

PKM Tarfrechner des Fraunhofer IVI

Nutzung der Webserviceschnittstelle PPE v7.1 durch Client-Systeme im Rahmen des Systems „CiCo-BW“

VERTRAULICH

Dateiname:	PKM-Tarfrechner_Nutzungsdokument_INOUT-BW_v1.5.docx
Erstellt am:	20.10.2021
Zuletzt geändert am:	20.03.2023
Version:	1.5
Ersteller:	Fraunhofer IVI Dresden

Autoren

Rostock, Sebastian / Fraunhofer IVI Dresden

Gründel, Torsten / Fraunhofer IVI Dresden

Irmer, Stefan / Fraunhofer IVI Dresden

Versionen

Version	Bearbeiter	Datum	Bemerkung
0.9	Rostock, Gründel	20.10.2021	Erstmalige Bereitstellung für das System „CiCo-BW“
1.0	Gründel	29.11.2021	Redaktionelle Überarbeitungen; Abschnitt Prüfergebnis erweitert
1.1	Irmer	13.07.2022	Abschnitt Ticketdaten: Vertriebsdatenfelder gemäß der Vorgaben von CiCo-Bw komplettiert; marginale zur Vorversion kompatible Konkretisierungen
1.2	Irmer	15.07.2022	Kleinere redaktionelle Ergänzungen.
1.3	Irmer	30.08.2022	Abschnitt Verbindung: Bezüge zu GTFS-Daten sowie Ausprägungen von Verkehrsmitteln und Linien ergänzt
1.4	Irmer	12.09.2022	Abschnitt Ticketdaten: Vertriebsdatenfelder um PVOrgID ergänzt, Inhalte in Abschnitt INOUTBW_ANZEIGEDATEN ergänzt
1.5	Irmer	20.03.2022	Ergänzung der Erweiterung „INOUTBW_VERKAUFSMELDUNG-BWTARIF“ im Abschnitt 3.2 (Response) und 3.3.7

Inhalt

1	Einleitung	5
2	Allgemeine Festlegungen	6
3	Anfrage Kaufangebote nach Verbindung (INOUT-Fahrt)	8
3.1	Grundlagen	8
3.2	Datenstrukturen	9
3.3	Wertebereiche	11
3.3.1	Anfrage: allgemeine Daten	11
3.3.2	Verbindung (INOUT-Fahrt)	11
3.3.3	Berechtigung	13
3.3.4	Nutzungszustand	14
3.3.5	Reisender	14
3.3.6	Ticketdaten: Anzeige- und Bezahlraten	14
3.3.7	Ticketdaten: Vertriebsdatenfelder	15
3.3.8	Ticketdaten: Eingabedaten und Erweiterungen	18
3.3.9	Angebotsdaten	18
3.3.10	Prüfergebnis	18
4	Clientspezifische Erweiterungen	19
4.1	INOUTBW_ANZEIGEDATEN	19

1 Einleitung

Diese Dokumentation beschreibt die anwendungsbezogen vorgesehene Nutzung der Schnittstelle des PKM-Tarifrechners für den Anwendungsfall „CiCo-BW“; insbesondere die hierfür relevanten Kennungen, Datenstrukturen und Verarbeitungsprozesse.

Im Dokument «PKM Tarifrechner Spezifikation Webserviceschnittstelle_PPEv7.1» werden die hierfür zu verwendenden Datenstrukturen beschrieben. Dieses verweist dazu auf das Dokument «Auswahlmenü Technische Spezifikation_v1.2», in welchem grundlegende Datentypen spezifiziert werden. Der aktuell relevante Teil aus diesem Dokument (Beschreibung des ANY-Typs und weiterer grundlegender Datentypen) wurde für einen einfacheren Zugriff in den nachfolgenden Abschnitt kopiert. Ein Auswahlmenü an sich ist für den hier betrachteten Anwendungsfall aktuell noch nicht relevant, aber für eine zukünftige Verwendung zu berücksichtigen.

Der PKM-Tarifrechner an sich wird von der NVBW bereitgestellt. Testzugänge werden nach Abstimmung durch Fraunhofer IVI bereitgestellt.

Wichtiger Hinweis: Zur Nutzung der Schnittstelle des PKM-Tarifrechners für den Anwendungsfall „CiCo-BW“ können weitergehende fachlich-technische Abstimmungen, z.B. bzgl. der zur verwendenden Datengrundlage und der auszugebenden Informationen, zwischen den Anbietern der Client-Systeme und Fraunhofer IVI erforderlich werden. Änderungen aufgrund aktueller Projekterfordernisse sowie fachliche Weiterentwicklungen durch Fraunhofer IVI bleiben jederzeit vorbehalten. Soweit dadurch Anpassungen auf Seiten der Client-Systeme notwendig werden, sind diese nach Bekanntgabe innerhalb von drei Monaten durch die Anbieter umzusetzen.

2 Allgemeine Festlegungen

Damit der Tarifrechner die ihm gestellten Anfragen korrekt beantworten kann, muss er diese verstehen können. Dazu sind neben den Anfragearten und sonstigen konstanten Anfragemerkmalen insbesondere die Wertebereiche der in den jeweiligen Domänen gültigen Nummern abzustimmen.

Nach aktuellem Stand wird die nachfolgend vorgestellte Anfrageart mit den dort angegebenen Wertebereichen und Strukturmerkmalen unterstützt.

Weitere Anfragearten sind derzeit nicht vorgesehen; könnten aber nach Abstimmung für folgende Anwendungsfälle ergänzt werden:

- Stornierungen bzw. Neuberechnungen, bei denen von der üblichen Preis- und Produktermittlung abgewichen werden soll
- Ermittlung von Preisauskünften vor einer Fahrt anhand der Fahrplanauskunft efa-bw

Alle Anzeigetexte sind in der Schnittstelle mit dem Datentyp TEXT gekennzeichnet und können die HTML-Auszeichnungen `<wbr>`, `<br>`, `<em>`, `<strong>`, `<ul>`, `<ol>` und `<li>` enthalten. Zudem sind auch Sonderzeichen wie `&`, ` ` und `­` zulässig.

Textlängenbegrenzungen für die vom Tarifrechner bereitgestellten Textfelder sind nicht vereinbart.

Alle als **ERWEITERUNG** gekennzeichneten Datenstrukturen sind obligatorisch. Die Werte in einer **ERWEITERUNG** unterliegen der ANY-Kodierung. Dieser Datentyp beschreibt einen beliebigen Wert als typisierte Zeichenkette (`string`), als Auflistung (`array`) solcher Werte oder als Abbildung (`object`) auf solche Werte. Dabei gilt, dass ein primitiver Werte grundsätzlich in seine XML-Textdarstellung mit vorangestellter Typkennung überführt wird, mit besonderer Behandlung benannter Konstanten. Es werden die folgenden Typkennungen eingesetzt:

- U für die Konstante null, d.h. "U"
- T für die Konstante true, d.h. "T"
- F für die Konstante false, d.h. "F"
- I für Ganzzahlen, bspw. "I132", "I-456"
- S für Zeichenketten, bspw. "SDas ist ein Text!"
- B für Bytefolgen in Hexadezimaldarstellung, bspw. "B1F08154711BAADF00D"
- D für Datum, Uhrzeit und Zeitpunkt, bspw. "D09:30:00", "D2020-01-06"
- P für Zeitspannen bspw. "PP1M", "P-PT1H"

Weitere grundlegende Datentypen sind:

BOOLEAN: Dieser Datentyp beschreibt einen Wahrheitswert in JSON (vgl. XML boolean).

VERTRAULICH

DATE: Dieser Datentyp beschreibt einen gregorianischen Kalendertag als `STRING` (vgl. XML `date`).

DATETIME: Dieser Datentyp beschreibt einen Zeitpunkt als `STRING` (vgl. `DATE`, `TIME`, XML `dateTime`).

INTEGER: Dieser Datentyp beschreibt eine beliebige Ganzzahl in JSON (vgl. XML `long`).

NAME: Dieser Datentyp beschreibt die informative Bezeichnung eines Datensatzes als `STRING`, welche bei der Datenpflege sowie bei Abstimmungen unter den Beteiligten nützlich sein kann. Diese Bezeichnung unterliegt grundsätzlich keiner Abstimmung und ist damit im Client grundsätzlich zu ignorieren.

REFERENCE: Dieser Datentyp beschreibt einen Verweis auf einen Datensatz als `INTEGER`, welcher die 1-basierte Position des Datensatzes in der ihn verwaltenden Auflistung angibt.

STRING: Dieser Datentyp beschreibt eine beliebige Zeichenkette in JSON (vgl. XML `string`).

TEXT: Dieser Datentyp beschreibt einen Anzeige- oder Hinweistext als `STRING`, welcher zur Beschriftung eines Steuerelements eingesetzt werden kann. Ob dieser bspw. Zeilenumbrüche oder HTML-Auszeichnungen enthalten darf und welche Längenbeschränkung für ihn gelten, ist unter den Beteiligten abzustimmen.

TIME: Dieser Datentyp beschreibt eine sekundengenaue Tageszeit als `STRING` (vgl. XML `time`).

3 Anfrage Kaufangebote nach Verbindung (INOUT-Fahrt)

3.1 Grundlagen

Diese Anfrage verlangt mindestens eine Verbindung mit mindestens einem Reisenden. Eine Verbindung in der Anfrage entspricht jeweils genau einer vom Clientsystem erfassten INOUT-Fahrt. Eine INOUT-Fahrt ist immer auf mindestens einen Reisenden bezogen, beginnt mit der ersten Haltestelle nach einem Check-In, endet mit der letzten Haltestelle vor dem nächstfolgenden Check-Out und kann Umstiege enthalten. Der Client stellt in jeder Anfrage alle zur kompletten Preisberechnung erforderlichen Daten bereit. Es erfolgt keine Berechnung „vorläufiger“ Preise.

Fußwege werden grundsätzlich nicht an den Tarifrechner übermittelt; es sei denn, dies wird aufgrund tariflicher Anforderungen zwischen aufeinanderfolgenden Fahrten erforderlich.

Vom Client werden nur Haltestellen aus dem räumlichen Anwendungsgebiet von CiCo-BW an den Tarifrechner übergeben, d.h. alle Fahrtanteile außerhalb dieses Anwendungsgebiets werden vor der Übergabe an den Tarifrechner vom Client entfernt. Dies gilt auch für durchgeführte Haltestellen außerhalb dieses Anwendungsgebiets; in diesem Falle werden zwei getrennte INOUT-Fahrten übergeben und bepreist.

Die für eine konkrete Preisberechnung relevanten INOUT-Fahrten beinhalten in der Regel auch bereits bepreiste INOUT-Fahrten, z.B. für die Berücksichtigung fahrtübergreifender Preisdeckel. Das Clientsystem erkennt diese zusätzlich an den Tarifrechner zu übermittelnden bepreisten INOUT-Fahrten an deren Preisberücksichtigungsfristen. Eine bepreiste INOUT-Fahrt erhielt diese Frist zuvor über die Erweiterung INOUTBW_PREISBERUECKSICHTIGUNGSFRIST der dazu erzeugten Ticketdaten. Eine bepreiste INOUT-Fahrt ist an den Tarifrechner zu übergeben, wenn ihre Preisberücksichtigungsfrist erst nach der Abfahrt der ersten unbepreisten INOUT-Fahrt endet. Ein alternatives Verfahren muss ggf. zwischen Fraunhofer IVI, NVBW und allen Clientsystemen einvernehmlich abgestimmt werden.

Zudem sind die Berechtigungsdatensätze zu übermitteln, die auf die übermittelten bereits bepreisten INOUT-Fahrten verweisen. Diese sind durch Kopie der folgenden Datenfelder aus den TICKETDATEN der jeweiligen vorherigen Käufe abzuleiten:

- TICKETDATEN.eingabedaten
- TICKETDATEN.fahrtenbezug
- TICKETDATEN.reisendenbezug
- TICKETDATEN.erweiterungsliste

Ebenso müssen in die Anfrage auch diejenigen Reisenden aus vorherigen Anfragen kopiert werden, auf die diese Berechtigungsdatensätze jeweils zeigten. Die Liste der Reisenden in einer Anfrage beinhaltet damit stets die aktuell Reisenden sowie alle Reisenden aus den zusätzlich übergebenen bereits bepreisten INOUT-Fahrten.

Die Preis- und Produktermittlung erfolgt immer für komplette INOUT-Fahrten; niemals nur für einzelne Abschnitte einer INOUT-Fahrt. Je neu angefragter INOUT-Fahrt werden vom Tarifrechner ein oder mehrere TICKETDATEN generiert.

In bestimmten Fällen muss eine angefragte INOUT-Fahrt tarifrechnerseitig nach bestimmten Regeln aufgebrochen und in neue tarifliche INOUT-Fahrten aufgespalten werden. Auch eine solche tarifliche Aufspaltung der übermittelten INOUT-Fahrt führt zur Rückgabe mehrerer `TICKETDATEN`, die über entsprechende Fahrtenbezüge den Abschnitten der übermittelten INOUT-Fahrten zugeordnet sind (vgl. auch Abschnitt 3.3.2).

Zu den übermittelten INOUT-Fahrten, die in einer vorherigen Anfrage an den Tarifrechner schon einmal bepreist wurden, werden in der aktuellen Anfrage keine `TICKETDATEN` mehr erzeugt. Der Tarifrechner erkennt derartige INOUT-Fahrten an den dazu zu übermittelnden `BERECHTIGUNG` Datenstrukturen, wie oben beschrieben.

Alle ggf. zu korrigierenden Fahrten werden durch das Clientsystem storniert (sofern bereits gegenüber dem Kunden abgerechnet) und zur Neuberechnung an den Tarifrechner übergeben.

Dieser kann nicht unterscheiden, ob es sich um eine reguläre Preisberechnung oder um einen Korrekturlauf handelt. Sollte sich zu einem späteren Zeitpunkt herausstellen, dass der Tarifrechner gesonderte Korrekturläufe durchführen soll, müsste ggf. eine gesonderte Anfrageart ergänzt werden, innerhalb derer eine von der regulären Preisberechnung abweichende Behandlung inkl. der Kennzeichnung preiskorrigierter Fahrten erfolgen kann.

Der Tarifrechner wird Fahrten für einen Zeitraum berechnen können, der maximal 45 Tage ab dem Anfragezeitpunkt in die Vergangenheit zurückreicht.

3.2 Datenstrukturen

REQUEST

```

anfrageliste: ANFRAGE[N]
  1: ANFRAGE
    anfrageart = "INOUTBW_CLIENTXY_KAUFANGEBOTE_NACH_VERBINDUNG-1.0"
    zeitpunkt: DATETIME
    herausgeber = 0
    sprache = 58
    kontext = 1
    verbindungsliste: VERBINDUNG[1...*]
      1: VERBINDUNG
        fahrtenliste: FAHRT[1...*]
          1: FAHRT
            halteliste: HALT[2...*]
              1: HALT
                ankunft: DATETIME
                abfahrt: DATETIME
                haltestelle: ELEMENT
                  nr: STRING (DHID)
                  name: NAME
                istDurchfahren: BOOLEAN
              ...: HALT
            verkehrsmittel: ELEMENT
              nr: STRING
              name: NAME
            liniendatenliste: LINIENDATEN[1...*]
              1: LINIENDATEN
                linie: ELEMENT
                  nr: STRING
                  name: NAME
                betreiber: ELEMENT
                  nr: STRING
                  name: NAME
              ...: LINIENDATEN
            ...: FAHRT
          erweiterungsliste: ERWEITERUNG[4...*]
            1: ERWEITERUNG
              nr = "INOUTBW_ISTSYSTEMCO"
              wert: ANY<BOOLEAN>

```

```

2: ERWEITERUNG
  nr = "INOUTBW_ISTERSTEKLASSE"
  wert: ANY<BOOLEAN>
3: ERWEITERUNG
  nr = "INOUTBW_ISTAUSGESCHLOSSEN"
  wert: ANY<BOOLEAN>
...: ERWEITERUNG
  nr = "INOUTBW_REISENDENBEZUG"
  wert: ANY<REFERENCE>
...: VERBINDUNG
berechtigungsliste?: BERECHTIGUNG[1...*]
1: BERECHTIGUNG (Datenfelder kopiert von TICKETDATEN aus ANTWORT)
  eingabedaten: EINGABEDATEN
  fahrtenbezug: ABSCHNITT[1]
  reisendenbezug: REFERENCE[1]
  erweiterungsliste: ERWEITERUNG[1...*]
    1: ERWEITERUNG
      nr = "INOUTBW_PREISBERUECKSICHTIGUNGSFRIST"
      wert: ANY<DATETIME>
    ...: ERWEITERUNG
...: BERECHTIGUNG
nutzungszustandsliste: NUTZUNGSZUSTAND[]
reisendenliste: REISENDER[1...*]
1: REISENDER
  typ: ELEMENT
  nr: REISENDERTYP
  name: NAME
  rabattmedienliste?: ELEMENT[]
    1: ELEMENT
      nr: RABATTMEDIENTYP
      name: NAME
    ...: ELEMENT
  erweiterungsliste: ERWEITERUNG[1]
    1: ERWEITERUNG
      nr = "INOUTBW_ISTHAUPTNUTZER"
      wert: ANY<BOOLEAN>
...: REISENDER
...: ANFRAGE

```

RESPONSE

```

antwortliste: ANTWORT[N]
1: ANTWORT
  ticketdatenliste?: TICKETDATEN[1...*]
    1: TICKETDATEN
      anzeigetext: STRING
      tarifyinfotext?: STRING
      tarifyinfoseite?: STRING
      hinweistext?: STRING
      bezahldatenliste: BEZAHLDATEN[1...*]
        1: BEZAHLDATEN
          hinweistext?: STRING
          betrag: INTEGER
          steuer = 700
          rabatt?: INTEGER
        ...: BEZAHLDATEN
      ausgabedaten: AUSGABEDATEN
      erweiterungsliste: ERWEITERUNG[1...*]
        1: ERWEITERUNG
          nr = "INOUTBW_VERTRIEBSDATENFELDER"
          wert: ANY<VERTRIEBSDATENFELDER>
        2: ERWEITERUNG
          nr = "INOUTBW_VERKAUFSMELDUNG-BWTARIF"
          wert: ANY<VERKAUFSMELDUNG-BWTARIF>
        ...: ERWEITERUNG
      eingabedaten: EINGABEDATEN
      fahrtenbezug: ABSCHNITT[1]
      reisendenbezug: REFERENCE[1]
      berechtigungsbezug?: REFERENCE[1...*]
      erweiterungsliste: ERWEITERUNG[1...*]
        1: ERWEITERUNG
          nr = "INOUTBW_PREISBERUECKSICHTIGUNGSFRIST"
          wert: ANY<DATETIME>
        2: ERWEITERUNG
          nr = "INOUTBW_ANZEIGEDATEN"
          wert: ANY<ANZEIGEDATEN>
        ...: ERWEITERUNG

```

```

...: TICKETDATEN
angebotsdatenliste?: ANGEBOTSDATEN[1]
1: ANGEBOTSDATEN
  anzeigetext?: STRING
  hinweistext?: STRING
  ticketdatenbezug: REFERENCE[1...*]
  nutzungszustandsbezug?: REFERENCE[1]
pruefergebnisliste?: PRUEFERGEBNIS[1...*]
1: PRUEFERGEBNIS
  nr: STRING
  anzeigetext?: STRING
  entwicklerinfo?: STRING
  fahrtenbezug?: ABSCHNITT[1...*]
  reisendenbezug?: REFERENCE[1...*]
  berechtigungsbezug?: REFERENCE[1...*]
  nutzungszustandsbezug?: REFERENCE[1]
...: PRUEFERGEBNIS
nutzungszustandsliste?: NUTZUNGSZUSTAND[1]
...: ANTWORT

```

3.3 Wertebereiche

3.3.1 Anfrage: allgemeine Daten

Den Empfehlungen zur Nutzung der Webserviceschnittstelle folgend, wird als Nummer des Herausgebers des einzusetzenden RN-Tarifmoduls die Organisationsnummer des primären Kundenvertragspartners (KVP) gemäß VDV KA, bzw. bei unbestimmtem KVP die Nummer 0 eingesetzt (siehe `ANFRAGE.herausgeber`).

Als unterstützte Sprache wird derzeit ausschließlich Deutsch mit der Nummer 58 angeboten (siehe `ANFRAGE.sprache`). Eine zukünftige Erweiterung auf andere Sprachen ist möglich.

Da es derzeit keine Notwendigkeit zur Unterscheidung des Einsatzorts des RN-Tarifmoduls gibt, wird für den Modulkontext die Konstante 1 eingesetzt (siehe `ANFRAGE.kontext`).

Die Anfrageart ist gemäß Abschnitt 3.2 zu bestücken (siehe `ANFRAGE.anfrageart`).
TODO: Anfrageart clientspezifisch festlegen.

3.3.2 Verbindung (INOUT-Fahrt)

Die Datenstruktur `VERBINDUNG` wird im `REQUEST` mit genau einer `INOUT-Fahrt` bestückt. Umstiege während einer `INOUT-Fahrt` sind jeweils durch Ausprägungen von `FAHRT` abzubilden. Die Ausprägungen `CICO` und `CIBO` sind durch den Tarifrheber völlig gleich zu behandeln. Wurden keine Fahrten zwischen einzelnen `CI-` bzw. `CO-Vorgängen` erfasst, so werden diese `CI-` bzw. `CO-Vorgänge` nicht übermittelt. `CI-` bzw. `CO-Vorgänge` sowie Fahrten dürfen sich nicht überlappen.

Tarifrheberseitig kann in bestimmten, tarifbezogenen Fällen eine Aufspaltung von übermittelten `INOUT-Fahrten` erfolgen. Ein Ticketdatensatz und ein Prüfergebnis in der `RESPONSE` beziehen sich jeweils auf genau eine, komplette `INOUT-Fahrt`. Bei einer tariflichen Aufspaltung durch den Tarifrheber entstehen neue tarifliche `INOUT-Fahrten`, so dass demzufolge auch mehrere Ticketdatensätze bzw. Prüfergebnisse erzeugt werden.

VERTRAULICH

Diese unterschiedlichen Ticketdatensätze bzw. Prüfergebnisse werden den jeweiligen Verbindungsabschnitten der angefragten Verbindung zugeordnet. Diese Zuordnung erfolgt über die Fahrtenbezüge `TICKETDATEN.fahrtenbezug` und `PRUEFERGEBNIS.fahrtenbezug`. Jeder Verbindungsabschnitt repräsentiert eine neue tarifliche INOUT-Fahrt, sodass der Beginn eines solchen Abschnitts als IN-Vorgang und das Ende eines solchen Abschnitts als OUT-Vorgang zu interpretieren ist.

Falls mindestens einer dieser Abschnitte aufgrund der `INOUTBW_PREISBERUECKSICHTIGUNGSFRIST` bei einer späteren Anfrage erneut an den Tarifrchner übergeben werden muss, so sind alle aus einer tariflichen Aufspaltung entstandenen Abschnitte, d.h. die gesamte betreffende Verbindung, an den Tarifrchner zu übergeben. Im REQUEST sind dann die Fahrtenbezüge gemäß der Zuordnung der jeweiligen Berechtigungen zu den Verbindungsabschnitten über `BERECHTIGUNG.fahrtenbezug` herzustellen. Damit wird eine zuvor schon einmal vorgenommene tarifliche Aufspaltung in späteren Anfragen ersichtlich, ohne dass die Verbindung selbst durch den Client verändert werden muss.

