
**Echtzeit Kommunikations- und Aus-
kunftsplattform EKAP**

Teil 2: EKAP-Schnittstellenbeschreibung

Gesamtbearbeitung

Ausschuss für Kundenservice, -information und -dialog (K3),
Fachausschuss für Telematik und Informationssysteme (ATI) und
Unterausschuss „intermodal transport control system“ (UA itcs)

Sachbearbeitung

Dipl.-Ing. Berthold Radermacher, VDV, Köln
Dipl.-Ing. (FH) Andreas Wehrmann, VDV, Köln
AK3 des IP-KOM-ÖV Forschungsprojektes

Gefördert durch:



Bundesministerium
für Wirtschaft
und Energie

aufgrund eines Beschlusses
des Deutschen Bundestages

Das dieser VDV-Schrift zugrundeliegende Vorhaben IP-KOM-ÖV wurde mit Mitteln des Bundesministeriums für Wirtschaft und Energie unter dem Förderkennzeichen 19P10003 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Bearbeiter im AK 3 des IP-KOM-ÖV Forschungsprojektes:

u. a.:

Dipl.-Inform. Anselmo Stelzer, TU Darmstadt, Darmstadt

M. Sc. Frank Englert, TU Darmstadt, Darmstadt

Dipl.-Inf. Christine Keller, TU Dresden, Dresden

Dipl.-Medieninf. Romina Kühn, TU Dresden, Dresden

Dipl.-Inf. Katja Tietze, TU Dresden, Dresden

Dipl.-Math. Peter von Grumbkow, HaCon Ing.-Ges. mbH, Hannover

Dipl.-Ing. Dirk Weißer, Init, Karlsruhe

Dipl.-Ing. Stephan Hörold, TU Ilmenau, Ilmenau

Dipl.-Medienwiss. Cindy Mayas, TU Ilmenau, Ilmenau

Dipl.-Inform. Günther Gruber, Mentz Datenverarbeitung GmbH, München

Dipl.-Math. Werner Kohl, Mentz Datenverarbeitung GmbH, München

Dipl.-Ing. ETH Walter Meier-Leu, Weisskopf Engineering AG, Schaffhausen

Inhaltsverzeichnis

Inhaltsverzeichnis	3
Vorwort	14
1. Einleitung	16
2. Anwendungsbereich	17
3. Notation der XML-Elemente und -Strukturen	17
3.1. Darstellung von XML-Elementen im Text	17
3.2. Darstellung von Beziehungen	17
3.3. Tabellennotation von XML-Strukturen	18
3.3.1. Gruppierung	19
3.3.2. Elementname	19
3.3.3. Multiplizität & Choice (Min:Max)	19
3.3.4. Datentyp	20
3.3.5. Erläuterung	20
4. Nachrichtenübermittlung	20
4.1. Einsatz der SIRI-Verfahren	20
4.2. HTTP und REST	22
4.3. Rollen von Server und Client	23
5. Identifikation von Objekten über Systemgrenzen hinweg	23
5.1. Haltestellen und Haltepunkte	23
5.2. Orte und Gemeinden	24
5.3. Adressen und POIs	25
5.4. Verkehrsunternehmen und Verkehrsverbünde	25
5.5. Linien und Linienrichtungen	25
5.6. Fahrten	26
5.7. Fahrzeuge	27
5.8. Fahrzeugtypen	27
5.9. Verkehrstage	28
5.10. Eigentümer	28
5.11. Haltestellen- und Fahrzeugeinrichtungen	28
5.12. Teilnehmende Systeme / IT-Systeme	28
5.13. Ereignismeldungen	29
5.14. Tarifverantwortliche	29
5.15. Tarifzonen	29

5.16. Fahrscheine und Vielfahrerkarten	29
6. Dienste und XML-Schemata	29
6.1. Bereitgestellte Dienste	30
6.2. Dienstübergreifend genutzte XML-Schemata	31
6.3. Importierte Schemata	34
6.4. Fehlerzustände beim Betrieb von TRIAS-Diensten	34
6.4.1. Fehlercodes aus SIRI	34
6.4.2. Allgemeine TRIAS-Fehlerzustände	35
7. Gemeinsam genutzte XML-Strukturen	35
7.1. Trias, das Wurzelement	35
7.1.1. ServiceRequestStructure	36
7.1.2. SubscriptionRequestStructure	36
7.1.3. RequestPayloadStructure	37
7.1.4. ServiceDeliveryStructure	38
7.1.5. DeliveryPayloadStructure	38
7.2. Trias_UTILITY	39
7.2.1. Einfache Typen	39
7.2.2. InternationalTextStructure	40
7.2.3. GeoPositionStructure	40
7.3. Trias_ModesSupport	40
7.3.1. Einfache Typen	40
7.3.2. IndividualTransportOptionsStructure	42
7.3.3. PtSubmodeChoiceGroup	43
7.3.4. ModeStructure	43
7.3.5. PtModeFilterStructure	43
7.4. Trias_Common	44
7.4.1. Einfache Typen	44
7.4.2. ErrorMessageStructure	44
7.4.3. PrivateCodeStructure	44
7.4.4. OperatorFilterStructure	44
7.4.5. LineDirectionStructure	45
7.4.6. LineDirectionFilterStructure	45
7.4.7. OperatingDaysStructure	45
7.4.8. WeekdayTimePeriodStructure	45
7.4.9. GeneralAttributeStructure	46

7.5. Trias_LocationSupport	46
7.5.1. Einfache Typen	46
7.5.2. StopPointStructure	47
7.5.3. StopPlaceStructure	47
7.5.4. LocalityStructure	48
7.5.5. PointOfInterestStructure	48
7.5.6. AddressStructure	48
7.5.7. LocationStructure	49
7.5.8. LocationRefStructure	49
7.6. Trias_JourneySupport	49
7.6.1. ServiceViaPointStructure	50
7.6.2. ViaStructure	50
7.6.3. DatedJourneyStructure	50
7.6.4. TripLocationStructure	51
7.6.5. CallAtStopStructure	51
7.6.6. DatedCallAtLocationStructure	52
7.6.7. ContinuousServiceStructure	53
7.6.8. VehiclePositionStructure	54
7.6.9. ProgressBetweenStopsStructure	54
7.6.10. LocationContextStructure	54
7.6.11. AbstractResponseContextStructure	55
7.6.12. LegAttributeStructure	55
7.7. Trias_FacilitySupport	55
7.7.1. siri:CommonFacilityGroup	55
7.7.2. siri:StopFacilityGroup	57
7.7.3. siri:ServiceFacilityGroup	58
7.7.4. siri:AllFacilitiesGroup	58
7.8. Trias_SituationSupport	59
7.8.1. SituationsStructure	59
7.8.2. SituationFullRefStructure	59
7.9. Trias_RequestSupport	60
7.9.1. Einfache Typen	60
7.9.2. AbstractTriasServiceRequestStructure	60
7.9.3. AbstractTriasSubscriptionRequestStructure	60
7.9.4. AbstractTriasResponseStructure	61

7.10. Trias_FaresSupport	62
7.10.1. Einfache Typen	62
7.10.2. FareZoneListInAreaStructure	63
7.10.3. BookingInfoStructure	63
7.10.4. TicketStructure	63
7.10.5. TripFaresResultStructure	64
7.10.6. FaresPassengerStructure	64
7.10.7. FaresParamStructure	65
8. Ortsinformationsdienst	65
8.1. Beschreibung	65
8.2. Einfache Datentypen	65
8.3. Anfragestrukturen	65
8.3.1. LocationInformationRequestStructure	66
8.3.2. InitialLocationInputStructure	66
8.3.3. GeoRestrictionsStructure	66
8.3.4. GeoCircleStructure	67
8.3.5. GeoRectangleStructure	67
8.3.6. GeoAreaStructure	67
8.3.7. LocationParamStructure	67
8.4. Antwortstrukturen	68
8.4.1. LocationInformationResponseStructure	68
8.4.2. LocationResultStructure	69
9. Dienst Verbindungsauskunft	69
9.1. Beschreibung	69
9.2. Anfragestrukturen	69
9.2.1. TripRequestStructure	69
9.2.2. TripParamStructure	69
9.2.3. NumberOfResultsGroup	71
9.2.4. NotViaStructure	71
9.2.5. NoChangeAtStructure	71
9.3. Antwortstrukturen	71
9.3.1. TripResponseStructure	72
9.3.2. TripResponseContextStructure	72
9.3.3. TripResultStructure	72
9.3.4. TripStructure	73

9.3.5. TripLegStructure	74
9.3.6. TimedLegStructure	74
9.3.7. InterchangeLegStructure	74
9.3.8. ContinuousLegStructure	75
9.3.9. LegBoardStructure	76
9.3.10. LegAlightStructure	76
9.3.11. LegIntermediateStructure	77
9.3.12. LegTrackStructure	78
9.3.13. TrackSectionStructure	78
9.3.14. NavigationPathStructure	78
9.3.15. NavigationSectionStructure	79
9.3.16. AccessPathStructure	79
10. Dienst Abfahrtstafeln	80
10.1. Beschreibung	80
10.2. Anfragestrukturen	80
10.2.1. StopEventRequestStructure	80
10.2.2. StopEventParamStructure	80
10.3. Antwortstrukturen	81
10.3.1. StopEventResponseStructure	81
10.3.2. StopEventResponseContextStructure	81
10.3.3. StopEventResultStructure	81
10.3.4. StopEventStructure	82
10.3.5. CallAtNearStopStructure	82
11. Dienst Logische Ortung	82
11.1. Beschreibung	82
11.2. Anfragestrukturen	83
11.2.1. PositioningRequestStructure	83
11.2.2. PositioningParamStructure	83
11.2.3. TimedPositionStructure	83
11.2.4. TimedStopStructure	83
11.3. Antwortstrukturen	84
11.3.1. PositioningResponseStructure	84
11.3.2. PositioningResultStructure	85
11.3.3. PositioningStructure	85
11.3.4. RankedPositionStructure	85

12. Dienst Fahrtinformation (EKAP)	85
12.1. Beschreibung	85
12.2. Anfragestrukturen	85
12.2.1. TripInfoRequestStructure	85
12.2.2. TripInfoParamStructure	86
12.3. Antwortstrukturen	86
12.3.1. TripInfoResponseStructure	86
12.3.2. TripInfoResponseContextStructure	87
12.3.3. TripInfoResultStructure	87
13. Anschlussdienste	87
13.1. Beschreibung	87
13.1.1. Dienst Anschlussvoranmeldung	89
13.1.2. Dienst Anschlussstatus	89
13.1.3. Dienst Info bei Anschlussverlust	90
13.1.4. Dienst Anschlussrückmeldung	90
13.2. Einfache Typen	90
13.3. Komplexe Strukturen	90
13.3.1. DatedConnectionStructure	90
13.3.2. FeederDistributorStructure	91
13.3.3. GeneralizedConnectionStructure	91
13.3.4. ConnectionStatusStructure	91
13.3.5. RecommendationStructure	91
13.4. Anfragestrukturen Anschlussmeldung	92
13.4.1. ConnectionDemandRequestStructure	92
13.4.2. ConnectionDemandDeleteRequestStructure	92
13.5. Antwortstrukturen Anschlussmeldung	92
13.5.1. ConnectionDemandResponseStructure	92
13.5.2. ConnectionDemandDeleteResponseStructure	93
13.6. Anfragestrukturen Anschlussstatus	93
13.6.1. ConnectionStatusRequestStructure	93
13.6.2. ConnectionStatusNotificationStructure	93
13.7. Antwortstrukturen Anschlussstatus	94
13.7.1. ConnectionStatusResponseStructure	94
13.8. Anfragestrukturen Anschlussrückmeldung	94

13.8.1. ConnectionReportRequestStructure	94
13.9. Antwortstrukturen Anschlussrückmeldung	95
13.9.1. ConnectionReportResponseStructure	95
14. Dienst Fahrpreis- und Tarifberechnung	95
14.1. Beschreibung	95
14.2. Anfragestrukturen	95
14.2.1. FaresRequestStructure.....	95
14.2.2. StopFaresRequestStructure.....	96
14.2.3. StaticFaresRequestStructure	96
14.2.4. TripFaresRequestStructure	96
14.2.5. MultiTripFaresRequestStructure.....	96
14.3. Antwortstrukturen	97
14.3.1. FaresResponseStructure.....	97
14.3.2. FaresResultStructure.....	97
14.3.3. StopFaresResultStructure	97
14.3.4. StaticFaresResultStructure	98
14.3.5. TripTicketReferenceStructure	98
14.3.6. MultiTripFaresResultStructure.....	98
15. Dienst Buchungsinformationen.....	99
15.1. Beschreibung	99
15.2. Anfragestrukturen	99
15.2.1. BookingInfoRequestStructure	99
15.3. Antwortstrukturen	99
15.3.1. BookingInfoResponseStructure	99
15.3.2. BookingInfoResultStructure	100
16. Dienst IV-Routing.....	100
16.1. Einfache Typen.....	100
16.2. Anfragestrukturen	100
16.2.1. IndividualRouteRequestStructure.....	100
16.2.2. IndividualTripParamStructure	101
16.2.3. IndividualRouteContextStructure	102
16.3. Antwortstrukturen	102
16.3.1. IndividualRouteResponseStructure.....	102
16.3.2. RouteResultStructure	103
16.3.3. RouteStructure	104

17. Dienst Kartendienst	104
17.1. Einfache Datentypen	104
17.2. Anfragestrukturen	104
17.2.1. MapServiceRequestStructure	104
17.2.2. ImageCoordinatesRequestStructure	105
17.2.3. GeoCoordinatesRequestStructure	105
17.2.4. MapAspectStructure	106
17.2.5. MapSizeStructure	106
17.2.6. MapCoordinateStructure	106
17.3. Antwortstrukturen	106
17.3.1. MapServiceResponseStructure	106
17.3.2. ImageCoordinatesResponseStructure	107
17.3.3. GeoCoordinatesResponseStructure	107
17.3.4. MapResultStructure	108
17.3.5. ImagePointResultStructure	108
17.3.6. GeoCoordinateResultStructure	109
18. Dienst Schadensmeldung / Zustand von Einrichtungen	109
18.1. Beschreibung	109
18.2. Einfache Typen	109
18.3. Komplexe Strukturen	110
18.3.1. FacilityStructure	110
18.3.2. VehicleFacilityStructure	110
18.3.3. InfrastructureFacilityStructure	110
18.3.4. VehicleFacilityRefGroup	110
18.3.5. InfrastructureFacilityRefGroup	111
18.3.6. FacilityDataFilterGroup	111
18.3.7. FacilityStatusStructure	111
18.4. Anfrage Schadensmeldung	112
18.4.1. FacilityStatusReportStructure	112
18.5. Antwort Schadensmeldung	112
18.5.1. FacilityStatusReportResponseStructure	112
18.6. Anfrage Zustand von Einrichtungen	113
18.6.1. FacilityRequestStructure	113
18.7. Antwort Zustand von Einrichtungen	113
18.7.1. FacilityResponseStructure	113

18.7.2. FacilityResultStructure	114
19. Dienst Benachrichtigungsdienst	114
19.1. Beschreibung	114
19.2. Komplexe Strukturen	115
19.2.1. TripMonitoringParamStructure	115
19.3. Anfragestrukturen	116
19.3.1. TripMonitoringSubscriptionRequestStructure	116
19.4. Antwortstrukturen	116
19.4.1. TripMonitoringDeliveryStructure	116
20. Dienst Personalisierungsdienst	117
20.1. Beschreibung	117
20.2. Interaktionen	118
20.2.1. Lebenszyklus eines Wertes	118
20.2.2. Werteliste ermitteln	119
20.2.3. Werte speichern und abrufen	119
20.3. Einfache Typen	121
20.4. Anfragestrukturen	122
20.4.1. PersonalisationRequestStructure	122
20.4.2. PersonalisationSaveValueRequestStructure	122
20.4.3. PersonalisationRetrieveValueRequestStructure	122
20.4.4. PersonalisationDeleteValueRequestStructure	122
20.4.5. PersonalisationEnumerateValuesRequestStructure	123
20.5. Antwortstrukturen	123
20.5.1. PersonalisationResponseStructure	123
20.5.2. PersonalisationSaveValueResponseStructure	123
20.5.3. PersonalisationRetrieveValueResponseStructure	124
20.5.4. PersonalisationDeleteValueResponseStructure	124
20.5.5. PersonalisationEnumerateValuesResponseStructure	124
21. Dienst Fahrzeuginformationen	124
21.1. Beschreibung	124
21.2. Anfragestrukturen	125
21.2.1. VehicleDataRequestStructure	125
21.3. Antwortstrukturen	125
21.3.1. VehicleDataResponseStructure	125
21.3.2. VehicleStatusStructure	125

21.3.3. VehicleActivityStructure	125
21.3.4. StopSequenceStructure	126
21.3.5. StopInformationStructure	126
21.3.6. DisplayContentStructure	126
21.3.7. LineInformationStructure	127
21.3.8. DestinationStructure	127
21.3.9. ViaPointStructure	127
21.3.10. AnnouncementStructure	127
21.3.11. ConnectionStructure	128
21.3.12. VehicleTypeStructure	128
22. Dienst Haltewunsch an Fahrzeug	128
22.1. Beschreibung	128
22.2. Anfragestrukturen	129
22.2.1. StopRequestRequestStructure	129
22.3. Antwortstrukturen	129
22.3.1. StopRequestResponseStructure	129
23. Dienst Diensteregister	129
23.1. Beschreibung	129
23.2. Einfache Typen	129
23.3. Anfragestrukturen	130
23.3.1. ServiceRegisterRequestStructure	130
23.3.2. ServiceRegisterParamStructure	130
23.3.3. ServiceRegisterRegisterRequestStructure	130
23.3.4. ServiceRegisterUpdateRequestStructure	131
23.3.5. ServiceRegisterLookupRequestStructure	131
23.3.6. ServiceRegisterUnregisterRequestStructure	131
23.4. Antwortstrukturen	131
23.4.1. ServiceRegisterResponseStructure	131
23.4.2. ServiceRegisterResultStructure	132
23.4.3. ServiceRegisterRegisterResponseStructure	132
23.4.4. ServiceRegisterUpdateResponseStructure	132
23.4.5. ServiceRegisterLookupResponseStructure	133
23.4.6. ServiceRegisterUnregisterResponseStructure	133
23.4.7. TriasServiceStructure	133

24. Dienst Authentifizierung.....	133
25. Normative Referenzen	136
26. Begriffe	136
27. Abkürzungen.....	136
28. Abbildungsverzeichnis.....	136
29. Tabellenverzeichnis	137
30. Literaturverzeichnis.....	145

GELBEDRUCK

Vorwort

Auf Initiative des VDV und gefördert durch das BMWi begann im September 2010 das Forschungs- und Standardisierungsprojekt

Internet Protokoll basierte Kommunikationsdienste im öffentlichen Verkehr (IP-KOM-ÖV).

Das Projekt wird von 14 Partnern aus Industrie, Universitäten und Verkehrsunternehmen getragen. Es dient der Erarbeitung moderner Kommunikationskonzepte für die umfassende und kontinuierliche Fahrgastinformation.

Eine umfassende Fahrgastinformation stellt heutzutage ein entscheidendes Wettbewerbsmerkmal im öffentlichen Personenverkehr dar, nicht nur im Vergleich mit anderen Verkehrsunternehmen sondern auch im Vergleich zum Individualverkehr.

Bereits heute ist es üblich, dass Verkehrsunternehmen ihre Fahrgäste nicht nur über die geplanten Fahrten informieren, sondern auch Echtzeitinformationen z. B. zu Verspätungen, Störungen oder Fahrtzieländerungen bereitstellen. Diese Informationen werden zum einen über öffentliche Anzeiger bzw. Ansagen in Fahrzeugen oder an Haltestellen allen dort befindlichen Personen zur Verfügung gestellt. Zum anderen lassen sich solche Informationen mit speziellen Applikationen oder über Web-Angebote individuell abfragen.

Bislang ist es aber nicht möglich, Fahrgäste im öffentlichen Verkehr direkt mit Informationen zu ihrer persönlich relevanten Fahrt zu versorgen, den Fahrgast also auch im Störfall mit Hilfe des öffentlichen Verkehrs auf dem schnellsten Weg zu seinem Ziel zu führen.

Die weit verbreiteten Smartphones und Tablets bieten hierfür vielfältige Möglichkeiten und ermöglichen eine hohe Akzeptanz der Benutzer. Die Informationsübertragung erfolgt dabei IP-basiert und sollte bevorzugt zwischen einem zentralen Informations-Server und dem Kundenendgerät erfolgen. Für den Fall, dass der zentrale Datenserver nicht erreichbar ist, sollte auch eine Kommunikation zwischen Kundenendgerät und Fahrzeug möglich sein.

Das Forschungs- und Standardisierungsprojekt IP-KOM-ÖV arbeitet deshalb an drei Schwerpunkten (vgl. Abbildung 1).

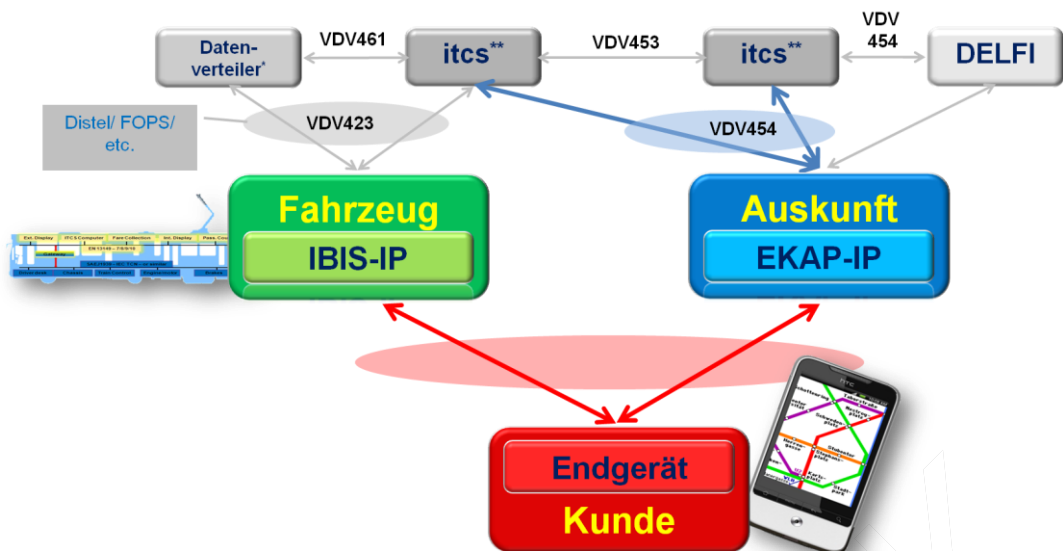


Abbildung 1: Umfeld und Schwerpunkte im Projekt IP-KOM-ÖV

Erster Schwerpunkt (grün in Abbildung 1) ist die Spezifikation eines performanten IP-basierten Kommunikationsprotokolls im Fahrzeug (IBIS-IP). Dabei geht es zum einen darum, den gewachsenen Bedürfnissen der Fahrgastinformation gerecht zu werden und zum anderen um die Definition einer IP-basierten Schnittstelle zur Übertragung der Informationen vom Fahrzeug zum mobilen Kundenendgerät. Hierzu wird der in den achtziger Jahren entwickelte IBIS-Wagenbus aus der VDV-Schrift 300 auf eine moderne Ethernet-Informationsarchitektur umgesetzt.

Zweiter Schwerpunkt (rot in Abbildung 1) ist die individuelle Fahrgastinformation unter Verwendung mobiler Geräte des Fahrgasts (Smartphones, Tablet-PC u. ä.) Hierzu wurden im ersten Schritt die Bedürfnisse von Fahrgästen zu individuellen Informationen ermittelt. Im zweiten Schritt werden einheitliche Schnittstellen zwischen der Echtzeit-Kommunikations- und Auskunftsplattform (EKAP) und den mobilen Kundenendgeräten bzw. zwischen der EKAP und den Hintergrundsystemen entwickelt. Hierbei werden ausschließlich die Datenmodellierungen und Architekturen erforscht und spezifiziert. Aufbauend auf diesen Datenmodellierungen werden semantische Modelle erarbeitet, die helfen, die Fahrgastinformationsdaten für Kommunikationsdienste auf Basis von innovativen Technologien des Semantic Web zur Verfügung zu stellen. Die Entwicklung einer Applikation für mobile Endgeräte ist ausdrücklich nicht vorgesehen.

Dritter Schwerpunkt (blau in Abbildung 1) ist die Definition und Schaffung einer Echtzeit-Kommunikations- und Auskunftsplattform (EKAP). Die EKAP bündelt Informationen von itcs- und anderen Auskunfts- und Informationssystemen und stellt die Vielzahl an Informationen über geeignete Schnittstellen den Applikationen auf den Kundenendgeräten zur Verfügung. Diese Plattform ermöglicht es, Kunden dynamisch mit individuellen Störungsmeldungen versorgen zu können.

Neben den Forschungsarbeiten ist die Standardisierung der Ergebnisse ein wesentliches Ziel des Projektes, um eine nachhaltige Nutzung zu gewährleisten.

Darüber hinaus wird die Praxistauglichkeit dieses neuen Standards in Labor- und Feldtests verifiziert.

1. Einleitung

In diesem Dokument werden die Dienste, die in VDV-Schrift 430 und VDV-Schrift 431-1 beschrieben sind, als XML-Schnittstellen definiert. Dadurch entstehen Schnittstellenstandards, die es Software-Entwicklern und Unternehmen erlauben, Anwendungen zu realisieren, wo mobile Apps der Fahrgäste, Fahrzeuge, Portalsysteme und echtzeitfähige Auskunftssysteme (EKAPs) miteinander kommunizieren.

Bei der Ausarbeitung dieser Schnittstellendefinitionen wurde Wert darauf gelegt, Kompatibilität zu anderen Standards auf dem Sektor des öffentlichen Verkehrs herzustellen. Hier sind vor allem TransModel als Begriffsglossar, IFOPT für die Modellierung von Haltestellen, SIRI für den Austausch von Echtzeitdaten und für sein ausgefeiltes Nachrichtenaustauschverfahren, sowie JourneyWeb und DELFI als Schnittstellen zum Abrufen von Fahrplaninformationen und Verbindungsauskünften zu nennen.

GELBEDRUCK

2. Anwendungsbereich

Die in diesem Dokument definierten Schnittstellen spezifizieren Dienste, die zum einen zwischen mobilen Apps und Fahrzeugen des ÖV Verwendung finden, zum anderen zwischen Portalsystemen und Auskunftssystemen (EKAPs) als Hintergrundsystem. Sie sind für den Zweck der Interaktion mit dem Fahrgast gedacht. In erster Linie soll der Fahrgast informiert werden. Es gibt aber auch Dienste, bei denen der Fahrgast von sich aus aktiv wird, so z. B. beim Haltewunsch oder der Anschlussvoranmeldung.

3. Notation der XML-Elemente und -Strukturen

Die in diesem Dokument vorgestellten TRIAS¹-Schnittstellen werden mit Hilfe von XML-Schema definiert. Die Objekte, die über die Schnittstelle ausgetauscht werden, liegen folglich als XML-Elemente vor. Die Beschreibung der XML-Elemente wird in diesem Dokument in einer Tabellenform vorgenommen, die aus SIRI (CEN, TS 15531 Part 1) stammt. Sie ist sehr kompakt und übersichtlich und bietet eine Vielzahl an strukturellen Informationen, die ansonsten nur in der XML-Schema-Definition sichtbar wird. Dieses Kapitel erläutert die Notation der Tabellenform, die ab Kapitel 7 intensiv genutzt wird.

Alle Namen von Elementen, Datentypen und Attributen sind in Englisch gehalten, um eine etwaige Normierung auf europäischer Ebene vorzubereiten und den Austausch mit europäischen Partnern zu erleichtern.

3.1. Darstellung von XML-Elementen im Text

In diesem Dokument soll eine konsistente Notation der XML-Elemente helfen, technisch wichtige Information beim Lesen bereit zu stellen.

- XML-Elemente werden in Groß-Klein-Schreibweise (Upper Camel Case) fett und kursiv geschrieben, z. B.: ***VehicleJourneyRef***. Die Elementnamen sind – wo immer möglich und sinnvoll – an Begriffe aus TransModel angelehnt. Fehlt in TransModel ein geeigneter Begriff für ein Konzept oder Objekt, so wurde versucht, den entsprechenden Begriff aus JourneyWeb oder das passende Konzept aus DELFI zu übernehmen.
- Datentypen werden kursiv dargestellt, z. B. : *xsd:boolean*.
- Code-Beispiele werden in kleinerer Schrift wiedergegeben.

3.2. Darstellung von Beziehungen

Beziehungen zwischen Objekten können mittels

- impliziter Mechanismen,
- internen Referenzen oder
- externen Referenzen

ausgedrückt werden. Ein impliziter Mechanismus ist z. B. das Enthaltensein eines Elements in einem anderen. Damit wird eine unmittelbare Kindbeziehung ausgedrückt. Ei-

¹ Travellers Realtime Information and Advisory Standard

ne interne Referenz ist ein Objektschlüssel, der innerhalb der Schnittstelle definiert wird (z. B. ein Identifikator einer Meldung). Eine externe Referenz ist ein Objektschlüssel, der außerhalb der Schnittstelle festgelegt wird (z. B. eine Haltestellennummer). Externe Referenzen bestehen manchmal auch aus zusammengesetzten Schlüsseln (siehe die ausführliche Darstellung in Kapitel 5).

Es ist wichtig, den Unterschied zwischen einem Identifikator (Objektschlüssel) und einer Referenz auf das Objekt festzuhalten. In TRIAS gelten folgende Regeln:

- Ein Identifikator ist ein Kindelement des definierenden Elements, das einen eindeutigen Code (Primärschlüssel) für das definierende Element angibt. Diese Identifikatoren enden auf ein signalisierendes Hauptwort wie „Code“ oder „Identifizier“ (manchmal auch „Number“ in SIRI), z. B. erhält eine Fahrplanfahrt (Journey) den Schlüssel **JourneyCode**.
- Wird ein Objekt von einem anderen Objekt referenziert, endet das referenzierende Element (Fremdschlüssel) auf „Ref“. Zum Beispiel lautet die Referenz auf eine Fahrplanfahrt (etwas aus einer Abfahrtstafel heraus): JourneyRef.
- Die Instanz eines Objekts und die Referenz darauf verwenden einen gemeinsamen zugrunde liegenden Datentyp. Zum Beispiel sind JourneyCode und JourneyRef beide vom Typ JourneyCodeType.

3.3. Tabellennotation von XML-Strukturen

In diesem Dokument werden XML-Strukturen in einer Tabellennotation dargestellt (vgl. Tabelle 1). Für jedes wichtige TRIAS-Anfrage/Antwort-Element findet sich eine eigene Tabelle. Weitere Tabellen werden für alle wesentlichen Kindelemente, aus denen die komplexen Strukturen aufgebaut sind, angegeben. Um Platz zu sparen, werden die Spaltenüberschriften nur im Beispiel in Tabelle 1 angezeigt und bei allen folgenden Tabellen nicht wiederholt. In den Tabellen wird ein konsistenter Satz an Regeln zur Beschreibung der XML-Elemente und der daran geknüpften Bedingungen verwendet.

Gruppierung	Elementname	Min : Max	Datentyp	Erläuterung
	ContinuousServiceStructure		+Structure	Eine Fahrgastbewegung mit Hilfe eines kontinuierlichen, nicht fahrplangebundenen Verkehrsmittels.
	a ContinuousMode	-1:1	walk demandResponsive replacement-Service	Modalität für kontinuierliche Verkehre
	b IndividualMode		walk cycle taxi self-drive-car others-drive-car motorcycle truck	Verkehrsmittelmodalität für Individualverkehr
Dated-Service	OperatingDay	1:1	→OperatingDay	Betriebstag der Fahrt. Vgl. 7.4.1.
	VehicleRef	0:1	→Vehicle	Fahrzeug-ID. Vgl. 7.4.1.
Service Journey	JourneyRef	1:1	→Journey	Fahrt-ID. Vgl. 7.4.1.
Linien-Entity	LineRef	1:1	→Line	Linien-ID. Vgl. 7.4.1.
	DirectionRef	1:1	→Direction	Richtungs-ID. Vgl. 7.4.1.
Service	Mode	1:1	+Mode	Verkehrsmitteltyp. Vgl. 7.3.4.

	<i>PublishedLineName</i>	1:1	<i>International-Text</i>	Liniennummer oder -name, wie in der Öffentlichkeit bekannt.
	<i>OperatorRef</i>	0:1	→ <i>Operator</i>	Operator-ID. Vgl. 7.4.1.
	<i>RouteDescription</i>	0:1	<i>International-Text</i>	Beschreibung des Fahrwegs.
	<i>Via</i>	0:*	+ <i>ServiceViaPoint</i>	Wichtige Halte auf dem Fahrweg. Vgl. 7.6.1.
	<i>Attribute</i>	0:*	+ <i>GeneralAttribute</i>	Hinweise und Attribute (mit Klassifikationen) zur Fahrt. Vgl. 7.4.9.
<i>Service Origin</i>	<i>OriginStopPointRef</i>	0:1	→ <i>StopPoint</i>	ID des ersten Haltepunkts der Fahrt; Starthaltestelle. Vgl. 7.5.1.
	<i>OriginText</i>	0:1	<i>International-Text</i>	Name des ersten Haltepunkts der Fahrt, der Starthaltestelle.
<i>ServiceDestination</i>	<i>DestinationStopPointRef</i>	0:1	→ <i>StopPoint</i>	ID des letzten Haltepunkts der Fahrt; Endhaltestelle. Vgl. 7.5.1.
	<i>DestinationText</i>	0:1	<i>International-Text</i>	Name des letzten Haltepunkts der Fahrt, der Endhaltestelle.
	<i>SituationFullRef</i>	0:*	+ <i>SituationFullRef</i>	Verweis auf eine Störungsnachricht. Diese Nachricht kann im Kontext der Meldung (ResponseContext) zu finden sein oder auf anderem Wege bekannt gemacht werden.

Tabelle 1: Beispiel (aus Abschnitt 7.6) für die tabellarische Notation einer XML-Struktur

3.3.1. Gruppierung

In der ersten Spalte befindet sich gelegentlich ein Bezeichner, der die Elemente in sinnvolle Gruppierungen einteilt, z. B. *Service* oder *ServiceOrigin*. Dies dient rein zu Dokumentationszwecken und entspricht in den meisten Fällen den Namen einer XML-Gruppe, die im XML-Schema verwendet wurde. Die Verwendung von Gruppierungen hat nur den Zweck, die Elemente zu organisieren und damit für mehr Klarheit und bessere Wiederverwendbarkeit zu sorgen.

3.3.2. Elementname

Elementnamen werden kursiv in der zweiten Spalte wiedergegeben, z. B. *OperatingDay*. Handelt es sich um ein verpflichtendes Element, so wird es **fett** gedruckt. Optionale Elemente werden nicht fett gedruckt. Der Name der Struktur selbst ist links oben in der Tabelle angegeben.

Elemente, die geerbt (XML: "derived by extension") oder anonym verwendet werden, tragen im Namensfeld drei Doppelpunkte ":::" zur Kennzeichnung.

3.3.3. Multiplizität & Choice (Min:Max)

Die Bedingungen, ob ein Element verpflichtend oder optional ist oder ob es einfach oder mehrfach innerhalb des übergeordneten Elements auftreten kann, werden in der dritten Spalte Min:Max angegeben. Dabei werden die üblichen UML-Konventionen „min:max“ angewendet, so steht z. B. „0:1“ für ein optionales, einfaches Element, „**1:1**“ zeigt ein verpflichtendes, einfaches Element an, „0:*“ steht für ein optionales, mehrfaches Element usw. Verpflichtende Elemente werden **fett** gedruckt.

In manchen Fällen muss ein Element aus seiner Menge ausgewählt werden (XML-Choice). Dies wird durch ein vorangestelltes Minuszeichen symbolisiert, z. B. „-**1:1**“. In diesem Fall steht vor dem Elementnamen noch ein Kleinbuchstabe, der die Auflistung

der Wahlmöglichkeiten anzeigt. Bei optionalen Auswahlmöglichkeiten (Choices) steht im Min-Wert eine Null: „-0:1“.

3.3.4. Datentyp

Die Datentypen werden in der vierten Spalte kursiv angegeben, z. B. *InternationalText*. Falls der Namensraum (namespace) vom TRIAS-Namensraum abweicht, wird er mit angegeben, z. B. „*xs:dateTime*“ oder „*siri:PtSituationElement*“.

- Ein komplexer Datentyp, der selbst Strukturen als Kindelemente enthält, wird in der Spalte Datentyp mit „+*Structure*“ gekennzeichnet.
- Wo Elemente als Referenzen (Fremdschlüssel) auf andere Objekte verwendet werden, wird als Datentyp der Typ des referenzierten Objekts mit vorangestelltem Pfeil verwendet. Zum Beispiel „→*StopPoint*“ als Typ einer Referenz (*StopPointRefStructure*) auf ein Objekt vom Typ „*StopPointType*“.
- Aufzählungstypen (Enumerated types) werden an den meisten Stellen unmittelbar mit den verwendbaren Werten dargestellt, z. B. „*walk | cycle*“. Nur in einigen Fällen mit sehr umfangreichen Aufzählungen, die an mehreren Stellen wieder verwendet werden, wird ein Typ deklariert und referenziert.
- Um Platz zu sparen werden bei der Angabe der Datentypen Abkürzungen verwendet, z. B. wird auf die Endungen „Structure“ und „Type“ durchgehend verzichtet. Statt beispielsweise „*InternationalTextStructure*“ wird also immer „*InternationalText*“ als Datentyp angegeben.

3.3.5. Erläuterung

Alle Elemente erhalten in der letzten Spalte eine Erläuterung ihres Verwendungszwecks. An vielen Stellen wird auf weitere Passagen im Text hingewiesen, so z. B. bei komplexen Kindelementen an die Stelle, wo ihre Tabellenbeschreibung zu finden ist. An einigen Stellen ist die Erläuterung zu umfangreich und würde die Tabellenform sprengen. Dann finden sich diese Anmerkungen im Text unterhalb der Tabelle.

4. Nachrichtenübermittlung

In diesem Kapitel wird erläutert, wie TRIAS-Nachrichten ausgetauscht werden. Es kommen zwei grundlegende Verfahren zum Einsatz

- Anfrage mit synchroner Antwort (Request-Response-Verfahren),
- Abonnements mit asynchronen Nachrichten (Publish-Subscribe-Verfahren).

Diese Verfahren sind bereits etabliert und im Einsatz, z. B. in den SIRI-Schnittstellen.

4.1. Einsatz der SIRI-Verfahren

In SIRI wurden die eingangs aufgezählten Nachrichtenübermittlungsverfahren bereits definiert und beschrieben, vgl. (CEN, TS 15531, Part 2). Daher werden diese Verfahren hier aufgegriffen. Das hat zum einen den Vorteil, dass bereits getestete Verfahren verwendet werden können, zum anderen kann bei der Implementierung der TRIAS-Dienste evtl. auf eine bereits vorhandene SIRI-Implementierung zurückgegriffen werden, was Kosten und Zeit sparen kann.

Das grundlegende Verfahren ist die Anfrage mit synchroner Antwort. Ein Client stellt eine Anfrage an einen Server, der unmittelbar antwortet. In der SIRI-Terminologie ist der

Anfrager der *Data Consumer*, der antwortende Server wird mit *Data Producer* bezeichnet (vgl. Abbildung 2).



Abbildung 2: Anfrage mit synchroner Antwort (Abbildung entnommen aus SIRI, (CEN, TS 15531, Part 2)).

Anfragen mit synchroner Antwort werden bei fast allen TRIAS-Diensten verwendet (eine Ausnahme ist nur der Benachrichtigungsdienst). Die Rolle des Anfragers übernimmt z. B. das Portalsystem, das Anfragen an die EKAP richtet. Aber auch die mobile App stellt Anfragen an das Fahrzeug oder EKAP-Komponenten stellen untereinander Anfragen.

Etwas komplizierter ist der Abonnement-Mechanismus. Ein Datenkonsument interessiert sich für neue Nachrichten, weiß aber nicht, wann diese auftreten werden. Statt regelmäßig nachzufragen und so eine Grundlast zu erzeugen (und zu riskieren, dass er von der neuen Nachricht erst erfährt, wenn er das nächste Mal nachfragt), kann er ein Abonnement einrichten.

Abbildung 3 zeigt die grundlegenden Zusammenhänge. Der Datenkonsument hat zwei Rollen zu erfüllen, die des Abonnenten (Subscriber) und die des Empfängers von Nachrichten (Notification Consumer). Der Datenkonsument bittet den Server um die Einrichtung eines Abonnements (Subscription Request). Dabei teilt er dem Server mit, bei welcher Art Ereignisse er informiert werden möchte. Der Server richtet das Abonnement ein, indem er es beim Abo-Verwalter (Subscription Manager) registriert. Danach geschieht erst etwas, wenn ein Ereignis eintritt, das dem Konsumenten zu melden ist. In diesem Fall schickt der Server als Benachrichtigungsersteller (Notification Producer) dem Datenkonsumenten die Nachricht mit dem neuen Ereignis (Delivery). Dies wiederholt sich so lange, bis das Abonnement ausläuft oder vom Datenkonsumenten beendet wird.

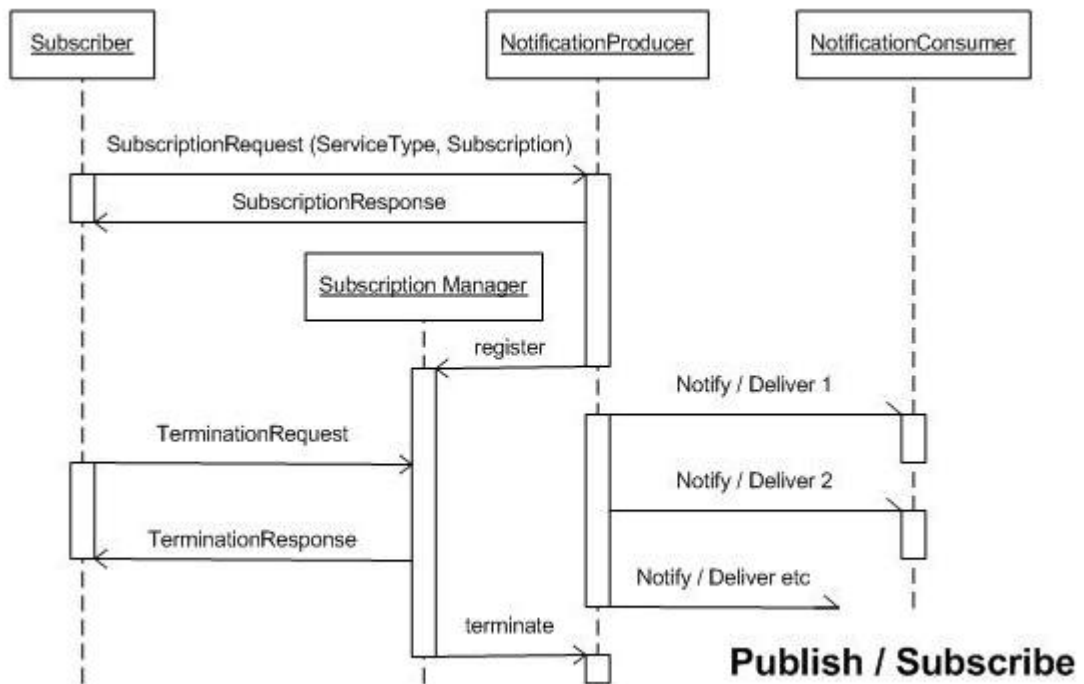


Abbildung 3: Abonnement-Verfahren mit asynchronen Benachrichtigungen (Abbildung entnommen aus SIRI, (CEN, TS 15531, Part 2)).

Sowohl Client als auch Server haben zwei Rollen zu erfüllen, nämlich der Client die Rolle des Subscribers und des Notification Consumers, der Server die Rolle des Notification Producers und des Subscription Managers. In den meisten Implementierungen wird dies aber nicht unterschieden und jeweils eine einzige Softwarekomponente erfüllt beide Rollen.

Das Abo-Verfahren wird komplettiert durch weitere Anfragen. Die Statusanfrage erlaubt es, den Status des Schnittstellenpartners abzufragen und dessen Verfügbarkeit zu testen. Die Heartbeat-Anfrage, die ein Server regelmäßig sendet, ermöglicht es dem Datenkonsumenten umgekehrt zu erkennen, wann ein Server verfügbar ist und Signale (Ping oder Heartbeat) aussendet. Details dazu finden sich in SIRI (CEN, TS 15531, Part 2), Kapitel 5.

In den TRIAS-Diensten kommt der Abo-Mechanismus beim Benachrichtigungsdienst vor, wo ein Datenkonsument (Portalsystem oder mobile App) sich über Störungen oder andere Ereignisse und Vorkommnisse informieren lassen will.

4.2. HTTP und REST

Die Umsetzung der SIRI-Nachrichtenverfahren geschieht in TRIAS mit Hilfe von HTTP/1.1 (Hypertext Transfer Protocol²) als Transportprotokoll und XML (Extensible Markup Language³) für die Nachrichteninhalte.

Eine HTTP-Anfrage wird vom Server unmittelbar unter Nutzung des schon geöffneten IP-Ports beantwortet. Zum Beispiel sendet ein Client eine Anfrage nach einer Verbin-

² <http://tools.ietf.org/html/rfc2616>

³ <http://www.w3.org/XML/>

dungsauskunft als HTTP-Anfrage mit dem XML-Element *Trias* und *TripRequest* als einem der Kindelemente im POST-Block. Der Server antwortet synchron in der HTTP-Antwort mit dem XML-Element *Trias* und *TripResponse* als einem der Kindelemente.

Falls mehrere Anfragen in schneller Folge abgesendet werden, kann der HTTP-Mechanismus „Keep-Alive“ zum Einsatz kommen, bei dem der bereits geöffnete Port eine Zeit lang leben bleibt und wiederbenutzt werden kann, um häufiges Öffnen und Schließen des Ports zu sparen.

Für größere Nachrichten empfiehlt sich der Einsatz eines Komprimierungsverfahrens. Solche Methoden sind ebenfalls für HTTP spezifiziert.

4.3. Rollen von Server und Client

Bei der Nutzung des synchronen Anfrage-Antwort-Verfahrens ist der Datenkonsument (der Anfrager) ein HTTP-Client, der Datenproduzent (der antwortende Server) ein HTTP-Server.

Lediglich beim Benachrichtigungsdienst, wenn das Abonnement-Verfahren zum Einsatz kommt, ist die Lage komplizierter. Hier müssen sowohl Datenkonsument als auch Datenproduzent die Rollen von Client und Server im HTTP-Sinne beide ausfüllen. Denn in dem Moment, wo der Datenproduzent (Notification Producer) eine neue Nachricht an den Datenkonsumenten senden will, wird er zum Client im HTTP-Sinn und der Datenkonsument zum Server im HTTP-Sinn.

5. Identifikation von Objekten über Systemgrenzen hinweg

Damit verschiedene Systeme dasselbe Objekt referenzieren können, ist eine Objekt-ID notwendig, die allen Systemen bekannt ist. Im Rahmen der TRIAS-Schnittstellen sind Haltestellen, Linien und Verkehrsunternehmen Beispiele für solche Objekttypen, zu denen Informationen über die Schnittstellendienste ausgetauscht werden. Daher braucht man für sie (und weitere Objekttypen) Referenzierungssysteme, die allgemein bekannt sind und verwendet werden können.

Das bedeutet nicht notwendigerweise, dass ein Softwaresystem diese Objektschlüssel auch selbst im Betrieb verwenden muss. Es genügt, wenn es die allgemeinen Objektreferenzen verstehen und auf die intern verwendeten Identifikatoren abbilden kann.

Für die in diesem Kapitel vorgestellten Schemata zur Objektreferenzierung wird eine an IFOPT angelehnte Syntax verwendet. Sie benutzt den Doppelpunkt zur Abgrenzung von Namensräumen. Aus diesem Grund ist ein Doppelpunkt ein syntaktisches Trennzeichen und darf in Identifikatoren nicht verwendet werden.

In den folgenden Abschnitten wird für verschiedene Objekttypen vorgestellt, welche Referenzierungssysteme verwendet werden sollen.

5.1. Haltestellen und Haltepunkte

Für die Referenzierung von Haltestellen und Haltepunkten gibt es von CEN die europäische Norm IFOPT (CEN, EN 28701:2012, 2012). Dort ist in Kapitel 6.8.1 eine Syntax für den Aufbau eines Referenzierungsschlüssels vorgesehen. Einige Systeme in Deutschland unterstützen diese Syntax bereits. Eine bundesweite Einführung wird im BMVBS-

Projekt DELFIplus vorbereitet. In den TRIAS-Schnittstellen sollen die Ergebnisse aus diesem Projekt zur Anwendung kommen.

Aufbau eines IFOPT-Objektschlüssels

Länderkürzel:Region:Haltestellennummer:Bereich:Haltepunkt

Das folgende Beispiel zeigt den (hierarchischen) Aufbau der Schlüssel für eine Haltestelle, einen Haltestellenbereich und einen Haltepunkt

Praxis-Beispiel:

Haltestelle Karlsplatz (Stachus) in München:

Haltestellenobjekt	Eindeutige ID
Haltestelle Karlsplatz (Stachus) in München	de:9162:1
Haltestellenbereich U-Bahn U4/5	de:9162:1:2
Haltepunkt U4/5 Richtung Odeonsplatz	de:9162:1:2:URiOd

Client-Systeme, die selbst keine eigene Datenversorgung haben, können die Objektreferenzen für Haltestellen und Haltepunkte mit Hilfe des TRIAS-Schnittstellendienstes Ortsauflösung (vgl. 8.1) von der EKAP beziehen.

5.2. Orte und Gemeinden

Zur eindeutigen Referenzierung von Gemeinden existiert in Deutschland der Amtliche Gemeindegchlüssel (AGS⁴), früher auch Gemeindekennziffer (GKZ) genannt. Für die Orte innerhalb einer Gemeinde ist die Situation je nach Bundesland unterschiedlich. In Bayern z. B. gibt es je Gemeinde eine Liste von Orten mit amtlich festgelegten Orts-IDs. Wo diese Identifikatoren fehlen, AW2013Vdv2

müssen eigene Festlegungen getroffen werden, damit systemübergreifend ein gleiches Verständnis von Orten vorliegt, so dies notwendig ist. Dabei kann z. B. die Ortsliste aus dem Bestand der DELFI-Meta-Daten verwendet werden.

Für den Betrieb von TRIAS-Schnittstellen empfiehlt sich die Verwendung eines Ortsschlüssels, der sich vom Aufbau her an die IFOPT-Norm für Haltestellen anlehnt:

Aufbau eines Ortsschlüssels

⁴ Siehe auch: Statistisches Bundesamt, <https://www.destatis.de/DE/ZahlenFakten/LaenderRegionen/Regionales/Gemeindeverzeichnis/Gemeindeverzeichnis.html>

Länderkürzel:AGS:Ort

Beispiel	
Ilmenau	de:16070029:1

Client-Systeme, die selbst keine eigene Datenversorgung haben, können die Objektreferenzen für Gemeinden und Orte mit Hilfe des TRIAS-Schnittstellendienstes Ortsauflösung von der EKAP beziehen.

5.3. Adressen und POIs

Für den Betrieb von TRIAS-Schnittstellen ist es nicht notwendig, dass Adressen und wichtige Punkte (Points Of Interest, POI) systemübergreifend referenziert werden können. Es genügt, deren Lage durch Koordinatenpositionen mitzuteilen.

