Attribut leer übergeben wird, wird ein monatlicher Abrechnungszeitraum gesetzt. Die Datentypen ‚datetime_var' und ‚datetime_intervall' können alternativ zueinander verwendet werden. Bei Verwendung des Datentyps ‚datetime_var' werden die Zeitangaben, die nicht gesetzt werden als Wildcard angenommen. Wird beispielsweise nur das Attribut ‚minute' mit dem Wert ‚0' und das Attribut ‚second' mit dem Wert ‚0' angegeben, so endet der Abrechnungszeitraum an jedem Tag zu jeder vollen Stunde zur 0. Minute und zur 0. Sekunde. Der Wert des Attributs ‚day_of_month' muss zwischen ‚1' und ‚31', des Attributs ‚hour' zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen. Bei Verwendung des Datentyps ‚datetime_intervall' wird ein Startzeitpunkt angegeben und davon ausgehend eine Periode, die zyklisch das Ende des Abrechnungszeitraums angibt.
		second	unsignedByte	
	datetime_intervall	*start_time*	dateTime	
		period	unsignedInt	
capture_period			unsignedInt	Aktualisiert die Messwertliste, aktualisiert den Eingangswert des Zählers und die abgeleiteten Register. (Registrierperiode). Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen, in Sekunden angegeben werden und zwischen ‚1' und ‚86400' liegen.

Name des XML-Elements				XML-Datentyp	Beschreibung
capture_time		*hour*		unsignedByte	Uhrzeit zu der der abrechnungstechnische Kalendertag beginnt. Zu diesem Zeitpunkt wird der Wert für eine Spontanauslesung gemäß TAF-6 aufgezeichnet. Das Attribut muss angegeben werden. Der Wert des Attributs ‚hour' muss zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen.
		minute		unsignedByte	
		second		unsignedByte	
consumer_reference		*logical_name*		string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *kaf_han_wan_container*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318011.LogicalDevice-Letztverbraucher. sm", das Attribut ‚class_id' dem Wert ‚32795' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
		class_id		unsignedShort	
		class_version		unsignedByte	
delivery_references	*delivery_reference*	*comm_profile_reference*	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *kaf_han_wan_container*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318011.LogicalDevice-EMT.sm", das Attribut ‚class_id' dem Wert ‚32795' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
			class_id	unsignedShort	
			class_version	unsignedByte	

Name des XML-Elements					XML-Datentyp	Beschreibung
		delivery_time	*datetime_var*	*year*	unsignedShort	Uhrzeit des Messwert-Versands. Das Attribut muss angegeben werden. Die Datentypen ‚datetime_var' und ‚datetime_intervall' können alternativ zueinander verwendet werden. Bei Verwendung des Datentyps ‚datetime_var' muss der Wert des Attributs ‚year' zwischen ‚2000' und ‚2199', des Attributs ‚month' zwischen ‚1' und ‚12', des Attributs ‚day_of_month' zwischen ‚1' und ‚31', des Attributs ‚hour' zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen. Der Wert des Attributs ‚day_of_week' muss entweder auf ‚Mon' oder ‚Tue' oder ‚Wed' oder ‚Thu' oder ‚Fri' oder ‚Sat' oder ‚Sun' gesetzt werden. Die Zeitangaben, die nicht gesetzt werden, werden als Wildcard angenommen. Wird beispielsweise nur das Attribut ‚minute' mit dem Wert ‚0' und das Attribut ‚second' mit dem Wert ‚0' angegeben, erfolgt der Messwertversand an jedem Tag zu jeder vollen Stunde zur 0. Minute und zur 0. Sekunde. Bei Verwendung des Datentyps ‚datetime_intervall' wird ein Startzeitpunkt angegeben und davon ausgehend eine Periode, die zyklisch den Versandzeitpunkt angibt.
				month	string	
				day_of_month	string	
				day_of_week	string	
				hour	unsignedByte	
				minute	unsignedByte	
				second	unsignedByte	

Name des XML-Elements					XML-Datentyp	Beschreibung
			datetime_intervall	*start_time*	dateTime	
				period	unsignedInt	
		qos_profile_reference	*logical_name*		string	Objekt-Referenz (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version) der COSEM-IC *qos_sq_setup*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name‘ muss dem Schema ‚01005e318013.LogicalDevice-EMT.sm“, das Attribut ‚class_id‘ dem Wert ‚32799‘ und das Attribut ‚class_version‘ dem Wert ‚0‘ entsprechen.
			class_id		unsignedShort	
			class_version		unsignedByte	
		data_references	*data_reference*	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version) der COSEM-IC *profile_generic*. Die Werte, die in diesem Attribut angegeben sind, werden an den in ‚comm_profile_reference‘ festgelegten EMT versendet. Das Attribut muss angegeben werden.
				class_id	unsignedShort	

Name des XML-Elements					XML-Datentyp		Beschreibung
				class_version	unsignedByte		Der Wert des Attributs ‚logical_name' muss dem Schema ‚OBIS.LogicalDevice-Zähler.sm' oder dem Schema ‚OBIS.LogicalDevice-TAF.sm', das Attribut ‚class_id' dem Wert ‚4' oder dem Wert ‚32822' und das Attribut ‚class_version' dem Wert ‚0' entsprechen. Der Wert des Attributs ‚attr_reference' muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.
				attr_references	attr_reference	unsignedByte	
validity_window		*start_time*			dateTime		Zeitangabe des Gültigkeitszeitraums. Das Element *start_time* muss immer und ohne die Verwendung von Wildcards angegeben werden. Das Element *end_time* muss entweder ohne Wildcards angegeben werden oder vollständig auf leer gesetzt werden. Ist *end_time* auf leer gesetzt, wird der TAF nicht eigenständig durch das SMGW beendet. Der Wert des Elements *end_time* muss vor dem 01.01.2038 liegen. Das Attribut muss angegeben werden.
		end_time			dateTime		

Name des XML-Elements		XML-Datentyp	Beschreibung
			Der Wert muss dem Schema „yyyy-mm-ddThh:mm:ssZ“ (mit 4-stelliger Abgabe des Jahres yyyy , 2-stelliger Angabe der Minuten mm, 2-stelliger Angabe der Stunden hh und 2-stelliger Angabe der Sekunden ss) entsprechen. Der Wert ist als UTC Zeit mit einem ‚Z‘ am Ende anzugeben.
metering_point_id		string	Zählpunktbezeichnung Das Attribut kann angegeben werden.
auto_cleanup		Boolean	Legt fest, ob ein TAF und die damit verbundenen Register und Messwerte automatisch entsprechend der folgenden Parameter gelöscht werden. Das Attribut kann angegeben werden. Der Wert muss entweder auf ‚false‘ oder ‚true‘ gesetzt werden. Es wird im SMGW nicht ausgewertet.
archive_duration		unsignedInt	Legt fest, wie lange (in Tagen) für diesen TAF relevante Messwerte vorgehalten werden müssen. Nach dem Erreichen dieser Dauer werden die für diesen TAF relevanten Messwerte automatisch durch das SMGW gelöscht. Wenn weniger als 15 Monate (458 Tage) angegeben sind, werden die TAF relevanten Daten erst nach 458 Tagen gelöscht. Das Attribut muss angegeben werden.

Name des XML-Elements		XML-Datentyp	Beschreibung
			Der Wert muss den Vorgaben des Datentyps entsprechen.
taf_state		string	(read-only) Ein TAF durchläuft grundsätzlich drei Zustände *ready*, *running* und *archive*. Ist ein TAF im Zustand ‚archive' oder im Zustand ‚running', werden einmal täglich alle den Input-Registern zugeordneten originären Messwerte, die „archive_duration" Tage (aber mindestens 458 Tage) alt oder älter sind, vom SMGW eigenständig entfernt. Ein TAF kann im Zustand "*ready*" durch den GWA rückstandsfrei entfernt werden. Ein TAF kann zusätzlich im Zustand "*archive*" durch den GWA rückstandsfrei entfernt werden, wenn er mindestens "*archive_duration*" Tage im SMGW vorgehalten worden ist. Alle von einem TAF benötigten Laufzeit-Objekte (also alle Objekte, die nicht in der Domain des EMT angeordnet sind und nicht von anderen Objekten des SMGW referenziert werden) werden mit Wechsel in den Zustand "*Running*" instanziiert und mit Wechsel in den Zustand "*archive*" entfernt.
on_demand_delivery_ref	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der für diesen Anwendungsfall festgelegten Instanz einer COSEM-IC *data*. Diese Referenz verweist auf ein Objekt des Typs *on_demand_delivery*, dessen Attribute '*value'* einen Inhalt mit Datentyp T*YPE_Delivery_Reference* enthält. Dieses Attribut wird dazu benutzt, eine Spontanablesung nach wie folgt zu ermöglichen:
	class_id	unsignedShort	
	class_version	unsignedByte	

Name des XML-Elements				XML-Datentyp	Beschreibung
					- Solange das Attribut ‚*value*‘ auf ‚null‘ gesetzt ist, liegt keine Anforderung zu einer Spontan-Ablesung vor. - Sobald das Attribut ‚*value*‘ mit einem Wert vom ‚*TYP_Delivery_Reference*‘ gefüllt wird, soll die Spontan-Ablesung gemäß der Informationen, die ‚*TYP_Delivery_Reference*‘ liefern, ausgeführt werden. Wurde die Spontan-Ablesung ausgeführt, wird der Inhalt wieder zu ‚null‘ gesetzt. Das Attribut kann angegeben werden. Der Wert des Attributs ‚logical_name‘ muss dem Schema ‚01005e31803b.LogicalDevice-TAF. sm“, das Attribut ‚class_id‘ dem Wert ‚32820‘ und das Attribut ‚class_version‘ dem Wert ‚0‘ entsprechen.
sensor_signature				Boolean	Der Wert des Attributs muss entweder auf ‚false‘ oder ‚true‘ gesetzt werden. Er wird im SMGW nicht ausgewertet.
season_profiles	*season_profile*	*season_profile_name*		string	Das Attribut definiert einen Zeitraum für den ein bestimmtes Schaltprogramm gültig ist. Eine *Season* endet mit dem Start der nächsten Season. Bei den Zeitangaben kann mit Wildcards gearbeitet werden, wenn der beispielsweise der Wert des Attributs ‚month‘ auf ‚4‘ und der des Attributs ‚day_of_month‘ auf ‚1‘ gesetzt ist und ‚year‘ nicht angegeben wurde, dann startet die Season jedes Jahr am 01.04.
		season_start	year	unsignedShort	
			month	string	

Name des XML-Elements				XML-Datentyp	Beschreibung
			day_of_month	string	Das Attribut muss angegeben werden. Der Wert des Attributs ‚season_profile_name' muss den Vorgaben des Datentyps entsprechen und hat eine maximale Länge von 255 Zeichen. Der Wert des Attributs ‚year' muss zwischen ‚2000' und ‚2199', des Attributs ‚month' zwischen ‚1' und ‚12' und des Attributs ‚day_of_month' zwischen ‚1' und ‚31' liegen. Der Wert des Attributs ‚week_profile_name' muss den Vorgaben des Datentyps entsprechen und hat eine maximale Länge von 255 Zeichen. Darüber hinaus muss ein entsprechender Eintrag im Attribut ‚week_profiles' vorhanden sein.
		week_profile_name		string	
week_profiles	*week_profile*	*week_profile_name*		string	Das Attribut definiert, für welchen Wochentag welche Schaltzeitpunkte relevant sind. Das Attribut muss angegeben werden. Der Wert des Attributs ‚week_profile_name' muss den Vorgaben des Datentyps entsprechen und hat eine maximale Länge von 255 Zeichen. Über diesen Wert wird das entsprechend ‚week_profile' aus den ‚season_profiles' referenziert. Die Werte der Attribute ‚monday', ‚tuesday', ‚wednesday', ‚thursday', ‚friday', ‚saturday', und ‚sunday' müssen den Vorgaben des Datentyps entsprechen. Sie referenzieren eine bestimmte ‚day_id' im Attribut ‚day_profiles'.
		monday		UnsignedInt	
		tuesday		UnsignedInt	
		wednesday		UnsignedInt	
		thursday		UnsignedInt	

Name des XML-Elements						XML-Datentyp	Beschreibung
		friday				UnsignedInt	
		saturday				UnsignedInt	
		sunday				UnsignedInt	
day_profiles	*day_profile*	*day_id*				UnsignedInt	Angabe der Schaltzeiten an einem Tag. Das Attribut muss angegeben werden. Der Wert des Attributs ‚day_id' muss den Vorgaben des Datentyps entsprechen. Über die ‚day_id' werden Schaltzeiten aus den ‚week_profiles' referenziert. Der Wert des Attributs ‚hour' muss zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen. Beim Attribut ‚start_time' kann mit Wildcards gearbeitet werden, so dass beispielsweise in jeder Stunde in eine bestimmte Tarifstufe geschalten wird. Der Wert des Attributs ‚tariff_number' muss zwischen ‚1' und ‚62' liegen und darf nicht größer sein, als der Wert, der in ‚number_of_tariffs' hinterlegt wurde.
		day_time_profiles	*day_profile*	*start_time*	*hour*	Unsigned Byte	
					minute	Unsigned Byte	
					second	Unsigned Byte	
				tariff_number		UnsignedInt	

Name des XML-Elements				XML-Datentyp	Beschreibung
special_days_entries	*special_days_entry*	*special_day_date*	*month*	string	Gibt Tage an, an denen spezielle Schaltzeiten gelten. Das Attribut muss angegeben werden. Der Wert des Attributs ‚month' muss zwischen ‚1' und ‚12' und des Attributs ‚day_of_month' zwischen ‚1' und ‚31' liegen. Der Wert des Attributs ‚day_id' muss den Vorgaben des Datentyps entsprechen. Über die ‚day_id' wird ein bestimmter Tag aus den ‚day_profiles' referenziert. In dem ‚day_profile' werden die Schaltzeiten für den Tag festgelegt.
			day_of_month	string	
		day_id		UnsignedInt	

Beim *taf02* sind ist die folgende OBIS-Kennzahl zu verwenden:

- *taf02:* *01005E318022*

Die Instanzen der Klasse sind der Logical Device ID *ldevid* des TAFs zugeordnet. Bei mehreren taf02 erfolgt die Unterscheidung also durch die Logical Device ID und nicht durch die OBIS-Kennzahl. Der GWA kann TAF-Profile im SMGW anlegen. Profile werden mittels des HTTP-Verbs POST auf dem SMGW angelegt.

Vorbedingung ist, dass das TAF-Profil, das angelegt werden soll, zuvor nicht bereits angelegt wurde. Weiterhin müssen folgende Referenzen vor dem Einspielen eines TAFs auf dem Gateway vorhanden sein:

- *input_reference*
- *consumer_reference*
- *com_profile_reference*
- *qos_profile_reference*

Die entsprechenden Profile müssen durch den GWA also vor dem Einspielen eines TAFs ins SMGW eingebracht worden sein. Das folgende Beispiel verdeutlicht das Einbringen eines TAF-Profils mit dem HTTP-Verb POST:

Tabelle 42: Anlegen eines *taf02*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Anlegen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Anlegen einer neuen Instanz	POST	<PoC>/cosem/ldevs/	taf02	201	Leer

Der GWA kann TAF-Profile aus dem SMGW auslesen. Profile werden mittels des HTTP-Verbs GET ausgelesen. Vorbedingung ist, dass das TAF-Profil, das ausgelesen werden soll, zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 43: Auslesen eines *taf02*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Auslesen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Auslesen einer bestehenden Instanz	GET	<PoC>/cosem/ldevs/taf0212345679.sm/objects/01005E318022	---	200	taf02

Der GWA kann TAF-Profile aus dem SMGW löschen. Profile werden mittels des HTTP-Verbs DELETE gelöscht. Vorbedingung ist, dass das TAF-Profil im Zustand „*archive*“ oder im Zustand „*ready*“ im SMGW vorhanden ist. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 44: Löschen eines *taf02*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Löschen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Löschen einer bestehenden Instanz	DELETE	<PoC>/cosem/ldevs/taf0212345679.sm/	---	200	---

Der GWA kann TAF-Profile im SMGW aktualisieren. Profile werden mittels des HTTP-Verbs UPDATE aktualisiert. Die Aktualisierung erfolgt stets auf den gesamten Container. Der GWA kann im TAF-Profil nur das Attribut „*end_time*“ anpassen und den entsprechenden TAF so beenden. Vorbedingung ist, dass das TAF-Profil, das aktualisiert werden soll, zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 45: Aktualisieren eines *taf02*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Aktualisieren eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Aktualisieren einer bestehenden Instanz	PUT	<PoC>/cosem/ldevs/taf0212345679.sm/objects/01005E318022	*taf02* mit angepasster *end_time*	200	---

IC *taf07*

Instanzen der Klasse *taf07* werden vom GWA gemäß WKS1 zur Konfiguration des Tarifanwendungsfalls TAF7 im SMGW benutzt. Diese Klasse kennt keine Methoden. Die Attribute dieser Klasse sind wie folgt:

Tabelle 46: Attribute der IC *taf07*

Name des XML-Elements	XML-Datentyp	Beschreibung
logical_name	string	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
taf_name	string	Freier Bezeichner des Auswertungsprofils gemäß [BSI TR-03109-1]. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
taf_identifier	string	Für das SMGW eindeutiger Bezeichner für das Auswertungsprofil. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
pseudonym	string	(optional) Mit diesem Attribut kann ein Pseudonym angegeben werden. Bei der Übertragung der Messwerte werden sämtlich Hinweise auf den Zähler (Logical Device ID des Zählers) durch das Pseudonym ersetzt. Somit ist ein

Name des XML-Elements			XML-Datentyp	Beschreibung
				Rückschluss auf den konkreten Zähler oder das konkrete Smart Meter Gateway aus dem die Messwerte stammen nicht mehr möglich.
input_references	*input_reference*	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version) des Typs *gateway_signed_extended_register*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name‘ muss auf den Wert des Attributs ‚value‘ einer auf dem SMGW vorhandenen Instanz der IC ‚*e_meter_sensor_setup*‘ oder der IC ‚*generic_meter_sensor_setup*‘ und somit dem Schema ‚value.LogicalDevice-Zähler.sm‘ entsprechen. Beispielsweise kann der Wert auf , 0100010800ff.1emh0005288229.sm‘ gesetzt werden. Der Wert des Attributs ‚class_id‘ muss auf ‚3‘ gesetzt werden. Der Wert des Attributs ‚class_version‘ muss auf ‚0‘ gesetzt werden.
		class_id	unsignedShort	
		class_version	unsignedByte	
number_of_tariffs			byte	Anzahl der Tarifstufen Das Attribut muss angegeben werden. Der Wert wird vom SMGW nicht ausgewertet.
active_tariff			byte	(read-only) Dieses Attribut spiegelt den jeweils aktuell in dem TAF geltenden Tarif wieder.
end_of_billing_period	*datetime_var*	*day_of_month*	hexBinary	Schreibt die abgeleiteten Register in die turnusgemäße Abrechnungsliste, entspricht dem Abrechnungszeitraum aus [BSI TR-03109] und dem Turnus aus den dortigen Abbildungen.

Name des XML-Elements			XML-Datentyp	Beschreibung
		hour	unsignedByte	Das Attribut muss angegeben werden. Wenn das Attribut leer übergeben wird, wird ein stündlicher Abrechnungszeitraum gesetzt. Die Datentypen ‚datetime_var' und ‚datetime_intervall' können alternativ zueinander verwendet werden. Bei Verwendung des Datentyps ‚datetime_var' werden die Zeitangaben, die nicht gesetzt werden als Wildcard angenommen. Wird beispielsweise nur das Attribut ‚minute' mit dem Wert ‚0' und das Attribut ‚second' mit dem Wert ‚0' angegeben, so endet der Abrechnungszeitraum an jedem Tag zu jeder volle Stunde zur 0. Minute und zur 0. Sekunde. Der Wert des Attributs ‚day_of_month' muss zwischen ‚1' und ‚31', des Attributs ‚hour' zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen. Bei Verwendung des Datentyps ‚datetime_intervall' wird ein Startzeitpunkt angegeben und davon ausgehend eine Periode, die zyklisch das Ende des Abrechnungszeitraums angibt.
		minute	unsignedByte	
		second	unsignedByte	
	datetime_intervall	*start_time*	dateTime	
		period	unsignedInt	
capture_period			unsignedInt	Aktualisiert die Messwertliste, aktualisiert den Eingangswert des Zählers und die abgeleiteten Register (Registrierperiode). Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen, in Sekunden angegeben werden und zwischen ‚1' und ‚86400' liegen.
capture_time	*hour*		unsignedByte	Uhrzeit zu der der abrechnungstechnische Kalendertag beginnt.

Name des XML-Elements					XML-Datentyp	Beschreibung
			minute		unsignedByte	Zu diesem Zeitpunkt wird der Wert für eine Spontanauslesung gemäß TAF-6 aufgezeichnet. Das Attribut muss angegeben werden. Der Wert des Attributs ‚hour' muss zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen.
			second		unsignedByte	
consumer_reference			*logical_name*		string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *kaf_han_wan_container*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318011.LogicalDevice-Letztverbraucher.sm", das Attribut ‚class_id' dem Wert ‚32795' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
			class_id		unsignedShort	
			class_version		unsignedByte	
delivery_references	*delivery_reference*	*comm_profile_reference*	*logical_name*		string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *kaf_han_wan_container*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318011.LogicalDevice-EMT.sm", das Attribut ‚class_id' dem Wert ‚32795' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
			class_id		unsignedShort	
			class_version		unsignedByte	
		delivery_time	*datetime_var*	*year*	unsignedShort	Uhrzeit des Messwert-Versands. Das Attribut muss angegeben werden. Die Datentypen ‚datetime_var' und ‚datetime_intervall' können alternativ zueinander verwendet werden.
				month	string	

Name des XML-Elements					XML-Datentyp	Beschreibung
				day_of_month	string	Bei Verwendung des Datentyps ‚datetime_var' muss der Wert des Attributs ‚year' zwischen ‚2000' und ‚2199', des Attributs ‚month' zwischen ‚1' und ‚12', des Attributs ‚day_of_month' zwischen ‚1' und ‚31', des Attributs ‚hour' zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen. Der Wert des Attributs ‚day_of_week' muss entweder auf ‚Mon' oder ‚Tue' oder ‚Wed' oder ‚Thu' oder ‚Fri' oder ‚Sat' oder ‚Sun' gesetzt werden. Die Zeitangaben, die nicht gesetzt werden, werden als Wildcard angenommen. Wird beispielsweise nur das Attribut ‚minute' mit dem Wert ‚0' und das Attribut ‚second' mit dem Wert ‚0' angegeben, erfolgt der Messwertversand an jedem Tag zu jeder vollen Stunde zur 0. Minute und zur 0. Sekunde. Bei Verwendung des Datentyps ‚datetime_intervall' wird ein Startzeitpunkt angegeben und davon ausgehend eine Periode, die zyklisch den Versandzeitpunkt angibt.
				day_of_week	string	
				hour	unsignedByte	
				minute	unsignedByte	
				second	unsignedByte	
			datetime_intervall	*start_time*	dateTime	
				period	unsignedInt	

Name des XML-Elements							XML-Datentyp	Beschreibung
		qos_profile_reference	*logical_name*				string	Objekt-Referenz (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version) der COSEM-IC *qos_sq_setup*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name‘ muss dem Schema ‚01005e318013.LogicalDevice-EMT.sm“, das Attribut ‚class_id‘ dem Wert ‚32799‘ und das Attribut ‚class_version‘ dem Wert ‚0‘ entsprechen.
			class_id				unsignedShort	
			class_version				unsignedByte	
		data_references	*data_reference*	*logical_name*			string	Objekt-Referenz (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version) des profile_generic. Die Werte, die in diesem Attribut angegeben sind, werden an den in ‚comm_profile_reference‘ festgelegten EMT versendet. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name‘ muss dem Schema , OBIS.LogicalDevice-Zähler.sm‘ oder dem Schema , OBIS.LogicalDevice-TAF.sm‘, das Attribut ‚class_id‘ dem Wert ‚4‘ oder dem Wert ‚32822‘ und das Attribut ‚class_version‘ dem Wert ‚0‘ entsprechen. Der Wert des Attributs ‚attr_reference‘ muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.
				class_id			unsignedShort	
				class_version			unsignedByte	
				attr_references	attr_reference	unsignedByte		
validity_window			*start_time*				dateTime	Zeitangabe des Gültigkeitszeitraums. Das Element *start_time* muss immer und ohne die Verwendung von Wildcards angegeben werden. Das Element *end_time* muss entweder ohne Wildcards angegeben werden oder vollständig
			end_time					

Name des XML-Elements		XML-Datentyp	Beschreibung
			auf leer gesetzt werden. Ist *end_time* auf leer gesetzt, wird der TAF nicht eigenständig durch das SMGW beendet. Der Wert des Elements *end_time* muss vor dem 01.01.2038 liegen. Das Attribut muss angegeben werden. Der Wert muss dem Schema „yyyy-mm-ddThh:mm:ssZ“ (mit 4-stelliger Abgabe des Jahres yyyy , 2-stelliger Angabe der Minuten mm, 2-stelliger Angabe der Stunden hh und 2-stelliger Angabe der Sekunden ss) entsprechen. Der Wert ist als UTC Zeit mit einem ‚Z‘ am Ende anzugeben.
metering_point_id		string	Zählpunktbezeichnung. Das Attribut kann angegeben werden.
auto_cleanup		Boolean	Legt fest, ob ein TAF und die damit verbundenen Register und Messwerte automatisch entsprechend der folgenden Parameter gelöscht werden. Das Attribut kann angegeben werden. Der Wert muss entweder auf ‚false‘ oder ‚true‘ gesetzt werden. Es wird im SMGW nicht ausgewertet.
archive_duration		unsignedInt	Legt fest, wie lange (in Tagen) für diesen TAF relevante Messwerte vorgehalten werden müssen. Nach dem Erreichen dieser Dauer werden die für diesen TAF relevanten Messwerte automatisch durch das SMGW gelöscht.

Name des XML-Elements		XML-Datentyp	Beschreibung
			Wenn weniger als 15 Monate (458 Tage) angegeben sind, werden die TAF relevanten Daten erst nach 458 Tagen gelöscht. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
taf_state		string	(read-only) Ein TAF durchläuft grundsätzlich drei Zustände *ready*, *running* und *archive*. Ist ein TAF im Zustand ‚archive' oder im Zustand ‚running', werden einmal täglich alle den Input-Registern zugeordneten originären Messwerte, die „archive_duration" Tage (aber mindestens 458 Tage) alt oder älter sind, vom SMGW eigenständig entfernt. Ein TAF kann im Zustand "*ready*" durch den GWA rückstandsfrei entfernt werden. Ein TAF kann zusätzlich im Zustand "*archive*" durch den GWA rückstandsfrei entfernt werden, wenn er mindestens "*archive_duration*" Tage im SMGW vorgehalten worden ist. Alle von einem TAF benötigten Laufzeit-Objekte (also alle Objekte, die nicht in der Domain des EMT angeordnet sind und nicht von anderen Objekten des SMGW referenziert werden) werden mit Wechsel in den Zustand "*Running*" instanziiert und mit Wechsel in den Zustand "*archive*" entfernt.
on_demand_delivery_ref	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) des für diesen Anwendungsfall festgelegten Instanz einer Interface-Klasse vom Typ IC *on_demand_delivery*, dessen Attribute „*delivery_reference"* als Unterelement von *value* dazu benutzt, eine Spontanablesung wie folgt zu ermöglichen:
	class_id	unsignedShort	
	class_version	unsignedByte	

Name des XML-Elements		XML-Datentyp	Beschreibung
			- Solange ‚*value*' zu ‚*null*' gesetzt ist, liegt keine Anforderung zu einer Spontan-Ablesung vor. - Sobald ‚*value*' mit Daten ungleich ‚*null*' gefüllt ist, wird die Spontan-Ablesung entsprechend der angegebenen Daten ausgeführt werden. Wurde die Spontan-Ablesung ausgeführt, wird der Inhalt wieder zu ‚*null*' gesetzt. Das Attribut kann angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e31803b.LogicalDevice-TAF. sm", das Attribut ‚class_id' dem Wert , 32820' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
sensor_signature		Boolean	Der Wert des Attributs muss entweder auf ‚false' oder ‚true' gesetzt werden. Er wird im SMGW nicht ausgewertet.