Die Nummern der in einer VERBINDUNG aufgeführten Haltestellen werden als haltestellengenaue, aber keinesfalls steiggenaue DHID angegeben (derzeit: Nr = `NVBW_HST_DHID` aus `stops.txt`, Name = `stop_name` aus `stops.txt`). In der VERBINDUNG sind generell alle Haltestellen aufgeführt, die bei den einzelnen Fahrten passiert werden, also auch die durchfahrenen (`HALT.istDurchfahren`). Dazu sind an allen Haltestellen die Ankunfts- bzw. Abfahrtszeit bzw. der Durchfahrtszeitpunkt anzugeben (`HALT.ankunft` und `HALT.abfahrt`). Es werden ausschließlich die realen Fahrtzeiten bzw. die Zeiten aus dem Fahrplan übermittelt, nicht jedoch die Zeiten von CI- bzw. CO-Vorgängen.

Die Verkehrsmittel werden gemäß dem durch NVBW bereitgestellten GTFS-Datensatz mit folgenden Ausprägungen angegeben (derzeit: Nr = `NVBW_Verkehrsmittel` aus `routes.txt`).

Nr	Name
„0“	„Zug“ (TODO: zu ersetzen durch 13 bis 16 - Klärung NVBW)
„1“	„S-Bahn“
„2“	„U-Bahn“
„3“	„Stadtbahn“
„4“	„Straßen-/Trambahn“
„5“	„Stadtbus“
„6“	„Regionalbus“
„7“	„Schnellbus“
„8“	„Seil-/Zahnradbahn“
„9“	„Schiff“
„10“	„AST/Rufbus“
„11“	„Sonstige“
„12“	„Flugzeug“
„13“	„Zug (Nahverkehr)“
„14“	„Zug (Fernverkehr)“
„15“	„Zug (Fernverkehr mit Zuschlag)“
„16“	„Zug (Fernverkehr mit speziellem Fahrpreis)“
„17“	„Schienenersatzverkehr (SEV)“
„18“	„Zug Shuttle“
„19“	„Bürgerbus“

VERTRAULICH

Die Linien werden gemäß dem durch NVBW bereitgestellten GTFS-Datensatz mit folgenden (beispielhaften) Ausprägungen angegeben (derzeit: Nr = NVBW_DLID aus routes.txt, Name = route_short_name aus routes.txt).

Nr	Name
„de:kvv:22070E:“	„S7“
„de:vvs:vvs-34-97:“	„N97“
„de:h3nv:30091N:“	„91“
„de:vrn:49976_:“	„976“
„de:nvbw:94019_:“	„19“
...	

Die Linienbetreiber werden gemäß dem durch NVBW bereitgestellten GTFS-Datensatz mit folgenden (beispielhaften) Ausprägungen angegeben (derzeit nicht benötigt).

Nr	Name

Der Umgang mit und die Kodierung von Schienenersatzverkehr (SEV) ist noch abzustimmen (TODO).

Die Erweiterung „INOUTBW_ISTSYSTEMCO“ dient der Kennzeichnung des CO-Vorgangs als systemseitig (true) bzw. kundenseitig (false).

Die Erweiterung „INOUTBW_ISTERSTELASSE“ kennzeichnet die gesamte INOUT-Fahrt als erste Klasse (true) bzw. zweite Klasse (false).

Die Erweiterung „INOUTBW_ISTAUSGESCHLOSSEN“ kennzeichnet die gesamte INOUT-Fahrt als ausgeschlossen (true) von der Preis- und Produktermittlung, sodass dafür insbesondere keine Ticketdaten ermittelt werden. Eine ausgeschlossene INOUT-Fahrt kann jedoch bei der Preis- und Produktermittlung anderer INOUT-Fahrten eine Rolle spielen. Insbesondere im Rahmen eines Stornierungsvorgangs gelöschte oder ersetzte INOUT-Fahrten sind als ausgeschlossen zu kennzeichnen.

Für jeden Reisenden, der eine bestimmte INOUT-Fahrt absolviert, muss die zugehörige Verbindung eine Erweiterung „INOUTBW_REISENDENBEZUG“ besitzen, welche auf den jeweiligen Reisenden (Listenposition) verweist.

3.3.3 Berechtigung

Die Bestückung der Datenstruktur `BERECHTIGUNG` in der Anfrage ist in Abschnitt 3.1 beschrieben.

Der Fahrtenbezug und der Reisendenbezug der Berechtigungsdaten sind durch den Client so anzupassen, dass sie zu der jeweils aktuellen Zusammenstellung von Verbindungen und Reisenden passen.

Der `TICKETDATEN.berechtigungsbezug` aus der `RESPONSE` verweist auf alle Instanzen von `BERECHTIGUNG` im `REQUEST`, welche die Ermittlung der jeweiligen `TICKETDATEN` tariflich beeinflussten.

3.3.4 Nutzungszustand

Der Nutzungszustand ermöglicht bei Bedarf die Verwaltung nutzerindividueller Merkmale zur Tarifierung. Er wird vom Tarifrechner erzeugt, muss clientseitig gespeichert und in der nachfolgenden Anfrage wieder an den Tarifrechner mitgegeben werden.

Hinweis: Die Funktionalität „Nutzungszustand“ ist für eine zukünftige Nutzung vorgesehen.

3.3.5 Reisender

Als REISENDERTYP sind zunächst folgende Ausprägungen zulässig:

"1" - Erwachsener/Vollzahler/Jedermann"
"2" - Kind

Die Mitnahme von weiteren Personen sowie die Mitnahme von Sachen und Tieren sind für eine zukünftige Verwendung vorgesehen, werden aber derzeit planmäßig noch nicht vom Tarifrechner unterstützt.

Je Reisendem sind folgende Rabattmedien als RABATTMEDIENTYP zulässig:

Nr	Bedeutung
„51“	BahnCard 25 2. Klasse
„52“	BahnCard 50 2. Klasse
„53“	BahnCard 25 1. Klasse
„54“	BahnCard 50 1. Klasse

Aus der Liste der zulässigen Bahncards ist pro Reisendem maximal eine Ausprägung zulässig. Ein Wechsel innerhalb einer INOUT-Fahrt ist nicht zulässig.

Für jede CICO-Fahrt gibt es immer genau einen erwachsenen Hauptnutzer (Accountinhaber). Ein Reisender wird in der Anfrage über die Erweiterung „INOUTBW_ISTHAUPTNUTZER“ als Hauptnutzer (true) bzw. – sobald zukünftig zulässig – als Mitnahme (false) gekennzeichnet.

Hauptnutzer werden nur in der Ausprägung „1“ (Erwachsener/Vollzahler/Jedermann) oder „2“ (Kind) vom Tarifrechner verarbeitet.

Der Hauptnutzer sowie – sobald zukünftig zulässig – alle mitzunehmenden Personen, Sachen, Tiere werden als Reisende abgebildet. Eine konkrete Zusammenstellung der Reisenden gilt pro INOUT-Fahrt.

Obergrenzen für derartige Zubuchungsoptionen werden sobald zukünftig zulässig vereinbart.

3.3.6 Ticketdaten: Anzeige- und Bezahlzeiten

Der Tarifrechner wird keinerlei personenbezogene Daten verarbeiten. Bei Bedarf können jedoch nach gesonderter Abstimmung z.B. Platzhalter für die Anzeige von personalisierten Informationen in Anzeigetexte oder andere Ausgabedaten integriert werden.

Die Anzeigetexte in der App und auf einem Kaufbeleg/Rechnung werden wie folgt mit Attributen der Standard-Webservice-Schnittstelle bestückt. Statische Texte, Texte zur Start- und Zielhaltestelle sowie Abfahrts- bzw. Ankunftszeiten sind clientseitig zu bestücken.

VERTRAULICH

Sollte sich die Notwendigkeit für zusätzliche clientspezifische (tarifliche) Anzeigeelemente ergeben, dann wären diese über eine separate Erweiterungsstruktur zu den Ausgabedaten der Ticketdaten umzusetzen und in Kapitel 4 zu beschreiben.

TODO Festlegung der Anzeigetexte in Abstimmung mit dem Client-System

Der Produktname ergibt sich aus `TICKETDATEN.anzeigetext`; sowohl für die Darstellung in der App als auch für den Kaufbeleg/Rechnung.

Der Preis zu einem Ticketdatensatz ergibt sich aus `BEZAHLDATEN.betrag`. Die Bezahlungen werden generell mit Preisen in Eurocent ausgegeben.

Eine Rabattierung ist derzeit im Anwendungsfall „CiCo-BW“ nicht relevant. Sollte sich zukünftig bei der Preisberechnung eine Rabattierung ergeben, so wird über `BEZAHLDATEN.rabatt` die gesamte Ersparnis ausgewiesen. TODO: Der Umfang dieser Ersparnis ist noch abzustimmen (z.B. ob auch eine Ersparnis aufgrund von BahnCard-Nutzung eingerechnet wird). Dem Nutzer gegenüber kann dazu über `BEZAHLDATEN.hinweistext` ein gesonderter Hinweis zu der vorgenommenen Rabattierung angezeigt werden. TODO: Klärung, ob so gewünscht.

Aus `BEZAHLDATEN.betrag` und `BEZAHLDATEN.rabatt` werden clientseitig die preisbezogenen Anzeigetexte für die App als auch für den Kaufbeleg/Rechnung erzeugt. Der Gesamtpreis bzw. Gesamtrabatt zu einer INOUT-Fahrt ist vom Client durch Addition der Einzelbeträge aus den jeweiligen `BEZAHLDATEN.betrag` bzw. `BEZAHLDATEN.rabatt` zu berechnen. Kumulierte Preise und Ersparnisse in Bezug auf einen Historienzeitraum sind ebenfalls clientseitig zu berechnen.

`BEZAHLDATEN.steuer` weist den einheitlichen und entfernungsunabhängigen Mehrwertsteuersatz von 700 aus (in 1/100 Prozent).

3.3.7 Ticketdaten: Vertriebsdatenfelder

Daten zur Erzeugung von Vertriebsdatenmeldungen werden als Erweiterung „INOUTBW_VERTRIEBSDATENFELDER“ der Ausgabedaten der Ticketdaten bereitgestellt. Der Wert „VERTRIEBSDATENFELDER“ setzt sich aus den folgenden Feldern zusammen:

`PvOrgID: INTEGER`

Dieses Datenfeld gibt die ID des Produktverantwortlichen an.

`TicketTypeID: STRING`

Dieses Datenfeld gibt die ID des Tickettyps, z.B. 101 oder Fehlercode für unbekannte Restposten (z.B. aus Tariferverfehlerdifferenzen) an.

`TicketTypDesc: STRING`

Dieses Datenfeld gibt die Bezeichnung des Tickettyps, z.B.: Einzelticket Erwachsener oder Erstattung an.

`PreisStufe: STRING?`

Dieses optionale Datenfeld gibt die Anzahl der Zonen an.

`PreisEinzel: INTEGER`

Dieses Datenfeld gibt den Preis für ein Ticket in Cent ohne Trennzeichen an.

`StartAbrechnung: STRING?`

Dieses optionale Datenfeld gibt den Start des abgerechneten Tarifprodukts an.

`ZielAbrechnung: STRING?`

Dieses optionale Datenfeld gibt das Ziel des abgerechneten Tarifprodukts an.

RaeumlicheGueltigkeit: STRING?

Dieses optionale Datenfeld gibt die räumliche Gültigkeit des abgerechneten Tarifprodukts an.

BeginnGueltigkeit: DATETIME

Dieses Datenfeld gibt den Gültigkeitsbeginn des Tickets (Date-Time) oder Erstattungszeitpunkt an.

EndeGueltigkeit: DATETIME

Dieses Datenfeld gibt das Gültigkeitsende des Tickets (Date-Time) oder Erstattungszeitpunkt an.

TarifVersion: STRING

Dieses Datenfeld gibt die aktuelle Version des Produktmoduls des PV an.

Daten zur Erzeugung von Verkaufsmeldungen des bwtarif werden als Erweiterung „IN-OUTBW_VERKAUFSMELDUNG-BWTARIF“ der Ausgabedaten der Ticketdaten bereitgestellt. Dies trifft nur für Tickets des bwtarif zu. Alle anderen Tickets besitzen diese Erweiterung nicht. Der Wert „VERKAUFSMELDUNG-BWTARIF“ setzt sich aus den folgenden Feldern zusammen:

PvOrgID: INTEGER

Dieses Datenfeld gibt die ID des Produktverantwortlichen an.

ID_REL: INTEGER

Objekt-ID der Relation

ID_FKA: INTEGER

Objekt-ID der Fahrkartenart

ID_TP_EIN: INTEGER

Objekt-ID des Einstiegstarifpunkts

ID_TP_AUS: INTEGER

Objekt-ID des Ausstiegstarifpunkt

ID_PRODUKTGRUPPE: INTEGER

Objekt-ID der Produktgruppe

ID_WEGTEXT: INTEGER

Objekt-ID der Wegealternative

Datum_Fahrt_Ab: DATE

Beginn des Gültigkeitszeitraums der Fahrkarte

Datum_Fahrt_Bis: DATE

Ende des Gültigkeitszeitraums der Fahrkarte

Tarifentfernung: INTEGER

Objekt-ID der Preisstufe

Umsatz: INTEGER

Gesamtpreis

ID_VERSION: INTEGER

Angabe der Tarifdatenversion, auf deren Grundlage der Verkauf erfolgte

Diese Struktur basiert auf den in Vertriebskonzept Anlage 1 Tarif- und Vertriebsdatenschnittstellen bwtarif - Vertriebskonzept Anlage 1 aufgeführten Feldern für Verkaufsmeldungen. Hier

VERTRAULICH

sind nur Felder mit tariflichen Informationen enthalten, alle weiteren Inhalte der Verkaufsmeldung müssen von Client ergänzt werden.

Über die Fahrt- und Reisendenbezüge an den Ticketdaten sind diese Daten den bereits in der Anfrage enthaltenen CICO-Fahrten bzw. Reisenden zugeordnet, so dass clientseitig eine entsprechende Weitergabe in die Einnahmemeldung erfolgen kann.

Pro INOUT-Fahrt und Reisenden können Ticketdaten mit nachfolgenden Informationen bereitgestellt werden. Sind Felder im Einzelfall ggf. nicht relevant oder vom Tarifrechner nicht bestückbar, wird für diese Felder jeweils null (ANY-Typkennung "U") geliefert.

Wurde eine Verbindung tariflich durch den Tarifrechner aufgespalten, so werden die nachfolgenden Datenfelder getrennt für jede der entstandenen tariflichen INOUT-Fahrten erzeugt. TODO: Klärung sobald tarifliche Aufspaltung im Anwendungsfall „CiCo-BW“ relevant wird.

Datenfeld	Datentyp	Beschreibung
TODO		
ABS	ABS [1..*]	Liste der Reiseabschnitte

Reiseabschnitt ABS:

TODO	INTEGER	gibt die laufende Nummer (Listenposition) des Teilweges (FAHRT) innerhalb der zugrunde liegenden INOUT-Fahrt (VERBINDUNG) an.
TODO	STRING	Einstiegs-Halt als DHID (zur Validierung, welcher Abschnitt zugrunde liegt)
TODO	STRING	Ausstiegs-Halt als DHID (zur Validierung, welcher Abschnitt zugrunde liegt)
TODO		

Derzeit besteht keine Anforderung, einzelne Datenfelder rückwirkend in Abhängigkeit von nachfolgenden INOUT-Fahrten anzupassen. Sollte dies zukünftig erforderlich werden, wird der Tarifrechner ein solches Datenfeld jeweils neu auch für die zurückliegenden INOUT-Fahrten berechnen, die per `TICKETDATEN.berechtigungsbezug` referenziert werden (vgl. Abschnitt 3.3.3), und das Neuberechnete Feld als Bestandteil einer komplett neu berechneten Erweiterung „INOUTBW_VERTRIEBSDATENFELDER“ bereitstellen. Diese Erweiterung kann also mehrfach bereitgestellt werden, wobei die Zuordnung zu den per `TICKETDATEN.berechtigungsbezug` referenzierten INOUT-Fahrten unter Beachtung der Reihenfolge erfolgt. Zuerst werden die Erweiterungen passend zur Anzahl und Reihenfolge der referenzierten Berechtigungen aufgeführt. Danach stehen die Erweiterungen zur aktuellen tariflichen INOUT-Fahrt mit ihrem Reisenden.

Einige der vom Clientsystem an das „CiCo-BW“ Meldesystem möglicherweise zu übergebenden Daten, wie z.B. Fahrt- und Fahrzeugdaten, Linien- und Verkehrsmitteldaten sowie Stornoinformationen, werden nicht vom Tarifrechner bestückt, sondern sind außerhalb des Tarifrechners vom Clientsystem zu ergänzen.

Ebenso müssen einige Datenfelder ggf. in das finale Zielformat umformatiert werden. Sofern als Zieldatentypen DECIMAL oder FLOAT spezifiziert wurden, liefert der Tarifrechner stattdessen ein INTEGER in der geforderten Genauigkeit z.B. als 1/100 oder 1/1000. TODO Final klären.

3.3.8 Ticketdaten: Eingabedaten und Erweiterungen

Der Tarifrechner stellt in der Response die Datenstruktur `EINGABEDATEN` bereit. Diese wird vom Client nicht ausgewertet, sondern bei nachfolgenden Anfragen wieder an den Tarifrechner übergeben (siehe Abschnitt 3.1).

Die Erweiterung „`INOUTBW_PREISBERUECKSICHTIGUNGSFRIST`“ an den Ticketdaten wird vom Client ausgewertet (siehe Abschnitt 3.1). Alle Erweiterungen werden bei nachfolgenden Anfragen wieder an den Tarifrechner übergeben (siehe Abschnitt 3.1).

Der Wert in `INOUTBW_PREISBERUECKSICHTIGUNGSFRIST` hat keine tarifliche Bedeutung; er darf vom Client nur zu dem in Abschnitt 3.1 beschriebenen Zweck benutzt werden.

3.3.9 Angebotsdaten

Die Angebotsdaten stellen den Bezug zu den Ticketdaten her.

3.3.10 Prüfergebnis

In Fehlerfällen werden tarifrechnerseitig lediglich allgemeine Prüfergebnisse bereitgestellt, welche für die Kundenanzeige nicht zwischen verschiedenen Fehlerfällen differenzieren.

Kann eine Anfrage strukturell nicht verarbeitet werden, wird `Pruefergebnis.nr = ANFRAGE_UNGUELTIG` ausgegeben.

Kann für eine INOUT-Fahrt einer Anfrage insbesondere aus tariflichen Gründen kein Angebot berechnet werden, wird `Pruefergebnis.nr = KEIN_ANGEBOT_VERFUEGBAR` ausgegeben. Die Preisberechnung für INOUT-Fahrten einer Anfrage erfolgt bis zu der INOUT-Fahrt, für die das Prüfergebnis `KEIN_ANGEBOT_VERFUEGBAR` ausgegeben wird. Für die nachfolgenden INOUT-Fahrten dieser Anfrage wird `Pruefergebnis.nr = HISTORIE_UNGUELTIG` ausgegeben und die Preisberechnung abgebrochen, da ansonsten falsche Preise berechnet werden könnten.

Bei sonstigen Fehlern wird `Pruefergebnis.nr = UNERWARTETER_ABBRUCH` ausgegeben.

Die Entwicklerinfo kann für Detailinformationen zur Fehlersuche genutzt werden, es sei denn, es handelt sich um `UNERWARTETER_ABBRUCH`.

TODO: Klärung, ob/welche Fehlermeldungen für die Kundenanzeige aus Sicht der Client-Systeme erforderlich werden.

4 Clientspezifische Erweiterungen

4.1 INOUTBW_ANZEIGEDATEN

Informationen, die gegenüber dem Kunden zur Anzeige gebracht (ggf. auch über die Rechnung) und nicht über die Standardattribute der Webservice-Schnittstelle bestückt werden, stehen – soweit vereinbart – als spezifische Erweiterung „INOUTBW_ANZEIGEDATEN“ der Ticketdaten zur Verfügung.

Der Wert „ANZEIGEDATEN“ setzt sich aus den folgenden Feldern zusammen:

PvKuerzel: STRING

Dieses Datenfeld gibt das Kürzel des Produktverantwortlichen, z.B. „VVS“ an.

PvName: STRING

Dieses Datenfeld gibt den Namen des Produktverantwortlichen, z.B. „Verkehrs- und Tarifverbund Stuttgart GmbH“ an.

PvAnschriftStrasse: STRING

Dieses Datenfeld gibt Straße und Hausnummer der Anschrift des Produktverantwortlichen, z.B. „Rotebühlstraße 121“ an.

PvAnschriftOrt: STRING

Dieses Datenfeld gibt den Ort der Anschrift des Produktverantwortlichen inkl. vorangestellter Postleitzahl, z.B. „70178 Stuttgart“ an.

PvUstID: STRING

Dieses Datenfeld gibt die Umsatzsteuer-Identifikationsnummer des Produktverantwortlichen, z.B. „DE 147800319“ an.

TODO Klärung, ob/inwieweit clientspezifisch zu differenzieren.

