

# Echtzeit Kommunikations- und Aus- kunftsplattform EKAP

---

## Teil 2: EKAP-Schnittstellenbeschreibung V1.2

---

### **Gesamtbearbeitung**

Ausschuss für Kundenservice, -information und -dialog (K3),  
Ausschuss für Telematik und Informationssysteme (ATI)

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.



---

# Echtzeit Kommunikations- und Auskunftsplattform EKAP

---

## Teil 2: EKAP-Schnittstellenbeschreibung V1.2

---

### Gesamtbearbeitung

Kernteam Information KTI des K3, Unterausschuss „intermodal transport control system“ (UA itcs)

### Autorenverzeichnis

M. Sc. Frank Englert, TU Darmstadt, Darmstadt (bis V1.0)  
Dipl.-Inf. Günther Gruber, MENTZ GmbH, München (bis V1.0)  
B. Sc. Stephan Großberndt, side by site GmbH & Co. KG, Köln (ab V1.1)  
Dipl.-Ing. Stephan Hörold, TU Ilmenau, Ilmenau  
Dipl.-Inf. Christine Keller, TU Dresden, Dresden (bis V1.0)  
Dipl.-Medieninf. Romina Kühn, TU Dresden, Dresden (bis V1.0)  
Dipl.-Math. Werner Kohl, MENTZ GmbH, München  
Dipl.-Medienwiss. Cindy Mayas, TU Ilmenau, Ilmenau (bis V1.0)  
Dipl.-Ing. ETH Walter Meier-Leu, Weisskopf Engineering AG, Schaffhausen  
Dipl.-Ing. Berthold Radermacher, VDV, Köln  
Dipl.-Inform. Anselmo Stelzer, TU Darmstadt, Darmstadt (bis V1.0)  
Dipl.-Inf. Katja Tietze, TU Dresden, Dresden (bis V1.0)  
Dipl.-Ing. (FH) Andreas Wehrmann, VDV, Köln (bis V1.0)  
Dipl.-Ing. Dirk Weißer, Init, Karlsruhe  
Dipl.-Math. Peter von Grumbkow, HaCon Ing.-Ges. mbH, Hannover  
Dipl.-Ing. (FH) Waldemar Isajkin, Init, Karlsruhe (ab V1.2)

Der Anwender ist für die sorgfältige und ordnungsgemäße Anwendung der Schrift verantwortlich. Stellt der Anwender Gefährdungen oder Unregelmäßigkeiten im Zusammenhang mit der Anwendung dieser Schrift fest, wird eine unmittelbare Benachrichtigung an den VDV erbeten. Eine Haftung des VDV oder der Mitwirkenden an der Schrift ist, soweit gesetzlich zulässig, ausgeschlossen.

© Verband Deutscher Verkehrsunternehmen e. V. Köln 2017 | Alle Rechte, einschließlich des Nachdrucks von Auszügen, der fotomechanischen oder datenverarbeitungstechnischen Wiedergabe und der Übersetzung, vorbehalten.

## Inhaltsverzeichnis

<b>Vorwort zur Version 1.0</b> .....	<b>16</b>
<b>1. Einleitung</b> .....	<b>18</b>
<b>2. Anwendungsbereich</b> .....	<b>18</b>
<b>3. Notation der XML-Elemente und -Strukturen</b> .....	<b>18</b>
<b>3.1. Darstellung von XML-Elementen im Text</b> .....	<b>19</b>
<b>3.2. Darstellung von Beziehungen</b> .....	<b>19</b>
<b>3.2.1. Gruppierung</b> .....	<b>20</b>
<b>3.2.2. Elementname</b> .....	<b>21</b>
<b>3.2.3. Multiplizität &amp; Choice (Min:Max)</b> .....	<b>21</b>
<b>3.2.4. Datentyp</b> .....	<b>21</b>
<b>3.2.5. Erläuterung</b> .....	<b>21</b>
<b>4. Nachrichtenübermittlung</b> .....	<b>22</b>
<b>4.1. Einsatz der SIRI-Verfahren</b> .....	<b>22</b>
<b>4.2. HTTP und REST</b> .....	<b>23</b>
<b>4.3. Nachrichten-Kodierung</b> .....	<b>24</b>
<b>4.4. Rollen von Server und Client</b> .....	<b>24</b>
<b>5. Identifikation von Objekten über Systemgrenzen hinweg</b> .....	<b>25</b>
<b>5.1. Haltestellen und Haltepunkte</b> .....	<b>25</b>
<b>5.2. Orte und Gemeinden</b> .....	<b>26</b>
<b>5.3. Adressen und POIs</b> .....	<b>26</b>
<b>5.4. Organisationen</b> .....	<b>27</b>
<b>5.5. Linien und Linienrichtungen</b> .....	<b>27</b>
<b>5.6. Fahrten</b> .....	<b>28</b>
<b>5.7. Fahrzeuge</b> .....	<b>29</b>
<b>5.8. Fahrzeugtypen</b> .....	<b>29</b>
<b>5.9. Verkehrstage</b> .....	<b>29</b>
<b>5.10. Eigentümer</b> .....	<b>29</b>
<b>5.11. Haltestellen- und Fahrzeugeinrichtungen</b> .....	<b>30</b>
<b>5.12. Teilnehmende Systeme / IT-Systeme</b> .....	<b>30</b>
<b>5.13. Ereignismeldungen</b> .....	<b>30</b>
<b>5.14. Tarifverantwortliche</b> .....	<b>30</b>
<b>5.15. Tarifzonen</b> .....	<b>30</b>
<b>5.16. Fahrscheine und Vielfahrerkarten</b> .....	<b>31</b>

<b>6. Dienste, XML-Schemata und Konventionen .....</b>	<b>32</b>
6.1. Bereitgestellte Dienste.....	32
6.2. Dienstübergreifend genutzte XML-Schemata.....	34
6.3. Importierte Siri-Schemata.....	34
6.4. Fehlerzustände beim Betrieb von TRIAS-Diensten.....	34
6.4.1. Fehlercodes aus SIRI.....	35
6.4.2. Allgemeine TRIAS-Fehlerzustände .....	36
6.5. Haltsequenznummern und Fahrtabschnitte in TRIAS-Diensten .....	36
<b>7. Gemeinsam genutzte XML-Strukturen .....</b>	<b>37</b>
7.1. Trias, das Wurzelement .....	37
7.1.1. ServiceRequestStructure .....	37
7.1.2. SubscriptionRequestStructure .....	38
7.1.3. RequestPayloadStructure .....	39
7.1.4. ServiceDeliveryStructure.....	40
7.1.5. DeliveryPayloadStructure.....	40
7.2. Trias_Utility.....	41
7.2.1. Einfache Typen .....	41
7.2.2. InternationalTextStructure .....	42
7.2.3. GeoPositionStructure .....	42
7.2.4. WebLinkStructure .....	42
7.3. Trias_ModesSupport.....	42
7.3.1. Einfache Typen .....	43
7.3.2. IndividualTransportOptionsStructure.....	44
7.3.3. PtSubmodeChoiceGroup.....	45
7.3.4. ModeStructure.....	45
7.3.5. PtModeFilterStructure .....	46
7.4. Trias_Common .....	46
7.4.1. Einfache Typen .....	46
7.4.2. ErrorMessageStructure .....	46
7.4.3. PrivateCodeStructure .....	47
7.4.4. OperatorFilterStructure .....	47
7.4.5. LineDirectionStructure .....	47
7.4.6. LineDirectionFilterStructure.....	47
7.4.7. SharingServiceStructure .....	48
7.4.8. OperatingDaysStructure.....	48

7.4.9. WeekdayTimePeriodStructure .....	48
7.4.10. GeneralAttributeStructure .....	49
7.5. Trias_LocationSupport .....	49
7.5.1. Einfache Typen .....	49
7.5.2. StopPointStructure .....	50
7.5.3. StopPlaceStructure .....	50
7.5.4. LocalityStructure .....	51
7.5.5. PointOfInterestStructure .....	51
7.5.6. PointOfInterestCategoryStructure .....	51
7.5.7. OsmTagStructure .....	52
7.5.8. PointOfInterestFilterStructure .....	52
7.5.9. AddressStructure .....	52
7.5.10. LocationStructure .....	53
7.5.11. LocationRefStructure .....	53
7.6. Trias_JourneySupport .....	54
7.6.1. ServiceViaPointStructure .....	54
7.6.2. ViaStructure .....	54
7.6.3. ServiceSectionStructure .....	55
7.6.4. DatedServiceGroup .....	55
7.6.5. DatedJourneyStructure .....	55
7.6.6. ParallelServiceStructure .....	56
7.6.7. TripLocationStructure .....	56
7.6.8. ServiceCallStructure .....	56
7.6.9. CallAtStopStructure .....	57
7.6.10. DatedCallAtLocationStructure .....	58
7.6.11. ContinuousServiceStructure .....	59
7.6.12. VehiclePositionStructure .....	59
7.6.13. ProgressBetweenStopsStructure .....	59
7.6.14. LegTrackStructure .....	60
7.6.15. TrackSectionStructure .....	60
7.6.16. LocationContextStructure .....	60
7.6.17. AbstractResponseContextStructure .....	61
7.6.18. ServiceAttributeStructure .....	61
7.6.19. PassengerAccessibilityStructure .....	61
7.7. Trias_FacilitySupport .....	62

7.7.1. siri:CommonFacilityGroup .....	62
7.7.2. siri:StopFacilityGroup .....	64
7.7.3. siri:ServiceFacilityGroup .....	65
7.7.4. siri:AllFacilitiesGroup .....	66
7.8. Trias_SituationSupport .....	66
7.8.1. SituationsStructure .....	66
7.8.2. SituationFullRefStructure .....	67
7.9. Trias_RequestSupport .....	68
7.9.1. Einfache Typen .....	68
7.9.2. AbstractTriasServiceRequestStructure .....	68
7.9.3. AbstractTriasSubscriptionRequestStructure .....	69
7.9.4. AbstractTriasResponseStructure .....	70
7.10. Trias_FaresSupport .....	71
7.10.1. Einfache Typen .....	71
7.10.2. FareZoneStructure .....	71
7.10.3. FareZoneListInAreaStructure .....	71
7.10.4. BookingInfoStructure .....	72
7.10.5. TicketStructure .....	72
7.10.6. TripFaresResultStructure .....	73
7.10.7. FaresPassengerStructure .....	74
7.10.8. FaresParamStructure .....	74
8. Ortsinformationsdienst .....	75
8.1. Beschreibung .....	75
8.2. Einfache Datentypen .....	75
8.3. Anfragestrukturen .....	76
8.3.1. LocationInformationRequestStructure .....	76
8.3.2. InitialLocationInputStructure .....	76
8.3.3. GeoRestrictionsStructure .....	76
8.3.4. GeoCircleStructure .....	77
8.3.5. GeoRectangleStructure .....	77
8.3.6. GeoAreaStructure .....	77
8.3.7. LocationParamStructure .....	77
8.4. Antwortstrukturen .....	79
8.4.1. LocationInformationResponseStructure .....	79
8.4.2. LocationResultStructure .....	80

<b>9. Dienst Verbindungsauskunft .....</b>	<b>80</b>
<b>9.1. Beschreibung .....</b>	<b>80</b>
<b>9.2. Anfragestrukturen .....</b>	<b>80</b>
<b>9.2.1. TripRequestStructure .....</b>	<b>80</b>
<b>9.2.2. TripParamStructure .....</b>	<b>81</b>
<b>9.2.3. NumberOfResultsGroup .....</b>	<b>82</b>
<b>9.2.4. NotViaStructure.....</b>	<b>82</b>
<b>9.2.5. NoChangeAtStructure .....</b>	<b>83</b>
<b>9.3. Antwortstrukturen .....</b>	<b>84</b>
<b>9.3.1. TripResponseStructure .....</b>	<b>84</b>
<b>9.3.2. TripResponseContextStructure .....</b>	<b>85</b>
<b>9.3.3. TripResultStructure .....</b>	<b>85</b>
<b>9.3.4. TripStructure .....</b>	<b>86</b>
<b>9.3.5. TripLegStructure.....</b>	<b>86</b>
<b>9.3.6. TimedLegStructure .....</b>	<b>87</b>
<b>9.3.7. InterchangeLegStructure.....</b>	<b>88</b>
<b>9.3.8. ContinuousLegStructure.....</b>	<b>89</b>
<b>9.3.9. LegBoardStructure .....</b>	<b>90</b>
<b>9.3.10. LegAlightStructure.....</b>	<b>91</b>
<b>9.3.11. LegIntermediateStructure.....</b>	<b>92</b>
<b>9.3.12. NavigationPathStructure .....</b>	<b>92</b>
<b>9.3.13. NavigationSectionStructure .....</b>	<b>93</b>
<b>9.3.14. AccessPathStructure .....</b>	<b>93</b>
<b>10. Dienst Abfahrtstafeln.....</b>	<b>94</b>
<b>10.1. Beschreibung .....</b>	<b>94</b>
<b>10.2. Anfragestrukturen .....</b>	<b>94</b>
<b>10.2.1. StopEventRequestStructure.....</b>	<b>94</b>
<b>10.2.2. StopEventParamStructure.....</b>	<b>94</b>
<b>10.3. Antwortstrukturen .....</b>	<b>95</b>
<b>10.3.1. StopEventResponseStructure.....</b>	<b>95</b>
<b>10.3.2. StopEventResponseContextStructure .....</b>	<b>95</b>
<b>10.3.3. StopEventResultStructure.....</b>	<b>95</b>
<b>10.3.4. StopEventStructure .....</b>	<b>96</b>
<b>10.3.5. CallAtNearStopStructure .....</b>	<b>96</b>



<b>11. Dienst Logische Ortung .....</b>	<b>97</b>
<b>11.1. Beschreibung .....</b>	<b>97</b>
<b>11.2. Anfragestrukturen .....</b>	<b>97</b>
<b>11.2.1. PositioningRequestStructure.....</b>	<b>97</b>
<b>11.2.2. PositioningParamStructure.....</b>	<b>97</b>
<b>11.2.3. TimedPositionStructure.....</b>	<b>98</b>
<b>11.2.4. TimedStopStructure.....</b>	<b>98</b>
<b>11.3. Antwortstrukturen .....</b>	<b>98</b>
<b>11.3.1. PositioningResponseStructure.....</b>	<b>98</b>
<b>11.3.2. PositioningResultStructure.....</b>	<b>99</b>
<b>11.3.3. PositioningStructure.....</b>	<b>99</b>
<b>11.3.4. RankedPositionStructure .....</b>	<b>99</b>
<b>12. Dienst Fahrtinformation (EKAP) .....</b>	<b>100</b>
<b>12.1. Beschreibung .....</b>	<b>100</b>
<b>12.2. Anfragestrukturen .....</b>	<b>100</b>
<b>12.2.1. TripInfoRequestStructure.....</b>	<b>100</b>
<b>12.2.2. TripInfoParamStructure.....</b>	<b>101</b>
<b>12.3. Antwortstrukturen .....</b>	<b>101</b>
<b>12.3.1. TripInfoResponseStructure.....</b>	<b>101</b>
<b>12.3.2. TripInfoResponseContextStructure.....</b>	<b>102</b>
<b>12.3.3. TripInfoResultStructure.....</b>	<b>102</b>
<b>13. Anschlussdienste .....</b>	<b>103</b>
<b>13.1. Beschreibung .....</b>	<b>103</b>
<b>13.1.1. Dienst Anschlussvoranmeldung.....</b>	<b>104</b>
<b>13.1.2. Dienst Anschlussstatus.....</b>	<b>104</b>
<b>13.1.3. Dienst Info bei Anschlussverlust.....</b>	<b>105</b>
<b>13.1.4. Dienst Anschlussrückmeldung.....</b>	<b>105</b>
<b>13.2. Einfache Typen.....</b>	<b>105</b>
<b>13.3. Komplexe Strukturen .....</b>	<b>106</b>
<b>13.3.1. DatedConnectionStructure.....</b>	<b>106</b>
<b>13.3.2. FeederDistributorStructure .....</b>	<b>106</b>
<b>13.3.3. GeneralizedConnectionStructure .....</b>	<b>106</b>
<b>13.3.4. ConnectionStatusStructure.....</b>	<b>107</b>
<b>13.3.5. RecommendationStructure .....</b>	<b>107</b>
<b>13.4. Anfragestrukturen Anschlussmeldung.....</b>	<b>108</b>

13.4.1. ConnectionDemandRequestStructure.....	108
13.4.2. ConnectionDemandDeleteRequestStructure .....	108
13.5. Antwortstrukturen Anschlussmeldung.....	108
13.5.1. ConnectionDemandResponseStructure.....	108
13.5.2. ConnectionDemandDeleteResponseStructure .....	109
13.6. Anfragestrukturen Anschlussstatus.....	109
13.6.1. ConnectionStatusRequestStructure.....	109
13.6.2. ConnectionStatusNotificationStructure .....	110
13.7. Antwortstrukturen Anschlussstatus.....	110
13.7.1. ConnectionStatusResponseStructure.....	110
13.8. Anfragestrukturen Anschlussrückmeldung .....	111
13.8.1. ConnectionReportRequestStructure .....	111
13.9. Antwortstrukturen Anschlussrückmeldung .....	111
13.9.1. ConnectionReportResponseStructure .....	111
14. Dienst Fahrpreis- und Tarifberechnung .....	112
14.1. Beschreibung .....	112
14.2. Anfragestrukturen .....	112
14.2.1. FaresRequestStructure.....	112
14.2.2. StopFaresRequestStructure.....	112
14.2.3. StaticFaresRequestStructure .....	113
14.2.4. TripFaresRequestStructure .....	113
14.2.5. MultiTripFaresRequestStructure.....	113
14.3. Antwortstrukturen .....	114
14.3.1. FaresResponseStructure.....	114
14.3.2. FaresResultStructure.....	114
14.3.3. StopFaresResultStructure.....	114
14.3.4. StaticFaresResultStructure .....	115
14.3.5. TripTicketReferenceStructure .....	115
14.3.6. MultiTripFaresResultStructure.....	115
15. Dienst Anreicherung.....	116
15.1. Beschreibung .....	116
15.2. Anfragestrukturen .....	117
15.2.1. RefineRequestStructure (in Trias_Refine.xsd) .....	117
15.2.2. IndividualRouteRefineRequestStructure (in Trias_IndividualTrips.xsd) .....	118

15.2.3. LocationInformationRefineRequestStructure (in Trias_Locations.xsd) .....	118
15.2.4. StopEventRefineRequestStructure (in Trias_StopEvents.xsd) .....	118
15.2.5. TripInfoRefineRequestStructure (in Trias_TripInfo.xsd) .....	118
15.2.6. TripRefineRequestStructure (in Trias_Trips.xsd) .....	118
15.2.7. IndividualTripRefineParamStructure (in Trias_IndividualTrips.xsd)	119
15.2.8. LocationInformationRefineParamStructure (in Trias_Locations.xsd) .....	119
15.2.9. StopEventRefineParamStructure (in Trias_StopEvents.xsd).....	119
15.2.10. TripInfoRefineParamStructure (in Trias_TripInfo.xsd) .....	120
15.2.11. TripRefineParamStructure (in Trias_Trips.xsd) .....	120
15.3. Antwortstrukturen .....	121
15.3.1. RefineResponseStructure (in Trias_Refine.xsd).....	121
15.3.2. IndividualRouteRefineResponseStructure (in Trias_IndividualTrips.xsd) .....	121
15.3.3. LocationInformationRefineResponseStructure (in Trias_Locations.xsd) .....	122
15.3.4. StopEventRefineResponseStructure (in Trias_StopEvents.xsd)....	122
15.3.5. TripInfoRefineResponseStructure (in Trias_TripInfo.xsd) .....	122
15.3.6. TripRefineResponseStructure (in Trias_Trips.xsd) .....	122
16. Dienst Buchungsinformation .....	123
16.1. Beschreibung .....	123
16.2. Anfragestrukturen .....	123
16.2.1. BookingInfoRequestStructure .....	123
16.3. Antwortstrukturen .....	123
16.3.1. BookingInfoResponseStructure .....	123
16.3.2. BookingInfoResultStructure .....	124
17. Dienst IV-Routing.....	125
17.1. Einfache Typen.....	125
17.2. Anfragestrukturen .....	125
17.2.1. IndividualRouteRequestStructure.....	125
17.2.2. IndividualTripParamStructure .....	125
17.2.3. IndividualRouteLocationContextStructure.....	127
17.3. Antwortstrukturen .....	127
17.3.1. IndividualRouteResponseStructure.....	127
17.3.2. RouteResultStructure .....	128

17.3.3. RouteStructure.....	129
<b>18. Kartendienst.....</b>	<b>130</b>
18.1. Einfache Datentypen .....	130
18.2. Anfragestrukturen .....	130
18.2.1. MapServiceRequestStructure .....	130
18.2.2. ImageCoordinatesRequestStructure .....	131
18.2.3. GeoCoordinatesRequestStructure.....	131
18.2.4. MapAspectStructure .....	131
18.2.5. MapSizeStructure.....	132
18.2.6. MapCoordinateStructure .....	132
18.3. Antwortstrukturen .....	132
18.3.1. MapServiceResponseStructure .....	132
18.3.2. ImageCoordinatesResponseStructure .....	133
18.3.3. GeoCoordinatesResponseStructure.....	133
18.3.4. MapResultStructure .....	134
18.3.5. ImagePointResultStructure .....	134
18.3.6. GeoCoordinateResultStructure.....	135
<b>19. Dienst Schadensmeldung / Zustand von Einrichtungen.....</b>	<b>136</b>
19.1. Beschreibung .....	136
19.2. Einfache Typen.....	136
19.3. Komplexe Strukturen .....	136
19.3.1. FacilityStructure.....	136
19.3.2. VehicleFacilityStructure .....	136
19.3.3. InfrastructureFacilityStructure.....	137
19.3.4. VehicleFacilityRefGroup.....	137
19.3.5. InfrastructureFacilityRefGroup .....	137
19.3.6. FacilityDataFilterGroup.....	138
19.3.7. FacilityStatusStructure .....	138
19.4. Anfrage Schadensmeldung .....	138
19.4.1. FacilityStatusReportStructure.....	138
19.5. Antwort Schadensmeldung .....	139
19.5.1. FacilityStatusReportResponseStructure.....	139
19.6. Anfrage Zustand von Einrichtungen.....	139
19.6.1. FacilityRequestStructure.....	139
19.7. Antwort Zustand von Einrichtungen.....	140

19.7.1. FacilityResponseStructure .....	140
19.7.2. FacilityResultStructure .....	140
20. Benachrichtigungsdienst .....	141
20.1. Beschreibung .....	141
20.2. Komplexe Strukturen .....	142
20.2.1. TripMonitoringParamStructure .....	142
20.3. Anfragestrukturen .....	143
20.3.1. TripMonitoringSubscriptionRequestStructure.....	143
20.4. Antwortstrukturen .....	143
20.4.1. TripMonitoringDeliveryStructure .....	143
21. Personalisierungsdienst .....	145
21.1. Beschreibung .....	145
21.2. Interaktionen.....	145
21.2.1. Lebenszyklus eines Wertes.....	146
21.2.2. Werteliste ermitteln.....	147
21.2.3. Werte speichern und abrufen.....	147
21.3. Einfache Typen.....	149
21.4. Anfragestrukturen .....	149
21.4.1. PersonalisationRequestStructure.....	149
21.4.2. PersonalisationSaveValueRequestStructure .....	149
21.4.3. PersonalisationRetrieveValueRequestStructure .....	149
21.4.4. PersonalisationDeleteValueRequestStructure.....	150
21.4.5. PersonalisationEnumerateValuesRequestStructure .....	150
21.5. Antwortstrukturen .....	150
21.5.1. PersonalisationResponseStructure.....	150
21.5.2. PersonalisationSaveValueResponseStructure .....	150
21.5.3. PersonalisationRetrieveValueResponseStructure .....	151
21.5.4. PersonalisationDeleteValueResponseStructure.....	151
21.5.5. PersonalisationEnumerateValuesResponseStructure .....	152
22. Dienst Fahrzeuginformationen .....	152
22.1. Beschreibung .....	152
22.2. Anfragestrukturen .....	152
22.2.1. VehicleDataRequestStructure .....	152
22.3. Antwortstrukturen .....	152
22.3.1. VehicleDataResponseStructure .....	152

22.3.2. VehicleStatusStructure.....	153
22.3.3. VehicleActivityStructure.....	153
22.3.4. StopSequenceStructure .....	153
22.3.5. StopInformationStructure.....	154
22.3.6. DisplayContentStructure .....	154
22.3.7. LineInformationStructure .....	154
22.3.8. DestinationStructure.....	155
22.3.9. ViaPointStructure.....	155
22.3.10. AnnouncementStructure .....	155
22.3.11. ConnectionStructure.....	156
22.3.12. VehicleTypeStructure .....	156
23. Dienst Fahrzeugaktionen .....	157
23.1. Beschreibung .....	157
23.2. Anfragestrukturen .....	157
23.2.1. VehicleInteractionRequestStructure.....	157
23.2.2. ActivateOutsideSpeakerRequestStructure .....	157
23.2.3. StopRequestRequestStructure .....	157
23.3. Antwortstrukturen .....	158
23.3.1. VehicleInteractionResponseStructure.....	158
23.3.2. ActivateOutsideSpeakerResponseStructure .....	158
23.3.3. StopRequestResponseStructure .....	158
24. Dienst Diensteregister .....	159
24.1. Beschreibung .....	159
24.2. Einfache Typen.....	159
24.3. Anfragestrukturen .....	159
24.3.1. ServiceRegisterRequestStructure .....	159
24.3.2. ServiceRegisterParamStructure .....	160
24.3.3. ServiceRegisterRegisterRequestStructure .....	160
24.3.4. ServiceRegisterUpdateRequestStructure .....	160
24.3.5. ServiceRegisterLookupRequestStructure.....	161
24.3.6. ServiceRegisterUnregisterRequestStructure.....	161
24.4. Antwortstrukturen .....	161
24.4.1. ServiceRegisterResponseStructure .....	161
24.4.2. ServiceRegisterResultStructure .....	162
24.4.3. ServiceRegisterRegisterResponseStructure .....	162

24.4.4. ServiceRegisterUpdateResponseStructure .....	162
24.4.5. ServiceRegisterLookupResponseStructure.....	162
24.4.6. ServiceRegisterUnregisterResponseStructure.....	162
24.4.7. TriasServiceStructure .....	163
25. Dienst Authentifizierung.....	164
26. Normative Referenzen .....	166
27. Begriffe .....	166
28. Versionshistorie.....	167
28.1. Version 1.1 (Schrift und Schema).....	167
28.1.1. Funktionale Erweiterungen .....	167
28.1.2. Technische Ergänzungen/Korrekturen.....	167
28.1.3. Dokumentarische Korrekturen.....	168
28.2. Version 1.2 (Schrift und Schema) .....	168
28.2.1. Funktionale Erweiterungen .....	168
28.2.2. Technische Ergänzungen/Korrekturen.....	169
28.2.3. Dokumentarische Ergänzungen/Korrekturen .....	170
29. Abkürzungen.....	170
30. Abbildungsverzeichnis.....	171
31. Tabellenverzeichnis.....	171
32. Literaturverzeichnis.....	181
Impressum .....	182

## Vorwort zur Version 1.0

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).

Erster Schwerpunkt (grün in Abbildung 1) ist die Spezifikation eines performanten IP-basierten Kommunikationsprotokolls im Fahrzeug (IBIS-IP, VDV301). 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.

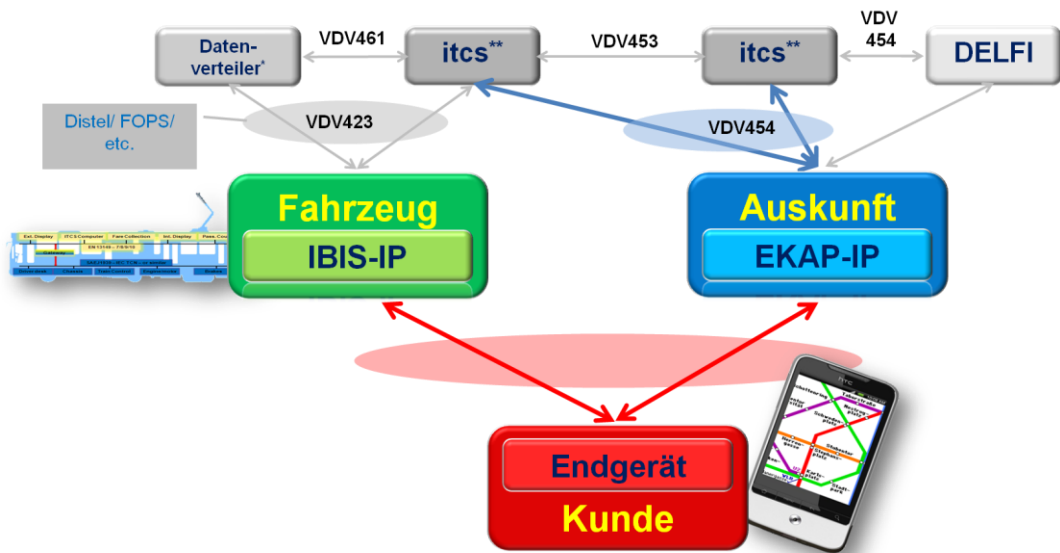


Abbildung 1: Umfeld und Schwerpunkte im Projekt IP-KOM-ÖV

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, womit 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.

## 2. Anwendungsbereich

Die in diesem Dokument definierten Schnittstellen spezifizieren Dienste, die

- zwischen mobilen Apps und Fahrzeugen des ÖV
- zwischen Portalsystemen und Auskunftssystemen (EKAPs) als Hintergrundsystem und
- EKAP und Fahrgastinformationssystemen (z.B. Anschlussanzeigen in Fahrzeugen, Abfahrtsanzeigen und Übersichtsanzeigen an Haltestellen, etc.)

Verwendung finden.

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 TRIAS<sup>1</sup>-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, 2011) 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.

---

<sup>1</sup> Travellers Realtime Information and Advisory Standard

### 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. Eine 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 „Identifier“ (manchmal auch „Number“ in SIRI), z. B. erhält eine Fahrplanfahrt (Journey) den Schlüssel ***JourneyCode***.
- Wird ein Objekt von einem anderen Objekt aus 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*.

### 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
<b>ContinuousServiceStructure</b>			+Structure	Eine Fahrgastbewegung mit Hilfe eines kontinuierlichen, nicht fahrplangebundenen Verkehrsmittels.
	a <b>ContinuousMode</b>	-1:1	walk   demandResponsive   replacementService	Modalität für kontinuierliche Verkehre
	b <b>IndividualMode</b>		walk   cycle   taxi   self-drive-car   others-drive-car   motorcycle   truck	Verkehrsmittelmodalität für Individualverkehr
Dated-Service	<b>OperatingDay</b>	1:1	→OperatingDay	Betriebstag der Fahrt.
	VehicleRef	0:1	→Vehicle	Fahrzeug-ID.
Service-Journey	<b>JourneyRef</b>	1:1	→Journey	Fahrt-ID.
Line-identity	<b>LineRef</b>	1:1	→Line	Linien-ID.
	<b>DirectionRef</b>	1:1	→Direction	Richtungs-ID.
Service	<b>Mode</b>	1:1	+Mode	Verkehrsmitteltyp.
	<b>PublishedLineName</b>	1:*	International-Text	Liniennummer oder -name, wie in der Öffentlichkeit bekannt.
	OperatorRef	0:1	→Operator	Operator-ID.
	RouteDescription	0:*	International-Text	Beschreibung des Fahrwegs.
	Via	0:*	+ServiceViaPoint	Wichtige Halte auf dem Fahrweg.
	Attribute	0:*	+GeneralAttribute	Hinweise und Attribute (mit Klassifikationen) zur Fahrt.
ServiceOrigin	OriginStopPointRef	0:1	→StopPoint	ID des ersten Haltepunkts der Fahrt; Starthaltestelle.
	OriginText	0:*	International-Text	Name des ersten Haltepunkts der Fahrt, der Starthaltestelle.
Service-Destination	DestinationStopPointRef	0:1	→StopPoint	ID des letzten Haltepunkts der Fahrt; Endhaltestelle.
	DestinationText	1:*	International-Text	Name des letzten Haltepunkts der Fahrt, der Endhaltestelle oder Fahrtziel.
	SituationFullRef	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 1: Beispiel (aus einem späteren Abschnitt) für die tabellarische Notation einer XML-Struktur

### 3.2.1. 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.

### 3.2.2. 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 (siehe beispielhaft Tabelle 5). Tabelle 5: Beschreibung der Struktur ***ServiceRequestStructure***.

### 3.2.3. 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“.

### 3.2.4. Datentyp

Die Datentypen werden in der vierten Spalte kursiv angegeben, z. B. *InternationalText*. Falls der Namensraum (namespace) vom TRIAS-Namensraum abweicht, wird er mitangegeben, 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 wiederverwendet 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.

### 3.2.5. 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, an der 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, 2011). 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, 2011)).

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.

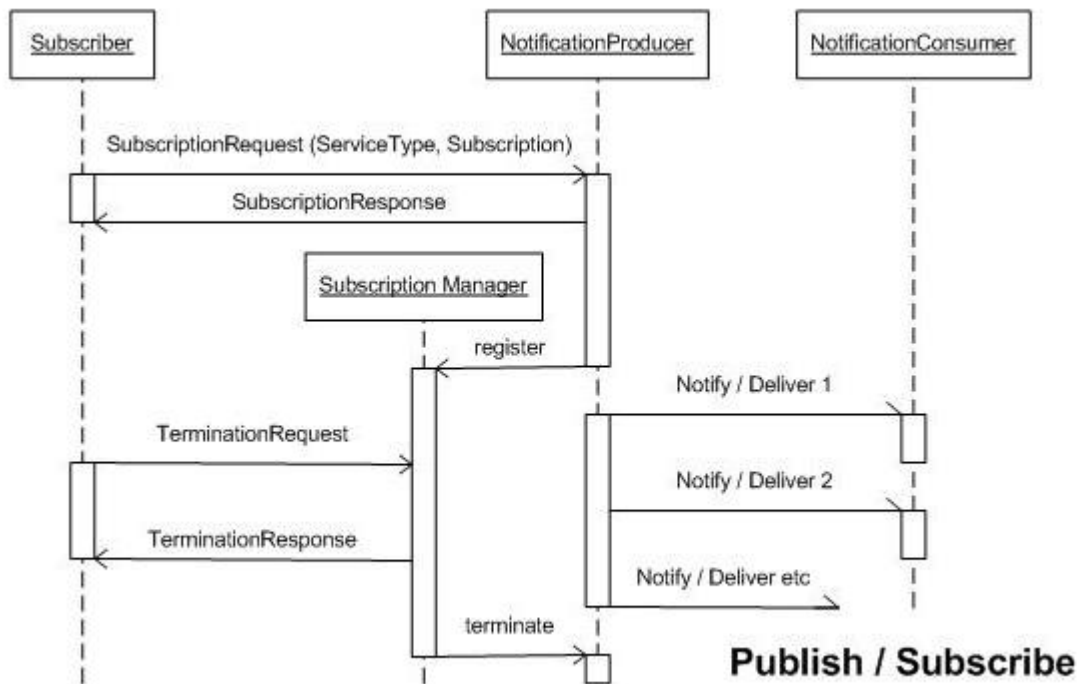


Abbildung 3: Abonnement-Verfahren mit asynchronen Benachrichtigungen (Abbildung entnommen aus SIRI, (CEN, TS 15531 Part 2, 2011)).

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, 2011), Kapitel 5.

In den TRIAS-Diensten kommt der Abo-Mechanismus beim Benachrichtigungsdienst vor, über den sich ein Datenkonsument (z. B. Portalsystem, mobile App, Fahrgastinformationssystem, etc.) 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 Protocol<sup>2</sup>) als Transportprotokoll und XML (Extensible Markup Language<sup>3</sup>) für die Nachrichteninhalte.

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

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

Zur Nachrichten-Kodierung ist grundsätzlich "UTF-8" zu verwenden. Dies gilt sowohl für die Header, das XML-Element und den Inhalt von Anfragen und Antworten. Dies gewährleistet eine problemlose Nutzung auch bei Verwendung von Sprachen mit nicht-lateinischen Zeichen. Zusätzlich müssen sich Clients, die mit mehreren TRIAS-Schnittstellen verschiedener Anbieter kommunizieren, nicht um eine unterschiedliche, Anbieter-spezifische Konvertierung der empfangenen Daten kümmern.

### **4.4. 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. Wenn 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.

---

3 <http://www.w3.org/XML/>



## 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.

Basierend auf der Norm IFOPT wurde in Deutschland die Deutschlandweite Haltestellen ID entwickelt. Das Format ist in der VDV-Schrift 432 (VDV-Schrift 432, 07/2016) beschrieben.

## 5.2. Orte und Gemeinden

Zur eindeutigen Referenzierung von Gemeinden existiert in Deutschland der Amtliche Gemeindegemeinschaftsschlüssel (AGS<sup>4</sup>), 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, 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

**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.

Die Kategorisierung von POI basiert auf dem Tagging-Schema von OpenStreetMap<sup>5</sup>. Ein POI kann mehrere Schlüssel-Wert-Paare zugeordnet bekommen (z.B. für eine Fahrrad-Ladestation amenity=charging\_station und bike=yes).

---

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

<sup>5</sup> [http://wiki.openstreetmap.org/wiki/DE:Map\\_Features](http://wiki.openstreetmap.org/wiki/DE:Map_Features)

## 5.4. Organisationen

Zur eindeutigen Referenzierung von Organisationen (z. B. Verkehrsunternehmen und Verkehrsverbänden, Aufgabenträger, etc.) wird ein Organisationscode verwendet. Damit diese Codes über mehrere Systeme hinweg eindeutig bleiben, empfiehlt sich der Aufbau einer übergreifenden Datenbank von Organisationen. Damit die Organisations-IDs über mehrere Länder hinweg eindeutig bleiben, wird dem Organisationscode ein Länderkürzel als Namensraum vorangestellt.

### Aufbau einer Organisations-ID

**Länderkürzel:Organisationscode**

### Beispiele

<b>Verkehrs- und Tarifverbund Stuttgart</b>	de:vvs
<b>Stuttgarter Straßenbahn AG</b>	de:ssb
<b>Fernverkehr Deutsche Bahn</b>	de:dbag
<b>DB Regio Baden-Württemberg</b>	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.<sup>6</sup> Damit die Linien-IDs über mehrere Datenlieferanten hinweg eindeutig bleiben, wird dem Linienschlüssel die Organisations-ID (vgl. **Fehler! Verweisquelle konnte nicht gefunden werden.**) als Namensraum vorangestellt.

### Aufbau einer Linien-ID

**Länderkürzel:Organisationscode:Linien-schlüssel**

### Beispiel

<b>Stadtbahn-Linie U1 in Stuttgart</b>	de:vvs:20001
--	--------------

Zur eindeutigen Referenzierung von Linienrichtungen wird der Richtungscode des verantwortlichen Datenlieferanten verwendet. Der Richtungscode ist vom Datenlieferanten frei

---

<sup>6</sup> 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.

wählbar und wird für den Fahrgast erst durch begleitende Texte verständlich. Die Richtungs-ID wird nur im Kontext einer Linie verwendet, so dass das Voranstellen des Linienschlüssels als Namensraum nicht notwendig ist.

#### Aufbau einer Richtungs-ID

**Richtungscode**

#### Beispiele

<b>Hin</b>	H
<b>Rück</b>	R
<b>Hin</b>	1
<b>Rück</b>	2
<b>Stadteinwärts</b>	E
<b>Stadtauswärts</b>	A

## 5.6. Fahrten

Zur eindeutigen Referenzierung von Fahrten (engl. *Vehicle journey* oder kurz: *Journey*) wird der Fahrten Schlüssel des verantwortlichen Datenlieferanten verwendet. Damit die Fahrt-IDs über mehrere Datenlieferanten hinweg eindeutig bleiben, wird dem Fahrten Schlüssel die Organisations-ID (vgl. 5.4) als Namensraum vorangestellt.

Der Fahrten Schlüssel ist vom Datenlieferanten frei wählbar, solange er im Namensraum einer Linie eindeutig ist.