5.4. Verkehrsunternehmen und Verkehrsverbünde

Zur eindeutigen Referenzierung von Verkehrsunternehmen und Verkehrsverbänden wird ein Organisationscode verwendet. Damit diese Codes über mehrere Systeme hinweg eindeutig bleiben, empfiehlt sich der Aufbau einer übergreifenden (im besten Falle nationalen) Datenbank von Verkehrsunternehmen.

Aufbau eines Organisationscodes
Länderkürzel:Organisationscode

Beispiele	
Verkehrs- und Tarifverbund Stuttgart	de:vvs
Stuttgarter Straßenbahn AG	de:ssb
Fernverkehr Deutsche Bahn	de:dbag
DB Regio Baden-Württemberg	de:dbregiobw

5.5. Linien und Linienrichtungen

Zur eindeutigen Referenzierung von Linien wird der Linienschlüssel des verantwortlichen Datenlieferanten verwendet. Als verantwortlicher Datenlieferant kommt das beauftragte Verkehrsunternehmen (Konzessionär) oder der zuständige Verkehrsverbund in

Frage.⁵ Damit diese Codes über mehrere Datenlieferanten hinweg eindeutig bleiben, wird der Organisationscode (vgl. 5.4) als Namensraum vorangestellt.

Aufbau eines Linienschlüssels

Länderkürzel:Organisationscode:Linienschlüssel

Beispiel

Stadtbahn-Linie U1 in Stuttgart de:vvs:20001

Zur eindeutigen Referenzierung von Linienrichtungen wird der Richtungscode des verantwortlichen Datenlieferanten verwendet. Der Richtungscode ist vom Datenlieferanten frei wählbar und wird für den Fahrgast erst durch begleitende Texte verständlich. Der Richtungscode wird nur im Kontext einer Linie verwendet, so dass das Voranstellen des Linienschlüssels als Namensraum nicht notwendig ist.

Aufbau eines Richtungscode

Richtungscode

Beispiele

Hin	H
Rück	R
Hin	1
Rück	2
Stadteinwärts	E
Stadtauswärts	A

5.6. Fahrten

Zur eindeutigen Referenzierung von Fahrten (engl. *Vehicle journey* oder kurz: *Journey*) wird der Fahrtenschlüssel des verantwortlichen Datenlieferanten verwendet. Der Fahrtenschlüssel ist vom Datenlieferanten frei wählbar, solange er im Namensraum einer Linie eindeutig ist.

⁵ Damit in Datensammelsystemen Dopplungen von Datenlieferungen zu einer Linie von mehreren Datenlieferanten vermieden werden können, empfiehlt sich der Aufbau einer übergreifenden (im besten Falle nationalen) Datenbank von Linien.

Aufbau eines Fahrtschlüssels

Länderkürzel:Organisationscode:Linien-schlüssel:Fahrtenschlüssel

Beispiel

Fahrt 1512 der Linie U1 in Stuttgart de:vvs:20001:1512

Falls eine Organisation (Verkehrsunternehmen) ihre Fahrten nicht in Linien organisiert (z. B. Bahnfernverkehr), kann der Linienschlüssel leer bleiben.

Beispiel

ICE 612 der DB AG de:dbag::612

5.7. Fahrzeuge

Zur eindeutigen Referenzierung von Fahrzeugen (engl. *Vehicle*) wird der Fahrzeugcode des verantwortlichen Datenlieferanten verwendet.

In Echtzeitschnittstellen (VDV 454, SIRI ET) teilen Leitstellen zu einer Fahrplanfahrt den Fahrzeugcode mit, so dass eine EKAP für einen bestimmten Betriebstag wissen kann, welche Fahrt von welchem Fahrzeug durchgeführt wird. Für jeden Betriebstag muss daher die Zuordnung von Fahrzeugcode zu Fahrtenschlüssel eineindeutig sein.

Aufbau eines Fahrzeugcodes

Länderkürzel:Organisationscode:Fahrzeugcode

Beispiel

Fahrzeug 5812 der SSB AG de:ssb:5812

5.8. Fahrzeugtypen

Der Fahrzeugtyp (engl. *Vehicle type*) wird nur vom Fahrzeug an die mobile App mitgeteilt (in Form von Code und menschenlesbarem Text). Der Fahrzeugtypcode wird nicht in Folgeaufrufen verwendet und wird daher im Rahmen von TRIAS nicht weiter betrachtet.

5.9. Verkehrstage

Eine Fahrplanfahrt wird erst in Verbindung mit einem Verkehrstag (engl. *Operating Day*) zu einer spezifischen Fahrt. Ein Verkehrstag kann auch Uhrzeiten nach Mitternacht einschließen und daher von einem Kalendertag abweichen. Ob eine solche Abweichung existiert und wie groß sie ist, ist für die Fahrgastinformation nicht relevant. Den Fahrgästen gegenüber werden nur Uhrzeiten und Datumsangaben nach dem Kalendertagsprinzip bekannt gegeben.

Ein Verkehrstag wird in TRIAS nach der Norm ISO 8601 dargestellt.

Beispiel

29. März 2013

2013-03-29

5.10. Eigentümer

Mit dem Begriff Eigentümer (engl. *Owner*) sind hier die Betreiber von Haltestelleneinrichtungen und Fahrgastinformationsgeräten gemeint. In der Regel sind das Verkehrsunternehmen, aber auch z. B. Kinobetreiber können einen Monitor für die Anzeige von aktuellen Haltestellenabfahrten aufstellen und betreiben. Die Referenzierung von Eigentümern erfolgt auf genau dieselbe Weise wie die von Verkehrsunternehmen und -verbänden (vgl. 5.4).

5.11. Haltestellen- und Fahrzeugeinrichtungen

Haltestellen- und Fahrzeugeinrichtungen (wie z. B. Aufzüge oder Fahrscheinautomaten) werden durch Codes referenziert, die vom Eigentümer (vgl. 5.10) vergeben werden. Im Kontext eines Eigentümers ist der Code einer Einrichtung also global eindeutig.

5.12. Teilnehmende Systeme / IT-Systeme

Die TRIAS-Dienste werden von IT-Systemen angeboten und in Anspruch genommen. Sie sind die teilnehmenden Systeme (engl. *Participants*) an einem umfassenden Systemverbund zur Steuerung des Betriebs des ÖV und zur Fahrgastinformation. Damit diese Systeme unterscheidbar und ansprechbar sind, benötigen sie Kennungen (in VDV 453/454 als *Leitstellenkennung* bekannt).

Aufbau einer Systemkennung

Länderkürzel:Organisationscode:Systemkennung

Beispiel

Öffentliche EKAP des VVS

de:vvs:publicEKAP

5.13. Ereignismeldungen

Ereignis- und Störungsmeldungen (engl. *Situations*) werden mit Hilfe der in SIRI SX definierten Strukturen übertragen. Dort ist auch die Vergabe von IDs für die Ereignismeldungen geregelt. Die Meldungs-IDs werden im Kontext des teilnehmenden Systems (vgl. 5.12) übertragen und sind somit global eindeutig.

5.14. Tarifverantwortliche

Eine Organisation, die verantwortlich ist für die Festlegung von Tarifstrukturen und die Entwicklung von Fahrscheinprodukten, wird als Tarifverantwortlicher (engl. *Fares authority*) bezeichnet. Für Verbundtarife sind dies meist die Verkehrs- und Tarifverbünde, für Haustarife die Verkehrsunternehmen selbst. Die Referenzierung von Tarifverantwortlichen erfolgt auf genau dieselbe Weise wie die von Verkehrsunternehmen und -verbänden (vgl. 5.4).

5.15. Tarifzonen

Die Codierung von Tarifzonen (engl. *Fare zones*) liegt in der Obhut der jeweiligen Tarifverantwortlichen (vgl. 5.14). Tarifzonen werden im Kontext der jeweiligen Tarifverantwortlichen angegeben und werden so global eindeutig.

5.16. Fahrscheine und Vielfahrerkarten

Die Codierung von Fahrscheinen (engl. *Ticket*) liegt in der Obhut der jeweiligen Tarifverantwortlichen (vgl. 5.14). Fahrscheine werden im Kontext der jeweiligen Tarifverantwortlichen angegeben und werden so global eindeutig.

Die Codierung von Vielfahrerkarten (engl. *TravellerCard*), z. B. BahnCard50 der Deutschen Bahn AG, liegt in der Obhut der jeweiligen Tarifverantwortlichen (vgl. 5.14). Der Code einer Vielfahrerkarte muss im Namensraum des Tarifverantwortlichen angegeben werden.

Aufbau eines Codes für eine Vielfahrerkarte

Länderkürzel:Organisationscode:TravellerCardCode

Beispiel

BahnCard50 der DB AG

de:dbag:BC50

6. Dienste und XML-Schemata

In diesem Dokument werden Schnittstellendefinitionen für Dienste zwischen Softwarekomponenten dargestellt. Für eine ausführliche Erläuterung der Aufgabenstellung dieser Dienste und der möglichen Systemarchitekturen sei hier auf die grundlegenden VDV-Schriften (VDV-Schrift 431-1) und (VDV-Schrift 430) verwiesen.

Die TRIAS-Schnittstellendienste sind als XML-Schemata definiert. Eine Übersicht über die Dienste und ihre Implementierung als XML-Schema bietet der erste Abschnitt dieses Kapitels. Einige Strukturdefinitionen sind in mehreren Diensten nützlich und werden daher in eigenen Schemadateien als gemeinsame Basis hierarchisch definiert, so dass eine Wiederverwendbarkeit ermöglicht wird. Das dabei verfolgte Konzept orientiert sich stark an den Grundsätzen der Objektorientierung. Die gemeinsam genutzten Strukturdefinitionen sind im zweiten Abschnitt beschrieben. Der dritte Abschnitt stellt die XML-Schemata vor, die aus SIRI importiert werden. Eine Klassifikation der Fehlerzustände findet sich im vierten Abschnitt.

6.1. Bereitgestellte Dienste

Die TRIAS-Schnittstellenfamilie umfasst derzeit folgende Dienste:

Dienst	Bezeichnung des Anfrageelements	Schema-Datei	Kapitel
Ortsinformation	LocationInformationRequest	Trias_Locations.xsd	8
Verbindungsankunft	TripRequest	Trias_Trips.xsd	9
Abfahrtstafeln	StopEventRequest	Trias_StopEvents.xsd	10
Logische Ortung	PositioningRequest	Trias_Positioning.xsd	11
Fahrtinformation (EKAP)	TripInfoRequest	Trias_TripInfo.xsd	12
Anschlussmeldung	ConnectionDemandRequest	Trias_Connections.xsd	13.1.1
Anschlussstatus	ConnectionStatusRequest	Trias_Connections.xsd	13.1.2
Info bei Anschlussverlust	ConnectionStatusResponse	Trias_Connections.xsd	13.1.3
Anschlussrückmeldung	ConnectionReportRequest	Trias_Connections.xsd	13.1.4
Fahrpreis- und Tarifberechnung	FaresRequest	Trias_Fares.xsd	14
Buchungsinformation	BookingInfoRequest	Trias_Booking.xsd	15
IV-Routing	IndividualRouteRequest	Trias_IndividualTrips.xsd	16

Dienst	Bezeichnung des Anfrageelements	Schema-Datei	Kapitel
Kartendienst	MapServiceRequest ImageCoordinatesRequest GeoCoordinatesRequest	Trias_Maps.xsd	17
Schadensmeldung	FacilityStatusReport	Trias_Facilities.xsd	18.4
Zustand von Einrichtungen	FacilityRequest	Trias_Facilities.xsd	18.6
Benachrichtigungsdienst	SubscriptionRequest	Trias.xsd	7.1.2 und 19
Personalisierungsdienst	PersonalisationRequest	Trias_Personalisation.xsd	20
Fahrzeuginformationen	VehicleDataRequest	Trias_VehicleInterface.xsd	21
Haltewunsch	StopRequestRequest	Trias_VehicleInterface.xsd	22
Diensteregister	ServiceRegisterRequest	Trias_ServiceRegister.xsd	23
Authentifizierung	AbstractTriasServiceRequest (vererbt auf alle TRIAS-Nachrichten)	Trias_RequestSupport.xsd	7.9 und 24

Tabelle 2: Liste der TRIAS-Dienste und ihrer Anfrageelemente.

6.2. Dienstübergreifend genutzte XML-Schemata

Um Strukturen, die in mehr als einem Dienst verwendet werden, nicht mehrfach und damit redundant definieren zu müssen, werden gemeinsam benutzte Basis-XML-Schemata eingeführt, die sich hierarchisch inkludieren. Die Inklusionsreihenfolge und der Zuschnitt der Schemadateien sind dabei so gewählt, dass inhaltlich verwandte Elemente in einer Datei zusammenstehen und dass jedes Schema möglichst nur so viel inkludiert, wie für die eigenen Aufgaben notwendig ist.

Die gemeinsam genutzten Basis-Schemadateien werden ausführlich in Kapitel 7 erläutert. Die hierarchische Abhängigkeit zwischen den einzelnen Schemata zeigen die folgenden Abbildungen. Die blau eingefärbten Schemadateien werden von SIRI importiert.

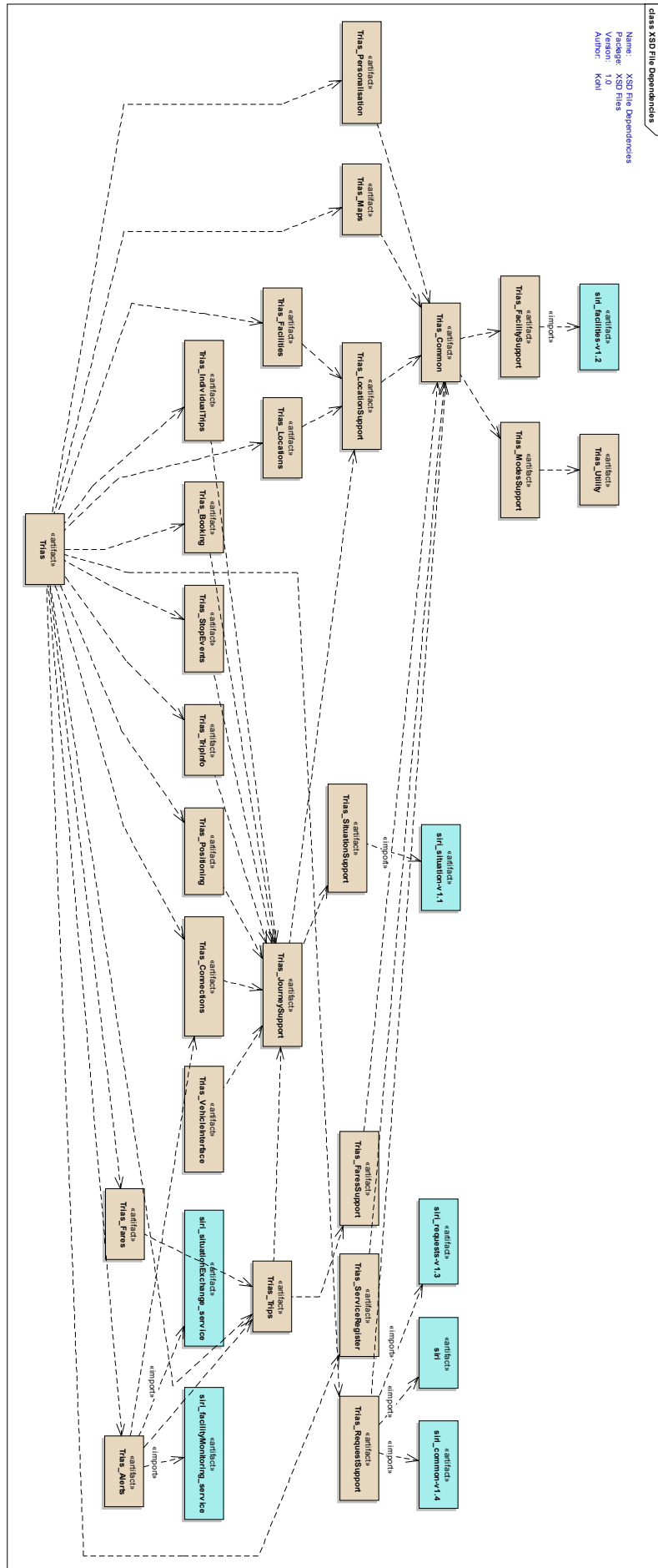


Abbildung 4: Gesamtübersicht über die Hierarchie der Schemadateien.

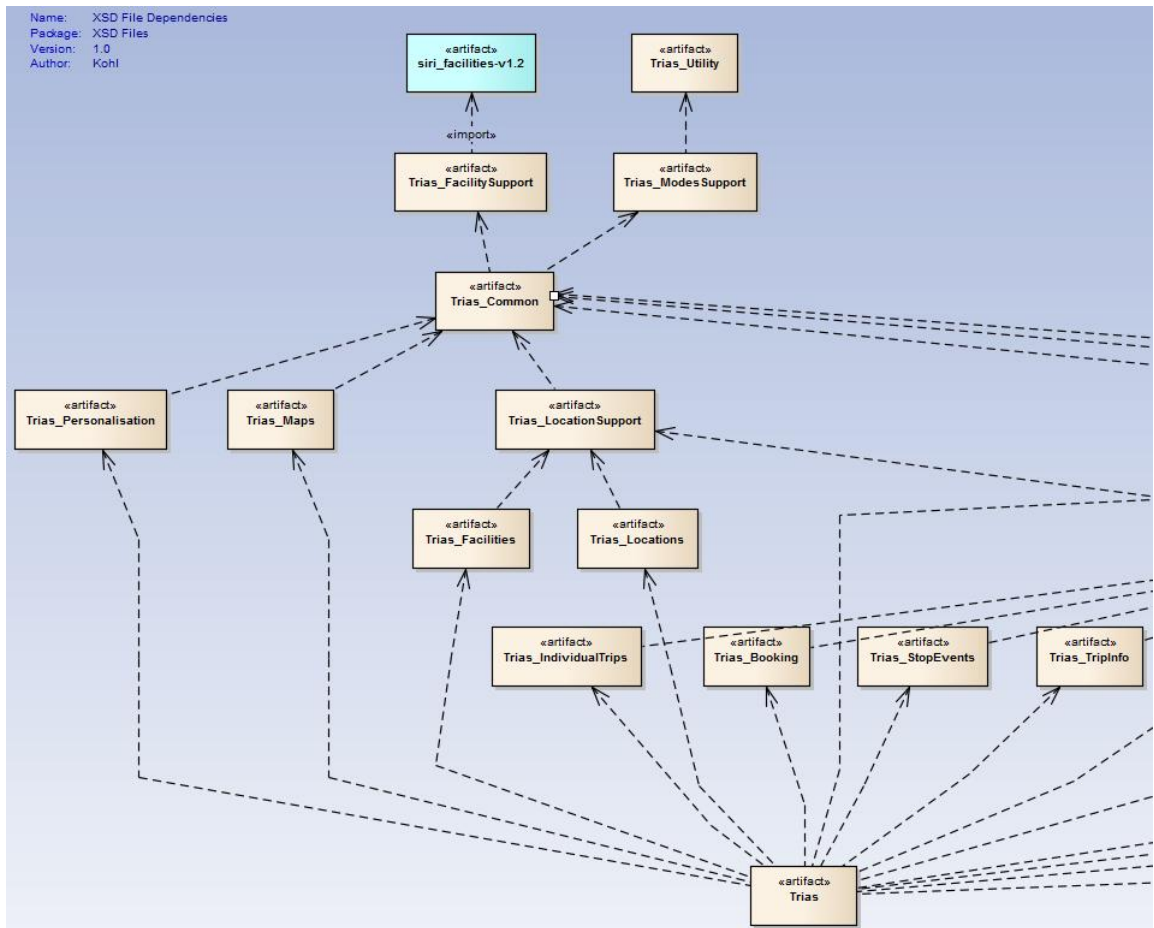


Abbildung 5: Linker Teil der Schema-Hierarchie.

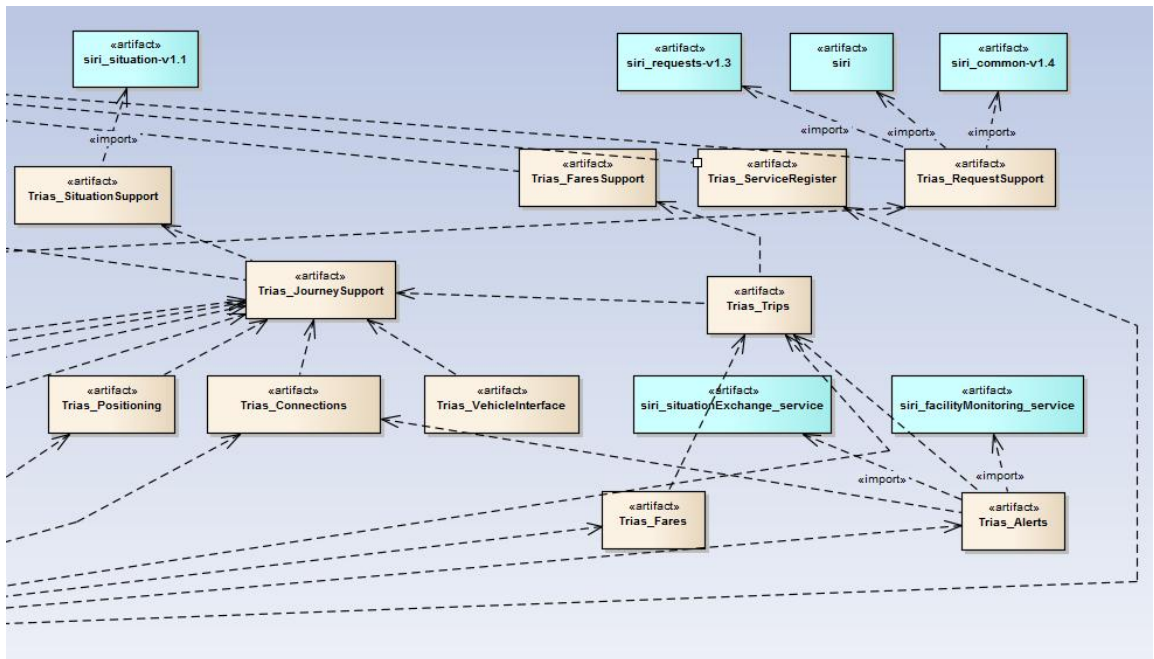


Abbildung 6: Rechter Teil der Schema-Hierarchie.

6.3. Importierte Schemata

Aus den SIRI-Schnittstellen werden einige Schemadateien nach TRIAS importiert. Diese Dateien sind in den Abbildungen des vorigen Abschnitts blau eingefärbt dargestellt. Durch diesen Import von SIRI-Definitionen wird erreicht, dass die SIRI-Verfahren für den Austausch von Nachrichten auch für die TRIAS-Meldungen anwendbar sind. Außerdem können bestimmte Strukturdefinitionen aus SIRI wiederverwendet werden, was die Konsistenz zwischen diesen Schnittstellenstandards sicherstellt. Dies betrifft unter anderem die Definition von Verkehrsmittelarten (modes), Störungsereignissen (situations) und Haltestelleneinrichtungen bzw. Fahrzeugausstattungen (facilities).

SIRI wird in der derzeit aktuellen Version 1.4 importiert.

6.4. Fehlerzustände beim Betrieb von TRIAS-Diensten

Die Fehlerzustände beim Betrieb von TRIAS-Diensten werden durch Fehlercodes signalisiert, die in der Struktur *ErrorMessage* übermittelt werden können. *ErrorMessage* kann an den meisten Stellen mehrfach auftreten und daher auch eine mehrfache, vielschichtige Fehlersituation beschreiben. In *ErrorMessage* können Fehlercodes auftreten, die

- aus den SIRI-Diensten geerbt werden,
- allgemeine, dienstübergreifende TRIAS-Fehlersituationen beschreiben oder
- dienstspezifische Fehlersituationen anzeigen.

Die TRIAS-Fehlercodes sind durch ein Präfix gekennzeichnet, das den jeweiligen Dienst angibt (z. B. **STOPEVENT_**) oder anzeigt, dass es sich um einen allgemeinen Fehlerzustand handelt (**TRIASGENERIC_**).

6.4.1. Fehlercodes aus SIRI

In SIRI (CEN, TS 15531, Part 2), Kapitel 5.7, werden eine Reihe von Fehlercodes definiert, die für das Nachrichtenübermittlungsverfahren eine wichtige Rolle spielen. Diese Codes sind in die Gruppen Erfolg (Success), Systemfehler (Systemic Error) und Anwendungsfehler (Application Error) eingeteilt (vgl. Tabelle 3).

Group	Condition	Description (Beschreibung)
Success	OK (true)	Request successful. (Anfrage erfolgreich bearbeitet.)
Systemic Error	RequestTimeout	Server not responding. (Server antwortet nicht.)
	InvalidRequest	The server does not "understand" the request. The client should not repeat the request. (Der Server "versteht" die Anfrage nicht. Der Client braucht die Anfrage nicht zu wiederholen.)
	Unauthorized	User name and password are required for the request, or credentials not satisfied. (Benutzername und Passwort sind für die Anfrage erforderlich, oder die Berechtigungen reichen nicht aus.)
	Forbidden	The server "understands" the request, but cannot carry it out. (Der Server "versteht" die Anfrage, kann sie aber nicht ausführen.)
	NotFound	The requested URL was not found. (Die angefragte URL konnte nicht gefunden werden.)
Application Error	VersionNotSupported	Service is not available. (Die angefragte Version des Dienstes ist nicht verfügbar.)
	CapabilityNotSupported	Service does not support the requested capability. (Die angeforderte Funktionalität wird vom Dienst nicht unterstützt.)

Group	Condition	Description (Beschreibung)
	ServiceNotAvailable	Functional service is not available to use (but it is still capable of giving this response). (Der funktionale Dienst kann keine Anfragen abarbeiten (obwohl er in der Lage ist, eine Antwort zu geben).)
	AccessNotAllowed	Requestor is not authorised to the service or data requested. (Der Anfrager ist für den Zugriff auf den Dienst oder die Daten nicht autorisiert.)
	NoInfoForTopic	Valid request was made but service does not hold any data for the requested topic expression. (Die Anfrage ist gültig, der Dienst kann aber über den angefragten Fachinhalt keine Auskunft geben.)
	UnknownSubscriber	Subscriber not found. (Der Abonnent wurde nicht gefunden.)
	UnknownSubscription	Subscription not found. (Das Abonnement wurde nicht gefunden.)
	AllowedResource-UsageExceeded	Valid request was made, but request would exceed the permitted resource usage of the client. (Die Anfrage ist gültig, sie überschreitet aber das dem Client zugestandene Ressourcen-Limit.)
	OtherError	Other Error Type. (Sonstiger Fehler.)

Tabelle 3: Liste der Fehlercodes, wie sie in SIRI für das Nachrichtenübermittlungsverfahren definiert werden.

6.4.2. Allgemeine TRIAS-Fehlerzustände

In *ErrorMessage* können folgende allgemeine Fehlerzustände auftreten:

TRIAS- GENERIC_LANGUAGENOTSUPPORTED	In der Anfrage wurde eine Sprache für die Anzeige der Ergebnistexte spezifiziert, die vom Server nicht unterstützt werden kann (zumindest im Kontext der vorliegenden Anfrage).
TRIASGE- NERIC_EXCEPTIONFROMREQUESTEDLAN- GUAGE	In der Anfrage wurde eine Sprache für die Anzeige der Ergebnistexte spezifiziert, die vom Server nicht bei allen Textelementen der Antwort unterstützt werden konnte.
TRIAS- GENERIC_DATAVERSIONNOTAVAILABLE	Die in der Anfrage angeforderte Datenversion konnte vom Server nicht berücksichtigt werden.
AUTH_FAILURE	Dieser Fehler tritt auf, wenn eine Anfrage mit ungültiger oder nicht prüfbarer Signatur empfangen wurde.
AUTH_MISSING	Dieser Fehlercode tritt auf, wenn der Server zwingend eine Authentifizierung benötigt, aber eine Nachricht ohne Signatur empfangen wurde.
AUTH_USER_UNKNOWN	Dieser Fehlercode wird zurückgegeben, wenn die Authentifikation fehlschlägt, weil der Benutzer unbekannt ist.

Tabelle 4: Generische TRIAS-Fehlermeldungen, die in allen Nachrichten auftreten können.

7. Gemeinsam genutzte XML-Strukturen

In diesem Kapitel werden die XML-Strukturen erläutert, die als Basisobjekte in den dienstübergreifend genutzten XML-Schemadateien definiert werden. Die Gliederung ergibt sich anhand der einzelnen Schemadateien.

7.1. Trias, das Wurzelement

TRIAS steht für *Travellers' Realtime Information and Advisory Standard*.

In der XML-Schema-Definition Trias.xsd wird das allgemeine Wurzelement *TRIAS* definiert, das als gemeinsame Basis für alle Nachrichten aller TRIAS-Dienste dient.

An komplexen Strukturen finden sich in Trias.xsd folgende:

7.1.1. ServiceRequestStructure

ServiceRequestStructure			+Structure	Grundstruktur für jede TRIAS-Anfrage (ohne Abonnement).
<i>Service Request Context</i>	:::	1:1	<i>AbstractTriasServiceRequest</i>	Gemeinsamer Anfragekontext (vgl. 7.9.2).
	RequestPayload	1:1	<i>RequestPayload</i>	Dienstspezifischer Anfrageinhalt (vgl. 7.1.3).

Tabelle 5: Beschreibung der Struktur **ServiceRequestStructure**.

7.1.2. SubscriptionRequestStructure

SubscriptionRequestStructure			+Structure	Grundstruktur für jede TRIAS-Abonnement-Anfrage.
<i>SubscriptionRequestContext</i>	:::	1:1	<i>AbstractTriasSubscriptionRequest</i>	Gemeinsamer Anfragekontext (vgl. 7.9.3).
<i>AlertSettingsGroup</i>	<i>AlertTimeWindow</i>	0:*	<i>WeekdayTimePeriod</i>	Zeitfenster, in denen Benachrichtigungen zugestellt werden dürfen (vgl. 7.4.8).
	<i>MaximumAlertFrequency</i>	0:1	<i>xs:duration</i>	Maximale Frequenz für Benachrichtigungen aus gleichem Grund.
	<i>MaximumTimeBeforeEvent</i>	0:1	<i>xs:duration</i>	Früheste Zeit für eine Benachrichtigung gemessen vom Beginn des Ereignisses. Nur im Zusammenhang mit Ereignissen, deren Beginn im Voraus bekannt ist.
<i>SubscriptionRequest</i>	a <i>SituationExchangeSubscriptionRequest</i>	-1:1	<i>SituationExchangeSubscriptionRequest</i>	Inhalt der Abonnement-Anfrage für allgemeine Ereignis- und Störungsbearbeitung (vgl. Kapitel 19).
	b <i>FacilityMonitoringSubscriptionRequest</i>		<i>FacilityMonitoringSubscriptionRequest</i>	Inhalt der Abonnement-Anfrage für Zustandsmeldungen zur Infrastruktur von Haltestellen und Fahrzeugen (vgl. Kapitel 19).
	c <i>TripMonitoringSubscriptionRequest</i>		<i>TripMonitoringSubscriptionRequest</i>	Inhalt der Abonnement-Anfrage für Benachrichtigungen zu einer bestimmten Reiseverbindung (vgl. 19.3.1).

Tabelle 6: Beschreibung der Struktur **SubscriptionRequestStructure**.

7.1.3. RequestPayloadStructure

RequestPayloadStructure			+Structure	Element zur Auswahl des gewünschten TRIAS-Dienstes.
a	BookingInfoRequest		<i>BookingInfoRequest</i>	Anfrage nach Buchungsinformationen (vgl. 15.2.1).
b	ConnectionDemandRequest		<i>ConnectionDemandRequest</i>	Anfrage zur Anschlussvoranmeldung (vgl. 13.4.1).
c	ConnectionDemandDeleteRequest		<i>ConnectionDemandDeleteRequest</i>	Löschung einer Anschlussvoranmeldung (vgl. 13.4.2).
d	ConnectionReportRequest		<i>ConnectionReportRequest</i>	Meldung des Fahrgasts, ob ein Anschluss geklappt hat (vgl. 13.9.1).
e	ConnectionStatusRequest		<i>ConnectionStatusRequest</i>	Anfrage zum Anschlussstatus (vgl. 13.6.1).
f	FacilityRequest		<i>FacilityRequest</i>	Anfrage zu Fahrzeug- und Infrastruktureinrichtungen (vgl. 18.6.1).
g	FacilityStatusReport		<i>FacilityStatusReport</i>	Übermittlung des Zustands von Fahrzeug- und Infrastruktureinrichtungen in einem aktiven Abonnement (vgl. 18.4.1).
h	FaresRequest		<i>FaresRequest</i>	Anfrage zum Tariffberechnungsdienst (vgl. 14.2.1).
i	GeoCoordinatesRequest		<i>GeoCoordinatesRequest</i>	Anfrage nach Geo-Koordinaten (vgl. 17.2.3).
j	ImageCoordinatesRequest		<i>ImageCoordinatesRequest</i>	Anfrage nach Bildkoordinaten (vgl. 17.2.2).
k	IndividualRouteRequest		<i>IndividualRouteRequest</i>	Anfrage nach einer IV-Route (vgl. 16.2.1).
l	LocationInformationRequest		<i>LocationInformationRequest</i>	Anfrage zum Ortsinformationsdienst (vgl. 8.3.1).
m	MapServiceRequest		<i>MapServiceRequest</i>	Anfrage zum Kartendienst (vgl. 17.2.1).
n	PersonalisationRequest		<i>PersonalisationRequest</i>	Anfrage zum Personalisierungsdienst (vgl. 20.4.1).
o	PositioningRequest		<i>PositioningRequest</i>	Anfrage zum Positionierungsdienst (vgl. 11.2.1).
p	ServiceRegisterRequest		<i>ServiceRegisterRequest</i>	Anfrage zum Dienstregisterdienst (vgl. 23.3.1).
q	StopEventRequest		<i>StopEventRequest</i>	Anfrage nach Abfahrtstafeln (vgl. 10.2.1).
r	StopRequestRequest		<i>StopRequestRequest</i>	Bekanntgabe eines Haltewunsches im Fahrzeug (vgl. 22.2.1).
s	TripInfoRequest		<i>TripInfoRequest</i>	Anfrage nach Fahrtinformationen (vgl. 12.2.1).
t	TripRequest		<i>TripRequest</i>	Anfrage nach einer intermodalen Verbindungsbe- rechnung (vgl. 9.2.1).
u	VehicleDataRequest		<i>VehicleDataRequest</i>	Anfrage nach Fahrzeuginformationen (vgl. 21.2.1).

Tabelle 7: Beschreibung der Struktur **RequestPayloadStructure**.

7.1.4. ServiceDeliveryStructure

ServiceDeliveryStructure		+Structure	Grundstruktur für jede TRIAS-spezifische Antwort.	
	...	1:1	<i>AbstractTriasResponse</i>	Gemeinsamer Antwortkontext (vgl. 7.9.4).
	<i>DeliveryPayload</i>	1:1	<i>DeliveryPayload</i>	Dienstspezifischer Antwortinhalt (vgl. 7.1.5).

Tabelle 8: Beschreibung der Struktur **ServiceDeliveryStructure**.

7.1.5. DeliveryPayloadStructure

DeliveryPayloadStructure		+Structure	Element zur Auswahl der passenden TRIAS-Antwort.
a	BookingInfoResponse	<i>BookingInfoResponse</i>	Antwort mit Buchungsinformationen (vgl. 15.3.1).
b	ConnectionDemandResponse	<i>ConnectionDemandResponse</i>	Antwort zur Anschlussvoranmeldung (vgl. 13.5.1).
c	ConnectionDemandDeleteResponse	<i>ConnectionDemandDeleteResponse</i>	Bestätigung zur Löschung einer Anschlussvoranmeldung (vgl. 0).
d	ConnectionReportResponse	<i>ConnectionReportResponse</i>	Bestätigung zur Anschlusserrichtungsmeldung (vgl. 13.10.1).
e	ConnectionStatusNotification	<i>ConnectionStatusNotification</i>	Übermittlung des Anschlussstatus in einem aktiven Abonnement (vgl. 13.6.2).
f	ConnectionStatusResponse	<i>ConnectionStatusResponse</i>	Antwort zu Anschlussstatusanfrage (vgl. 13.7.1).
g	FacilityMonitoringDelivery	<i>si-ri:FacilityMonitoringDelivery</i>	Aktualisierungsnachricht im Rahmen eines Abonnements zu Ausstattungsmerkmalen und Einrichtungen (vgl. Kapitel 19).
h	FacilityResponse	<i>FacilityResponse</i>	Antwort zu Fahrzeug- und Infrastruktureinrichtungen (vgl. 18.7.1).
i	FacilityStatusReportResponse	<i>FacilityStatusReportResponse</i>	Bestätigung zur Übermittlung des Zustands von Fahrzeug- und Infrastruktureinrichtungen (Schadensmeldung, vgl. 18.5.1).
j	FaresResponse	<i>FaresResponse</i>	Antwort zur Tarifberechnungsanfrage (vgl. 14.3.1).
k	GeoCoordinatesResponse	<i>GeoCoordinatesResponse</i>	Antwort zur Anfrage nach Geo-Koordinaten (vgl. 17.3.3).
l	ImageCoordinatesResponse	<i>ImageCoordinatesResponse</i>	Antwort zur Anfrage nach Bildkoordinaten (vgl. 17.3.2).
m	IndividualRouteResponse	<i>IndividualRouteResponse</i>	Antwort mit berechneten IV-Routen (vgl. 16.3.1).
n	LocationInformationResponse	<i>LocationInformationResponse</i>	Antwort des Ortsinformationsdiensts (vgl. 8.4.1).
o	MapServiceResponse	<i>MapServiceResponse</i>	Antwort des Kartendienstes (vgl. 17.3.1).
p	PersonalisationResponse	<i>PersonalisationResponse</i>	Antwort des Personalisierungsdienstes (vgl. 20.5.1).
q	PositioningResponse	<i>PositioningResponse</i>	Antwort des Positionierungsdienstes (vgl. 11.3.1).
r	ServiceRegisterResponse	<i>ServiceRegisterResponse</i>	Antwort des Dienstregisterdienstes (vgl. 23.4.1).

<i>s</i>	SituationExchangeDelivery	<i>si-ri:SituationExchangeDelivery</i>	Aktualisierungsnachricht im Rahmen eines Abonnements zu Störungsinformationen (vgl. Kapitel 19).
<i>t</i>	StopEventResponse	<i>StopEventResponse</i>	Antwort mit Haltestellen-Abfahrtstafeln (vgl. 10.2.1).
<i>u</i>	StopRequestResponse	<i>StopRequestResponse</i>	Bestätigung zum Haltewunsch im Fahrzeug (vgl. 22.3.1).
<i>v</i>	TripInfoResponse	<i>TripInfoResponse</i>	Antwort mit Fahrtinformationen (vgl. 12.3.1).
<i>w</i>	TripMonitoringDelivery	<i>TripMonitoringDelivery</i>	Aktualisierungsnachricht im Rahmen eines Abonnements zu Verbindungszuständen (vgl. 19.4.1).
<i>x</i>	TripResponse	<i>TripResponse</i>	Antwort zur intermodalen Verbindungsberechnung (vgl. 9.3.1).
<i>y</i>	VehicleDataResponse	<i>VehicleDataResponse</i>	Antwort mit Fahrzeuginformationen (vgl. 21.3.1).

Tabelle 9: Beschreibung der Struktur **DeliveryPayloadStructure**.

7.2. Trias_Utility

In der XML-Schema-Definition *Trias_Utility.xsd* werden eine Reihe von Typen und Strukturen definiert, die als Basistypen in anderen Definitionen wieder verwendet werden. Die Definitionen in *Trias_Utility* haben keinen unmittelbaren Bezug zur Fachthematik des ÖV.

7.2.1. Einfache Typen

Folgende einfache Typen (simple types) werden definiert:

<i>PercentType</i>	<i>xs:nonNegativeInteger</i>	Prozentangabe als Ganzzahlwert. Maximalwert ist 100.
<i>OpenPercentType</i>	<i>xs:nonNegativeInteger</i>	Prozentangabe als Ganzzahlwert, nach oben unbeschränkt.
<i>BitStringType</i>	<i>xs:string</i>	Zeichenkette, die nur aus Nullen und Einsen bestehen kann.
<i>DistanceType</i>	<i>xs:nonNegativeInteger</i>	Typ zur Angabe von Distanzen (in Metern).
<i>LengthType</i>	<i>xs:nonNegativeInteger</i>	Typ zur Angabe von Längen (in Metern).
<i>SpeedType</i>	<i>xs:nonNegativeInteger</i>	Typ zur Angabe einer Geschwindigkeit (in Meter pro Sekunde).
<i>PriorityType</i>	<i>xs:nonNegativeInteger, [1,5]</i>	Prioritätswerte von 1 (höchste Priorität) bis 5 (niedrigste Priorität).
<i>LongitudeType</i>	<i>xs:decimal</i>	Geographische Länge.
<i>LatitudeType</i>	<i>xs:decimal</i>	Geographische Breite.
<i>AltitudeType</i>	<i>xs:decimal</i>	Höhe über dem Meeresspiegel in Meter.
<i>AbsoluteBearingType</i>	<i>xs:nonNegativeInteger</i>	Kompassrichtung in Grad. Nord = 0 Grad, im Uhrzeigersinn aufsteigende Werte.
<i>PhoneNumberType</i>	<i>xs:normalizedString</i>	Typ zur Angabe einer Telefonnummer.

Tabelle 10: Liste der einfachen Typdefinitionen in *Trias_Utility.xsd*.

7.2.2. InternationalTextStructure

An komplexen Strukturen findet sich in Trias_Utility Folgende:

<i>InternationalTextStructure</i>			+Structure	Ein Text mit einer Text-ID und Angabe der Sprache, in der er verfasst ist.
	Text	1:1	<i>xs:normalizedString</i>	Text.
	<i>TextId</i>	0:1	<i>xs:NMTOKEN</i>	ID des Texts.
	<i>Language</i>	0:1	<i>xs:language</i>	Sprache, in der der Text verfasst ist.

Tabelle 11: Beschreibung der Struktur **InternationalTextStructure**.

Elemente vom Typ *InternationalText* werden verwendet, um Texte in unterschiedlichen Sprachen angeben zu können.

7.2.3. GeoPositionStructure

<i>GeoPositionStructure</i>			+Structure	Geographische Position in WGS84.
	Longitude	1:1	<i>Longitude</i>	Geographische Länge bzgl. des Greenwich-Meridians. Wertebereich von -180 Grad (West) bis +180 Grad (Ost).
	Latitude	1:1	<i>Latitude</i>	Geographische Breite bzgl. des Äquators. Wertebereich von -90 Grad (Süden) bis +90 Grad (Norden).
	<i>Altitude</i>	0:1	<i>Altitude</i>	Höhe über dem Meeresspiegel in Meter.

Tabelle 12: Beschreibung der Struktur **GeoPositionStructure**.

7.3. Trias_ModesSupport

In der XML-Schema-Definition Trias_ModesSupport.xsd werden eine Reihe von Basistypen und -strukturen definiert, die zur Klassifizierung von Verkehrsmitteln dienen. Diese Definitionen lehnen sich sehr stark an die TPEG-Codierung an, die auch in SIRI verwendet wird.

7.3.1. Einfache Typen

Folgende einfache Typen (simple types) werden definiert:

<i>IndividualModesEnumeration</i>	<i>walk cycle taxi self-drive-car others-drive-car motorcycle truck</i>	Klassifizierung der Individualverkehrsarten.
<i>ContinuousModesEnumeration</i>	<i>walk demandResponsive replacementService</i>	Klassifizierung von kontinuierlichen Verkehrsarten, die zu jeder beliebigen Zeit (ohne Fahrplan) stattfinden können.
<i>InterchangeModesEnumeration</i>	<i>walk parkAndRide bikeAndRide carHire bikeHire protectedConnection guaranteedConnection remainInVehicle changeWithinVehicle checkIn checkOut</i>	Klassifizierung von Umsteigevorgängen
<i>PtModesEnumeration</i>	<i>all unknown air bus trolleyBus tram coach rail intercityRail urbanRail metro water cableway funicular taxi</i>	Klassifizierung der ÖV-Verkehrsmittel (nach TPEG pti_table 01).
<i>RailSubmodeEnumeration</i>	<i>unknown undefined</i>	Unter-Klassifizierung der Züge (nach TPEG pti_table

	<i>local highSpeedRail suburbanRailway regionalRail interregionalRail longDistance international sleeperRailService nightRail carTransportRailService touristRailway railShuttle replacementRailService specialTrain crossCountryRail rackAndPinionRailway</i>	02).
<i>CoachSubmodeEnumeration</i>	<i>unknown undefined internationalCoach nationalCoach shuttleCoach regionalCoach specialCoach sightseeingCoach touristCoach commuterCoach</i>	Unter-Klassifizierung der Überlandbusse (nach TPEG pti_table 03).
<i>MetroSubmodeEnumeration</i>	<i>unknown undefined metro tube urbanRailway</i>	Unter-Klassifizierung der U-Bahnen (nach TPEG pti_table 04).
<i>BusSubmodeEnumeration</i>	<i>unknown undefined localBus regionalBus expressBus nightBus postBus specialNeedsBus mobilityBus mobilityBusForRegisteredDisabled sightseeingBus shuttleBus schoolBus schoolAndPublicServiceBus railReplacementBus demandAndResponseBus airportLinkBus</i>	Unter-Klassifizierung der Busse (nach TPEG pti_table 05).
<i>TramSubmodeEnumeration</i>	<i>unknown undefined cityTram localTram regionalTram sightseeingTram shuttleTram</i>	Unter-Klassifizierung der Straßenbahnen (nach TPEG pti_table 06).
<i>WaterSubmodeEnumeration</i>	<i>unknown undefined internationalCarFerry nationalCarFerry regionalCarFerry localCarFerry internationalPassengerFerry nationalPassengerFerry regionalPassengerFerry localPassengerFerry postBoat trainFerry roadFerryLink airportBoatLink highSpeedVehicleService highSpeedPassengerService sightseeingService schoolBoat cableFerry riverBus scheduledFerry shuttleFerryService</i>	Unter-Klassifizierung der Wasserverkehrsmittel (nach TPEG pti_table 07).
<i>AirSubmodeEnumeration</i>	<i>unknown undefined internationalFlight domesticFlight</i>	Unter-Klassifizierung der Luftverkehrsmittel (nach TPEG pti_table 08).

	<i>intercontinentalFlight domesticScheduledFlight shuttleFlight intercontinentalCharterFlight internationalCharterFlight roundTripCharterFlight sightseeingFlight helicopterService domesticCharterFlight SchengenAreaFlight airshipService shortHaulInternationalFlight canalBarge</i>	
<i>TelecabinSubmodeEnumeration</i>	<i>unknown undefined telecabin cableCar lift chairLift dragLift telecabinLink</i>	Unter-Klassifizierung der Lift- und Aufzugsarten (nach TPEG pti_table 09).
<i>FunicularSubmodeEnumeration</i>	<i>unknown funicular allFunicularServices undefinedFunicular</i>	Unter-Klassifizierung der Seilbahnen (nach TPEG pti_table 10).
<i>TaxiSubmodeEnumeration</i>	<i>unknown undefined communalTaxi waterTaxi railTaxi bikeTaxi blackCab miniCab allTaxiServices</i>	Unter-Klassifizierung der Taxiarten (nach TPEG pti_table 11).

Tabelle 13: Liste der einfachen Typdefinitionen in Trias_ModesSupport.xsd.

Im Folgenden werden die komplexen Strukturen beschrieben, die in Trias_ModesSupport definiert sind.

7.3.2. IndividualTransportOptionsStructure

<i>IndividualTransportOptionsStructure</i>			+Structure	Arten von IV und deren Nutzungsgrenzen, wie sie der Benutzer vorgegeben hat.
	Mode	1:1	<i>IndividualModesEnumeration</i>	Angabe des IV-Typs. Zugelassen sind hier Werte für Fußweg, Fahrrad, Taxi, selbst gefahrenes Auto, durch andere gefahrenes Auto, Motorrad und LKW. Der Modus „selbst gefahrenes Auto“ benötigt beim Umstieg in ein anderes Verkehrsmittel einen längerfristigen Parkplatz und ist daher ein verallgemeinertes Synonym für Park&Ride. Der Modus „durch andere gefahrenes Auto“ benötigt dagegen nur einen Platz zum Aussteigen lassen.
	<i>MaxDistance</i>	0:1	<i>Distance</i>	Maximale Distanz, bis zu der die Nutzung dieses IV-Typs zugelassen ist.
	<i>MaxDuration</i>	0:1	<i>xs:duration</i>	Maximale Zeitdauer, bis zu der die Nutzung dieses IV-Typs zugelassen ist.
	<i>MinDistance</i>	0:1	<i>Distance</i>	Minimale Distanz, ab der die Nutzung dieses IV-Typs zugelassen ist.
	<i>MinDuration</i>	0:1	<i>xs:duration</i>	Minimale Zeitdauer, ab der die Nutzung dieses IV-Typs zugelassen ist.
	<i>Speed</i>	0:1	<i>OpenPercent</i>	Relative Geschwindigkeit in Prozent. Wert 100 stellt Standardgeschwindigkeit dar. Werte kleiner 100 verringern die Geschwindigkeit, Werte größer 100 vergrößern die Geschwindigkeit anteilig.

Tabelle 14: Beschreibung der Struktur *IndividualTransportOptionsStructure*.

7.3.3. PtSubmodeChoiceGroup

PtSubmodeChoiceGroup			+Group	Gruppe zur Auswahl der Verkehrsmitteluntertypen.	
	a	<i>AirSubmode</i>	-0:1	<i>AirSubmodeEnumeration</i>	Untertypen der Luftverkehrsmittel.
	b	<i>BusSubmode</i>		<i>BusSubmodeEnumeration</i>	Untertypen der Busse.
	c	<i>CoachSubmode</i>		<i>CoachSubmodeEnumeration</i>	Untertypen der Überlandbusse.
	d	<i>FunicularSubmode</i>		<i>FunicularSubmodeEnumeration</i>	Untertypen der Seilbahnen.
	e	<i>MetroSubmode</i>		<i>MetroSubmodeEnumeration</i>	Untertypen der U-Bahnen.
	f	<i>RailSubmode</i>		<i>RailSubmodeEnumeration</i>	Untertypen der Züge.
	g	<i>TelecabinSubmode</i>		<i>TelecabinSubmodeEnumeration</i>	Untertypen der Lift- und Aufzugsarten.
	h	<i>TramSubmode</i>		<i>TramSubmodeEnumeration</i>	Untertypen der Straßenbahnen.
	i	<i>WaterSubmode</i>		<i>WaterSubmodeEnumeration</i>	Untertypen der Wasserverkehrsmittel.

Tabelle 15: Beschreibung der Gruppe **PtSubmodeChoiceGroup**.

7.3.4. ModeStructure

ModeStructure			+Structure	Verkehrsmittel mit Klassifizierung und Namen.
<i>Mode</i>	<i>PtMode</i>	1:1	<i>PtModeEnumeration</i>	Angabe der ÖV-Verkehrsmittelart.
<i>PtSubmodeChoice</i>	:::	-0:1	<i>PtSubmodeChoice</i>	Verkehrsmitteluntertypen (vgl. 7.3.3)
	<i>Name</i>	0:1	<i>InternationalText</i>	Verkehrsmittelname.
	<i>ShortName</i>	0:1	<i>InternationalText</i>	Kurzname oder Abkürzung.
	<i>Description</i>	0:1	<i>InternationalText</i>	Beschreibender Text.

