

| | | |
|--|--|--|
| | VDE-AR-E 2XXX | |
| | Dies ist eine VDE-Anwendungsregel im Sinne von VDE 0022. Sie ist nach der Durchführung des vom VDE-Präsidium beschlossenen Genehmigungsverfahrens unter der oben angeführten Nummer in das VDE-Vorschriftenwerk aufgenommen und in der „etz Elektrotechnik + Automation“ bekannt gegeben worden. | |
| <p><i>Entwurf</i></p> <p>Schnittstelle zur Anzeige von Smart Meter Gateway Daten für den Letztverbraucher (IF_3D_CON)</p> | | |
| | | |

Inhalt

| | | |
|-------|--|----|
| 1 | Anwendungsbereich..... | 1 |
| 2 | Abkürzungsverzeichnis | 3 |
| 3 | Normative Verweisungen und Abgrenzungen | 4 |
| 3.1 | REQ.21 – Energy Service Provider Interface..... | 4 |
| 3.2 | BSI TR03109..... | 4 |
| 3.3 | PTB-A 50.8 | 4 |
| 3.4 | FNN-Lastenheft Smart-Meter-Gateway | 5 |
| 3.5 | FNN-Lastenheft Daten und Prozesse für den Display-Datendienst (3D)..... | 5 |
| 3.6 | Smart Meter Gateway, Teil 2: Klassen-Definition zur TR 03109 nach COSEM..... | 5 |
| 3.7 | Abgrenzung..... | 6 |
| 4 | Systemumgebung Display-Datendienst und IF_3D_CON Schnittstelle | 7 |
| 5 | Informationen der Schnittstelle IF_3D_CON zur Überprüfung der Rechnung durch den Letztverbraucher | 9 |
| 5.1 | Anforderungen an das Datenformat | 9 |
| 5.2 | Aufbau der Modellierung | 12 |
| 5.3 | Nutzung der Schnittstelle IF_3D_CON für Zwecke der Messdatenübertragung ohne eichrechtliche Relevanz..... | 13 |
| 5.4 | Übersicht über das Datenmodell..... | 13 |
| 5.4.1 | Allgemeine Festlegungen | 13 |
| 5.4.2 | Basisdaten | 14 |
| 5.4.3 | Prüfungsdaten | 19 |
| 5.4.4 | Beschreibung zusätzlicher Klassen | 26 |
| 6 | Regeln zur Anwendung des Datenmodells..... | 28 |
| 6.1 | Basisdaten | 28 |
| 6.2 | Prüfungsdaten | 37 |
| 7 | Sicherheits- und Datenintegritätsanforderungen an die Schnittstelle IF_3D_CON | 49 |
| 7.1 | Überprüfung der inneren Signatur | 49 |
| 8 | Services der Schnittstelle IF_3D_CON | 50 |
| 8.1 | Beschreibung des XML-Dateidownloads zur Rechnungsprüfung | 50 |
| 8.2 | Beschreibung des XML-Dateidownloads zur Darstellung von Verbrauchsdaten (Energy Awareness) | 51 |
| 8.3 | Regelungen zur Dateibereitstellung | 51 |
| 8.3.1 | Namensschema für die Dateibereitstellung..... | 51 |
| 8.3.2 | Automatisierung der Dateibereitstellung..... | 52 |

| | | |
|------|-------------------------|----|
| 9 | Literaturverweise | 53 |
| 10 | Anhang..... | 54 |
| 10.1 | XML-Schema | 54 |

1 Anwendungsbereich

Für den Roll Out der intelligenten Messsysteme (iMSys) gemäß des Gesetzes zur Digitalisierung der Energiewende wird eine gegen Manipulation geschützte und eichrechtskonforme Anzeige der Verbrauchs- und Erzeugungsdaten für Zwecke der Rechnungsprüfung benötigt. Mit Hilfe dieser sollen Letztverbraucher in die Lage versetzt werden beweissichere Überprüfungen von Abrechnungen der Energieversorgungsunternehmen vornehmen zu können. Zusätzlich soll dem Letztverbraucher die Möglichkeit gegeben werden, seine derzeitigen Verbrauchswerte abzurufen.

Die Anforderungen an eine solche Anzeige werden im Wesentlichen in 2 Dokumenten beschrieben:

Die Technische Richtlinie BSI TR-03109-1 „Anforderungen an die Interoperabilität der Kommunikationssicherheit eines intelligenten Messsystems“ (kurz: TR) [BSI TR-03109-1], herausgegeben vom Bundesamt für Sicherheit in der Informationstechnik (BSI), beschreibt wie verschiedene Anwendungsfälle mit intelligenten Messsystem umgesetzt werden können und welche Daten für den jeweiligen Anwendungsfall benötigt werden. Im Sinne dieses Dokumentes sind dabei insbesondere die Anwendungsfälle für Tarifierung und Bilanzierung (Kapitel 4.2.2) relevant, in denen 8 sogenannte Tarifierungsanwendungsfälle (TAFs) beschrieben werden. Daneben werden insbesondere Anforderungen an den Datenschutz und die Datensicherheit beschrieben.

Die PTB-Anforderungen PTB-A 50.8 „Smart Meter Gateway“ [PTB-A50.8], herausgegeben von der Physikalisch-Technischen Bundesanstalt (PTB), beschreiben Anforderungen an intelligente Messsysteme aus eichrechtlicher Sicht und geht dazu zum Teil auf die Technische Richtlinie des BSI ein, indem es zum Beispiel benötigte Daten in eichrechtlich relevant einstuft und zusätzlich alle für die Rechnungsprüfung benötigten Daten benennt. Zusätzlich beschreibt das Dokument, wie eine mit dem Eichrecht konforme Übertragung dieser Daten möglich ist.

Offen blieb zunächst die Thematik der Verortung der Anzeige Smart Meter-Daten für den Letztverbraucher nach Vorgaben von TR und PTB. Da bei ausgewählten Tarifierungsarten nicht alle Daten lokal im SMGW¹ verfügbar sein werden, ist nur die Bereitstellung einer Anzeigemöglichkeit über „Wide Area Network“ (WAN) für Letztverbraucher verpflichtend.

Durch die Einführung einer neuen Instanz eines neuen Externen Marktteilnehmers (EMT), dem sogenannten Display-Datendienstes („DDD“ oder „3D“), zur Bereitstellung der Metering-Daten aus dem SMGW und ergänzender Tarifdaten vom Lieferanten für Letztverbraucher wird eine entsprechende Infrastruktur für die Anzeige von Smart Metering-Daten für Letztverbraucher über das WAN geschaffen. Im Rahmen dieser VDE-Anwendungsregel wird die Datenschnittstelle des DDD zu einer externen Anzeigelösung spezifiziert. Dabei wird auf „Green Button“ als einen etablierten Standard zur Datenbereitstellung von Smart Metering Daten für Letztverbraucher zurückgegriffen. Dieser international eingesetzte US-Standard beschreibt insbesondere ein Datenformat für die Übertragung von Messwerten an den Letztverbraucher. Dieser Standard wird in dieser VDE- Anwendungsregel für deutsche Anforderungen erweitert.

¹ Zunächst war die Schnittstelle für eine Anzeige der Verbrauchs- und Erzeugungsdaten des Letztverbrauchers über die „Home Area Network“ (HAN) Schnittstelle des SMGWs vorgesehen. Um den TAF7 zu realisieren wäre es bei der Lösung über die HAN-Schnittstelle nötig, dass die Tarifdaten der Lieferanten an die SMGWs gesendet werden müssten. Je nach Anzahl der Energielieferanten und Netzbetreiber, sowie der unterschiedlichen Tarife, müssten entsprechend viele Daten über GWA an die SMGWs gesendet werden und würden dadurch auch einen höheren Aufwand bei der Konfiguration der Kommunikationsprofile im SMGW bedeuten.

Durch die Einführung einer standardisierten Datenschnittstelle für Metering Daten über Display-Datendienst werden Hersteller von Metering-Lösungen in die Lage versetzt eichrechtskonforme Anzeigelösungen für Letztverbraucher über WAN zu realisieren, die mit DDD anderer Hersteller kompatibel sein werden. Durch die Nutzung des international eingesetzten US Standards Green Button soll zum einen dem Letztverbraucher ermöglicht werden Applikationen internationaler Anbieter für Analyse eigener Metering Daten zu verwenden. Zum anderen wird die Übertragbarkeit der Display-Lösung auf internationale Märkte vereinfacht.

Im Rahmen dieser VDE Anwendungsregel werden die erforderlichen Erweiterungen des Green Button-Standards beschrieben, die notwendig sind, um ergänzende Anforderungen gemäß dem Gesetz zur Digitalisierung der Energiewende einzuhalten. Hierzu gehören insbesondere die zusätzlichen Anforderungen an die Datenintegrität und zusätzliche Anforderungen an den Datenumfang für Zwecke der Rechnungsprüfung dokumentiert. Diese Anwendungsregel beschreibt somit eine Schnittstelle zur Bereitstellung von Letztverbraucherdaten durch den Display-Datendienst zur eichrechtlich konformen Anzeige und Rechnungsprüfung an einem geeigneten Kundendisplay.

2 Abkürzungsverzeichnis

| | |
|---------|---|
| 3D | Display-Datendienst |
| BfDI | Bundesbeauftragter für den Datenschutz und die Informationsfreiheit |
| BSI | Bundesamt für Sicherheit in der Informationstechnik |
| COSEM | Companion Specification for Energy Metering (Teil der IEC62056-Normenreihe) |
| DDD | Display-Datendienst |
| EDIFACT | Electronic Data Interchange For Administration, Commerce and Transport |
| EMT | Externer Marktteilnehmer |
| ESPI | Energy Service Provider Interface |
| FNN | Forum Netztechnik/Netzbetrieb im VDE |
| HAN | Home Area Network |
| iMSys | Intelligentes Messsystem |
| kWh | Kilowattstunden |
| MWh | Megawattstunden |
| NAESB | North American Energy Standards Board |
| OBIS | Object Identification System (Teil der IEC62056-Normenreihe) |
| PTB | Physikalisch-Technische Bundesanstalt |
| SMGW | Smart Meter Gateway |
| TAF | Tarifanwendungsfall |
| TLS | Transport Layer Security |
| TOU | Time-Of-Use |
| TR | Technische Regel |
| UML | Unified Modelling Language |
| UOM | Unit-of-Measurement |
| VDE | Verband der Elektrotechnik Elektronik Informationstechnik e.V. |
| WAN | Wide Area Network |
| XML | Extensible Markup Language |
| XSD | XML-Schema Definition |

3 Normative Verweisungen und Abgrenzungen

3.1 REQ.21 – Energy Service Provider Interface

Aus den USA stammt ein bereits etabliertes Datenformat für den Transport von Energieverbrauchs- und –erzeugungsdaten zu Letztverbrauchern, verbunden mit einer Referenzarchitektur für eine automatische und kaskadierende Autorisierung zur Weitergabe von Letztverbraucherdaten an Dritte. Dieses Datenformat ist seit Herbst 2011 vom North American Energy Standards Board (NAESB) als „Energy Service Provider Interface“ (ESPI, REQ.21) [NAESB REQ.21] standardisiert. Dieser bildet die Grundlage für die unter dem Marketingnamen „Green Button“ veröffentlichte Teilmenge für die Übertragung von Energieverbrauchsdaten zu Endkunden. Rund um Green Button wurden erfolgreich Anwendungen implementiert und werden für die Kommunikation von Energieverbrauchsdaten zum Letztverbraucher in den USA genutzt. Green Button wird nach eigenen Angaben inzwischen von mehr als 60 Millionen Haushalten genutzt.

Weitere Informationen zu Green Button können auf folgender Website abgerufen werden: <http://www.greenbuttondata.org>.

3.2 BSI TR03109

Die Technische Richtlinie BSI TR-03109-1 wurde vom Bundesamt für Sicherheit in der Informationstechnik veröffentlicht und beschreibt technische Vorgaben für die Übertragung von Daten vom und zum Smart Meter Gateway. In Sinne dieses Dokuments von besonderer Relevanz ist dabei der Übertragungsweg vom Smart Meter Gateway bis zum EMT. Den externen Marktteilnehmer stellt im Fall der Display Daten Dienst dar, der diese Daten im Sinne dieser Anwendungsregel an den Letztverbraucher weiterleitet.

Die in der Technischen Regel beschriebenen Anwendungsfälle für Tarifierung und Bilanzierung (Kapitel 4.2.2), die sogenannten Tarifierungsanwendungsfälle (TAFs) bilden einen Rahmen zur Nutzung der rohen Messdaten, zur Generierung von daraus abgeleiteten Messwerten und damit schlussendlich zu Zwecken der Abrechnung innerhalb des Smart Meter Gateways. Einen besonderen Anwendungsfall beschreibt dabei der Tarifierungsanwendungsfall 7. Dieser beschreibt eine zentrale Tarifierung, die auch außerhalb des Smart Meter Gateways (SMGW) möglich ist. Der TAF erlaubt die Erfassung und Versendung von Zählerstandsgängen an autorisierte Dritte, die darauf basierend eine Tarifierung in ihren eigenen Systemen vornehmen. Die Anwendung der Tarifierung auf Basis der Rohdaten muss für den Letztverbraucher aber in jedem Fall zum Zweck der Rechnungsprüfung nachvollzogen werden können.

Für die Sicherstellung der Datenauthentizität und Datenintegrität der Letztverbraucherdaten auf diesen Weg sind insbesondere die benötigten Vorgaben zur Berechnung und Prüfung der inneren Signatur zu beachten, die in der Technischen Regel aufgeführt werden.

3.3 PTB-A 50.8

Die PTB-Anforderungen PTB-A 50.8 „Smart Meter Gateway“, herausgegeben von der Physikalisch-Technischen Bundesanstalt (PTB), beschreiben Anforderungen an intelligente Messsysteme aus eichrechtlicher Sicht. Insbesondere legt das Dokument fest, welche Informationen dem Letztverbraucher angezeigt werden müssen, damit eine Rechnungsprüfung in den jeweiligen Tarifierungsanwendungsfällen durchgeführt werden kann.

Kapitel 7 der PTB-Anforderungen legt fest, dass die Rechnungsprüfung mittels einer Transparenzsoftware erfolgen kann. Die Transparenzsoftware dient als Hilfsmittel für den Letztverbraucher, die Rechnung, die der Rechnungssteller mit nicht der eichrechtlichen Überwachung unterliegender Software erstellt hat, zu überprüfen und damit den Geschäftsvorgang abzuschließen. Das von der Software erzeugte Ergebnis soll beweiskräftig sein und im Zweifelsfall als Nachweis dienen, dass Angaben in der Rechnung falsch sind. In dieser Anwendungsregel wird die Übertragung der dazu benötigten Daten von Display Daten Dienst an die Transparenzsoftware beschrieben.

3.4 FNN-Lastenheft Smart-Meter-Gateway

Das "FNN-Lastenheft Smart-Meter-Gateway" [FNN SMGW] beschreibt die funktionalen Merkmale des Smart-Meter-Gateways (SMGW) als Kommunikationsmodul des intelligenten Messsystems gemäß den gesetzlichen Vorgaben. Hier werden auch die Anwendungsfälle, der interne Datenfluss und die daraus resultierenden Parameter definiert. Des Weiteren werden die zu unterstützenden Zählerprotokolle beschrieben. Ein SMGW nach dem Lastenheft Smart-Meter-Gateway kann dabei durchaus auch ohne Elektrizitätszähler und nur in Verbindung mit Sensoren anderer Sparten eingesetzt werden.

3.5 FNN-Lastenheft Daten und Prozesse für den Display-Datendienst (3D)

Das FNN-Lastenheft Daten und Prozesse für den Daten-Displaydienst (3D) [FNN 3D-EMT] dient der Spezifikationen, von Prozessen und Daten, die erforderlich sind, um die für eine eichrechtskonforme Anzeige benötigten Daten für einen Letztverbraucher im WAN bereit zu stellen. Der Display-Datendienst dient dabei als zentrale Sammelstelle von Mess- und eichrechtlich benötigten Daten zur Weitergabe an eine Anzeigesoftware, z.B. eine Transparenzsoftware, welche eine eichrechtlich konforme Überprüfung von Abrechnungsdaten ermöglicht.

Folgende Festlegungen werden dazu in diesem Dokument getroffen:

- Spezifikation der Formate und Verfahren der Daten zwischen SMGW und Daten-Displaydienst (3D)
- Spezifikation für Methoden der 3D-Software zur Abfrage der Eichlog-Daten vom Admin
- Spezifikation der Verwaltungsprozeduren zwischen dem Vertragspartner des Letztverbrauchers, dem 3D-Portal und dem Gateway-Administrator

3.6 Smart Meter Gateway, Teil 2: Klassen-Definition zur TR 03109 nach COSEM

Die Spezifikation Smart Meter Gateway, Teil 2: Klassen-Definitionen zur TR 03109 nach COSEM [FNN COSEM] ergänzt die vorhandenen normativen Dokumente um weitere Festlegungen, mit denen Datenmodelle und Methoden für die Umsetzung der Anforderungen nach TR 03109 des BSI vereinbart werden können.

Die Spezifikation ist Teil der in Arbeit befindlichen Beschreibung zum Smart-Meter-Gateway. Sie ergänzt für diesen Zweck die in den normativen Dokumenten DIN EN 62056-62 (COSEM-Interface-Classes) und DIN EN 62056-61 (OBIS) bereitgestellten Definitionen um jene Elemente, die aus nationaler Sicht zusätzlich benötigt werden.

3.7 Abgrenzung

Diese Anwendungsregel beschreibt, aufbauend auf dem Green Button Datenformat eine Erweiterung für dieses Datenformat, um dem Letztverbraucher in Deutschland die eichrechtliche Rechnungsüberprüfung anhand einer Display-Software zu ermöglichen. Bestandteile des Green Button Datenformats, welche von diesem übernommen werden, sind, sofern in diesem Dokument aufgeführt, entsprechend gekennzeichnet².

Die definierten Erweiterungen zum Green Button Datenformat sind, dort wo es sinnvoll erschien, in weiten Teilen von den Definitionen der COSEM-Klassen für den Nachrichtenaustausch zwischen SMGW-Admin und SMGW bzw. SMGW und 3D-EMT entlehnt. Dies soll dabei unterstützen ein Mapping von vorhandenen Daten im 3D-EMT auf das hier definierte Format möglichst einfach zu halten.

² Es ist zu beachten, dass bei einer Implementierung der Anwendungsregel, der NAESB Standard REQ.21 ESPI (NAESB 2014) von der NAESB für die Implementierung der IF_3D_CON-Schnittstelle zu erwerben ist!