#### Aufbau einer Fahrt-ID

**Länderkürzel:Organisationscode:Linienschlüssel:Fahrtenschlüssel**

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

#### Beispiele

<b>Fahrt 1512 der Linie U1 in Stuttgart</b>	de:vvs:20001:1512
<b>ICE 612 der DB AG</b>	de:dbag::612

## 5.7. Fahrzeuge

Zur eindeutigen Referenzierung von Fahrzeugen (engl. *Vehicle*) wird der Fahrzeugcode des verantwortlichen Datenlieferanten verwendet. Damit die Fahrzeug-IDs über mehrere Datenlieferanten hinweg eindeutig bleiben, wird dem Fahrzeugcode die Organisations-ID (vgl. **Fehler! Verweisquelle konnte nicht gefunden werden.**) als Namensraum vorangestellt.

In Echtzeitschnittstellen (VDV 454, SIRI ET) teilen Leitstellen zu einer Fahrplanfahrt die Fahrzeug-ID 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 Fahrzeug-ID zur Fahrt-ID eindeutig sein.

### Aufbau einer Fahrzeug-ID

**Länderkürzel:Organisationscode:Fahrzeugcode**

### Beispiel

**Fahrzeug 5812 der SSB AG**

de:ssb:5812

## 5.8. Fahrzeugtypen

Der Fahrzeugtyp (engl. *Vehicle type*) und die damit verbundenen Fahrzeugausstattungsattribute werden 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 ist in TRIAS die Referenz auf den Betriebstag-Code der Fahrplandaten.

Diese Betriebstag-Codes sollten in TRIAS nach der Norm ISO 8601 dargestellt werden.

### 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 ge-

nau dieselbe Weise wie die von Verkehrsunternehmen und -verbänden (vgl. **Fehler! Verweisquelle konnte nicht gefunden werden.**).

### 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 Tarifverbände, für Haustarife die Verkehrsunternehmen selbst. Die Referenzierung von Tarifverantwortlichen erfolgt auf genau dieselbe Weise wie die von Verkehrsunternehmen und -verbänden (vgl. **Fehler! Verweisquelle konnte nicht gefunden werden.**).

### 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). IDs von Fahrscheinen werden im Namensraum der jeweiligen Tarifverantwortlichen angegeben und sind so global eindeutig.

### Aufbau eines Codes für einen Fahrschein

**Länderkürzel:Organisationscode:FahrscheinCode**

### Beispiel

**Einzel-Ticket für Erwachsene im VVS-  
Gebiet für 2 Zonen**      de:vvs:EinzelErw2Z

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, XML-Schemata und Konventionen

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, 2014) und (VDV-Schrift 430, 2014) 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
<b>Ortsinformation</b>	LocationInformationRequest	Trias_Locations.xsd	8
<b>Verbindungsauskunft</b>	TripRequest	Trias_Trips.xsd	9
<b>Abfahrtstafeln</b>	StopEventRequest	Trias_StopEvents.xsd	10
<b>Logische Ortung</b>	PositioningRequest	Trias_Positioning.xsd	11
<b>Fahrtinformation (EKAP)</b>	TripInfoRequest	Trias_TripInfo.xsd	12
<b>Anschlussmeldung</b>	ConnectionDemandRequest	Trias_Connections.xsd	13.1.1
<b>Anschlussstatus</b>	ConnectionStatusRequest	Trias_Connections.xsd	13.1.2
<b>Info bei Anschlussverlust</b>	ConnectionStatusResponse	Trias_Connections.xsd	13.1.3
<b>Anschlussrückmeldung</b>	ConnectionReportRequest	Trias_Connections.xsd	13.1.4
<b>Fahrpreis- und Tarifberechnung</b>	FaresRequest	Trias_Fares.xsd	0
<b>Buchungsinformation</b>	BookingInfoRequest	Trias_Booking.xsd	16



Dienst	Bezeichnung des Anfrageelements	Schema-Datei	Kapitel
<b>IV-Routing</b>	IndividualRouteRequest	Trias_- IndividualTrips.xsd	17
<b>Kartendienst</b>	MapServiceRequest ImageCoordinatesRequest GeoCoordinatesRequest	Trias_Maps.xsd	18
<b>Schadensmeldung</b>	FacilityStatusReport	Trias_Facilities.xsd	19.4
<b>Zustand von Einrichtungen</b>	FacilityRequest	Trias_Facilities.xsd	19.6
<b>Benachrichtigungsdienst</b>	SubscriptionRequest	Trias.xsd	0 und 20
<b>Personalisierungsdienst</b>	PersonalisationRequest	Trias_- Personalisation.xsd	21
<b>Fahrzeuginformation</b>	VehicleDataRequest	Trias_- VehicleInterface.xsd	22
<b>Fahrzeuginteraktion</b>	VehicleInteractionRequest	Trias_- VehicleInterface.xsd	23
<b>Diensteregister</b>	ServiceRegisterRequest	Trias_- ServiceRegister.xsd	24
<b>Authentifizierung</b>	AbstractTriasServiceRequest (vererbt auf alle TRIAS-Nachrichten)	Trias_- RequestSupport.xsd	7.9 und 25

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.

## 6.3. Importierte Siri-Schemata

Aus der SIRI-Schnittstellen-Spezifikation der Version 1.4 werden die Schemadateien

- siri.xsd
- siri\_facilities-v1.2.xsd
- siri\_situation-v1.1.xsd
- siri\_requests-v1.3.xsd
- siri\_common-v1.4.xsd
- siri\_situationExchange\_service.xsd
- siri\_facilityMonitoring\_service.xsd

nach TRIAS importiert.

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).

## 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, 2011), 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	<b>OK (true)</b>	Request successful. (Anfrage erfolgreich bearbeitet.)
Systemic Error	<b>RequestTimeout</b>	Server not responding. (Server antwortet nicht.)
	<b>InvalidRequest</b>	The server does not "understand" the request. The client should not repeat the request. (Der Server "versteh" die Anfrage nicht. Der Client braucht die Anfrage nicht zu wiederholen.)
	<b>Unauthorized</b>	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.)
	<b>Forbidden</b>	The server "understands" the request, but cannot carry it out. (Der Server "versteh" die Anfrage, kann sie aber nicht ausführen.)
	<b>NotFound</b>	The requested URL was not found. (Die angefragte URL konnte nicht gefunden werden.)
Application Error	<b>VersionNotSupported</b>	Service is not available. (Die angefragte Version des Dienstes ist nicht verfügbar.)
	<b>CapabilityNotSupported</b>	Service does not support the requested capability. (Die angeforderte Funktionalität wird vom Dienst nicht unterstützt.)
	<b>ServiceNotAvailable</b>	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).)
	<b>AccessNotAllowed</b>	Requestor is not authorised to the service or data requested. (Der Anfrager ist für den Zugriff auf den Dienst oder die Daten nicht autorisiert.)
	<b>NoInfoForTopic</b>	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.)
	<b>UnknownSubscriber</b>	Subscriber not found. (Der Abonnent wurde nicht gefunden.)
	<b>UnknownSubscription</b>	Subscription not found. (Das Abonnement wurde nicht gefunden.)
	<b>AllowedResourceUsageExceeded</b>	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.)
<b>OtherError</b>	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:

<b>FehlerCode</b>	<b>Fehlerbedeutung</b>
<b>AUTH_FAILURE</b>	Dieser Fehler tritt auf, wenn eine Anfrage mit ungültiger oder nicht prüfbarer Signatur empfangen wurde.
<b>AUTH_MISSING</b>	Dieser Fehlercode tritt auf, wenn der Server zwingend eine Authentifizierung benötigt, aber eine Nachricht ohne Signatur empfangen wurde.
<b>AUTH_USER_UNKNOWN</b>	Dieser Fehlercode wird zurückgegeben, wenn die Authentifikation fehlschlägt, weil der Benutzer unbekannt ist.
<b>TRIASGENERIC_ERROR</b>	Bei der Verarbeitung der Anfrage ist ein Fehler aufgetreten, der nicht durch einen speziellen Fehlercode abgedeckt wird, Einzelheiten werden im Text der Fehlermeldung genannt.
<b>TRIASGENERIC_SERVICENOTSUPPORTED</b>	In der Anfrage wurde ein Dienst spezifiziert, der vom Server nicht unterstützt wird (z.B. Dienst <i>ConnectionDemand</i> ).
<b>TRIASGENERIC_REQUESTNOTSUPPORTED</b>	Es wurde eine Anfrage spezifiziert, die vom Server nicht unterstützt wird (z.B. Anfrage <i>FacilityStatusReport</i> ).
<b>TRIASGENERIC_FEATURENOTSUPPORTED</b>	In der Anfrage wurde ein Feature spezifiziert, das vom Server nicht unterstützt wird (z.B. Parameter <i>NotVia</i> in <i>TripRequest</i> )
<b>TRIASGENERIC_LANGUAGENOTSUPPORTED</b>	In der Anfrage wurde eine Sprache für die Anzeige der Ergebnistexte spezifiziert, die vom Server nicht unterstützt wird (zumindest im Kontext der vorliegenden Anfrage).
<b>TRIASGENERIC_EXCEPTIONFROMREQUESTEDLANGUAGE</b>	In der Anfrage wurde eine Sprache für die Anzeige der Ergebnistexte spezifiziert, die vom Server nicht bei allen Textelementen der Antwort unterstützt wird.
<b>TRIASGENERIC_DATAVERSIONNOTAVAILABLE</b>	Die in der Anfrage angeforderte Datenversion konnte vom Server nicht berücksichtigt werden.

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

## 6.5. Haltsequenznummern und Fahrtabschnitte in TRIAS-Diensten

In diversen TRIAS-Diensten werden Haltsequenznummern oder Fahrtabschnitte verwendet. Eine Haltsequenznummer gibt an, an wievielter Stelle ein Halt in der Haltstellenfolge einer Fahrt steht. Dies wird beispielsweise in den Diensten *StopEvents* und *TripInfo* verwendet. Ein Fahrtabschnitt besteht aus einer Haltsequenznummer, die den Beginn des Abschnitts kennzeichnet und aus einer zweiten Haltsequenznummer, die das Ende des Abschnitts kennzeichnet. Solche Fahrtabschnitte werden genutzt, um deutlich zu machen, dass bestimmte Eigenschaften einer Fahrt nur auf gewissen Abschnitten der Fahrt gelten (siehe *ServiceAttributeStructure*, *ServiceSectionStructure* und *ParallelServiceStructure*).

Die Verwendung von Haltsequenznummern in TRIAS-Diensten unterliegt gewissen, stets gleichen Konventionen, die im Folgenden zusammengefasst werden:

- Alle Haltsequenznummern beziehen sich stets auf die vollständige Fahrt (wie sie im Fahrplan der EKAP enthalten ist). Auch in Kontexten, in denen nur ein Teil einer Fahrt genutzt wird (z.B. im Dienst *Trips*) beziehen sich die Haltsequenznummern auf die gesamte Fahrt und nicht auf den beauskunfteten Abschnitt.
- Haltsequenznummern sind stets als *StopSeqNumber* bezeichnet.
- Haltsequenznummern werden stets von 1 an gezählt.

Im Bezug auf Fahrtabschnitte gibt es einige zusätzlichen Konventionen, die im Folgenden zusammengefasst werden:

- Fahrtabschnitte werden stets mit der Gruppe *StopSeqIntervalGroup* und ihren beiden Elementen *FromStopSeqNumber* und *ToStopSeqNumber* gekennzeichnet. Für diese Elemente gelten obigen Regeln für Haltsequenznummern.
- *FromStopSeqNumber* und *ToStopSeqNumber* sind jeweils optional. Werden sie angegeben, beziehen sie sich stets auf die komplette Fahrt, auch in Zusammenhängen, in denen nur ein Teil einer Fahrt genutzt wird (z.B. in *Trips*).
- Wird *FromStopSeqNumber* im Dienst *Trips* nicht angegeben, so beginnt der zugehörige Fahrtabschnitt mit dem Fahrtabschnitt, der tatsächlich genutzt wird. Über „davor“ wird keine Aussage getroffen. Wird *ToStopSeqNumber* im Dienst *Trips* nicht angegeben, so endet der zugehörige Fahrtabschnitt mit dem Fahrtabschnitt der tatsächlich genutzt wird. Über „danach“ wird keine Aussage getroffen. Auf diese Weise kann für Fahrteigenschaften, die für den gesamten genutzten Fahrtabschnitt gelten, auf die Angabe eines expliziten Fahrtabschnittes verzichtet werden.
- Wird *FromStopSeqNumber* in den Diensten *StopsEvents* oder *TriplInfo* nicht angegeben, so beginnt der zugehörige Fahrtabschnitt mit der Fahrt. Wird *ToStopSeqNumber* in den Diensten *StopEvents* oder *TriplInfo* nicht angegeben, so endet der zugehörige Fahrtabschnitt mit der Fahrt. Auf diese Weise kann für Fahrteigenschaften, die für gesamte Fahrt gelten, auf die Angabe eines expliziten Fahrtabschnittes verzichtet werden.

## 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

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

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

## 7.1.2. SubscriptionRequestStructure

<b>SubscriptionRequestStructure</b>			<b>+Structure</b>	<b>Grundstruktur für jede TRIAS-Abonnement-Anfrage.</b>
<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. 7.4.9).
	<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>	<b>a</b> <i>SituationExchangeSubscriptionRequest</i>	-1:1	<i>SituationExchangeSubscriptionRequest</i>	Inhalt der Abonnement-Anfrage für allgemeine Ereignis- und Störungsbenachrichtigung (vgl. Kapitel 20).
	<b>b</b> <i>FacilityMonitoringSubscriptionRequest</i>		<i>FacilityMonitoringSubscriptionRequest</i>	Inhalt der Abonnement-Anfrage für Zustandsmeldungen zur Infrastruktur von Haltestellen und Fahrzeugen (vgl. Kapitel 20).
	<b>c</b> <i>TripMonitoringSubscriptionRequest</i>		<i>TripMonitoringSubscriptionRequest</i>	Inhalt der Abonnement-Anfrage für Benachrichtigungen zu einer bestimmten Reiseverbindung (vgl. 20.3.1).

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

### 7.1.3. RequestPayloadStructure

<b>RequestPayloadStructure</b>		<b>+Structure</b>	<b>Element zur Auswahl des gewünschten TRIAS-Dienstes.</b>
<b>a</b>	<b>BookingInfoRequest</b>	<i>BookingInfoRequest</i>	Anfrage nach Buchungsinformationen (vgl. 16.2.1).
<b>b</b>	<b>ConnectionDemandRequest</b>	<i>ConnectionDemandRequest</i>	Anfrage zur Anschlussvoranmeldung (vgl. 13.4.1).
<b>c</b>	<b>ConnectionDemandDeleteRequest</b>	<i>ConnectionDemandDeleteRequest</i>	Löschung einer Anschlussvoranmeldung (vgl. 13.4.2).
<b>d</b>	<b>ConnectionReportRequest</b>	<i>ConnectionReportRequest</i>	Meldung des Fahrgasts, ob ein Anschluss geklappt hat (vgl. 13.8.1).
<b>e</b>	<b>ConnectionStatusRequest</b>	<i>ConnectionStatusRequest</i>	Anfrage zum Anschlussstatus (vgl. 13.6.1).
<b>f</b>	<b>FacilityRequest</b>	<i>FacilityRequest</i>	Anfrage zu Fahrzeug- und Infrastruktureinrichtungen (vgl. 19.6.1).
<b>g</b>	<b>FacilityStatusReport</b>	<i>FacilityStatusReport</i>	Übermittlung des Zustands von Fahrzeug- und Infrastruktureinrichtungen in einem aktiven Abonnement (vgl. 19.4.1).
<b>h</b>	<b>FaresRequest</b>	<i>FaresRequest</i>	Anfrage zum Tarifberechnungsdienst (vgl. 14.2.1).
<b>i</b>	<b>GeoCoordinatesRequest</b>	<i>GeoCoordinatesRequest</i>	Anfrage nach Geo-Koordinaten (vgl. 18.2.3).
<b>j</b>	<b>ImageCoordinatesRequest</b>	<i>ImageCoordinatesRequest</i>	Anfrage nach Bildkoordinaten (vgl. 18.2.2).
<b>k</b>	<b>IndividualRouteRequest</b>	<i>IndividualRouteRequest</i>	Anfrage nach einer IV-Route (vgl. 17.2.1).
<b>l</b>	<b>LocationInformationRequest</b>	<i>LocationInformationRequest</i>	Anfrage zum Ortsinformationsdienst (vgl. 8.3.1).
<b>m</b>	<b>MapServiceRequest</b>	<i>MapServiceRequest</i>	Anfrage zum Kartendienst (vgl. 18.2.1).
<b>n</b>	<b>PersonalisationRequest</b>	<i>PersonalisationRequest</i>	Anfrage zum Personalisierungsdienst (vgl. 21.4.1).
<b>o</b>	<b>PositioningRequest</b>	<i>PositioningRequest</i>	Anfrage zum Positionierungsdienst (vgl. 11.2.1).
<b>p</b>	<b>RefineRequest</b>	<i>RefineRequest</i>	Anfrage zur Anreicherung von Strukturen (vgl. 15.2.1).
<b>q</b>	<b>ServiceRegisterRequest</b>	<i>ServiceRegisterRequest</i>	Anfrage zum Dienstregisterdienst (vgl. 24.3.1).
<b>r</b>	<b>StopEventRequest</b>	<i>StopEventRequest</i>	Anfrage nach Abfahrtstafeln (vgl. 10.2.1).
<b>s</b>	<b>TripInfoRequest</b>	<i>TripInfoRequest</i>	Anfrage nach Fahrtinformationen (vgl. 12.2.1).
<b>t</b>	<b>TripRequest</b>	<i>TripRequest</i>	Anfrage nach einer intermodalen Verbindungsberechnung (vgl. 9.2.1).
<b>u</b>	<b>VehicleDataRequest</b>	<i>VehicleDataRequest</i>	Anfrage nach Fahrzeuginformationen (vgl. 22.2.1).
<b>v</b>	<b>VehicleInteractionRequest</b>	<i>VehicleInteractionRequest</i>	Anfragen, die an ein Fahrzeug gerichtet werden, um dort eine Aktion auszulösen (vgl. 23.2.1). Beinhaltet u. a. den StopRequestRequest.

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

## 7.1.4. ServiceDeliveryStructure

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

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

## 7.1.5. DeliveryPayloadStructure

<b>DeliveryPayloadStructure</b>			<b>+Structure</b>	<b>Element zur Auswahl der passenden TRIAS-Antwort.</b>
<b>a</b>	<b>BookingInfoResponse</b>		<i>BookingInfoResponse</i>	Antwort mit Buchungsinformationen (vgl. 16.3.1).
<b>b</b>	<b>ConnectionDemandResponse</b>		<i>ConnectionDemandResponse</i>	Antwort zur Anschlussvoranmeldung (vgl. 13.5.1).
<b>c</b>	<b>ConnectionDemandDeleteResponse</b>		<i>ConnectionDemandDeleteResponse</i>	Bestätigung zur Löschung einer Anschlussvoranmeldung (vgl. 13.5.2).
<b>d</b>	<b>ConnectionReportResponse</b>		<i>ConnectionReportResponse</i>	Bestätigung zur Anschlusserrichtungsmeldung (vgl. 13.9.1).
<b>e</b>	<b>ConnectionStatusNotification</b>		<i>ConnectionStatusNotification</i>	Übermittlung des Anschlussstatus in einem aktiven Abonnement (vgl. 13.6.2).
<b>f</b>	<b>ConnectionStatusResponse</b>		<i>ConnectionStatusResponse</i>	Antwort zu Anschlussstatusanfrage (vgl. 13.7.1).
<b>g</b>	<b>FacilityMonitoringDelivery</b>		<i>si-ri:FacilityMonitoringDelivery</i>	Aktualisierungsnachricht im Rahmen eines Abonnements zu Ausstattungsmerkmalen und Einrichtungen (vgl. Kapitel 20).
<b>h</b>	<b>FacilityResponse</b>		<i>FacilityResponse</i>	Antwort zu Fahrzeug- und Infrastruktureinrichtungen (vgl. 19.7.1).
<b>i</b>	<b>FacilityStatusReportResponse</b>		<i>FacilityStatusReportResponse</i>	Bestätigung zur Übermittlung des Zustands von Fahrzeug- und Infrastruktureinrichtungen (Schadensmeldung, vgl. 19.5.1).
<b>j</b>	<b>FaresResponse</b>		<i>FaresResponse</i>	Antwort zur Tarfberechnungsanfrage (vgl. 14.3.1).
<b>k</b>	<b>GeoCoordinatesResponse</b>		<i>GeoCoordinatesResponse</i>	Antwort zur Anfrage nach Geo-Koordinaten (vgl. 18.3.3).
<b>l</b>	<b>ImageCoordinatesResponse</b>		<i>ImageCoordinatesResponse</i>	Antwort zur Anfrage nach Bildkoordinaten (vgl. 18.3.2).
<b>m</b>	<b>IndividualRouteResponse</b>		<i>IndividualRouteResponse</i>	Antwort mit berechneten IV-Routen (vgl. 17.3.1).
<b>n</b>	<b>LocationInformationResponse</b>		<i>LocationInformationResponse</i>	Antwort des Ortsinformationsdiensts (vgl. 8.4.1).
<b>o</b>	<b>MapServiceResponse</b>		<i>MapServiceResponse</i>	Antwort des Kartendienstes (vgl. 18.3.1).
<b>p</b>	<b>PersonalisationResponse</b>		<i>PersonalisationResponse</i>	Antwort des Personalisierungsdienstes (vgl. 21.5.1).
<b>q</b>	<b>PositioningResponse</b>		<i>PositioningResponse</i>	Antwort des Positionierungsdienstes (vgl. 11.3.1).



<b>r</b>	<b>RefineResponse</b>	<i>Refine-Response</i>	Antwort des Anreicherungsdiensts (vgl. 15.3.1)
<b>s</b>	<b>ServiceRegisterResponse</b>	<i>ServiceRegisterResponse</i>	Antwort des Dienstregisterdiensts (vgl. 24.4.1).
<b>t</b>	<b>SituationExchangeDelivery</b>	<i>si-ri:SituationExchangeDelivery</i>	Aktualisierungsnachricht im Rahmen eines Abonnents zu Störungsinformationen (vgl. Kapitel 20).
<b>u</b>	<b>StopEventResponse</b>	<i>StopEventResponse</i>	Antwort mit Haltestellen-Abfahrtstafeln (vgl. 10.2.1).
<b>v</b>	<b>TripInfoResponse</b>	<i>TripInfoResponse</i>	Antwort mit Fahrtinformationen (vgl. 12.3.1).
<b>w</b>	<b>TripMonitoringDelivery</b>	<i>TripMonitoringDelivery</i>	Aktualisierungsnachricht im Rahmen eines Abonnents zu Verbindungszuständen (vgl. 20.4.1).
<b>x</b>	<b>TripResponse</b>	<i>TripResponse</i>	Antwort zur intermodalen Verbindungsberechnung (vgl. 9.3.1).
<b>y</b>	<b>VehicleDataResponse</b>	<i>VehicleDataResponse</i>	Antwort mit Fahrzeuginformationen (vgl. 22.3.1).
<b>z</b>	<b>VehicleInteractionResponse</b>	<i>VehicleInteractionResponse</i>	Fahrzeugantwort auf eine Interaktionsanfrage (vgl. 23.3.1). Beinhaltet u. a. die StopRequestResponse.

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:

<b>Typname</b>	<b>Basistyp</b>	<b>Beschreibung</b>
<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>	Geografische Länge.
<i>LatitudeType</i>	<i>xs:decimal</i>	Geografische 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

<b>InternationalTextStructure</b>			<b>+Structure</b>	<b>Ein Text mit einer Text-ID und Angabe der Sprache, in der er verfasst ist.</b>
	<b>Text</b>	1:1	<i>xs:normalizedString</i>	Text.
	<b>TextId</b>	0:1	<i>xs:NMTOKEN</i>	ID des Texts.
	<b>Language</b>	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. Um mehrere Sprachen ausgeben zu können, z. B. in mehrsprachigen Regionen, ist dieser Typ „unbound“ im Schema vorgesehen.

## 7.2.3. GeoPositionStructure

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

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

## 7.2.4. WebLinkStructure

<b>WebLinkStructure</b>			<b>+Structure</b>	<b>URL mit Beschriftungstext zu einer Ressource im Web</b>
	<b>Label</b>	1:*	<i>International-Text</i>	Beschriftungstext des Links (s. 7.2.2).
	<b>Url</b>	1:1	<i>xs:anyURI</i>	URL zur Web-Ressource.

Tabelle 13: Beschreibung der Struktur **WebLinkStructure**.

Um weitergehende Informationen zu einem Objekt abrufen zu können, werden in die TRIAS-Antworten an manchen Stellen Elemente des Typs *WebLinkStructure* eingefügt. Ein Client kann die darin enthaltenen URLs nutzen, um weitere Aktionen auszuführen oder Informationen abzurufen. Da die Clients von unterschiedlichem Typ (z.B. App oder Web-Browser) sein können, sind die Anbieter der Web-Ressourcen aufgerufen, eine möglichst breite Unterstützung unterschiedlicher Clienttypen anzubieten.

## 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:

Typname	Werte	Beschreibung
<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>walk</i> : Fußweg <i>demandResponsive</i> : Bedarfsverkehr ohne Fahrplan auf Anforderung <i>replacementService</i> : z.B. Shuttle-Service im Ersatzverkehr
<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   local   highSpeedRail   suburbanRailway   regionalRail   interregionalRail   longDistance   international   sleeperRailService   nightRail   carTransportRailService   touristRailway   railShuttle   replacementRailService   specialTrain   crossCountryRail   rackAndPinionRailway</i>	Unter-Klassifizierung der Züge (nach TPEG pti_table 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   scheduled-</i>	Unter-Klassifizierung der Wasserverkehrsmittel (nach TPEG pti_table 07).

	<i>Ferry   shuttleFerryService</i>	
<i>AirSubmodeEnumeration</i>	<i>unknown   undefined   internationalFlight   domesticFlight   intercontinentalFlight   domesticScheduledFlight   shuttleFlight   intercontinentalCharterFlight   internationalCharterFlight   roundTripCharterFlight   sightseeingFlight   helicopterService   domesticCharterFlight   SchengenAreaFlight   airshipService   shortHaulInternationalFlight   canalBarge</i>	Unter-Klassifizierung der Luftverkehrsmittel (nach TPEG pti_table 08).
<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 14: 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

<b>IndividualTransportOptionsStructure</b>		<b>+Structure</b>	<b>Arten von IV und deren Nutzungsgrenzen, wie sie der Benutzer vorgegeben hat.</b>
<i>Mode</i>	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 15: Beschreibung der Struktur **IndividualTransportOptionsStructure**.

### 7.3.3. PtSubmodeChoiceGroup

<b>PtSubmodeChoiceGroup</b>			<b>+Group</b>	<b>Gruppe zur Auswahl der Verkehrsmitteluntertypen.</b>
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 16: Beschreibung der Gruppe **PtSubmodeChoiceGroup**.

### 7.3.4. ModeStructure

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

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

### 7.3.5. PtModeFilterStructure

<b>PtModeFilterStructure</b>			<b>+Structure</b>	<b>Struktur zum Filtern nach Verkehrsmitteltypen</b>
	<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>	<i>...</i>	0:*	<i>PtSubmodeChoice</i>	ÖV-Untertypen.

Tabelle 18: Beschreibung der Struktur **PtModeFilterStructure**.

Die Angabe von PtModes und PtSubmodeChoice ist additiv: egal wie der Modus ist, werden die Angaben addiert. Im Modus „exclude“ werden die PtModes und die Submodes ausgeschlossen. Im Modus „include“ werden die PtModes und die Submodes eingeschlossen.

## 7.4. Trias\_Common

### 7.4.1. Einfache Typen

Folgende einfache Typen (simple types) werden definiert:

<b>Typname</b>	<b>Basistyp</b>	<b>Beschreibung</b>
<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 19: 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

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

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

### 7.4.3. PrivateCodeStructure

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

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

### 7.4.4. OperatorFilterStructure

<i>OperatorFilterStructure</i>			<i>+Structure</i>	<b>Struktur zum Filtern nach Verkehrsunternehmen.</b>
	<i>Exclude</i>	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> .
	<i>OperatorRef</i>	0:*	<i>→Operator</i>	Referenz auf Verkehrsunternehmen. Vgl. 7.4.1.

Tabelle 22: Beschreibung der Struktur **OperatorFilterStructure**

### 7.4.5. LineDirectionStructure

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

Tabelle 23: Beschreibung der Struktur **LineDirectionStructure**.

### 7.4.6. LineDirectionFilterStructure

<i>LineDirectionFilterStructure</i>			<i>+Structure</i>	<b>Filterstruktur zum Ein/Ausschließen von Linien(richtungen)</b>
	<b>Line</b>	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 24: Beschreibung der Struktur **LineDirectionFilterStructure**.

### 7.4.7. SharingServiceStructure

<b>SharingServiceStructure</b>		<b>+Structure</b>		<b>Struktur zur Beschreibung eines Mobilitätsdienstes mit Ausleihfahrzeugen</b>
	<b>OperatorRef</b>	1:1	→Operator	Operator-ID. Vgl. 7.4.1.
	<i>Name</i>	0:1	xs:string	Name des Mobilitätsdienstes.
	<i>SharingModel</i>	0:1	singleStationBased / multipleStationBased / nonStationBased	Typ des Ausleih- und Rückgabeverfahrens.
SharingServiceUsage	<i>TimeBufferBefore</i>	0:1	xs:duration	Typische Zeit, die ein Benutzer einplanen muss, um sich am System anzumelden und das Fahrzeug in Fahrbereitschaft zu versetzen.
	<i>TimeBufferAfter</i>	0:1	xs:duration	Typische Zeit, die ein Benutzer einplanen muss, um das Fahrzeug ordnungsgemäß abzustellen, zu verschließen und sich am System abzumelden.
	<i>InfoURL</i>	0:1	+WebLink	Link zu Web-Seite mit weiterführenden Informationen (vgl. 7.2.4).

Tabelle 25: Beschreibung der Struktur **SharingServiceStructure**.

### 7.4.8. OperatingDaysStructure

<b>OperatingDaysStructure</b>		<b>+Structure</b>		<b>Struktur für die Definition von Verkehrstagen mittels Bit-Kette.</b>
	<b>From</b>	1:1	xs:date	Startdatum des Zeitraums.
	<b>To</b>	1:1	xs:date	Enddatum des Zeitraums.
	<b>Pattern</b>	1:1	BitString	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 26: Beschreibung der Struktur **OperatingDaysStructure**.

### 7.4.9. WeekdayTimePeriodStructure

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

Tabelle 27: Beschreibung der Struktur **WeekdayTimePeriodStructure**.



## 7.4.10. GeneralAttributeStructure

<b>GeneralAttributeStructure</b>			<b>+Structure</b>	<b>Struktur für die Definition von Attributen/Hinweisen.</b>
	<b>Text</b>	1:*	+International Text	Attributtext für die Fahrgastinformation.
	<b>Code</b>	1:1	xs:NMTOKEN	Interner Attribute-Code. Kann verwendet werden, um mehrfaches Auftreten desselben Attributs zu erkennen.
AllFacilities	:::	0:1	+AllFacilitiesGroup	Klassifizierung nach TPEG. Vgl. 7.7.4.
	Mandatory	0:1	xs:boolean	Legt fest, ob das Attribut in jedem Fall angezeigt werden muss. Voreinstellung ist <i>false</i> .
	Importance	0:1	Percent	Wichtigkeit für die Priorisierung von Attributen gegeneinander.
	InfoURL	0:1	xs:anyURI	URL zu weiteren Informationen über dieses Attribut. Falls vorhanden, soll der gesamte Text als Link zu dieser URL gekennzeichnet werden.
	Status	0:1	Unknown / Planned / AsPlanned / NotAsPlanned / RealtimeUpdate	Gibt den Status eines Attributs an, z.B. in einer Anreicherungsanfrage. <ul style="list-style-type: none"> <li>• <i>Planned</i> bedeutet, dass die Angabe des Attributs auf den Planungsangaben beruht (z.B. ein Zug soll planmäßig einen Restaurantwagen mit sich führen).</li> <li>• <i>AsPlanned</i> bedeutet, dass bereits bekannt ist, dass das Attribut wie geplant vorhanden ist/sein wird (z.B. ein Zug führt wie geplant einen Restaurantwagen mit sich).</li> <li>• <i>NotAsPlanned</i> bedeutet, dass bereits bekannt ist, dass ein Attribut nicht wie geplant vorhanden ist (z.B. ein Zug führt im Gegensatz zur Planung doch keinen Restaurantwagen mit sich).</li> <li>• <i>RealtimeUpdate</i> wird verwendet, um ein Attribut mitzuteilen, das erst nach der Planungszeit bekannt wurde/entstand (z.B. Hinweis auf die Fahrzeugausstattung).</li> </ul>

Tabelle 28: 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:

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

Tabelle 29: 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

<b>StopPointStructure</b>			<b>+Structure</b>	<b>Modellierung eines Haltepunkts.</b>
<i>Stop-Point</i>	<b>StopPointRef</b>	1:1	→ <i>StopPoint</i>	Referenz auf einen Code für einen Haltepunkt. Vgl. 7.5.1.
	<b>StopPointName</b>	1:*	+ <i>International Text</i>	Name des Haltepunkts für Fahrgastinformation.
	<i>NameSuffix</i>	0:*	+ <i>International Text</i>	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „gegenüber vom Haupteingang“.
	<i>PlannedBay</i>	0:*	+ <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:*	+ <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 Schlüsselsystem. 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.
<i>StopAttributes</i>	<i>WheelchairAccessible</i>	0:1	<i>xs:boolean</i>	Rollstuhltauglichkeit dieses Haltepunkts. Default ist <i>false</i> .
	<i>Lighting</i>	0:1	<i>xs:boolean</i>	Angabe zur Beleuchtung dieses Haltepunkts. Default ist <i>false</i> .
	<i>Covered</i>	0:1	<i>xs:boolean</i>	Angabe, ob dieser Haltepunkt Witterungsschutz bietet (vor Regen, Schnee, Sturm etc.). Default ist <i>false</i> .

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

### 7.5.3. StopPlaceStructure

<b>StopPlaceStructure</b>			<b>+Structure</b>	<b>Modellierung einer Haltestelle.</b>
<i>StopPlace</i>	<b>StopPlaceRef</b>	1:1	→ <i>StopPlace</i>	Referenz auf einen Code für eine Haltestelle. Vgl. 7.5.1.
	<b>StopPlaceName</b>	1:*	+ <i>International Text</i>	Name der Haltestelle für Fahrgastinformation.
	<i>NameSuffix</i>	0:*	+ <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 Schlüsselsystem.
	<i>LocalityRef</i>	0:1	→ <i>Locality</i>	Referenz auf die Ortschaft, zu der diese Haltestelle gehört. Vgl. 7.5.1.

Stop- pAttribu- tes	<i>WheelchairAccessible</i>	0:1	<i>xs:boolean</i>	Rollstuhlgängigkeit dieser Haltestelle insgesamt. Default ist <i>false</i> .
	<i>Lighting</i>	0:1	<i>xs:boolean</i>	Angabe zur Beleuchtung dieser Haltestelle. Default ist <i>false</i> .
	<i>Covered</i>	0:1	<i>xs:boolean</i>	Angabe, ob diese Haltestelle Witterungsschutz bietet (vor Regen, Schnee, Sturm etc.). Default ist <i>false</i> .

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

#### 7.5.4. LocalityStructure

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

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

#### 7.5.5. PointOfInterestStructure

<b>PointOfInterestStructure</b>			<b>+Structure</b>	<b>Modellierung eines wichtigen Punkts (POI).</b>
	<b>PointOfInterestCode</b>	1:1	→ <i>PointOfInterest</i>	Identifikator des POI.
	<b>PointOfInterestName</b>	1:*	+ <i>International Text</i>	Name des POI für Fahrgastinformation.
	<i>NameSuffix</i>	0:*	+ <i>International Text</i>	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „Messe/Exhibition Center“.
	<i>PointOfInterestCategory</i>	0:*	+ <i>PointOfInterestCategory</i>	Kategorien, die diesem POI zugeordnet sind. Vgl. 7.5.6. Falls mehrere aufgeführt sind, sind sie nach absteigender Relevanz sortiert.
	<i>PrivateCode</i>	0:*	+ <i>PrivateCode</i>	Privater Code für diesen POI in einem anderen Schlüsselssystem.
	<i>LocalityRef</i>	0:1	→ <i>Locality</i>	Referenz auf die zugeordnete Ortschaft, zu der diesem POI gehört. Vgl. 7.5.1.

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

#### 7.5.6. PointOfInterestCategoryStructure

<b>PointOfInterestCategoryStructure</b>			<b>+Structure</b>	<b>Modellierung einer POI-Kategorie-Liste.</b>
	<b>OsmTag</b>	1:*	+ <i>OsmTag</i>	Liste der POI-Kategorien, definiert durch Schlüssel-Wert-Paare wie in OpenStreetMap. <sup>7</sup> Vgl. 7.5.7

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

<sup>7</sup> [http://wiki.openstreetmap.org/wiki/DE:Map\\_Features](http://wiki.openstreetmap.org/wiki/DE:Map_Features)

### 7.5.7. OsmTagStructure

<b>OsmTagStructure</b>			<b>+Structure</b>	<b>Modellierung einer POI-Kategorie.</b>
	<b>Tag</b>	1:1	<i>xs:NMTOKEN</i>	Name des OpenStreetMap-Tags (z.B. amenity, leisure, tourism, bike, ...)
	<b>Value</b>	1:1	<i>xs:NMTOKEN</i>	Wert des OpenStreetMap-Tags (z.B. yes, hostel, charging_station, ...)

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

### 7.5.8. PointOfInterestFilterStructure

<b>PointOfInterestFilterStructure</b>			<b>+Structure</b>	<b>Struktur zum Filtern nach POI-Kategorien</b>
	<i>Exclude</i>	0:1	<i>xs:boolean</i>	Legt fest, ob die nachfolgenden Kategorien bei der POI-Suche als einzige eingeschlossen ( <i>Exclude=false</i> ) oder ausgeschlossen werden sollen ( <i>Exclude=true</i> ). Default ist <i>false</i> .
	<b>PointOfInterestCategory</b>	1:*	<i>+PointOfInterestCategory</i>	Bezeichner für POI-Kategorien. Vgl. 7.5.6. Wenn mehrere aufgelistet sind, werden die Kategorien bei der Suche mit einem logischen „ODER“ (im Fall von <i>Exclude=false</i> ) bzw. mit einem logischen „UND“ (im Fall von <i>Exclude=true</i> ) berücksichtigt.

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

### 7.5.9. AddressStructure

<b>AddressStructure</b>			<b>+Structure</b>	<b>Modellierung einer Adresse.</b>
	<b>AddressCode</b>	1:1	<i>→Address</i>	Identifikator der Adresse. Vgl. 7.5.1.
	<i>PrivateCode</i>	0:*	<i>+PrivateCode</i>	Privater Code für diese Adresse in einem anderen Schlüsselsystem.
	<b>AddressName</b>	1:*	<i>+InternationalText</i>	Formatierte Adressbeschriftung für Fahrgastinformation, enthält alle relevanten Adress-Bestandteile, z.B. „Lister Str. 15, 30163 Hannover“.
	<i>NameSuffix</i>	0:*	<i>+InternationalText</i>	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „Messe/Exhibition Center“.
<i>AddressDetail</i>	<i>CountryName</i>	0:1	<i>xs:string</i>	Angabe zum Land.
	<i>PostalCode</i>	0:1	<i>xs:string</i>	Postleitzahl.
	<i>LocalityName</i>	0:1	<i>xs:string</i>	Name der Stadt oder Ortschaft, in der die Adresse liegt.
	<i>LocalityRef</i>	0:1	<i>→Locality</i>	Referenz auf die Stadt oder Ortschaft, zu der dieser Adresse gehört. Vgl. 7.5.1.
	<i>StreetName</i>	0:1	<i>xs:string</i>	Name der Straße, in der die Adresse liegt, z.B. „Barbarastr.“.
	<i>HouseNumber</i>	0:1	<i>xs:string</i>	Hausnummer inklusive Zusatz, z.B. „3-9, Block 6“. 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	<i>xs:string</i>	Name der kreuzenden Straße.

