

Anlage 3 zu den Bewerbungsbedingungen, zu den Vorgaben zur Erstellung des Angebotes und zum Lastenheft: Schnittstellendefinition VDV 457-2

Inhalt

1	Einführung.....	2
2	Generelle Anforderungen.....	2
2.1	Datenaufzeichnung im Fahrzeug, Zeitstempel	2
2.2	Datenlieferung, Dateinamenskonvention	3
3	PassengerCountingMessage (PCM).....	4
3.1	HeaderData.....	5
3.1.1	APC-Device.....	6
3.2	PassengerCountingEvent (PCE).....	8
3.2.1	HeaderCountingEvent.....	8
3.2.2	JourneyInfo.....	11
3.2.3	Line	11
3.2.4	StopInformation	13
3.2.5	GNSS.....	15
3.2.6	CountingArea	17
3.2.7	APCInventory.....	22
3.2.8	Composition	24
3.2.9	Beispiel: Wartesaalerfassung mit Intervallmessung.....	25
4	WayPointsMessage (WPM)	27
4.1	HeaderData.....	28
4.2	Coordinates	28
4.3	LineInfo.....	28
4.4	JourneyInfo	28
5	Anhang: XML Basistypen	29
5.1	Enumerations aus PassengerCountingServiceBGS	29
5.1.1	PassengerCounting	29
5.1.2	GNSS.....	30
5.2	XML allgemein.....	30
5.3	VDV IBIS-IP Basistypen.....	31

Neu in Version vom 19.02.2026: CountingOperationState= Muss; Kapitel 3.2.9 ergänzt

Neu in Version vom 23.02.2026: Muss-Anforderungen bei manuellen Zähldaten ergänzt

Neu in Version vom 31.03.2026: Wartesaalintervalle ohne Fahrgastwechsel in Kapitel 3.2.9

1 Einführung

Dieses Dokument gibt Interpretationshilfen zur Erstellung und Verwendung von Fahrgastzähldaten gemäß dem Standard „457-2“ und definiert zusätzliche Empfehlungen und verbindliche Vorgaben zur Verwendung in den AFZS-Clustern in Baden-Württemberg.

Adressaten des Dokuments sind sowohl die Datenlieferanten (Fahrgastzählsysteme) als auch die Datenempfänger (Hintergrundsysteme).

Liegt dieses Dokument als Anlage zu einer Leistungsbeschreibung oder einem Lastenheft bei, so gelten die Festlegungen verbindlich, solange in der Leistungsbeschreibung oder dem Lastenheft keine abweichenden Festlegungen getroffen werden. Der Standard VDV 457-2 beschreibt das Format für den Transport von Zählrohdaten zwischen einem im Fahrzeug verbauten Fahrgastzählsystem und einem weiterverarbeitenden Hintergrundsystem. Die Daten werden in einem XML-Format transportiert, zu dem ein XSD veröffentlicht wurde:

- IBIS-IP_PassengerCountingServiceBGS[Version].xsd

Dieses XSD verweist auf weitere XSD-files aus dem „IBIS-IP“ Umfeld, in denen weitere Datentypen und Enumerations definiert sind. Die wichtigsten Typen sind in Kapitel 5 aufgeführt.

Die IBIS-IP Datentypen enden fast immer mit „Value“. Das Element „ErrorCode“ kann i.d.R. entfallen.

2 Generelle Anforderungen

2.1 Datenaufzeichnung im Fahrzeug, Zeitstempel

Wichtig ist die Einhaltung der im Anforderungskatalog AFZS des Landes Baden-Württemberg formulierten Anforderungen wie

- Türscharfe Erfassung der Fahrgastzahlen von Personen $\geq 1,20$ m Körpergröße („Adult“)
- Wartesaalerfassung
- Erfassung von Tür-, System- und Fehlerstatus
- Erfassung der GNSS-Positionen (z.B. GPS)

sowie die differenzierte Erfassung der Zeitpunkte der Messung (Türöffnung, Wartesaal...).

Der Zeitgeber der Onboard-Unit muss mindestens täglich mit GPS oder NTP synchronisieren. Der Zeitgeber muss nach Kaltstart mit einer intern gepufferten Zeit arbeiten, damit alle Datensätze einen Zeitstempel erhalten. Der Zeitgeber ist dann mit GPS oder NTP zu synchronisieren, sobald Internet oder GPS verfügbar sind.

Alle Zeitangaben sollen in UTC-Zeit erfolgen.

Unter „Messung“ und den damit verbundenen Angaben zu Zeit und Ort wird im Folgenden der Vorgang der Zählung verstanden, also das Erfassen der Fahrgäste während der geöffneten Türe. Dies ist nicht zu verwechseln mit dem Zeitpunkt und Ort, an dem das Zentralgerät die Messwerte ggf. erst nach Abfahrt und somit zeitversetzt von den Sensoren abfragt.

2.2 Datenlieferung, Dateinamenskonvention

Es spielt keine Rolle, ob die XML-Dateien direkt im AFZS-Gerät im Fahrzeug oder nachgelagert in einem Hintergrundsystem auf Basis der auf andere Weise übertragenen und gesammelten Rohdaten generiert werden. Die Art der Datenbereitstellung, etwa in gezippter Form auf einem SFTP-Server, ist projektspezifisch festzulegen.

Erwartet wird eine tägliche Datenlieferung in möglichst wenigen Einzeldateien je Fahrzeug. Im Idealfall umfasst ein Set von Dateien die Daten eines ganzen Betriebstages eines Fahrzeugs. Die Aufteilung eines Tages in mehrere Dateien, die größere Blöcke zusammenfassen (z.B. nach Umläufen), ist aber zulässig. Die Daten verteilen sich entsprechend Datentyp auf zwei Dateien:

- **PassengerCountingMessage** („pcm“)
- **WayPointMessage** („wpm“)

Die Dateinamen sollen Datum, Zeit und Fahrzeug sowie Datentyp erkennen lassen und sich an der folgenden Notation orientieren:

Datum_Zeit_Fahrzeug_typ.xml

Datum	YYYY-MM-DD oder YYYYMMDD	Datum der Generierung der Datei
Zeit	hh-mm-ss oder hhmmss	Zeitpunkt der Generierung (für alle Dateien eines Sets aus „pcm“ und „wpm“ identisch) Wenn dieselben Zählungen zu einem späteren Zeitpunkt lediglich neu aufbereitet werden, soll der Dateiname unverändert bleiben, um doppelte Daten zu vermeiden.
Fahrzeug	Text ohne Leerzeichen	Siehe VehicleID, z.B. FR-VG123
typ	3 Zeichen	„pcm“, „wpm“

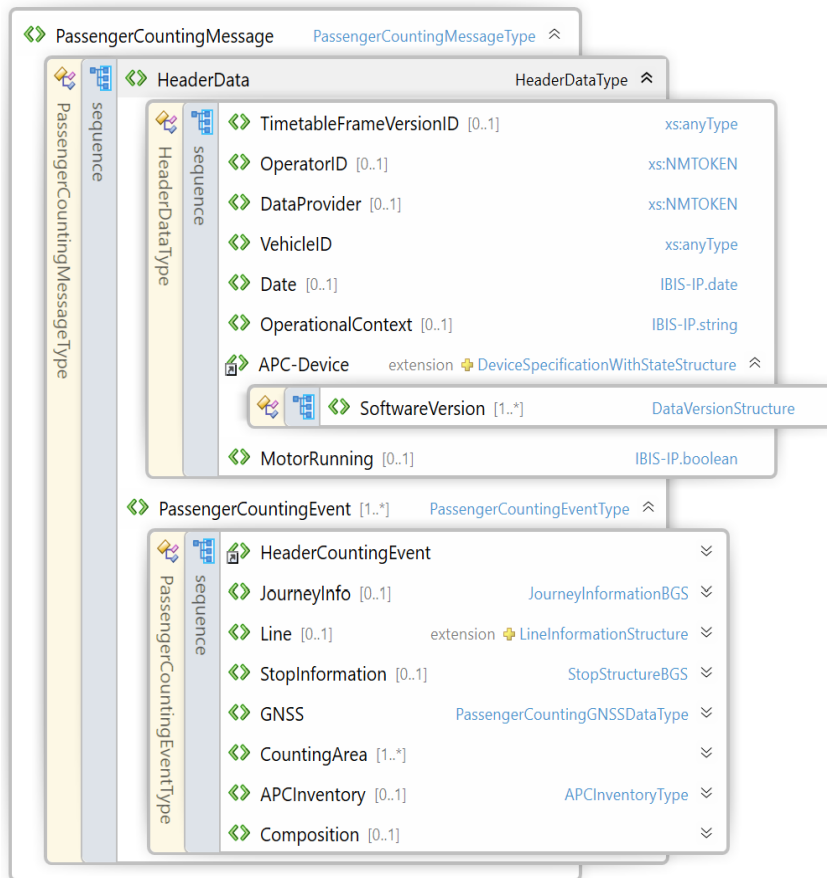
Beispiel: 2025-05-19_14-09-36_FR-VG123_pcm.xml

Die Lieferung zusätzlicher, herstellersistenspezifischer Logfiles parallel zu den VDV-Daten ist möglich und erwünscht, vorzugsweise in gezippter Form. Diese Logfiles sollen strukturiert (Tags oder Spaltentrenner, jeder Eintrag mit Zeitstempel), maschinell verarbeitbar (eindeutige Tags und Codes) und menschenlesbar (Klartextelemente) sein.

3 PassengerCountingMessage (PCM)

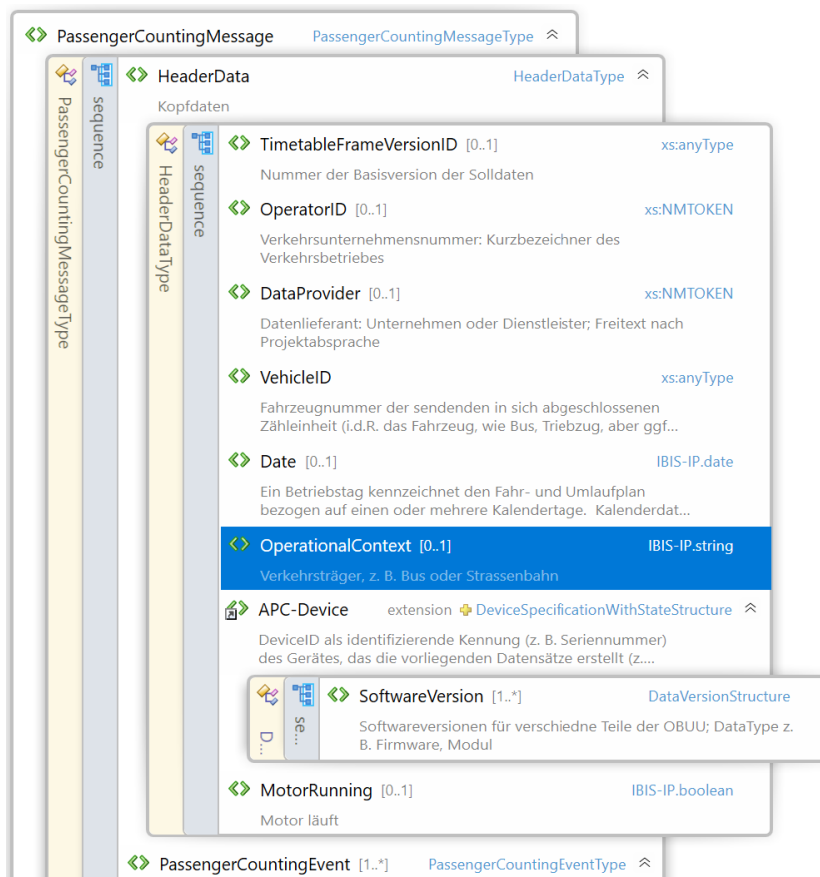
Hier gibt es einen Header (1x pro Datei), gefolgt von vielen PassengerCountingEvents (je Halt)

HeaderData und PassengerCountingEvent sind „Muss“



In den nachfolgenden Unterkapiteln werden die einzelnen Bestandteile näher beschrieben.