Beim *taf07* sind ist die folgende OBIS-Kennzahl zu verwenden:

- *taf07:* *01005E318027*

Die Instanzen der Klasse sind der Logical Device ID ldevid des TAFs zugeordnet. Bei mehreren *taf07* erfolgt die Unterscheidung also durch die Logical Device ID und nicht durch die OBIS Kennzahl. Der GWA kann TAF-Profile im SMGW anlegen. Profile werden mittels des HTTP-Verbs POST auf dem SMGW angelegt.

Vorbedingung ist, dass das TAF-Profil, das angelegt werden soll, zuvor nicht bereits angelegt wurde. Weiterhin müssen folgende Referenzen vor dem Einspielen eines TAFs auf dem Gateway vorhanden sein:

- *Input_reference*
- *consumer_reference*
- *com_profile_reference*
- *qos_profile_reference*

Die entsprechenden Profile müssen durch den GWA also vor dem Einspielen eines TAFs ins SMGW eingebracht worden sein. Das folgende Beispiel verdeutlicht das Einbringen eines TAF-Profils mit dem HTTP-Verb POST:

Tabelle 47: Anlegen eines *taf07*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Anlegen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Anlegen einer neuen Instanz	POST	<PoC>/cosem/ldevs/	taf07	201	Leer

Der GWA kann TAF-Profile aus dem SMGW auslesen. Profile werden mittels des HTTP-Verbs GET ausgelesen. Vorbedingung ist, dass das TAF-Profil, das ausgelesen werden soll, zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 48: Auslesen eines *taf07*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Auslesen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Auslesen einer bestehenden Instanz	GET	<PoC>/cosem/ldevs/taf0712345679.sm/objects/01005E318022	---	200	taf07

Der GWA kann TAF-Profile aus dem SMGW löschen. Profile werden mittels des HTTP-Verbs DELETE gelöscht. Vorbedingung ist, dass das TAF-Profil im Zustand „*archive*“ oder im Zustand „*ready*“ im SMGW vorhanden ist. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 49: Löschen eines *taf07*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Löschen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Löschen einer bestehenden Instanz	DELETE	<PoC>/cosem/ldevs/taf0712345679.sm/	---	200	---

Der GWA kann TAF-Profile im SMGW aktualisieren. Profile werden mittels des HTTP-Verbs UPDATE aktualisiert. Die Aktualisierung erfolgt stets auf den gesamten Container. Der GWA kann im TAF-Profil nur das Attribut „*end_time*“ anpassen und den entsprechenden TAF so beenden. Vorbedingung ist, dass das TAF-Profil, das aktualisiert werden soll, zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 50: Aktualisieren eines *taf07*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Aktualisieren eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Aktualisieren einer bestehenden Instanz	PUT	<PoC>/cosem/ldevs/taf0712345679.sm/objects/01005E318022	*taf07* mit angepasster *end_time*	200	---

IC *taf09*

Instanzen der Klasse *taf09* werden vom GWA zur Konfiguration des Tarifanwendungsfalls TAF9 im SMGW benutzt. Die Klasse kennt folgende Parameter:

Tabelle 51: Attribute der IC *taf09*

Name des XML-Elements			XML-Datentyp	Beschreibung
logical_name			string	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
taf_name			string	Freier Bezeichner des Auswertungsprofils gemäß [BSI TR-03109-1]. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
taf_identifier			string	Für das SMGW eindeutiger Bezeichner für das Auswertungsprofil. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
input_references	*input_reference*	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) des Typs *gateway_signed_extended_register*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss auf den Wert des Attributs ‚value' einer auf dem SMGW vorhandenen Instanz der IC
		class_id	unsignedShort	
		class_version	unsignedByte	

Name des XML-Elements			XML-Datentyp	Beschreibung
				‚*e_meter_sensor_setup*' oder der IC ‚*generic_meter_sensor_setup*' und somit dem Schema ‚value.LogicalDevice-Zähler.sm' entsprechen. Der Wert muss den zugelassenen Messgrößen der [BSI TR-03109-1] für diesen TAF entsprechen (bspw. 0100100700ff.1emh0005288229.sm). Der Wert des Attributs ‚class_id' muss auf ‚3' gesetzt werden. Der Wert des Attributs ‚class_version' muss auf ‚0' gesetzt werden.
number_of_tariffs			byte	Anzahl der Tarifstufen. Das Attribut muss angegeben werden. Bei diesem TAF muss das Attribut immer mit ‚1' belegt werden.
active_tariff			byte	(read-only) Dieses Attribut spiegelt den jeweils aktuell in dem TAF geltenden Tarif wider.
end_of_billing_period	*datetime_var*	*day_of_month*	string	Schreibt die abgeleiteten Register in die turnusgemäße Abrechnungsliste, entspricht dem Abrechnungszeitraum aus [BSI TR-03109] und dem Turnus aus den dortigen Abbildungen. Das Attribut muss angegeben werden. Die Datentypen ‚datetime_var' und ‚datetime_intervall' können alternativ zueinander verwendet werden.
		hour	unsignedByte	
		minute	unsignedByte	
		second	unsignedByte	

Name des XML-Elements			XML-Datentyp	Beschreibung
	datetime_intervall	*start_time*	dateTime	Bei Verwendung des Datentyps ‚datetime_var' werden die Zeitangaben, die nicht gesetzt werden als Wildcard angenommen. Wird beispielsweise nur das Attribut ‚minute' mit dem Wert ‚0' und das Attribut ‚second' mit dem Wert ‚0' angegeben, so endet der Abrechnungszeitraum an jedem Tag zu jeder vollen Stunde zur 0. Minute und zur 0. Sekunde. Der Wert des Attributs ‚day_of_month' muss zwischen ‚1' und ‚31', des Attributs ‚hour' zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen. Bei Verwendung des Datentyps ‚datetime_intervall' wird ein Startzeitpunkt angegeben und davon ausgehend eine Periode, die zyklisch das Ende des Abrechnungszeitraums angibt. Bei diesem TAF legt dieser Zeitpunkt die Blockbildung für die Übertragung fest. Falls keine Blockbildung benötigt wird (Versand unmittelbar zur Messwertbildung) ist das Attribut ‚period' mit dem Wert ‚0' zu belegen.
		period	unsignedInt	
capture_period			unsignedInt	Aktualisiert die Messwertliste, aktualisiert den Eingangswert des Zählers und die abgeleiteten Register. (Registrierperiode). Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen, in Sekunden angegeben werden und zwischen ‚1' und ‚86400' liegen. Bei diesem TAF entspricht das Attribut der Abtastrate aus der [BSI TR-03109].

Name des XML-Elements				XML-Datentyp	Beschreibung
capture_time			*hour*	unsignedByte	Uhrzeit zu der der abrechnungstechnische Kalendertag beginnt. Zu diesem Zeitpunkt wird der Wert für eine Spontanauslesung gemäß TAF-6 aufgezeichnet. Das Attribut muss angegeben werden. Der Wert des Attributs ‚hour' muss zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen. Bei diesem TAF wird das Attribut nicht ausgewertet.
			minute	unsignedByte	
			second	unsignedByte	
consumer_reference			*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *kaf_han_wan_container*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318011.LogicalDevice-Letztverbraucher.sm", das Attribut ‚class_id' dem Wert ‚32795' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
			class_id	unsignedShort	
			class_version	unsignedByte	
delivery_references	*delivery_reference*	*comm_profile_reference*	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *kaf_han_wan_container*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318011.LogicalDevice-EMT. sm", das Attribut ‚class_id' dem Wert ‚32795' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
			class_id	unsignedShort	
			class_version	unsignedByte	

Name des XML-Elements					XML-Datentyp	Beschreibung
		delivery_time	*datetime_var*	*year*	unsignedShort	Uhrzeit des Messwert-Versands. Das Attribut muss angegeben werden. Die Datentypen ‚datetime_var' und ‚datetime_intervall' können alternativ zueinander verwendet werden. Bei Verwendung des Datentyps ‚datetime_var' muss der Wert des Attributs ‚year' zwischen ‚2000' und ‚2199', des Attributs ‚month' zwischen ‚1' und ‚12', des Attributs ‚day_of_month' zwischen ‚1' und ‚31', des Attributs ‚hour' zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen. Der Wert des Attributs ‚day_of_week' muss entweder auf ‚Mon' oder ‚Tue' oder ‚Wed' oder ‚Thu' oder ‚Fri' oder ‚Sat' oder ‚Sun' gesetzt werden. Die Zeitangaben, die nicht gesetzt werden, werden als Wildcard angenommen. Wird beispielsweise nur das Attribut ‚minute' mit dem Wert ‚0' und das Attribut ‚second' mit dem Wert ‚0' angegeben, erfolgt der Messwertversand an jedem Tag zu jeder vollen Stunde zur 0. Minute und zur 0. Sekunde. Bei Verwendung des Datentyps ‚datetime_intervall' wird ein Startzeitpunkt angegeben und davon ausgehend eine Periode, die zyklisch den Versandzeitpunkt angibt.
				month	string	
				day_of_month	string	
				day_of_week	string	
				hour	unsignedByte	
				minute	unsignedByte	
				second	unsignedByte	
			datetime_intervall	*start_time*	dateTime	
				period	unsignedInt	

Name des XML-Elements							XML-Datentyp		Beschreibung
									Falls Messwerte unmittelbar nach der Messwertbildung oder unmittelbar nach Ende der Blockbildung versendet werden sollen, ist das Attribut ‚period' mit dem Wert ‚0' zu belegen.
		qos_profile_reference	*logical_name*				string		Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *qos_sq_setup*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318013.LogicalDevice-EMT.sm", das Attribut ‚class_id' dem Wert ‚32799' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
			class_id				unsignedShort		
			class_version				unsignedByte		
		data_references	*data_reference*	*logical_name*			string		Objekt-Referenz (in Form des „*Logical Device Name*", der Klassen-ID und Klassen-Version) der COSEM-IC *profile_generic*. Die Werte, die in diesem Attribut angegeben sind, werden an den in ‚*comm_profile_reference*' festgelegten EMT versendet. Das Attribut muss angegeben werden. Der Wert des Attributs ‚*logical_name*' muss dem Schema ‚*OBIS.LogicalDevice-Zähler.sm*' oder dem Schema ‚*OBIS.LogicalDevice-TAF.sm*', das Attribut ‚*class_id*' dem Wert ‚*4*' oder dem Wert ‚*32822*' und das Attribut ‚*class_version*' dem Wert ‚*0*' entsprechen. Der Wert des Attributs ‚attr_reference' muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.
				class_id			unsignedShort		
				class_version			unsignedByte		
				attr_references	attr_reference	unsignedByte			

Name des XML-Elements		XML-Datentyp	Beschreibung
validity_window	*start_time* *end_time*	dateTime	Zeitangabe des Gültigkeitszeitraums. Das Element *start_time* muss immer und ohne die Verwendung von Wildcards angegeben werden. Das Element *end_time* muss entweder ohne Wildcards angegeben werden oder vollständig auf leer gesetzt werden. Ist *end_time* auf leer gesetzt, wird der TAF nicht eigenständig durch das SMGW beendet. Der Wert des Elements *end_time* muss vor dem 01.01.2038 liegen. Das Attribut muss angegeben werden. Der Wert muss dem Schema „yyyy-mm-ddThh:mm:ssZ“ (mit 4-stelliger Abgabe des Jahres *yyyy*, 2-stelliger Angabe der Minuten *mm*, 2-stelliger Angabe der Stunden *hh* und 2-stelliger Angabe der Sekunden *ss*) entsprechen. Der Wert ist als UTC Zeit mit einem ‚Z‘ am Ende anzugeben.
metering_point_id		string	Zählpunktbezeichnung. Das Attribut kann angegeben werden.
auto_cleanup		Boolean	Legt fest, ob ein TAF und die damit verbundenen Register und Messwerte automatisch entsprechend der folgenden Parameter gelöscht werden. Das Attribut kann angegeben werden. Der Wert muss entweder auf ‚false‘ oder ‚true‘ gesetzt werden. Es wird im SMGW nicht ausgewertet.

Name des XML-Elements	XML-Datentyp	Beschreibung
archive_duration	unsignedInt	Legt fest, wie lange (in Tagen) für diesen TAF relevante Messwerte vorgehalten werden müssen. Nach dem Erreichen dieser Dauer werden die für diesen TAF relevanten Messwerte automatisch durch das SMGW gelöscht. Wenn weniger als 15 Monate (458 Tage) angegeben sind, werden die TAF relevanten Daten erst nach 458 Tagen gelöscht. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
taf_state	string	(read-only) Ein TAF durchläuft grundsätzlich drei Zustände *ready*, *running* und *archive*. Ist ein TAF im Zustand ‚archive' oder im Zustand ‚running', werden einmal täglich alle den Input-Registern zugeordneten originären Messwerte, die „archive_duration" Tage (aber mindestens 458 Tage) alt oder älter sind, vom SMGW eigenständig entfernt. Ein TAF kann im Zustand "*ready*" durch den GWA rückstandsfrei entfernt werden. Ein TAF kann zusätzlich im Zustand "*archive*" durch den GWA rückstandsfrei entfernt werden, wenn er mindestens "*archive_duration*" Tage im SMGW vorgehalten worden ist. Alle von einem TAF benötigten Laufzeit-Objekte (also alle Objekte, die nicht in der Domain des EMT angeordnet sind und nicht von anderen Objekten des SMGW referenziert werden) werden mit Wechsel in den

Name des XML-Elements					XML-Datentyp	Beschreibung
						Zustand "*Running*" instanziiert und mit Wechsel in den Zustand "*archive*" entfernt.
sensor_signature					Boolean	Der Wert des Attributs muss entweder auf ‚false' oder ‚true' gesetzt werden. Er wird im SMGW nicht ausgewertet.
pseudonym					string	(optional) Mit diesem Attribut kann ein Pseudonym angegeben werden. Bei der Übertragung der Messwerte werden sämtlich Hinweise auf den Zähler (Logical Device ID des Zählers) durch das Pseudonym ersetzt. Somit ist ein Rückschluss auf den konkreten Zähler oder das konkrete Smart Meter Gateway aus dem die Messwerte stammen nicht mehr möglich.
value_monitors	*monitored_obj*	*logical_name*			string	(optional) Mit diesem Attribut können zu überwachende Grenzwerte angegeben werden. Im Attribut ‚monitored_obj' wird dabei die zu überwachende Messgröße angegeben, im Attribut ‚thresholds' die Grenzwerte, die für dieses Objekt überwacht werden müssen. Falls ein Grenzwert auf Überschreitung geprüft werden soll, ist das Attribut ‚direction' mit dem Wert ‚true' zu belegen.
		class_id			unsignedShort	
		class_version			unsignedByte	
	thresholds	*threshold*	*direction*		Boolean	
			over_limit	*value*	long	

Name des XML-Elements					XML-Datentyp	Beschreibung
				scaler	Byte	Falls ein Grenzwert auf Unterschreitung geprüft werden soll, ist das Attribut ‚direction' mit dem Wert ‚false' zu belegen. Die beiden Werte in ‚over_limit' und ‚under_limit' wirken dabei als Hysterese-Band. Der Messwertversand wird bei Prüfung auf Überschreitung erst dann erneut ausgelöst, wenn der Werte in ‚under_limit' unterschritten wurde und wird bei Prüfung auf Unterschreitung erst dann erneut ausgelöst, wenn der Wert in ‚over_limit' überschritten wurde.
				unit	unsignedByte	
			under_limit	*value*	long	
				scaler	Byte	
				unit	unsignedByte	

Diese Klasse verfügt über die folgenden Methoden:

Tabelle 52: Beschreibung der Methoden der IC *taf09*

Methode	Beschreibung
onDemandReadout()	Die Methode löst einen einmaligen Versand der aktuellen Messwerte im Bedarfsfall an die EMTs, die in ‚comm_profile_reference' definiert sind aus.

Beim *taf09* ist die folgende OBIS Kennzahl zu verwenden:

- *taf09*: 01005E318029

Die Instanzen der Klasse sind der Logical Device ID ldevid des TAFs zugeordnet. Bei mehreren *taf09* erfolgt die Unterscheidung also durch die Logical Device ID und nicht durch die OBIS Kennzahl. Der GWA kann TAF-Profile im SMGW anlegen. Profile werden mittels des HTTP-Verbs POST auf dem SMGW angelegt.

Vorbedingung ist, dass das TAF-Profil, das angelegt werden soll, zuvor nicht bereits angelegt wurde. Weiterhin müssen folgende Referenzen vor dem Einspielen eines TAFs auf dem Gateway vorhanden sein:

- *Input_reference*
- *consumer_reference*
- *com_profile_reference*
- *qos_profile_reference*

Die entsprechenden Profile müssen durch den GWA also vor dem Einspielen eines TAFs ins SMGW eingebracht worden sein. Das folgende Beispiel verdeutlicht das Einbringen eines TAF-Profils mit dem HTTP-Verb POST:

Tabelle 53: Anlegen eines *taf09*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Anlegen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Anlegen einer neuen Instanz	POST	<PoC>/cosem/ldevs/	taf09	201	Leer

Der GWA kann TAF-Profile aus dem SMGW auslesen. Profile werden mittels des HTTP-Verbs GET ausgelesen. Vorbedingung ist, dass das TAF-Profil, das ausgelesen werden soll, zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 54: Auslesen eines *taf09*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Auslesen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Auslesen einer bestehenden Instanz	GET	<PoC>/cosem/ldevs/taf0912345679.sm/objects/01005E318029	---	200	taf09

Der GWA kann TAF-Profile aus dem SMGW löschen. Profile werden mittels des HTTP-Verbs DELETE gelöscht. Vorbedingung ist, dass das TAF-Profil im Zustand „*archive*“ oder im Zustand „*ready*“ im SMGW vorhanden ist. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 55: Löschen eines *taf09*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Löschen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Löschen einer bestehenden Instanz	DELETE	<PoC>/cosem/ldevs/taf0912345679.sm/	---	200	---

Der GWA kann TAF-Profile im SMGW aktualisieren. Profile werden mittels des HTTP-Verbs UPDATE aktualisiert. Die Aktualisierung erfolgt stets auf den gesamten Container. Der GWA kann im TAF-Profil nur die Attribute „*end_time*“ und „*thresholds*“ anpassen und den entsprechenden TAF so beenden. Vorbedingung ist, dass das TAF-Profil, das aktualisiert werden soll, zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 56: Aktualisieren eines *taf09*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Aktualisieren eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Aktualisieren einer bestehenden Instanz	PUT	<PoC>/cosem/ldevs/taf0912345679.sm/objects/01005E318029	*taf09* mit angepasster *end_time* und/oder angepassten *thresholds*	200	---

IC *taf10*

Instanzen der Klasse *taf10* werden vom GWA zur Konfiguration des Tarifanwendungsfalls TAF10 im SMGW benutzt. Die Klasse kennt folgende Parameter:

Tabelle 57: Attribute der IC *taf10*

Name des XML-Elements			XML-Datentyp	Beschreibung
logical_name			string	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
taf_name			string	Freier Bezeichner des Auswertungsprofils gemäß [BSI TR-03109-1]. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
taf_identifier			string	Für das SMGW eindeutiger Bezeichner für das Auswertungsprofil. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
input_references	*input_reference*	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) des Typs *gateway_signed_extended_register*. Das Attribut muss angegeben werden.
		class_id	unsignedShort	

Name des XML-Elements				XML-Datentyp	Beschreibung
		class_version		unsignedByte	Der Wert des Attributs ‚logical_name' muss auf den Wert des Attributs ‚value' einer auf dem SMGW vorhandenen Instanz der IC ‚*e_meter_sensor_setup*' oder der IC ‚*generic_meter_sensor_setup*' und somit dem Schema ‚value.LogicalDevice-Zähler.sm' entsprechen. Der Wert muss den zugelassenen Messgrößen der [BSI TR-03109-1] für diesen TAF entsprechen (bspw. 0100100700ff.1emh0005288229.sm). Der Wert des Attributs ‚class_id' muss auf ‚3' gesetzt werden. Der Wert des Attributs ‚class_version' muss auf ‚0' gesetzt werden.
number_of_tariffs				byte	Anzahl der Tarifstufen. Das Attribut muss angegeben werden. Bei diesem TAF muss das Attribut immer mit ‚1' belegt werden.
active_tariff				byte	(read-only) Dieses Attribut spiegelt den jeweils aktuell in dem TAF geltenden Tarif wider.
end_of_billing_period	*datetime_var*	*day_of_month*		string	Schreibt die abgeleiteten Register in die turnusgemäße Abrechnungsliste, entspricht dem Abrechnungszeitraum aus [BSI TR-03109] und dem Turnus aus den dortigen Abbildungen. Das Attribut muss angegeben werden.
		hour		unsignedByte	

Name des XML-Elements			XML-Datentyp	Beschreibung
		minute	unsignedByte	Die Datentypen ‚datetime_var' und ‚datetime_intervall' können alternativ zueinander verwendet werden. Bei Verwendung des Datentyps ‚datetime_var' werden die Zeitangaben, die nicht gesetzt werden als Wildcard angenommen. Wird beispielsweise nur das Attribut ‚minute' mit dem Wert ‚0' und das Attribut ‚second' mit dem Wert ‚0' angegeben, so endet der Abrechnungszeitraum an jedem Tag zu jeder vollen Stunde zur 0. Minute und zur 0. Sekunde. Der Wert des Attributs ‚day_of_month' muss zwischen ‚1' und ‚31', des Attributs ‚hour' zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen. Bei Verwendung des Datentyps ‚datetime_intervall' wird ein Startzeitpunkt angegeben und davon ausgehend eine Periode, die zyklisch das Ende des Abrechnungszeitraums angibt. Bei diesem TAF legt dieser Zeitpunkt die Blockbildung für die Übertragung fest. Falls keine Blockbildung benötigt wird (Versand unmittelbar zur Messwertbildung) ist das Attribut ‚period' mit dem Wert ‚0' zu belegen.
		second	unsignedByte	
	datetime_intervall	*start_time*	dateTime	
		period	unsignedInt	
capture_period			unsignedInt	Aktualisiert die Messwertliste, aktualisiert den Eingangswert des Zählers und die abgeleiteten Register. (Registrierperiode). Das Attribut muss angegeben werden.

Name des XML-Elements			XML-Datentyp	Beschreibung
				Der Wert muss den Vorgaben des Datentyps entsprechen, in Sekunden angegeben werden und zwischen ‚1' und ‚86400' liegen. Bei diesem TAF entspricht das Attribut der Abtastrate aus der [BSI TR-03109].
capture_time		*hour*	unsignedByte	Uhrzeit zu der der abrechnungstechnische Kalendertag beginnt. Zu diesem Zeitpunkt wird der Wert für eine Spontanauslesung gemäß TAF-6 aufgezeichnet. Das Attribut muss angegeben werden. Der Wert des Attributs ‚hour' muss zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen. Bei diesem TAF wird das Attribut nicht ausgewertet.
		minute	unsignedByte	
		second	unsignedByte	
consumer_reference		*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *kaf_han_wan_container*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318011.LogicalDevice-Letztverbraucher.sm", das Attribut ‚class_id' dem Wert ‚32795' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
		class_id	unsignedShort	
		class_version	Unsigned Byte	
		logical_name	string	

Name des XML-Elements					XML-Datentyp	Beschreibung
delivery_references	*delivery_reference*	*comm_profile_reference*	*class_id*		unsignedShort	Objekt-Referenz (in Form des „Logical Device Name“, der Klassen-ID und Klassen-Version) der COSEM-IC *kaf_han_wan_container*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name‘ muss dem Schema ‚01005e318011.LogicalDevice-EMT. sm“, das Attribut ‚class_id‘ dem Wert ‚32795‘ und das Attribut ‚class_version‘ dem Wert ‚0‘ entsprechen.
			class_version		unsignedByte	
		delivery_time	*datetime_var*	*year*	unsignedShort	Uhrzeit des Messwert-Versands. Das Attribut muss angegeben werden. Die Datentypen ‚datetime_var‘ und ‚datetime_intervall‘ können alternativ zueinander verwendet werden. Bei Verwendung des Datentyps ‚datetime_var‘ muss der Wert des Attributs ‚year‘ zwischen ‚2000‘ und ‚2199‘, des Attributs ‚month‘ zwischen ‚1‘ und ‚12‘, des Attributs ‚day_of_month‘ zwischen ‚1‘ und ‚31‘, des Attributs ‚hour‘ zwischen ‚0‘ und ‚23‘, des Attributs ‚minute‘ zwischen ‚0‘ und ‚59‘ und des Attributs ‚second‘ zwischen ‚0‘ und ‚59‘ liegen. Der Wert des Attributs ‚day_of_week‘ muss entweder auf ‚Mon‘ oder ‚Tue‘ oder ‚Wed‘ oder ‚Thu‘ oder ‚Fri‘ oder ‚Sat‘ oder ‚Sun‘ gesetzt werden.
				month	string	
				day_of_month	string	
				day_of_week	string	
				hour	unsignedByte	
				minute	unsignedByte	

Name des XML-Elements					XML-Datentyp	Beschreibung
				second	unsignedByte	Die Zeitangaben, die nicht gesetzt werden, werden als Wildcard angenommen. Wird beispielsweise nur das Attribut ‚minute' mit dem Wert ‚0' und das Attribut ‚second' mit dem Wert ‚0' angegeben, erfolgt der Messwertversand an jedem Tag zu jeder vollen Stunde zur 0. Minute und zur 0. Sekunde. Bei Verwendung des Datentyps ‚datetime_intervall' wird ein Startzeitpunkt angegeben und davon ausgehend eine Periode, die zyklisch das Ende des Abrechnungszeitraums angibt. Falls Messwerte unmittelbar nach der Messwertbildung oder unmittelbar nach Ende der Blockbildung versendet werden sollen ist das Attribut ‚period' mit dem Wert ‚0' zu belegen.
			datetime_intervall	*start_time*	dateTime	
				period	unsignedInt	
		qos_profile_reference	*logical_name*		string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *qos_sq_setup*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318013.LogicalDevice-EMT.sm", das Attribut ‚class_id' dem Wert ‚32799' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
			class_id		unsignedShort	
			class_version		unsignedByte	
		data_references	*data_reference*	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *profile_generic*. Die Werte, die in diesem Attribut angegeben sind, werden an den in ‚comm_profile_reference' festgelegten EMT versendet.
				class_id	unsignedShort	

Name des XML-Elements					XML-Datentyp		Beschreibung
				class_version	unsignedByte		Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚*OBIS.LogicalDevice-Zähler.sm*' oder dem Schema ‚*OBIS.LogicalDevice-TAF.sm*', das Attribut ‚class_id' dem Wert ‚4' oder dem Wert ‚32822' und das Attribut ‚class_version' dem Wert ‚0' entsprechen. Der Wert des Attributs ‚attr_reference' muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.
				attr_references	attr_reference	unsignedByte	
validity_window		*start_time*			dateTime		Zeitangabe des Gültigkeitszeitraums. Das Element *start_time* muss immer und ohne die Verwendung von Wildcards angegeben werden. Das Element *end_time* muss entweder ohne Wildcards angegeben werden oder vollständig auf leer gesetzt werden. Ist *end_time* auf leer gesetzt, wird der TAF nicht eigenständig durch das SMGW beendet. Der Wert des Elements *end_time* muss vor dem 01.01.2038 liegen. Das Attribut muss angegeben werden. Der Wert muss dem Schema „yyyy-mm-ddThh:mm:ssZ" (mit 4-stelliger Abgabe des Jahres *yyyy*, 2-stelliger Angabe der Minuten mm, 2-stelliger Angabe der Stunden *hh* und 2-stelliger Angabe der Sekunden *ss*) entsprechen. Der Wert ist als UTC Zeit mit einem ‚Z' am Ende anzugeben.
		end_time					
metering_point_id					string		Zählpunktbezeichnung. Das Attribut kann angegeben werden.