Tabelle 16: Beschreibung der Struktur **ModeStructure**.

7.3.5. PtModeFilterStructure

PtModeFilterStructure			+Structure	Struktur zum Filtern nach Verkehrsmitteltypen
	<i>Exclude</i>	0:1	<i>xs:boolean</i>	Indikator, ob die in der Liste angegebenen Verkehrsmittel ausgeschlossen (Wert <i>true</i>) oder als einzige verwendet werden sollen (Wert <i>false</i>). Voreinstellung ist <i>true</i> .
	<i>PtMode</i>	0:*	<i>PtModesEnumeration</i>	ÖV-Verkehrsmitteltypen.
<i>PtSubmodeChoice</i>	:::	0:*	<i>PtSubmodeChoice</i>	ÖV-Untertypen.

Tabelle 17: Beschreibung der Struktur **PtModeFilterStructure**.

7.4. Trias_Common

7.4.1. Einfache Typen

Folgende einfache Typen (simple types) werden definiert:

<i>ParticipantCodeType</i>	<i>xs:normalizedString</i>	ID eines Kommunikationspartners.
<i>OperatorCodeType</i>	<i>xs:NMTOKEN</i>	ID eines Verkehrsunternehmens.
<i>LineCodeType</i>	<i>xs:NMTOKEN</i>	ID einer Linie.
<i>DirectionCodeType</i>	<i>xs:NMTOKEN</i>	ID einer Linienrichtung.
<i>JourneyCodeType</i>	<i>xs:NMTOKEN</i>	ID einer Fahrplanfahrt.
<i>VehicleCodeType</i>	<i>xs:NMTOKEN</i>	ID eines Fahrzeugs.
<i>FacilityCodeType</i>	<i>xs:NMTOKEN</i>	ID einer Einrichtung/Ausstattung.
<i>OwnerCodeType</i>	<i>xs:NMTOKEN</i>	ID einer verantwortlichen Organisation (Eigentümer).
<i>OperatingDayCodeType</i>	<i>xs:NMTOKEN</i>	ID eines Verkehrstags.

Tabelle 18: Liste der einfachen Typdefinitionen in Trias_Common.xsd.

Damit die Codes von Linien, Verkehrsunternehmen etc. systemübergreifend verstanden werden können, sind bestimmte Vereinbarungen zu treffen. Diese sind in Kapitel 5 beschrieben.

Im Folgenden werden die komplexen Strukturen beschrieben, die in Trias_Common definiert sind.

7.4.2. ErrorMessageStructure

ErrorMessageStructure		+Structure	Struktur zur Meldung von Fehlerzuständen.
	Code	1:1	<i>xs:normalizedString</i> Code des Fehlerzustands.
	Text	0:1	<i>+InternationalText</i> Beschreibung des Fehlerzustands.

Tabelle 19: Beschreibung der Struktur **ErrorMessageStructure**.

7.4.3. PrivateCodeStructure

PrivateCodeStructure		+Structure	Objekt-ID innerhalb eines proprietären (privaten) Schlüsselsystems (Fremdschlüssel).
	System	1:1	<i>xs:NMTOKEN</i> Bezeichnung des Schlüsselsystems.
	Value	1:1	<i>xs:NMTOKEN</i> Code/Objekt-ID.

Tabelle 20: Beschreibung der Struktur **PrivateCodeStructure**.

7.4.4. OperatorFilterStructure

OperatorFilterStructure		+Structure	Struktur zum Filtern nach Verkehrsunternehmen.
	Exclude	0:1	<i>xs:boolean</i> Indikator, ob die in der Liste angegebenen Verkehrsunternehmen ausgeschlossen (Wert <i>true</i>) oder als einzige verwendet werden sollen (Wert <i>false</i>). Voreinstellung ist <i>true</i> .
	OperatorRef	0:*	<i>→Operator</i> Referenz auf Verkehrsunternehmen. Vgl. 7.4.1.

Tabelle 21: Beschreibung der Struktur **OperatorFilterStructure**.

7.4.5. LineDirectionStructure

<i>LineDirectionStructure</i>		+Structure	Linien-ID, evtl. verfeinert auf eine Richtung
<i>LineRef</i>	1:1	→ <i>LineCode</i>	Referenz auf die Linie. Vgl. 7.4.1.
<i>DirectionRef</i>	0:1	→ <i>DirectionCode</i>	Referenz auf die Linienrichtung. Vgl. 7.4.1.

Tabelle 22: Beschreibung der Struktur *LineDirectionStructure*.

7.4.6. LineDirectionFilterStructure

<i>LineDirectionFilterStructure</i>		+Structure	Filterstruktur zum Ein/Ausschließen von Linien(richtungen)
<i>Line</i>	1:*	+ <i>LineDirection</i>	Referenz auf die Linie (vgl. 7.4.5).
<i>Exclude</i>	0:1	<i>xs:boolean</i>	Indikator, ob die Linien(richtungen) dieser Liste in die Suche aufgenommen oder von ihr ausgeschlossen werden sollen. Default ist Ausschluss (Exclude).

Tabelle 23: Beschreibung der Struktur *LineDirectionFilterStructure*.

7.4.7. OperatingDaysStructure

<i>OperatingDaysStructure</i>		+Structure	Struktur für die Definition von Verkehrstagen mittels Bit-Kette.
<i>From</i>	1:1	<i>xs:date</i>	Startdatum des Zeitraums.
<i>To</i>	1:1	<i>xs:date</i>	Enddatum des Zeitraums.
<i>Pattern</i>	1:1	<i>BitString</i>	Bitmuster für die Verkehrstage im Zeitraum von Startdatum (<i>From</i>) bis Enddatum (<i>To</i>). Die Länge des Bitmusters in <i>Pattern</i> entspricht der Anzahl der Tage von <i>From</i> bis <i>To</i> . Eine „1“ bedeutet, dass das in Frage kommende Ereignis an dem Tag stattfindet, der der Position in der Bitkette entspricht.

Tabelle 24: Beschreibung der Struktur *OperatingDaysStructure*.

7.4.8. WeekdayTimePeriodStructure

<i>WeekdayTimePeriodStructure</i>		+Structure	Struktur für die Definition von Verkehrstagen.
<i>Weekday</i>	0:1	<i>Sunday</i> <i>Monday</i> <i>Tuesday</i> <i>Wednesday</i> <i>Thursday</i> <i>Friday</i> <i>Saturday</i> <i>PublicHoliday</i>	Wochentagstyp.
<i>StartTime</i>	1:1	<i>xs:time</i>	Startzeit des Zeitintervalls.
<i>Duration</i>	1:1	<i>xs:duration</i>	Dauer des Zeitintervalls.

Tabelle 25: Beschreibung der Struktur *WeekdayTimePeriodStructure*.

7.4.9. GeneralAttributeStructure

GeneralAttributeStructure			+Structure	Struktur für die Definition von Attributen/Hinweisen.
	Text	1:1	+InternationalText	Attributtext für die Fahrgastinformation.
	Code	1:1	xs:NMTOKEN	Interner Attribute-Code. Kann verwendet werden, um mehrfaches Auftreten desselben Attributs zu erkennen.
<i>AllFacilities</i>	:::	0:1	+AllFacilitiesGroup	Klassifizierung nach TPEG. Vgl. 7.7.4.
	<i>Mandatory</i>	0:1	xs:boolean	Legt fest, ob das Attribut in jedem Fall angezeigt werden muss. Voreinstellung ist <i>false</i> .
	<i>Importance</i>	0:1	Percent	Wichtigkeit für die Priorisierung von Attributen gegeneinander.
	<i>InfoURL</i>	0:1	xs:anyURI	URL zu weiteren Informationen über dieses Attribut. Falls vorhanden, soll der gesamte Text als Link zu dieser URL gekennzeichnet werden.

Tabelle 26: Beschreibung der Struktur **GeneralAttributeStructure**.

7.5. Trias_LocationSupport

In der XML-Schema-Definition *Trias_LocationSupport.xsd* werden eine Reihe von Basistypen und -strukturen definiert, die als Ortsbezüge (Haltestellen, Haltepunkte, Ortschaften und POIs) und Beschreibungen von Haltestellen und Haltepunkten in anderen Definitionen wieder verwendet werden können.

7.5.1. Einfache Typen

Folgende einfache Typen (simple types) werden definiert:

<i>StopPointCodeType</i>	<i>xs:normalizedString</i>	Code für einen Haltepunkt.
<i>StopPlaceCodeType</i>	<i>xs:normalizedString</i>	Code für eine Haltestelle.
<i>LocalityCodeType</i>	<i>xs:normalizedString</i>	Code für eine Ortschaft.
<i>PointOfInterestCodeType</i>	<i>xs:normalizedString</i>	Code für einen POI.
<i>AddressCodeType</i>	<i>xs:normalizedString</i>	Code für eine Adresse.

Tabelle 27: Liste der einfachen Typdefinitionen in *Trias_LocationSupport.xsd*.

Damit die Codes von Haltestellen, Haltepunkten etc. systemübergreifend verstanden werden können, sind bestimmte Vereinbarungen zu treffen. Diese sind in Kapitel 5 beschrieben.

Im Folgenden werden die komplexen Strukturen beschrieben, die in *Trias_LocationSupport* definiert sind.

7.5.2. StopPointStructure

StopPointStructure			+Structure	Modellierung eines Haltepunkts.
Stop-Point	StopPointRef	1:1	→ <i>StopPoint</i>	Referenz auf einen Code für einen Haltepunkt. Vgl. 7.5.1.
	StopPointName	1:1	+ <i>International Text</i>	Name des Haltepunkts für Fahrgastinformation.
	<i>NameSuffix</i>	0:1	+ <i>International Text</i>	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „gegenüber vom Haupteingang“.
	<i>PlannedBay</i>	0:1	+ <i>International Text</i>	Name des Steigs/Haltepunkts, wo in das Fahrzeug ein- oder ausgestiegen werden muss (bei Verwendung in Zusammenhang mit einer konkreten Verbindungsauskunft, wenn in StopPointName ein allgemeiner Name angegeben ist, ähnlich Haltestellenname). Nach Planungsstand.
	<i>EstimatedBay</i>	0:1	+ <i>International Text</i>	Name des Steigs/Haltepunkts, wo in das Fahrzeug ein- oder ausgestiegen werden muss (bei Verwendung in Zusammenhang mit einer konkreten Verbindungsauskunft, wenn in StopPointName ein allgemeiner Name angegeben ist, ähnlich Haltestellenname). Nach letztem Prognosestand.
	<i>PrivateCode</i>	0:*	+ <i>PrivateCode</i>	Privater Code für diesen Haltepunkt in einem anderen Schlüsselssystem. Vgl. 7.4.3.
	<i>ParentRef</i>	0:1	→ <i>StopPlace</i>	Referenz auf die Haltestelle, zu der dieser Haltepunkt gehört. Vgl. 7.5.1.
	<i>LocalityRef</i>	0:1	→ <i>Locality</i>	Referenz auf die Ortschaft, zu der dieser Haltepunkt gehört. Vgl. 7.5.1.
StopAttributes	<i>WheelchairAccessible</i>	0:1	<i>xs:boolean</i>	Rollstuhltauglichkeit dieses Haltepunkts.
	<i>Lighting</i>	0:1	<i>xs:boolean</i>	Angabe zur Beleuchtung dieses Haltepunkts.
	<i>Covered</i>	0:1	<i>xs:boolean</i>	Angabe, ob dieser Haltepunkt Witterungsschutz bietet (vor Regen, Schnee, Sturm etc.).

Tabelle 28: Beschreibung der Struktur **StopPointStructure**.

7.5.3. StopPlaceStructure

StopPlaceStructure			+Structure	Modellierung einer Haltestelle.
Stop-Place	StopPlaceRef	1:1	→ <i>StopPlace</i>	Referenz auf einen Code für eine Haltestelle. Vgl. 7.5.1.
	StopPlaceName	1:1	+ <i>International Text</i>	Name der Haltestelle für Fahrgastinformation.
	<i>NameSuffix</i>	0:1	+ <i>International Text</i>	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „Messe/Exhibition Center“.
	<i>PrivateCode</i>	0:*	+ <i>PrivateCode</i>	Privater Code für diese Haltestelle in einem anderen Schlüsselssystem.
	<i>LocalityRef</i>	0:1	→ <i>Locality</i>	Referenz auf die Ortschaft, zu der diese Haltestelle gehört. Vgl. 7.5.1.
StopAttributes	<i>WheelchairAccessible</i>	0:1	<i>xs:boolean</i>	Rollstuhltauglichkeit dieser Haltestelle insgesamt.
	<i>Lighting</i>	0:1	<i>xs:boolean</i>	Angabe zur Beleuchtung dieser Haltestelle.
	<i>Covered</i>	0:1	<i>xs:boolean</i>	Angabe, ob diese Haltestelle Witterungsschutz bietet (vor Regen, Schnee, Sturm etc.).

Tabelle 29: Beschreibung der Struktur **StopPlaceStructure**.

7.5.4. LocalityStructure

LocalityStructure			+Structure	Modellierung einer Ortschaft/Stadt.
	LocalityCode	1:1	→Locality	Identifikator der Ortschaft/Stadt. Vgl. 7.5.1.
	LocalityName	1:1	+International Text	Name der Ortschaft für Fahrgastinformation.
	<i>PrivateCode</i>	0:*	+PrivateCode	Privater Code für diesen Haltepunkt in einem anderen Schlüsselssystem.
	<i>ParentRef</i>	0:1	→Locality	Referenz auf eine übergeordnete Ortschaft, zu der diese Ortschaft gehört, z. B. Beziehung Stadtteil zu Stadt. Vgl. 7.5.1.
Area	<i>Points</i>	3:*	+GeoPosition	Polygonzug, der das Gebiet der Ortschaft beschreibt.

Tabelle 30: Beschreibung der Struktur **LocalityStructure**.

7.5.5. PointOfInterestStructure

PointOfInterestStructure			+Structure	Modellierung eines wichtigen Punkts (POI).
	PointOfInterestCode	1:1	→PointOfInterest	Identifikator des wichtigen Punkts.
	PointOfInterestName	1:1	+International Text	Name des wichtigen Punkts für Fahrgastinformation.
	<i>NameSuffix</i>	0:1	+International Text	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „Messe/Exhibition Center“.
	<i>PrivateCode</i>	0:*	+PrivateCode	Privater Code für diesen Haltepunkt in einem anderen Schlüsselssystem.
	<i>LocalityRef</i>	0:1	→Locality	Referenz auf die zugeordnete Ortschaft, zu der dieser wichtige Punkt gehört. Vgl. 7.5.1.

Tabelle 31: Beschreibung der Struktur **PointOfInterestStructure**.

7.5.6. AddressStructure

AddressStructure			+Structure	Modellierung einer Adresse.
	AddressCode	1:1	→Address	Identifikator der Adresse. Vgl. 7.5.1.
	AddressName	1:1	+International Text	Adressbeschriftung für Fahrgastinformation.
	<i>NameSuffix</i>	0:1	+International Text	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „Messe/Exhibition Center“.
Address	<i>CountryName</i>	0:1	xs:string	Angabe zum Land.
	<i>PostalCode</i>	0:1	xs:string	Postleitzahl.
	CityName	1:1	xs:string	Name der Stadt, in der die Adresse liegt.
	StreetName	1:1	xs:string	Name der Straße, in der die Adresse liegt.
	<i>HouseNumber</i>	0:1	xs:string	Hausnummer. Falls leer, kann a) in <i>CrossingStreet</i> eine Kreuzung angegeben werden oder b) die Straße ist als Ganzes gemeint.
	<i>CrossingStreet</i>	0:1	xs:string	Name der kreuzenden Straße.
	<i>PrivateCode</i>	0:*	+PrivateCode	Privater Code für diesen Haltepunkt in einem anderen Schlüsselssystem.
	<i>LocalityRef</i>	0:1	→Locality	Referenz auf die zugeordnete Ortschaft, zu der dieser wichtige Punkt gehört. Vgl. 7.5.1.

Tabelle 32: Beschreibung der Struktur **AddressStructure**.

7.5.7. LocationStructure

LocationStructure			+Structure	Modell eines allgemeinen Ortspunkts (Haltepunkt, Haltestelle, Koordinatenposition, Ortschaft oder POI).
a	<i>StopPoint</i>	-0:1	→ <i>StopPoint</i>	Referenz auf einen Code für einen Haltepunkt. Vgl. 7.5.1.
b	<i>StopPlace</i>		→ <i>StopPlace</i>	Referenz auf einen Code für eine Haltestelle. Vgl. 7.5.1.
c	<i>Locality</i>		→ <i>Locality</i>	Referenz auf einen Code für eine Ortschaft. Vgl. 7.5.1.
d	<i>PointOfInterest</i>		→ <i>PointOfInterest</i>	Referenz auf einen Code für einen POI. Vgl. 7.5.1.
e	<i>Address</i>		→ <i>Address</i>	Referenz auf eine Adresse. Vgl. 7.5.1.
LocationName		1:1	+ <i>InternationalText</i>	Name oder Bezeichnung des Ortspunkts.
GeoPosition		1:1	+ <i>GeoPosition</i>	Koordinatenposition.
<i>Attribute</i>		0:*	+ <i>GeneralAttribute</i>	Attribute, die dem Ortspunkt zugeordnet sind. Vgl. 7.4.9.
<i>Extension</i>		0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 33: Beschreibung der Struktur **LocationStructure**.

7.5.8. LocationRefStructure

LocationRefStructure			+Structure	Referenz auf einen allgemeinen Ortspunkt (Haltepunkt, Haltestelle, Koordinatenposition, Ortschaft oder POI).
a	StopPointRef	-1:1	→ <i>StopPoint</i>	Referenz auf einen Code für einen Haltepunkt. Vgl. 7.5.1.
b	StopPlaceRef		→ <i>StopPlace</i>	Referenz auf einen Code für eine Haltestelle. Vgl. 7.5.1.
c	GeoPosition		+ <i>GeoPosition</i>	Koordinatenposition.
d	LocalityRef		→ <i>Locality</i>	Referenz auf einen Code für eine Ortschaft. Vgl. 7.5.1.
e	PointOfInterestRef		→ <i>PointOfInterest</i>	Referenz auf einen Code für einen POI. Vgl. 7.5.1.
f	AddressRef		→ <i>Address</i>	Referenz auf eine Adresse. Vgl. 7.5.1.
<i>LocationName</i>		0:1	+ <i>InternationalText</i>	Name oder Bezeichnung des Ortspunkts.

Tabelle 34: Beschreibung der Struktur **LocationRefStructure**.

7.6. Trias_JourneySupport

In der XML-Schema-Definition Trias_JourneySupport.xsd werden Strukturen beschrieben, die den Fahrbetrieb des ÖV beschreiben. Dazu gehören u. a. die Beschreibung einer Fahrzeugfahrt, Informationen zu Ankünften und Abfahrten an Haltestellen, sowie die Bewegung eines Fahrzeugs entlang des Fahrwegs.

Im Folgenden werden die komplexen Strukturen beschrieben, die in Trias_JourneySupport definiert sind.

7.6.1. ServiceViaPointStructure

ServiceViaPointStructure			+Structure	Via-Punkt auf dem Fahrweg.
Stop-Point	StopPointRef	1:1	→ <i>StopPoint</i>	Referenz auf einen Code für einen Haltepunkt. Vgl. 7.5.1.
	StopPointName	1:1	+ <i>InternationalText</i>	Name des Haltepunkts für Fahrgastinformation.
	<i>NameSuffix</i>	0:1	+ <i>InternationalText</i>	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „gegenüber vom Haupteingang“.
	<i>PlannedBay</i>	0:1	+ <i>InternationalText</i>	Name des Steigs/Haltepunkts, wo in das Fahrzeug ein- oder ausgestiegen werden muss (bei Verwendung in Zusammenhang mit einer konkreten Verbindungsauskunft, wenn in StopPointName ein allgemeiner Name angegeben ist, ähnlich Haltestellenname). Nach Planungsstand.
	<i>EstimatedBay</i>	0:1	+ <i>InternationalText</i>	Name des Steigs/Haltepunkts, wo in das Fahrzeug ein- oder ausgestiegen werden muss (bei Verwendung in Zusammenhang mit einer konkreten Verbindungsauskunft, wenn in StopPointName ein allgemeiner Name angegeben ist, ähnlich Haltestellenname). Nach letztem Prognosestand.
	<i>DisplayPriority</i>	0:1	<i>Priority</i>	Priorität, mit der dieser Via-Punkt angezeigt werden soll (z. B. falls Platz knapp ist und nicht alle Via-Punkte angezeigt werden können).

Tabelle 35: Beschreibung der Struktur **ServiceViaPointStructure**.

7.6.2. ViaStructure

ViaStructure			+Structure	Angaben zu einer Via-Bedingung.
	ViaPoint	1:1	+ <i>LocationRef</i>	Referenz auf den Via-Punkt. Vgl. 7.5.8.
	<i>DwellTime</i>	0:*	<i>xs:duration</i>	Vom Benutzer vorgeschriebene Mindestaufenthaltszeit am Via-Punkt.

Tabelle 36: Beschreibung der Struktur **ViaStructure**.

7.6.3. DatedJourneyStructure

DatedJourneyStructure			+Structure	Fahrplanfahrt an bestimmtem Tag.
<i>DatedService</i>	OperatingDayRef	1:1	→ <i>OperatingDay</i>	Betriebstag der Fahrt. Vgl. 7.4.1.
	<i>VehicleRef</i>	0:1	→ <i>Vehicle</i>	Fahrzeug-ID. Vgl. 7.4.1.
<i>ServiceJourney</i>	JourneyRef	1:1	→ <i>Journey</i>	Fahrt-ID. Vgl. 7.4.1.
<i>LineIdentity</i>	LineRef	1:1	→ <i>Line</i>	Linien-ID. Vgl. 7.4.1.
	DirectionRef	1:1	→ <i>Direction</i>	Richtungs-ID. Vgl. 7.4.1.
Service	Mode	1:1	+ <i>Mode</i>	Verkehrsmitteltyp. Vgl. 7.3.4.
	PublishedLineName	1:1	+ <i>InternationalText</i>	Liniennummer oder -name, wie in der Öffentlichkeit bekannt.
	<i>OperatorRef</i>	0:1	→ <i>Operator</i>	Operator-ID. Vgl. 7.4.1.
	<i>RouteDescription</i>	0:1	+ <i>InternationalText</i>	Beschreibung des Fahrwegs.
	<i>Via</i>	0:*	+ <i>ServiceViaPoint</i>	Wichtige Halte auf dem Fahrweg. Vgl. 7.6.1.
	<i>Attribute</i>	0:*	+ <i>GeneralAttribute</i>	Hinweise und Attribute (mit Klassifikationen) zur Fahrt. Vgl. 7.4.9.

Service Origin	OriginStopPointRef	0:1	→StopPoint	ID des ersten Haltepunkts der Fahrt; Starthaltestelle. Vgl. 7.5.1.
	OriginText	0:1	+InternationalText	Name des ersten Haltepunkts der Fahrt, der Starthaltestelle.
ServiceDestination	DestinationStopPointRef	0:1	→StopPoint	ID des letzten Haltepunkts der Fahrt; Endhaltestelle. Vgl. 7.5.1.
	DestinationText	0:1	+InternationalText	Name des letzten Haltepunkts der Fahrt, der Endhaltestelle.
ServiceStatus	Unplanned	0:1	xs:boolean	Gibt an, ob es sich um eine zusätzliche, ungeplante Fahrt handelt. Voreinstellung ist <i>false</i> .
	Cancelled	0:1	xs:boolean	Gibt an, ob diese Fahrt zur Gänze entfällt. Voreinstellung ist <i>false</i> .
	Deviation	0:1	xs:boolean	Gibt an, ob diese Fahrt einen anderen Weg nimmt. Voreinstellung ist <i>false</i> .
	Occupancy	0:1	manySeatsAvailable fewSeatsAvailable noSeatsAvailable standingAvailable full	Auslastungszustand des Fahrzeugs.
	SituationFullRef	0:*	+SituationFullRef	Verweis auf eine Störungsnachricht. Diese Nachricht kann im <i>StopEventResponseContext</i> (vgl. 10.3.2) zu finden sein oder auf anderem Wege bekannt gemacht werden.

Tabelle 37: Beschreibung der Struktur **DatedJourneyStructure**.

7.6.4. TripLocationStructure

TripLocationStructure			+Structure	Fahrplanfahrt als momentaner Aufenthaltsort eines Fahrgasts
	OperatingDayRef	1:1	→OperatingDay	Betriebstag der Fahrt. Vgl. 7.4.1.
	JourneyRef	1:1	→Journey	Fahrt-ID. Vgl. 7.4.1.
LineIdentity	LineRef	1:1	→Line	Linien-ID. Vgl. Vgl. 7.4.1.
	DirectionRef	1:1	→Direction	Richtungs-ID. Vgl. 7.4.1.

Tabelle 38: Beschreibung der Struktur **TripLocationStructure**.

7.6.5. CallAtStopStructure

CallAtStopStructure			+Structure	Halt einer Fahrt an einem Haltepunkt oder Haltestelle.
StopPoint	StopPointRef	1:1	→StopPoint	Referenz auf einen Code für einen Haltepunkt. Vgl. 7.5.1.
	StopPointName	1:1	+InternationalText	Name des Haltepunkts für Fahrgastinformation.
	NameSuffix	0:1	+InternationalText	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „gegenüber vom Haupteingang“.
	PlannedBay	0:1	+InternationalText	Name des Steigs/Haltepunkts, wo in das Fahrzeug ein- oder ausgestiegen werden muss (bei Verwendung in Zusammenhang mit einer konkreten Verbindungsauskunft, wenn in StopPointName ein allgemeiner Name angegeben ist, ähnlich Haltestellenname). Nach Planungsstand.

	<i>EstimatedBay</i>	0:1	+ <i>InternationalText</i>	Name des Steigs/Haltepunkts, wo in das Fahrzeug ein- oder ausgestiegen werden muss (bei Verwendung in Zusammenhang mit einer konkreten Verbindungsauskunft, wenn in <i>StopPointName</i> ein allgemeiner Name angegeben ist, ähnlich Haltestellenname). Nach letztem Prognosestand.
<i>Service Arrival</i>	<i>TimetabledTime</i>	1:1	<i>xs:dateTime</i>	Ankunftszeit nach Fahrplan.
	<i>RecordedAtTime</i>	0:1	<i>xs:dateTime</i>	Tatsächliche Ankunftszeit.
	<i>EstimatedTime</i>	0:1	<i>xs:dateTime</i>	Erwartete Ankunftszeit.
	<i>EstimatedTimeLow</i>	0:1	<i>xs:dateTime</i>	Untere Schranke für erwartete Ankunftszeit.
	<i>EstimatedTimeHigh</i>	0:1	<i>xs:dateTime</i>	Obere Schranke für erwartete Ankunftszeit.
<i>Service Departure</i>	<i>TimetabledTime</i>	1:1	<i>xs:dateTime</i>	Abfahrtszeit nach Fahrplan.
	<i>RecordedAtTime</i>	0:1	<i>xs:dateTime</i>	Tatsächliche Abfahrtszeit.
	<i>EstimatedTime</i>	0:1	<i>xs:dateTime</i>	Erwartete Abfahrtszeit.
	<i>EstimatedTimeLow</i>	0:1	<i>xs:dateTime</i>	Untere Schranke für erwartete Abfahrtszeit.
	<i>EstimatedTimeHigh</i>	0:1	<i>xs:dateTime</i>	Obere Schranke für erwartete Abfahrtszeit.
<i>StopCall Status</i>	<i>StopSeqNumber</i>	1:1	<i>xs:positiveInteger</i>	Laufende Nummer des Halts im Fahrweg der Fahrt.
	<i>DemandStop</i>	0:1	<i>xs:boolean</i>	Bedarfshalt. Fahrzeug bedient diesen Halt nur nach Voranmeldung.
	<i>UnplannedStop</i>	0:1	<i>xs:boolean</i>	Halt, der laut Planung nicht vorgesehen war.
	<i>NotServicedStop</i>	0:1	<i>xs:boolean</i>	Entgegen der Planung findet kein Halt statt.
	<i>SituationFullRef</i>	0:*	+ <i>SituationFullRef</i>	Verweis auf eine Störungsnachricht. Diese Nachricht kann im ResponseContext der Antwort zu finden sein oder auf anderem Wege bekannt gemacht werden. Vgl. 7.8.2.

Tabelle 39: Beschreibung der Struktur **CallAtStopStructure**.

7.6.6. DatedCallAtLocationStructure

<i>DatedCallAtLocationStructure</i>			+ <i>Structure</i>	Bedienung eines allgemeinen Ortspunkts durch eine Fahrt an einem bestimmten Tag
<i>DatedJourneyRef</i>	<i>JourneyRef</i>	1:1	→ <i>Journey</i>	Fahrt-ID. Vgl. 7.4.1.
	<i>OperatingDayRef</i>	1:1	→ <i>OperatingDay</i>	Betriebstag der Fahrt. Vgl. 7.4.1.
	<i>CallLocation</i>	1:1	+ <i>LocationRef</i>	Verallgemeinerter Abfahrtspunkt. Im Normalfall eine Haltestelle, kann aber auch eine Adresse oder Koordinate sein, wenn es sich um flexible Linien oder Flächenbedarfsverkehre handelt. Vgl. 7.5.8.
<i>Service Arrival</i>	<i>TimetabledTime</i>	1:1	<i>xs:dateTime</i>	Ankunftszeit nach Fahrplan.
	<i>RecordedAtTime</i>	0:1	<i>xs:dateTime</i>	Tatsächliche Ankunftszeit.
	<i>EstimatedTime</i>	0:1	<i>xs:dateTime</i>	Erwartete Ankunftszeit.
	<i>EstimatedTimeLow</i>	0:1	<i>xs:dateTime</i>	Untere Schranke für erwartete Ankunftszeit.
	<i>EstimatedTimeHigh</i>	0:1	<i>xs:dateTime</i>	Obere Schranke für erwartete Ankunftszeit.
<i>Service Departure</i>	<i>TimetabledTime</i>	1:1	<i>xs:dateTime</i>	Abfahrtszeit nach Fahrplan.
	<i>RecordedAtTime</i>	0:1	<i>xs:dateTime</i>	Tatsächliche Abfahrtszeit.
	<i>EstimatedTime</i>	0:1	<i>xs:dateTime</i>	Erwartete Abfahrtszeit.
	<i>EstimatedTimeLow</i>	0:1	<i>xs:dateTime</i>	Untere Schranke für erwartete Abfahrtszeit.
	<i>EstimatedTimeHigh</i>	0:1	<i>xs:dateTime</i>	Obere Schranke für erwartete Abfahrtszeit.
<i>StopCall Status</i>	<i>StopSeqNumber</i>	1:1	<i>xs:positiveInteger</i>	Laufende Nummer des Halts im Fahrweg der Fahrt.

	<i>DemandStop</i>	0:1	<i>xs:boolean</i>	Bedarfhalt. Fahrzeug bedient diesen Halt nur nach Voranmeldung.
	<i>UnplannedStop</i>	0:1	<i>xs:boolean</i>	Halt, der laut Planung nicht vorgesehen war.
	<i>NotServicedStop</i>	0:1	<i>xs:boolean</i>	Entgegen der Planung findet kein Halt statt.

Tabelle 40: Beschreibung der Struktur **DatedCallAtLocationStructure**.

7.6.7. ContinuousServiceStructure

ContinuousServiceStructure			+Structure	Eine Fahrgastbewegung mit Hilfe eines kontinuierlichen, nicht fahrplangebundenen Verkehrsmittels.
	a ContinuousMode	-1:1	<i>walk demandResponsive replacement-Service</i>	Modalität für kontinuierliche Verkehre.
	b IndividualMode		<i>walk cycle taxi self-drive-car others-drive-car motorcycle truck</i>	Verkehrsmittelmodalität für Individualverkehr.
<i>Dated-Service</i>	OperatingDay	1:1	<i>→OperatingDay</i>	Betriebstag der Fahrt. Vgl. 7.4.1.
	<i>VehicleRef</i>	0:1	<i>→Vehicle</i>	Fahrzeug-ID. Vgl. 7.4.1.
<i>Service Journey</i>	JourneyRef	1:1	<i>→Journey</i>	Fahrt-ID. Vgl. 7.4.1.
<i>Linelen- tity</i>	LineRef	1:1	<i>→Line</i>	Linien-ID. Vgl. 7.4.1.
	DirectionRef	1:1	<i>→Direction</i>	Richtungs-ID. Vgl. 7.4.1.
<i>Service</i>	Mode	1:1	+Mode	Verkehrsmitteltyp. Vgl. 7.3.4.
	PublishedLineName	1:1	<i>International-Text</i>	Liniennummer oder -name, wie in der Öffentlichkeit bekannt.
	<i>OperatorRef</i>	0:1	<i>→Operator</i>	Operator-ID. Vgl. 7.4.1.
	<i>RouteDescription</i>	0:1	<i>International-Text</i>	Beschreibung des Fahrwegs.
	<i>Via</i>	0:*	+ServiceViaPoint	Wichtige Halte auf dem Fahrweg. Vgl. 7.6.1.
<i>Service Origin</i>	<i>OriginStopPointRef</i>	0:1	<i>→StopPoint</i>	ID des ersten Haltepunkts der Fahrt; Starthaltestelle. Vgl. 7.5.1.
	<i>OriginText</i>	0:1	<i>International-Text</i>	Name des ersten Haltepunkts der Fahrt, der Starthaltestelle.
<i>ServiceDes- tination</i>	<i>DestinationStopPointRef</i>	0:1	<i>→StopPoint</i>	ID des letzten Haltepunkts der Fahrt; Endhaltestelle. Vgl. 7.5.1.
	<i>DestinationText</i>	0:1	<i>International-Text</i>	Name des letzten Haltepunkts der Fahrt, der Endhaltestelle.
	<i>SituationFullRef</i>	0:*	+SituationFullRef	Verweis auf eine Störungsnachricht. Diese Nachricht kann im Kontext der Meldung (ResponseContext) zu finden sein oder auf anderem Wege bekannt gemacht werden.

Tabelle 41: Beschreibung der Struktur **ContinuousServiceStructure**.

7.6.8. VehiclePositionStructure

VehiclePositionStructure			+Structure	Geographische und logische Position eines Fahrzeugs.
	<i>GeoPosition</i>	0:1	+GeoPosition	Geographische Position (vgl. 7.2.3).
	<i>Progress</i>	0:1	<i>Not yet operated Operation finished At stop Between stops</i>	Logische Position bezogen auf Haltestellenfolge im Fahrplan.
	<i>Bearing</i>	0:1	<i>AbsoluteBearing</i>	Kompassrichtung in Grad, in die sich das Fahrzeug bewegt (vgl. 7.2.1).
	<i>ProgressBetweenStops</i>	0:1	+ProgressBetweenStops	Position zwischen der zuletzt besuchten Haltestelle und der aktuellen Position (vgl. 7.6.9).

Tabelle 42: Beschreibung der Struktur **VehiclePositionStructure**.

7.6.9. ProgressBetweenStopsStructure

ProgressBetweenStopsStructure			+Structure	Position zwischen der zuletzt besuchten Haltestelle und der aktuellen Position.
	<i>LinkDistance</i>	0:1	<i>Distance</i>	Gesamtdistanz in Meter zwischen der letzten und der nächsten Haltestelle.
	<i>Percentage</i>	0:1	<i>Percent</i>	Prozentanteil, den das Fahrzeug entlang der Streckendistanz (LinkDistance) zurückgelegt hat.

Tabelle 43: Beschreibung der Struktur **ProgressBetweenStopsStructure**.

7.6.10. LocationContextStructure

LocationContextStructure			+Structure	Angabe eines Orts und der Möglichkeiten, wie ein Benutzer ihn per IV erreichen kann.
	a LocationRef	-1:1	+LocationRef	Angabe eines räumlichen Orts (vgl. 7.5.8).
	b TripLocation		+TripLocation	Aufenthaltort in einem (sich bewegenden) Fahrzeug (vgl. 7.6.4).
	<i>DepArrTime</i>	0:1	<i>xs:dateTime</i>	Beabsichtigte Abfahrts- oder Ankunftszeit an dem in <i>Location</i> oder <i>TripLocation</i> bezeichneten Ort.
	<i>IndividualTransportOptions</i>	0:*	+IndividualTransportOptions	Angaben des Benutzers, wie er/sie den Ort mittels IV erreichen/verlassen könnte (vgl. 7.3.2).

Tabelle 44: Beschreibung der Struktur **LocationContextStructure**.

Elemente vom Typ *LocationContextStructure* werden vor allem dazu benutzt, um den Start- (oder Ziel-) Kontext zu beschreiben, in dem ein Reisender sich befindet. Beim Dienst Verbindungsauskunft z. B. dienen Elemente dieses Typs als Angabe von Start und Ziel. Die Implementierung des Suchalgorithmus ist dabei selbst dafür zuständig, die Ortsangaben, etwa eine Koordinate, auf die internen Elemente (z. B. Knoten oder Kanten) des Such-Netzwerks abzubilden.

Die *IndividualTransportOptions* geben dabei an, mit welchen IV-Verkehrsmitteln der Benutzer bereit ist, die Haltestelle zu erreichen/verlassen. Im Standardfall ist das ein Fußweg, es kommen aber auch das Fahrrad, PKW und Taxi in Betracht. Beim Fahrrad muss man in den Optionen z. B. der Verbindungsauskunft wählen, ob man das Fahrrad in den ÖV-Fahrzeugen mitnehmen will. Dies zieht dann unter Umständen eine andere Reiseplanung nach sich. Bei Pkw wird zwischen „selber fahren“ und „gefahren werden“ unterschieden. Im ersten Fall muss in der Reiseplanung ein Weg zu einem Parkplatz berücksichtigt werden, wohingegen im zweiten Fall ein Anhalten zum Aussteigen genügt.

7.6.11. AbstractResponseContextStructure

AbstractResponseContextStructure			+Structure	Basisstruktur für Antwortkontext. Hier können Objekte abgelegt werden, die mehrfach auftreten und durch Referenzen auf den Kontext ersetzt werden können.
Locations	Location	0:*	+Location	Modellierung von Ortspunkten (vgl. 7.5.7).
Situations	Situation	0:*	+siri:PtSituationElement	SIRI-Modellierung eines Ereignisses oder einer Störung.

Tabelle 45: Beschreibung der Struktur **AbstractResponseContextStructure**.

7.6.12. LegAttributeStructure

LegAttributeStructure			+Structure (abgeleitet von GeneralAttributeStructure, vgl. 7.4.9)	Definition von Attributen und Hinweisen, die nur auf Teilen einer Verbindung gültig sind.
	FromStopSeqNumber	0:1	xs:positiveInteger	Fahrwegpositionsnummer des Haltepunkts, ab dem das Attribut gültig ist. Falls leer, dann gültig ab Beginn des Fahrwegs.
	ToStopSeqNumber	0:1	xs:positiveInteger	Fahrwegpositionsnummer des Haltepunkts, bis zu dem das Attribut gültig ist. Falls leer, dann gültig bis zum Ende des Fahrwegs.

Tabelle 46: Beschreibung der Struktur **LegAttributeStructure**.

7.7. Trias_FacilitySupport

In der XML-Schema-Definition Trias_FacilitySupport.xsd werden Strukturdefinitionen aus dem SIRI-FM-Dienst zur Verfügung gestellt, die für die Übermittlung von Meldungen zu Infrastruktureinrichtungen und Fahrzeugausstattungen verwendet werden können. Die hier definierten Strukturen haben den Sinn, den Import des SIRI-Schemas an einer Stelle zu kapseln und eine Abstraktionsebene zu schaffen, die eine Erweiterung erlaubt, ohne dass die SIRI-Definitionen verändert werden müssten.

Im Folgenden werden die komplexen Strukturen beschrieben, die in Trias_FacilitySupport definiert sind.

7.7.1. siri:CommonFacilityGroup

Die Gruppe *CommonFacilityGroup* wird in SIRI in der Schemadatei siri_facilities-v1.2.xsd definiert. Sie ist hier nur aus Gründen der Vollständigkeit und leichteren Verständlichkeit wiedergegeben.

siri:CommonFacilityGroup			+Group	Klassifizierung von allgemeinen Ausstattungsmerkmalen und Einrichtungen (nach TPEG pti_table 23).
	FareClassFacility	0:*	unknown / firstClass / secondClass / thirdClass / economyClasses / businessClass	Tarifklassen.
	TicketingFacility	0:*	unknown / ticketMachines / ticketOffice	Einrichtungen zum Fahrscheinkauf.

			 ticketOnDemandMachines ticketSales mobileTicketing ticketCollection centralReservations localTickets nationalTickets internationalTickets	
<i>NuisanceFacility</i>	0:*		unknown smoking noSmoking mobilePhoneUseZone mobilePhoneFreeZone	Aufenthaltsbereiche.
<i>MobilityFacility</i>	0:*		unknown suitableForWheelChairs lowFloor boardingAssistance stepFreeAccess tactilePlatformEdges onboardAssistance unaccompaniedMinorAssistance audioInformation visualInformation displaysForVisuallyImpaired audioForHearingImpaired	Ausstattungsmerkmale für Mobilitätseingeschränkte.
<i>PassengerInformationFacility</i>	0:*		unknown nextStopIndicator stopAnnouncements passengerInformationDisplay audioInformation visualInformation tactilePlatformEdges tactileInformation walkingGuidance journeyPlanning lostFound informationDesk interactiveKiosk-Display printedPublicNotice	Einrichtungen zur Fahrgastinformation.
<i>PassengerCommsFacility</i>	0:*		unknown faccomms_1 passengerWifi	Kommunikationseinrichtungen für Fahrgäste.

			telephone audioServices videoServices businessServices internet postoffice letterbox	
	RefreshmentFacility	0:*	unknown restaurantService snacksService trolley bar foodNotAvailable beveragesNot Available bistro foodVendingMachine beverageVendingMachine	Versorgung mit Erfrischungen, Speisen und Getränken.
	AccessFacility	0:*	unknown lift escalator travelator ramp stairs shuttle narrowEntrance barrier palletAccess_ owFloor validator	Zugangsmerkmale zu Haltestellen oder Fahrzeugen.
	SanitaryFacility	0:*	unknown toilet noToilet shower wheelchairAccessToilet babyChange	Sanitäre Einrichtungen.
	LuggageFacility	0:*	unknown bikeCarriage baggageStorage leftLuggage porterage baggageTrolleys	Einrichtungen zur Gepäckbeförderung oder -aufbewahrung.

Tabelle 47: Beschreibung der Gruppe *siri:CommonFacilityGroup*.

7.7.2. siri:StopFacilityGroup

Die Gruppe *StopFacilityGroup* wird in SIRI in der Schemadatei *siri_facilities-v1.2.xsd* definiert. Sie ist hier nur aus Gründen der Vollständigkeit und leichteren Verständlichkeit wiedergegeben.

<i>siri:StopFacilityGroup</i>		+Group	Klassifizierung von Ausstattungsmerkmalen und Einrichtungen an Haltestellen (nach TPEG <i>pti_table 23</i>).	
<i>CommonFacilityGroup</i>	:::	0:*	<i>siri:CommonFacilityGroup</i>	Allgemeine Ausstattungsmerkmale. Vgl. 7.7.1.
	<i>AssistanceFacility</i>	0:*	unknown police firstAid sosPoint specificAssistance unaccompaniedMinorAssistance	Einrichtungen für Hilfesuchende.

			<i>nce</i> <i>boardingAssistance</i>	
	<i>HireFacility</i>	0:*	<i>unknown</i> <i>carHire</i> <i>motorCycleHire</i> <i>cycleHire</i> <i>taxi</i> <i>recreationDeviceHire</i>	Vermietungs- und Leihangebote.

Tabelle 48: Beschreibung der Gruppe **siri:StopFacilityGroup**.

7.7.3. siri:ServiceFacilityGroup

Die Gruppe *ServiceFacilityGroup* wird in SIRI in der Schemadatei *siri_facilities-v1.2.xsd* definiert. Sie ist hier nur aus Gründen der Vollständigkeit und leichteren Verständlichkeit wiedergegeben.

siri:ServiceFacilityGroup			+Group	Klassifizierung von Ausstattungsmerkmalen und Einrichtungen in Fahrzeugen (nach TPEG pti_table 23).
<i>CommonFacilityGroup</i>	:::	0:*	<i>siri:CommonFacilityGroup</i>	Allgemeine Ausstattungsmerkmale. Vgl. 7.7.1.
	<i>AccommodationFacility</i>	0:*	<i>unknown</i> <i>sleeper</i> <i>couchette</i> <i>specialSeating</i> <i>freeSeating</i> <i>recliningSeats</i> <i>babyCompartment</i> <i>familyCarriage</i>	Abteilarten.

Tabelle 49: Beschreibung der Gruppe **siri:ServiceFacilityGroup**.

7.7.4. siri:AllFacilitiesGroup

Die Gruppe *AllFacilitiesGroup* wird in SIRI in der Schemadatei *siri_facilities-v1.2.xsd* definiert. Sie ist hier nur aus Gründen der Vollständigkeit und leichteren Verständlichkeit wiedergegeben.

siri:AllFacilitiesGroup			+Group	Umfassende Gruppe mit allen Klassifizierungen von Ausstattungsmerkmalen und Einrichtungen (nach TPEG pti_table 23).
<i>ServiceFacilityGroup</i>	:::	0:*	<i>siri:ServiceFacilityGroup</i>	Ausstattungsmerkmale von Fahrzeugen. Vgl.7.7.3.
	<i>AssistanceFacility</i>	0:*	<i>unknown</i> <i>police</i> <i>firstAid</i> <i>sosPoint</i> <i>specificAssistance</i> <i>unaccompaniedMinorAssistance</i> <i>boardingAssistance</i>	Einrichtungen für Hilfesuchende.

	<i>HireFacility</i>	0:*	<i>unknown</i> <i>carHire</i> <i>motorCycleHire</i> <i>cycleHire</i> <i>taxi</i> <i>recreationDeviceHire</i>	Vermietungs- und Leihangebote.
--	---------------------	-----	---	--------------------------------

Tabelle 50: Beschreibung der Gruppe **siri:AllFacilitiesGroup**.

7.8. Trias_SituationSupport

In der XML-Schema-Definition Trias_SituationSupport.xsd werden Strukturdefinitionen aus dem SIRI-SX-Dienst zur Verfügung gestellt, die für die Übermittlung von Störungs- und Ereignismeldungen verwendet werden können. Die hier definierten Strukturen haben den Sinn, den Import des SIRI-Schemas an einer Stelle zu kapseln und eine Abstraktionsebene zu schaffen, die eine Erweiterung erlaubt, ohne dass die SIRI-Definitionen verändert werden müssten.

Im Folgenden werden die komplexen Strukturen beschrieben, die in Trias_SituationSupport definiert sind.

7.8.1. SituationsStructure

SituationsStructure		+Structure	Container für die strukturierte Beschreibung einer Situation im ÖV oder im Straßenverkehr, wie z. B. einer Störung im ÖV oder auf der Straße oder eines Ereignisses mit Auswirkungen auf das Verkehrsgeschehen.	
	<i>PtSituation</i>	0:*	<i>siri:PtSituationElement</i>	Kapselung der SIRI-Strukturdefinition für ÖV-Ereignisse, vgl. (CEN, TS 15531 Part 5).
	<i>RoadSituation</i>	0:*	<i>siri:RoadSituationElement</i>	Kapselung der SIRI-Strukturdefinition für IV-Ereignisse, vgl. (CEN, TS 15531 Part 5).

Tabelle 51: Beschreibung der Struktur **SituationsStructure**.

7.8.2. SituationFullRefStructure

SituationFullRefStructure		+Structure (abgeleitet von <i>siri:SituationFullRefStructure</i>)	Referenz auf eine Situationsbeschreibung.	
<i>SituationFullIdentity</i>	<i>VersionCountryRef</i>	0:1	<i>ifopt:CountryRef</i>	Referenziert das Land, um ggf. die <i>ParticipantRef</i> eindeutig zu machen.
	ParticipantRef	1:1	<i>ParticipantRef</i>	Eindeutige ID des Schnittstellenpartners (vgl. 5.12). Stellt Namensraum für die ID der Situation bereit.
	SituationNumber	1:1	<i>EntryQualifier</i>	Eindeutige ID der Situation.
<i>SituationUpdateIdentity</i>	<i>VersionCountryRef</i>	0:1	<i>ifopt:CountryRef</i>	Referenziert das Land, um ggf. die <i>ParticipantRef</i> eindeutig zu machen.
	<i>UpdateParticipantRef</i>	0:1	<i>ParticipantRef</i>	Eindeutige ID des Schnittstellenpartners (vgl. 5.12). Stellt Namensraum für die ID der Situation bereit.
	<i>Version</i>	0:1	<i>SituationVersion</i>	Versionsnummer des Updates zur Situation. Kann bei der Erstmeldung entfallen.

Tabelle 52: Beschreibung der Struktur **SituationFullRefStructure**.

7.9. Trias_RequestSupport

In der XML-Schema-Definition Trias_RequestSupport.xsd werden eine Reihe von Basistypen und -strukturen definiert, die die Aufgabe haben, die SIRI-Nachrichtenaustauschverfahren für die TRIAS-Dienste nutzbar zu machen.

7.9.1. Einfache Typen

Folgende einfache Typen (simple types) werden definiert:

<i>DataVersionType</i>	<i>xs:NMTOKEN</i>	Datentyp zur Angabe der Datenversion.
<i>CalcTimeType</i>	<i>xs:integer</i>	Datentyp für die Rechenzeit in Millisekunden.
<i>SignatureType</i>	<i>xs:string</i>	Datentyp für Signaturen.
<i>CertificateIdType</i>	<i>xs:NMTOKEN</i>	Datentyp für Zertifikat-IDs.

Tabelle 53: Liste der einfachen Typdefinitionen in Trias_RequestSupport.xsd.

Im Folgenden werden die komplexen Strukturen beschrieben, die in Trias_RequestSupport definiert sind.

7.9.2. AbstractTriasServiceRequestStructure

AbstractTriasServiceRequestStructure			+Structure	Basisstruktur für alle direkten Anfragen (ohne Abonnement)
<i>siri:ContextualisedRequest</i>	<i>ServiceRequestContext</i>	0:1	<i>+siri:ServiceRequestContext</i>	Allgemeine Nachrichteneigenschaften, die üblicherweise durch die Konfiguration bekannt sind und nicht je Anfrage ausgetauscht werden müssen. Siehe auch (GEN, TS 15531, Part 2), Kap.6.1.2.
	RequestTimestamp	1:1	<i>xs:dateTime</i>	Zeitstempel der Anfrage.
<i>RequestorEndpoint</i>	<i>Address</i>	0:1	<i>siri:EndpointAddress</i>	Adresse, an die die Antwort gesendet werden soll. Kann auch mittels RequestorRef aus der Konfiguration ermittelt werden.
	RequestorRef	1:1	<i>→siri:ParticipantCode</i>	ID des Anfragers.
	<i>MessageIdentifier</i>	0:1	<i>siri:MessageQualifier</i>	Beliebige, eindeutige ID, mit der diese Nachricht referenziert werden kann.
<i>ServiceRequestContext</i>	<i>DataVersion</i>	0:1	<i>DataVersion</i>	Datenversion, die vom Server bei der Bearbeitung benutzt werden soll.
	<i>Language</i>	0:1	<i>xs:language</i>	Bevorzugte Sprache, in der Texte in der Antwort zurückgegeben werden sollen.
<i>MessageIntegrityProperties</i>	<i>Signature</i>	0:1	<i>Signature</i>	Signatur der Nachricht.
	<i>CertificateId</i>	0:1	<i>CertificateId</i>	Zertifikat-ID für die Überprüfung der Nachricht.
<i>ServiceRequestContext</i>	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 54: Beschreibung der Struktur **AbstractTriasServiceRequestStructure**.