PKM Tarfrechner des Fraunhofer IVI

Spezifikation Webserviceschnittstelle PPE v7.1

VERTRAULICH

Dateiname:	PKM Tarfrechner Spezifikation Webserviceschnittstelle_PPEv7.1.1.docx
Erstellt am:	14.03.2017
Zuletzt geändert am:	17.05.2021
Version:	7.1.1
Ersteller:	Fraunhofer IVI Dresden

Autoren

Rostock, Sebastian / Fraunhofer IVI Dresden

Versionen

Version	Bearbeiter	Datum	Bemerkung
0.1	Rostock	14.03.2017	Dokument erstellt.
0.2	Rostock	24.03.2017	Einbettung der Inhalte aus der PV Schnittstelle.
0.3	Rostock	03.04.2017	Dokumentation aller Schnittstellen synchronisiert.
0.4	Rostock	22.06.2017	Überarbeitung nach Kommentierung.
0.5	Rostock	28.09.2017	Überarbeitung nach internem Review bzgl. JSON.
0.6	Rostock	25.10.2017	Überarbeitung nach Kommentierung.
0.7	Rostock	24.11.2017	Überarbeitung der Auswahlmechanik.
0.8	Rostock	26.02.2018	Dokumentation synchronisiert mit PKM-Schnittstellen.
0.9	Rostock	25.07.2018	Fixierung der Version und Dokumentation des Fault.
5.0.33	Rostock	08.04.2019	Synchronisierung der Versionen von Dokumenten und Webservice. Semantische Haupt- und Nebenversion werden gleichzeitig angepasst. In der dritten Versionskomponente zählt jedes Artefakt unabhängig seine eigenen Anpassungen.
6.0.0	Rostock	23.04.2019	Ergänzung der Art einer Anfrage und Ersetzen ihrer Pflichtfelder mit primitiven Attributen. Korrektur des Einheitslayouts.
6.0.1	Rostock	07.01.2020	Ergänzung der Kurzbezeichnung PPE sowie der nach Detailtiefe unterschiedenen Anfragearten. Verteilung ausgelagert.
7.0.0	Rostock	10.01.2020	Generelle Ergänzung/Umgestaltung nach neuesten Erkenntnissen.
7.0.1	Rostock	24.01.2020	Ergänzung des Auswahlmenüs. Verbesserung der Benennung sowie der Dokumentenstruktur.
7.0.2	Rostock	30.01.2020	Redaktion der Änderungen. Zentralisierung der Empfehlungen.
7.0.3	Rostock	05.02.2020	Ergänzung von Empfehlungen.
7.0.4	Rostock	25.02.2020	Überarbeitung von Reisenden und Fahrten. Ergänzung von Empfehlungen zu Modulkontexten und Anfragearten.
7.0.5	Rostock	28.02.2020	Ergänzung von Storno- und Fahrgastdaten. Ergänzung von und Gliederung der Empfehlungen.
7.0.6	Rostock	05.03.2020	Ergänzung von Linienlisten und Datenstrukturdiagrammen.
7.0.7	Rostock	20.03.2020	Redaktion der Datenfeldnamen.
7.0.8	Rostock	06.04.2020	Rechtschreibkorrektur.
7.0.9	Rostock	21.04.2020	Stornodaten redaktionell überarbeitet und Internetadresse zu Stornierungsregelungen ergänzt.
7.0.10	Rostock	28.05.2020	Korrektur der Beschreibung der EINGABEDATEN.
7.0.11	Rostock	16.07.2020	Kennungen der Prüfergebnisse zu fehlerhaften Anfragen und Antworten angepasst.
7.1.0	Rostock	23.07.2020	Anpassung der Datenstrukturen des Auswahlmenüs.
7.1.1	Rostock	17.05.2021	Verweis auf PKM Tarifrechner Grundlagen gelöscht.

Inhalt

1	Einleitung	5
2	Spezifikation.....	6
2.1	Webserviceschnittstelle	6
2.2	Anfragebezogene Datenstrukturen	6
2.2.1	Datenstrukturdiagramm	6
2.2.2	REQUEST	7
2.2.3	ANFRAGE	7
2.2.4	EINSCHRAENKUNG	8
2.2.5	REISENDER.....	8
2.2.6	VERBINDUNG	9
2.2.7	FAHRT.....	9
2.2.8	HALT	10
2.2.9	BERECHTIGUNG	10
2.2.10	FAHRGASTDATEN	11
2.2.11	IDMEDIENDATEN	11
2.2.12	DISTANZDATEN	11
2.2.13	LINIENDATEN	11
2.2.14	FAHRTDATEN.....	12
2.2.15	FAHRZEUGDATEN	12
2.2.16	GEOPOSITION.....	12
2.3	Antwortbezogene Datenstrukturen.....	13
2.3.1	Datenstrukturdiagramm	13
2.3.2	RESPONSE.....	13
2.3.3	ANTWORT.....	13
2.3.4	TICKETDATEN.....	14
2.3.5	ANGEBOTSDATEN.....	15
2.3.6	BEZAHLDATEN.....	15
2.3.7	STORNODATEN	16
2.3.8	AUSGABEDATEN	16
2.3.9	VDVKAEFSDATEN.....	17
2.3.10	VDVEINHEITSLAYOUT	18
2.3.11	PRUEFERGEBNIS	18
2.4	Allgemeine Datenstrukturen.....	19
2.4.1	Datenstrukturdiagramm	19
2.4.2	ELEMENT.....	19
2.4.3	ABSCHNITT	19
2.4.4	ERWEITERUNG	20
2.4.5	EINGABEDATEN.....	20
2.4.6	NUTZUNGSZUSTAND	21
3	Empfehlungen.....	23
3.1	Bestückung von Anfragen	23
3.2	Bestückung von Elementen	24
3.3	Bestückung von Reisenden	24
3.4	Bestückung von Eingabedaten.....	24

VERTRAULICH

3.5	Bestückung von Verbindungen	25
3.6	Bestückung von Berechtigungen.....	25
3.7	Bestückung von Einschränkungen	26
3.8	Bestückung von Ticketdaten	26

1 Einleitung

Die Datenstrukturen der Webserviceschnittstelle kodieren die fachlichen Datenstrukturen der Ein- und Ausgangsschnittstellen des Tarifmoduls sowie darüber hinaus gehende Merkmale zur Auswahl des anzuwendenden RN-Tarifmoduls. Der Webservice-Request enthält dazu eine Auflistung beliebig vieler, inhaltlich voneinander unabhängiger und daher parallel durch unterschiedliche RN-Tarifmodule verarbeitbarer Anfragen, zu welchen in der Webservice-Response ebenso vielen Antworten geliefert werden.

Die hierfür verwendete Notation zur Definition der Datenstrukturen entspricht der im Dokument «Auswahlmenü Technische Spezifikation» vorgestellten. Darüber hinaus wird auch auf die dort spezifizierten Datentypen aufgesetzt.

Sollten beim Einlesen einer Anfrage, der Auswertung dieser Anfrage im Rahmen des Fachablaufs zur Produktermittlung oder bei der Bestückung ihrer Antwort eine Ausnahme ausgelöst werden, so wird diese Anfrage in besonderer Weise beantwortet. Ihre Antwort enthält dann nur ein Prüfergebnis mit einem Entwicklerhinweistext. Letzterer beginnt grundsätzlich mit einem der Sätze «Anfrage ... konnte nicht vollständig eingelesen werden», «Antwort ... konnte nicht vollständig bestückt werden», «Ausnahme in Kernablauf» oder «Ausnahme im Fachablauf ...», welchem dann eine genauere Beschreibung der Fehlerursache folgt. Wenn der Fehler vor Auswertung der Produktermittlungsstrategie des RN-Tarifmoduls eintritt, trägt das Prüfergebnis die Nummer «ANFRAGE_UNGUELTIG». Andernfalls trägt es die Nummer «ANTWORT_UNGUELTIG».

Alle Merkmale von erweiterbaren Fachobjekte, jedoch nicht die Erweiterungen selbst, bspw. Angebotsdaten, Ticketdaten und Eingabedaten, werden inhaltlich ausschließlich vom PV verantwortet. Dies gilt insb. für Anzeigetexte dieser Fachobjekte. Zugesicherte sowie speziell formatierte Anzeigetexte müssen mit dem PV vereinbart, abgestimmt und ggf. in den Erweiterungen der entsprechenden Fachobjekte eingetragen werden.

Die der Spezifikation folgen Empfehlungen enthalten grundlegende Vorschläge zur Verwendung der spezifizierten Datenstrukturen, insbesondere zu Abstimmung und einheitlicher bzw. anwendungsspezifischer Verwendung von Kennungen.

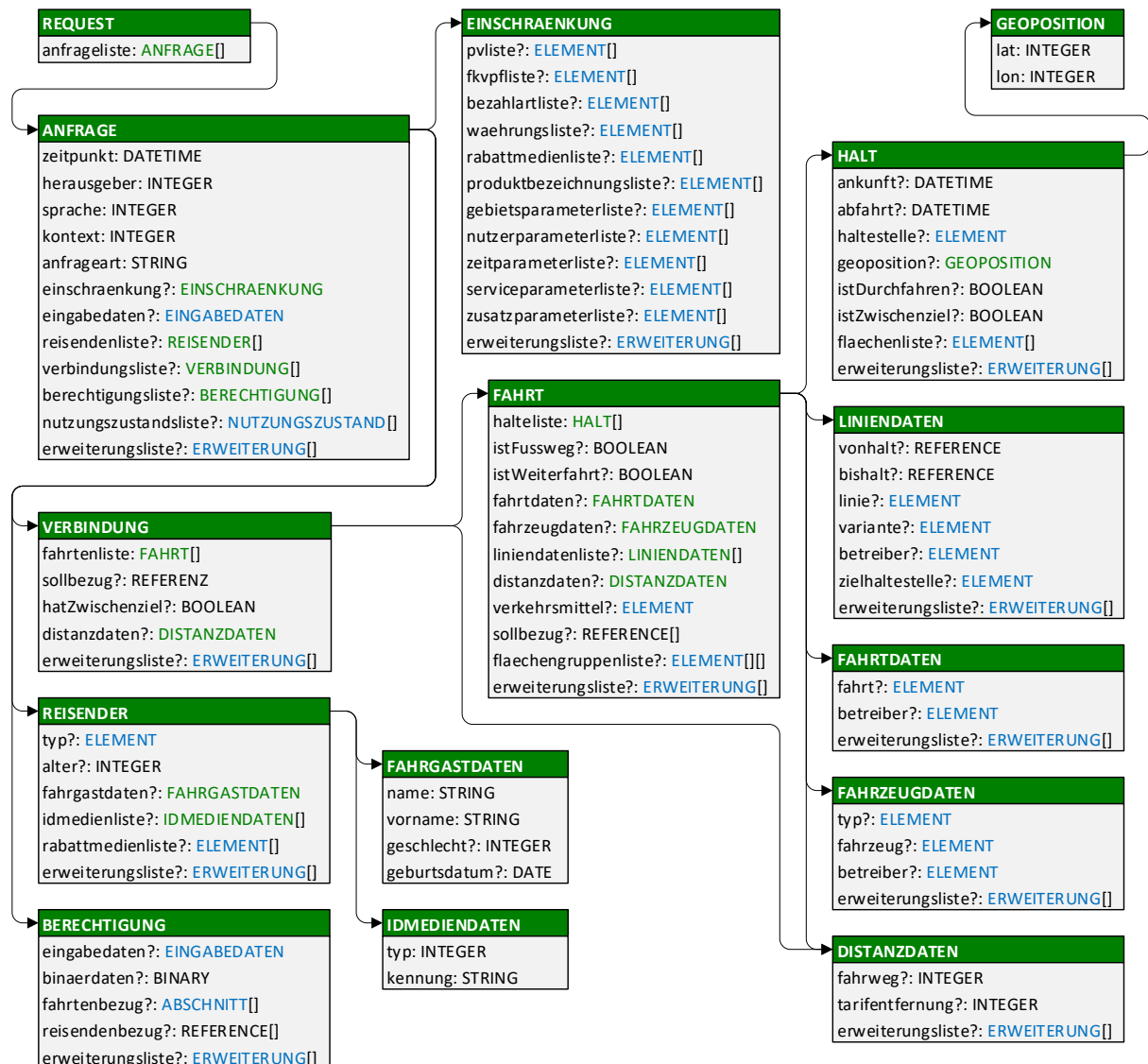
2 Spezifikation

2.1 Webserviceschnittstelle

Die Webserviceschnittstelle nutzt als Request und Response die in den nachfolgenden Abschnitten dokumentierten JSON-Datenstrukturen `REQUEST` bzw. `RESPONSE` als Nutzdaten des HTTP POST. Im Fehlerfall nutzt die Schnittstelle eine einfache Zeichenkette sowie die HTTP Fehlernummern.

2.2 Anfragebezogene Datenstrukturen

2.2.1 Datenstrukturdiagramm



2.2.2 REQUEST

Diese Datenstruktur kodiert den Webservice-Request.

`anfrageliste: ANFRAGE[]`

Dieses Datenfeld enthält die Auflistung der inhaltlich voneinander unabhängigen Anfragen.

2.2.3 ANFRAGE

Diese Datenstruktur kodiert eine Anfrage zur Ermittlung von Produktangeboten, die zu einer gegebenen Menge von Reisenden und deren ggf. bereits vorhandenen Berechtigungen insgesamt passen. Welche der hier aufgeführten Daten dazu benötigt werden, ist grundsätzlich unter den Beteiligten abzustimmen. Die hier beschriebenen Datenstrukturen wurden primär für die katalog- bzw. verbindungsbezogene Ermittlung konzipiert, lassen sich aber an vielen Stellen anwendungsspezifisch erweitern. Die katalogbezogene Produktermittlung bestimmt diese Ergebnisse zu Eingabedaten mit produktidentifizierenden Parametern sowie Merkmalen der zeitlichen und räumlichen Gültigkeit gewünschter Berechtigungen. Die verbindungsbezogene Produktermittlung bestimmt die Ergebnisse dagegen zu gegebenen Verbindungen, wobei diesbezüglich auch Gültigkeit und Anwendbarkeit gegebener Berechtigungen geprüft werden kann.

`zeitpunkt: DATETIME`

`herausgeber: INTEGER`

Diese Datenfelder nennen Zeitpunkt und Organisationsnummer zur Auswahl des anzuwendenden RN-Tarifmoduls.

`sprache: INTEGER`

Dieses Datenfeld nennt die Sprache der Anzeigetexte in der Antwort und dient der Prüfung der Einsetzbarkeit des anzuwendenden RN-Tarifmoduls.

`kontext: INTEGER`

Dieses Datenfeld nennt die Kennung des Einsatzorts als Nummer des aktuellen Modulkontextadapters und dient der Prüfung der Einsetzbarkeit des anzuwendenden RN-Tarifmoduls.

`anfrageart: STRING`

Dieses Datenfeld enthält eine Kennung, die festlegt, wie die Anfrage zu verarbeiten ist und welche Daten dazu in die Antwort einzutragen sind.

`einschraenkung?: EINSCHRAENKUNG`

Dieses Datenfeld enthält Kriterien zur groben Einschränkung der zu ermittelnden Produktangebote.

`eingabedaten?: EINGABEDATEN`

Dieses Datenfeld enthält produktidentifizierende, zeitliche, räumliche sowie ergänzende Merkmale gewünschter Berechtigungen, zu welchen Produktangebote ermittelt werden sollen.

`reisendenliste?: REISENDER[]`

Dieses Datenfeld enthält die Auflistung der Reisenden (Personen, Tiere, Gepäck, Fahrräder usw.), für welche gemeinsam Produktangebote ermittelt werden sollen.

`verbindungsliste?: VERBINDUNG[]`

Dieses Datenfeld enthält die Auflistung der Verbindungen, zu welchen Produktangebote ermittelt werden sollen.

`berechtigungsliste?: BERECHTIGUNG[]`

Dieses Datenfeld enthält die Auflistung der Berechtigungen, die bereits von Reisenden ggf. gemeinsam genutzt werden.

`nutzungszustandsliste?: NUTZUNGSZUSTAND[]`

Dieses Datenfeld enthält die Auflistung der zu Kunden erfassten tariflichen Nutzungszustandsdaten vor dem Kauf.

erweiterungsliste?: ERWEITERUNG[]

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen.

2.2.4 EINSCHRAENKUNG

Diese Datenstruktur kodiert Kriterien zur Einschränkung der zu ermittelnden Produktangebote. Die Datenfelder stehen hierbei für die Auflistungen der zulässigen Ausprägungen in den Merkmalen dieser Angebote, insbesondere denen der EINGABEDATEN. Eine nicht angegebene und damit leere Auflistung gibt dabei keine Einschränkung des entsprechenden Merkmals an. Welche Nummern in den Ausprägungen eingesetzt werden und ob bspw. die Unterscheidung von Bar- und Zeittarif über die zulässigen Tarife, Zeitparameter oder Zusatzparameter erfolgt, ist in jedem Fall abzustimmen.

pvlste?: ELEMENT[]

Dieses Datenfeld enthält die Auflistung der zulässigen Produktverantwortlichen.

fkvpvlste?: ELEMENT[]

Dieses Datenfeld enthält die Auflistung der zulässigen fremden/sekundären Kundenvertragspartner.

bezahlartliste?: ELEMENT[]

waehrungsliste?: ELEMENT[]

rabattmedienliste?: ELEMENT[]

Diese Datenfelder enthalten die Auflistungen der zulässigen Bezahlarten, Währungen und Rabattmedien.

produktbezeichnungsliste?: ELEMENT[]

gebietsparameterliste?: ELEMENT[]

nutzerparameterliste?: ELEMENT[]

zeitparameterliste?: ELEMENT[]

serviceparameterliste?: ELEMENT[]

zusatzparameterliste?: ELEMENT[]

Diese Datenfelder enthalten die Auflistungen der zulässigen Ausprägungen für die Identifikationsparameter Produktbezeichnung, Gebietsparameter, Nutzerparameter, Zeitparameter, Serviceparameter und Zusatzparameter gemäß PKM Spezifikation.

erweiterungsliste?: ERWEITERUNG[]

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen.

2.2.5 REISENDER

Diese Datenstruktur kodiert Merkmale eines reisenden Fahrgasts, Tiers oder Gegenstands.

typ?: ELEMENT

Dieses Datenfeld nennt den Typ des Fahrgasts, Tiers bzw. Gegenstands.

alter?: INTEGER

Dieses Datenfeld nennt das Alter des Fahrgasts zum Fahrtantritt in Jahren.

fahrgastdaten?: FAHRGASTDATEN

Dieses Datenfeld enthält personenbezogene Merkmale des Fahrgasts.

idmedienliste?: IDMEDIENDATEN[]

Dieses Datenfeld enthält die Auflistung der Identifikationsmedien, über die der Fahrgast verfügt.

rabattmedienliste?: ELEMENT[]

Dieses Datenfeld enthält die Auflistung der Rabattmedien, über die der Fahrgast als Kunde verfügt.

erweiterungsliste?: ERWEITERUNG[]

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen.

2.2.6 VERBINDUNG

Diese Datenstruktur kodiert im Wesentlichen eine Auflistung von Fahrten, die bspw. durch ein Fahrplanauskunftssystem zu einem Reisewunsch ermittelt oder aus einem Erfassungsprotokoll bspw. eines IN/OUT-Systems rekonstruiert wurden.

`fahrtenliste: FAHRT[]`

Dieses Datenfeld enthält die Auflistung der zeitlich aufeinander folgenden und sich diesbezüglich nicht überschneidenden Fahrten oder Fußwege.

`sollbezug?: REFERENZ`

Dieses Datenfeld ist nur bei einer vom regulären Soll abweichenden Verbindung anwendbar (bspw. Umweg oder Verspätung) und verweist dann auf diese Sollverbindung.

`hatZwischenziel?: BOOLEAN`

Dieses Datenfeld enthält «true», wenn die Verbindung mit Zwischenzielwahl ermittelt wurde. Dies kann für die Behandlung von Rund- und Rückfahrten relevant sein. Wenn es nicht angegeben ist, gilt es als «false».

`distanzdaten?: DISTANZDATEN`

Dieses Datenfeld enthält Angaben zur der durch diese Verbindung überwundenen Entfernung.

`erweiterungsliste?: ERWEITERUNG[]`

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen.

2.2.7 FAHRT

Diese Datenstruktur kodiert die Merkmale einer Fahrt, d.h. einer durch Ein- und Ausstieg begrenzten Fortbewegung in einem Fahrzeug unter bestimmten Linienbezeichnungen, bzw. eines Fußwegs.

`halteliste: HALT[]`

Dieses Datenfeld enthält die Auflistung der zeitlich aufeinander folgenden und sich diesbezüglich nicht überschneidenden tatsächlichen oder möglichen Halte mit mindestens zwei Elementen.

`istFussweg?: BOOLEAN`

Dieses Datenfeld enthält «true», wenn diese Fahrt für einen Fußweg steht. In diesem Fall sind die Merkmale von Linie, Fahrt, Fahrzeug und Verkehrsmittel grundsätzlich irrelevant. Wenn es nicht angegeben ist, gilt es als «false».

`istWeiterfahrt?: BOOLEAN`

Dieses Datenfeld enthält «true», wenn diese Fahrt bspw. wegen eines Linienwechsels die vorherige im gleichen Fahrzeug fortsetzt. Damit wird angezeigt, dass zwischen diesen beiden Fahrten kein Umstieg erfolgt. Die fortgesetzte Fahrt muss in der Fahrtenliste direkt davor angegeben sein. Wenn das Datenfeld nicht angegeben ist, gilt es als «false».

`fahrt Daten?: FAHRTDATEN`

`fahrzeugdaten?: FAHRZEUGDATEN`

`liniendatenliste?: LINIENDATEN[]`

Diese Datenfelder enthalten betriebliche Merkmale der Linien, der Fahrt bzw. des Fahrzeugs.

`distanzdaten?: DISTANZDATEN`

Dieses Datenfeld enthält Angaben zur der durch diese Fahrt überwundenen Entfernung.

`verkehrsmittel?: ELEMENT`

Dieses Datenfeld nennt den Verkehrsmitteltyp.

`sollbezug?: REFERENCE[]`

Dieses Datenfeld ist nur bei einer vom regulären Soll abweichenden Fahrt anwendbar (bspw. Umweg oder Verspätung) und verweist dann auf diese Sollfahrten in der Sollverbindung.

flaechengruppenliste?: ELEMENT[][]

Dieses Datenfeld enthält die Auflistung der durchfahrenen Gruppen tariflicher Flächen, bspw. Tarifzonen, welche in der angegebenen Reihenfolge durchfahren und bspw. zur Konsistenzsicherung mit denen der PV-Tarifmodule abgeglichen werden können. Eine Gruppe nennt dazu alternative Flächen, wenn der durch sie betreffende Fahrabschnitt tariflich zu jeder dieser Flächen gezählt werden könnte.

erweiterungsliste?: ERWEITERUNG[]

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen.

2.2.8 HALT

Diese Datenstruktur kodiert die Merkmale eines zeitlich begrenzten tatsächlichen oder möglichen Aufenthalts an einer auf der Strecke einer Fahrt liegenden Haltestelle bzw. Geoposition.

ankunft?: DATETIME

Dieses Datenfeld nennt den Zeitpunkt, ab dem der Aufenthalt an der Haltestelle bzw. Geoposition beginnt. Er muss beim letzten Halt einer Fahrt angegeben sein.

abfahrt?: DATETIME

Dieses Datenfeld nennt den Zeitpunkt, ab dem der Aufenthalt an der Haltestelle bzw. Geoposition endet. Er muss beim ersten Halt einer Fahrt angegeben sein.

haltestelle?: ELEMENT

Dieses Datenfeld nennt die Haltestelle, sofern der Aufenthalt an einer solchen erfolgt.

geoposition?: GEOPOSITION

Dieses Datenfeld nennt die Geoposition, an welcher der Aufenthalt erfolgt.

flaechenliste?: ELEMENT[]

Dieses Datenfeld enthält die Auflistung der tariflichen Flächen, bspw. Tarifzonen, zu denen die Haltestelle bzw. Geoposition gezählt werden könnte. Diese können bspw. zur Konsistenzsicherung mit denen der PV-Tarifmodule abgeglichen werden.

istDurchfahren?: BOOLEAN

Dieses Datenfeld ist nur dann «true», wenn an der Haltestelle bzw. Geoposition nicht gehalten wird. Wenn es nicht angegeben ist, gilt es als «false». Es kann am ersten bzw. letzten Halt einer Fahrt nicht «true» sein.

istZwischenziel?: BOOLEAN

Dieses Datenfeld ist nur dann «true», wenn die Haltestelle bzw. Geoposition aus einer Zwischenzielwahl herrührt. Wenn es nicht angegeben ist, gilt es als «false».

erweiterungsliste?: ERWEITERUNG[]

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen, bspw. Größe der Abweichung von Soll- und Ist-Zeitangaben oder Orts- und Gemeindeangaben zur Haltestelle.