3.1 HeaderData



VehicleID	Muss	KFZ-Kennzeichen, Format „FR-VG123“ ohne Leerzeichen Straßenbahn: „FR-123“ (Stadtkennung-Fahrzeugnummer) Bahn: UIC-Nummer (im Ausnahmefall und nach Absprache: anders aufgebaute Fahrzeugnummer wie „VT-123“)
DataProvider	Muss	Datenlieferant/Systemintegrator (Kurzbezeichnung der Firma). Die genaue Bezeichnung ist vor Implementierung mit dem Clusterverantwortlichen abzustimmen.
OperatorId	Muss	Verkehrsunternehmen (Kurzbezeichnung), ggf. auch Aufgabenträger. Muss <u>zusammen mit „Vehicle“</u> eindeutig sein in Baden-Württemberg. Die ID ist vor Implementierung mit dem Clusterverantwortlichen abzustimmen.
TimetableFrame-VersionID	weglassen	wird bei der Weiterverarbeitung ignoriert
Date	weglassen	wird bei der Weiterverarbeitung ignoriert
OperationalContext	weglassen	wird bei der Weiterverarbeitung ignoriert
MotorRunning / Value	Wünschenswert	IBIS-IP.boolean xs:boolean Mögliche Werte: true, false, 1 (for true), 0 (for false) Den jeweils gültigen Status des Zündungssignals (bei E-Fahrzeugen eine Entsprechung) zum Zeitpunkt der Generierung von HeaderData verwenden.

3.1.1 APC-Device

PassengerCountingMessage/HeaderData/APC-Device/DeviceSpecification[]/DeviceClass
 PassengerCountingMessage/HeaderData/APC-Device/DeviceSpecification[]/DeviceID/Value
 PassengerCountingMessage/HeaderData/APC-Device/DeviceState[]
 PassengerCountingMessage/HeaderData/APC-Device/SoftwareVersion[]/DataType/Value
 PassengerCountingMessage/HeaderData/APC-Device/SoftwareVersion[]/VersionRef/Value

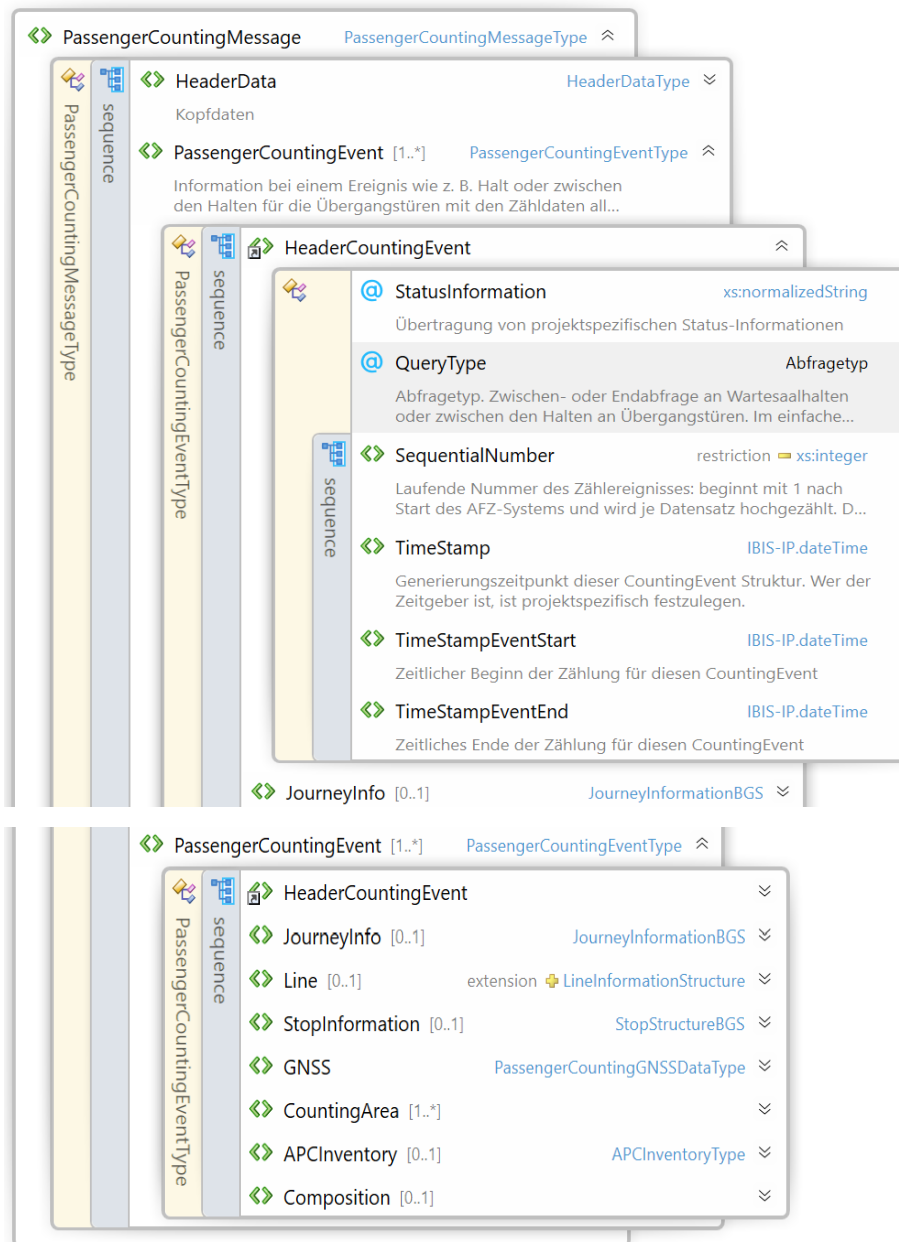
DeviceSpecificationWithStateStructure

DeviceSpecification/ DeviceClass	Muss	<p>Hinweis: Die Sensoren kommen in 0.</p> <p>Immer „APC“ für Fahrgastzählssystem Zentralgerät, andere Geräte werden nicht erwartet</p> <pre> <xs:simpleType name="DeviceClassEnumeration"> <xs:restriction base="xs:string"> <xs:enumeration value="OnBoardUnit"/> <xs:enumeration value="SideDisplay"/> <xs:enumeration value="FrontDisplay"/> <xs:enumeration value="InteriorDisplay"/> <xs:enumeration value="Validator"/> <xs:enumeration value="TicketVendingMachine"/> <xs:enumeration value="AnnouncementSystem"/> <xs:enumeration value="MMI"/> <xs:enumeration value="VideoSystem"/> <xs:enumeration value="APC"/> <xs:enumeration value="MobileInterface"/> <xs:enumeration value="Other"/> <xs:enumeration value="TestDevice"/> </xs:restriction> </xs:simpleType> </pre>
DeviceSpecification/ DeviceID/Value	Muss	<p>Identifizierende Kennung (z. B. Seriennummer) des Gerätes, das die vorliegenden Datensätze erstellt (z.B. OBU)</p> <pre> <xs:complexType name="DeviceSpecificationWithStateStructure"> <xs:sequence> <xs:element name="DeviceSpecification" type="DeviceSpecificationStructure"/> <xs:element name="DeviceState" type="DeviceStateEnumeration"/> </xs:sequence> </xs:complexType> <xs:complexType name="DeviceSpecificationStructure"> <xs:sequence> <xs:element name="DeviceClass" type="DeviceClassEnumeration"/> <xs:element name="DeviceID" type="IBIS-IP.NMTOKEN"/> </xs:sequence> </xs:complexType> </pre>
DeviceState	Muss	<p>z.B. „running“</p> <pre> <xs:simpleType name="DeviceStateEnumeration"> <xs:restriction base="xs:string"> <xs:enumeration value="defective"/> <xs:enumeration value="notavailable"/> <xs:enumeration value="running"/> <xs:enumeration value="readyForShutdown"/> </xs:restriction> </xs:simpleType> </pre>
SoftwareVersion/ DataType/Value	Muss	<p>Die Versionsstände für verschiedene Softwaremodule des Zentralgeräts, soweit trennbar; Mögliche Werte z.B. „firmware“, „APC“, „GPS“, „5G“. Mindestens die Softwareversion des für die Zählung relevanten Softwaremoduls muss angegeben werden.</p>

		<p>Bitte im Projekt dokumentieren, welche Softwaremodule es gibt und wie diese benannt sind.</p> <p>Im Falle einer einzigen, zentralen Softwareversion des Zentralgeräts: „firmware“ verwenden</p> <p>Wenn möglich die für die Zählfunktion relevante Softwareversion angeben, sollte die Gerätefirmware aus mehreren Softwaremodulen zusammensetzen. Dann „APC“ verwenden.</p> <p>Hinweis: Firmware der Sensoren kommt in 0</p> <pre> <xs:complexType name="DataVersionStructure"> <xs:sequence> <xs:element name="DataType" type="IBIS-IP.string"> </xs:element> <xs:element name="VersionRef" type="IBIS-IP.NMTOKEN"> </xs:element> </xs:sequence> </xs:complexType> </pre>
SoftwareVersion/ VersionRef/Value	Muss	<p>String</p> <p>Beispiel: 6.4.21.8</p>

3.2 PassengerCountingEvent (PCE)

Eine Datei enthält üblicherweise viele PassengerCountingEvents (eines pro Halt).



3.2.1 HeaderCountingEvent

Abhängig davon, ob es sich um ein autonomes oder bordrechnergestütztes Zählsystem handelt, kommen zwei Varianten in Frage, wie das PassengerCountingEvent ausgestaltet werden soll. Der Unterschied besteht in der Definition des Haltes und der Logik, wie die Zählungen abgelegt werden, insbesondere in Hinblick auf die zeitliche Differenzierung. Der Unterschied wird am deutlichsten im HeaderCountingEvent sichtbar, das im Folgenden in zwei Varianten definiert ist.

```

PassengerCountingMessage/PassengerCountingEvent/HeaderCountingEvent/QueryType
PassengerCountingMessage/PassengerCountingEvent/HeaderCountingEvent/SequentialNumber
PassengerCountingMessage/PassengerCountingEvent/HeaderCountingEvent/TimeStamp/Value
PassengerCountingMessage/PassengerCountingEvent/HeaderCountingEvent/TimeStampEventStart/Value
PassengerCountingMessage/PassengerCountingEvent/HeaderCountingEvent/TimeStampEventEnd/Value
  
```



```

<xs:simpleType name="Abfragetyp">
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="arrival"/>
    <xs:enumeration value="at stop"/>
    <xs:enumeration value="departure"/>
    <xs:enumeration value="between stops"/>
    <xs:enumeration value="unknown"/>
  </xs:restriction>
</xs:simpleType>

```

3.2.1.1 Variante 1: Autonomes Fahrgastzählsystem

Das autonome Zählsystem hat keine genaue Kenntnis vom Erreichen und Verlassen einer Haltestelle und ob es sich um eine planmäßige Haltestelle handelt. Es orientiert sich an Geschwindigkeit und Türsignal und zeichnet Zähldaten entsprechend auf. Charakteristisch ist darüber hinaus, dass die Zählung für die spätere Wartesaalverrechnung während des Haltes (genauer: während der Türöffnungszeit) kontinuierlich in kleinen Intervallen abgespeichert wird. Typisch sind 10 bis 15 Sekunden.