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

## 7.5.10. LocationStructure

<b>LocationStructure</b>			<b>+Structure</b>	<b>Modell eines allgemeinen Ortspunkts (Haltepunkt, Haltestelle, Koordinatenposition, Ortschaft, POI oder Adresse).</b>
a	<i>StopPoint</i>	-0:1	<b>+StopPoint</b>	Angaben zu einem Haltepunkt. Vgl. 7.5.2.
b	<i>StopPlace</i>		<b>+StopPlace</b>	Angaben zur Haltestelle. Vgl. 7.5.3.
c	<i>Locality</i>		<b>+Locality</b>	Angaben zu einer Stadt/Ortschaft. Vgl. 7.5.4.
d	<i>PointOfInterest</i>		<b>+PointOfInterest</b>	Angaben zu einem POI. Vgl. 7.5.5.
e	<i>Address</i>		<b>+Address</b>	Angaben zu einer Adresse. Vgl. 7.5.9.
<b>LocationName</b>		<b>1:*</b>	<b>+InternationalText</b>	Name oder Bezeichnung des Ortspunkts.
<b>GeoPosition</b>		<b>1:1</b>	<b>+GeoPosition</b>	Koordinatenposition. Vgl. 7.2.3..
<i>Attribute</i>		<b>0:*</b>	<b>+GeneralAttribute</b>	Attribute, die dem Ortspunkt zugeordnet sind. Vgl. 7.4.10.
<i>Extension</i>		<b>0:1</b>	<b>xs:anyType</b>	Erweiterungen.

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

## 7.5.11. LocationRefStructure

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

Tabelle 39: 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

<i>ServiceViaPointStructure</i>			<i>+Structure</i>	<b>Via-Punkt auf dem Fahrweg.</b>
<i>Stop-Point</i>	<i>StopPointRef</i>	1:1	<i>→StopPoint</i>	Referenz auf einen Code für einen Haltepunkt. Vgl. 7.5.1.
	<i>StopPointName</i>	1:*	<i>+International Text</i>	Name des Haltepunkts für Fahrgastinformation.
	<i>NameSuffix</i>	0:*	<i>+International Text</i>	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „gegenüber vom Haupteingang“.
	<i>PlannedBay</i>	0:*	<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:*	<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>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 40: Beschreibung der Struktur **ServiceViaPointStructure**.

### 7.6.2. ViaStructure

<i>ViaStructure</i>			<i>+Structure</i>	<b>Angaben zu einer Via-Bedingung.</b>
	<i>ViaPoint</i>	1:1	<i>+LocationRef</i>	Referenz auf den Via-Punkt. Vgl. 7.5.11.
	<i>DwellTime</i>	0:*	<i>xs:duration</i>	Vom Benutzer vorgeschriebene Mindestaufenthaltszeit am Via-Punkt.

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

### 7.6.3. ServiceSectionStructure

<b>ServiceSectionStructure</b>			<b>+Structure</b>	<b>Eigenschaften einer Fahrt zusammen mit dem Fahrabschnitt, auf dem diese Eigenschaften gelten.</b>
Stop-SeqInterval	FromStopSeqNumber	0:1	xs:positiveInteger	Fahrtwegpositionsnummer des Haltepunkts, ab dem die Eigenschaften gültig sind. Falls leer, dann gültig ab Beginn des Fahrwegs.
	ToStopSeqNumber	0:1	xs:positiveInteger	Fahrtwegpositionsnummer des Haltepunkts, bis zu dem die Eigenschaften gültig sind. Falls leer, dann gültig bis zum Ende des Fahrwegs.
LineIdentity	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.
	PublishedLineName	1:*	+InternationalText	Liniennummer oder -name, wie in der Öffentlichkeit bekannt.
	OperatorRef	0:1	→Operator	Operator-ID. Vgl. 7.4.1.
	RouteDescription	0:*	+InternationalText	Beschreibung des Fahrwegs, z.B. „rechte Rheinstraße“.
	Via	0:*	+ServiceViaPoint	Wichtige Halte auf dem Fahrweg. Vgl. 7.6.1.

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

### 7.6.4. DatedServiceGroup

<b>DatedServiceGroup</b>			<b>+Group</b>	<b>Gruppe zur Beschreibung der Fahrt einer Linie an einem bestimmten Tag.</b>
	OperatingDayRef	1:1	→OperatingDay	Betriebstag der Fahrt. Vgl. 7.4.1.
	VehicleRef	0:1	→Vehicle	Fahrzeug-ID. Vgl. 7.4.1.
ServiceJourney	JourneyRef	1:1	→Journey	Fahrt-ID. Vgl. 7.4.1.
	ServiceSection	1:*	+ServiceSection	Fahrtabschnitte mit Eigenschaften. Vgl. 0.
	Attribute	0:*	+ServiceAttribute	Hinweise und Attribute (mit Klassifikationen) zur Fahrt. Vgl. 7.6.18.

Tabelle 43: Beschreibung der Gruppe **DatedServiceGroup**.

### 7.6.5. DatedJourneyStructure

<b>DatedJourneyStructure</b>			<b>+Structure</b>	<b>Fahrplanfahrt an bestimmtem Tag.</b>
	...	1:1	+DatedServiceGroup	Fahrt einer Linie am Stichtag (vgl. 7.6.4).
ServiceOrigin	OriginStopPointRef	0:1	→StopPoint	ID des ersten Haltepunkts der Fahrt; Starthaltestelle. Vgl. 7.5.1.
	OriginText	0:*	+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	1:*	+InternationalText	Name des letzten Haltepunkts der Fahrt, der Endhaltestelle oder Fahrtziel.
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> .

	<i>Deviation</i>	0:1	<i>xs:boolean</i>	Gibt an, ob diese Fahrt einen anderen Weg nimmt. Voreinstellung ist <i>false</i> .
	<i>Occupancy</i>	0:1	<i>manySeatsAvailable   fewSeatsAvailable   noSeatsAvailable   standingAvailable   full</i>	Auslastungszustand des Fahrzeugs.
	<i>SituationFullRef</i>	0:*	<i>+SituationFullRef</i>	Verweis auf eine Störungsnachricht. Diese Nachricht kann im Kontext der Antwort (Response context) zu finden sein oder auf anderem Wege bekannt gemacht werden. Vgl. 7.8.2.

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

### 7.6.6. ParallelServiceStructure

Verkehrt eine Fahrt auf einem Abschnitt gemeinsam mit einer anderen Fahrt (z.B. ICE zwischen Köln und Berlin, Trennung in Hamm), ist es hilfreich, den Fahrgast über die Gefahr zu informieren, in den falschen Zugteil einzusteigen.

<b>ParallelServiceStructure</b>			<i>+Structure</i>	<b>Enthält einen Abschnitt, auf dem eine weitere Fahrt gemeinsam fährt (z.B. bei Flügelungen) und die entsprechende Parallelfahrt.</b>
<i>StopSeqInterval</i>	<i>FromStopSeqNumber</i>	0:1	<i>xs:positiveInteger</i>	Fahrtwegpositionsnummer des Haltepunkts, ab dem die Parallelfahrt beginnt. Falls leer, dann gültig ab Beginn des Fahrwegs.
	<i>ToStopSeqNumber</i>	0:1	<i>xs:positiveInteger</i>	Fahrtwegpositionsnummer des Haltepunkts, an dem Parallelfahrt endet. Falls leer, dann gültig bis zum Ende des Fahrwegs.
	<b>Service</b>	<b>1:1</b>	<b><i>+DatedJourney</i></b>	Parallelfahrt. Vgl. 7.6.5.

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

### 7.6.7. TripLocationStructure

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

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

### 7.6.8. ServiceCallStructure

<b>ServiceCallStructure</b>			<i>+Structure</i>	<b>Enthält Informationen zu Ankunft oder Abfahrt einer Fahrt an einem Punkt (z.B. Zeiten).</b>
<i>ServiceTime</i>	<b><i>TimetabledTime</i></b>	<b>1:1</b>	<b><i>xs:dateTime</i></b>	Zeit nach Fahrplan.
	<i>RecordedAtTime</i>	0:1	<i>xs:dateTime</i>	Tatsächliche Zeit.
	<i>EstimatedTime</i>	0:1	<i>xs:dateTime</i>	Erwartete Zeit.
	<i>EstimatedTimeLow</i>	0:1	<i>xs:dateTime</i>	Untere Schranke für erwartete Zeit.
	<i>EstimatedTimeHigh</i>	0:1	<i>xs:dateTime</i>	Obere Schranke für erwartete Zeit.

Tabelle 47: Beschreibung der Struktur **ServiceCallStructure**.



## 7.6.9. CallAtStopStructure

<b>CallAtStopStructure</b>			<b>+Structure</b>	<b>Halt einer Fahrt an einem Haltepunkt oder Haltestelle.</b>
<i>Stop-Point</i>	<b>StopPointRef</b>	1:1	→ <i>StopPoint</i>	Referenz auf einen Code für einen Haltepunkt. Vgl. 7.5.1.
	<b>StopPointName</b>	1:*	+ <i>InternationalText</i>	Name des Haltepunkts für Fahrgastinformation.
	<i>NameSuffix</i>	0:*	+ <i>InternationalText</i>	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „gegenüber vom Haupteingang“.
	<i>PlannedBay</i>	0:*	+ <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:*	+ <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>ServiceArrival</i>	0:1	+ <i>ServiceCall</i>	Informationen zur Ankunft. Vgl. 7.6.8.
	<i>ServiceDeparture</i>	0:1	+ <i>ServiceCall</i>	Informationen zur Abfahrt. Vgl. 7.6.8.
<i>Stop-CallStatus</i>	<i>StopSeqNumber</i>	0:1	<i>xs:positiveInteger</i>	Laufende Nummer des Halts im Fahrweg der Fahrt. Gezählt ab der Starthaltestelle der Fahrt (als Nummer 1).
	<i>DemandStop</i>	0:1	<i>xs:boolean</i>	Bedarfshalt. Fahrzeug bedient diesen Halt nur nach Voranmeldung. Default ist <i>false</i> .
	<i>UnplannedStop</i>	0:1	<i>xs:boolean</i>	Halt, der laut Planung nicht vorgesehen war. Default ist <i>false</i> .
	<i>NotServicedStop</i>	0:1	<i>xs:boolean</i>	Entgegen der Planung findet kein Halt statt. Default ist <i>false</i> .
	<i>NoBoardingAtStop</i>	0:1	<i>xs:boolean</i>	An diesem Halt der Fahrt darf nicht eingestiegen werden. Default ist <i>false</i> .
	<i>NoAlightingAtStop</i>	0:1	<i>xs:boolean</i>	An diesem Halt der Fahrt darf nicht ausgestiegen werden. Default ist <i>false</i> .
	<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 48: Beschreibung der Struktur **CallAtStopStructure**.

## 7.6.10. DatedCallAtLocationStructure

<b>DatedCallAtLocationStructure</b>			<b>+Structure</b>	<b>Bedienung eines allgemeinen Ortspunkts durch eine Fahrt an einem bestimmten Tag</b>
<i>DatedJourneyRef</i>	<b>JourneyRef</b>	1:1	→ <i>Journey</i>	Fahrt-ID. Vgl. 7.4.1.
	<b>OperatingDayRef</b>	1:1	→ <i>OperatingDay</i>	Betriebstag der Fahrt. Vgl. 7.4.1.
<i>LineDirection</i>	<b>LineRef</b>	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.
	<i>OperatorRef</i>	0:1	→ <i>Operator</i>	Operator-ID. Vgl. 7.4.1.
	<b>CallLocation</b>	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. 7.5.11.
	<i>ServiceArrival</i>	0:1	+ <i>ServiceCall</i>	Informationen zur Ankunft. Vgl. 7.6.8.
	<i>ServiceDeparture</i>	0:1	+ <i>ServiceCall</i>	Informationen zur Abfahrt. Vgl. 7.6.8.
<i>StopCallStatus</i>	<i>StopSeqNumber</i>	0:1	<i>xs:positiveInteger</i>	Laufende Nummer des Halts im Fahrweg der Fahrt. Gezählt ab der Starthaltestelle der Fahrt (als Nummer 1).
	<i>DemandStop</i>	0:1	<i>xs:boolean</i>	Bedarfshalt. Fahrzeug bedient diesen Halt nur nach Voranmeldung. Default ist <i>false</i> .
	<i>UnplannedStop</i>	0:1	<i>xs:boolean</i>	Halt, der laut Planung nicht vorgesehen war. Default ist <i>false</i> .
	<i>NotServicedStop</i>	0:1	<i>xs:boolean</i>	Entgegen der Planung findet kein Halt statt. Default ist <i>false</i> .
	<i>NoBoardingAtStop</i>	0:1	<i>xs:boolean</i>	An diesem Halt der Fahrt darf nicht eingestiegen werden. Default ist <i>false</i> .
	<i>NoAlightingAtStop</i>	0:1	<i>xs:boolean</i>	An diesem Halt der Fahrt darf nicht ausgestiegen werden. Default ist <i>false</i> .

Tabelle 49: Beschreibung der Struktur **DatedCallAtLocationStructure**.

### 7.6.11. ContinuousServiceStructure

<b>ContinuousServiceStructure</b>			<b>+Structure</b>	<b>Eine Fahrgastbewegung mit Hilfe eines kontinuierlichen, nicht fahrplangebundenen Verkehrsmittels.</b>	
	a	<b>ContinuousMode</b>	-1:1	walk   demandResponsive   replacementService	Modalität für kontinuierliche Verkehre.
	b	<b>IndividualMode</b>		walk   cycle   taxi   self-drive-car   others-drive-car   motorcycle   truck	Verkehrsmittelmodalität für Individualverkehr.
	a	...	-0:1	+DatedServiceGroup	Beschreibung einer ÖV-Fahrmöglichkeit am Stichtag (vgl. 7.6.4).
	b	SharingService		+SharingService	Beschreibung eines Mobilitätsangebots mit Ausleihfahrzeugen (vgl. 7.4.7).
ServiceOrigin		OriginStopPointRef	0:1	→StopPoint	ID des ersten Haltepunkts der Fahrt; Starthaltestelle. Vgl. 7.5.1.
		OriginText	0:*	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:*	InternationalText	Name des letzten Haltepunkts der Fahrt, der Endhaltestelle oder Fahrtziel.
		SituationFullRef	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 50: Beschreibung der Struktur **ContinuousServiceStructure**.

### 7.6.12. VehiclePositionStructure

<b>VehiclePositionStructure</b>			<b>+Structure</b>	<b>Geografische und logische Position eines Fahrzeugs.</b>	
		GeoPosition	0:1	+GeoPosition	Geografische Position (vgl. 7.2.3).
		Progress	0:1	Not yet operated   Operation finished   At stop   Between stops	Logische Position bezogen auf Haltestellenfolge im Fahrplan.
		Bearing	0:1	AbsoluteBearing	Kompassrichtung in Grad, in die sich das Fahrzeug bewegt (vgl. 7.2.1).
		ProgressBetweenStops	0:1	+ProgressBetweenStops	Position zwischen der zuletzt besuchten Haltestelle und der aktuellen Position (vgl. 7.6.13).

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

### 7.6.13. ProgressBetweenStopsStructure

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

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

## 7.6.14. LegTrackStructure

<b>LegTrackStructure</b>			+Structure	Container für die Streckenabschnitte entlang eines Verbindungsteilwegs.
	<b>TrackSection</b>	1:*	+TrackSection	Ein oder mehrere Streckenabschnitte. Vgl. 7.6.15.

Tabelle 53: Beschreibung der Struktur **LegTrackStructure**.

## 7.6.15. TrackSectionStructure

<b>TrackSectionStructure</b>			+Structure	Ein Streckenabschnitt in einem Verbindungsteilweg.
	<i>TrackStart</i>	0:1	+LocationRef	Beginn (Ort) des Streckenabschnitts. Vgl. 7.5.11.
	<i>TrackEnd</i>	0:1	+LocationRef	Ende (Ort) des Streckenabschnitts. Vgl. 7.5.11.
<i>Projection</i>	<b>Position</b>	2:*	+GeoPosition	Geografische Projektion des Streckenabschnitts als Polygonzug. Vgl. 7.2.3.
	<i>RoadName</i>	0:1	xs:string	Name der Straße, auf der dieser Streckenabschnitt liegt.
	<i>Duration</i>	0:1	xs:duration	Zeitdauer, die der Fahrgast braucht, um diesen Streckenabschnitt zu bewältigen.
	<i>Length</i>	0:1	LengthType	Länge des Streckenabschnitts.
	<i>Extension</i>	0:1	xs:anyType	Erweiterungen.

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

## 7.6.16. LocationContextStructure

<b>LocationContextStructure</b>			+Structure	Angabe eines Orts und der Möglichkeiten, wie ein Benutzer ihn per IV erreichen kann.	
	a	<b>LocationRef</b>	-1:1	+LocationRef	Angabe eines räumlichen Orts (vgl. 7.5.11).
	b	<b>TripLocation</b>		+TripLocation	Aufenthaltort in einem (sich bewegenden) Fahrzeug (vgl. 7.6.6).
	<i>DepArrTime</i>		0:1	xs:dateTime	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 55: 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.

## 7.6.17. AbstractResponseContextStructure

<b>AbstractResponseContextStructure</b>			<b>+Structure</b>	<b>Basisstruktur für Antwortkontext. Hier können Objekte abgelegt werden, die mehrfach auftreten und durch Referenzen auf den Kontext ersetzt werden können.</b>
<i>Locations</i>	<i>Location</i>	0:*	<b>+Location</b>	Modellierung von Ortspunkten (vgl. 7.5.10).
<i>Situations</i>	<i>Situation</i>	0:*	<b>+siri:PtSituationElement</b>	SIRI-Modellierung eines Ereignisses oder einer Störung (vgl. 7.8.1).

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

## 7.6.18. ServiceAttributeStructure

<b>ServiceAttributeStructure</b>			<b>+Structure (abgeleitet von GeneralAttributeStructure, vgl. 7.4.10)</b>	<b>Definition von Attributen und Hinweisen, die nur auf Teilen einer Verbindung gültig sind.</b>
	<i>Scope</i>	0:1	<i>onRide   atStop   atBoardOnly   atAlightOnly   atBoardAndAlight</i>	Definiert, für was ein Attribut oder Hinweis gültig ist. Ein Attribut kann gültig sein während der Fahrt (z.B. „Rollstuhlstellplatz“), an jedem Halt (z.B. „Einstiegs- hilfe vorhanden“), nur an Einstiegshalten (z.B. „Mit- fahrt nur bei Anmeldung“), nur an Ausstiegshalten (z.B. „Hält nur nach Knopfdruck“) oder an Einstiegs- und Ausstiegshalten (z.B. „Vorsicht Stufe“).
<i>Stop- SeqIn- terval</i>	<i>FromStopSeqNumber</i>	0:1	<i>xs:positiveInteger</i>	Fahrwegpositionsnummer des Haltepunkts, ab dem das Attribut gültig ist. Falls leer, dann gültig ab Be- ginn des Fahrwegs.
	<i>ToStopSeqNumber</i>	0:1	<i>xs:positiveInteger</i>	Fahrwegpositionsnummer des Haltepunkts, bis zu dem das Attribut gültig ist. Falls leer, dann gültig bis zum Ende des Fahrwegs.

Tabelle 57: Beschreibung der Struktur **ServiceAttributeStructure**.

## 7.6.19. PassengerAccessibilityStructure

<b>PassengerAccessibilityStructure</b>			<b>+Structure</b>	<b>Struktur für die Definition besonderer Bedürfnisse und Einschränkungen von Fahrgästen.</b>
<i>Base- TripMo- bilityFil- ter</i>	<i>NoSingleStep</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer keine Stufe bewältigen kann. Default ist <i>false</i> .
	<i>NoStairs</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer keine Treppe bewältigen kann. Default ist <i>false</i> .
	<i>NoEscalator</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer keine Rolltreppe benutzen kann. Default ist <i>false</i> .
	<i>NoElevator</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer keinen Aufzug benutzen kann. Default ist <i>false</i> .
	<i>NoRamp</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer keine Rampe bewältigen kann. Default ist <i>false</i> .
<i>TripMo- bilityFil- ter</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 auf <i>true</i> gesetzt. Default ist <i>false</i> .
	<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 auf <i>true</i> gesetzt. Default ist <i>false</i> .
	<i>WalkSpeed</i>	0:1	<i>OpenPercent</i>	Veränderung der Standardgehgeschwindigkeit in Prozent. Der Wert 100 stellt den Standard dar. Wer-

				te kleiner 100 stellen eine langsamere Geschwindigkeit dar, Werte größer 100 eine schnellere.
As- sistance	<i>BoardingAssistance</i>	0:1	<i>xs:boolean</i>	Gibt an, ob beim Einsteigen Hilfe durch das Fahr- oder Stationspersonal benötigt wird. Default ist <i>false</i> .
	<i>AlightingAssistance</i>	0:1	<i>xs:boolean</i>	Gibt an, ob beim Aussteigen Hilfe durch das Fahr- oder Stationspersonal benötigt wird. Default ist <i>false</i> .
Pas- senger- Profile	<i>WheelchairUser</i>	0:1	<i>xs:boolean</i>	Fahrgast nutzt einen Rollstuhl. Default ist <i>false</i> .
	<i>WalkingFrame</i>	0:1	<i>xs:boolean</i>	Fahrgast nutzt einen Rollator. Default ist <i>false</i> .
	<i>WalkingStick</i>	0:1	<i>xs:boolean</i>	Fahrgast nutzt einen Gehstock. Default ist <i>false</i> .
	<i>WalkingImpaired</i>	0:1	<i>xs:boolean</i>	Fahrgast ist gehbehindert. Default ist <i>false</i> .
	<i>Pram</i>	0:1	<i>xs:boolean</i>	Fahrgast führt einen Kinderwagen mit sich. Default ist <i>false</i> .
	<i>HeavyLuggage</i>	0:1	<i>xs:boolean</i>	Fahrgast führt schweres Gepäck mit sich. Default ist <i>false</i> .
	<i>VisuallyImpaired</i>	0:1	<i>xs:boolean</i>	Fahrgast ist sehbeeinträchtigt. Default ist <i>false</i> .
	<i>HearingImpaired</i>	0:1	<i>xs:boolean</i>	Fahrgast ist hörbeeinträchtigt. Default ist <i>false</i> .
	<i>ReadingImpaired</i>	0:1	<i>xs:boolean</i>	Fahrgast hat Schwierigkeiten mit dem Leseverständnis. Default ist <i>false</i> .

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

## 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.

<i>siri:CommonFacilityGroup</i>		+Group	Klassifizierung von allgemeinen Ausstattungsmerkmalen und Einrichtungen (nach TPEG <i>pti_table 23</i> ).
<i>FareClassFacility</i>	0:*	<i>unknown</i>   <i>firstClass</i>   <i>secondClass</i>   <i>thirdClass</i>   <i>economyClass</i>   <i>businessClass</i>	Tarifklassen.
<i>TicketingFacility</i>	0:*	<i>unknown</i>   <i>ticketMachines</i>   <i>ticketOffice</i>   <i>ticketOnDemandMachines</i>   <i>ticketSales</i>   <i>mobileTicketing</i>   <i>ticketCollec-</i>	Einrichtungen zum Fahrscheinkauf.

			<i>tion   centralReservations   localTickets   nationalTickets   internationalTickets</i>	
	<i>NuisanceFacility</i>	0:*	<i>unknown   smoking   noSmoking   mobile-PhoneUse-Zone   mobilePhone-FreeZone</i>	Aufenthaltsbereiche.
	<i>MobilityFacility</i>	0:*	<i>unknown   suitableFor-WheelChairs   lowFloor   boardingAssistance   stepFreeAccess   tactile-PlatformEdges   onboardAssistance   unaccompaniedMinorAssistance   audioInformation   visualInformation   displaysForVisuallyImpaired   audio-ForHearingImpaired</i>	Ausstattungsmerkmale für Mobilitätseingeschränkte.
	<i>PassengerInformation-Facility</i>	0:*	<i>unknown   nextStopIndicator   stopAnnouncements   passengerInformationDisplay   audioInformation   visualInformation   tactilePlatformEdges   tactileInformation   walkingGuidance   journeyPlanning   lost-Found   informationDesk   interactiveKiosk-Display   printedPublicNotice</i>	Einrichtungen zur Fahrgastinformation.
	<i>PassengerCommsFacility</i>	0:*	<i>unknown   faccomms_1   passengerWifi   telephone   audioServices   videoServices   businessServices   internet   postoffice   letterbox</i>	Kommunikationseinrichtungen für Fahrgäste.

	<i>RefreshmentFacility</i>	0:*	<i>unknown   restaurantService   snacksService   trolley   bar   foodNotAvailable   beveragesNotAvailable   bistro   foodVendingMachine   beverageVendingMachine</i>	Versorgung mit Erfrischungen, Speisen und Getränken.
	<i>AccessFacility</i>	0:*	<i>unknown   lift   escalator   travelator   ramp   stairs   shuttle   narrowEntrance   barrier   palletAccess_lowFloor   validator</i>	Zugangsmerkmale zu Haltestellen oder Fahrzeugen.
	<i>SanitaryFacility</i>	0:*	<i>unknown   toilet   noToilet   shower   wheelchairAccessToilet   babyChange</i>	Sanitäre Einrichtungen.
	<i>LuggageFacility</i>	0:*	<i>unknown   bikeCarriage   baggageStorage   leftLuggage   portage   baggageTrolleys</i>	Einrichtungen zur Gepäckbeförderung oder -aufbewahrung.

Tabelle 59: 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.

<b>siri:StopFacilityGroup</b>		<b>+Group</b>	<b>Klassifizierung von Ausstattungsmerkmalen und Einrichtungen an Haltestellen (nach TPEG pti_table 23).</b>	
<i>CommonFacilityGroup</i>	...	0:*	<i>siri:CommonFacilityGroup</i>	Allgemeine Ausstattungsmerkmale. Vgl. 7.7.1.
	<i>AssistanceFacility</i>	0:*	<i>unknown   police   firstAid   sosPoint   specificAssistance   unaccompaniedMinorAssistance   boardingAssistance</i>	Einrichtungen für Hilfesuchende.
	<i>HireFacility</i>	0:*	<i>unknown   carHire   motorCycleHire   cycleHire   taxi   recreation-</i>	Vermietungs- und Leihangebote.



			<i>DeviceHire</i>	
--	--	--	-------------------	--

Tabelle 60: 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.

<b>siri:ServiceFacilityGroup</b>			<b>+Group</b>	<b>Klassifizierung von Ausstattungsmerkmalen und Einrichtungen in Fahrzeugen (nach TPEG <i>pti_table 23</i>).</b>
<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 61: 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.

<i>siri:AllFacilitiesGroup</i>			+Group	Umfassende Gruppe mit allen Klassifizierungen von Ausstattungsmerkmalen und Einrichtungen (nach TPEG <i>pti_table 23</i> ).
<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 62: 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

<i>SituationsStructure</i>			+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, 2011).
	<i>RoadSituation</i>	0:*	<i>siri:RoadSituationElement</i>	Kapselung der SIRI-Strukturdefinition für IV-Ereignisse, vgl. (CEN, TS 15531 Part 5, 2011).

Tabelle 63: Beschreibung der Struktur *SituationsStructure*.

## 7.8.2. SituationFullRefStructure

<b>SituationFullRefStructure</b>			<i>+Structure (abgeleitet von si- ri:SituationFull RefStructure)</i>	<b>Referenz auf eine Situationsbeschreibung.</b>
<i>SituationFullIdentity</i>	<i>VersionCountryRef</i>	0:1	<i>ifopt:CountryRef</i>	Referenziert das Land, um ggf. die <i>ParticipantRef</i> eindeutig zu machen.
	<b>ParticipantRef</b>	<b>1:1</b>	<i>ParticipantRef</i>	Eindeutige ID des Schnittstellenpartners (vgl. 5.12). Stellt Namensraum für die ID der Situation bereit.
	<b>SituationNumber</b>	<b>1:1</b>	<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 64: 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:

Typname	Basistyp	Beschreibung
DataVersionType	xs:NMTOKEN	Datentyp zur Angabe der Datenversion.
CalcTimeType	xs:integer	Datentyp für die Rechenzeit in Millisekunden.
SignatureType	xs:string	Datentyp für Signaturen.
CertificateIdType	xs:NMTOKEN	Datentyp für Zertifikat-IDs.

Tabelle 65: 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)
si-ri:ContextualisedRequest	ServiceRequestContext	0:1	+si-ri:ServiceRequestContext	Allgemeine Nachrichteneigenschaften, die üblicherweise durch die Konfiguration bekannt sind und nicht je Anfrage ausgetauscht werden müssen. Siehe auch (CEN, TS 15531 Part 2, 2011), Kap.6.1.2.
	<b>RequestTimestamp</b>	<b>1:1</b>	xs:dateTime	Zeitstempel der Anfrage.
RequestorEndpoint	Address	0:1	si-ri:EndpointAddress	Adresse, an die die Antwort gesendet werden soll. Kann auch mittels RequestorRef aus der Konfiguration ermittelt werden.
	<b>RequestorRef</b>	<b>1:1</b>	→si-ri:ParticipantCode	ID des Anfragers.
	MessageIdentifier	0:1	si-ri:MessageQualifier	Beliebige, eindeutige ID, mit der diese Nachricht referenziert werden kann.
ServiceRequestContext	DataVersion	0:1	DataVersion	Datenversion, die vom Server bei der Bearbeitung benutzt werden soll.
	Language	0:*	xs:language	Bevorzugte Sprachen, in der Texte in der Antwort zurückgegeben werden sollen.
MessageIntegrityProperties	Signature	0:1	Signature	Signatur der Nachricht.
	CertificateId	0:1	CertificateId	Zertifikat-ID für die Überprüfung der Nachricht.
ServiceRequestContext	Extension	0:1	xs:anyType	Erweiterungen.

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

### 7.9.3. AbstractTriasSubscriptionRequestStructure

<b>AbstractTriasSubscriptionRequestStructure</b>			<b>+Structure</b>	<b>Basisstruktur für alle Anfragen zur Einrichtung eines Abonnements.</b>
<i>si-ri:AbstractSubscriptionRequest</i>	<b>RequestTimestamp</b>	1:1	<i>xs:dateTime</i>	Zeitstempel der Anfrage.
<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.
	<b>RequestorRef</b>	1:1	<i>→si-ri:ParticipantCode</i>	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 Abommeneigenschaften, die üblicherweise durch die Konfiguration bekannt sind und nicht explizit angegeben werden müssen. Siehe auch (CEN, TS 15531 Part 2, 2011), 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:*	<i>xs:language</i>	Bevorzugte Sprachen, 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 67: Beschreibung der Struktur **AbstractTriasSubscriptionRequestStructure**.

## 7.9.4. AbstractTriasResponseStructure

<b>AbstractTriasResponseStructure</b>		<b>+Structure</b>		<b>Basisstruktur für alle Antworten.</b>
<i>si-ri:ProducerResponse</i>	<b>RequestTimestamp</b>	1:1	<i>xs:dateTime</i>	Zeitstempel der Antwort.
<i>si-ri:ProducerResponseEndpoint</i>	<i>ProducerRef</i>	0:1	<i>→si-ri:ParticipantCode</i>	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	<i>→si-ri:MessageQualifier</i>	Referenz auf die Anfragenachricht, die diese Antwortnachricht ausgelöst hat.
<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>si-ri:ErrorCondition</i>	SIRI-Fehlerzustände, die die Bearbeitung der Anfrage als Ganzes betreffen. Siehe auch (CEN, TS 15531 Part 2, 2011), 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.
	<b>Language</b>	1:1	<i>xs:language</i>	Standardsprache, in der Text in der Antwort zurückgegeben wird, falls nicht pro Element anders angegeben (vgl. 7.2.2).
	<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 68: 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:

Typname	Basistyp	Beschreibung
<i>FaresAuthorityCodeType</i>	<i>xs:NMTOKEN</i>	Code für ein Tarifgebiet oder einen Unternehmens- tarif, 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 Tarif- gebiets oder eines Unternehmenstarifs.
<i>TravellerCardCodeType</i>	<i>xs:NMTOKEN</i>	Code für eine Vielfahrerkarte, z. B. „Bahn- Card50“ 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 Hin- sicht.

Tabelle 69: 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. FareZoneStructure

<i>FareZoneStructure</i>			+Structure	Modell einer Tarifzone mit öffentlich bekannter Bezeichnung.
	<i>FareZoneRef</i>	1:1	→ <i>FareZoneCode</i>	Code für eine Tarifzone (vgl. 7.10.1).
	<i>FareZoneText</i>	1:1	<i>xs:string</i>	Bezeichnung der Tarifzone für die Fahrgäste.

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

### 7.10.3. FareZoneListInAreaStructure

<i>FareZoneListInAreaStructure</i>			+Structure	Liste von Tarifzonen bezogen auf ein Tarifgebiet.
<i>FaresAu- thority</i>	<i>FaresAuthorityRef</i>	1:1	→ <i>FaresAuthori- tyCode</i>	Code für ein Tarifgebiet oder einen Unternehmens- tarif (vgl. 7.10.1).
	<i>FaresAuthorityText</i>	1:1	<i>xs:string</i>	Beschreibung oder Name des Tarifgebiets.
	<i>FareZone</i>	1:*	+ <i>FareZone</i>	Eine oder mehrere Tarifzonen (Vgl. 7.10.2).

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

## 7.10.4. BookingInfoStructure

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

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

## 7.10.5. TicketStructure

<b>TicketStructure</b>			<b>+Structure</b>	<b>Modellierung eines Fahrscheins und zugehöriger Informationen.</b>
	<b>TicketId</b>	1:1	<i>→TicketCode</i>	Eindeutige Ticket-ID (vgl. 7.10.1).
	<b>TicketName</b>	1:1	<i>xs:string</i>	Name/Bezeichnung des Tickets.
<i>FaresAuthority</i>	<b>FaresAuthorityRef</b>	1:1	<i>→FaresAuthorityCode</i>	Code für ein Tarifgebiet oder einen Unternehmens-tarif (vgl. 7.10.1).
	<b>FaresAuthorityText</b>	1:1	<i>xs:string</i>	Beschreibung oder Name des Tarifgebiets.
<i>Ticket-Price</i>	<i>Price</i>	0:1	<i>xs:decimal</i>	Ticketpreis als Dezimalzahl.
	<i>NetPrice</i>	0:1	<i>xs:decimal</i>	Netto-Ticketpreis als Dezimalzahl für Abrechnungszwecke.
	<i>Currency</i>	0:1	<i>xs:NMTOKEN</i>	Währungscode nach ISO 4217, z. B. „EUR“ oder „GBP“.
	<i>VatRate</i>	0:1	<i>VatRateEnumeration</i>	Mehrwertsteuersatz (vgl. 7.10.1). Voreinstellung ist <i>unknown</i> .
<i>Tariff-Level</i>	<i>TariffLevel</i>	0:1	<i>xs:string</i>	Tarifstufe (Beispiel aus Nürnberg „10“ oder „10+T“)
	<i>TariffLevelLabel</i>	0:*	<i>+InternationalText</i>	Bezeichnung für Tarifstufen in diesem Zusammenhang (Beispiel aus Nürnberg „Preisstufe“, „Tarifstufe“)
<i>Ticket-Validity</i>	<i>TravelClass</i>	0:1	<i>TravelClassEnumeration</i>	Reiseklasse, für die das Ticket gültig ist (vgl. 7.10.1).
	<i>RequiredCard</i>	0:*	<i>→TravellerCardCode</i>	Eine oder mehrere Vielfahrerkarten, die nötig sind, um dieses Ticket erwerben oder nutzen zu dürfen (vgl. 7.10.1).
	<i>ValidFor</i>	0:*	<i>PassengerCategoryEnumeration</i>	Personengruppen, die dieses Ticket nutzen dürfen (vgl. 7.10.1).
	<i>ValidityDuration</i>	0:1	<i>xs:duration</i>	Maximale zeitliche Gültigkeit des Tickets ab Kauf oder Entwertung.
	<i>ValidityDurationText</i>	0:*	<i>+InternationalText</i>	Beschreibung der zeitlichen Gültigkeit.
	<i>ValidityFareZones</i>	0:1	<i>+FareZoneListInArea</i>	Räumliche Gültigkeit des Tickets ausgedrückt durch eine Liste der Tarifzonen, für die das Ticket gilt.
	<i>ValidityAreaText</i>	0:*	<i>+InternationalText</i>	Beschreibung der räumlichen Gültigkeit.
<i>Ticket-Booking</i>	<i>InfoUrl</i>	0:*	<i>+WebLink</i>	URL zu Informationsseiten für dieses Ticket (vgl. 7.2.4).
	<i>SaleUrl</i>	0:*	<i>+WebLink</i>	URL zu Online-Kaufmöglichkeiten, um dieses Ticket zu erwerben (vgl. 7.2.4).
	<i>BookingInfo</i>	0:*	<i>+BookingInfo</i>	Beschreibung der Buchungsmöglichkeiten (vgl.



				7.10.4).
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

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

### 7.10.6. TripFaresResultStructure

<b>TripFaresResultStructure</b>			<b>+Structure</b>	<b>Fasst die Ergebnisdaten für die Tarifauskunft zu einer Verbindung (oder zu Teilen einer Verbindung) zusammen.</b>
	<i>ErrorMessage</i>	0:*	<b>+ErrorMessage</b>	Fehlermeldungen bezogen auf diese Tarifauskunft. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. auch 7.4.2.
<i>TripLegRange</i>	<i>FromTripLegIdRef</i>	0:1	<i>xs:NMTOKEN</i>	Referenz auf einen Teilweg der Verbindung als Beginn der Gültigkeit dieser Tarifauskunft.
	<i>ToTripLegIdRef</i>	0:1	<i>xs:NMTOKEN</i>	Referenz auf einen Teilweg der Verbindung als Ende der Gültigkeit dieser Tarifauskunft.
	<i>PassedZones</i>	0:1	<b>+FareZoneListInArea</b>	Die durchfahrenen Tarifzonen auf diesem Abschnitt der Verbindung (vgl. 7.10.3).
	<i>Ticket</i>	0:*	<b>+Ticket</b>	Fahrscheine, die auf diesem Abschnitt der Verbindung gültig sind (vgl. 7.10.5).
	<i>StaticInfoURL</i>	0:*	<b>+WebLink</b>	URL zu Informationsseiten (vgl. 7.2.4).

Tabelle 74: Beschreibung der Struktur **TripFaresResultStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<b>Fehlercode</b>	<b>Fehlerbeschreibung</b>
<b>FARES_OUTOFAREA</b>	Der in der Verbindungsauskunft gefundene Weg verlässt das Tarifgebiet.
<b>FARES_JOURNEYNOTPERMITTED</b>	Ein in der Verbindungsauskunft benutztes Verkehrsmittel ist für den Tarif nicht zulässig.
<b>FARES_ADDITIONALCHARGES</b>	Zusätzliche Gebühren sind voraussichtlich zu entrichten (z. B. Mautzuschläge oder Reservierungsgebühren).
<b>FARES_ADDITIONALTICKETS</b>	Zusätzliche Fahrscheine sind notwendig, da nicht für alle Verkehrsmittel ein passender Fahrschein ermittelt werden konnte.
<b>FARES_ROUTENOTFEASIBLE</b>	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).
<b>FARES_ALREADYCOVERED</b>	Diese Verbindung (oder der in <i>TripLegRange</i> angegebene Teil davon) kann mit der Fahrberechtigung benutzt werden, die in der Anfrage mitgegeben wurde.

Tabelle 75: Liste der Fehlerzustände in **TripFaresResult**.

### 7.10.7. FaresPassengerStructure

<b>FaresPassengerStructure</b>			<b>+Structure</b>	<b>Profil eines Reisenden für die Tarifiermittlung.</b>	
	a	<b>Age</b>	-1:1	<i>xs:nonNegativeInteger</i>	Alter des Reisenden.
	b	<b>PassengerCategory</b>		<i>PassengerCategoryEnumeration</i>	Personengruppe, die der Reisende zugerechnet werden kann (vgl. 7.10.1).
		<i>TravellerCard</i>	0:*	<i>→TravellerCardCode</i>	Eine oder mehrere Vielfahrerkarten, die der Reisende erworben hat und verwenden kann (vgl. 7.10.1).
	a	<i>ZonesAlreadyPaid</i>	-0:1	<i>+FareZoneListInArea</i>	Liste von Tarifzonen, für die der Fahrgast bereits ein gültiges Ticket hat (vgl. 7.10.3).
	b	<i>OwnedTicket</i>		-0:*	<i>→TicketCode</i>

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

Durch die Angabe der Elemente *ZonesAlreadyPaid* bzw. *OwnedTicket* kann zum Ausdruck gebracht werden, dass der Fahrgast bereits Fahrberechtigungen z.B. in Form von Fahrscheinen (wie etwa einer Monatskarte oder eines Jobtickets) besitzt. Der Server soll dann versuchen festzustellen, ob diese Berechtigungen bereits ausreichen, um auch diese Verbindung benutzen zu dürfen, oder im anderen Fall Tickets für die verbleibenden Teile empfehlen, für die die vorhandenen Berechtigungen nicht ausreichen. Für die Teile der Verbindung, für die kein weiterer Fahrschein notwendig ist, wird in *TripFaresResult* (vgl. 7.10.6) der Fehlerzustand *FARES\_ALREADYCOVERED* (aus Tabelle 75) angegeben.