Name des XML-Elements	XML-Datentyp	Beschreibung
auto_cleanup	Boolean	Legt fest, ob ein TAF und die damit verbundenen Register und Messwerte automatisch entsprechend der folgenden Parameter gelöscht werden. Das Attribut kann angegeben werden. Der Wert muss entweder auf ‚false' oder ‚true' gesetzt werden. Es wird im SMGW nicht ausgewertet.
archive_duration	unsignedInt	Legt fest, wie lange (in Tagen) für diesen TAF relevante Messwerte vorgehalten werden müssen. Nach dem Erreichen dieser Dauer werden die für diesen TAF relevanten Messwerte automatisch durch das SMGW gelöscht. Wenn weniger als 15 Monate (458 Tage) angegeben sind, werden die TAF relevanten Daten erst nach 458 Tagen gelöscht. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
taf_state	string	(read-only) Ein TAF durchläuft grundsätzlich drei Zustände *ready*, *running* und *archive*. Ist ein TAF im Zustand ‚archive' oder im Zustand ‚running', werden einmal täglich alle den Input-Registern zugeordneten originären

Name des XML-Elements			XML-Datentyp	Beschreibung
				Messwerte, die „archive_duration“ Tage (aber mindestens 458 Tage) alt oder älter sind, vom SMGW eigenständig entfernt. Ein TAF kann im Zustand "*ready*" durch den GWA rückstandsfrei entfernt werden. Ein TAF kann zusätzlich im Zustand "*archive*" durch den GWA rückstandsfrei entfernt werden, wenn er mindestens "*archive_duration*" Tage im SMGW vorgehalten worden ist. Alle von einem TAF benötigten Laufzeit-Objekte (also alle Objekte, die nicht in der Domain des EMT angeordnet sind und nicht von anderen Objekten des SMGW referenziert werden) werden mit Wechsel in den Zustand "*Running*" instanziiert und mit Wechsel in den Zustand "*archive*" entfernt.
sensor_signature			Boolean	Der Wert des Attributs muss entweder auf ‚false‘ oder ‚true‘ gesetzt werden. Er wird im SMGW nicht ausgewertet.
pseudonym			string	(optional) Mit diesem Attribut kann ein Pseudonym angegeben werden. Bei der Übertragung der Messwerte werden sämtlich Hinweise auf den Zähler (Logical Device ID des Zählers) durch das Pseudonym ersetzt. Somit ist ein Rückschluss auf den konkreten Zähler oder das konkrete Smart Meter Gateway aus dem die Messwerte stammen nicht mehr möglich.
value_monitors	monitored_obj	logical_name	string	(optional)

Name des XML-Elements					XML-Datentyp	Beschreibung
		class_id			unsignedShort	Mit diesem Attribut können zu überwachende Grenzwerte angegeben werden. Im Attribut ‚monitored_obj' wird dabei die zu überwachende Messgröße angegeben, im Attribut ‚thresholds' die Grenzwerte, die für dieses Objekt überwacht werden müssen. Falls ein Grenzwert auf Überschreitung geprüft werden soll, ist das Attribut ‚direction' mit dem Wert ‚true' zu belegen. Falls ein Grenzwert auf Unterschreitung geprüft werden soll, ist das Attribut ‚direction' mit dem Wert ‚false' zu belegen. Die beiden Werte in ‚over_limit' und ‚under_limit' wirken dabei als Hysterese-Band. Der Messwertversand wird bei Prüfung auf Überschreitung erst dann erneut ausgelöst, wenn der Werte in ‚under_limit' unterschritten wurde und wird bei Prüfung auf Unterschreitung erst dann erneut ausgelöst, wenn der Wert in ‚over_limit' überschritten wurde.
		class_version			unsignedByte	
	thresholds	threshold	direction		Boolean	
			over_limit	value	long	
				scaler	Byte	
				unit	unsignedByte	
			under_limit	value	long	
				scaler	Byte	
				unit	unsignedByte	

Diese Klasse verfügt über die folgenden Methoden:

Tabelle 58: Beschreibung der Methoden der IC *taf10*

Methode	Beschreibung
onDemandReadout()	Die Methode löst einen einmaligen Versand der aktuellen Messwerte im Bedarfsfall an die EMTs, die in ‚comm_profile_reference' definiert sind aus.

Beim *taf10* ist die folgende OBIS Kennzahl zu verwenden:

- *taf10*: 01005E31802A

Die Instanzen der Klasse sind der Logical Device ID ldevid des TAFs zugeordnet. Bei mehreren *taf10* erfolgt die Unterscheidung also durch die Logical Device ID und nicht durch die OBIS Kennzahl. Der GWA kann TAF-Profile im SMGW anlegen. Profile werden mittels des HTTP-Verbs POST auf dem SMGW angelegt.

Vorbedingung ist, dass das TAF-Profil, das angelegt werden soll, zuvor nicht bereits angelegt wurde. Weiterhin müssen folgende Referenzen vor dem Einspielen eines TAFs auf dem Gateway vorhanden sein:

- *input_reference*
- *consumer_reference*
- *com_profile_reference*
- *qos_profile_reference*

Die entsprechenden Profile müssen durch den GWA also vor dem Einspielen eines TAFs ins SMGW eingebracht worden sein. Das folgende Beispiel verdeutlicht das Einbringen eines TAF-Profils mit dem HTTP-Verb POST:

Tabelle 59: Anlegen eines *taf10*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Anlegen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Anlegen einer neuen Instanz	POST	<PoC>/cosem/ldevs/	*taf10*	201	Leer

Der GWA kann TAF-Profile aus dem SMGW auslesen. Profile werden mittels des HTTP-Verbs GET ausgelesen. Vorbedingung ist, dass das TAF-Profil, das ausgelesen werden soll, zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 60: Auslesen eines *taf10*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Auslesen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Auslesen einer bestehenden Instanz	GET	<PoC>/cosem/ldevs/taf1012345679.sm/objects/01005E31802A	---	200	*taf10*

Der GWA kann TAF-Profile aus dem SMGW löschen. Profile werden mittels des HTTP-Verbs DELETE gelöscht. Vorbedingung ist, dass das TAF-Profil im Zustand „*archive*“ oder im Zustand „*ready*“ im SMGW vorhanden ist. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 61: Löschen eines *taf10*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Löschen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Löschen einer bestehenden Instanz	DELETE	<PoC>/cosem/ldevs/taf1012345679.sm/	---	200	---

Der GWA kann TAF-Profile im SMGW aktualisieren. Profile werden mittels des HTTP-Verbs UPDATE aktualisiert. Die Aktualisierung erfolgt stets auf den gesamten Container. Der GWA kann im TAF-Profil nur die Attribute „*end_time*“ und „*thresholds*“ anpassen und den entsprechenden TAF so beenden. Vorbedingung ist, dass das TAF-Profil, das aktualisiert werden soll, zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 62: Aktualisieren eines *taf10*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Aktualisieren eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Aktualisieren einer bestehenden Instanz	PUT	<PoC>/cosem/ldevs/taf1012345679.sm/objects/01005E31802A	*taf10* mit angepasster *end_time und oder angepassten thresholds*	200	---

IC *taf14*

Instanzen der Klasse *taf14* werden vom GWA zur Konfiguration des Tarifanwendungsfalls TAF14 im SMGW benutzt. Die Klasse kennt keine Methoden. Die Klasse kennt folgende Parameter:

Tabelle 63: Attribute der IC *taf14*

Name des XML-Elements			XML-Datentyp	Beschreibung
logical_name			String	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
taf_name			String	Freier Bezeichner des Auswertungsprofils gemäß [BSI TR-03109-1]. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
taf_identifier			String	Für das SMGW eindeutiger Bezeichner für das Auswertungsprofil. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
input_references	*input_reference*	*logical_name*	String	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) des Typs *gateway_signed_extended_register*. Das Attribut muss angegeben werden.
		class_id	unsignedShort	

Name des XML-Elements			XML-Datentyp	Beschreibung
		class_version	unsignedByte	Der Wert des Attributs ‚logical_name' muss auf den Wert des Attributs ‚value' einer auf dem SMGW vorhandenen Instanz der IC ‚*e_meter_sensor_setup*' oder der IC ‚*generic_meter_sensor_setup*' und somit dem Schema ‚value.LogicalDevice-Zähler.sm' entsprechen. Der Wert muss den zugelassenen Messgrößen der [BSI TR-03109-1] für diesen TAF entsprechen (bspw. 0100100700ff.1emh0005288229.sm). Der Wert des Attributs ‚class_id' muss auf ‚3' gesetzt werden. Der Wert des Attributs ‚class_version' muss auf ‚0' gesetzt werden.
number_of_tariffs			Byte	Anzahl der Tarifstufen. Das Attribut muss angegeben werden. Bei diesem TAF muss das Attribut immer mit ‚1' belegt werden.
active_tariff			Byte	(read-only) Dieses Attribut spiegelt den jeweils aktuell in dem TAF geltenden Tarif wider.
end_of_billing_period	*datetime_var*	*day_of_month*	String	Schreibt die abgeleiteten Register in die turnusgemäße Abrechnungsliste, entspricht dem Abrechnungszeitraum aus [BSI TR-03109] und dem Turnus aus den dortigen Abbildungen. Das Attribut muss angegeben werden.
		hour	unsignedByte	

Name des XML-Elements			XML-Datentyp	Beschreibung
		minute	unsignedByte	Die Datentypen ‚datetime_var' und ‚datetime_intervall' können alternativ zueinander verwendet werden. Bei Verwendung des Datentyps ‚datetime_var' werden die Zeitangaben, die nicht gesetzt werden als Wildcard angenommen. Wird beispielsweise nur das Attribut ‚minute' mit dem Wert ‚0' und das Attribut ‚second' mit dem Wert ‚0' angegeben, so endet der Abrechnungszeitraum an jedem Tag zu jeder vollen Stunde zur 0. Minute und zur 0. Sekunde. Der Wert des Attributs ‚day_of_month' muss zwischen ‚1' und ‚31', des Attributs ‚hour' zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen. Bei Verwendung des Datentyps ‚datetime_intervall' wird ein Startzeitpunkt angegeben und davon ausgehend eine Periode, die zyklisch das Ende des Abrechnungszeitraums angibt. Bei diesem TAF wird der Parameter nicht ausgewertet.
		second	unsignedByte	
	datetime_intervall	*start_time*	dateTime	
		period	unsignedInt	
capture_period			unsignedInt	Aktualisiert die Messwertliste, aktualisiert den Eingangswert des Zählers und die abgeleiteten Register. (Registrierperiode). Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen, in Sekunden angegeben werden und zwischen ‚1' und ‚86400' liegen. Bei diesem TAF entspricht das Attribut der Abtastrate aus der [BSI TR-03109].

Name des XML-Elements				XML-Datentyp	Beschreibung
capture_time			*hour*	unsignedByte	Uhrzeit zu der der abrechnungstechnische Kalendertag beginnt. Zu diesem Zeitpunkt wird der Wert für eine Spontanauslesung gemäß TAF-6 aufgezeichnet. Das Attribut muss angegeben werden. Der Wert des Attributs ‚hour' muss zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen. Bei diesem TAF wird das Attribut nicht ausgewertet.
			minute	unsignedByte	
			second	unsignedByte	
consumer_reference			*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *kaf_han_wan_container*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318011.LogicalDevice-Letztverbraucher.sm", das Attribut ‚class_id' dem Wert ‚32795' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
			class_id	unsignedShort	
			class_version	unsignedByte	
delivery_references	*delivery_reference*	*comm_profile_reference*	*logical_name*	string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *kaf_han_wan_container*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318011.LogicalDevice-EMT. sm", das Attribut ‚class_id' dem Wert ‚32795' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
			class_id	unsignedShort	
			class_version	unsignedByte	

Name des XML-Elements					XML-Datentyp	Beschreibung
		delivery_time	*datetime_var*	*year*	unsignedShort	Uhrzeit des Messwert-Versands. Das Attribut muss angegeben werden. Die Datentypen ‚datetime_var' und ‚datetime_intervall' können alternativ zueinander verwendet werden. Bei Verwendung des Datentyps ‚datetime_var' muss der Wert des Attributs ‚year' zwischen ‚2000' und ‚2199', des Attributs ‚month' zwischen ‚1' und ‚12', des Attributs ‚day_of_month' zwischen ‚1' und ‚31', des Attributs ‚hour' zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen. Der Wert des Attributs ‚day_of_week' muss entweder auf ‚Mon' oder ‚Tue' oder ‚Wed' oder ‚Thu' oder ‚Fri' oder ‚Sat' oder ‚Sun' gesetzt werden. Die Zeitangaben, die nicht gesetzt werden, werden als Wildcard angenommen. Wird beispielsweise nur das Attribut ‚minute' mit dem Wert ‚0' und das Attribut ‚second' mit dem Wert ‚0' angegeben, erfolgt der Messwertversand an jedem Tag zu jeder vollen Stunde zur 0. Minute und zur 0. Sekunde. Bei Verwendung des Datentyps ‚datetime_intervall' wird ein Startzeitpunkt angegeben und davon ausgehend eine Periode, die zyklisch den Versandzeitpunkt angibt.
				month	string	
				day_of_month	string	
				day_of_week	string	
				hour	unsignedByte	
				minute	unsignedByte	
				second	unsignedByte	
			datetime_intervall	*start_time*	dateTime	
				period	unsignedInt	

Name des XML-Elements							XML-Datentyp		Beschreibung
									Falls Messwerte unmittelbar nach der Messwertbildung versendet werden sollen ist das Attribut ‚period' mit dem Wert ‚0' zu belegen.
		qos_profile_reference	*logical_name*				string		Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *qos_sq_setup*. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318013.LogicalDevice-EMT.sm", das Attribut ‚class_id' dem Wert ‚32799' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
			class_id				unsignedShort		
			class_version				unsignedByte		
		data_references	*data_reference*	*logical_name*			string		Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *profile_generic*. Die Werte, die in diesem Attribut angegeben sind, werden an den in ‚comm_profile_reference' festgelegten EMT versendet. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚OBIS.LogicalDevice-Zähler. sm' oder dem Schema ‚OBIS.LogicalDevice-TAF.sm', das Attribut ‚class_id' dem Wert ‚4' oder dem Wert ‚32822' und das Attribut ‚class_version' dem Wert ‚0' entsprechen. Der Wert des Attributs ‚attr_reference' muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.
				class_id			unsignedShort		
				class_version			unsignedByte		
				attr_references			attr_reference	unsignedByte	
validity_window			*start_time*				dateTime		

Name des XML-Elements		XML-Datentyp	Beschreibung
	end_time		Zeitangabe des Gültigkeitszeitraums. Das Element *start_time* muss immer und ohne die Verwendung von Wildcards angegeben werden. Das Element *end_time* muss entweder ohne Wildcards angegeben werden oder vollständig auf leer gesetzt werden. Ist *end_time* auf leer gesetzt, wird der TAF nicht eigenständig durch das SMGW beendet. Der Wert des Elements *end_time* muss vor dem 01.01.2038 liegen. Das Attribut muss angegeben werden. Der Wert muss dem Schema „yyyy-mm-ddThh:mm:ssZ“ (mit 4-stelliger Abgabe des Jahres yyyy , 2-stelliger Angabe der Minuten mm, 2-stelliger Angabe der Stunden hh und 2-stelliger Angabe der Sekunden ss) entsprechen. Der Wert ist als UTC Zeit mit einem ‚Z‘ am Ende anzugeben.
metering_point_id		string	Zählpunktbezeichnung. Das Attribut kann angegeben werden.
auto_cleanup		Boolean	Legt fest, ob ein TAF und die damit verbundenen Register und Messwerte automatisch entsprechend der folgenden Parameter gelöscht werden. Das Attribut kann angegeben werden. Der Wert muss entweder auf ‚false‘ oder ‚true‘ gesetzt werden. Es wird im SMGW nicht ausgewertet.

Name des XML-Elements	XML-Datentyp	Beschreibung
archive_duration	unsignedInt	Legt fest, wie lange (in Tagen) für diesen TAF relevante Messwerte vorgehalten werden müssen. Nach dem Erreichen dieser Dauer werden die für diesen TAF relevanten Messwerte automatisch durch das SMGW gelöscht. Wenn weniger als 15 Monate (458 Tage) angegeben sind, werden die TAF relevanten Daten erst nach 458 Tagen gelöscht. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen.
taf_state	string	(read-only) Ein TAF durchläuft grundsätzlich drei Zustände *ready*, *running* und *archive*. Ist ein TAF im Zustand ‚archive' oder im Zustand ‚running', werden einmal täglich alle den Input-Registern zugeordneten originären Messwerte, die „archive_duration" Tage (aber mindestens 458 Tage) alt oder älter sind, vom SMGW eigenständig entfernt. Ein TAF kann im Zustand "*ready*" durch den GWA rückstandsfrei entfernt werden. Ein TAF kann zusätzlich im Zustand "*archive*" durch den GWA rückstandsfrei entfernt werden, wenn er mindestens "*archive_duration*" Tage im SMGW vorgehalten worden ist. Alle von einem TAF benötigten Laufzeit-Objekte (also alle Objekte, die nicht in der Domain des EMT angeordnet sind und nicht von anderen Objekten des SMGW referenziert

Name des XML-Elements				XML-Datentyp	Beschreibung
					werden) werden mit Wechsel in den Zustand "*Running*" instanziiert und mit Wechsel in den Zustand "*archive*" entfernt.
sensor_signature				Boolean	Der Wert des Attributs muss entweder auf ‚false' oder ‚true' gesetzt werden. Er wird im SMGW nicht ausgewertet.
pseudonym				string	(optional) Mit diesem Attribut kann ein Pseudonym angegeben werden. Bei der Übertragung der Messwerte werden sämtlich Hinweise auf den Zähler (Logical Device ID des Zählers) durch das Pseudonym ersetzt. Somit ist ein Rückschluss auf den konkreten Zähler oder das konkrete Smart Meter Gateway aus dem die Messwerte stammen nicht mehr möglich.
value_monitors	monitored_obj	logical_name		string	(optional) Mit diesem Attribut können zu überwachende Grenzwerte angegeben werden. Im Attribut ‚monitored_obj' wird dabei die zu überwachende Messgröße angegeben, im Attribut ‚thresholds' die Grenzwerte, die für dieses Objekt überwacht werden müssen. Falls ein Grenzwert auf Überschreitung geprüft werden soll, ist das Attribut ‚direction' mit dem Wert ‚true' zu belegen.
		class_id		unsignedShort	
		class_version		unsignedByte	
	thresholds	threshold	direction	Boolean	

Name des XML-Elements					XML-Datentyp	Beschreibung
			over_limit	value	long	Falls ein Grenzwert auf Unterschreitung geprüft werden soll, ist das Attribut ‚direction' mit dem Wert ‚false' zu belegen. Die beiden Werte in ‚over_limit' und ‚under_limit' wirken dabei als Hysterese-Band. Der Messwertversand wird bei Prüfung auf Überschreitung erst dann erneut ausgelöst, wenn der Werte in ‚under_limit' unterschritten wurde und wird bei Prüfung auf Unterschreitung erst dann erneut ausgelöst, wenn der Wert in ‚over_limit' überschritten wurde.
				scaler	Byte	
				unit	unsignedByte	
			under_limit	value	long	
				scaler	Byte	
				unit	unsignedByte	

Beim *taf14* ist die folgende OBIS Kennzahl zu verwenden:

- *taf14*: 01005E31802E

Die Instanzen der Klasse sind der Logical Device ID ldevid des TAFs zugeordnet. Bei mehreren *taf14* erfolgt die Unterscheidung also durch die Logical Device ID und nicht durch die OBIS Kennzahl. Der GWA kann TAF-Profile im SMGW anlegen. Profile werden mittels des HTTP-Verbs POST auf dem SMGW angelegt.

Vorbedingung ist, dass das TAF-Profil, das angelegt werden soll, zuvor nicht bereits angelegt wurde. Weiterhin müssen folgende Referenzen vor dem Einspielen eines TAFs auf dem Gateway vorhanden sein:

- *input_reference*
- *consumer_reference*
- *com_profile_reference*
- *qos_profile_reference*

Die entsprechenden Profile müssen durch den GWA also vor dem Einspielen eines TAFs ins SMGW eingebracht worden sein. Das folgende Beispiel verdeutlicht das Einbringen eines TAF-Profils mit dem HTTP-Verb POST:

Tabelle 64: Anlegen eines *taf14*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Anlegen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Anlegen einer neuen Instanz	POST	<PoC>/cosem/ldevs/	*taf14*	201	Leer

Der GWA kann TAF-Profile aus dem SMGW auslesen. Profile werden mittels des HTTP-Verbs GET ausgelesen. Vorbedingung ist, dass das TAF-Profil, das ausgelesen werden soll, zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 65: Auslesen eines *taf14*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Auslesen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Auslesen einer bestehenden Instanz	GET	<PoC>/cosem/ldevs/taf1412345679.sm/objects/01005E31802E	---	200	*taf14*

Der GWA kann TAF-Profile aus dem SMGW löschen. Profile werden mittels des HTTP-Verbs DELETE gelöscht. Vorbedingung ist, dass das TAF-Profil im Zustand „*archive*“ oder im Zustand „*ready*“ im SMGW vorhanden ist. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 66: Löschen eines *taf14*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Löschen eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Löschen einer bestehenden Instanz	DELETE	<PoC>/cosem/ldevs/taf1412345679.sm/	---	200	---

Der GWA kann TAF-Profile im SMGW aktualisieren. Profile werden mittels des HTTP-Verbs UPDATE aktualisiert. Die Aktualisierung erfolgt stets auf den gesamten Container. Der GWA kann im TAF-Profil nur die Attribute „*end_time*“ und „*thresholds*“ anpassen und den entsprechenden TAF so beenden. Vorbedingung ist, dass das TAF-Profil, das aktualisiert werden soll, zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 67: Aktualisieren eines *taf14*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Aktualisieren eines TAF-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Aktualisieren einer bestehenden Instanz	PUT	<PoC>/cosem/ldevs/taf1412345679.sm/objects/01005E31802E	*taf14* mit angepasster *end_time und oder angepassten thresholds*	200	---

4.4.3.4 Verwaltung der CLS-Profile

IC *proxy_server*

Instanzen der Klasse *proxy_server* werden vom GWA gemäß WKS1 zur Konfiguration der CLS-Proxy-Komponente des SMGW benutzt. Die Parameter sind wie folgt:

Tabelle 68: Attribute der IC *proxy_server*

Name des XML-Elements		XML-Datentyp	Beschreibung
logical_name		string	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
wan_peer	*logical_name*	string	Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) auf ein COSEM-Objekt vom Typ *kaf_han_wan_container* zur Konfiguration der CLS-Gegenstelle im WAN. Das Attribut muss angegeben werden. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318011.LogicalDevice-EMT. sm", das Attribut ‚class_id' dem Wert ‚32795' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
	class_id	unsignedShort	
	class_version	unsignedByte	
han_peer	*logical_name*	string	Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) auf ein COSEM-Objekt vom Typ *kaf_han_wan_container* zur Konfiguration der CLS-Gegenstelle im HAN. Das Attribut muss angegeben werden.
	class_id	unsignedShort	
	class_version	unsignedByte	

Name des XML-Elements		XML-Datentyp	Beschreibung
			Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318011.LogicalDevice-Letztverbraucher.sm", das Attribut ‚class_id' dem Wert ‚32795' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
session_status	*connected*	boolean	(read-only) Zustand der Proxy-Verbindung. Der Wert ‚*true*' zeigt an, dass die Proxy-Verbindung besteht.
	session_timer	unsignedInt	(read-only) Angabe der seit Verbindungsbeginn vergangenen Dauer (in Sekunden). Der Wert entspricht den Vorgaben des Datentyps.
connect_time	*year*	unsignedShort	(optional) Mit dem Attribut kann angegeben werden, zu welchen Zeitpunkten eine CLS Verbindung vom SMGW hergestellt werden soll (HKS5). Der Wert des Attributs ‚year' muss zwischen ‚2000' und ‚2199', des Attributs ‚month' zwischen ‚1' und ‚12', des Attributs ‚day_of_month' zwischen ‚1' und ‚31', des Attributs ‚hour' zwischen ‚0' und ‚23', des Attributs ‚minute' zwischen ‚0' und ‚59' und des Attributs ‚second' zwischen ‚0' und ‚59' liegen. Der Wert des Attributs ‚day_of_week' muss entweder auf ‚Mon' oder ‚Tue' oder ‚Wed' oder ‚Thu' oder ‚Fri' oder ‚Sat' oder ‚Sun' gesetzt werden. Die nicht gesetzten Zeitangaben werden als Wildcard angenommen. Wird beispielsweise nur das Attribut ‚minute' mit dem Wert ‚0' und das Attribut ‚second' mit dem Wert ‚0' angegeben, erfolgt der Verbindungsaufbau an jedem Tag zu jeder vollen Stunde zur 0. Minute und zur 0. Sekunde.
	month	string	
	day_of_month	string	
	day_of_week	string	
	hour	unsignedByte	
	minute	unsignedByte	
	second	unsignedByte	

Name des XML-Elements	XML-Datentyp	Beschreibung
CLS_connect	Boolean	(Optional) Wenn das Attribut mit ‚true' belegt ist, darf die CLS Verbindung durch das CLS Gerät aus dem HAN initiiert werden (HKS3). Wenn das Attribut mit ‚false' belegt ist, darf die CLS Verbindung durch das CLS Gerät aus dem HAN nicht initiiert werden.

Diese Klasse verfügt über die Methoden *connect* und *disconnect*.

Die Methode *connect* verbindet die mittels der Parameter *wan_peer* und *han_peer* definierten beiden CLS-Gegenstellen miteinander, indem der Wert des Parameters *session_timer* auf „*0*“ gesetzt wird, sodass die Proxy-Verbindung umgehend hergestellt wird. Bei erfolgreichem Verbindungsaufbau wird der Wert des Parameters *connected* auf „*true*“ gesetzt, sonst auf „*false*“.

Die Methode *disconnect* trennt die mittels der Parameter *wan_peer* und *han_peer* definierten beiden CLS-Gegenstellen. Der Wert des Parameters *connected* wird dabei auf „*false*“ gesetzt.

Bei Auslieferung ist im SMGW kein CLS-Profil vorhanden.

Vor dem Einspielen der Klasse müssen folgende Referenzen auf dem SMGW vorhanden sein:

- *wan_peer*
- *han_peer*

Beim CLS-Profil sind die folgenden OBIS-Kennzahlen zu verwenden:

- *proxy_server:* *01005E318017*

Die Instanzen der Klasse sind der Logical Device ID *ldevid* des CLS-Profils zugeordnet. Bei mehreren CLS-Profilen erfolgt die Unterscheidung also durch die Logical Device ID und nicht durch die OBIS Kennzahl. Der GWA kann CLS-Profile im SMGW anlegen. Profile werden mittels des HTTP-Verbs POST auf dem SMGW angelegt. Vorbedingung ist, dass das CLS-Profil, das angelegt werden soll, zuvor nicht bereits angelegt wurde. Weiterhin müssen die Referenzen des Profils in Form der entsprechenden WAN- und HAN-Profile vorhanden sein. Falls die Referenzen nicht vorhanden sind, ist ein Einspielen nicht möglich. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 69: Anlegen eines CLS-Profils

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Anlegen eines CLS-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Anlegen einer neuen Instanz	POST	<PoC>/cosem/ldevs/	*proxy_server*	201	Leer

Der GWA kann CLS-Profile aus dem SMGW auslesen. Profile werden mittels des HTTP-Verbs GET ausgelesen. Vorbedingung ist, dass das CLS-Profil, das ausgelesen werden soll, zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 70: Auslesen eines CLS-Profils

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Auslesen eines CLS-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Auslesen einer bestehenden Instanz	GET	<PoC>/cosem/ldevs/cls12345678.sm/objects/01005E318017	---	200	proxy_server

Der GWA kann CLS-Profile aus dem SMGW löschen. Profile werden mittels des HTTP-Verbs DELETE gelöscht. Vorbedingung ist, dass das CLS-Profil, das gelöscht werden soll, zuvor durch den GWA eingebracht wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 71: Löschen eines CLS-Profils

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Löschen eines CLS-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Löschen einer bestehenden Instanz	DELETE	<PoC>/cosem/ldevs/cls12345678.sm	---	200	---

Der GWA kann CLS-Profile im SMGW aktualisieren. Profile werden mittels des HTTP-Verbs PUT aktualisiert. Vorbedingung ist, dass das CLS-Profil, das aktualisiert werden soll, zuvor durch den GWA angelegt wurde. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 72: Aktualisieren eines CLS-Profils

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Aktualisieren eines CLS-Profils	Request-Body	HTTP Response Statuscode	Response-Body
Aktualisieren einer bestehenden Instanz	PUT	<PoC>/cosem/ldevs/cls12345678.sm/objects/01005E318017	*proxy_server* mit den gewünschten Änderungen	200	---

4.4.3.5 Pseudonymisierung

Messdaten werden im XML in Objekten der IC ***profile_generic*** an den EMT übertragen. Jedes Objekt der IC ***profile_generic*** Objekte enthält für jeden Messwert ein Objekt der IC **gateway_signed_*extended_register***.