Statusinformation	optional	Text, systemspezifisch, im Projekt abzusprechen. Wird vom HGS ignoriert
QueryType	Muss	<p>Mögliche Werte:</p> <p>„unknown“ darf bei autonomen Systemen für <u>jede</u> Abfrage verwendet werden</p> <p>Alternativ (falls das System eine „Abfahrt“ erkennt):</p> <p>„at stop“ (für jede Zwischenabfrage),</p> <p>„departure“ (letzte Abfrage der Sensoren nach Abfahrt),</p> <p>Eine Datei soll nur eine der beiden Alternativen enthalten und diese nicht mischen, damit der Algorithmus des weiterverarbeitenden Systems entsprechend eingestellt werden kann.</p> <p>Bei Verwendung von „at stop“ ist stets ein abschließendes „departure“ zu schreiben. Das empfangene System verlässt sich darauf, dass es immer ein „departure“ gibt.</p> <p>Bei Verwendung von „unknown“ entfällt „departure“. Das auswertende System baut die Halte aufgrund der Zeitstempel und Geokoordinaten zu Halten zusammen.</p> <p><u>Generell gilt:</u></p> <p>„unknown“ liefert das Ergebnis für die im Intervall seit dem Öffnen der Tür oder dem letzten „unknown“ gezählten Fahrgäste.</p> <p>„at stop“ und liefert das Ergebnis für die im Intervall seit dem Öffnen der Tür oder dem letzten „at stop“ gezählten Fahrgäste.</p> <p>Das optionale „departure“ liefert das Ergebnis für die im letzten Intervall nach dem letzten „at stop“ gezählten Fahrgäste.</p> <p>Man muss also alle Ereignisse addieren, um die gesamte Fahrgastzahl des Haltes zu erhalten.</p> <p>Das auswertende System muss alle Ereignisse eines CountingEvents zusammen betrachten und einem Gesamthalt (an Haltestelle oder auf Strecke) zuordnen. Ggf. sind mehrere CountingEvents zu einem Gesamthalt zusammenzufassen (z.B. beim Vorrücken, getrennten Ankunfts- und</p>

		Abfahrtspositionen oder bei separat in CountingEvents abgelegten Türfreigaben) und als gemeinsamen Wartesaal interpretieren
Sequential-Number	Muss	<p>Laufende Nummer des Zählereignisses: beginnt mit 1 nach Start des AFZ-Systems (bzw. Beginn einer Datei) und wird je Datensatz hochgezählt. Die Nummer soll den Verlust von Datensätzen bei der Übertragung erkennbar machen. Nicht zu verwechseln mit der laufenden Nummer innerhalb einer Fahrt (siehe StopInformation/StopIndex).</p> <p>Die laufende Nummer ist bei der Entstehung der Erhebungs- und Statusdaten (-sätze) zu vergeben. Die laufende Nummer ist über alle Erhebungs- und Statusdatensätze fortlaufend zu vergeben.</p>
TimeStamp /Value	Muss	<p>Generierungszeitpunkt dieser CountingEvent Struktur.</p> <p>Wird nicht ausgewertet.</p>
TimeStampEventStart /Value	Muss	<p>Zeitlicher Beginn der Zählung für diesen CountingEvent.</p> <p>Zeitpunkt der Aktivierung des Sensors.</p> <p>Bei Wartesaalerfassung: Beginn des jeweiligen Wartesaalzeitraums.</p>
TimeStampEventEnd /Value	Muss	<p>Zeitliches Ende der Zählung für diesen CountingEvent.</p> <p>Bei („unknown“ bzw. „at stop“): Ende des Wartesaalzeitraums bzw. Zeitpunkt der Türschließung (was zuerst eintritt).</p> <p>Bei „departure“: Zeitpunkt des Endes der Zählung des Sensors.</p>

3.2.1.2 Variante 2: Mit Ortung der Haltestelle und des Halt-Status

Diese Variante soll bei manuellen Zählern und bei Daten aus VORSYSTEMEN zur Anwendung kommen, sofern diese nicht im Format 457-3 geliefert werden. In diesem Fall werden zur Wartesaalerfassung zwei Zeitfenster ausgegeben, das erste als „at stop“ oder „between stops“ und abschließend „departure“. Wenn der Wartesaalzeitraum nicht erreicht wird, ist direkt „departure“ zu verwenden.

Der Datensatz steht unabhängig davon, ob der Halt an einer Haltestelle stattfindet oder nicht. Diese Status-Details können in **StopInfo** abgelegt werden.

Statusinformation	optional	<p>Text, systemspezifisch, im Projekt abzusprechen.</p> <p>Wird vom HGS ignoriert</p>
QueryType	Muss	<p>Mögliche Werte:</p> <p>„at stop“ (Zwischenabfrage bei Wartesaalerfassung)</p> <p>„departure“ (letzte Abfrage der Sensoren nach Abfahrt)</p> <p>Bei Verwendung von „at stop“ ist stets ein abschließendes „departure“ zu schreiben. Das empfangene System verlässt sich darauf, dass es immer ein „departure“ gibt.</p> <p><u>Es gilt:</u></p> <p>„at stop“ liefert das Ergebnis für das Intervall seit dem Öffnen der ersten Tür</p> <p>„departure“ liefert das Ergebnis für das zweite Intervall nach „at stop“.</p> <p>Man muss also die „at stop“ und „departure“ addieren, um die gesamte Fahrgastzahl des Haltes zu erhalten.</p>

		Das auswertende System muss alle Ereignisse eines CountingEvents zusammen betrachten und einem Gesamthalt (an Haltestelle oder auf Strecke) zuordnen. Ggf. sind mehrere CountingEvents zu einem Gesamthalt zusammenzufassen (z.B. beim Vorrücken, getrennten Ankunfts- und Abfahrtspositionen oder bei separat in CountingEvents abgelegten Türfreigaben) und als gemeinsamen Wartesaal interpretieren.
Sequential-Number	Muss	Wie Variante 1
TimeStamp /Value	Muss	Wie Variante 1. Wird nicht ausgewertet.
TimeStampEventStart /Value	Muss	Zeitlicher Beginn der Zählung für diesen CountingEvent. Zeitpunkt der Aktivierung des Sensors. Bei Wartesaalerfassung („at stop“): Beginn des jeweiligen Wartesaalzeitraums.
TimeStampEventEnd /Value	Muss	Zeitliches Ende der Zählung für diesen CountingEvent. Bei „departure“: Zeitpunkt des Endes der Zählung des Sensors. Bei „at stop“: Ende des Wartesaalzeitraums bzw. Zeitpunkt der Türschließung (was zuerst eintritt).

3.2.2 JourneyInfo

Kein „Muss“, wird hier nicht weiter behandelt. Erfordert mehr als die Anbindung an IBIS oder IBIS-IP

„Muss“-Anforderung bei manuellen Zählern

Bei bordrechnergestützten Zählsystemen kann die Struktur gefüllt werden.

3.2.3 Line

Angaben zur Linie, auf der sich das Fahrzeug zum Zeitpunkt der Messung befindet

„Muss“ bei bordrechnergestützten Systemen.

„Muss“-Anforderung bei manuellen Zählern

„Muss“ in Teilen bei autonomen Zählsystemen, falls IBIS oder IBIS-IP angeschlossen ist. In diesem Fall ist die Information zu speichern, die zum Zeitpunkt **TimeStampEventStart** auf dem Fahrzeug gemeldet wurde (also bei Ankunft bzw. Start des jeweiligen Zeitfensters bei der Wartesaalerfassung).

Wird vom HGS meist ignoriert, ist aber hilfreich bei der Orientierung innerhalb der XML-Datei und bei der Fehlersuche.

PassengerCountingMessage/PassengerCountingEvent/Line/LineNumber/Value
 PassengerCountingMessage/PassengerCountingEvent/Line/LineRef/Value
 PassengerCountingMessage/PassengerCountingEvent/Line/Destination
 PassengerCountingMessage/PassengerCountingEvent/Line/DirectionType

LineRef/Value	wünschenswert	IBIS-IP.NMTOKEN, Hier vorzugsweise DLID
LineName/Value	kann entfallen	InternationalTextType
LineShortName/Value	kann entfallen	InternationalTextType
LineNumber/Value	Muss	IBIS-IP.int, rein numerisch: IBIS-Nummer
Destination	wünschenswert	Vorzugsweise Zieltext. Projektspezifische Absprache notwendig
DirectionType	wünschenswert	Zahl: 1 oder 2

XML-Strukturen:

```

<xs:element name="Line" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Linienangaben: z. B. RE1, 16, M10</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="LineInformationStructure">
        <xs:sequence>
          <xs:element name="Destination" type="xs:NMTOKEN" minOccurs="0">
            <xs:annotation>
              <xs:documentation>ID der Endhaltestelle der Fahrt</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="DirectionType" type="xs:int" minOccurs="0">
            <xs:annotation>
              <xs:documentation>LaufwegsRichtung. Richtungsnummer aus VDV452: LI_RI_NR (1 oder 2)  
Bezeichnung der Linienrichtung</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

<xs:complexType name="LineInformationStructure">
  <xs:sequence>
    <xs:element name="LineRef" type="IBIS-IP.NMTOKEN"/>
    <xs:element name="LineName" type="InternationalTextType" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="LineShortName" type="InternationalTextType" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="LineNumber" type="IBIS-IP.int" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="InternationalTextType">
  <xs:sequence>
    <xs:element name="Value" type="xs:string"/>
    <xs:element name="Language" type="xs:language"/>
    <xs:element name="ErrorCode" type="ErrorCodeEnumeration" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

```

3.2.4 StopInformation

Angaben zum Halt, an dem sich das Fahrzeug zum Zeitpunkt der Messung befindet.

„Muss“ bei bordrechnergestützten Systemen: Haltestelle und Zeiten

„Muss“-Anforderung bei manuellen Zähldaten

„Muss“, falls IBIS oder IBIS-IP angeschlossen ist: Haltestelle. In diesem Fall ist die Information zu speichern, die zum Zeitpunkt **TimeStampEventStart** auf dem Fahrzeug gemeldet wurde (also bei Ankunft bzw. Start des jeweiligen Zeitfensters bei der Wartesaalerfassung).