4 Systemumgebung Display-Datendienst und IF_3D_CON Schnittstelle

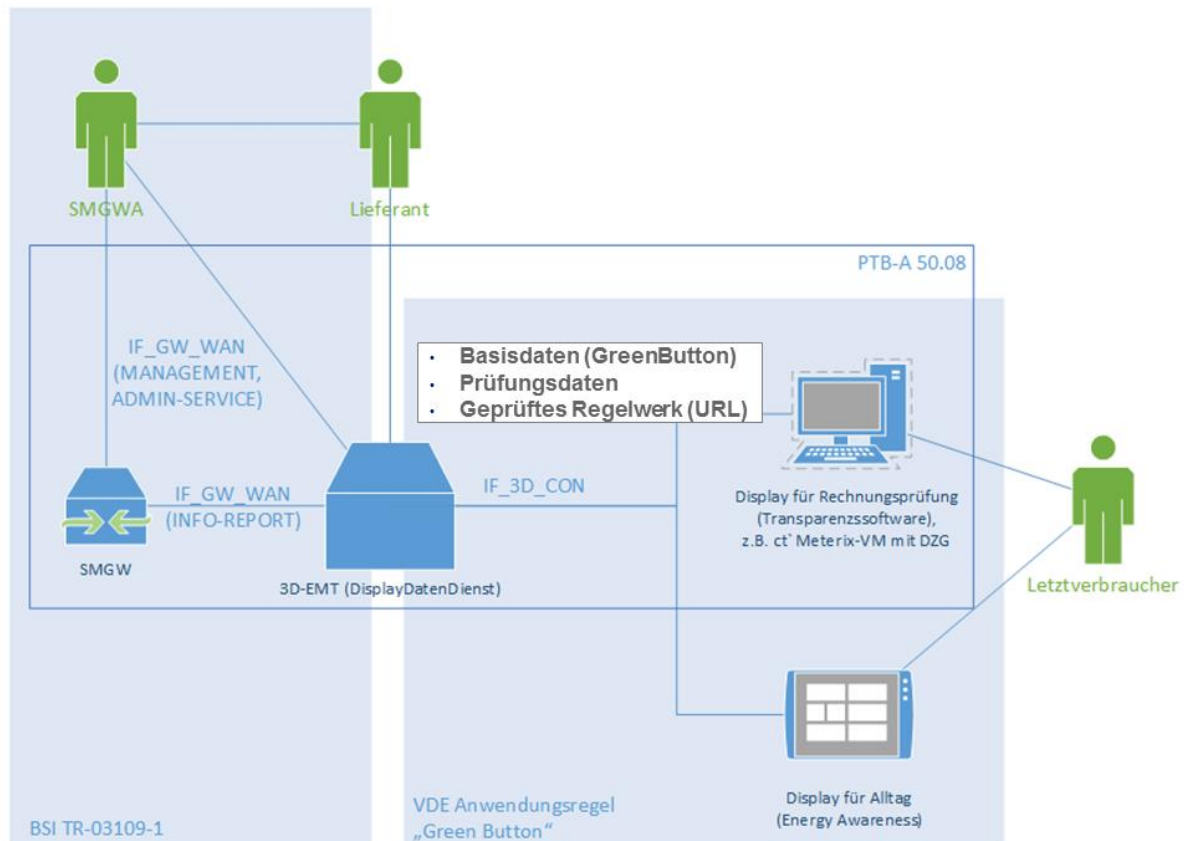


Abbildung 1: Übersicht der Systemumgebung

Im Abbildung 1 ist ein Ausschnitt der Smart Grid Systemarchitektur zu sehen. Die Übersicht zeigt 4 Domänen die es zu beachten gilt. Domäne 1 beinhaltet die Schnittstelle zum EMT Display-Datendienst (3D-EMT) sowie den Display-Datendienst selbst. Der Display-Datendienst ist verantwortlich für die das Zusammenführen von Daten des Letztverbraucher. Die entsprechenden Daten und Funktionen werden in anderen Dokumenten spezifiziert. Der Display-Datendienst stellt die Letztverbraucherdaten im Green Button Datenformat bzw. im erweiterten Green Button Datenformat nach dieser Anwendungsregel zur Rechnungsprüfung über das 3D-Portal zur Verfügung. Das 3D-Portal ist eine Portallösung zur Bereitstellung von Energieverbrauchsdaten und deren Visualisierung für den Letztverbraucher. Die Domäne 2 stellt die in diesem Dokument beschriebene Schnittstelle „IF_3D_CON“ dar, d.h. Die Datenübertragung vom Display-Datendienst zum Letztverbraucherportal. Die Domäne 3 stellt die Nutzung eines Kundendisplays auf Basis der Daten für den Alltag z.B. für Zwecke der Energy-Awareness dar. Hierfür sind keine besonderen Rahmenbedingungen festgelegt, jedoch sollten für diese Domäne auch Sicherheitsvorgaben wie in Abschnitt 7 beachtet werden. Für diese Domäne ist die Nutzung des Green Button Datenformats (spezifiziert in NAESB REQ. 21 ESPI), ohne die im Weiteren in diesem Dokument spezifizierten Erweiterung empfohlen, um die Kompatibilität zu existierender Green Button Software zu wahren. Ziel dieser Anwendungsregel ist die Schnittstellendefinition zur Bereitstellung von Letztverbraucherdaten zur Rechnungsprüfung in Domäne 4.

Die Rechnungsprüfung muss mittels einer Transparenzsoftware in einer sicheren Umgebung nach PTB-A 50.8 [PTB-A50.8] realisiert werden.

5 Informationen der Schnittstelle IF_3D_CON zur Überprüfung der Rechnung durch den Letztverbraucher

5.1 Anforderungen an das Datenformat

Eichrechtlich benötigte Informationen für die Rechnungsprüfung durch den Letztverbraucher werden in den PTB-Anforderungen PTB-A 50.8 beschrieben und wurden daraus entnommen und abgeleitet. Tabelle 1 listet diese Anforderungen hier noch einmal zur Übersicht auf. Diese bezieht sich insbesondere auf die Tabelle 4.17 der PTB-A. Die einzelnen Anforderungen sind dabei durchnummeriert, um direkt darauf referenzieren zu können.

Tabelle 1: Eichtechnisch relevante Informationen für die Rechnungsprüfung nach PTB-A 50.8

| Anforderung | Größe und Art der Information | Einheit | Bemerkung | TAF |
|--------------|---|------------------------------|---|----------|
| PTB01 | Kennung des Letztverbrauchers | | Eindeutiger Name oder Benutzerkennung | Alle TAF |
| PTB02 | Kennung des Lieferanten (Rechnungsstellers) | | Eindeutiger Name oder Code, dessen Bedeutung bekannt gemacht worden ist | Alle TAF |
| PTB03 | Abrechnungszeitraum | tt.mm.(jj)jj tt.mm.(jj)jj | | Alle TAF |
| PTB04 | Gateway-Kennzeichnung | | | Alle TAF |
| PTB05 | Zählerkennzeichnung | | Alle dem Letztverbraucher zugeordneten Zähler. Es können Summen und Differenzen von Ständen verschiedener Zähler gebildet werden. Es muss erkennbar sein, welche Zähler mit welcher Operation verknüpft werden. | Alle TAF |
| PTB06 | Originäre Messwertliste, Zählerstandsgang + OBIS-Kennzahl | kWh, MWh, m ³ | Zählerstände der Registrierperioden des abgeschlossenen Abrechnungszeitraums. Der Letztverbraucher sollte auf dem Kundendisplay eine Hilfe vorfinden die eine Legende der verwendeten OBIS Kennzahlen enthält | Alle TAF |
| PTB07 | Länge der Registrierperiode | h,min,s | Bezogen auf die originäre Messwertliste. | Alle TAF |
| PTB08 | Aktivierte Tarifierungsfälle + Anfangs- und Endzeitpunkt | | Die im SMGW für den betreffenden Letztverbraucher aktivierten TAF müssen mit den vertraglich gebuchten übereinstimmen. Kennungen müssen im Versorgungsvertrag erscheinen. | Alle TAF |
| PTB09 | Tarifumschaltzeitpunkte | tt.mm.(jj)jj, hh:mm:ss | Liste der Tarifumschaltzeitpunkte des abgeschlos- | TAF2 |

| Anforderung | Größe und Art der Information | Einheit | Bemerkung | TAF |
|--------------|---|-----------------------------|---|----------|
| | | | senen Abrechnungszeit- raumes (Schaltprogramm) | |
| PTB10 | Register der Tarifstufen | kWh, MWh, m ³ | Endstände aller Tarifstu- fenregister des abge- schlossenen Abrechnungs- zeitraums | TAF2 |
| PTB11 | Datum | tt.mm.(jj)jj | Jeder Tag | TAF6 |
| PTB12 | Zählerstand am Ende des jeweiligen Tages | kWh, MWh, m ³ | | TAF6 |
| PTB13 | Fehlerstatus/Fehlernummer | | Information, ob angezeigte Messwert wegen eines Fehlers ungültig sind. | Alle TAF |
| PTB14 | Innere Signatur | | Messwerte müssen durch die vom SMGW gebildete innere Signatur in Kombi- nation mit dem öffentli- chen Schlüssel des SMGWs auf Unverändertheit über- prüfbar sein. | Alle TAF |

Zusätzlich definiert die PTB noch zwei weitere Anforderungen an die Transparenzsoftware, die in dieser Anwendungsregel berücksichtigt werden:

- *Anforderung TS1.3:* das Regelwerk muss einen eindeutigen Identifikator besitzen, damit der Letztverbraucher die Übereinstimmung der angewandten Regeln mit dem Vertrag überprüfen kann. Der Identifikator muss von der Transparenzsoftware angezeigt werden können.
- *Anforderung TS8.5:* Die Transparenzsoftware muss die originäre Messwertlisten des Abrechnungszeitraumes laden und deren Authentizität und Integrität prüfen können. Für diesen Prüfschritt ist es erforderlich, dass die Transparenzsoftware das kryptographische Zertifikat des SMGW, das die Messwerte erfasst und signiert hat, laden und auf Authentizität und Integrität überprüfen kann.

Aus der technischen Regel des Bundesamtes für Sicherheit in der Informationstechnik stammen Anforderungen an die Tarifregelwerke und die Daten mit welchen diese übertragen werden können. Zum Teil decken sich diese Daten mit den eichtechnischen Anforderungen der PTB-Anforderungen. Der Vollständigkeit halber sind alle Anforderungen an Regelwerke hier noch einmal in Tabelle 2 aufgelistet.

Tabelle 2: Anforderungen aus der Technischen Richtlinie des BSI (TAF 1,2,6,7)

| ID | Anforderung | Herkunft der Anforderung |
|-------------|-------------------------|---|
| RW01 | Geräte-ID | TR03109-1 Tabelle 20 (TAF1) Tabelle 21 (TAF2) Tabelle 27 (TAF6) Tabelle 28 (TAF7) |
| RW02 | Beziehungen von Geräten | Abgeleitet aus A5b |

| | | |
|--------------|--|---|
| RW03 | OBIS-Kennzahl | TR03109-1 Tabelle 20 (TAF1) Tabelle 21 (TAF2) Tabelle 27 (TAF6) Tabelle 28 (TAF7) |
| RW04 | Zählpunktbezeichnung | TR03109-1 Tabelle 20 (TAF1) Tabelle 21 (TAF2) Tabelle 27 (TAF6) Tabelle 28 (TAF7) |
| RW05 | Abrechnungszeitraum | TR03109-1 Tabelle 20 (TAF1) Tabelle 21 (TAF2) Tabelle 28 (TAF7) |
| RW06 | Letztverbrauchererkennung | TR03109-1 Tabelle 20 (TAF1) Tabelle 21 (TAF2) Tabelle 27 (TAF6) Tabelle 28 (TAF7) |
| RW07 | Zugriffsberechtigungen | TR03109-1 Tabelle 20 (TAF1) Tabelle 21 (TAF2) Tabelle 28 (TAF7) |
| RW08 | Versandzeitpunkte | TR03109-1 Tabelle 20 (TAF1) Tabelle 21 (TAF2) Tabelle 28 (TAF7) |
| RW09 | Gültigkeitszeitraum | TR03109-1 Tabelle 20 (TAF1) Tabelle 21 (TAF2) Tabelle 27 (TAF7) |
| RW10 | Definition der Tarifstufen | TR03109-1 Tabelle 21 (TAF2) |
| RW10a | OBIS-Kennzahl für Tarifstufe | TR03109-1 Tabelle 21 (TAF2) |
| RW10b | Tarifstufe zum Startzeitpunkt | TR03109-1 Tabelle 21 (TAF2) |
| RW10c | Tarifumschaltzeitpunkte | TR03109-1 Tabelle 21 (TAF2) |
| RW11 | Registrierperiode | TR03109-1 Tabelle 28 (TAF7) |
| RW12 | Beginn des abrechnungstechnischen Kalendertags | TR03109-1 Tabelle 27 (TAF6) |

Eine Besonderheit stellt der Tarifierungsfall 7 dar: Er ermöglicht eine Tarifierung auf Seiten des Versorgers und ermöglicht so die Anwendung von nahezu beliebigen Regelwerken, insbesondere von

Kombinationen aller beschriebenen Tarifierungsfälle. Es wird daher davon ausgegangen, dass auch die Anforderungen an die in diesem Dokument eigentlich nicht betrachteten weiteren Tarifierungsfälle (TAF 3,4,5 und 8 bis 12) auch relevant werden. Daher sind diese im Folgenden noch einmal explizit aufgeführt (Tabelle 3), soweit sie in der Technischen Richtlinie des BSI benannt werden:

Tabelle 3: Weitere Anforderungen aus der technischen Richtlinie

| ID | Anforderung | Herkunft der Anforderung |
|--------------|---|--|
| RW101 | Schwellwert | TR03109-1 Tabelle 23 (TAF3) Tabelle 34 (TAF12) |
| RW102 | Ereignis | TR03109-1 Tabelle 26 (TAF5) Tabelle33 (TAF9) |
| RW103 | Liste eingetretener Ereignisse mit Zeitpunkten (Tarifumschaltungen) | Hergeleitet aus TR03109-1 Tabelle 26 (TAF5) |
| RW104 | Anzahl Minimalwerte n | TR03109-1 Tabelle29 (TAF8) |
| RW105 | Anzahl Maximalwerte m | TR03109-1 Tabelle29 (TAF8) |
| RW106 | Verfügbare Energiemenge | TR03109-1 Tabelle34 (TAF12) |
| RW107 | Toleranzzeitraum | TR03109-1 Tabelle34 (TAF12) |
| RW108 | Startzeitpunkt | TR03109-1 Tabelle34 (TAF12) |
| RW109 | Geräte-IDs der Unterbrecher | TR03109-1 Tabelle34 (TAF12) |

Zusätzlich soll jeder Tarifierungsfall mit einem Namen und einer eindeutigen ID identifiziert werden können.

5.2 Aufbau der Modellierung

Wie bereits in den obigen Kapiteln angedeutet, steht das GreenButton-Datenformat im Mittelpunkt der Modellierung. Dieses wurde entwickelt, um einheitlich Messdaten von Smart Metern als Intervalldaten an Letztverbraucher zu übertragen. Dieses Datenformat weitestgehend genutzt werden, um Verbrauchsdaten an das Display für den Alltag zu übertragen, für welches keine eichtechnischen

Anforderungen vorliegen. Für die Bereitstellung von Daten für die Rechnungsprüfung wurde das GreenButton-Datenformat erheblich erweitert, um die in Kapitel 5.1 genannten Anforderungen abdecken zu können.

Um eine Strukturierung des Datenformats vornehmen zu können, wurde eine logische Dreiteilung der Daten vorgenommen:

- *Basisdaten*: Dienen der Übertragung von Messwerten. Diese können weitestgehend durch das GreenButton-Datenmodell abgebildet werden. Zusätzlich werden hier die Kennungen zur Zuordnung der Daten mit versendet.
- *Prüfungsdaten*: Beinhalten die Auswertepprofile und weitere eichrechtlich relevante Informationen zur Nachvollziehbarkeit einer Rechnung. Auswertepprofile parametrisieren die Regeln, welche auf den Messdaten angewandt werden um zu Werten für die Abrechnung zu kommen. Auswertepprofile sind individuell für einen Letztverbraucher und insbesondere für TAF7 relevant. Zusätzlich werden protokollierte Log-Daten aus dem Smart Meter Gateway für die Rechnungsprüfung benötigt
- *Regelwerke*: Allgemeine Beschreibung von möglichen Berechnungen auf Basis von Messdaten. Regelwerke gelten allgemein für eine Gruppe von Letztverbrauchern. Durch Auswertepprofile werden diese parametrisiert und individualisiert werden. Die Regelwerke selbst werden nicht über die Schnittstelle übertragen, sondern nur eine URI zu deren Bezugsort.

Datenelemente aus GreenButton werden in diesem Dokument aufgeführt und kurz erläutert. Weiterführende Dokumentation zu den Datenelementen ist in REQ.21 ESPI zu finden. Übernommene Klassen aus dem GreenButton-Standard sind zur Kenntlichmachung farblich grün hinterlegt.

Alle Erweiterungen des GreenButton-Datenmodells, welche zum Zwecke der Rechnungsprüfung benötigt werden, sind in den Grafiken mit `<<german_invoice_validation>>` gekennzeichnet.

5.3 Nutzung der Schnittstelle IF_3D_CON für Zwecke der Messdatenübertragung ohne eichrechtliche Relevanz

Das in diesem Dokument beschriebene Datenformat kann auch ohne zur Übertragung von Messzeitreihen ohne eichrechtliche Relevanz genutzt werden (z.B. für Zwecke der Energy Awareness). In diesem Fall müssen nicht alle Daten wie Einträge in Log-Files oder innere Signaturen übertragen. In diesem Falle empfiehlt es sich bei der Datenübertragung nur die Basisdaten zur Übertragung von abgeleiteten Messwertlisten zu nutzen.

5.4 Übersicht über das Datenmodell

5.4.1 Allgemeine Festlegungen

Das Datenmodell ist in 3 Komponenten aufgeteilt: Die Basisdaten, die Prüfungsdaten und benötigte Daten für die Regelwerke. Die Basisdaten basieren weitestgehend auf dem GreenButton-Datenmodell und erlauben auch die Kommunikation der Messdaten für das „Alltags-Display“ (vgl. Abbildung 1) sowie Messwertlisten aus der Tarifierung in den Smart Metern. Jede Nachricht **muss** mindestens die in Kapitel 6.1 als Erforderlich gekennzeichneten Daten enthalten.

Die Basisdaten ermöglichen das Versenden von Messwertlisten. Dies können sowohl originäre Messwertlisten als auch abgeleitete Messwertlisten sein. Die einzelnen Messwertlisten können über die jeweilige OBIS-Kennziffer eindeutig zugeordnet werden. Das Format der Basisdaten enthält allerdings eichrechtlich relevante Informationen für die Rechnungsüberprüfung nicht vollständig, insbesondere wenn der Tarifierungsfall 7 (dezentrale Tarifierung) genutzt wird. Diese werden erst mit den Daten für die Prüfungsdaten vervollständigt. Eine Nachricht **muss** die Prüfungsdaten enthalten, sofern die Nachricht für eine eichrechtlich konforme Rechnungsprüfung genutzt werden soll. Für alle anderen Zwecke sind die Prüfungsdaten **optional** und können die Basisdaten mit weiteren Informationen zur Tarifierung anreichern.

Die benötigten Daten für das Regelwerk bestehen im Rahmen der Schnittstelle IF_3D_CON im Wesentlichen aus einem Datenfeld für eine URI, welche den genauen Ort für das zu ladende Regelwerk spezifiziert. Bei Nutzung des Tarifierungsfalls 7 wird das Mitsenden der Prüfungsdaten und der Daten für das Regelwerk empfohlen, um die Tarifierungsinformationen nachvollziehbar darstellen zu können.

Regelungen des Green Button – Datenmodells wie zum Beispiel die Festlegung von Referenzierungen über das Atom Syndication Format wurden soweit es für Kompatibilitätszwecke notwendig ist, übernommen. Referenzierungen die im Datenmodell über das Atom Syndication Format erfolgen, sind in den Abbildungen der folgenden Kapitel mit <<link>> gekennzeichnet.

5.4.2 Basisdaten

Die Basisdaten werden als Grundlage der Datenübertragung vom Display-Datendienst zum Letztverbraucher genutzt. Das Datenmodell basiert dabei weitestgehend auf dem GreenButton-Datenmodell.

Die Basisdaten beschreiben ein allgemeines Datenformat für die Übertragung von Messdaten, die in Intervallen vorliegen („Messwertlisten“). Sie ermöglichen auch die Zuordnung der Messwertlisten zu einem Letztverbraucher und einem Lieferanten. Die Messdaten sind einem Zählpunkt (*UsagePoint*) zugeordnet. Zusammen mit der Zählernummer (*meterId*) und dem OBIS-Code (*obisCode*) in der Klasse *ReadingType* lassen sich die Messdaten immer eindeutig zuordnen. Jeder einzelne Messwert wird für die Anwendung in Deutschland signiert und entsprechend übertragen. Über die Klasse *ReadingQuality* lassen sich Messwerte als „originär“, „abgeleitet“ oder „Ersatzwerte“ kennzeichnen. Die Zuordnung von Messwertlisten zu einem Zählpunkt erfolgt über die Zählpunktbezeichnung und die Zählernummer und, sofern sich der OBIS-Code bei der Ableitung nicht ändert, über den OBIS-Code.

In der Klasse *ReadingType* können Details der entsprechenden Messung weiter spezifiziert werden. Für die Details der einzelnen Datenelemente wird hier auf die GreenButton-Spezifikation (ESPI, REQ.21) verwiesen.

Eine Anforderung des BSI wird mit der Erweiterung des Datentyps *IntervalReading* erreicht: die Signatur der einzelnen Messwerte. Dazu wird der Datenelement *signature* eingeführt, der die interne Signatur eines Messwertes abbildet und anhand derer die Integrität des entsprechenden Messwertes geprüft werden kann.