Zur Identifikation des Ursprungs der Messwerte sind in diesen Objekten folgende Attribute vorhanden:

- Logical Device Name des Objekts der IC ***profile_generic***: Setzt sich zusammen aus der OBIS Kennzahl des Objekts (siehe Kapitel 9) sowie der Logical Device ID des Tarifprofils. Die Logical Device ID des Tarifprofils kann durch den GWA beliebig konfiguriert werden. Somit ist abhängig von der Konfiguration der Logical Device ID des Tarifprofils kein Rückschluss auf den Zähler, das SMGW oder den Letztverbraucher möglich.
- Logical Device Name des Objekts der IC **gateway_signed_*extended_register***: Setzt sich zusammen aus der OBIS Kennzahl des Objekts (entsprechend des enthaltenen Messwerts) und der Logical Device ID des Tarifprofils oder des Zählerprofils. Die Logical Device ID des Zählerprofils wird verwendet, wenn der Messwert unverändert vom Zähler durchgereicht wurde (bspw. TAF-1), die des Tarifprofils, wenn der Messwert durchs Tarifprofil erzeugt wurde (bspw. TAF-2). Die Logical Device ID des Zählerprofils entspricht der tatsächlichen Kennung des Zählers, die u.a. auch auf dem Zähler aufgedruckt ist. Somit ist aus der Logical Device ID des Zählerprofils ein Rückschluss auf den Zähler und damit unter Umständen auch auf den Letztverbraucher möglich.

Die Logical Device ID des SMGW wird nicht zur Identifikation der Messwert verwendet und ist daher in den Objekten der IC ***profile_generic*** nicht vorhanden. Daher muss bei der Logical Device ID des SMGWs auch keine Ersetzung durch ein Pseudonym vorgenommen werden.

Um pseudonymisierte Messwerte zu erhalten, muss der GWA das SMGW wie folgt parametrisieren:

- TAF-Profil (IC ***taf01***, ***taf02***, ***taf07***, ***taf09***, ***taf10***, ***taf14***) einspielen, bei dem die Logical Device ID keine Rückschlüsse auf den Zähler, das SMGW oder den Letztverbraucher zulässt. Die TAF-ID kann frei festgelegt werden. Zudem muss im TAF-Profil das Attribut ‚pseudonym' gesetzt sein. Nur wenn dieses Attribut gesetzt ist, werden die Logical Device IDs in den Objekten der IC **gateway_signed_*extended_register*** durch das Pseudonym ersetzt. Wenn dieses Attribut nicht gesetzt ist, werden die Messwerte nicht pseudonymisiert übertragen.
- EMT-Kommunikationsprofil (IC ***kaf_han_wan_container***) einspielen, das als Inhalt des Attributs
  - *dest_address* die Zieladresse des GWA
  - *cert_tls* das TLS Zertifikat des GWA
  - *cert_enc* das ENC Zertifikat des EMT
  - *cert_sig* das SIG Zertifikat des EMT

  enthält.

Damit werden die pseudonymisierten Messwerte für den EMT verschlüsselt und signiert und anschließend an den GWA übertragen. Der GWA muss die Signatur des SMGW prüfen und entfernen. Der GWA leitet die Daten dann an den EMT weiter. Der EMT kann die Daten entschlüsseln, ein Rückschluss auf das konkrete SMGW oder den konkreten Zähler sind nicht mehr möglich.

4.4.4 GWA-Zugriff auf das SMGW-Sicherheitsmodul

Für die Interaktion zwischen SMGW und seinem Sicherheitsmodul ist es notwendig, dass der GWA sich gegenüber dem Sicherheitsmodul authentisiert und derart das Sicherheitsmodul in die Lage versetzt, den Zustand AUTH gemäß [BSI TR-03109-2] einzunehmen.

Der Zugriff auf das Sicherheitsmodul erfolgt gemäß [DKE-AK 142, Kapitel 6.3.4].

Bei erfolgreichem Erreichen der PACE-gesicherten Kommunikationsverbindung des Sicherheitsmoduls zum SMGW kann nachfolgend der notwendige AUTH-Zustand durch den GWA zum Sicherheitsmodul aufgebaut werden und somit weitere auf Geheiß des GWA durchzuführende Aktionen des Sicherheitsmoduls veranlassen. Die hierfür notwendigen Kommandos des GWA an der WAN-Schnittstelle und die dabei zu verwendenden COSEM-IC-Klassen sind vom SMGW-Hersteller wie folgt spezifiziert worden.

Neben der Beschreibung der zu verwendenden COSEM-IC sind in den Unterkapiteln dieses Kapitels auch Kommunikationssequenzen für die Anwendungsfälle „Personalisierung“, „Zertifikatswechsel“, „Root-Zertifikatswechsel“ und „Gateway-Administrator-Wechsel“ beschrieben.

IC *security_module*

Instanzen der Klasse *security_module* wird zur Ansteuerung des Sicherheitsmoduls des Smart Meter Gateways benötigt. Die Parameter sind wie folgt:

Tabelle 73: Beschreibung der Attribute der IC *security_module*

Name des XML-Elements	XML-Datentyp	Beschreibung
logical_name	Octet-String	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der *tr_certificate_setup*. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
root_certificates	TYPE_KeyRef_certificate	(read-only) Array *TYPE_KeyRef_certificate* (read-only) *TYPE_KeyRef_certificate* ::= Structure*:* *{ KeyRef* Octet-String *certificate* Octet-String *}* Mit diesem Attribut können die auf dem Sicherheitsmodul vorhandenen Root-Zertifikate abgefragt werden. *KeyRef* bezeichnet dabei die Referenz auf die entsprechende Speicherbank im Sicherheitsmodul. Das Attribut wird in der vorliegenden SMGW-Version nicht genutzt.

Name des XML-Elements	XML-Datentyp	Beschreibung
GWA_keys	*TYPE_KeyRef_Key*	(read-only) Array *TYPE_KeyRef_Key* (read-only) *TYPE_KeyRef_Key*::= Structure*:* *{ KeyRef* Octet-String *Key* Octet-String *}* Mit diesem Attribut können die, auf dem Sicherheitsmodul vorhandenen GWA Keys, abgefragt werden. Das Attribut wird in der vorliegenden SMGW-Version nicht genutzt.
SM_state	*TYPE_SM_state*	(read-only) *TYPE_SM_state*::= Structure*:* *{ SEID* Unsigned8 *ext_auth* Boolean *used_key_objects* Unsigned8 *free_key_objects* Unsigned8 *manufacture_specific* Octet-String *}* Falls die Informationen über *used_key_objects* und *free_key_objects* nicht bereitgestellt werden, werden beide Werte mit 0 belegt. Das Attribut wird in der vorliegenden SMGW-Version nicht genutzt.

Diese Klasse verfügt über die folgenden Methoden:

Tabelle 74: Beschreibung der Methoden der IC *security_module*

Methode	Beschreibung
getAUTHChallenge	Rückgabewert: Octet-String Über diese Methode kann der GWA eine Challenge vom Sicherheitsmodul anfordern. Bis die Challenge signiert vom GWA zurückgegeben wird, hält das Sicherheitsmodul die Challenge intern in einem Register, welches nicht überschrieben werden darf.
responseAUTHChallenge (Octet-String)	Mit dieser Methode wird die, mit dem privaten Schlüssel AUT des GWA signierte, 16 Byte lange Challenge in das SMGW zurückgeschrieben. Zuvor wurde die Challenge über die Methode *getAUTHChallenge* angefordert.
terminateAUTH	Mit dieser Methode wird der AUTH-Zustand des Sicherheitsmoduls durch das SMGW beendet.
createInitialKeyPairs	Mit dieser Methode werden die einleitend beschriebenen neuen Schlüsselbänke angelegt. • *Key.WAN_TLS_2* • *Key.WAN_SIG_2* • *Key.WAN_ENC_2* • *Key.LMN_1* • *Key.LMN_2* • *Key.HAN_1* • *Key.HAN_2* Die Nutzung dieser Methode setzt den AUTH-Zustand voraus und muss im Rahmen des Personalisierungsprozesses durch den GWA ausgeführt werden.

Methode	Beschreibung
writeImportCertificate(TYPE_GWA_ImportCert)	Mit dieser Methode können Importzertifikate vom GWA in das Sicherheitsmodul geschrieben werden. *TYPE_GWA_ImportCert ::= Structure* { • *ImportCertificate_TLS* *Octet-String* • *ImportCertificate_ENC* *Octet-String* • *ImportCertificate_SIG* *Octet-String* • *ImportCertificate_AUT* *Octet-String* } Die Nutzung dieser Methode setzt den AUTH-Zustand voraus. Die Parameter der Methode müssen den Vorgaben des Datentyps entsprechen und gültige Importzertifikate darstellen.
getGWA_newKeys	Rückgabewert: *TYPE_KeyRef_set ::= Structure* *{* • *TLS* *Octet-String* • *ENC* *Octet-String* • *SIG* *Octet-String* • *AUT* *Octet-String* }

Methode	Beschreibung
	Über diese Methode können die KeyRefs der Schlüssel des GWA abgefragt werden. Da das Sicherheitsmodul zum Schlüsselwechsel über zwei Bänke verfügt, werden vom SMGW die KeyRefs der Bank ausgegeben, die mit neuen Schlüsseln beschrieben werden können. Die KeyRefs müssen zur Generierung der entsprechenden Import-Zertifikate verwendet werden.
deactivateGWA_oldKeys	Mit dieser Methode können die alten Keys des GWA (Key.GWA_x_1 bzw. Key.GWA_x_2) deaktiviert werden. Die Methode kann angewendet werden, nachdem neue GWA-Keys über die entsprechenden Import-Zertifikate ins Sicherheitsmodul geschrieben wurden. Die Nutzung dieser Methode setzt den AUTH-Zustand voraus.
getRootCertificate_newKey	Rückgabewert: Octet-String Über diese Methode kann die KeyRef für einen neuen Root Schlüssel ausgegeben werden.
writeRootCertificate(TYPE_root_certificate_set)	Mit dieser Methode kann ein neues Root-Zertifikat in das Sicherheitsmodul geschrieben werden. *TYPE_root_certificate_set* ::= Structure { • *import_certificate* Octet-String • *X.509_root_certificate* Octet-String • *X.509_link_root* Octet-String } Die Nutzung dieser Methode setzt den AUTH-Zustand voraus. Die Parameter der Methode müssen den Vorgaben des Datentyps entsprechen und gültige Importzertifikate darstellen. Der Wert des Parameters ‚import_certificate' muss einem gültigen

Methode	Beschreibung
	Importzertifikat entsprechen. Der Wert des Parameters ‚X.509_root_certificate' und des Parameters ‚X.509_link_root' müssen gültigen X.509 Zertifikaten im DER-Format hexbinary codiert entsprechen.
deleteRootCertificate(Octet-String)	Mit der Methode kann ein Root-Zertifikat einschließlich des entsprechenden öffentlichen Schlüssels aus dem Sicherheitsmodul entfernt werden. Als Parameter wird die Subject KeyID des X.509-Zertifikats übergeben. Die Nutzung dieser Methode setzt den AUTH-Zustand voraus. Die Parameter der Methode müssen den Vorgaben des Datentyps und einer gültigen sowie auf dem SMGW vorhandenen Subject KeyID entsprechen. Der Aufruf der Methode wird vom SMGW nicht unterstützt.
deactivateWAN_oldKeys	Mit dieser Methode können die alten WAN-Keys (Key.WAN_y_x) im Sicherheitsmodul deaktiviert werden. Die Methode sollte erst ausgeführt werden, wenn alle Kommunikationsprofile im SMGW dementsprechend angepasst wurden. Die Nutzung dieser Methode setzt den AUTH-Zustand voraus.
generateInitialWANCSR(issuer_cert, curveID)	Rückgabewert: *TYPE_CSR_set* ::= Structure { • *TLS* Octet-String • *SIG* Octet-String • *ENC* Octet-String • *container* Octet-String

Methode	Beschreibung
	} *issuer_cert* Octet-String *curveID* Enum*:= {* *CURVE_OID_BRAINPOOL_P256R1,* *CURVE_OID_BRAINPOOL_P384R1,* *CURVE_OID_BRAINPOOL_P512R1,* *CURVE_OID_NIST_P256,* *CURVE_OID_NIST_P384* *}* Mit dieser Methode wird die Erzeugung der Schlüsselpaare und der entsprechenden *Certificate Signing Requests* für die Wirkzertifikate angestoßen. Über die gesamte Datenstruktur, die als Rückgabewert zurückgegeben wird, wird eine äußere Signatur mit dem, dem Gütesiegelzertifikat (SIG) zugehörigen privaten Schlüssel gebildet. Der Container enthält entsprechend der Vorgaben der PKI die signierten CSRs. Die Nutzung dieser Methode setzt den AUTH-Zustand voraus. Der Parameter ‚issuer_cert' muss dem Datentyp entsprechen und ein gültiges Zertifikat im DER-Format hexbinary codiert enthalten. Der Parameter ‚curve_id' muss einem der angegeben Werte entsprechen.
generateWANCSR(issuer_cert, curveID)	Rückgabewert: *TYPE_CSR_set* (siehe oben)

Methode	Beschreibung
	Mit dieser Methode wird die Erzeugung eines neuen WAN-Schlüsselpaars und der entsprechenden *Certificate Signing Requests* für das SMGW angestoßen. Über die gesamte Datenstruktur, die als Rückgabewert zurückgegeben wird, wird eine äußere Signatur mit dem vorhergehenden privaten Schlüssel (SIG) gebildet. Mit dem *issuer_cert* wird das TLS-Zertifikat der entsprechenden SubCA übergeben, das für den CSR benötigt wird. Die Nutzung dieser Methode setzt den AUTH-Zustand voraus. Der Parameter ‚issuer_cert' muss dem Datentyp entsprechen und ein gültiges Zertifikat im DER-Format hexbinary codiert enthalten. Der Parameter ‚curve_id' muss einem der angegeben Werte entsprechen.
generateHANCertificate	Rückgabewert: Octet-String Mit dieser Methode wird die Erzeugung eines neuen SMGW-HAN-Schlüsselpaars und des entsprechenden selbstsignierten Zertifikats angestoßen. Die Nutzung dieser Methode setzt den AUTH-Zustand voraus.
deactivateHAN_oldKey	Mit dieser Methode können die alten HAN-Keys (Key.HAN_x) im Sicherheitsmodul deaktiviert werden. Die Methode sollte erst ausgeführt werden, wenn alle Kommunikationsprofile im SMGW dementsprechend angepasst wurden. Die Nutzung dieser Methode setzt den AUTH-Zustand voraus.
generate_newLMNKeyPair	Rückgabewert: KeyRef des neu erzeugten Schlüssels als Octet-String

Methode	Beschreibung
	Mit dieser Methode werden die SMGW-Schlüssel für das LMN neu erzeugt. Die alten Schlüssel bleiben zunächst erhalten. Anschließend müssen die Zählerprofile auf die neuen KeyRefs umgestellt werden. Die Nutzung dieser Methode setzt den AUTH-Zustand voraus.
deactivate_oldLMNKeyPair	Mit dieser Methode wird das alte Schlüsselpaar deaktiviert. Die Methode sollte erst ausgeführt werden, wenn alle LMN-Teilnehmer auf das neue Schlüsselpaar umgestellt sind. Die Nutzung dieser Methode setzt den AUTH-Zustand voraus.

Bei Auslieferung ist im SMGW eine Instanz der IC *security_module* vorhanden.

Für diese Instanz wird die folgende OBIS Kennzahl verwendet:

- *security_module:* *01005E3180A2*

Die Instanzen der Klasse sind der Logical Device ID *ldevid* des SMGW zugeordnet. Der GWA kann Methoden der Klasse ausführen. Er kann die Klasse nicht lesen, aktualisieren, löschen oder erzeugen.

Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 75: Aufruf einer Methode der IC *security_module*

Zugriffs-Dienst	HTTP-Verb	URI zum Auslesen der Klasse security_module	Request-Body	HTTP Response Statuscode	Response-Body
Aufruf der Methode	POST	<PoC>/cosem/ldevs/eppc0110123456.sm/objects/01005e3180a2/methods/generateInitialWANCSR	*issuer_cert, curveID*	200	*TYPE_CSR_set*

Für bestimmte Methoden ist der AUTH-Zustand verpflichtende Voraussetzung. Der AUTH-Zustand wird über die Methoden:

- *getAUTHChallenge*
- *terminateAUTH*

hergestellt und wieder beendet.

IC *tr_certificate_setup*

Instanzen der Klasse *tr_certificate_setup* dienen zur Verwaltung der Zertifikate des Gateways. Diese Klasse kennt keine Methoden. Die Attribute dieser Klasse sind wie folgt:

Tabelle 76: Attribute der IC *tr_certificate_setup*

Name des XML-Elements	XML-Datentyp	Beschreibung
logical_name	Octet-String	Octet-String; OBIS-Code und Geräte-ID zur Identifizierung der Instanz der *tr_certificate_setup*. Das Attribut muss angegeben werden. Der Wert muss dem Schema ‚OBIS.LogicalDevice.sm' entsprechen und hat eine maximale Länge von 255 Zeichen.
C_sub_CA	Octet-String	Enthält das SubCA-Zertifikat für die mit der vorliegenden Klasse eingespielten Zertifikate. Das Attribut kann angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen und ein gültiges hexbinary-kodiertes Zertifikat im DER-Format enthalten.
C_TLS	Octet-String	TLS-Authentifizierungs-Zertifikat des SMGW. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen und ein gültiges hexbinary-kodiertes Zertifikat im DER-Format enthalten.

Name des XML-Elements	XML-Datentyp	Beschreibung
key_id_C_TLS	Octet-String	Key-Identifier zum TLS-Authentifizierungs-Zertifikat. Das Attribut wird in der vorliegenden Produktvariante nicht benutzt. Das Attribut kann angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.
C_ENC	Octet-String	Schlüsseltransport-Zertifikat. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen und ein gültiges hexbinary-kodiertes Zertifikat im DER-Format enthalten.
key_id_C_ENC	Octet-String	Key-Identifier zum Schlüsseltransport-Zertifikat. Das Attribut wird in der vorliegenden Produktvariante nicht benutzt. Das Attribut kann angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.
C_SIG	Octet-String	Inhaltsdaten-Signatur-Zertifikat. Das Attribut muss angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen und ein gültiges hexbinary-kodiertes Zertifikat im DER-Format enthalten.

Name des XML-Elements	XML-Datentyp	Beschreibung
key_id_C_SIG	Octet-String	Key-Identifier zum Inhaltsdaten-Signatur-Zertifikat. Das Attribut wird in der vorliegenden Produktvariante nicht benutzt. Das Attribut kann angegeben werden. Der Wert muss den Vorgaben des Datentyps entsprechen. Es wird im SMGW nicht ausgewertet.

Bei Auslieferung ist im SMGW eine Instanz der IC *tr_certificate_setup* vorhanden.

Für diese Instanz wird die folgende OBIS Kennzahl verwendet:

- *tr_certificate_setup:* *01005e3180a5*

Die Instanzen der Klasse sind der Logical Device ID ldevid des SMGW zugeordnet. Der GWA kann die Klasse nur aktualisieren, um die Zertifikate des SMGW zu aktualisieren. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 77: Aktualisierung der Zertifikate des SMGW

Zugriffs-Dienst	HTTP-Verb	Beispiel URI zum Aktualisieren der Klasse *tr_certificate_setup*	Request-Body	HTTP Response Statuscode	Response-Body
Aktualisierung der Klasseninstanz	PUT	<PoC>/cosem/ldevs/eppc0110123456.sm/objects/01005e3180a5	*tr_certificate_setup*	200	---

4.4.4.1 Personalisierung

Zur Personalisierung des SMGW müssen vom Gateway Administrator folgende Schritte durchgeführt werden:

1. Authentisierung des GWA am Sicherheitsmodul: Aufruf der Methode *getAUTHChallenge* und *responseAUTHChallenge (Detailbeschreibung der Methoden in* Tabelle 74*)*
2. *Certificate Signing Request (CSR)* erzeugen: Aufruf der Methode *generateInitialWANCSR (Detailbeschreibung der Methode in* Tabelle 74*)*
3. Authentisierung des GWA am Sicherheitsmodul beenden: Aufruf der Methode *terminateAUTH (Detailbeschreibung der Methode in* Tabelle 74*)*
4. Aktualisierung der Zertifikate des Gateways: Aktualisierung der Klasse *tr_certificate_setup (Detailbeschreibung der IC* in Tabelle 76*)*

4.4.4.2 Zertifikatswechsel

1. Zum Zertifikatswechsel des SMGW müssen vom Gateway Administrator folgende Schritte durchgeführt werden: Authentisierung des GWA am Sicherheitsmodul: Aufruf der Methode *getAUTHChallenge* und *responseAUTHChallenge (*Detailbeschreibung der Methoden in Tabelle 76*)*
2. *Certificate Signing Request (CSR)* erzeugen: Aufruf der Methode *generateWANCSR (Detailbeschreibung der Methoden in* Tabelle 74*)*
3. Authentisierung des GWA am Sicherheitsmodul beenden: Aufruf der Methode *terminateAUTH (Detailbeschreibung der Methoden in* Tabelle 74*)*

4. Aktualisierung der Zertifikate des Gateways: Aktualisierung der Klasse *tr_certificate_setup (Detailbeschreibung der IC in* Tabelle 76*)*

Nach erneutem Aufbau des Management-Kanals:

1. Authentisierung des GWA am Sicherheitsmodul: Aufruf der Methode *getAUTHChallenge* und *responseAUTHChallenge (Detailbeschreibung der Methoden in* Tabelle 74*)*
2. Deaktivierung der alten WAN-Schlüssel: Aufruf der Methode *deactivateWAN_oldKeys (Detailbeschreibung der Methoden in* Tabelle 74*)*
3. Authentisierung des GWA am Sicherheitsmodul beenden: Aufruf der Methode *terminateAUTH (Detailbeschreibung der Methoden in* Tabelle 74*)*

4.4.4.3 Root-Zertifikatswechsel

Zum Wechsel des Root-Zertifikats müssen vom Gateway Administrator folgende Schritte durchgeführt werden:

1. Abfrage der ID einer freien Speicherbank für ein zusätzliches Root-Zertifikat auf dem Sicherheitsmodul, der Rückgabewert der Methode wird vom Gateway Administrator zur Erstellung des Import Zertifikats benötigt, das im nächsten Schritt ins SMGW geschrieben wird: Aufruf der Methode *getRootCertificate_newKey (Detailbeschreibung der Methode in* Tabelle 74*)*
2. Authentisierung des GWA am Sicherheitsmodul: Aufruf der Methode *getAUTHChallenge* und *responseAUTHChallenge (Detailbeschreibung der Methoden in* Tabelle 74*)*
3. Schreiben eines neuen Root-Zertifikats auf das Sicherheitsmodul: Aufruf der Methode writeRootCertificate *(Detailbeschreibung der Methode in* Tabelle 74*)*
4. Optionales Löschen des alten Root-Zertifikat: Aufruf der Methode *deleteRootCertificate (Detailbeschreibung der Methode in* Tabelle 74*)*
5. Authentisierung des GWA am Sicherheitsmodul beenden: Aufruf der Methode *terminateAUTH (Detailbeschreibung der Methoden in* Tabelle 74*)*

4.4.4.4 Gateway-Administrator-Wechsel

Zum Wechsel des Gateway Administrators müssen vom bestehenden Gateway Administrator folgende Schritte durchgeführt werden:

1. Abfrage der ID der freien Speicherbänke für die Schlüssel des neuen Gateway Administrators auf dem Sicherheitsmodul, der Rückgabewert der Methode wird zur Erstellung der Importzertifikate benötigt, die im nächsten Schritt ins SMGW geschrieben werden: Aufruf der Methode *getGWA_newKeys (Detailbeschreibung der Methoden in* Tabelle 74*)*
2. Einspielen der Kommunikationsprofile des neuen Gateway Administrators: Aktualisieren der Klasse kaf_han_wan_container für Management, Admin-Service und NTP-TLS (Detailbeschreibung der IC in Tabelle 24).

Hierfür sind die folgenden *Logical Device IDs* zu verwenden:
- *gwapre-management.sm*
- *gwapre-adminservice.sm*
- *gwapre-ntptls.sm*

3. Authentisierung des GWA am Sicherheitsmodul: Aufruf der Methode *getAUTHChallenge* und *responseAUTHChallenge (Detailbeschreibung der Methoden in* Tabelle 74*)*
4. Einspielen der öffentlichen Schlüssel des neuen Gateway Administrators in Form von Import-Zertifikaten ins Sicherheitsmodul: Aufruf der Methode *writeImportCertificate (Detailbeschreibung der Methoden in* Tabelle 74)
5. Authentisierung des GWA am Sicherheitsmodul beenden: Aufruf der Methode *terminateAUTH (Detailbeschreibung der Methoden in* Tabelle 74*)*
6. Wechsel des Gateway Administrators beauftragen: Aufruf der Methode *switchGwa (Detailbeschreibung der Methode in* Tabelle 86*)*

Anschließend versendet das SMGW ein Event an den neuen Gateway Administrator, um diesen über den Wechsel zu informieren. Weiterhin wird der Gateway Administrator per Event über die IP-Adresse des zu ihm gewechselten SMGWs informiert. Der neue Gateway Administrator muss daraufhin ein Wake-Up-Paket an das SMGW versenden und initiiert damit einen Management-Kanal zwischen dem SMGW und dem neuen Gateway Administrator. Der neue Gateway Administrator muss die folgenden Schritte durchführen, um den Wechsel fortzuführen. Zunächst wird die Authentisierung am Sicherheitsmodul herbeigeführt und wieder beendet, um den erfolgreichen Wechsel des AUTH-Schlüssels zu testen:

1. Authentisierung des GWA am Sicherheitsmodul: Aufruf der Methode *getAUTHChallenge* und *responseAUTHChallenge* (Detailbeschreibung der Methoden in Tabelle 74)
2. Authentisierung des GWA am Sicherheitsmodul beenden: Aufruf der Methode *terminateAUTH* (Detailbeschreibung der Methoden in Tabelle 74)

Wenn dieser Schritt durchlaufen ist, gilt der Wechsel des Gateway Administrators als erfolgreich. Ein Zurückfallen zum alten Gateway Administrator ist im Anschluss daran ausgeschlossen. Der neue Gateway Administrator muss anschließend die folgenden Schritte durchführen:

1. Einspielen der Kommunikationsprofile des neuen Gateway Administrators: Aktualisieren der Klasse *kaf_han_wan_container* für Management, Admin-Service und NTP-TLS (Detailbeschreibung der IC in Tabelle 24). Hierfür sind die folgenden Logical Device-IDs zu verwenden:
   - *gwa-management.sm*
   - *gwa-adminservice.sm*
   - *gwa-ntptls.sm*

Nachdem die Kommunikationsprofile des neuen Gateway Administrators aufgespielt sind, unterbricht dieser den bestehenden Management Kanal und initiiert diesen neu, indem er ein Wake-Up-Paket an das SMGW versendet. Um den Gateway-Administrator-Wechsel abzuschließen führt der neue Gateway Administrator folgende Schritte durch:

1. Authentisierung des GWA am Sicherheitsmodul: Aufruf der Methode *getAUTHChallenge* und *responseAUTHChallenge (Detailbeschreibung der Methoden in* Tabelle 74*)*
2. Schlüssel des alten Gateway Administrators auf dem Sicherheitsmodul deaktivieren: Aufruf der Methode *deactivateGWA_oldKeys (Detailbeschreibung der Methoden in* Tabelle 74*)*
3. Authentisierung des GWA am Sicherheitsmodul beenden: Aufruf der Methode *terminateAUTH (Detailbeschreibung der Methoden in* Tabelle 74*)*

4.4.5 Verwaltung der Logdaten

Für Logdaten werden im SMGW die folgenden Klassen verwendet:

Tabelle 78: Überblick über die COSEM-ICs zur Verwaltung der Logdaten

Name der COSEM-IC	Aufgabe
profile_generic (in der Nutzungsvariante *Logbook)*	Wird verwendet, um Logeinträge aufzunehmen und zu übertragen. Dabei wird die Datenstruktur sowohl bei Push- als auch bei Pull-Abfragen des GWA (Logs bzw. Alarme) verwendet.
log_entry	Hilfsklasse für *profile_generic* (in der Nutzungsvariante *Logbook)*
event	Hilfsklasse für *log_entry* in der Nutzungsvariante *Logbook*

IC *profile_generic* (in der Nutzungsvariante *Logbook*)

Instanzen der Klasse *profile_generic* dienen in der Nutzungsvariante *Logbook* der Speicherung von Logbuch-Einträgen sowie ihrer späteren Übermittlung an den GWA. Diese Klasse kennt keine Methoden. Die Attribute dieser Klasse sind wie folgt:

Tabelle 79: Attribute der IC *profile_generic in der Nutzungsvariante Logbook*

Name des XML-Elements			XML-Datentyp	Beschreibung
logical_name			***string***	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Der Wert entspricht dem Schema ‚OBIS.LogicalDevice-SMGW.sm' und hat eine maximale Länge von 255 Zeichen.
capture_objects	*capture_object*	*class_id*	unsignedShort	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *log_entry*. Der Wert des Attributs ‚logical_name' entspricht dem Schema ‚*01005e3180a3*.LogicalDevice-SMGW.sm'. Der Wert des Attributs ‚class_id' entspricht dem Wert ‚32819', der des Attributs ‚class_version' dem Wert ‚0'. Die Attribute ‚attr_reference' und ‚index_reference' werden mit ausgegeben und entsprechen dem Datentyp. Es werden jedoch unabhängig von diesen Angaben stets alle Attribute der Hilfsklasse ‚log_entry' sowie der Klasse ‚event' ausgegeben.
		class_version	unsignedByte	
		attr_reference	byte	
		index_reference	unsignedInt	
buffer	*logbook*	*entry*	*TYPE_entry*	Inkludierung (nicht Referenzierung) eines Objekts vom Typ *log_entry* (Definition s. dort)

IC *log_entry*

Instanzen der Klasse *log_entry* dienen der Beschreibung eines einzelnen Logbuch-Eintrags. Diese Klasse kennt keine Methoden. Die Attribute dieser Klasse sind wie folgt:

Tabelle 80: Attribute der IC *log_entry*

Name des XML-Elements	XML-Datentyp	Beschreibung
record_number	unsignedInt	Ein-eindeutiger Bezeichner (dieser wird mit Ablegen des Eintrags im Logbuch durch das SMGW erzeugt). Der Wert entspricht dem Datentyp und gibt die Nummer des Eintrags im Logbuch an.
parent_record_number	unsignedInt	Dient der Markierung verketteter Ereignisse. Enthält die „*Parent Record Number*“ denselben Wert, wie die „*Record Number*“, so handelt es sich um das Wurzel-Element einer Kette von zusammenhängenden Ereignissen oder um ein Einzel-Ereignis. Der Wert enthält stets denselben Wert wie ‚record_number‘.
repetition_counter	unsignedInt	Der „Repetition Counter“ ist bei einem neu auftretenden Ereignis mit „1“ zu besetzen. Bei nachfolgenden Ereignissen kann der „Repetition Counter“ benutzt werden, um das mehrfache Einfügen der Ereignisse in einem Logbuch zu vermeiden. Der Wert wird entsprechend der Vorgaben des Datentyps ausgegeben.
Event	TYPE_event	Inkludierung (nicht Referenzierung) eines Objekts vom Typ *event* (Definition s. dort).