Wird vom HGS meist ignoriert, ist aber hilfreich bei der Orientierung innerhalb der XML-Datei und bei der Fehlersuche.

Hinweis zu den Zeiten (falls angegeben): Zusammenfassung eines kompletten Haltes mit Ankunft und Abfahrt, falls Türen mehrfach auf- und zu gehen. Erfordert Interpretation der Ankunft und Abfahrt im itcs/Bordrechner.

Haltestelle

PassengerCountingMessage/PassengerCountingEvent/StopInformation/StopIndex/Value
 PassengerCountingMessage/PassengerCountingEvent/StopInformation/StopRef/Value
 PassengerCountingMessage/PassengerCountingEvent/StopInformation/StopRef/ErrorCode
 PassengerCountingMessage/PassengerCountingEvent/StopInformation/StopName/Value
 PassengerCountingMessage/PassengerCountingEvent/StopInformation/StopName/Language
 PassengerCountingMessage/PassengerCountingEvent/StopInformation/TypeOfStopRef/Value

Zeiten

PassengerCountingMessage/PassengerCountingEvent/StopInformation/AimedArrivalTime
 PassengerCountingMessage/PassengerCountingEvent/StopInformation/AimedDepartureTime
 PassengerCountingMessage/PassengerCountingEvent/StopInformation/ActualArrivalTime
 PassengerCountingMessage/PassengerCountingEvent/StopInformation/ActualDepartureTime
 PassengerCountingMessage/PassengerCountingEvent/StopInformation/PassengerRelated/Value

Falls mit Türfreigabe gearbeitet wird und nicht mit individuellem Türsignal:

PassengerCountingMessage/PassengerCountingEvent/StopInformation/DoorReleaseStart/Value
 PassengerCountingMessage/PassengerCountingEvent/StopInformation/DoorReleaseEnd/Value

StopIndex/Value	wünschenswert	IBIS-IP.int Fortlaufende Nummer innerhalb der Fahrt
StopRef/Value	wünschenswert	IBIS-IP.NMTOKEN Wenn möglich DHID
StopName/Value	Muss	Text
TypeOfStopRef/Value	optional	IBIS-IP.NMTOKEN Bahnhofsnummer, VDV432, „DIVA“, „IFOPT“, „DHID“, „IBNR“
ActualArrivalTime/Value	optional	IBIS-IP.dateTime, 2025-06-11T03:55:34+00:00 Ankunftszeit am Halt (lt. Bordrechner)
ActualDepartureTime/Value	optional	IBIS-IP.dateTime, 2025-06-11T03:55:34+00:00 Abfahrtszeit am Halt (lt. Bordrechner)
DoorReleaseStart/Value	optional	IBIS-IP.dateTime, 2025-06-11T03:55:34+00:00 Türfreigabe
DoorReleaseEnd/Value	optional	IBIS-IP.dateTime, 2025-06-11T03:55:34+00:00 Ende der Türfreigabe

XML-Struktur StopStructureBGS

```

<xs:complexType name="StopStructureBGS">
  <xs:sequence>
    <xs:element name="StopIndex" type="IBIS-IP.int" minOccurs="0">
      <xs:annotation>
        <xs:documentation>StopIndex on current trip </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="StopRef" type="IBIS-IP.NMTOKEN">
      <xs:annotation>
        Reference to the stoppoint from the planning system
      </xs:annotation>
    </xs:element>
    <xs:element name="StopName" type="InternationalTextType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="StopAlternativeName" type="InternationalTextType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="AimedArrivalTime" type="IBIS-IP.dateTime" minOccurs="0">
      <xs:annotation>
        Laut zuletzt gültigem Soll-Fahrplan geplante Ankunftszeit (NeTex)
      </xs:annotation>
    </xs:element>
    <xs:element name="AimedDepartureTime" type="IBIS-IP.dateTime" minOccurs="0">
      <xs:annotation>
        Laut zuletzt gültigem Soll-Fahrplan geplante Abfahrtszeit (NeTex)
      </xs:annotation>
    </xs:element>
    <xs:element name="ActualArrivalTime" type="IBIS-IP.dateTime" minOccurs="0">
      <xs:annotation>
        Zeitpunkt bei dem das Fahrzeug den Halt erreicht. (Erste Türfreigabe erteilt und
Fahrzeug steht.) Die Angabe erfolgt mit Datum und Zeit. (NeTex)
      </xs:annotation>
    </xs:element>
    <xs:element name="ActualDepartureTime" type="IBIS-IP.dateTime" minOccurs="0">
      <xs:annotation>
        Zeitpunkt, bei dem das Fahrzeug den Halt verlässt. (Sowohl Türfreigabe nicht erteilt
bzw. Tür ist geschlossen, als auch Fahrzeug steht nicht mehr) Die Angabe erfolgt mit Datum und Zeit.
(NeTex)
      </xs:annotation>
    </xs:element>
    <xs:element name="DoorReleaseStart" type="IBIS-IP.dateTime" minOccurs="0">
      <xs:annotation>
        Türfreigabe zum ersten Mal erteilt, definiert Zeitspanne in der die Türen geöffnet
werden und damit Fahrgäste aussteigen oder einsteigen könnten
      </xs:annotation>
    </xs:element>
    <xs:element name="StopPoint" type="xs:unsignedInt" minOccurs="0">
      <xs:annotation>
        Nummer des Haltepunktes innerhalb der Haltestelle. Ist NULL, wenn der Halt nicht an
einer Haltestelle aufgezeichnet wurde. Dies ist auch der Fall, wenn das Fahrzeug weder auf einem Umlauf
noch auf einer Linie angemeldet war
      </xs:annotation>
    </xs:element>
    <xs:element name="PassengerRelated" type="IBIS-IP.boolean" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Fahrgastrelevant</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="TypeOfStopRef" minOccurs="0">
      <xs:annotation>
        Art der Haltestellen-ID: z. B. Bahnhofsnnummer, VDV432
      </xs:annotation>
    </xs:element>
    <xs:element name="DoorReleaseEnd" type="IBIS-IP.dateTime" minOccurs="0">
      <xs:annotation>
        Türfreigabe zum letzten Mal beendet. Definiert Zeitspanne in der die Türen geöffnet
werden und damit Fahrgäste aussteigen oder einsteigen könnten.
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

3.2.5 GNSS

Die Koordinate muss vom Zeitpunkt der Türöffnung oder Türschließung oder dazwischen stammen und nicht vom Zeitpunkt des Generierens des Datensatzes, vor Ankunft oder nach Abfahrt.

Muss (alle):

PassengerCountingMessage/PassengerCountingEvent/GNSS/GNSS_Point_Structure/**Longitude/Degree/Value**
 PassengerCountingMessage/PassengerCountingEvent/GNSS/GNSS_Point_Structure/**Latitude/Degree/Value**
 PassengerCountingMessage/PassengerCountingEvent/GNSS/GNSS_Type
 PassengerCountingMessage/PassengerCountingEvent/GNSS/GNSSCoordinateSystem

Muss (mindestens eines):

PassengerCountingMessage/PassengerCountingEvent/GNSS/**HorizontalDilutionOfPrecision/Value**
 PassengerCountingMessage/PassengerCountingEvent/GNSS/**SignalQuality**
 PassengerCountingMessage/PassengerCountingEvent/GNSS/**NumberOfSatellites/Value**

Kann:

PassengerCountingMessage/PassengerCountingEvent/GNSS/GNSS_Point_Structure/Longitude/Direction/Value
 PassengerCountingMessage/PassengerCountingEvent/GNSS/GNSS_Point_Structure/Latitude/Direction/Value
 PassengerCountingMessage/PassengerCountingEvent/GNSS/GNSS_Point_Structure/Altitude/Value
 PassengerCountingMessage/PassengerCountingEvent/GNSS/GNSS_Point_Structure/Direction/Value
 PassengerCountingMessage/PassengerCountingEvent/GNSS/SpeedOverGround/Value
 PassengerCountingMessage/PassengerCountingEvent/GNSS/VerticalDilutionOfPrecision/Value
 PassengerCountingMessage/PassengerCountingEvent/GNSS/time/Value

Latitude/Degree/Value Longitude/Degree/Value	Muss	GNSSCoordinateStructure / IBIS-IP.double 8.9588167 48.9463183
Latitude/Direction/Value Longitude/ Direction /Value	optional	Bewegungsrichtung
Altitude/Value	optional	IBIS-IP.double
SpeedOverGround/Value	optional	IBIS-IP.double
SignalQuality	wünschenswert	GNSSQualityEnumeration: dGPS, Estimated, GPS, NotValid, Unknown
NumberOfSatellites/Value	wünschenswert	IBIS-IP.int: Anzahl Satelliten
HorizontalDilutionOfPrecision/Value	wünschenswert	IBIS-IP.double: Genauigkeit der Koordinate Eins von SignalQuality NumberOfSatellites und HorizontalDilutionOfPrecision sollte vorhanden sein
GNSS_Type	Muss	GNSSTypeEnumeration Normalerweise GPS GALILEO, GLONASS, other
GNSSCoordinateSystem	Muss	GNSSCoordinateSystemEnumeration Normalerweise „WGS84“

XML-Strukturen