2.2.9 BERECHTIGUNG

Diese Datenstruktur kodiert Merkmale einer bereits gekauften Berechtigung.

eingabedaten?: EINGABEDATEN

Dieses Datenfeld enthält die tariflichen Merkmale der Berechtigung, welche bspw. aus einer manuellen Eingabe oder einem vorangegangenen Produktangebot stammen können.

binaerdaten?: BINARY

Dieses Datenfeld enthält die Daten der elektronisch abgelegten Berechtigung, welche bspw. von einem Nutzermedium eingelesen oder aus einer Kundendatenbank stammen können.

fahrtenbezug?: ABSCHNITT[]

Dieses Datenfeld enthält die Auflistung der Abschnitte einer oder mehrerer Fahrten, für welche die Berechtigung gilt.

reisendenbezug?: REFERENCE[]

Dieses Datenfeld enthält die Auflistung der Verweise auf die Reisenden, an welche die Berechtigung gebunden ist.

erweiterungsliste?: ERWEITERUNG[]

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen, bspw. Merkmale eines Fahrausweises aus proprietären Kundendatenbanken oder abgeleitet aus Fotografien vom Papierfahrtschein.

2.2.10 FAHRGASTDATEN

Diese Datenstruktur kodiert Merkmale zur Bestückung des Personenbezugs bspw. über das TLV-Element «0xDB» einer elektronischen Berechtigung gemäß VDV KA NM Spezifikation.

name: STRING

vorname: STRING

geschlecht?: INTEGER

geburtsdatum?: DATE

Diese Datenfelder nennen die entsprechenden Merkmale des Fahrgasts.

2.2.11 IDMEDIENDATEN

Diese Datenstruktur kodiert Merkmale zur Bestückung des Personenbezugs bspw. über das TLV-Element «0xD7» einer elektronischen Berechtigung gemäß VDV KA NM Spezifikation.

typ: INTEGER

kennung: STRING

Diese Datenfelder nennen die entsprechenden Merkmale des Identifikationsmediums.

2.2.12 DISTANZDATEN

Diese Datenstruktur kodiert Entfernungsmerkmale einer Fahrt oder Verbindung.

fahrweg?: INTEGER

Dieses Datenfeld gibt die Entfernung in Metern an.

tarifentfernung?: INTEGER

Dieses Datenfeld gibt die tarifliche Entfernung in einer abgestimmten Granularität an, bspw. Luftlinie oder Tarifikilometer.

erweiterungsliste?: ERWEITERUNG[]

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen.

2.2.13 LINIENDATEN

Diese Datenstruktur kodiert betriebliche Merkmale der Linie zu einem Fahrtabschnitt.

vonhalt?: REFERENCE

Dieses Datenfeld enthält den Verweis auf den Halt, ab dem diese Liniendaten gelten. Wenn es nicht angegeben ist, beginnt gelten diese ab dem ersten Halt der Fahrt.

bishalt?: REFERENCE

Dieses Datenfeld enthält den Verweis auf den Halt, bis zu dem diese Liniendaten gelten. Wenn es nicht angegeben ist, gelten diese bis zum letzten Halt der Fahrt.

linie?: ELEMENT

variante?: ELEMENT

Diese Datenfelder nennen Nummer und Variante der Linie. Mit der Linienvariante ist eine gerichtete, fahrplanmäßig befahrene Haltestellenfolge eindeutig bezeichnet. Bezogen auf die Linie können damit bspw. die Hin- und Rückrichtung oder eine abweichende und/oder verkürzte Haltestellenfolge ausgedrückt werden.

betreiber?: ELEMENT

zielhaltestelle?: ELEMENT

Diese Datenfelder nennen Betreiber und Zielhaltestelle der Linie.

erweiterungsliste?: ERWEITERUNG[]

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen.

2.2.14 FAHRTDATEN

Diese Datenstruktur kodiert betriebliche Merkmale einer Fahrt.

fahrt?: ELEMENT

Dieses Datenfeld nennt die Nummer der Fahrt. Diese Nummer wird i.d.R. aus Betriebsplanungssystemen bezogen. Im Bereich der Deutschen Bahn ist dies bspw. die Zugnummer, im Bereich des (Stadt-)Busverkehrs die Kombination aus Kursnummer und laufender Nummer der Fahrt auf diesem Kurs.

betreiber?: ELEMENT

Dieses Datenfeld nennt den Betreiber der Fahrt.

erweiterungsliste?: ERWEITERUNG[]

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen.

2.2.15 FAHRZEUGDATEN

Diese Datenstruktur kodiert betriebliche Merkmale eines Fahrzeugs.

typ?: ELEMENT

Dieses Datenfeld nennt den Typ des Fahrzeugs.

fahrzeug?: ELEMENT

Dieses Datenfeld nennt die Nummer des Fahrzeugs.

betreiber?: ELEMENT

Dieses Datenfeld nennt den Betreiber des Fahrzeugs.

erweiterungsliste?: ERWEITERUNG[]

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen.

2.2.16 GEOPOSITION

Diese Datenstruktur kodiert eine geographische Position auf der Erde.

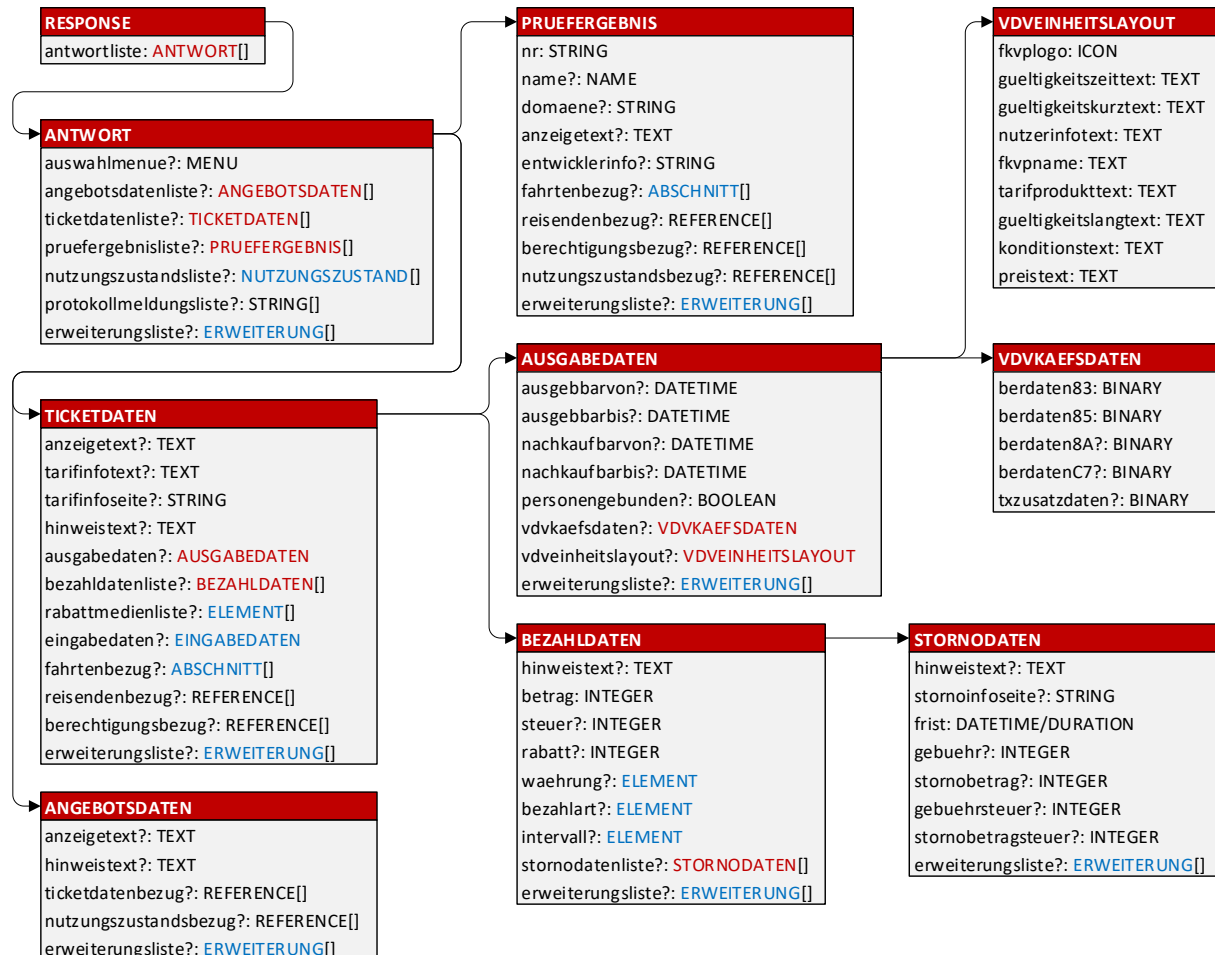
lat: INTEGER

lon: INTEGER

Diese Datenfelder nennen Breitengrad (N) und Längengrad (E) im WGS84 in ganzen Mikrowinkelgrad.

2.3 Antwortbezogene Datenstrukturen

2.3.1 Datenstrukturdiagramm



2.3.2 RESPONSE

Diese Datenstruktur kodiert die Webservices-Response.

```
antwortliste: ANTWORT[]
```

Dieses Datenfeld enthält die Auflistung der Antworten zu den Anfragen des Webservice-Request. Reihenfolge und Anzahl der Antworten entsprechen denen der Anfragen.

2.3.3 ANTWORT

Diese Datenstruktur kodiert Informationen zu ermittelten Produktangeboten und durchgeführten Prüfungen.

```
auswahlmenue?: MENU
```

Dieses Datenfeld enthält das Auswahlmenü, welches nach vorheriger Abstimmung auch auf Datensätze von Anfrage und/oder Antwort verweisen kann. Dazu enthält es entsprechende Anbindungen (**BINDING**) an Merkmale (**FEATURE**), deren zulässige Eigenschaften (**PROPERTY**) in ihrem Wert (**value**) einen entsprechenden Verweis als **REFERENCE** enthalten (vgl. Dokument «Auswahlmenü Technische Spezifikation»).

angebotsdatenliste?: ANGEBOTSDATEN[]

Dieses Datenfeld enthält die Auflistung der zueinander alternativen Produktangebote, welche grundsätzlich in der hier gegebenen Reihenfolge angezeigt werden sollten. Abhängig vom Detailgrad können diese Angebote auch einfache Tarifinformationen bspw. nur mit Anzeigetexten und Preisinformationen darstellen.

ticketdatenliste?: TICKETDATEN[]

Dieses Datenfeld enthält die Auflistung der von den Produktangeboten gemeinsam genutzten Ticketdaten.

pruefergebnisliste?: PRUEFERGEBNIS[]

Dieses Datenfeld enthält die Auflistung der Ergebnisse von Prüfergebnissen bspw. der Gültigkeit, Zulässigkeit oder Fehlerfreiheit der in der Anfrage angegebenen Daten.

nutzungszustandsliste?: NUTZUNGSZUSTAND[]

Dieses Datenfeld enthält die Auflistung der zu Kunden erfassten tariflichen Nutzungszustandsdaten, die von den Produktangeboten gemeinsam benutzt werden und nach dem Kauf der entsprechenden Produktangebote gelten.

protokollmeldungsliste?: STRING[]

Dieses Datenfeld enthält eine Auflistung informativer Protokollmeldungen.

erweiterungsliste?: ERWEITERUNG[]

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen.

2.3.4 TICKETDATEN

Diese Datenstruktur kodiert Informationen zu einer Berechtigung, die für bestimmte Reisende gemeinsam sowie ggf. auf mehreren Fahrtabschnitten und ggf. unter Voraussetzung gegebener Berechtigungen gilt und welche ggf. auch zur Ausgabe der Berechtigung eingesetzt werden können. Wenn sie weder über Eingabedaten noch über Ausgabedaten verfügt, kennzeichnet dies die Nichtkaufbarkeit einer entsprechenden Berechtigung.

anzeigetext?: TEXT

Dieses Datenfeld enthält den Anzeigetext für den Kunden, bspw. mit Produktbezeichnung, Preisstufe und Serviceklasse.

tarifinfotext?: TEXT

Dieses Datenfeld enthält einen allgemeinen tariflichen Hinweistext zum angebotenen Produkt.

tarifinfoseite?: STRING

Dieses Datenfeld enthält die Internetadressen zu weiterführenden Tarifinformationen.

hinweistext?: TEXT

Dieses Datenfeld enthält einen speziellen Hinweistext für den Kunden, bspw. einen Hinweis zum angebotenen Angebot, zur Ermittlung des Angebots, zu Besonderheiten bei Streifenkarten, zu Produktabhängigkeiten oder einen Kaufhinweis.

ausgabedaten?: AUSGABEDATEN

Dieses Datenfeld enthält die tariflichen Merkmale zur Ausgabe der Berechtigung.

bezahldatenliste?: BEZAHLDATEN[]

Dieses Datenfeld enthält die ggf. nach Bezahlart, Währung und/oder Zahlungsintervall differenzierten Zahlungsbeträge zusammen mit deren Stornierungsmöglichkeiten.

rabattmedienliste?: ELEMENT[]

Dieses Datenfeld enthält die Auflistung der Rabattmedien, die zur Nutzung der Berechtigung vorausgesetzt sind, bspw. Sozialpass oder Bahncard.

eingabedaten?: EINGABEDATEN

Dieses Datenfeld enthält genau die tariflichen Merkmale der Berechtigung, mit denen diese Berechtigung nachfolgend nochmals angefragt werden kann (vgl. Katalog, Favorit).

fahrtenbezug?: ABSCHNITT[]

Dieses Datenfeld enthält die Auflistung der Abschnitte einer oder mehrerer Fahrten, auf denen die Berechtigung gilt.

reisendenbezug?: REFERENCE[]

Dieses Datenfeld enthält die Auflistung der Verweise auf die Reisenden, die für diese Berechtigung vorausgesetzt sind, bspw. bei personenbezogenen Produkten.

berechtigungsbezug?: REFERENCE[]

Dieses Datenfeld enthält die Auflistung der Verweise auf die Berechtigungen, die für diese Berechtigung vorausgesetzt sind, bspw. Grundfahrtschein bei Zusatztickets.

erweiterungsliste?: ERWEITERUNG[]

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen, bspw. Visualisierungsdaten zum Freigaberaum oder Verweise auf Tarifbestimmungen und/oder Beförderungsbedingungen.

2.3.5 ANGEBOTS DATEN

Diese Datenstruktur kodiert ein Produktangebot als Zusammenstellung gemeinsam wirkender Ticketdaten, die nur gemeinsam den in der Anfrage gestellten Reisewunsch befriedigen.

anzeigetext?: TEXT

Dieses Datenfeld enthält den Anzeigetext zum Produktangebot, welcher dem Kunden bspw. Anzahl und Bezeichnung der beteiligten Produkte angeben kann.

hinweistext?: TEXT

Dieses Datenfeld enthält den Hinweistext zum Produktangebot, welcher dem Kunden bspw. Bezeichnung und Hinweise der beteiligten Produkte nennen sowie Hinweise zum Umgang mit der spezifischen Ticketzusammenstellung angeben kann.

ticketdatenbezug?: REFERENCE[]

Dieses Datenfeld enthält die Auflistung der Verweise auf die Ticketdaten des Produktangebots. Mehrfach vorkommende Verweise zeigen indirekt die Anzahl der entsprechenden Ticketdaten an.

nutzungszustandsbezug?: REFERENCE[]

Dieses Datenfeld enthält die Auflistung der Verweise auf die Nutzungszustandsdaten, die nach dem Kauf der Berechtigungen dieses Produktangebots für die jeweiligen Kunden gelten. Diese Daten können nur dann vorliegen, wenn in der Anfrage entsprechende Nutzungszustandsdaten von vor dem Kauf angegeben wurden. Die Zustandsdaten nach dem Kauf dürfen nur dann gespeichert werden, wenn die zuletzt gespeicherten Nutzungszustandsdaten vor dem Kauf gleich den in der Anfrage aufgeführten Nutzungszustandsdaten sind. Die Speicherung erfolgt i.d.R. PV- und kundenspezifisch beim KVP.

erweiterungsliste?: ERWEITERUNG[]

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen.

2.3.6 BEZAHL DATEN

Diese Datenstruktur kodiert einen ggf. bezahlart-, währungs- und zahlungsintervallspezifischen Zahlungsbetrag mit Angabe des inklusiven Mehrwertsteuersatzes sowie dazugehörigen Stornierungsmodalitäten.

hinweistext?: TEXT

Dieses Datenfeld enthält einen Hinweistext, bspw. zu Rabatt oder Bezahlarten.

betrag: INTEGER

Dieses Datenfeld nennt den Zahlungsbetrag als Bruttobetrag.

steuer?: INTEGER

Dieses Datenfeld nennt im Zahlungsbetrag enthaltenen Mehrwertsteuersatz in 1/100 Prozent.

`rabatt?: INTEGER`

Dieses Datenfeld nennt die nicht im Zahlbetrag enthaltene Ersparnis aufgrund des eingesetzten Rabattmodells.

`waehrung?: ELEMENT`

Dieses Datenfeld nennt die Währung von Zahl- und Rabattbetrag.

`bezahllart?: ELEMENT`

Dieses Datenfeld nennt die Bezahlart.

`intervall?: ELEMENT`

Dieses Datenfeld nennt das Zahlungsintervall.

`stornodatenliste?: STORNODATEN[]`

Dieses Datenfeld enthält die nach Stornierungsfrist differenzierten Stornogebühren und/oder Stornobeträge. Zu einem gegebenen Zeitpunkt ist die Stornierung mit der frühesten zukünftigen Stornierungsfrist anzuwenden.

`erweiterungsliste?: ERWEITERUNG[]`

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen.

2.3.7 STORNODATEN

Diese Datenstruktur kodiert fristbezogene Merkmale zur Stornierung einer bereits gekauften Berechtigung.

`hinweistext?: TEXT`

Dieses Datenfeld enthält einen Hinweistext zum Stornierungsvorgang für den Kunden.

`stornoinfoseite?: STRING`

Dieses Datenfeld enthält die Internetadressen zu den Stornierungsregelungen.

`frist: DATETIME/DURATION`

Dieses Datenfeld nennt die Stornierungsfrist als absoluten Zeitpunkt oder als Zeitspanne ab dem konkreten Gültigkeitsbeginn der Berechtigung in Richtung Zukunft.

`gebuehr?: INTEGER`

`stornobetrag?: INTEGER`

Diese Datenfelder nennen Stornogebühr und Stornobetrag als Bruttobeträge.

`gebuehrsteuer?: INTEGER`

`stornobetragsteuer?: INTEGER`

Diese Datenfelder nennen den in Stornogebühr bzw. Stornobetrag enthaltenen Mehrwertsteuersatz in 1/100 Prozent.

`erweiterungsliste?: ERWEITERUNG[]`

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen, bspw. Merkmale zu weiteren Konditionen der Stornierung.

2.3.8 AUSGABEDATEN

Diese Datenstruktur kodiert ggf. zeitlich eingeschränkt zulässige Merkmale zur Ausgabe einer gedruckten und/oder elektronischen Berechtigung.

`ausgebbarvon?: DATETIME`

`ausgebbarbis?: DATETIME`

Diese Datenfelder geben Beginn und Ende des Zeitraums an, in welchem die Berechtigung auf ein Nutzermedium ausgegeben werden darf.

nachkaufbarvon?: DATETIME

nachkaufbarbis?: DATETIME

Diese Datenfelder geben Beginn und Ende des Zeitraums an, in welchem sowohl der gewünschte Gültigkeitsbeginn als auch das gewünschte Gültigkeitsende der parallel zu diesen Ausgabedaten angegebenen Eingabedaten für einen wiederholten Kauf liegen dürfen.

personengebunden?: BOOLEAN

Dieses Datenfeld ist nur dann «true», wenn die Berechtigung außerhalb des Tarifmoduls personalisiert werden muss. Diese ist dann von Bedeutung, wenn der notwendige Personenbezug bspw. noch nicht in den Einheitslayout- oder Binärdaten der Berechtigung enthalten ist. Wenn es nicht angegeben ist, gilt es als «false».

vdvkaefdaten?: VDVKAEFSDATEN

Dieses Datenfeld enthält die Daten zur elektronischen Ausgabe gemäß VDV KA Spezifikation.

vdveinheitslayout?: VDVEINHEITSLAYOUT

Dieses Datenfeld enthält Anzeigedaten für das Handyticketeinheitslayout gemäß VDV KA Schrift 733.

erweiterungsliste?: ERWEITERUNG[]

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen, bspw. spezifische abgestimmte Daten zur gedruckten und/oder elektronischen Ausgabe sowie zur Buchung des Verkaufs (Einnahmemeldung, Verkaufsdatenprotokoll, Registrierungsdaten, Fahrscheinverkaufsdaten).

2.3.9 VDVKAEFSDATEN

Diese Datenstruktur kodiert die vom PV verantworteten Daten zur Ausgabe einer elektronischen Berechtigung gemäß VDV KA Spezifikation.

berdaten83: BINARY

Dieses Datenfeld enthält die Bytefolge für den Inhalt von TLV-Element «0x83 – Verzeichniseintrag der Berechtigung», in welchem vor Ausgabe der Berechtigung noch die ID von Schlüssel und Berechtigung zu ergänzen sind.

berdaten85: BINARY

Dieses Datenfeld enthält die Bytefolge für den Inhalt von TLV-Element «0x85 – Statischer Teil der separaten Berechtigungsdaten».

berdaten8A?: BINARY

Dieses Datenfeld enthält die Bytefolge für den Inhalt von TLV-Element «0x8A – Dynamischer Teil der separaten Berechtigungsdaten».

berdatenC7?: BINARY

Dieses Datenfeld enthält die Bytefolge für den Inhalt von TLV-Element «0xC7 – Infotext der Berechtigung».

txzusatzdaten?: BINARY

Dieses Datenfeld enthält die Bytefolge für den Inhalt von «Berechtigung Tarifbereich Zusatz» des TXABER.

mkpv?: INTEGER

mkkvp?: INTEGER

mktr?: INTEGER

mkauth?: INTEGER

Diese Datenfelder enthalten die ID der Schlüssel zur Ausgabe der elektronischen Berechtigung gemäß VDV KA NM Spezifikation.

Diese Datenfelder ENTFALLEN wahrscheinlich nach Abstimmung mit VDV ETS, da der PV dies Schlüssel nicht wissen kann. Grundsätzlich müsste für die Pflege der Schlüssel eine Tabelle gepflegt werden, die von den Spalten [vonzeitpunkt, biszeitpunkt, pvn, kvpn, efmprodnr] auf die Spalten [mkpv, mkkvp, mktr, mkauth, pvttest, kvptest] abbildet. Diese muss vom Berechtigungssignierenden gepflegt werden, da dieser auch die Schlüssel in seinem SAM haben muss.