### 7.10.8. FaresParamStructure

<b>FaresParamStructure</b>			<b>+Structure</b>	<b>Parameter für die Tarifiermittlung.</b>
<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. 7.10.7).

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

## 8. Ortsinformationsdienst

### 8.1. Beschreibung

Der Ortsinformationsdienst umfasst vier Funktionalitäten, die in der VDV-Schrift 431-1 als getrennte Dienste beschrieben werden

- 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 werden durch Abstraktion in einem einzigen Dienst gebündelt. Dadurch entstehen auch weitere Einsatzmöglichkeiten des Dienstes.

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:

<b>Typname</b>	<b>Basistyp</b>	<b>Beschreibung</b>
<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 78: 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

<b>LocationInformationRequestStructure</b>			<b>+Structure</b>	<b>Fasst die Daten Ortsobjektanfrage zusammen.</b>	
	<b>a</b>	<b>InitialInput</b>	-1:1	+InitialLocationInput	Eingabedaten für eine initiale Ortsinformationsanfrage. Vgl. 8.3.2.
	<b>b</b>	<b>LocationRef</b>		+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, vgl. <i>Complete</i> in Kapitel 0). 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. 7.5.11.
		<i>Restrictions</i>	0:1	+LocationParameter	Weitere Anfrageparameter. Vgl. 8.3.7.
		<i>Extension</i>	0:1	xs:anyType	Erweiterungen.

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

#### 8.3.2. InitialLocationInputStructure

<b>InitialLocationInputStructure</b>			<b>+Structure</b>	<b>Fasst die Anfrageparameter zusammen, die eine initiale Suche nach Ortsobjekten benötigt.</b>	
		<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>GeoPosition</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 80: Beschreibung der Struktur **InitialLocationInputStructure**.

#### 8.3.3. GeoRestrictionsStructure

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

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

### 8.3.4. GeoCircleStructure

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

Tabelle 82: Beschreibung der Struktur **GeoCircleStructure**.

### 8.3.5. GeoRectangleStructure

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

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

### 8.3.6. GeoAreaStructure

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

Tabelle 84: Beschreibung der Struktur **GeoAreaStructure**

### 8.3.7. LocationParamStructure

<b>LocationParamStructure</b>			<b>+Structure</b>	<b>Fasst Anfrageparameter zusammen, die im Ortsinformationsdienst verwendet werden.</b>
<i>LocationDataFilter</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	<i>+PtModeFilter</i>	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>OperatorFilter</i>	0:1	<i>+OperatorFilter</i>	Die Suche wird auf solche Ortsobjekte eingeschränkt, die von bestimmten Unternehmen betrieben/nicht betrieben werden (vgl. 7.4.4).
	<i>LocalityRef</i>	0:*	<i>→LocalityCode</i>	Erlaubte Lokalitäten. Falls angegeben, dürfen nur solche Ortsobjekte zurückgegeben werden, die mindestens einer der gegebenen Lokalitäten zugeordnet sind. Vgl. 7.5.1.
	<i>PointOfInterestFil-</i>	0:1	<i>+PointOfIn</i>	Ermöglicht, eine POI-Suche auf be-

	<i>ter</i>		<i>terestFilter</i>	stimmte POI-Kategorien einzuschränken (vgl. 7.5.6).
<i>LocationPolicy</i>				
	<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 Ortsobjekten 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 Dienstantwort 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 85: Beschreibung der Struktur ***LocationParamStructure***.

## 8.4. Antwortstrukturen

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

### 8.4.1. LocationInformationResponseStructure

<b>LocationInformationResponseStructure</b>		<b>+Structure</b>	<b>Fasst die Ergebnisdaten für eine Ortsinformationsanfrage zusammen.</b>
<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:*	<i>+ErrorMessage</i>	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
<i>LocationResult</i>	0:*	<i>+LocationResult</i>	Gefundene Ortsobjektergebnisse. Die Ortsobjekte müssen nach dem Übereinstimmungsgrad mit den Eingabedaten sortiert sein, d.h. das erste ist das am besten passende Objekt. Vgl. 0.

Tabelle 86: Beschreibung der Struktur **LocationInformationResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

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

Tabelle 87: Liste der Fehlerzustände in **LocationInformationResponse**.

## 8.4.2. LocationResultStructure

<i>LocationResultStructure</i>			+Structure	Resultatstruktur für ein Ortsobjekt.
	<i>Location</i>	1:1	+Location	Eigentliches Ortsobjekt. Vgl. 7.5.10.
	<i>Complete</i>	1:1	xs:boolean	Gibt an, ob das Ortsobjekt schon vollständig ausdifferenziert ist oder ob es noch verfeinert werden muss, damit es z.B. für einen TripRequest verwendet werden kann. Nicht vollständige Ortsobjekte müssen durch einen erneuten LocationInformationRequest ausdifferenziert werden. (Vgl. LocationRef in Kapitel 8.3.1)
	<i>Probability</i>	0:1	xs:float	Wahrscheinlichkeit, dass dieses Ortsobjekt dem gesuchten entspricht. Wird mit einem Wert zwischen 0 und 1 angegeben.
	<i>Mode</i>	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 88: 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.

In der XML-Schema-Definition *Trias\_Trips.xsd* werden Datentypen und Strukturen definiert, die für den Dienst Verbindungsauskunft verwendet werden.

### 9.2. Anfragestrukturen

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

#### 9.2.1. TripRequestStructure

<i>TripRequestStructure</i>			+Structure	Fasst die Anfragedaten für eine Verbindungsauskunft zusammen.
	<i>Origin</i>	1:*	+LocationContent	Ortsdaten für den Abfahrtsort. Vgl. 7.6.16.
	<i>Destination</i>	1:*	+LocationContent	Ortsdaten für den Zielort. Vgl. 7.6.16.
	<i>Via</i>	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.
	<i>NotVia</i>	0:*	+NotVia	Haltestellen oder Haltepunkte, die die Verbindung nicht passieren darf. Vgl. 9.2.4.
	<i>NoChangeAt</i>	0:*	+NoChangeAt	Haltestellen oder Haltepunkte, an denen die Verbindung keinen Umstieg vorsehen darf. Vgl. 9.2.5.
	<i>Params</i>	0:1	+TripParam	Parameter, die die Suche und Rückgabewerte beeinflussen können. Vgl. 9.2.2.

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



Die Elemente *Origin* und *Destination* treten in der Regel nur einfach auf. Nur im Fall, dass man mehrere Start- oder Zielpunkte, die jeweils eine eigene Abfahrts- bzw. Ankunftszeit mit sich bringen, definieren möchte, sollten mehrere *Origin*- bzw. *Destination*-Elemente übergeben werden. In diesem Fall sucht der Server den für die Gesamtverbindung optimalen Punkt heraus. Die Wahl des optimalen *Origin*- oder *Destination*-Punkts kann von der Uhrzeit abhängen und sich daher mit jeder gefundenen Verbindung ändern.

## 9.2.2. TripParamStructure

TripParamStructure			+Structure	Fasst die Anfragedaten für eine Verbindungsauskunft zusammen.	
TripDataFilter	PtModeFilter	0:1	+PtModeFilter	Filter nach Verkehrsmitteltypen. Vgl. 7.3.5.	
	LineFilter	0:1	+LineDirectionFilter	Erlaubte Linien (ggf. verfeinert auf Richtungen). Vgl. 7.4.6.	
	OperatorFilter	0:1	+OperatorFilter	Filter nach Verkehrsunternehmen. Vgl. 7.4.4.	
Base-TripMobilityFilter	NoSingleStep	0:1	xs:boolean	Legt fest, ob der Benutzer keine Stufe bewältigen kann. Default ist <i>false</i> .	
	NoStairs	0:1	xs:boolean	Legt fest, ob der Benutzer keine Treppe bewältigen kann. Default ist <i>false</i> .	
	NoEscalator	0:1	xs:boolean	Legt fest, ob der Benutzer keine Rolltreppe benutzen kann. Default ist <i>false</i> .	
	NoElevator	0:1	xs:boolean	Legt fest, ob der Benutzer keinen Aufzug benutzen kann. Default ist <i>false</i> .	
	NoRamp	0:1	xs:boolean	Legt fest, ob der Benutzer keine Rampe bewältigen kann. Default ist <i>false</i> .	
TripMobilityFilter	LevelEntrance	0:1	xs:boolean	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 auf <i>true</i> gesetzt. Default ist <i>false</i> .	
	BikeTransport	0:1	xs:boolean	Legt fest, ob der Benutzer ein Fahrrad an Bord der Verkehrsmittel mitnehmen will. Falls ja, wird dieser Parameter auf <i>true</i> gesetzt. Default ist <i>false</i> .	
	WalkSpeed	0:1	OpenPercent	Veränderung der Standardgehgeschwindigkeit in Prozent. Der Wert 100 stellt den Standard dar. Werte kleiner 100 stellen eine langsamere Geschwindigkeit dar, Werte größer 100 eine schnellere.	
Base-TripPolicy	a	NumberOfResults	-0:1	xs:positiveInteger	Anzahl der Verbindungsauskünfte, die der Benutzer mindestens erwartet.
	b	:::	-0:1	NumberOfResultsGroup	Angabe der gewünschten Verbindungen vor/nach dem angegebenen Zeitpunkt am Start oder Ziel (vgl. 9.2.3).
		IgnoreRealtimeData	0:1	xs:boolean	Wenn dieser Parameter gesetzt ist, sollen in der Verbindungssuche keine Echtzeitdaten oder Störungsinformationen sondern nur Sollfahrplandaten berücksichtigt werden. Default ist <i>false</i> .
		ImmediateTripStart	0:1	xs:boolean	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. Default ist <i>false</i> .
TripPolicy		InterchangeLimit	0:1	xs:positiveInteger	Anzahl der maximal zugelassenen Umsteigevorgänge.
		AlgorithmType	0:1	fastest / minChanges / leastWalking	Art der Zielfunktion, nach der der Algorithmus die Verbindung optimieren soll.

			<i>leastCost</i>	
	<i>ItModesToCover</i>	0:*	<i>IndividualModesEnumeration</i>	Für jeden IV-Typ (vgl. 0) in dieser Liste soll eine eigene monomodale Verbindung gefunden werden – zusätzlich zu den intermodalen Verbindungen.
<i>Base-TripContentFilter</i>	<i>IncludeTrackSections</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat TrackSection-Elemente (vgl. 7.6.15) für die detaillierte geografische 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 geografische 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>IncludeEstimatedTimes</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Informationen zur Echtzeitsituation mitausgegeben werden sollen. Default ist <i>false</i> .
	<i>IncludeSituationInfo</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat textuelle Echtzeitmeldungen 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 Tarifermittlung (vgl. 7.10.8).
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

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

### 9.2.3. NumberOfResultsGroup

<b>NumberOfResultsGroup</b>			<i>+Group</i>	<b>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.</b>
	<i>NumberOfResultsBefore</i>	1:1	<i>xs:positiveInteger</i>	Anzahl gewünschter Verbindungen vor dem angegebenen Zeitpunkt.
	<i>NumberOfResultsAfter</i>	1:1	<i>xs:positiveInteger</i>	Anzahl gewünschter Verbindungen nach dem angegebenen Zeitpunkt.

Tabelle 91: Beschreibung der Gruppe **NumberOfResultsGroup**.

### 9.2.4. NotViaStructure

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

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

### 9.2.5. NoChangeAtStructure

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

Tabelle 93: 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

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

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

In *ErrorMessage* können folgende Fehlerzustände auftreten:

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

Tabelle 95: Liste der Fehlerzustände in **TripResponse**.

### 9.3.2. TripResponseContextStructure

<b>TripResponseContextStructure</b>	+Structure (abgeleitet von AbstractResponseContextStructure)	Container für Daten, die in der Antwort mehrfach auftreten und referenziert werden. Vgl. 7.6.17.
-------------------------------------	---	--

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

### 9.3.3. TripResultStructure

<b>TripResultStructure</b>		+Structure	Fasst die Ergebnisdaten für eine einzelne intermodale Verbindungsauskunft zusammen.	
	<b>ResultId</b>	1:1	xs:NMTOKEN	ID des Resultats für spätere Referenzierung bzw. für Debug-Zwecke.
	<i>ErrorMessage</i>	0:*	+ErrorMessage	Fehlermeldungen bezogen auf dieses Verbindungsresultat. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. auch 7.4.2.
	<b>Trip</b>	1:1	+Trip	Daten zu einer intermodalen Verbindung. Vgl. 9.3.4.
	<i>TripFares</i>	0:*	+TripFaresResult	Ticket- und Fahrpreisinformationen zur Verbindung als Ganzes oder zu Teilen der Verbindung (vgl. 7.10.6).

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

In *ErrorMessage* können folgende Fehlerzustände auftreten:

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

Tabelle 98: Liste der Fehlerzustände in **TripResult**.

### 9.3.4. TripStructure

<b>TripStructure</b>			<b>+Structure</b>	<b>Daten zu einer einzelnen intermodalen Verbindung.</b>
	<b>TripId</b>	1:1	<i>xs:NMTOKEN</i>	ID der Verbindung für spätere Referenzierung bzw. für Debug-Zwecke.
	<b>Duration</b>	1:1	<i>xs:duration</i>	Gesamtdauer der Verbindung.
	<b>StartTime</b>	1:1	<i>xs:dateTime</i>	Startzeitpunkt der Verbindung.
	<b>EndTime</b>	1:1	<i>xs:dateTime</i>	Endzeitpunkt der Verbindung.
	<b>Interchanges</b>	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.
	<b>TripLeg</b>	1:*	<b>+TripLeg</b>	Teilweg/e dieser Verbindung. Vgl. 9.3.5.
<i>Operating-Days</i>	<i>OperatingDays</i>	0:1	<b>+OperatingDays</b>	Verkehrstage für diese Verbindung. Vgl. 7.4.8.
	<i>OperatingDaysDescription</i>	0:*	<b>+InternationalText</b>	Menschenlesbare Beschreibung der Verkehrstage, z. B. „Montag bis Freitag“ oder „Sonn- und Feiertag“.
	<i>SituationFullRef</i>	0:*	<b>+SituationFullRef</b>	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 99: Beschreibung der Struktur **TripStructure**

### 9.3.5. TripLegStructure

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

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

### 9.3.6. TimedLegStructure

<i>TimedLegStructure</i>			<i>+Structure</i>	<b>Beinhaltet einen fahrplangebundenen Verbindungsanteil (Teilweg).</b>
	<b>LegBoard</b>	1:1	<i>+LegBoard</i>	Beginn (Haltepunkt) des Teilwegs. Vgl. 9.3.9.
	<i>LegIntermediates</i>	0:*	<i>+LegIntermediates</i>	Dazwischenliegende, durchfahrene Haltepunkte auf dem Teilweg zwischen <i>LegBoard</i> und <i>LegAlight</i> . Vgl. 9.3.11..
	<b>LegAlight</b>	1:1	<i>+LegAlight</i>	Ende (Haltepunkt) des Teilwegs. Vgl. 9.3.10.
	<b>Service</b>	1:1	<i>+DatedJourney</i>	Angaben zum Verkehrsmittel, wie Linie, Verkehrsmitteltyp etc. Vgl. 7.6.5
<i>Operating-Days</i>	<i>OperatingDays</i>	0:1	<i>+OperatingDays</i>	Verkehrstage für diese Verbindung. Vgl. 7.4.8.
	<i>OperatingDaysDescription</i>	0:*	<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. 7.6.14.
	<i>ParallelService</i>	0:*	<i>+ParallelService</i>	Parallelfahrten (z.B. bei Flügelungen). Vgl. 7.6.6.
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

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

### 9.3.7. InterchangeLegStructure

<b>InterchangeLegStructure</b>			<b>+Structure</b>	<b>Beinhaltet einen Verbindungsanteil (Teilweg), der einen Umsteigevorgang zwischen zwei Verkehrsmitteln darstellt.</b>	
	a	<b>InterchangeMode</b>	-1:1	walk   parkAndRide   bikeAndRide   carHire   bikeHire   protectedConnection   guaranteedConnection   remainInVehicle   changeWithinVehicle   checkIn   checkOut	Klassifizierung von Umsteigevorgängen
	b	<b>ContinuousMode</b>		walk   demandResponsive   replacementService	Modalität für kontinuierliche Verkehre.
	<b>LegStart</b>		1:1	+LocationRef	Beginn (Ort) dieses Teilwegs. Vgl. 7.5.11.
	<b>LegEnd</b>		1:1	+LocationRef	Ende (Ort) dieses Teilwegs. Vgl. 7.5.11.
Time-Window	<b>TimeWindowStart</b>		0:1	xs:dateTime	Frühester Zeitpunkt für den Start dieses Teilwegs.
	<b>TimeWindowEnd</b>		0:1	xs:dateTime	Spätester Zeitpunkt für das Ende dieses Teilwegs.
InterchangeDuration	<b>Duration</b>		1:1	xs:duration	Notwendige Gesamtumsteigezeit.
	<b>WalkDuration</b>		0:1	xs:duration	Fußweganteil der Gesamtumsteigezeit.
	<b>BufferTime</b>		0:1	xs:duration	Pufferzeitanteil der Gesamtumsteigezeit. Pufferzeiten („Check-In-Zeiten“) sind bei manchen Verkehrsmitteln vorgeschrieben, z. B. im Flugverkehr, bei Fähren oder auch Hochgeschwindigkeitszügen.
	<b>LegDescription</b>		0:*	+InternationalText	Beschreibung des Umsteigevorgangs.
	<b>Length</b>		0:1	LengthType	Länge des Umsteigewegs.
	<b>Attribute</b>		0:*	+GeneralAttribute	Hinweise und Attribute (mit Klassifikationen) zum Umsteigevorgang. Vgl. 7.4.10.
	<b>NavigationPath</b>		0:1	+NavigationPath	Detaillierte Informationen zum geometrischen Verlauf, der Wegfolge und Zugänglichkeit. Vgl. 9.3.12.
	<b>SituationFullRef</b>		0:*	+SituationFullRef	Verweis auf eine Störungsnachricht. Diese Nachricht kann im TripResponseContext (vgl. 9.3.2) zu finden sein oder auf anderem Wege bekannt gemacht werden.
	<b>Extension</b>		0:1	xs:anyType	Erweiterungen.

Tabelle 102: Beschreibung der Struktur **InterchangeLegStructure**.



### 9.3.8. ContinuousLegStructure

<b>ContinuousLegStructure</b>			<b>+Structure</b>	<b>Beinhaltet einen Verbindungsanteil (Teilweg), der nicht fahrplangebunden ist (z. B. Fußweg).</b>
	<b>LegStart</b>	1:1	+LocationRef	Beginn (Ort) des Teilwegs dieser Verbindung. Vgl. 7.5.11.
	<b>LegEnd</b>	1:1	+LocationRef	Ende (Ort) des Teilwegs dieser Verbindung. Vgl. 7.5.11.
	<b>Service</b>	1:1	+ContinuousService	Angaben zum „Verkehrsmittel“ (z. B. Fußweg). Vgl. 7.6.11.
Time-Window	TimeWindowStart	0:1	xs:dateTime	Frühester Zeitpunkt für den Start dieses Teilwegs.
	TimeWindowEnd	0:1	xs:dateTime	Spätester Zeitpunkt für das Ende dieses Teilwegs.
	<b>Duration</b>	1:1	xs:duration	Dauer dieses Teilwegs.
	LegDescription	0:*	+InternationalText	Beschreibung dieses Verbindungsanteils.
	Length	0:1	LengthType	Länge dieses Teilwegs.
	LegTrack	0:1	+LegTrack	Detaillierter (geometrischer) Verlauf. Vgl. 7.6.14
	NavigationPath	0:1	+NavigationPath	Detaillierte Informationen zum geometrischen Verlauf, der Wegfolge und Zugänglichkeit. Vgl. 9.3.12.
	SituationFullRef	0:*	+SituationFullRef	Verweis auf eine Störungsnachricht. Diese Nachricht kann im TripResponseContext (vgl. 9.3.2) zu finden sein oder auf anderem Wege bekannt gemacht werden. Vgl. 7.8.2.
	Extension	0:1	xs:anyType	Erweiterungen.

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

### 9.3.9. LegBoardStructure

<b>LegBoardStructure</b>			<b>+Structure</b>	<b>Beschreibt die Einstiegssituation in ein Verkehrsmittel.</b>
<i>Stop-Point</i>	<b>StopPointRef</b>	1:1	→ <i>StopPoint</i>	Referenz auf einen Code für einen Haltepunkt. Vgl. 7.5.1.
	<b>StopPointName</b>	1:*	+ <i>International Text</i>	Name des Haltepunkts für Fahrgastinformation.
	<i>NameSuffix</i>	0:*	+ <i>International Text</i>	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „gegenüber vom Haupteingang“.
	<i>PlannedBay</i>	0:*	+ <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:*	+ <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>ServiceArrival</i>	0:1	+ <i>ServiceCall</i>	Informationen zur Ankunft. Vgl. 7.6.8.
	<b>ServiceDeparture</b>	1:1	+ <i>ServiceCall</i>	Informationen zur Abfahrt. Vgl. 7.6.8.
	<i>MeetsViaRequest</i>	0:1	<i>xs:boolean</i>	Dieser Halt erfüllt eine der in der Anfrage vorgegebenen Via-Bedingungen. Default ist <i>false</i> .
<i>Stop-CallStatus</i>	<i>StopSeqNumber</i>	0:1	<i>xs:positiveInteger</i>	Laufende Nummer des Halts im Fahrweg der Fahrt. Gezählt ab der Starthaltestelle der Fahrt (als Nummer 1).
	<i>DemandStop</i>	0:1	<i>xs:boolean</i>	Bedarfhalt. Fahrzeug bedient diesen Halt nur nach Voranmeldung. Default ist <i>false</i> .
	<i>UnplannedStop</i>	0:1	<i>xs:boolean</i>	Halt, der laut Planung nicht vorgesehen war. Default ist <i>false</i> .
	<i>NotServicedStop</i>	0:1	<i>xs:boolean</i>	Entgegen der Planung findet kein Halt statt. Default ist <i>false</i> .
	<i>NoBoardingAtStop</i>	0:1	<i>xs:boolean</i>	An diesem Halt der Fahrt darf nicht eingestiegen werden. Default ist <i>false</i> .
	<i>NoAlightingAtStop</i>	0:1	<i>xs:boolean</i>	An diesem Halt der Fahrt darf nicht ausgestiegen werden. Default ist <i>false</i> .

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

### 9.3.10. LegAlightStructure

<b>LegAlightStructure</b>			<b>+Structure</b>	<b>Beschreibt die Ausstiegssituation aus einem Verkehrsmittel.</b>
<i>Stop-Point</i>	<b>StopPointRef</b>	1:1	→ <i>StopPoint</i>	Referenz auf einen Code für einen Haltepunkt. Vgl. 7.5.1.
	<b>StopPointName</b>	1:*	+ <i>InternationalText</i>	Name des Haltepunkts für Fahrgastinformation.
	<i>NameSuffix</i>	0:*	+ <i>InternationalText</i>	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „gegenüber vom Haupteingang“.
	<i>PlannedBay</i>	0:*	+ <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:*	+ <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.
	<b>ServiceArrival</b>	1:1	+ <i>ServiceCall</i>	Informationen zur Ankunft. Vgl. 7.6.8.
	<i>ServiceDeparture</i>	0:1	+ <i>ServiceCall</i>	Informationen zur Abfahrt. Vgl. 7.6.8.
	<i>MeetsViaRequest</i>	0:1	<i>xs:boolean</i>	Dieser Halt erfüllt eine der in der Anfrage vorgegebenen Via-Bedingungen. Default ist <i>false</i> .
<i>Stop-CallStatus</i>	<i>StopSeqNumber</i>	0:1	<i>xs:positiveInteger</i>	Laufende Nummer des Halts im Fahrweg der Fahrt. Gezählt ab der Starthaltestelle der Fahrt (als Nummer 1).
	<i>DemandStop</i>	0:1	<i>xs:boolean</i>	Bedarfhalt. Fahrzeug bedient diesen Halt nur nach Voranmeldung. Default ist <i>false</i> .
	<i>UnplannedStop</i>	0:1	<i>xs:boolean</i>	Halt, der laut Planung nicht vorgesehen war. Default ist <i>false</i> .
	<i>NotServicedStop</i>	0:1	<i>xs:boolean</i>	Entgegen der Planung findet kein Halt statt. Default ist <i>false</i> .
	<i>NoBoardingAtStop</i>	0:1	<i>xs:boolean</i>	An diesem Halt der Fahrt darf nicht eingestiegen werden. Default ist <i>false</i> .
	<i>NoAlightingAtStop</i>	0:1	<i>xs:boolean</i>	An diesem Halt der Fahrt darf nicht ausgestiegen werden. Default ist <i>false</i> .

Tabelle 105: Beschreibung der Struktur **LegAlightStructure**.

### 9.3.11. LegIntermediateStructure

<b>LegIntermediateStructure</b>			<b>+Structure</b>	<b>Zwischenhalt auf einem Teilweg.</b>
<i>Stop-Point</i>	<b>StopPointRef</b>	1:1	→ <i>StopPoint</i>	Referenz auf einen Code für einen Haltepunkt. Vgl. 7.5.1.
	<b>StopPointName</b>	1:*	+ <i>InternationalText</i>	Name des Haltepunkts für Fahrgastinformation.
	<i>NameSuffix</i>	0:*	+ <i>InternationalText</i>	Namenszusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „gegenüber vom Haupteingang“.
	<i>PlannedBay</i>	0:*	+ <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 Planungsstand.
	<i>EstimatedBay</i>	0:*	+ <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.
	<b>ServiceArrival</b>	1:1	+ <i>ServiceCall</i>	Informationen zur Ankunft. Vgl. 7.6.8.
	<b>ServiceDeparture</b>	1:1	+ <i>ServiceCall</i>	Informationen zur Abfahrt. Vgl. 7.6.8.
	<i>MeetsViaRequest</i>	0:1	<i>xs:boolean</i>	Dieser Halt erfüllt eine der in der Anfrage vorgegebenen Via-Bedingungen. Default ist <i>false</i> .
<i>Stop-CallStatus</i>	<i>StopSeqNumber</i>	0:1	<i>xs:positiveInteger</i>	Laufende Nummer des Halts im Fahrweg der Fahrt. Gezählt ab der Starthaltestelle der Fahrt (als Nummer 1).
	<i>DemandStop</i>	0:1	<i>xs:boolean</i>	Bedarfhalt. Fahrzeug bedient diesen Halt nur nach Voranmeldung. Default ist <i>false</i> .
	<i>UnplannedStop</i>	0:1	<i>xs:boolean</i>	Halt, der laut Planung nicht vorgesehen war. Default ist <i>false</i> .
	<i>NotServicedStop</i>	0:1	<i>xs:boolean</i>	Entgegen der Planung findet kein Halt statt. Default ist <i>false</i> .
	<i>NoBoardingAtStop</i>	0:1	<i>xs:boolean</i>	An diesem Halt der Fahrt darf nicht eingestiegen werden. Default ist <i>false</i> .
	<i>NoAlightingAtStop</i>	0:1	<i>xs:boolean</i>	An diesem Halt der Fahrt darf nicht ausgestiegen werden. Default ist <i>false</i> .

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

### 9.3.12. NavigationPathStructure

<b>NavigationPathStructure</b>			<b>+Structure</b>	<b>Container für Wegbeschreibungen.</b>
	<b>NavigationSection</b>	1:*	+ <i>NavigationSection</i>	Ein oder mehrere Streckenabschnitte. Vgl. 0

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

### 9.3.13. NavigationSectionStructure

<b>NavigationSectionStructure</b>			<b>+Structure</b>	<b>Beschreibung eines Wegstücks, evtl. mit Angabe der geografischen Einbettung, der Abbiegeanweisungen und der Wegbeschaffenheit (Zugänglichkeit für mobilitätseingeschränkte Personen).</b>
	<i>TrackSection</i>	0:1	<b>+TrackSection</b>	Geografische Beschreibung des Streckenabschnitts. Vgl. 7.6.15.
	<i>TurnDescription</i>	0:*	<b>+InternationalText</b>	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:*	<b>+InternationalText</b>	Textueller Richtungshinweis zum besseren Verständnis des nachfolgenden Streckenabschnittes, z. B. „Folgen Sie der Beschilderung nach Hamburg“.
	<i>Bearing</i>	0:1	<b>AbsoluteBearing</b>	Himmelsrichtung, die nach dem Manöver eingeschlagen ist. Sie bezieht sich nicht auf das gesamte Wegstück.
	<i>SituationFullRef</i>	0:*	<b>+SituationFullRef</b>	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	<b>+AccessPath</b>	Beschreibung der Zugänglichkeit des Wegstücks. Vgl. 9.3.14.

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

### 9.3.14. AccessPathStructure

<b>AccessPathStructure</b>			<b>+Structure</b>	<b>Beschreibung der Zugänglichkeit eines Wegstücks.</b>
	<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 109: 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 Parametern kann eine bestimmte Haltestelle oder Haltestellen im Umkreis eines Ortes angefragt werden, dabei können weitere Einschränkungen vorgegeben werden, die sich als Filter auf die Ergebnisse auswirken.

In der XML-Schema-Definition *Trias\_StopEvents.xsd* werden Datentypen und Strukturen definiert, die für den Dienst Abfahrtstafeln verwendet werden.

### 10.2. Anfragestrukturen

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

#### 10.2.1. StopEventRequestStructure

<b>StopEventRequestStructure</b>			<b>+Structure</b>	<b>Fasst die Anfragedaten für eine Abfahrts- oder Ankunftstafel zusammen.</b>
	<i>Location</i>	1:1	<i>+LocationContext</i>	Ortsdaten für die Abfahrts-/Ankunftstafel. Vgl. 7.6.16.
	<i>Params</i>	0:1	<i>+StopEventParam</i>	Spezifische Anfrageparameter. Vgl. 10.2.2.

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

#### 10.2.2. StopEventParamStructure

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

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

### 10.3. Antwortstrukturen

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

#### 10.3.1. StopEventResponseStructure

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

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

In *ErrorMessage* können folgende Fehlerzustände auftreten:

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

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

#### 10.3.2. StopEventResponseContextStructure

<b>StopEventResponseContextStructure</b>	<b>+Structure (abgeleitet von AbstractResponseContextStructure)</b>	<b>Container für Daten, die in der Antwort mehrfach auftreten und referenziert werden. Vgl. 7.6.17.</b>

Tabelle 114: Beschreibung der Struktur **StopEventResponseContextStructure**.

#### 10.3.3. StopEventResultStructure

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

Tabelle 115: Beschreibung der Struktur **StopEventResultStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

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

Tabelle 116: Liste der Fehlerzustände in **StopEventResult**.

### 10.3.4. StopEventStructure

<b>StopEventStructure</b>		<b>+Structure</b>	<b>Daten zu einem einzelnen Abfahrts- oder Ankunftsereignis.</b>
	<i>PreviousCall</i>	0:*	+CallAtNearStop Abfahrts-/Ankunftsereignisse an Haltestellen vor der gefundenen Haltestelle. Vgl. 10.3.5.
	<b>ThisCall</b>	1:1	+CallAtNearStop Abfahrts-/Ankunftsereignis an der gefundenen Haltestelle. Vgl. 10.3.5.
	<i>OnwardCall</i>	0:*	+CallAtNearStop Abfahrts-/Ankunftsereignis an Haltestellen nach der gefundenen Haltestelle. Vgl. 10.3.5.
	<b>Service</b>	1:1	+DatedJourney Angaben zum Verkehrsmittel, Linie etc. Vgl. 7.6.5
<i>Operating-Days</i>	<i>OperatingDays</i>	0:1	+OperatingDays Verkehrstage für dieses Abfahrts-/Ankunftsereignis. Vgl. 7.4.8..
	<i>OperatingDaysDescription</i>	0:*	+InternationalText Menschenlesbare Beschreibung der Verkehrstage, z. B. „Montag bis Freitag“ oder „Sonn- und Feiertag“.
	<i>ParallelService</i>	0:*	+ParallelService Parallelfahrten (z.B. bei Flügelungen). Vgl. 7.6.6.
	<i>Extension</i>	0:1	xs:anyType Erweiterungen.

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

### 10.3.5. CallAtNearStopStructure

<b>CallAtNearStopStructure</b>		<b>+Structure</b>	<b>Abfahrt oder Ankunft an einer Haltestelle in der Umgebung.</b>
	<b>CallAtStop</b>	1:1	+CallAtStop Abfahrt oder Ankunft an einem Haltepunkt. Vgl. 7.6.9.
	<i>WalkDistance</i>	0:1	Distance 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 118: 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.

In der XML-Schema-Definition *Trias\_Positioning.xsd* werden Datentypen und Strukturen definiert, die für den Dienst Logische Ortung verwendet werden.

### 11.2. Anfragestrukturen

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

#### 11.2.1. PositioningRequestStructure

<i>PositioningRequestStructure</i>			+Structure	Fasst die Anfragedaten für eine logische Ortung zusammen.	
	a	<b>LastPositions</b>	-1:1	+TimedPosition	Bewegungsmuster des Fahrgasts als Folge von Koordinaten mit Zeitstempel. Vgl. 0.
	b	<b>StopSequence</b>		+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 119: Beschreibung der Struktur **PositioningRequestStructure**.

#### 11.2.2. PositioningParamStructure

<i>PositioningParamStructure</i>			+Structure	Fasst die Anfrageparameter zusammen, die die Bestimmung einer logischen Ortung des Fahrgasts im ÖV-Netz steuern.
<i>Positioning-DataFilter</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>Positioning-Policy</i>	<i>NumberOfResults</i>	0:1	xs:positiveInteger	Maximale Zahl von Ortungsvorschlägen, die in der Antwort zurückgegeben werden dürfen.

Tabelle 120: Beschreibung der Struktur **PositioningParamStructure**.

### 11.2.3. TimedPositionStructure

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

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

### 11.2.4. TimedStopStructure

<b>TimedStopStructure</b>		<b>+Structure</b>	<b>Beschreibt einen Haltepunkt evtl. mit Zeiten, wann er erreicht und/oder verlassen wird.</b>	
<i>Stop-Point</i>	<b>StopPointRef</b>	1:1	<i>→StopPoint</i>	Referenz auf einen Code für einen Haltepunkt. Vgl. 7.5.1.
	<b>StopPointName</b>	1:*	<i>+InternationalText</i>	Name des Haltepunkts für Fahrgastinformation.
	<b>NameSuffix</b>	0:*	<i>+InternationalText</i>	Namenzusatz, der bei Platzmangel evtl. auch weggelassen werden kann, z. B.: „gegenüber vom Haupteingang“.
	<b>PlannedBay</b>	0:*	<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.
	<b>EstimatedBay</b>	0:*	<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.
	<b>ArrivalTime</b>	0:1	<i>+ServiceCall</i>	Informationen zur Ankunft. Vgl. 7.6.8.
	<b>DepartureTime</b>	0:1	<i>+ServiceCall</i>	Informationen zur Abfahrt. Vgl. 7.6.8.

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

## 11.3. Antwortstrukturen

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

### 11.3.1. PositioningResponseStructure

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

Tabelle 123: Beschreibung der Struktur **PositioningResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

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

Tabelle 124: Liste der Fehlerzustände in **PositioningResponse**.

### 11.3.2. PositioningResultStructure

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

Tabelle 125: Beschreibung der Struktur **PositioningResultStructure**.

### 11.3.3. PositioningStructure

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

Tabelle 126: Beschreibung der Struktur **PositioningStructure**.

### 11.3.4. RankedPositionStructure

<i>RankedPositionStructure</i>			<i>+Structure</i>	<b>Positionsvorschlag mit Wahrscheinlichkeitseinstufung.</b>
	<b>a</b>	<b>StationaryLocation</b>	-1:1	<i>+LocationRef</i> Position im ÖV-Netz außerhalb Fahrzeugen. Vgl. 7.5.11.
	<b>b</b>	<b>TripLocation</b>		<i>+DatedJourney</i> Position im ÖV-Netz in einer Fahrt. Vgl. 7.6.5.
		<b>Ranking</b>	1:1	<i>Percent</i> Wahrscheinlichkeitseinstufung in Prozent. Der Wert 100 bedeutet absolute Gewissheit. Vgl. 7.2.1.

Tabelle 127: Beschreibung der Struktur **RankedPositionStructure**.

## 12. Dienst Fahrtinformation (EKAP)

### 12.1. Beschreibung

Im Dienst Fahrtinformation (EKAP) liefert eine EKAP Informationen zu einer bestimmten Fahrt. Dieser Dienst bezieht die Daten vom Hintergrundsystem EKAP im Unterschied zum Dienst Fahrzeuginformationen (s. Kapitel 22), der die Daten vom Fahrzeug direkt bezieht.

In der XML-Schema-Definition *Trias\_TripInfo.xsd* werden Datentypen und Strukturen definiert, die für den Dienst Fahrtinformation (EKAP) verwendet werden.

### 12.2. Anfragestrukturen

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

#### 12.2.1. TripInfoRequestStructure

<i>TripInfoRequestStructure</i>			<i>+Structure</i>	Fasst die Anfragedaten für eine Fahrtinformationsanfrage zusammen.	
<b>a</b>	<b><i>JourneyRef</i></b>	-1:1	→ <i>Journey</i>	Referenz auf eine Fahrt. Vgl. 7.4.1.	
	<b><i>OperatingDayRef</i></b>	-1:1	→ <i>OperatingDay</i>	Referenz auf einen Verkehrstag. Vgl. 7.4.1.	
	<b>b</b>	<b><i>VehicleRef</i></b>	-1:1	→ <i>Vehicle</i>	Referenz auf ein Fahrzeug. Vgl. 7.4.1.
		<b><i>TimeOfOperation</i></b>		<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. 0.	

Tabelle 128: 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

<b>TripInfoParamStructure</b>			<b>+Structure</b>	<b>Fasst die Parameter für eine Fahrtinformationsanfrage zusammen.</b>
<i>TripInfoPolicy</i>	<i>UseTimetabledDataOnly</i>	0:1	<i>xs:boolean</i>	Legt fest, dass nur Solldaten für die Auswahl der Fahrt für das Fahrzeug und der aktuellen Position in der Fahrt verwendet wird. 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>IncludeEstimatedTimes</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Echtzeitinformationen (Prognosen, Ausfälle, Umleitungen, etc.) 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 Fahrt 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>IncludeSituationInfo</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat textuelle Meldungen (z. B. Meldungen zu Störungen, Events, etc.) ausgegeben werden sollen. Default ist <i>true</i> .
	<i>IncludeTrackSections</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat auch die geografische Beschreibung der Route dieser Fahrt enthalten sein soll. Default ist <i>false</i> .
	<i>IncludeTrackProjection</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat auch die geografische Projektion (die Koordinatenfolge) der Route dieser Fahrt enthalten sein soll. Default ist <i>false</i> .
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 129: Beschreibung der Struktur **TripInfoParamStructure**.

## 12.3. Antwortstrukturen

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

### 12.3.1. TripInfoResponseStructure

<b>TripInfoResponseStructure</b>			<b>+Structure</b>	<b>Fasst die Ergebnisdaten für eine Fahrtinformationsanfrage zusammen.</b>
	<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>TripInfoResponseContext</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 130: Beschreibung der Struktur **TripInfoResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<b>Fehlercode</b>	<b>Fehlerbeschreibung</b>
<b>TRIPINFO_JOURNEYUNKNOWN</b>	Die angefragte Fahrt-ID ( <i>JourneyRef</i> ) ist unbekannt.
<b>TRIPINFO_VEHICLEUNKNOWN</b>	Die angefragte Fahrzeug-ID ( <i>VehicleRef</i> ) ist unbekannt.
<b>TRIPINFO_NOJOURNEYFOUND</b>	Für die angefragte Fahrzeug-ID ( <i>VehicleRef</i> ) kann für die Uhrzeit keine passende Fahrt gefunden werden.
<b>TRIPINFO_NOGEOINFO</b>	Geografische Information für diese Fahrt ist nicht verfügbar.