Im SMGW sind zwei Instanzen der Klasse *profile_generic* in der Nutzungsvariante *Logbook* vorhanden. Die zwei Instanzen können über ihre OBIS-Kennzahl identifiziert werden. Die beiden OBIS Kennzahlen lauten wie folgt:

- *Systemlog:* *0000636201ff*
- *Eichlog:* *0000636202ff*

Zur Eingrenzung der Logbuchabfrage können folgende Parameter verwendet werden:

- *q.fromtime*
- *q.totime*
- *q.fromidx*
- *q.count*

Die Parameter *q.fromtime* und *q.totime* können in Lokal- oder in UTC-Zeit angegeben werden. Bei einer Angabe in Lokalzeit muss das Schema "YYYY-MM-DD.hh.mm.ss" verwendet werden. Bei einer Angabe in UTC-Zeit muss das Schema "YYYY-MM-DDThh:mm:ssZ" verwendet werden.

Wenn keine Parameter bei der Abfrage angegeben werden, werden, absteigend sortiert, maximal die letzten 1000 Einträge des jeweiligen Logbuchs zurückgegeben.

Im Folgenden ist das zu erwartende Antwortverhalten für bestimmte Parameterkombinationen beschrieben:

- *q.fromtime, q.totime:* Alle Einträge innerhalb der angegebenen Zeitspanne werden zurückgegeben
- *q.fromtime, q.count:* Beginnend von *q.fromtime* werden aufsteigend die nächsten *q.count* Einträge zurückgegeben
- q.fromidx, q.count: Beginnend von *q.fromidx* werden aufsteigend die nächsten *q.count* Einträge zurückgegeben

Wenn die Parameter bei der Abfrage so angegeben werden, dass das Ergebnis mehr als 1000 Logeinträge erhält, wird eine Fehlermeldung zurückgegeben. Die Parameter müssen so angepasst werden, dass die Antwort maximal 1000 Einträge enthält, damit eine gültige Antwort generiert wird.

Der Gateway Administrator kann jedes der beiden Logbücher mittels des HTTP-Verbs GET auslesen. Die Instanzen der Klasse sind der Logical Device ID *ldevid* des SMGW zugeordnet.

Tabelle 81: Auslesen des Systemlogs

Zugriffs-Dienst	HTTP-Verb	Beispiel URI zum Auslesen des Systemlogs	Request-Body	HTTP Response Statuscode	Response-Body
Auslesen der bestehenden Klasseninstanz	GET	<PoC>/cosem/ldevs/eppc0110123456.sm/objects/0000636201ff	---	200	*profile_generic* in der Nutzungsvariante *Logbook*

Tabelle 82: Auslesen des Eichlogs

Zugriffs-Dienst	HTTP-Verb	Beispiel URI zum Auslesen des Eichlogs	Request-Body	HTTP Response Statuscode	Response-Body
Auslesen der bestehenden Klasseninstanz	GET	<PoC>/cosem/ldevs/eppc0110123456.sm/objects/0000636202ff	---	200	*profile_generic* in der Nutzungsvariante *Logbook*

Um nur eine bestimmte Anzahl an Logeintragen abzufragen, kann die Abfrage einer begrenzten Anzahl an Logeinträgen verwendet werden. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 83: Auslesen eines Logbuchs mit Verwendung von Query-Parametern

Zugriffs-Dienst	HTTP-Verb	Beispiel URI zum Auslesen bestimmter Einträge des Eichlogs	Request-Body	HTTP Response Statuscode	Response-Body
Auslesen einer bestimmten Anzahl an Logeinträgen	GET	<PoC>/cosem/ldevs/eppc0110123456.sm/objects/0000636202ff/?q.fromidx=0&q.count=1	---	200	*profile_generic* in der Nutzungsvariante *Logbook*

IC *event*

Instanzen der Klasse *event* dienen der Beschreibung eines einzelnen Events. Diese Klasse kennt keine Methoden. Die Attribute dieser Klasse sind wie folgt:

Tabelle 84: Attribute der IC *event*

Name des XML-Elements		XML-Datentyp	Beschreibung
seconds_index		unsignedInt	Anzahl der Sekunden seit Inbetriebnahme des SMGW. Der Wert des Attributs ist immer auf ‚0' gesetzt.
timestamp		dateTime	Zeitangabe des Events gemäß dem Muster „*yyyy-mm-dd*T*hh:mm:ssZ*" (mit 4-stelliger Abgabe des Jahres *yyyy*, 2-stelliger Angabe des Monats mm, 2-stelliger Angabe des Tages dd, 2-stelliger Angabe des Tages dd, 2-stelliger Angabe der Stunden *hh*, 2-stelliger Angabe der Minuten mm und 2-stelliger Angabe der Sekunden *ss*. Der Wert wird immer als UTC Zeit angegeben. Die Angabe wird durch das ‚Z' am Ende gekennzeichnet.
level		string	Textuelle Angabe des Ereignis-Typs. Auswahl zwischen den Werten „INFO" (für informelle Events), „WARNING" (für warnende Events), „ERROR" (für Fehler beschreibende Events), „FATAL" (für fatale Fehler beschreibende Events) oder „EXTENSION" (für Events mit erweiterter Fehlerbeschreibung)
type	*version*	unsignedShort	Version der Logmeldung. Der Wert des Attributs wird als default auf ‚1' gesetzt.

Name des XML-Elements		XML-Datentyp	Beschreibung
	length	unsignedShort	Länge der Logmeldung. Der Wert des Attributs wird als default auf ‚0' gesetzt.
	device_type	unsignedShort	Device Type, von dem die Logmeldung stammt. Der Wert des Attributs wird als default auf ‚1' gesetzt.
	module	unsignedShort	Angabe des SMGW-Moduls. Der Wert des Attributs wird als default auf ‚0' gesetzt.
	function	unsignedShort	Angabe der funktionalen Einheit in einem SMGW-Modul. Der Wert des Attributs wird als default auf ‚0' gesetzt.
	vendor_id	string	Angabe der SMGW-Herstellerkennung. Der Wert des Attributs ist immer auf ‚PPC' gesetzt.
	event_id	unsignedInt	Angabe der Kennung des Events. Der Wert des Attributs gibt an, um welche Logmeldung es sich handelt.
	event_sub_id	unsignedInt	Angabe der Subkennung des Events. Der Wert des Attributs wird als default auf ‚0' gesetzt.
subject_identity		unsignedInt	Identität der Quelle, die das Ereignis ausgelöst hat. Der Wert des Attributs wird als default auf ‚0' gesetzt.

Name des XML-Elements		XML-Datentyp	Beschreibung
user_id entity	*logical_name*	string	Das Attribut kann vom SMGW gesetzt werden, wenn die Meldung einem konkreten User zugeordnet ist. Der Wert des Attributs ‚logical_name' muss dem Schema ‚01005e318011.LogicalDevice-Letztverbraucher. sm", das Attribut ‚class_id' dem Wert ‚32795' und das Attribut ‚class_version' dem Wert ‚0' entsprechen.
	class_id	unsignedShort	
	class_version	unsignedByte	
outcome		string	Textueller Bezeichner des Ergebnisses der mit dem Ereignis verbundenen Aktion; Auswahl zwischen den Werten „SUCCESS" (Ergebnis: erfolgreich), „FAILURE" (Ergebnis: fehlerhaft) oder „EXTENSION" (erweiterter Meldungstext).
message_extensions	*message_extension*	string	*message_extensions* Die Werte identifizieren eine Erweiterung der Logmeldung. Wenn die Logmeldung sich beispielsweise auf das Hinzufügen eines neuen Zählers bezieht, enthält dieses Attribut die Zählernummer des neuen Zählers.
Destination		hexBinary oder anyURI	Falls *hexBinary*, dann mit max. 4 Byte (IPv4), falls *anyURI*, dann mit max. 1000 Zeichen. Das Attribut kann vom SMGW gesetzt werden, wenn der Wert einem bestimmten Ziel zugeordnet werden kann.

Name des XML-Elements	XML-Datentyp	Beschreibung
Evidence	hexBinary	Enthält die Signatur über den jeweiligen Logeintrag. Die Signatur wird über folgende Attribute gebildet: - *timestamp* - *event_id* - *event_sub_id* - *vendor_id* - *message_extensions* - *outcome* Das SMGW erstellt dabei eine ECDSA-Signatur mit dem Algorithmus *brainpoolP256r1* gemäß RFC 5639.

Für diese Instanz wird die folgende OBIS-Kennzahl verwendet:

- *event:* *01005e3180a3*

4.4.6 Funktionen des Smart Meter Gateways

Das SMGW bietet dem Gateway Administrator die Möglichkeit einen Neustart und einen Selbsttest auszulösen. Hierzu wird die IC *smgw_info* genutzt.

IC *smgw_info*

Instanzen der Klasse *smgw_info* dienen der Beschreibung von Informationen über das jeweilige SMGW. Die Attribute dieser Klasse sind wie folgt:

Tabelle 85: Attribute der IC *smgw_info*

Name des XML-Elements	XML-Datentyp	Beschreibung
logical_name	string	(read-only) OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Der Wert entspricht immer ‚01005e3180a0.LogicalDevice-SMGW.sm‘.
manufacturer	String	(read-only) Enthält die dreistellige Herstellerkennzeichnung. Der Wert ist immer auf ‚PPC‘ gesetzt.
batch	String	(read-only) Gibt die Produktionscharge des SMGW an. Der Wert ist immer auf ‚02‘ gesetzt.

Name des XML-Elements	XML-Datentyp	Beschreibung
fw_versions	TYPE_fw_version	(read-only) *Array TYPE_fw_version* *TYPE_fw_version := Structure {* *version string* *active boolean* *}* In der vorliegenden SMGW wird immer nur ein Array mit einem Eintrag ausgegeben, der die derzeit aktive Firmware-Version angibt. Der Wert des Attributs ‚version' enthält zwei 5-stellige Zahlen, die durch einen Bindestrich getrennt sind. Durch die erste 5-stellige Zahl wird die Version des installierten Bootstreams (Bootloader und Betriebssystem), durch die zweite die Version der Applikationssoftware angegeben. Der Wert des Attributs ‚active' ist immer auf ‚true' gesetzt.
ipv4_address	String	(read-only) Gibt die aktuelle IP Adresse des SMGW an der WAN Schnittstelle an. Der Wert entspricht einer gültigen IPv4 Adresse entsprechen (vier Blöcke, durch ‚.' getrennt, je 8 Bit, d.h. Wertebereich von 0 -255).
smgw_id	String	(read-only) Gibt die kanonische Geräte ID des SMGW an. Der Wert entspricht dem folgenden Schema ‚GeräteID.sm' und hat eine Länge von 17 Zeichen.

Die Methoden dieser Klasse sind wie folgt:

Tabelle 86: Methoden der IC *smgw_info*

Name der Methode	Beschreibung
reset	Löst einen Neustart des SMGW aus.
selfTest	Löst einen Selbsttest des SMGW aus.
switchGwa (switch_time_max)	Die Methode löst den Aufbau eines Admin-Service Kanals zum neuen GWA aus, um diesen mittels eines Events über den durchgeführten GWA-Wechsel zu informieren. Falls die Übermittlung des Events zum neuen GWA fehlschlägt, ist der alte GWA mit einem Event darüber in Kenntnis zu setzen. Darüber hinaus werden alle bestehenden Kanäle zum alten GWA geschlossen. *switch_time_max* := Unsigned32 Max: 1 209 000 (14 Tage) Gibt die Zeit in Sekunden an, die maximal ohne bestehende Verbindung zum neuen GWA zur Verfügung steht, um den GWA Wechsel erfolgreich durchzuführen.

Name der Methode	Beschreibung
setUserLogEntryMaxCount (max_entries)	Mit dieser Methode wird die maximale Anzahl der Einträge für alle Letztverbraucher Logbücher global auf dem SMGW festgelegt. *max_entries* := Unsigned32 Min: 30 000 Max: 120 000 Der Wert wird nur übernommen, wenn er größer ist als die maximale Anzahl der bestehenden Logeinträge in einem der Letztverbraucher-Logbücher auf dem SMGW. Ein Löschen von bestehenden Logeinträgen im Smart Meter Gateway ist durch diesen Mechanismus ausgeschlossen. Um den neuen Wert zu aktivieren, muss im Anschluss die Methode ‚reset' der Klasse ‚*smgw_info*' aufgerufen werden.

Im SMGW ist eine Instanz der Klasse *smgw_info* vorhanden. Die Instanz kann über ihre OBIS-Kennzahl identifiziert werden. Die OBIS Kennzahl lautet wie folgt:

- *smgw_info:* *01005e3180a0*

Die Instanzen der Klasse sind der Logical Device ID *ldevid* des SMGW zugeordnet.

Der GWA kann die Instanz der Klasse aus dem SMGW auslesen. Das Auslesen erfolgt mittels des HTTP-Verbs GET. Das folgende Beispiel verdeutlicht die Nutzung:

Tabelle 87: Auslesen des IC *smgw_info*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für das Auslesen der Klasse *smgw_info*	Request-Body	HTTP Response Statuscode	Response-Body
Auslesen einer bestehenden Instanz	GET	<PoC>/cosem/ldevs/eppc0110123456.sm/objects/01005e3180a0	---	200	*smgw_info*

Der Gateway Administrator kann jede der beiden Methoden der Klasse mittels des HTTP-Verbs POST aufrufen. Das folgende Beispiel verdeutlicht die Nutzung der *reset*-Methode:

Tabelle 88: Ausführen der Methode *reset* der IC *smgw_info*

Zugriffs-Dienst	HTTP-Verb	URI zum Auslösen des Reset	Request-Body	HTTP Response Statuscode	Response-Body
Aufruf der Methode	POST	<PoC>/cosem/ldevs/eppc0110123456.sm/objects/01005e3180a0/methods/reset	---	200	---

Das folgende Beispiel verdeutlicht die Nutzung der *selfTest*-Methode:

Tabelle 89: Ausführen der Methode *selfTest* der IC *smgw_info*

Zugriffs-Dienst	HTTP-Verb	URI zum Auslösen des Selbsttests	Request-Body	HTTP Response Statuscode	Response-Body
Aufruf der Methode	POST	<PoC>/cosem/ldevs/eppc0110123456.sm/objects/01005e3180a0/methods/selfTest	---	200	---

4.5 WKS2: „ADMIN-SERVICE“-Webservice-Schnittstelle

Das SMGW nutzt einen Dienst beim GWA, um tarifierte Messwerte, Netzzustandsdaten oder Benachrichtigungen an den GWA auszuliefern, die dieser dann an einen EMT weiterleitet. Das WAN-Kommunikationsszenario, dem dieses Kommunikationsmuster zugrunde liegt, wird im Folgenden als WKS2 oder „ADMIN-SERVICE“ bezeichnet. Dieser Webservice unterstützt ausschließlich die in diesem Kapitel genannten Administrationsbefehle. Diese Administrationsbefehle werden dabei durch die vom SMGW-Hersteller gelieferten XML-Schema-Dateien aus der Archivdatei *SMGW-XSD_v.1.0.zip* festgelegt.

4.5.1 WAF2: Zugriff auf Dienste des GWA

Zur Auslieferung von Messwerten und Netzzustandsdaten über WKS2 „ADMIN-SERVICE“ werden im SMGW folgende COSEM-ICs durch den GWA verwendet:

Tabelle 90: Überblick über die COSEM-ICs in WAF2 „Zugriff auf Dienste des GWA“

Name der COSEM-IC	Aufgabe
profile_generic	Wird in der Nutzungsvariante „*Simple Data*“ verwendet, um Mess- und Registerwerte zu übertragen.
gateway_signed_extended_register	Hilfsklasse im *Profile Generic,* Nutzungsvariante „*Simple Data*“.

Bei der Beschreibung der Klassen wurde auf die Angabe „read-only“ verzichtet, da alle Daten an den GWA gesendet werden und somit nicht ins SMGW geschrieben oder aktiv ausgelesen werden.

IC *profile_generic* (in der Nutzungsvariante *Simple Data*)

Instanzen der Klasse *profile_generic* beschreiben in der Nutzungsvariante *Simple Data* zur Übertragung von Messwerten an einen Gateway Administrator. Diese Klasse besitzt keine Methoden. Die Parameter sind wie folgt:

Tabelle 91: Attribute der IC *profile_generic* in der Nutzungsvariante *Simple Data*

Name des XML-Elements				XML-Datentyp	Beschreibung
logical_name				string	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Der Wert entspricht immer ‚OBIS.LogicalDevice-TAF.sm' oder OBIS.LogicalDevice-TAF.sm', wobei ‚OBIS' entsprechend Kapitel 9 zu setzen ist.
capture_objects	*capture_object*	*logical_name*		string	Objekt-Referenz (in Form des „Logical Device Name", der Klassen-ID und Klassen-Version) der COSEM-IC *gateway_signed_extended_register*. Die Werte der Attribute ergeben sich aus den übermittelten Messwerten.
		class_id		unsignedShort	
		class_version		unsignedByte	
		attr_reference		byte	
		index_reference		unsignedInt	
buffer	*simple_data*	*column*	*entry*	*TYPE_extended_register*	Inkludierung (nicht Referenzierung) eines Objekts vom Typ *gateway_signed_extended_register*. Definition s. dort.
capture_period				unsignedInt	Aktualisiert die Messwertliste, aktualisiert die abgeleiteten Register (Registrierperiode) und wird in Sekunden angegeben. Das Attribut wird in der vorliegenden SMGW Version nicht genutzt.

IC gateway_signed_*extended_register*

Instanzen der Klasse *gateway_signed_extended_register* dienen der Modellierung eines Prozess-Werts mit seinen zugehörigen Informationen bzgl. der Skalierung, Einheit, Status und Zeitpunkt der Erfassung des Registerwerts. Diese Klasse kennt keine Methoden. Die Attribute dieser Klasse sind wie folgt:

Tabelle 92: Attribute der IC *gateway_signed_extended_register*

Name des XML-Elements		XML-Datentyp	Beschreibung
logical_name		String	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse. Der Wert entspricht immer dem Schema ‚OBIS.LogicalDevice.sm' und hat eine maximale Länge von 255 Zeichen.
value	*bit-string* oder *double-long* oder *double-long-unsigned* o. *octet-string* oder *visible-string* oder *UTF8-string* oder *Integer* oder *long* oder *unsigned* oder *long-unsigned* oder *long64* oder *long64-unsigned* oder	string oder int oder unsignedInt oder string oder string oder string oder byte oder short oder unsignedByte oder unsignedShort oder long oder unsignedLong oder	Enthält den aktuellen Prozess- bzw. Status-Wert. Im Fall des Vorhandenseins des Elements *bit-string* gemäß dem Muster „[0-1]{1,}", im Fall des Vorhandenseins des Elements *visible-string* mit der Maximallänge 255. Als Attribut wird immer ‚double-long' genutzt.

Name des XML-Elements		XML-Datentyp	Beschreibung
	float32 oder *float64*	float oder double	
Scaler		Byte	Angabe des Skalierungsfaktors
Unit		unsignedByte	Angabe der Maßeinheit der im Register vorliegenden Größe. Der Wert des Attributs ‚unit' entspricht dem Wert aus IEC-62056-6-2, Seite 31, Tabelle 3.
status	*bit-string* oder *double-long-unsigned* o. *octet-string* oder *visible-string* oder *UTF8-string* oder *unsigned* oder *long-unsigned* oder *long64-unsigned*	string oder unsignedInt oder string oder string oder string oder unsignedByte oder unsignedShort oder unsignedLong	Textuelle Angabe der Statusinformationen dieses *gateway_signed_extended_register*, im Fall des Vorhandenseins des Elements *bit-string* gemäß dem Muster „[0-1]{1,}", im Fall des Vorhandenseins des Elements *visible-string* mit der Maximallänge 255 Zeichen. Als Attribut wird immer ‚unsigned' genutzt.
capture_time		dateTime	Angabe des Zeitpunkts der Erfassung des Registerwerts. Der Wert entspricht dem Schema „yyyy-mm-ddThh:mm:ssZ" (mit 4-stelliger Abgabe des Jahres *yyyy*, 2-stelliger Angabe des Tages *dd*, 2-stelliger Angabe der Stunde *hh*, 2-stelliger Angabe der Minuten *mm*, 2-stelliger Angabe der Stunden *hh* und 2-stelliger Angabe der

Name des XML-Elements	XML-Datentyp	Beschreibung
		Sekunden *ss*) entsprechen. Der Wert wird als UTC Zeit mit einem ‚Z' am Ende angegeben.
smgw_signature	hexBinary	Innere Signatur zu jedem Messwert. Die innere Signatur wird über folgende Attribute gebildet: - *logical_name* - *status* - *capture_time* - *value* - *scaler* - *unit* Das SMGW erstellt dabei eine ECDSA-Signatur mit dem Algorithmus *brainpoolP256r1* gemäß RFC 5639.

Im SMGW werden Messwerte gebildet und anschließend als *profile_generic in der Nutzungsvariante Simple Data* versendet. Dabei kommen die folgenden OBIS-Kennzahlen zum Einsatz:

- *Profile_generic: siehe Kapitel 9*
- Die OBIS-Kennzahlen der einzelnen *gateway_signed_extended_register* ergeben sich aus den Messwerten, die enthalten sind.

Das *profile_generic in der Nutzungsvariante Simple Data* ist der Logical Device ID des TAF zugeordnet, durch den es gebildet wurde. Die einzelnen *gateway_signed_extended_register* sind der Logical Device ID des TAF oder der Logical Device ID des Zählers zugeordnet. Die Messwerte, die nur transparent durchgereicht werden, sind der Logical Device ID des Zählers zugeordnet. Die Messwerte, die vom TAF gebildet werden (abgeleitete Register), sind der Logical Device ID des TAF zugeordnet.

Tabelle 93: Übertragung von Messdaten an den GWA/EMT

Zugriffs-Dienst	HTTP-Verb	URI zur Übertragung der Messwerte	Request-Body	HTTP Response Statuscode	Response-Body
Übergabe der Messwerte vom SMGW an den GWA/EMT	POST	URI aus Kommunikationsprofil	*profile_generic* in der Nutzungsvariante *Simple Data*	200	

4.5.2 WAF3: Alarmierung und Benachrichtigung

Der WAN-Anwendungsfall WAF3 dient gemäß [BSI TR-03109-1] dazu, unerwartete Ereignisse oder Fehlersituationen, die während des Betriebs des SMGW auftreten, zur Analyse und weiteren Bearbeitung an den GWA zu melden.

Zur Übermittlung von derlei Nachrichten greift das SMGW auf einen WAN-Service des GWA gemäß Kommunikationsszenario „ADMIN-SERVICE“ (WKS2) zu (vgl. Kap. 4.5), der die Zustellung solcher Ereignisse durch das SMGW ermöglicht.

Für Alarmierung und Benachrichtigung des GWA werden im SMGW die folgenden Klassen verwendet:

Tabelle 94: Überblick über die COSEM-ICs in WAF3 „Alarmierung und Benachrichtigung“

Name der COSEM-IC	Aufgabe
profile_generic (in der Nutzungsvariante *Logbook)*	Wird dazu verwendet, Logeinträge abzubilden und zu übertragen. Dabei wird die Datenstruktur sowohl bei Push- als auch bei Pull-Abfragen des GWA (Logs bzw. Alarme) verwendet.

Name der COSEM-IC	Aufgabe
log_entry	Hilfsklasse für die COSEM-IC *profile_generic* (in der Nutzungsvariante *Logbook)*
event	Hilfsklasse für die COSEM-IC *log_entry* (in der Nutzungsvariante *Logbook*)

Die für den Versand eines Events verwendete Objektstruktur ist somit identisch zu der in 4.4.5 dargestellten Struktur für das Verwalten von Logdaten. Es wird nur ein Element in ‚log_entry' angegeben.

Zum Versand von Alarmen werden die folgenden OBIS-Kennzahlen verwendet:

- *Systemlog:* *0000636201ff*
- *Eichlog:* *0000636202ff*
- *event:* *01005e3180a3*

Die Übertragung an den Gateway Administrator erfolgt mittels des HTTP-Verbs POST. Die folgende Tabelle verdeutlicht die Übertragung der Alarme an den GWA:

Tabelle 95: Übertragen von Alarmen an den GWA

Zugriffs-Dienst	HTTP-Verb	URI zur Übertragung von Events	Request-Body	HTTP Response Statuscode	Response-Body
Übergabe der Alarme an den GWA	POST	URI aus Kommunikationsprofil	*profile_generic* (in der Nutzungsvariante *Logbook)*	200	

4.5.3 WAF4: Auslieferung von Messwerten und Netzzustandsdaten

Der Vorgang der Auslieferung von Messwerten und Netzzustandsdaten wird vollständig im Kapitel

WAF2: Zugriff auf Dienste des GWA“ beschrieben.

4.6 WKS3: „Info-Report“-Schnittstelle zu einem EMT

Bei der Übertragung von Daten des SMGW an einen externen Marktteilnehmer (EMT), also dem WAN-Kommunikationsszenario WKS3 „INFO-REPORT“, tritt der WAN-Anwendungsfall WAF5 („Übertragung von Daten an EMT“) mit den folgenden Funktionalitäten auf:

- **Turnusmäßige Auslieferung von tarifierten Messwerten:** Das SMGW liefert gemäß eines Auswertungs- und Kommunikationsprofils regelmäßig abrechnungsrelevante Messwerte zur Tarifierung an einen EMT aus.
- **Turnusmäßige Netzzustandsdatenauslieferung**: Das SMGW liefert gemäß eines Auswertungs- und Kommunikationsprofils regelmäßig Messwerte zum Netzzustand an einen EMT aus.
- **Spontane Messwertauslesung:** Ein EMT hat keinen direkten Zugriff auf die Daten des SMGW. Daher wird die Spontanablesung dadurch nachgebildet werden, dass der GWA ein geeignetes Auswertungs- und Kommunikationsprofil in das SMGW einbringt (falls noch nicht vorhanden), das die Auslieferung der benötigten Messwerte an den EMT auslöst. Das anschließende WAN-Kommunikationsverhalten entspricht dann einer Weitergabe von Messwerten wie bei einer turnusmäßigen Auslieferung.