2.3.10 VDVEINHEITSLAYOUT

Diese Datenstruktur kodiert in Verbindung mit den VDVKAEFSDATEN die von einem Produktverantwortlichen verantworteten Inhalte des Handyticketeinheitslayouts gemäß VDV Schrift 733. Für ein vollständiges Handyticketeinheitslayout müssten diese Inhalte noch um die bei der Ausgabe der Berechtigung erzeugten Informationen ergänzt werden.

fkvplogo: ICON

Dieses Datenfeld enthält die Kennung des graphischen Logos des fremd-KVP (Abschnitt 1/1).

gueltigkeitszeittext: TEXT

Dieses Datenfeld enthält den Gültigkeitsbeginn bzw. Gültigkeitszeitraum (Abschnitt 1/3).

gueltigkeitskurztext: TEXT

Dieses Datenfeld enthält Kurzangaben zur zeitlichen und räumlichen Gültigkeit (Abschnitt 1/4).

nutzerinfotext: TEXT

Dieses Datenfeld enthält die Referenz zum Ticketnutzer (Abschnitt 1/6).

fkvpname: TEXT

Dieses Datenfeld enthält den Namen des fremd-KVP (Abschnitt 2/1).

tarifprodukttext: TEXT

Dieses Datenfeld enthält die Bezeichnung des Tarifprodukts (Abschnitt 2/2).

gueltigkeitslangtext: TEXT

Dieses Datenfeld enthält ausführliche zeitliche und räumliche Informationen (Abschnitt 2/3).

konditionstext: TEXT

Dieses Datenfeld enthält die Konditionen bzw. Tarifhinweise (Abschnitt 2/4).

preistext: TEXT

Dieses Datenfeld enthält den Zahlbetrag mit Währung und Mehrwertsteuerangabe (Abschnitt 2/6).

2.3.11 PRUEFERGEBNIS

Diese Datenstruktur kodiert das Ergebnis einer Prüfung der in der Anfrage bereitgestellten Informationen und kann bspw. Aussagen über die Gültigkeit, Zulässigkeit oder Korrektheit von Datenstrukturen und Zusammenhängen machen sowie auch angeben, dass eine bestimmte Prüfung nicht durchgeführt wurde.

nr: STRING

name?: NAME

domaene?: STRING

Siehe Datenstruktur ELEMENT.

anzeigetext?: TEXT

Dieses Datenfeld enthält den Anzeigetext für die Kunden.

entwicklerinfo?: STRING

Dieses Datenfeld enthält den Hinweistext für die Entwickler.

fahrtenbezug?: ABSCHNITT[]

Dieses Datenfeld enthält die Auflistung der geprüften Abschnitte einer oder mehrerer Fahrten.

reisendenbezug?: REFERENCE[]

berechtigungsbezug?: REFERENCE[]

nutzungszustandsbezug?: REFERENCE[]

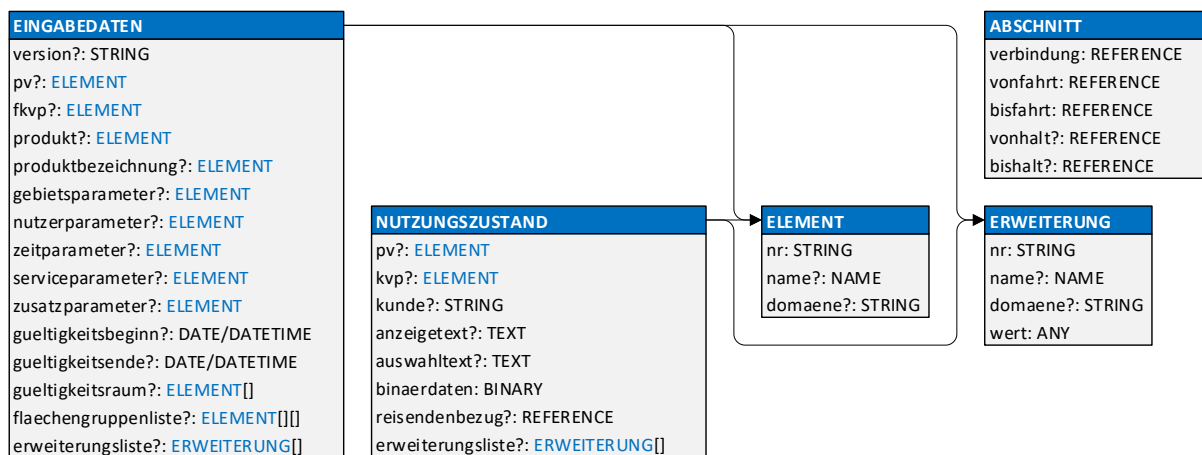
Diese Datenfelder enthalten die Auflistungen der Verweise auf die geprüften Reisenden, Berechtigungen und Nutzungszustandsdaten der Anfrage.

erweiterungsliste?: ERWEITERUNG[]

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen.

2.4 Allgemeine Datenstrukturen

2.4.1 Datenstrukturdiagramm



2.4.2 ELEMENT

Diese Datenstruktur kodiert eine nummerierte und ggf. benannte Ausprägung einer tariflich relevanten Größe. Anwendung findet sie bspw. bei der Angabe von Haltestellen, Linien- und Fahrzeugmerkmalen.

nr: STRING

Dieses Datenfeld nennt den technischen Identifikator bzw. die Kennung der Ausprägung im Bezugsrahmen.

name?: NAME

Dieses Datenfeld nennt die informative Bezeichnung der Ausprägung, bspw. zum Einsatz in Fehlermeldungen zu unbekannten oder nicht unterstützten Ausprägungen.

domaene?: STRING

Dieses Datenfeld nennt den technischen Identifikator bzw. die Kennung des Bezugsrahmens, in welchem die Kennung der Ausprägung definiert ist und welche angibt, wie die Kennung des Elements zu interpretieren ist. Wenn es nicht angegeben ist, wird der Bezugsrahmen des Einsatzorts/Empfängers dieser Ausprägung genutzt.

2.4.3 ABSCHNITT

Diese Datenstruktur kodiert mit seinen Positionsangaben einen logischen Abschnitt einer Fahrt oder Verbindung und realisiert damit einen Fahrtenbezug.

`verbindung: REFERENCE`

Dieses Datenfeld enthält den Verweis auf die Verbindung, deren Abschnitt beschrieben wird.

`vonfahrt: REFERENCE`

Dieses Datenfeld enthält den Verweis auf die Fahrt, mit welcher der Abschnitt beginnt.

`bisfahrt: REFERENCE`

Dieses Datenfeld enthält den Verweis auf die Fahrt, mit welcher der Abschnitt endet.

`vonhalt?: REFERENCE`

Dieses Datenfeld enthält den Verweis auf den Halt in der ersten Fahrt des Abschnitts, mit welcher der Abschnitt beginnt. Wenn es nicht angegeben ist, beginnt der Abschnitt mit dem ersten Halt dieser Fahrt.

`bishalt?: REFERENCE`

Dieses Datenfeld enthält den Verweis auf den Halt in der letzten Fahrt des Abschnitts, mit welcher der Abschnitt beginnt. Wenn es nicht angegeben ist, endet der Abschnitt mit dem letzten Halt dieser Fahrt.

2.4.4 ERWEITERUNG

Diese Datenstruktur kodiert ein anwendungsspezifisches Merkmal mit beliebigem Wert, welches über eine unter den Beteiligten abgestimmte Kennung identifiziert und damit als Ausprägung einer (tariflich) relevanten Größe verstanden werden kann. Eine Liste solcher Merkmale entspricht einer Abbildung von Merkmalskennungen auf beliebige Werte. Unbekannte Merkmale können somit ignoriert werden.

Erweiterungen an nahezu allen Datenstrukturen in Anfrage und Antwort erlauben das dynamische Ergänzen von Informationen und damit das frühzeitige Experimentieren mit Merkmalen, für welche in späteren Versionen der Schnittstelle eigene optimierte Datenstrukturen bereitgestellt werden könnten.

`nr: STRING`

`name?: NAME`

`domaene?: STRING`

Siehe Datenstruktur `ELEMENT`.

`wert: ANY`

Dieses Datenfeld enthält den konkreten Wert in einer unter den Beteiligten abgestimmten Datenstruktur.

2.4.5 EINGABEDATEN

Diese Datenstruktur kodiert angebotsidentifizierende Merkmale, welche bspw. als Eingabedaten für nachgelagerte Verkaufssysteme eingesetzt werden können. Anwendung finden solche Eingabedaten bei der Ermittlung von Produktangeboten sowie bei der Beschreibung von Berechtigungen, die in der Anfrage mitgegeben werden.

`version?: STRING`

Dieses Datenfeld dient der Erkennung kompatibler Varianten von Eingabedaten.

`pv?: ELEMENT`

Dieses Datenfeld nennt den Produktverantwortlichen.

`fkvp?: ELEMENT`

Dieses Datenfeld nennt den fremden/sekundären Kundenvertragspartner, d.h. eine zum Verkauf berechnigte Organisation, in deren Namen die Ausgabe der Berechtigung erfolgt, die aber nicht zwangsläufig auch als Vertragspartner des Kunden auftritt.

produkt?: ELEMENT

Dieses Datenfeld nennt das Tarifprodukt.

produktbezeichnung?: ELEMENT

gebietsparameter?: ELEMENT

nutzerparameter?: ELEMENT

zeitparameter?: ELEMENT

serviceparameter?: ELEMENT

zusatzparameter?: ELEMENT

Diese Datenfelder enthalten die Werte der gleichnamigen Identifikationsparameter gemäß PKM Spezifikation, welche alternativ zur Produktnummer zur Kennzeichnung des Tarifprodukts eingesetzt werden können.

gueltigkeitsbeginn?: DATE/DATETIME

gueltigkeitsende?: DATE/DATETIME

Diese Datenfelder enthalten Gültigkeitsbeginn und Gültigkeitsende der Berechtigung. Bei den in der Anfrage gegebenen Eingabedaten stehen dies für den gewünschten Gültigkeitsbeginn bzw. das gewünschte Gültigkeitsende.

gueltigkeitsraum?: ELEMENT[]

Dieses Datenfeld enthält die Auflistung der Flächen, die den Bezug zur (gewünschten) räumlichen Gültigkeit der Berechtigung ausdrücken.

flaechengruppenliste?: ELEMENT[][]

Dieses Datenfeld enthält die Auflistung der durchfahrenen Gruppen tariflicher Flächen einer Verbindung (vgl. analog FAHRT).

erweiterungsliste?: ERWEITERUNG[]

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen, bspw. Mitnahmen, die nicht als Reisende abgebildet werden können, Verkehrsmittel- und Linienbezug aus dem Entwertungstempel der Berechtigung sowie andere den Kundenwunsch zur Berechtigung beschreibende Parameter oder Merkmale aus einer vorangegangenen verbindungsbasierten Angebotsermittlung.

2.4.6 NUTZUNGSZUSTAND

Diese Datenstruktur kodiert von einem Produktverantwortlichen verantwortete, produkt- und berechtigungsübergreifende Nutzungszustandsdaten als Teil der Kundendaten eines Kundenvertragspartners. Solche Daten können für ein Rabattmodell relevanten Informationen enthalten, bspw. tarifliche Zähler und Statistiken zu historischen Käufen, Fahrten und Berechtigungen. Sie sollten darüber hinaus über Versions- und Strukturkennungen, Verfallszeitpunkte und ggf. Signaturen verfügen.

Zustandsdaten als Teil der Kundendaten eines Kundenvertragspartners erlauben die produkt- und berechtigungsübergreifende Erfassung und Anpassung von Informationen bspw. zu tariflichen Zählern und Statistiken eines Rabattmodells oder zu tariflich relevanten historischen Käufen, Fahrten und Berechtigungen.

Bis auf Anzeigetext, Binärdaten und Erweiterungen werden die Merkmale der Zustandsdaten von der Anfrage unverändert in die Antwort übernommen.

pv?: ELEMENT

Dieses Datenfeld nennt den Produktverantwortlichen, der die Zustandsdaten des Kunden inhaltlich verantwortet.

kvp?: ELEMENT

Dieses Datenfeld nennt den Kundenvertragspartner, der die Zustandsdaten des Kunden technisch verwaltet.

kunde?: STRING

Dieses Datenfeld enthält die Kennung des Kunden, bspw. als anonymisierten Streuwert.

anzeigetext?: TEXT

Dieses Datenfeld enthält die textuelle Beschreibung der für den Kunden relevanten Teile der Zustandsdaten.

`auswahltext?: TEXT`

Dieses Datenfeld enthält die Beschriftung der Zustandsdaten, die bei der indirekten Auswahl von Produktangeboten über die anzupassenden Zustandsdaten eingesetzt werden kann.

`binaerdaten: BINARY`

Dieses Datenfeld enthält die Zustandsdaten als Bytefolge. Wenn solche Zustandsdaten vorliegen, kann die Angebotserstellung mit individuellen Merkmalen, wie bspw. Preisen erfolgen. Wenn die Bytefolge leer ist, erfolgt die Angebotserstellung mit neu initialisierten Zustandsdaten ohne Historie.

`reisendenbezug?: REFERENCE`

Dieses Datenfeld enthält den Verweis auf den Reisenden, dessen Zustandsdaten angegeben sind. Es ist abwesend, wenn der Kunde, dessen Zustandsdaten durch den Kauf modifiziert werden sollen, selbst nicht mitreist.

`erweiterungsliste?: ERWEITERUNG[]`

Dieses Datenfeld enthält weitere anwendungsspezifische Merkmale für zukünftige Erweiterungen, bspw. den Anzeigetext zu den im aktuellen Monat gekauften Fahrten, Tarifzonen oder Berechtigungen.

3 Empfehlungen

3.1 Bestückung von Anfragen

ANFRAGE > herausgeber

Hier sollte möglichst die «Organisation_NUMMER» des primären Kundenvertragspartners gemäß VDV KA eingesetzt werden. Wenn der KVP keine solche Nummer besitzt, sollte eine eigene eindeutig abgestimmt und eingesetzt werden. Wenn der KVP unbestimmt ist, sollte hier die Konstante 0 eingesetzt werden.

ANFRAGE > sprache

Hier sollte der «Sprache_CODE» gemäß VDV KA eingesetzt werden, bspw. 58 für Deutsch und 44 für Englisch.

ANFRAGE > kontext

Der Modulkontext dient grundsätzlich der Differenzierung des Einsatzortes und sollte nicht zur Differenzierung des Einsatzzweckes verwendet werden. Dies ist der Anfrageart vorbehalten. Wenn eine Differenzierung des Einsatzortes und die damit verbundene Abstimmung und Aktualisierung zentral verwalteter Modulkontexte nicht nötig sind, sollte hier die Konstante 1 eingesetzt werden.

ANFRAGE > anfrageart

Die Anfrageart sollte ihren Einsatzzweck möglichst sprechend und präzise wiedergeben. Sie kennzeichnet eine konkrete Aufgabenstellung mit den dazu einzubeziehenden Datenfeldern und -strukturen der Anfrage sowie den dazu bereitzustellenden Datenfeldern und -strukturen der Antwort. Bei der Spezifikation von Anfragearten können folgende Fragestellung helfen:

- Werden gegebene Einschränkungen einbezogen? Alternativ könnte auch ein breites oder vorab abgestimmtes Spektrum an ggf. später zu filternden Angeboten ermittelt werden.
- Werden gegebene Reisenden einbezogen? Ein Reisender könnte auch implizit angenommen werden.
- Werden gegebene Reisende ohne konkrete Merkmale einbezogen? Dies geht, wenn nur deren Anzahl von Interesse ist.
- Werden gegebene Reisenden mit personenbezogenen Fahrgastdaten einbezogen? Dies kann für die Personalisierung von Berechtigungen erforderlich sein. Alternativ könnte der Personenbezug auch außerhalb des Tarifrechners vor der Berechtigungsausgabe erfolgen (vgl. Datenfeld personengebunden in AUSGABEDATEN).
- Werden gegebene Berechtigungen einbezogen? Diese könnten auf ihre Gültigkeit bezüglich der gegebenen Verbindungen hin geprüft und/oder als Grundfahrscheine bei der Ermittlung von Angeboten beachtet werden.
- Werden gegebene Berechtigungen als Eingabedaten und/oder Binärdaten einbezogen?
- Werden Nutzungszustandsdaten einbezogen? Diese können für eine Best-Price-Ermittlung oder spezielle Rabattmodelle relevant sein.
- Soll eine Preis- und Produktermittlung auf Grund von Verbindungen und/oder Eingabedaten erfolgen?
- Werden die ggf. zu Verbindungen und Fahrten angegebenen Sollbezüge einbezogen?
- Werden die ggf. zu Verbindungen und Fahrten angegebenen Tarifentfernungen einbezogen?
- Werden die ggf. zu Fahrten und Halten angegebenen Flächenelemente einbezogen?
- Werden die ggf. zu Fahrten angegebenen Linien-, Fahrt- bzw. Fahrzeugdaten einbezogen?
- Werden die zu Halten angegebenen Haltestellen und/oder Geopositionen einbezogen?
- Wird ein Auswahlménú zur Navigation zwischen den ermittelten Angeboten geliefert? Dieses kann bei der gezielten und übersichtlichen Wahl eines von dutzenden Angeboten helfen.
- Wird ein Auswahlménú als Übersicht zu den ermittelten Angeboten geliefert? Dieses kann Anbindungen bereitstellen, über welche bspw. ein Fahrplanauskunftssystem eine generische Tabelle aller Angebote mit entsprechend beschrifteten Spalten darstellen kann.
- Wird ein Auswahlménú zur Unterstützung der Parameterauswahl durch die Kunden für die Bestückung von Reisenden, Berechtigungen und/oder anderen Merkmale einer nachfolgenden Anfrage geliefert? Dieses kann bspw. nur auf die für eine gegebene Verbindung relevanten Tarife und deren Besonderheiten eingehen, um den Aufwand bei der Eingabe des solcher Merkmale zu reduzieren.
- Werden tarifübergreifend kombinierte Angebote ermittelt? Es könnten auch nur tarifinterne Angebote sein.
- Werden die Ticketdaten mit verkaufsermöglichenden Ein-/Ausgabedaten bzw. Erweiterungen geliefert? Angebote könnten auch nur vorläufig/informativ sein und bspw. nur Preise, Preisstufen, Produktnamen, Produkthinweise usw. bereitstellen. Vorläufige Angebote könnten bei einer vorläufigen Best-Price-Ermittlung vorkommen.
- Soll eine auf Nutzungszustandsdaten beruhende Best-Price-Ermittlung abgeschlossen werden? Dies könnte zur Rechnungsstellung nötig werden, bei welcher zuvor vorläufige Angebote abgeschlossen werden müssen.
- Werden die Ticketdaten für die Handyticketausgabe mit Daten für das Einheitslayout sowie die elektronische Berechtigung geliefert?

ANFRAGE > nutzungszustandsliste

Wenn Nutzungszustandsdaten einbezogen werden sollen, erfolgt dies meist im Kontext genau eines Kunden, so dass hier nur die Angabe der Bytefolge seiner Zustandsdaten notwendig ist, nicht aber die der übrigen Merkmale zur Differenzierung von PV, KVP und Kunde.

3.2 Bestückung von Elementen

ELEMENT > nr

Die Kennung eines Elements ist grundsätzlich unter den Beteiligten abzustimmen bzw. bekanntzugeben, da die damit rechnenden Systeme und Komponenten wissen müssen, was diese Kennungen bedeuten. Sofern nötig, werden solche Kennungen durch das RN-Tarifmodul in die Domänen der eingebetteten PV-Tarifmodule überführt.

ELEMENT > domaene

Die Domäne sollte eine konkrete Version einer Abstimmung oder eines Standards referenzieren. Für alle Kennung gilt grundsätzlich, dass diese unter den Beteiligten abzustimmen sind. Bezüglich der Domäne ist auch abzustimmen, an welchen Stellen sie explizit anzugeben ist bzw. implizit eine bestimmte Domäne gilt. Eine Reduktion bzw. Abbildung der in den Quellsystemen bereits eingesetzten Ausprägung auf die fachlich-tariflich relevanten ist damit von Vorteil. Im RN-Tarifmodul kann in jedem Fall eine Abbildung auf die in den eingebetteten PV-Tarifmodul relevanten bzw. bekannten Ausprägungen erfolgen.

3.3 Bestückung von Reisenden

REISENDER > typ

REISENDER > rabattmedienliste

Hier sollte der «Profil_CODE» bzw. «Rabattmedientyp_CODE» gemäß VDV KA eingesetzt werden.

3.4 Bestückung von Eingabedaten

EINGABEDATEN > version

Diese Versionskennung sollte auf die Tarifmodule Bezugnehmen, die diese Eingabedaten bereitstellen sowie unterstützen, bspw. als Auflistung der Versionskennungen/Dateinamen der kompatiblen PV-Tarifmodule. Dadurch könnte insbesondere ein älteres PV-Tarifmodul diese Eingabedaten als kompatibel erkennen.

EINGABEDATEN > pv

Hier sollte die «Organisation_NUMMER» des Produktverantwortlichen gemäß VDV KA eingesetzt werden. Wenn ein Produktverantwortlicher oder Tarif keine solche Nummer besitzt, sollte eine eigene Nummer abgestimmt und eingesetzt werden.

EINGABEDATEN > fkvp

Hier sollte ebenfalls eine «Organisation_NUMMER» gemäß VDV KA eingesetzt werden.

EINGABEDATEN > produktbezeichnung

EINGABEDATEN > gebietsparameter

EINGABEDATEN > nutzerparameter

EINGABEDATEN > zeitparameter

EINGABEDATEN > serviceparameter

EINGABEDATEN > zusatzparameter

Hier sollten die Nummern der Adapter zu Identifikationsparameterwerten eingesetzt werden, damit diese aus der Domäne des RN-Tarifmoduls in die der entsprechenden PV-Tarifmodule übertragen werden können. Tarife, die ihre Produkte nach einer Gattung und einer Preisstufe differenzieren, sollten hierfür die Identifikationsparameter Produktbezeichnung bzw. Gebietsparameter verwenden.

EINGABEDATEN > gueltigkeitsbeginn

EINGABEDATEN > gueltigkeitsende

Hier sollte stets mindestens eine dieser Zeitangaben angegeben werden.

EINGABEDATEN > gueltigkeitsraum

Die Anzahl der hier benötigten Flächen sowie die Rolle der Flächen an den unterschiedlichen Positionen in dieser Auflistung sind anwendungsspezifisch. Beispiele hierfür sind [Start] bei Produkten mit reichweitenbegrenztem Freigaberaum (bspw. Startort und mehrere umliegende Flächen, Kurzstrecke) oder [Start; Ziel; Weg] bei Produkten mit relationsbezogenem Freigaberaum.

3.5 Bestückung von Verbindungen

HALT > haltestelle

Hier sollte als Nummern die bundesweit einheitlichen DHID eingesetzt werden.

LINIENDATEN > linie

LINIENDATEN > variante

LINIENDATEN > betreiber

Hier sollte die «Linie_NUMMER, «Linienvariante_NUMMER» bzw. «Organisation_NUMMER» gemäß VDV KA eingesetzt werden.

FAHRTDATEN > fahrt

FAHRTDATEN > betreiber

Hier sollte die «Fahrt_NUMMER» bzw. «Organisation_NUMMER» gemäß VDV KA eingesetzt werden.