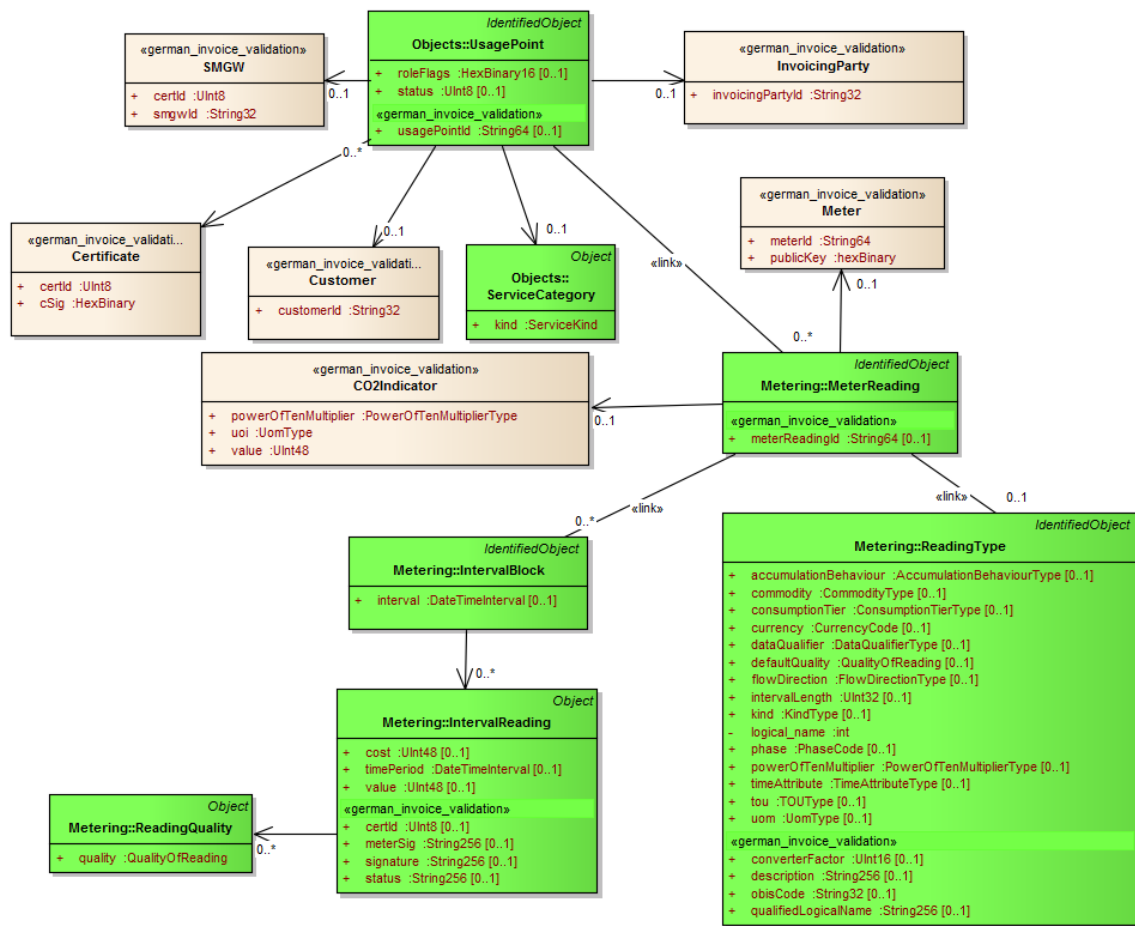


Abbildung 2: Übersicht über die Basisdaten im Datenmodell

5.4.2.1 Von den Basisdaten abgedeckte Anforderungen

Die folgende Abbildung des Datenmodells auf die Anforderungen bezieht sich auf die Tabellen 1 – 3 aus Kapitel 6.1.

5.4.2.1.1 Anforderung PTB01: Kennung des Letztverbrauchers

Die Kennung des Letztverbrauchers wird über die Klasse *Customer* und das beinhaltende Datenelement *customerId* abgebildet. *customerId* muss eine beliebige je Liefervertragspartner eindeutige Zeichenfolge zur Identifizierung enthalten („Kundennummer“). Die Kennung sollte dem Letztverbraucher bekannt sein.

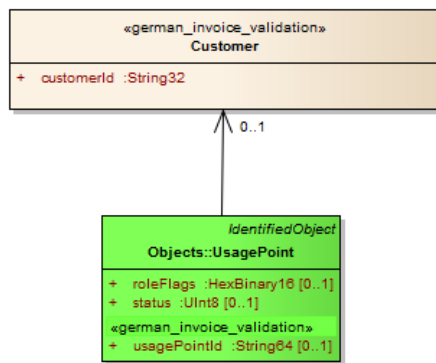


Abbildung 3: Die Klasse Customer zur Abbildung der Letztverbrauchererkennung

5.4.2.1.2 Anforderung PTB02: Kennung des Lieferanten (Rechnungssteller)

Die Kennung des Rechnungsstellers wird über die Klasse *InvoicingParty* und das beinhaltende Datenelement *invoicingPartyId* abgebildet. *invoicingPartyId* muss eine eindeutige Kennung des Rechnungsstellers enthalten. Diese Kennung sollte dem Letztverbraucher bekannt gemacht worden sein, damit dieser diese zuordnen kann.

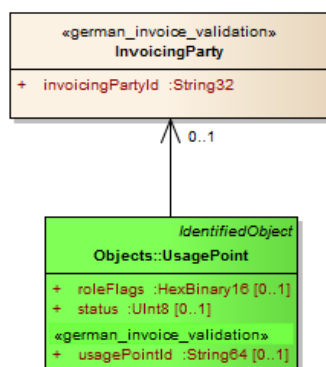


Abbildung 4: Die Klasse InvoicingParty zur Abbildung des Rechnungsstellers

5.4.2.1.3 Anforderung PTB04: Gateway-Kennzeichnung

Die Gateway-Kennzeichnung wird über die Klasse *SMGW* abgebildet. Das Datenelement *smgwId* enthält die eindeutige Kennzeichnung des Smart Meter Gateways als Zeichenfolge. Das Zertifikat des SMGW soll über die Klasse *Certificate* kommuniziert werden, um eine Überprüfung der Signatur des Smart Gateways zu ermöglichen.

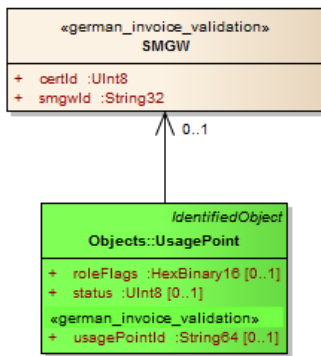


Abbildung 5: Die Klasse SMGW im Datenmodell zur Abbildung der Smart Meter Gateway Informationen

5.4.2.1.4 Anforderung PTB05: Zählerkennzeichnung

Die Zählerkennzeichnung wird über das Datenelement *meterId* in der Klasse *Meter* dargestellt.

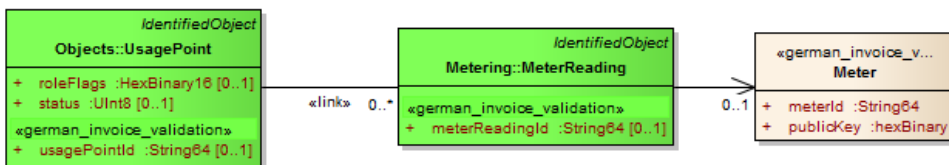


Abbildung 6: Die Klasse Zähler im Datenmodell zur Abbildung der Zählerkennung

5.4.2.1.5 Anforderung PTB06: Originäre Messwertliste, Zählerstandsgang + OBIS-Kennzahl

Die originäre Messwertliste und Zählerstandsgänge können über die Klasse *Metering* und die beinhaltenden Klassen *MeterReading*, *ReadingType*, *IntervalBlock*, *IntervalReading* und *ReadingQuality* abgebildet werden. Bei diesen Klassen handelt es sich um Klassen aus ESPI REQ.21 (GreenButton). Der OBIS-Code wird über das Feld *obisCode* in der Klasse *ReadingType* abgebildet. Hierbei handelt es sich um eine Erweiterung der GreenButton-Klasse zur Abbildung von deutschen Anforderungen.

Originäre Messewertlisten im Sinne dieser Anwendungsregel sind die vom Smart Meter Gateway für den Versand bereitgestellten Messwertlisten. Dies können auch im Smart Meter Gateway eichrechtlich korrekt gebildete abgeleitete Messwertlisten sein.

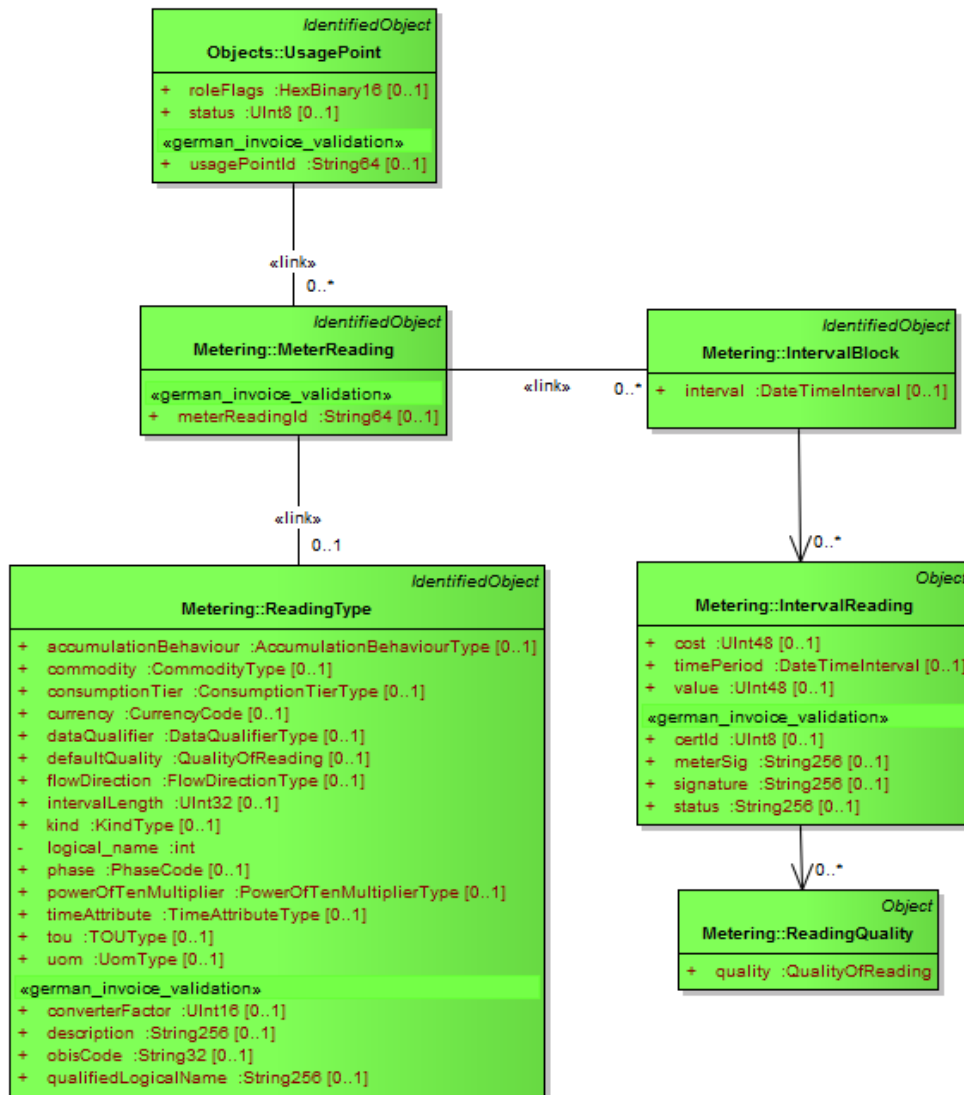


Abbildung 7: Informationen und Daten zu Messwertlisten im Datenmodell

5.4.2.1.6 Anforderung PTB07: Länge der Registrierperiode

Die Länge der Registrierperiode wird über das Feld *intervalLength* der Klasse *ReadingType* abgebildet.

5.4.2.1.7 Anforderung PTB10: Register der Tarifstufen

Register der verschiedenen Tarifstufen werden als abgeleitete Messwertlisten abgebildet. Verschiedene Messwertlisten können über den OBIS-Code und dessen Beschreibung in der Klasse *ReadingType* unterschieden werden. Zusätzlich kann optional das Feld *TOUType* der Klasse „*ReadingType*“ genutzt werden, sofern eine zeitbasierte Aufteilung zugrunde liegt.

5.4.2.1.8 Anforderung PTB11: Datum

Das Datum einer Ablesung für TAF06 geht aus der Klasse *IntervalBlock* und dem darin enthaltenen Datenelement *interval* hervor. Zusätzlich wird der Abruf der Messdaten in den Logdaten mitgeteilt.

5.4.2.1.9 Anforderung PTB12: Zählerstand am Ende des jeweiligen Tages

Der Zählerstand am Ende des Ablesetages geht aus der originären Messwertliste hervor (siehe PTB06).

5.4.2.1.10 Anforderung PTB13: Fehlerstatus/Fehlernummer

Der Status eines Messwertes wird über die Klasse *ReadingQuality* abgebildet. Entsprechende Statuswerte zur Fehlerindikation sind vorgesehen.

5.4.2.1.11 Anforderung PTB14: Innere Signatur

Die innere Signatur eines Messwertes wird über das Datenelement *signature* in der Klasse *IntervalReading* dargestellt. Weitere Informationen zur Überprüfung der inneren Signatur sind in Kapitel 7.1 aufgeführt.

5.4.2.1.12 Anforderung RW01: Geräte-ID

Die Geräte-ID wird im Sinne dieser Anwendungsregel als Zähler-ID verstanden und kann analog Anforderung PTB05 über das Feld „*meterId*“ dargestellt.

5.4.2.1.13 Anforderung RW03: OBIS-Kennzahl

Die OBIS-Kennzahl wird analog PTB06 über das Feld *obisCode* in der Klasse *ReadingType* abgebildet. Zusätzlich kann in dem Feld *description* eine freie Beschreibung zu der OBIS-Kennziffer hinterlegt werden.

5.4.2.1.14 Anforderung RW04: Zählpunktbezeichnung

Die Zählpunktbezeichnung wird über das Feld *usagepointId* in der Klasse *UsagePoint* abgebildet

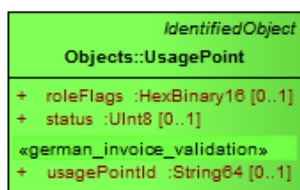


Abbildung 8: Der Zählpunkt im Datenmodell

5.4.2.1.15 Anforderung RW06: Letztverbrauchererkennung

Die Letztverbrauchererkennung wird analog der Anforderung PTB01 über die Klasse *Customer* abgebildet. Hier wird das Datenelement *customerId* genutzt.

5.4.2.1.16 Anforderung RW07: Zugriffsberechtigungen

Zugriffsberechtigungen werden durch das Datenmodell dieser Anwendungsregel nicht übertragen. Eine entsprechende Überprüfung der Zugriffsberechtigung auf die Messdaten ist im Smart Meter-Gateway zu realisieren.

5.4.2.1.17 Anforderung RW11: Registrierperiode

Die Registrierperiode wird analog PTB07 über das Feld *intervalLength* der Klasse *ReadingType* abgebildet

5.4.2.1.18 Anforderung RW12: Beginn des abrechnungstechnischen Kalendertags

Der Beginn des abrechnungstechnischen Kalendertags wird über die Intervall-Blöcke (Klasse *IntervalBlock*) dargestellt und nicht gesondert übertragen.

5.4.3 Prüfungsdaten

Zur vollständigen Abbildung von Auswerteprofilen, die auf ein bestimmtes Regelwerk angewendet werden können, wird eine umfangreiche Erweiterung der Basisdaten benötigt. Der zentrale Verbindungspunkt zwischen den Basisdaten und den Daten für das Auswerteprofil ist die Klasse „Analy-

sisProfile“. Zusätzlich sind hier die protokollierten Log-Daten des SMGW beinhaltet, die ebenfalls für eine Rechnungsprüfung benötigt werden.

Diese Erweiterung wird im Folgenden zuerst anhand der Anforderungen beschrieben.

5.4.3.1 Von den Prüfungsdatendaten abgedeckte Anforderungen

5.4.3.1.1 Anforderung PTB03: Abrechnungszeitraum

Der Abrechnungszeitraum wird das Datenelement *billingPeriod* in der Klasse *AnalysisProfile* abgedeckt.

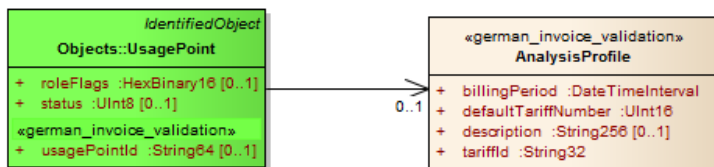


Abbildung 9: Die Klasse *AnalysisProfile* im Datenmodell als Einstiegspunkt zu den Prüfungsdaten im Datenmodell

5.4.3.1.2 Anforderung PTB08: Aktivierte Tarifierungsfälle + Anfangs- und Endzeitpunkt

Die für den Rechnungszeitraum aktivierten Tarifierungsfälle werden über die Klasse *TariffStage* abgebildet. Das Datenelemente *tariffUseCase* beschreibt mit einer Integer-Zahl den zugrunde liegenden Tarifierungsfall so wie diese im Smart Meter Umfeld beschrieben worden sind. Dem Datenelement liegt eine entsprechende Aufzählung der Tarifierungsfälle zugrunde. Mit der *tariffStage* wird die Tarifstufe wird auch über eine Integerzahl definiert, wobei diese frei wählbar ist und dem Letztverbraucher eine Zuordnung ermöglichen sollte.

Jede *TariffStage* ist mit einer Klasse *DateTimePeriod* verbunden. Hier muss eines der Datenelemente *datetimeInterval* oder *datetimeVar* gefüllt werden. Während das erstgenannte Datenelement einen abgeschlossen Zeitraum beschreibt, kann mit dem zweiten Datenelement ein Zeitpunkt definiert werden. Ist nur der Zeitpunkt definiert, so gilt dieser als Aktivierungszeitpunkt des Tarifs. Dieser gilt dann erstmal auf unbestimmte Dauer.

5.4.3.1.3 Anforderung PTB09: Tarifumschaltzeitpunkte

Die Tarifumschaltzeitpunkte beschreiben Zeitpunkte des Wechsels in eine andere Tarifstufe. Daher ist der Klasse *TariffStage* die Klasse *TariffChangeTrigger* zugeordnet, mit der sich die Wechselzeitpunkte und die Gründe für einen Wechsel beschreiben lassen. Dabei sind drei Arten von Tarifstufenwechseln vorgesehen: Tarifwechsel aufgrund der Überschreitung eines bestimmten Wertes (*ThresholdTrigger*), Tarifwechsel aufgrund eines aufgetretenen Events (*ExternalEventTrigger*) oder ein Tarifwechsel aufgrund eines bestimmten erreichten Zeitpunkts (*TimeTrigger*). *TimeTrigger* ermöglicht die Abbildung von Regeln für die Berechnung von regelmäßigen Umschaltzeitpunkten.

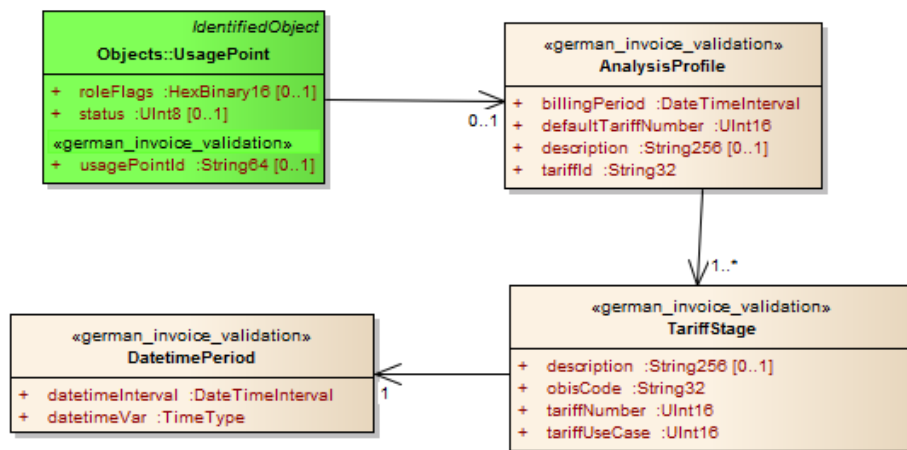


Abbildung 10: Definition von verschiedenen Tarifstufen im Datenmodell

5.4.3.1.4 Anforderung PTB11: Datum

Das Datum einer Ablesung die in einem Bedarfsfall (Tarifwechsel, Lieferantenwechsel, etc.) durchgeführt wird nicht im speziellen abgebildet. Eine Ablesung die zu einem Tarif-, Lieferanten- oder sonstigem Wechsel führen, werden über ein verändertes Datum *billingPeriod* in der Klasse *AnalysisProfile* abgebildet.

5.4.3.1.5 Anforderung RW02: Beziehungen von Geräten

Beziehungen von Geräten lassen sich als mathematische Operationen abbilden. Diese sind verbunden mit dem entsprechenden Zählpunkten. Diese können addiert, subtrahiert, multipliziert oder dividiert werden. Eine entsprechende Klasse „MathOperation“ besteht aus zwei Operanden (Klasse „Operand1“ und Klasse „Operand2“), die auf entsprechende Messwertlisten verweisen und einem Operator (Klasse „Operator“), der die durchzuführende mathematische Operation beschreibt.

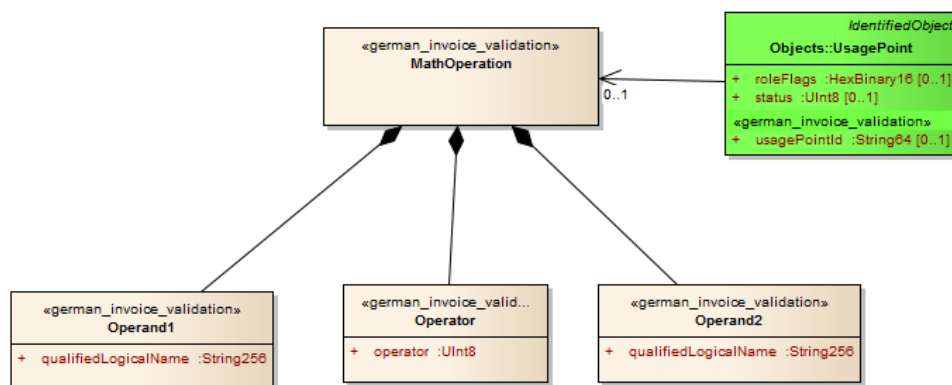


Abbildung 11: Gerätebeziehungen als Operationen im Datenmodell

5.4.3.1.6 Anforderung RW05: Abrechnungszeitraum

Der Abrechnungszeitraum wird über das Datenelement *billingPeriod* der Klasse *AnalysisProfile* abgebildet.