Tabelle 131: Liste der Fehlerzustände in *TripInfoResponse*.

### 12.3.2. TripInfoResponseContextStructure

<b>TripInfoResponseContextStructure</b>	+Structure (abgeleitet von AbstractResponseContextStructure)	<b>Container für Daten, die in der Antwort mehrfach auftreten und referenziert werden. Vgl. 7.6.17.</b>
---	--	---

Tabelle 132: Beschreibung der Struktur *TripInfoResponseContextStructure*.

### 12.3.3. TripInfoResultStructure

<b>TripInfoResultStructure</b>		<b>+Structure</b>	<b>Resultatstruktur, die die Fahrtinformationen zusammenfasst.</b>	
	<i>PreviousCall</i>	0:*	+CallAtStop	Bereits zurückgelegte Halte. Umfasst auch den aktuellen Halt, falls sich die Fahrt gerade an einer Haltestelle befindet. Vgl. 7.6.9.
	<i>CurrentPosition</i>	0:1	+VehiclePosition	Aktuelle Position des Fahrzeugs. Vgl. 7.6.12.
	<i>OnwardCall</i>	0:*	+CallAtStop	Die noch bevorstehenden Halte der Fahrt. Vgl. 7.6.9.
	<i>Service</i>	0:1	+DatedJourney	Angaben zum Verkehrsmittel, Linie etc. Vgl. 7.6.5
<i>Operating-Days</i>	<i>OperatingDays</i>	0:1	+OperatingDays	Verkehrstage für diese Fahrt. Vgl. 7.4.8.
	<i>OperatingDaysDescription</i>	0:*	+InternationalText	Menschenlesbare Beschreibung der Verkehrstage, z. B. „Montag bis Freitag“ oder „Sonn- und Feiertag“.
	<i>ParallelService</i>	0:*	+ParallelService	Parallelfahrten (z.B. bei Flügelungen). Vgl. 7.6.6.
	<i>JourneyTrack</i>	0:1	+LegTrack	Geografische Beschreibung der kompletten Fahrzeugfahrt. Vgl. 7.6.14.
	<i>Extension</i>	0:1	xs:anyType	Erweiterungen.

Tabelle 133: 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.

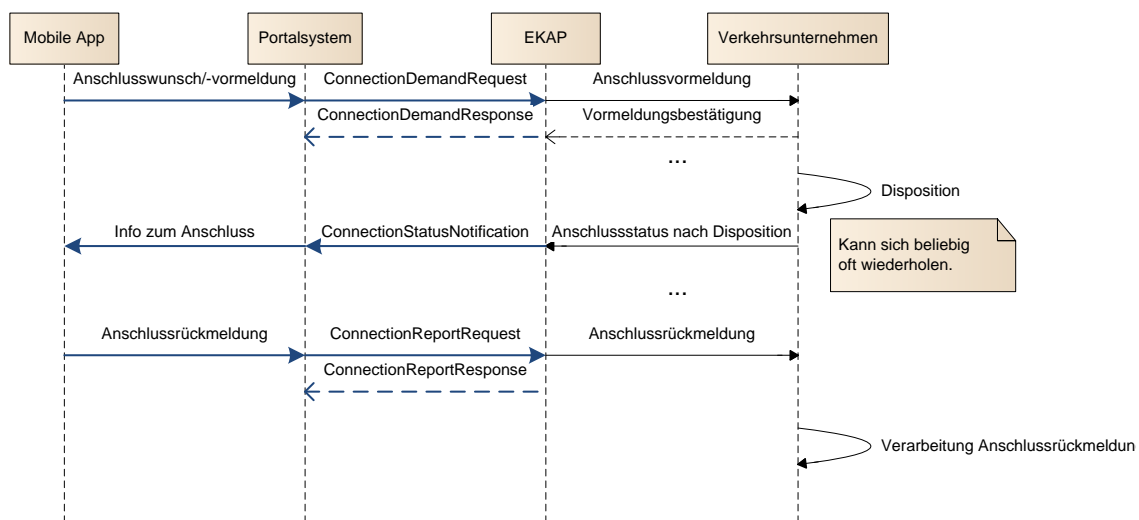


Abbildung 4: Ablauf der Anschlussdienste mit aktiver Benachrichtigung bei Statusänderung

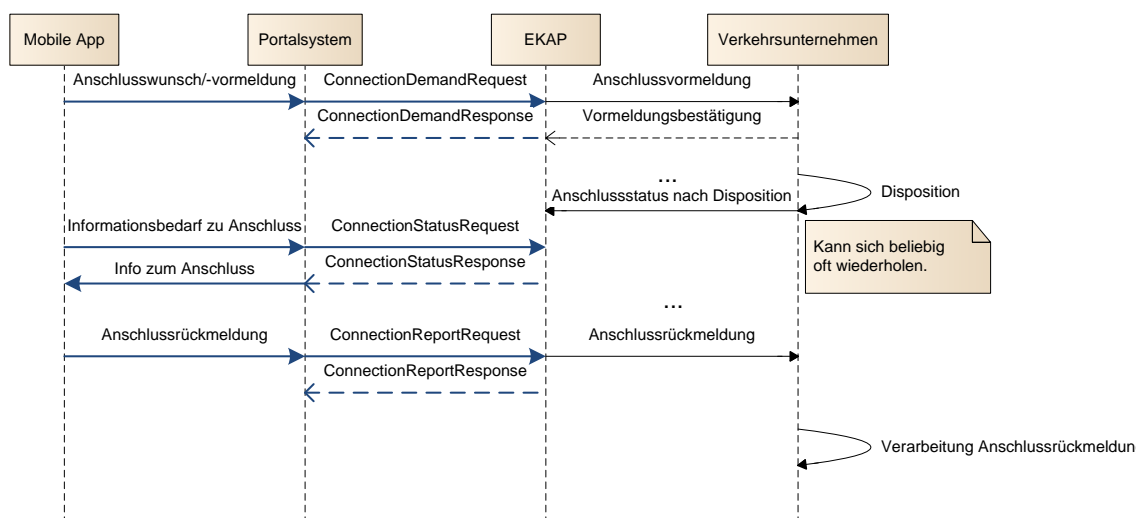


Abbildung 5: Ablauf der Anschlussdienste mit passiver Benachrichtigung bei Statusänderung

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 4)
  - b. passive Information mittels Anfrage an die EKAP (Abbildung 5)
4. Rückmeldung des Reisenden oder seiner Applikation zur Anslusserreichung.

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 und/oder Leitsysteme 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 20).

### 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:

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

Tabelle 134: Beschreibung der einfachen Typen

### 13.3. Komplexe Strukturen

Im Folgenden werden die komplexen Strukturen beschrieben, die in der XML-Schema-Definition *Trias\_Connections.xsd* definiert sind.

#### 13.3.1. DatedConnectionStructure

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

Tabelle 135: Beschreibung der Struktur **DatedConnectionStructure**

#### 13.3.2. FeederDistributorStructure

<b>FeederDistributorStructure</b>			<b>+Structure</b>	<b>Beinhaltet einen Zubringer oder einen Abbringer an einem definierten Ort zu einer definierten Betriebszeit.</b>
<i>DatedJourneyRef</i>	<b>JourneyRef</b>	1:1	<i>→Journey</i>	Referenz auf die Fahrt des Zubringers oder Abbringers. Vgl. 7.4.1.
	<b>OperatingDayRef</b>	1:1	<i>→OperatingDay</i>	Referenz auf den Betriebstag. Vgl. 7.4.1.
<i>LineDirection</i>	<b>LineRef</b>	1:1	<i>→LineCode</i>	Referenz auf die Linie. Vgl. 7.4.1.
	<b>DirectionRef</b>	0:1	<i>→DirectionCode</i>	Referenz auf die Linienrichtung. Vgl. 7.4.1.
	<b>OperatorRef</b>	0:1	<i>→Operator</i>	Operator-ID. Vgl. 7.4.1.
	<b>ConnectionLocation</b>	1:1	<i>+CallAtStop</i>	Ort des Anschlusses, vgl. 7.6.9.

Tabelle 136: Beschreibung der Struktur **FeederDistributorStructure**

#### 13.3.3. GeneralizedConnectionStructure

<b>GeneralizedConnectionStructure</b>				<b>+Structure</b>	<b>Definiert eine Umstiegsrelation. Enthält die Verbindungsarten Einstieg (Pickup), Ausstieg (Set-Down) und Umstieg (DatedConnection)</b>
	<b>a</b>	<b>DatedConnection</b>	-1:1	<i>+DatedConnection</i>	Umstiegsverbindung, für die der Status abgefragt werden soll. Vgl. 13.3.1.
	<b>b</b>	<b>PickUpLocation</b>		<i>+DatedCallAtLocation</i>	Zustieg in Fahrzeug, für den der Status abgefragt werden soll. Vgl. 0.
	<b>c</b>	<b>SetDownLocation</b>		<i>+DatedCallAtLocation</i>	Ausstieg aus Fahrzeug, für den der Status abgefragt werden soll. Vgl. 0.

Tabelle 137: Beschreibung des Typs **GeneralizedConnectionStructure**

### 13.3.4. ConnectionStatusStructure

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

Tabelle 138: Beschreibung der Struktur **ConnectionStatusStructure**

### 13.3.5. RecommendationStructure

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

<b>RecommendationStructure</b>			<i>+Structure</i>	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.
	<b>RecommendationId</b>	1:1	<i>xs:NMTOKEN</i>	ID der Anfrage für spätere Referenzierung bzw. für Debug-Zwecke
	<b>Text</b>	1:*	<i>+InternationalText</i>	Beschreibung der Alternative.
	<b>Type</b>	1:1	<i>RecommendationTypeEnumeration</i>	Typ der Empfehlung, vgl. 13.2.

Tabelle 139: Beschreibung der Struktur **RecommendationStructure**

## 13.4. Anfragestrukturen Anschlussmeldung

### 13.4.1. ConnectionDemandRequestStructure

<b>ConnectionDemandRequestStructure</b>			<i>+Structure</i>	<b>Beinhaltet die Anfragedaten für eine Meldung von Umsteigern, Einsteigern oder Aussteigern bei regulären oder Bedarfsverkehren.</b>
	<b>RequestId</b>	1:1	<i>xs:NMTOKEN</i>	ID der Anfrage für spätere Referenzierung bzw. für Debug-Zwecke
	<b>Connection</b>	1:1	<i>+Generalized Connection</i>	Anschlussrelation. Vgl. 13.3.3.
	<i>NumberOfPersons</i>	0:1	<i>xs:positiveInteger</i>	Anzahl der Um-, Ein- oder Aussteiger.
	<i>TravelProbability</i>	0:1	<i>Percent</i>	(Kumulierte) Reisewahrscheinlichkeit für die angegebenen Um-, Ein- oder Aussteiger. Vgl. 7.2.1.
	<i>RequiredInterchangeDuration</i>	0:1	<i>xs:duration</i>	Gibt die Umsteigezeit an, die von den Reisenden, die diesen Anschlusswunsch initiiert haben, voraussichtlich benötigt wird, um vom Zubringer zum Abbringer zu gelangen.
	<i>PassengerAccessibilityNeeds</i>	0:*	<i>+PassengerAccessibility</i>	Pro (anonymen) Fahrgast mit besonderen Bedürfnissen kann an dieser Stelle ein Element übermittelt werden, das diese Bedürfnisse ausdrückt. Vgl. 7.6.19.
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 140: Beschreibung der Struktur **ConnectionDemandRequestStructure**.

### 13.4.2. ConnectionDemandDeleteRequestStructure

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

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

## 13.5. Antwortstrukturen Anschlussmeldung

### 13.5.1. ConnectionDemandResponseStructure

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

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

In *ErrorMessage* können folgende Fehlerzustände auftreten:

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

Tabelle 143: Liste der Fehlerzustände in **ConnectionDemandResponse**.

### 13.5.2. ConnectionDemandDeleteResponseStructure

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

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

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<b>Fehlercode</b>	<b>Fehlerbeschreibung</b>
<b>CONNECTIONDEMAND_REQUESTID_UNKNOWN</b>	Die Anfrage-ID ist der EKAP nicht bekannt.
<b>CONNECTIONDEMAND_DELETIONNOTPOSSIBLE</b>	Die Anschlussmeldung konnte nicht storniert werden.

Tabelle 145: Liste der Fehlerzustände in **ConnectionDemandDeleteResponse**.

## 13.6. Anfragestrukturen Anschlussstatus

### 13.6.1. ConnectionStatusRequestStructure

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

Tabelle 146: Beschreibung der Struktur **ConnectionStatusRequestStructure**

### 13.6.2. ConnectionStatusNotificationStructure

<b>ConnectionStatusNotificationStructure</b>			+Structure	Push-Information zu Anschlussstatus. Genutzt von TripMonitoring oder optional aktiv gesendet von EKAP nach ConnectionDemands.
	<b>ConnectionStatus</b>	1:1	+ConnectionStatus	Enthält den tatsächlichen Verbindungsstatus. Vgl. 0.

Tabelle 147: Beschreibung der Struktur **ConnectionStatusNotificationStructure**

## 13.7. Antwortstrukturen Anschlussstatus

### 13.7.1. ConnectionStatusResponseStructure

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

Tabelle 148: Beschreibung der Struktur **ConnectionStatusResponseStructure**

In *ErrorMessage* können folgende Fehlerzustände auftreten:

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

Tabelle 149: Liste der Fehlerzustände in ConnectionStatusResponseStructure.

## 13.8. Anfragestrukturen Anschlussrückmeldung

### 13.8.1. ConnectionReportRequestStructure

<b>ConnectionReportRequestStructure</b>			<i>+Structure</i>	<b>Beinhaltet die Daten für eine Meldung, die besagt, ob der Anschluss für den Reisenden zustande gekommen ist.</b>
	<i>RequestId</i>	1:1	<i>xs:NMTOKEN</i>	ID der Anfrage für spätere Referenzierung bzw. für Debug-Zwecke.
	<i>Connection</i>	1:1	<i>+Generalized Connection</i>	Anschlussrelation, über die berichtet wird. Vgl. 13.3.3.
	<i>Succeeded</i>	1:1	<i>xs:boolean</i>	Gibt an, ob der Anschluss aus Sicht des Reisenden zustande gekommen ist.
	<i>Reason</i>	0:*	<i>+International Text</i>	Optional ist die Angabe eines Grundes für den Anschlussbruch oder das Erreichen des Anschlusses möglich.
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

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

## 13.9. Antwortstrukturen Anschlussrückmeldung

### 13.9.1. ConnectionReportResponseStructure

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

Tabelle 151: Beschreibung der Struktur **ConnectionReportResponseStructure**

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<b>CONNECTIONREPORT_FEEDER_UNKNOWN</b>	Der Zubringer ist der EKAP nicht bekannt.
<b>CONNECTIONREPORT_DISTRIBUTOR_UNKNOWN</b>	Der Abbringer ist der EKAP nicht bekannt.
<b>CONNECTIONREPORT_DEPARTURE_BEFORE_ARRIVAL</b>	Die Sollabfahrt des Abbringers liegt vor der Sollankunft des Zubringers. Umstieg folglich nicht möglich.
<b>CONNECTIONREPORT_FEEDER_LOCATION_UNKNOWN</b>	Der referenzierte Ort des Umstiegs für den Zubringer ist unbekannt.
<b>CONNECTIONREPORT_DISTRIBUTOR_LOCATION_UNKNOWN</b>	Der referenzierte Ort des Umstiegs für den Abbringer ist unbekannt.

Tabelle 152: Liste der Fehlerzustände in ConnectionReportResponse.

## 14. Dienst Fahrpreis- und Tarifberechnung

### 14.1. Beschreibung

Dieser Dienst stellt allgemeine, haltestellenbezogene oder verbindungsbezogene Tarifinformationen bereit. In der XML-Schema-Definition *Trias\_Fares.xsd* werden Datentypen und Strukturen definiert, die für den Dienst Fahrpreis- und Tarifberechnung verwendet werden.

### 14.2. Anfragestrukturen

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

#### 14.2.1. FaresRequestStructure

<i>FaresRequestStructure</i>			+Structure	Fasst die Daten für eine Tarifierfrage zusammen.
a	<b>StopFaresRequest</b>	-1:1	+ <i>StopFaresRequest</i>	Haltestellenbezogene Tarifierfrage. Vgl. 14.2.2.
b	<b>StaticFaresRequest</b>		+ <i>StaticFaresRequest</i>	Allgemeine Tarifierfrage. Vgl. 0
c	<b>TripFaresRequest</b>		+ <i>TripFaresRequest</i>	Verbindungsbezogene Tarifierfrage. Vgl. 0..
d	<b>MultiTripFaresRequest</b>		+ <i>MultiTripFaresRequest</i>	Aggregierte Tarifierfrage für mehrere Verbindungen. Vgl. 14.2.5.
<i>Params</i>		0:1	+ <i>FaresParam</i>	Parameter für Tarifierfrage. Vgl. 7.10.8.
<i>Extension</i>		0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 153: Beschreibung der Struktur **FaresRequestStructure**.

#### 14.2.2. StopFaresRequestStructure

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

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

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



### 14.2.3. StaticFaresRequestStructure

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

<b>StaticFaresRequestStructure</b>			<i>+Structure</i>	<b>Fasst die Daten für eine allgemeine (statische) Tarifrfrage zusammen.</b>
	<i>Date</i>	0:1	<i>xs:date</i>	Stichtag für die Gültigkeit der Tarifauskunft.
	<i>TicketRef</i>	0:*	<i>→TicketCode</i>	Code von Tickets, für die weitere Informationen angefragt werden. Falls kein TicketRef angegeben wird, soll der Server über alle verfügbaren Tickets Auskunft geben.

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

### 14.2.4. TripFaresRequestStructure

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

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

Tabelle 156: 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.

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

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

### 14.3. Antwortstrukturen

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

#### 14.3.1. FaresResponseStructure

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

Tabelle 158: Beschreibung der Struktur **FaresResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<b>Fehlercode</b>	<b>Fehlerbeschreibung</b>
<b>FARES_DATEOUTOFFRANGE</b>	Die Tarifrage kann nicht bearbeitet werden, weil für das gewünschte Datum keine Informationen zur Verfügung stehen.
<b>FARES_STOPPOINTUNKNOWN</b>	Die Tarifrage kann nicht bearbeitet werden, weil der angefragte Haltepunkt unbekannt ist.

Tabelle 159: Liste der Fehlerzustände in **FaresResponse**.

#### 14.3.2. FaresResultStructure

<b>FaresResultStructure</b>			<b>+Structure</b>	<b>Resultatstruktur für die Tarifinformationen.</b>
	<b>ResultId</b>	1:1	<i>xs:NMTOKEN</i>	ID des Resultats für spätere Referenzierung.
<b>a</b>	<b>StopFaresResult</b>	-1:1	<b>+StopFaresResult</b>	Antwort auf haltestellenbezogene Tarifrage. Vgl. 14.3.3.
<b>b</b>	<b>StaticFaresResult</b>		<b>+StaticFaresResult</b>	Antwort auf allgemeine Tarifrage. Vgl. 14.3.4.
<b>c</b>	<b>TripFaresResult</b>		<b>+TripFaresResult</b>	Antwort auf verbindungsbezogene Tarifrage. Vgl. 7.10.6.
<b>d</b>	<b>MultiTripFaresResult</b>		<b>+MultiTripFaresResult</b>	Antwort auf Tarifrage für mehrere Verbindungen. Vgl. 14.3.6.

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

#### 14.3.3. StopFaresResultStructure

<b>StopFaresResultStructure</b>			<b>+Structure</b>	<b>Resultatstruktur für haltestellenbezogene Tarifinformationen.</b>
	<b>FareZoneListInArea</b>	1:*	<b>+FareZoneListInArea</b>	Liste der Tarifzonen, in denen die angefragte Haltestelle liegt. Vgl. 7.10.3.
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

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

#### 14.3.4. StaticFaresResultStructure

<b>StaticFaresResultStructure</b>			<b>+Structure</b>	<b>Resultatstruktur für allgemeine Tarifinformationen.</b>
	<i>Ticket</i>	0:*	<b>+Ticket</b>	Liste der verfügbaren Tickets. Vgl. 7.10.5.
	<i>StaticInfoUrl</i>	0:1	<b>+WebLinkl</b>	Links auf Informationsseiten im Web (vgl. 7.2.4).
	<i>Extension</i>	0:1	<b>xs:anyType</b>	Erweiterungen.

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

#### 14.3.5. TripTicketReferenceStructure

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

Tabelle 163: Beschreibung der Struktur **TripTicketReferenceStructure**.

#### 14.3.6. MultiTripFaresResultStructure

<b>MultiTripFaresResultStructure</b>			<b>+Structure</b>	<b>Fasst die Ergebnisdaten für eine Tarifauskunft zu mehreren Verbindungen zusammen.</b>
	<i>ErrorMessage</i>	0:*	<b>+ErrorMessage</b>	Fehlermeldungen bezogen auf diese Tarifauskunft. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. auch 7.4.2.
	<b>TripTicketReference</b>	<b>1:*</b>	<b>+TripTicketReference</b>	Verknüpfung von Tickets mit Verbindungen (oder Teilen davon). Vgl. 14.3.5.
	<i>Ticket</i>	0:*	<b>+Ticket</b>	Fahrscheine, die auf diesem Abschnitt der Verbindung gültig sind (vgl. 7.10.5).
	<i>PassedZones</i>	0:1	<b>+FareZoneListInArea</b>	Die Menge der durchfahrenen Tarifzonen, über alle Verbindungen gesehen (vgl. 7.10.3).
	<i>StaticInfoURL</i>	0:*	<b>+WebLink</b>	URL zu Informationsseiten (vgl. 7.2.4).

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

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<b>Fehlercode</b>	<b>Fehlerbeschreibung</b>
<b>FARES_OUTOFAREA</b>	Der in der Verbindungsauskunft gefundene Weg verlässt das Tarifgebiet.
<b>FARES_JOURNEYNOTPERMITTED</b>	Ein in der Verbindungsauskunft benutztes Verkehrsmittel ist für den Tarif nicht zulässig.
<b>FARES_ADDITIONALCHARGES</b>	Zusätzliche Gebühren sind voraussichtlich zu entrichten (z. B. Mautzuschläge oder Reservierungsgebühren).
<b>FARES_ADDITIONALTICKETS</b>	Zusätzliche Fahrscheine sind notwendig, da nicht für alle Verkehrsmittel oder für alle angegebenen Verbindungen ein passender Fahrschein ermittelt werden konnte.
<b>FARES_ROUTENOTFEASIBLE</b>	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 165: Liste der Fehlerzustände in *MultiTripFaresResultStructure*.

## 15. Dienst Anreicherung

### 15.1. Beschreibung

Dieser Dienst ist dazu gedacht, für bereits vorher bekannte Objekte zusätzliche (oder aktualisierte) Informationen von einer EKAP holen zu können.

Auf welchem Wege diese Objekte bereits vorab bekannt geworden sind, ist dabei nicht relevant; das kann per TRIAS-Schnittstelle oder auch auf einem anderen Wege geschehen sein.

Die Motivation für den Anreicherungsdienst kam aus den Bedürfnissen des Forschungsprojektes DELFIplus, bei dem die verteilte DELFI-Auskunft in ein hybrides Auskunftssystem überführt wurde, welches auf einem deutschlandweiten Datenbestand Verbindungen berechnet, nicht-integrierte Informationen wie z.B. Tarife oder Echtzeitinformationen von lokalen EKAPs holen möchte, die über entsprechendes Spezialwissen verfügen. Der Anreicherungsdienst wurde aber über die unmittelbaren Bedürfnisse des Forschungsprojektes DELFIplus hinaus spezifiziert, so dass sinnvolle allgemeine Anreicherungsmöglichkeiten im Rahmen der VDV431 geschaffen werden.

Aus diesem Szenario lässt sich leicht ableiten, dass ein anfragendes System Informationen von zwei (oder mehr) verschiedenen EKAPs erhalten möchte. Z.B. könnten für eine Verbindungsauskunft nachträglich Prognoseinformationen von einer anderen EKAP abgefragt werden. In diesem Fall ergibt sich die Schwierigkeit, dass jede EKAP ihr eigenes ID-System mitbringt, innerhalb dessen die Objekte, die sie kennt, referenziert werden können. Wenn nun eine Verbindungsauskunft auf EKAP A gerechnet wird und Prognoseinformationen für ein Verkehrsmittel aus diesem Verbindungsergebnis von EKAP B angereichert werden sollen, so kann man nicht ohne Weiteres davon ausgehen, dass EKAP B die betreffende Fahrt-ID, die von EKAP A verwendet wurde, kennt. Aus diesem Grund wird in den Anreicherungsanfragen die Information mitgegeben, ob in den Anfragedaten „Fremd-IDs“ enthalten sind.

Falls eine EKAP mit fremden IDs konfrontiert wird, muss sie versuchen, die entsprechenden Objekte auf andere Weise zu erkennen und im eigenen Datenbestand wiederzufinden. In der Antwort wird die EKAP notwendigerweise ihre eigenen Objekt-IDs verwenden. Damit das anfragende System die in der Anreicherungsantwort enthaltenen Objekte zweifelsfrei zuordnen kann, ist es unerlässlich, dass die EKAP in ihrer Antwort exakt die Struktur der Anreiche-

rungsanfrage widerspiegelt (Prinzip der Strukturhaltung). Wenn also z.B. eine Anreicherungsanfrage für den StopEvents-Dienst (*StopEventRefineRequest*) zehn Elemente vom Typ *StopEventResult* enthält und die EKAP für die Elemente 2, 7 und 10 eine Anreicherung bereitstellen kann, ist es erforderlich, dass die EKAP auch die anderen Elemente unverändert und in der ursprünglichen Reihenfolge zurückgibt.

Die Anreicherung ist für folgende Dienste möglich:

- IndividualRoute  
*Beispiel: eine vorher ermittelte PKW-Route wird nachträglich mit der geografischen Streckenführung angereichert*
- LocationInformation  
*Beispiel: zu einer Haltestellen-ID werden die Details der Haltestelle ermittelt (Name, Ort, Koordinaten, ...)*
- StopEvent  
*Beispiel: Zu einer Haltestellenabfahrt (unter vielen) werden nachträglich die vorausgehenden und nachfolgenden Halte der Fahrt ermittelt*
- TripInfo  
*Beispiel: zu einer Fahrplanfahrt werden nachträglich die Echtzeitinformationen ermittelt (Zuglauf)*
- Trip  
*Beispiel: zu einer Verbindungsauskunft werden nachträglich die Fahrpreiskosten und mögliche Tickets abgefragt*

Um die Kommunikation effizient zu halten, besteht mit einem allgemeinen Refine-Dienst die Möglichkeit, mehrere Anreicherungsanfragen in einem HTTP-Request zu verpacken.

In der XML-Schema-Definition *Trias\_Refine.xsd* werden Datentypen und Strukturen definiert, die für den Dienst Anreicherung verwendet werden.

## 15.2. Anfragestrukturen

Eine Informationsanreicherung wird mittels eines Elements **RefineRequest** vom Typ *RefineRequestStructure* angefordert.

### 15.2.1. RefineRequestStructure (in Trias\_Refine.xsd)

<b>RefineRequestStructure</b>		<b>+Structure</b>	<b>Fordert zusätzliche Informationen an.</b>
<i>IndividualRouteRefineRequest</i>	0:*	<i>+IndividualRouteRefineRequest</i>	IV-Routing-Strukturen, die angereichert werden sollen. Vgl. 15.2.2.
<i>LocationInformationRefineRequest</i>	0:*	<i>+LocationInformationRefineRequest</i>	Ortsinformationen, die angereichert werden sollen. Vgl. 15.2.3.
<i>StopEventRefineRequest</i>	0:*	<i>+StopEventRefineRequest</i>	Abfahrtstafeln, die angereichert werden sollen. Vgl. 15.2.4.
<i>TripInfoRefineRequest</i>	0:*	<i>+TripInfoRefineRequest</i>	Fahrtinformationen, die angereichert werden sollen. Vgl. 15.2.5.
<i>TripRefineRequest</i>	0:*	<i>+TripRefineRequest</i>	Verbindungsauskünfte, die angereichert werden sollen. Vgl. 15.2.6.

Tabelle 166: Beschreibung der Struktur **RefineRequestStructure**.

### 15.2.2. IndividualRouteRefineRequestStructure (in Trias\_IndividualTrips.xsd)

<b>IndividualRouteRefineRequestStructure</b>		<b>+Structure</b>	<b>Fordert zusätzliche Informationen für ein IV-Routing an.</b>
	<i>RefineParams</i>	0:1	+ <i>IndividualTripRefineParam</i> Wählt die gewünschten Anreicherungen aus. Vgl. 15.2.7.
	<b>IndividualRouteResult</b>	1:1	+ <i>RouteResult</i> Die anzureichernde IV-Route. Vgl. 17.3.2.
	<i>Extension</i>	0:1	<i>xs:anyType</i> Erweiterungen.

Tabelle 167: Beschreibung der Struktur **IndividualRouteRefineRequestStructure**.

### 15.2.3. LocationInformationRefineRequestStructure (in Trias\_Locations.xsd)

<b>LocationInformationRefineRequestStructure</b>		<b>+Structure</b>	<b>Fordert zusätzliche Informationen für eine Ortsinformation an.</b>
	<i>RefineParams</i>	0:1	+ <i>LocationInformationRefineParam</i> Optionen für die Anreicherungsfunktion. Vgl. 15.2.8.
	<b>LocationResult</b>	1:1	+ <i>LocationResult</i> Die anzureichernde Ortsinformation. Vgl. 0.
	<i>Extension</i>	0:1	<i>xs:anyType</i> Erweiterungen.

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

### 15.2.4. StopEventRefineRequestStructure (in Trias\_StopEvents.xsd)

<b>StopEventRefineRequestStructure</b>		<b>+Structure</b>	<b>Fordert zusätzliche Informationen für Abfahrtstafeln an.</b>
	<i>RefineParams</i>	0:1	+ <i>StopEventRefineParam</i> Wählt die gewünschten Anreicherungen aus. Vgl. 15.2.9.
	<b>StopEventResult</b>	1:*	+ <i>StopEventResult</i> Die anzureichernden Abfahrtstafeln. Vgl. 10.3.3.
	<i>Extension</i>	0:1	<i>xs:anyType</i> Erweiterungen.

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

### 15.2.5. TripInfoRefineRequestStructure (in Trias\_TripInfo.xsd)

<b>TripInfoRefineRequestStructure</b>		<b>+Structure</b>	<b>Fordert zusätzliche Informationen für Fahrtinformationen an.</b>
	<i>RefineParams</i>	0:1	+ <i>TripInfoRefineParam</i> Wählt die gewünschten Anreicherungen aus. Vgl. 15.2.10.
	<b>TripInfoResult</b>	1:1	+ <i>TripInfoResult</i> Die anzureichernden Fahrtinformationen. Vgl. 12.3.3.
	<i>Extension</i>	0:1	<i>xs:anyType</i> Erweiterungen.

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

### 15.2.6. TripRefineRequestStructure (in Trias\_Trips.xsd)

<b>TripRefineRequestStructure</b>		<b>+Structure</b>	<b>Fordert zusätzliche Informationen für eine Verbindungsauskunft an.</b>
	<i>RefineParams</i>	0:1	+ <i>TripRefineParam</i> Wählt die gewünschten Anreicherungen aus. Vgl. 15.2.11.
	<b>TripResult</b>	1:1	+ <i>TripResult</i> Die anzureichernde Verbindungsauskunft. Vgl. 9.3.3.
	<i>Extension</i>	0:1	<i>xs:anyType</i> Erweiterungen.

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

### 15.2.7. IndividualTripRefineParamStructure (in Trias\_IndividualTrips.xsd)

<b>IndividualTripRefineParamStructure</b>			<b>+Structure</b>	<b>Parametrisiert die Anreicherungsanfragen des IV-Routings</b>
<i>Refine-Options</i>	<i>ForeignObjectRefs</i>	0:1	<i>xs:boolean</i>	Falls true, zeigt dieses Element an, dass in den Anfragedaten Objektreferenzen von anderen EKAPs enthalten sein können. Default ist <i>false</i> .
<i>Base-TripContentFilter</i>	<i>IncludeTrackSections</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat TrackSection-Elemente (vgl. 7.6.15) für die detaillierte geografische Beschreibung des Wegs ausgegeben werden sollen. Default ist <i>false</i> .
	<i>IncludeLegProjection</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat der detaillierte geografische Verlauf des Wegs als Koordinatenfolge ausgegeben werden soll. Default ist <i>false</i> .
	<i>IncludeTurnDescription</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Routenhinweise mit Abbiegeempfehlungen ausgegeben werden sollen. Default ist <i>false</i> .
	<i>IncludeAccessibility</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Informationen zur Barrierefreiheit ausgegeben werden sollen. Default ist <i>false</i> .
	<i>IncludeEstimatedTimes</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Informationen zur Echtzeitsituation ausgegeben werden sollen. Default ist <i>false</i> .
	<i>IncludeSituationInfo</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Störungsmeldungen ausgegeben werden sollen. Default ist <i>false</i> .
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

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

### 15.2.8. LocationInformationRefineParamStructure (in Trias\_Locations.xsd)

<b>LocationInformationRefineParamStructure</b>			<b>+Structure</b>	<b>Parametrisiert die Anreicherungsanfragen der Ortsinformationen.</b>
<i>Refine-Options</i>	<i>ForeignObjectRefs</i>	0:1	<i>xs:boolean</i>	Falls true, zeigt dieses Element an, dass in den Anfragedaten Objektreferenzen von anderen EKAPs enthalten sein können. Default ist <i>false</i> .
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

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

### 15.2.9. StopEventRefineParamStructure (in Trias\_StopEvents.xsd)

<b>StopEventRefineParamStructure</b>			<b>+Structure</b>	<b>Parametrisiert die Anreicherungsanfragen von Abfahrtstafeln</b>
<i>Refine-Options</i>	<i>ForeignObjectRefs</i>	0:1	<i>xs:boolean</i>	Falls <i>true</i> , zeigt dieses Element an, dass in den Anfragedaten Objektreferenzen von anderen EKAPs enthalten sein können. Default ist <i>false</i> .
<i>StopEventContentFilter</i>	<i>IncludePreviousCalls</i>	0:1	<i>xs:boolean</i>	Gibt an, ob je Fahrt die vorausgehenden Halte angeführt werden sollen. Default ist <i>false</i> .
	<i>IncludeOnwardCalls</i>	0:1	<i>xs:boolean</i>	Gibt an, ob je Fahrt die nachfolgenden Halte angeführt werden sollen. Default ist <i>false</i> .
	<i>IncludeOperatingDays</i>	0:1	<i>xs:boolean</i>	Gibt an, ob die Verkehrstage der Fahrten angegeben werden sollen. Default ist <i>false</i> .
	<i>IncludeRealtimeData</i>	0:1	<i>xs:boolean</i>	Steuert, ob Echtzeitdaten berücksichtigt und ausgegeben werden sollen. Default ist <i>false</i> .
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

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

## 15.2.10. TripInfoRefineParamStructure (in Trias\_TripInfo.xsd)

<b>TripInfoRefineParamStructure</b>			<b>+Structure</b>	<b>Parametrisiert die Anreicherungsanfragen nach Fahrtinformationen</b>
<i>Refine-Options</i>	<i>ForeignObjectRefs</i>	0:1	<i>xs:boolean</i>	Falls <i>true</i> , zeigt dieses Element an, dass in den Anfragedaten Objektreferenzen von anderen EKAPs enthalten sein können. 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>IncludeEstimatedTimes</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Echtzeitinformationen (Prognosen, Ausfälle, Umleitungen, etc.) 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>IncludeSituationInfo</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat textuelle Meldungen (z. B. Meldungen zu Störungen, Events, etc.) ausgegeben werden sollen. Default ist <i>true</i> .
	<i>IncludeTrackSection</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat auch die geografische Beschreibung der Route dieser Fahrt enthalten sein soll. Default ist <i>false</i> .
	<i>IncludeLegProjection</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat auch die geografische Projektion (die Koordinatenfolge) der Route dieser Fahrt enthalten sein soll. Default ist <i>false</i> .
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 175: Beschreibung der Struktur **TripInfoRefineParamStructure**.

## 15.2.11. TripRefineParamStructure (in Trias\_Trips.xsd)

<b>TripRefineParamStructure</b>			<b>+Structure</b>	<b>Parametrisiert die Anreicherungsanfragen zur Verbindungsauskunft</b>
<i>Refine-Options</i>	<i>ForeignObjectRefs</i>	0:1	<i>xs:boolean</i>	Falls <i>true</i> , zeigt dieses Element an, dass in den Anfragedaten Objektreferenzen von anderen EKAPs enthalten sein können. Default ist <i>false</i> .
	<i>RefineLegRef</i>	1:*	<i>→xs:NMTOKEN</i>	Gibt an, für welche Teile einer Verbindungsauskunft (Legs) die gewünschten Anreicherungen durchgeführt werden sollen.
<i>Base-TripContentFilter</i>	<i>IncludeTrackSections</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat TrackSection-Elemente (vgl. 7.6.15) für die detaillierte geografische Beschreibung des Wegs ausgegeben werden sollen. Default ist <i>false</i> .
	<i>IncludeLegProjection</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat der detaillierte geografische Verlauf des Wegs als Koordinatenfolge ausgegeben werden soll. Default ist <i>false</i> .
	<i>IncludeTurnDescription</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Routenhinweise mit Abbiegeempfehlungen ausgegeben werden sollen. Default ist <i>false</i> .
	<i>IncludeAccessibility</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Informationen zur Barrierefreiheit ausgegeben werden sollen. Default ist <i>false</i> .
	<i>IncludeEstimatedTimes</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Informationen zur Echtzeitsituation ausgegeben werden sollen. Default ist <i>false</i> .
	<i>IncludeSituationInfo</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Störungsmeldungen ausgegeben 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 ausgegeben werden sollen. Default ist <i>false</i> .
	<i>IncludeFares</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Tarifinformationen ausge-



				geben 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 ausgegeben werden sollen. Default ist <i>false</i> .
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.

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

### 15.3. Antwortstrukturen

Das Ergebnis einer Anreicherungsanfrage wird mittels eines Elements **RefineResponse** vom Typ *RefineResponseStructure* übertragen.

In den jeweiligen dienstspezifischen Antwortstrukturen (z.B. *StopEventRefineResponse* oder *TripRefineResponse*) ist für die Übermittlung etwaiger Fehlerzustände das Element *ErrorMessage* enthalten. In *ErrorMessage* können folgende Fehlerzustände auftreten:

<b>Fehlercode</b>	<b>Fehlerbeschreibung</b>
<b>REFINE_OBJECTNOTFOUND</b>	Das anzureichernde Objekt konnte im eigenen Datenbestand nicht oder nicht zweifelsfrei gefunden werden.

Tabelle 177: Liste der Fehlerzustände in den Response-Strukturen der Refine-Anfragen.

#### 15.3.1. RefineResponseStructure (in Trias\_Refine.xsd)

<b>RefineResponseStructure</b>		<b>+Structure</b>	<b>Fasst die Ergebnisdaten für eine Anreicherungsanfrage zusammen.</b>
<i>IndividualRouteRefineResponse</i>	0:*	<b>+IndividualRouteRefineResponse</b>	Die angereicherten IV-Routen. Vgl. 15.3.2.
<i>LocationInformationRefineResponse</i>	0:*	<b>+LocationInformationRefineResponse</b>	Die angereicherten Ortsinformationen. Vgl. 0.
<i>StopEventRefineResponse</i>	0:*	<b>+StopEventRefineResponse</b>	Die angereicherten Abfahrtstafeln. Vgl. 15.3.4.
<i>TripInfoRefineResponse</i>	0:*	<b>+TripInfoRefineResponse</b>	Die angereicherten Fahrtinformationen. Vgl. 15.3.5.
<i>TripRefineResponse</i>	0:*	<b>+TripRefineResponse</b>	Die angereicherten Verbindungsauskünfte. Vgl. 15.3.6.