7.9.3. AbstractTriasSubscriptionRequestStructure

AbstractTriasSubscriptionRequestStructure			+Structure	Basisstruktur für alle Anfragen zur Einrichtung eines Abonnements.
<i>siri:</i>	RequestTimestamp	1:1	<i>xs:dateTime</i>	Zeitstempel der Anfrage.

<i>ri:AbstractSubscriptionRequest</i>				
<i>RequestorEndpoint</i>	<i>Address</i>	0:1	<i>si-ri:EndpointAddress</i>	Adresse, an die die Antwort gesendet werden soll. Kann auch mittels RequestorRef aus der Konfiguration ermittelt werden.
	RequestorRef	1:1	→si-ri:ParticipantCode	ID des Anfragers.
	<i>MessageIdentifier</i>	0:1	<i>si-ri:MessageQualifier</i>	Beliebige, eindeutige ID, mit der diese Nachricht referenziert werden kann.
<i>SubscriberEndpoint</i>	<i>ConsumerAddress</i>	0:1	<i>si-ri:EndpointAddress</i>	Adresse, an die die im Rahmen des Abonnements entstandenen Nachrichten gesendet werden sollen. Diese Angabe kann entfallen, falls ConsumerAddress identisch zu RequestorEndpoint:Address.
	<i>SubscriptionFilterIdentifier</i>	0:1	<i>xs:NMTOKEN</i>	ID eines vorkonfigurierten Filters, dem die Nachrichten für dieses Abonnement unterzogen werden sollen.
<i>si-ri:AbstractSubscriptionRequest</i>	<i>SubscriptionContext</i>	0:1	<i>si-ri:SubscriptionContext</i>	Allgemeine Abbonemenseigenschaften, die üblicherweise durch die Konfiguration bekannt sind und nicht explizit angegeben werden müssen. Siehe auch (CEN, TS 15531, Part 2), Kap.7.1.1.2.
<i>SubscriptionRequestContext</i>	<i>DataVersion</i>	0:1	<i>DataVersion</i>	Datenversion, die vom Server bei der Bearbeitung benutzt werden soll.
	<i>Language</i>	0:1	<i>xs:language</i>	Bevorzugte Sprache, in der Texte in der Antwort zurückgegeben werden sollen.
<i>MessageIntegrityProperties</i>	<i>Signature</i>	0:1	<i>Signature</i>	Signatur der Nachricht.
	<i>CertificateId</i>	0:1	<i>CertificateId</i>	Zertifikat-ID für die Überprüfung der Nachricht.
<i>SubscriptionRequestContext</i>	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 55: Beschreibung der Struktur **AbstractTriasSubscriptionRequestStructure**.

7.9.4. AbstractTriasResponseStructure

AbstractTriasResponseStructure		+Structure	Basisstruktur für alle Antworten.	
<i>si-ri:ProducerResponse</i>	RequestTimestamp	1:1	<i>xs:dateTime</i>	Zeitstempel der Antwort.
<i>si-ri:ProducerResponseEndpoint</i>	<i>ProducerRef</i>	1:1	→si-ri:ParticipantCode	ID des antwortenden Teilnehmers.
	<i>Address</i>	0:1	<i>si-ri:EndpointAddress</i>	Adresse, an die eine etwaige Empfangsbestätigung für den Erhalt der Nachricht gesendet werden soll. Kann auch mittels RequestorRef aus der Konfiguration ermittelt werden.
	<i>ResponseMessageIdentifier</i>	0:1	<i>si-ri:MessageQualifier</i>	Beliebige, eindeutige ID, mit der diese Nachricht referenziert werden kann.
	<i>RequestMessageRef</i>	0:1	→si-ri:MessageQu	Referenz auf die Anfragenachricht, die diese Antwortnachricht ausgelöst hat.

			<i>alifier</i>	
<i>ResponseStatus</i>	<i>Status</i>	0:1	<i>xs:boolean</i>	Indikator, ob die gesamte Anfrage komplett erfolgreich bearbeitet werden konnte. Default ist <i>true</i> .
	<i>ErrorCondition</i>	0:1	<i>siri:ErrorCondition</i>	SIRI-Fehlerzustände, die die Bearbeitung der Anfrage als Ganzes betreffen. Siehe auch (CEN, TS 15531, Part 2), Kap. 5.7.
	<i>MoreData</i>	0:1	<i>xs:boolean</i>	Indikator, ob noch weitere Aktualisierungen vorliegen, die abgerufen werden könnten. Default ist <i>false</i> .
<i>ServiceResponseContext</i>	<i>DataVersion</i>	0:1	<i>DataVersion</i>	Datenversion, die vom Server bei der Bearbeitung benutzt wurde.
	<i>Language</i>	0:1	<i>xs:language</i>	Sprache, in der Texte in der Antwort zurückgegeben wurden.
	<i>CalcTime</i>	0:1	<i>CalcTime</i>	Rechenzeit für die Bearbeitung der Anfrage.
<i>MessageIntegrityProperties</i>	<i>Signature</i>	0:1	<i>Signature</i>	Signatur der Nachricht.
	<i>CertificateId</i>	0:1	<i>CertificateId</i>	Zertifikat-ID für die Überprüfung der Nachricht.
<i>ServiceResponseContext</i>	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 56: Beschreibung der Struktur **AbstractTriasResponseStructure**.

7.10. Trias_FaresSupport

In der XML-Schema-Definition *Trias_FaresSupport.xsd* werden eine Reihe von Basistypen und -strukturen definiert, die für die Tarifberechnung einer Reise verwendet werden.

7.10.1. Einfache Typen

Folgende einfache Typen (simple types) werden definiert:

<i>FaresAuthorityCodeType</i>	<i>xs:NMTOKEN</i>	Code für ein Tarifgebiet oder einen Unternehmensstarif, z. B. „VVS“ oder „DBAG“.
<i>FareZoneCodeType</i>	<i>xs:NMTOKEN</i>	Code für eine Tarifzone in einem Tarifgebiet oder einem Unternehmenstarif.
<i>TicketCodeType</i>	<i>xs:NMTOKEN</i>	Code für ein Ticket. Eindeutig innerhalb eines Tarifgebiets oder eines Unternehmenstarifs.
<i>TravellerCardCodeType</i>	<i>xs:NMTOKEN</i>	Code für eine Vielfahrerkarte, z. B. „BahnCard50“ oder „BahnCard25First“.
<i>TravelClassEnumeration</i>	<i>all first second third business economy</i>	Reiseklasse.
<i>VatRateEnumeration</i>	<i>no full half mixed unknown</i>	Aufzählung möglicher Mehrwertsteuersätze.
<i>PassengerCategoryEnumeration</i>	<i>Adult Child Senior Youth Disabled</i>	Kategorisierung von Fahrgästen in tariflicher Hinsicht.

Tabelle 57: Liste der einfachen Typdefinitionen in *Trias_FaresSupport.xsd*.

Damit die Codes von Tarifgebieten, Tarifzonen etc. systemübergreifend verstanden werden können, sind bestimmte Vereinbarungen zu treffen. Diese sind in Kapitel 5 beschrieben.

Im Folgenden werden die komplexen Strukturen beschrieben, die in Trias_FaresSupport definiert sind.

7.10.2. FareZoneListInAreaStructure

FareZoneListInAreaStructure		+Structure	Liste von Tarifzonen bezogen auf ein Tarifgebiet.
	FaresAuthorityRef	1:1	→FaresAuthorityCode Code für ein Tarifgebiet oder einen Unternehmens-tarif (vgl. 7.10.1).
	FareZoneRef	1:*	→FareZoneCode Code für eine Tarifzone (vgl. 7.10.1).

Tabelle 58: Beschreibung der Struktur **FareZoneListInAreaStructure**.

7.10.3. BookingInfoStructure

BookingInfoStructure		+Structure	Beschreibung einer Buchungsmöglichkeit für das angefragte Objekt.
	<i>BookingAgencyName</i>	0:1	+InternationalText Name der Buchungsagentur (Vertragspartner).
	<i>BookingUrl</i>	0:1	xs:anyURI URL für Online-Buchung.
	<i>InfoUrl</i>	0:1	xs:anyURI URL zu Informationsseiten.
	<i>PhoneNumber</i>	0:1	PhoneNumber Telefonnummer zum Buchen (vgl. 7.2.1).
	<i>BookingDeadline</i>	0:1	xs:duration Minimale Vorlaufzeit für Buchungen, bevor die Fahrt beginnt.
	<i>Extension</i>	0:1	xs:anyType Erweiterungen.

Tabelle 59: Beschreibung der Struktur **BookingInfoStructure**.

7.10.4. TicketStructure

TicketStructure		+Structure	Modellierung eines Fahrscheins und zugehöriger Informationen.
	TicketId	1:1	→TicketCode Eindeutige Ticket-ID (vgl. 7.10.1).
	TicketName	1:1	xs:string Name/Bezeichnung des Tickets.
	FaresAuthorityRef	1:1	→FaresAuthorityCode Code für ein Tarifgebiet oder einen Unternehmens-tarif (vgl. 7.10.1).
Ti- cketPri- ce	<i>Price</i>	0:1	xs:decimal Ticketpreis als Dezimalzahl.
	<i>NetPrice</i>	0:1	xs:decimal Netto-Ticketpreis als Dezimalzahl für Abrechnungszwecke.
	<i>Currency</i>	0:1	xs:NMTOKEN Währungscode nach ISO 4217, z. B. „EUR“ oder „GBP“.
	<i>VatRate</i>	0:1	VatRateEnumeration Mehrwertsteuersatz (vgl. 7.10.1). Voreinstellung ist <i>unknown</i> .
Ticket- Validity	<i>TravelClass</i>	0:1	TravelClassEnumeration Reiseklasse, für die das Ticket gültig ist (vgl. 7.10.1).
	<i>RequiredCard</i>	0:*	→TravellerCardCode Eine oder mehrere Vielfahrerkarten, die nötig sind, um dieses Ticket erwerben oder nutzen zu dürfen (vgl. 7.10.1).
	<i>ValidFor</i>	0:*	PassengerCategoryEnumeration Personengruppen, die dieses Ticket nutzen dürfen (vgl. 7.10.1).
Ticket- Booking	<i>BookingInfo</i>	0:*	+BookingInfo Beschreibung der Buchungsmöglichkeiten (vgl. 7.10.3).
	<i>Extension</i>	0:1	xs:anyType Erweiterungen.

Tabelle 60: Beschreibung der Struktur **TicketStructure**.

7.10.5. TripFaresResultStructure

TripFaresResultStructure			+Structure	Fasst die Ergebnisdaten für die Tarifauskunft zu einer Verbindung (oder zu Teilen einer Verbindung) zusammen.
	<i>ErrorMessage</i>	0:*	+ErrorMessage	Fehlermeldungen bezogen auf diese Tarifauskunft. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. auch 7.4.2.
<i>TripLeg-Range</i>	<i>FromTripLegIdRef</i>	0:1	xs:NMTOKEN	Referenz auf einen Teilweg der Verbindung als Beginn der Gültigkeit dieser Tarifauskunft.
	<i>ToTripLegIdRef</i>	0:1	xs:NMTOKEN	Referenz auf einen Teilweg der Verbindung als Ende der Gültigkeit dieser Tarifauskunft.
	<i>PassedZones</i>	0:1	+FareZoneListInArea	Die durchfahrenen Tarifzonen auf diesem Abschnitt der Verbindung (vgl. 7.10.2).
	<i>Ticket</i>	0:*	+Ticket	Fahrscheine, die auf diesem Abschnitt der Verbindung gültig sind (vgl. 7.10.4).
	<i>StaticInfoURL</i>	0:*	xs:anyURI	URL zu Informationsseiten.

Tabelle 61: Beschreibung der Struktur **TripFaresResultStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

FARES_OUTOFAREA	Der in der Verbindungsauskunft gefundene Weg verlässt das Tarifgebiet.
FARES_JOURNEYNOTPERMITTED	Ein in der Verbindungsauskunft benutztes Verkehrsmittel ist für den Tarif nicht zulässig.
FARES_ADDITIONALCHARGES	Zusätzliche Gebühren sind voraussichtlich zu entrichten (z. B. Mautzuschläge oder Reservierungsgebühren).
FARES_ADDITIONALTICKETS	Zusätzliche Fahrscheine sind notwendig, da nicht für alle Verkehrsmittel ein passender Fahrschein ermittelt werden konnte.
FARES_ROUTENOTFEASIBLE	Es kann kein Fahrschein ermittelt werden, weil die Route der Verbindungsauskunft nicht in Einklang mit den Tarifregeln steht (z. B. wegen Rundreisen, Stichfahrten oder Überschreitung der zulässigen Gesamtdauer).

Tabelle 62: Liste der Fehlerzustände in *TripFaresResult*.

7.10.6. FaresPassengerStructure

FaresPassengerStructure			+Structure	Profil eines Reisenden für die Tarifermittlung.
	a	Age	xs:nonNegativeInteger	Alter des Reisenden.
	b	PassengerCategory		
		<i>TravellerCard</i>	0:*	->TravellerCardCode

Tabelle 63: Beschreibung der Struktur **FaresPassengerStructure**.

7.10.7. FaresParamStructure

<i>FaresParamStructure</i>		+Structure	Parameter für die Tarifiermittlung.
<i>Fares-DataFilter</i>	<i>FareAuthorityFilter</i>	0:*	→ <i>FaresAuthorityCode</i> Codes für Tarifgebiete oder Unternehmenstarife, die berücksichtigt werden sollen (vgl. 7.10.1).
	<i>PassengerCategory</i>	0:*	<i>PassengerCategoryEnumeration</i> Personengruppen, die berücksichtigt werden sollen (vgl. 7.10.1).
	<i>TravelClass</i>	0:1	<i>TravelClassEnumeration</i> Reiseklasse, die berücksichtigt werden soll (vgl. 7.10.1).
	<i>Traveller</i>	0:*	+ <i>FaresPassenger</i> Anzahl Reisende, für die die Tarifiermittlung ausgeführt werden soll (vgl. 7.10.6).

Tabelle 64: Beschreibung der Struktur **FaresParamStructure**.

8. Ortsinformationsdienst

8.1. Beschreibung

Der Ortsinformationsdienst umfasst vier Funktionalitäten, die in der ursprünglichen Dienstbeschreibung noch als getrennte Dienste aufgefasst wurden

- Start-/Ziel-Identifikation bei Eingabe einer Zeichenkette,
- Objektinformationsdienst zum Abrufen aller Ortsobjekte,
- Geografischer Kontextdienst zum Abrufen von Ortsobjekten in einem Kartenausschnitt,
- Koordinaten-zu-Adressdienst zum Abrufen der nächsten Adresse für gegebene Koordinaten.

Diese Funktionalitäten konnten durch Abstraktion in einem einzigen Dienst gebündelt werden. Durch diese Verallgemeinerung sind auch weitere Einsatzmöglichkeiten des Dienstes entstanden. Beispielsweise (aber nicht abschließend):

- Abruf der nächsten Haltestelle(n) für gegebene Koordinaten.
- Ortsabhängiges Patternmatching einer Zeichenkette durch Berücksichtigung von gleichzeitig übergebenen Koordinaten.

In der XML-Schema-Definition *Trias_Locations.xsd* werden Datentypen und Strukturen definiert, die für den Ortsinformationsdienst verwendet werden.

8.2. Einfache Datentypen

Folgende einfache Typen (simple types) werden definiert:

<i>LocationTypeEnumeration</i>	<i>stop address poi coord locality</i>	Typ eines Ortsobjektes.
<i>LocationUsageEnumeration</i>	<i>origin destination via</i>	Verwendungszweck eines Ortsobjektes.

Tabelle 65: Liste der einfachen Typdefinitionen in *Trias_Locations.xsd*.

8.3. Anfragestrukturen

Ortsobjekte werden mittels eines Elements **LocationInformationRequest** vom Typ *LocationInformationRequestStructure* angefordert.

8.3.1. LocationInformationRequestStructure

LocationInformationRequestStructure			+Structure	Fasst die Daten Ortsobjektanfrage zusammen.
	a	InitialInput	-1:1	+InitialLocationInput Eingabedaten für eine initiale Ortsinformationsanfrage. Vgl. 8.3.2.
	b	LocationRef		+LocationRef Referenz auf ein Ortsobjekt, welches weiter verfeinert werden soll. Bei hierarchisch organisierten Ortsobjekten kann es sinnvoll sein, die Ortsidentifikation in mehreren Stufen durchzuführen. Dabei erzeugt eine initiale Anfrage an den Ortsinformationsdienst eine Menge von „groben“ Ortsobjekten (z. B. Straßen), die ggf. noch weiter verfeinert werden müssen (z. B. zu Hausnummernbereichen). Die „groben“ Objekte werden dem Benutzer gezeigt und er wählt eines davon aus. Um dieses nun weiter zu verfeinern, wird seine Referenz hier dem Ortsinformationsdienst übergeben. Vgl. 7.5.8.
	<i>Restrictions</i>		0:1	+LocationParameter Weitere Anfrageparameter. Vgl. 8.3.7.
	<i>Extension</i>		0:1	xs:anyType Erweiterungen.

Tabelle 66: Beschreibung der Struktur **LocationInformationRequestStructure**.

8.3.2. InitialLocationInputStructure

InitialLocationInputStructure			+Structure	Fasst die Anfrageparameter zusammen, die eine initiale Suche nach Ortsobjekten benötigt.
	<i>LocationName</i>	0:1	xs:string	Eingabezeichenkette, die als Muster für die zu findenden Ortsobjekte dienen soll. Falls angegeben, sollen Ortsobjekte umso mehr bevorzugt werden, je ähnlicher ihr Name der Zeichenkette ist. Falls gleichzeitig <i>Coordinate</i> angegeben wird, muss der Dienst beide Anforderungen sinnvoll zueinander gewichten.
	<i>GeoPosition</i>	0:1	+GeoPosition	Geografische Position, in dessen Nähe die zu findenden Ortsobjekte liegen sollen. Falls angegeben, sollen solche Ortsobjekte bevorzugt werden, die in der Nähe dieser Geoposition liegen. Falls gleichzeitig <i>LocationName</i> angegeben wird, muss der Dienst beide Anforderungen sinnvoll zueinander gewichten. Vgl. 7.2.3.
	<i>GeoRestriction</i>	0:1	+GeoRestrictions	Geografischer Filter. Falls angegeben, müssen alle gefundenene Ortsobjekte diesem Filter entsprechen. Vgl. 8.3.3.

Tabelle 67: Beschreibung der Struktur **InitialLocationInputStructure**.

8.3.3. GeoRestrictionsStructure

GeoRestrictionsStructure			+Structure	Definiert einen geografischen Filter.
	a	Circle	-1:1	+GeoCircle Der Filter wird durch einen Kreis definiert. Vgl. 8.3.4.
	b	Rectangle		+GeoRectangle Der Filter wird durch ein Rechteck definiert. Vgl. 8.3.5.
	c	Area		+GeoArea Der Filter wird durch ein Polygon definiert. Vgl. 8.3.6.

Tabelle 68: Beschreibung der Struktur **GeoRestrictionsStructure**.

8.3.4. GeoCircleStructure

GeoCircleStructure			+Structure	Definiert einen geografischen Kreis.
	Center	1:1	+GeoPosition	Zentrum des Kreises. Vgl. 7.2.3.
	Radius	1:1	Distance	Radius des Kreises in Metern.

Tabelle 69: Beschreibung der Struktur **GeoCircleStructure**.

8.3.5. GeoRectangleStructure

GeoRectangleStructure			+Structure	Definiert ein geografisches Rechteck.
	UpperLeft	1:1	+GeoPosition	Linke obere Ecke des Rechtecks. Vgl. 7.2.3.
	LowerRight	1:1	+GeoPosition	Rechte untere Ecke des Rechtecks. Vgl. 7.2.3.

Tabelle 70: Beschreibung der Struktur **GeoRectangleStructure**.

8.3.6. GeoAreaStructure

GeoAreaStructure			+Structure	Definiert ein geografisches Polygon.
	PolylinePoint	3:*	+GeoPosition	Eckpunkte des Polygons. Vgl. 7.2.3.

Tabelle 71: Beschreibung der Struktur **GeoAreaStructure**.

8.3.7. LocationParamStructure

LocationParamStructure			+Structure	Fasst Anfrageparameter zusammen, die im Ortsinformationsdienst verwendet werden.
<i>Location DataFilter</i>	<i>Type</i>	0:*	<i>stop address poi coord locality</i>	Erlaubte Ortsobjektstypen. Falls welche angegeben werden, dürfen nur Ortsobjekte zurückgegeben werden, die von einem der angegebenen Typen sind. Falls keine angegeben werden, sind alle Objekttypen erlaubt.
	<i>Usage</i>	0:1	<i>origin destination via</i>	Verwendung des Ortsobjektes. Falls angegeben, teilt dies dem Dienst mit, als was das gesuchte Ortsobjekt verwendet werden soll. Der Ortsinformationsdienst darf dann nur Objekte zurückgeben, die für die angegebene Verwendung freigegeben sind.
	<i>PtModes</i>	0:1	+PtModeFilter	Erlaubte Verkehrsmittel. Falls angegeben, dürfen nur solche Ortsobjekte zurückgegeben werden, an denen Verkehre fahren, die dem Filter entsprechen. Dies schließt automatisch alle Nicht-Haltestellen aus. Vgl. 7.3.5.
	<i>LocalityRef</i>	0:*	→LocalityCode	Erlaubte Lokalitäten. Falls angegeben, dürfen nur solche Ortsobjekte zurückgegeben werden, die mindestens einer der gegebenen Lokalitäten zugeordnet sind. Vgl. 7.5.1.
<i>Location Policy</i>	<i>Language</i>	0:1	<i>xs:language</i>	Bevorzugte Sprache für die zurückgegebenen Texte. Dies muss nicht unbedingt Einfluss auf die Namen der Ortsobjekte haben.
	<i>NumberOfResults</i>	0:1	<i>xs:positiveInteger</i>	Anzahl der maximal zurückzugebenen Ortsobjekte. Der Dienst kann durchaus weniger Objekte zurückgeben, falls sinnvoll oder falls sonst der Dienst überfordert wäre. Falls mehr Objekte die Anfrage erfüllen (z. B. wenn alle Objekte abgerufen werden sollen), kann mit diesem Parameter die Menge der Objekte, die in einem Abruf maximal übertragen werden, begrenzt werden. Ein Ortsinformationsdienst muss in der Lage sein, mindestens 500 Ortsobjekte in einer Antwort zurückzugeben.
	<i>ContinueAt</i>	0:1	<i>xs:nonNegativeInteger</i>	Falls angegeben, weist dieser Parameter den Dienst an, wieviele Objekte in der Rückgabe übersprungen werden sollen. Falls bei einem Abruf von Ortsobjek-

				ten nicht alle passenden Objekte geliefert werden konnten, teilt der Dienst dies in seiner Antwort im Feld <i>ContinueAt</i> mit (vgl. 8.4.1). Um die weiteren Objekte abzurufen, wird die Anfrage an den Ortsinformationsdienst exakt wiederholt, wobei dieser Parameter angegeben wird, indem der Wert aus der letzten Diensteantwort eingefüllt wird.
	<i>IncludePtModes</i>	0:1	<i>xs:boolean</i>	Teilt dem Dienst mit, an Haltestellen die verfügbaren Verkehrsmittel mit zurück zu geben. Default ist <i>false</i> .

Tabelle 72: Beschreibung der Struktur **LocationParamStructure**.

8.4. Antwortstrukturen

Das Ergebnis einer Objektinformationsanfrage wird mittels eines Elements **LocationInformationResponse** vom Typ *LocationInformationResponseStructure* übertragen.

8.4.1. LocationInformationResponseStructure

LocationInformationResponseStructure		+Structure	Fasst die Ergebnisdaten für eine Ortsinformationsanfrage zusammen.	
	<i>ContinueAt</i>	0:1	<i>xs:nonNegativeInteger</i>	In einem Folgeabruf zu überspringende Ortsobjekte. Falls gesetzt, zeigt der Dienst an, dass noch weitere Ortsobjekte zu der Anfrage passen, die in der Antwort nicht enthalten sind. Wird der Abruf wiederholt und dabei der Parameter <i>ContinueAt</i> auf den hier übermittelten Wert gesetzt (vgl. 8.3.7), liefert der Dienst die folgenden Ortsobjekte.
	<i>ErrorMessage</i>	0:*	+ErrorMessage	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
	<i>Location</i>	0:*	+LocationResult	Gefundene Ortsobjektergebnisse. Die Ortsobjekte müssen nach dem Übereinstimmungsgrad mit den Eingabedaten sortiert sein, d.h. das erste ist das am besten passende Objekt. Vgl. 8.4.2.

Tabelle 73: Beschreibung der Struktur **LocationInformationResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

LOCATION_NORESULTS	Zu den Eingabedaten konnten keine Ortsobjekte gefunden werden.
LOCATION_UNSUPPORTEDTYPE	Es wurden nur Objekttypen angefragt, die vom Dienst nicht unterstützt werden.
LOCATION_UNSUPPORTEDCOMBINATION	Die verwendete Kombination von Eingabedaten (Zeichenkette, Koordinaten, Geo-Restriktion) wird vom Dienst nicht unterstützt.
LOCATION_NOREFINEMENT	Das angegebene Ortsobjekt konnte nicht verfeinert werden.
LOCATION_USAGEIGNORED	Der Verwendungszweck wurde ignoriert.
LOCATION_UNSUPPORTEDPTMODES	Der Dienst unterstützt keine Einschränkung der Verkehrsmittel.
LOCATION_UNSUPPORTEDLOCALITY	Der Dienst unterstützt keine Einschränkung durch Lokalitäten.

Tabelle 74: Liste der Fehlerzustände in *LocationInformationResponse*.

8.4.2. LocationResultStructure

LocationResultStructure			+Structure	Resultatstruktur für ein Ortsobjekt.
	Location	1:1	+Location	Eigentliches Ortsobjekt. Vgl. 7.5.7.
	Complete	1:1	xs:boolean	Gibt an, ob das Ortsobjekt schon vollständig ausdifferenziert ist, oder ob es noch verfeinert werden kann.
	Probability	0:1	xs:float	Wahrscheinlichkeit, dass dieses Ortsobjekt dem gesuchten entspricht. Wird mit einem Wert zwischen 0 und 1 angegeben.
	Mode	0:*	+Mode	Auflistung der Verkehrsmittel, die an dem Ortsobjekt verkehren. Sollte nur bei Haltestellen gefüllt sein und nur dann, wenn es in der Anfrage angefordert wurde. Vgl. 7.3.4.

Tabelle 75: Beschreibung der Struktur **LocationResultStructure**.

9. Dienst Verbindungsauskunft

9.1. Beschreibung

Dieser Dienst berechnet intermodale Verbindungen von einem Startpunkt zu einem Zielpunkt. Dabei werden diverse Benutzerpräferenzen berücksichtigt.

9.2. Anfragestrukturen

Eine intermodale Verbindungsauskunft wird mittels eines Elements **TripRequest** vom Typ **TripRequestStructure** angefordert.

9.2.1. TripRequestStructure

TripRequestStructure			+Structure	Fasst die Anfragedaten für eine Verbindungsauskunft zusammen.
	Origin	1:1	+LocationContent	Ortsdaten für den Abfahrtsort. Vgl. 7.6.10.
	Destination	1:1	+LocationContent	Ortsdaten für den Zielort. Vgl. 7.6.10.
	Via	0:*	+Via	Ein oder mehrere Via-Orte. Die angegebenen Via-Orte müssen in der vorgegebenen Reihenfolge erreicht werden. Der Server darf eine Via-Haltestelle durch eine äquivalente Haltestelle ersetzen. Vgl. 7.6.2.
	NotVia	0:*	+NotVia	Haltestellen oder Haltepunkte, die die Verbindung nicht passieren darf. Vgl. 9.2.4.
	NoChangeAt	0:*	+NoChangeAt	Haltestellen oder Haltepunkte, wo die Verbindung keinen Umstieg vorsehen darf. Vgl. 9.2.5.
	Params	0:1	+TripParam	Parameter, die die Suche und Rückgabewerte beeinflussen können. Vgl. 9.2.2.

Tabelle 76: Beschreibung der Struktur **TripRequestStructure**.

9.2.2. TripParamStructure

TripParamStructure			+Structure	Fasst die Anfragedaten für eine Verbindungsauskunft zusammen.
TripDat	PtModeFilter	0:1	+PtModeFilter	Filter nach Verkehrsmitteltypen. Vgl. 7.3.5.

<i>aFilter</i>	<i>LineFilter</i>	0:1	<i>+LineDirectionFilter</i>	Erlaubte Linien (ggf. verfeinert auf Richtungen). Vgl. 7.4.6.	
	<i>OperatorFilter</i>	0:1	<i>+OperatorFilter</i>	Filter nach Verkehrsunternehmen. Vgl.7.4.4.	
<i>BaseTripMobilityFilter</i>	<i>NoSingleStep</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer eine Stufe bewältigen kann. Falls nein, wird dieser Parameter gesetzt.	
	<i>NoStairs</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer eine Treppe bewältigen kann. Falls nein, wird dieser Parameter gesetzt.	
	<i>NoEscalator</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer eine Rolltreppe benutzen kann. Falls nein, wird dieser Parameter gesetzt.	
	<i>NoElevator</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer einen Aufzug benutzen kann. Falls nein, wird dieser Parameter gesetzt.	
	<i>NoRamp</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer eine Rampe bewältigen kann. Falls nein, wird dieser Parameter gesetzt.	
<i>TripMobilityFilter</i>	<i>LevelEntrance</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer beim Ein- und Aussteigen in und aus Fahrzeugen einen ebenen Zugang benötigt. Dazu reicht u.U. auch ein Hublift am Fahrzeug oder am Bahnsteig. Falls der ebene Zugang notwendig ist, wird dieser Parameter gesetzt.	
	<i>BikeTransport</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer ein Fahrrad an Bord der Verkehrsmittel mitnehmen will. Falls ja, wird dieser Parameter gesetzt.	
	<i>WalkSpeed</i>	0:1	<i>OpenPercent</i>	Veränderung der Standardgeschwindigkeit in Prozent. Der Wert 100 stellt den Standard dar. Werte kleiner 100 stellen eine langsamere Geschwindigkeit dar, Werte größer 100 eine schnellere.	
<i>BaseTripPolicy</i>					
	a	<i>NumberOfResults</i>	-0:1	<i>xs:positiveInteger</i>	Anzahl der Verbindungsauskünfte, die der Benutzer mindestens erwartet.
	b	...	-0:1	<i>NumberOfResultsGroup</i>	Angabe der gewünschten Verbindungen vor/nach dem angegebenen Zeitpunkt am Start oder Ziel (vgl. 9.2.3).
		<i>IgnoreRealtimeData</i>	0:1	<i>xs:boolean</i>	Wenn dieser Parameter gesetzt ist, sollen in der Verbindungssuche keine Echtzeitdaten oder Störungsinformationen sondern nur Sollfahrplandaten berücksichtigt werden.
		<i>ImmediateTripStart</i>	0:1	<i>xs:boolean</i>	Wenn dieser Parameter gesetzt ist, soll die zu suchende Verbindung unmittelbar an der angegebenen Startsituation beginnen. Eine Optimierung der Abfahrtszeit am Start nach der Regel „Starte so spät wie möglich, solange nur die gleiche Ankunftszeit am Ziel gewährleistet ist“ ist dann nicht notwendig.
<i>TripPolicy</i>	<i>InterchangeLimit</i>	0:1	<i>xs:positiveInteger</i>	Anzahl der maximal zugelassenen Umsteigevorgänge.	
	<i>AlgorithmType</i>	0:1	<i>fastest</i> / <i>minChanges</i> / <i>leastWalking</i> / <i>leastCost</i>	Art der Zielfunktion, nach der der Algorithmus die Verbindung optimieren soll.	
	<i>ItModesToCover</i>	0:*	<i>IndividualModesEnumeration</i>	Für jeden IV-Typ (vgl. 7.3.1) in dieser Liste soll eine eigene monomodale Verbindung gefunden werden – zusätzlich zu den intermodalen Verbindungen.	
<i>BaseTripContentFilter</i>	<i>IncludeTrackSections</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat TrackSection-Elemente (vgl. 9.3.13) für die detaillierte geographische Beschreibung des Wegs mitausgegeben werden sollen. Default ist <i>false</i> .	
	<i>IncludeLegProjection</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat der detaillierte geographische Verlauf des Wegs als Koordinatenfolge mitausgegeben werden soll. Default ist <i>false</i> .	
	<i>IncludeTurnDescription</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Routenhinweise mit Abbiegeempfehlungen mitausgegeben werden sollen. Default ist <i>false</i> .	

	<i>IncludeAccessibility</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Informationen zur Barrierefreiheit mitausgegeben werden sollen. Default ist <i>false</i> .
<i>TripContentFilter</i>	<i>IncludeIntermediateStops</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat die Zwischenhalte mitausgegeben werden sollen. Default ist <i>false</i> .
	<i>IncludeFares</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Tarifinformationen mitausgegeben werden sollen. Default ist <i>false</i> .
	<i>IncludeOperatingDays</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Informationen zu den Verkehrstagen mitausgegeben werden sollen. Default ist <i>false</i> .
	<i>FaresParam</i>	0:1	<i>+FaresParam</i>	Parameter für die Tarifiermittlung (vgl. 7.10.7).
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 77: Beschreibung der Struktur **TripParamStructure**.

9.2.3. NumberOfResultsGroup

NumberOfResultsGroup			<i>+Group</i>	Festlegung der Anzahl gewünschter Verbindungen vor und nach dem angegebenen Zeitpunkt am Start oder Ziel. Diese Gruppe kann nicht verwendet werden, wenn am Start UND am Ziel ein Zeitpunkt vorgeschrieben ist.
	NumberOfResultsBefore	1:1	<i>xs:positiveInteger</i>	Anzahl gewünschter Verbindungen vor dem angegebenen Zeitpunkt.
	NumberOfResultsAfter	1:1	<i>xs:positiveInteger</i>	Anzahl gewünschter Verbindungen nach dem angegebenen Zeitpunkt.

Tabelle 78: Beschreibung der Gruppe **NumberOfResultsGroup**.

9.2.4. NotViaStructure

NotViaStructure			<i>+Structure</i>	Angaben zu einer Nicht-Via-Bedingung. Diese Art Bedingung verhindert, dass eine Verbindungsauskunft über die angegebene Haltestelle oder Haltepunkt führt.
	a StopPointRef	-1:1	<i>→StopPoint</i>	Referenz auf einen Nicht-Via-Haltepunkt. Vgl. 7.5.1.
	b StopPlaceRef		<i>→StopPlace</i>	Referenz auf eine Nicht-Via-Haltestelle. Vgl. 7.5.1.

Tabelle 79: Beschreibung der Struktur **NotViaStructure**.

9.2.5. NoChangeAtStructure

NoChangeAtStructure			<i>+Structure</i>	Angaben zu einer Nicht-Umsteigen-Bedingung. Diese Art Bedingung verhindert, dass in einer Verbindungsauskunft an der angegebenen Haltestelle oder Haltepunkt umgestiegen werden muss.
	a StopPointRef	-1:1	<i>→StopPoint</i>	Referenz auf einen Haltepunkt. Vgl. 7.5.1.
	b StopPlaceRef		<i>→StopPlace</i>	Referenz auf eine Haltestelle. Vgl. 7.5.1.

Tabelle 80: Beschreibung der Struktur **NoChangeAtStructure**.

9.3. Antwortstrukturen

Das Ergebnis einer intermodalen Verbindungsanfrage wird mittels eines Elements **TripResponse** vom Typ *TripResponseStructure* übertragen.

9.3.1. TripResponseStructure

TripResponseStructure			+Structure	Fasst die Ergebnisdaten für eine intermodale Verbindungsauskunft zusammen.
	ErrorMessage	0:*	+ErrorMessage	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. auch 7.4.2.
	TripResponseContext	0:1	+TripResponseContext	Container für Daten, die in der Antwort mehrfach auftreten und referenziert werden. Vgl. 9.3.2.
	TripResult	0:*	+TripResult	Container für eine Verbindungsauskunft. Vgl. 9.3.3.

Tabelle 81: Beschreibung der Struktur **TripResponseStructure**.

In **ErrorMessage** können folgende Fehlerzustände auftreten:

TRIP_NOTRIPFOUND	Zu den angegebenen Start- und Zielorten, der gewünschten Abfahrts- oder Ankunftszeit sowie unter Berücksichtigung der gegebenen Parameter konnte keine Verbindung gefunden werden.
TRIP_ORIGINUNKNOWN	Der angegebene Ort (Adresse, Haltestelle, ...) für den Start der Verbindung ist unbekannt.
TRIP_DESTINATIONUNKNOWN	Der angegebene Ort (Adresse, Haltestelle, ...) für das Ziel der Verbindung ist unbekannt.
TRIP_VIAUNKNOWN	Einer der angegebenen Via-Punkte ist unbekannt.
TRIP_NOTVIAUNKNOWN	Eine der angegebenen Nicht-Via-Haltestellen ist unbekannt.
TRIP_NOCHANGEATUNKNOWN	Eine der angegebenen Nicht-Umsteigen -Haltestellen ist unbekannt.
TRIP_NOORIGIN	Es wurde kein Startpunkt angegeben.
TRIP_NODESTINATION	Es wurde kein Zielpunkt angegeben.
TRIP_ORIGINDESTINATIONIDENTICAL	Start und Ziel sind gleich.
TRIP_DATETIMEERROR	Datum und/oder Uhrzeit sind unverständlich.
TRIP_DEPARTUREAFTERARRIVAL	Die gewünschte Abfahrtszeit an allen Startpunkten liegt nach der gewünschten Ankunftszeit an allen Zielpunkten.
TRIP_DATEOUTOFFRANGE	Für das angefragte Datum liegen keine Fahrplandaten vor.

Tabelle 82: Liste der Fehlerzustände in **TripResponse**.

9.3.2. TripResponseContextStructure

TripResponseContextStructure	+Structure (abgeleitet von AbstractResponseContextStructure)	Container für Daten, die in der Antwort mehrfach auftreten und referenziert werden. Vgl. 7.6.11.
-------------------------------------	---	--

Tabelle 83: Beschreibung der Struktur **TripResponseContextStructure**.

9.3.3. TripResultStructure

TripResultStructure			+Structure	Fasst die Ergebnisdaten für eine einzelne intermodale Verbindungsauskunft zusammen.
	ResultId	1:1	xs:NMTOKEN	ID des Resultats für spätere Referenzierung bzw. für

				Debug-Zwecke.
	<i>ErrorMessage</i>	0:*	+ <i>ErrorMessage</i>	Fehlermeldungen bezogen auf dieses Verbindungsergebnis. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. auch 7.4.2.
	<i>Trip</i>	1:1	+ <i>Trip</i>	Daten zu einer intermodalen Verbindung. Vgl. 9.3.4.
	<i>TripFares</i>	0:*	+ <i>TripFaresResult</i>	Ticket- und Fahrpreisinformationen zur Verbindung als Ganzes oder zu Teilen der Verbindung (vgl. 7.10.5).

Tabelle 84: Beschreibung der Struktur **TripResultStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

TRIP_ORIGINEQUIVALENT	Die gewünschte Starthaltestelle wurde durch eine äquivalente Haltestelle ersetzt.
TRIP_DESTINATIONEQUIVALENT	Die gewünschte Zielhaltestelle wurde durch eine äquivalente Haltestelle ersetzt.
TRIP_VIAEQUIVALENT	Eine gewünschte Via-Haltestelle wurde durch eine äquivalente Haltestelle ersetzt.
TRIP_REALTIMEINCOMPLETE	Für mindestens ein Verkehrsmittel in dieser Verbindung liegen keine Echtzeitdaten vor.
TRIP_ITTIMEEXTENDED	Die vorgegebene maximale Zeit im Individualverkehr (meist Fußweg oder Fahrrad) wurde vom System verlängert, da ansonsten keine Verbindung gefunden werden kann.
TRIP_ITMODECHANGED	Das vorgegebene Individualverkehrsmittel wurde vom System ersetzt, da ansonsten keine Verbindung gefunden werden kann. Üblicherweise ist dies ein Wechsel von Fußweg auf Taxi.
TRIP_INCONVENIENTWAITING	Die Verbindung enthält eine lange Wartezeit.

Tabelle 85: Liste der Fehlerzustände in **TripResult**.

9.3.4. TripStructure

<i>TripStructure</i>			+ <i>Structure</i>	Daten zu einer einzelnen intermodalen Verbindung.
	<i>TripId</i>	1:1	<i>xs:NMTOKEN</i>	ID der Verbindung für spätere Referenzierung bzw. für Debug-Zwecke.
	<i>Duration</i>	1:1	<i>xs:duration</i>	Gesamtdauer der Verbindung.
	<i>StartTime</i>	1:1	<i>xs:dateTime</i>	Startzeitpunkt der Verbindung.
	<i>EndTime</i>	1:1	<i>xs:dateTime</i>	Endzeitpunkt der Verbindung.
	<i>Interchanges</i>	1:1	<i>xs:nonNegativeInteger</i>	Anzahl der notwendigen Umsteigevorgänge.
	<i>Distance</i>	0:1	<i>Distance</i>	Gesamtdistanz der Verbindung als Länge des zurückzulegenden Weges.
	<i>TripLeg</i>	1:*	+ <i>TripLeg</i>	Teilweg/e dieser Verbindung. Vgl. 9.3.5.
<i>OperatingDays</i>	<i>OperatingDays</i>	0:1	+ <i>OperatingDays</i>	Verkehrstage für diese Verbindung. Vgl. 7.4.7.
	<i>OperatingDaysDescription</i>	0:1	+ <i>InternationalText</i>	Menschenlesbare Beschreibung der Verkehrstage, z. B. „Montag bis Freitag“ oder „Sonn- und Feiertag“.
	<i>SituationFullRef</i>	0:*	+ <i>SituationFullRef</i>	Verweis auf eine Störungsnachricht. Diese Nachricht kann im <i>TripResponseContext</i> (vgl.9.3.2) zu finden sein oder auf anderem Wege bekannt gemacht werden. Vgl. 7.8.2.
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 86: Beschreibung der Struktur **TripStructure**

9.3.5. TripLegStructure

<i>TripLegStructure</i>			+Structure	Teilweg zu einer Verbindung.
	LegId	1:1	<i>xs:NMTOKEN</i>	ID des Teilwegs dieser Verbindung für spätere Referenzierung. Eindeutig innerhalb <i>TripResult</i> .
	a TimedLeg	-1:1	+ <i>TimedLeg</i>	Ausprägung des Teilwegs als fahrplanbehafteter Teilweg. Vgl. 9.3.6.
	b InterchangeLeg		+ <i>InterchangeLeg</i>	Ausprägung des Teilwegs als Umstieg zwischen Verkehrsmitteln. Vgl. 9.3.7.
	c ContinuousLeg		+ <i>ContinuousLeg</i>	Ausprägung des Teilwegs als Fortbewegung mit einem kontinuierlich verfügbaren Verkehrsmittel. Vgl. 9.3.8.

Tabelle 87: Beschreibung der Struktur *TripLegStructure*.

9.3.6. TimedLegStructure

<i>TimedLegStructure</i>			+Structure	Beinhaltet einen fahrplangebundenen Verbindungsanteil (Teilweg).
	LegBoard	1:1	+ <i>LegBoard</i>	Beginn (Haltepunkt) des Teilwegs. Vgl. 9.3.8.
	<i>LegIntermediates</i>	0:*	+ <i>LegIntermediates</i>	Dazwischenliegende, durchfahrene Haltepunkte auf dem Teilweg zwischen <i>LegBoard</i> und <i>LegAlign</i> . Vgl. 9.3.11..
	LegAlign	1:1	+ <i>LegAlign</i>	Ende (Haltepunkt) des Teilwegs. Vgl. 9.3.10.
	Service	1:1	+ <i>DatedJourney</i>	Angaben zum Verkehrsmittel, wie Linie, Verkehrsmitteltyp etc. Vgl. 7.6.2.
	<i>LegAttribute</i>	0:*	+ <i>LegAttribute</i>	Attribute, die nur auf einem Teil der Verbindung gültig sind und nicht für die ganze Fahrt. Vgl. 7.6.12.
<i>OperatingDays</i>	<i>OperatingDays</i>	0:1	+ <i>OperatingDays</i>	Verkehrstage für diese Verbindung. Vgl. 7.4.7.
	<i>OperatingDaysDescription</i>	0:1	+ <i>InternationalText</i>	Menschenlesbare Beschreibung der Verkehrstage, z. B. „Montag bis Freitag“ oder „Sonn- und Feiertag“.
	<i>LegTrack</i>	0:1	+ <i>LegTrack</i>	Detaillierter (geometrischer) Verlauf. Vgl. 9.3.12.
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 88: Beschreibung der Struktur *TimedLegStructure*.

9.3.7. InterchangeLegStructure

<i>InterchangeLegStructure</i>			+Structure	Beinhaltet einen Verbindungsanteil (Teilweg), der einen Umsteigevorgang zwischen zwei Verkehrsmitteln darstellt.
	a InterchangeMode	-1:1	<i>walk parkAndRide bikeAndRide carHire bikeHire protected-Connection guaranteed-Connection remainInVehicle change-WithinVehicle checkIn checkOut</i>	Klassifizierung von Umsteigevorgängen
	b ContinuousMode		<i>walk demandResponsive replacement-</i>	Modalität für kontinuierliche Verkehre.

			<i>Service</i>	
	LegStart	1:1	+LocationRef	Beginn (Ort) dieses Teilwegs. Vgl. 7.5.8.
	LegEnd	1:1	+LocationRef	Ende (Ort) dieses Teilwegs. Vgl. 7.5.8.
<i>TimeWindow</i>	<i>TimeWindowStart</i>	0:1	<i>xs:dateTime</i>	Frühester Zeitpunkt für den Start dieses Teilwegs.
	<i>TimeWindowEnd</i>	0:1	<i>xs:dateTime</i>	Spätester Zeitpunkt für das Ende dieses Teilwegs.
<i>InterchangeDuration</i>	Duration	1:1	<i>xs:duration</i>	Notwendige Gesamtumsteigezeit.
	<i>WalkDuration</i>	0:1	<i>xs:duration</i>	Fußweganteil der Gesamtumsteigezeit.
	<i>BufferTime</i>	0:1	<i>xs:duration</i>	Pufferzeitanteil der Gesamtumsteigezeit. Pufferzeiten („Check-In-Zeiten“) sind bei manchen Verkehrsmitteln vorgeschrieben, z. B. im Flugverkehr, bei Fähren oder auch Hochgeschwindigkeitszügen.
	<i>LegDescription</i>	0:1	+InternationalText	Beschreibung des Umsteigevorgangs.
	<i>Length</i>	0:1	LengthType	Länge des Umsteigewegs.
	<i>Attribute</i>	0:*	+GeneralAttribute	Hinweise und Attribute (mit Klassifikationen) zum Umsteigevorgang. Vgl. 7.4.9.
	<i>NavigationPath</i>	0:1	+NavigationPath	Detaillierte Informationen zum geometrischen Verlauf, der Wegfolge und Zugänglichkeit. Vgl. 9.3.14.
	<i>SituationFullRef</i>	0:*	+SituationFullRef	Verweis auf eine Störungsnachricht. Diese Nachricht kann im <i>TripResponseContext</i> (vgl. 9.3.2) zu finden sein oder auf anderem Wege bekannt gemacht werden.
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 89: Beschreibung der Struktur **InterchangeLegStructure**.

9.3.8. ContinuousLegStructure

ContinuousLegStructure			+Structure	Beinhaltet einen Verbindungsanteil (Teilweg), der nicht fahrplangebunden ist (z. B. Fußweg).
	LegStart	1:1	+LocationRef	Beginn (Ort) des Teilwegs dieser Verbindung. Vgl. 7.5.8.
	LegEnd	1:1	+LocationRef	Ende (Ort) des Teilwegs dieser Verbindung. Vgl. 7.5.8.
	Service	1:1	+ContinuousService	Angaben zum „Verkehrsmittel“ (z. B. Fußweg). Vgl. 7.6.7.
<i>TimeWindow</i>	<i>TimeWindowStart</i>	0:1	<i>xs:dateTime</i>	Frühester Zeitpunkt für den Start dieses Teilwegs.
	<i>TimeWindowEnd</i>	0:1	<i>xs:dateTime</i>	Spätester Zeitpunkt für das Ende dieses Teilwegs.
	Duration	1:1	<i>xs:duration</i>	Dauer dieses Teilwegs.
	<i>LegDescription</i>	0:1	+InternationalText	Beschreibung dieses Verbindungsanteils.
	<i>Length</i>	0:1	LengthType	Länge dieses Teilwegs.
	<i>LegTrack</i>	0:1	+LegTrack	Detaillierter (geometrischer) Verlauf. Vgl. 9.3.12.
	<i>NavigationPath</i>	0:1	+NavigationPath	Detaillierte Informationen zum geometrischen Verlauf, der Wegfolge und Zugänglichkeit. Vgl. 9.3.14.
	<i>SituationFullRef</i>	0:*	+SituationFullRef	Verweis auf eine Störungsnachricht. Diese Nachricht kann im <i>TripResponseContext</i> (vgl. 9.3.2) zu finden sein oder auf anderem Wege bekannt gemacht werden. Vgl. 7.8.2.
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 90: Beschreibung der Struktur **ContinuousLegStructure**.