5.4.3.1.7 Anforderung RW07: Zugriffsberechtigungen

Zugriffsberechtigungen werden durch das Datenmodell dieser Anwendungsregel nicht übertragen. Eine entsprechende Überprüfung der Zugriffsberechtigung auf die Messdaten ist im Smart Meter-Gateway zu realisieren.

5.4.3.1.8 Anforderung RW08 Versandzeitpunkte

Erfolgte Versandzeitpunkte können werden mit den Log-Einträgen im Smart Meter Gateway protokolliert und können über diese vom Letztverbraucher nachvollzogen werden. Entsprechende Logeinträge werden über die Klasse *LogEntry* abgebildet.

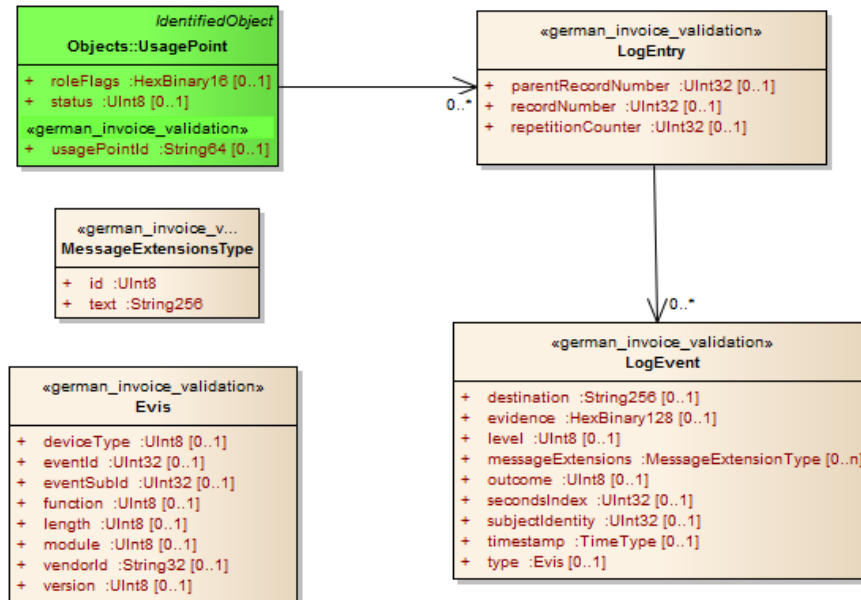


Abbildung 12: Darstellung von Logeinträgen im Datenmodell

5.4.3.1.9 Anforderung RW09: Gültigkeitszeitraum

Der Gültigkeitszeitraum eines Regelwerks wird im Regelwerk selber codiert. Eine Übertragung der Gültigkeitszeiträume für Regelwerke erfolgt über das Auswerteprofil nicht erneut. Start und Endzeitpunkte von Zuordnung von Tarifen und Tarifstufen zu einzelnen Messwertlisten werden analog der Anforderungen PTB08 und PTB09 abgebildet.

5.4.3.1.10 Anforderung RW10: Definition der Tarifstufen

Die Tarifstufen werden analog Anforderung PTB08 über die Klasse *TariffStage* abgebildet.

5.4.3.1.11 Anforderung RW10a: OBIS-Kennzahl für Tarifstufe

Die OBIS-Kennzahlen werden über die Basisdaten abgebildet. Das Wertefeld für die Tarifstufe, so wie es in der OBIS-Kennzahlen Systematik beschrieben ist, sollte dem Datenelement *tariffNumber* in der Klasse *tariffStage* entsprechen.

5.4.3.1.12 Anforderung RW10b: Tarifstufe zum Startzeitpunkt

Gültigkeitszeiträume für Tarifstufen werden analog TAF08 und TAF09 über die Klassen *tariffStage*, „DateTimePeriod“ und den entsprechenden Triggern für die Tarifumschaltung beschrieben. In der Klasse *AnalysisProfile* kann das Datenelement *defaultTariffNumber* genutzt werden um die Tarifstufe zum Startzeitpunkt als Standardtarifstufe zu definieren.

5.4.3.1.13 Anforderung RW10c: Tarifumschaltzeitpunkte

Tarifumschaltzeitpunkte werden analog Anforderung PTB09 beschrieben.

5.4.3.1.14 Anforderung RW101: Schwellwert

Schwellwerte als Anlass für einen Tarifstufenwechsel werden als *ThresholdTrigger* abgebildet. Ein entsprechender Trigger wird über die Klasse *TariffThreshold* detailliert über die Datenelemente *overLimit*, *underLimit*, *direction* und *tariffNumber* definiert.

Aus dem FNN COSEM-Dokument [FNN COSEM], an den sich die Darstellung des Schwellwertes hier anlehnt:

Enthält die Schaltstufen, wobei tariff_number = Tarifstufe. Entspricht einer der Tarifstufen aus der Anzahl der Tarifstufen, die mit den Grundparametern festgelegt werden (n). Die Schaltstufe muss einem Leistungswert entsprechen. Bei Überschreitung von ‚over_limit‘ wird in die referenzierte Tarifstufe geschaltet. Bei Unterschreitung von ‚under_limit‘ wird in diejenige Tarifstufe geschaltet, die durch den neuen Messwert festgelegt wird. Jeder Eintrag in einem ‚tariff_threshold‘ beschreibt den Übergang in die nächste Tarifstufe. Die Beschreibung bezieht sich nur auf die untere Grenze der Tarifstufe. Zur Abbildung einer Hysterese werden ‚over_limit‘ und ‚under_limit‘ herangezogen.

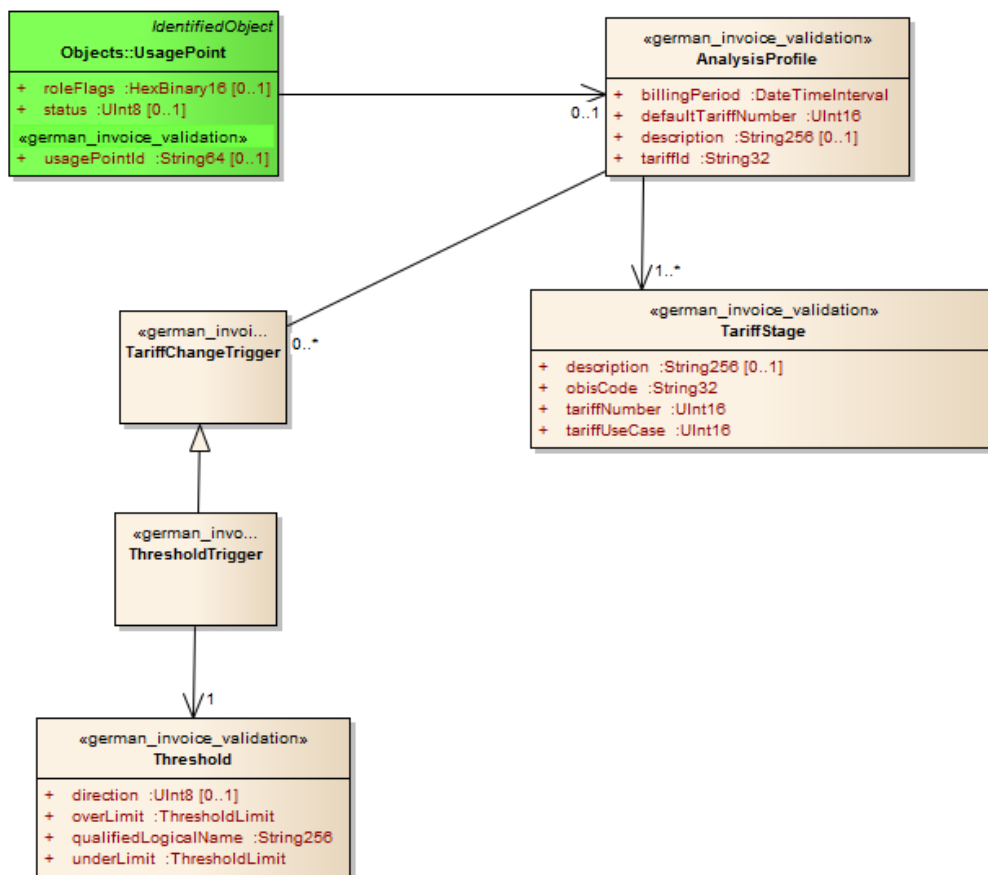


Abbildung 13: Darstellung von Schwellwerten im Datenmodell

5.4.3.1.15 Anforderung RW102: Ereignis

Ereignisse werden über die Klasse *EventTrigger* dargestellt. Eine detaillierte Beschreibung eines Ereignisses erfolgt über die Klasse *ExternalEvent*, die über die Datenelemente *eventID* und *eventPeriod* verfügt. Ein Ereignis kann sowohl eine zeitliche als auch eine mengenmäßige Begrenzung beinhalten.

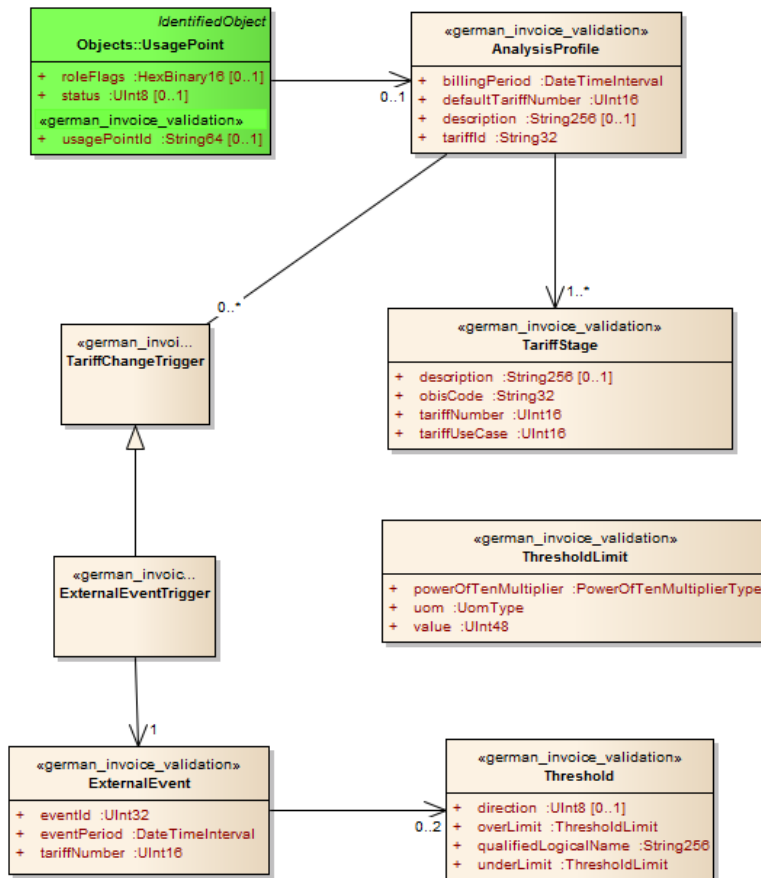


Abbildung 14: Darstellung von Ereignissen im Datenmodell

5.4.3.1.16 Anforderung RW103: Liste eingetretener Ereignisse mit Zeitpunkten

Siehe Anforderung RW102. Zusätzlich werden eingetretene externe Ereignisse im Log protokolliert und über die Logeinträge mitkommuniziert

5.4.3.1.17 Anforderung RW104: Anzahl Minimalwerte n

Die Anzahl der Minimalwerte wird über die Klasse *MinMaxCounter* dargestellt. Das Datenelement *count* gibt die Anzahl der Minimalwerte an. Zur Kennzeichnung von Minimalwerten muss das Datenelement *maxCounter* auf den Wert *false* gesetzt werden.

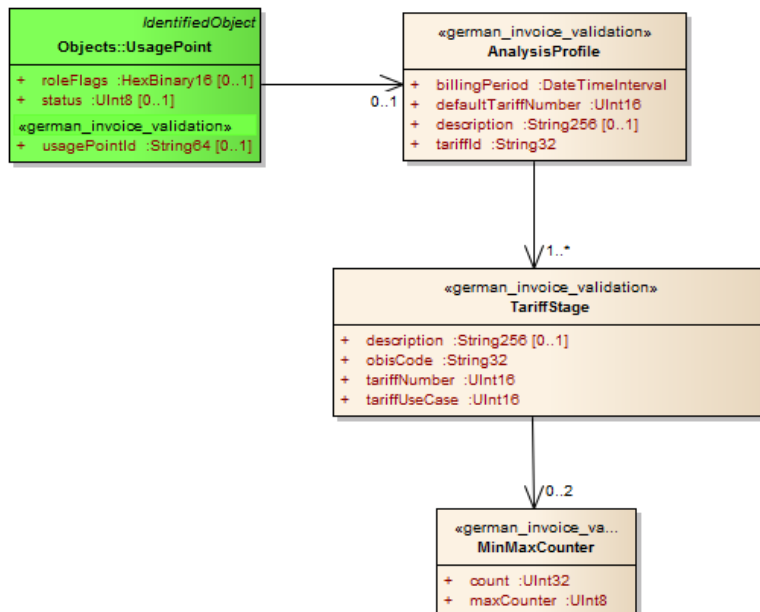


Abbildung 15: Darstellung von Minimal- und Maximalwerten im Datenmodell

5.4.3.1.18 Anforderung RW105: Anzahl Maximalwerte n

Die Anzahl der Maximalwerte wird über die Klasse *MinMaxCounter* dargestellt. Das Datenelement *count* gibt die Anzahl der Maximalwerte an. Zur Kennzeichnung von Minimalwerten muss das Datenelement *maxCounter* auf den Wert „true“ gesetzt werden.

5.4.3.1.19 Anforderung RW106: Verfügbare Energiemenge

In dieser Anwendungsregel wird davon ausgegangen, dass über das Auswerteprofil die ursprünglich eingekaufte bzw. bezahlte Energiemenge kommuniziert wird. Die ausgehend von diesem Wert nach der Messung noch verfügbare Energiemenge kann über die verbrauchte Menge errechnet werden. Die ursprünglich verfügbare Energiemenge wird über die Klasse *Prepaid* und über die Datenelemente *uom* für die Maßeinheit und *powerOfTenMultiplier* für die Multiplizität der Maßeinheit dargestellt (vgl. Abbildung 16).

5.4.3.1.20 Anforderung RW107: Toleranzzeitraum

Der Toleranzzeitraum wird über das Datenelement *toleranceInterval* in der Klasse *Prepaid* beschrieben. Es handelt sich um einen Intervall-Datentyp.

5.4.3.1.21 Anforderung RW108: Startzeitpunkt

Der Startzeitpunkt wird über die Klasse *DatetimePeriod* dargestellt, auf die von der Klasse *Prepaid* verwiesen wird.

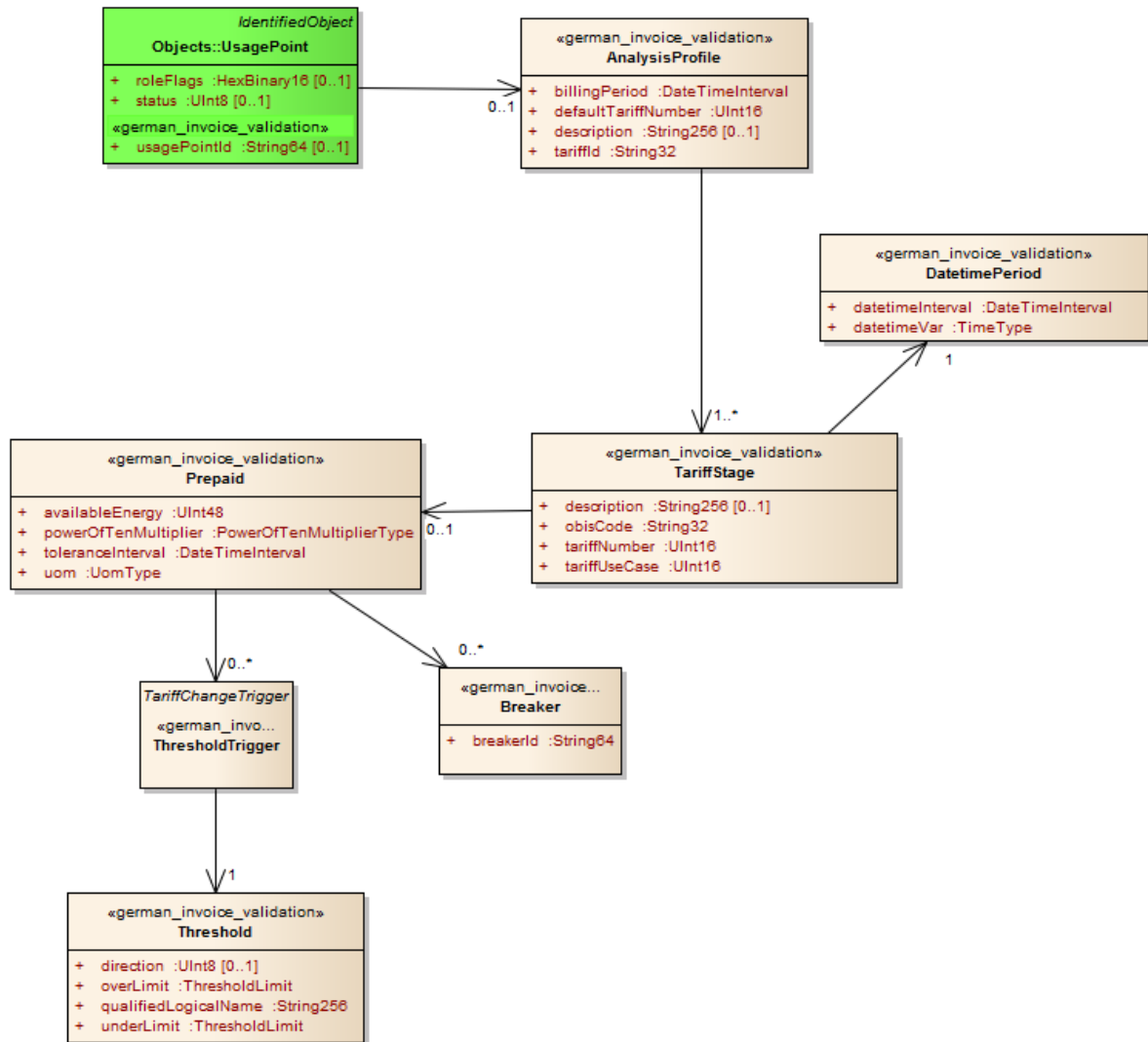


Abbildung 16: Datenelemente zur Abbildung von Vorab-bezahlten Tarifinformationen (Prepaid)

5.4.3.1.22 Anforderung RW109: Geräte-IDs der Unterbrecher

Die Geräte-IDs der Unterbrecher werden über die Klasse *Breaker* mit dem Datenelement *breakerId* als eindeutige Identifikation beschrieben.

5.4.4 Beschreibung zusätzlicher Klassen

5.4.4.1 AnalysisProfile

Die Klasse *AnalysisProfile* dient der Identifizierung des Auswerteprofiles und stellt die Verbindung des Auswerteprofiles zu den Basisdaten her. Die Klasse *AnalysisProfile* enthält die Datenelemente *billing-Period* und *tariffId*. Das Element *tariffId* beschreibt eine je Liefervertragspartner eindeutige Identifizierung für den Tariftyp, dem das Auswerteprofil zugrunde liegt.

5.4.4.2 RuleSystem

Die Klasse *RuleSystem* enthält den Verweis auf das zugrundeliegende Regelwerk. Dieses Regelwerk wird über eine URI identifiziert. Die URI beschreibt die Adresse, über die das Regelwerk vom Display-DatenDienst bezogen werden kann. Anforderung TS 1.3 [PTB-A50.8] fordert außerdem eindeutige eine Identifizierung des Regelsystems, welches über das Datenelement *id* erfolgen kann.

5.4.4.3 *Certificate*

Mit der Klasse *Certificate* wird das Inhaltsdaten-Zertifikat des SMGW übertragen. Hiermit kann Anforderung TS8.5 der PTB abgedeckt werden.

6 Regeln zur Anwendung des Datenmodells

Im Folgenden sind die Regeln zur Anwendung des Datenmodells beschrieben. Diese werden ausgehend von der zentralen Klasse *UsagePoint* aufgeteilt in Basisdaten und Prüfungsdaten.