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

#### 15.3.2. IndividualRouteRefineResponseStructure (in Trias\_IndividualTrips.xsd)

<b>IndividualRouteRefineResponseStructure</b>		<b>+Structure</b>	<b>Fasst die Ergebnisdaten für eine IV-Routing-Anreicherung zusammen.</b>
<i>ErrorMessage</i>	0:*	<b>+ErrorMessage</b>	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Vgl. 7.4.2.
<i>IndividualRouteResponseContext</i>	0:1	<b>+TripResponseContext</b>	Container für Daten, die in der Antwort mehrfach auftreten und referenziert werden. Vgl. 9.3.2.
<i>IndividualRouteResult</i>	0:*	<b>+RouteResult</b>	Container für ein IV-Routing. Vgl. 17.3.2.

Tabelle 179: Beschreibung der Struktur **IndividualRouteRefineResponseStructure**.

### 15.3.3. LocationInformationRefineResponseStructure (in Trias\_Locations.xsd)

<b>LocationInformationRefineResponseStructure</b>			<b>+Structure</b>	<b>Fasst die Ergebnisdaten für eine Ortsinformation-Anreicherung zusammen.</b>
	<i>ErrorMessage</i>	0:*	<b>+ErrorMessage</b>	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Vgl. 7.4.2.
	<i>LocationResult</i>	0:1	<b>+LocationResult</b>	Container für eine Ortsinformation. Vgl. 0.

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

### 15.3.4. StopEventRefineResponseStructure (in Trias\_StopEvents.xsd)

<b>StopEventRefineResponseStructure</b>			<b>+Structure</b>	<b>Fasst die Ergebnisdaten für eine Abfahrtstafel-Anreicherung zusammen.</b>
	<i>ErrorMessage</i>	0:*	<b>+ErrorMessage</b>	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Vgl. 7.4.2.
	<i>StopEventResponseContext</i>	0:1	<b>+StopEventResponseContext</b>	Container für Daten, die in der Antwort mehrfach auftreten und referenziert werden. Vgl. 10.3.2.
	<i>StopEventResult</i>	0:*	<b>+StopEventResult</b>	Container für Abfahrtstafeln. Vgl. 10.3.3.

Tabelle 181: Beschreibung der Struktur **StopEventRefineResponseStructure**.

### 15.3.5. TripInfoRefineResponseStructure (in Trias\_TripInfo.xsd)

<b>TripInfoRefineResponseStructure</b>			<b>+Structure</b>	<b>Fasst die Ergebnisdaten für eine Fahrtinformationen-Anreicherung zusammen.</b>
	<i>ErrorMessage</i>	0:*	<b>+ErrorMessage</b>	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Vgl. 7.4.2.
	<i>TripInfoResponseContext</i>	0:1	<b>+TripInfoResponseContext</b>	Container für Daten, die in der Antwort mehrfach auftreten und referenziert werden. Vgl. 12.3.2.
	<i>TripInfoResult</i>	0:1	<b>+TripInfoResult</b>	Container für eine Fahrtinformation. Vgl. 12.3.3.

Tabelle 182: Beschreibung der Struktur **TripInfoRefineResponseStructure**.

### 15.3.6. TripRefineResponseStructure (in Trias\_Trips.xsd)

<b>TripRefineResponseStructure</b>			<b>+Structure</b>	<b>Fasst die Ergebnisdaten für eine intermodale Verbindungsauskunft-Anreicherung zusammen.</b>
	<i>ErrorMessage</i>	0:*	<b>+ErrorMessage</b>	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Vgl. 7.4.2.
	<i>TripResponseContext</i>	0:1	<b>+TripResponseContext</b>	Container für Daten, die in der Antwort mehrfach auftreten und referenziert werden. Vgl. 9.3.2.
	<i>UnknownLegRef</i>	0:*	<b>→xs:NMTOKEN</b>	Referenzen auf die Legs, die nicht in den lokalen Daten wiedergefunden werden konnten
	<i>TripResult</i>	0:1	<b>+TripResult</b>	Container für eine Verbindungsauskunft. Vgl. 9.3.3.

Tabelle 183: Beschreibung der Struktur **TripRefineResponseStructure**.

## 16. Dienst Buchungsinformation

### 16.1. 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.

In der XML-Schema-Definition *Trias\_Booking.xsd* werden Datentypen und Strukturen definiert, die für den Dienst Buchungsinformation verwendet werden.

### 16.2. Anfragestrukturen

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

#### 16.2.1. BookingInfoRequestStructure

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

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

Tabelle 184: Beschreibung der Struktur **BookingInfoRequestStructure**.

### 16.3. Antwortstrukturen

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

#### 16.3.1. BookingInfoResponseStructure

<b>BookingInfoResponseStructure</b>			+Structure	Fasst die Ergebnisdaten für eine Buchungsinformationsanfrage zusammen.
	ErrorMessage	0:*	+ErrorMessage	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
	BookingInfoResult	0:1	+BookingInfoResult	Struktur für ein Buchungsinformationsresultat. Vgl. 16.3.2.

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

In *ErrorMessage* können folgende Fehlerzustände auftreten:

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

Tabelle 186: Liste der Fehlerzustände in **BookingInfoResponse**.

### 16.3.2. BookingInfoResultStructure

<b>BookingInfoResultStructure</b>		<b>+Structure</b>	<b>Resultatstruktur für die Buchungsinformationen.</b>
	<b>BookingInfo</b>	1:*	+BookingInfo
	Extension	0:1	xs:anyType

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

## 17. Dienst IV-Routing

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

### 17.1. Einfache Typen

Folgender einfacher Typ (simple type) wird definiert:

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

Tabelle 188: Liste der einfachen Typdefinitionen in *Trias\_IndividualTrips.xsd*.

### 17.2. Anfragestrukturen

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

#### 17.2.1. IndividualRouteRequestStructure

<i>IndividualRouteRequestStructure</i>		+Structure	Fasst die Anfragedaten für ein IV-Routing zusammen.
<b>Origin</b>	1:1	+ <i>IndividualRouteLocationContext</i>	Ortsdaten für den Abfahrtsort. Vgl. 17.2.3.
<b>Destination</b>	1:1	+ <i>IndividualRouteLocationContext</i>	Ortsdaten für den Zielort. Vgl. 17.2.3.
<i>Via</i>	0:*	+ <i>Via</i>	Ein oder mehrere <i>Via</i> -Orte. Die angegebenen <i>Via</i> -Orte müssen in der vorgegebenen Reihenfolge erreicht werden. Der Server darf eine <i>Via</i> -Haltestelle durch eine äquivalente Haltestelle ersetzen. Vgl. 7.6.2.
<b>Mode</b>	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. 17.2.2.

Tabelle 189: Beschreibung der Struktur ***IndividualRouteRequestStructure***.

#### 17.2.2. IndividualTripParamStructure

<i>IndividualTripParamStructure</i>		+Structure	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>Base-TripMobilityFilter</i>	<i>NoSingleStep</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer keine Stufe bewältigen kann. Default ist <i>false</i> .
	<i>NoStairs</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer keine Treppe bewältigen kann. Default ist <i>false</i> .
	<i>NoEscalator</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer keine Rolltreppe benutzen kann. Default ist <i>false</i> .
	<i>NoElevator</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer keinen Aufzug benutzen kann. Default ist <i>false</i> .
	<i>NoRamp</i>	0:1	<i>xs:boolean</i>	Legt fest, ob der Benutzer keine Rampe bewältigen

				kann. Default ist <i>false</i> .	
<i>Base-TripPolicy</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. Default ist <i>false</i> .
		<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. Default ist <i>false</i> .
<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>Base-TripContentFilter</i>		<i>IncludeTrackSections</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat <i>TrackSection</i> -Elemente (vgl. 7.6.15) für die detaillierte geografische 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 geografische 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>IncludeEstimatedTimes</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat Informationen zur Echtzeitsituation mitausgegeben werden sollen. Default ist <i>false</i> .
		<i>IncludeSituationInfo</i>	0:1	<i>xs:boolean</i>	Legt fest, ob im Resultat textuelle Echtzeitmeldungen mitausgegeben werden sollen. Default ist <i>false</i> .
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterungen.	

Tabelle 190: Beschreibung der Struktur **IndividualTripParamStructure**.

### 17.2.3. IndividualRouteLocationContextStructure

Elemente vom Typ *IndividualRouteLocationContextStructure* werden 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.

<i>IndividualRouteLocationContextStructure</i>			+Structure	Ortsangabe für Start- oder Zielorte von IV-Routen.
	<i>LocationRef</i>	1:1	+LocationRef	Referenz auf ein Ortsobjekt. Vgl. 7.5.11.
	<i>DepArrTime</i>	0:1	xs:dateTime	Abfahrts- bzw. Ankunftszeit.

Tabelle 191: Beschreibung der Struktur *IndividualRouteLocationContextStructure*.

## 17.3. Antwortstrukturen

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

### 17.3.1. IndividualRouteResponseStructure

<i>IndividualRouteResponseStructure</i>			+Structure	Fasst die Ergebnisdaten für eine IV-Routing-Auskunft zusammen.
	<i>ErrorMessage</i>	0:*	+ErrorMessage	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. auch 7.4.2.
	<i>IndividualRouteResponseContext</i>	0:1	+TripResponseContext	Container für Daten, die in der Antwort mehrfach auftreten und referenziert werden. Vgl. 9.3.2.
	<i>RouteResult</i>	0:*	+RouteResult	Container für eine Verbindungsauskunft. Vgl. 17.3.2.

Tabelle 192: Beschreibung der Struktur *IndividualRouteResponseStructure*.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

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

Tabelle 193: Liste der Fehlerzustände in *IndividualRouteResponse*.

### 17.3.2. RouteResultStructure

<b>RouteResultStructure</b>			<b>+Structure</b>	<b>Fasst die Ergebnisdaten für eine einzelne IV-Route zusammen.</b>
	<b>ResultId</b>	<b>1:1</b>	<b>xs:NMTOKEN</b>	ID des Resultats für spätere Referenzierung bzw. für Debug-Zwecke.
	<b>ErrorMessage</b>	<b>0:*</b>	<b>+ErrorMessage</b>	Fehlermeldungen bezogen auf diese IV-Route. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. auch 7.4.2.
	<b>Route</b>	<b>1:1</b>	<b>+Route</b>	Daten zu einer IV-Route. Vgl. 17.3.3.

Tabelle 194: Beschreibung der Struktur *RouteResultStructure*.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<b>Fehlercode</b>	<b>Fehlerbeschreibung</b>
<b>ROUTE_MODEPARAMETERSIGNORED</b>	Bei dieser IV-Route wurde mindestens einer der Parameter für diesen IV-Modus ignoriert. Vgl. auch 7.3.2.
<b>ROUTE_MOBILITYFILTERIGNORED</b>	Bei dieser IV-Route wurde mindestens einer der Mobilitätsfilter ignoriert.
<b>ROUTE_BANIGNORED</b>	Bei dieser IV-Route wurde mindestens einer der Ausschlussfilter (Autobahn, Mautstraßen, Fähren) ignoriert.

Tabelle 195: Liste der Fehlerzustände in *RouteResult*.



### 17.3.3. RouteStructure

<b>RouteStructure</b>		<b>+Structure</b>	<b>Daten zu einer einzelnen IV-Route.</b>
<b>Routeld</b>	1:1	<i>xs:NMTOKEN</i>	ID der Verbindung für spätere Referenzierung bzw. für Debug-Zwecke.
<b>Duration</b>	1:1	<i>xs:duration</i>	Gesamtdauer der IV-Route.
<b>StartTime</b>	1:1	<i>xs:dateTime</i>	Startzeitpunkt der IV-Route.
<b>EndTime</b>	1:1	<i>xs:dateTime</i>	Endzeitpunkt der IV-Route.
<b>Distance</b>	0:1	<i>Distance</i>	Gesamtdistanz der IV-Route als Länge des zurückzulegenden Weges.
<b>RouteLeg</b>	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.
<b>SituationFullRef</b>	0:*	<i>+SituationFullRef</i>	Verweis auf Störungsnachrichten. Diese Nachrichten können im Element <i>IndividualRouteResponseContext</i> vom Typ <i>TripResponseContext</i> (vgl. 9.3.2) zu finden sein oder auf anderem Wege bekannt gemacht werden. Vgl. 7.8.2.
<b>Extension</b>	0:1	<i>xs:anyType</i>	Erweiterungen.

Tabelle 196: Beschreibung der Struktur **RouteStructure**

## 18. Kartendienst

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

### 18.1. Einfache Datentypen

Folgende einfache Typen (simple types) werden definiert:

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

Tabelle 197: Liste der einfachen Typdefinitionen in *Trias\_Maps.xsd*.

### 18.2. Anfragestrukturen

#### 18.2.1. 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.

<b>MapServiceRequestStructure</b>			+Structure	Fasst die Anfragedaten für den Abruf einer Karte zusammen.
<i>MapProperties</i>	<b>Aspect</b>	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. 18.2.4.
	<b>Size</b>	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. 18.2.5.
	<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.
	<b>Layer</b>	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>Opaqueness</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 198: Beschreibung der Struktur **MapServiceRequestStructure**.

### 18.2.2. 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.

<i>ImageCoordinatesRequestStructure</i>			+Structure	Fasst die Anfragedaten für den Abruf von Bildkoordinaten zusammen.
<i>MapProperties</i>	<b>Aspect</b>	1:1	+MapAspect	Tatsächlicher geografischer Ausschnitt der referenzierten Karte. Vgl. 18.2.4.
	<b>Size</b>	1:1	+MapSize	Bildgröße der referenzierten Karte. Vgl. 18.2.5.
	<b>Point</b>	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 199: Beschreibung der Struktur **ImageCoordinatesRequestStructure**.

### 18.2.3. 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 Umrechnungsfunktion. Solche geografischen Positionen werden mittels eines Elements **GeoCoordinatesRequest** vom Typ *GeoCoordinatesRequestStructure* angefordert.

<i>GeoCoordinatesRequestStructure</i>			+Structure	Fasst die Anfragedaten für den Abruf von geografischen Koordinaten zusammen.
<i>MapProperties</i>	<b>Aspect</b>	1:1	+MapAspect	Tatsächlicher geografischer Ausschnitt der referenzierten Karte. Vgl. 18.2.4.
	<b>Size</b>	1:1	+MapSize	Bildgröße der referenzierten Karte. Vgl. 18.2.5.
	<b>ImagePoint</b>	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. 18.2.6.

Tabelle 200: Beschreibung der Struktur **GeoCoordinatesRequestStructure**.

### 18.2.4. MapAspectStructure

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

Tabelle 201: Beschreibung der Struktur **MapAspectStructure**.

## 18.2.5. MapSizeStructure

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

Tabelle 202: Beschreibung der Struktur **MapSizeStructure**.

## 18.2.6. MapCoordinateStructure

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

Tabelle 203: Beschreibung der Struktur **MapCoordinateStructure**.

## 18.3. Antwortstrukturen

### 18.3.1. MapServiceResponseStructure

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

<i>MapServiceResponseStructure</i>			<i>+Structure</i>	<b>Fasst die Ergebnisdaten für eine Kartenanfrage zusammen.</b>
	<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>MapResult</i>	0:1	<i>+MapResult</i>	Ergebnis der Kartenanfrage.

Tabelle 204: Beschreibung der Struktur **MapServiceResponseStructure**.

In **ErrorMessage** können folgende Fehlerzustände auftreten:

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

Tabelle 205: Liste der Fehlerzustände in **MapServiceResponse**.

### 18.3.2. ImageCoordinatesResponseStructure

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

<b>ImageCoordinatesResponseStructure</b>		<b>+Structure</b>	<b>Fasst die Ergebnisdaten für eine Bildkoordinatenanfrage zusammen.</b>
	<i>ErrorMessage</i>	0:*	<b>+ErrorMessage</b> Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
	<i>Result</i>	0:*	<b>+ImagePointResult</b> Einzelergebnisse der Umrechnung von Geo- zu Bildkoordinaten. Es können maximal so viele Elemente sein, wie geografische Punkte in der Anfrage enthalten waren. Zur Identifizierung ist in jedem Einzelergebnis der angefragte Punkt enthalten. Vgl. 18.3.5.

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

In *ErrorMessage* können folgende Fehlerzustände auftreten:

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

Tabelle 207: Liste der Fehlerzustände in **ImageCoordinatesResponse**.

### 18.3.3. GeoCoordinatesResponseStructure

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

<b>GeoCoordinatesResponseStructure</b>		<b>+Structure</b>	<b>Fasst die Ergebnisdaten für eine Geokoordinatenanfrage zusammen.</b>
	<i>ErrorMessage</i>	0:*	<b>+ErrorMessage</b> Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
	<i>Result</i>	0:*	<b>+GeoCoordinateResult</b> Einzelergebnisse der Umrechnung von Bild- zu Geokoordinaten. Es können maximal so viele Elemente sein, wie Bildpunkte in der Anfrage enthalten waren. Zur Identifizierung ist in jedem Einzelergebnis der angefragte Punkt enthalten. Vgl. 18.3.6.

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

In *ErrorMessage* können folgende Fehlerzustände auftreten:

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

Tabelle 209: Liste der Fehlerzustände in **GeoCoordinatesResponse**.

### 18.3.4. MapResultStructure

<b>MapResultStructure</b>		<b>+Structure</b>	<b>Fasst die Daten einer generierten Karte zusammen.</b>
<b>File</b>	<b>1:1</b>	<i>xs:base64Binary</i>	Die Bilddaten der generierten Karte.
<b>ImageType</b>	<b>1:1</b>	<i>xs:string</i>	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.
<b>Aspect</b>	<b>1:1</b>	<i>+MapAspect</i>	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. 18.2.4.

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

### 18.3.5. ImagePointResultStructure

<b>ImagePointResultStructure</b>		<b>+Structure</b>	<b>Fasst die Daten einer Umrechnung von Geo- zu Bildkoordinaten zusammen.</b>
<i>ErrorMessage</i>	<b>0:*</b>	<i>+ErrorMessage</i>	Fehlermeldungen bezogen auf die Umrechnung des nachfolgenden Geokoordinaten-Paars. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
<b>Point</b>	<b>1:1</b>	<i>+GeoPosition</i>	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>	<b>0:1</b>	<i>+MapCoordinate</i>	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. 18.2.6.

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

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<b>Fehlercode</b>	<b>Fehlerbeschreibung</b>
<b>MAP_POINTNOTONMAP</b>	Der angegebene Punkt liegt außerhalb der Karte. Trotzdem konnte eine Konversion durchgeführt werden.
<b>MAP_UNSUPPORTEDPOINT</b>	Der angefragte Punkt liegt außerhalb des umrechenbaren Gebietes.

Tabelle 212: Liste der Fehlerzustände in *ImagePointResultStructure*.

### 18.3.6. GeoCoordinateResultStructure

<b>GeoCoordinateResultStructure</b>		<b>+Structure</b>	<b>Fasst die Daten einer Umrechnung von Bild- zu Geokoordinaten zusammen.</b>
	<i>ErrorMessage</i>	0:*	<b>+ErrorMessage</b> 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	<b>+GeoPosition</b> 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.
	<b>ImagePoint</b>	<b>1:1</b>	<b>+MapCoordinate</b> Bildkoordinaten, für den geografische Koordinaten berechnet werden sollten. Dies muss einer der Punkte aus der zugehörigen Dienstanfrage sein. Vgl. 18.2.6.

Tabelle 213: Beschreibung der Struktur *GeoCoordinateResultStructure*.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<b>Fehlercode</b>	<b>Fehlerbeschreibung</b>
<b>MAP_POINTNOTONMAP</b>	Der angegebene Punkt liegt außerhalb der Karte. Trotzdem konnte eine Konversion durchgeführt werden.
<b>MAP_UNSUPPORTEDPOINT</b>	Der angefragte Punkt liegt außerhalb des umrechenbaren Gebietes.

Tabelle 214: Liste der Fehlerzustände in *GeoCoordinateResultStructure*.

## 19. Dienst Schadensmeldung / Zustand von Einrichtungen

### 19.1. Beschreibung

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

In der XML-Schema-Definition *Trias\_Facilities.xsd* werden Datentypen und Strukturen definiert, die für den Dienst Schadensmeldung / Zustand von Einrichtungen verwendet werden.

### 19.2. Einfache Typen

Folgende einfache Typen (simple types) werden definiert:

Typename	Werte	Beschreibung
<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.

Tabelle 215: Beschreibung der Struktur einfacher Typen

### 19.3. Komplexe Strukturen

#### 19.3.1. FacilityStructure

<i>FacilityStructure</i>			+Structure	Beschreibung und Zustand einer Einrichtung.	
	<b>a</b>	<b>VehicleFacility</b>	-1:1	+VehicleFacility	Definition einer Fahrzeugeinrichtung. Vgl. 19.3.2.
	<b>b</b>	<b>InfrastructureFacility</b>		+InfrastructureFacility	Definition einer Infrastruktureinrichtung. Vgl. 19.3.3.
	<b>Condition</b>		1:1	+FacilityStatus	Zustand der Einrichtung. Vgl. 19.3.7.
<i>Extension</i>		0:1	<i>xs:anyType</i>	Erweiterung.	

Tabelle 216: Beschreibung der Struktur **FacilityStructure**.

#### 19.3.2. VehicleFacilityStructure

<i>VehicleFacilityStructure</i>			+Structure	Beschreibung einer Fahrzeugeinrichtung.
<i>VehicleFacilityRef</i>	:::	1:1	+VehicleFacilityRefGroup	Referenz auf eine Einrichtung. Vgl. 19.3.4.
<i>ServiceFacility</i>	:::	1:1	+ServiceFacilityGroup	Klassifizierung der Einrichtung. Vgl. 7.7.3.
<i>FacilityDescription</i>		0:*	+InternationalText	Name oder Beschreibung der Einrichtung. Vgl. 7.2.2.
<i>LocationDescription</i>		0:*	+InternationalText	Beschreibung, wo die Einrichtung zu finden ist. Vgl. 7.2.2.

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



### 19.3.3. InfrastructureFacilityStructure

<b>InfrastructureFacilityStructure</b>			<b>+Structure</b>	<b>Beschreibung einer Infrastruktureinrichtung.</b>
<i>InfrastructureFacilityRef</i>	:::	1:1	+InfrastructureFacilityRefGroup	Referenz auf eine Einrichtung. Vgl. 19.3.5.
<i>StopFacility</i>	:::	1:1	+StopFacilityGroup	Klassifizierung der Einrichtung. Vgl. 7.7.2.
	<i>FacilityDescription</i>	0:*	+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:*	+InternationalText	Beschreibung, wo die Einrichtung zu finden ist. Vgl. 7.2.2.

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

### 19.3.4. VehicleFacilityRefGroup

<b>VehicleFacilityRefGroup</b>			<b>+Group</b>	<b>Referenzierung einer Fahrzeugeinrichtung durch Referenz auf die Einrichtung selbst oder auf ein übergeordnetes Objekt.</b>
	<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>		→JourneyCode	Referenz auf Fahrt. Vgl. 7.4.1.
	<i>VehicleRef</i>		→VehicleCode	Referenz auf Fahrzeug. Vgl. 7.4.1.

Tabelle 219: Beschreibung der Gruppe **VehicleFacilityRefGroup**.

### 19.3.5. InfrastructureFacilityRefGroup

<b>InfrastructureFacilityRefGroup</b>			<b>+Group</b>	<b>Referenzierung einer Infrastruktureinrichtung durch Referenz auf die Einrichtung selbst oder auf ein übergeordnetes Objekt.</b>
	<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>StopPointRef</i>		→StopPoint	Referenz auf einen Haltepunkt. Vgl. 7.5.1.
	<i>StopPlaceRef</i>		→StopPlace	Referenz auf eine Haltestelle. Vgl. 7.5.1.
	<i>OperatorRef</i>		→OperatorCode	Referenz auf Verkehrsunternehmen. Vgl. 7.4.1.
	<i>LineRef</i>		→LineCode	Referenz auf Linie. Vgl. 7.4.1.

Tabelle 220: Beschreibung der Gruppe **InfrastructureFacilityRefGroup**.

### 19.3.6. FacilityDataFilterGroup

<b>FacilityDataFilterGroup</b>			<b>+Group</b>	<b>Eine Menge von Objektreferenzen als Filter zum Eingrenzen der in Frage kommenden Einrichtungen.</b>
	<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>OperatorCode</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>JourneyCode</i>	Referenz auf Fahrt. Vgl. 7.4.1.
	<i>VehicleRef</i>	0:*	→ <i>VehicleCode</i>	Referenz auf Fahrzeug. Vgl. 7.4.1.

Tabelle 221: Beschreibung der Gruppe **FacilityDataFilterGroup**.

### 19.3.7. FacilityStatusStructure

<b>FacilityStatusStructure</b>			<b>+Structure</b>	<b>Der Zustand einer Einrichtung.</b>
	<b>Availability</b>	1:1	<i>FacilityAvailabilityEnumeration</i>	Verfügbarkeit der Einrichtung. Vgl. 19.2
	<b>Status</b>	1:1	<i>FacilityStatusTypeEnumeration</i>	Klassifizierung des Zustands. Vgl. 19.2
	<i>StatusDescription</i>	0:*	<i>+InternationalText</i>	Beschreibung des Zustands. Vgl. 7.2.2.

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

## 19.4. Anfrage Schadensmeldung

Meldung eines Schadens an einer Einrichtung wird mittels des Elementes **FacilityStatusReport** vom Typ *FacilityStatusReportStructure* von einer Person übermittelt.

### 19.4.1. FacilityStatusReportStructure

<b>FacilityStatusReportStructure</b>			<b>+Structure</b>	<b>Fasst die Daten für eine Schadensmeldung an einer Einrichtung zusammen.</b>
<b>a</b>	<b>VehicleFacility</b>	-1:1	<i>+VehicleFacility</i>	Definition einer Fahrzeugeinrichtung. Vgl. 19.3.2.
<b>b</b>	<b>InfrastructureFacility</b>		<i>+InfrastructureFacility</i>	Definition einer Infrastruktureinrichtung. Vgl. 19.3.3.
	<b>Condition</b>	1:1	<i>+FacilityStatus</i>	Zustand der Einrichtung. Vgl. 19.3.7.
	<i>Extension</i>	0:1	<i>xs:anyType</i>	Erweiterung.

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

## 19.5. Antwort Schadensmeldung

Die Antwort auf eine Schadensmeldung (FacilityStatusReport, siehe 19.4) wird mittels eines Elements **FacilityStatusReportResponse** vom Typ *FacilityStatusReportResponseStructure* übertragen.

### 19.5.1. FacilityStatusReportResponseStructure

<i>FacilityStatusReportResponseStructure</i>		+Structure	Antwort auf die Schadensmeldung.
	<b>ErrorMessage</b>	1:1	+ErrorMessage e
			Fehlermeldung. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.

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

In *ErrorMessage* können folgende Fehlerzustände auftreten:

Fehlercode	Fehlerbeschreibung
<b>FACILITYSTATUSREPORT_FACILITYUNKNOWN</b>	Die angegebene Einrichtung ist unbekannt.
<b>FACILITYSTATUSREPORT_OWNERUNKNOWN</b>	Der angegebene Eigentümer ist unbekannt.
<b>FACILITYSTATUSREPORT_OPERATORUNKNOWN</b>	Das angegebene Verkehrsunternehmen ist unbekannt.
<b>FACILITYSTATUSREPORT_LINEUNKNOWN</b>	Die angegebene Linie ist unbekannt.
<b>FACILITYSTATUSREPORT_JOURNEYUNKNOWN</b>	Die angegebene Fahrt ist unbekannt.
<b>FACILITYSTATUSREPORT_VEHICLEUNKNOWN</b>	Das angegebene Fahrzeug ist unbekannt.
<b>FACILITYSTATUSREPORT_STOPPOINTUNKNOWN</b>	Der angegebene Haltepunkt ist unbekannt.
<b>FACILITYSTATUSREPORT_STOPPLACEUNKNOWN</b>	Die angegebene Haltestelle ist unbekannt.

Tabelle 225: Liste der Fehlerzustände in **FacilityStatusReportResponse**.

## 19.6. Anfrage Zustand von Einrichtungen

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

### 19.6.1. FacilityRequestStructure

<i>FacilityRequestStructure</i>			+Structure	Fasst die Daten für eine Schadensmeldung an einer Einrichtung zusammen.
<i>FacilityDataFilter</i>	:::	1:1	+FacilityDataFilterGroup	Objektreferenzen als Filter. Vgl. 19.3.6.
	<i>Extension</i>	0:1	xs:anyType	Erweiterung.

Tabelle 226: Beschreibung der Struktur **FacilityRequestStructure**.

## 19.7. Antwort Zustand von Einrichtungen

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

### 19.7.1. FacilityResponseStructure

<b>FacilityResponseStructure</b>		<b>+Structure</b>	<b>Antwort auf die Schadensmeldung.</b>
	<i>ErrorMessage</i>	0:*	+ <i>ErrorMessage</i> Fehlermeldung bezogen auf die Gesamtbeantwortung der Anfrage. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
	<i>FacilityResult</i>	0:*	+ <i>FacilityResult</i> Resultatstruktur. Vgl. 19.7.2.

Tabelle 227: Beschreibung der Struktur **FacilityResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<b>Fehlercode</b>	<b>Fehlerbeschreibung</b>
<b>FACILITYREQUEST_FACILITYUNKNOWN</b>	Die angegebene Einrichtung ist unbekannt.
<b>FACILITYREQUEST_OWNERUNKNOWN</b>	Der angegebene Eigentümer ist unbekannt.
<b>FACILITYREQUEST_OPERATORUNKNOWN</b>	Das angegebene Verkehrsunternehmen ist unbekannt.
<b>FACILITYREQUEST_LINEUNKNOWN</b>	Die angegebene Linie ist unbekannt.
<b>FACILITYREQUEST_JOURNEYUNKNOWN</b>	Die angegebene Fahrt ist unbekannt.
<b>FACILITYREQUEST_VEHICLEUNKNOWN</b>	Das angegebene Fahrzeug ist unbekannt.
<b>FACILITYREQUEST_STOPPOINTUNKNOWN</b>	Der angegebene Haltepunkt ist unbekannt.
<b>FACILITYREQUEST_STOPPLACEUNKNOWN</b>	Die angegebene Haltestelle ist unbekannt.

Tabelle 228: Liste der Fehlerzustände in **FacilityResponse**.

### 19.7.2. FacilityResultStructure

<b>FacilityResultStructure</b>		<b>+Structure</b>	<b>Resultatstruktur für die Facility-Zustandsanfrage.</b>
	<i>ErrorMessage</i>	0:*	+ <i>ErrorMessage</i> Fehlermeldungen bezogen auf das einzelne FacilityResult. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
	<b>Facility</b>	1:1	+ <i>Facility</i> Informationen zur Einrichtung. Vgl. 19.3.1

Tabelle 229: Beschreibung der Struktur **FacilityResultStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<b>Fehlercode</b>	<b>Fehlerbeschreibung</b>
<b>FACILITYREQUEST_STATUSNOTCONFIRMED</b>	Für die Einrichtung liegt ein Zustandsbericht vor, der aber noch nicht bestätigt ist.

Tabelle 230: Liste der Fehlerzustände in **FacilityResultStructure**.

## 20. Benachrichtigungsdienst

### 20.1. 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

- Geplante Maßnahmen, 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 0):

- 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 6 zu sehen. Um Nachrichten zu erhalten, muss ein Portalsystem ein Abonnement beim Benachrichtigungsdienst erstellen (1). Bei der Erstellung des Abonnements (SubscriptionRequest) kann angegeben werden, welche Typen von Nachrichten dem Portalsystem übermittelt werden sollen. Die Erstellung des Abonnements wird vom Benachrichtigungsdienst synchron mit einer sir:SubscriptionResponse beantwortet.

Bei jeder neuen Meldung von einem Datenlieferant (2) ermittelt der Benachrichtigungsdienst die betroffenen Abonnements und leitet die Nachrichten an das zugehörige Portalsystem weiter. (3) Eine Nachricht vom Benachrichtigungsdienst, die eine Datenlieferung (ServiceDelivery) im Rahmen eines bestehenden Abonnements überliefert (z.B. als TripMonitoringDelivery), wird vom Portalsystem synchron mit einer DataReceivedAcknowledgement-Nachricht bestätigt.

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).

Eine Aktualisierung des Abonnements durch das Portalsystem (z.B. wegen geänderter TripMonitoringParam) ist nicht vorgesehen, stattdessen muss das Abonnement abgemeldet (4) und neu eingerichtet (1) werden.

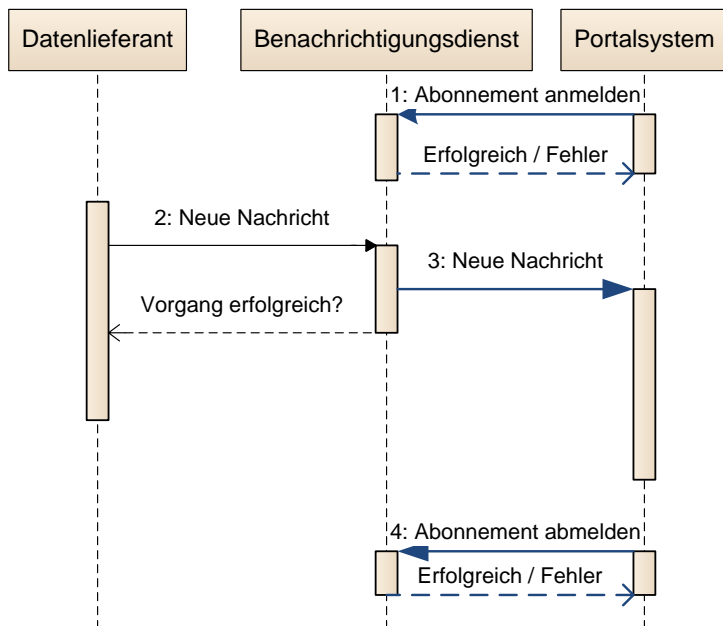


Abbildung 6: Funktionsweise des Benachrichtigungsdienstes

In den XML-Schema-Definitionen *Trias.xsd* und *Trias\_Alerts.xsd* werden Datentypen und Strukturen definiert, die für den Benachrichtigungsdienst verwendet werden.

## 20.2. Komplexe Strukturen

### 20.2.1. TripMonitoringParamStructure

<i>TripMonitoringParamStructure</i>			+Structure	Parameter für die Verbindungsüberwachung.
<i>TripMonitoringPolicy</i>	<i>Severity</i>	0:1	<i>unknown   verySlight   slight   normal   severe   verySevere   nolmpact   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. Default ist <i>false</i> .
	<i>IncludeAlternatives</i>	0:1	<i>xs:boolean</i>	Gibt an, ob direkt Alternativen mit zurückgegeben werden sollen. Default ist <i>false</i> .

Tabelle 231: 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 insbeson-

dere dann sinnvoll, wenn mehrere EKAPs zusammengeschaltet werden und dabei unterschiedliche (geografische) Zuständigkeiten haben.

## 20.3. Anfragestrukturen

### 20.3.1. TripMonitoringSubscriptionRequestStructure

<i>TripMonitoringSubscriptionRequestStructure</i>			<i>+Structure</i> (abgeleitet von <i>siri:AbstractSubscriptionStructure</i> )	<b>Einrichten einer Verbindungsüberwachung.</b>
	<b>Trip</b>	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. 20.2.1.

Tabelle 232: Beschreibung der Struktur *TripMonitoringSubscriptionRequestStructure*

## 20.4. Antwortstrukturen

### 20.4.1. TripMonitoringDeliveryStructure

<i>TripMonitoringDeliveryStructure</i>			<i>+Structure</i> (abgeleitet von <i>siri:AbstractServiceDeliveryStructure</i> )	<b>Liefert Informationen zu einer überwachten Verbindung.</b>
<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. 0.
<i>AlternativeTrip</i>	<i>TripResponse</i>	0:1	<i>+TripResponse</i>	Enthält eine Verbindungsalternative (vgl. 9.3.1).
	<i>ErrorMessage</i>	0:*	<i>+ErrorMessage</i>	Fehlermeldungen bezogen auf die Gesamtbeantwortung der Anfrage. Vgl. 7.4.2.

Tabelle 233: Beschreibung der Struktur *TripMonitoringDeliveryStructure*

In *ErrorMessage* können folgende Fehlerzustände auftreten:

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

Tabelle 234: Liste der Fehlerzustände in ***TripMonitoringDeliveryStructure***.



## 21. Personalisierungsdienst

### 21.1. 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 verwendeten 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.

In der XML-Schema-Definition *Trias\_Personalisation.xsd* werden Datentypen und Strukturen definiert, die für den Personalisierungsdienst verwendet werden.

### 21.2. 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.

### 21.2.1. Lebenszyklus eines Wertes

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

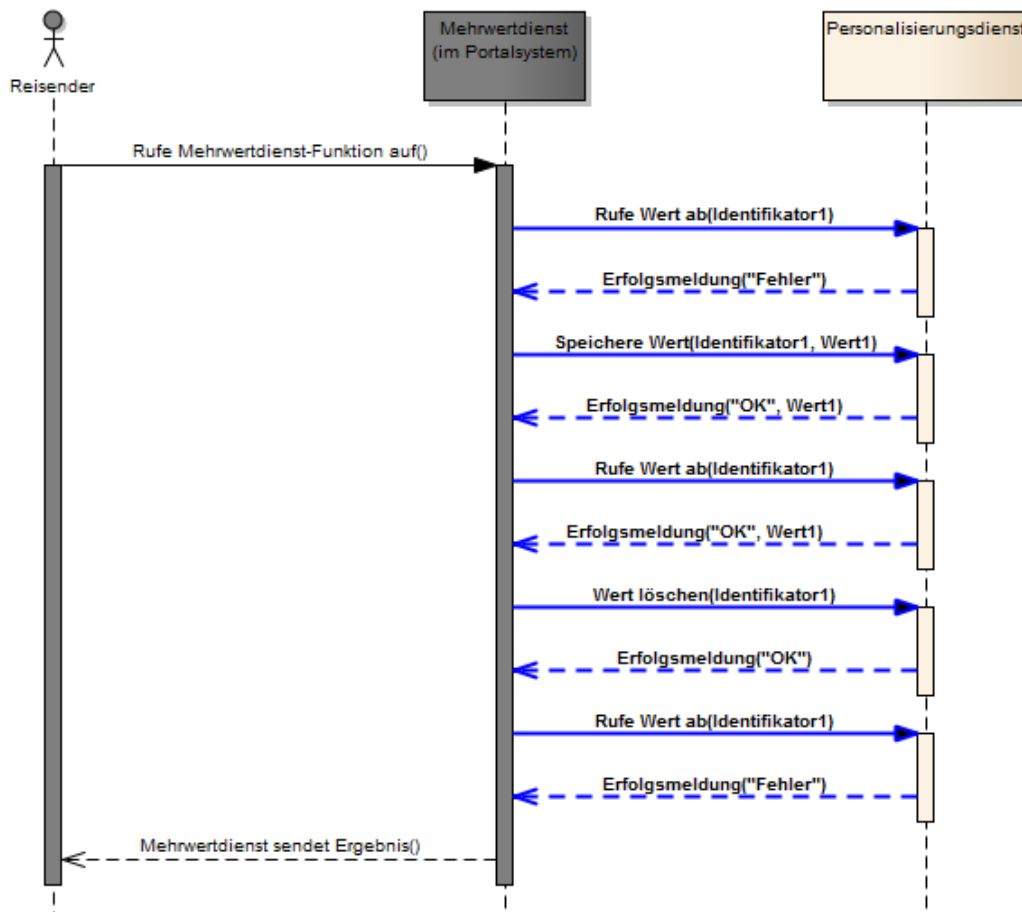


Abbildung 7: Sequenzdiagramm Lebenszyklus eines Wertes

### 21.2.2. Werteliste ermitteln

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

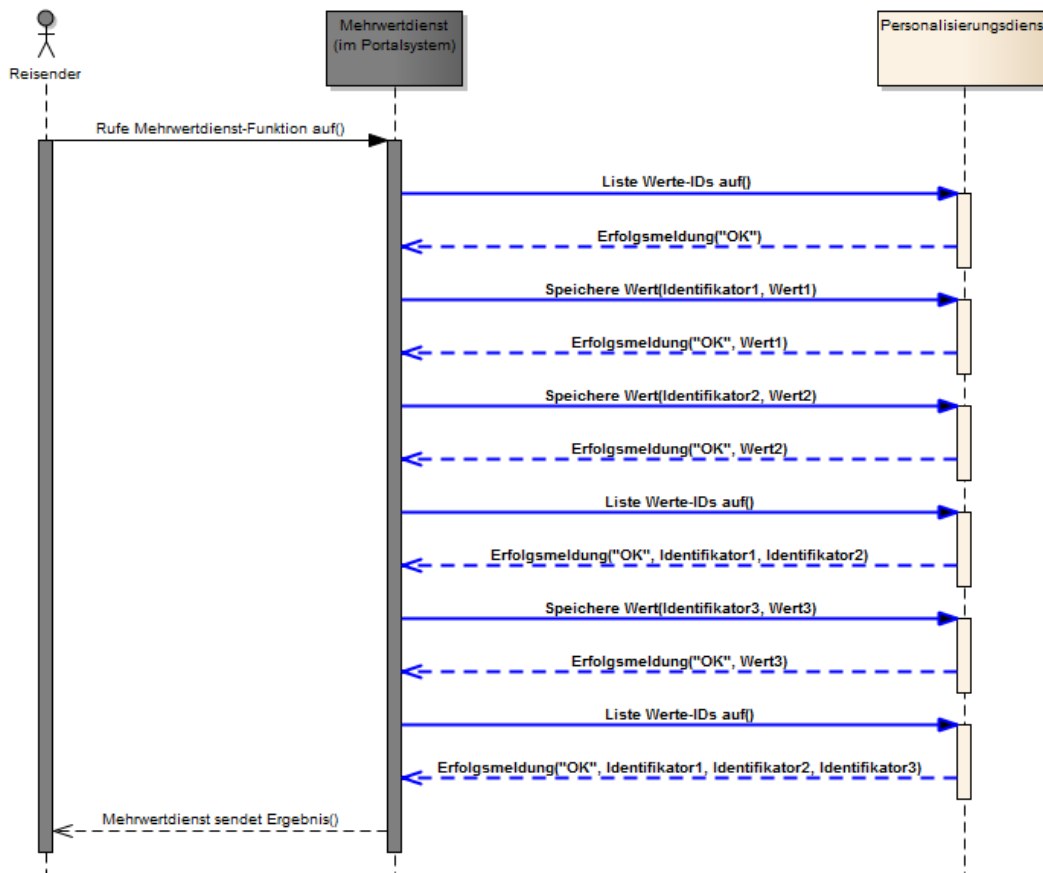


Abbildung 8: Sequenzdiagramm Werteliste ermitteln

### 21.2.3. Werte speichern und abrufen

Im folgenden Beispiel wird gezeigt, wie ein Reisender auf unterschiedliche Mehrwertdienste zugreift, welche den Personalisierungsdienst nutzen, um Konfigurationseinstellungen abzufragen 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 anbieterspezifisch 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 Hersteller stammen, o-

der weil der Entwickler von MWD 1 die in MWD 1 verwendeten Identifikatoren öffentlich bekanntgegeben hat.