```
<xs:complexType name="PassengerCountingGNSSDataType">
  <xs:sequence>
    <xs:element name="GNSS_Point_Structure" type="GNSSPointStructure">
      <xs:annotation>
        Ortung. Direction: The direction is needed because longitude and latitude are in some
        coordinate system only positive values (see NMEA-Protocol, WGS 84). It should contain the angle (0 to
        359) clockwise. North is 0, east is 90. GNSSCoordinateSystem: Koordinatensystem. Standard für diese
        Schnittstelle ist WGS84
      </xs:annotation>
    </xs:element>
    <xs:element name="time" type="IBIS-IP.time" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

```

    <xs:element name="date" type="IBIS-IP.date" minOccurs="0"/>
    <xs:element name="SpeedOverGround" type="IBIS-IP.double" minOccurs="0"/>
    <xs:element name="SignalQuality" type="GNSSQualityEnumeration" minOccurs="0"/>
    <xs:element name="NumberOfSatellites" type="IBIS-IP.int" minOccurs="0"/>
    <xs:element name="HorizontalDilutionOfPrecision" type="IBIS-IP.double" minOccurs="0"/>
    <xs:element name="VerticalDilutionOfPrecision" type="IBIS-IP.double" minOccurs="0"/>
    <xs:element name="TrackDegreeTrue" type="IBIS-IP.double" minOccurs="0"/>
    <xs:element name="TrackDegreeMagnetic" type="IBIS-IP.double" minOccurs="0"/>
    <xs:element name="GNSS_Type" type="GNSSTypeEnumeration"/>
    <xs:element name="GNSSCoordinateSystem" type="GNSSCoordinateSystemEnumeration" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GNSSPointStructure">
  <xs:annotation>
    <xs:documentation>Structure for describing a GNSS-Point</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="PointRef" type="IBIS-IP.NMTOKEN" minOccurs="0">
      <xs:annotation>
        Reference to the point in schedule data
      </xs:annotation>
    </xs:element>
    <xs:element name="Longitude" type="GNSSCoordinateStructure"/>
    <xs:element name="Latitude" type="GNSSCoordinateStructure"/>
    <xs:element name="Altitude" type="IBIS-IP.double" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

  <xs:complexType name="GNSSCoordinateStructure">
    <xs:sequence>
      <xs:element name="Degree" type="IBIS-IP.double"/>
      <xs:element name="Direction" type="IBIS-IP.string"/>
    </xs:sequence>
  </xs:complexType>

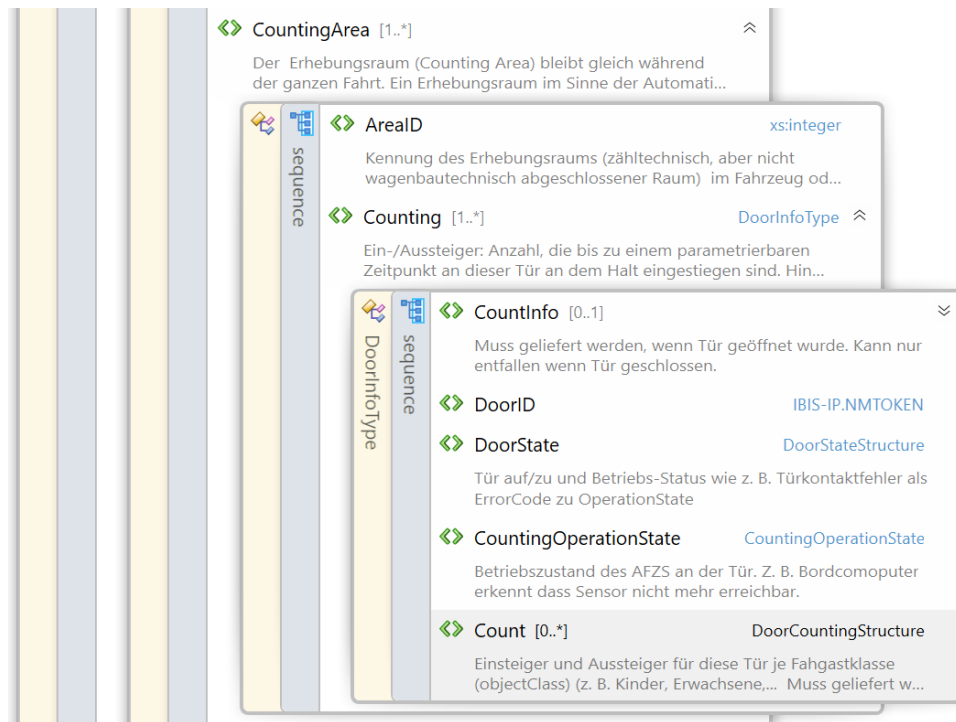
<xs:simpleType name="GNSSQualityEnumeration">
  <xs:restriction base="xs:string">
    <xs:enumeration value="dGPS"/>
    <xs:enumeration value="Estimated"/>
    <xs:enumeration value="GPS"/>
    <xs:enumeration value="NotValid"/>
    <xs:enumeration value="Unknown"/>
  </xs:restriction>
</xs:simpleType>

```


3.2.6 CountingArea

CountingArea enthält mit Elementen der „Counting“ Struktur die eigentliche Zählung während eines Haltes oder Zeitintervalls (bei Wartesaalerfassung) mit den Zählwerten und Statusinformationen aller Türen.

Eine Tür darf in einem PassengerCountingEvent nur einmal vorkommen. Sollte eine Tür innerhalb des Zeitintervalls mehrmals öffnen und schließen, so ist die Zeit der ersten Türöffnung und letzten Türschließung zu verwenden.



PassengerCountingMessage/...

PassengerCountingEvent/CountingArea/AreaID

PassengerCountingEvent/CountingArea/Counting/DoorID/Value
 PassengerCountingEvent/CountingArea/Counting/CountingOperationState
 PassengerCountingEvent/CountingArea/Counting/CountInfo/DoorOpenTime/Value
 PassengerCountingEvent/CountingArea/Counting/CountInfo/TimeStampCountingStart/Value - nein
 PassengerCountingEvent/CountingArea/Counting/CountInfo/DoorClosingTime/Value
 PassengerCountingEvent/CountingArea/Counting/CountInfo/FirstBoardingOrAlighting/Value
 PassengerCountingEvent/CountingArea/Counting/CountInfo/LastBoardingOrAlighting/Value

PassengerCountingEvent/CountingArea/Counting/DoorState/OpenState/Value
 PassengerCountingEvent/CountingArea/Counting/DoorState/OperationState/Value
 PassengerCountingEvent/CountingArea/Counting/Count/ObjectClass
 PassengerCountingEvent/CountingArea/Counting/Count/In/Value
 PassengerCountingEvent/CountingArea/Counting/Count/Out/Value
 PassengerCountingEvent/CountingArea/Counting/Count/CountQuality

ArealID	Muss	Typ Integer. Beim Bus immer „1“. Bei Bahnen fortlaufend 1, 2, 3 je Wagen (während der Fahrt konstant, auch bei Kopfmachen). Busanhänger immer „2“. (bei autonomen AFZS im Busanhänger eigene VehicleID!)
Counting	Muss	Je Tür einmal vorhanden Weglassen für eine Tür, die nicht geöffnet war

Counting/DoorID	Muss	<p>Vorgabe beim Bus: 1, 2, 3: Türen von vorne nach hinten durchnummeriert.</p> <p>Bei Bahnen mit beidseitigen Türen links ungerade und rechts gerade, oder fortlaufend rechts herum, oder 11, 12, 13 links und 21, 22, 23 rechts</p> <p>Projektabsprache erforderlich</p> <p>Datentyp IBIS-IP.NMTOKEN = Buchstabe, Zahlen, Minuszeichen und Punkt. KEINE Leerzeichen und Komma</p> <p>Bei Zählsystemen, die keine türgenaue Erfassung ermöglichen: „0“.</p>
Counting/CountingOperationState	Muss	<p>Betriebszustand des AFZS an der Tür.</p> <p>Beispiel: „normal“</p> <pre><xs:simpleType name="CountingOperationState"> <xs:restriction base="xs:string"> <xs:enumeration value="normal"/> <xs:enumeration value="faulty"/> <xs:enumeration value="configuration problem"/> <xs:enumeration value="sensor covered"/> </xs:restriction> </xs:simpleType></pre> <p>Wenn ein neuer VDV-Standard erweiterte Werte zulässt, sollen diese auch verwendet werden, soweit sie einen Mehrwert bringen. Dieser Grundsatz gilt für Zählsystem im Fahrzeug wie für die weiterverarbeitende Software.</p>
Counting / CountInfo	Muss	<p>Erstes Element in „Counting“ vom Typ DoorInfoType</p> <p>Muss geliefert werden, wenn Tür geöffnet wurde.</p> <p><u>Muss</u> entfallen, wenn Tür geschlossen ist (wichtig zur Unterscheidung echt gezählter 0 Fahrgäste).</p> <p>Es folgen die Elemente in „CountInfo“:</p>
Counting / CountInfo/DoorOpenTime	Muss	<p>je nach Implementierung wird der eine oder andere Zeitpunkt gespeichert.</p> <ul style="list-style-type: none"> - individuelle Türsignale je Tür (bevorzugt) oder - zentrales Türsummensignal (auch möglich) oder - Türfreigabesignal (Straßenbahn). <p>Bei Wartesaalerfassung TimeStampCountingStart verwenden, wenn es sich um das Folge-Event handelt und die Tür bereits offensteht.</p>
Counting / CountInfo/DoorClosingTime	Muss	Siehe DoorOpenTime
Counting / CountInfo/TimeStampCountingStart	Muss	<p>ALTERNATIV zu DoorOpenTime</p> <p>Zeit für den Beginn eines Wartesaalintervalls bei bereits geöffneter Tür.</p>
Counting / CountInfo/TimeStampCountingEnd	Muss	<p>ALTERNATIV zu DoorClosingTime</p> <p>Zeit für das Ende eines Wartesaalintervalls bei noch geöffneter Tür.</p>
Counting / CountInfo/FirstBoardingOrAlighting	optional	Zeitpunkt des ersten Fahrgastes im Zeitfenster dieses Events. Element weglassen, falls das System den Zeitpunkt nicht erfassen kann.

Counting / CountInfo/ LastBoardingOrAlighting	optional	Zeitpunkt des letzten Fahrgastes. Siehe oben
Counting / DoorState	Muss	<p>DoorStateStructure: Tür auf/zu und Betriebsstatus wie z.B. Türkontaktfehler als ErrorCode zu OperationState</p> <pre><xs:complexType name="DoorStateStructure"> <xs:sequence> <xs:element name="OpenState" type="DoorOpenStateStructure"/> <xs:element name="OperationState" type="DoorOperationStateStructure" minOccurs="0"/> </xs:sequence> </xs:complexType></pre> <p>Es folgen die Elemente in „DoorState“:</p>
Counting/ OpenState/Value	Muss	<pre><xs:enumeration value="DoorsOpen"/> <xs:enumeration value="AllDoorsClosed"/> <xs:enumeration value="SingleDoorOpen"/> <xs:enumeration value="SingleDoorClosed"/></pre> <p>Information on the state of the doors in a vehicle</p> <pre><xs:complexType name="DoorOpenStateStructure"> <xs:sequence> <xs:element name="Value" type="DoorOpenStateEnumeration"/> <xs:element name="ErrorCode" type="ErrorCodeEnumeration" minOccurs="0"/> </xs:sequence> </xs:complexType></pre>
Counting/ OperationState/Value	Muss	<pre><xs:enumeration value="Locked"/> <xs:enumeration value="Normal"/> <xs:enumeration value="EmergencyRelease"/></pre> <pre><xs:complexType name="DoorOperationStateStructure"> <xs:sequence> <xs:element name="Value" type="DoorOperationStateEnumeration"/> <xs:element name="ErrorCode" type="ErrorCodeEnumeration" minOccurs="0"/> </xs:sequence> </xs:complexType></pre>
Counting/Count	Muss	<p>DoorCountingStructure: Einsteiger und Aussteiger für diese Tür je Objektklasse (ObjectClass) Muss geliefert werden, wenn Tür geöffnet wurde. Muss entfallen, wenn Tür nicht geöffnet war. Muss für jede konfigurierte Objektklasse geliefert werden. Es folgen die Elemente in „Count“:</p>
Counting / Count / In/Value	Muss	Integer: Anzahl Fahrgäste/Objekte
Counting / Count / Out/Value	Muss	Integer: Anzahl Fahrgäste/Objekte
Counting / Count / ObjectClass	Muss	<p>DoorCountingObjectClassEnumeration: „Adult“, „Child“, „Bike“, „Wheelchair“, „Pram“, „Unidentified“, „Other“. Nur Adult und Child sind Pflicht. Bei Personenzählung ohne Unterscheidung nach Personengröße „Unidentified“ verwenden.</p>

Counting / Count / CountQuality	Muss	<pre><xs:simpleType name="DoorCountingQualityEnumeration"> <xs:restriction base="xs:string"> <xs:enumeration value="Defect"/> <xs:enumeration value="Regular"/> <xs:enumeration value="Sabotage"/> <xs:enumeration value="Other"/> </xs:restriction> </xs:simpleType></pre>
--	------	---