Zur Aufrechterhaltung der (Abwärts-) Komptabilität zwischen dem ESPI REQ.21 (GreenButton) und dem Datenmodell dieser Anwendungsregel, sind in dem Datenmodell einige Elemente als optional gekennzeichnet, die im Folgenden als „Erforderlich“ beschrieben werden. Führend sind hier die Angaben in den nachfolgenden Tabellen.

Regelungen des Green Button – Datenmodells, wie zum Beispiel der Festlegung von Referenzierungen über den Atom Syndication Format oder Inhaltsvorgaben für einzelne Datenelemente, wurden soweit es für Kompatibilitätszwecke notwendig ist, übernommen. Dies gilt auch für Zeitangaben, die soweit nicht anders definiert, in UTC-Zeit erfolgen.

6.1 Basisdaten

Datenelemente, deren Verwendung mit **Erforderlich*** beschrieben ist, sind erforderlich, sofern die Daten für eine eichrechtlich relevante Überprüfung übermittelt werden (Abrechnungsdaten).

| Nr. | Beschreibung | Verwendung |
|------|--|---------------------|
| 1.01 | UsagePoint Die Klasse <i>UsagePoint</i> repräsentiert den Zählpunkt und stellt das zentrale Datenelement einer Nachricht dar. Jede Nachricht muss mindestens einen Zählpunkt beinhalten. Eine Instanz der Klasse <i>UsagePoint</i> : <ul style="list-style-type: none">• muss auf eine Instanz der Klasse <i>InvoicingParty</i> verweisen• muss auf eine Instanz der Klasse <i>Customer</i> verweisen• muss auf eine Instanz der Klasse <i>SMGW</i> verweisen• muss auf eine Instanz der Klasse <i>ServiceCategory</i> verweisen• muss auf mindestens eine Instanz der Klasse <i>MeterReading</i> verweisen• kann auf Instanzen der Klasse <i>LogEntry</i> verweisen• kann auf Instanzen der Klasse <i>MathOperation</i> verweisen• kann auf eine Instanz der Klasse <i>AnalysisProfile</i> verweisen | Erforderlich |
| 1.02 | UsagePoint – roleFlags Das Datenelement <i>roleFlags</i> spezifiziert die Rolle eines Zählpunkts. Die Werte des Datenelements sind nach ESPI REQ.21 spezifiziert. Das Datenelement <i>roleFlags</i> ist optional . | Optional |
| 1.03 | UsagePoint - status Das Datenelement <i>status</i> beschreibt den aktuellen Status des Zählpunktes. Nach ESPI REQ.21 kann der Status die Werte 0 (off) oder 1 (on) annehmen. Das Datenelement <i>status</i> ist optional . | Optional |
| 1.04 | UsagePoint - usagePointId Die <i>usagePointId</i> entspricht der Zählpunktbezeichnung nach dem aktuellen MeteringCode. Eine Instanz der Klasse <i>UsagePoint</i> muss genau eine Zählpunktbezeichnung enthalten. | Erforderlich |
| 1.05 | ServiceCategory Die Klasse <i>ServiceCategory</i> repräsentiert nach ESPI REQ.21 die Sparte des Produkts (Service), welches dem Letztverbraucher am Zählpunkt zur Ver- | Erforderlich |

| | | |
|------|---|---------------------|
| | <p>fügung gestellt wird.</p> <p>Die Klasse <i>ServiceCategory</i> enthält keine Verweise auf weitere Klassen.</p> | |
| 1.06 | <p>ServiceCategory - kind</p> <p><i>kind</i> beschreibt als Datenelement die konkrete Sparte des Zählpunktes. Gültige Werte nach ESPI REQ.21 sind:</p> <ul style="list-style-type: none"> 0 – electricity (Elektrizität) 1 – gas (Gas) 2 – water (Wasser) 4 – pressure (Druck) 5 – heat (Wärme) 6 – cold (Kälte) 7 – communication (Kommunikation) 8 – time (Zeit) <p>Eine Instanz der Klasse <i>ServiceCategory</i> muss genau einen der genannten Werte für das Datenelement <i>kind</i> beinhalten.</p> | Erforderlich |
| 1.07 | <p>MeterReading</p> <p>Die Klasse <i>MeterReading</i> repräsentiert den Kopf einer Messwertliste. Die Klasse enthält untergeordnet die entsprechenden Werte und Zusatzinformationen einer Messwertliste. Jede Nachricht muss mindestens eine Instanz der Klasse <i>MeterReading</i> enthalten.</p> <p>Eine Instanz der Klasse <i>MeterReading</i>:</p> <ul style="list-style-type: none"> • Muss auf eine Instanz der Klasse <i>ReadingType</i> verweisen • Muss auf mindestens eine Instanz der Klasse <i>IntervalBlock</i> verweisen • Kann auf eine Instanz der Klasse <i>CO2Indicator</i> verweisen | Erforderlich |
| 1.08 | <p>MeterReading – meterReadingId</p> <p>Die <i>meterReadingId</i> identifiziert eine Messwertliste eindeutig. Die ID kann zum Beispiel aus der Zählpunktbezeichnung, der Gerätenummer und der OBIS-Kennziffer zusammengesetzt werden.</p> <p>Eine Instanz der Klasse <i>MeterReading</i> muss ein Datenelement vom Typ <i>meterReadingId</i> enthalten.</p> | Erforderlich |
| 1.09 | <p>CO2Indicator</p> <p>Die Klasse <i>CO2Indicator</i> kann genutzt werden, um einer Messwertliste Informationen über damit verbundene CO₂-Werte zu übermitteln. Die Nutzung der Klasse <i>CO2Indicator</i> ist optional.</p> <p>Die Klasse <i>CO2Indicator</i> verweist auf keine weiteren Klassen.</p> | Optional |
| 1.10 | <p>CO2Indicator – powerOfTenMultiplier</p> <p>Das Datenelement <i>powerOfTenMultiplier</i> repräsentiert den Einheitenvorsatz des übermittelten CO₂-Wertes. Gültige Werte nach ESPI REQ.21 sind:</p> <ul style="list-style-type: none"> 0 = None 1 = deca=x10 2 = hecto=x100 -3 = mili=x10⁻³ 3 = kilo=x1000 6 = Mega=x10⁶ -6 = micro=x10⁻³ 9 = Giga=x10⁹ <p>Bei Instanziierung der (optionalen) Klasse <i>CO2Indicator</i> muss das Datenelement <i>powerOfTenMultiplier</i> mit einem entsprechenden Wert gefüllt werden.</p> | Erforderlich |
| 1.11 | <p>CO2Indicator – uoi</p> <p>Das Datenelement <i>uoi</i> repräsentiert die Maßeinheit des übermittelten CO₂-Wertes. Gültige Werte entsprechen ESPI REQ.21. Eigene Maßeinheiten können hier ergänzt werden.</p> | Erforderlich |

| | | |
|------|--|----------------------|
| | Bei Instanziierung der (optionalen) Klasse CO2Indicator muss das Datenelement <i>uoi</i> mit einem entsprechenden Wert gefüllt werden. | |
| 1.12 | CO2Indicator – value Das Datenelement <i>value</i> repräsentiert den eigentlichen CO ₂ -Wert, dessen Maßeinheit durch die weiteren Datenelemente der Klasse bestimmt wird. Bei Instanziierung der (optionalen) Klasse CO2Indicator muss das Datenelement <i>value</i> mit einem entsprechenden ganzzahligen Wert gefüllt werden. | Erforderlich |
| 1.13 | ReadingType Die Klasse <i>ReadingType</i> spezifiziert die Inhalte einer Messwertliste. Jede Messwertliste muss eine Instanz der Klasse <i>ReadingType</i> beinhalten. Die Klasse <i>readingType</i> verweist auf keine weiteren Klassen. | Erforderlich |
| 1.14 | ReadingType – accumulationBehaviour Das Datenelement <i>accumulationBehaviour</i> beschreibt, wie die Werte über die Zeit kumuliert werden. Die Liste der gültigen Werte ist nach ESPI REQ.21 im <i>AccumulationBehaviourType</i> definiert. Beispiele sind: 0 = nicht anwendbar 3 = kumuliert Die Nutzung des Datenelements <i>accumulationBehaviour</i> ist optional. Eine Verwendung sollte nur bei aufsummierten Werten in der Messwertliste stattfinden. | Optional |
| 1.15 | ReadingType – commodity Das Datenelement <i>commodity</i> beschreibt das gelieferte Produkt genauer. Gültige Werte nach ESP REQ.21 sind: 0 = Not Applicable (nicht anwendbar) 1 = Electricity metered value (Elektrizität) 4 = Air (Luft) 7 = NaturalGas (Erdgas) 8 = Propane (PropanGas) 9 = PotableWater (Trinkwasser) 10 = Steam (Dampf) 11 = WasteWater (Abwasser) 12 = HeatingFluid (Wärmflüssigkeit) 13 = CoolingFluid (Kühlflüssigkeit) Die Nutzung des Datenelements <i>commodity</i> ist optional . | Optional |
| 1.16 | ReadingType – consumptionTier Code für die Stufe des Verbrauchs, welcher mit der Messung verbunden wird Beispiele nach ESPI REQ.21 sind: 0 = Nicht anwendbar 1 = Block Tier 1 2 = Block Tier 2 Das Datenelement <i>consumptionTier</i> wird im Sinne dieser Anwendungsregel nicht genutzt. | Nicht genutzt |
| 1.17 | ReadingType – currency Das Datenelement <i>currency</i> codiert die Währung, welche mit den Kostenangaben der einzelnen Messwerte verbunden ist. Beispiele für gültige Werte nach ESPI REQ.21 sind: 0 - Not Applicable 840 - US Dollar 978 - Euro Die Nutzung des Datenelements <i>currency</i> ist optional . Bei Nutzung des Datenelements <i>cost</i> in der Klasse <i>IntervalReading</i> wird die Nutzung des Datenelements <i>currency</i> empfohlen | Optional |
| 1.18 | ReadingType – dataQualifier | Optional |

| | | |
|------|--|---------------------|
| | <p>Das Datenelement <i>dataQualifier</i> codiert bestimmte Eigenschaften der Messwerte der Messwertliste. Gültige Werte nach ESPI REQ.21 sind:</p> <p>Valid values include:</p> <ul style="list-style-type: none"> 0 = Not Applicable 2 = Durchschnitt 8 = Maximum 9 = Minimum 12 = Normal <p>Die Nutzung des Datenelements <i>dataQualifier</i> ist optional.</p> | |
| 1.19 | <p>ReadingType – defaultQuality</p> <p>Das Datenelement <i>defaultQuality</i> beschreibt den Standardwert für den Messwertstatus. Dieser kann genutzt werden, falls für einzelne Messwerte in der Klasse <i>ReadingQuality</i> das Datenelement <i>quality</i> nicht gefüllt wurde. Gültige Werte nach ESPI REQ.21 sind:</p> <ul style="list-style-type: none"> 0 – valid (gültig) 7 – manually edited (manuell geändert) 8 - estimated using reference day (geschätzt basierend auf einem Referenztag) 9 - estimated using linear interpolation (geschätzt mittels linearer Interpolation) 10 – questionable (unplausibel) 11 – derived (abgeleitet) 12 - projected (forecast) (hochgerechnet (Prognose)) 13 – mixed (gemischt) 14 – raw (roh/unverarbeitet) 15 – normalized for weather (normalisiert im Bezug zum Wetter) 16 – andere (other) 17 – validated (gültig) 18 – verified (überprüft) <p>Die Nutzung des Datenelements <i>defaultQuality</i> ist optional.</p> | Optional |
| 1.20 | <p>ReadingType – flowDirection</p> <p>Das Datenelement <i>flowDirection</i> beschreibt die Fließrichtung, die mit den Messwerten verbunden ist. Gültige Werte nach ESPI REQ.21 sind:</p> <ul style="list-style-type: none"> 0 = nicht anwendbar 1 = Vorwärts 19 = Rückwärts <p>Die Nutzung des Datenelements <i>flowDirection</i> ist optional.</p> | Optional |
| 1.21 | <p>ReadingType – intervallLength</p> <p>Das Datenelement <i>intervallLength</i> beschreibt die Standard-Intervalllänge die für die einzelnen Messwerte zugrunde liegt. Die Intervalllänge wird nach ESPI REQ.21 in Sekunden spezifiziert.</p> <p>Die Standard-Intervalllänge muss in einer Instanz der Klasse <i>ReadingType</i> beschrieben werden. Im Falle einer originären Messwertliste muss dieser Wert der Registrierperiode entsprechen.</p> | Erforderlich |
| 1.22 | <p>ReadingType – kind</p> <p>Das Datenelement <i>kind</i> codiert eine generelle Klassifikation der in der Messwertliste beinhalteten Werte dar. Gültige Werte nach ESPI REQ.21 sind:</p> <ul style="list-style-type: none"> 0 = Not Applicable (nicht anwendbar) 3 = Currency (Währung) 4 = Current (Stromstärke, Ampere) 5 = CurrentAngle (Strömungswinkel) 7 = Date (Datum) 8 = Demand (Abruf) 12 = Energy (Energie) 15 = Frequency (Frequenz) 37 = Power (Leistung) | Optional |

| | | |
|------|--|---------------------|
| | 38 = PowerFactor (Leistungsfaktor) 40 = QuantityPower (Leistungsmenge) 54 = Voltage (Spannung) 55 = VoltageAngle (Spannungswinkel) 64 = DistortionPowerFactor (Klirrfaktor) 155 = VolumetricFlow (Volumenstrom) Die Nutzung des Datenelements <i>kind</i> ist optional . | |
| 1.23 | ReadingType – phase Das Datenelement <i>phase</i> codiert die Phase, die mit den Messwerten der Messwertliste in Verbindung steht. Beispiele nach ESPI REQ.21 sind: 0 = Not Applicable (nicht anwendbar) 129 = Phase AN 128 = Phase A 132 = Phase AB Die Nutzung des Datenelements <i>phase</i> ist optional . | Optional |
| 1.24 | ReadingType – powerOfTenMultiplier Das Datenelement <i>powerOfTenMultiplier</i> repräsentiert den Einheitenvorsatz der Maßeinheit der in der Messwertliste übermittelten Werte. Gültige Werte nach ESPI REQ.21 sind: 0 = None 1 = deca= $\times 10$ 2 = hecto= $\times 100$ -3 = mili= $\times 10^{-3}$ 3 = kilo= $\times 1000$ 6 = Mega= $\times 10^6$ -6 = micro= $\times 10^{-6}$ 9 = Giga= $\times 10^9$ Das Datenelement <i>powerOfTenMultiplier</i> muss mit einem entsprechenden Wert gefüllt werden. | Erforderlich |
| 1.25 | ReadingType – timeAttribute Das Datenelement <i>timeAttribute</i> wird genutzt, um bestimmte Intervalltypen zu spezifizieren, mit denen die Messwertlisten beschrieben werden können. Beispiele für gültige Werte nach ESPI REQ.21 sind: 0 = Not Applicable 1 = 10-Minuten 2 = 15-Minuten 11 = täglich 24 = wöchentlich Die Nutzung des Datenelements <i>timeAttribute</i> ist optional . | Optional |
| 1.26 | ReadingType – tou Das Datenelement <i>tou</i> kann genutzt werden, um bestimmte Time-Of-Use-Blöcke zu codieren. Beispiele nach ESPI REQ.21 sind: 0 = Nicht anwendbar 1 = TOU A 2 = TOU B Die Nutzung des Datenelements <i>tou</i> ist optional . | Optional |
| 1.27 | Readingtype – uom Das Datenelement <i>uom</i> codiert die Maßeinheit, welche für alle Messwerte der Messwertliste gilt. Gültige Werte nach ESPI REQ.21 sind: 0 = Not Applicable 5 = A (Current) 29 = Voltage 31 = J (Energy joule) 33 = Hz (Frequency) 38 = Real power (Watts) 42 = m3 (Cubic Meter) | Erforderlich |

| | | |
|------|---|---------------------|
| | 61 = VA (Apparent power) 63 = VAR (Reactive power) 65 = CosPhi (Power factor) 67 = V ² (Volts squared) 69 = A ² (Amp squared) 71 = VAh (Apparent energy) 72 = Real energy (Watt-hours) 73 = VARh (Reactive energy) 106 = Ah (Ampere-hours / Available Charge) 119 = ft ³ (Cubic Feet) 122 = ft ³ /h (Cubic Feet per Hour) 125 = m ³ /h (Cubic Meter per Hour) 128 = US gl (US Gallons) 129 = US gl/h (US Gallons per Hour) Das Datenelement <i>uom</i> muss mit einem entsprechenden Wert gefüllt werden. | |
| 1.28 | ReadingType – converterFactor Das Datenelement <i>converterFactor</i> kann genutzt werden um einen Wand- lurfaktor mit aufzunehmen. Die Nutzung des Datenelements <i>converterFactor</i> ist optional . | Optional |
| 1.29 | ReadingType – description Das Datenelement <i>description</i> kann genutzt werden um eine Beschrei- bung der Messwertliste einzufügen. Diese Beschreibung kann 256 Zeichen beinhalten und sollte den OBIS-Code für den Letztverbraucher näher spe- zifizieren. Die Nutzung des Datenelements <i>description</i> ist optional . | Optional |
| 1.30 | ReadingType – obisCode Das Datenelement <i>obisCode</i> codiert die Messwerte der Messwertliste wie im Object Identification System (OBIS) nach DIN EN 62056-61 und für die Nutzung in den EDIFACT-Nachrichtentypen des deutschen Energiemarktes beschrieben. Das Datenelement <i>obisCode</i> muss mit einem entsprechenden Wert (OBIS- Kennzahl) gefüllt werden. | Erforderlich |
| 1.31 | ReadingType – qualifiedLogicalName Das Datenelement <i>qualifiedLogicalName</i> ist ein eindeutiger Bezeichner einer Messreihe, die von einem Smart Meter Gateway übertragen werden kann. Er setzt sich nach der COSEM-Definition zusammen aus <OBIS>.<SMGw-ID>.sm und wird für die Bildung der inneren Signatur ge- nutzt. Jede Instanz der Klasse <i>readingType</i> muss ein Datenelement <i>qualifiedLo- gicalName</i> enthalten. | Erforderlich |
| 1.32 | IntervalBlock Die Klasse <i>IntervalBlock</i> enthält die einzelnen Intervallwerte. Eine Nach- richt muss mindestens eine Instanz der Klasse <i>IntervalBlock</i> enthalten. Jede Instanz der Klasse <i>IntervalBlock</i> : <ul style="list-style-type: none"> • muss auf mindestens eine Instanz der Klasse <i>IntervalReading</i> ver- weisen | Erforderlich |
| 1.33 | IntervalBlock – interval Das Datenelement <i>interval</i> beschreibt die gesamte Zeitperiode, für die die nachfolgenden Messwerte in der Messwertliste enthalten sind. Die Zeit- periode wird durch einen Startzeitpunkt und eine Dauer definiert. Der Startzeitpunkt wird als ganzzahliger Wert beschrieben. Dieser Wert reprä- sentiert die vergangenen Sekunden seit dem 01.01.1970 00:00:00Uhr. Die | Erforderlich |

| | | |
|------|---|----------------------|
| | Dauer wird ebenfalls als ganzzahliger Sekundenwert beschrieben. Jede Instanz der Klasse <i>IntervalBlock</i> muss ein Datenelement vom Typ <i>interval</i> enthalten. | |
| 1.34 | IntervalReading Die Klasse <i>IntervalReading</i> repräsentiert die Daten zu einem konkreten Messwert. Jede Nachricht muss mindestens eine Instanz der Klasse <i>IntervalReading</i> enthalten. Jede Instanz der Klasse <i>IntervalReading</i> : <ul style="list-style-type: none"> muss auf eine Instanz der Klasse <i>ReadingQuality</i> verweisen. | Erforderlich |
| 1.35 | IntervalReading – cost Das Datenelement <i>cost</i> kann genutzt werden, um die Kosten, die mit diesem konkreten Messwert verbunden sind, darzustellen. Die Nutzung des Datenelements <i>cost</i> ist optional . | Optional |
| 1.36 | IntervalReading – timePeriod Das Datenelement <i>timePeriod</i> beschreibt das Intervall für das der angegebene Messwert gültig ist. Das Intervall wird durch einen Startzeitpunkt und eine Dauer definiert. Der Startzeitpunkt wird als ganzzahliger Wert beschrieben. Dieser Wert repräsentiert die vergangenen Sekunden seit dem 01.01.1970 00:00:00Uhr (UTC-Zeit). Für eichrechtlich relevante Daten muss hier zwingend die „capture_time“ eingetragen werden, also der sekundengenaue Zeitpunkt der Messwerterfassung. Dieser wird zur Überprüfung der inneren Signatur benötigt. Die Dauer wird ebenfalls als ganzzahliger Sekundenwert beschrieben und sollte dem Datenelement <i>intervalLength</i> der Klasse <i>ReadingType</i> entsprechen. Jede Instanz der Klasse <i>IntervalReading</i> muss ein Datenelement vom Typ <i>timePeriod</i> enthalten. | Erforderlich |
| 1.37 | IntervalReading – value Das Datenelement <i>value</i> repräsentiert den Wert der Messung. Dieser wird als ganzzahliger Wert definiert. Jede Instanz der Klasse <i>IntervalReading</i> muss ein Datenelement <i>value</i> enthalten. | Erforderlich |
| 1.38 | IntervalReading – certId Das Datenelement <i>certId</i> enthält die ID des Zertifikats der signierenden Instanz des Messwertes, sofern die Signatur nicht vom Smart Meter Gateway stammt. Dies kann zum Beispiel der Fall sein, wenn ein Messwert von einer Marktrolle nachträglich manuell geändert wurde. Das entsprechende Zertifikat wird in der Klasse <i>Certificate</i> übertragen. Ist <i>certId</i> nicht angegeben, so wird davon ausgegangen, dass das Zertifikat des Smart Meter Gateway zur Überprüfung der Signatur genutzt wird. Eine Instanz der Klasse <i>IntervalReading</i> muss ein Datenelement <i>certId</i> beinhalten, sofern eine eichrechtlich-konforme Prüfung durchgeführt werden soll und die Signatur nicht vom Smart Meter Gateway erstellt wurde. In den weiteren Fällen ist die Nutzung des Datenelements optional . | Erforderlich* |
| 1.39 | IntervalReading – signature Das Datenelement <i>signature</i> repräsentiert die innere Signatur des Messwertes. Diese wird vom Smart Meter Gateway gebildet und den Messwerten der Messwertliste zugeordnet. Eine Instanz der Klasse <i>IntervalReading</i> muss ein Datenelement <i>signature</i> enthalten. | Erforderlich* |
| 1.40 | IntervalReading – status | Erforderlich* |