9.3.9. LegBoardStructure

LegBoardStructure			+Structure	Beschreibt die Einstiegssituation in ein Verkehrsmittel.
Stop-Point	StopPointRef	1:1	→ <i>StopPoint</i>	Referenz auf einen Code für einen Haltepunkt. Vgl. 7.5.1.
	StopPointName	1:1	+ <i>InternationalText</i>	Name des Haltepunkts für Fahrgastinformation.
	<i>NameSuffix</i>	0:1	+ <i>InternationalText</i>	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „gegenüber vom Haupteingang“.
	<i>PlannedBay</i>	0:1	+ <i>InternationalText</i>	Name des Steigs/Haltepunkts, wo in das Fahrzeug ein- oder ausgestiegen werden muss (bei Verwendung in Zusammenhang mit einer konkreten Verbindungsauskunft, wenn in StopPointName ein allgemeiner Name angegeben ist, ähnlich Haltestellenname). Nach Planungsstand.
	<i>EstimatedBay</i>	0:1	+ <i>InternationalText</i>	Name des Steigs/Haltepunkts, wo in das Fahrzeug ein- oder ausgestiegen werden muss (bei Verwendung in Zusammenhang mit einer konkreten Verbindungsauskunft, wenn in StopPointName ein allgemeiner Name angegeben ist, ähnlich Haltestellenname). Nach letztem Prognosestand.
Service Arrival	<i>TimetabledTime</i>	0:1	<i>xs:dateTime</i>	Ankunftszeit nach Fahrplan.
	<i>RecordedAtTime</i>	0:1	<i>xs:dateTime</i>	Tatsächliche Ankunftszeit.
	<i>EstimatedTime</i>	0:1	<i>xs:dateTime</i>	Erwartete Ankunftszeit.
	<i>EstimatedTimeLow</i>	0:1	<i>xs:dateTime</i>	Untere Schranke für erwartete Ankunftszeit.
	<i>EstimatedTimeHigh</i>	0:1	<i>xs:dateTime</i>	Obere Schranke für erwartete Ankunftszeit.
Service Departure	TimetabledTime	1:1	<i>xs:dateTime</i>	Abfahrtszeit nach Fahrplan.
	<i>RecordedAtTime</i>	0:1	<i>xs:dateTime</i>	Tatsächliche Abfahrtszeit.
	<i>EstimatedTime</i>	0:1	<i>xs:dateTime</i>	Erwartete Abfahrtszeit.
	<i>EstimatedTimeLow</i>	0:1	<i>xs:dateTime</i>	Untere Schranke für erwartete Abfahrtszeit.
	<i>EstimatedTimeHigh</i>	0:1	<i>xs:dateTime</i>	Obere Schranke für erwartete Abfahrtszeit.
	<i>MeetsViaRequest</i>	0:1	<i>xs:boolean</i>	Dieser Halt erfüllt eine der in der Anfrage vorgegebenen Via-Bedingungen.
StopCall Status	StopSeqNumber	1:1	<i>xs:positiveInteger</i>	Laufende Nummer des Halts im Fahrweg der Fahrt.
	<i>DemandStop</i>	0:1	<i>xs:boolean</i>	Bedarfshalt. Fahrzeug bedient diesen Halt nur nach Voranmeldung.
	<i>UnplannedStop</i>	0:1	<i>xs:boolean</i>	Halt, der laut Planung nicht vorgesehen war.
	<i>NotServicedStop</i>	0:1	<i>xs:boolean</i>	Entgegen der Planung findet kein Halt statt.

Tabelle 91: Beschreibung der Struktur **LegBoardStructure**.

9.3.10. LegAlightStructure

LegAlightStructure			+Structure	Beschreibt die Ausstiegssituation aus einem Verkehrsmittel.
Stop-Point	StopPointRef	1:1	→ <i>StopPoint</i>	Referenz auf einen Code für einen Haltepunkt. Vgl. 7.5.1.
	StopPointName	1:1	+ <i>InternationalText</i>	Name des Haltepunkts für Fahrgastinformation.
	<i>NameSuffix</i>	0:1	+ <i>InternationalText</i>	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „gegenüber vom Haupteingang“.

	<i>PlannedBay</i>	0:1	+International Text	Name des Steigs/Haltepunkts, wo in das Fahrzeug ein- oder ausgestiegen werden muss (bei Verwendung in Zusammenhang mit einer konkreten Verbindungsauskunft, wenn in StopPointName ein allgemeiner Name angegeben ist, ähnlich Haltestellenname). Nach Planungsstand.
	<i>EstimatedBay</i>	0:1	+International Text	Name des Steigs/Haltepunkts, wo in das Fahrzeug ein- oder ausgestiegen werden muss (bei Verwendung in Zusammenhang mit einer konkreten Verbindungsauskunft, wenn in StopPointName ein allgemeiner Name angegeben ist, ähnlich Haltestellenname). Nach letztem Prognosestand.
Service Arrival	TimetabledTime	1:1	<i>xs:dateTime</i>	Ankunftszeit nach Fahrplan.
	<i>RecordedAtTime</i>	0:1	<i>xs:dateTime</i>	Tatsächliche Ankunftszeit.
	<i>EstimatedTime</i>	0:1	<i>xs:dateTime</i>	Erwartete Ankunftszeit.
	<i>EstimatedTimeLow</i>	0:1	<i>xs:dateTime</i>	Untere Schranke für erwartete Ankunftszeit.
	<i>EstimatedTimeHigh</i>	0:1	<i>xs:dateTime</i>	Obere Schranke für erwartete Ankunftszeit.
Service Departu re	<i>TimetabledTime</i>	0:1	<i>xs:dateTime</i>	Abfahrtszeit nach Fahrplan.
	<i>RecordedAtTime</i>	0:1	<i>xs:dateTime</i>	Tatsächliche Abfahrtszeit.
	<i>EstimatedTime</i>	0:1	<i>xs:dateTime</i>	Erwartete Abfahrtszeit.
	<i>EstimatedTimeLow</i>	0:1	<i>xs:dateTime</i>	Untere Schranke für erwartete Abfahrtszeit.
	<i>EstimatedTimeHigh</i>	0:1	<i>xs:dateTime</i>	Obere Schranke für erwartete Abfahrtszeit.
	<i>MeetsViaRequest</i>	0:1	<i>xs:boolean</i>	Dieser Halt erfüllt eine der in der Anfrage vorgegebenen Via-Bedingungen.
StopCall Status	StopSeqNumber	1:1	<i>xs:positiveInteger</i>	Laufende Nummer des Halts im Fahrweg der Fahrt.
	<i>DemandStop</i>	0:1	<i>xs:boolean</i>	Bedarfshalt. Fahrzeug bedient diesen Halt nur nach Voranmeldung.
	<i>UnplannedStop</i>	0:1	<i>xs:boolean</i>	Halt, der laut Planung nicht vorgesehen war.
	<i>NotServicedStop</i>	0:1	<i>xs:boolean</i>	Entgegen der Planung findet kein Halt statt.

Tabelle 92: Beschreibung der Struktur **LegAlightStructure**.

9.3.11. LegIntermediateStructure

LegIntermediateStructure			+Structure	Zwischenhalt auf einem Teilweg.
Stop- Point	StopPointRef	1:1	→ <i>StopPoint</i>	Referenz auf einen Code für einen Haltepunkt. Vgl. 7.5.1.
	StopPointName	1:1	+International Text	Name des Haltepunkts für Fahrgastinformation.
	<i>NameSuffix</i>	0:1	+International Text	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „gegenüber vom Haupteingang“.
	<i>PlannedBay</i>	0:1	+International Text	Name des Steigs/Haltepunkts, wo in das Fahrzeug ein- oder ausgestiegen werden muss (bei Verwendung in Zusammenhang mit einer konkreten Verbindungsauskunft, wenn in StopPointName ein allgemeiner Name angegeben ist, ähnlich Haltestellenname). Nach Planungsstand.
	<i>EstimatedBay</i>	0:1	+International Text	Name des Steigs/Haltepunkts, wo in das Fahrzeug ein- oder ausgestiegen werden muss (bei Verwendung in Zusammenhang mit einer konkreten Verbindungsauskunft, wenn in StopPointName ein allgemeiner Name angegeben ist, ähnlich Haltestellenname). Nach letztem Prognosestand.

Service Arrival	TimetabledTime	1:1	<i>xs:dateTime</i>	Ankunftszeit nach Fahrplan.
	<i>RecordedAtTime</i>	0:1	<i>xs:dateTime</i>	Tatsächliche Ankunftszeit.
	<i>EstimatedTime</i>	0:1	<i>xs:dateTime</i>	Erwartete Ankunftszeit.
	<i>EstimatedTimeLow</i>	0:1	<i>xs:dateTime</i>	Untere Schranke für erwartete Ankunftszeit.
	<i>EstimatedTimeHigh</i>	0:1	<i>xs:dateTime</i>	Obere Schranke für erwartete Ankunftszeit.
Service Departure	TimetabledTime	1:1	<i>xs:dateTime</i>	Abfahrtszeit nach Fahrplan.
	<i>RecordedAtTime</i>	0:1	<i>xs:dateTime</i>	Tatsächliche Abfahrtszeit.
	<i>EstimatedTime</i>	0:1	<i>xs:dateTime</i>	Erwartete Abfahrtszeit.
	<i>EstimatedTimeLow</i>	0:1	<i>xs:dateTime</i>	Untere Schranke für erwartete Abfahrtszeit.
	<i>EstimatedTimeHigh</i>	0:1	<i>xs:dateTime</i>	Obere Schranke für erwartete Abfahrtszeit.
	<i>MeetsViaRequest</i>	0:1	<i>xs:boolean</i>	Dieser Halt erfüllt eine der in der Anfrage vorgegebenen Via-Bedingungen.
StopCall Status	StopSeqNumber	1:1	<i>xs:positiveInteger</i>	Laufende Nummer des Halts im Fahrweg der Fahrt.
	<i>DemandStop</i>	0:1	<i>xs:boolean</i>	Bedarfshalt. Fahrzeug bedient diesen Halt nur nach Voranmeldung.
	<i>UnplannedStop</i>	0:1	<i>xs:boolean</i>	Halt, der laut Planung nicht vorgesehen war.
	<i>NotServicedStop</i>	0:1	<i>xs:boolean</i>	Entgegen der Planung findet kein Halt statt.

Tabelle 93: Beschreibung der Struktur **LegIntermediateStructure**.

9.3.12. LegTrackStructure

LegTrackStructure			+Structure	Container für die Streckenabschnitte entlang eines Verbindungsteilwegs.
	TrackSection	1:*	+TrackSection	Ein oder mehrere Streckenabschnitte. Vgl. 9.3.13.

Tabelle 94: Beschreibung der Struktur **LegTrackStructure**.

9.3.13. TrackSectionStructure

TrackSectionStructure			+Structure	Ein Streckenabschnitt in einem Verbindungsteilweg.
	<i>TrackStart</i>	0:1	+LocationRef	Beginn (Ort) des Streckenabschnitts. Vgl. 7.5.8.
	<i>TrackEnd</i>	0:1	+LocationRef	Ende (Ort) des Streckenabschnitts. Vgl. 7.5.8.
Projection	Position	2:*	+GeoPosition	Geographische Projektion des Streckenabschnitts als Polygonzug. Vgl. 7.2.3.
	<i>RoadName</i>	0:1	<i>xs:string</i>	Name der Straße, auf der dieser Streckenabschnitt liegt.
	<i>Duration</i>	0:1	<i>xs:duration</i>	Zeitdauer, die der Fahrgast braucht, um diesen Streckenabschnitt zu bewältigen.
	<i>Length</i>	0:1	<i>LengthType</i>	Länge des Streckenabschnitts.
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 95: Beschreibung der Struktur **TrackSectionStructure**.

9.3.14. NavigationPathStructure

NavigationPathStructure			+Structure	Container für Wegbeschreibungen.
	NavigationSection	1:*	+NavigationSection	Ein oder mehrere Streckenabschnitte. Vgl. 9.3.15

Tabelle 96: Beschreibung der Struktur **NavigationPathStructure**.

9.3.15. NavigationSectionStructure

NavigationSectionStructure		+Structure	Beschreibung eines Wegstücks, evtl. mit Angabe der geographischen Einbettung, der Abbiegeanweisungen und der Wegbeschaffenheit (Zugänglichkeit für mobilitätseingeschränkte Personen).
<i>TrackSection</i>	0:1	+TrackSection	Geographische Beschreibung des Streckenabschnitts. Vgl. 9.3.13.
<i>TurnDescription</i>	0:1	+International Text	Beschreibung des durchzuführenden Manövers. Es sollte in textueller Form die Inhalte von <i>Manoeuvre</i> , <i>TurnAction</i> und <i>TrackSection.RoadName</i> beschreiben.
<i>Manoeuvre</i>	0:1	<i>origin destination continue keep turn leave enter</i>	Codierung des durchzuführenden Manövers.
<i>TurnAction</i>	0:1	<i>sharp left left half left straight on half right right sharp right uturn</i>	Codierung der Abbiegevorgänge.
<i>DirectionHint</i>	0:1	+International Text	Textueller Richtungshinweis zum besseren Verständnis des nachfolgenden Streckenabschnittes, z. B. „Folgen Sie der Beschilderung nach Hamburg“.
<i>Bearing</i>	0:1	<i>AbsoluteBearing</i>	Himmelsrichtung, die nach dem Manöver eingeschlagen ist. Sie bezieht sich nicht auf das gesamte Wegstück.
<i>SituationFullRef</i>	0:*	+SituationFullRef	Verweise auf Störungsnachrichten. Diese Nachrichten können im <i>TripResponseContext</i> (vgl. 9.3.2) zu finden sein oder auf anderem Wege bekannt gemacht werden. Vgl. 7.8.2.
<i>AccessPath</i>	0:1	+AccessPath	Beschreibung der Zugänglichkeit des Wegstücks. Vgl. 9.3.16.

Tabelle 97: Beschreibung der Struktur **NavigationSectionStructure**.

9.3.16. AccessPathStructure

AccessPathStructure		+Structure	Beschreibung der Zugänglichkeit eines Wegstücks.
<i>Transition</i>	0:1	<i>up down level upAndDown downAndUp</i>	Angabe, ob Weg eben geht oder aufwärts/abwärts führt.
<i>AccessFeatureType</i>	0:1	<i>lift stairs seriesOfStairs escalator ramp footpath</i>	Wegtyp.
<i>Count</i>	0:1	<i>xs:positiveInteger</i>	Anzahl, wie oft der Wegtyp vorkommt.

Tabelle 98: Beschreibung der Struktur **AccessPathStructure**.

10. Dienst Abfahrtstafeln

10.1. Beschreibung

Dieser Dienst informiert über Ankünfte und Abfahrten von ÖV-Fahrten an Haltestellen für einen bestimmten Zeitpunkt oder Zeitraum. In den Anfrageparametern können Einschränkungen vorgegeben werden, die sich als Filter auf die Ergebnisse auswirken.

10.2. Anfragestrukturen

Eine Abfahrtstafel (bzw. Ankunftstafel) wird mittels eines Elements **StopEventRequest** vom Typ *StopEventRequestStructure* angefordert.

10.2.1. StopEventRequestStructure

StopEventRequestStructure			+Structure	Fasst die Abfragedaten für eine Abfahrts- oder Ankunftstafel zusammen.
	Location	1:1	+LocationContext	Ortsdaten für die Abfahrts-/Ankunftstafel. Vgl. 7.6.10.
	Params	0:1	+StopEventParameter	Spezifische Anfrageparameter. Vgl. 10.2.2.

Tabelle 99: Beschreibung der Struktur **StopEventRequestStructure**.

10.2.2. StopEventParamStructure

StopEventParamStructure			+Structure	Fasst die Anfrageparameter zusammen, die die Berechnung einer Abfahrts- oder Ankunftstafel steuern.
StopEventDataFilter	PtModeFilter	0:1	+PtModeFilter	Erlaubte Verkehrsmittel. Vgl. 7.3.5.
	LineFilter	0:1	+LineDirectionFilter	Erlaubte Linien (ggf. verfeinert auf Richtungen). Vgl. 7.4.6.
	OperatorFilter	0:1	+OperatorFilter	Erlaubte Verkehrsunternehmen. Vgl. 7.4.4.
StopEventPolicy	NumberOfResults	0:1	xs:positiveInteger	Maximale Zahl von Abfahrts-/Ankunftsereignissen, die in der Antwort zurückgegeben werden sollen.
	TimeWindow	0:1	xs:duration	Zeitfenster, in dem Abfahrts-/Ankunftsereignisse in der Antwort zurückgegeben werden sollen. Wird gerechnet ab dem in <i>LocationContext</i> angegebenen Zeitpunkt.
	StopEventType	0:1	departure arrival both	Gibt an, ob Abfahrts- oder Ankunftsereignisse oder beides zurückgegeben werden sollen.
StopEventContentFilter	IncludePreviousCalls	0:1	xs:boolean	Gibt an, ob je Fahrt die vorausgehenden Halte angeführt werden sollen.
	IncludeOnwardCalls	0:1	xs:boolean	Gibt an, ob je Fahrt die nachfolgenden Halte angeführt werden sollen.
	IncludeOperatingDays	0:1	xs:boolean	Gibt an, ob die Verkehrstage der Fahrten angegeben werden sollen.
	IncludeRealtimeData	0:1	xs:boolean	Steuert, ob Echtzeitdaten berücksichtigt und ausgegeben werden sollen.

Tabelle 100: Beschreibung der Struktur **StopEventParamStructure**.

10.3. Antwortstrukturen

Das Ergebnis einer Abfahrtstafelanfrage wird mittels eines Elements **StopEventResponse** vom Typ *StopEventResponseStructure* übertragen.

10.3.1. StopEventResponseStructure

StopEventResponseStructure		+Structure	Fasst die Ergebnisdaten für eine Abfahrts- oder Ankomstafelanfrage zusammen.
	<i>ErrorMessage</i>	0:*	+ErrorMessage Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
	<i>StopEventResponseContext</i>	0:1	+StopEventResponseContext Container für Daten, die in der Antwort mehrfach auftreten und referenziert werden. Vgl. 10.3.2.
	<i>StopEventResult</i>	0:*	+StopEventResult Container für ein Abfahrts- oder Ankomstereignis. Vgl. 10.3.3.

Tabelle 101: Beschreibung der Struktur **StopEventResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

STOPEVENT_DATEOUTOFRANGE	Für das angefragte Datum liegen keine Fahrplandaten vor.
STOPEVENT_LOCATIONUNKNOWN	Der Ort (Adresse, Haltestelle, ...), für den die Abfahrtstafel/Ankomststafel angefordert wurde, ist unbekannt.
STOPEVENT_LOCATIONUNSERVED	Der Ort (Adresse, Haltestelle, ...), für den die Abfahrtstafel/Ankomststafel angefordert wurde, wird überhaupt nicht von öffentlichen Verkehrsmitteln bedient.
STOPEVENT_NOEVENTFOUND	Im fraglichen Zeitraum wurde keine Abfahrt/Ankomst unter Einhaltung der gegebenen Optionen gefunden.

Tabelle 102: Liste der Fehlerzustände in **StopEventResponse**.

10.3.2. StopEventResponseContextStructure

StopEventResponseContextStructure	+Structure (abgeleitet von AbstractResponseContextStructure)	Container für Daten, die in der Antwort mehrfach auftreten und referenziert werden. Vgl. 7.6.11.

Tabelle 103: Beschreibung der Struktur **StopEventResponseContextStructure**.

10.3.3. StopEventResultStructure

StopEventResultStructure		+Structure	Fasst die Ergebnisdaten für ein einzelnes Abfahrts- oder Ankomstereignis zusammen.
	ResultId	1:1	xs:NMTOKEN ID des Resultats für spätere Referenzierung bzw. für Debug-Zwecke.
	<i>ErrorMessage</i>	0:*	+ErrorMessage Fehlermeldungen bezogen auf dieses Abfahrts-/Ankomstereignis. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. auch 7.4.2.
	StopEvent	1:1	+StopEvent Daten zu einem Abfahrts- oder Ankomstereignis. Vgl. 10.3.4.

Tabelle 104: Beschreibung der Struktur **StopEventResultStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

STOPEVENT_LASTSERVICEOFTHISLINE	Diese Abfahrt/Ankunft ist die letzte dieser Linie an dieser Haltestelle an diesem Betriebstag.
STOPEVENT_NOREALTIME	Für diese Abfahrt/Ankunft liegen keine Echtzeitdaten oder Prognosen vor.

Tabelle 105: Liste der Fehlerzustände in **StopEventResult**.

10.3.4. StopEventStructure

StopEventStructure			+Structure	Daten zu einem einzelnen Abfahrts- oder Ankunftsereignis.
	<i>PreviousCall</i>	0:*	+CallAtNearStop	Abfahrts-/Ankunftsereignisse an Haltestellen vor der angefragten Haltestelle. Vgl. 10.3.5.
	ThisCall	1:1	+CallAtNearStop	Abfahrts-/Ankunftsereignis an der angefragten Haltestelle. Vgl. 10.3.5.
	<i>OnwardCall</i>	0:*	+CallAtNearStop	Abfahrts-/Ankunftsereignis nach der angefragten Haltestelle. Vgl. 10.3.5.
	Service	1:1	+DatedJourney	Angaben zum Verkehrsmittel, Linie etc. Vgl. 7.6.2
<i>OperatingDays</i>	<i>OperatingDays</i>	0:1	+OperatingDays	Verkehrstage für dieses Abfahrts-/Ankunftsereignis. Vgl. 7.4.7..
	<i>OperatingDaysDescription</i>	0:1	+InternationalText	Menschenlesbare Beschreibung der Verkehrstage, z. B. „Montag bis Freitag“ oder „Sonn- und Feiertag“.
	<i>Extension</i>	0:1	xs:anyType	Erweiterungen.

Tabelle 106: Beschreibung der Struktur **StopEventStructure**.

10.3.5. CallAtNearStopStructure

CallAtNearStopStructure			+Structure	Abfahrt oder Ankunft an einer Haltestelle in der Umgebung.
	CallAtStop	1:1	+CallAtStop	Abfahrt oder Ankunft an einem Haltepunkt. Vgl.7.6.5..
	<i>WalkDistance</i>	0:1	<i>Distance</i>	Distanz des Haltepunkts vom angefragten Ort in Metern. Der angefragte Ort kann z. B. eine Adresse sein.
	<i>WalkDuration</i>	0:1	xs:duration	Zeitliche Distanz des Haltepunkts vom angefragten Ort. Der angefragte Ort kann z. B. eine Adresse sein. Der Zeitbedarf ergibt sich durch die IV-Einstellungen in der Anfrage: es wird also z. B. berücksichtigt, ob ein Fahrrad benutzt werden kann, um vom angefragten Ort zum AbfahrtsHaltepunkt zu gelangen.

Tabelle 107: Beschreibung der Struktur **CallAtNearStopStructure**.

11. Dienst Logische Ortung

11.1. Beschreibung

Der Dienst Logische Ortung hat die Aufgabe, den Aufenthaltsort des Fahrgasts im ÖV-Netz zu bestimmen. Er benutzt dabei das Bewegungsmuster des Fahrgasts, das entweder sein Mobilgerät aufgezeichnet hat oder durch den Fahrplan des Fahrzeugs bestimmt wird, in dem er sich gerade befindet. Als Resultat erhält man mögliche Aufenthaltsorte mit Angabe der jeweiligen Wahrscheinlichkeit.

11.2. Anfragestrukturen

Eine logische Ordnung wird mittels eines Elements **PositioningRequest** vom Typ *PositioningRequestStructure* angefordert.

11.2.1. PositioningRequestStructure

PositioningRequestStructure			+Structure	Fasst die Anfragedaten für eine logische Ordnung zusammen.	
	a	LastPositions	-1:1	+TimedPosition	Bewegungsmuster des Fahrgasts als Folge von Koordinaten mit Zeitstempel. Vgl. 11.2.3.
	b	StopSequence		+TimedStop	Bewegungsmuster des Fahrgasts als Folge von Haltepunkten mit Uhrzeiten. Vgl. 11.2.4.
		<i>Params</i>	0:1	+PositioningParameter	Spezifische Anfrageparameter. Vgl. 11.2.2.

Tabelle 108: Beschreibung der Struktur **PositioningRequestStructure**.

11.2.2. PositioningParamStructure

PositioningParamStructure			+Structure	Fasst die Anfrageparameter zusammen, die die Bestimmung einer logischen Ordnung des Fahrgasts im ÖV-Netz steuern.
<i>PositioningDataFilter</i>	<i>PtModeFilter</i>	0:1	+PtModeFilter	Verkehrsmittel-Filter. Vgl. 7.3.5.
	<i>LineFilter</i>	0:1	+LineDirectionFilter	Linien-Filter (ggf. verfeinert auf Richtungen). Vgl. 7.4.6.
	<i>OperatorFilter</i>	0:1	+OperatorFilter	Verkehrsunternehmen-Filter. Vgl. 7.4.4.
<i>PositioningPolicy</i>	<i>NumberOfResults</i>	0:1	<i>xs:positiveInteger</i>	Maximale Zahl von Ortungsvorschlägen, die in der Antwort zurückgegeben werden dürfen.

Tabelle 109: Beschreibung der Struktur **PositioningParamStructure**.

11.2.3. TimedPositionStructure

TimedPositionStructure			+Structure	Geographische Position mit Zeitstempel.
	Timestamp	1:1	<i>xs:dateTime</i>	Zeitstempel, wann der Fahrgast diesen Punkt passiert hat.
	Position	1:1	+GeoPosition	Koordinatenposition. Vgl. 7.2.3.
	<i>Speed</i>	0:1	<i>Speed</i>	Geschwindigkeit, mit der der Fahrgast die Position passiert hat. Vgl. 7.2.1.
	<i>Direction</i>	0:1	<i>AbsoluteBearing</i>	Kompassrichtung, in der der Fahrgast die Position passiert hat. Vgl. 7.2.1.

Tabelle 110: Beschreibung der Struktur **TimedPositionStructure**.

11.2.4. TimedStopStructure

TimedStopStructure			+Structure	Beschreibt einen Haltepunkt evtl. mit Zeiten, wann er erreicht und/oder verlassen wird.
<i>Stop-Point</i>	StopPointRef	1:1	→ <i>StopPoint</i>	Referenz auf einen Code für einen Haltepunkt. Vgl. 7.5.1.
	StopPointName	1:1	+ <i>InternationalText</i>	Name des Haltepunkts für Fahrgastinformation.
	<i>NameSuffix</i>	0:1	+ <i>InternationalText</i>	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „gegenüber vom Haupteingang“.

	<i>PlannedBay</i>	0:1	+ <i>International Text</i>	Name des Steigs/Haltepunkts, wo in das Fahrzeug ein- oder ausgestiegen werden muss (bei Verwendung in Zusammenhang mit einer konkreten Verbindungsauskunft, wenn in <i>StopPointName</i> ein allgemeiner Name angegeben ist, ähnlich Haltestellenname). Nach Planungsstand.
	<i>EstimatedBay</i>	0:1	+ <i>International Text</i>	Name des Steigs/Haltepunkts, wo in das Fahrzeug ein- oder ausgestiegen werden muss (bei Verwendung in Zusammenhang mit einer konkreten Verbindungsauskunft, wenn in <i>StopPointName</i> ein allgemeiner Name angegeben ist, ähnlich Haltestellenname). Nach letztem Prognosestand.
<i>ArrivalTime</i>	<i>TimetabledTime</i>	0:1	<i>xs:dateTime</i>	Geplante Ankunftszeit.
	<i>RecordedAtTime</i>	0:1	<i>xs:dateTime</i>	Tatsächliche Ankunftszeit.
	<i>EstimatedTime</i>	0:1	<i>xs:dateTime</i>	Erwartete Ankunftszeit.
	<i>EstimatedTimeLow</i>	0:1	<i>xs:dateTime</i>	Untere Schranke für erwartete Ankunftszeit.
	<i>EstimatedTimeHigh</i>	0:1	<i>xs:dateTime</i>	Obere Schranke für erwartete Ankunftszeit.
<i>DepartureTime</i>	<i>TimetabledTime</i>	0:1	<i>xs:dateTime</i>	Geplante Abfahrtszeit.
	<i>RecordedAtTime</i>	0:1	<i>xs:dateTime</i>	Tatsächliche Abfahrtszeit.
	<i>EstimatedTime</i>	0:1	<i>xs:dateTime</i>	Erwartete Abfahrtszeit.
	<i>EstimatedTimeLow</i>	0:1	<i>xs:dateTime</i>	Untere Schranke für erwartete Abfahrtszeit.
	<i>EstimatedTimeHigh</i>	0:1	<i>xs:dateTime</i>	Obere Schranke für erwartete Abfahrtszeit.

Tabelle 111: Beschreibung der Struktur **TimedStopStructure**.

11.3. Antwortstrukturen

Das Ergebnis einer Ortungsanfrage wird mittels eines Elements **PositioningResponse** vom Typ **PositioningResponseStructure** übertragen.

11.3.1. PositioningResponseStructure

PositioningResponseStructure		+ <i>Structure</i>	Fasst die Ergebnisdaten für eine Ortungsanfrage zusammen.
	<i>ErrorMessage</i>	0:*	+ <i>ErrorMessage</i>
	<i>PositioningResult</i>	0:1	+ <i>PositioningResult</i>
			Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
			Struktur für ein Ortungsergebnis. Vgl. 11.3.2.

Tabelle 112: Beschreibung der Struktur **PositioningResponseStructure**.

In **ErrorMessage** können folgende Fehlerzustände auftreten:

POSITIONING_NOMATCH	Kein passendes Resultat gefunden.
POSITIONING_DATEOUTOFRANGE	Für das angefragte Datum liegen keine Fahrplandaten vor.
POSITIONING_SPEEDTOOFAST	Die dem Bewegungsmuster zugrundeliegende Geschwindigkeit ist zu schnell.
POSITIONING_COORDOUTOFRANGE	Die angegebenen Koordinaten liegen außerhalb des betrachteten Gebiets.
POSITIONING_STOPUNKNOWN	Eine angegebene Haltestelle ist unbekannt.

Tabelle 113: Liste der Fehlerzustände in **PositioningResponse**.

11.3.2. PositioningResultStructure

<i>PositioningResultStructure</i>		+Structure	Resultatstruktur für das Ortungsergebnis.
	<i>ResultId</i>	1:1	<i>xs:NMTOKEN</i> ID des Resultats für spätere Referenzierung bzw. für Debug-Zwecke.
	<i>Positioning</i>	1:1	+ <i>Positioning</i> Container für die Ortungsvorschläge. Vgl. 11.3.3
	<i>Extension</i>	0:1	<i>xs:anyType</i> Erweiterungen.

Tabelle 114: Beschreibung der Struktur *PositioningResultStructure*.

11.3.3. PositioningStructure

<i>PositioningStructure</i>		+Structure	Container für die Ortungsvorschläge.
	<i>RankedPosition</i>	1:*	+ <i>RankedPosition</i> Ein oder mehrere Ortungsvorschläge. Vgl. 11.3.4.

Tabelle 115: Beschreibung der Struktur *PositioningStructure*.

11.3.4. RankedPositionStructure

<i>RankedPositionStructure</i>		+Structure	Positionsvorschlag mit Wahrscheinlichkeitseinstufung.
	<i>a StationaryLocation</i>	-1:1	+ <i>LocationRef</i> Position im ÖV-Netz außerhalb Fahrzeugen. Vgl. 7.5.8.
	<i>b TripLocation</i>		+ <i>DatedJourney</i> Position im ÖV-Netz in einer Fahrt. Vgl.7.6.2.
	<i>Ranking</i>	1:1	<i>Percent</i> Wahrscheinlichkeitseinstufung in Prozent. Der Wert 100 bedeutet absolute Gewissheit. Vgl. 7.2.1.

Tabelle 116: Beschreibung der Struktur *RankedPositionStructure*.

12. Dienst Fahrtinformation (EKAP)

12.1. Beschreibung

Im Dienst Fahrtinformation (EKAP) liefert eine EKAP Informationen zu einer bestimmten Fahrt.

12.2. Anfragestrukturen

Die Fahrtinformationen werden mittels eines Elements *TripInfoRequest* vom Typ *TripInfoRequestStructure* angefordert.

12.2.1. TripInfoRequestStructure

<i>TripInfoRequestStructure</i>		+Structure	Fasst die Anfragedaten für eine Fahrtinformationsanfrage zusammen.
a	<i>JourneyRef</i>	-1:1	→ <i>Journey</i> Referenz auf eine Fahrt. Vgl. 7.4.1.
	<i>OperatingDayRef</i>	-1:1	→ <i>OperatingDay</i> Referenz auf einen Verkehrstag. Vgl. 7.4.1.
b	<i>VehicleRef</i>	-1:1	→ <i>Vehicle</i> Referenz auf ein Fahrzeug. Vgl. 7.4.1.
	<i>TimeOfOperation</i>		<i>xs:dateTime</i> Zeitpunkt, wann das Fahrzeug unterwegs ist. Dieser Wert wird in den meisten Anwendungsfällen dem Zeitpunkt „Jetzt“ entsprechen.

	<i>Params</i>	0:1	+ <i>TripInfoParam</i>	Parameter, die die Suche und Rückgabewerte beeinflussen können. Vgl. 12.2.2.
--	---------------	-----	------------------------	--

Tabelle 117: Beschreibung der Struktur ***TripInfoRequestStructure***.

Die Informationen zu einer Fahrt können mittels einer Fahrt-ID (*JourneyRef*) oder einer Fahrzeug-ID (*VehicleRef*) angefragt werden.

Bei Verwendung der Fahrzeug-ID wird mit der zusätzlichen Information des Zeitpunkts in *TimeOfOperation* eindeutig die Fahrt aus der Menge aller Fahrten ausgewählt, die das Fahrzeug an dem gewählten Tag durchführt.

12.2.2. TripInfoParamStructure

<i>TripInfoParamStructure</i>			+ <i>Structure</i>	Fasst die Parameter für eine Fahrtinformationsanfrage zusammen.
<i>TripInfoPolicy</i>	<i>UseTimetabledDataOnly</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Informationen zu den Verkehrstagen mitausgegeben werden sollen. Default ist <i>false</i> .
<i>TripInfoContentFilter</i>	<i>IncludeCalls</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat die Halte der Fahrt ausgegeben werden sollen. Default ist <i>true</i> .
	<i>IncludePosition</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat die aktuelle Position der ausgegeben werden soll. Default ist <i>true</i> .
	<i>IncludeService</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Verkehrsmittelinformationen zur Fahrt ausgegeben werden sollen. Default ist <i>true</i> .
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 118: Beschreibung der Struktur ***TripInfoParamStructure***.

12.3. Antwortstrukturen

Das Ergebnis einer Fahrtinformationsanfrage wird mittels eines Elements ***TripInfoResponse*** vom Typ *TripInfoResponseStructure* übertragen.

12.3.1. TripInfoResponseStructure

<i>TripInfoResponseStructure</i>			+ <i>Structure</i>	Fasst die Ergebnisdaten für eine Fahrtinformationsanfrage zusammen.
	<i>ErrorMessage</i>	0:*	+ <i>ErrorMessage</i>	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
	<i>TripInfoResponseContent</i>	0:1	+ <i>TripInfoResponseContext</i>	Container für Daten, die in der Antwort mehrfach auftreten und referenziert werden. Vgl. 12.3.2.
	<i>TripInfoResult</i>	0:1	+ <i>TripInfoResult</i>	Container für die Fahrtinformationen. Vgl. 12.3.3.

Tabelle 119: Beschreibung der Struktur ***TripInfoResponseStructure***.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<i>TRIPINFO_JOURNEYUNKNOWN</i>	Die angefragte Fahrt-ID (<i>JourneyRef</i>) ist unbekannt.
<i>TRIPINFO_VEHICLEUNKNOWN</i>	Die angefragte Fahrzeug-ID (<i>VehicleRef</i>) ist unbekannt.
<i>TRIPINFO_NOJOURNEYFOUND</i>	Für die angefragte Fahrzeug-ID (<i>VehicleRef</i>) kann für die Uhrzeit keine passende Fahrt gefunden werden.

Tabelle 120: Liste der Fehlerzustände in ***TripInfoResponse***.

12.3.2. TripInfoResponseContextStructure

<i>TripInfoResponseContextStructure</i>	+Structure (abgeleitet von AbstractResponse- ContextStructure)	Container für Daten, die in der Antwort mehrfach auftreten und referenziert werden. Vgl. 7.6.11.
--	---	---

Tabelle 121: Beschreibung der Struktur ***TripInfoResponseContextStructure***.

12.3.3. TripInfoResultStructure

<i>TripInfoResultStructure</i>		+Structure	Resultatstruktur, die die Fahrtinformationen zusammenfasst.
	<i>PreviousCall</i>	0:*	+CallAtNearStop Bereits zurückgelegte Halte. Umfasst auch den aktuellen Halt, falls sich die Fahrt gerade an einer Haltestelle befindet. Vgl. 10.3.5.
	<i>CurrentPosition</i>	0:1	+VehiclePosition Aktuelle Position des Fahrzeugs. Vgl. 7.6.8.
	<i>OnwardCall</i>	0:*	+CallAtNearStop Die noch bevorstehenden Halte der Fahrt. Vgl. 10.3.5.
	<i>Service</i>	0:1	+DatedJourney Angaben zum Verkehrsmittel, Linie etc. Vgl. 7.6.2.
<i>OperatingDays</i>	<i>OperatingDays</i>	0:1	+OperatingDays Verkehrstage für diese Verbindung. Vgl. 7.4.7.
	<i>OperatingDaysDescription</i>	0:1	+InternationalText Menschenlesbare Beschreibung der Verkehrstage, z. B. „Montag bis Freitag“ oder „Sonn- und Feiertag“.
<i>ServiceFacility</i>	:::	0:1	+siri:ServiceFacilityGroup Klassifizierung der Ausstattungsmerkmale. Vgl. 7.7.3.
	<i>Extension</i>	0:1	xs:anyType Erweiterungen.

Tabelle 122: Beschreibung der Struktur ***TripInfoResultStructure***.

13. Anschlussdienste

13.1. Beschreibung

Unter dem Begriff „Anschlussdienste“ werden unterschiedliche Dienste des TRIAS-Standards zusammengefasst, die der Kommunikation zu Anschlüssen dienen. Die Anschlussdienste setzen sich aus den Diensten

- Anschlussmeldung,
- Anschlussstatus,
- Info zu Anschlussverlust und
- Anschlussrückmeldung

zusammen. Nachfolgend sind zwei Abläufe dokumentiert, die eine mögliche Nutzung der Dienste darstellen.

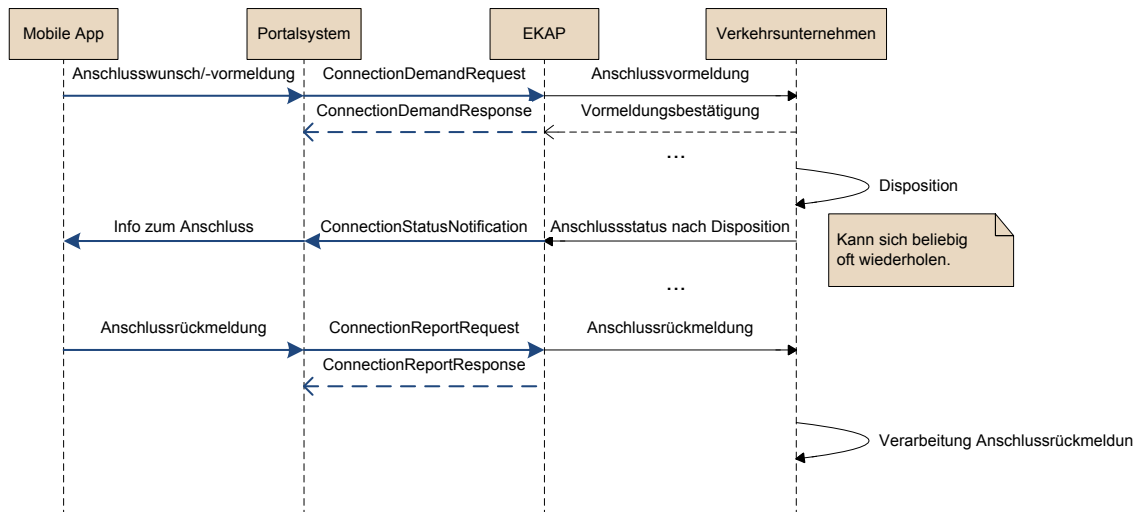


Abbildung 7: Ablauf der Anschlussdienste mit aktiver Benachrichtigung bei Statusänderung

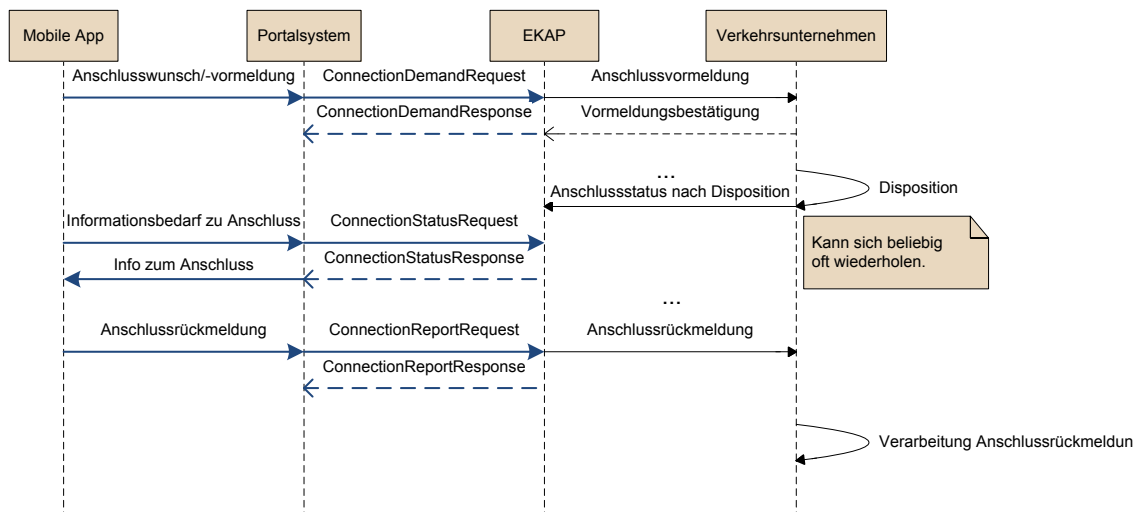


Abbildung 8: Ablauf der Anschlussdienste mit passiver Benachrichtigung bei Statusänderung

Der Ablauf gestaltet sich im Allgemeinen folgendermaßen:

1. Anschlussmeldung durch Reisenden/Applikation, Zugbegleiter oder System über das Portalsystem und die EKAP an das/die beteiligte/n Verkehrsunternehmen
 2. Verarbeitung der Anschlussmeldung durch Verkehrsunternehmen und Anschlussdisposition
 3. Information über Dispositionsmaßnahme
 - a. aktive Information durch EKAP (Abbildung 7)
 - b. passive Information mittels Anfrage an die EKAP (Abbildung 8)
 4. Rückmeldung des Reisenden oder seiner Applikation zur Anchlusserreichung.
- Der Ablauf ist nicht zwingend vorgegeben. Insbesondere kann es sinnvoll sein, einzelne Dienste ohne Bezug zu den anderen zu nutzen. Beispielsweise ist die Abfrage des Anschlussstatus durch die Verbindungsauskunft denkbar. Die Dienste sind genauer in den folgenden Unterkapiteln beschrieben.

13.1.1. Dienst Anschlussvoranmeldung

Mit Hilfe dieses Dienstes können Reisende ihre Anschlusswünsche mitteilen. Auf diese Weise werden Dispositionsverantwortliche in die Lage versetzt, Umsteigerzahlen abzuschätzen und in der Entscheidungsfindung einer Anschlussdisposition zu berücksichtigen. Der Mehrwert für die Reisenden ist entsprechend eine verbesserte Anschlussdisposition.

Anschlussbeziehungen umfassen dabei allerdings nicht nur ein Zubringer/Abbringer-Paar, sondern schließen auch Anschlussbeziehungen von einem Startort auf einen Abbringer mit ein. Der Abbringer wiederum kann ein normal verkehrendes Angebot sein, aber auch ein Anrufsammeltaxi (AST) oder Bedarfsverkehr mit fester oder variabler Linienführung und festen oder variablen Halten.

Somit kann und soll die Anschlussmeldung auch als Bestellung für einen Bedarfsverkehr eingesetzt werden.

Der Dienst übermittelt unterschiedliche Grade der Wahrscheinlichkeit, mit der ein Nutzer die gewählte Verbindung nimmt. Auf diese Weise können auch nicht sicher gewählte Verbindungen anhand der Wahrscheinlichkeiten für die Disposition verwendet werden.

Weiterhin besteht die Möglichkeit für die beteiligten Verkehrsunternehmen auf Basis der gemeldeten Umsteiger Rückschlüsse auf die Anzahl der Reisenden im Fahrzeug zu schließen und entsprechende Kapazitäten zu disponieren. Dies gilt insbesondere für die Einstiegs- und Ausstiegsmeldungen.

13.1.2. Dienst Anschlussstatus

Der Dienst Anschlussstatus ermöglicht es den Verkehrsunternehmen, andere Beteiligte über den Status einer Anschlussbeziehung (erwartetes Zustandekommen des Anschlusses) zu informieren.

In erster Linie dient dies der Kundeninformation. Aufgrund der Information über den Anschlussstatus wissen ein Kunde und auch seine Applikation, ob er seine Reisekette (auch bei Verspätung des Zubringers) in der geplanten Weise fortsetzen kann. Seine Applikation kann entsprechend reagieren und Alternativen suchen.

Auch andere Verkehrsunternehmen können Nutzer dieser Information sein. Sie können auf eine Disposition reagieren und von sich aus weitere Maßnahmen im Fall eines abgelehnten Anschlusses einleiten. Ferner lassen sich Prognosen zu Reisendenströmen aufgrund der aktuellen Verkehrslage stellen.

Zur sinnvollen Nutzung des Dienstes ist es erforderlich, dass die Betriebsleitsysteme Anschlussstatusinformationen liefern, sobald sie bekannt werden. Das kann durch eine Dispositionshandlung des Disponenten geschehen oder implizit durch Einflüsse des Betriebsablaufes. Das Betriebsleitsystem meldet die Anschlussinformationen an die Datendrehscheibe einer oder mehrerer EKAPs. Dort kann das Benachrichtigungssystem auf diese Daten zugreifen.

Der Anschlussstatus kann auch im Rahmen einer Verbindungsüberwachung durch den Benachrichtigungsdienst übermittelt werden (siehe Kapitel 19).

13.1.3. Dienst Info bei Anschlussverlust

Durch das Nicht-Zustandekommen eines Anschlusses kann ein Reisender nicht mehr seine ursprünglich geplante Reisekette wahrnehmen. Mit diesem Dienst kann ein Verkehrsunternehmen auf alternative Abbringer, auf die Bestellung von Taxen, Bussen oder Hotelzimmern, die Bereitstellung von Ersatz- oder Sonderfahrten, Umleitungen oder eine Kombination aus unterschiedlichen Maßnahmen verweisen. Die App des Fahrgastes kann anhand der vorgeschlagenen Alternativen prüfen, ob es sich um einen für den Fahrgast sinnvollen Vorschlag handelt und ihn im positiven Fall in die Suche nach Alternativen mit einbeziehen.

Der Dienst ist in die Antworten des Anschlussstatus integriert.

13.1.4. Dienst Anschlussrückmeldung

Mit Hilfe dieses Dienstes können Reisende das Transportunternehmen darüber informieren, ob ein Anschluss aus Sicht des Reisenden erfolgreich disponiert wurde bzw. erfolgreich zustande gekommen ist. Dazu sendet der Reisende eine Nachricht mit einem Anschluss, bestehend aus Zu- und Abbringer und einer Information, ob der Anschluss für ihn zustande kam.

Die Übermittlung der Anschlussrückmeldung kann aber auch automatisch durch die Applikation erfolgen, wenn beispielsweise anhand einer geplanten Verbindung bekannt ist, welche Umstiegsverbindungen genutzt werden sollen. Diese können durch die Applikation überprüft werden, und es kann ein automatisches Feedback bei erkanntem Anschluss oder Anschlussbruch erfolgen. Ein weiteres Beispiel ist mit dem automatischen Erkennen der Fahrzeuge verbunden. Mit Hilfe dieser Funktion kann erkannt werden, wann das Fahrzeug gewechselt wird und dazu eine Anschluss Erfolgsmeldung versendet werden.

13.2. Einfache Typen

Folgende einfache Typen (simple types) werden definiert:

<i>ConnectionStatusEnumeration</i>	<i>unknown planned confirmed broken</i>	Klassifizierung der Verbindungsstatus.
<i>RecommendationTypeEnumeration</i>	<i>NextService DifferentRoute Hotel Taxi Bus Helpdesk Hotline Driver Other</i>	Klassifizierung der Ausweichempfehlungen bei Verbindungsverlust.

13.3. Komplexe Strukturen

Im Folgenden werden die komplexen Strukturen beschrieben, die in Trias_Connections definiert sind.

13.3.1. DatedConnectionStructure

<i>DatedConnectionStructure</i>		<i>+Structure</i>	Beinhaltet einen Zubringer und einen Abbringer für einen konkreten Betriebstag.
	ConnectionId	1:1	<i>xs:NMTOKEN</i> ID des Anschlusses für spätere Referenzierung bzw. für Debug-Zwecke.
	Feeder	1:1	<i>+FeederDistributor</i> Zubringer des gemeldeten Anschlusswunsches, vgl. 13.3.2.

	Distributor	1:1	+FeederDistributor	Abbringer des gemeldeten Anschlusswunsches, vgl. 13.3.2.
--	--------------------	-----	--------------------	--

Tabelle 123: Beschreibung der Struktur **DatedConnectionStructure**

13.3.2. FeederDistributorStructure

FeederDistributorStructure			+Structure	Beinhaltet einen Zubringer oder einen Abbringer an einem definierten Ort zu einer definierten Betriebszeit.
<i>DatedJourneyRef</i>	JourneyRef	1:1	→Journey	Referenz auf die Fahrt des Zubringers oder Abbringers. Vgl. 7.4.1.
	OperatingDayRef	1:1	→OperatingDay	Referenz auf den Betriebstag. Vgl. 7.4.1.
	ConnectionLocation	1:1	+CallAtStop	Ort des Anschlusses, vgl. 7.6.5.

Tabelle 124: Beschreibung der Struktur **FeederDistributorStructure**

13.3.3. GeneralizedConnectionStructure

GeneralizedConnectionStructure			+Structure	Definiert eine Umstiegsrelation. Enthält die Verbindungsarten Einstieg (Pickup), Ausstieg (SetDown) und Umstieg (DatedConnection)	
	a	DatedConnection	-1:1	+DatedConnection	Umstiegsverbindung, für die der Status abgefragt werden soll. Vgl. 13.3.1.
	b	PickUpLocation		+DatedCallAtLocation	Zustieg in ÖV-Mittel, für den der Status abgefragt werden soll. Vgl. 7.6.6.
	c	SetDownLocation		+DatedCallAtLocation	Ausstieg aus ÖV-Mittel, für den der Status abgefragt werden soll. Vgl. 7.6.6.

Tabelle 125: Beschreibung des Typs **GeneralizedConnectionStructure**

13.3.4. ConnectionStatusStructure

ConnectionStatusStructure			+Structure	Enthält den tatsächlichen Verbindungsstatus. Dieser besteht aus einem Ein-, Aus- oder Umstieg und einem zugehörigen Status.
	Connection	1:1	+GeneralizedConnection	Ein-, Aus- oder Umstieg. Vgl. 13.3.3.
	Status	1:1	ConnectionStatusEnumeration	Verbindungsstatus. Vgl. 13.2.
	<i>Alternative</i>	0:*	+Recommendation	Alternativen nach Verbindungsbruch, vgl. 13.3.5.

Tabelle 126: Beschreibung der Struktur **ConnectionStatusStructure**

13.3.5. RecommendationStructure

Dieses Element bildet den Dienst „Info bei Anschlussverlust“ ab.

RecommendationStructure			+Structure	Enthält Alternativenvorschläge bei Anschlussverlust, die über eine bloße Verbindungsalternative hinausgehen. Als Alternative kann neben
--------------------------------	--	--	------------	--

				anderen bestehenden Fahrten auf bereitgestellte Ersatzverkehre, Übernachtungsmöglichkeiten oder andere, durch Disposition neu zu schaffende, Möglichkeiten hingewiesen werden.
	RecommendationId	1:1	xs:NMTOKEN	ID der Anfrage für spätere Referenzierung bzw. für Debug-Zwecke
	Text	1:1	+InternationalText	Beschreibung der Alternative.
	Type	1:1	RecommendationTypeEnum	Typ der Empfehlung, vgl. 13.2.