FAHRZEUGDATEN > typ

FAHRZEUGDATEN > fahrzeug

FAHRZEUGDATEN > betreiber

Hier sollte der «FahrzeugTyp_CODE», die «Fahrzeug_NUMMER» bzw. die «Organisation_NUMMER» gemäß VDV KA eingesetzt werden.

3.6 Bestückung von Berechtigungen

BERECHTIGUNG > binaerdaten

Diese Bytefolge sollte ein TLV-Element darstellen, an dessen Typkennung erkannt werden kann, gemäß welcher Kodierung sein Wert interpretiert werden muss.

Für eine unbekannte Kodierung sollte die Typkennung 0x01 eingesetzt werden.

Eine elektronische Berechtigung gemäß VDV KA sollte dagegen die Typkennung 0x02 sowie ein Wert verwenden, der aus der Verkettung der Bytefolgen der TLV-Elemente

- «0x83 – Statischer Teil des Verzeichniseintrags zur Berechtigung»,
- «0x84 – Dynamischer Teil des Verzeichniseintrags zur Berechtigung»,
- «0x85 – Statischer Teil der separaten Berechtigungsdaten»,
- «0x8A – Dynamischer Teil der separaten Berechtigungsdaten»,
- «0xC7 – Infotext der Berechtigung»

sowie der Bytefolge des TLV-Elements der letzten Transaktion

- «0xF1 – Fahrttransaktion»,
- «0xF3 – Sperrtransaktion»,
- «0xF6 – Ausgabetransaktion»,
- «0xF9 – Rückgabetransaktion»,
- «0xFC – Belastungstransaktion» oder
- «0xFD – Einzahlungsbeleg»

besteht.

3.7 Bestückung von Einschränkungen

```
EINSCHRAENKUNG > pvliste  
EINSCHRAENKUNG > fkvpliste  
EINSCHRAENKUNG > bezahlartliste  
EINSCHRAENKUNG > waehrungsliste  
EINSCHRAENKUNG > rabattmedienliste  
EINSCHRAENKUNG > produktbezeichnungsliste  
EINSCHRAENKUNG > gebietsparameterliste  
EINSCHRAENKUNG > nutzerparameterliste  
EINSCHRAENKUNG > zeitparameterliste  
EINSCHRAENKUNG > serviceparameterliste  
EINSCHRAENKUNG > zusatzparameterliste
```

In diesen Filterlisten sollten die Kennungen der Ausprägungen eingetragen werden, die in Reisenden, Ticketdaten bzw. Eingabedaten verwendet werden.

3.8 Bestückung von Ticketdaten

```
TICKETDATEN > rabattmedienliste
```

Hier sollte der «Rabattmedientyp_CODE» gemäß VDV KA eingesetzt werden.

```
BEZAHLDATEN > waehrung  
BEZAHLDATEN > bezahlart  
BEZAHLDATEN > intervall
```

Hier sollte der «Waehrung_CODE», «BezahlArt_CODE» bzw. «Zahlungsintervall_CODE» gemäß VDV KA eingesetzt werden.

Auswahlmenü des Fraunhofer IVI

Technische Spezifikation Datenstruktur und Webserviceschnittstelle

VERTRAULICH

Dateiname:	Auswahlmenü Technische Spezifikation_v1.2.2.docx
Erstellt am:	10.12.2019
Zuletzt geändert am:	19.08.2021
Version:	1.2.2
Ersteller:	Fraunhofer IVI Dresden

Autoren

Rostock, Sebastian / Fraunhofer IVI Dresden

Schlenzka, Michael / Fraunhofer IVI Dresden

Versionen

Version	Bearbeiter	Datum	Bemerkung
1.0.0	Rostock	10.12.2019	Dokument erstellt.
1.0.1	Rostock	12.12.2019	Request und Response ergänzt.
1.0.2	Rostock	13.12.2019	Primitive Datentypen sowie Datenstrukturvarianten ergänzt.
1.0.3	Rostock	16.12.2019	Auswertungslogik ergänzt und Beschreibungen bereinigt.
1.0.4	Rostock	20.12.2019	Anfrage über GET und Wertebereiche in Tabellen ergänzt. Dokumenttitel überarbeitet.
1.0.5	Rostock	10.01.2020	Verteilung ausgelagert. Datenstrukturvarianten korrigiert. Diagramme ergänzt. Datentypen und Datenstrukturen separiert. Gliederung angepasst.
1.0.6	Rostock	14.01.2020	Primitive Datentypen ergänzt.
1.0.7	Rostock	16.01.2020	Einstiegspunkt in die Wechselbeziehungen am Auswahlménú ergänzt. Datenstrukturvariante der Anbindung korrigiert.
1.0.8	Rostock	04.02.2020	Umgang mit Ergänzungen. Konkretisierung des «JOIN» auf Tabellen. Vorgabewerte bei Anwendungen jetzt auch über Verweise möglich. Empfehlungen zu Anbindungsstrukturen.
1.0.9	Rostock	07.02.2020	Datenstrukturvarianten dokumentiert. Kodierung von ANY spezifiziert.
1.0.10	Rostock	20.02.2020	Benennung des Auswahlménús überarbeitet. Hinweis zu unbenutzten Datentypen BOOLEAN und BINARY ergänzt.
1.0.11	Rostock	20.03.2020	Korrektur der ANY-Kodierung. Empfehlung zur Merkmalsbestückung.
1.0.12	Rostock	26.03.2020	Empfehlung zu Anbindungen für Schalter bzw. Zwischenergebnisse.
1.0.13	Rostock	15.04.2020	Anbindung ohne Merkmal und Vorgabewert ergänzt. Empfehlung zu Anbindungen und deren Darstellung.
1.0.14	Rostock	20.04.2020	Beispiele ergänzt.
1.0.15	Rostock	15.07.2020	Empfehlung zur Wiederherstellung von Einstellungen verdeutlicht.
1.1.0	Rostock	16.07.2020	Verzichtbare alternative Datentypen entfernt.
1.1.1	Rostock	30.07.2020	Beschreibungen zu Wechselbeziehungen vereinfacht.
1.1.2	Rostock	21.09.2020	Empfehlungen zur Implementation.
1.1.3	Rostock	15.12.2020	Empfehlungen zu Eigenschaften.
1.2.0	Schlenzka	22.04.2021	Überarbeitung Webserviceschnittstelle.
1.2.1	Rostock	28.07.2021	Ergänzung von Empfehlungen. Zeitonenverschiebungen ignorieren.
1.2.2	Rostock, Schlenzka	19.08.2021	Ergänzung der Webserviceschnittstelle bzgl. Antwortstrukturen.

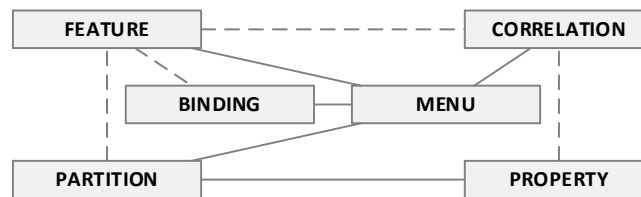
Inhalt

1	Einleitung	5
2	Spezifikation.....	7
2.1	Webserviceschnittstelle	7
2.2	Datenstrukturdiagramm	8
2.3	Strukturierte Datentypen	8
2.3.1	REQUEST	8
2.3.2	RESPONSE.....	9
2.3.3	MENU	9
2.3.4	BINDING.....	10
2.3.5	FEATURE.....	11
2.3.6	PARTITION.....	12
2.3.7	PROPERTY	12
2.3.8	CORRELATION.....	14
2.4	Primitive Datentypen.....	15
2.4.1	UUID.....	15
2.4.2	NAME	15
2.4.3	ROLE.....	15
2.4.4	TEXT	15
2.4.5	ICON.....	16
2.4.6	LANG.....	16
2.4.7	BINARY	16
2.4.8	PATTERN.....	16
2.4.9	DURATION	16
2.4.10	DATETIME.....	16
2.4.11	DATE	16
2.4.12	TIME	16
2.4.13	ANY	16
2.4.14	STRING	17
2.4.15	INTEGER.....	17
2.4.16	BOOLEAN	17
2.4.17	REFERENCE.....	17
2.4.18	FEATURE_REF	17
2.4.19	PARTITION_REF.....	17
2.4.20	PROPERTY_REF.....	17
2.4.21	CORRELATION_REF.....	18
3	Empfehlung.....	19
3.1	Anbindungen.....	19
3.1.1	Grundlagen	19
3.1.2	Menüanbindung	19
3.1.3	Anzeigeanbindung	20
3.1.4	Eingabeanbindung	20
3.1.5	Ausgabeanbindung	21
3.2	Implementation	22
3.2.1	Grundlagen.....	22

VERTRAULICH

3.2.2	Ermittlung von Eigenschaften	22
3.2.3	Abbildung von Ausprägungen	22
3.2.4	Abbildung von Wechselbeziehungen	23
3.2.5	Auswertung von Wechselbeziehungen	23

1 Einleitung



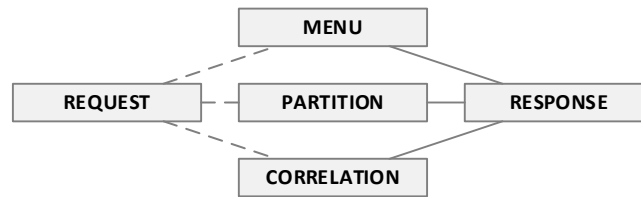
Das Auswahlmenü (MENU) ist eine statische JSON-Datenstruktur, die einem Client zur Interpretation und Auswertung bereitgestellt wird und den Nutzer bei der gezielten Auswahl eines Datensatzes unterstützen soll. Einem Client soll es damit ermöglicht werden, neue Auswahlprozesse über ein aktualisiertes Auswahlmenü zu realisieren, ohne den Client dafür umprogrammieren zu müssen. Es entstand als Tarifmenü zur Auswahl eines Produktangebots und hatte wegen dieses ersten Anwendungsfalls einen tariflichen Bezug. Dieser hatte jedoch keinen Einfluss auf die spezifizierten Inhalte, sodass der Name allgemeiner gewählt wurde.

Der Auswahlprozess beschreibt die Bestückung bestimmter Merkmale (FEATURE) mit konkreten Ausprägungen, deren Eigenschaften (PROPERTY) wechselseitig voneinander abhängen können. Ein Beispiel hierfür ist die Auswahl von Start- und Zieltarifzonen nach der Preisstufe. Wenn für das Merkmal Preisstufe die Eigenschaft Verbundgebiet gewählt wurde, sind die Merkmale Startzone und Zielzone implizit bestückt und bedürfen keiner Auswahl. Für eine geringere Preisstufe könnten dagegen Start- und ggf. auch Zielzone abgefragt werden, wobei hier nur die zur gewählten Preisstufe passenden Tarifzonen als Eigenschaften der Merkmale Startzone und Zielzone wählbar sind.

Die Auswertung eines Auswahlmenüs bildet die Grundlage eines gerichteten azyklischen Graphen aus Eingabe- und Auswahlritten ab. Ein solcher Graph hat meist genau einen Startschritt. In jedem der Schritte wird die Ausprägung eines bestimmten Merkmals manuell oder automatisch bestimmt. Jeder Navigationslauf durch den Graphen führt damit zu einer Merkmalsbestückung, die bspw. ein Produktangebot identifiziert. Die auf dem Navigationspfad bis zu einem bestimmten Schritt zusammengestellte Merkmalsbestückung entscheidet über die in den nachfolgenden Schritten eingebbaren bzw. auswählbaren Ausprägungen. Ob und in welcher Weise diese Abfolge von Schritten durch das Auswahlmenü und/oder den Client vorgegeben wird, ist grundsätzlich unter den Beteiligten abzustimmen. Hierbei ist insbesondere die Anbindung (BINDING) der Merkmale an die Datenkanäle und Steuerelemente des Clients sowie deren Struktur abzustimmen. Siehe dazu auch die Empfehlungen in Kapitel 3.

Die Menge aller in einem Auswahlmenü verwendeten Eigenschaften (PROPERTY) ist zu deren Priorisierung sowie zur Wiederverwendung von Wertebereichen in geordnete Partitionen (PARTITION) unterteilt. Die Wechselbeziehungen (CORRELATION) zwischen den Eigenschaften der Ausprägungen der Merkmale (FEATURE) können als Tabelle verstanden werden, in welcher jede Spalte einem Merkmal entspricht und in welcher jede Zeile für eine zulässige Merkmalsbestückung steht. Die Ermittlung der in einem Schritt für ein Merkmal zulässigen Ausprägungen entspricht der Zusammenstellen der in der entsprechenden Spalte vorkommenden Ausprägungen. Die Bestückung eines Merkmals führt zur Reduktion der Tabelle auf die Zeilen, in denen die bestückte Ausprägung in der entsprechenden Spalte enthalten ist.

Neben der vollständigen Bereitstellung als Datei kann ein Auswahlmenü auch über eine entsprechende Webserviceschnittstelle in nachladbarer Form bereitgestellt werden, was im Zusammenhang mit der Verwendung von Zwischenspeichern im Client sinnvoll sein kann.



Eine Anfrage (REQUEST) an eine solche Schnittstelle kann über Kennungen nach den nachladbaren Inhalten fragen, welche dann in der Antwort (RESPONSE) enthalten sind.

Die primitiven Datentypen BINARY und BOOLEAN werden in dieser Spezifikation zwar beschrieben, aber von den anderen Datentypen nicht direkt verwendet. Sie finden nur indirekt im Datentyp ANY Anwendung und sind damit vorrangig zur erleichterten Wiederverwendung in den auf dieser Spezifikation aufbauenden vorgesehen.

Die im Folgenden verwendete Notation zur Definition der Datenstrukturen gibt zu jedem Datenfeld einen Namen und einen ggf. zusammengesetzten Datentyp an. Vor letzterem steht stets ein Doppelpunkt. Bei optionalen Datenfeldern folgt dem Namen ein Fragezeichen. Bei Auflistungstypen (JSON array) folgt dem Datentyp der Listenelemente eine öffnende sowie eine schließende eckige Klammer. Bei Abbildungstypen (JSON object) folgt dem Datentyp der Werte eine öffnende geschweifte Klammer, der Datentyp der Schlüssel sowie eine schließende geschweifte Klammer. Sofern nur bestimmte Zusammenstellungen von Datenfeldern als Datenstrukturvarianten zulässig sind, werden die Sequenzen ihrer ggf. optionalen Datenfelder mit Leerzeichen separiert und in runde Klammern eingeschlossen aufgelistet.

2 Spezifikation

2.1 Webserviceschnittstelle

Die Webserviceschnittstelle nutzt als Request und Response die in den nachfolgenden Abschnitten spezifizierten Datenstrukturen REQUEST bzw. RESPONSE als Nutzdaten des HTTP POST.

Neben der Anfragemethode HTTP POST kann ein Request auch über HTTP GET gestellt werden. Damit kann bspw. das Puffern der Response durch den Cache des Browsers erfolgen. Im Fehlerfall nutzt diese Anfragemethode eine einfache Zeichenkette sowie die HTTP Fehlernummern. Die Datenstrukturen der Response unterscheiden sich je nach Anfragetyp:

`/menus?select={DATETIME}`

Diese Anfrage enthält den Zeitpunkt (DATETIME) zur Ermittlung anwendbarer Auswahlmenüs. Die Response enthält die Liste aller ab diesem Zeitpunkt gültigen Auswahlmenüs (MENU[]), aufsteigend geordnet nach ihrem Gültigkeitsbeginn und je Gültigkeitsbeginn nur in der höchsten Revision und ggf. als reduzierte Kopfdatensätze. Eine Anfrage der ab Jahresbeginn anwendbaren Auswahlmenüs erfolgt bspw. mit: `/menus?select=2021-01-01T00:00:00Z`.

`/menus/{UUID}`

Diese Anfrage enthält den Identifikator (UUID) des angefragten Auswahlmenüs (MENU). Partitionen und Wechselbeziehungen können hier nur als reduzierte Kopfdatensätze geliefert werden.

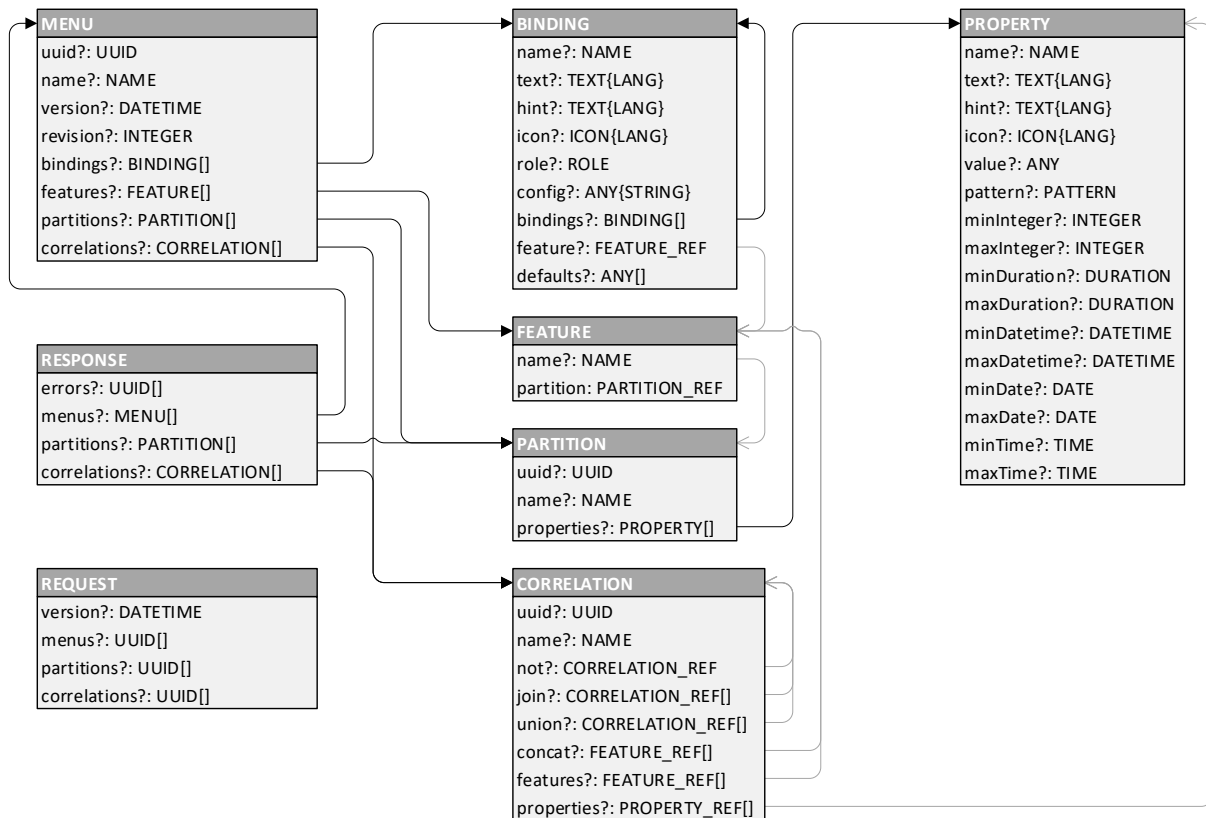
`/partitions/{UUID}`

Diese Anfrage enthält den Identifikator (UUID) der angefragten Partition (PARTITION). Die Response enthält den Volldatensatz der angefragten Partition oder die Fehlermarkierung.

`/correlations/{UUID}`

Diese Anfrage enthält den Identifikator (UUID) der angefragten Wechselbeziehung (CORRELATION). Die Response enthält den Volldatensatz der angefragten Wechselbeziehung oder die Fehlermarkierung.

2.2 Datenstrukturdiagramm



Diese an UML-Klassendiagramme angelehnte Übersicht zeigt alle Datenstrukturen, wobei ausgefüllte Pfeile für Kompositionen und die übrigen Pfeile für Aggregationen stehen. Eine Erweiterung der in diesem Dokument spezifizierten Datenstrukturen um eigene anwendungsspezifische Datenfelder darf die hier spezifizierte Verarbeitung nicht beeinträchtigen.

2.3 Strukturierte Datentypen

2.3.1 REQUEST

Diese Datenstruktur beschreibt den Webservice-Request zum Nachladen von Auswahlmenüs bzw. deren Bestandteilen.

version?: DATETIME

Dieses Datenfeld enthält den Zeitpunkt zur Ermittlung anwendbarer Auswahlmenüs. Seine Zeitonenverschiebung wird ignoriert. Die Response enthält dazu alle ab diesem Zeitpunkt gültigen Auswahlmenüs, aufsteigend geordnet nach ihrem Gültigkeitsbeginn und je Gültigkeitsbeginn nur in der höchsten Revision.

menus?: UUID[]

Dieses Datenfeld enthält die Identifikatoren der angefragten Auswahlmenüs.

partitions?: UUID[]

Dieses Datenfeld enthält die Identifikatoren der angefragten Partitionen.

correlations?: UUID[]

Dieses Datenfeld enthält die Identifikatoren der angefragten Wechselbeziehungen.

Es gibt folgende anwendungsspezifische Datenstrukturvarianten:

(version)

Ein Request zur Ermittlung anwendbarer Auswahlmenüs kann nicht zum Nachladen verwendet werden.

(menus? partitions? correlations?)

Ein Request zum Nachladen von Volldatensätzen kann nicht nach anwendbaren Auswahlmenüs fragen.

2.3.2 RESPONSE

Diese Datenstruktur beschreibt die Webservice-Response mit den zur Anfrage ermittelten Identifikatoren und Datensätzen. Jeder darüber gelieferte Datensatz ist grundsätzlich bis zu den eingebetteten Datensätzen aufgelöst, die über einen eigenen Identifikator (UUID) verfügen und damit als reduzierte Kopfdatensätze vorliegen können. Der Volldatensatz zu einem solchen Kopfdatensatz kann dann separat angefragt und nachgeladen werden.

errors?: UUID[]

Dieses Datenfeld enthält die Identifikatoren angefragter Datensätze, die nicht existieren.

menus?: MENU[]

Dieses Datenfeld enthält die angefragten Auswahlmenüs.

partitions?: PARTITION[]

Dieses Datenfeld enthält die angefragten Partitionen.

correlations?: CORRELATION[]

Dieses Datenfeld enthält die angefragten Wechselbeziehungen.

Es gibt folgende anwendungsspezifische Datenstrukturvarianten:

(errors)

Eine Response mit ungültigen Identifikatoren enthält nur diese.

(menus? partitions? correlations?)

Eine Response ohne ungültige Identifikatoren kann nur Datensätze enthalten.