4.6.1 WAF5: Übertragung von Daten an externe Marktteilnehmer

Das SMGW übergibt die Daten an die bei einem EMT betriebene Dienstschnittstelle, die im Kommunikationsprofil angegeben ist, die die zuverlässige Auslieferung durch das SMGW ermöglicht.

Zur Auslieferung von Messwerten und Netzzustandsdaten werden im SMGW vom GWA an der WAN-Schnittstelle folgende COSEM-ICs verwendet:

Tabelle 96: Überblick über die COSEM-ICs in WAF5 „Übertragung von Daten an EMT“

Name der COSEM-IC	Aufgabe
profile_generic	Wird in der Nutzungsvariante „*Simple Data*“ verwendet, um mess- und Registerwerte zu übertragen.
gateway_signed_extended_register	Hilfsklasse im *profile_generic* in der Nutzungsvariante „*Simple Data*“.

Diese COSEM-ICs sind schon vollständig in WAF2 beschrieben; der Unterschied zwischen WKS2 und WKS3 besteht lediglich im Ziel der auszuliefernden Daten, welches bei WKS2 der GWA ist, während dies in WKS3 ein EMT ist.

4.6.2 Statuswort der übermittelten Messwerte

Im Attribut „status“ der COSEM-IC *gateway_signed_extended_register* wird zu jedem Messwert ein Statuswort übertragen.

Das Statuswort setzt sich aus dem Statuswort des SMGWs und dem Statuswort des Zählers zusammen. Insofern ergibt sich folgende Zusammensetzung.

Statuswort:= SMGW-Statuswort gefolgt von Zähler-Statuswort

Sowohl das SMGW-Statuswort als auch das Zähler-Statuswort bestehen aus jeweils 32 Bit, so dass sich eine Gesamtlänge von 64 Bit für das Statuswort, das mit jedem Messwert übermittelt wird, ergibt. Das Statuswort wird in hexadezimaler Form übertragen und wird binär jeweils mit dem Least-Significant-Bit zuerst dargestellt.

Das Zähler-Statuswort wird vom Zähler mit den Messwerten mitgeliefert und wird unverändert an den EMT übermittelt. Das Zähler-Statuswort folgt dabei der Beschreibung aus [FNN_LH_BZ].

Das SMGW-Statuswort wird wie folgt belegt:

Tabelle 97: Bedeutung der Bits des SMGW-Statusworts

Bit-Position	Mögliche Werte	Bedeutung
0	Immer ‚1'	Statuswort-Identifikation (Least-Significant-Bit)
1	Wird aktuell nicht genutzt, daher immer ‚0'	Abrechnungsrelevanz
2	Immer ‚1'	Statuswort-Identifikation
3	Immer ‚0'	Statuswort-Identifikation
4	Immer ‚0'	Statuswort-Identifikation
5	Immer ‚0'	Statuswort-Identifikation
6	Immer ‚0'	Statuswort-Identifikation
7	Immer ‚0'	Statuswort-Identifikation
8	Wird aktuell nicht genutzt, daher immer ‚0'	Fataler Fehler
9	Wird aktuell nicht genutzt, daher immer ‚0'. Im Falle einer ungültigen Systemzeit stellt das SMGW den Messbetrieb ein und versendet/erfasst keine Messwerte mehr.	Systemzeit ist ungültig
10	Immer ‚0'	Reserviert
11	Immer ‚0'	Reserviert
12	Wird aktuell nicht genutzt, daher immer ‚0'	Warnung

13	‚0' wenn ein Messwert vorhanden ist ‚1' wenn zu diesem Registrierzeitpunkt kein Messwert erfasst werden konnte	Messwert vorhanden oder nicht vorhanden
14	‚0' wenn der Messwert gültig ist ‚1' wenn der Messwert nicht gültig ist (mechanische/magnetische Manipulation oder fataler Fehler des Zählers liegen vor	Messwert gültig oder nicht gültig
15	Immer ‚0'	Reserviert
16	‚0' wenn der Messwert keinen Versand durch eine Schwellwertüber- oder unterschreitung ausgelöst hat ‚1' wenn der Messwert aufgrund einer Schwellwertüber- oder unterschreitung den Messwertversand ausgelöst hat	Reserviert
17	Immer ‚0'	Reserviert
18	Immer ‚0'	Reserviert
19	Immer ‚0'	Reserviert
20	Immer ‚0'	Reserviert
21	Immer ‚0'	Reserviert
22	Immer ‚0'	Reserviert
23	Immer ‚0'	Reserviert
24	Immer ‚0'	Reserviert
25	Immer ‚0'	Reserviert
26	Immer ‚0'	Reserviert
27	Immer ‚0'	Reserviert
28	Immer ‚0'	Reserviert
29	Immer ‚0'	Reserviert
30	Immer ‚0'	Reserviert
31	Immer ‚0'	Reserviert

Für die Bildung der inneren Signatur gemäß Tabelle 92 werden aus dem Statuswort nur die Bits 13 und 14 des SMGW-Statusworts herangezogen. Abhängig von der Belegung der beiden Bits werden dabei folgenden Werte bei der Signaturbildung verwendet:

- Bit 13 und Bit 14 sind mit dem Wert ‚0' belegt ‚0'
- Bit 13 ist mit dem Wert ‚1' und Bit 14 ist mit dem Wert ‚0' belegt ‚3'
- Bit 13 ist mit dem Wert ‚0' und Bit 14 ist mit dem Wert ‚1' belegt ‚4'

4.7 WKS6: CLS-Tunnel-Funktionalität (WAF6)

Das SMGW bietet eine Proxy-Funktionalität gemäß [RFC 1928] zur Kommunikation zwischen einem externen Marktteilnehmer (EMT) im WAN und einem Controllable Local System (CLS) im HAN**.** Über die CLS-Schnittstelle des SMGW können steuerbare Komponenten im HAN des Anschlussnutzers gesicherte Kommunikationsverbindungen ins WAN zu einem EMT aufbauen. Das SMGW stellt dazu seine TLS-Funktionalität gemäß [RFC 5246] zur Verfügung. Das SMGW lässt an der CLS-Schnittstelle mit seinen beiden Endpunkten im HAN und im WAN lediglich eine bidirektionale Authentisierung mittels TLS-Zertifikaten zu.

Die dann über diese TLS-Verbindung ablaufende Kommunikation und die dazugehörigen Protokolle, die dem Monitoring oder der Steuerung der CLS-Komponente dienen, sind für das SMGW transparent. Die über diesen Transportkanal laufenden Daten und Metadaten werden vom SMGW nicht verarbeitet.

Über die CLS-Schnittstelle wird es CLS-Systemen im HAN ermöglicht, über das SMGW mit konfigurierten Teilnehmern im WAN über die Proxy-Funktionalität des SMGW zu kommunizieren. Die CLS-Proxy-Funktionalität des SMGW wird über das vom erfolgreich authentisierten GWA in das SMGW eingespielte *Proxy-Kommunikationsprofil* konfiguriert, sodass das SMGW nur die Kommunikation zwischen derart vom GWA konfigurierten CLS-Systemen im HAN und konfigurierten Teilnehmern im WAN ermöglicht.

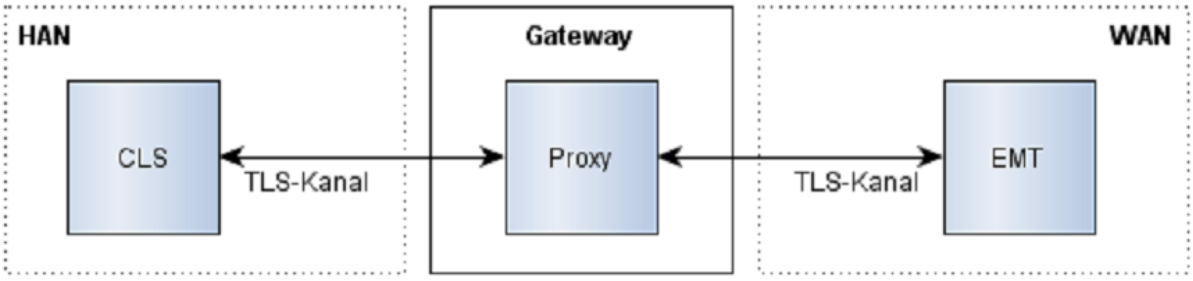


Abbildung 7: Schematischer Aufbau der CLS-Proxy-Kommunikation

In einem solchen Proxy-Kommunikationsprofil wird ein CLS mit einem bestimmten EMT verknüpft, indem die notwendigen Kommunikationsparameter der Verbindungsendpunkte spezifiziert werden. Die Konfiguration eines Proxy-Kommunikationsprofils durch den GWA erfolgt mittels der COSEM-Interface-Klasse *proxy_server*.

Die HAN-seitige Kommunikation zwischen CLS-System und dem SMGW wird gleichfalls über TLS gemäß [RFC 5246] gesichert. Das SMGW kann jedoch hierbei sowohl als TLS-Client als auch als TLS-Server agieren. Zur Authentisierung an der CLS-Schnittstelle werden X.509-Zertifikate eingesetzt, die die kryptografischen Anforderungen aus [BSI TR-03109-3, Kap. 5] erfüllen. Das SMGW prüft hierbei, dass das von der Gegenseite verwendete Zertifikat binär identisch zum zuvor vom GWA eingebrachten Zertifikat ist (DIRECT TRUST) und die zeitliche Gültigkeit des Zertifikats. Im Fehlerfall wird der Verbindungsaufbau vom SMGW abgelehnt und ein Eintrag im System-Log erzeugt.

Die WAN-seitige Kommunikation wird zwischen SMGW und WAN-Teilnehmer über eine beidseitig authentifizierte TLS-Verbindung abgesichert, in der das SMGW ausschließlich als TLS-Client auftritt und sein Sicherheitsmodul für die asymmetrischen kryptographischen Operationen verwendet.

Als Steuerungsprotokoll wird SOCKSv5 gemäß [RFC 1928] mit „TLS for SOCKSv5“ gemäß [SOCKS-SSL] eingesetzt werden. Die nach Aufbau der Proxy-Funktionalität fließenden (verschlüsselten) Daten des Proxy-Tunnels sind für das SMGW transparent.

Die Initiierung einer solchen transparenten Datenkommunikation gemäß dem Proxy-Kommunikationsprofil kann entweder durch das CLS oder durch den GWA oder autorisierten EMT erfolgen. Da jedoch ein EMT keine direkte Verbindung zum SMGW aufnehmen kann, muss die Initiierung indirekt über den GWA erfolgen, der hierfür den entsprechenden Administrationsbefehl unter Verwendung der Methode *connect* der COSEM-IC *proxy_server* gemäß Kap. 4.4.3.4 an das SMGW senden muss. Im Falle der Initiierung durch das CLS verwendet das CLS gemäß [RFC 1928] SOCKSv5-konforme Kommandos bei der initialen Kommunikation mit dem SMGW. In jedem Falle (d. h. sowohl einer nicht erfolgreichen wie auch erfolgreichen Verbindungsaufnahme und Kommunikation) werden nach Abbau der Verbindung alle mit dieser Verbindung assoziierten Ressourcen freigegeben.

Die folgenden Abbildungen zeigen schematisch den allgemeinen Protokollablauf bei SOCKSv5:

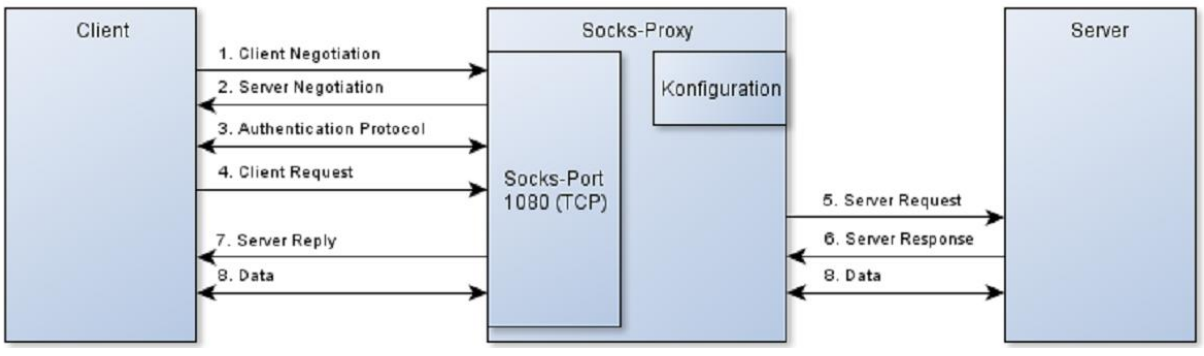


Abbildung 8: Protokollablauf bei SOCKSv5

Das SMGW stellt dabei (unabhängig von der jeweiligen Variante der Initiierung der Kommunikation) sicher, dass seine Funktionalität nicht durch CLS-Systeme kompromittiert werden kann. Insbesondere bleibt das SMGW funktional verfügbar, auch wenn CLS-Systeme unabsichtlich oder absichtlich fehlerhaft funktionieren. Die Kommunikation von CLS-Systemen

wird von der restlichen Kommunikation durch die Firewall-Funktionalität des SMGW geeignet separiert.

Folgende Maßnahmen werden dafür umgesetzt:

- Physische Trennung der Schnittstellen zum WAN, HAN und LMN.
- Beschränkung der maximalen Anzahl parallel kommunizierender CLS-Systeme des SMGW.
- Beschränkung der Rechenzeit für Betriebssystemprozesse, die für CLS-Funktionalität zuständig sind, so dass die Verfügbarkeit der restlichen Funktionalität nicht eingeschränkt werden kann.
- Beschränkung der durch CLS-Funktionalität verwendbaren Bandbreite für die WAN-Anbindung, so dass die Verfügbarkeit der restlichen Funktionalität nicht eingeschränkt werden kann.

4.8 WKS5: Zeitsynchronisation („NTP-TLS“)

Das SMGW besitzt eine Echtzeituhr (*Real Time Clock,* RTC) und eine Systemuhr. Die Echtzeituhr dient dazu, auch im Fall eines Spannungsverlusts ein gültiges Zeitsignal weiterzuführen. Die Systemuhr hingegen übernimmt während des Startvorgangs des Betriebssystems die Uhrzeit der Echtzeituhr und übernimmt ab diesem Zeitpunkt im Betriebssystem die Zeitführung. Erst wenn ein Spannungsverlust erkannt oder ein regulärer Neustart ausgelöst wurde, ist die Echtzeituhr wieder für die Zeitführung zuständig. Beiden Uhren liegt jeweils ein Quarz-Baustein zugrunde.

Als Anforderung für die maximal erlaubte Abweichung der Uhrzeit auf dem SMGW von UTC werden 3 % einer Registrierperiode festgelegt. Eine Registrierperiode als kleinstes Zeitraster für die Tarifierung ist im SMGW als 15 Minuten festgelegt, sodass die Uhrzeit auf dem SMGW nicht mehr als +/- 27 Sekunden von UTC abweichen darf.

Bei Spannungsverlust des SMGWs wird die Echtzeituhr durch den integrierten SuperCap weitergeführt. In dieser Zeit beträgt der Drift selbst unter den ungünstigsten Bedingungen weniger als -21 bzw. +6 Sekunden und somit weniger als 3% der kleinsten abrechnungsrelevanten Registrierperiodenlänge von 15 Minuten. Der Drift wird gemäß [Quarz RTC] wie folgt berechnet:

$N^2 * f1 – f2 – Tb*f3 = fe$

$fe * To = d <= fv * Tr$

N	-	Abweichung von der Betriebstemperatur (+25°C) des SMGWs
f1	-	Temperaturbedingte Abweichung in der Genauigkeit (-0,035 ppm / °C²)
f2	-	Frequenzbedingte Abweichung in der Genauigkeit (±5ppm)
Tb	-	Alter des SMGWs (8 Jahre)
f3	-	Alterungsbedingte Abweichung in der Genauigkeit (±3ppm / Jahr)
fv	-	Zulässiger prozentualer Fehler der Verschiebung der Registrierzeitpunkte gegen die gesetzliche Zeit (0,03)
Tr	-	Sollwert der Registrierperiodenlänge
fe	-	Effektive Abweichung in der Genauigkeit (-116,5 ppm / +29 ppm)
To	-	Zeitspanne über die der Drift gemessen wird (48 * 60 * 60 Sekunden)
d	-	Abweichung der Uhrzeit gegenüber der gesetzlichen Zeit

Die maximale negative Abweichung ergibt sich bei einer Betriebstemperatur von -25C° und beträgt -20,1312 Sekunden. Die maximale positive Abweichung ergibt sich bei einer Betriebstemperatur von +25C° und beträgt +5,0112 Sekunden.

Sobald das Betriebssystem des SMGW gestartet ist, wird die Uhrzeit der Echtzeituhr für die Systemuhr übernommen.

Die Gültigkeit der Uhrzeit der Systemuhr wird während des Startvorgangs der SMGW-Applikation überprüft. Dabei wird die Uhrzeit mit dem Erstellungsdatum des Betriebssystems verglichen. Wenn der Spannungsausfall länger als zwei Tage dauert und der integrierte Energiespeicher nicht für die Weiterführung der Echtzeituhr ausreicht, startet die Echtzeituhr bei Spannungswiederkehr mit dem Datum 01.01.1970. Insofern wird in diesem Fall erkannt, dass die Uhrzeit der Echtzeituhr nicht mehr gültig ist. Dieser Umstand wird mit der Logmeldung RTC_RECOVERFAIL protokolliert und die Systemzeit als „ungültig“ markiert. Kann die Systemzeit durch eine anschließende Synchronisation mit der gesetzlichen Zeit wiederhergestellt werden, wird dies mit der Logmeldung RTC_SYNCRECOVERED protokolliert. Das SMGW befindet sich in diesem Fall bis zur erfolgreichen Synchronisation im Zustand des eingeschränkten Betriebs (siehe Kapitel 2.6).

Während des Startvorgangs der SMGW-Applikation und darüber hinaus einmal pro Stunde wird versucht die Systemzeit mit der gesetzlichen Zeit synchronisieren.

Die Zeitsynchronisierung erfolgt im Kommunikationsszenario WKS5 („NTP-TLS“) gemäß [BSI TR-03109-1] mit einem vom GWA bereitgestellten NTP-Service gemäß [RFC 5905]. Die Kommunikation mit dem NTP-Service erfolgen dabei über einen gemäß TLS v1.2 [RFC 5246] gesicherten Kanal. Über diesen Kanal werden als Nutzinformationen ausschließlich die NTP-Protokolldaten gemäß [RFC 5905] übertragen. Dabei wird zunächst der TLS-gesicherte Kanal aufgebaut und danach die NTP-Zeitsynchronisation durchgeführt, damit die Zeit zum TLS-Kanalaufbau das NTP-Protokoll nicht beeinflusst.

Ist die Zeitsynchronisation erfolgreich, wird dies mit der Logmeldung RTC_SYNCED protokolliert und die Systemzeit als „gültig“ markiert. Um zu vermeiden, dass das Eichtechnische Logbuch zu schnell gefüllt wird, wird die Logmeldung RTC_SYNCED nur dann geschrieben, wenn eine der folgenden Bedingungen zutrifft:

- Die Logmeldung tritt zum ersten Mal für diesen Tag auf.
- Es wurde zuvor RTC_TOLERANCEEXCEEDED protokolliert.
- Es wurde zuvor RTC_SYNCPERIODEEXCEEDED protokolliert.
- Es wurde zuvor RTC_SERVERNOTRESPONDING protokolliert.

Nach jeder erfolgreichen Synchronisation der Systemuhr mit der gesetzlichen Zeit wird auch die Echtzeituhr mit der Systemuhr synchronisiert.

Ist die festgestellte Abweichung der Systemuhr zur gesetzlichen Zeit außerhalb der erlaubten Fehlertoleranz (3% der kleinsten abrechnungsrelevanten Registrierperiode) wird dies zusätzlich mit der Logmeldung RTC_TOLERANCEEXCEEDED protokolliert und an den GWA gesendet.

Konnte die Zeitsynchronisation nicht durchgeführt werden, da der Zeitserver nicht auf die NTP-Anfragen des SMGW geantwortet hat, wird dies mit der Logmeldung RTC_SERVERNOTRESPONDING protokolliert.

Der Zeitpunkt der letzten erfolgreichen Zeitsynchronisation wird gespeichert. Dieser Zeitpunkt wird regelmäßig mit der aktuellen Systemzeit verglichen. Konnte die Systemuhr des SMGW über ein Zeitfenster von 24 Stunden nicht mit der gesetzlichen Zeit synchronisiert werden, wird dies mit der Logmeldung RTC_SYNCFAIL protokolliert und an den GWA gesendet. Wurde die

Systemuhr des SMGW über ein Zeitfenster von 48 Stunden nicht mit der gesetzlichen Zeit synchronisiert, wird dies mit der Logmeldung RTC_SYNCPERIODEXCEEDED protokolliert, an den GWA gesendet, die Systemzeit als „ungültig“ markiert und der Betrieb eingeschränkt (siehe Kapitel 2.6), bis die Systemzeit wieder erfolgreich mit der gesetzlichen Zeit synchronisiert wurde. Diese Regel gilt auch im Falle einer Spannungsunterbrechung.

Die technischen Daten des der Systemuhr zugrunde liegenden Quarz-Bausteins werden in [Quarz System] beschrieben. Auf Basis dieses Quarz-Bausteins kann eine gültige Uhrzeit über mehrere Stunden aufrechterhalten werden, ohne dass eine Zeitsynchronisation mit dem vom GWA bereitgestellten Zeitserver erfolgt. In dieser Zeit beträgt der Drift selbst unter den ungünstigsten Bedingungen weniger als 3% der kleinsten abrechnungsrelevanten Registrierperiodenlänge von 15 Minuten. Der Drift wird wie folgt berechnet:

$$f1 + f2 + Tb*f3 + fk = fe \approx f1$$
$$fe * To = d \leq fv * Tr$$

f1	Maximale temperaturbedingte Abweichung (-40°C bis +85°C) in der Genauigkeit (±50 ppm)
f2	Frequenzbedingte Abweichung in der Genauigkeit (±20ppm)
Tb	Alter des SMGWs (8 Jahre)
f3	Alterungsbedingte Abweichung in der Genauigkeit (±2ppm / Jahr)
Fk	Korrekturwert für die ermittelte Abweichung in der Genauigkeit (≈ -1 * (f2 + Tb*f3))
Fv	Zulässiger prozentualer Fehler der Verschiebung der Registrierzeitpunkte gegen die gesetzliche Zeit (0,03)
Tr	Sollwert der Registrierperiodenlänge
Fe	Effektive Abweichung in der Genauigkeit (≈ ±50 ppm)
To	Zeitspanne über die der Drift gemessen wird (48 * 60 * 60 Sekunden)
D	Abweichung der Uhrzeit gegenüber der gesetzlichen Zeit

Trotz der Eigenschaft, dass das SMGW eine gültige Uhrzeit über mehrere Stunden ohne Synchronisation aufrechterhalten kann, wird die Synchronisation jede Stunde durchgeführt. Durch ein solch kurzes Intervall wird selbst bei einer Fehlfunktion des Quarzbausteins (z.B. aufgrund eines Hardware-Defekts) nahezu ausgeschlossen, dass ein potentieller Verstoß gegen die erlaubte Fehlertoleranz unbemerkt bleibt, der ansonsten zu einem Versand von irrtümlich als „gültig“ markierten Messwerten führen könnte.

4.9 WKS7: Wake-Up-Paket (WAF7)

Die zwischen dem SMGW und Teilnehmern im WAN verwendeten TLS-Verbindungen werden immer vom SMGW als TLS-Client initiiert. Über ein sogenanntes *Wake-Up-Paket* (als UDP-Datenpaket gemäß [BSI TR-03109-1, Anhang A]) ist es möglich, dass der GWA den Aufbau des TLS-Kanals durch das SMGW für das WAN-Kommunikationsszenario WKS1 „Management“ (s. Kap.4.4) anfordern kann (vgl. Abbildung 8 in [BSI TR-03109-1] zum Anwendungsfall „Wake-Up Service“).

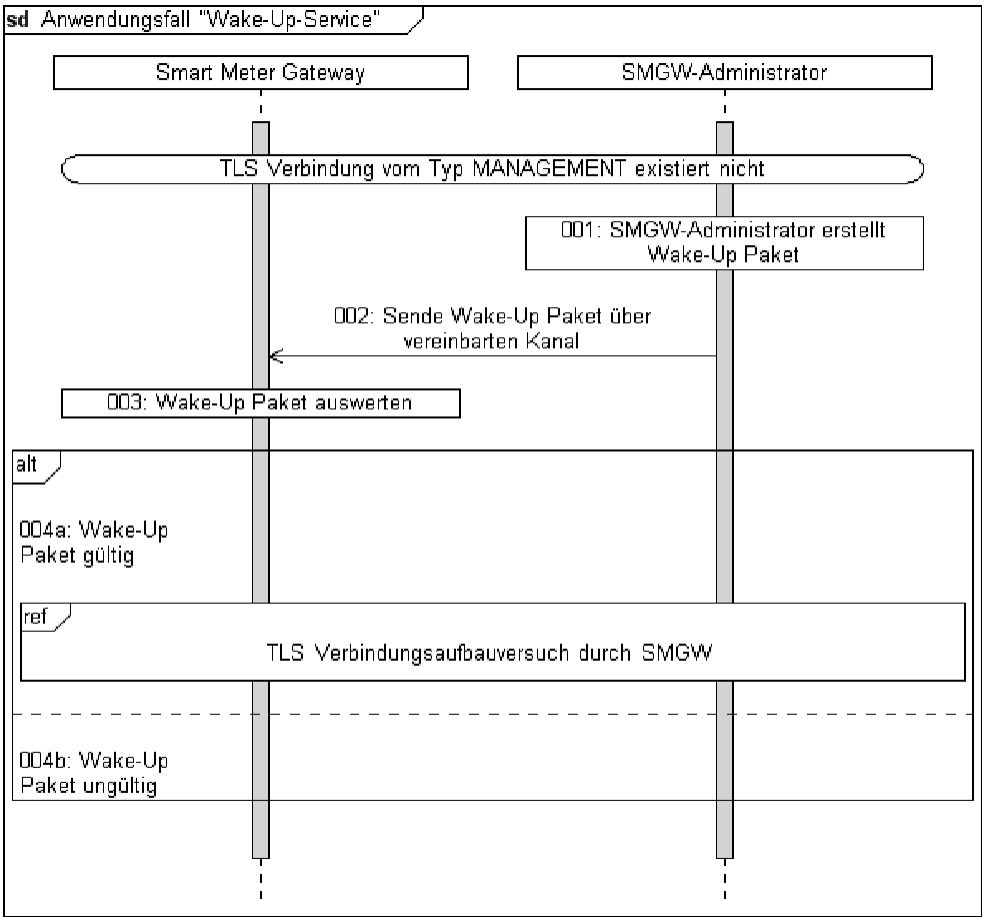


Abbildung 9: Sequenzdiagramm für den Anwendungsfall „Wake-Up Service“

Das Wake-Up-Paket enthält u. a. die Geräteidentifizierung des adressierten SMGW und einen Zeitstempel. Diese Felder sind mit dem privaten Schlüssel des GWA für die Inhaltsdatensignierung GWADM_SIG_PRV signiert. Die Informationen im Wake-Up-Paket sind nicht vertraulich und werden daher nicht verschlüsselt. Der Aufbau des Wake-Up-Pakets ist in [BSI TR-03109-1, Anhang A] beschrieben und wird vom SMGW in der folgenden Form erwartet:

Header (2 Bytes)		*Vers. (1 B)*	*RecipientId (9 Bytes)*								*Timestamp (8 Bytes)*			
‚W‘	‚U‘	01h	0Eh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh
Timestamp (continued)				*Padding / Reserved (11 Bytes + 1 Byte Padding-Length)*											
xxh	xxh	xxh	xxh	0Bh	0Bh	0Bh	0Bh	0Bh	0Bh	0Bh	0Bh	0Bh	0Bh	0Bh	0Bh
Sig. Frm	*OID Len*	*SignatureAlgorithmOID (OID-Length Bytes)*													
01h	0Ah	04h	00h	7Fh	00h	07h	01h	01h	04h	01h	xxh				
ECDSA-Signature (r) (L Bytes)															
xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh
xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh
xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh
ECDSA-Signature (s) (L Bytes)															
xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh
xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh
xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh	xxh

Abbildung 10: Struktur des Wake-Up Pakets

Das Wake-Up-Paket enthält hierbei die folgenden Felder:

- Das Feld **Header** dient zur Kennzeichnung des Wake-Up Pakets und ermöglicht eine erste einfache (hardwarenahe) Überprüfung bzw. Klassifizierung der empfangen Pakete.
- Das Feld **VersionId** bezeichnet die verwendete Version der Wake-Up-Paket-Definition. Bei eventuellen zukünftigen Erweiterungen werden neue Versionsnummern vergeben.
- Das Feld **RecipientId** dient zur eindeutigen Identifizierung des SMGWs. Die Vergabe und Kodierung der RecipientId erfolgt gemäß [VDE AR-N 4400:2011-09] Kapitel 4.2 nach [DIN 43863- 27085:2012-04] "Herstellerübergreifende Identifikationsnummer für Messeinrichtungen" erfolgen. Nur das adressierte SMGW darf das Wake-Up-Paket verarbeiten. Hiermit soll verhindert werden, dass das Wake-Up-Paket von einem Angreifer missbraucht wird, um eine Vielzahl von SMGW in der Verantwortung eines SMGW-Administrators zu einem gleichzeitigen TLS Call-Back zu verleiten (DoS-Attacke).
- Das Feld **Timestamp** enthält die aktuelle Zeit (in UTC) zum Zeitpunkt der Erstellung des Wake-Up-Pakets. Geringfügige Unterschiede zwischen den jeweiligen Uhrzeiten auf den Servern und den SMGW sind üblich. Der Timestamp muss daher im Zeitfenster von +/- **27 Sekunden** relativ zur Uhrzeit des SMGWs liegen.