Tabelle 127: Beschreibung der Struktur *RecommendationStructure*

13.4. Anfragestrukturen Anschlussmeldung

13.4.1. ConnectionDemandRequestStructure

ConnectionDemandRequestStructure			+Structure	Beinhaltet die Anfragedaten für eine Meldung von Umsteigern, Einsteigern oder Aussteigern bei regulären oder Bedarfsverkehren.
	RequestId	1:1	xs:NMTOKEN	ID der Anfrage für spätere Referenzierung bzw. für Debug-Zwecke
	Connection	1:1	+GeneralizedConnection	Anschlussrelation. Vgl. 13.3.3.
	NumberOfPersons	0:1	xs:positiveInteger	Anzahl der Um-, Ein- oder Aussteiger.
	TravelProbability	0:1	Percent	(Kumulierte) Reisewahrscheinlichkeit für die angegebenen Um-, Ein- oder Aussteiger. Vgl. 7.2.1.
	Extension	0:1	xs:anyType	Erweiterungen.

Tabelle 128: Beschreibung der Struktur *ConnectionDemandRequestStructure*.

13.4.2. ConnectionDemandDeleteRequestStructure

ConnectionDemandDeleteRequestStructure			+Structure	Anfragestruktur, um eine frühere Anschlussvoranmeldung zu stornieren.
	RequestId	1:1	xs:NMTOKEN	ID der Anfrage, die storniert werden soll.

Tabelle 129: Beschreibung der Struktur *ConnectionDemandDeleteRequestStructure*.

13.5. Antwortstrukturen Anschlussmeldung

13.5.1. ConnectionDemandResponseStructure

ConnectionDemandResponseStructure			+Structure	Antwort auf eine Anfrage vom Typ <i>ConnectionDemandRequest</i>
	ErrorMessage	0:*	+ErrorMessage	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Vgl. auch 7.4.2.

Tabelle 130: Beschreibung der Struktur *ConnectionDemandResponseStructure*.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

CONNECTIONDEMAND_FEEDER_UNKNOWN	Der Zubringer ist der EKAP nicht bekannt.
CONNECTIONDEMAND_DISTRIBUTOR_UNKNOWN	Der Abbringer ist der EKAP nicht bekannt.
CONNECTIONDEMAND_DEPARTURE_BEFORE_ARRIVAL	Die Sollabfahrt des Abbringers liegt vor der Sollankunft des Zubringers. Umstieg folglich nicht möglich.
CONNECTIONDEMAND_FEEDER_LOCATION_UNKNOWN	Der referenzierte Ort des Umstiegs für den Zubringer ist unbekannt.
CONNECTIONDEMAND_DISTRIBUTOR_LOCATION_UNKNOWN	Der referenzierte Ort des Umstiegs für den Abbringer ist unbekannt.

Tabelle 131: Liste der Fehlerzustände in **ConnectionDemandResponse**.

13.5.2. ConnectionDemandDeleteResponseStructure

ConnectionDemandDeleteResponseStructure		+Structure	Antwort auf eine Anfrage vom Typ ConnectionDemandDeleteRequest .
	<i>ErrorMessage</i>	0:*	+ <i>ErrorMessage</i>
			Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Vgl. auch 7.4.2.

Tabelle 132: Beschreibung der Struktur **ConnectionDemandDeleteResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

CONNECTIONDEMAND_REQUESTID_UNKNOWN	Die Anfrage-ID ist der EKAP nicht bekannt.
CONNECTIONDEMAND_DELETIONNOTPOSSIBLE	Die Anschlussmeldung konnte nicht storniert werden.

Tabelle 133: Liste der Fehlerzustände in **ConnectionDemandDeleteResponse**.

13.6. Anfragestrukturen Anschlussstatus

13.6.1. ConnectionStatusRequestStructure

ConnectionStatusRequestStructure		+Structure	Dient der aktiven Abfrage eines Verbindungsstatus.
	<i>RequestId</i>	1:1	<i>xs:NMTOKEN</i>
	<i>Connection</i>	1:1	+ <i>GeneralizedConnection</i>
	<i>Extension</i>	0:1	<i>xs:anyType</i>
			ID der Anfrage für spätere Referenzierung bzw. für Debug-Zwecke.
			Anschlussrelation. Vgl. 13.3.3.
			Erweiterungen.

Tabelle 134: Beschreibung der Struktur **ConnectionStatusRequestStructure**

13.6.2. ConnectionStatusNotificationStructure

ConnectionStatusNotificationStructure	+Structure	Push-Information zu Anschlussstatus. Genutzt von <i>TripMonitoring</i> oder optional aktiv gesendet von EKAP nach <i>ConnectionDemands</i> .
--	------------	--

	ConnectionStatus	1:1	+ConnectionStatus	Enthält den tatsächlichen Verbindungsstatus. Vgl. 13.3.4.
--	-------------------------	-----	-------------------	---

Tabelle 135: Beschreibung der Struktur **ConnectionStatusNotificationStructure**

13.7. Antwortstrukturen Anschlussstatus

13.7.1. ConnectionStatusResponseStructure

ConnectionStatusResponseStructure			+Structure	Liefert den Verbindungsstatus oder eine Fehlermeldung auf die Anfrage ConnectionStatusRequest.
	ErrorMessage	0:*	+ErrorMessage	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Vgl. 7.4.2.
	ConnectionStatus	0:1	+ConnectionStatus	Enthält den tatsächlichen Verbindungsstatus. Vgl. 13.3.4.

Tabelle 136: Beschreibung der Struktur **ConnectionStatusResponseStructure**

In *ErrorMessage* können folgende Fehlerzustände auftreten:

CONNECTIONSTATUS_FEEDER_UNKNOWN	Der Zubringer ist der EKAP nicht bekannt.
CONNECTIONSTATUS_DISTRIBUTOR_UNKNOWN	Der Abbringer ist der EKAP nicht bekannt.
CONNECTIONSTATUS_DEPARTURE_BEFORE_ARRIVAL	Die Sollabfahrt des Abbringers liegt vor der Sollankunft des Zubringers. Umstieg folglich nicht möglich.
CONNECTIONSTATUS_FEEDER_LOCATION_UNKNOWN	Der referenzierte Ort des Umstiegs für den Zubringer ist unbekannt.
CONNECTIONSTATUS_DISTRIBUTOR_LOCATION_UNKNOWN	Der referenzierte Ort des Umstiegs für den Abbringer ist unbekannt.

Tabelle 137: Liste der Fehlerzustände in **ConnectionStatusResponseStructure**.

13.8. Anfragestrukturen Anschlussrückmeldung

13.8.1. ConnectionReportRequestStructure

ConnectionReportRequestStructure			+Structure	Beinhaltet die Daten für eine Meldung, die besagt, ob der Anschluss für den Reisenden zustande gekommen ist.
	RequestId	1:1	xs:NMTOKEN	ID der Anfrage für spätere Referenzierung bzw. für Debug-Zwecke.
	Connection	1:1	+GeneralizedConnection	Anschlussrelation, über die berichtet wird. Vgl. 13.3.3.
	Succeeded	1:1	xs:boolean	Gibt an, ob der Anschluss aus Sicht des Reisenden zustande gekommen ist.

	<i>Reason</i>	0:1	<i>+International Text</i>	Optional ist die Angabe eines Grundes für den Anschlussbruch oder das Erreichen des Anschlusses möglich.
	<i>Extensions</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 138: Beschreibung der Struktur **ConnectionReportRequestStructure**.

13.9. Antwortstrukturen Anschlussrückmeldung

13.9.1. ConnectionReportResponseStructure

ConnectionReportResponseStructure		<i>+Structure</i>	Antwort auf eine Anfrage vom Typ ConnectionReportRequest
	<i>ErrorMessage</i>	0:*	<i>+ErrorMessage</i>
			Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Vgl. 7.4.2.

Tabelle 139: Beschreibung der Struktur **ConnectionReportResponseStructure**

In *ErrorMessage* können folgende Fehlerzustände auftreten:

CONNECTIONREPORT_FEEDE-PORT_FEEDER_UNKNOWN	Der Zubringer ist der EKAP nicht bekannt.
CONNECTIONREPORT_DISTRI-BUTOR_UNKNOWN	Der Abbringer ist der EKAP nicht bekannt.
CONNECTIONREPORT_DEPART-URE_BEFORE_ARRIVAL	Die Sollabfahrt des Abbringers liegt vor der Sollankunft des Zubringers. Umstieg folglich nicht möglich.
CONNECTIONREPORT_FEEDE-R_LOCATION_UNKNOWN	Der referenzierte Ort des Umstiegs für den Zubringer ist unbekannt.
CONNECTIONREPORT_DISTRI-BUTOR_LOCATION_UNKNOWN	Der referenzierte Ort des Umstiegs für den Abbringer ist unbekannt.

Tabelle 140: Liste der Fehlerzustände in **ConnectionReportResponse**.

14. Dienst Fahrpreis- und Tarifberechnung

14.1. Beschreibung

Dieser Dienst stellt allgemeine, haltestellenbezogene oder verbindungsbezogene Tarifinformationen bereit.

14.2. Anfragestrukturen

Eine Anfrage an den Tarifberechnungsdienst wird mittels eines Elements **FaresRequest** vom Typ *FaresRequestStructure* gestellt.

14.2.1. FaresRequestStructure

FaresRequestStructure		<i>+Structure</i>	Fasst die Daten für eine Tarifierfrage zusammen.
a	StopFaresRequest	-1:1	<i>+StopFaresRequest</i>
b	StaticFaresRequest		<i>+StaticFaresRequest</i>
c	TripFaresRequest		<i>+TripFaresRequest</i>
			Haltestellenbezogene Tarifierfrage. Vgl. 14.2.2.
			Allgemeine Tarifierfrage. Vgl. 14.2.3
			Verbindungsbezogene Tarifierfrage. Vgl. 14.2.3..

			<i>quest</i>	
d	MultiTripFaresRequest		+MultiTripFaresRequest	Aggregierte Tarifrage für mehrere Verbindungen. Vgl. 14.2.5.
	<i>Params</i>	0:1	+ FaresParam	Parameter für Tarifrage. Vgl. 7.10.7.
	<i>Extension</i>	0:1	xs:anyType	Erweiterungen.

Tabelle 141: Beschreibung der Struktur **FaresRequestStructure**.

14.2.2. StopFaresRequestStructure

Die haltestellenbezogene Tarifrage ermittelt Tarifrinformationen, die für eine bestimmte Haltestelle gelten, z. B. die Tarifzonen, in denen die Haltestelle liegt.

StopFaresRequestStructure			+Structure	Fasst die Daten für eine haltestellenbezogene Tarifrfrage zusammen.
	StopPointRef	1:1	→StopPointCode	Referenziert den Haltepunkt. Vgl. 7.5.1.
	<i>Date</i>	0:1	xs:date	Stichtag für die Gültigkeit der Tarifauskunft.

Tabelle 142: Beschreibung der Struktur **StopFaresRequestStructure**.

14.2.3. StaticFaresRequestStructure

Die statische Tarifrfrage ermittelt allgemeine Tarifrinformationen wie z. B. eine Liste der erhältlichen Fahrscheinarten oder eine URL auf weiterführende Tarifrinformationen (z. B. Tarifzonenpläne, Tarifbestimmungen etc.).

StaticFaresRequestStructure			+Structure	Fasst die Daten für eine allgemeine (statische) Tarifrfrage zusammen.
	<i>Date</i>	0:1	xs:date	Stichtag für die Gültigkeit der Tarifauskunft.

Tabelle 143: Beschreibung der Struktur **StaticFaresRequestStructure**.

14.2.4. TripFaresRequestStructure

Die verbindungsbezogene Tarifrfrage ermittelt zu einer bestimmten Verbindung die in Frage kommenden Fahrschein und ihre Preise.

TripFaresRequestStructure			+Structure	Fasst die Daten für eine verbindungsbezogene Tarifrfrage zusammen.
	Trip	1:1	+Trip	Beinhaltet die Verbindung, für die die Tarifauskunft zu ermitteln ist. Vgl. 9.3.4.

Tabelle 144: Beschreibung der Struktur **TripFaresRequestStructure**.

14.2.5. MultiTripFaresRequestStructure

Der Unterschied von *MultiTripFaresRequestStructure* zu *TripFaresRequestStructure* besteht darin, dass der Server bei *MultiTripFaresRequestStructure* aufgefordert ist, eine möglichst günstige Ticketkombination herauszufinden, die die Verbindungen abdeckt, also z. B. eine Tageskarte, wenn genügend Verbindungen am gleichen Tag stattfinden sollen.

MultiTripFaresRequestStructure			+Structure	Fasst die Daten für eine Tarifrage für mehrere Verbindungen zusammen.
	<i>Trip</i>	1:*	+Trip	Beinhaltet die Verbindungen, für die die Tarifauskunft zu ermitteln ist. Vgl. 9.3.4.

Tabelle 145: Beschreibung der Struktur **MultiTripFaresRequestStructure**.

14.3. Antwortstrukturen

Das Ergebnis einer Tarifrage wird mittels eines Elements **FaresResponse** vom Typ **FaresResponseStructure** übertragen.

14.3.1. FaresResponseStructure

FaresResponseStructure			+Structure	Fasst die Ergebnisdaten für eine Tarifrage zusammen.
	<i>ErrorMessage</i>	0:*	+ErrorMessage	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
	<i>FaresResult</i>	0:*	+FaresResult	Struktur für ein Tarifergebnis. Vgl. 14.3.2.

Tabelle 146: Beschreibung der Struktur **FaresResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

FARES_DATEOUTOFRANGE	Die Tarifrage kann nicht bearbeitet werden, weil für das gewünschte Datum keine Informationen zur Verfügung stehen.
FARES_STOPPOINTUNKNOWN	Die Tarifrage kann nicht bearbeitet werden, weil der angefragte Haltepunkt unbekannt ist.

Tabelle 147: Liste der Fehlerzustände in **FaresResponse**.

14.3.2. FaresResultStructure

FaresResultStructure			+Structure	Resultatstruktur für die Buchungsinformationen.
	ResultId	1:1	xs:NMTOKEN	ID des Resultats für spätere Referenzierung.
a	StopFaresResult	-1:1	+StopFaresResult	Antwort auf haltestellenbezogene Tarifrage. Vgl. 14.3.3.
b	StaticFaresResult		+StaticFaresResult	Antwort auf allgemeine Tarifrage. Vgl. 14.3.4.
c	TripFaresResult		+TripFaresResult	Antwort auf verbindungsbezogene Tarifrage. Vgl. 7.10.5.
d	MultiTripFaresResult		+MultiTripFaresResult	Antwort auf Tarifrage für mehrere Verbindungen. Vgl. 14.3.6.

Tabelle 148: Beschreibung der Struktur **FaresResultStructure**.

14.3.3. StopFaresResultStructure

StopFaresResultStructure			+Structure	Resultatstruktur für haltestellenbezogene Tarifinformationen.
	FareZoneListInArea	1:*	+FareZoneListInArea	Liste der Tarifzonen, in denen die angefragte Haltestelle liegt. Vgl. 7.10.2.
	<i>Extension</i>	0:1	xs:anyType	Erweiterungen.

Tabelle 149: Beschreibung der Struktur **StopFaresResultStructure**.

14.3.4. StaticFaresResultStructure

StaticFaresResultStructure		+Structure	Resultatstruktur für allgemeine Tarifinformationen.
<i>Ticket</i>	0:*	+ <i>Ticket</i>	Liste der verfügbaren Tickets. Vgl. 7.10.4.
<i>StaticInfoUrl</i>	0:1	<i>xs:anyURI</i>	Links auf Informationsseiten im Web.
<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 150: Beschreibung der Struktur **StaticFaresResultStructure**.

14.3.5. TripTicketReferenceStructure

TripTicketReferenceStructure		+Structure	Verknüpfung von Tickets mit Verbindungen (oder Teilen davon).
TicketRef	1:1	→ <i>TicketCode</i>	Referenz auf ein Ticket.
FromTripIdRef	1:1	<i>xs:NMTOKEN</i>	Referenz auf eine Verbindung, ab der ein Ticket gültig ist.
<i>FromTripLegIdRef</i>	0:1	<i>xs:NMTOKEN</i>	Referenz auf einen Teilweg, ab dem das Ticket gültig ist.
ToTripIdRef	1:1	<i>xs:NMTOKEN</i>	Referenz auf eine Verbindung, bis zu der ein Ticket gültig ist.
<i>ToTripLegIdRef</i>	0:1	<i>xs:NMTOKEN</i>	Referenz auf einen Teilweg, bis zu dem das Ticket gültig ist.

Tabelle 151: Beschreibung der Struktur **TripTicketReferenceStructure**.

14.3.6. MultiTripFaresResultStructure

MultiTripFaresResultStructure		+Structure	Fasst die Ergebnisdaten für eine Tarifauskunft zu mehreren Verbindungen zusammen.
<i>ErrorMessage</i>	0:*	+ <i>ErrorMessage</i>	Fehlermeldungen bezogen auf diese Tarifauskunft. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. auch 7.4.2.
TripTicketReference	1:*	+ <i>TripTicketReference</i>	Verknüpfung von Tickets mit Verbindungen (oder Teilen davon). Vgl. 14.3.5.
<i>Ticket</i>	0:*	+ <i>Ticket</i>	Fahrscheine, die auf diesem Abschnitt der Verbindung gültig sind (vgl. 7.10.4).
<i>PassedZones</i>	0:1	+ <i>FareZoneListInArea</i>	Die Menge der durchfahrenen Tarifzonen, über alle Verbindungen gesehen (vgl. 7.10.2).
<i>StaticInfoURL</i>	0:*	<i>xs:anyURI</i>	URL zu Informationsseiten.

Tabelle 152: Beschreibung der Struktur **MultiTripFaresResultStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

FARES_OUTOFAREA	Der in der Verbindungsauskunft gefundene Weg verlässt das Tarifgebiet.
FARES_JOURNEYNOTPERMITTED	Ein in der Verbindungsauskunft benutztes Verkehrsmittel ist für den Tarif nicht zulässig.
FARES_ADDITIONALCHARGES	Zusätzliche Gebühren sind voraussichtlich zu entrichten (z. B. Mautzuschläge oder Reservierungsgebühren).
FARES_ADDITIONALTICKETS	Zusätzliche Fahrscheine sind notwendig, da nicht für alle Verkehrsmittel oder für alle angegebenen Verbindungen ein passender Fahrschein ermittelt werden konnte.

FARES_ROUTENOTFEASIBLE	Es kann kein Fahrschein ermittelt werden, weil die Route der Verbindungsauskunft nicht in Einklang mit den Tarifregeln steht (z. B. wegen Rundreisen, Stichfahrten oder Überschreitung der zulässigen Gesamtdauer).
-------------------------------	---

Tabelle 153: Liste der Fehlerzustände in *MultiTripFaresResultStructure*.

15. Dienst Buchungsinformationen

15.1. Beschreibung

Der Dienst Buchungsinformation stellt Informationen zur Verfügung, mit deren Hilfe Kontakt zu einem Buchungssystem hergestellt werden kann. Das zuständige Buchungssystem kann für ein Verkehrsunternehmen oder für eine einzelne ÖV-Fahrt abgefragt werden. Ein Buchungssystem führt z. B. die Vorbestellung eines Bedarfsverkehrs, eine Sitzplatzreservierung oder auch den Kauf eines Fahrscheins durch.

15.2. Anfragestrukturen

Eine Buchungsinformation wird mittels eines Elements **BookingInfoRequest** vom Typ *BookingInfoRequestStructure* angefordert.

15.2.1. BookingInfoRequestStructure

In einer *BookingInfoRequestStructure* kann wahlweise eine bestimmte ÖV-Fahrt oder ein Verkehrsunternehmen angegeben werden, für die die Buchungsinformation ermittelt werden soll.

BookingInfoRequestStructure		+Structure	Fasst die Anfragedaten nach Buchungsinformationen zusammen.
a	Service	-1:1	+DatedJourney
b	OperatorRef		→Operator
	Extension	0:1	xs:anyType
			Definition einer ÖV-Fahrt an einem bestimmten Tag. Vgl. 7.6.2.
			Referenz auf ein Verkehrsunternehmen. Vgl. 7.4.1.
			Erweiterungen.

Tabelle 154: Beschreibung der Struktur **BookingInfoRequestStructure**.

15.3. Antwortstrukturen

Das Ergebnis einer Buchungsinformationsanfrage wird mittels eines Elements **BookingInfoResponse** vom Typ *BookingInfoResponseStructure* übertragen.

15.3.1. BookingInfoResponseStructure

BookingInfoResponseStructure		+Structure	Fasst die Ergebnisdaten für eine Buchungsinformationsanfrage zusammen.
	ErrorMessage	0:*	+ErrorMessage
	BookingInfoResult	0:1	+BookingInfoResult
			Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
			Struktur für ein Buchungsinformationsresultat. Vgl. 15.3.2.

Tabelle 155: Beschreibung der Struktur **BookingInfoResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

BOOKINGINFO_DATEINVALID	Zum angegebenen Datum liegen keine Informationen vor.
BOOKINGINFO_VEHICLEUNKNOWN	Das angegebene Fahrzeug ist unbekannt.
BOOKINGINFO_OPERATORUNKNOWN	Das angegebene Verkehrsunternehmen ist unbekannt.
BOOKINGINFO_JOURNEYUNKNOWN	Die angegebene Fahrt ist unbekannt.
BOOKINGINFO_LINEUNKNOWN	Die angegebene Linie ist unbekannt.
BOOKINGINFO_MODEUNKNOWN	Die angegebene Verkehrsmittelart ist unbekannt.
BOOKINGINFO_NOINFORMATION	Es liegt keine passende Information vor.

Tabelle 156: Liste der Fehlerzustände in **BookingInfoResponse**.

15.3.2. BookingInfoResultStructure

BookingInfoResultStructure		+Structure	Resultatstruktur für die Buchungsinformationen.
<i>BookingInfo</i>	1:*	+BookingInfo	Container für die Buchungsinformationen. Vgl. 7.10.3.
<i>Extension</i>	0:1	xs:anyType	Erweiterungen.

Tabelle 157: Beschreibung der Struktur **BookingInfoResultStructure**.

16. Dienst IV-Routing

In der XML-Schema-Definition *Trias_IndividualTrips.xsd* werden Datentypen und Strukturen definiert, die für ein IV-Routing verwendet werden.

16.1. Einfache Typen

Folgender einfacher Typ (simple type) wird definiert:

<i>IndividualTripsAlgorithmTypeEnumeration</i>	<i>fastest shortest beautiful optimal economic</i>	Algorithmus-Typ für die Berechnung von IV-Routen.
--	--	---

Tabelle 158: Liste der einfachen Typdefinitionen in *Trias_IndividualTrips.xsd*.

16.2. Anfragestrukturen

Eine Route im Individualverkehr wird mittels eines Elements **IndividualRouteRequest** vom Typ *IndividualRouteRequestStructure* angefordert.

16.2.1. IndividualRouteRequestStructure

IndividualRouteRequestStructure		+Structure	Fasst die Anfragedaten für ein IV-Routing zusammen.
<i>Origin</i>	1:1	+IndividualRouteLocationContext	Ortsdaten für den Abfahrtsort. Vgl. 16.2.3.
<i>Destination</i>	1:1	+IndividualRo	Ortsdaten für den Zielort. Vgl. 16.2.3.

			<i>uteLocation-Context</i>	
	<i>Via</i>	0:*	+ <i>Via</i>	Ein oder mehrere Via-Orte. Die angegebenen Via-Orte müssen in der vorgegebenen Reihenfolge erreicht werden. Der Server darf eine Via-Haltestelle durch eine äquivalente Haltestelle ersetzen. Vgl. 7.6.2.
	Mode	1:*	+ <i>IndividualTransportOptions</i>	IV-Modi, für die eine IV-Route ermittelt werden soll. Für jeden IV-Modus können weitere steuernde Parameter angegeben werden. Vgl. 7.3.2.
	<i>Params</i>	0:1	+ <i>IndividualTripParam</i>	Parameter, die die Suche und Rückgabewerte beeinflussen können. Vgl. 16.2.2.

Tabelle 159: Beschreibung der Struktur **IndividualRouteRequestStructure**.

16.2.2. IndividualTripParamStructure

IndividualTripParamStructure			+ <i>Structure</i>	Fasst die Parameter zusammen, die die IV-Routen-Suche und Rückgabewerte beeinflussen können. Diese Parameter gelten für alle IV-Modi, für die das IV-Routing durchgeführt werden soll. Sollen unterschiedliche Parametersätze für unterschiedliche IV-Modi verwendet werden, müssen mehrere unabhängige IV-Routensuchen durchgeführt werden.	
<i>BaseTripMobilityFilter</i>	<i>NoSingleStep</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer eine Stufe bewältigen kann. Falls nein, wird dieser Parameter gesetzt.	
	<i>NoStairs</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer eine Treppe bewältigen kann. Falls nein, wird dieser Parameter gesetzt.	
	<i>NoEscalator</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer eine Rolltreppe benutzen kann. Falls nein, wird dieser Parameter gesetzt.	
	<i>NoElevator</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer einen Aufzug benutzen kann. Falls nein, wird dieser Parameter gesetzt.	
	<i>NoRamp</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer eine Rampe bewältigen kann. Falls nein, wird dieser Parameter gesetzt.	
<i>BaseTripPolicy</i>					
	a	<i>NumberOfResults</i>	-0:1	<i>xs:positiveInteger</i>	Anzahl der Verbindungsauskünfte, die der Benutzer mindestens erwartet.
	b	:::	-0:1	<i>NumberOfResultsGroup</i>	Angabe der gewünschten Verbindungen vor/nach dem angegebenen Zeitpunkt am Start oder Ziel (vgl. 9.2.3).
		<i>IgnoreRealtimeData</i>	0:1	<i>xs:boolean</i>	Wenn dieser Parameter gesetzt ist, sollen in der Verbindungssuche keine Echtzeitdaten oder Störungsinformationen sondern nur Sollfahrplandaten berücksichtigt werden.
		<i>ImmediateTripStart</i>	0:1	<i>xs:boolean</i>	Wenn dieser Parameter gesetzt ist, soll die zu suchende Verbindung unmittelbar an der angegebenen Startsituation beginnen. Eine Optimierung der Abfahrtszeit am Start nach der Regel „Starte so spät wie möglich, solange nur die gleiche Ankunftszeit am Ziel gewährleistet ist“ ist dann nicht notwendig.
<i>IndividualTripPolicy</i>		<i>AlgorithmType</i>	0:1	<i>fastest shortest beautiful optimal economic</i>	Art der Zielfunktion, nach der der Routing-Algorithmus die Route optimieren soll. Falls nicht vorgegeben, benutzt der Dienst seine eigene Voreinstellung.
		<i>BanMotorways</i>	0:1	<i>xs:boolean</i>	Falls gesetzt, sollen im IV-Routing keine Autobahnen genutzt werden. Default ist <i>false</i> .
		<i>BanTollRoads</i>	0:1	<i>xs:boolean</i>	Falls gesetzt, sollen im IV-Routing keine Mautstraßen genutzt werden. Default ist <i>false</i> .
		<i>BanFerries</i>	0:1	<i>xs:boolean</i>	Falls gesetzt, sollen im IV-Routing keine Fähren

				oder Schiffe genutzt werden. Default ist <i>false</i> .
	<i>BanTunnels</i>	0:1	<i>xs:boolean</i>	Falls gesetzt, sollen im IV-Routing keine Tunnel (Unterführungen aber schon) genutzt werden. Der IV-Routendienst legt den Unterschied zwischen Tunnel und Unterführung selbst fest. Default ist <i>false</i> .
	<i>BanBridges</i>	0:1	<i>xs:boolean</i>	Falls gesetzt, sollen im IV-Routing keine größeren Brücken (Überführungen aber schon) genutzt werden. Der IV-Routendienst legt selbst fest, welche Brücken als „größer“ gelten. Default ist <i>false</i> .
	<i>AllowUnpavedRoads</i>	0:1	<i>xs:boolean</i>	Falls gesetzt, dürfen im IV-Routing auch nicht-befestigte Straßen genutzt werden, sonst nicht. Default ist <i>false</i> .
<i>BaseTripContentFilter</i>	<i>IncludeTrackSections</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat TrackSection-Elemente (vgl. 9.3.13) für die detaillierte geographische Beschreibung des Wegs mitausgegeben werden sollen. Default ist <i>false</i> .
	<i>IncludeLegProjection</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat der detaillierte geographische Verlauf des Wegs als Koordinatenfolge mitausgegeben werden soll. Default ist <i>false</i> .
	<i>IncludeTurnDescription</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Routenhinweise mit Abbiegeempfehlungen mitausgegeben werden sollen. Default ist <i>false</i> .
	<i>IncludeAccessibility</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Informationen zur Barrierefreiheit mitausgegeben werden sollen. Default ist <i>false</i> .
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 160: Beschreibung der Struktur **IndividualTripParamStructure**.

16.2.3. IndividualRouteContextStructure

Elemente vom Typ *IndividualRouteContextStructure* werden vor allem dazu benutzt, um den Start- bzw. Ziel-Kontext zu beschreiben, der für den Reisenden am Anfang bzw. Ende seiner Reise angenommen werden soll. Beim Dienst IV-Routing dienen Elemente dieses Typs als Angabe von Start und Ziel. Die Implementierung des Suchalgorithmus ist dabei selbst dafür zuständig, die Ortsangaben, etwa eine Koordinate, auf die internen Elemente (z. B. Knoten oder Kanten) des Such-Netzwerks abzubilden.

IndividualRouteContextStructure		<i>+Structure</i>	Ortsangabe für Start- oder Zielorte von IV-Routen.
	<i>LocationRef</i>	1:1	<i>+LocationRef</i>
	<i>DepArrTime</i>	0:1	<i>xs:dateTime</i>
			Referenz auf ein Ortsobjekt. Vgl. 7.5.8.
			Abfahrts- bzw. Ankunftszeit.

Tabelle 161: Beschreibung der Struktur **IndividualRouteContextStructure**.

16.3. Antwortstrukturen

Das Ergebnis einer IV-Routing-Anfrage wird mittels eines Elements **IndividualRouteResponse** vom Typ *IndividualRouteResponseStructure* übertragen.

16.3.1. IndividualRouteResponseStructure

IndividualRouteResponseStructure		<i>+Structure</i>	Fasst die Ergebnisdaten für eine IV-Routing-Auskunft zusammen.
	<i>ErrorMessage</i>	0:*	<i>+ErrorMessage</i>
			Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. auch 7.4.2.

	<i>IndividualRouteResponseContext</i>	0:1	<i>+TripResponseContext</i>	Container für Daten, die in der Antwort mehrfach auftreten und referenziert werden. Vgl. 9.3.2.
	<i>RouteResult</i>	0:*	<i>+RouteResult</i>	Container für eine Verbindungsauskunft. Vgl. 16.3.2.

Tabelle 162: Beschreibung der Struktur **IndividualRouteResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

ROUTE_NOROUTEFOUND	Zu den angegebenen Start- und Zielorten, der gewünschten Abfahrts- oder Ankunftszeit sowie unter Berücksichtigung der gegebenen Parameter konnte keine IV-Route gefunden werden.
ROUTE_ORIGINUNKNOWN	Der angegebene Ort (Adresse, Haltestelle, ...) für den Start der IV-Route ist unbekannt.
ROUTE_DESTINATIONUNKNOWN	Der angegebene Ort (Adresse, Haltestelle, ...) für das Ziel der IV-Route ist unbekannt.
ROUTE_VIAUNKNOWN	Einer der angegebenen Via-Punkte ist unbekannt.
ROUTE_ORIGINDESTINATIONIDENTICAL	Start und Ziel sind gleich.
ROUTE_UNSUPPORTEDMODE	Einer der angefragten IV-Modi wird nicht unterstützt.
ROUTE_UNSUPPORTEDMOBILITYFILTER	Einer der angefragten Mobilitätsfilter wird nicht unterstützt.
ROUTE_UNSUPPORTEDALGORITHM	Der angefragte Algorithmustyp wird nicht unterstützt.
ROUTE_UNSUPPORTEDBAN	Einer der angefragten Ausschlussfilter (Autobahn, Mautstraßen, Fähren) wird nicht unterstützt.
ROUTE_NODATETIME	Weder Abfahrts- noch Ankunftszeit wurden angegeben.
ROUTE_DATETIMEERROR	Datum und/oder Uhrzeit sind unverständlich.
ROUTE_DEPARTUREAFTERARRIVAL	Die gewünschte Abfahrtszeit an allen Startpunkten liegt nach der gewünschten Ankunftszeit an allen Zielpunkten.
ROUTE_DATEOUTOFRANGE	Für das angefragte Datum liegen keine Routingdaten vor, z. B. weil das Datum in der Vergangenheit oder weit in der Zukunft liegt.

Tabelle 163: Liste der Fehlerzustände in **IndividualRouteResponse**.

16.3.2. RouteResultStructure

RouteResultStructure		<i>+Structure</i>	Fasst die Ergebnisdaten für eine einzelne IV-Route zusammen.	
	ResultId	1:1	<i>xs:NMTOKEN</i>	ID des Resultats für spätere Referenzierung bzw. für Debug-Zwecke.
	<i>ErrorMessage</i>	0:*	<i>+ErrorMessage</i>	Fehlermeldungen bezogen auf diese IV-Route. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. auch 7.4.2.
	Route	1:1	<i>+Route</i>	Daten zu einer IV-Route. Vgl. 16.3.3.

Tabelle 164: Beschreibung der Struktur **RouteResultStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

ROUTE_MODEPARAMETERSIGNORED	Bei dieser IV-Route wurde mindestens einer der Parameter für diesen IV-Modus ignoriert. Vgl. auch 7.3.2.
ROUTE_MOBILITYFILTERIGNORED	Bei dieser IV-Route wurde mindestens einer der Mobilitätsfilter ignoriert.

ROUTE_BANIGNORED	Bei dieser IV-Route wurde mindestens einer der Ausschlussfilter (Autobahn, Mautstraßen, Fähren) ignoriert.
-------------------------	--

Tabelle 165: Liste der Fehlerzustände in **RouteResult**.

16.3.3. RouteStructure

RouteStructure		+Structure	Daten zu einer einzelnen IV-Route.
RoutelId	1:1	<i>xs:NMTOKEN</i>	ID der Verbindung für spätere Referenzierung bzw. für Debug-Zwecke.
Duration	1:1	<i>xs:duration</i>	Gesamtdauer der IV-Route.
StartTime	1:1	<i>xs:dateTime</i>	Startzeitpunkt der IV-Route.
EndTime	1:1	<i>xs:dateTime</i>	Endzeitpunkt der IV-Route.
Distance	0:1	<i>Distance</i>	Gesamtdistanz der IV-Route als Länge des zurückzulegenden Weges.
RouteLeg	1:*	<i>+ContinuousLeg</i>	Teilwege dieser IV-Route. Es muss genau ein RouteLeg mehr existieren, als Vias angefragt wurden. Für IV-Routen ist in <i>ContinuousLeg.Service</i> lediglich <i>IndividualMode</i> und ggf. <i>SituationFullRef</i> gefüllt. Vgl. 9.3.8.
SituationFullRef	0:*	<i>+SituationFullRef</i>	Verweis auf Störungsnachrichten. Diese Nachrichten können im <i>TripResponseContext</i> (vgl. 9.3.2) zu finden sein oder auf anderem Wege bekannt gemacht werden. Vgl. 7.8.2.
Extension	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 166: Beschreibung der Struktur **RouteStructure**

17. Dienst Kartendienst

In der XML-Schema-Definition *Trias_Maps.xsd* werden Datentypen und Strukturen definiert, die für den Kartendienst verwendet werden.

17.1. Einfache Datentypen

Folgende einfache Typen (simple types) werden definiert:

<i>MapLayersEnumeration</i>	<i>physical satellite street rail names stops traffic</i>	Zusätzliche Layer in der Karte.
-----------------------------	---	---------------------------------

Tabelle 167: Liste der einfachen Typdefinitionen in *Trias_Maps.xsd*.

17.2. Anfragestrukturen

17.2.1. MapServiceRequestStructure

Eine Karte wird mittels eines Elements **MapServiceRequest** vom Typ *MapServiceRequestStructure* angefordert. Dieser liefert eine Bilddatei zurück, welche die angeforderte Karte enthält. Sollen zusätzliche Objekte auf Karten eingezeichnet (z. B. Haltestellen) oder aktive Elemente integriert werden (z. B. für Mouse-Over-Effekte, für Verlinkungen, usw.), dann muss dies durch den Client auf Basis einer vom Kartendienst generierten Hintergrundkarte gemacht werden.

MapServiceRequestStructure	+Structure	Fasst die Anfragedaten für den Abruf einer Karte zusammen.
-----------------------------------	-------------------	---

<i>MapProperties</i>	Aspect	1:1	+MapAspect	Geografischer Ausschnitt der zu generierenden Karte. Die generierte Karte darf einen anderen Kartenausschnitt als angegeben umfassen. Allerdings muss der Mittelpunkt nahezu unverändert bleiben und der tatsächliche Kartenausschnitt soll dem angefragten so ähnlich wie möglich sein. Vgl. 17.2.4.
	Size	1:1	+MapSize	Bildgröße der zu generierenden Karte. Der Kartendienst muss in der Lage sein, Bildgrößen bis mindestens 1920x1080 Bildpunkten („FullHD“) zu erzeugen. Die generierte Karte muss exakt dieser vorgegebenen Bildgröße entsprechen. Vgl. 17.2.5.
	<i>ImageType</i>	0:1	<i>xs:string</i>	Datenformat der zu generierenden Karte. Dieser muss als Medientyp (ehemals MIME-Typ) eines Bilddatenformates (Subtypen des Typs „image“) angegeben werden. Die Liste erlaubter Werte wird durch die IANA definiert. Falls nicht angegeben, muss der Kartendienst „image/png“ nutzen.
	Layer	1:*	<i>physical satellite street rail names stops traffic</i>	Gibt die Layer der Karte an. Dazu zählt die Hintergrundkarte, aber auch zusätzliche Informationen, die auf der Karte integriert werden sollen.
	<i>Opacity</i>	0:1	<i>Percent</i>	Undurchsichtigkeit des Kartenhintergrundes, soweit kein Hintergrundlayer gewählt wurde. Zwischen 0 (ganz durchsichtig) und 100 (ganz undurchsichtig).
	<i>BackgroundColor</i>	0:1	<i>xs:string</i>	Farbe des Kartenhintergrundes, soweit kein Hintergrundlayer gewählt wurde. Erlaubte Werte sind alle Farbangaben, die dem CSS3-Standard der W3C entsprechen.
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 168: Beschreibung der Struktur **MapServiceRequestStructure**.

17.2.2. ImageCoordinatesRequestStructure

In einigen Anwendungsfällen sollen geografische Objekte auf einer Karte eingezeichnet werden, oder als aktives, ggf. sogar verschiebbares Objekt zur Verfügung stehen. Um dies zu erreichen, muss der Client das Objekt nachträglich über das Kartenbild einzeichnen. Damit er dies tun kann, ist es hilfreich, aus den geografischen Koordinaten eines Objektes Bildkoordinaten generieren zu können. Solche Bildkoordinaten werden mittels eines Elements **ImageCoordinatesRequest** vom Typ *ImageCoordinatesRequestStructure* angefordert.

ImageCoordinatesRequestStructure			+Structure	Fasst die Anfragedaten für den Abruf von Bildkoordinaten zusammen.
<i>MapProperties</i>	Aspect	1:1	+MapAspect	Tatsächlicher geografischer Ausschnitt der referenzierten Karte. Vgl. 17.2.4.
	Size	1:1	+MapSize	Bildgröße der referenzierten Karte. Vgl. 17.2.5.
	Point	1:*	+GeoPosition	Geografische Punkte, für die Bildkoordinaten berechnet werden sollen. Diese dürfen außerhalb des angegebenen Kartenausschnitt liegen, allerdings kann ein Kartendienst die Bearbeitung bei zu weit außerhalb liegenden Punkten ablehnen. Vgl. 7.2.3.

Tabelle 169: Beschreibung der Struktur **ImageCoordinatesRequestStructure**.

17.2.3. GeoCoordinatesRequestStructure

Um einer Position auf einem Kartenbild die entsprechende geografische Position zuzuordnen zu können (z. B. nach dem Klick in eine Karte), benötigt man eine passende Um-

rechnungsfunktion. Solche geografischen Positionen werden mittels eines Elements **GeoCoordinatesRequest** vom Typ *GeoCoordinatesRequestStructure* angefordert.

GeoCoordinatesRequestStructure			+Structure	Fasst die Anfragedaten für den Abruf von geografischen Koordinaten zusammen.
<i>MapProperties</i>	Aspect	1:1	+MapAspect	Tatsächlicher geografischer Ausschnitt der referenzierten Karte. Vgl. 17.2.4.
	Size	1:1	+MapSize	Bildgröße der referenzierten Karte. Vgl. 17.2.5.
	ImagePoint	1:*	+MapCoordinate	Bildpunkte, für die geografische Koordinaten berechnet werden sollen. Diese können außerhalb des Kartenbildes liegen, allerdings kann ein Kartendienst die Bearbeitung von zu weit außerhalb liegenden Punkten ablehnen. Vgl. 17.2.6.

Tabelle 170: Beschreibung der Struktur **GeoCoordinatesRequestStructure**.

17.2.4. MapAspectStructure

MapAspectStructure			+Structure	Geografischer Kartenausschnitt.
	UpperLeft	1:1	+GeoPosition	Obere linke Ecke des geografischen Kartenausschnitts. Vgl. 7.2.3.
	LowerRight	1:1	+GeoPosition	Untere rechte Ecke des geografischen Kartenausschnitts. Vgl. 7.2.3.

Tabelle 171: Beschreibung der Struktur **MapAspectStructure**.

17.2.5. MapSizeStructure

MapSizeStructure			+Structure	Bildgröße einer Karte.
	Width	1:1	<i>xs:nonNegativeInteger</i>	Breite der Karte in Pixeln.
	Height	1:1	<i>xs:nonNegativeInteger</i>	Höhe der Karte in Pixeln.

Tabelle 172: Beschreibung der Struktur **MapSizeStructure**.

17.2.6. MapCoordinateStructure

MapCoordinateStructure			+Structure	Koordinaten eines Bildpunktes.
	X	1:1	<i>xs:integer</i>	X-Koordinate. Kann negativ sein oder größer als die Breite des zugrundeliegenden Bildes.
	Y	1:1	<i>xs:integer</i>	Y-Koordinate. Kann negativ sein oder größer als die Breite des zugrundeliegenden Bildes.

Tabelle 173: Beschreibung der Struktur **MapCoordinateStructure**.

17.3. Antwortstrukturen

17.3.1. MapServiceResponseStructure

Das Ergebnis einer Kartenanfrage wird mittels eines Elements **MapServiceResponse** vom Typ *MapServiceResponseStructure* übertragen.

MapServiceResponseStructure			+Structure	Fasst die Ergebnisdaten für eine Kartenanfrage zusammen.
	ErrorMessage	0:*	+ErrorMessage	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.

	<i>MapResult</i>	0:1	+ <i>MapResult</i>	Ergebnis der Kartenanfrage.
--	------------------	-----	--------------------	-----------------------------

Tabelle 174: Beschreibung der Struktur **MapServiceResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

MAP_UNSUPPORTEDSIZE	Die angefragte Bildgröße wird nicht unterstützt.
MAP_UNSUPPORTEDMEDIATYPE	Der angefragte Medientyp (ehemals MIME-Type) wird nicht unterstützt.
MAP_UNSUPPORTEDASPECT	Der angeforderte Kartenausschnitt liegt außerhalb des vom Kartendienst unterstützten Gebietes, ist zu groß oder ist zu klein.
MAP_LAYERIGNORED	Mindestens einer der angefragten Layer wurde ignoriert.
MAP_UNSUPPORTEDSTYLE	Der Kartendienst unterstützt die Angabe einer Hintergrundfarbe oder einer Undurchsichtigkeit nicht.
MAP_NOMAP	Es konnte keine der Anfrage entsprechende Karte generiert werden.

Tabelle 175: Liste der Fehlerzustände in *MapServiceResponse*.

17.3.2. ImageCoordinatesResponseStructure

Das Ergebnis einer Bildkoordinatenanfrage wird mittels eines Elements **ImageCoordinatesResponse** vom Typ *ImageCoordinatesResponseStructure* übertragen.

ImageCoordinatesResponseStructure		+ <i>Structure</i>	Fasst die Ergebnisdaten für eine Bildkoordinatenanfrage zusammen.
	<i>ErrorMessage</i>	0:*	+ <i>ErrorMessage</i>
	<i>Result</i>	0:*	+ <i>ImagePointResult</i>
			Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
			Einzelergebnisse der Umrechnung von Geo- zu Bildkoordinaten. Es können maximal soviele Elemente sein, wie geografische Punkte in der Anfrage enthalten waren. Zur Identifizierung ist in jedem Einzelergebnis der angefragte Punkt enthalten. Vgl. 17.3.5.

Tabelle 176: Beschreibung der Struktur **ImageCoordinatesResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

MAP_UNSUPPORTEDSIZE	Die angefragte Bildgröße wird nicht unterstützt.
MAP_UNSUPPORTEDASPECT	Der angeforderte Kartenausschnitt liegt außerhalb des vom Kartendienst unterstützten Gebietes, ist zu groß oder ist zu klein.
MAP_TOOMANYPPOINTS	Es wurden zu viele Punktobjekte zur Umrechnung angegeben.
MAP_UNSUPPORTEDPOINT	Mindestens einer der angefragten Punkte liegt außerhalb des umrechenbaren Gebietes.

Tabelle 177: Liste der Fehlerzustände in *ImageCoordinatesResponse*.

17.3.3. GeoCoordinatesResponseStructure

Das Ergebnis einer Geokoordinatenanfrage wird mittels eines Elements **GeoCoordinatesResponse** vom Typ *GeoCoordinatesResponseStructure* übertragen.

GeoCoordinatesResponseStructure		+Structure	Fasst die Ergebnisdaten für eine Geokoordinatenanfrage zusammen.
<i>ErrorMessage</i>	0:*	+ErrorMessage	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
<i>Result</i>	0:*	+GeoCoordinateResult	Einzelresultate der Umrechnung von Bild- zu Geokoordinaten. Es können maximal soviele Elemente sein, wie Bildpunkte in der Anfrage enthalten waren. Zur Identifizierung ist in jedem Einzelresultat der angefragte Punkt enthalten. Vgl. 17.3.6.

Tabelle 178: Beschreibung der Struktur **GeoCoordinatesResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

MAP_UNSUPPORTEDSIZE	Die angefragte Bildgröße wird nicht unterstützt.
MAP_UNSUPPORTEDASPECT	Der angeforderte Kartenausschnitt liegt außerhalb des vom Kartendienst unterstützten Gebietes, ist zu groß oder ist zu klein.
MAP_TOOMANYPOINTS	Es wurden zu viele Punktobjekte zur Umrechnung angegeben.
MAP_UNSUPPORTEDPOINT	Mindestens einer der angefragten Punkte liegt außerhalb des umrechenbaren Gebietes.

Tabelle 179: Liste der Fehlerzustände in **GeoCoordinatesResponse**.

17.3.4. MapResultStructure

MapResultStructure		+Structure	Fasst die Daten einer generierten Karte zusammen.
<i>File</i>	1:1	xs:base64Binary	Die Bilddaten der generierten Karte.
<i>ImageType</i>	1:1	xs:string	Datenformat der generierten Karte. Dieser muss als Medientyp (ehemals MIME-Typ) eines Bilddatenformates (Subtypen des Typs „image“) angegeben werden. Die Liste erlaubter Werte wird durch die IANA definiert.
<i>Aspect</i>	1:1	+MapAspect	Tatsächlicher geografischer Ausschnitt der generierten Karte. Er darf von dem angefragten Kartenausschnitt abweichen. Allerdings muss der Mittelpunkt nahezu unverändert bleiben und der tatsächliche Kartenausschnitt soll dem angefragten so ähnlich wie möglich sein. Vgl. 17.2.4.

Tabelle 180: Beschreibung der Struktur **MapResultStructure**.

17.3.5. ImagePointResultStructure

ImagePointResultStructure		+Structure	Fasst die Daten einer Umrechnung von Geo- zu Bildkoordinaten zusammen.
<i>ErrorMessage</i>	0:*	+ErrorMessage	Fehlermeldungen bezogen auf die Umrechnung des nachfolgenden Geokoordinaten-Paars. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
<i>Point</i>	1:1	+GeoPosition	Geografischer Punkt, für den Bildkoordinaten berechnet werden sollten. Dies muss einer der Punkte aus der zugehörigen Dienstanfrage sein. Vgl. 7.2.3.
<i>ImagePoint</i>	0:1	+MapCoordinate	Bildkoordinaten für den angefragten geografischen Punkt, falls die Umrechnung durchgeführt werden konnte. Der Bildpunkt kann außerhalb des zugrundeliegenden Kartenbildes liegen (inklusive negativer Koordinatenwerte). Falls dieses Element fehlt, muss mindestens ein Fehlercode gesetzt sein. Vgl. 17.2.6.

Tabelle 181: Beschreibung der Struktur *ImagePointResultStructure*.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

MAP_POINTNOTONMAP	Der angegebene Punkt liegt außerhalb der Karte. Trotzdem konnte eine Konversion durchgeführt werden.
MAP_UNSUPPORTEDPOINT	Der angefragte Punkt liegt außerhalb des umrechenbaren Gebietes.

Tabelle 182: Liste der Fehlerzustände in *ImagePointResultStructure*.

17.3.6. GeoCoordinateResultStructure

GeoCoordinateResultStructure		+Structure	Fasst die Daten einer Umrechnung von Bild- zu Geokoordinaten zusammen.
<i>ErrorMessage</i>	0:*	+ErrorMessage	Fehlermeldungen bezogen auf die Umrechnung des nachfolgenden Geokoordinaten-Paars. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
<i>Point</i>	0:1	+GeoPosition	Geografische Koordinaten für den angefragten Bildpunkt, falls die Umrechnung durchgeführt werden konnte, berechnet werden sollten. Falls dieses Element fehlt, muss mindestens ein Fehlercode gesetzt sein. Vgl. 7.2.3.
ImagePoint	1:1	+MapCoordinate	Bildkoordinaten, für den geografische Koordinaten berechnet werden sollten. Dies muss einer der Punkte aus der zugehörigen Dienstanfrage sein. Vgl. 17.2.6.

Tabelle 183: Beschreibung der Struktur *GeoCoordinateResultStructure*.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

MAP_POINTNOTONMAP	Der angegebene Punkt liegt außerhalb der Karte. Trotzdem konnte eine Konversion durchgeführt werden.
MAP_UNSUPPORTEDPOINT	Der angefragte Punkt liegt außerhalb des umrechenbaren Gebietes.

Tabelle 184: Liste der Fehlerzustände in *GeoCoordinateResultStructure*.

18. Dienst Schadensmeldung / Zustand von Einrichtungen

18.1. Beschreibung

Dieser Dienst erlaubt es, den Zustand einer Haltestelleneinrichtung oder Fahrzeugausstattung abzufragen oder zu melden (Schadensmeldung).

18.2. Einfache Typen

Folgende einfache Typen (simple types) werden definiert:

<i>FacilityStatusTypeEnumeration</i>	<i>OK dirty destroyed damaged stolen out of order</i>	Zustand der Einrichtung.
<i>FacilityAvailabilityEnumeration</i>	<i>unknown available notAvailable partiallyAvailable added removed</i>	Verfügbarkeit der Einrichtung.

18.3. Komplexe Strukturen

18.3.1. FacilityStructure

FacilityStructure			+Structure	Beschreibung und Zustand einer Einrichtung.	
	a	VehicleFacility	-1:1	+VehicleFacility	Definition einer Fahrzeugeinrichtung. Vgl. 18.3.2.
	b	InfrastructureFacility		+InfrastructureFacility	Definition einer Infrastruktureinrichtung. Vgl. 18.3.3.
	Condition		1:1	+FacilityStatus	Zustand der Einrichtung. Vgl. 18.3.7.
<i>Extension</i>		0:1	<i>xs:anyType</i>	Erweiterung.	