| | | |
|------|--|----------------------|
| | <p>Das Datenelement <i>status</i> repräsentiert das Statuswort, welches aus dem Statuswort des Zählers und des Smart Meter Gateways zusammensetzt. Dieses Statuswort wird laut FNN Lastenheft Smart Meter Gateway [FNN SNGW] als octet-String übertragen und wird für die Überprüfung der inneren Signatur benötigt.</p> <p>Eine Instanz der Klasse <i>IntervalReading</i> muss ein Datenelement <i>status</i> beinhalten.</p> | |
| 1.41 | <p>ReadingQuality</p> <p>Die Klasse <i>ReadingQuality</i> repräsentiert den Status eines Messwertes. Jeder Messwert muss mit einer entsprechenden Statusinformation gekennzeichnet werden.</p> <p>Die Klasse <i>ReadingQuality</i> verweist auf keine weiteren Klassen.</p> | Erforderlich |
| 1.42 | <p>ReadingQuality – quality</p> <p>Das Datenelement <i>quality</i> repräsentiert die eigentliche Statusinformation zu einem Messwert. Gültige Werte nach ESPI REQ.21 sind:</p> <ul style="list-style-type: none"> 0 – valid (gültig) 7 – manually edited (manuell geändert) 8 - estimated using reference day (geschätzt basierend auf einem Referenztag) 9 - estimated using linear interpolation (geschätzt mittels linearer Interpolation) 10 – questionable (unplausibel) 11 – derived (abgeleitet) 12 - projected (forecast) (hochgerechnet (Prognose)) 13 – mixed (gemischt) 14 – raw (roh/unverarbeitet) 15 – normalized for weather (normalisiert im Bezug zum Wetter) 16 – andere (other) 17 – validated (gültig) 18 – verified (überprüft) <p>Jede Instanz der der Klasse <i>ReadingQuality</i> muss ein Datenelement <i>quality</i> mit einem entsprechenden Wert enthalten.</p> | Erforderlich |
| 1.43 | <p>Meter</p> <p>Die Klasse <i>Meter</i> identifiziert den Zähler über die enthaltene Zählerkennung. Eine originäre Messliste muss eine Zählernummer enthalten.</p> <p>Die Klasse <i>Meter</i> verweist auf keine weiteren Klassen.</p> | Erforderlich |
| 1.44 | <p>Meter – meterId</p> <p>Das Datenelement <i>meterId</i> repräsentiert die Zählernummer des Smart Meters.</p> <p>Eine Instanz der Klasse <i>Meter</i> muss ein Datenelement <i>meterId</i> enthalten.</p> | Erforderlich |
| 1.45 | <p>Meter – publicKey</p> <p>Das Datenelement <i>publicKey</i> der Klasse <i>Meter</i> kann genutzt werden um den öffentlichen Schlüssel des Smart Meters zu übertragen. Dies ist insbesondere der Fall, wenn das Smart Meter die Zähler mit Zählersignatur versendet. Der Wert <i>publicKey</i> ermöglicht damit die Überprüfung der Zählersignatur.</p> <p>Die Nutzung des Datenelements <i>publicKey</i> ist optional.</p> | Optional |
| 1.46 | <p>SMGW</p> <p>Die Klasse <i>SMGW</i> repräsentiert Informationen zum Smart Meter Gateway, von welchem die beinhaltenden Messwertlisten stammen. Eine Nachricht muss eine Instanz von <i>SMGW</i> enthalten.</p> <p>Die Klasse <i>SMGW</i> verweist auf keine weiteren Klassen.</p> | Erforderlich |
| 1.47 | <p>SMGW - certId</p> <p>Das Datenelement <i>certId</i> enthält die ID des Zertifikats des signierenden Smart Meter Gateways. Das Zertifikat wird in der Klasse <i>Certificate</i> über-</p> | Erforderlich* |

| | | |
|------|--|----------------------|
| | tragen. Als Default erhält die ID des Zertifikats des SMGW den Wert 1. Jede Instanz von <i>SMGW</i> muss ein Datenelement vom Typ <i>certId</i> beinhalten, sofern eine eichrechtlich-konforme Prüfung durchgeführt werden soll. | |
| 1.48 | SMGW – smgwId Das Datenelement <i>smgwId</i> repräsentiert die Kennung des Smart Meter Gateways, von denen die Daten der Nachricht und der öffentliche Schlüssel stammen. Jede Instanz der Klasse <i>SMGW</i> muss ein Datenelement vom Typ <i>smgwId</i> enthalten. | Erforderlich |
| 1.49 | Certificate Die Klasse <i>Certificate</i> repräsentiert das Zertifikat, welches für die Inhaltsdatensignierung genutzt wird. Wird von einer anderen Rolle als dem SMGW ein Messwert signiert (z.B. bei manueller Änderung eines Messwerts), so ist das entsprechende Zertifikat hier zusätzlich einzufügen. Für eine eichrechtlich-konforme Überprüfung der Daten muss die Nachricht mindestens eine Instanz der Klasse <i>Certificate</i> beinhalten. Die Klasse <i>Certificate</i> verweist auf keine weiteren Klassen. | Erforderlich* |
| 1.50 | Certificate – certId Das Datenelement <i>certId</i> identifiziert ein Zertifikat eindeutig. Das Zertifikat des SMGW hat per Default die ID mit der Nummer 1. Jede Instanz der Klasse <i>Certificate</i> muss ein Datenelement vom Typ <i>CertId</i> enthalten. | Erforderlich* |
| 1.51 | Certificate – cSig Das Datenelement <i>cSig</i> enthält das Inhaltsdaten-Signatur-Zertifikat. Jede Instanz der Klasse <i>Certificate</i> muss ein Datenelement vom Typ <i>cSig</i> enthalten. | Erforderlich* |
| 1.52 | InvoicingParty Die Klasse <i>InvoicingParty</i> repräsentiert den Rechnungssteller bzw. die Marktrolle, die für die Tarifierung der Messdaten für den Letztverbraucher verantwortlich ist. Jede Nachricht muss eine Instanz der Klasse <i>InvoicingParty</i> beinhalten. Die Klasse <i>InvoicingParty</i> verweist auf keine weiteren Klassen. | Erforderlich |
| 1.53 | InvoicingParty – invoicingPartyId Das Datenelement <i>invoicingPartyId</i> wird genutzt, um die eindeutige Kennung des Marktteilnehmers, welcher für die Tarifierung der Messdaten für den Letztverbraucher verantwortlich ist. Dies kann zum Beispiel die Marktpartner-ID aus den EDIFACT-Nachrichten sein. Die eindeutige Kennung muss dem Letztverbraucher vorab bekannt gemacht werden. Jede Instanz der Klasse <i>InvoicingParty</i> muss ein Datenelement vom Typ <i>invoicingPartyId</i> beinhalten. | Erforderlich |
| 1.54 | Customer Die Klasse <i>Customer</i> repräsentiert den Letztverbraucher, welcher über eine je Rechnungsstellende eindeutige Identifikationsnummer beschrieben wird. Jede Nachricht muss eine Instanz der Klasse <i>Customer</i> beinhalten. Die Klasse <i>Customer</i> verweist auf keine weiteren Klassen. | Erforderlich |
| 1.55 | Customer – customerId Das Datenelement <i>customerId</i> beinhaltet eine eindeutige Identifikation des Letztverbrauchers. Diese Identifikation sollte mindestens je Marktteilnehmer eindeutig sein. Genutzt werden kann hier auch die Vertragsnummer des Letztverbrauchers beim Marktteilnehmer. Die Letztverbrauchererkennung ist dem Letztverbraucher vorab bekannt zu machen. Jede Instanz der Klasse <i>Customer</i> muss ein Datenelement vom Typ <i>custo-</i> | Erforderlich |

| | | |
|--|--------------------------|--|
| | <i>merId</i> beinhalten. | |
|--|--------------------------|--|

6.2 Prüfungsdaten

Die Prüfungsdaten werden zur eichrechtlich konformen Rechnungsprüfung benötigt. Zentrales Datenelement der Prüfungsdaten ist die Klasse *AnalysisProfile*.

| Nr. | Beschreibung | Verwendung |
|------|--|--|
| 2.01 | AnalysisProfile Die Klasse <i>AnalysisProfile</i> repräsentiert das Auswerteprofil. Bei der Übertragung von Daten zu Zwecken der eichrechtskonformen Rechnungsüberprüfung muss die Nachricht eine Instanz der Klasse enthalten. Bei der Übertragung von Daten für das Alltagsdisplay ist die Nutzung der Klasse <i>AnalysisProfile</i> optional . Eine Instanz der Klasse <i>AnalysisProfile</i> : <ul style="list-style-type: none"> • muss auf mindestens eine Instanz der Klasse <i>TariffStage</i> verweisen • kann auf eine Klasse <i>RuleSystem</i> verweisen | Erforderlich (Rechnungsprüfung) Optional (Alltagsdisplay) |
| 2.02 | AnalysisProfile – billingPeriod Das Datenelement <i>billingPeriod</i> spezifiziert den Abrechnungszeitraum für den dieses Auswerteprofil Gültigkeit hat. Eine Instanz der Klasse <i>AnalysisProfile</i> muss genau ein Datenelement <i>billingPeriod</i> enthalten. | Erforderlich |
| 2.03 | AnalysisProfile – description Das Datenelement <i>description</i> kann zur Beschreibung des Tarifs genutzt werden, für den das Auswerteprofil gültig ist. Das Datenelement <i>description</i> ist optional . | Optional |
| 2.04 | AnalysisProfile – tariffId Das Datenelement <i>tariffId</i> entspricht der Tarifkennbezeichnung des Lieferanten. Eine Instanz der Klasse <i>AnalysisProfile</i> muss genau ein Datenelement <i>tariffId</i> enthalten | Erforderlich |
| 2.05 | AnalysisProfile – defaultTariffNumber Das Datenelement <i>defaultTariffNumber</i> ist eine Referenz auf die Standardtarifstufe, die zu Beginn eines Abrechnungszeitraums gültig ist. Eine Instanz der Klasse <i>AnalysisProfile</i> muss genau ein Datenelement <i>defaultTariffNumber</i> enthalten. | Erforderlich |
| 2.06 | RuleSystem Die Klasse <i>RuleSystem</i> repräsentiert das Regelwerk. Eine Instanz der Klasse <i>RuleSystem</i> ist optional . Eine Nutzung im Rahmen des Tarifierungsfalls 7 wird empfohlen. | Optional |
| 2.07 | RuleSystem - rules Das Datenelement <i>rules</i> spezifiziert eine URI/URL als Link zum Ablageort des Regelwerks. Dieses kann dann von der Basissoftware dort heruntergeladen werden. Eine Instanz der Klasse <i>RuleSystem</i> muss einen Link zum Regelwerk enthalten. | Erforderlich |
| 2.08 | TariffStage Die Klasse <i>TariffStage</i> spezifiziert einzelnen Tarifstufen des Auswerteprofiles. Ein Auswerteprofil muss mindestens eine Instanz der Klasse <i>TariffStage</i> enthalten. Eine Instanz der Klasse <i>TariffStage</i> : <ul style="list-style-type: none"> • muss auf eine Instanz der Klasse <i>DatetimePeriod</i> verweisen | Erforderlich |

| | | |
|------|--|-------------------------------|
| | <ul style="list-style-type: none"> • kann auf eine oder zwei Instanz der Klasse <i>MinMaxCounter</i> verweisen • kann auf Instanzen der Klasse <i>TariffChangeTrigger</i> verweisen • kann auf eine Instanz der Klasse <i>Prepaid</i> verweisen | |
| 2.09 | TariffStage – tariffNumber Das Datenelement <i>tariffNumber</i> spezifiziert die Tarifnummer der Tarifstufe. Eine Instanz der Klasse <i>TariffStage</i> muss genau ein Datenelement <i>tariffNumber</i> enthalten. | Erforderlich |
| 2.10 | TariffStage – tariffUseCase Das Datenelement <i>tariffUseCase</i> spezifiziert den Tarifierfassungsfall für den diese Tarifstufe gültig ist. Die angegebene Nummer entspricht dabei dem Anwendungsfall wie er in der Technischen Richtlinie des BSI definiert wurde. Eine Instanz der Klasse <i>TariffStage</i> muss genau ein Datenelement der Klasse <i>tariffUseCase</i> beinhalten. | Erforderlich |
| 2.11 | TariffStage – description Das Datenelement <i>description</i> dient zur freien Beschreibung der Tarifstufe. Die Nutzung ist optional . | Optional |
| 2.12 | TariffStage – obisCode Das Datenelement <i>obisCode</i> der Klasse <i>TariffStage</i> beschreibt den OBIS-Code der Tarifstufe. Mit Hilfe des OBIS-Codes ist eine Zuordnung der Tarifstufen zu Messwertlisten ermöglicht. Jede Instanz der Klasse <i>TariffStage</i> muss genau ein Datenelement <i>obisCode</i> enthalten. | Erforderlich |
| 2.13 | DatetimePeriod Die Klasse <i>DateTimePeriod</i> beschreibt den Gültigkeitszeitraum einer Tarifstufe. Eine Instanz der Klasse <i>TariffStage</i> muss genau auf eine Instanz der Klasse <i>DateTimePeriod</i> verweisen. Eine Instanz der Klasse <i>DateTimePeriod</i> muss: <ul style="list-style-type: none"> • entweder ein Datenelement <i>datetimeInterval</i> beinhalten • oder ein Datenelement <i>datetimeVar</i> beinhalten Eine Instanz der Klasse <i>DateTimePeriod</i> verweist auf keine weiteren Klassen. | Erforderlich |
| 2.14 | DatetimePeriod – datetimeInterval Das Datenelement <i>datetimeInterval</i> spezifiziert den Gültigkeitszeitraum, zu dem eine Tarifstufe gültig ist. Bedingung: Entweder muss das Datenelement <i>datetimeInterval</i> oder es muss das Datenelement <i>datetimeVar</i> in der Instanz der Klasse <i>DateTimePeriod</i> angegeben werden. | Erforderlich (bedingt) |
| 2.15 | DatetimePeriod – datetimeVar Das Datenelement <i>datetimeVar</i> spezifiziert den Gültigkeitszeitraum unter der Angabe des Startzeitpunkts. Bedingung: Entweder muss das Datenelement <i>datetimeInterval</i> oder es muss das Datenelement <i>datetimeVar</i> in der Instanz der Klasse <i>DateTimePeriod</i> angegeben werden. | Erforderlich (bedingt) |
| 2.16 | MinMaxCounter Die Klasse <i>MinMaxCounter</i> beschreibt die Anzahl und Eigenschaft der Minimal- bzw. Maximalwerte, die für eine Tarifstufe ermittelt bzw. angegeben werden sollen. Die Verwendung der Klasse <i>MinMaxCounter</i> ist optional Die Klasse <i>MinMaxCounter</i> kann maximal zweimal instanziiert werden, um so Minimalwerte und Maximalwerte zu ermitteln bzw. anzugeben. | Optional |

| | | |
|------|---|---------------------|
| | Die Klasse <i>MinMaxCounter</i> verweist auf keine weiteren Klassen. | |
| 2.17 | MinMaxCounter – count Das Datenelement <i>count</i> spezifiziert die Anzahl der zu ermittelnden bzw. anzugebenden Minimal- oder Maximalwerte. Eine Instanz der Klasse <i>MinMaxCounter</i> muss ein Datenelement <i>count</i> enthalten werden. | Erforderlich |
| 2.18 | MinMaxCounter – maxCounter Das Datenelement <i>maxCounter</i> spezifiziert einen boolschen Wert (Flag). Ist der Wert 1=true, werden Maximalwerte angegeben bzw. ermittelt. Ist der Wert des Datenelements 0=false, werden Minimalwerte angegeben bzw. ermittelt. Eine Instanz der Klasse <i>MinMaxCounter</i> muss ein Datenelement vom Typ <i>maxCounter</i> enthalten. Das Datenelement muss den Wert 0 oder den Wert 1 annehmen. | Erforderlich |
| 2.19 | Prepaid Die Klasse <i>Prepaid</i> dient zur Auswertung von Prepaid-Tarifen (TAF12). Eine Instanz der Klasse <i>Prepaid</i> ist optional . Eine Instanz der Klasse <i>Prepaid</i> : <ul style="list-style-type: none"> • kann auf Instanzen der Klasse <i>Breaker</i> verweisen • kann auf Instanzen der Klasse <i>ThresholdTrigger</i> verweisen | Optional |
| 2.20 | Prepaid – availableEnergy Das Datenelement <i>availableEnergy</i> spezifiziert die bezahlte und damit verfügbare Menge an Energie des Prepaid-Tarifs. Eine Instanz der Klasse <i>Prepaid</i> muss ein Datenelement vom Typ <i>availableEnergy</i> enthalten. | Erforderlich |
| 2.21 | Prepaid – powerOfTenMultipller Das Datenelement <i>powerOfTenMultipller</i> spezifiziert den Einheitenvorsatz der bezahlten und verfügbaren Menge des Prepaid-Tarifs. Gültige Werte nach ESPI REQ.21 sind: 0 = None 1 = deca= $\times 10$ 2 = hecto= $\times 100$ -3 = mili= $\times 10^{-3}$ 3 = kilo= $\times 1000$ 6 = Mega= $\times 10^6$ -6 = micro= $\times 10^{-3}$ 9 = Giga= $\times 10^9$ Eine Instanz der Klasse <i>Prepaid</i> muss ein Datenelement vom Typ <i>powerOfTenMultipller</i> enthalten. | Erforderlich |
| 2.22 | Prepaid – toleranceInterval Das Datenelement <i>toleranceInterval</i> spezifiziert die Länge des Zeitraums in dem der Letztverbraucher nach Verbrauch der verfügbaren Energiemenge noch Energie beziehen kann. Eine Instanz der Klasse <i>Prepaid</i> muss ein Datenelement vom Typ <i>toleranceInterval</i> enthalten. | Erforderlich |
| 2.23 | Prepaid – uom Das Datenelement <i>uom</i> spezifiziert die Einheit in der die bezahlte und verfügbare Menge des Prepaid-Tarifs angegeben wird. Eine Instanz der Klasse <i>Prepaid</i> muss ein Datenelement vom Typ <i>uom</i> enthalten. | Erforderlich |
| 2.24 | Breaker Die Klasse <i>Breaker</i> beschreibt die Unterbrecher, die zum Zeitpunkt des Verbrauches der verfügbaren Energiemenge, ein Signal erhalten. | Optional |