- Dem Feld Timestamp folgen 12 **Padding** Bytes, um den Datensatz auf 32 (2*16) Bytes aufzufüllen.
- Das letzte Byte enthält die Anzahl der vorher kodierten Paddingbytes.
- Anschließend wird vom SMGW-Administrator ein Hash gemäß [BSI TR-03109-3] über diesen n*16 Bytes Datensatz generiert. Der Hash wird mit ECDSA und einer elliptischen Kurve gemäß [BSI TR-03109-3] signiert.
- Das Feld **SignatureFormat** dient zur Kennzeichnung, welches Signaturformat im Wake-Up-Paket verwendet wurde. Bisher ist nur das „PlainFormat(1)“ vorgesehen.
- Das Feld **SignatureAlgorithmOIDLength** enthält die kodierte Bytelänge des darauf folgenden Felds *SignatureAlgorithmOID*.
- Das Feld **SignatureAlgorithmOID** enthält die OID des verwendeten Signaturalgorithmus gemäß [BSI TR-03109-3] (s. Kapitel 10 Verzeichnis der im SMGW verwendeten Object Identifier (OID)).
- Die erzeugte **ECDSA-Signatur (r, s)** wird im "Plain Format" (s. Kapitel 10 Verzeichnis der im SMGW verwendeten Object Identifier (OID)) kodiert und an das Wake-Up-Paket angehängt.

Als Herstellerkennzeichnung wird dabei gemäß FLAG-Registrierung die Kennung „*PPC*“ (als 3 ASCII-Großbuchstaben = „50 50 43 h“) verwendet.

Zur Abwehr von DoS-Angriffen ist die Anzahl der möglichen Wake-Up-Paket-Signaturprüfungen einschränkt, indem der Empfang der Wake-Up-Pakete für einen gewissen Zeitraum gestoppt wird, falls mehr als 12 ungültige Wake-Up-Pakete zeitlich kurz aufeinander folgend vom SMGW entgegengenommen wurden.

Schlägt die Überprüfung der Kennzeichnung, der Geräteidentifizierung, des Zeitstempels oder der Signatur fehl, wird der weitere Prüfvorgang beim ersten Fehler unterbrochen, die Nachricht sofort verworfen und der System-Log-Eintrag *„Ein WakeUp-Paket wurde verworfen“* unter Angabe des Grundes erstellt. Es wird keine Meldung zum Teilnehmer im WAN zurückgesendet und der Vorgang der Verarbeitung des Pakets beendet.

Gelingt der Prüfvorgang des Wake-Up-Pakets in allen Schritten, wird ein TLS-Kanal zum GWA im WAN initiiert, wobei die Adressierungsdaten aus dem vom GWA im SMGW konfigurierten Kommunikationsprofil entnommen werden. Das SMGW unternimmt bis zu drei Versuche zum Verbindungsaufbau. Gelingt nach diesen bis zu drei Versuchen kein derartiger Aufbau eines TLS-Kanals, erstellt das SMGW einen entsprechenden Eintrag im System-Log. Erhält das SMGW weitere gültige Wake-Up-Pakete und wurde ein solcher TLS-Kanal schon aufgebaut, baut das SMGW keinen weiteren solchen TLS-Kanal auf. Eine andere Aktion außer dem Aufbau des TLS-Kanals zu dem konfigurierten externen Kommunikationspartner im WAN ist im SMGW programmatisch nicht möglich.

Die Möglichkeit zur Konfiguration des Wake-Up-Service durch den GWA ist nicht vorhanden.

5 Update-Funktionalitäten

Im folgenden Kapitel wird die Update Funktionalität des SMGWs beschrieben. Die Beschreibung schließt dabei die Übermittlung des Firmware Updates vom SMGW-Hersteller (GWH) zum GWA und den Update Vorgang vom GWA zum SMGW ein.

5.1 Sicherer Kommunikationskanal zwischen GWH und GWA

Falls noch nicht vorhanden, muss zur Auslieferung von Firmware-Updates vom Hersteller zum GWA ein sicherer E-Mail-Kommunikationsweg zwischen SMGW-Hersteller und GWA etabliert werden. Dazu werden bei beiden Parteien jeweils ein oder mehrere Ansprechpartner benannt. Diese Ansprechpartner tauschen ihre S/MIME-Zertifikate aus. Nach dem Austausch der Zertifikate gleichen die beiden Ansprechpartner per Festnetz-Telefonat die sog. „Fingerprints“ ihrer E-Mail-Verschlüsselungszertifikate ab. Dieser Prozess wird wiederholt, sobald einer der beiden Ansprechpartner sein E-Mail-Verschlüsselungszertifikat erneuert.

5.2 Information des GWA über ein neues Firmware Update

Der SMGW Hersteller betreibt eine HTTPS-gesicherte Internetseite und informiert den Gateway Administrator über diese Internetseite über den jeweils aktuellsten Firmwarestand. Sobald ein neues Firmware Update vorliegt, werden die entsprechende Revisionsbezeichnung und das Datum auf der gesicherten Internetseite bekannt gegeben.

Die Zugangsdaten für die gesicherte Internetseite werden dem Gateway Administrator über den sicheren E-Mail-Kommunikationsweg übermittelt und bei Bedarf aktualisiert.

Über den sicheren E-Mail-Kommunikationsweg informiert der SMGW-Hersteller den Gateway Administrator auch über die Verfügbarkeit neuer Firmware Versionen.

5.3 Download, Prüfung und Entfernung der äußeren Signatur durch den GWA

Um die Integrität und Authentizität der Firmware Datei sicherzustellen, ist diese vom Hersteller des SMGWs signiert worden. Die Signatur (äußere Signatur) wurde dabei mit Signatur-Schlüsselmaterial des Herstellers aus der SM-PKI als CMS-Container in der Ausführung Signed-Data gemäß RFC 5652, Abschnitt 5 gebildet.

Der GWA muss diese Signatur überprüfen. Es dürfen nur Firmware Dateien verwendet werden, deren Signatur mit einem gültigen Zertifikat aus der SM-PKI geprüft wurde. Das Signatur-Zertifikat des SMGW-Herstellers kann aus der SM-PKI bezogen werden und muss erfolgreich gegen das Root Zertifikat der SM-PKI validiert werden können.

Falls die Überprüfung der äußeren Signatur fehlschlägt, muss die Firmware-Datei umgehend gelöscht werden und der SMGW-Hersteller darüber informiert werden. Die betroffene Firmware-Datei darf auf keinen Fall für das Update eines SMGWs verwendet werden.

Nach der erfolgreichen Prüfung, vor der Übermittlung ans SMGW muss diese Signatur entfernt werden, lediglich der Inhalt des (äußeren) CMS Containers darf an das SMGW übermittelt werden.

5.4 Aufruf des Update Prozesses beim SMGW

Instanzen der Klasse *fw_update* werden vom GWA zur Initiierung und Durchführung des Firmware-Updates des SMGWs benutzt. Diese Klasse besitzt keine Methoden. Die Attribute sind wie folgt:

Tabelle 98: Attribute der IC *fw_update*

Name des XML-Elements	XML-Datentyp	Beschreibung
logical_name	***string***	OBIS-Code und Geräte-ID zur Identifizierung der Instanz der Klasse.
update_status	Enum := { *idle*, *running*, *waiting*, *interrupted* }	(read only); Der Status des Firmware-Updates. - „*idle*“: Derzeit wird kein Firmware Update durchgeführt. - „*running*“: Die Firmware wird im Moment heruntergeladen und das Firmware Update wird durchgeführt. - „*waiting*“: Ein Zeitpunkt in der Zukunft ist in time_of_update angegeben. - „*interrupted*“: Das Firmware Update wurde unterbrochen. Eine entsprechende Meldung wurde in das Systemlog eingetragen und an den GWA gesendet. Das Attribut wird vom SMGW nicht verwendet.
time_of_update	TYPE_iso8601_datetime	Der Wert des Attributs muss den Vorgaben des Datentyps entsprechend und wird in der vorliegenden SMGW Version nicht ausgewertet.

Name des XML-Elements	XML-Datentyp	Beschreibung
time_of_activation	TYPE_iso8601_datetime	Der Wert des Attributs muss den Vorgaben des Datentyps entsprechend und wird in der vorliegenden SMGW Version nicht ausgewertet.
firmware_uri	TYPE_uriPath	In diesem Attribut kann die URI übergeben werden, von der sich das SMGW die Firmware vom GWA herunterladen kann.
target_reference	TYPE_Obj_Ref	(optional) Das Attribut wird in der vorliegenden SMGW Version nicht genutzt.

Im SMGW ist eine Instanz der Klasse *fw_update* vorhanden, die der OBIS-Kennzahl „01005e3180a1“ zugeordnet ist.

Um ein Firmware-Update zu initiieren, muss ein gültiges Wake-Up Paket vom GWA an das SMGW übermittelt werden und anschließend vom SMGW ein TLS-Kanal nach WAN Kommunikationsszenario WKS1 aufgebaut werden. In diesem TLS-Kanal kann der GWA mittels des HTTP-Verbs PUT ein Update auf die Instanz der Klasse durchführen. Das folgende Beispiel verdeutlicht die Initiierung.

Tabelle 99: Aktualisierung der IC *fw_update*

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für Initiierung des Firmware Updates	Request-Body	HTTP Response Statuscode	Response-Body
Update der bestehenden Klasseninstanzen	PUT	<PoC>/cosem/ldevs/eppc0110123456.sm/objects/*01005e3180a1*	*fw_update*	200	Leer

Anschließend baut das SMGW eigenständig einen TLS-Kanal zum GWA nach WAN-Kommunikationsszenario WKS2 auf. Das SMGW beginnt mit dem Download der Firmware-Datei, indem es mittels des HTTP-Verbs GET den Download startet. Das Firmware-File wird nicht in einer XML-Struktur übertragen, sondern als Octet-Stream. Das folgende Beispiel verdeutlicht den Download der Firmware-Datei.

Tabelle 100: Abfrage der Firmware Datei beim GWA

Zugriffs-Dienst	HTTP-Verb	Beispiel URI für Download der Firmware-Datei	Request-Body	HTTP Response Statuscode	Response-Body
Abfrage der Firmware Datei	GET	*firmware_uri* aus *fw_update*	leer	200	Die Firmware-Datei als Octet-Stream.

Falls der Download unterbrochen wurde (z.B. wegen einer instabilen Kommunikationsverbindung), wird der Download mittels des HTTP-Verbs GET und einem zusätzlichen Range-Header gemäß [RFC 7233] fortgesetzt, bis der Download abgeschlossen wurde.

Der erste GET-Request des SMGW an den GWA wird dabei noch ohne Range Header angegeben. In der Response liefert der GWA im HTTP-Header *content-length* die gesamte Größe des Firmware Pakets. Der Beginn des Downloads wird durch die Logmeldung (SMGW_Logging) „SWUP_Download“ im System-Log verzeichnet.

Wenn der Download unterbrochen wird, teilt das SMGW den Download einzelne Pakete auf. Dabei wird der HTTP-Header *Content-Range* genutzt. Die Anfragen mit dem HTTP-Header *Content-Range* werden dabei vom SMGW automatisch ohne zusätzliche Konfiguration durch oder Interaktion mit dem GWA erzeugt.

Auf eine solche Anfrage gibt der GWA den geforderten Bereich des Firmware-Paketes zurück. In diesem Fall wird im HTTP-Header *Content-Range* der angefragte Bereich und im HTTP-Status-Code „*206 Partial Content*“ angegeben. Im HTTP-Header *content-length* wird nur die Größe des angefragten Bereichs angegeben. Bei der Produktion definiert der Hersteller, wie oft ein Firmware-Download nach einer Unterbrechung neu angestoßen wird, bevor er endgültig abgebrochen wird. Dieser Wert ist für den GWA nicht modifizierbar. Der Wert wird zurückgesetzt, falls bereits ein Teil der Firmware heruntergeladen werden konnte, wodurch es in diesem Fall zu einer deutlich häufigeren Wiederholung der Anfragen des SMGWs kommen kann.

Bei jeder Unterbrechung eines Downloads wird im System-Log die Meldung (SMGW_Logging) „SWUP_DOWNLOADINT“, bei jeder Wiederaufnahme die Meldung „SWUP_DOWNLOADRESUMED“ geschrieben.

Falls der GWA einen anderen HTTP-Status-Code als 200 oder 206 verwendet, wird das als negative Rückmeldung gewertet. Der Download-Versuch wird dann vom SMGW entweder erfolgreich neu angestoßen, oder solange wiederholt bis die maximale Anzahl der Wiederholungen erreicht ist.

Sobald der Download endgültig abgebrochen wird, wird der GWA vom SMGW durch die Logmeldung (SMGW_Logging) „SWUP_DOWNLOADCANCELED“, die zusätzlich auch als Benachrichtigung über einen TLS-Kanal des WAN-Kommunikationsszenarios WKS2 versendet, informiert.

Die Firmware-Datei ist Hersteller-spezifisch, d.h. ein Update eines Herstellers kann nicht für SMGW eines anderen Herstellers verwendet werden.

Zusätzlich zu der zuvor vom GWA entfernten Signatur ist die Firmware Datei mit einer weiteren Signatur (innere Signatur) geschützt. Die innere Signatur wird nach der Übermittlung der Firmware Datei an das Smart Meter Gateway durch selbiges überprüft. Das zum Überprüfen der Signatur notwendige Schlüsselmaterial wird während der Vorpersonalisierung in das SMGW eingespielt.

Nach erfolgreicher Übermittlung und anschließender Signaturprüfung installiert das SMGW die neue Firmware eigenständig. Die erfolgreiche Installation wird im Systemlog und im eichtechnischen Log mit dem Eintrag „SWUP_INSTALLED“ (SMGW_Logging) protokolliert und dem GWA zusätzlich als Benachrichtigung mitgeteilt.

Es können nur Firmware Dateien eingespielt werden, deren Versionsstand größer als der bereits vorhandene Versionsstand ist. Das SMGW prüft das automatisch bei der Installation und bricht die Installation ab, falls dieses Kriterium nicht erfüllt ist.

Falls die Installation nicht erfolgreich abgeschlossen werden konnte wird das Firmware-Update (sofern notwendig) rückgängig gemacht und das SMGW fällt auf die alte Firmware-Installation zurück. Sowohl die fehlgeschlagene Installation (Logmeldung „SWUP_INSTALL_FAILED“, SMGW_Logging) als auch der Rückfall auf die alte Firmware-Installation (Logmeldung „SWUP_ROLLBACK“, SMGW_Logging) werden im Systemlog und im eichtechnischen Log protokolliert und dem GWA zusätzlich als Benachrichtigung mitgeteilt.

Nach der Installation wird die neue Firmware durch Herunterfahren und Neustarten des SMGWs aktiv. Während des Neustarts führt das SMGW einen Selbsttest durch, dessen Ergebnis im System- und im Eichtechnischen Log verzeichnet wird und das dem GWA zusätzlich als Benachrichtigung mitgeteilt wird.

Ein Firmware-Update (Download und anschließende Installation) dauert bei einer stabilen und performanten WAN Anbindung ca. 25 Minuten. Bei einer instabilen Verbindung und einer niedrigen Datenübertragungsrate kann der Vorgang auch mehrere Stunden in Anspruch nehmen.

6 Logging und Auditing

Das SMGW erzeugt Audit-Daten für alle Ereignisse, die dem System-Log, dem Letztverbraucher-Log bzw. dem eichtechnischen Log zugewiesen sind. Beim Start und beim Beenden der Audit-Funktionalität des SMGW werden hierüber entsprechende Einträge in die zugehörigen Audit-Daten geschrieben.

Die Obergrenze der Größe des Log-Speichers ist im SMGW auf **128 MB** festgelegt. Dies wird als hinreichend angesehen, da die Log-Nachrichten vom SMGW aggregiert werden. Aggregation bedeutet, dass bei mehrfachem Auftreten des gleichen Sicherheitsereignisses nur noch Anzahl und Zeitpunkt protokolliert werden. Detailinformationen (z.B. Paketanalysedaten) werden in diesem Falle mit dem aktuellsten Wert überschrieben.

Das SMGW unterscheidet hierbei zwischen den folgenden Typen von Logs:

a) **System-Log** mit Daten bzgl. des Systemzustands, die vom GWA resp. Service-Techniker zwecks Prüfung des derzeitigen funktionalen Zustands benötigt werden.
b) **Letztverbraucher-Log** mit allen Informationen, mit denen ein Letztverbraucher seinen Verbrauch und seine Abrechnungen nachvollziehen kann.
c) **Eichtechnisches Log** mit allen eichtechnischen Informationen des SMGW (wie z. B. Kalibrierungsdaten und Auswertungsprofile).

Im jedem der drei Logs des SMGW werden folgende Informationen gespeichert:

- Laufende Nummerierung der Einträge
- Art des Ereignisses (Kennziffer eines geänderten Parameters oder eines Fehlers)
- Identifikation des Parameters und Wert des veränderten Parameters nach der Änderung
- Identität des auslösenden Subjekts (falls zutreffend)
- Vermerk, ob das Ereignis das SMGW selbst, die Anzeige des SMGWs, einen angeschlossenen Zähler oder einen Kommunikationsadapter des SMGWs betrifft
- Zeitpunkt des Ereignisses bzw. der Zeitpunkt der Veränderung

6.1 System-Log

Das SMGW protokolliert jedes wichtige System-Ereignis (z.B. Fehlermeldungen, Ausfall der WAN-Verbindung, sicherheitsrelevante Ereignisse, Aktivitäten des GWA, etc.) im System-Log. Dieses Log kann nur vom autorisierten GWA an der WAN-Schnittstelle sowie dem autorisierten Service-Techniker an der HAN-Schnittstelle eingesehen werden. Die Informationen des System-Logs dienen dazu, den momentanen Status des SMGW zu erkennen und eventuelle Fehlerquellen oder Störungen zu identifizieren.

Sollte das System-Log funktional nicht verfügbar sein, z.B. da sich das SMGW in der Anlaufphase befindet und daher der SMGW-interne Log-Dienst noch nicht verfügbar ist, erstellt das SMGW – falls technisch möglich - im Syslog seines Betriebssystems einen entsprechenden Eintrag.

Die Logeinträge des Systemlogs werden automatisch nach einer Vorhaltezeit von **15 Monaten** durch das SMGW entfernt. Falls die Kapazität des System-Logs bereits zuvor überschritten wird, werden die ältesten Einträge überschrieben.

Eine Aufstellung der vom SMGW im System-Log protokollierten Ereignisse findet sich in [SMGW_Logging]. Insbesondere protokolliert das SMGW die folgenden Ereignisse im System-Log:

Audit der SMGW-Sicherheitsfunktionalitäten

- Starten und Stoppen der eichtechnischen Audit-Funktionalität
- Inbetriebnahme des SMGW
- Auslösung eines SMGW-Selbsttests
- (erfolgreiches oder fehlgeschlagene) Ausführung des SMGW-Updates

Sicherheitsschutzverletzung

- eine vom SMGW anhand der kryptografischen Prüfsumme festgestellten Verfälschung der gespeicherten Daten; eine vom SMGW anhand der kryptografischen Prüfsumme festgestellten Verletzung der Authentizität und Integrität der Firmware- oder Softwarekomponente des SMGW
- Fehlerhafte Prüfung der Integrität der SMGW-Hardware resp. -Firmware sowie fehlerhafte Prüfung der Integrität der SMGW-Software-Applikation
- Versuchte Zugriffe auf nicht erlaubte Ressourcen des SMGW (Audit)
- Jedes Auftreten eines vom SMGW als potentielle Sicherheitsschutzverletzung eingeschätzten Ereignisses (vgl. [SMGW_Logging])

Mess-Fehler

- Statusinformation „*Fataler Fehler*“ eines angeschlossenen Zählers: Auditierung der Fehlermeldung einschl. Vermerk zum Austausch des Zählers
- Nicht erfolgreiche Prüfung der Signatur eines Messwerts: Auditierung des Systemfehlers einschl. der „inneren Signatur“ dieses Messwerts
- Nicht zutreffende Angaben bzgl. des SML-Protokoll-Version und -Typs

Zeit-Fehler

- Überschreiten der maximal zulässigen Abweichung der Systemuhr des SMGW (> 3% der Registrierperiode): Auditierung der Messdaten einschl. Vermerk „*inoperable*“
- Fehlerhafte Synchronisation der gesetzlichen Zeit bei Ausfall der Synchronisation für längere Zeit als das Synchronisations-Toleranzintervall (siehe [PTB-A 50.8], Abschnitt 4.2.1.2)

Kommunikations- und Systemfehler

- Fehlgeschlagene Prüfung eines kryptografischen Schlüssels oder Zertifikats aufgrund eines ungültigen DIRECT-TRUST-Vergleichs oder aufgrund einer fehlerhaften Gültigkeitsprüfung (Prüfung des Gültigkeitszeitraums, Prüfung der Zertifikatskette)

- Fehler vom SMGW in Form der HTTP-Statuscodes bei der Kommunikation an der WAN-Schnittstelle ausgegeben werden (vgl. Kap. 4.3)
- Misslingen des TLS-Kanal-Aufbaus nach erfolgreich geprüftem Wake-Up-Call
- Eintreten des Fallback-Falls im Quality-Of-Service (für *log_fallback_failures* = ‚*true*‘)

6.2 Eichtechnisches Log

Das eichtechnische Log dient der Registrierung von Änderungen an eichtechnisch relevanten Soft- und Firmware-Teilen sowie den Konfigurationsprofilen und den zugehörigen Parametern. Des Weiteren werden im eichtechnischen Log eichtechnisch relevante Ereignisse gespeichert, sodass nachträglich erkennbar ist, ob und welche Messwerte verfälscht worden sind.

Das SMGW erlaubt einen lesenden Zugriff auf das eichtechnische Log ausschließlich dem erfolgreich authentisierten GWA an der WAN-Schnittstelle. Hierzu verwendet der GWA die COSEM-Interface-Klasse *profile_generic* (in der Nutzungsvariante *Logbook)*, in der die Audit-Einträge als Instanzen der COSEM-Interface-Klasse *log_entry* zum eichtechnischen Log des SMGW zusammengefasst sind. Ein Löschen des eichtechnischen Logs oder seiner Teile ist nicht möglich. Kann das SMGW keine weiteren Einträge im eichtechnischen Log erstellen, stoppt das SMGW den Empfang von Messwerten und informiert den GWA.

Das SMGW nimmt die folgenden Einträge im eichtechnischen Log vor.

<u>Audit der SMGW-Sicherheitsfunktionalitäten</u>

- Starten und Stoppen der eichtechnischen Audit-Funktionalität
- Inbetriebnahme des SMGW
- Auslösung eines SMGW-Selbsttests

<u>GWA-Administration</u>

- Anschluss und die Registrierung eines jeden neuen Zählers unter Angabe des ‚*device_identifier*‘.
- Einbringen eines Zählerprofils unter Angabe des ‚*logical_name*‘
- Entfernung eines Zählers
- Änderung (einschließlich Parametrierung) an Auswertungsprofilen, sowie Einbringen und Löschen von Auswertungsprofilen. Hierdurch wird auch die Zuordnung eines Zählers zu einem Letztverbraucher sichergestellt.
- Fehler bei der Parametrierung der Profile (Zählerprofile, Auswertungsprofile) in Bezug auf die angegebenen Ober- oder Untergrenzen für die Parameter des Profils, soweit diese nicht schon als syntaktische Fehler durch den internen XML-Parser erkannt wurden
- Vergabe eines Passworts des Letztverbrauchers für die HAN-Anzeige

FW-Update

- Firmware-Update: Das SMGW zeichnet das Datum und den Zeitpunkt des Downloads, die Identifikation der heruntergeladenen rechtlich relevanten Software, die Identifikation der herunterladenden Stelle und einen Erfolgseintrag auf. Für jeden eigenen neuen Downloadversuch, unabhängig davon, ob dieser erfolgreich war oder nicht, wird ein Eintrag erzeugt.

Allgemeine Systemfehler

- Fataler Systemfehler des SMGW: Auditierung des Systemfehlers einschl. Vermerk „*inoperable*“ (sofern technisch möglich)
- Fehler wegen der Überschreitung des SMGW-internen Zeitlimits für die Abarbeitung eines externen funktionalen Aufrufs

Mess-Fehler

- Statusinformation „Fataler Fehler“ eines angeschlossenen Zählers: Auditierung der Fehlermeldung einschl. Vermerk zur Deaktivierung des Zählers

Kommunikationsfehler

- Übertragungsfehler oder dauerhaftes Ausbleiben von erwarteten Nachrichten bei Überschreitung der maximalen Anzahl von Wiederholungen oder des Überwachungsintervalls

Zeit-Fehler

- Überschreiten der maximal zulässigen Abweichung der Systemuhr des SMGW (> 3% der Registrierperiode): Auditierung des Ereignisses
- Fehlerhafte Synchronisation der gesetzlichen Zeit bei Ausfall der Synchronisation für längere Zeit als das Synchronisations-Toleranzintervall (siehe [PTB-A 50.8], Abschnitt 4.2.1.2)

Sicherheitsschutzverletzung

- eine vom SMGW anhand der kryptografischen Prüfsumme festgestellten Verfälschung der gespeicherten Daten; eine vom SMGW anhand der kryptografischen Prüfsumme festgestellten Verletzung der Authentizität und Integrität der Firmware- oder Softwarekomponente des SMGW
- Versuchte Zugriffe auf nicht erlaubte Ressourcen (Audit)

6.3 Eichrechtlich relevante Fehlermeldungen

Folgende Logmeldungen dokumentieren eine eichrechtlich relevante Fehlermeldung. D.h. mit Eintritt des entsprechenden Fehlers, ist die eichrechtkonforme Verwendung des SMGWs nicht mehr gewährleistet. Das Gerät muss ausgetauscht oder repariert und geeicht werden. Alle

Daten, die nach Fehlereintritt entstanden sind, dürfen nicht für die Rechnungsbildung verwendet werden.