Tabelle 185: Beschreibung der Struktur **FacilityStructure**.

18.3.2. VehicleFacilityStructure

VehicleFacilityStructure			+Structure	Beschreibung einer Fahrzeugeinrichtung.
<i>VehicleFacilityRef</i>	:::	1:1	+VehicleFacilityRefGroup	Referenz auf eine Einrichtung. Vgl. 18.3.4.
<i>ServiceFacility</i>	:::	1:1	+ServiceFacilityGroup	Klassifizierung der Einrichtung. Vgl. 7.7.3.
	<i>FacilityDescription</i>	0:1	+InternationalText	Name oder Beschreibung der Einrichtung. Vgl. 7.2.2.
	<i>LocationDescription</i>	0:1	+InternationalText	Beschreibung, wo die Einrichtung zu finden ist. Vgl. 7.2.2.

Tabelle 186: Beschreibung der Struktur **VehicleFacilityStructure**.

18.3.3. InfrastructureFacilityStructure

InfrastructureFacilityStructure			+Structure	Beschreibung einer Infrastruktureinrichtung.
<i>InfrastructureFacilityRef</i>	:::	1:1	+InfrastructureFacilityRefGroup	Referenz auf eine Einrichtung. Vgl. 18.3.5.
<i>StopFacility</i>	:::	1:1	+StopFacilityGroup	Klassifizierung der Einrichtung. Vgl. 7.7.2.
	<i>FacilityDescription</i>	0:1	+InternationalText	Name oder Beschreibung der Einrichtung. Vgl. 7.2.2.
	<i>Location</i>	0:1	+GeoPosition	Koordinatenposition der Einrichtung. Vgl. 7.2.3.
	<i>LocationDescription</i>	0:1	+InternationalText	Beschreibung, wo die Einrichtung zu finden ist. Vgl. 7.2.2.

Tabelle 187: Beschreibung der Struktur **InfrastructureFacilityStructure**.

18.3.4. VehicleFacilityRefGroup

VehicleFacilityRefGroup			+Group	Referenzierung einer Fahrzeugeinrichtung durch Referenz auf die Einrichtung selbst oder auf ein übergeordnetes Objekt.
	<i>FacilityRef</i>	0:1	→FacilityCode	Referenz auf Einrichtung. Vgl. 7.4.1.
	<i>OwnerRef</i>		→OwnerCode	Referenz auf Eigentümer/Verantwortlichen. Vgl. 7.4.1.
	<i>OperatorRef</i>		→OperatorCode	Referenz auf Verkehrsunternehmen. Vgl. 7.4.1.
	<i>LineRef</i>		→LineCode	Referenz auf Linie. Vgl. 7.4.1.

	<i>JourneyRef</i>		→ <i>Journey-Code</i>	Referenz auf Fahrt. Vgl. 7.4.1.
	<i>VehicleRef</i>		→ <i>Vehicle-Code</i>	Referenz auf Fahrzeug. Vgl. 7.4.1.

Tabelle 188: Beschreibung der Gruppe **VehicleFacilityRefGroup**.

18.3.5. InfrastructureFacilityRefGroup

InfrastructureFacilityRefGroup			+Group	Referenzierung einer Infrastruktureinrichtung durch Referenz auf die Einrichtung selbst oder auf ein übergeordnetes Objekt.
	<i>FacilityRef</i>	0:1	→ <i>FacilityCode</i>	Referenz auf Einrichtung. Vgl. 7.4.1.
	<i>OwnerRef</i>		→ <i>OwnerCode</i>	Referenz auf Eigentümer/Verantwortlichen. Vgl. 7.4.1.
	<i>StopPointRef</i>		→ <i>StopPoint</i>	Referenz auf einen Haltepunkt. Vgl. 7.5.1.
	<i>StopPlaceRef</i>		→ <i>StopPlace</i>	Referenz auf eine Haltestelle. Vgl. 7.5.1.
	<i>OperatorRef</i>		→ <i>Operator-Code</i>	Referenz auf Verkehrsunternehmen. Vgl. 7.4.1.
	<i>LineRef</i>		→ <i>LineCode</i>	Referenz auf Linie. Vgl. 7.4.1.

Tabelle 189: Beschreibung der Gruppe **InfrastructureFacilityRefGroup**.

18.3.6. FacilityDataFilterGroup

FacilityDataFilterGroup			+Group	Eine Menge von Objektreferenzen als Filter zum Eingrenzen der in Frage kommenden Einrichtungen.
	<i>FacilityRef</i>	0:*	→ <i>FacilityCode</i>	Referenz auf Einrichtung. Vgl. 7.4.1.
	<i>OwnerRef</i>	0:*	→ <i>OwnerCode</i>	Referenz auf Eigentümer/Verantwortlichen. Vgl. 7.4.1.
	<i>StopPointRef</i>	0:*	→ <i>StopPoint</i>	Referenz auf einen Haltepunkt. Vgl. 7.5.1.
	<i>StopPlaceRef</i>	0:*	→ <i>StopPlace</i>	Referenz auf eine Haltestelle. Vgl. 7.5.1.
	<i>OperatorRef</i>	0:*	→ <i>Operator-Code</i>	Referenz auf Verkehrsunternehmen. Vgl. 7.4.1.
	<i>LineRef</i>	0:*	→ <i>LineCode</i>	Referenz auf Linie. Vgl. 7.4.1.
	<i>JourneyRef</i>	0:*	→ <i>Journey-Code</i>	Referenz auf Fahrt. Vgl. 7.4.1.
	<i>VehicleRef</i>	0:*	→ <i>Vehicle-Code</i>	Referenz auf Fahrzeug. Vgl. 7.4.1.

Tabelle 190: Beschreibung der Gruppe **FacilityDataFilterGroup**.

18.3.7. FacilityStatusStructure

FacilityStatusStructure			+Structure	Der Zustand einer Einrichtung.
	Availability	1:1	<i>FacilityAvailabilityEnumeration</i>	Verfügbarkeit der Einrichtung. Vgl. 18.2
	Status	1:1	<i>FacilityStatusTypeEnumeration</i>	Klassifizierung des Zustands. Vgl. 18.2
	<i>StatusDescription</i>	0:1	+InternationalText	Beschreibung des Zustands. Vgl. 7.2.2.

Tabelle 191: Beschreibung der Struktur **FacilityStatusStructure**.

18.4. Anfrage Schadensmeldung

Eine Schadensmeldung an einer Einrichtung wird mittels eines Elements **FacilityStatusReport** vom Typ *FacilityStatusReportStructure* gestellt.

18.4.1. FacilityStatusReportStructure

FacilityStatusReportStructure			+Structure	Fasst die Daten für eine Schadensmeldung an einer Einrichtung zusammen.	
	a	VehicleFacility	-1:1	+VehicleFacility	Definition einer Fahrzeugeinrichtung. Vgl. 18.3.2.
	b	InfrastructureFacility		+InfrastructureFacility	Definition einer Infrastruktureinrichtung. Vgl. 18.3.3.
	Condition		1:1	+FacilityStatus	Zustand der Einrichtung. Vgl. 18.3.7.
	<i>Extension</i>		0:1	<i>xs:anyType</i>	Erweiterung.

Tabelle 192: Beschreibung der Struktur **FacilityStatusReportStructure**.

18.5. Antwort Schadensmeldung

Die Antwort auf eine Schadensmeldung (FacilityStatusReport, siehe 18.4) wird mittels eines Elements **FacilityStatusReportResponse** vom Typ *FacilityStatusReportResponseStructure* übertragen.

18.5.1. FacilityStatusReportResponseStructure

FacilityStatusReportResponseStructure			+Structure	Antwort auf die Schadensmeldung.
	ErrorMessage	1:1	+ErrorMessage	Fehlermeldung. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.

Tabelle 193: Beschreibung der Struktur **FacilityStatusReportResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

FACILITYSTATUSREPORT_FACILITYUNKNOWN	Die angegebene Einrichtung ist unbekannt.
FACILITYSTATUSREPORT_OWNERUNKNOWN	Der angegebene Eigentümer ist unbekannt.
FACILITYSTATUSREPORT_OPERATORUNKNOWN	Das angegebene Verkehrsunternehmen ist unbekannt.
FACILITYSTATUSREPORT_LINEUNKNOWN	Die angegebene Linie ist unbekannt.
FACILITYSTATUSREPORT_JOURNEYUNKNOWN	Die angegebene Fahrt ist unbekannt.
FACILITYSTATUSREPORT_VEHICLEUNKNOWN	Das angegebene Fahrzeug ist unbekannt.
FACILITYSTATUSREPORT_STOPPOINTUNKNOWN	Der angegebene Haltepunkt ist unbekannt.
FACILITYSTATUSREPORT_STOPPLACEUNKNOWN	Die angegebene Haltestelle ist unbekannt.

Tabelle 194: Liste der Fehlerzustände in **FacilityStatusReportResponse**.

18.6. Anfrage Zustand von Einrichtungen

Eine Anfrage nach dem aktuellen Zustand von Einrichtungen wird mittels eines Elements **FacilityRequest** vom Typ *FacilityRequestStructure* gestellt.

18.6.1. FacilityRequestStructure

FacilityRequestStructure			+Structure	Fasst die Daten für eine Schadensmeldung an einer Einrichtung zusammen.
<i>FacilityDataFilter</i>	:::	1:1	+FacilityDataFilterGroup	Objektreferenzen als Filter. Vgl. 18.3.6.
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterung.

Tabelle 195: Beschreibung der Struktur **FacilityRequestStructure**.

18.7. Antwort Zustand von Einrichtungen

Die Antwort auf eine Zustandsanfrage wird mittels eines Elements **FacilityResponse** vom Typ *FacilityResponseStructure* übertragen.

18.7.1. FacilityResponseStructure

FacilityResponseStructure			+Structure	Antwort auf die Schadensmeldung.
	<i>ErrorMessage</i>	0:*	+ErrorMessage	Fehlermeldung bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
	<i>FacilityResult</i>	0:*	+FacilityResult	Resultatstruktur. Vgl. 18.7.2.

Tabelle 196: Beschreibung der Struktur **FacilityResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

FACILITYREQUEST_FACILITYUNKNOWN	Die angegebene Einrichtung ist unbekannt.
FACILITYREQUEST_OWNERUNKNOWN	Der angegebene Eigentümer ist unbekannt.
FACILITYREQUEST_OPERATORUNKNOWN	Das angegebene Verkehrsunternehmen ist unbekannt.
FACILITYREQUEST_LINEUNKNOWN	Die angegebene Linie ist unbekannt.
FACILITYREQUEST_JOURNEYUNKNOWN	Die angegebene Fahrt ist unbekannt.
FACILITYREQUEST_VEHICLEUNKNOWN	Das angegebene Fahrzeug ist unbekannt.
FACILITYREQUEST_STOPPOINTUNKNOWN	Der angegebene Haltepunkt ist unbekannt.
FACILITYREQUEST_STOPPLACEUNKNOWN	Die angegebene Haltestelle ist unbekannt.

Tabelle 197: Liste der Fehlerzustände in **FacilityResponse**.

18.7.2. FacilityResultStructure

FacilityResultStructure		+Structure	Resultatstruktur für die Facility-Zustandsanfrage.
<i>ErrorMessage</i>	0:*	<i>+ErrorMessage</i>	Fehlermeldungen bezogen auf das einzelne FacilityResult. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
Facility	1:1	+Facility	Informationen zur Einrichtung. Vgl. 18.3.1

Tabelle 198: Beschreibung der Struktur **FacilityResultStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

FACILITYRE-QUEST_STATUSNOTCONFIRMED	Für die Einrichtung liegt ein Zustandsbericht vor, der aber noch nicht bestätigt ist.
--	---

Tabelle 199: Liste der Fehlerzustände in **FacilityResultStructure**.

19. Dienst Benachrichtigungsdienst

19.1. Beschreibung

Der Benachrichtigungsdienst dient der aktiven Benachrichtigung von Benutzern über aktuelle Geschehnisse. Die Benutzer können Abonnements einrichten, um sich bei Auftreten neuer Informationen benachrichtigen zu lassen

Der Dienst informiert unter anderem über

- Störungen oder Ereignisse auf einer Verbindung, einer Strecke oder an einer Haltestelle,
- alternative Verbindungen (aufgrund von Störungen oder der Prozessdatenlage),
- den Status eines Anschlusses und zusätzliche Informationen bei Anschlussverlust.

Diese Aufzählung ist nicht abschließend.

Dabei verfügt der Benachrichtigungsdienst über Funktionalität, die es ihm ermöglicht, betroffene Objekte (Verbindungen, Anschlüsse etc.) zu ermitteln. Teile des Benachrichtigungsdienstes (Ereignismeldungen und Änderungen an der Fahrzeugausstattung bzw. an Haltestelleneinrichtungen) wurden aus dem SIRI-Standard übernommen.

Eine Benachrichtigung besteht aus einer eindeutigen ID, einem Typ und den Nutzdaten. Folgende Arten von Abonnements können eingerichtet werden (vgl. auch die allgemeine Beschreibung von Abonnementanfragen in 7.1.2):

- **SituationExchangeSubscriptionRequest** (aus SIRI SX): Benachrichtigung über Ereignisse und Störungen,
- **FacilityMonitoringSubscriptionRequest** (aus SIRI FM): Änderungen an der Fahrzeugausstattung bzw. an Haltestelleneinrichtungen,
- **TripMonitoringSubscriptionRequest** (neu in TRIAS): Überwachung einer geplanten Verbindung.

Die Funktionsweise des Benachrichtigungsdienstes ist in Abbildung 9 zu sehen. Um Nachrichten zu erhalten, muss ein Portalsystem ein Abonnement beim Benachrichtigungsdienst erstellen (1).

Bei der Erstellung des Abonnements kann angegeben werden, welche Typen von Nachrichten dem Portalsystem übermittelt werden sollen. Bei jeder neuen Meldung von

einem Datenlieferant (2) ermittelt der Benachrichtigungsdienst die betroffenen Abonnements und leitet die Nachrichten an das zugehörige Portalsystem weiter. (3)

Soll das Portalsystem keine neuen Nachrichten mehr empfangen, so muss es das Abonnement am Benachrichtigungsdienst abmelden (4) (oder das Abonnement erlischt von selbst nach Ablauf des Gültigkeitszeitraums).

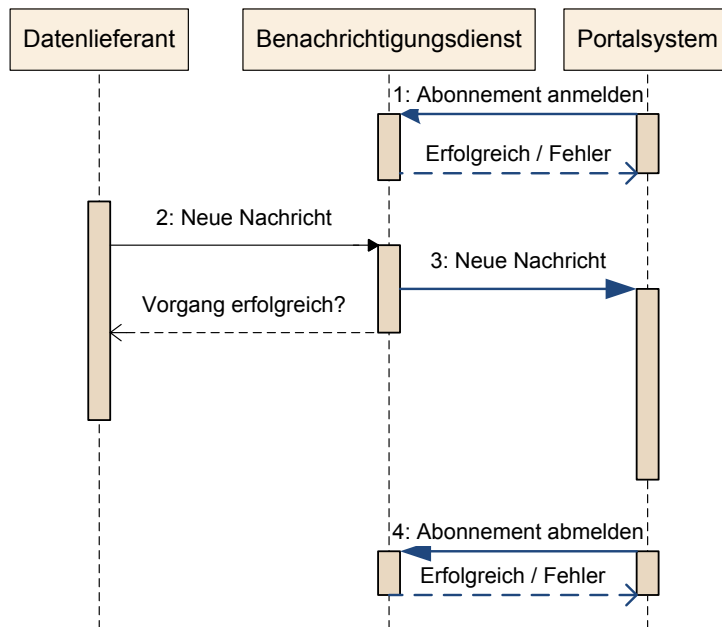


Abbildung 9: Funktionsweise des Benachrichtigungsdienstes

19.2. Komplexe Strukturen

Im Folgenden werden die komplexen Strukturen beschrieben, die in `Trias_Alerts` definiert sind.

19.2.1. TripMonitoringParamStructure

<i>TripMonitoringParamStructure</i>			+Structure	Parameter für die Verbindungsüberwachung.
<i>TripMonitoringPolicy</i>	<i>Severity</i>	0:1	<i>unknown</i> <i>verySlight</i> <i>slight</i> <i>normal</i> <i>severe</i> <i>verySevere</i> <i>noImpact</i> <i>undefined</i>	Prioritäten von Ereignissen (nach TPEG-Tabelle 26).
	<i>MinimumDelayChangeThreshold</i>	0:1	<i>xs:duration</i>	Verspätungsänderung, ab der wieder benachrichtigt wird.
	<i>AcceptThirdPartyInformation</i>	0:1	<i>xs:boolean</i>	Gibt an, ob Informationen von anderen Plattformen abgeholt werden sollen.

	<i>IncludeAlternatives</i>	0:1	<i>xs:boolean</i>	Gibt an, ob direkt Alternativen mit zurückgegeben werden sollen.
--	----------------------------	-----	-------------------	--

Tabelle 200: Beschreibung der Struktur *TripMonitoringParamStructure*

Der Parameter *AcceptThirdPartyInformation* gibt an, ob die angefragte EKAP die Antwort rein aus eigenen Informationsquellen erstellen soll, oder ob sie (fehlende) Informationen aus weiteren Datenquellen, wie beispielsweise weitere EKAPs, einbeziehen darf. Somit lässt sich durch die anfragende Plattform steuern, woher die Daten stammen. Der Anfragende kann damit entweder selbst fehlende Informationen an verschiedenen Plattformen anfragen oder diese Funktion von dem angefragten System ausführen lassen. Der Parameter ist insbesondere dann sinnvoll, wenn mehrere EKAPs zusammenschaltet werden und dabei unterschiedliche (geografische) Zuständigkeiten haben.

19.3. Anfragestrukturen

19.3.1. TripMonitoringSubscriptionRequestStructure

<i>TripMonitoringSubscriptionRequestStructure</i>			<i>+Structure</i> (abgeleitet von <i>AbstractSubscriptionStructure</i>)	Einrichten einer Verbindungsüberwachung.
	<i>Trip</i>	1:1	<i>+Trip</i>	Zu überwachende Verbindung. Vgl. 9.3.4.
	<i>TripRequest</i>	0:1	<i>+TripRequest</i>	Ursprüngliche Verbindungsanfrage. Vgl. 9.2.1.
	<i>MonitoringParameter</i>	0:1	<i>+TripMonitoringParam</i>	Weitere Parameter zur Konfiguration der Verbindungsüberwachung. Vgl. 19.2.1.

Tabelle 201: Beschreibung der Struktur *TripMonitoringSubscriptionRequestStructure*

19.4. Antwortstrukturen

19.4.1. TripMonitoringDeliveryStructure

<i>TripMonitoringDeliveryStructure</i>			<i>+Structure</i> (abgeleitet von <i>siri:AbstractServiceDeliveryStructure</i>)	Liefert Informationen zu einer überwachten Verbindung.
<i>MonitoringAlertReason</i>	<i>Situations</i>	0:1	<i>Situations</i>	(Störungs-)Ereignisse als Grund für die Meldung (vgl. 7.8.1).
	<i>FacilityCondition</i>	0:*	<i>+siri:FacilityCondition</i>	Ein oder mehrere Zustände von Facilities als Grund für die Meldung, vgl. 7.7.
	<i>ConnectionStatus</i>	0:*	<i>+ConnectionStatus</i>	Der Status eines Anschlusses in der überwachten Verbindung. Vgl. 13.3.4.
<i>Alternati</i>	<i>TripResponse</i>	0:1	<i>+TripResponse</i>	Enthält eine Verbindungsalternative (vgl. 9.3.1).

<i>veTrip</i>			e	
	<i>ErrorMessage</i>	0:*	+ <i>ErrorMessage</i> e	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Vgl. 7.4.2.

Tabelle 202: Beschreibung der Struktur *TripMonitoringDeliveryStructure*

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<i>ALERT_TRIPREQUEST_ORIGIN_UNKNOWN</i>	Der Abfahrtsort (Adresse, Haltestelle, ...), des TripRequests ist unbekannt.
<i>ALERT_TRIPREQUEST_DESTINATION_UNKNOWN</i>	Der Ankunftsort (Adresse, Haltestelle, ...), des TripRequests ist unbekannt.
<i>ALERT_TRIP_UNKNOWN</i>	Die zu überwachende Verbindung ist unbekannt.
<i>ALERT_THRESHOLD_NEGATIVE</i>	Verspätungsänderung, ab der wieder benachrichtigt wird, hat einen negativen Wert.
<i>ALERT_FACILITY_UNKNOWN</i>	Die zu überwachende Einrichtung ist unbekannt.
<i>ALERT_MONITORED_OBJECT_UNKNOWN</i>	Das Objekt, zu dem Ereignisse und Störungen gemeldet werden sollen, ist unbekannt.

Tabelle 203: Liste der Fehlerzustände in *TripMonitoringDeliveryStructure*.

20. Dienst Personalisierungsdienst

20.1. Beschreibung

Dieser Dienst stellt Funktionen bereit, über die Daten für die personalisierte Konfiguration beliebiger Dienste hinterlegt werden können. Unter „Konfiguration“ sind hier nicht nur explizite Einstellungen im engeren Sinn zu verstehen, sondern allgemein benutzerbezogene Daten. All diese Informationen können das Verhalten derjenigen Dienste, die den Personalisierungsdienst benutzen, beeinflussen und stellen somit eine Konfiguration für den verwendenden Dienst dar.

Es ist wichtig, zu beachten, dass der Dienst keine eigenen personalisierten Funktionen zur Verfügung stellt. Seine Aufgabe besteht in der Verwaltung von Benutzereinstellungen. Andere Dienste können auf den Personalisierungsdienst zurückgreifen, um ihre Funktionen personalisiert zur Verfügung zu stellen. Der Zugriff auf die personalisierten Daten erfolgt mit Hilfe des Authentifizierungsdienstes, um einen Schutz der Daten, entsprechend der rechtlichen Vorgaben zum Datenschutz, zu ermöglichen. Diese Vorgaben sind bei der Umsetzung des Personalisierungsdienstes zu berücksichtigen, werden in der Dienstbeschreibung des Personalisierungsdienstes jedoch nicht behandelt. Generell gilt, dass Benutzer nur auf von ihnen gespeicherte Daten zugreifen können. Daten von anderen Benutzern bleiben stets vollkommen unsichtbar und unerreichbar. Wie diese Trennung der benutzerbezogenen Daten erreicht wird, ist herstellerabhängig und wird hier nicht vorgeschrieben.

Der Dienst speichert beliebige Datenwerte, wobei jedem Datenwert ein – für den aktuellen Benutzer – eindeutiger Schlüssel zugeordnet wird. Über diesen Schlüssel kann der Datenwert wieder abgerufen werden. Bei den Werten handelt es sich um Zeichenketten mit beliebigem Format, sodass prinzipiell jegliche Datenstrukturen abgelegt werden können.

Der Dienst bietet keine Zuordnung von Werten zu einem oder mehreren anderen Diensten an. Diese kann über herstellerspezifische Schnittstellen unterstützt werden.

20.2. Interaktionen

Die Funktionen dieses Dienstes stehen oft für sich alleine und sind sehr generisch. Daher werden im Folgenden drei beispielhafte Abläufe im Zusammenhang mit dem Dienst beschrieben, bei denen die Funktionsaufrufe in einen größeren Zusammenhang eingebettet sind.

Auf den Personalisierungsdienst kann von verschiedenen Komponenten aus zugegriffen werden, zum Beispiel von der EKAP oder von Mehrwertdiensten aus. Bei den im Folgenden gezeigten Zugriffen auf den Personalisierungsdienst handelt es sich um beispielhafte Abläufe. Dabei sollen die anderen beteiligten Komponenten nicht konkret festgelegt werden. Aus diesem Grund wird in den untenstehenden Diagrammen von beliebigen Mehrwertdiensten - welche in beliebigen Ausprägungen vorkommen können und im Rahmen dieser Schrift ebenfalls nicht näher spezifiziert werden - gesprochen. Zugriffe von anderen Komponenten aus finden nach demselben Muster statt.

20.2.1. Lebenszyklus eines Wertes

Das erste Beispiel zeigt auf, wie ein Wert über den Personalisierungsdienst abgelegt und wieder gelöscht werden kann.

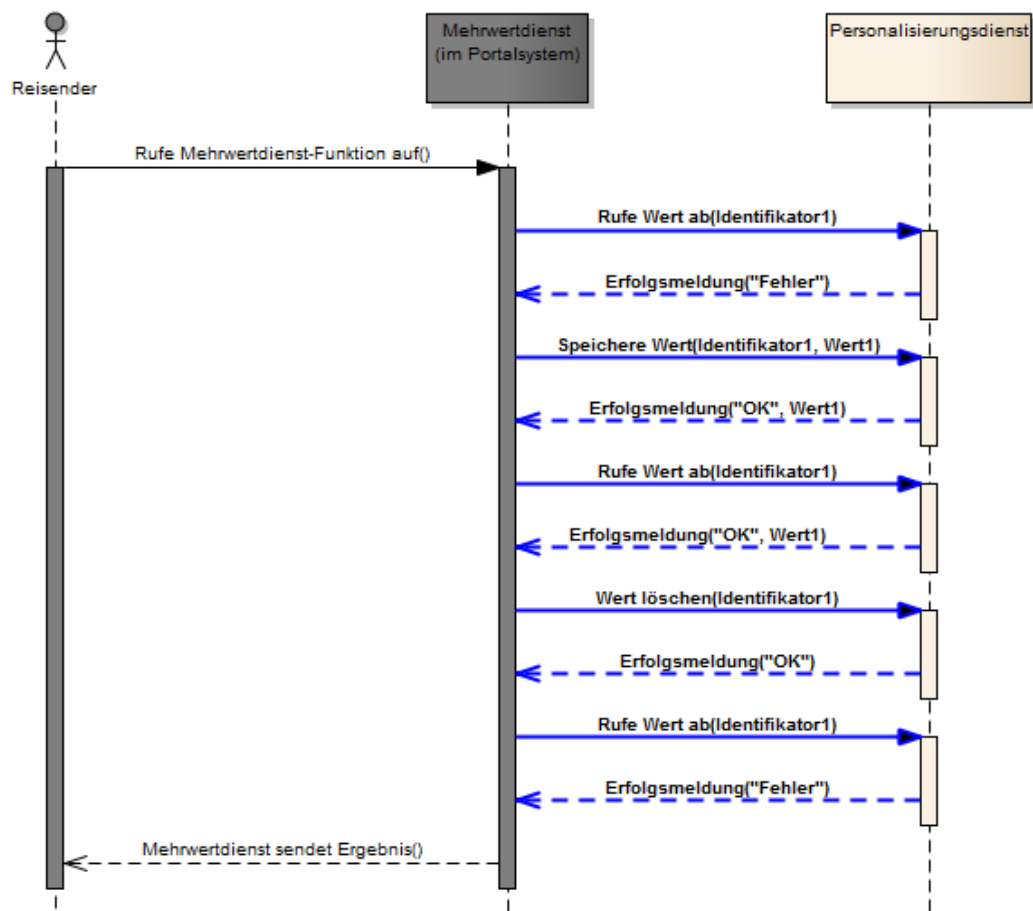


Abbildung 10: Sequenzdiagramm Lebenszyklus eines Wertes

20.2.2. Werteliste ermitteln

In diesem Beispiel ist dargestellt, wie die Funktion zum Auflisten der verfügbaren Werte funktioniert.

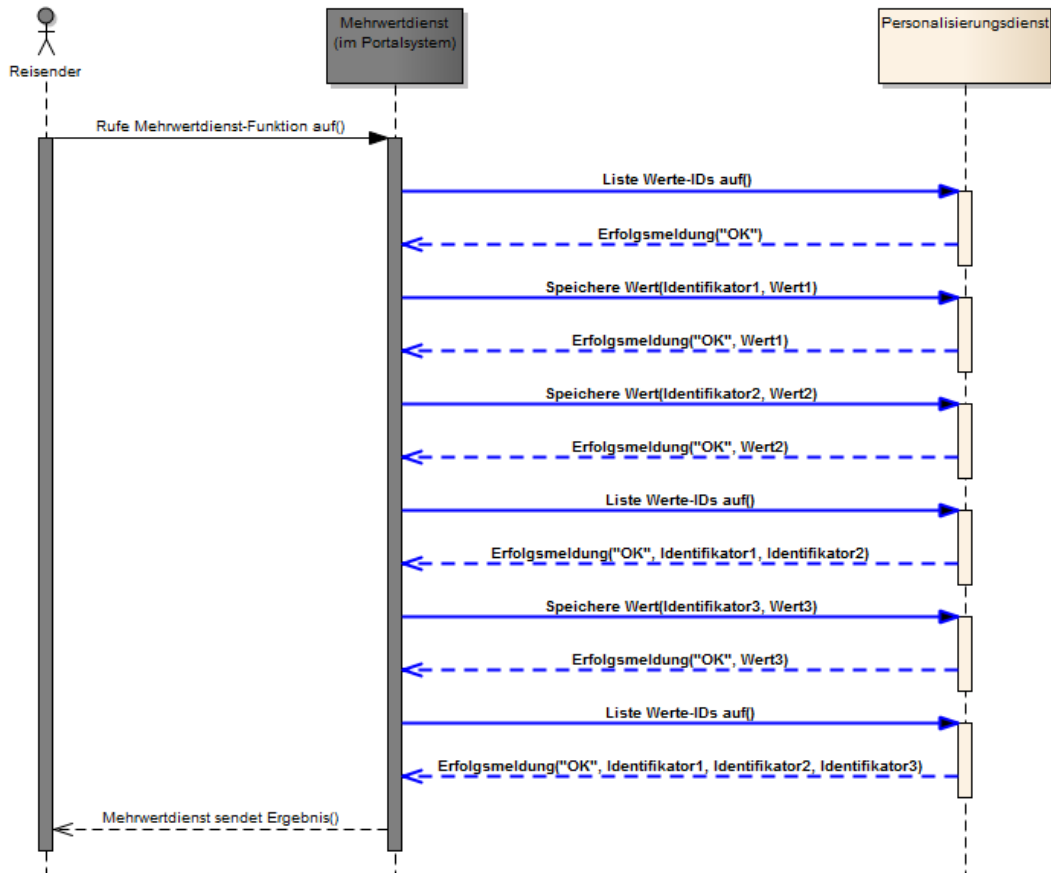


Abbildung 11: Sequenzdiagramm Werteliste ermitteln

20.2.3. Werte speichern und abrufen

Im folgenden Beispiel wird gezeigt, wie ein Reisender auf unterschiedliche Mehrwertdienste zugreift, welche den Personalisierungsdienst nutzen, um Konfigurationseinstellungen abzulegen und wieder abzurufen.

Dabei wird illustriert, dass zu jedem gespeicherten Wert ein für den Benutzer eindeutiger Identifikator gehört. Erstmaliges Speichern eines Wertes für einen Identifikator legt einen Wert im Speicher an, erneutes Speichern unter demselben Identifikator überschreibt den ursprünglichen Wert.

Des Weiteren wird im Beispiel gezeigt, dass die Werte vom Personalisierungsdienst prinzipiell dauerhaft gespeichert werden, auch wenn beispielsweise inzwischen ein anderer Mehrwertdienst genutzt wird. Hierbei ist zu beachten, dass Anbieter gespeicherte Werte mit einem Löschdatum versehen können, um ungenutzte Daten nicht unbegrenzt vorhalten zu müssen. Die genaue Umsetzung entsprechender Löschungen ist anbieter-spezifisch und wird in dieser Schrift nicht festgelegt.

Um klarzustellen, dass in der standardisierten Form keine Zuordnung zwischen Werten und Diensten stattfindet, wird ferner vorgeführt, dass ein Mehrwertdienst (MWD 2 in der Abbildung) einen ursprünglich von einem anderen Mehrwertdienst (MWD 1 in der Abbildung) abgelegten Wert überschreiben kann. Die Voraussetzung dazu ist lediglich, dass MWD 2 den Identifikator des Werts kennt, sei es, weil MWD 1 und 2 vom selben Her-

steller stammen, oder weil der Entwickler von MWD 1 die in MWD 1 verwendeten Identifikatoren öffentlich bekanntgegeben hat.

Der Dienst kann bei Bedarf durch herstellerspezifische Schnittstellen um die Funktionalität, den Zugriff auf bestimmte Werte dienstspezifisch einzuschränken, erweitert werden und somit den Zugriff ausgewählter Dienste auf einzelne Werte einschränken. Hierzu kann, je nach Implementierung und Konfiguration, auch der Authentifizierungsdienst genutzt werden.

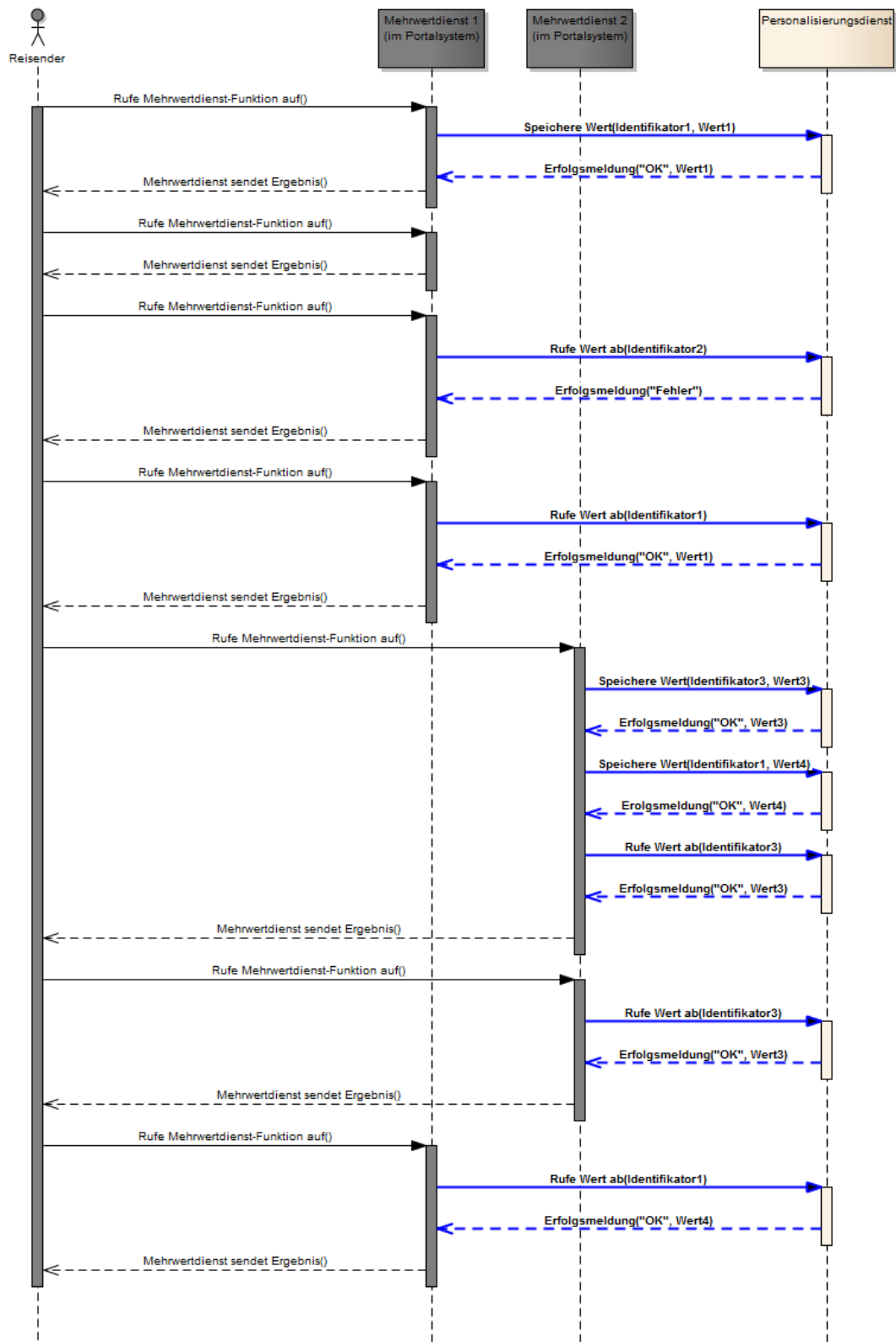


Abbildung 12: Sequenzdiagramm Personalisierung

20.3. Einfache Typen

Folgende einfache Typen (simple types) werden definiert:

<i>ValueIdType</i>	<i>xs:string</i>	Identifikator eines Wertes.
--------------------	------------------	-----------------------------

Tabelle 204: Liste der einfachen Typdefinitionen in Trias_Personalisation.xsd.

20.4. Anfragestrukturen

20.4.1. PersonalisationRequestStructure

PersonalisationRequest			+Structure	Repräsentiert eine Anfrage an den Personalisierungsdienst.	
	a	SaveValue	-1:1	+PersonalisationSaveValueRequest	Falls dies vorhanden ist, soll durch die Anfrage ein personalisierter Wert gespeichert werden; genauere Angaben dazu sind in dieser Struktur enthalten (vgl. 20.4.2).
	b	RetrieveValue		+PersonalisationRetrieveValueRequest	Falls dies vorhanden ist, soll durch die Anfrage ein personalisierter Wert abgerufen werden; genauere Angaben dazu sind in dieser Struktur enthalten (vgl. 20.4.3).
	c	DeleteValue		+PersonalisationDeleteValueRequest	Falls dies vorhanden ist, soll durch die Anfrage ein personalisierter Wert gelöscht werden; genauere Angaben dazu sind in dieser Struktur enthalten (vgl. 20.4.4).
	d	EnumerateValue		+PersonalisationEnumerateValuesRequest	Falls dies vorhanden ist, sollen durch die Anfrage verfügbare personalisierte Werte abgerufen werden; genauere Angaben dazu sind in dieser Struktur enthalten (vgl. 20.4.5).

Tabelle 205: Beschreibung der Struktur **PersonalisationRequestStructure**.

20.4.2. PersonalisationSaveValueRequestStructure

PersonalisationSaveValueRequest			+Structure	Enthält genauere Angaben zum Speichern eines personalisierten Wertes.
	ValueId	1:1	→ValueId	Der Identifikator des zu speichernden Wertes. Vgl. 20.3.
	Value	1:1	xs:string	Der zu speichernde Wert.

Tabelle 206: Beschreibung der Struktur **PersonalisationSaveValueRequestStructure**.

20.4.3. PersonalisationRetrieveValueRequestStructure

PersonalisationRetrieveValueRequest			+Structure	Enthält genauere Angaben zum Abrufen eines personalisierten Wertes.
	ValueId	1:1	→ValueId	Der Identifikator des abzurufenden Wertes. Vgl. 20.3.

Tabelle 207: Beschreibung der Struktur **PersonalisationRetrieveValueRequestStructure**.

20.4.4. PersonalisationDeleteValueRequestStructure

PersonalisationDeleteValueRequest			+Structure	Enthält genauere Angaben zum Löschen eines personalisierten Wertes.
	ValueId	1:1	→ValueId	Der Identifikator des zu löschenden Wertes. Vgl. 20.3.

Tabelle 208: Beschreibung der Struktur **PersonalisationDeleteValueRequestStructure**.

20.4.5. PersonalisationEnumerateValuesRequestStructure

PersonalisationEnumerateValuesRequest	+Structure	Drückt aus, dass eine Liste der gespeicherten Werte abgerufen werden soll.
--	------------	--

Tabelle 209: Beschreibung der Struktur **PersonalisationEnumerateValuesRequestStructure**.

20.5. Antwortstrukturen

20.5.1. PersonalisationResponseStructure

PersonalisationResponse			+Structure	Repräsentiert die Antwort auf eine Anfrage an den Personalisierungsdienst.
	<i>ErrorMessage</i>	0:*	+ErrorMessage	Enthält etwaige Fehlermeldungen, die die generelle Verarbeitung der Nachricht durch einen Personalisierungsdienst betreffen.
	a SaveValue	-1:1	+PersonalisationSaveValueResponse	Falls dies vorhanden ist, drückt die Antwort aus, ob das Speichern eines personalisierten Wertes erfolgreich war; genauere Angaben dazu sind in dieser Struktur enthalten (vgl. 20.5.2).
	b RetrieveValue		+PersonalisationRetrieveValueResponse	Falls dies vorhanden ist, drückt die Antwort aus, ob das Abrufen eines personalisierten Wertes erfolgreich war; genauere Angaben dazu sind in dieser Struktur enthalten (vgl. 20.5.3).
	c DeleteValue		+PersonalisationDeleteValueResponse	Falls dies vorhanden ist, drückt die Antwort aus, ob das Löschen eines personalisierten Wertes erfolgreich war; genauere Angaben dazu sind in dieser Struktur enthalten (vgl. 20.5.4).
	d EnumerateValue		+PersonalisationEnumerateValuesResponse	Falls dies vorhanden ist, drückt die Antwort aus, ob das Auflisten aller vorhandenen personalisierten Werte erfolgreich war; genauere Angaben dazu sind in dieser Struktur enthalten (vgl. 20.5.5).

Tabelle 210: Beschreibung der Struktur **PersonalisationResponseStructure**.

20.5.2. PersonalisationSaveValueResponseStructure

PersonalisationSaveValueResponse			+Structure	Enthält genauere Angaben zum abgeschlossenen Speichern eines personalisierten Wertes.
	<i>ErrorMessage</i>	0:*	+ErrorMessage	Gibt an, ob der Speichervorgang erfolgreich war.
	<i>Value</i>	0:1	xs:string	Der soeben gespeicherte Wert.

Tabelle 211: Beschreibung der Struktur **PersonalisationSaveValueResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

PERSONALISATIONSVALUEQUEST_INVALID_ID	Der angegebene Identifikator hat ein ungültiges Format.
--	---

Tabelle 212: Liste der Fehlerzustände in **PersonalisationSaveValueResponseStructure**.

20.5.3. PersonalisationRetrieveValueResponseStructure

PersonalisationRetrieveValueResponse			+Structure	Enthält genauere Angaben zum abgeschlossenen Abrufen eines personalisierten Wertes.
	ErrorMessage	0:*	+ErrorMessage	Gibt an, ob der Abruf erfolgreich war.
	Value	0:1	xs:string	Falls der Abruf erfolgreich war, der abgerufene Wert.

Tabelle 213: Beschreibung der Struktur **PersonalisationRetrieveValueResponseStructure**.

In **ErrorMessage** können folgende Fehlerzustände auftreten:

PERSONALISATIONRETRIEVEVALUERE-QUEST_UNKNOWN_ID	Zum angegebenen Identifikator ist im Personalisierungsdienst kein Wert abgelegt.
--	--

Tabelle 214: Liste der Fehlerzustände in **PersonalisationRetrieveValueResponseStructure**.

20.5.4. PersonalisationDeleteValueResponseStructure

PersonalisationDeleteValueResponse			+Structure	Enthält genauere Angaben zum abgeschlossenen Löschen eines personalisierten Wertes.
	ErrorMessage	0:*	+ErrorMessage	Gibt an, ob die Löschung erfolgreich war.

Tabelle 215: Beschreibung der Struktur **PersonalisationDeleteValueResponseStructure**.

In **ErrorMessage** können folgende Fehlerzustände auftreten:

PERSONALISATIONDELETEVALUERE-QUEST_UNKNOWN_ID	Zum angegebenen Identifikator ist im Personalisierungsdienst kein Wert abgelegt.
--	--

Tabelle 216: Liste der Fehlerzustände in **PersonalisationDeleteValueResponseStructure**.

20.5.5. PersonalisationEnumerateValuesResponseStructure

PersonalisationEnumerateValuesResponse			+Structure	Enthält genauere Angaben zur Auflistung aller gespeicherten personalisierten Werte.
	ErrorMessage	0:*	+ErrorMessage	Gibt an, ob der Abruf erfolgreich war.
	ValueId	0:*	→ValueId	Falls der Abruf erfolgreich war, ist für jeden gespeicherten Wert ein solches Element vorhanden, welches jeweils einen Identifikator eines gespeicherten Wertes enthält. Vgl. 20.3.

Tabelle 217: Beschreibung der Struktur **PersonalisationEnumerateValuesResponseStructure**.

21. Dienst Fahrzeuginformationen

21.1. Beschreibung

Dieser Dienst dient dazu, dass zwischen einem Fahrzeug und einer mobilen Applikation, die von einem Fahrgast benutzt wird, Informationen, die das Fahrzeug betreffen, ausgetauscht werden können.

21.2. Anfragestrukturen

21.2.1. VehicleDataRequestStructure

VehicleDataRequestStructure			+Structure	Fasst die Abfragedaten für eine Abfrage der Fahrzeugdaten zusammen.
	<i>VehicleStatus</i>	0:1	<i>xs:boolean</i>	Es sollen vom Fahrzeug Fahrzeug-Status-Informationen geschickt werden.
	<i>VehicleActivity</i>	0:1	<i>xs:boolean</i>	Es sollen vom Fahrzeug Fahrzeug-Aktivitäts-Informationen geschickt werden.

Tabelle 218: Beschreibung der Struktur **VehicleDataRequestStructure**.

21.3. Antwortstrukturen

21.3.1. VehicleDataResponseStructure

VehicleDataResponseStructure			+Structure	Fasst die Ergebnisdaten für eine Abfrage der Fahrzeugdaten zusammen
	VehicleCode	1:1	→ <i>VehicleCode</i>	Eindeutige Fahrzeug-ID, mit der eine Referenzierung auf die EKAP-Daten ermöglicht werden soll, vgl. 7.4.1.
	<i>VehicleStatus</i>	0:1	+ <i>VehicleStatus</i>	Informationen, die den Zustand des Fahrzeugs betreffen, siehe 21.3.2.
	<i>VehicleActivity</i>	0:1	+ <i>VehicleActivity</i>	Informationen, die die Aktivität des Fahrzeugs betreffen, siehe 21.3.3.

Tabelle 219: Beschreibung der Struktur **VehicleDataResponseStructure**.

21.3.2. VehicleStatusStructure

VehicleStatusStructure			+Structure	Informationen, die den Zustand des einzelnen Fahrzeugs betreffen.
	<i>DoorState</i>	0:1	<i>DoorsOpen AllDoorsClosed</i>	Informationen über den Türzustand.
	<i>VehicleStopRequested</i>	0:1	<i>xs:boolean</i>	Information darüber, ob bereits ein Haltewunsch für die kommende Haltestelle dem Fahrzeugsystem bekannt ist.
	<i>VehicleTypeRef</i>	0:1	→ <i>VehicleType</i>	Informationen über den Fahrzeugtyp und damit der Ausstattung des Fahrzeugs, vgl. 7.4.1.

Tabelle 220: Beschreibung der Struktur **VehicleStatusStructure**.

21.3.3. VehicleActivityStructure

VehicleActivityStructure			+Structure	Informationen, die die Aktivität des einzelnen Fahrzeugs betreffen.
	<i>TimetableDelay</i>	0:1	<i>xs:int</i>	Abweichung gegenüber dem Fahrplan in Sekunden, Verfrühungen werden als negative Werte dargestellt.
	<i>RouteDeviation</i>	0:1	<i>onroute offroute unknown</i>	Information darüber, ob das Fahrzeug sich auf dem geplanten Linienweg befindet oder nicht.
	<i>JourneyMode</i>	0:1	<i>NoTrip AdditionalTrip ServiceTrip</i>	Information über die Art der Fahrt, die das Fahrzeug durchführt (Planfahrt, Verstärkerfahrt u. ä.).
<i>ServicePattern</i>	<i>StopSequence</i>	0:*	+ <i>StopSequence</i>	Informationen zur Haltepunktfolge, die benötigt wird, um bspw. eine Perlschnur darzustellen, siehe

<i>Position</i>				21.3.4.
	<i>CurrentStopIndex</i>	0:1	<i>xs:int</i>	Index über den nächsten angefahrenen Haltepunkt in der Haltepunktfolge.
	<i>LocationState</i>		<i>AfterStop AtStop BeforeStop BetweenStop</i>	Information darüber, ob das Fahrzeug sich noch an der Haltestelle befindet, kurz davor oder dahinter oder zwischen zwei Halten.
	<i>NextExitSide</i>	0:1	<i>both left right unknown</i>	Information über die Ausstiegsseite am nächsten Haltepunkt.

Tabelle 221: Beschreibung der Struktur **VehicleActivityStructure**.

21.3.4. StopSequenceStructure

StopSequenceStructure			+Structure	Informationen zur Haltepunktfolge.
	StopPoint	2:*	+StopInformation	Informationen über den Haltepunkt, siehe 21.3.5.

Tabelle 222: Beschreibung der Struktur **StopSequenceStructure**.

21.3.5. StopInformationStructure

StopInformationStructure			+Structure	Informationen über den einzelnen Haltepunkt.
	StopIndex	1:1	<i>xs:int</i>	Index des aktuellen Haltepunkts in der Haltepunktfolge.
	<i>StopRef</i>	0:1	→<i>StopPoint</i>	Referenz auf den Haltepunkt, siehe 7.5.1.
	StopName	1:*	+<i>InternationalText</i>	Name des Haltepunkts.
	<i>StopAlternativeName</i>	0:*	+<i>InternationalText</i>	Alternativbezeichnung des Haltepunkts.
	<i>Platform</i>	0:*	<i>xs:string</i>	Bezeichnung des Haltestellensteiges
	<i>DisplayContent</i>	0:*	+<i>DisplayContent</i>	Information über die Zusammensetzung des Zieltextinhalts, siehe 21.3.6.
	<i>ArrivalScheduled</i>	0:1	<i>xs:dateTime</i>	Angabe der geplanten Ankunftszeit.
	<i>DepartureScheduled</i>	0:1	<i>xs:dateTime</i>	Angabe der geplanten Abfahrtszeit.
	<i>RecordedArrivalTime</i>	0:1	<i>xs:dateTime</i>	Information über die tatsächliche Ankunftszeit (wird für den Feldtest in Stuttgart und während des Migrationszeitraums benötigt).
	<i>DistanceToNextStop</i>	0:1	<i>xs:double</i>	Abstand zum nächsten Halt in [m].
	<i>AnnouncementNextStop</i>	0:*	+<i>Announcement</i>	Informationen über die Haltestellenansage, siehe 21.3.10.
	<i>Farezone</i>	0:*	<i>xs:NMTOKEN</i>	Informationen über die Tarifzonen, in denen dieser Haltepunkt liegt.
	<i>Connection</i>	0:*	+<i>Connection</i>	Informationen über Anschlüsse, siehe 21.3.11.

Tabelle 223: Beschreibung der Struktur **StopInformationStructure**.

21.3.6. DisplayContentStructure

DisplayContentStructure			+Structure	Information über die Zusammensetzung des einzelnen Zieltextinhalts.
	<i>Line</i>	0:1	+<i>LineInformation</i>	Informationen zur Bezeichnung der Linie, siehe 21.3.7.
	Destination	1:1	+<i>Destination</i>	Informationen über den Inhalt des Zieltextes, siehe 21.3.8.

	<i>Via</i>	0:*	+ <i>ViaPoint</i>	Informationen über Zwischenhaltestellen, siehe 21.3.9.
	<i>AdditionalInformation</i>	0:*	+ <i>InternationalText</i>	Zusatzinformationen wie „Eilkurs“, „Entlastungsbuss“ etc.
<i>DisplayPolicy</i>	<i>PeriodDuration</i>	1:1	<i>xs:duration</i>	Information über die Periodendauer bei Wechsel zwischen verschiedenen Anzeigeeinhalten.
	<i>Duration</i>	1:1	<i>xs:duration</i>	Information über die Anzeigedauer dieses Anzeigeeinhalts innerhalb einer Anzeigeperiode (bei Wechsel zwischen verschiedenen Anzeigeeinhalten).