| | | |
|------|---|--------------------------------|
| | <p>Instanzen der Klasse <i>Breaker</i> sind optional. Eine Instanz der Klasse <i>Breaker</i> verweist auf keine weiteren Klassen.</p> | |
| 2.25 | <p>Breaker – breakerId Das Datenelement <i>breakerId</i> spezifiziert die eindeutige Kennzeichnung des Unterbrechers. Eine Instanz der Klasse <i>Breaker</i> muss ein Datenelement vom Typ <i>breakerId</i> enthalten.</p> | Erforderlich |
| 2.26 | <p>TariffChangeTrigger Die Klasse <i>TariffChangeTrigger</i> abstrahiert alle weiteren Triggerformen und damit alle weiteren Tarifumschaltgründe. Eine Instanz der Klasse <i>TariffChangeTrigger</i>: <ul style="list-style-type: none"> muss auf genau eine Instanz der folgenden Klassen verweisen: <ul style="list-style-type: none"> <i>ThresholdTrigger</i> <i>ExternalTrigger</i> <i>TimeTrigger</i> Eine Instanz der Klasse <i>TariffChangeTrigger</i> ist optional.</p> | Optional |
| 2.27 | <p>ThresholdTrigger Die Klasse <i>ThresholdTrigger</i> beschreibt einen Trigger, der einen Tarifstufenwechsel durch Grenzwertüberschreitung oder -unterschreitung auslöst. Eine Instanz der Klasse <i>ThresholdTrigger</i> ist optional. Eine Instanz der Klasse <i>ThresholdTrigger</i>: <ul style="list-style-type: none"> muss auf eine Instanz der Klasse <i>Threshold</i> verweisen </p> | Optional |
| 2.28 | <p>ExternalEventTrigger Die Klasse <i>ExternalEventTrigger</i> beschreibt einen Trigger, der durch externe Events ausgelöst werden kann, um in die referenzierte Tarifstufe zu wechseln. Eine Instanz der Klasse <i>ExternalEventTrigger</i> ist optional.</p> | Optional |
| 2.30 | <p>Threshold Die Klasse <i>Threshold</i> beschreibt einen mengenbasierten Grenzwert, der aufgrund Grenzwertüberschreitung oder –unterschreitung einen Tarifstufenwechsel auslöst. Eine Instanz der Klasse muss angegeben werden, sofern eine Instanz der Klasse <i>ThresholdTrigger</i> existiert. Eine Instanz der Klasse <i>ExternalEvent</i> kann Instanzen der Klasse <i>Threshold</i> referenzieren, hierbei ist eine Instanz der Klasse <i>Threshold</i> optional. <ul style="list-style-type: none"> Eine Instanz der Klasse <i>Threshold</i> verweist auf keine weiteren Klassen. </p> | Erforderlich / Optional |
| 2.31 | <p>Threshold – direction Das Datenelement <i>direction</i> spezifiziert, ob die Grenzwerte, welche in den Datenelementen <i>overLimit</i> oder <i>underLimit</i> angegeben werden, über- oder unterschritten werden müssen, um einen Tarifstufenwechsel oder ein externes Event auszulösen. Durch 1=true oder 0=false wird festgelegt, ob ein Über- oder ein Unterschreiten eines Grenzwertes geprüft wird. Bei Prüfung auf Überschreitung (1=true) tritt das Ereignis bei Überschreitung des ‚over_limit‘ ein. Bei Prüfung auf Unterschreitung (0 = false) tritt das Ereignis bei Unterschreitung des ‚under_limit‘ ein. Das Datenelement <i>direction</i> ist optional.</p> | Optional |
| 2.32 | <p>Threshold – qualifiedLogicalName Das Datenelement <i>qualifiedLogicalName</i> spezifiziert die Referenz auf Messwertlisten auf die sich die Grenzwertdefinition beziehen. Das Datenelement <i>qualifiedLogicalName</i> referenziert damit auf das Datenelement <i>qualifiedLogicalName</i> der Klasse <i>MeterReading</i> in den Basisdaten. Hiermit</p> | Erforderlich |

| | | |
|------|---|----------------------------|
| | können auch Grenzwerte in Messwertlisten referenziert werden, die nicht die aktuelle Tarifstufe betreffen (z.B. Grenzwerte der Leistungsmessung). Eine Instanz der Klasse <i>Threshold</i> muss ein Datenelement <i>qualifiedLogical-Name</i> enthalten. | |
| 2.33 | Threshold – overLimit Das Datenelement <i>overLimit</i> spezifiziert den oberen Grenzwert eines Bandes zwischen den Grenzwerten abgebildet durch <i>overLimit</i> und <i>underLimit</i> . Die Datenelemente <i>overLimit</i> und <i>underLimit</i> können auf den gleichen Wert gesetzt werden. Das Datenelement <i>overLimit</i> ist vom Datentyp <i>ThresholdLimit</i> . Eine Instanz der Klasse <i>Threshold</i> muss ein Datenelement <i>overLimit</i> enthalten. | Erforderlich |
| 2.34 | Threshold – underLimit Das Datenelement <i>underLimit</i> spezifiziert den unteren Grenzwert eines Bandes zwischen den Grenzwerten abgebildet durch <i>overLimit</i> und <i>underLimit</i> . Die Datenelemente <i>overLimit</i> und <i>underLimit</i> können auf den gleichen Wert gesetzt werden. Das Datenelement <i>underLimit</i> ist vom Datentyp <i>ThresholdLimit</i> . Eine Instanz der Klasse <i>Threshold</i> muss ein Datenelement <i>underLimit</i> enthalten. | Erforderlich |
| 2.35 | ThresholdLimit Die Klasse <i>ThresholdLimit</i> spezifiziert den Datentyp für Grenzwerte inkl. Einheit und Einheitenvorsatz. Eine Instanz der Klasse ist je nach Verwendung des Datentyps optional . Die Klasse <i>ThresholdLimit</i> verweist auf keine weiteren Klassen. | Optional (Datentyp) |
| 2.36 | ThresholdLimit – powerOfTenMultiplier Das Datenelement <i>powerOfTenMultiplier</i> spezifiziert den Einheitenvorsatz des Grenzwertes. Gültige Werte nach ESPI REQ.21 sind: 0 = None 1 = deca=x10 2 = hecto=x100 -3 = mili=x10 ⁻³ 3 = kilo=x1000 6 = Mega=x10 ⁶ -6 = micro=x10 ⁻³ 9 = Giga=x10 ⁹ Das Datenelement muss genau einmal angegeben werden. | Erforderlich |
| 2.37 | ThresholdLimit – uom Das Datenelement <i>uom</i> spezifiziert die Einheit des Grenzwertes. Das Datenelement muss genau einmal angegeben werden. | Erforderlich |
| 2.38 | ThresholdLimit – value Das Datenelement <i>value</i> beschreibt den Wert des Grenzwertes. Das Datenelement muss genau einmal angegeben werden. | Erforderlich |
| 2.39 | ExternalEvent Eine Instanz der Klasse <i>ExternalEvent</i> beschreibt ein externes Event. Ist eine Instanz der Klasse <i>ExternalEventTrigger</i> vorhanden, muss diese auf genau eine Instanz der Klasse <i>ExternalEvent</i> verweisen. Eine Instanz der Klasse <i>ExternalEvent</i> : kann auf eine oder zwei Instanzen der Klasse <i>Threshold</i> verweisen. | Erforderlich |
| 2.40 | ExternalEvent – eventId Das Datenelement <i>eventId</i> spezifiziert die Kennzeichnung eines externen Events und dient zur Zuordnung von Events aus den Logdaten. | Erforderlich |

| | | |
|------|---|--------------------------------|
| | <ul style="list-style-type: none"> • Eine Instanz der Klasse <i>ExternalEvent</i> muss ein Datenelement vom Typ <i>eventId</i> enthalten. | |
| 2.41 | ExternalEvent – eventPeriod Das Datenelement <i>eventPeriod</i> spezifiziert den Zeitraum, in das Event gültig ist. Eine Instanz der Klasse <i>ExternalEvent</i> muss ein Datenelement vom Typ <i>eventId</i> enthalten. | Erforderlich |
| 2.42 | TimeTrigger Die Klasse <i>TimeTrigger</i> beschreibt einen Trigger, der anhand von zeitlichen Angaben einen Tarifstufenwechsel auslöst. Eine Instanz der Klasse <i>TimeTrigger</i> ist optional . Eine Instanz der Klasse <i>TimeTrigger</i> : <ul style="list-style-type: none"> • muss auf mindestens eine Instanz der Klasse <i>DayProfile</i> verweisen • kann auf Instanzen der Klasse <i>SpecialDayProfile</i> verweisen • kann auf Instanzen der Klasse <i>WeekProfile</i> verweisen • kann auf Instanzen der Klasse <i>SeasonProfile</i> verweisen | Optional |
| 2.43 | SeasonProfile Die Klasse <i>SeasonProfile</i> spezifiziert eine zeitliche Saison in der ein bestimmtes Wochenprofil (Zeit) referenziert wird. Eine Instanz der Klasse <i>SeasonProfile</i> : <ul style="list-style-type: none"> • muss auf eine Instanz der Klasse <i>WeekProfile</i> verweisen | Optional |
| 2.44 | SeasonProfile – seasonProfileName Das Datenelement <i>seasonProfileName</i> dient zur freien Beschreibung des Saisonnamens. Eine Instanz der Klasse <i>SeasonProfile</i> muss ein Datenelement vom Typ <i>seasonProfileName</i> enthalten. | Erforderlich |
| 2.45 | SeasonProfile – seasonStart Das Datenelement <i>seasonStart</i> spezifiziert den Start der Saison. Eine Instanz der Klasse <i>SeasonProfile</i> muss ein Datenelement vom Typ <i>seasonProfileName</i> enthalten. | Erforderlich |
| 2.46 | SeasonProfile – weekProfileName Das Datenelement <i>weekProfileName</i> spezifiziert die Kennzeichnung bzw. Identifizierung des referenzierten Wochenprofils. Eine Instanz der Klasse <i>SeasonProfile</i> muss ein Datenelement vom Typ <i>weekProfileName</i> enthalten. | Erforderlich |
| 2.47 | WeekProfile Die Klasse <i>WeekProfile</i> dient zur Beschreibung eines Wochenprofils. Hierbei kann jeder Tag einer Woche ein eigenes Tagesprofil zugeordnet bekommen. Eine Instanz der Klasse <i>SeasonProfile</i> muss auf eine Instanz der Klasse <i>WeekProfile</i> verweisen. Eine Instanz der Klasse <i>TimeTrigger</i> kann auf eine Instanz der Klasse <i>WeekProfile</i> verweisen. Die Klasse <i>WeekProfile</i> verweist auf keine weiteren Klassen. | Erforderlich / Optional |
| 2.48 | WeekProfile – monday Das Datenelement <i>monday</i> referenziert auf ein Tagesprofil (Datentyp <i>DayProfile</i>). Es ist das Datenelement <i>dayId</i> des zu referenzierenden Tagesprofils einzusetzen. Das Datenelement ist optional . | Optional |
| 2.49 | WeekProfile – tuesday Das Datenelement <i>tuesday</i> referenziert auf ein Tagesprofil (Datentyp <i>DayProfile</i>). Es ist das Datenelement <i>dayId</i> des zu referenzierenden Tagesprofils einzusetzen. Das Datenelement ist optional . | Optional |

| | | |
|------|---|--------------|
| 2.50 | WeekProfile – wednesday Das Datenelement <i>wednesday</i> referenziert auf ein Tagesprofil (Datentyp <i>DayProfile</i>). Es ist das Datenelement <i>dayId</i> des zu referenzierenden Tagesprofils einzusetzen. Das Datenelement ist optional . | Optional |
| 2.51 | WeekProfile – thursday Das Datenelement <i>thursday</i> referenziert auf ein Tagesprofil (Datentyp <i>DayProfile</i>). Es ist das Datenelement <i>dayId</i> des zu referenzierenden Tagesprofils einzusetzen. Das Datenelement ist optional . | Optional |
| 2.52 | WeekProfile – friday Das Datenelement <i>friday</i> referenziert auf ein Tagesprofil (Datentyp <i>DayProfile</i>). Es ist das Datenelement <i>dayId</i> des zu referenzierenden Tagesprofils einzusetzen. Das Datenelement ist optional . | Optional |
| 2.53 | WeekProfile – saturday Das Datenelement <i>saturday</i> referenziert auf ein Tagesprofil (Datentyp <i>DayProfile</i>). Es ist das Datenelement <i>dayId</i> des zu referenzierenden Tagesprofils einzusetzen. Das Datenelement ist optional . | Optional |
| 2.54 | WeekProfile – weekProfileName Das Datenelement <i>weekProfileName</i> spezifiziert die Kennzeichnung bzw. Identifizierung des Wochenprofils. Eine Instanz der Klasse <i>WeekProfile</i> muss ein Datenelement vom Typ <i>weekProfileName</i> enthalten. | Erforderlich |
| 2.55 | DayProfile Instanzen der Klasse <i>DayProfile</i> kapseln ein oder mehrere Instanzen der Klasse <i>DayTimeProfile</i> . Eine Instanz der Klasse <i>DayProfile</i> : <ul style="list-style-type: none"> • muss genau auf mindestens eine Instanz der Klasse <i>DayTimeProfile</i> verweisen. | Erforderlich |
| 2.56 | DayProfile – dayId Das datenelement <i>dayId</i> identifiziert ein Tagesprofil eindeutig. Es wird von anderen Klassen genutzt, um auf das entsprechende Tagesprofil zu referenzieren. Jede Instanz der Klasse <i>dayProfile</i> muss ein Datenelement vom Typ <i>dayId</i> beinhalten. | Erforderlich |
| 2.57 | DayTimeProfile Die Klasse <i>DayTimeProfile</i> spezifiziert die Tageszeit für die übergeordnete <i>DayProfile</i> -Instanz. Die Klasse <i>DayTimeProfile</i> verweist auf keine weiteren Klassen. | Erforderlich |
| 2.58 | DayTimeProfile – startTime Das Datenelement <i>startTime</i> spezifiziert den Startzeitpunkt des referenzierenden Tagesprofiles (Klasse <i>DayProfile</i>). Der Startzeitpunkt wird aus der (zeitunabhängigen) Angabe der Stunde (<i>TimeVarType - hour</i>) und der Minuten (<i>TimeVarType - minute</i>) gebildet. Als Defaultwert sollten für beide Werte 0 genutzt werden. Jede Instanz der Klasse <i>DayTimeProfile</i> muss ein Datenelement vom Typ <i>startTime</i> enthalten. | Erforderlich |
| 2.59 | DayTimeProfile – tariffNumber Das Datenelement <i>tariffNumber</i> verweist auf die Tarifstufe, die zu der angegebenen Startzeit (Datenelement <i>startTime</i>) gültig wird. Das Datenele- | Erforderlich |

| | | |
|------|---|---------------------|
| | <p>ment muss auf eine Instanz des Datenelements <i>tariffNumber</i> in der Klasse <i>TariffStage</i> verweisen.</p> <p>Jede Instanz der Klasse <i>DayTimeProfile</i> muss ein Datenelement vom Typ <i>tariffNumber</i> enthalten.</p> | |
| 2.60 | <p>SpecialDayProfile</p> <p>Die Klasse <i>SpecialDayProfile</i> spezifiziert Sondertage, die ein bestimmtes Tagesprofil abbilden müssen.</p> <p>Instanzen der Klasse <i>SpecialDayProfile</i> sind optional.</p> <p>Eine Instanz der Klasse <i>SpecialDayProfile</i>:</p> <ul style="list-style-type: none"> • muss auf eine Instanz der Klasse <i>DayProfile</i> verweisen. | Optional |
| 2.61 | <p>SpecialDayProfile – dayId</p> <p>Das Datenelement <i>dayId</i> der Klasse <i>SpecialDayProfile</i> referenziert auf das entsprechende Tagesprofil, welches die Startzeit definiert.</p> <p>Jede Instanz der Klasse <i>SpecialDayProfile</i> muss ein Datenelement vom Typ <i>dayId</i> enthalten.</p> | Erforderlich |
| 2.62 | <p>SpecialDayProfile – specialDayDate</p> <p>Das Datenelement <i>specialDayDate</i> spezifiziert das Datum des Sondertages.</p> <p>Das Datenelement <i>specialDayDate</i> spezifiziert das Datum des Sondertages.</p> <p>Das Datum wird über die Klasse <i>DayVarType</i> beschrieben.</p> <p>Eine Instanz der Klasse <i>SpecialDayProfile</i> muss ein Datenelement vom Typ <i>specialDayDate</i> enthalten.</p> | Erforderlich |
| 2.63 | <p>TimeVarType</p> <p>Die Klasse <i>TimeVarType</i> kann genutzt werden um feste (zeitzoneunabhängige) Zeitangaben als Einzelwerte mit Stunden- und Minutenangabe zu definieren</p> <p>Die Klasse <i>TimeVarType</i> verweist auf keine weiteren Klassen.</p> | Optional |
| 2.64 | <p>TimeVarType – hour</p> <p>Das Datenelement <i>hour</i> der Klasse <i>TimeVarType</i> kann genutzt werden um eine ganzzahlige Stundenangabe zu definieren.</p> <p>Die Nutzung des Datenelements <i>hour</i> ist optional.</p> | Optional |
| 2.65 | <p>TimeVarType – minute</p> <p>Das Datenelement <i>minute</i> der Klasse <i>TimeVarType</i> kann genutzt werden um eine ganzzahlige Minutenangabe zu definieren.</p> <p>Die Nutzung des Datenelements <i>minute</i> ist optional.</p> | Optional |
| 2.66 | <p>TimeVarType – second</p> <p>Das Datenelement <i>second</i> der Klasse <i>TimeVarType</i> kann genutzt werden um eine ganzzahlige Sekundenangabe zu definieren.</p> <p>Die Nutzung des Datenelements <i>second</i> ist optional.</p> | Optional |
| 2.67 | <p>TimeVarType – hundreds</p> <p>Das Datenelement <i>hundreds</i> der Klasse <i>TimeVarType</i> kann genutzt werden um eine ganzzahlige Hunderstel-Sekundenangabe zu definieren.</p> <p>Die Nutzung des Datenelements <i>hundreds</i> ist optional.</p> | Optional |
| 2.68 | <p>DayVarType</p> <p>Die Klasse <i>DayVarType</i> kann genutzt werden, um Datumsangaben darzustellen, zum Beispiel um Feiertage zu beschreiben. Ist in der Klasse <i>DayVarType</i> keine Jahresangabe enthalten, so gilt das angegebene Datum jährlich. Sind keine Monatsangaben enthalten, so gilt das angegebene Datum monatlich.</p> <ul style="list-style-type: none"> • Die Klasse <i>DayVarType</i> verweist auf keine weiteren Klassen. | Optional |
| 2.69 | <p>DayVarType – dayOfMonth</p> <p>Das Datenelement <i>dayOfMonth</i> kann genutzt werden, um den genauen Tag innerhalb eines Monats zu spezifizieren.</p> | Optional |

| | | |
|------|---|-----------------|
| | Die Nutzung des Datenelements <i>dayOfMonth</i> ist optional . | |
| 2.70 | DayVarType – dayOfWeek Das Datenelement <i>dayOfWeek</i> kann genutzt werden, um den Tag innerhalb einer Woche als Wert zwischen 1 und 7 zu spezifizieren. Dabei ist 1 = Montag und 7 = Sonntag, die weiteren Tage entsprechend kalendarischer Reihenfolge. Die Nutzung des Datenelements <i>dayOfWeek</i> ist optional . | Optional |
| 2.71 | DayVarType – month Das Datenelement <i>month</i> kann genutzt werden, um den Monat als ganzzahligen Wert zu beschreiben. Dabei ist 1 = Januar und 12 = Dezember, die weiteren Monate entsprechend kalendarischer Reihenfolge. Die Nutzung des Datenelements <i>month</i> ist optional . | Optional |
| 2.72 | DayVarType – year Das Datenelement <i>year</i> kann genutzt werden, um ein bestimmtes Jahr als ganzzahligen Wert zu beschreiben. Die Nutzung des Datenelements <i>year</i> ist optional . | Optional |
| 2.73 | LogEntry Die Klasse <i>LogEntry</i> spezifiziert Logeinträge. Eine Instanz der Klasse <i>LogEntry</i> : <ul style="list-style-type: none"> • kann auf eine Instanz der Klasse <i>LogEvent</i> verweisen | Optional |
| 2.74 | LogEntry – parentRecordNumber Das Datenelement <i>parentRecordNumber</i> dient der Markierung verketteter Ereignisse. Enthält die „Parent Record Number“ denselben Wert wie die „Record Number“, so handelt es sich um das Root-Element einer Kette von zusammenhängenden Ereignissen oder um ein Einzel-Ereignis. Enthält die „Parent Record Number“ den Wert der „Record Number“ eines zuvor bereits erzeugten Ereignisses, so handelt es sich um ein Kind-Element einer Kette von zusammenhängenden Ereignissen. Das Datenelement ist optional anzugeben. | Optional |
| 2.75 | LogEntry – recordNumber Das Datelement <i>recordNumber</i> ist der eineindeutige Bezeichner des Logeintrags. Dieser wird mit Ablegen des Eintrags im Logbuch durch die Geräte-Firmware erzeugt. Das Datenelement ist optional anzugeben. | Optional |
| 2.76 | LogEntry – repetitionCounter Das Datenelement <i>repetitionCounter</i> beschreibt den Wiederholungszähler und ist bei einem neu auftretenden Ereignis mit „1“ zu besetzen. Bei nachfolgenden Ereignissen kann der „Repetition Counter“ in Verbindung mit Logbüchern benutzt werden, um das mehrfache Einfügen der Ereignisse in ein Logbuch zu vermeiden. Dabei ist zu gewährleisten, dass immer der erste und letzte Logbuch-Eintrag vorhanden ist. Das Datenelement ist optional anzugeben. | Optional |
| 2.77 | LogEvent Die Klasse <i>LogEvent</i> spezifiziert Logereignisse. Instanzen der Klasse sind optional . | Optional |
| 2.78 | LogEvent – evidence Das Datenelement <i>evidence</i> enthält, falls vorhanden, die Signatur, die im Zusammenhang mit dem Logereignis aufgetreten ist. Das Datenelement ist optional anzugeben. | Optional |
| 2.79 | LogEvent – destination Das Datenelement <i>destination</i> beinhalten, falls vorhanden die URL des Adressaten, an den die Aktion gerichtet ist, die das Ereignis ausgelöst hat. Das Datenelement <i>destination</i> ist optional anzugeben. | Optional |