Der Dienst kann bei Bedarf durch herstellereinspezifische 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.

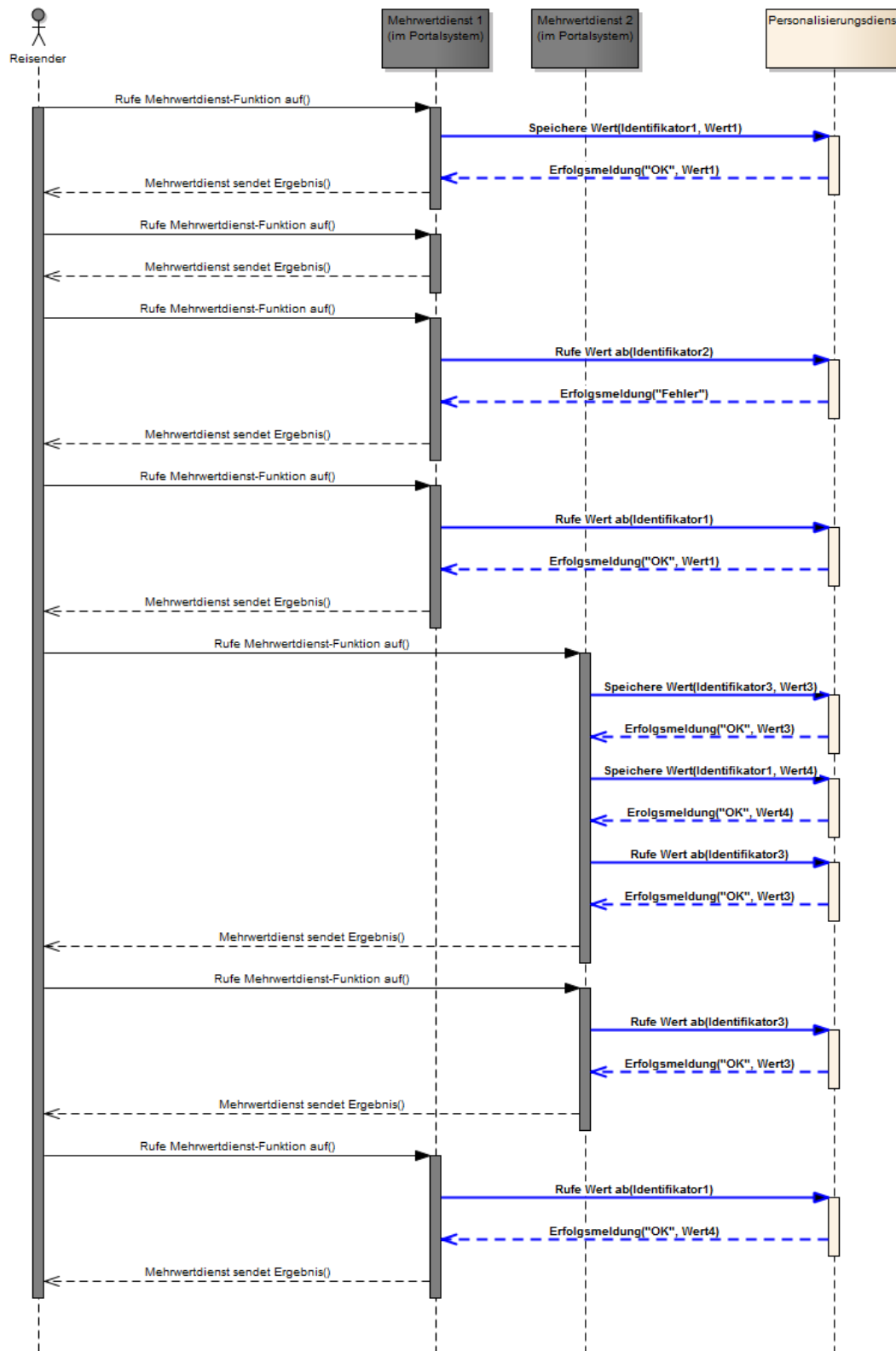


Abbildung 9: Sequenzdiagramm Personalisierung

## 21.3. Einfache Typen

Folgende einfache Typen (simple types) werden definiert:

Typname	Basistyp	Beschreibung
<i>ValueIdType</i>	<i>xs:string</i>	Identifikator eines Wertes.

Tabelle 235: Liste der einfachen Typdefinitionen in *Trias\_Personalisation.xsd*.

## 21.4. Anfragestrukturen

### 21.4.1. PersonalisationRequestStructure

<i>PersonalisationRequest</i>				+Structure	Repräsentiert eine Anfrage an den Personalisierungsdienst.
	<b>a</b>	<b>SaveValue</b>	-1:1	+ <i>PersonalisationSaveValueRequest</i>	Falls dies vorhanden ist, soll durch die Anfrage ein personalisierter Wert gespeichert werden; genauere Angaben dazu sind in dieser Struktur enthalten (vgl. 21.4.2).
	<b>b</b>	<b>RetrieveValue</b>		+ <i>PersonalisationRetrieveValueRequest</i>	Falls dies vorhanden ist, soll durch die Anfrage ein personalisierter Wert abgerufen werden; genauere Angaben dazu sind in dieser Struktur enthalten (vgl. 21.4.3).
	<b>c</b>	<b>DeleteValue</b>		+ <i>PersonalisationDeleteValueRequest</i>	Falls dies vorhanden ist, soll durch die Anfrage ein personalisierter Wert gelöscht werden; genauere Angaben dazu sind in dieser Struktur enthalten (vgl. 21.4.4).
	<b>d</b>	<b>EnumerateValue</b>		+ <i>PersonalisationEnumerateValuesRequest</i>	Falls dies vorhanden ist, sollen durch die Anfrage verfügbare personalisierte Werte abgerufen werden; genauere Angaben dazu sind in dieser Struktur enthalten (vgl. 21.4.5).

Tabelle 236: Beschreibung der Struktur *PersonalisationRequestStructure*.

### 21.4.2. PersonalisationSaveValueRequestStructure

<i>PersonalisationSaveValueRequest</i>			+Structure	Enthält genauere Angaben zum Speichern eines personalisierten Wertes.
	<b>ValueId</b>	1:1	→ <i>ValueId</i>	Der Identifikator des zu speichernden Wertes. Vgl. 21.3.
	<b>Value</b>	1:1	<i>xs:string</i>	Der zu speichernde Wert.

Tabelle 237: Beschreibung der Struktur *PersonalisationSaveValueRequestStructure*.

### 21.4.3. PersonalisationRetrieveValueRequestStructure

<i>PersonalisationRetrieveValueRequest</i>			+Structure	Enthält genauere Angaben zum Abrufen eines personalisierten Wertes.
	<b>ValueId</b>	1:1	→ <i>ValueId</i>	Der Identifikator des abzurufenden Wertes. Vgl. 21.3.

Tabelle 238: Beschreibung der Struktur *PersonalisationRetrieveValueRequestStructure*.

#### 21.4.4. PersonalisationDeleteValueRequestStructure

<i>PersonalisationDeleteValueRequest</i>			<i>+Structure</i>	Enthält genauere Angaben zum Löschen eines personalisierten Wertes.
	<i>ValueId</i>	1:1	→ <i>ValueId</i>	Der Identifikator des zu löschenden Wertes. Vgl. 21.3.

Tabelle 239: Beschreibung der Struktur *PersonalisationDeleteValueRequestStructure*.

#### 21.4.5. PersonalisationEnumerateValuesRequestStructure

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

Tabelle 240: Beschreibung der Struktur *PersonalisationEnumerateValuesRequestStructure*.

### 21.5. Antwortstrukturen

#### 21.5.1. PersonalisationResponseStructure

<i>PersonalisationResponse</i>			<i>+Structure</i>	Repräsentiert die Antwort auf eine Anfrage an den Personalisierungsdienst.
	<i>ErrorMessage</i>	0:*	<i>+ErrorMessage</i>	Enthält etwaige Fehlermeldungen, die die generelle Verarbeitung der Nachricht durch einen Personalisierungsdienst betreffen.
<b>a</b>	<b>SaveValue</b>	-1:1	<i>+PersonalisationSaveValueResponse</i>	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. 21.5.2).
	<b>RetrieveValue</b>		<i>+PersonalisationRetrieveValueResponse</i>	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. 21.5.3).
	<b>DeleteValue</b>		<i>+PersonalisationDeleteValueResponse</i>	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. 21.5.4).
	<b>EnumerateValue</b>		<i>+PersonalisationEnumerateValuesResponse</i>	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. 0).

Tabelle 241: Beschreibung der Struktur *PersonalisationResponseStructure*.

#### 21.5.2. PersonalisationSaveValueResponseStructure

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

Tabelle 242: Beschreibung der Struktur *PersonalisationSaveValueResponseStructure*.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<b>Fehlercode</b>	<b>Fehlerbeschreibung</b>
<b>PERSONALISATIONSVALUEREQUEST_INVALID_ID</b>	Der angegebene Identifikator hat ein ungültiges Format.

Tabelle 243: Liste der Fehlerzustände in **PersonalisationSaveValueResponseStructure**.

### 21.5.3. PersonalisationRetrieveValueResponseStructure

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

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

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<b>Fehlercode</b>	<b>Fehlerbeschreibung</b>
<b>PERSONALISATIONRETRIEVEVALUEREQUEST_UNKNOWN_ID</b>	Zum angegebenen Identifikator ist im Personalisierungsdienst kein Wert abgelegt.

Tabelle 245: Liste der Fehlerzustände in **PersonalisationRetrieveValueResponseStructure**.

### 21.5.4. PersonalisationDeleteValueResponseStructure

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

Tabelle 246: Beschreibung der Struktur **PersonalisationDeleteValueResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<b>Fehlercode</b>	<b>Fehlerbeschreibung</b>
<b>PERSONALISATIONDELETEVALUEREQUEST_UNKNOWN_ID</b>	Zum angegebenen Identifikator ist im Personalisierungsdienst kein Wert abgelegt.

Tabelle 247: Liste der Fehlerzustände in **PersonalisationDeleteValueResponseStructure**.

## 21.5.5. PersonalisationEnumerateValuesResponseStructure

<b>PersonalisationEnumerateValuesResponse</b>			<b>+Structure</b>	<b>Enthält genauere Angaben zur Auflistung aller gespeicherten personalisierten Werte.</b>
	<i>ErrorMessage</i>	0:*	<b>+ErrorMessage</b>	Gibt an, ob der Abruf erfolgreich war.
	<i>ValueId</i>	0:*	<b>→ValueId</b>	Falls der Abruf erfolgreich war, ist für jeden gespeicherten Wert ein solches Element vorhanden, welches jeweils einen Identifikator eines gespeicherten Wertes enthält. Vgl. 21.3.

Tabelle 248: Beschreibung der Struktur **PersonalisationEnumerateValuesResponseStructure**.

## 22. Dienst Fahrzeuginformationen

### 22.1. 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.

In der XML-Schema-Definition *Trias\_VehicleInterface.xsd* werden Datentypen und Strukturen definiert, die für den Dienst Fahrzeuginformationen verwendet werden.

### 22.2. Anfragestrukturen

#### 22.2.1. VehicleDataRequestStructure

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

Tabelle 249: Beschreibung der Struktur **VehicleDataRequestStructure**.

### 22.3. Antwortstrukturen

#### 22.3.1. VehicleDataResponseStructure

<b>VehicleDataResponseStructure</b>			<b>+Structure</b>	<b>Fasst die Ergebnisdaten für eine Abfrage der Fahrzeugdaten zusammen</b>
	<b>VehicleCode</b>	<b>1:1</b>	<b>→VehicleCode</b>	Eindeutige Fahrzeug-ID, mit der eine Referenzierung auf die EKAP-Daten ermöglicht werden soll, vgl. 7.4.1.
	<i>VehicleStatus</i>	0:1	<b>+VehicleStatus</b>	Informationen, die den Zustand des Fahrzeugs betreffen, siehe 22.3.2.
	<i>VehicleActivity</i>	0:1	<b>+VehicleActivity</b>	Informationen, die die Aktivität des Fahrzeugs betreffen, siehe 22.3.3.

Tabelle 250: Beschreibung der Struktur **VehicleDataResponseStructure**.



### 22.3.2. VehicleStatusStructure

<b>VehicleStatusStructure</b>			<b>+Structure</b>	<b>Informationen, die den Zustand des einzelnen Fahrzeugs betreffen.</b>
	<i>DoorState</i>	0:1	<i>DoorsOpen</i>   <i>All-DoorsClosed</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. Default ist <i>false</i> .
	<i>InPanic</i>	0:1	<i>xs:boolean</i>	Information darüber, ob ein Sicherheitsalarm ausgelöst worden ist. Default ist <i>false</i> .
	<i>VehicleTypeRef</i>	0:1	→ <i>VehicleType</i>	Informationen über den Fahrzeugtyp und damit der Ausstattung des Fahrzeugs, vgl. 7.4.1.
<i>ServiceFacility</i>	⋮	0:1	+ <i>siri:ServiceFacilityGroup</i>	Klassifizierung der Ausstattungsmerkmale. Vgl. 7.7.3.

Tabelle 251: Beschreibung der Struktur **VehicleStatusStructure**.

### 22.3.3. VehicleActivityStructure

<b>VehicleActivityStructure</b>			<b>+Structure</b>	<b>Informationen, die die Aktivität des einzelnen Fahrzeugs betreffen.</b>
	<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</i>   <i>offroute</i>   <i>unknown</i>	Information darüber, ob das Fahrzeug sich auf dem geplanten Linienweg befindet oder nicht.
	<i>JourneyMode</i>	0:1	<i>NoTrip</i>   <i>AdditionalTrip</i>   <i>ServiceTrip</i>	Information über die Art der Fahrt, die das Fahrzeug durchführt (Planfahrt, Verstärkerfahrt u. ä.).
<i>ServicePatternPosition</i>	<i>StopSequence</i>	0:*	+ <i>StopSequence</i>	Informationen zur Haltepunktfolge, die benötigt wird, um bspw. eine Perlschnur darzustellen, siehe 22.3.4.
	<i>CurrentStopIndex</i>	0:1	<i>xs:int</i>	Index über den nächsten angefahrenen Haltepunkt in der Haltepunktfolge.
	<i>LocationState</i>	0:1	<i>AfterStop</i>   <i>AtStop</i>   <i>BeforeStop</i>   <i>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</i>   <i>left</i>   <i>right</i>   <i>unknown</i>	Information über die Ausstiegsseite am nächsten Haltepunkt.

Tabelle 252: Beschreibung der Struktur **VehicleActivityStructure**.

### 22.3.4. StopSequenceStructure

<b>StopSequenceStructure</b>			<b>+Structure</b>	<b>Informationen zur Haltepunktfolge.</b>
	<b>StopPoint</b>	<b>2:*</b>	+ <i>StopInformation</i>	Informationen über den Haltepunkt, siehe 0.

Tabelle 253: Beschreibung der Struktur **StopSequenceStructure**.

## 22.3.5. StopInformationStructure

<b>StopInformationStructure</b>		<b>+Structure</b>		<b>Informationen über den einzelnen Haltepunkt.</b>
	<b>StopIndex</b>	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.
	<b>StopName</b>	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 22.3.6.
	<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 22.3.10.
	<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 0.

Tabelle 254: Beschreibung der Struktur **StopInformationStructure**.

## 22.3.6. DisplayContentStructure

<b>DisplayContentStructure</b>		<b>+Structure</b>		<b>Information über die Zusammensetzung des einzelnen Zieltextinhalts.</b>
	<i>Line</i>	0:1	+ <i>LineInformation</i>	Informationen zur Bezeichnung der Linie, siehe 22.3.7.
	<b>Destination</b>	1:1	+ <i>Destination</i>	Informationen über den Inhalt des Zieltextes, siehe 22.3.8.
	<i>Via</i>	0:*	+ <i>ViaPoint</i>	Informationen über Zwischenhaltestellen, siehe 22.3.9.
	<i>AdditionalInformation</i>	0:*	+ <i>InternationalText</i>	Zusatzinformationen wie „Eilkurs“, „Entlastungsbus“ etc.
<i>DisplayPolicy</i>	<b>PeriodDuration</b>	1:1	<i>xs:duration</i>	Information über die Periodendauer bei Wechsel zwischen verschiedenen Anzeigeeinhalten.
	<b>Duration</b>	1:1	<i>xs:duration</i>	Information über die Anzeigedauer dieses Anzeigeeinhalts innerhalb einer Anzeigeperiode (bei Wechsel zwischen verschiedenen Anzeigeeinhalten).

Tabelle 255: Beschreibung der Struktur **DisplayContentStructure**.

## 22.3.7. LineInformationStructure

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

Tabelle 256: Beschreibung der Struktur **LineInformationStructure**

### 22.3.8. DestinationStructure

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

Tabelle 257: Beschreibung der Struktur **DestinationStructure**

### 22.3.9. ViaPointStructure

<b>ViaPointStructure</b>			<b>+Structure</b>	<b>Informationen über Zwischenhaltestellen.</b>
	<b>ViaPointRef</b>	<b>1:1</b>	<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>+International Text</i>	Bezeichnung des Zwischenhaltepunktes.
	<i>PlaceShortName</i>	0:*	<i>+International Text</i>	Kurzbezeichnung des Zwischenhaltepunktes.
	<i>ViaPointDisplayPriority</i>	0:1	<i>xs:nonNegativeInteger</i>	Anzeigepriorität des Zwischenhaltepunktes.

Tabelle 258: Beschreibung der Struktur **ViaPointStructure**

### 22.3.10. AnnouncementStructure

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

Tabelle 259: Beschreibung der Struktur **AnnouncementStructure**

### 22.3.11. ConnectionStructure

<b>ConnectionStructure</b>		<b>+Structure</b>	<b>Information über die einzelnen Anschlüsse an einer Haltestelle inkl. Nachbarhaltestellen.</b>
<b>ConnectionRef</b>	1:1	<i>xs:NMTOKEN</i>	Haltepunktsbezogener Index über alle verfügbaren Anschlüsse.
<b>ConnectionType</b>	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.
<b>DisplayContent</b>	1:1	<i>+DisplayContent</i>	Information über die Zusammensetzung des Zieltextinhalts, siehe 22.3.6.
<b>Platform</b>	1:1	<i>xs:string</i>	Abfahrtsstelle des Abbringers.
<i>WalkDuration</i>	0:1	<i>xs:duration</i>	Durchschnittliche Fußwegzeit zum Abfahrtshaltepunkt des Anschlusses.
<b>ConnectionState</b>	1:1	<i>ConnectionOK / Connection-Broken / 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 22.3.12.
<i>ExpectedDepartureTime</i>	0:1	<i>xs:dateTime</i>	erwartete Abfahrtszeit des Abbringers.

Tabelle 260: Beschreibung der Struktur **ConnectionStructure**

### 22.3.12. VehicleTypeStructure

<b>VehicleTypeStructure</b>		<b>+Structure</b>	<b>Information über die Art eines Fahrzeugs.</b>
<b>VehicleTypeRef</b>	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 261: Beschreibung der Struktur **VehicleTypeStructure**

## 23. Dienst Fahrzeugaktionen

### 23.1. Beschreibung

Dieser Dienst dient der Übertragung eines Interaktionswunsches (z.B. eines Haltewunsches) von einer mobilen Applikation eines Fahrgasts an das Fahrzeug.

In der XML-Schema-Definition *Trias\_VehicleInterface.xsd* werden Datentypen und Strukturen definiert, die für den Dienst Fahrzeugaktionen verwendet werden.

### 23.2. Anfragestrukturen

#### 23.2.1. VehicleInteractionRequestStructure

<i>VehicleInteractionRequestStructure</i>			+Structure	Anfrage, die eine Interaktion mit einem Fahrzeug auslösen soll.
	<b>a</b>	<b>ActivateOutsideSpeakerRequest</b>	-1:1	+ActivateOutsideSpeakerRequest Anfrage, um die Außenlautsprecher eines Fahrzeugs zu aktivieren. Vgl. 23.2.2.
	<b>b</b>	<b>StopRequestRequest</b>		+StopRequestRequest Anfrage, um einen Haltewunsch an das Fahrzeug abzusetzen. Vgl. 23.2.3.

Tabelle 262: Beschreibung der Struktur *VehicleInteractionRequestStructure*

#### 23.2.2. ActivateOutsideSpeakerRequestStructure

<i>ActivateOutsideSpeakerRequestStructure</i>			+Structure	Anfrage zur Aktivierung der Fahrzeugaußenlautsprecher.
	<i>ActivateOutsideSpeaker</i>	0:1	<i>xs:boolean</i>	Gibt an, ob der Außenlautsprecher aktiviert werden soll.

Tabelle 263: Beschreibung der Struktur *ActivateOutsideSpeakerRequestStructure*

#### 23.2.3. StopRequestRequestStructure

<i>StopRequestRequestStructure</i>			+Structure	Fasst die Informationen über die Anfragestruktur bei einem Haltewunsch an das Fahrzeug zusammen.
	<b>StopRef</b>	1:1	→ <i>StopPoint</i>	Referenz auf die gewünschte Ausstiegshaltestelle, vgl. 7.5.1.
	<i>StopName</i>	0:*	+ <i>InternationalText</i>	Bezeichner der Ausstiegshaltestelle.
	<i>Intention</i>	0:1	<i>Boarding</i>   <i>Alighting</i>	Gibt an zu welchem Zweck der Haltewunsch ausgelöst wurde. Default ist <i>Alighting</i> .
	<i>AssistanceRequired</i>	0:1	<i>xs:boolean</i>	Fahrgast benötigt Hilfe beim Ein-/Ausstieg. Default ist <i>false</i> .
<i>Pas-senger-Profile</i>	<i>WheelchairUser</i>	0:1	<i>xs:boolean</i>	Fahrgast nutzt einen Rollstuhl. Default ist <i>false</i> .
	<i>WalkingFrame</i>	0:1	<i>xs:boolean</i>	Fahrgast nutzt einen Rollator. Default ist <i>false</i> .
	<i>WalkingStick</i>	0:1	<i>xs:boolean</i>	Fahrgast nutzt einen Gehstock. Default ist <i>false</i> .
	<i>WalkingImpaired</i>	0:1	<i>xs:boolean</i>	Fahrgast ist gehbehindert. Default ist <i>false</i> .
	<i>Pram</i>	0:1	<i>xs:boolean</i>	Fahrgast führt einen Kinderwagen mit sich. Default ist <i>false</i> .
	<i>HeavyLuggage</i>	0:1	<i>xs:boolean</i>	Fahrgast führt schweres Gepäck mit sich. Default ist <i>false</i> .
	<i>VisuallyImpaired</i>	0:1	<i>xs:boolean</i>	Fahrgast ist sehbeeinträchtigt. Default ist <i>false</i> .
	<i>HearingImpaired</i>	0:1	<i>xs:boolean</i>	Fahrgast ist hörbeeinträchtigt. Default ist <i>false</i> .
	<i>ReadingImpaired</i>	0:1	<i>xs:boolean</i>	Fahrgast hat Schwierigkeiten mit dem Leseverständnis. Default ist <i>false</i> .

Tabelle 264: Beschreibung der Struktur *StopRequestRequestStructure*

## 23.3. Antwortstrukturen

### 23.3.1. VehicleInteractionResponseStructure

<b>VehicleInteractionResponseStructure</b>			<b>+Structure</b>	<b>Antwort des Fahrzeugs auf einen Interaktionswunsch.</b>
	<b>a</b>	<b>ActivateOutsideSpeakerResponse</b>	<b>-1:1</b>	<b>+ActivateOutsideSpeakerResponse</b> Fahrzeugantwort auf die Aktivierung der Außenlautsprecher. Vgl. 23.3.2.
	<b>b</b>	<b>StopRequestResponse</b>		<b>+StopRequestResponse</b> Fahrzeugantwort auf einen Haltewunsch. Vgl. 23.3.3.

Tabelle 265: Beschreibung der Struktur **VehicleInteractionResponseStructure**

### 23.3.2. ActivateOutsideSpeakerResponseStructure

<b>ActivateOutsideSpeakerResponseStructure</b>			<b>+Structure</b>	<b>Anfrage zur Aktivierung der Fahrzeugaußenlautsprecher.</b>
	<b>OutsideSpeakerActivated</b>	<b>1:1</b>	<b>xs:boolean</b>	Gibt an, ob die Außenlautsprecher aktiviert wurden.

Tabelle 266: Beschreibung der Struktur **ActivateOutsideSpeakerResponseStructure**

### 23.3.3. StopRequestResponseStructure

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

Tabelle 267: Beschreibung der Struktur **StopRequestResponseStructure**

## 24. Dienst Diensteregister

### 24.1. Beschreibung

Der Dienst Diensteregister führt Buch über verfügbare TRIAS-Dienste.

In der XML-Schema-Definition *Trias\_ServiceRegister.xsd* werden Datentypen und Strukturen definiert, die für den Dienst Diensteregister verwendet werden.

### 24.2. Einfache Typen

Folgende einfache Typen (simple types) werden definiert:

Typname	Basistype und Werte	Typbeschreibung
<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   IndividualRoutes   IndividualRoutesRefine   Locations   LocationsRefine   Maps   Positioning   ServiceRegister   StopEvents   StopEventsRefine   TripInfo   TripInfoRefine   Trips   TripsRefine</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 268: Beschreibung von einfachen Typen

### 24.3. Anfragestrukturen

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

#### 24.3.1. 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 Anfragedaten an das Diensteregister zusammen.
	<b>a</b>	<b><i>RegisterRequest</i></b>	-1:1	+ <i>ServiceRegisterRequest</i> Anfrage, um eine Dienstinanz im Dienstregister anzumelden. Vgl. 24.3.3.
	<b>b</b>	<b><i>UpdateRequest</i></b>		+ <i>ServiceRegisterUpdateRequest</i> Anfrage, um eine Dienstinanz im Dienstregister zu aktualisieren. Vgl. 24.3.4.
	<b>c</b>	<b><i>LookupRequest</i></b>		+ <i>ServiceRegisterLookupRequest</i> Anfrage, um nach passenden Diensten im Dienstregister zu suchen. 24.3.5.
	<b>d</b>	<b><i>UnregisterRequest</i></b>		+ <i>ServiceRegisterUnregisterRequest</i> Anfrage, um eine Dienstinanz im Dienstregister zu löschen. Vgl. 24.3.6.
		<i>Params</i>	0:1	+ <i>ServiceRegisterParam</i> Anfrageparameter. Vgl. 24.3.2.

Tabelle 269: Beschreibung der Struktur ***ServiceRegisterRequestStructure***.

### 24.3.2. ServiceRegisterParamStructure

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

Tabelle 270: Beschreibung der Struktur **ServiceRegisterParamStructure**.

### 24.3.3. ServiceRegisterRegisterRequestStructure

<b>ServiceRegisterRegisterRequestStructure</b>		<b>+Structure</b>	<b>Anfrage an das Diensteregister, um einen Dienst anzumelden.</b>
<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>ServiceAddress</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 271: Beschreibung der Struktur **ServiceRegisterRegisterRequestStructure**.

### 24.3.4. ServiceRegisterUpdateRequestStructure

<b>ServiceRegisterUpdateRequestStructure</b>		<b>+Structure</b>	<b>Anfrage an das Diensteregister, um die Eintragungen zu einem Dienst zu aktualisieren.</b>
	<b>ServiceId</b>	<b>1:1</b>	<i>TriasServiceId</i> ID des Dienstes, 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>ServiceAddress</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 272: Beschreibung der Struktur **ServiceRegisterUpdateRequestStructure**.



### 24.3.5. ServiceRegisterLookupRequestStructure

<b>ServiceRegisterLookupRequestStructure</b>		+Structure	<b>Anfrage an das Diensteregister, um passende Dienste zu finden.</b>	
	<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>ServiceAddress</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 273: Beschreibung der Struktur **ServiceRegisterLookupRequestStructure**.

### 24.3.6. ServiceRegisterUnregisterRequestStructure

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

Tabelle 274: Beschreibung der Struktur **ServiceRegisterUnregisterRequestStructure**.

## 24.4. Antwortstrukturen

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

### 24.4.1. ServiceRegisterResponseStructure

<b>ServiceRegisterResponseStructure</b>		+Structure	<b>Fasst die Ergebnisdaten für eine Dienstregisteranfrage zusammen.</b>	
	<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>ServiceRegisterResult</i>	0:1	<i>+ServiceRegisterResult</i>	Struktur für ein Dienstregisterresultat. Vgl. 24.4.2.

Tabelle 275: Beschreibung der Struktur **ServiceRegisterResponseStructure**.

In *ErrorMessage* können folgende Fehlerzustände auftreten:

<b>Fehlercode</b>	<b>Fehlerbeschreibung</b>
<b>SERVICEREGISTER_SERVICEIDUNKNOWN</b>	Die Anfrage an das Diensteregister enthält eine unbekannte Service-ID.
<b>SERVICEREGISTER_NOMATCH</b>	Die Suchanfrage an das Diensteregister liefert keine Übereinstimmung.
<b>SERVICEREGISTER_TOOMANYMATCHES</b>	Die Suchanfrage an das Diensteregister liefert zu viele Übereinstimmungen

Tabelle 276: Liste der Fehlerzustände in **ServiceRegisterResponse**.

## 24.4.2. ServiceRegisterResultStructure

<b>ServiceRegisterResultStructure</b>			<b>+Structure</b>	<b>Resultatstruktur für die Dienstregisteranfrage.</b>
	<b>ResultId</b>	1:1	xs:NMTOKEN	ID des Resultats für spätere Referenzierung.
	<b>ErrorMessage</b>	0:*	+ErrorMessage	Fehlermeldungen bezogen auf die Anfrage an das Dienstregister. Siehe die nachstehende Tabelle für mögliche Werte. Vgl. 7.4.2.
<b>a</b>	<b>RegisterResponse</b>	-1:1	+ServiceRegisterRegisterResponse	Antwort zur Dienstregistrierung. Vgl. 24.4.3.
<b>b</b>	<b>UpdateResponse</b>		+ServiceRegisterUpdateResponse	Antwort zur Dienstaktualisierung. Vgl. 24.4.4.
<b>c</b>	<b>LookupResponse</b>		+ServiceRegisterLookupResponse	Antwort zur Suche nach passenden Diensten. Vgl. 24.4.5.
<b>d</b>	<b>UnregisterResponse</b>		+ServiceRegisterUnregisterResponse	Antwort zur Dienstlöschung. Vgl. 24.4.6.

Tabelle 277: Beschreibung der Struktur **ServiceRegisterResultStructure**.

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

## 24.4.3. ServiceRegisterRegisterResponseStructure

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

Tabelle 278: Beschreibung der Struktur **ServiceRegisterRegisterResponseStructure**.

## 24.4.4. ServiceRegisterUpdateResponseStructure

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

Tabelle 279: Beschreibung der Struktur **ServiceRegisterUpdateResponseStructure**.

## 24.4.5. ServiceRegisterLookupResponseStructure

<b>ServiceRegisterLookupResponseStructure</b>			<b>+Structure</b>	<b>Antwort auf eine Dienstsuche.</b>
	<b>Service</b>	1:*	TriasService	Ein oder mehrere Dienste, die auf die Anfragekriterien passen. Vgl. 24.4.7.

Tabelle 280: Beschreibung der Struktur **ServiceRegisterLookupResponseStructure**.

## 24.4.6. ServiceRegisterUnregisterResponseStructure

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

Tabelle 281: Beschreibung der Struktur **ServiceRegisterUnregisterResponseStructure**.

### 24.4.7. TriasServiceStructure

<i>TriasServiceStructure</i>			<i>+Structure</i>	Definition einer Instanz eines TRIAS-Dienstes.
	<b>ServiceId</b>	1:1	<i>TriasServiceId</i>	ID der Instanz.
<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>ServiceAddress</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 282: Beschreibung der Struktur ***TriasServiceStructure***.

## 25. 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-Verfahrens<sup>8</sup> empfohlen. Jedoch können auch andere Verfahren zum Einsatz kommen. Die Kommunikationspartner müssen sich dazu bilateral abstimmen.

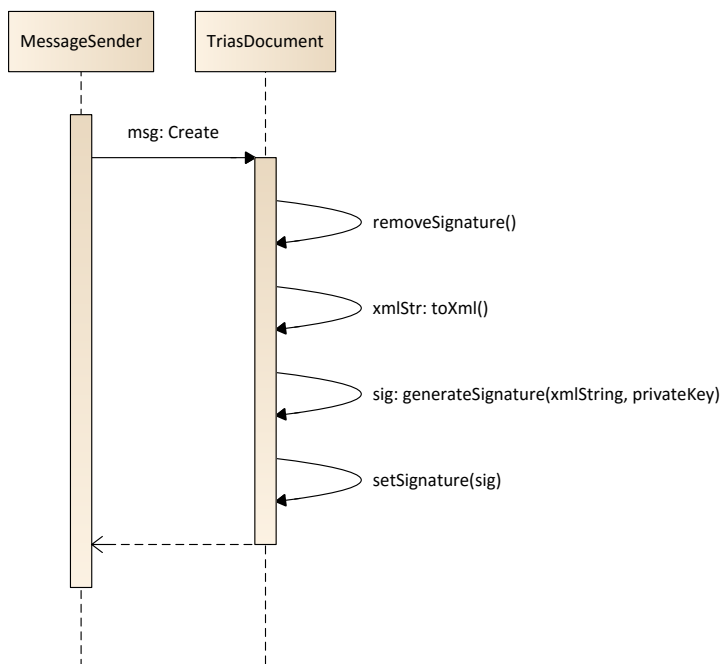


Abbildung 10: Erzeugung einer Nachrichtensignatur

<sup>8</sup> Digital Signature Standard: [http://csrc.nist.gov/publications/fips/fips186-3/fips\\_186-3.pdf](http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf)

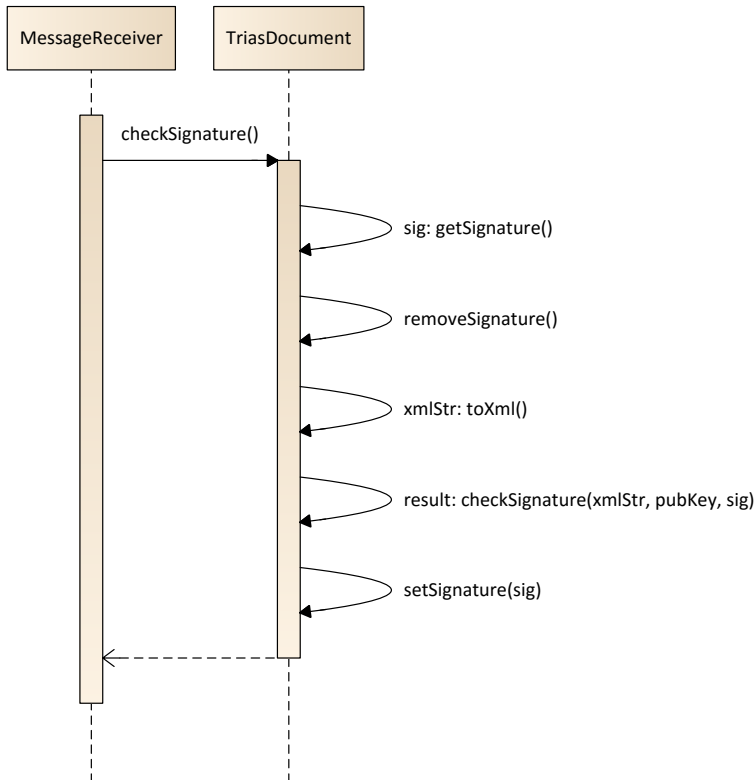


Abbildung 11: Überprüfung einer Signatur

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 Normalform<sup>9</sup> 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 10 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 11 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.

<sup>9</sup> RFC 3076: <http://www.ietf.org/rfc/rfc3076.txt>

## **26. Normative Referenzen**

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

## **27. Begriffe**

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

## 28. Versionshistorie

### 28.1. Version 1.1 (Schrift und Schema)

#### 28.1.1. Funktionale Erweiterungen

- Schema-Version im Wurzelement auf „1.1“ erhöht.
- Neue Struktur *WebLinkStructure* (Link + Beschreibung) definiert als Ersatz für Elemente vom Typ *xs:anyUri*, die nur einen Link ohne Beschreibung darstellen.
- In *Trias\_FaresSupport.xsd* eine neue Struktur *FareZoneStructure* aufgenommen, um zu den Referenzen auf Tarifzonenobjekte auch Tarifzonenbezeichnungen angeben zu können. Wird z.B. in *PassedZones* oder *ZonesAlreadyPaid* verwendet.
- Dem Element *FaresAuthorityRef* wird ein neues Element *FaresAuthorityText* beiseite gestellt, um Tarifverantwortliche mit Bezeichnungen benennen zu können.
- In *TicketStructure* weitere Elemente *ValidityDuration*, *ValidityDurationText*, *ValidityFareZones* und *ValidityAreaText*.
- *StaticFaresRequestStructure* bietet nun die Möglichkeit, eine TicketID anzugeben, um Informationen speziell zu diesem Ticket abzurufen.
- *FaresPassengerStructure* um *ZonesAlreadyPaid* und *OwnedTickets* erweitert, um die bereits vorhandenen Fahrberechtigungen modellieren zu können.
- POI-Kategorien (*POICategory*) als Liste von Key-Value-Paaren (Attribute aus Open Street Map) eingeführt. Ein POI kann einer oder mehreren POI-Kategorien zugeordnet werden. Die Suche nach POIs kann nach diesen POI-Kategorien gefiltert werden (in *LocationParamStructure*).
- Neues Element *OperatorFilter* in *LocationParamStructure* aufgenommen.
- Neues Element *SharingService* zur Modellierung von Fahrzeugverleihanbietern (Car-Sharing, Bike-Sharing) in *ContinuousServiceStructure* aufgenommen.
- In der *TripRequestStructure* werden die Elemente *Origin* und *Destination* nun auch mehrfach zugelassen.