Tabelle 224: Beschreibung der Struktur ***DisplayContentStructure***.

21.3.7. LineInformationStructure

<i>LineInformationStructure</i>			+ <i>Structure</i>	Informationen zur Bezeichnung der Linie.
	<i>LineRef</i>	1:1	→ <i>Line</i>	Referenz auf eine Linie, siehe 7.4.1.
	<i>LineName</i>	1:*	+ <i>InternationalText</i>	Name der Linie.
	<i>LineShortName</i>	0:*	+ <i>InternationalText</i>	Kurzbezeichnung der Linie.
	<i>LineNumber</i>	1:1	<i>xs:int</i>	Nummer der Linie.

Tabelle 225: Beschreibung der Struktur ***LineInformationStructure***

21.3.8. DestinationStructure

<i>DestinationStructure</i>			+ <i>Structure</i>	Informationen über den Inhalt des Zieltextes.
	<i>DestinationRef</i>	1:1	<i>xs:NMTOKEN</i>	Index des Zieltextes.
	<i>DestinationName</i>	0:*	+ <i>InternationalText</i>	Zieltext.
	<i>DestinationShortName</i>	0:*	+ <i>InternationalText</i>	Zieltextkurzbezeichnung.

Tabelle 226: Beschreibung der Struktur ***DestinationStructure***

21.3.9. ViaPointStructure

<i>ViaPointStructure</i>			+ <i>Structure</i>	Informationen über Zwischenhaltestellen.
	<i>ViaPointRef</i>	1:1	<i>xs:int</i>	Index des Haltepunkts innerhalb der Auflistung der Zwischenhaltepunkte.
	<i>PlaceRef</i>	0:1	→ <i>StopPoint</i>	Referenz auf den Haltepunkt, siehe 7.5.1.
	<i>PlaceName</i>	0:*	+ <i>InternationalText</i>	Bezeichnung des Zwischenhaltepunktes.
	<i>PlaceShortName</i>	0:*	+ <i>InternationalText</i>	Kurzbezeichnung des Zwischenhaltepunktes.
	<i>ViaPointDisplayPriority</i>	0:1	<i>xs:nonNegativeInteger</i>	Anzeigepriorität des Zwischenhaltepunktes.

Tabelle 227: Beschreibung der Struktur ***ViaPointStructure***

21.3.10. AnnouncementStructure

<i>AnnouncementStructure</i>			+ <i>Structure</i>	Informationen über die einzelne Haltestellenansage
	<i>AnnouncementRef</i>	1:1	<i>xs:NMTOKEN</i>	Index der Ansage.
	<i>AnnouncementText</i>	0:*	+ <i>International</i>	Ansagetext als für den Fahrgast lesbare Information.

			<i>Text</i>	
	<i>AnnouncementTTSText</i>	0:*	<i>+InternationalText</i>	Ansagetext, für ein TextToSpeech-System.

Tabelle 228: Beschreibung der Struktur **AnnouncementStructure**

21.3.11. ConnectionStructure

ConnectionStructure		<i>+Structure</i>	Information über die einzelnen Anschlüsse an einer Haltestelle inkl. Nachbarhaltestellen.	
	ConnectionRef	1:1	<i>xs:NMTOKEN</i>	Haltepunktsbezogener Index über alle verfügbaren Anschlüsse.
	ConnectionType	1:1	<i>Interchange ProtectedConnection</i>	Art des Anschlusses (gesicherter Anschluss oder einfache Umsteigebeziehung).
	<i>ConnectionStop</i>	0:1	<i>→StopPoint</i>	Referenz auf einen Nachbarhaltepunkt, siehe 7.5.1. Falls nicht vorhanden, findet der Anschluss am selben Haltepunkt statt.
	DisplayContent	1:1	<i>+DisplayContent</i>	Information über die Zusammensetzung des Zieltextinhalts, siehe 21.3.6.
	Platform	1:1	<i>xs:string</i>	Abfahrtsstelle des Abbringers.
	<i>WalkDuration</i>	0:1	<i>xs:duration</i>	Durchschnittliche Fußwegzeit zum Abfahrtshaltepunkt des Anschlusses.
	ConnectionState	1:1	<i>ConnectionOK ConnectionBroken NoInformationAvailable</i>	Information darüber, ob der Anschluss gehalten werden kann oder nicht.
	<i>Transportmode</i>	0:1	<i>+VehicleType</i>	Information über die Art des Abbringer-Fahrzeugs, siehe 21.3.12.
	<i>ExpectedDepartureTime</i>	0:1	<i>xs:dateTime</i>	erwartete Abfahrtszeit des Abbringers.

Tabelle 229: Beschreibung der Struktur **ConnectionStructure**

21.3.12. VehicleTypeStructure

VehicleTypeStructure		<i>+Structure</i>	Information über die Art eines Fahrzeugs.	
	VehicleTypeRef	1:1	<i>→VehicleType</i>	Informationen über den Fahrzeugtyp und damit der Ausstattung des Fahrzeugs, vgl. 7.4.1.
	<i>VehicleName</i>	0:*	<i>+InternationalText</i>	Bezeichner des Fahrzeugtyps.

Tabelle 230: Beschreibung der Struktur **VehicleTypeStructure**

22. Dienst Haltewunsch an Fahrzeug

22.1. Beschreibung

Dieser Dienst dient der Übertragung eines Haltewunsches von einer mobilen Applikation eines Fahrgasts an das Fahrzeug.

22.2. Anfragestrukturen

22.2.1. StopRequestRequestStructure

<i>StopRequestRequestStructure</i>			<i>+Structure</i>	Fasst die Informationen über die Anfragestruktur bei einem Haltewunsch an das Fahrzeug zusammen.
	<i>StopRef</i>	1:1	<i>→StopPoint</i>	Referenz auf die gewünschte Ausstiegshaltestelle, vgl. 7.5.1.
	<i>StopName</i>	0:*	<i>+International Text</i>	Bezeichner der Ausstiegshaltestelle.
	<i>DisabledPerson</i>	0:1	<i>xs:boolean</i>	Information darüber, ob die anfragende Person mobilitätseingeschränkt ist.

Tabelle 231: Beschreibung der Struktur *StopRequestRequestStructure*

22.3. Antwortstrukturen

22.3.1. StopRequestResponseStructure

<i>StopRequestResponseStructure</i>			<i>+Structure</i>	Fasst die Informationen über die Antwortstruktur bei einem Haltewunsch an das Fahrzeug zusammen.
	<i>StopRequestReceived</i>	1:1	<i>xs:boolean</i>	Information darüber, dass der Haltewunsch angekommen ist.
	<i>RequestedStop</i>	0:1	<i>xs:NMTOKEN</i>	Referenz auf die gewünschte Ausstiegshaltestelle, vgl. 7.5.1.
	<i>EstimatedArrivalTime</i>	0:1	<i>xs:dateTime</i>	Prognostizierte Ankunftszeit an der Ausstiegshaltestelle.

Tabelle 232: Beschreibung der Struktur *StopRequestResponseStructure*

23. Dienst Diensteregister

23.1. Beschreibung

Der Diensteregisterdienst führt Buch über verfügbare TRIAS-Dienste.

23.2. Einfache Typen

Folgende einfache Typen (simple types) werden definiert:

<i>TriasServiceIdType</i>	<i>xs:NMTOKEN</i>	ID einer Instanz eines TRIAS-Dienstes.
<i>InterfaceVersionType</i>	<i>xs:NMTOKEN</i>	Versionsnummer einer Dienstschnittstellendefinition.
<i>TriasServiceTypeEnumeration</i>	<i>Alerts BookingInfo ConnectionDemand Facilities Fares IndividualTrips Locations Maps Positioning ServiceRegister StopEvents TripInfo Trips</i>	Art des Dienstes.
<i>ServiceAddressType</i>	<i>xs:anyURI</i>	Adresse (URL) eines Online-Dienstes.
<i>ServiceUsageEnumeration</i>	<i>Consumer Provider</i>	Verwendung des Dienstes als Anbieter oder Client.

Tabelle 233: Beschreibung von einfachen Typen

23.3. Anfragestrukturen

Eine Anfrage an das Diensteregister wird mittels eines Elements **ServiceRegisterRequest** vom Typ *ServiceRegisterRequestStructure* gestellt.

23.3.1. ServiceRegisterRequestStructure

Mit einer Anfrage vom Typ *ServiceRegisterRequestStructure* kann ein TRIAS-Dienst im Dienstregister wahlweise aufgenommen, gelöscht oder aktualisiert werden, oder es können alle registrierten Dienste gesucht werden, die auf die angegebenen Filterkriterien passen.

<i>ServiceRegisterRequestStructure</i>			+Structure	Fasst die Abfragedaten an das Diensteregister zusammen.	
	a	RegisterRequest	-1:1	+ServiceRegisterRegisterRequest	Anfrage, um eine Dienstinstanz im Dienstregister anzumelden. Vgl. 23.3.3.
	b	UpdateRequest		+ServiceRegisterUpdateRequest	Anfrage, um eine Dienstinstanz im Dienstregister zu aktualisieren. Vgl. 23.3.4.
	c	LookupRequest		+ServiceRegisterLookupRequest	Anfrage, um nach passenden Diensten im Dienstregister zu suchen. Vgl. 23.3.5.
	d	UnregisterRequest		+ServiceRegisterUnregisterRequest	Anfrage, um eine Dienstinstanz im Dienstregister zu löschen. Vgl. 23.3.6.
	Params		0:1	+ServiceRegisterParam	Anfrageparameter. Vgl. 23.3.2.

Tabelle 234: Beschreibung der Struktur **ServiceRegisterRequestStructure**.

23.3.2. ServiceRegisterParamStructure

<i>ServiceRegisterParamStructure</i>			+Structure	Fasst die Parameter für eine Anfrage an das Diensteregister zusammen.
	Extension	0:1	xs:anyType	Erweiterungen.

Tabelle 235: Beschreibung der Struktur **ServiceRegisterParamStructure**.

23.3.3. ServiceRegisterRegisterRequestStructure

<i>ServiceRegisterRegisterRequestStructure</i>			+Structure	Anfrage an das Diensteregister, um einen Dienst anzumelden.
<i>TriasServiceProperties</i>	ServiceType	0:1	<i>TriasServiceTypeEnumeration</i>	Art des Dienstes.
	Version	0:1	<i>InterfaceVersion</i>	Versionsnummer der Dienstschnittstellendefinition.
	ServiceAddress	0:1	<i>ServiceAddresses</i>	Adresse (URL) eines Online-Dienstes.
	ParticipantRef	0:1	→ <i>ParticipantCode</i>	ID eines Kommunikationspartners. Vgl. 7.4.1.
	ServiceUsage	0:1	<i>Consumer / Provider</i>	Verwendung des Dienstes als Anbieter oder Client.

Tabelle 236: Beschreibung der Struktur **ServiceRegisterRegisterRequestStructure**.

23.3.4. ServiceRegisterUpdateRequestStructure

ServiceRegisterUpdateRequestStructure			+Structure	Anfrage an das Diensteregister, um die Eintragungen zu einem Dienst zu aktualisieren.
	ServiceId	1:1	<i>TriasServiceId</i>	ID des Diensts, der aktualisiert werden soll.
<i>TriasServiceProperties</i>	<i>ServiceType</i>	0:1	<i>TriasServiceTypeEnumeration</i>	Art des Dienstes.
	<i>Version</i>	0:1	<i>InterfaceVersion</i>	Versionsnummer der Dienstschnittstellendefinition.
	<i>ServiceAddress</i>	0:1	<i>ServiceAddresses</i>	Adresse (URL) eines Online-Dienstes.
	<i>ParticipantRef</i>	0:1	→ <i>ParticipantCode</i>	ID eines Kommunikationspartners. Vgl. 7.4.1.
	<i>ServiceUsage</i>	0:1	<i>Consumer Provider</i>	Verwendung des Dienstes als Anbieter oder Client.

Tabelle 237: Beschreibung der Struktur **ServiceRegisterUpdateRequestStructure**.

23.3.5. ServiceRegisterLookupRequestStructure

ServiceRegisterLookupRequestStructure			+Structure	Anfrage an das Diensteregister, um passende Dienste zu finden.
	<i>ServiceId</i>	0:1	<i>TriasServiceId</i>	ID des Diensts, nach dem gesucht wird.
<i>TriasServiceProperties</i>	<i>ServiceType</i>	0:1	<i>TriasServiceTypeEnumeration</i>	Art des Dienstes.
	<i>Version</i>	0:1	<i>InterfaceVersion</i>	Versionsnummer der Dienstschnittstellendefinition.
	<i>ServiceAddress</i>	0:1	<i>ServiceAddresses</i>	Adresse (URL) eines Online-Dienstes.
	<i>ParticipantRef</i>	0:1	→ <i>ParticipantCode</i>	ID eines Kommunikationspartners. Vgl. 7.4.1.
	<i>ServiceUsage</i>	0:1	<i>Consumer Provider</i>	Verwendung des Dienstes als Anbieter oder Client.

Tabelle 238: Beschreibung der Struktur **ServiceRegisterLookupRequestStructure**.

23.3.6. ServiceRegisterUnregisterRequestStructure

ServiceRegisterUnregisterRequestStructure			+Structure	Anfrage an das Diensteregister, um einen Dienst zu löschen.
	<i>ServiceId</i>	0:1	<i>TriasServiceId</i>	ID des Diensts, der gelöscht werden soll.

Tabelle 239: Beschreibung der Struktur **ServiceRegisterUnregisterRequestStructure**.

23.4. Antwortstrukturen

Das Ergebnis einer Dienstregisteranfrage wird mittels eines Elements **ServiceRegisterResponse** vom Typ *ServiceRegisterResponseStructure* übertragen.

23.4.1. ServiceRegisterResponseStructure

ServiceRegisterResponseStructure			+Structure	Fasst die Ergebnisdaten für eine Dienstregisteranfrage zusammen.
	<i>ErrorMessage</i>	0:*	+ <i>ErrorMessage</i>	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Ta-

				belle für mögliche Werte. Vgl. 7.4.2.
	<i>ServiceRegisterResult</i>	0:1	+ <i>ServiceRegisterResult</i>	Struktur für ein Dienstregisterresultat. Vgl. 23.4.2.

Tabelle 240: Beschreibung der Struktur **ServiceRegisterResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

SERVICEREGIS- TER_SERVICEIDUNKNOWN	Die Anfrage an das Dienstregister enthält eine unbekannte Service-ID.
SERVICEREGISTER_NOMATCH	Die Suchanfrage an das Dienstregister liefert keine Übereinstimmung.
SERVICEREGISTER_TOOMANYMATCHES	Die Suchanfrage an das Dienstregister liefert zu viele Übereinstimmungen

Tabelle 241: Liste der Fehlerzustände in **ServiceRegisterResponse**.

23.4.2. ServiceRegisterResultStructure

ServiceRegisterResultStructure		+Structure	Resultatstruktur für die Dienstregisteranfrage.
	ResultId	1:1	<i>xs:NMTOKEN</i> ID des Resultats für spätere Referenzierung.
	<i>ErrorMessage</i>	0:*	+ <i>ErrorMessage</i> Fehlermeldungen bezogen auf die Anfrage an das Dienstregister. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
a	RegisterResponse	-1:1	+ <i>ServiceRegisterResponse</i> Antwort zur Dienstregistrierung. Vgl. 23.4.3.
b	UpdateResponse		+ <i>ServiceRegisterUpdateResponse</i> Antwort zur Dienstaktualisierung. Vgl. 23.4.4.
c	LookupResponse		+ <i>ServiceRegisterLookupResponse</i> Antwort zur Suche nach passenden Diensten. Vgl. 23.4.5.
d	UnregisterResponse		+ <i>ServiceRegisterUnregisterResponse</i> Antwort zur Dienstlöschung. Vgl. 23.4.6.

Tabelle 242: Beschreibung der Struktur **ServiceRegisterResultStructure**.

Die Verwendung von *ErrorMessage* in *ServiceRegisterResultStructure* ist für spätere Erweiterungen reserviert.

23.4.3. ServiceRegisterRegisterResponseStructure

ServiceRegisterRegisterResponseStructure		+Structure	Antwort auf die Dienstregistrierung.
	ServiceId	1:1	<i>TriasServiceId</i> ID des Diensts, wie er im Register geführt wird. Diese ID muss bei Folgeanfragen an das Dienstregister verwendet werden.

Tabelle 243: Beschreibung der Struktur **ServiceRegisterRegisterResponseStructure**.

23.4.4. ServiceRegisterUpdateResponseStructure

ServiceRegisterUpdateResponseStructure		+Structure	Antwort auf die Dienstaktualisierung.
	ServiceId	1:1	<i>TriasServiceId</i> ID des Diensts, wie er im Register geführt wird. Diese ID muss bei Folgeanfragen an das Dienstregister verwendet werden.

Tabelle 244: Beschreibung der Struktur **ServiceRegisterUpdateResponseStructure**.

23.4.5. ServiceRegisterLookupResponseStructure

<i>ServiceRegisterLookupResponseStructure</i>		+Structure	Antwort auf eine Dienstsuche.
	<i>Service</i>	1:*	<i>TriasService</i>
			Ein oder mehrere Dienste, die auf die Anfragekriterien passen. Vgl. 23.4.7.

Tabelle 245: Beschreibung der Struktur *ServiceRegisterLookupResponseStructure*.

23.4.6. ServiceRegisterUnregisterResponseStructure

<i>ServiceRegisterUnregisterResponseStructure</i>		+Structure	Antwort auf eine Dienstlöschung aus dem Register.
			Das Antwortelement bleibt bis auf weiteres leer.

Tabelle 246: Beschreibung der Struktur *ServiceRegisterUnregisterResponseStructure*.

23.4.7. TriasServiceStructure

<i>TriasServiceStructure</i>		+Structure	Definition einer Instanz eines TRIAS-Dienstes.
	<i>ServiceId</i>	1:1	<i>TriasServiceId</i>
	<i>ServiceType</i>	0:1	<i>TriasServiceTypeEnumeration</i>
<i>TriasServiceProperties</i>	<i>Version</i>	0:1	<i>InterfaceVersion</i>
	<i>ServiceAddress</i>	0:1	<i>ServiceAddresses</i>
	<i>ParticipantRef</i>	0:1	<i>ParticipantCode</i>
	<i>ServiceUsage</i>	0:1	<i>Consumer Provider</i>
			ID der Instanz.
			Art des Dienstes.
			Versionsnummer der Dienstschnittstellendefinition.
			Adresse (URL) eines Online-Dienstes.
			ID eines Kommunikationspartners. Vgl. 7.4.1.
			Verwendung des Dienstes als Anbieter oder Client.

Tabelle 247: Beschreibung der Struktur *TriasServiceStructure*.

24. Dienst Authentifizierung

Einige Teile der TRIAS-Schnittstelle werden zum Austausch von wichtigen betrieblichen Daten verwendet. Um die Unverfälschtheit dieser Daten sicherzustellen, sind Mechanismen zur Authentifizierung und zur Autorisierung der Schnittstellenpartner notwendig. In diesem Kapitel soll daher ein Mechanismus beschrieben werden, um die Authentizität von Schnittstellenpartnern überprüfen zu können.

Zur Authentifizierung der Schnittstellenpartner kommt ein PKI-Verfahren zum Einsatz. Es wird die Verwendung des DSA-Verfahrens⁶ empfohlen. Jedoch können auch andere Verfahren zum Einsatz kommen. Die Kommunikationspartner müssen sich dazu bilateral abstimmen.

⁶ Digital Signature Standard: http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf

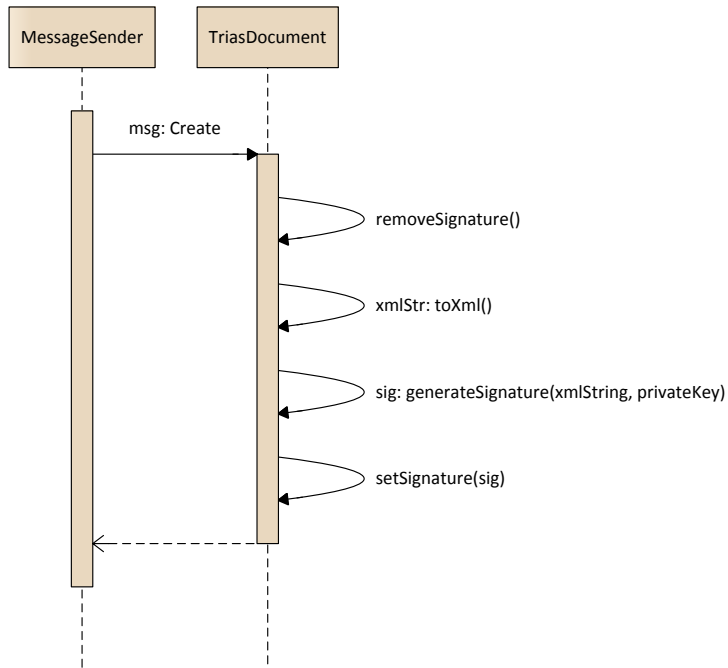


Abbildung 13: Erzeugung einer Nachrichtensignatur

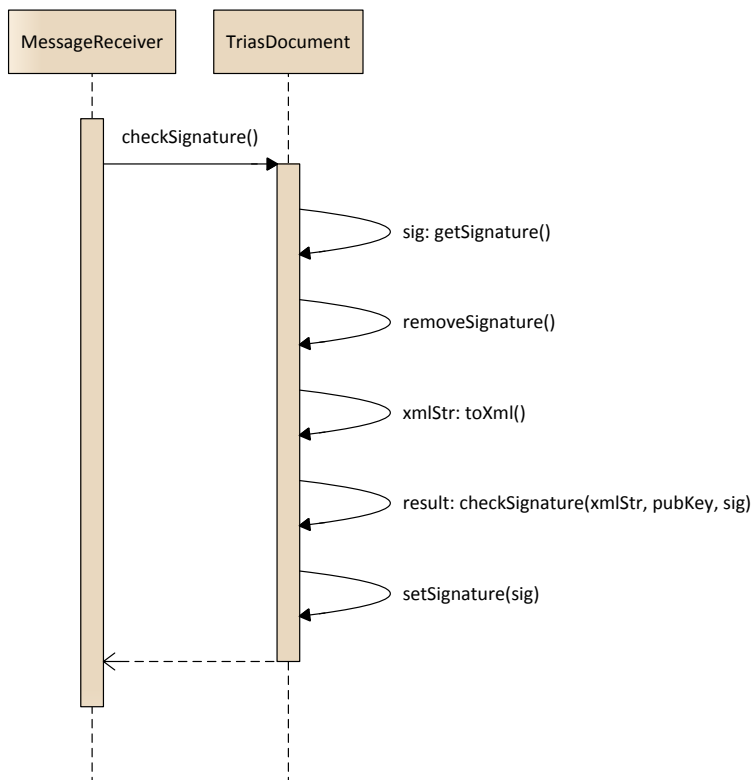


Abbildung 14: Überprüfung einer Signatur

Der Authentifizierungsdienst des Empfängers überprüft die CertificateId und die SignatureId aus dem ServiceRequest (vgl. die Definition der Nachrichteneigenschaften in 7.9.2). Das Feld CertificateId enthält eine Referenz auf diesen Schlüssel. Der Schlüssel muss also vorab ausgetauscht werden. Das Feld SignatureId enthält die Signatur der

versendeten TRIAS-Nachricht. Zur Berechnung der Signatur wird die XML-Nachricht in der kanonischen Normalform⁷ verwendet. Bei der Berechnung der Signatur darf das Feld SignatureId in der XML-Nachricht nicht vorhanden sein, es muss also vom Empfänger wieder entfernt werden. Dabei muss die kanonische Normalform erhalten bleiben.

Der Ablauf der Signierung einer Nachricht ist schematisch in Abbildung 13 dargestellt. Zunächst wird das Feld SignatureId komplett aus der Nachricht entfernt. Dann wird die Nachricht in ihre XML-Repräsentation in der kanonischen Normalform exportiert. Mithilfe der XML-Repräsentation und dem privaten Schlüssel kann nun die korrekte Signatur der Nachricht berechnet und im Feld SignatureId gespeichert werden. Das Ergebnis ist die korrekt signierte, und vor Modifikation durch Dritte geschützte Nachricht.

Der schematische Ablauf der Signaturprüfung ist in Abbildung 14 dargestellt. Zunächst wird das Feld SignatureId aus der Nachricht entfernt und die Nachricht in ihre kanonische XML-Darstellung überführt. Anschließend kann mithilfe des öffentlichen Schlüssels des Nachrichtensenders, der Nachricht in XML-Darstellung und der Signatur überprüft werden, ob diese Signatur gültig ist. Anschließend muss, um die Nachricht nicht zu verfälschen, die Signatur wieder auf ihren ursprünglichen Wert gesetzt werden.

⁷ RFC 3076: <http://www.ietf.org/rfc/rfc3076.txt>

25. Normative Referenzen

- CEN-IFOPT (CEN, EN 28701:2012, 2012)
- ISO 8601 Zeitformate (ISO 8601:2004, 2004)

26. Begriffe

Es gelten die in VDV-Schrift 430 Teil 1, 2013 und in VDV-Schrift 431, Teil 1, 2013 festgelegten Begriffsbestimmungen.

27. Abkürzungen

Es gelten die in VDV-Schrift 430 Teil 1, 2013 und in VDV-Schrift 431, Teil 1, 2013 angegebenen Abkürzungen. Darüber hinaus werden die in Tabelle 248 aufgeführten Abkürzungen verwendet.

Begriff	Beschreibung
HTTP	Hypertext Transfer Protocol. Übertragungsprotokoll basierend auf TCP/IP, wird vor allem im Internet zum Austausch von Informationen benutzt.
IFOPT	Identification of Fixed Objects in Public Transport (CEN, EN 28701:2012, 2012)
JourneyWeb	Britischer Standard zur Verknüpfung von regionalen Fahrplanauskunftssystemen für eine Großbritannienweite nationale Auskunft (Department for Transport, 2012)
Transmodel	Reference Data Model for Public Transport (CEN, EN 12896:2006, 2006)
TRIAS	Travellers' Realtime Information and Advisory Standard. Mit diesem Akronym wird die Familie der in diesem Dokument definierten Schnittstellendienste bezeichnet.
UML	Unified Modeling Language Standardisierte, grafische Modellierungssprache zur Spezifikation von Software.

Tabelle 248: Liste der Abkürzungen.

28. Abbildungsverzeichnis

Abbildung 1: Umfeld und Schwerpunkte im Projekt IP-KOM-ÖV	15
Abbildung 2: Anfrage mit synchroner Antwort (Abbildung entnommen aus SIRI, (CEN, TS 15531, Part 2)).....	21

Abbildung 3: Abonnement-Verfahren mit asynchronen Benachrichtigungen (Abbildung entnommen aus SIRI, (CEN, TS 15531, Part 2)).....	22
Abbildung 4: Gesamtübersicht über die Hierarchie der Schemadateien.....	32
Abbildung 6: Rechter Teil der Schema-Hierarchie.	33
Abbildung 5: Linker Teil der Schema-Hierarchie.	33
Abbildung 7: Ablauf der Anschlussdienste mit aktiver Benachrichtigung bei Statusänderung.....	88
Abbildung 8: Ablauf der Anschlussdienste mit passiver Benachrichtigung bei Statusänderung.....	88
Abbildung 9: Funktionsweise des Benachrichtigungsdienstes.....	115
Abbildung 10: Sequenzdiagramm Lebenszyklus eines Wertes	118
Abbildung 11: Sequenzdiagramm Werteliste ermitteln	119
Abbildung 12: Sequenzdiagramm Personalisierung	121
Abbildung 13: Erzeugung einer Nachrichtensignatur	134
Abbildung 14: Überprüfung einer Signatur	134

29. Tabellenverzeichnis

Tabelle 1: Beispiel (aus Abschnitt 7.6) für die tabellarische Notation einer XML- Struktur	19
Tabelle 2: Liste der TRIAS-Dienste und ihrer Anfrageelemente.....	31
Tabelle 3: Liste der Fehlercodes, wie sie in SIRI für das Nachrichtenübermittlungsverfahren definiert werden.....	35
Tabelle 4: Generische TRIAS-Fehlermeldungen, die in allen Nachrichten auftreten können.....	35
Tabelle 5: Beschreibung der Struktur ServiceRequestStructure	36
Tabelle 6: Beschreibung der Struktur SubscriptionRequestStructure	36
Tabelle 7: Beschreibung der Struktur RequestPayloadStructure	37
Tabelle 8: Beschreibung der Struktur ServiceDeliveryStructure	38
Tabelle 9: Beschreibung der Struktur DeliveryPayloadStructure	39
Tabelle 10: Liste der einfachen Typdefinitionen in Trias_Utility.xsd.....	39
Tabelle 11: Beschreibung der Struktur InternationalTextStructure	40
Tabelle 12: Beschreibung der Struktur GeoPositionStructure	40
Tabelle 13: Liste der einfachen Typdefinitionen in Trias_ModesSupport.xsd.....	42
Tabelle 14: Beschreibung der Struktur IndividualTransportOptionsStructure	42
Tabelle 15: Beschreibung der Gruppe PtSubmodeChoiceGroup	43

Tabelle 16: Beschreibung der Struktur ModeStructure	43
Tabelle 17: Beschreibung der Struktur PtModeFilterStructure	43
Tabelle 18: Liste der einfachen Typdefinitionen in Trias_Common.xsd	44
Tabelle 19: Beschreibung der Struktur ErrorMessageStructure	44
Tabelle 20: Beschreibung der Struktur PrivateCodeStructure	44
Tabelle 21: Beschreibung der Struktur OperatorFilterStructure	44
Tabelle 22: Beschreibung der Struktur LineDirectionStructure	45
Tabelle 23: Beschreibung der Struktur LineDirectionFilterStructure	45
Tabelle 24: Beschreibung der Struktur OperatingDaysStructure	45
Tabelle 25: Beschreibung der Struktur WeekdayTimePeriodStructure	45
Tabelle 26: Beschreibung der Struktur GeneralAttributeStructure	46
Tabelle 27: Liste der einfachen Typdefinitionen in Trias_LocationSupport.xsd.	46
Tabelle 28: Beschreibung der Struktur StopPointStructure	47
Tabelle 29: Beschreibung der Struktur StopPlaceStructure	47
Tabelle 30: Beschreibung der Struktur LocalityStructure	48
Tabelle 31: Beschreibung der Struktur PointOfInterestStructure	48
Tabelle 32: Beschreibung der Struktur AddressStructure	48
Tabelle 33: Beschreibung der Struktur LocationStructure	49
Tabelle 34: Beschreibung der Struktur LocationRefStructure	49
Tabelle 35: Beschreibung der Struktur ServiceViaPointStructure	50
Tabelle 36: Beschreibung der Struktur ViaStructure	50
Tabelle 37: Beschreibung der Struktur DatedJourneyStructure	51
Tabelle 38: Beschreibung der Struktur TripLocationStructure	51
Tabelle 39: Beschreibung der Struktur CallAtStopStructure	52
Tabelle 40: Beschreibung der Struktur DatedCallAtLocationStructure	53
Tabelle 41: Beschreibung der Struktur ContinuousServiceStructure	53
Tabelle 42: Beschreibung der Struktur VehiclePositionStructure	54
Tabelle 43: Beschreibung der Struktur ProgressBetweenStopsStructure	54
Tabelle 44: Beschreibung der Struktur LocationContextStructure	54
Tabelle 45: Beschreibung der Struktur AbstractResponseContextStructure	55
Tabelle 46: Beschreibung der Struktur LegAttributeStructure	55
Tabelle 47: Beschreibung der Gruppe siri:CommonFacilityGroup	57
Tabelle 48: Beschreibung der Gruppe siri:StopFacilityGroup	58
Tabelle 49: Beschreibung der Gruppe siri:ServiceFacilityGroup	58
Tabelle 50: Beschreibung der Gruppe siri:AllFacilitiesGroup	59

Tabelle 51: Beschreibung der Struktur SituationsStructure	59
Tabelle 52: Beschreibung der Struktur SituationFullRefStructure	59
Tabelle 53: Liste der einfachen Typdefinitionen in Trias_RequestSupport.xsd.....	60
Tabelle 54: Beschreibung der Struktur AbstractTriasServiceRequestStructure	60
Tabelle 55: Beschreibung der Struktur AbstractTriasSubscriptionRequestStructure	61
Tabelle 56: Beschreibung der Struktur AbstractTriasResponseStructure	62
Tabelle 57: Liste der einfachen Typdefinitionen in Trias_FaresSupport.xsd.....	62
Tabelle 58: Beschreibung der Struktur FareZoneListInAreaStructure	63
Tabelle 59: Beschreibung der Struktur BookingInfoStructure	63
Tabelle 60: Beschreibung der Struktur TicketStructure	63
Tabelle 61: Beschreibung der Struktur TripFaresResultStructure	64
Tabelle 62: Liste der Fehlerzustände in <i>TripFaresResult</i>	64
Tabelle 63: Beschreibung der Struktur FaresPassengerStructure	64
Tabelle 64: Beschreibung der Struktur FaresParamStructure	65
Tabelle 65: Liste der einfachen Typdefinitionen in Trias_Locations.xsd.....	65
Tabelle 66: Beschreibung der Struktur LocationInformationRequestStructure	66
Tabelle 67: Beschreibung der Struktur InitialLocationInputStructure	66
Tabelle 68: Beschreibung der Struktur GeoRestrictionsStructure	66
Tabelle 69: Beschreibung der Struktur GeoCircleStructure	67
Tabelle 70: Beschreibung der Struktur GeoRectangleStructure	67
Tabelle 71: Beschreibung der Struktur GeoAreaStructure	67
Tabelle 72: Beschreibung der Struktur LocationParamStructure	68
Tabelle 73: Beschreibung der Struktur LocationInformationResponseStructure ...	68
Tabelle 74: Liste der Fehlerzustände in <i>LocationInformationResponse</i>	68
Tabelle 75: Beschreibung der Struktur LocationResultStructure	69
Tabelle 76: Beschreibung der Struktur TripRequestStructure	69
Tabelle 77: Beschreibung der Struktur TripParamStructure	71
Tabelle 78: Beschreibung der Gruppe NumberOfResultsGroup	71
Tabelle 79: Beschreibung der Struktur NotViaStructure	71
Tabelle 80: Beschreibung der Struktur NoChangeAtStructure	71
Tabelle 81: Beschreibung der Struktur TripResponseStructure	72
Tabelle 82: Liste der Fehlerzustände in <i>TripResponse</i>	72
Tabelle 83: Beschreibung der Struktur TripResponseContextStructure	72
Tabelle 84: Beschreibung der Struktur TripResultStructure	73
Tabelle 85: Liste der Fehlerzustände in <i>TripResult</i>	73

Tabelle 86: Beschreibung der Struktur TripStructure	73
Tabelle 87: Beschreibung der Struktur TripLegStructure	74
Tabelle 88: Beschreibung der Struktur TimedLegStructure	74
Tabelle 89: Beschreibung der Struktur InterchangeLegStructure	75
Tabelle 90: Beschreibung der Struktur ContinuousLegStructure	75
Tabelle 91: Beschreibung der Struktur LegBoardStructure	76
Tabelle 92: Beschreibung der Struktur LegAlightStructure	77
Tabelle 93: Beschreibung der Struktur LegIntermediateStructure	78
Tabelle 94: Beschreibung der Struktur LegTrackStructure	78
Tabelle 95: Beschreibung der Struktur TrackSectionStructure	78
Tabelle 96: Beschreibung der Struktur NavigationPathStructure	78
Tabelle 97: Beschreibung der Struktur NavigationSectionStructure	79
Tabelle 98: Beschreibung der Struktur AccessPathStructure	79
Tabelle 99: Beschreibung der Struktur StopEventRequestStructure	80
Tabelle 100: Beschreibung der Struktur StopEventParamStructure	80
Tabelle 101: Beschreibung der Struktur StopEventResponseStructure	81
Tabelle 102: Liste der Fehlerzustände in StopEventResponse	81
Tabelle 103: Beschreibung der Struktur StopEventResponseContextStructure	81
Tabelle 104: Beschreibung der Struktur StopEventResultStructure	81
Tabelle 105: Liste der Fehlerzustände in StopEventResult	82
Tabelle 106: Beschreibung der Struktur StopEventStructure	82
Tabelle 107: Beschreibung der Struktur CallAtNearStopStructure	82
Tabelle 108: Beschreibung der Struktur PositioningRequestStructure	83
Tabelle 109: Beschreibung der Struktur PositioningParamStructure	83
Tabelle 110: Beschreibung der Struktur TimedPositionStructure	83
Tabelle 111: Beschreibung der Struktur TimedStopStructure	84
Tabelle 112: Beschreibung der Struktur PositioningResponseStructure	84
Tabelle 113: Liste der Fehlerzustände in PositioningResponse	84
Tabelle 114: Beschreibung der Struktur PositioningResultStructure	85
Tabelle 115: Beschreibung der Struktur PositioningStructure	85
Tabelle 116: Beschreibung der Struktur RankedPositionStructure	85
Tabelle 117: Beschreibung der Struktur TripInfoRequestStructure	86
Tabelle 118: Beschreibung der Struktur TripInfoParamStructure	86
Tabelle 119: Beschreibung der Struktur TripInfoResponseStructure	86
Tabelle 120: Liste der Fehlerzustände in TripInfoResponse	86

Tabelle 121: Beschreibung der Struktur <i>TripInfoResponseContextStructure</i>	87
Tabelle 122: Beschreibung der Struktur <i>TripInfoResultStructure</i>	87
Tabelle 123: Beschreibung der Struktur <i>DatedConnectionStructure</i>	91
Tabelle 124: Beschreibung der Struktur <i>FeederDistributorStructure</i>	91
Tabelle 125: Beschreibung des Typs <i>GeneralizedConnectionStructure</i>	91
Tabelle 126: Beschreibung der Struktur <i>ConnectionStatusStructure</i>	91
Tabelle 127: Beschreibung der Struktur <i>RecommendationStructure</i>	92
Tabelle 128: Beschreibung der Struktur <i>ConnectionDemandRequestStructure</i>	92
Tabelle 129: Beschreibung der Struktur <i>ConnectionDemandDeleteRequestStructure</i>	92
Tabelle 130: Beschreibung der Struktur <i>ConnectionDemandResponseStructure</i>	92
Tabelle 131: Liste der Fehlerzustände in <i>ConnectionDemandResponse</i>	93
Tabelle 132: Beschreibung der Struktur <i>ConnectionDemandDeleteResponseStructure</i>	93
Tabelle 133: Liste der Fehlerzustände in <i>ConnectionDemandDeleteResponse</i>	93
Tabelle 134: Beschreibung der Struktur <i>ConnectionStatusRequestStructure</i>	93
Tabelle 135: Beschreibung der Struktur <i>ConnectionStatusNotificationStructure</i>	94
Tabelle 136: Beschreibung der Struktur <i>ConnectionStatusResponseStructure</i>	94
Tabelle 137: Liste der Fehlerzustände in <i>ConnectionStatusResponseStructure</i>	94
Tabelle 138: Beschreibung der Struktur <i>ConnectionReportRequestStructure</i>	95
Tabelle 139: Beschreibung der Struktur <i>ConnectionReportResponseStructure</i>	95
Tabelle 140: Liste der Fehlerzustände in <i>ConnectionReportResponse</i>	95
Tabelle 141: Beschreibung der Struktur <i>FaresRequestStructure</i>	96
Tabelle 142: Beschreibung der Struktur <i>StopFaresRequestStructure</i>	96
Tabelle 143: Beschreibung der Struktur <i>StaticFaresRequestStructure</i>	96
Tabelle 144: Beschreibung der Struktur <i>TripFaresRequestStructure</i>	96
Tabelle 145: Beschreibung der Struktur <i>MultiTripFaresRequestStructure</i>	97
Tabelle 146: Beschreibung der Struktur <i>FaresResponseStructure</i>	97
Tabelle 147: Liste der Fehlerzustände in <i>FaresResponse</i>	97
Tabelle 148: Beschreibung der Struktur <i>FaresResultStructure</i>	97
Tabelle 149: Beschreibung der Struktur <i>StopFaresResultStructure</i>	97
Tabelle 150: Beschreibung der Struktur <i>StaticFaresResultStructure</i>	98
Tabelle 151: Beschreibung der Struktur <i>TripTicketReferenceStructure</i>	98
Tabelle 152: Beschreibung der Struktur <i>MultiTripFaresResultStructure</i>	98
Tabelle 153: Liste der Fehlerzustände in <i>MultiTripFaresResultStructure</i>	99

Tabelle 154: Beschreibung der Struktur BookingInfoRequestStructure	99
Tabelle 155: Beschreibung der Struktur BookingInfoResponseStructure	99
Tabelle 156: Liste der Fehlerzustände in BookingInfoResponse	100
Tabelle 157: Beschreibung der Struktur BookingInfoResultStructure	100
Tabelle 158: Liste der einfachen Typdefinitionen in Trias_IndividualTrips.xsd.	100
Tabelle 159: Beschreibung der Struktur IndividualRouteRequestStructure	101
Tabelle 160: Beschreibung der Struktur IndividualTripParamStructure	102
Tabelle 161: Beschreibung der Struktur IndividualRouteContextStructure	102
Tabelle 162: Beschreibung der Struktur IndividualRouteResponseStructure	103
Tabelle 163: Liste der Fehlerzustände in IndividualRouteResponse	103
Tabelle 164: Beschreibung der Struktur RouteResultStructure	103
Tabelle 165: Liste der Fehlerzustände in RouteResult	104
Tabelle 166: Beschreibung der Struktur RouteStructure	104
Tabelle 167: Liste der einfachen Typdefinitionen in Trias_Maps.xsd.....	104
Tabelle 168: Beschreibung der Struktur MapServiceRequestStructure	105
Tabelle 169: Beschreibung der Struktur ImageCoordinatesRequestStructure	105
Tabelle 170: Beschreibung der Struktur GeoCoordinatesRequestStructure	106
Tabelle 171: Beschreibung der Struktur MapAspectStructure	106
Tabelle 172: Beschreibung der Struktur MapSizeStructure	106
Tabelle 173: Beschreibung der Struktur MapCoordinateStructure	106
Tabelle 174: Beschreibung der Struktur MapServiceResponseStructure	107
Tabelle 175: Liste der Fehlerzustände in MapServiceResponse	107
Tabelle 176: Beschreibung der Struktur ImageCoordinatesResponseStructure	107
Tabelle 177: Liste der Fehlerzustände in ImageCoordinatesResponse	107
Tabelle 178: Beschreibung der Struktur GeoCoordinatesResponseStructure	108
Tabelle 179: Liste der Fehlerzustände in GeoCoordinatesResponse	108
Tabelle 180: Beschreibung der Struktur MapResultStructure	108
Tabelle 181: Beschreibung der Struktur ImagePointResultStructure	109
Tabelle 182: Liste der Fehlerzustände in ImagePointResultStructure	109
Tabelle 183: Beschreibung der Struktur GeoCoordinateResultStructure	109
Tabelle 184: Liste der Fehlerzustände in GeoCoordinateResultStructure	109
Tabelle 185: Beschreibung der Struktur FacilityStructure	110
Tabelle 186: Beschreibung der Struktur VehicleFacilityStructure	110
Tabelle 187: Beschreibung der Struktur InfrastructureFacilityStructure	110
Tabelle 188: Beschreibung der Gruppe VehicleFacilityRefGroup	111

Tabelle 189:	Beschreibung der Gruppe	<i>InfrastructureFacilityRefGroup</i>	111
Tabelle 190:	Beschreibung der Gruppe	<i>FacilityDataFilterGroup</i>	111
Tabelle 191:	Beschreibung der Struktur	<i>FacilityStatusStructure</i>	111
Tabelle 192:	Beschreibung der Struktur	<i>FacilityStatusReportStructure</i>	112
Tabelle 193:	Beschreibung der Struktur	<i>FacilityStatusReportResponseStructure</i>	112
Tabelle 194:	Liste der Fehlerzustände in	<i>FacilityStatusReportResponse</i>	112
Tabelle 195:	Beschreibung der Struktur	<i>FacilityRequestStructure</i>	113
Tabelle 196:	Beschreibung der Struktur	<i>FacilityResponseStructure</i>	113
Tabelle 197:	Liste der Fehlerzustände in	<i>FacilityResponse</i>	113
Tabelle 198:	Beschreibung der Struktur	<i>FacilityResultStructure</i>	114
Tabelle 199:	Liste der Fehlerzustände in	<i>FacilityResultStructure</i>	114
Tabelle 200:	Beschreibung der Struktur	<i>TripMonitoringParamStructure</i>	116
Tabelle 201:	Beschreibung der Struktur	<i>TripMonitoringSubscriptionRequestStructure</i>	116
Tabelle 202:	Beschreibung der Struktur	<i>TripMonitoringDeliveryStructure</i>	117
Tabelle 203:	Liste der Fehlerzustände in	<i>TripMonitoringDeliveryStructure</i>	117
Tabelle 204:	Liste der einfachen Typdefinitionen in	Trias_Personalisation.xsd	121
Tabelle 205:	Beschreibung der Struktur	<i>PersonalisationRequestStructure</i>	122
Tabelle 206:	Beschreibung der Struktur	<i>PersonalisationSaveValueRequestStructure</i>	122
Tabelle 207:	Beschreibung der Struktur	<i>PersonalisationRetrieveValueRequestStructure</i>	122
Tabelle 208:	Beschreibung der Struktur	<i>PersonalisationDeleteValueRequestStructure</i>	122
Tabelle 209:	Beschreibung der Struktur	<i>PersonalisationEnumerateValuesRequestStructure</i>	123
Tabelle 210:	Beschreibung der Struktur	<i>PersonalisationResponseStructure</i>	123
Tabelle 211:	Beschreibung der Struktur	<i>PersonalisationSaveValueResponseStructure</i>	123
Tabelle 212:	Liste der Fehlerzustände in	<i>PersonalisationSaveValueResponseStructure</i>	123
Tabelle 213:	Beschreibung der Struktur	<i>PersonalisationRetrieveValueResponseStructure</i>	124
Tabelle 214:	Liste der Fehlerzustände in	<i>PersonalisationRetrieveValueResponseStructure</i>	124
Tabelle 215:	Beschreibung der Struktur	<i>PersonalisationDeleteValueResponseStructure</i>	124

Tabelle	216:	Liste	der	Fehlerzustände	in	
		<i>PersonalisationDeleteValueResponseStructure</i>				124
Tabelle	217:	Beschreibung	der	Struktur		
		<i>PersonalisationEnumerateValuesResponseStructure</i>				124
Tabelle	218:	Beschreibung	der	Struktur	<i>VehicleDataRequestStructure</i>	125
Tabelle	219:	Beschreibung	der	Struktur	<i>VehicleDataResponseStructure</i>	125
Tabelle	220:	Beschreibung	der	Struktur	<i>VehicleStatusStructure</i>	125
Tabelle	221:	Beschreibung	der	Struktur	<i>VehicleActivityStructure</i>	126
Tabelle	222:	Beschreibung	der	Struktur	<i>StopSequenceStructure</i>	126
Tabelle	223:	Beschreibung	der	Struktur	<i>StopInformationStructure</i>	126
Tabelle	224:	Beschreibung	der	Struktur	<i>DisplayContentStructure</i>	127
Tabelle	225:	Beschreibung	der	Struktur	<i>LineInformationStructure</i>	127
Tabelle	226:	Beschreibung	der	Struktur	<i>DestinationStructure</i>	127
Tabelle	227:	Beschreibung	der	Struktur	<i>ViaPointStructure</i>	127
Tabelle	228:	Beschreibung	der	Struktur	<i>AnnouncementStructure</i>	128
Tabelle	229:	Beschreibung	der	Struktur	<i>ConnectionStructure</i>	128
Tabelle	230:	Beschreibung	der	Struktur	<i>VehicleTypeStructure</i>	128
Tabelle	231:	Beschreibung	der	Struktur	<i>StopRequestRequestStructure</i>	129
Tabelle	232:	Beschreibung	der	Struktur	<i>StopRequestResponseStructure</i>	129
Tabelle	233:	Beschreibung	von	einfachen	Typen 129
Tabelle	234:	Beschreibung	der	Struktur	<i>ServiceRegisterRequestStructure</i>	130
Tabelle	235:	Beschreibung	der	Struktur	<i>ServiceRegisterParamStructure</i>	130
Tabelle	236:	Beschreibung	der	Struktur	<i>ServiceRegisterRegisterRequestStructure</i>	130
Tabelle	237:	Beschreibung	der	Struktur	<i>ServiceRegisterUpdateRequestStructure</i>	131
Tabelle	238:	Beschreibung	der	Struktur	<i>ServiceRegisterLookupRequestStructure</i>	131
Tabelle	239:	Beschreibung	der	Struktur	<i>ServiceRegisterUnregisterRequestStructure</i>	131
Tabelle	240:	Beschreibung	der	Struktur	<i>ServiceRegisterResponseStructure</i>	132
Tabelle	241:	Liste	der	Fehlerzustände	in	<i>ServiceRegisterResponse</i> 132
Tabelle	242:	Beschreibung	der	Struktur	<i>ServiceRegisterResultStructure</i>	132
Tabelle	243:	Beschreibung	der	Struktur	<i>ServiceRegisterRegisterResponseStructure</i>	132
Tabelle	244:	Beschreibung	der	Struktur	<i>ServiceRegisterUpdateResponseStructure</i>	132

Tabelle	245:	Beschreibung	der	Struktur	
		ServiceRegisterLookupResponseStructure			133
Tabelle	246:	Beschreibung	der	Struktur	
		ServiceRegisterUnregisterResponseStructure			133
Tabelle 247:		Beschreibung der Struktur		TriasServiceStructure	133
Tabelle 248:		Liste der Abkürzungen.....			136

30. Literaturverzeichnis

CEN, EN 12896:2006. (2006). *Reference Data Model for Public Transport*. CEN - Europäisches Komitee für Normung.

CEN, EN 28701:2012. (2012). *Intelligent transport systems - Public transport - Identification of Fixed Objects in Public Transport (IFOPT, EN 28701:2012)*. CEN - Europäisches Komitee für Normung.

CEN, TS 15531 Part 1. *SIRI - Service Interface for Realtime Information, Part 1*. CEN - Europäisches Komitee für Normung.

CEN, TS 15531 Part 5. *SIRI - Service Interface for Realtime Information, Part 5*. CEN - Europäisches Komitee für Normung.

CEN, TS 15531, Part 2. *SIRI - Service Interface for Realtime Information, Part 2*. CEN - Europäisches Komitee für Normung.

Department for Transport. (19. 04 2012). *Dft - Journeyweb*. Abgerufen am 15. 05 2013 von <http://www.dft.gov.uk/journeyweb/>

ISO 8601:2004. (2004). *ISO 8601:2004, Data elements and interchange formats - Information interchange - Representation of dates and times*. ISO, International Organization for Standardization.

VDV-Schrift 430. *Kundenschnittstelle - Architektur*. Köln: VDV - Die Verkehrsunternehmen.

VDV-Schrift 431-1. *Systemarchitektur EKAP*. Köln: VDV - Die Verkehrsunternehmen.

Verband Deutscher Verkehrsunternehmen e. V. (VDV)
Kamekestraße 37-39 · 50672 Köln
T 0221 57979-0 · F 0221 57979-8000
info@vdv.de · www.vdv.de