Die Felder DoorOpen/CountingStart bzw. DoorClosing/CountingEnd sind wechselseitig leer und werden von der Situation abhängig befüllt, und zwar stets nur eines von beiden. Das leere Feld wird in der XML-Struktur entsprechend der Definition „xs:choice“ durch Weglassen des Tags abgebildet

XML-Strukturen

```
<xs:complexType name="DoorInfoType">
<xs:sequence>
  <xs:element name="CountInfo" minOccurs="0">
    <xs:annotation>
      Muss geliefert werden, wenn Tür geöffnet wurde. Kann nur entfallen wenn Tür geschlossen
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:choice>
          <xs:element name="DoorOpenTime" type="IBIS-IP.dateTime">
            <xs:annotation>
              Türöffnungszeitpunkts
            </xs:annotation>
          </xs:element>
          <xs:element name="TimeStampCountingStart" type="IBIS-IP.dateTime">
            <xs:annotation>
              Angabe des Startzeitpunkts der Zählphase an dieser Tür, wenn
der Start unabhängig von der Türöffnung erfolgte.</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:choice>
        <xs:choice>
          <xs:element name="DoorClosingTime" type="IBIS-IP.dateTime">
            <xs:annotation>
              Angabe des Türschließungszeitpunkts
            </xs:annotation>
          </xs:element>
          <xs:element name="TimeStampCountingEnd" type="IBIS-IP.dateTime">
            <xs:annotation>
              Angabe des Endzeitpunkts der Zählphase an dieser Tür, wenn die
Beendigung unabhängig von der Türschließung erfolgte
            </xs:annotation>
          </xs:element>
        </xs:choice>
        <xs:element name="FirstBoardingOrAlighting" type="IBIS-IP.dateTime">
          <xs:annotation>
            Zeitpunkt an dem der erste Fahrgast diese Tür durchschritten hat
          </xs:annotation>
        </xs:element>
        <xs:element name="LastBoardingOrAlighting" type="IBIS-IP.dateTime">
          <xs:annotation>
            Zeitpunkt an dem das letzte Zählobjekt erkannt wurde
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DoorID" type="IBIS-IP.NMTOKEN"/>
  <xs:element name="DoorState" type="DoorStateStructure">
    <xs:annotation>
      Tür auf/zu und Betriebs-Status wie z. B. Türkontaktfehler als ErrorCode zu OperationState
    </xs:annotation>
  </xs:element>
  <xs:element name="CountingOperationState" type="CountingOperationState">
    <xs:annotation>
      Vetriebszustand des AFZS an der Tür. Bordcomputer erkennt dass Sensor nicht mehr erreichbar
    </xs:annotation>
  </xs:element>
</xs:sequence>
</xs:complexType>
```

```
<xs:element name="Count" type="DoorCountingStructure" minOccurs="0" maxOccurs="unbounded">
  </xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="DoorStateStructure">
  <xs:sequence>
    <xs:element name="OpenState" type="DoorOpenStateStructure"/>
    <xs:element name="OperationState" type="DoorOperationStateStructure" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="DoorOpenStateStructure">
  <xs:sequence>
    <xs:element name="Value" type="DoorOpenStateEnumeration"/>
    <xs:element name="ErrorCode" type="ErrorCodeEnumeration" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="DoorOperationStateStructure">
  <xs:sequence>
    <xs:element name="Value" type="DoorOperationStateEnumeration"/>
    <xs:element name="ErrorCode" type="ErrorCodeEnumeration" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="DoorOpenStateEnumeration">
  <xs:annotation>
    <xs:documentation>Information on the state of the doors in a vehicle</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="DoorsOpen"/>
    <xs:enumeration value="AllDoorsClosed"/>
    <xs:enumeration value="SingleDoorOpen"/>
    <xs:enumeration value="SingleDoorClosed"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="DoorOperationStateEnumeration">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Locked"/>
    <xs:enumeration value="Normal"/>
    <xs:enumeration value="EmergencyRelease"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="DoorCountingObjectClassEnumeration">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Adult"/>
    <xs:enumeration value="Child"/>
    <xs:enumeration value="Bike"/>
    <xs:enumeration value="WheelChair"/>
    <xs:enumeration value="Pram"/>
    <xs:enumeration value="Unidentified"/>
    <xs:enumeration value="Other"/>
  </xs:restriction>
</xs:simpleType>
```

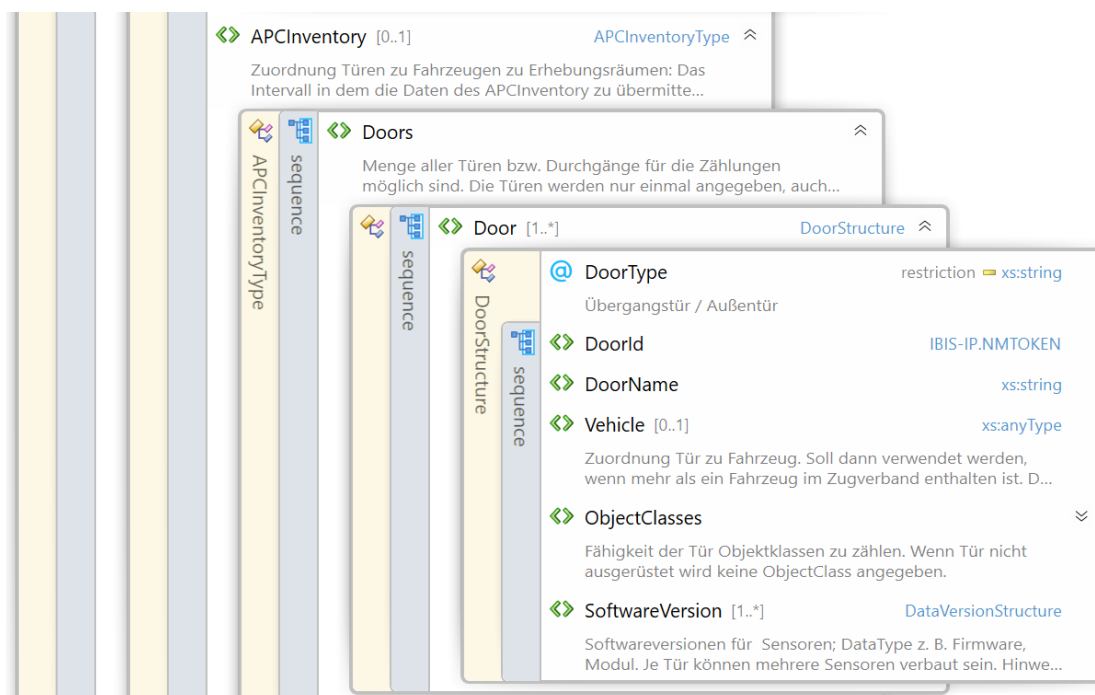
3.2.7 APCInventory

Beschreibt Anzahl der Türen und Softwareversionen der Sensoren

„Muss“-Anforderung

PassengerCountingMessage/PassengerCountingEvent/APCInventory/Statusinformation
 PassengerCountingMessage/PassengerCountingEvent/APCInventory/Doors[]/DoorId
 PassengerCountingMessage/PassengerCountingEvent/APCInventory/Doors[]/DoorName
 PassengerCountingMessage/PassengerCountingEvent/APCInventory/Doors[]/Vehicle
 PassengerCountingMessage/PassengerCountingEvent/APCInventory/Doors[]/ObjectClasses(ObjectClass[]/Value
 PassengerCountingMessage/PassengerCountingEvent/APCInventory/Doors[]/SoftwareVersion/DataType/Value
 PassengerCountingMessage/PassengerCountingEvent/APCInventory/Doors[]/SoftwareVersion/VersionRef/Value
 PassengerCountingMessage/PassengerCountingEvent/APCInventory/Doors[]/DoorType

PassengerCountingMessage/PassengerCountingEvent/APCInventory/Composition[]/VehicleRef
 PassengerCountingMessage/PassengerCountingEvent/APCInventory/Composition[]/Position
 PassengerCountingMessage/PassengerCountingEvent/APCInventory/Composition[]/DirectionInComposition
 PassengerCountingMessage/PassengerCountingEvent/APCInventory/ChangeOfDirection
 PassengerCountingMessage/PassengerCountingEvent/APCInventory/DirectionOfComposition



Doors		Liste aller Türen („Door“-Elemente) DoorStructure
.../DoorType	optional	Immer „Ausstentuer“ oder Element weglassen
.../DoorId	Muss	Nummer wie oben bei CountingArea
.../DoorName	optional	
.../Vehicle	Muss	KFZ-Kennzeichen oder Fahrzeugnummer, siehe oben
.../ObjectClass	optional	Mehrere Elemente möglich Dokumentiert, welche Objektklassen der Sensor erfassen kann (was auf dem Fahrzeug konfiguriert ist) Werte siehe ObjectClass in „Count“
SoftwareVersion	Muss	SoftwareVersion des Zählsensors <pre><xs:complexType name="DataVersionStructure"> <xs:sequence> <xs:element name="DataType" type="IBIS-IP.string"> </xs:element> </xs:sequence> </xs:complexType></pre>

		<pre> <xs:element name="VersionRef" type="IBIS-IP.NMTOKEN"> </xs:element> </xs:sequence> </xs:complexType> </pre> <p>Es folgen die Elemente von „SoftwareVersion“</p>
.../DataType/Value		
../VersionRef/Value		
CountingAreas	optional	Beim Bus alles weglassen, oder immer AreaID „1“
.../AreaID	optional	integer
.../AreaName	optional	String, „1. Klasse“, „Oberdeck“
.../AreaAccess	optional	

XML-Strukturen

```

<xs:complexType name="APCInventoryType">
  <xs:sequence>
    <xs:element name="Doors">
      <xs:annotation>
        <xs:documentation>Menge aller Türen bzw. Durchgänge für die Zählungen möglich sind. Die
Türen werden nur einmal angegeben, auch wenn sie in mehreren Erhebungsräumen enthalten sein sollten.
Die Zuordnung zu Erhebungsräumen erfolgt im Element "CountingAreas"</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Door" type="DoorStructure" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="CountingAreas" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Erhebungsräume</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="CountingArea" maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="AreaID" type="xs:integer">
                  <xs:annotation>
                    <xs:documentation>Erhebungsraum-
ID</xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element name="AreaName"
type="xs:normalizedString" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>Bezeichnung des
Erhebungsraums, kann zur fachlichen Unterscheidung verwendet werden, z.B. "1. Klasse", "Unterdeck",
"Oberdeck"</xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element name="AreaAccess"
maxOccurs="unbounded">
                  <xs:annotation>
                    <xs:documentation>Zugänge (Türen) zum
Erhebungsraum</xs:documentation>
                  </xs:annotation>
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element
name="DoorId" type="IBIS-IP.NMTOKEN"/>
                      <xs:element
name="DirectionInverted" type="xs:boolean" minOccurs="0">
                        <xs:annotation>
                          <xs:documentation>Zählwert (bei Zwischentür) ist als Zutritt / Abgang zu interpretieren. Die