#### 28.1.2. Technische Ergänzungen/Korrekturen

- Versionsnummern aus den Dateinamen der Schemadateien entfernt.
- Mehrfache import-Anweisungen aus dem siri-Namensraum pro Schemadatei entfernt. Stattdessen Import der SIRI-Haupt-Schemadatei.
- Einheitliche Vergabe des TRIAS-Namensraums „[www.vdv.de/trias](http://www.vdv.de/trias)“.
- In *NavigationSectionStructure* Korrektur der Schreibung von *Manoeuvre*.
- Im Benachrichtigungsdienst (*TripMonitoringPolicyGroup*) jeweils den Defaultwert *false* für die Elemente *AcceptThirdPartyInformation* und *IncludeAlternatives* aufgenommen.
- Die beiden unbenutzten Schemadateien *Trias\_Authorisation.xsd* und *Trias\_PushToDevice.xsd* entfernt. Das waren Relikte aus dem Forschungsprojekt IP-KOM-ÖV, die nicht mehr benötigt werden.
- Tippfehler im Element *IndividualTransportOptions* (sic!) in der *LocationContextStructure* korrigiert.
- In *DatedJourneyStructure* und *ContinuousServiceStructure* sind *OriginStopPointRef* bzw. *DestinationStopPointRef* nun optional, wenn man die *ServiceOriginGroup* bzw. *ServiceDestinationGroup* verwenden möchte.

- Weitere Fehlercodes *TRIASGENERIC\_ERROR*, *TRIASGENERIC\_SERVICENOTSUPPORTED*, *TRIASGENERIC\_REQUESTNOTSUPPORTED* und *TRIASGENERIC\_FEATURENOTSUPPORTED* definiert.
- *Trias\_JourneySupport.xsd:DatedCallAtLocationStructure->ServiceDeparture* steht `<xs:documentation>Arrival times of the service at this stop.</xs:documentation>`  
Da es das *ServiceDeparture*-Element ist, steht nun stattdessen "Departure times".

### 28.1.3. Dokumentarische Korrekturen

- In *TicketStructure* die Dokumentation der Elemente *InfoUrl* und *SaleUrl* ergänzt.
- In *AddressStructure* das Element *PrivateCode* an die richtige Position verschoben.
- In *AddressStructure* das Element *CityName* in die korrekte Benennung *LocalityName* umbenannt.
- In *AddressStructure* die Dokumentation des Elements *LocalityRef* ergänzt.
- In *LocationStructure* die Referenzen auf die Ortsobjekte in die im Schema vorhandenen Kindelemente gewandelt.
- Diverse Tippfehler, Formatierungsfehler und falsche/fehlende Querverweise korrigiert.
- Erläuterung zum Verkehrstag und Betriebstag-Code in Kapitel 5.9 ergänzt
- Kapitel 5.3 Ergänzung mit Verweis auf OSM
- Ergänzung 7.5.5. *PointOfInterestCategoryStructure* und 7.5.6. *OsmTagStructure*
- Ergänzungen und Klärungen im Kapitel 19 zum Ablauf (*DataReceivedAcknowledgement*)

## 28.2. Version 1.2 (Schrift und Schema)

### 28.2.1. Funktionale Erweiterungen

- Anreicherungsdienst (vgl. *RefineRequest* in Kap. 15) als neuen Dienst hinzugefügt.
- Zwei neue optionale Element in *StopCallStatusGroup* aufgenommen: *NoBoardingAtStop* und *NoAlightingAtStop*.
- Neues optionales Element *JourneyTrack* in *TripInfoResultStructure*.
- In *TripInfoParamStructure* zwei zusätzliche Filterelemente *IncludeTrackSections* und *IncludeTrackProjection*.
- *BaseTripContentFilterGroup* wurde erweitert um *IncludeEstimatedTimes* und *IncludeSituationInfo*.
- Es wurde die Möglichkeit geschaffen, Wechsel von Fahrteigenschaften (zb. Zugnummer, Gattung) abzubilden. Zu diesem Zweck wurde die neue *ServiceSectionStructure* geschaffen und die *ServiceJourneyGroup* sowie die *ServiceGroup* umgebaut.
- *ServiceAttributeStructure* wurde um ein Element *Scope* erweitert, um den Bezug des Attributes angeben zu können.
- Parallelfahrten können nun abschnittsweise angegeben werden (z.B. Flügelungen). Dazu wurde eine neue *ParallelServiceStructure* definiert und in *TimedLegStructure*, *TripInfoResultStructure* sowie *StopEventStructure* eingebaut.
- Es ist nun möglich, eine Zustiegs- bzw. Ausstiegshilfe anzufordern, hierfür wurde der Anschlussdienst und der Dienst für Fahrzeuginformationen entsprechend erweitert.
- Der Fahrzeugdienst wurde um die Möglichkeit erweitert, Ansagen auf den Außenlautsprecher auszugeben. In diesem Zuge wurde eine Aufteilung der Funktionalität innerhalb des Fahrzeugdienstes in Interaktion und Information vorgenommen.



- Erweiterung der Anfragestrukturen für Anschlüsse und den Haltewunsch um eine *PassengerProfileGroup*, in der generische Mobilitätseinschränkungen abgebildet sind.
- Erweiterung von *TripInfoContentFilterGroup* um *IncludeEstimatedTimes* und *IncludeSituationInfo*.

### 28.2.2. Technische Ergänzungen/Korrekturen

- Alle optionalen Elemente vom Typ *xs:boolean* wurden im Schema von *FIXED*-Werten auf *DEFAULT*-Werte umgestellt. In der Schrift sind die jeweils gültigen Default-Werte angegeben.
- Das verpflichtende Element *StopSeqNumber* (in den Strukturen *CallAtStopStructure*, *DatedCallAtLocationStructure*, *LegBoardStructure*, *LegAlightStructure* und *LegIntermediateStructure*) ist nun nur noch optional.
- Die nicht verwendete Datenstruktur *RouteDescriptionGroup* wurde aus dem Schema entfernt.
- Elemente vom Typ *InternationalTextStructure* können nun mehrfach vorkommen, um Mehrsprachigkeit zu ermöglichen.
- Das Element *Language* wurde aus der *LocationParamStructure* entfernt.
- In Requests (*ServiceRequestContext*, *SubscriptionRequestContext*) kann nun der Client präferierte Sprachen angeben.
- In Responses (*ServiceResponseContext*) kann die EKAP mehrere ausgelieferte Sprachen definieren.
- Das Element *Location* in *LocationInformationResponseStructure* wurde zu *Location-Result* umbenannt.
- Für alle XML-Inhalte ist nun die Codierung UTF-8 explizit vorgeschrieben.
- *JourneyAttributeStructure* wurde ersetzt durch *ServiceAttributeStructure*.
- *LegAttribute* wurde aus *TimeLegStructure* entfernt. Fahrtattribute werden stattdessen in der *ServiceJourneyGroup* unterhalb der *DatedJourneyStructure* angegeben.
- Zusammenfassung der *StopRequestRequestStructure* und der *ActivateOutside-SpeakerRequestStructure* in einer *VehicleInteractionRequestStructure*.
- Damit Leitstellen eine Fahrt zuverlässig erkennen können, wurden *FeederDistributorStructure* und *DatedCallAtLocationStructure* um *LineDirectionGroup* und *OperatorRef* erweitert.
- *TicketStructure* um *TariffLevelGroup* zur Abbildung von Preis- und Tarifstufen erweitert
- Einführung eines Typs *ServiceCallStructure*, welcher die *ServiceTimeGroup* enthält. Verwendung dieses Typs in *TimeStopStructure*, *LegBoardStructure*, *LegAlightStructure*, *LegIntermediateStructure*, *CallAtStopStructure* und *DatedCallAtLocationStructure* um jeweils Informationen zu Ankunft bzw. Abfahrt einer Fahrt an einem Punkt aufzunehmen.

### 28.2.3. Dokumentarische Ergänzungen/Korrekturen

- Kapitel 6 wurde inhaltlich erweitert und entsprechend umbenannt.
- Das neue Kapitel 6.5 zur Erläuterung von Haltesequenznummern und Fahrtabschnitten wurde eingefügt.
- Im Unterkapitel 17.2.3 wurde der Name des Elements *IndividualRouteContextStructure* in *IndividualRouteLocationContextStructure* korrigiert.
- Im Unterkapitel 12.3.3 wurde der Datentyp von *PreviousCall* und *OnwardCall* von *CallAtNearStop* zu *CallAtStop* korrigiert.
- Im Unterkapitel 13.8.1 wurde der Name des Elements *Extensions* in *Extension* korrigiert.

## 29. 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 283 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 283: Liste der Abkürzungen.

### 30. Abbildungsverzeichnis

Abbildung 1: Umfeld und Schwerpunkte im Projekt IP-KOM-ÖV .....	17
Abbildung 2: Anfrage mit synchroner Antwort (Abbildung entnommen aus SIRI, (CEN, TS 15531 Part 2, 2011)). .....	22
Abbildung 3: Abonnement-Verfahren mit asynchronen Benachrichtigungen (Abbildung entnommen aus SIRI, (CEN, TS 15531 Part 2, 2011)). .....	23
Abbildung 4: Ablauf der Anschlussdienste mit aktiver Benachrichtigung bei Statusänderung .....	103
Abbildung 5: Ablauf der Anschlussdienste mit passiver Benachrichtigung bei Statusänderung .....	103
Abbildung 6: Funktionsweise des Benachrichtigungsdienstes .....	142
Abbildung 7: Sequenzdiagramm Lebenszyklus eines Wertes.....	146
Abbildung 8: Sequenzdiagramm Werteliste ermitteln .....	147
Abbildung 9: Sequenzdiagramm Personalisierung .....	148
Abbildung 10: Erzeugung einer Nachrichtensignatur.....	164
Abbildung 11: Überprüfung einer Signatur.....	165

### 31. Tabellenverzeichnis

Tabelle 1: Beispiel (aus einem späteren Abschnitt) für die tabellarische Notation einer XML-Struktur .....	20
Tabelle 2: Liste der TRIAS-Dienste und ihrer Anfrageelemente. ....	33
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.....	36
Tabelle 5: Beschreibung der Struktur <b>ServiceRequestStructure</b> . ....	37
Tabelle 6: Beschreibung der Struktur <b>SubscriptionRequestStructure</b> . ....	38
Tabelle 7: Beschreibung der Struktur <b>RequestPayloadStructure</b> . ....	39
Tabelle 8: Beschreibung der Struktur <b>ServiceDeliveryStructure</b> . ....	40
Tabelle 9: Beschreibung der Struktur <b>DeliveryPayloadStructure</b> . ....	41
Tabelle 10: Liste der einfachen Typdefinitionen in Trias_Utility.xsd.....	41
Tabelle 11: Beschreibung der Struktur <b>InternationalTextStructure</b> . ....	42
Tabelle 12: Beschreibung der Struktur <b>GeoPositionStructure</b> . ....	42
Tabelle 13: Beschreibung der Struktur <b>WebLinkStructure</b> . ....	42
Tabelle 14: Liste der einfachen Typdefinitionen in Trias_ModesSupport.xsd. ....	44
Tabelle 15: Beschreibung der Struktur <b>IndividualTransportOptionsStructure</b> . ....	44

Tabelle 16:	Beschreibung der Gruppe <b>PtSubmodeChoiceGroup</b> .....	45
Tabelle 17:	Beschreibung der Struktur <b>ModeStructure</b> .....	45
Tabelle 18:	Beschreibung der Struktur <b>PtModeFilterStructure</b> .....	46
Tabelle 19:	Liste der einfachen Typdefinitionen in Trias_Common.xsd.....	46
Tabelle 20:	Beschreibung der Struktur <b>ErrorMessageStructure</b> .....	46
Tabelle 21:	Beschreibung der Struktur <b>PrivateCodeStructure</b> .....	47
Tabelle 22:	Beschreibung der Struktur <b>OperatorFilterStructure</b> .....	47
Tabelle 23:	Beschreibung der Struktur <b>LineDirectionStructure</b> .....	47
Tabelle 24:	Beschreibung der Struktur <b>LineDirectionFilterStructure</b> .....	47
Tabelle 25:	Beschreibung der Struktur <b>SharingServiceStructure</b> .....	48
Tabelle 26:	Beschreibung der Struktur <b>OperatingDaysStructure</b> .....	48
Tabelle 27:	Beschreibung der Struktur <b>WeekdayTimePeriodStructure</b> .....	48
Tabelle 28:	Beschreibung der Struktur <b>GeneralAttributeStructure</b> .....	49
Tabelle 29:	Liste der einfachen Typdefinitionen in Trias_LocationSupport.xsd.....	49
Tabelle 30:	Beschreibung der Struktur <b>StopPointStructure</b> .....	50
Tabelle 31:	Beschreibung der Struktur <b>StopPlaceStructure</b> .....	51
Tabelle 32:	Beschreibung der Struktur <b>LocalityStructure</b> .....	51
Tabelle 33:	Beschreibung der Struktur <b>PointOfInterestStructure</b> .....	51
Tabelle 34:	Beschreibung der Struktur <b>PointOfInterestCategoryStructure</b> .....	51
Tabelle 35:	Beschreibung der Struktur <b>OsmTagStructure</b> .....	52
Tabelle 36:	Beschreibung der Struktur <b>PointOfInterestFilterStructure</b> .....	52
Tabelle 37:	Beschreibung der Struktur <b>AddressStructure</b> .....	52
Tabelle 38:	Beschreibung der Struktur <b>LocationStructure</b> .....	53
Tabelle 39:	Beschreibung der Struktur <b>LocationRefStructure</b> .....	53
Tabelle 40:	Beschreibung der Struktur <b>ServiceViaPointStructure</b> .....	54
Tabelle 41:	Beschreibung der Struktur <b>ViaStructure</b> .....	54
Tabelle 42:	Beschreibung der Struktur <b>ServiceSectionStructure</b> .....	55
Tabelle 43:	Beschreibung der Gruppe <b>DatedServiceGroup</b> .....	55
Tabelle 44:	Beschreibung der Struktur <b>DatedJourneyStructure</b> .....	56
Tabelle 45:	Beschreibung der Struktur <b>ParallelServiceStructure</b> .....	56
Tabelle 46:	Beschreibung der Struktur <b>TripLocationStructure</b> .....	56
Tabelle 47:	Beschreibung der Struktur <b>ServiceCallStructure</b> .....	56
Tabelle 48:	Beschreibung der Struktur <b>CallAtStopStructure</b> .....	57
Tabelle 49:	Beschreibung der Struktur <b>DatedCallAtLocationStructure</b> .....	58
Tabelle 50:	Beschreibung der Struktur <b>ContinuousServiceStructure</b> .....	59

Tabelle 51:	Beschreibung der Struktur <b>VehiclePositionStructure</b> .....	59
Tabelle 52:	Beschreibung der Struktur <b>ProgressBetweenStopsStructure</b> .....	59
Tabelle 53:	Beschreibung der Struktur <b>LegTrackStructure</b> .....	60
Tabelle 54:	Beschreibung der Struktur <b>TrackSectionStructure</b> .....	60
Tabelle 55:	Beschreibung der Struktur <b>LocationContextStructure</b> .....	60
Tabelle 56:	Beschreibung der Struktur <b>AbstractResponseContextStructure</b> .....	61
Tabelle 57:	Beschreibung der Struktur <b>ServiceAttributeStructure</b> .....	61
Tabelle 58:	Beschreibung der Struktur <b>PassengerAccessibilityStructure</b> .....	62
Tabelle 59:	Beschreibung der Gruppe <b>siri:CommonFacilityGroup</b> .....	64
Tabelle 60:	Beschreibung der Gruppe <b>siri:StopFacilityGroup</b> .....	65
Tabelle 61:	Beschreibung der Gruppe <b>siri:ServiceFacilityGroup</b> .....	65
Tabelle 62:	Beschreibung der Gruppe <b>siri:AllFacilitiesGroup</b> .....	66
Tabelle 63:	Beschreibung der Struktur <b>SituationsStructure</b> .....	66
Tabelle 64:	Beschreibung der Struktur <b>SituationFullRefStructure</b> .....	67
Tabelle 65:	Liste der einfachen Typdefinitionen in Trias_RequestSupport.xsd .....	68
Tabelle 66:	Beschreibung der Struktur <b>AbstractTriasServiceRequestStructure</b> .....	68
Tabelle 67:	Beschreibung der Struktur <b>AbstractTriasSubscriptionRequestStructure</b> .....	69
Tabelle 68:	Beschreibung der Struktur <b>AbstractTriasResponseStructure</b> .....	70
Tabelle 69:	Liste der einfachen Typdefinitionen in Trias_FaresSupport.xsd .....	71
Tabelle 70:	Beschreibung der Struktur <b>FareZoneStructure</b> .....	71
Tabelle 71:	Beschreibung der Struktur <b>FareZoneListInAreaStructure</b> .....	71
Tabelle 72:	Beschreibung der Struktur <b>BookingInfoStructure</b> .....	72
Tabelle 73:	Beschreibung der Struktur <b>TicketStructure</b> .....	73
Tabelle 74:	Beschreibung der Struktur <b>TripFaresResultStructure</b> .....	73
Tabelle 75:	Liste der Fehlerzustände in <b>TripFaresResult</b> .....	73
Tabelle 76:	Beschreibung der Struktur <b>FaresPassengerStructure</b> .....	74
Tabelle 77:	Beschreibung der Struktur <b>FaresParamStructure</b> .....	74
Tabelle 78:	Liste der einfachen Typdefinitionen in Trias_Locations.xsd .....	75
Tabelle 79:	Beschreibung der Struktur <b>LocationInformationRequestStructure</b> .....	76
Tabelle 80:	Beschreibung der Struktur <b>InitialLocationInputStructure</b> .....	76
Tabelle 81:	Beschreibung der Struktur <b>GeoRestrictionsStructure</b> .....	77
Tabelle 82:	Beschreibung der Struktur <b>GeoCircleStructure</b> .....	77
Tabelle 83:	Beschreibung der Struktur <b>GeoRectangleStructure</b> .....	77
Tabelle 84:	Beschreibung der Struktur <b>GeoAreaStructure</b> .....	77
Tabelle 85:	Beschreibung der Struktur <b>LocationParamStructure</b> .....	78

Tabelle 86:	Beschreibung der Struktur <b><i>LocationInformationResponseStructure</i></b> .....	79
Tabelle 87:	Liste der Fehlerzustände in <b><i>LocationInformationResponse</i></b> .....	79
Tabelle 88:	Beschreibung der Struktur <b><i>LocationResultStructure</i></b> .....	80
Tabelle 89:	Beschreibung der Struktur <b><i>TripRequestStructure</i></b> .....	80
Tabelle 90:	Beschreibung der Struktur <b><i>TripParamStructure</i></b> .....	82
Tabelle 91:	Beschreibung der Gruppe <b><i>NumberOfResultsGroup</i></b> .....	82
Tabelle 92:	Beschreibung der Struktur <b><i>NotViaStructure</i></b> .....	82
Tabelle 93:	Beschreibung der Struktur <b><i>NoChangeAtStructure</i></b> .....	83
Tabelle 94:	Beschreibung der Struktur <b><i>TripResponseStructure</i></b> .....	84
Tabelle 95:	Liste der Fehlerzustände in <b><i>TripResponse</i></b> .....	84
Tabelle 96:	Beschreibung der Struktur <b><i>TripResponseContextStructure</i></b> .....	85
Tabelle 97:	Beschreibung der Struktur <b><i>TripResultStructure</i></b> .....	85
Tabelle 98:	Liste der Fehlerzustände in <b><i>TripResult</i></b> .....	85
Tabelle 99:	Beschreibung der Struktur <b><i>TripStructure</i></b> .....	86
Tabelle 100:	Beschreibung der Struktur <b><i>TripLegStructure</i></b> .....	86
Tabelle 101:	Beschreibung der Struktur <b><i>TimedLegStructure</i></b> .....	87
Tabelle 102:	Beschreibung der Struktur <b><i>InterchangeLegStructure</i></b> .....	88
Tabelle 103:	Beschreibung der Struktur <b><i>ContinuousLegStructure</i></b> .....	89
Tabelle 104:	Beschreibung der Struktur <b><i>LegBoardStructure</i></b> .....	90
Tabelle 105:	Beschreibung der Struktur <b><i>LegAlightStructure</i></b> .....	91
Tabelle 106:	Beschreibung der Struktur <b><i>LegIntermediateStructure</i></b> .....	92
Tabelle 107:	Beschreibung der Struktur <b><i>NavigationPathStructure</i></b> .....	92
Tabelle 108:	Beschreibung der Struktur <b><i>NavigationSectionStructure</i></b> .....	93
Tabelle 109:	Beschreibung der Struktur <b><i>AccessPathStructure</i></b> .....	93
Tabelle 110:	Beschreibung der Struktur <b><i>StopEventRequestStructure</i></b> .....	94
Tabelle 111:	Beschreibung der Struktur <b><i>StopEventParamStructure</i></b> .....	95
Tabelle 112:	Beschreibung der Struktur <b><i>StopEventResponseStructure</i></b> .....	95
Tabelle 113:	Liste der Fehlerzustände in <b><i>StopEventResponse</i></b> .....	95
Tabelle 114:	Beschreibung der Struktur <b><i>StopEventResponseContextStructure</i></b> .....	95
Tabelle 115:	Beschreibung der Struktur <b><i>StopEventResultStructure</i></b> .....	95
Tabelle 116:	Liste der Fehlerzustände in <b><i>StopEventResult</i></b> .....	96
Tabelle 117:	Beschreibung der Struktur <b><i>StopEventStructure</i></b> .....	96
Tabelle 118:	Beschreibung der Struktur <b><i>CallAtNearStopStructure</i></b> .....	96
Tabelle 119:	Beschreibung der Struktur <b><i>PositioningRequestStructure</i></b> .....	97
Tabelle 120:	Beschreibung der Struktur <b><i>PositioningParamStructure</i></b> .....	97

Tabelle 121:	Beschreibung der Struktur <b>TimedPositionStructure</b> .	98
Tabelle 122:	Beschreibung der Struktur <b>TimedStopStructure</b> .	98
Tabelle 123:	Beschreibung der Struktur <b>PositioningResponseStructure</b> .	98
Tabelle 124:	Liste der Fehlerzustände in <b>PositioningResponse</b> .	99
Tabelle 125:	Beschreibung der Struktur <b>PositioningResultStructure</b> .	99
Tabelle 126:	Beschreibung der Struktur <b>PositioningStructure</b> .	99
Tabelle 127:	Beschreibung der Struktur <b>RankedPositionStructure</b> .	99
Tabelle 128:	Beschreibung der Struktur <b>TripInfoRequestStructure</b> .	100
Tabelle 129:	Beschreibung der Struktur <b>TripInfoParamStructure</b> .	101
Tabelle 130:	Beschreibung der Struktur <b>TripInfoResponseStructure</b> .	101
Tabelle 131:	Liste der Fehlerzustände in <b>TripInfoResponse</b> .	102
Tabelle 132:	Beschreibung der Struktur <b>TripInfoResponseContextStructure</b> .	102
Tabelle 133:	Beschreibung der Struktur <b>TripInfoResultStructure</b> .	102
Tabelle 134:	Beschreibung der einfachen Typen	105
Tabelle 135:	Beschreibung der Struktur <b>DatedConnectionStructure</b>	106
Tabelle 136:	Beschreibung der Struktur <b>FeederDistributorStructure</b>	106
Tabelle 137:	Beschreibung des Typs <b>GeneralizedConnectionStructure</b>	106
Tabelle 138:	Beschreibung der Struktur <b>ConnectionStatusStructure</b>	107
Tabelle 139:	Beschreibung der Struktur <b>RecommendationStructure</b>	107
Tabelle 140:	Beschreibung der Struktur <b>ConnectionDemandRequestStructure</b>	108
Tabelle 141:	Beschreibung der Struktur <b>ConnectionDemandDeleteRequestStructure</b>	108
Tabelle 142:	Beschreibung der Struktur <b>ConnectionDemandResponseStructure</b>	108
Tabelle 143:	Liste der Fehlerzustände in <b>ConnectionDemandResponse</b>	109
Tabelle 144:	Beschreibung der Struktur <b>ConnectionDemandDeleteResponseStructure</b>	109
Tabelle 145:	Liste der Fehlerzustände in <b>ConnectionDemandDeleteResponse</b>	109
Tabelle 146:	Beschreibung der Struktur <b>ConnectionStatusRequestStructure</b>	109
Tabelle 147:	Beschreibung der Struktur <b>ConnectionStatusNotificationStructure</b>	110
Tabelle 148:	Beschreibung der Struktur <b>ConnectionStatusResponseStructure</b>	110
Tabelle 149:	Liste der Fehlerzustände in <b>ConnectionStatusResponseStructure</b>	110
Tabelle 150:	Beschreibung der Struktur <b>ConnectionReportRequestStructure</b>	111
Tabelle 151:	Beschreibung der Struktur <b>ConnectionReportResponseStructure</b>	111
Tabelle 152:	Liste der Fehlerzustände in <b>ConnectionReportResponse</b>	111
Tabelle 153:	Beschreibung der Struktur <b>FaresRequestStructure</b>	112
Tabelle 154:	Beschreibung der Struktur <b>StopFaresRequestStructure</b>	112

Tabelle 155:	Beschreibung der Struktur <b><i>StaticFaresRequestStructure</i></b> .....	113
Tabelle 156:	Beschreibung der Struktur <b><i>TripFaresRequestStructure</i></b> .....	113
Tabelle 157:	Beschreibung der Struktur <b><i>MultiTripFaresRequestStructure</i></b> .....	113
Tabelle 158:	Beschreibung der Struktur <b><i>FaresResponseStructure</i></b> .....	114
Tabelle 159:	Liste der Fehlerzustände in <b><i>FaresResponse</i></b> .....	114
Tabelle 160:	Beschreibung der Struktur <b><i>FaresResultStructure</i></b> .....	114
Tabelle 161:	Beschreibung der Struktur <b><i>StopFaresResultStructure</i></b> .....	114
Tabelle 162:	Beschreibung der Struktur <b><i>StaticFaresResultStructure</i></b> .....	115
Tabelle 163:	Beschreibung der Struktur <b><i>TripTicketReferenceStructure</i></b> .....	115
Tabelle 164:	Beschreibung der Struktur <b><i>MultiTripFaresResultStructure</i></b> .....	115
Tabelle 165:	Liste der Fehlerzustände in <b><i>MultiTripFaresResultStructure</i></b> .....	116
Tabelle 166:	Beschreibung der Struktur <b><i>RefineRequestStructure</i></b> .....	117
Tabelle 167:	Beschreibung der Struktur <b><i>IndividualRouteRefineRequestStructure</i></b> .....	118
Tabelle 168:	Beschreibung der Struktur <b><i>LocationInformationRefineStructure</i></b> .....	118
Tabelle 169:	Beschreibung der Struktur <b><i>StopEventRefineRequestStructure</i></b> .....	118
Tabelle 170:	Beschreibung der Struktur <b><i>TripInfoRefineRequestStructure</i></b> .....	118
Tabelle 171:	Beschreibung der Struktur <b><i>TripRefineRequestStructure</i></b> .....	118
Tabelle 172:	Beschreibung der Struktur <b><i>IndividualTripRefineParamStructure</i></b> .....	119
Tabelle 173:	Beschreibung der Struktur <b><i>LocationInformationRefineParamStructure</i></b> .....	119
Tabelle 174:	Beschreibung der Struktur <b><i>StopEventRefineParamStructure</i></b> .....	119
Tabelle 175:	Beschreibung der Struktur <b><i>TripInfoRefineParamStructure</i></b> .....	120
Tabelle 176:	Beschreibung der Struktur <b><i>TripRefineParamStructure</i></b> .....	121
Tabelle 177:	Liste der Fehlerzustände in den Response-Strukturen der Refine- Anfragen.....	121
Tabelle 178:	Beschreibung der Struktur <b><i>RefineResponseStructure</i></b> .....	121
Tabelle 179:	Beschreibung der Struktur <b><i>IndividualRouteRefineResponseStructure</i></b> .....	121
Tabelle 180:	Beschreibung der Struktur <b><i>LocationInformationRefineResponseStructure</i></b> .....	122
Tabelle 181:	Beschreibung der Struktur <b><i>StopEventRefineResponseStructure</i></b> .....	122
Tabelle 182:	Beschreibung der Struktur <b><i>TripInfoRefineResponseStructure</i></b> .....	122
Tabelle 183:	Beschreibung der Struktur <b><i>TripRefineResponseStructure</i></b> .....	122
Tabelle 184:	Beschreibung der Struktur <b><i>BookingInfoRequestStructure</i></b> .....	123
Tabelle 185:	Beschreibung der Struktur <b><i>BookingInfoResponseStructure</i></b> .....	123
Tabelle 186:	Liste der Fehlerzustände in <b><i>BookingInfoResponse</i></b> .....	124
Tabelle 187:	Beschreibung der Struktur <b><i>BookingInfoResultStructure</i></b> .....	124



Tabelle 188:	Liste der einfachen Typdefinitionen in Trias_IndividualTrips.xsd.....	125
Tabelle 189:	Beschreibung der Struktur <b>IndividualRouteRequestStructure</b> . ....	125
Tabelle 190:	Beschreibung der Struktur <b>IndividualTripParamStructure</b> .....	126
Tabelle 191:	Beschreibung der Struktur <b>IndividualRouteLocationContextStructure</b> .	127
Tabelle 192:	Beschreibung der Struktur <b>IndividualRouteResponseStructure</b> . ....	127
Tabelle 193:	Liste der Fehlerzustände in <b>IndividualRouteResponse</b> . ....	128
Tabelle 194:	Beschreibung der Struktur <b>RouteResultStructure</b> .....	128
Tabelle 195:	Liste der Fehlerzustände in <b>RouteResult</b> .....	128
Tabelle 196:	Beschreibung der Struktur <b>RouteStructure</b> .....	129
Tabelle 197:	Liste der einfachen Typdefinitionen in Trias_Maps.xsd.....	130
Tabelle 198:	Beschreibung der Struktur <b>MapServiceRequestStructure</b> . ....	130
Tabelle 199:	Beschreibung der Struktur <b>ImageCoordinatesRequestStructure</b> . ....	131
Tabelle 200:	Beschreibung der Struktur <b>GeoCoordinatesRequestStructure</b> . ....	131
Tabelle 201:	Beschreibung der Struktur <b>MapAspectStructure</b> .....	131
Tabelle 202:	Beschreibung der Struktur <b>MapSizeStructure</b> . ....	132
Tabelle 203:	Beschreibung der Struktur <b>MapCoordinateStructure</b> .....	132
Tabelle 204:	Beschreibung der Struktur <b>MapServiceResponseStructure</b> . ....	132
Tabelle 205:	Liste der Fehlerzustände in <b>MapServiceResponse</b> . ....	132
Tabelle 206:	Beschreibung der Struktur <b>ImageCoordinatesResponseStructure</b> . ....	133
Tabelle 207:	Liste der Fehlerzustände in <b>ImageCoordinatesResponse</b> . ....	133
Tabelle 208:	Beschreibung der Struktur <b>GeoCoordinatesResponseStructure</b> . ....	133
Tabelle 209:	Liste der Fehlerzustände in <b>GeoCoordinatesResponse</b> . ....	134
Tabelle 210:	Beschreibung der Struktur <b>MapResultStructure</b> .....	134
Tabelle 211:	Beschreibung der Struktur <b>ImagePointResultStructure</b> .....	134
Tabelle 212:	Liste der Fehlerzustände in <b>ImagePointResultStructure</b> . ....	135
Tabelle 213:	Beschreibung der Struktur <b>GeoCoordinateResultStructure</b> . ....	135
Tabelle 214:	Liste der Fehlerzustände in <b>GeoCoordinateResultStructure</b> .....	135
Tabelle 215:	Beschreibung der Struktur einfacher Typen.....	136
Tabelle 216:	Beschreibung der Struktur <b>FacilityStructure</b> . ....	136
Tabelle 217:	Beschreibung der Struktur <b>VehicleFacilityStructure</b> . ....	136
Tabelle 218:	Beschreibung der Struktur <b>InfrastructureFacilityStructure</b> . ....	137
Tabelle 219:	Beschreibung der Gruppe <b>VehicleFacilityRefGroup</b> . ....	137
Tabelle 220:	Beschreibung der Gruppe <b>InfrastructureFacilityRefGroup</b> . ....	137
Tabelle 221:	Beschreibung der Gruppe <b>FacilityDataFilterGroup</b> .....	138
Tabelle 222:	Beschreibung der Struktur <b>FacilityStatusStructure</b> .....	138

Tabelle 223:	Beschreibung der Struktur <b>FacilityStatusReportStructure</b> .	138
Tabelle 224:	Beschreibung der Struktur <b>FacilityStatusReportResponseStructure</b> .	139
Tabelle 225:	Liste der Fehlerzustände in <b>FacilityStatusReportResponse</b> .	139
Tabelle 226:	Beschreibung der Struktur <b>FacilityRequestStructure</b> .	139
Tabelle 227:	Beschreibung der Struktur <b>FacilityResponseStructure</b> .	140
Tabelle 228:	Liste der Fehlerzustände in <b>FacilityResponse</b> .	140
Tabelle 229:	Beschreibung der Struktur <b>FacilityResultStructure</b> .	140
Tabelle 230:	Liste der Fehlerzustände in <b>FacilityResultStructure</b> .	140
Tabelle 231:	Beschreibung der Struktur <b>TripMonitoringParamStructure</b> .	142
Tabelle 232:	Beschreibung der Struktur <b>TripMonitoringSubscriptionRequestStructure</b> .	143
Tabelle 233:	Beschreibung der Struktur <b>TripMonitoringDeliveryStructure</b> .	143
Tabelle 234:	Liste der Fehlerzustände in <b>TripMonitoringDeliveryStructure</b> .	144
Tabelle 235:	Liste der einfachen Typdefinitionen in Trias_Personalisation.xsd.	149
Tabelle 236:	Beschreibung der Struktur <b>PersonalisationRequestStructure</b> .	149
Tabelle 237:	Beschreibung der Struktur <b>PersonalisationSaveValueRequestStructure</b> .	149
Tabelle 238:	Beschreibung der Struktur <b>PersonalisationRetrieveValueRequestStructure</b> .	149
Tabelle 239:	Beschreibung der Struktur <b>PersonalisationDeleteValueRequestStructure</b> .	150
Tabelle 240:	Beschreibung der Struktur <b>PersonalisationEnumerateValuesRequestStructure</b> .	150
Tabelle 241:	Beschreibung der Struktur <b>PersonalisationResponseStructure</b> .	150
Tabelle 242:	Beschreibung der Struktur <b>PersonalisationSaveValueResponseStructure</b> .	150
Tabelle 243:	Liste der Fehlerzustände in <b>PersonalisationSaveValueResponseStructure</b> .	151
Tabelle 244:	Beschreibung der Struktur <b>PersonalisationRetrieveValueResponseStructure</b> .	151
Tabelle 245:	Liste der Fehlerzustände in <b>PersonalisationRetrieveValueResponseStructure</b> .	151
Tabelle 246:	Beschreibung der Struktur <b>PersonalisationDeleteValueResponseStructure</b> .	151
Tabelle 247:	Liste der Fehlerzustände in <b>PersonalisationDeleteValueResponseStructure</b> .	151
Tabelle 248:	Beschreibung der Struktur <b>PersonalisationEnumerateValuesResponseStructure</b> .	152
Tabelle 249:	Beschreibung der Struktur <b>VehicleDataRequestStructure</b> .	152

Tabelle 250:	Beschreibung der Struktur <b>VehicleDataResponseStructure</b> .....	152
Tabelle 251:	Beschreibung der Struktur <b>VehicleStatusStructure</b> .....	153
Tabelle 252:	Beschreibung der Struktur <b>VehicleActivityStructure</b> .....	153
Tabelle 253:	Beschreibung der Struktur <b>StopSequenceStructure</b> .....	153
Tabelle 254:	Beschreibung der Struktur <b>StopInformationStructure</b> .....	154
Tabelle 255:	Beschreibung der Struktur <b>DisplayContentStructure</b> .....	154
Tabelle 256:	Beschreibung der Struktur <b>LineInformationStructure</b> .....	154
Tabelle 257:	Beschreibung der Struktur <b>DestinationStructure</b> .....	155
Tabelle 258:	Beschreibung der Struktur <b>ViaPointStructure</b> .....	155
Tabelle 259:	Beschreibung der Struktur <b>AnnouncementStructure</b> .....	155
Tabelle 260:	Beschreibung der Struktur <b>ConnectionStructure</b> .....	156
Tabelle 261:	Beschreibung der Struktur <b>VehicleTypeStructure</b> .....	156
Tabelle 262:	Beschreibung der Struktur <b>VehicleInteractionRequestStructure</b> .....	157
Tabelle 263:	Beschreibung der Struktur <b>ActivateOutsideSpeakerRequestStructure</b> .....	157
Tabelle 264:	Beschreibung der Struktur <b>StopRequestRequestStructure</b> .....	157
Tabelle 265:	Beschreibung der Struktur <b>VehicleInteractionResponseStructure</b> ..	158
Tabelle 266:	Beschreibung der Struktur <b>ActivateOutsideSpeakerResponseStructure</b> .....	158
Tabelle 267:	Beschreibung der Struktur <b>StopRequestResponseStructure</b> .....	158
Tabelle 268:	Beschreibung von einfachen Typen.....	159
Tabelle 269:	Beschreibung der Struktur <b>ServiceRegisterRequestStructure</b> .....	159
Tabelle 270:	Beschreibung der Struktur <b>ServiceRegisterParamStructure</b> .....	160
Tabelle 271:	Beschreibung der Struktur <b>ServiceRegisterRegisterRequestStructure</b> .....	160
Tabelle 272:	Beschreibung der Struktur <b>ServiceRegisterUpdateRequestStructure</b> .....	160
Tabelle 273:	Beschreibung der Struktur <b>ServiceRegisterLookupRequestStructure</b> .....	161
Tabelle 274:	Beschreibung der Struktur <b>ServiceRegisterUnregisterRequestStructure</b> .....	161
Tabelle 275:	Beschreibung der Struktur <b>ServiceRegisterResponseStructure</b> .....	161
Tabelle 276:	Liste der Fehlerzustände in <b>ServiceRegisterResponse</b> .....	161
Tabelle 277:	Beschreibung der Struktur <b>ServiceRegisterResultStructure</b> .....	162
Tabelle 278:	Beschreibung der Struktur <b>ServiceRegisterRegisterResponseStructure</b> .....	162
Tabelle 279:	Beschreibung der Struktur <b>ServiceRegisterUpdateResponseStructure</b> .....	162
Tabelle 280:	Beschreibung der Struktur <b>ServiceRegisterLookupResponseStructure</b> .....	162

Tabelle 281:	Beschreibung	der	Struktur
	<b><i>ServiceRegisterUnregisterResponseStructure</i></b>	.....	162
Tabelle 282:	Beschreibung der Struktur	<b><i>TriasServiceStructure</i></b>	..... 163
Tabelle 283:	Liste der Abkürzungen	.....	170

## 32. 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. (2011). *SIRI - Service Interface for Realtime Information, Part 1*. CEN - Europäisches Komitee für Normung.
- CEN, TS 15531 Part 2. (2011). *SIRI - Service Interface for Realtime Information, Part 2*. CEN - Europäisches Komitee für Normung.
- CEN, TS 15531 Part 5. (2011). *SIRI - Service Interface for Realtime Information, Part 5*. 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. (2014). *Kundenschnittstelle - Architektur*. Köln: VDV.
- VDV-Schrift 431-1. (2014). *Systemarchitektur EKAP*. Köln: VDV.
- VDV-Schrift 432. (07/2016). *Identifikation von Haltestellen - Anwendung der Global ID in Deutschland*. Köln.

---

# Impressum

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

## **Ansprechpartner**

Dipl.-Ing. Berthold Radermacher  
T 0221 57979-141  
F 0221 57979-8141  
radermacher@vdv.de

---

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

---