2.3.3 MENU

Diese Datenstruktur kodiert ein Auswahlmenü, das ggf. ab einem bestimmten Zeitpunkt anwendbar ist. Seine intern gemeinsam genutzten bzw. wiederverwendeten Komponenten werden in entsprechenden Auflistungen verwaltet und über deren Listenposition referenziert. Die Datenstruktur kann für eine Datei eingesetzt werden, um bspw. einen Client mit Daten für den Katalogverkauf zu versorgen. Sie kann aber auch für dynamisch generierte Auswahlmenüs genutzt werden, bei welchen wegen der kontextspezifischen Erzeugung die Versionsmerkmale dann grundsätzlich entfallen können.

uuid?: UUID

Dieses Datenfeld enthält den Identifikator des Auswahlmenüs zum Nachladen seines Volldatensatzes.

name?: NAME

Dieses Datenfeld enthält die informative Bezeichnung des Auswahlmenüs.

version?: DATETIME

Dieses Datenfeld enthält den Gültigkeitsbeginn des Auswahlmenüs als Zeitpunkt. Dessen Zeitonenverschiebung wird ignoriert. Wenn es nicht angegeben ist, gibt es keine zeitliche Einschränkung der Anwendbarkeit. Andernfalls ist zu einem gegebenen Zeitpunkt stets das Auswahlmenü mit dem jüngsten Gültigkeitsbeginn anzuwenden.

revision?: INTEGER

Dieses Datenfeld enthält die Revisionsnummer des Auswahlmenüs, welche zur Unterscheidung von Auswahlmenüs mit gleicher zeitlicher Einschränkung benötigt wird. Wenn es mehrere solche Auswahlmenüs gibt, ist das mit der größten Revisionsnummer anzuwenden. Die Revisionsnummer kann damit als überschreibende Korrekturversion verstanden werden.

bindings?: BINDING[]

Dieses Datenfeld enthält die Auflistung der Anbindungen an die bedienten Datenkanäle und/oder Steuerelemente. Die Rollen und Strukturen dieser Anbindungen muss grundsätzlich unter den Beteiligten abgestimmt werden. Ein Client muss alle Anbindungen mit ihm unbekannten Rollen ignorieren (siehe dazu auch Kapitel 3).

features?: FEATURE[]

Dieses Datenfeld enthält die Auflistung aller Merkmale. Verweise auf diese Merkmale beziehen sich auf deren Positionen in dieser Liste.

partitions?: PARTITION[]

Dieses Datenfeld enthält die Auflistung aller Partitionen. Verweise auf diese Partitionen beziehen sich auf deren Positionen in dieser Liste.

correlations?: CORRELATION[]

Dieses Datenfeld enthält die Auflistung aller adressierbaren Wechselbeziehungen zwischen den Eigenschaften der Ausprägungen der Merkmale. Verweise auf diese Wechselbeziehungen beziehen sich auf deren Positionen in dieser Liste. Die erste Wechselbeziehung in dieser Liste steht implizit für die Wurzel des azyklischen Graphs aller verwendeten Wechselbeziehungen und wird daher niemals von anderen Wechselbeziehungen referenziert.

Es gibt folgende anwendungsspezifische Datenstrukturvarianten:

(name? bindings features partitions correlations)

Ein (dynamisch) kontextspezifisch erzeugtes Auswahlmenü besitzt weder Identifikator noch Versionsmerkmale.

(name? version revision bindings features partitions correlations)

Ein (statisch) versioniert hinterlegtes Auswahlmenü besitzt Versionsmerkmale und Komponenten.

(uuid name? version revision)

Ein (statisch) nachladbares Auswahlmenü besitzt Identifikator und Versionsmerkmale, aber keine Komponenten. Es liegt damit nur als Kopfdatensatz vor.

(uuid name? version revision bindings features partitions correlations)

Ein (statisch) nachgeladenes Auswahlmenü besitzt Identifikator, Versionsmerkmale und Komponenten.

2.3.4 BINDING

Diese Datenstruktur beschreibt eine ggf. strukturierte sowie ggf. beschriftete Anbindung eines Datenkanals oder Steuerelements an ein Merkmal, dessen konkrete oder mögliche Ausprägung über diesen Datenkanal ausgelesen und/oder bestückt bzw. über dieses Steuerelement angezeigt und ggf. angepasst werden kann.

name?: NAME

Dieses Datenfeld enthält die informative Bezeichnung der Anbindung.

text?: TEXT{LANG}

Dieses Datenfeld enthält den ggf. sprachspezifischen Anzeigetext für das Steuerelement.

hint?: TEXT{LANG}

Dieses Datenfeld enthält den ggf. sprachspezifischen Hinweistext für das Steuerelement.

icon?: ICON{LANG}

Dieses Datenfeld enthält das ggf. sprachspezifische Piktogramm für das Steuerelement.

`config?: ANY{STRING}`

Dieses Datenfeld enthält zusätzliche Konfigurationsdaten für den Datenkanal oder das Steuerelement. Diese sind grundsätzlich unter den Beteiligten abzustimmen. Hier können bspw. die Art des Steuerelements, dessen Anzeigestil oder auch Kriterien zur Steuerung seiner Größe und Position angegeben werden. Ein Client muss alle ihm unbekannten Datenfelder der Konfigurationsdaten ignorieren.

`role?: ROLE`

Dieses Datenfeld nennt die Rolle bzw. Kennung des Datenkanals oder Steuerelements, an welcher der Client den Verwendungszweck sowie die Notwendigkeit der Anbindung erkennen kann. Der Verwendungszweck stellt dazu meist konkrete Anforderungen an die Rollen und Strukturen eingebetteter Anbindungen, welche durch den Client gegen ein entsprechendes Schema geprüft werden sollten.

`feature?: FEATURE_REF`

Dieses Datenfeld enthält den Verweis auf das angebundene Merkmal, auf dessen Ausprägung über den Datenkanal bzw. das Steuerelement zugegriffen werden kann.

`defaults?: ANY[]`

Dieses Datenfeld enthält die nach Priorität absteigend geordnete Auflistung der Vorgabewerte. Diese werden zur automatischen Bestückung der Ausprägung eines angeordneten Merkmals eingesetzt. Wenn zu diesem Merkmal mehrere zulässige Ausprägungen ermittelt wurden, wird grundsätzlich der höchstpriorisierte Vorgabewert eingesetzt. Wenn die Anbindung auf kein Merkmal verweist, kann der Anbindung über einen einzigen Vorgabewert eine von den Wechselbeziehungen unabhängige Konstante zugewiesen werden.

`bindings?: BINDING[]`

Dieses Datenfeld enthält die Auflistung der eingebetteten Anbindungen, welche bspw. die im komponierten Steuerelement enthaltenen und damit untergeordneten Steuerelement bestücken. Wenn das Steuerelement eine Liste darstellt, legt die Reihenfolge der hier aufgeführten Anbindungen grundsätzlich auch die Reihenfolge der Listeneinträge fest. Solche Listenelemente benötigen dann meist keine Rolle oder Kennung. Wenn diese Auflistung angegeben ist, gilt die Anbindung als einbettend.

Es gibt folgende anwendungsspezifische Datenstrukturvarianten:

`(name? text? hint? icon? role? config? defaults? bindings?)`

Eine Anbindung ohne Verweis auf ein Merkmal kann über ihren Vorgabewert eine Konstante angeben.

`(name? text? hint? icon? role? config? feature defaults? bindings?)`

Eine Anbindung mit Verweis auf ein Merkmal kann dafür Vorgabewerte angeben.

2.3.5 FEATURE

Diese Datenstruktur beschreibt ein Merkmal, dessen im Client verwaltete Ausprägung durch einen konkreten Wert sowie eine dazu passende Eigenschaft beschrieben wird.

Grundsätzlich kann über Wechselbeziehungen zu den Eigenschaften in den konkreten Ausprägungen anderer Merkmale auf die in der Partition eines bestimmten Merkmals aufgeführten Eigenschaften geschlossen werden, die in der Ausprägung dieses Merkmals eingesetzt werden dürfen. In Tabellen ausgedrückt bedeutet dies, dass die logische Tabelle aller Wechselbeziehungen bezüglich der bereits gewählten Merkmale gefiltert wird und anschließend alle in einer bestimmten Spalte der gefilterten Tabelle noch vorkommenden Eigenschaften zusammengestellt werden. Diese abgeleitete Auflistung von Eigenschaften liefert aber nicht nur die möglichen Kandidaten zur Bestückung des Merkmals, sondern auch mit ihrem höchstpriorisierten Element die indirekt bestückte Eigenschaft, welche bspw. zum Lesen von Ausgabe- und Anzeigedaten eingesetzt werden kann. Diese Auflistung dieser Kandidaten kann daher nie leer sein.

Wenn die Bestückung der Ausprägung des Merkmals über einen (ein)gegebenen Wert erfolgen soll, müssen dazu die bezüglich ihrer Filterkriterien zum Wert passenden Eigenschaften aus der oben beschriebenen Auflistung von Kandidaten bestimmt werden. Wenn mehrere solche Eigenschaften existieren, ist die höchstpriorisierte einzusetzen. Wenn keine existieren, ist der Wert zur Bestückung des Merkmals unzulässig.

Wenn die Bestückung des Merkmals über eine gegebene Eigenschaft erfolgen soll, muss diese in der oben beschriebenen Auflistung von Kandidaten enthalten und damit zulässig sein. Wenn dem so ist, wird als Wert in der Ausprägung des Merkmals der an der Eigenschaft angegebene Wert eingesetzt. Wenn kein solcher Wert angegeben ist, gilt der Wert des Merkmals als unbestimmt.

name?: NAME

Dieses Datenfeld enthält die informative Bezeichnung des Merkmals.

partition: PARTITION_REF

Dieses Datenfeld enthält den Verweis auf die Partition, welche alle für die Ausprägung dieses Merkmal grundsätzlich möglichen Eigenschaften auflistet.

2.3.6 PARTITION

Diese Datenstruktur beschreibt eine Partition oder Halbordnung der im Auswahlmenü zur Bestückung von Merkmalen bereitgestellten Eigenschaften.

uuid?: UUID

Dieses Datenfeld nennt den Identifikator der Partition zum Nachladen ihres Volldatensatzes.

name?: NAME

Dieses Datenfeld enthält die informative Bezeichnung der Partition.

properties?: PROPERTY[]

Dieses Datenfeld enthält die nach Priorität absteigend geordnete Auflistung der Eigenschaften, die zur Bestückung eines Merkmals eingesetzt werden können. Verweise auf diese Eigenschaften beziehen sich auf deren Positionen in dieser Liste.

Es gibt folgende anwendungsspezifische Datenstrukturvarianten:

(uuid name?)

Eine nachladbare Partition besitzt einen Identifikator, aber keine Eigenschaften. Sie liegt damit nur als Kopfdatensatz vor.

(uuid name? properties)

Eine nachgeladene Partition besitzt Identifikator und Eigenschaften.

(name? properties)

Eine nicht nachladbare Partition besitzt Eigenschaften, aber keinen Identifikator.

2.3.7 PROPERTY

Diese Datenstruktur beschreibt die ggf. beschriftete Eigenschaft eines Werts, welche zusammen mit diesem Wert die konkrete Ausprägung eines Merkmals bilden kann. Für jede Eigenschaft kann grundsätzlich entschieden werden, ob ein gegebener Wert diese erfüllt. Dazu können Eigenschaften einen Wertebereich zum Abgleich von Zahlen- oder Zeitwerten, einen regulären Ausdruck zum Abgleich von Textwerten oder einen konkreten Wert zum direkten Abgleich angeben.

Wenn eine Eigenschaft neben den Filterkriterien einen konkreten Wert angibt, ist dieser grundsätzlich auch beim Lesen des Werts des auf diese Eigenschaft in seiner Ausprägung verweisenden Merkmals einzusetzen, auch wenn zur Bestückung dieser Ausprägung ein anderer Wert verwendet wurde. Andernfalls ist der gegebene und wegen des Abgleichs mit den Filterkriterien in eine Text-, Zahlen- oder Zeitwert überführte Wert einzusetzen.

name?: NAME

Dieses Datenfeld enthält die informative Bezeichnung der Eigenschaft.

text?: TEXT{LANG}

Dieses Datenfeld enthält den ggf. sprachspezifischen Anzeigetext.

hint?: TEXT{LANG}

Dieses Datenfeld enthält den ggf. sprachspezifischen Hinweistext.

icon?: ICON{LANG}

Dieses Datenfeld enthält das ggf. sprachspezifische Piktogramm.

value?: ANY

Dieses Datenfeld enthält den konkreten Wert der Eigenschaft, welche zum direkten Abgleich sowie als priorisierter Wert der Ausprägung eines Merkmals eingesetzt wird. Wenn das Datenfeld nicht angegeben ist, gilt der Wert der Eigenschaft als unbestimmt.

pattern?: PATTERN

Dieses Datenfeld enthält den regulären Ausdruck zum Abgleich mit Textwerten.

minInteger?: INTEGER

Dieses Datenfeld enthält den inklusiven Beginn des Wertebereichs zum Abgleich mit Zahle.

maxInteger?: INTEGER

Dieses Datenfeld enthält das exklusive Ende des Wertebereichs zum Abgleich mit Zahle.

minDuration?: DURATION

Dieses Datenfeld enthält den inklusiven Beginn des Wertebereichs zum Abgleich mit Zeitspannen.

maxDuration?: DURATION

Dieses Datenfeld enthält das exklusive Ende des Wertebereichs zum Abgleich mit Zeitspannen.

minDatetime?: DATETIME

Dieses Datenfeld enthält den inklusiven Beginn des Wertebereichs zum Abgleich mit Zeitpunkten.

maxDatetime?: DATETIME

Dieses Datenfeld enthält das exklusive Ende des Wertebereichs zum Abgleich mit Zeitpunkten.

minDate: DATE

Dieses Datenfeld enthält den inklusiven Beginn des Wertebereichs zum Abgleich mit Kalendertagen.

maxDate: DATE

Dieses Datenfeld enthält das exklusive Ende des Wertebereichs zum Abgleich mit Kalendertagen.

minTime: TIME

Dieses Datenfeld enthält den inklusiven Beginn des Wertebereichs zum Abgleich mit Tageszeiten.

maxTime: TIME

Dieses Datenfeld enthält das exklusive Ende des Wertebereichs zum Abgleich mit Tageszeiten.

Es gibt folgende anwendungsspezifische Datenstrukturvarianten:

(name? text? hint? icon?)

Die Eigenschaft ohne Wert und ohne Filterkriterien kann nicht mit einer Eingabe abgeglichen werden.

(name? text? hint? icon? value)

Eine Eigenschaft mit konkretem Wert kann ohne weitere Filterkriterien mit einer Eingabe abgeglichen werden.

(name? text? hint? icon? value? pattern)

(name? text? hint? icon? value? minInteger maxInteger)

(name? text? hint? icon? value? minDuration maxDuration)

(name? text? hint? icon? value? minDatetime maxDatetime)

(name? text? hint? icon? value? minDate maxDate)

(name? text? hint? icon? value? minTime maxTime)

Eine Eigenschaft mit Filterkriterien hat diese nicht Datentypübergreifend.

2.3.8 CORRELATION

Diese Datenstruktur beschreibt Wechselbeziehungen zwischen Eigenschaften in Ausprägungen von Merkmalen. Diese Beziehungen können in Form einer konkreten Tabelle oder eines Terms angegeben werden, wobei letzterer nur eine speicheroptimierte Form einer Tabelle ist.

In einer Tabelle von Wechselbeziehungen stehen die Spalten für die Merkmale und die Zeilen für die zulässigen Kombinationen von Eigenschaften in den Ausprägungen dieser Merkmale. Für die in der Tabelle nicht genannten Merkmale gelten dazu keine Einschränkungen. Die zur Bestückung eines bestimmten Merkmals zulässigen Eigenschaften ergeben sich dann aus dem Filtern der Zeilen bezüglich der in anderen Merkmalen bereits bestückten Eigenschaften.

uuid?: UUID

Dieses Datenfeld nennt den Identifikator der Wechselbeziehung zum Nachladen seines Volldatensatzes.

name?: NAME

Dieses Datenfeld enthält die informative Bezeichnung der Wechselbeziehung.

not?: CORRELATION_REF

Dieses Datenfeld enthält einen Verweis auf die zu negierende Tabelle. Die damit beschriebene Wechselbeziehung folgt aus der Differenz zwischen allen möglichen Kombinationen (kartesisches Produkt) und denen in der zu negierenden Tabelle aufgeführten. Damit können bspw. Entscheidungen als «union(join(IF, THEN), join(not(IF), ELSE))» abgebildet werden.

join?: CORRELATION_REF[]

Dieses Datenfeld enthält die Auflistung der Verweise auf zu verschmelzenden Tabelle. Die damit beschriebene Wechselbeziehung folgt aus dem Verschmelzen der Tabellen mit gemeinsamen Spalten über einen «NATURAL JOIN» sowie der Tabellen ohne gemeinsame Spalten mit einem «CROSS JOIN».

union?: CORRELATION_REF[]

Dieses Datenfeld enthält die Auflistung der Verweise auf zu vereinenden Tabelle. Die damit beschriebene Wechselbeziehung folgt aus der Vereinigung der Tabellen mit einer «UNION».

concat?: FEATURE_REF[]

Dieses Datenfeld enthält die Auflistung der Verweise auf die zu verkettenden Merkmale. Die damit beschriebene Wechselbeziehung entspricht einer Tabelle, deren Spalteninhalte durch die Auflistungen der Eigenschaft in den Partitionen der entsprechenden Merkmale gegeben sind. Die Tabelle besitzt damit das Minimum der Längen dieser Auflistungen als Zeilenanzahl.

features?: FEATURE_REF[]

Dieses Datenfeld enthält die Auflistung der Verweise auf die Merkmale, die als Spalten der konkret angegebenen Tabelle eingesetzt werden. Ein Merkmal darf hier höchstens zweimal referenziert werden, wobei beide Verweise dann direkt aufeinander folgen müssen und die inklusiven Von- und Bis-Spalte des Merkmals angeben (Wertebereichsangebe). Diese Form der effizienten Kodierung von Wertebereichen ändert nichts daran, dass jedes Merkmal einzigartig ist und in der logischen Tabelle aller Wechselbeziehungen genau eine logische Spalte darstellt.

properties?: PROPERTY_REF[]

Dieses Datenfeld enthält die Auflistung der Verweise auf die Eigenschaften in den Zeilen der konkret angegebenen Tabelle. Diese Auflistung besteht aus der Verkettung der Zeilen der Tabelle, wobei jede Zeile aus den Verweisen

auf die Eigenschaften für die jeweiligen Spalten besteht. Die Anzahl der Verweise in dieser Auflistung ist damit stets ein ganzzahliges Vielfaches der Spaltenanzahl. Wenn über ein Paar aus Von- und Bis-Spalte ein Wertebereich angegeben wird, enthält dieser die Eigenschaften mit den Verweisen im angegebenen Bereich, inklusive der Bereichsgrenzen.

Es gibt folgende anwendungsspezifische Datenstrukturvarianten:

(uuid name? features)

Eine nachladbare Tabelle besitzt Identifikator und Spalten, aber keine Zeilen. Sie liegt damit nur als Kopfdatensatz vor.

(uuid name? features properties)

Eine nachgeladene Tabelle besitzt Identifikator, Spalten und Zeilen.

(name? features properties)

Eine nicht nachladbare Tabelle besitzt Spalten und Zeilen, aber keinen Identifikator.

(name? not)

(name? join)

(name? union)

(name? concat)

Eine als Term angegebene Tabelle verwendet dazu stets nur einen konkreten Operator.

2.4 Primitive Datentypen

2.4.1 UUID

Dieser Datentyp beschreibt einen universellen einzigartigen Identifikator als **STRING**, welcher die konkrete Befüllung eines Datensatzes kennzeichnet (versions-/inhaltsbezogen, ggf. streuwertbasiert). Dieser Identifikator wird beim Nachladen von Datensätzen verwendet und kann in Zwischenspeichern zur Erkennung der Notwendigkeit einer Aktualisierung eingesetzt werden.

2.4.2 NAME

Dieser Datentyp beschreibt die informative Bezeichnung eines Datensatzes als **STRING**, welche bei der Datenpflege sowie bei Abstimmungen unter den Beteiligten nützlich sein kann. Diese Bezeichnung unterliegt grundsätzlich keiner Abstimmung und ist damit im Client grundsätzlich zu ignorieren.

2.4.3 ROLE

Dieser Datentyp beschreibt die Kennung einer Anbindung als **STRING**, welche als technischer Identifikator einen Einsatzort oder Verwendungszweck kennzeichnet und damit grundsätzlich unter den Beteiligten abgestimmt werden muss.

2.4.4 TEXT

Dieser Datentyp beschreibt einen Anzeige- oder Hinweistext als **STRING**, welcher zur Beschriftung eines Steuerelements eingesetzt werden kann. Ob dieser bspw. Zeilenumbrüche oder HTML-Auszeichnungen enthalten darf und welche Längenbeschränkung für ihn gelten, ist unter den Beteiligten abzustimmen.

2.4.5 ICON

Dieser Datentyp beschreibt die Kennung einer Grafik als STRING, welche als Piktogramm zur bildhaften Beschriftung eines Steuerelements eingesetzt werden kann. Welche Kennungen an welchen Stellen bzw. für welche Steuerelemente verfügbar sind, ist unter den Beteiligten abzustimmen.

2.4.6 LANG

Dieser Datentyp beschreibt die Kennung einer Sprache als STRING, welche zur Differenzierung sprachspezifischer Texte oder Grafiken eingesetzt wird. Beispiele hierfür sind "de" für Deutsch, "en" für Englisch und "\$" für Default.

2.4.7 BINARY

Dieser Datentyp beschreibt eine beliebige Bytefolge in JSON (vgl. XML hexBinary).

2.4.8 PATTERN

Dieser Datentyp beschreibt einen Regulären Ausdruck als STRING.

2.4.9 DURATION

Dieser Datentyp beschreibt eine Zeitspanne als STRING (vgl. XML duration).

2.4.10 DATETIME

Dieser Datentyp beschreibt einen Zeitpunkt als STRING (vgl. DATE, TIME, XML dateTime).

2.4.11 DATE

Dieser Datentyp beschreibt einen gregorianischen Kalendertag als STRING (vgl. XML date).

2.4.12 TIME

Dieser Datentyp beschreibt eine sekundengenaue Tageszeit als STRING (vgl. XML time).

2.4.13 ANY

Dieser Datentyp beschreibt einen beliebigen Wert als typisierte Zeichenkette (string), als Auflistung (array) solcher Werte oder als Abbildung (object) auf solche Werte. Die Kodierung der typisierten Zeichenkette ist dem Dokument «PKM Tarifrrechner Grundlagen_v1.0.x» entnommen. Dort ist spezifiziert, dass ein primitiver Werte grundsätzlich in seine XML-Textdarstellung mit vorangestellter Typkennung überführt wird, mit besonderer Behandlung benannter Konstanten. Es werden die folgenden Typkennungen eingesetzt:

- U für die Konstante null, d.h. "U"
- T für die Konstante true, d.h. "T"
- F für die Konstante false, d.h. "F"
- I für Ganzzahlen, bspw. "I132", "I-456"
- S für Zeichenketten, bspw. "SDas ist ein Text!"
- B für Bytefolgen in Hexadezimaldarstellung, bspw. "B1F08154711BAADF00D"
- D für Datum, Uhrzeit und Zeitpunkt, bspw. "D09:30:00", "D2020-01-06"
- P für Zeitspannen bspw. "PP1M", "P-PT1H"

2.4.14 STRING

Dieser Datentyp beschreibt eine beliebige Zeichenkette in JSON (vgl. XML string).