```

Zählungen in der Struktur Erhebungsraum geben die In und Out bezogen auf diesen Erhebungsraum korrekt an. Dies ist eine Zusatzinfo, ob die in und out durch Invertierung der physikalischen Zählungsdaten entstanden sind. Invertierung=1</xs:documentation>

```

</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="DoorStructure">
  <xs:sequence>
    <xs:element name="DoorId" type="IBIS-IP.NMTOKEN"/>
    <xs:element name="DoorName" type="xs:string"/>
    <xs:element name="Vehicle" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Zuordnung Tür zu Fahrzeug. Soll dann verwendet werden, wenn mehr als
ein Fahrzeug im Zugverband enthalten ist. Die Fahrzeuge werden dann im Element "Composition" angegeben.
Fahrzeugbezeichnung für Eisenbahnfahrzeuge UIC-Fahrzeugnummer, für Busse nach Unternehmensvorgabe z. B.
KFZ.Kennezeichen, betriebliche Fahrzeugnummer. </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="ObjectClasses">
      <xs:annotation>
        <xs:documentation>Fähigkeit der Tür Objektklassen zu zählen. Wenn Tür nicht ausgerüstet
wird keine ObjectClass angegeben.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ObjectClass"
type="DoorCountingObjectClassEnumeration" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SoftwareVersion" type="DataVersionStructure" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>Softwareversionen für Sensoren; DataType z. B. Firmware, Modul. Je
Tür können mehrere Sensoren verbaut sein. Hinweis: Das Element SoftwareVersion im HeaderData bezieht
sich auf die OBU</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="DoorType">
    <xs:annotation>
      <xs:documentation>Übergangstür / Außentür</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="Uebergangstuer"/>
        <xs:enumeration value="Aussentuer"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

```

3.2.8 Composition

Nicht benötigt beim Bus

Eventuell bei Anhängerbetrieb sinnvoll, wenn es ein zentrales Steuergerät im Bus gibt, der mit den Sensoren im Anhänger verbunden sind.

3.2.9 Beispiel: Wartesaalerfassung mit Intervallmessung

Das folgende Bild soll verdeutlichen, wie die PassengerCountingEvents (PCE) bei längeren Haltezeiten aufgebaut sein sollen:

HGS-Verortung Haltestelle A														
PCE-Gruppe 1							PCE-Gruppe 2							
mit Wartesaal	PCE 1	PCE 2	PCE 3	PCE 4	PCE 5	PCE 6	PCE 7	PCE 8	PCE 9	PCE 10	PCE 11	PCE 12	PCE 13	PCE 14
<TimeStampEventStart>	07:00:00	07:00:15	07:00:30	07:00:45	07:01:00	07:01:15	07:01:30	07:02:35	07:02:50	07:03:05	07:03:20	07:03:35	07:03:50	07:04:05
<TimeStampEventEnd>	07:00:15	07:00:30	07:00:45	07:01:00	07:01:15	07:01:30	07:01:39	07:02:50	07:03:05	07:03:20	07:03:35	07:03:50	07:04:05	07:04:20
<QueryType>	unknown	unknown	unknown	unknown	unknown	unknown	unknown	unknown	unknown	unknown	unknown	unknown	unknown	unknown
<DoorOpenTime>	07:00:00							07:02:35						
<TimeStampCountingStart>		07:00:15	07:00:30	07:00:45	07:01:00			07:02:50	07:02:50	07:03:05	07:03:20	07:03:35		
<TimeStampCountingEnd>	07:00:15	07:00:30	07:00:45	07:01:00				07:02:50	07:03:05	07:03:20	07:03:35			
<DoorClosingTime>					07:01:15							07:03:50		
<DoorOpenTime>		07:00:22							07:02:57					
<TimeStampCountingStart>			07:00:30							07:03:05				
<TimeStampCountingEnd>		07:00:30							07:03:05					
<DoorClosingTime>			07:00:39							07:03:14				
<DoorOpenTime>				07:00:54							07:03:29			
<TimeStampCountingStart>					07:01:00	07:01:15	07:01:30					07:03:35	07:03:50	
<TimeStampCountingEnd>				07:01:00	07:01:15	07:01:30					07:03:35	07:03:50	07:04:05	
<DoorClosingTime>							07:01:39							07:04:20
Tür 1:														
Tür 2:														
Tür 3:														
Globaler Wartesaalzeitraum aus Sicht des HGS, solange der Start oder Open innerhalb von 3 min liegt														
Von der globalen Einsteigerzahl abzuziehende Aussteigerzahlen														

Das Beispiel zeigt 14 Zeitfenster à 15 Sekunden an einem Ort. Jedes Zeitfenster ist in einer PCE-Struktur gespeichert mit den jeweils relevanten Zeitstempeln. Die 14 Zeitfenster bilden gedanklich zwei PCE-Gruppen, da zwischenzeitlich alle Türen geschlossen wurden und mit der erneuten Türöffnung um 07:02:35 eine neue 15-Sekunden-Erfassung startet. Es ist auch möglich, dass das Fahrzeug zwischen Gruppe 1 und 2 vorgerückt ist oder umgesetzt wurde, etwa von einer Ankunfts- zur Abfahrtsposition.

Hinweis für das Zählsystem im Fahrzeug:

Wichtig ist, dass TimeStampEventStart und TimeStampEventEnd die Zählintervalle genau abbilden, hier 15 Sekunden. Die Felder DoorOpen/CountingStart bzw. DoorClosing/CountingEnd sind wechselseitig leer und werden von der Situation abhängig befüllt, und zwar stets nur eines von beiden. Das leere Feld wird in der XML-Struktur entsprechend der Definition „xs:choice“ durch Weglassen des Tags abgebildet.

Ein PCE kann von der Intervall-Aufzeichnung abweichen und länger aufgezeichnet werden, wenn in dem gesamten PCE-Zeitraum keine Ein- oder Ausstiege erfasst werden. Auf diese Weise lassen sich besonders bei langen Türöffnungen ohne Fahrgastwechsel unnötige PCE-Datensätze sparen. Ein Weglassen der PCEs ohne Fahrgastwechsel darf nicht erfolgen, um die Datenvollständigkeit bezüglich offener Türzustände zu wahren.

Beispiel für verlängerten Zeitraum 2 ohne Fahrgäste:

PCE-Nr	PCE-Start	PCE-Ende	Ein	Aus
1	07:00:00	07:00:15	0	6
2	07:00:15	07:03:00	0	0
3	07:03:00	07:03:10	1	3

Hinweis für die Weiterverarbeitung im Hintergrundsystem:

Die Länge des ersten Wartesaal-Zeitfensters wird projektspezifisch festgelegt und liegt zwischen 1 und 3 Minuten. Das Intervall für die Aufzeichnung der Zwischenergebnisse ist ebenfalls projektspezifisch und liegt in der Regel bei 15 Sekunden. Für die Zuordnung der PCE zum ersten bzw. zweiten Zeitfenster gilt die folgende Regel: Ein PCE gehört zum ersten Zeitfenster, solange TimeStampEventStart innerhalb des ersten Zeitfensters liegt. Im obigen Beispiel ist dies PCE 9 aus Gruppe 2 mit Start=07:02:50.

Innerhalb einer Gruppe passen die Intervalle (EventStart/EventEnd) genau zusammen. Die Zusammenfassung der beiden Gruppen zu einem Halt ergibt sich daraus, dass beide Gruppen jeweils für sich aufgrund der Ortungsdaten (GPS) räumlich derselben Haltestelle zugeordnet wurden.

Bei der zeitlichen Zuordnung des Haltes zum Fahrplan muss der gesamte Zeitraum beider Gruppen herangezogen werden, auch wenn Teile dieses Zeitraums über etwaige zeitliche Ortungsgrenzen hinausgehen. Der Halt mit allen Einsteigern und Aussteigern beider Gruppen ist in der Folge ebenfalls als Ganzes zu betrachten, insbesondere an Endpunkten: Es ist zunächst eine Wartesaalverrechnung vorzunehmen und erst danach sind die Aussteiger der ankommenden und die Einsteiger der abgehenden Fahrt zuzuordnen.

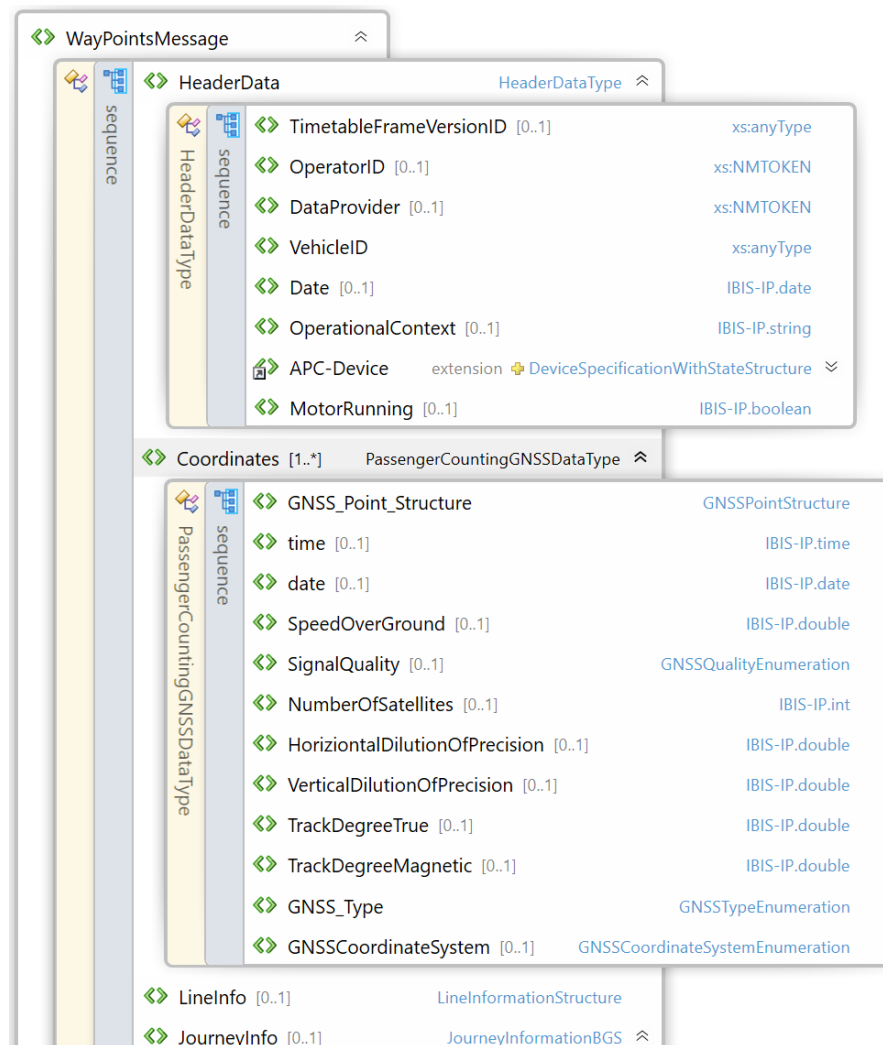
4 WayPointsMessage (WPM)

„Muss“-Anforderung bei allen AFZS

Keine Muss-Anforderung bei manuellen Zählrodaten mit JourneyInfo, Line und StopInformation