| | | |
|------|---|-----------------|
| 2.80 | LogEvent – type Das Datenelement <i>type</i> ist die eindeutige Beschreibung des Ereignisses mit dem Datentyp „Evis.“ Das Datenelement ist optional anzugeben. | Optional |
| 2.81 | LogEvent – level Das Datenelement <i>level</i> beschreibt die dem Ereignis zugeordnete Rubrik als ein Element der Enumeration aus {1=INFO, 2=WARNING, 3=ERROR, 4=FATAL, 5=EXTENSION}. Das Datenelement ist optional anzugeben. | Optional |
| 2.82 | LogEvent – messageExtension Das Datenelement <i>messageExtension</i> liefert die Texte aus ‚Message‘, jene Textteile, die in der Information zum ‚type‘ an Stelle der Platzhalter zu verwenden sind. Die Reihenfolge der Texte muss mit der Reihenfolge der Platzhalter übereinstimmen. Das Datenelement ist optional anzugeben. | Optional |
| 2.83 | LogEvent – outcome Das Datenelement <i>outcome</i> spezifiziert das Ergebnis der mit dem Ereignis verbundenen Aktion. Das Datenelement ist optional anzugeben. | Optional |
| 2.84 | LogEvent – subjectIdentity Das Datenelement <i>subjectIdentity</i> beschreibt die Identität der Quelle, die das Ereignis ausgelöst hat. Dieses Attribut wird als Default auf ‚0‘ gesetzt. Die Nutzung des Datenelements <i>subjectIdentity</i> ist optional. | |
| 2.85 | LogEvent – secondsIndex Das Datenelement <i>secondsIndex</i> beschreibt den monoton steigenden Sekundenindex aus dem SMGW. Das Datenelement ist optional anzugeben. | Optional |
| 2.86 | LogEvent – timestamp Das Datenelement <i>timestamp</i> beschreibt den Zeitstempel mit Zeitpunkt, wann das Ereignis eingetreten ist. Das Datenelement ist optional anzugeben. | Optional |
| 2.87 | Evis Die Klasse <i>Evis</i> beschreibt ein LogEvent eindeutig. Die Nutzung der Klasse <i>Evis</i> ist optional . | Optional |
| 2.88 | Evis – deviceType Das Datenelement <i>deviceType</i> beschreibt eine physische Ausprägung eines Gerätes. Im FNN werden diese Einheiten typischerweise über ein Lastenheft beschrieben. Z.B. FNN Lastenheft Basiszähler etc. Hinweis: Gerätetyp 00 beschreibt ein nicht spezifiziertes Gerät. Das Datenelement ist optional anzugeben. | Optional |
| 2.89 | Evis – eventId Das Datenelement <i>eventId</i> identifiziert ein Ereignis übergreifend über Gerätetyp, Modul und Funktion. Sie wird eindeutig pro vendorId beschrieben. Das Datenelement ist optional anzugeben. | Optional |
| 2.90 | Evis – eventSubId Das Datenelement <i>eventSubId</i> identifiziert die Detailmeldung. Das Datenelement ist optional anzugeben. | Optional |
| 2.91 | Evis – length Das Datenelement <i>length</i> beschreibt die Länge der Datenstruktur. Das Datenelement <i>length</i> ist optional anzugeben. | Optional |
| 2.92 | Evis – function Das Datenelement <i>function</i> identifiziert eine funktionale Einheit innerhalb | Optional |

| | | |
|-------|--|---------------------|
| | des Moduls. Die funktionale Einheit kann als Hardware oder Software ausgeprägt sein. Hinweis: Funktion 00 beschreibt eine nicht spezifizierte Funktion. Das Datenelement ist optional anzugeben. | |
| 2.93 | Evis – module Das Datenelement <i>module</i> identifiziert ein Modul als Quelle innerhalb eines Gerätes. Das Modul kann als Hardware oder Software ausgeprägt sein. Hinweis: Modul 00 beschreibt ein nicht spezifiziertes Modul. Das Datenelement ist optional anzugeben. | Optional |
| 2.94 | Evis – vendorId Das Datenelement <i>vendorId</i> gibt die Verwendung gemäß Flag-Kennung "the flag association" an (siehe FNN COSEM Definition). Das Datenelement ist optional anzugeben. | Optional |
| 2.95 | Evis – version Das Datenelement <i>version</i> spezifiziert die Version des Datentyps. Das Datenelement ist optional anzugeben. | Optional |
| 2.96 | MessageExtensionType Die Klasse <i>MessageExtensionType</i> liefert jene Textteile, die in der Information zum ‚event_type‘ an Stelle der Platzhalter zu verwenden sind. Die Reihenfolge der Texte muss mit der Reihenfolge der Platzhalter übereinstimmen und wird über das Datenelement <i>id</i> festgelegt. Die Nutzung der Klasse <i>MessageExtensionType</i> ist optional . | Optional |
| 2.97 | MessageExtensionType – id Das Datenelement <i>id</i> der Klasse <i>MessageExtensionType</i> identifiziert einen Textteil des Logeintrages und legt die Reihenfolge der Texte fest. Eine Instanz der Klasse <i>MessageExtension</i> muss immer ein Datenelement vom Typ <i>id</i> enthalten. | Erforderlich |
| 2.98 | MessageExtensionType – text Das Datenelement enthält den Text des Logeintrags Eine Instanz der Klasse <i>MessageExtension</i> muss immer ein Datenelement vom Typ <i>text</i> enthalten. | Erforderlich |
| 2.99 | MathOperation Die Klasse <i>MathOperation</i> spezifiziert mathematische Operationen zwischen Messwertlisten, um dadurch Zählerbeziehungen abzubilden. Eine Instanz der Klasse <i>MathOperation</i> : <ul style="list-style-type: none"> • muss auf eine Instanz der Klasse <i>Operand1</i> verweisen • muss auf eine Instanz der Klasse <i>Operator</i> verweisen • muss auf eine Instanz der Klasse <i>Operand2</i> verweisen | Optional |
| 2.100 | Operand1 Die Klasse <i>Operand1</i> spezifiziert den ersten Operand einer mathematischen Operation zwischen Messwertlisten. | Erforderlich |
| 2.101 | Operand1 – qualifiedLogicalName Das Datenelement <i>qualifiedLogicalName</i> spezifiziert den Identifikator der Messwertreihe dessen Werte mathematisch mit Werten einer weiteren Messwertreihe verknüpft werden sollen. Es muss eine Messwertliste über das Datenelement <i>qualifiedLogicalName</i> referenziert werden. | Erforderlich |
| 2.102 | Operator Die Klasse <i>Operator</i> spezifiziert den mathematischen Operator der mathematischen Operation als Container für das Datenelement <i>operator</i> . | Erforderlich |
| 2.103 | Operator – operator Das Datenelement <i>operator</i> spezifiziert den mathematischen Operator der mathematischen Operation als Element der Enumeration aus {1=Addition, | Erforderlich |

| | | |
|-------|---|---------------------|
| | 2=Subtraktion, 3=Division, 4=Multiplikation}. Für eine mathematische Operation muss ein Operator angegeben werden. | |
| 2.104 | Operand2 Die Klasse <i>Operand2</i> spezifiziert den zweiten Operand einer mathematischen Operation zwischen Messwertlisten. | Erforderlich |
| 2.105 | Operand2 – qualifiedLogicalName Das Datenelement <i>qualifiedLogicalName</i> spezifiziert den Identifikator der Messwertreihe dessen Werte mathematisch mit Werten einer weiteren Messwertreihe verknüpft werden sollen. Es muss eine Messwertliste über das Datenelement <i>qualifiedLogicalName</i> referenziert werden. | Erforderlich |

7 Sicherheits- und Datenintegritätsanforderungen an die Schnittstelle IF_3D_CON

Die TLS-abgesicherte Schnittstelle zum Download der XML-Datei vom 3D-Portal ist nach Anforderung des BfDI auf dem Sicherheitsniveau von Online-Banking abzusichern.

7.1 Überprüfung der inneren Signatur

Die innere Signatur für Messwerte wird laut Lastenheft SMGW [FNN SMGW] über den Inhalt der folgenden Datenelemente gebildet:

- Logical Name
- Status
- Capture Time
- Value
- Scalar_Unit

Die Daten werden in der Schnittstelle IF_3D_CON über den Inhalt folgender Datenelemente abgebildet:

- Logical Name → *ReadingType – qualifiedLogicalName*
- Status → *IntervalReading – status*
- Capture Time → *IntervalReading – timePeriod - start*
- Value → *IntervalReading - value*
- Scalar_Unit → *ReadingType – powerOfTenMultiplier*

Aus Kompatibilitätsgründen werden bestimmte Werte nicht in ihrem originalen Format verschickt. Vor der Bildung des Hash-Wertes zur Überprüfung der inneren Signatur müssen diese Werte wieder in das Format zum Zeitpunkt der Bildung des Hash-Wertes rücküberführt werden:

- *IntervalReading – timePeriod - start* wird als Wert vom Typ *TimeType* in UTC-Zeit übertragen und beschreibt die Anzahl der Millisekunden seit dem 01.01.1970 00:00:00. Als COSEM-Wert liegt Capture Time im Format nach ISO8601 vor. Ein entsprechendes Mapping ist durchzuführen.

Die innere Signatur wird im Regelfall vom Smart Meter Gateway generiert. In Ausnahmefällen (nach fehlerhafter Plausibilisierung und Ersatzwertbildung) kann diese auch von anderen Markttrollen gebildet werden, sofern diese die neuen Werte signieren. In diesem Fall ist das Zertifikat der entsprechenden Markttrolle der Nachricht hinzuzufügen und mit dem Datenelement *certId* zu referenzieren.

8 Services der Schnittstelle IF_3D_CON

Die Referenzarchitektur von Green Button beschreibt das Konzept einer kaskadierenden Autorisierung und sollte in einer Folgeversion der Anwendungsregel aufgegriffen werden.

Im Folgenden wird der zur Rechnungsprüfung durch den Letztverbraucher durchzuführende XML-Dateidownload beschrieben. Dieser Ablauf gilt auch für die Bereitstellung von Basisdaten. Der Prozessablauf orientiert sich an dem „Green Button Download My Data“ – Prinzip.

Vorteil dieses Prinzip ist es, dass der Letztverbraucher die Hoheit über seine Daten behält und beliebigen Programmen mit einer entsprechenden Importschnittstelle zur Verfügung stellen kann.

Nach dem Login beim 3D-Portal bekommt der Letztverbraucher alle für ihn zur Verfügung stehenden XML-Dateien angezeigt und kann diejenige Datei auswählen, die für ihn in diesem Sinne relevant ist.

8.1 Beschreibung des XML-Dateidownloads zur Rechnungsprüfung

In Abbildung 17 wird der Prozess zur Rechnungsprüfung durch den Letztverbraucher beschrieben.

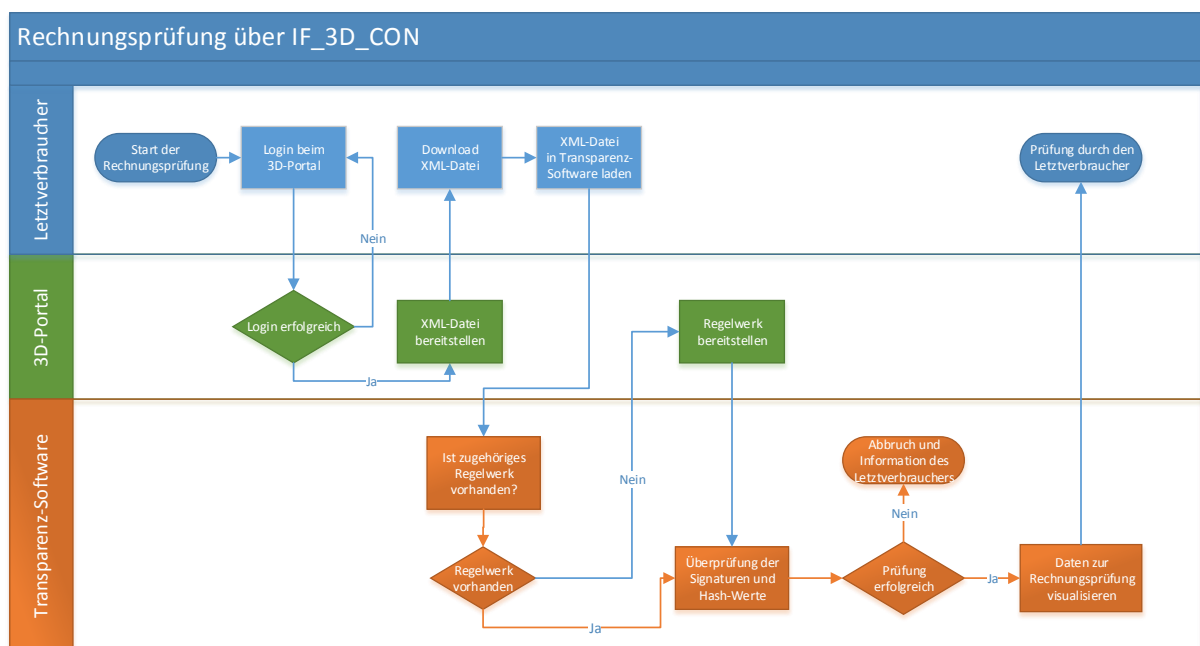


Abbildung 17: Prozess zur Rechnungsprüfung

Für jede Rechnung die ein Letztverbraucher erhält muss eine entsprechende Datei bereitgestellt werden. Diese muss bei Tarifierung im Smart Meter Gateway mindestens die abgeleiteten Messwertlisten mit Daten über den gesamten Rechnungszeitraum enthalten. Auswerteprofile können optional enthalten sein.

Bei dezentraler Tarifierung (TAF 7) müssen die originäre Messwertliste über den Abrechnungszeitraum, ein Auswerteprofil und ein Verweis auf das Regelwerk in der XML-Datei enthalten sein.

Die von dem 3D-Portal bereitgestellte Datei sollte über den Dateinamen eindeutig identifizierbar sein.

8.2 Beschreibung des XML-Dateidownloads zur Darstellung von Verbrauchsdaten (Energy Awareness)

In Abbildung 18 wird der Prozess der Datenbereitstellung zur Verbrauchsdatenvisualisierung beschrieben.

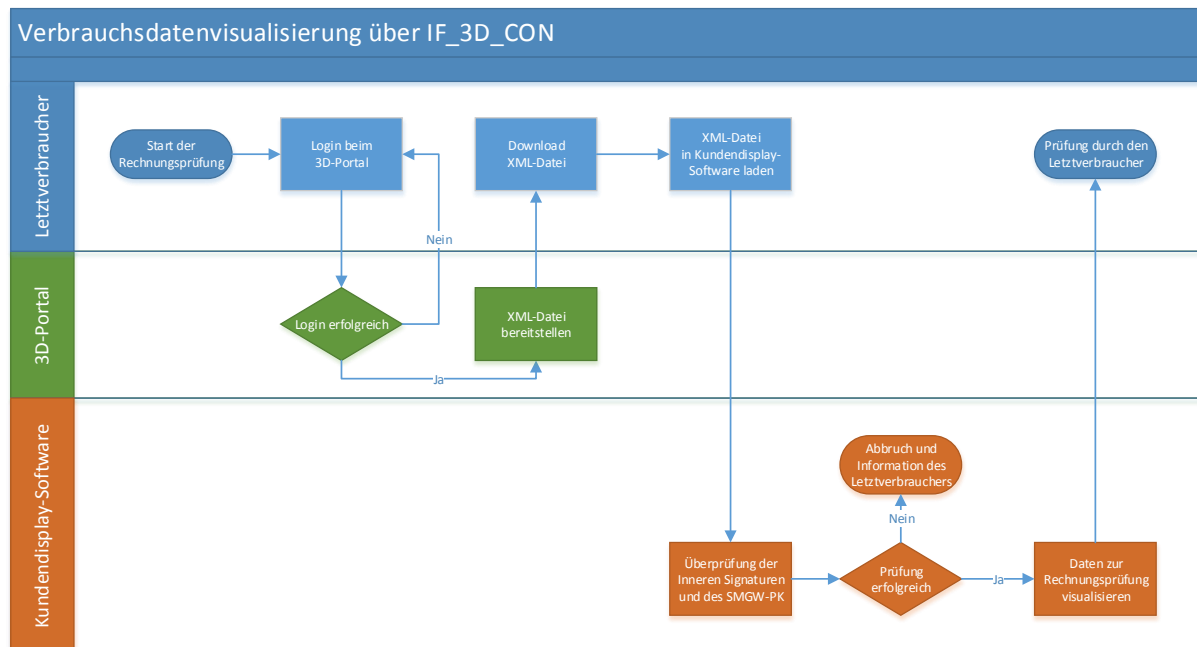


Abbildung 18: Prozess zur Verbrauchsdatenvisualisierung

Für die Verbrauchsdatenvisualisierung sollte jeden Monat eine Datei mit den abgeleiteten Messwertlisten für den entsprechenden Monat bereitgestellt werden. Die Datei sollte mindestens die Basisdaten enthalten.

Im Falle von TAF 7 sollte ein Regelwerk und ein entsprechendes Auswerteprofil zu Beginn der Lieferbeziehung bereitgestellt werden.

8.3 Regelungen zur Dateibereitstellung

8.3.1 Namensschema für die Dateibereitstellung

Für Abrechnungsdaten: Abrechnung_YYYYMMDD_YYMMDD.xml

YYYY ist mit der entsprechenden vierstelligen Jahreszahl, MM mit der zweistelligen Nummer des Monats und DD mit der Nummer des entsprechenden Tages zu füllen.

Eine Datei mit Abrechnungsdaten muss immer alle entsprechenden Daten eines Abrechnungszeitraums enthalten. Der enthaltene Abrechnungszeitraum wird im Dateinamen über die Felder DDMMYYYY gekennzeichnet. Das erste Feld wird mit entsprechenden Datumswerten Beginns des

Abrechnungszeitraum gefüllt, dass zweite Feld mit den entsprechende Datumswerten des Abrechnungsendes.

Für Energiedaten: Energiedaten_DDMMYYYY_DDMMYYYY.xml

YYYY ist mit der entsprechenden vierstelligen Jahreszahl, MM mit der zweistelligen Nummer des Monats und DD mit der Nummer des entsprechenden Tages zu füllen.

Die Energiedaten enthalten mindestens die entsprechenden Messwertlisten des über die Felder DDMMYYYY als Beginn und Enddatum gekennzeichneten Zeitraums.

Für Abrechnungsdaten kann der Dateiumfang sehr groß werden, eine entsprechend komprimierte Bereitstellung der Dateien in einem zip-Format ist möglich und sollte aus Sicht des Namensschemas nur zur Änderung der Dateiendung führen. Die im Archiv beinhaltete Datei sollte wieder den oben genannten Vorschlägen entsprechen.

8.3.2 Automatisierung der Dateibereitstellung

Im Sinne dieser Anwendungsregel muss mindestens die manuelle Downloadmöglichkeit für den Nutzer bereitgestellt werden.

Weitere technische Lösungen wie der direkte Download aus einem Programm heraus vom DatenDisplayDienst sind möglich. Technische Spezifikationen obliegen hier aber den entsprechenden Anbietern.

9 Literaturverweise

| | |
|------------------|--|
| [BSI TR-03109-1] | Bundesamt für Sicherheit in der Informationstechnik: Technische Richtlinie BSI TR-03109-1 "Anforderungen an die Interoperabilität der technischen Kommunikationseinheit eines intelligenten Messsystems" Version 1.0, BSI. |
| [FNN 3D-EMT] | FNN-Lastenheft Daten und Prozesse für den Daten-Displaydienst (3D), Entwurfsversion 0.9 vom 04.12.2015, VDE FNN. |
| [FNN COSEM] | Smart Meter Gateway, Teil 2: Datenmodell und Services nach TR 03109, Version 1.1 vom 28.04.2016, VDE FNN. |
| [FNN SMGW] | FNN-Lastenheft Smart-Meter-Gateway, Funktionale Anforderungen, Entwurfsversion 1.1 vom 08.07.2016 . |
| [NAESB REQ.21] | North American Standards Board: REQ.21 - "Energy Services Provider Interface", https://www.naesb.org/ESPI_Standards.asp |
| [OBIS] | IEC 62056-6-1: COSEM Object Identification System (OBIS) |
| [PTB-A50.8] | Pysikalisch-Technische Bundesanstalt: PTB-Anforderungen Smart Meter Gateway, Dezember 2014, PTB. |

10 Anhang

10.1 XML-Schema

if_3d_con.xsd