2.4.15 INTEGER

Dieser Datentyp beschreibt eine beliebige Ganzzahl in JSON (vgl. XML long).

2.4.16 BOOLEAN

Dieser Datentyp beschreibt einen Wahrheitswert in JSON (vgl. XML boolean).

2.4.17 REFERENCE

Dieser Datentyp beschreibt einen Verweis auf einen Datensatz als INTEGER, welcher die 1-basierte Position des Datensatzes in der ihn verwaltenden Auflistung angibt.

2.4.18 FEATURE_REF

Dieser Datentyp beschreibt einen Verweis auf ein Merkmal als REFERENCE, welcher die Position des Merkmals im umschließenden Auswahlmenü angibt.

2.4.19 PARTITION_REF

Dieser Datentyp beschreibt einen Verweis auf eine Partition als REFERENCE, welcher die Position der Partition im umschließenden Auswahlmenü angibt.

2.4.20 PROPERTY_REF

Dieser Datentyp beschreibt einen Verweis auf eine Eigenschaft als REFERENCE, welcher die Position der Eigenschaft in der umschließenden Partition angibt.

2.4.21 CORRELATION_REF

Dieser Datentyp beschreibt einen Verweis auf eine Wechselbeziehung als REFERENCE, welcher die Position der Wechselbeziehung im umschließenden Auswahlmenü angibt. Da die Wechselbeziehungen einen gerichteten azyklischen Graphen bilden müssen, sind rekursive Verweise unzulässig.

3 Empfehlung

3.1 Anbindungen

3.1.1 Grundlagen

Auch wenn die oben spezifizierten Datenstrukturen zur Anbindung von Merkmalen an Datenkanäle bzw. Steuerelemente (BINDING) viele Freiheitsgrade besitzen, kann eine Einschränkung auf die nachfolgend vorgestellten Anbindungsstrukturen ratsam sein, insbesondere wegen ihrer Einfachheit in Bestückung und Verarbeitung sowie ihrer Verständlichkeit. Erweitert werden können diese schließlich jederzeit. Mit Anbindungsstruktur ist an dieser Stelle die konkrete Bestückung der Rollen bzw. Kennungen (role) und Merkmale (feature) sowie die Einbettungen (bindings) von Anbindungen gemeint.

Ein Client sollte grundsätzlich alle Anbindungen mit unbekannten Rollen sowie alle Datenfelder von Konfigurationsdaten (config) mit unbekannten Kennungen ignorieren.

3.1.2 Menüanbindung

Diese Anbindungsstruktur kodiert ein eigenständiges Menü zur interaktiven Bestückung einer Datenstruktur. Sie liegt bei einer Anbindung vor, die ihrerseits Anbindungen mit den Rollen "INPUT" und "OUTPUT" einbettet, von denen die erste als Ein- und die zweite als Ausgabeanbindung strukturiert sind. Die Eingabeanbindung erlauben dabei die interaktive Anpassung der Merkmale, die über die Ausgabeanbindung in die Datenstruktur übersetzt werden.

Neben diesen notwendigen Anbindungen kann eine Menüanbindung zusätzlich noch optionale Anbindungen mit den Rollen "LABEL", "SETUP" und "RESTORE" enthalten, von denen die erste als Ein- und die anderen beiden als Ausgabeanbindung strukturiert sind. Die Anbindung mit der Rolle "LABEL" erlaubt die möglichst kompakte Darstellung der Merkmalsbestückung. Die Anbindung mit der Rolle "SETUP" dient der automatischen Bestückung bestimmter Merkmale über eine Datenstruktur ohne Interaktion. Über die Anbindung mit der Rolle "RESTORE" kann dem Client eine möglichst versionsstabile und auf Auflistungen verzichtende Datenstruktur zur Speicherung und Wiederherstellung von Einstellungen bereitgestellt werden, bspw. zur Anbindung von Favoriten- und Nutzerprofilspeichern. Ein Beispiel für die Struktur ist:

```
{
  "bindings": [
    {
      "role": "SETUP",
      "bindings": [
        {
          "role": "vonHst",
          "feature": 5
        },
        {
          "role": "bisHst",
          "feature": 9
        }
      ]
    },
    {
      "role": "INPUT",
      "bindings": [...]
    },
    {
      "role": "LABEL",
      "bindings": [
        {
          "role": "GROUP",
          "feature": ...,
          "bindings": [
            {
              "role": "LABEL",
              "text": ...,
              "feature": ...
            }
          ]
        }
      ]
    },
    {
      "role": "OUTPUT",
      "bindings": [
        {
          "role": "version",
          "defaults": ["S2020_04"]
        },
        {
          "role": "variante",
          "feature": 18
        },
        {
          "role": "elementliste",
          "bindings": [
            {
              "feature": 10
            },
            {
              "bindings": [
                {
                  "role": "nr",
                  "feature": 15
                }
              ]
            },
            {
              "feature": 11,
              "bindings": [
                {
                  "role": "nr",
                  "defaults": ["I54"]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```



```

    }},
    {"role": "RESTORE", "bindings": [...]}, ...
  ]}

```

Bei der oben zum Datentyp FEATURE beschriebenen Ermittlung der Eigenschaften, die in der Ausprägung eines Merkmals eingesetzt werden dürfen, sollte darauf geachtet werden, dass beim Abgleich mit Wechselbeziehungen zu bereits gewählten Merkmalen nur die in der Anbindungsstruktur davor aufgeführten anpassbaren Merkmale respektiert und alle danach aufgeführten ignoriert werden. Eine Ausnahme hiervon bildet nur die anfängliche Bestückung der Merkmale mit wiederhergestellten Einstellungen aus der Anbindung mit der Rolle "RESTORE".

Wenn in der zur Anbindung mit der Rolle "SETUP" bzw. "RESTORE" gegebenen Datenstruktur ein Datenfeld enthalten ist, zu welchem diese einbettende Anbindung keine passende Anbindung einbettet, wird dieses Datenfeld ignoriert. Andernfalls wird der Wert des Datenfeldes zur Bestückung der eingebetteten Anbindung eingesetzt. Letztere kann dabei entweder ebenfalls selbst einbettend sein und rekursiv bestückt werden oder das zu bestückende Merkmal anbinden. Wenn in die Gegenrichtung die getroffenen Einstellungen über die einbettende Anbindung mit der Rolle "RESTORE" gespeichert werden sollen, dürfen nur die dort aufgeführten Datenfelder übernommen werden. Die zuvor ignorierten Datenfelder sind auch hier zu ignorieren, damit Einstellungen insb. bei dynamisch erzeugten Auswahlmenüs nicht verloren gehen.

3.1.3 Anzeigeanbindung

Diese Anbindungsstruktur kodiert eine Erzeugungsvorschrift für beliebig gruppierte bzw. hierarchische Steuerelemente zur Anzeige von Merkmalen.

Ein Steuerelement zur Beschriftung wird durch eine Anbindung mit der Rolle "LABEL" kodiert. Dieses Steuerelement sollte nur dann angezeigt werden, wenn es selbst über Anzeigedaten (bspw. Anzeigetext) verfügt oder über die Eigenschaft des angeordneten Merkmals solche bereitgestellt werden.

Eine Steuerelementgruppe wird durch eine Anbindung mit der Rolle "GROUP" kodiert, deren eingebettete Anbindungen dann die gruppierten Steuerelemente kodieren. Über ein angeordnetes Merkmal und/oder den Vorgabewert kann zudem die Sichtbarkeit der Gruppe gesteuert bzw. festgelegt werden. Die Steuerelementgruppe ist hierbei nur dann sichtbar, wenn der Wert true ("T") ist. Nur wenn die Anbindung der Steuerelementgruppe über Anzeigedaten verfügt, sollte die Gruppierung auch in der Anzeige erkennbar sein, bspw. durch Beschriftung mit Anzeigetext, Hinweistext und/oder Piktogramm sowie durch Einrücken und/oder Umranden.

Ein Beispiel für die Struktur ist:

```

{"role": "GROUP", "feature": ..., "bindings": [
  {"role": "LABEL", "text": ...},
  {"role": "GROUP", "feature": ..., "bindings": [...]},
  {"role": "LABEL", "text": ..., "feature": ...}, ...
]}

```

3.1.4 Eingabeabbindung

Diese Anbindungsstruktur kodiert eine Erzeugungsvorschrift für beliebig gruppierte bzw. hierarchische Steuerelemente zur Anzeige sowie Bestückung von Merkmalen. Sie baut auf der Anzeigeanbindung auf und ergänzt diese.

Ein Steuerelement zur Werteingabe bzw. Wertauswahl wird durch eine auf das interaktiv anzupassende Merkmal verweisende Anbindung mit der Rolle "INPUT" bzw. "SELECT" kodiert.

Wenn ein Merkmal an einer bestimmten Position in der Anbindungsstruktur als allgemeines Zwischenergebnis im Auswahlprozess oder als genereller Vorgabewert bestückt werden soll, wird dies durch eine auf dieses Merkmal verweisende Anbindung mit der Rolle "HIDDEN" kodiert. Diese Anbindung stellt kein sichtbares Steuerelement dar und benötigt daher auch keine Anzeigemerkmale. Ein Beispiel für die Struktur ist:

```
{
  "role": "GROUP", "feature": "...", "bindings": [
    {
      "role": "GROUP", "feature": "...", "bindings": [...],
      "role": "LABEL", "text": "...", "feature": "...",
      "role": "INPUT", "text": "...", "feature": "...",
      "role": "SELECT", "text": "...", "feature": "...",
      "role": "HIDDEN", "feature": "...", "defaults": [...],
    }
  ]
}
```

Steuerelemente zur Werteingabe können für bestimmte Wertebereiche optimiert sein, bspw. zur Text-, Zahlen- oder Datumseingabe. Neben der Möglichkeit zur Erkennung des am besten passenden Steuerelements anhand der Filterkriterien in den zulässigen Eigenschaften der angeordneten Merkmale, sollten dem Client im Zweifel Konfigurationsdaten bereitgestellt werden, in denen bspw. der Datentyp der einzugebenden Werte explizit angegeben ist. Darüber könnte bspw. die Verwendung einer alphabetischen bzw. numerischen Tastatur vorgeben. In diesem Zusammenhang ist bei der Verwendung von Filterkriterien grundsätzlich darauf zu achten, dass diese nicht gemischt werden und dementsprechend alle Eigenschaften einer Partition auf dem gleichen Datentyp beruhen.

3.1.5 Ausgabeanbindung

Diese Anbindungsstruktur kodiert eine Erzeugungsvorschrift für beliebige Datenstrukturen als Ausgabewert und erlaubt damit die generische Bereitstellung von Auflistungen (array), Abbildungen (object) sowie den darin enthaltenen Elementen bzw. Datenfeldern. Bei Verzicht auf Auflistungen kann diese Erzeugungsvorschrift auch umgekehrt von einer gegebenen Datenstruktur auf eine Merkmalsbestückung abbilden, was bspw. bei der zuvor vorgestellten Menüanbindung für die Wiederherstellung von Einstellungen Anwendung findet.

Der Ausgabewert einer nicht einbettenden Anbindung entspricht dem Wert des Merkmals, auf welches die Anbindung verweist, bzw. dem Vorgabewert, wenn die Anbindung auf kein Merkmal verweist. Der Ausgabewert kann daher auch unbestimmt sein.

Bei einer einbettenden Anbindung mit Verweis auf ein Merkmal wird der Ausgabewert der einzigen eingebetteten Anbindung nur dann übernommen, wenn der Wert des Merkmals true ("T") ist. Andernfalls ist der Ausgabewert unbestimmt.

Der Ausgabewert einer einbettenden Anbindung ohne Verweis auf ein Merkmal wird dagegen aus der Struktur und den Ausgabewerten der eingebetteten Anbindungen abgeleitet. Wenn alle eingebetteten Anbindungen über eine Rolle verfügen, ist der Ausgabewert der einbettenden Anbindung eine Abbildung von Rollen auf Ausgabewerte der eingebetteten Anbindungen. Andernfalls ist der Ausgabewert der einbettenden Anbindung eine Auflistung der Ausgabewerte der eingebetteten Anbindungen. Unbestimmte Ausgabewerte entfallen grundsätzlich, aber leere Auflistungen bzw. Abbildungen bleiben grundsätzlich erhalten.

Ein Beispiel für die Struktur ist:

```
{
  "bindings": [
    {
      "role": "typ", "feature": "...",
      "role": "alter", "feature": "...",
      "role": "rabattmedienliste", "feature": "...", "bindings": [
        { "feature": "...", "feature": "..."}
      ]
    }
  ]
}
```

3.2 Implementation

3.2.1 Grundlagen

Der azyklische Graph aller Wechselbeziehungen (CORRELATION) zwischen den Eigenschaften in den Ausprägungen von Merkmalen kann zwar als große Tabelle verstanden werden, doch ist es aufgrund der Speicherkomplexität meist nicht möglich, diese Tabelle explizit auszurechnen. Aus diesem Grund empfiehlt sich die Abbildung dieser Tabelle als virtuelle Zeilenmenge, die sich aus einem normalisierten Term ergibt. Neben der Ermittlung und Auswertung dieser Normalform gibt es aber noch weitere Problemstellungen, zu denen hier Empfehlungen bereitgestellt werden.

3.2.2 Ermittlung von Eigenschaften

Auch wenn es in den oben spezifizierten Partitionen zulässig ist, Eigenschaften mit Abgleichskriterien unterschiedlicher Datentypen anzugeben, sollten Partitionen insbesondere für den Abgleich mit Eingabewerten keine gemischten Datentypen enthalten.

Wenn der Wert der Ausprägung eines Merkmals eingegeben wird, muss dieser mit den Eigenschaften (PROPERTY) aus der Partition dieses Merkmals abgeglichen werden, um dazu die anzuwendende Eigenschaft abzuleiten. Dazu müssen diese Eigenschaften ihrer Reihenfolge und damit ihrer Priorität entsprechend durchlaufen werden, um die erste zum Eingabewert passende zu ermitteln. Der Abgleich mit dem Eingabewert sollte dabei abhängig von den an der Eigenschaft angegebenen Abgleichskriterien zuerst über den Ausgabewert (value), danach über den Wertebereich (minInteger, maxInteger, ...) und zuletzt über das Textmuster (pattern) erfolgen, wobei abwesende Kriterien übersprungen werden und der Abgleich bei der ersten erfolgreichen Erkennung endet.

Abhängig vom Datentyp der vorliegenden Abgleichskriterien muss der Eingabewert für den Abgleich entsprechend umgewandelt werden, bspw. eine Eingabebene Ganzzahl in eine Zeichenkette oder umgekehrt. Wenn diese Umwandlung zu einem Fehler führen würde, sollte das Abgleichkriterium übersprungen werden.

3.2.3 Abbildung von Ausprägungen

Die Ausprägung eines Merkmals (FEATURE) beinhaltet gemäß Spezifikation stets einen Wert (ANY) und eine Eigenschaft (PROPERTY). Der Wert kann hierbei unbestimmt sein. Die Eigenschaft kann dies nur in der Initialisierungsphase. Grundsätzlich sollten zu jedem Merkmal die folgenden Informationen verwaltet werden:

- Priorität des Merkmals. Diese ergibt sich aus dem ersten Vorkommen einer ändernden Abbildung dieses Merkmals (vgl. Eingabeanbindung INPUT, SELECT, HIDDEN) in der Anbindungsstruktur.
- Aktuell gewählter Wert. Dieser stammt entweder direkt aus der Werteingabe (vgl. Eingabeanbindung INPUT) oder wurde aus der gewählten Eigenschaft abgeleitet und kann daher unbestimmt sein.
- Aktuell gewählte Eigenschaft. Diese stammt entweder aus der Wertauswahl (vgl. Eingabeanbindung SELECT) oder wurde aus dem gewählten Wert abgeleitet.

- List der Kandidaten, aus denen die aktuelle Eigenschaft gewählt wurde. Diese stammt aus der weiter unten beschriebenen Auswertung der Wechselbeziehungen unter Beachtung der Bestückung der Merkmale mit höherer Priorität, die also in der Anbindungsstruktur davor änderbar sind.
- Letzte manuelle Bestückung von Wert bzw. Eigenschaft. Diese sollte zur Wiederherstellung der manuellen Auswahl eingesetzt werden, wenn die Bestückung des Merkmals zwischenzeitlich infolge der Änderung der Bestückung höher priorisierter Merkmale ebenfalls geändert wurde. Diese letzte manuelle Bestückung hat dabei Vorrang vor den Vorgabewerten (defaults) der Anbindungen sowie den aus der Reihenfolge der Eigenschaften folgenden Priorität, sofern die Anbindungen zur Änderung dieser Merkmale weiterhin angezeigt werden.

3.2.4 Abbildung von Wechselbeziehungen

Die Wurzeln des azyklischen Graphs aller Wechselbeziehungen sollte zur effizienten Verarbeitung in eine Normalform überführt werden, welche aus einer Vereinigung (union) von Verschmelzungen (join) besteht, die ihrerseits nur noch Basistabellen (base) enthalten, welche aus verketteten (concat) bzw. konkreten (features) Tabellen abgeleitet wurden. Die Ableitung der Basistabellen einer verketteten bzw. konkreten Tabelle erzeugt zu dieser Tabelle ein neues Merkmal, welches für die Zeilennummer der Tabelle steht, und dieses mit jeweils einem der übrigen Merkmale der Tabelle zu einer zweispaltigen Basistabelle verbindet. Die Verschmelzung der abgeleiteten Basistabellen ergibt dann wieder die gegebene Tabelle, ergänzt um die zusätzliche Spalte mit den Zeilennummern. Eine Basistabelle bildet damit von einer Zeilennummer auf die Menge der in dieser Zeile zulässigen Eigenschaften eines Merkmals ab. Die Negation (not) einer Basistabelle bildet damit logisch auf die in der Zeile unzulässigen Eigenschaften, also die übrigen der Partition des Merkmals ab. Dafür reicht es datentechnisch aus, die Basistabelle als negiert zu kennzeichnen. Neben der semantischen Negation einer Basistabelle wird technisch zudem noch ihre strukturelle Umkehrung benötigt, bei welcher nicht von Zeilennummern auf Eigenschaften abgebildet wird, sondern von Eigenschaften auf Zeilennummern. Semantisch macht dies keinen Unterschied. Die Normalform kann über folgende Gleichheiten abgeleitet werden:

```

not(not(T)) = T
not(base(F1, F2, N)) = base(F1, F2, not(N))
not(join(T1, ..., Tn)) = union(not(T1), ..., not(Tn))
not(union(T1, ..., Tn)) = join(not(T1), ..., not(Tn))
base(F1, F2, N) = join(base(F1, F2, N), base(F2, F1, N))
join(T1, T2) = join(T2, T1)
join(T1, ..., T1) = join(T1)
join(join(T1, ..., Ti), Tk, ..., Tn) = join(T1, ..., Ti, Tk, ..., Tn)
join(union(T1, ..., Ti), Tk, ..., Tn) = union(join(T1, Tk, ..., Tn), ..., join(Ti, Tk, ..., Tn))
union(T1, T2) = union(T2, T1)
union(T1, ..., T1) = union(T1)
union(join(not(T1), T1, ..., Ti), Tk, ..., Tn) = union(Tk, ..., Tn)
union(union(T1, ..., Ti), Tk, ..., Tn) = union(T1, ..., Ti, Tk, ..., Tn)
concat(F1, ..., Fn) = join(base(Z, F1, FALSE), ..., base(Z, Fn, FALSE))
features(F1, ..., Fn) = join(base(Z, F1, FALSE), ..., base(Z, Fn, FALSE))

```

3.2.5 Auswertung von Wechselbeziehungen

Die Auswertung der Normalform des azyklischen Graphs aller Wechselbeziehungen erfolgt zuerst separat je Verschmelzung (join) und danach für deren Vereinigung (union). Hierfür müssen zu jeder Verschmelzung und je Merkmal folgende Zustandsdaten verwaltet werden:

- Liste der von diesem Merkmal aus auf andere Merkmale abbildende Basistabellen. Diese beinhaltet infolge der oben beschriebenen Normalisierung je nach Merkmal die von Zeilennummern auf Eigenschaften oder umgekehrt abbildenden Tabellen.
- Liste von Eigenschaften als Kandidaten zur Bestückung des Merkmals.
- Markierung der Änderung der Kandidatenliste. Technisch sollte diese über einen Ring (doppelt verkettete Liste) der geänderten Zustandsdaten realisiert werden, um die geänderten Zustandsdaten schnell an das Listenende zu verschieben bzw. dort einzufügen. Jede Änderung der Kandidatenliste markiert die Zustandsdaten als geändert.

Die Auswertung der Zustandsdaten einer Verschmelzung beginnt mit der Bestückung der Kandidatenliste je Merkmal. Wenn das Merkmal noch nicht bestückt wurde, enthält die Kandidatenliste alle Eigenschaften der Partition des Merkmals. Andernfalls enthält sie nur die Eigenschaft aus der aktuellen Bestückung des Merkmals.

Solange die Liste der geänderten Zustandsdaten nicht leer ist und damit Änderungen an den Kandidatenlisten behandelt werden müssen, wird wie folgt verfahren:

- Das erstes Element der Liste der geänderten Zustandsdaten wird entnommen. Diese Zustandsdaten verlieren dadurch zunächst ihre Änderungsmarkierung.
- Die Kandidatenliste dieser Zustandsdaten wird anschließend über jede der Basistabelle auf die dazu passenden zulässigen Eigenschaften anderer Merkmale abgebildet und auf die zugehörigen Zustandsdaten übertragen werden. Die Übertragung erfolgt dabei als Schnittmenge, wodurch die Kandidatenlisten auf die noch zulässigen reduziert werden.

Diese Iteration endet für jede Verschmelzung mit den darin je Merkmal zulässigen Kandidaten. Die insgesamt je Merkmal zulässigen Kandidaten ergeben sich dann aus der Vereinigung der Kandidatenliste der entsprechenden Zustandsdaten aus den Verschmelzungen, bei denen für kein Merkmal eine leere Kandidatenliste ermittelt wurde. Wenn sich bei der Auswertung einer Verschmelzung bereits initial eine leere Kandidatenliste ergibt, wenn also noch keine Merkmale bestückt wurden, kann die Verschmelzung verworfen werden.

Die je Merkmal zulässigen Kandidaten erlauben nun die Bestückung des nächsten noch nicht bestückten Merkmals entsprechend der spezifizierten Prioritäten und empfohlenen Reihenfolgen. Nachdem ein Merkmal bestückt wurde, muss die oben beschriebene Rechnung mit der neuen Merkmalsbestückung wiederholt werden, um die Kandidatenlisten zu aktualisieren. Da das bestückte Merkmal die Einschränkung seiner Kandidatenliste auf ein Element beschreibt, kann die Rechnung zur Bestückung weiterer Merkmale direkt auf den zuvor ermittelten Zustandsdaten fortgesetzt werden.

Im Ergebnis dieser Rechnung sind alle angeordneten Merkmale bestückt und enthalten alle deren Kandidatenlisten höchstens noch ein Element.