Separate Datei mit „wpm“ im Dateinamen

```
<xs:element name="WayPointsMessage">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="HeaderData" type="HeaderDataType"/>
      <xs:element name="Coordinates" type="PassengerCountingGNSSDataType"
        maxOccurs="unbounded"/>
      <xs:element name="LineInfo" type="LineInformationStructure" minOccurs="0"/>
      <xs:element name="JourneyInfo" type="JourneyInformationBGS" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```



4.1 HeaderData

„Muss“-Anforderung

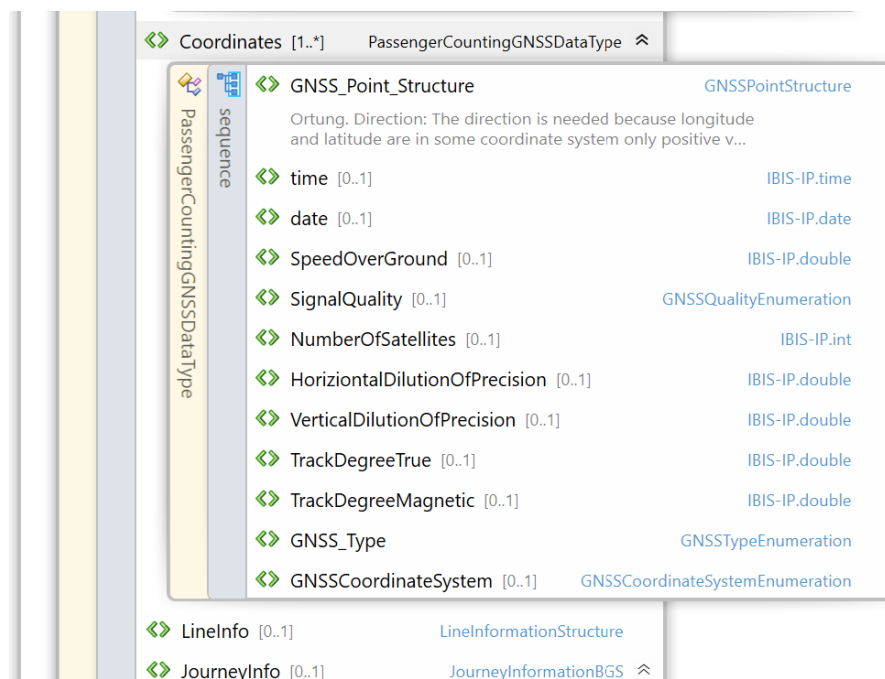
Siehe Kapitel 3.1

4.2 Coordinates

„Muss“-Anforderung

Alle 5 bis 15 Sekunden oder spätestens nach 50 Metern oder eine Kombination, auch während das Fahrzeug steht.

PassengerCountingGNSSDataType, Inhalte wie in GNSS Kapitel 3.2.5



4.3 LineInfo

Kann hier entfallen, ist bereits in PassengerCountingEvent enthalten

Siehe Kapitel 3.2.3

4.4 JourneyInfo

Kann hier entfallen, ist bereits in PassengerCountingEvent enthalten

Siehe Kapitel 3.2.2

5 Anhang: XML Basistypen

5.1 Enumerations aus PassengerCountingServiceBGS

Die in den XSD definierten Enumerations sind verbindlich zu verwenden.

5.1.1 PassengerCounting

<!--==== Definition of Enumerations BGS-->

```
<xs:simpleType name="Verkehrstraeger">
  <xs:restriction base="xs:string">
    <xs:enumeration value="STRAB"/>
    <xs:enumeration value="UBAHN"/>
    <xs:enumeration value="BUS"/>
    <xs:enumeration value="BAHN"/>
  </xs:restriction>
</xs:simpleType>
```

Für QueryType:

```
<xs:simpleType name="Abfragetyp">
  <xs:annotation>
    <xs:documentation>Im einfachen Busverkehr ohne Wartesaaleffekt oder Übergangstüren wird
    grundsätzlich 'departure' gesendet, auch wenn laut Fahrplan keine Haltestelle vorliegt.
  </xs:documentation>
</xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="arrival"/>
    <xs:enumeration value="at stop"/>
    <xs:enumeration value="departure"/>
    <xs:enumeration value="between stops"/>
    <xs:enumeration value="unknown"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="Fahrart">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Betriebsfahrt"/>
    <xs:enumeration value="produktive Fahrt"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="CountingOperationState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="normal"/>
    <xs:enumeration value="faulty"/>
    <xs:enumeration value="configuration problem"/>
    <xs:enumeration value="sensor covered"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="DirectionInComposition">
  <xs:restriction base="xs:string">
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="technical direction"/>
    <xs:enumeration value="against technical direction"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="DoorCountingQualityEnumeration">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Defect"/>
    <xs:enumeration value="Regular"/>
    <xs:enumeration value="Sabotage"/>
    <xs:enumeration value="Other"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="DoorOpenStateEnumeration">
  <xs:annotation>
    <xs:documentation>Information on the state of the doors in a vehicle</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="DoorsOpen"/>
    <xs:enumeration value="AllDoorsClosed"/>
  </xs:restriction>
</xs:simpleType>
```

```

        <xs:enumeration value="SingleDoorOpen"/>
        <xs:enumeration value="SingleDoorClosed"/>
    </xs:restriction>
</xs:simpleType>

```

5.1.2 GNSS

```

<xs:simpleType name="GNSSCoordinateSystemEnumeration">
  <xs:annotation>
    <xs:documentation>Information on the GNSS-Coordinate-System</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="CH1903"/>
    <xs:enumeration value="ETSR89"/>
    <xs:enumeration value="IERS"/>
    <xs:enumeration value="NAD27"/>
    <xs:enumeration value="NAD83"/>
    <xs:enumeration value="WGS84"/>
    <xs:enumeration value="WGS72"/>
    <xs:enumeration value="SGS85"/>
    <xs:enumeration value="P90"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="GNSSQualityEnumeration">
  <xs:annotation>
    <xs:documentation>Information on the GNSS-Quality</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="dGPS"/>
    <xs:enumeration value="Estimated"/>
    <xs:enumeration value="GPS"/>
    <xs:enumeration value="NotValid"/>
    <xs:enumeration value="Unknown"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="GNSSTypeEnumeration">
  <xs:annotation>
    <xs:documentation>Information on the type of GNSS</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="GPS"/>
    <xs:enumeration value="Glonass"/>
    <xs:enumeration value="Galileo"/>
    <xs:enumeration value="Beidou"/>
    <xs:enumeration value="IRNSS"/>
    <xs:enumeration value="other"/>
    <xs:enumeration value="DeadReckoning"/>
    <xs:enumeration value="MixedGNSSTypes"/>
  </xs:restriction>
</xs:simpleType>

```

5.2 XML allgemein

<xs:element name="note">	defines the element called "note"
<xs:complexType>	the "note" element is a complex type
<xs:sequence>	the complex type is a sequence of elements
<xs:element name="to" type="xs:string">	the element "to" is of type string (text)
<xs:element name="from" type="xs:string">	the element "from" is of type string
<xs:element name="heading" type="xs:string">	the element "heading" is of type string
<xs:element name="body" type="xs:string">	the element "body" is of type string

xs:choice Nur eines der Elemente darf in der XML-Datei vorkommen

xs:string

xs:decimal

xs:integer

xs:boolean true, false, 1 (for true), 0 (for false)

xs:date	Format YYYY-MM-DD, genauer: [-]CCYY-MM-DDThh:mm:ss[Z](+ -)hh:mm] Die Zeitzone kann als »Z« (UTC) oder als »(+ -)hh:mm« angegeben werden. Nicht angegebene Zeitzonen werden als »unbestimmt« betrachtet Gültige Werte sind unter anderem: "2001-10-26", aber auch "2001-10-26+02:00", "2001-10-26Z" („Z“ steht für UTC)
xs:time	Format hh:mm:ss[Z](+ -)hh:mm] Gültige Werte sind unter anderem 2001-10-26, 2001-10-26+02:00, 2001-10-26Z, 2001-10-26+00:00
xs:dateTime	Format [-]CCYY-MM-DDThh:mm:ss[Z](+ -)hh:mm]«. Die Zeitzone kann als »Z« (UTC) oder als »(+ -)hh:mm« angegeben werden. Nicht angegebene Zeitzonen werden als »unbestimmt« betrachtet Beispiele: "2001-10-26T21:32:52", "2001-10-26T21:32:52+02:00", "2001-10-26T19:32:52Z", "2001-10-26T19:32:52+00:00", "-2001-10-26T21:32:52" oder "2001-10-26T21:32:52.12679"
xs:NMTOKEN	Buchstabe, Zahlen, Minuszeichen und Punkt. KEINE Leerzeichen und Komma!

5.3 VDV IBIS-IP Basistypen

Diese Typen haben stets das Element „Value“. Daher ist „Value“ in sehr vielen Fällen der letzte Bezeichner in den XML-Dateien. „ErrorCode“ ist jeweils optional und wird daher fast immer weggelassen.

```

<xs:complexType name="IBIS-IP.string">
  <xs:sequence>
    <xs:element name="Value" type="xs:string"/>
    <xs:element name="ErrorCode" type="ErrorCodeEnumeration" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="DegreeType">
  <xs:sequence>
    <xs:element name="Degree" type="IBIS-IP.double"/>
    <xs:element name="Orientation" type="IBIS-IP.string"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="IBIS-IP.anyURI">
  <xs:sequence>
    <xs:element name="Value" type="xs:anyURI"/>
    <xs:element name="ErrorCode" type="ErrorCodeEnumeration" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="IBIS-IP.boolean">
  <xs:sequence>
    <xs:element name="Value" type="xs:boolean"/>
    <xs:element name="ErrorCode" type="ErrorCodeEnumeration" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="IBIS-IP.byte">
  <xs:sequence>
    <xs:element name="Value" type="xs:byte"/>
    <xs:element name="ErrorCode" type="ErrorCodeEnumeration" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="IBIS-IP.date">
  <xs:sequence>
    <xs:element name="Value" type="xs:date"/>
    <xs:element name="ErrorCode" type="ErrorCodeEnumeration" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="IBIS-IP.dateTime">
  <xs:sequence>
    <xs:element name="Value" type="xs:dateTime"/>
    <xs:element name="ErrorCode" type="ErrorCodeEnumeration" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="IBIS-IP.int">
  <xs:sequence>

```

```
        <xs:element name="Value" type="xs:int"/>
        <xs:element name="ErrorCode" type="ErrorCodeEnumeration" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="IBIS-IP.double">
    <xs:sequence>
        <xs:element name="Value" type="xs:double"/>
        <xs:element name="ErrorCode" type="ErrorCodeEnumeration" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="IBIS-IP.NMTOKEN">
    <xs:sequence>
        <xs:element name="Value" type="xs:NMTOKEN"/>
        <xs:element name="ErrorCode" type="ErrorCodeEnumeration" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="IBIS-IP.nonNegativeInteger">
    <xs:sequence>
        <xs:element name="Value" type="xs:nonNegativeInteger"/>
        <xs:element name="ErrorCode" type="ErrorCodeEnumeration" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="IBIS-IP.normalizedString">
    <xs:sequence>
        <xs:element name="Value" type="xs:normalizedString"/>
        <xs:element name="ErrorCode" type="ErrorCodeEnumeration" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
```