Tabelle 101: Eichrechtlich relevante Fehlermeldungen

ID	*Log-Level*	*Logmeldung*	*Eichrechtrelevante Fehler*
10065	WARNING	Selbsttest wurde mit [n] behebbaren Fehler(n) und [n] schweren Fehlern abgeschlossen. Fehler: [Fehler]	Ja (nur falls ein oder mehrere schwere Fehler aufgetreten sind)
10066	FATAL	Die Systemintegrität kann nicht mehr sichergestellt werden. Der Messbetrieb wird eingestellt.	Ja
10098	FATAL	Der Speicherplatz für eichrechtliche Protokolleinträge ist erschöpft. Der Messbetrieb wird eingestellt. Das SMGW muss sofort ausgetauscht werden. Der Messbetrieb wird eingestellt.	Ja
10223	FATAL	Die Speicherintegrität kann nicht mehr sichergestellt werden. Der Messbetrieb wird eingestellt. Grund: [Grund]	Ja
10259	ERROR	Der Speicherplatz des SMGWs ist erschöpft. Der Messbetrieb wird eingestellt. Das SMGW muss ausgetauscht werden.	Ja
10260	ERROR	Der Messbetrieb wird eingestellt.	Ja (falls nicht gefolgt von Logmeldung 10261)
10271	ERROR	Der Messbetrieb wird eingestellt.	Ja

7 Glossar

ARP	*Address Resolution Protocol*
CLS	*Controllable Local System*
COSEM	*Companion Specification for Energy Metering*
DI_reference	*Data Link Reference*
EMT	Externer (Markt-)Teilnehmer
FQDN	*Fully Qualified Domain Name*, Voll-qualifizierter Domänenname
GWA	Smart Meter Gateway Administrator
GWH	Smart Meter Gateway Hersteller
HAF	HAN-Anwendungsfall
HAN	*Home Area Network*
HCS	*Header Check Sequence*
HDLC	*High-Level Data Link Control*
HSM	*Hardware Security Module*
HTTP	*Hypertext Transfer Protocol*
IC	*Interface Class*
IP	*Internet Protocol*
LAF	LMN-Anwendungsfall
LF	Lieferant
LMN	*Local Metrological Network*
LSB	*Least significant bit*
MDL	Messdienstleister
MSB, MSP	Messstellenbetreiber, *Metering Service Provider*
OBIS	*COSEM Object Identification System*
RTC	*Real Time Clock*, Echtzeit-Uhr
RTT	*Round Trip Time*
SM-PKI	Smart-Metering-Public-Key-Infrastruktur
TAF	Tarifanwendungsfall
TCP	*Transmission Control Protocol*
TLS	*Transport Layer Security*
UTC	*Coordinated Universal Time*, koordinierte Weltzeit
VNB	Verteilnetzbetreiber

WAF	WAN-Anwendungsfall
WAN	*Wide Area Network*
WKS	WAN-Kommunikationsszenario
ZA	Zeitabweichung

8 XML- und Cosem-Datentypen samt Mapping auf IEC 62056-62

COSEM-Datentyp nach DIN EN 62056-62	Datentyp nach [XSD-Datentypen]	Datentyp nach [DKE-AK 142]
null-data	(Element Vorkommen minOccurs=”0”)	(OPTIONAL)
octet-string	hexBinary	octet-string
visible-string	string	(octet-string)
UTF8-string	string	UTF8-String
integer	byte	Integer8
long	short	Integer16
double-long	int	Integer32
long64	long	Integer64
unsigned	unsignedByte	Unsigned8
long-unsigned	unsignedShort	Unsigned16
double-long-unsigned	unsignedInt	Unsigned32
long64-unsigned	unsignedLong	Unsigned64
boolean	boolean	Boolean
date-time	hexBinary (restriction length=”12”)	Date-Time-Var
date	hexBinary (restriction length=”5”)	Date-Var
time	hexBinary (restriction length=”4”)	Time-Var
enum (Unsigned8)	string (enumeration value=”…”)	ENUM{}
float64	double (hexBinary(length=”8”))	
float32	float(hexBinary(length=”4”))	
bit-string	string (restriction pattern=”[0-1]{0,}“)	bit-string

9 Verzeichnis der im SMGW verwendeten *COSEM* Klassen

Name der COSEM-Interface-Class	OBIS-Kennzahl	Class ID[12]	HTTP-Verb
channel_setup	01005e318012	32797	DELETE, GET, POST, PUT
e_meter_sensor_setup	01005e318002	32809	DELETE, GET, POST, PUT
event	01005e3180a3	32819	Wird vom SMGW versendet
gateway_signed_extended_register	Abhängig von Messgröße		Wird vom SMGW versendet
fw_update	01005e3180a1	32782	PUT
generic_meter_sensor_setup	01005E318003		DELETE, GET, POST, PUT
ipv4_setup (CLS)	000019010001	42	PUT
ipv4_setup (HAN)	000019010002		PUT
ipv4_setup (WAN)	000019010003		PUT
kaf_han_wan_container	01005e318010	32795	DELETE, GET, POST, PUT
kaf_han_wan_index	01005e318011		DELETE, GET, POST, PUT
kaf_lmn_container	01005E318000	32800	DELETE, GET, POST, PUT
kaf_lmn_index	01005E318001		DELETE, GET, POST, PUT
on_demand_delivery	01005e31803b	32820	PUT
profile_generic (in der Nutzungsvariante Logbook) (Systemlog)	0000636201ff	7 (mit version=1)	GET
profile_generic (in der Nutzungsvariante Logbook) (Eichlog)	0000636202ff		GET
profile_generic (Simple Data) bei Messwerten aus taf01, taf02, taf06 oder taf07	01005e31803f		Wird vom SMGW versendet

12 Die Class-Version dieser Klassen ist stets “0“, außer bei der Klasse *profile_generic* mit *version* = *1*.

Name der COSEM-Interface-Class	OBIS-Kennzahl	Class ID[12]	HTTP-Verb
profile_generic (Simple Data) bei Messwerten aus taf09, taf10, taf14	01005e31803c		Wird vom SMGW versendet
proxy_server	01005E318017	32780	DELETE, GET, POST, PUT
qos_sq_setup	01005e318013	32799	DELETE, GET, POST, PUT
role_setup	01005e318014	32798	DELETE, GET, POST, PUT
security_module	01005e3180a2	32783	POST (Methodenaufruf)
smgw_info	01005e3180a0	32781	GET, POST
taf01	01005E318021	32785	DELETE, GET, POST, PUT
taf02	01005E318022	32786	DELETE, GET, POST, PUT
taf07	01005E318027	32791	DELETE, GET, POST, PUT
taf09	01005E318029	32793	DELETE, GET, POST, PUT
taf10	01005E31802A	32794	DELETE, GET, POST, PUT
taf14	01005E31802E	32798	DELETE, GET, POST, PUT
tr_certificate_setup	01005E3180a5	32821	PUT
user_setup	01005e318015	32796	DELETE, GET, POST, PUT

10 Verzeichnis der im SMGW verwendeten Object Identifier (OID)

ecdsaPlainSha256	"0.4.0.127.0.7.1.1.4.1.3"
ecdsaPlainSha384	"0.4.0.127.0.7.1.1.4.1.4"
ecdsaPlainSha512	"0.4.0.127.0.7.1.1.4.1.5"
ecka_eg_X963KDF_SHA256	"0.4.0.127.0.7.1.1.5.1.1.3"
ecka_eg_X963KDF_SHA384	"0.4.0.127.0.7.1.1.5.1.1.4"
ecka_eg_X963KDF_SHA512	"0.4.0.127.0.7.1.1.5.1.1.5"
aes_CBC_CMAC_Param	"0.4.0.127.0.7.1.3.1.1.1"
aes_CBC_CMAC_128	"0.4.0.127.0.7.1.3.1.1.2"
aes_CBC_CMAC_192	"0.4.0.127.0.7.1.3.1.1.3"
aes_CBC_CMAC_256	"0.4.0.127.0.7.1.3.1.1.4"
PACE_ecdhPaceAesCbcCmac128	"0.4.0.127.0.7.2.2.4.2.3"
PACE_ecdhPaceAesCbcCmac192	"0.4.0.127.0.7.2.2.4.2.4"
PACE_ecdhPaceAesCbcCmac256	"0.4.0.127.0.7.2.2.4.2.5"
rfc4211CertReqMsgs	"0.4.0.127.0.7.4.1.1.1"
ecPublicKey	"1.2.840.10045.2.1"
ecdsa_with_SHA256	"1.2.840.10045.4.3.2"
ecdsa_with_SHA384	"1.2.840.10045.4.3.3"
ecdsa_with_SHA512	"1.2.840.10045.4.3.4"
data	"1.2.840.113549.1.7.1"
signedData	"1.2.840.113549.1.7.2"
envelopedData	"1.2.840.113549.1.7.3"
contentType	"1.2.840.113549.1.9.3"
messageDigest	"1.2.840.113549.1.9.4"
signingTime	"1.2.840.113549.1.9.5"
ct_compressedData	"1.2.840.113549.1.9.16.1.9"
ct_authEnvelopedData	"1.2.840.113549.1.9.16.1.23"
alg_zlibCompress	"1.2.840.113549.1.9.16.3.8"
kpServerAuth	"1.3.6.1.5.5.7.3.1"
kpClientAuth	"1.3.6.1.5.5.7.3.2"
brainpoolP256r1	"1.3.36.3.3.2.8.1.1.7"
brainpoolP384r1	"1.3.36.3.3.2.8.1.1.11"

brainpoolP512r1	"1.3.36.3.3.2.8.1.1.13"
secp256r1 / prime256v1	"1.2.840.10045.3.1.7"
secp384r1 / prime384v1	"1.3.132.0.34"
aes128_CBC	"2.16.840.1.101.3.4.1.2"
aes128_wrap	"2.16.840.1.101.3.4.1.5"
aes128_GCM	"2.16.840.1.101.3.4.1.6"
aes192_CBC	"2.16.840.1.101.3.4.1.22"
aes192_wrap	"2.16.840.1.101.3.4.1.25"
aes192_GCM	"2.16.840.1.101.3.4.1.26"
aes256_CBC	"2.16.840.1.101.3.4.1.42"
aes256_wrap	"2.16.840.1.101.3.4.1.45"
aes256_GCM	"2.16.840.1.101.3.4.1.46"
sha256	"2.16.840.1.101.3.4.2.1"
sha384	"2.16.840.1.101.3.4.2.2"
sha512	"2.16.840.1.101.3.4.2.3"
commonName	"2.5.4.3"
serialNumber	"2.5.4.5"
country	"2.5.4.6"
organization	"2.5.4.10"
organizationalUnit	"2.5.4.11"

11 Verzeichnis der kryptografischen Kennungen

Tabelle 102: Rollen im SMGW und ihre TLS-Zertifikatskürzel

Rolle	TLS-Zertifikat
Letztverbraucher	CON_HAN_TLS_CRT
Service-Techniker	SRV_HAN_TLS_CRT
EMT	EMT_TLS_CRT
GWA	GWADM_TLS_CRT
Zähler	MTR_TLS_CRT
CLS	CLS_TLS_CRT
SMGW (WAN)	GW_WAN_TLS_CRT
SMGW (HAN)	GW_HAN_TLS_CRT
SMGW (LMN)	GW_LMN_TLS_CRT

Tabelle 103: Rollen im SMGW und ihre Signaturzertifikats- und Signaturschlüssel-Kürzel

Rolle	Signaturzertifikat	Signaturschlüssel
GWA	GWADM_SIG_CRT	GWADM_SIG_PRV
SMGW	GW_WAN_SIG_CRT	GW_WAN_SIG_PRV

12 Tabellenverzeichnis

Tabelle 1: Betriebszustände des SMGW 15
Tabelle 2: Vom SMGW unterstützte TLS-Ciphersuiten 27
Tabelle 3: Vom SMGW unterstützte EC-Kurven 27
Tabelle 4: Generische WAF-Rechtematrix 29
Tabelle 5: Allgmeine Zugriffs-Dienste des SMGW 30
Tabelle 6: HTTP-Statuscodes des SMGW auf Requests des GWA 33
Tabelle 7: HTTP-Statuscode im Fehlerfall 33
Tabelle 8: Überblick über die COSEM-ICs zur Schnittstellenkonfiguration des SMGW 34
Tabelle 9: Attribute der IC *ipv4_setup* 36
Tabelle 10: Aktualisierung der IPv4-Konfiguration der HAN-Schnittstelle 39
Tabelle 11: Aktualisierung der IPv4-Konfiguration der CLS-Schnittstelle 39
Tabelle 12: Aktualisierung der IPv4-Konfiguration der WAN-Schnittstelle 39
Tabelle 13: Überblick über die COSEM-ICs zur Profilverwaltung und -konfiguration des SMGW 40
Tabelle 14: Attribute der IC *kaf_lmn_container* 42
Tabelle 15: Attribute der IC *kaf_lmn_index* 43
Tabelle 16: Attribute der IC *e_meter_sensor_setup* 45
Tabelle 17: Mögliche Werte des Attributs *driver_reference* in IC *e_meter_sensor_setup* 51
Tabelle 18: Attribute der IC *generic_meter_sensor_setup* 53
Tabelle 19: Mögliche Werte des Attributs *driver_reference* in IC *generic_meter_sensor_setup* . 59
Tabelle 20: Anlegen eines Zählerprofils 61
Tabelle 21: Auslesen eines Zählerprofils 61
Tabelle 22: Löschen eines Zählerprofils 62
Tabelle 23: Aktualisieren eines Zählerprofils 62
Tabelle 24: Attribute der IC *kaf_han_wan_container* 63
Tabelle 25: Attribute der IC *kaf_han_wan_index* 66
Tabelle 26: Attribute der IC *channel_setup* 70
Tabelle 27: Attribute der IC *qos_sq_setup* 75
Tabelle 28: Attribute der IC *role_setup* 78
Tabelle 29: Attribute der IC *user_setup* 79
Tabelle 30: Anlegen eines HAN-/WAN-Profils 83
Tabelle 31: Auslesen eines HAN-/WAN-Profils 83
Tabelle 32: Löschen eines HAN-/WAN-Profils 84
Tabelle 33: Aktualisieren eines HAN-/WAN-Profils 84
Tabelle 34: Attribute der IC *taf01* 86
Tabelle 35: Anlegen eines *taf01* 95
Tabelle 36: Auslesen eines *taf01* 95
Tabelle 37: Löschen eines *taf01* 96
Tabelle 38: Aktualisieren eines *taf01* 96
Tabelle 39: Attribute der IC on_demand_delivery 97
Tabelle 40: Aktualisieren eines IC *on_demand_delivery* 99
Tabelle 41: Attribute der IC *taf02* 100
Tabelle 42: Anlegen eines *taf02* 113
Tabelle 43: Auslesen eines *taf02* 113
Tabelle 44: Löschen eines *taf02* 113
Tabelle 45: Aktualisieren eines *taf02* 114
Tabelle 46: Attribute der IC *taf07* 115
Tabelle 47: Anlegen eines *taf07* 124
Tabelle 48: Auslesen eines *taf07* 124
Tabelle 49: Löschen eines *taf07* 125
Tabelle 50: Aktualisieren eines *taf07* 125
Tabelle 51: Attribute der IC *taf09* 126
Tabelle 52: Beschreibung der Methoden der IC *taf09* 136
Tabelle 53: Anlegen eines *taf09* 136
Tabelle 54: Auslesen eines *taf09* 137
Tabelle 55: Löschen eines *taf09* 137

Tabelle 56: Aktualisieren eines *taf09* 137
Tabelle 57: Attribute der IC *taf10* 138
Tabelle 58: Beschreibung der Methoden der IC *taf10* 148
Tabelle 59: Anlegen eines *taf10* 148
Tabelle 60: Auslesen eines *taf10* 149
Tabelle 61: Löschen eines *taf10* 149
Tabelle 62: Aktualisieren eines *taf10* 149
Tabelle 63: Attribute der IC *taf14* 150
Tabelle 64: Anlegen eines *taf14* 160
Tabelle 65: Auslesen eines *taf14* 160
Tabelle 66: Löschen eines *taf14* 161
Tabelle 67: Aktualisieren eines *taf14* 161
Tabelle 68: Attribute der IC *proxy_server* 162
Tabelle 69: Anlegen eines CLS-Profils 165
Tabelle 70: Auslesen eines CLS-Profils 165
Tabelle 71: Löschen eines CLS-Profils 166
Tabelle 72: Aktualisieren eines CLS-Profils 166
Tabelle 73: Beschreibung der Attribute der IC *security_module* 169
Tabelle 74: Beschreibung der Methoden der IC *security_module* 171
Tabelle 75: Aufruf einer Methode der IC *security_module* 178
Tabelle 76: Attribute der IC *tr_certificate_setup* 179
Tabelle 77: Aktualisierung der Zertifikate des SMGW 182
Tabelle 78: Überblick über die COSEM-ICs zur Verwaltung der Logdaten 185
Tabelle 79: Attribute der IC *profile_generic in der Nutzungsvariante Logbook* 186
Tabelle 80: Attribute der IC *log_entry* 187
Tabelle 81: Auslesen des Systemlogs 189
Tabelle 82: Auslesen des Eichlogs 189
Tabelle 83: Auslesen eines Logbuchs mit Verwendung von Query-Parametern 190
Tabelle 84: Attribute der IC *event* 191
Tabelle 85: Attribute der IC *smgw_info* 195
Tabelle 86: Methoden der IC *smgw_info* 197
Tabelle 87: Auslesen des IC *smgw_info* 199
Tabelle 88: Ausführen der Methode *reset* der IC *smgw_info* 199
Tabelle 89: Ausführen der Methode *selfTest* der IC *smgw_info* 200
Tabelle 90: Überblick über die COSEM-ICs in WAF2 „Zugriff auf Dienste des GWA“ 200
Tabelle 91: Attribute der IC *profile_generic* in der Nutzungsvariante *Simple Data* 201
Tabelle 92: Attribute der IC *gateway_signed_extended_register* 202
Tabelle 93: Übertragung von Messdaten an den GWA/EMT 205
Tabelle 94: Überblick über die COSEM-ICs in WAF3 „Alarmierung und Benachrichtigung“ 205
Tabelle 95: Übertragen von Alarmen an den GWA 206
Tabelle 96: Überblick über die COSEM-ICs in WAF5 „Übertragung von Daten an EMT“ 207
Tabelle 97: Bedeutung der Bits des SMGW-Statusworts 208
Tabelle 98: Attribute der IC *fw_update* 220
Tabelle 99: Aktualisierung der IC *fw_update* 221
Tabelle 100: Abfrage der Firmware Datei beim GWA 222
Tabelle 101: Eichrechtlich relevante Fehlermeldungen 228
Tabelle 102: Rollen im SMGW und ihre TLS-Zertifikatskürzel 236
Tabelle 103: Rollen im SMGW und ihre Signaturzertifikats- und Signaturschlüssel-Kürzel 236

13 Abbildungsverzeichnis

Abbildung 1: Einbettung des *Smart Meter Gateways* in seine Einsatzumgebung 7
Abbildung 2: Das ETH *Smart Meter Gateway* (unterschiedliche Bedruckungsvarianten) 8
Abbildung 3: Beispiel eines 2D-Barcodes 12
Abbildung 4: Zustände des Siegels des Smart Meter Gateways 13
Abbildung 5: Protokollstapel für die WAN-Kommunikation gemäß [BSI TR-03109-1] 26
Abbildung 6: Objektmodell des SMGW 32
Abbildung 7: Schematischer Aufbau der CLS-Proxy-Kommunikation 210
Abbildung 8: Protokollablauf bei SOCKSv5 211
Abbildung 9: Sequenzdiagramm für den Anwendungsfall „Wake-Up Service“ 216
Abbildung 10: Struktur des Wake-Up Pakets 217

14 Literaturverzeichnis

[BSI TR-03109]	Technische Richtlinie BSI TR-03109, Version 1.1, Bundesamt für Sicherheit in der Informationstechnik, 22.09.2021
[BSI TR-03109-1]	Technische Richtlinie BSI TR-03109-1, Anforderungen an die Interoperabilität der Kommunikationseinheit eines Messsystems, Version 1.1, Bundesamt für Sicherheit in der Informationstechnik, 17.09.2021
[BSI TR-03109-1-I]	BSI TR-03109-1 Anlage I, CMS-Datenformat für die Inhaltsdatenverschlüsselung und -signatur, Version 1.0, Bundesamt für Sicherheit in der Informationstechnik, 18.03.2013
[BSI TR-03109-1-VI]	Technische Richtlinie BSI TR-03109-1 Anlage VI, Betriebsprozesse, Version 1.0, Bundesamt für Sicherheit in der Informationstechnik, 18.03.2013
[BSI TR-03109-2]	Technische Richtlinie BSI TR-03109-2, Smart Meter Gateway – Anforderungen an die Funktionalität und Interoperabilität des Sicherheitsmoduls, Version 1.1, Bundesamt für Sicherheit in der Informationstechnik, 15.12.2014
[BSI TR-03109-2-UC]	Technische Richtlinie BSI TR-03109-2, Anhang: Smart Meter Gateway – Sicherheitsmodul – Use Cases, Version 1.1, Bundesamt für Sicherheit in der Informationstechnik, 17.12.2014
[BSI TR-03109-3]	Technische Richtlinie BSI TR-03109-3, Kryptographische Vorgaben für die Infrastruktur von intelligenten Messsystemen, Version 1.1, Bundesamt für Sicherheit in der Informationstechnik, 17.04.2014
[BSI TR-03109-4]	Technische Richtlinie BSI TR-03109-4, Smart Metering PKI - Public Key Infrastruktur für Smart Meter Gateways, Version 1.2.1, Bundesamt für Sicherheit in der Informationstechnik, 09.08.2017
[BSI TR-03109-6]	Technische Richtlinie BSI TR-03109-6, Smart Meter Gateway Administration, Version 1.0, Bundesamt für Sicherheit in der Informationstechnik, 26.11.2015
[BSI TR-03116-3]	Technische Richtlinie BSI TR-03116-3, Kryptographische Vorgaben für Projekte der Bundesregierung, Teil 3 - Intelligente Messsysteme, Stand 2022, Bundesamt für Sicherheit in der Informationstechnik, 23.02.2022
[DIN 43863-5]	DIN 43863-5, Herstellerübergreifende Identifikationsnummer für Messeinrichtungen, April 2012, DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE
[DKE-AK 142]	Smart Meter Gateway, Teil 2: Klassen-Definition zur TR 03109 nach COSEM, Version 1.1, 28.10.2016, DKE AK 461.0.142
[EIA RS-485]	EIA Standard RS-485, Electrical Characteristics of Generators and Receivers for Use in Balanced Multipoint Systems, ANSI/TIA/EIA-485-A-98, 1983/R2003
[EN 13757-1]	DIN EN 13757-1:2014; Kommunikationssysteme für Zähler und deren Fernablesung Teil 1: Datenaustausch, Januar 2015, DIN Deutsches Institut für Normung e. V.
[EN 13757-3]	DIN EN 13757-3:2013, Kommunikationssysteme für Zähler und deren Fernablesung Teil 3: Spezielle Anwendungsschicht, August 2013, DIN Deutsches Institut für Normung e. V.

[EN 13757-4] DIN EN 13757-4:2014-02, Kommunikationssysteme für Zähler und deren Fernablesung Teil 4: Zählerauslesung über Funk, Fernablesung von Zählern im SRD-Band von 868 MHz bis 870 MHz, Februar 2014, DIN Deutsches Institut für Normung e. V.

[FIPS 180-4] NIST, FIPS Pub. 180-4, Secure Hash Standard, 2015, National Institute of Standards and Technology

[FIPS 197] NIST, FIPS Pub. 197, Advanced Encryption Standard (AES), 2001, National Institute of Standards and Technology

[FNN_LH_BZ] FNN Lastenheft Basiszähler Funktionale Merkmale Version 1.4.1, 08. Mai 2018, Forum Netztechnik/Netzbetrieb im VDE

[IEC-62056-6-1] IEC-62056-6-1, Datenkommunikation der elektrischen Energiemessung, Teil 6-1: OBIS Object Identification System, 2017, International Electrotechnical Commission

[IEC-62056-6-2] IEC-62056-6-2, Datenkommunikation der elektrischen Energiemessung - DLMS/COSEM, Teil 6-2: COSEM Interface classes, 2017, International Electrotechnical Commission

[IEEE 802.3i] IEEE Std 802.3i-1990 (Clauses 13 and 14), 10 Mb/s UTP MAU, 10 BASE-T

[OMS-2] Open Metering System Specification, Volume 2, Primary Communication, Issue 4.1.2, 2016-12-16, http://www.oms-group.org

[PTB-A 50.8] PTB-Anforderungen PTB-A 50.8, Smart Meter-Gateway, Entwurf vom 23.10.2013, Physikalisch-Technische Bundesanstalt

[Quarz RTC] Technisches Datenblatt des Quarz-Kristalls KX-327NHT, 13.01.2014, GEYER ELECTRONIC e.K.

[Quarz System] Technisches Datenblatt des Quarz-Kristalls KX-7T, 29.01.2013, GEYER ELECTRONIC e.K.

[RFC 791] IETF RFC 791, Internet Protocol, Darpa Internet Program, Protocol Specification, September 1981, http://www.rfc-editor.org/rfc/rfc791.txt

[RFC 793] IETF RFC 793, Transmission Control Protocol, Darpa Internet Program, Protocol Specification, September 1981, http://www.rfc-editor.org/rfc/rfc793.txt

[RFC 826] IETF RFC 826, An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware, David C. Plummer, November 1982, http://www.rfc-editor.org/rfc/rfc826.txt

[RFC 1928] IETF RFC 1928, SOCKS Protocol Version 5, March 1996, http://www.rfc-editor.org/rfc/rfc1928.txt

[RFC°2460] IETF RFC 2460, Internet Protocol, Version 6 (IPv6) Specification, December 1998, http://www.rfc-editor.org/rfc/rfc2460.txt

[RFC 2616] IETF RFC 2616, Hypertext Transfer Protocol - HTTP/1.1, R. Fielding, J. Gettys, J. Mogul, H. Frystyk, P. Masinter, P. Leach, T. Berners-Lee, http://rfc-editor.org/rfc/rfc2616.txt

[RFC 4493] IETF RFC 4493, The AES-CMAC Algorithm, J. H. Song, J. Lee, T. Iwata, June 2006, http://www.rfc-editor.org/rfc/rfc4493.txt

[RFC°4862] IETF RFC 4862, IPv6 Stateless Address Autoconfiguration, September 2007, http://www.rfc-editor.org/rfc/rfc4862.txt

[RFC 5084] IETF RFC 5084, Using AES-CCM and AES-GCM Authenticated Encryption in the Cryptographic Message Syntax (CMS), R. Housley, November 2007, http://www.rfc-editor.org/rfc/rfc5084.txt

[RFC 5114] IETF RFC 5114, Additional Diffie-Hellman Groups for Use with IETF Standards, M. Lepinski, S. Kent, January 2008, http://www.rfc-editor.org/rfc/rfc5114.txt

[RFC 5246] IETF RFC 5246, Transport Layer Security (TLS) Version 1.2, T. Dierks, E. Rescorla, August 2008, http://www.rfc-editor.org/rfc/rfc5246.txt

[RFC 5289] IETF RFC 5289, TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), E. Rescorla, August 2008, http://www.rfc-editor.org/rfc/rfc5289.txt

[RFC 5639] IETF RFC 5639, Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation, M. Lochter, J. Merkle, March 2010, http://www.rfc-editor.org/rfc/rfc5639.txt

[RFC 5652] IETF RFC 5652, Cryptographic Message Syntax (CMS), R. Housley, September 2009, http://www.rfc-editor.org/rfc/rfc5652.txt

[RFC 5905] IETF RFC 5905, Network Time Protocol Version 4: Protocol and Algorithms Specification, D. Mills, U. Delaware, J. Burbank, W. Kasch, June 2010, http://www.rfc-editor.org/rfc/rfc5905.txt

[RFC 7233] IETF RFC 7233, Hypertext Transfer Protocol (HTTP/1.1): Range Requests, R. Fielding, Y. Lafon, J. Reschke, June 2014, http://www.rfc-editor.org/rfc/rfc7233.txt

[Security_Target] Security Target, SMGW Version 2.0, 30.09.2022, Version 5.4, Power Plus Communications AG

[Sichere_Auslieferung] Auslieferungs- und Fertigungsprozeduren, Anhang Sichere Auslieferung, Version 1.4, 12.05.2021, Power Plus Communications AG

[SOCKS-SSL] IETF Internet-Draft, draft-ietf-aft-socks-ssl-00, Secure Sockets Layer for SOCKS Version 5, M. VanHeyningen, March 1997, http://tools.ietf.org/id/draft-ietf-aft-socks-ssl-00.txt

[SMGW_Logging] Logmeldungen, SMGW Version 2.0, Version 3.3, 18.07.2022, Power Plus Communications AG

[SMGW_PP] Protection Profile for the Gateway of a Smart Metering System (Gateway PP), v1.3, 31.03.2014, Bundesamt für Sicherheit in der Informationstechnik

[XSD-Datentypen] W3C, XML Schema Part 2: Datatypes Second Edition, W3C Recommendation, 28 October 2004, http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/

Power Plus Communications AG
Dudenstraße 6, 68167 Mannheim
Tel. 00 49 621 40165 100 | Fax. 00 49 621 40165 111
info@ppc-ag.de | www.ppc-ag.de