

MobilityBroker Schnittstelle mit Smartcar Erweiterung basierend auf IXSI - Interface for X-Sharing Information Version 4

Markus Beutel	RWTH Aachen University
Sevket Gökay	RWTH Aachen University
Peter von Grumbkow	HaCon Ing.-Ges. mbH
Dirk Hillbrecht	Cantamen GmbH
Karl-Heinz Krempels	RWTH Aachen University
Christian Samsel	RWTH Aachen University
Christoph Terwelp	RWTH Aachen University
Heike Twele	HaCon Ing.-Ges. mbH
Gerhard Wagner	Cantamen GmbH

Datum: 29. Mai 2018 Dokumentversion: 0.20

IXSI ist unter der Creative-Commons-Lizenz „Namensnennung Keine Bearbeitung 3.0 Deutschland“, CC BY-ND 3.0 DE, lizenziert. Eine Nutzung der Schnittstellenspezifikation ist frei möglich, auch in anderen Projekten. Bei einer Veröffentlichung müssen die Autoren genannt werden. Eine Abänderung der Spezifikation ist nicht gestattet. Die Details zur Lizenz finden sich in Kurzform (<http://creativecommons.org/licenses/by-nd/3.0/de/>) und in der rechtlich bindenden Vertragsform (<http://creativecommons.org/licenses/by-nd/3.0/de/legalcode>) auf den Seiten der Creative Commons. Das zugehörige XML Schema kann von <http://www.ixsi-schnittstelle.de> heruntergeladen werden.



Gefördert durch:



aufgrund eines Beschlusses
des Deutschen Bundestages

Inhaltsverzeichnis

1	Zusammenfassung	7
2	Rollenmodell	9
3	Hierarchiemodell	13
3.1	Basisdienst A – Sitzungen (session handling)	13
3.2	Basisdienst B – Abonnements (subscription handling)	14
3.3	Basisdienst C – tokens	14
3.4	Dienst 1 – Statische Daten (static data)	14
3.5	Dienst 2 – Verfügbarkeitsauskunft (availability query)	15
3.6	Dienst 3 – Verfügbarkeitsabonnement (availability subscription)	15
3.7	Dienst 4 – Buchung (booking)	16
3.8	Dienst 5 – Buchungsabonnement (booking subscription)	17
3.9	Dienst 6 – Preisauskunft (price information)	17
3.10	Dienst 7 – Abonnement Verbrauchsdaten / Abrechnung (consumption data) . .	18
3.11	Dienst 8 – Buchungsstatus ändern (freischalten / pausieren / abschließen) (boo- king state change)	18
3.12	Dienst 9 – Anlegen und Sperren von Nutzern (user management)	19
3.13	Dienst 10 – Fahrzeug- bzw. Buchungseinstellungsmanagement (booking settings)	19
3.14	Dienst 11 – Fernkonfiguration Navigationssystem und Fahrtverlauf überwachen	20
3.15	Dienst 12 – Dialog Intermodale Alternative	20
4	Interaktionsprotokolle	23
4.1	Überblick	23
4.2	Dienst 1 – Statische Daten	26
4.3	Dienst 2 – Verfügbarkeitsauskunft	27
4.4	Dienst 3 – Verfügbarkeitsabonnement	29
4.5	Dienst 4 – Buchung / Buchungsänderung	31
4.6	Dienst 5 – Buchungsabonnement	33
4.7	Dienst 6 – Preisauskunft	34
4.8	Dienst 7 – Verbrauchsdatenabonnement	35
4.9	Dienst 8 – Buchungsstatus ändern	36
4.10	Dienst 9 – Benutzermanagement	37
4.11	Dienst 10 – Fahrzeugeinstellungsmanagement	38
4.12	Dienst 11 – Fernkonfiguration Navigationssystem und Fahrtverlauf überwachen	39

4.13	Dienst 12 – Dialog Intermodale Alternative	40
5	Datenmodell	43
5.1	Basisdatentypen	43
5.2	Basisgruppen	53
5.3	Dienst 1 - Statische Daten	54
5.4	Dienst 2 - Verfügbarkeitsauskunft	63
5.5	Dienst 4 - Buchung	65
5.6	Dienst 5 - Buchungsabonnement	67
5.7	Dienst 6 - Preisauskunft	68
5.8	Dienst 7 - Verbrauchsdatenabonnement	69
5.9	Dienst 8 - Fahrzeugfreischaltung	70
5.10	Dienst 9 - Usermanagement	70
5.11	Dienst 10 - Fahrzeugeinstellungsmanagement	71
5.12	Dienst 11 - Fernkonfiguration Navigationssystem und Fahrtverlauf überwachen	72
5.13	Dienst 12 - Dialog Intermodale Alternative	73
5.14	Authentifizierung	73
5.15	Fehlerbehandlung	75
6	Technische Realisierung	77
6.1	Nachrichtenkodierung	77
6.2	Kommunikationskanal	77
6.3	Authentifizierung	77
6.4	Verbindungssicherheit	78
7	Nachrichten	79
7.1	Basisnachrichten	79
7.2	Basisgruppen	83
7.3	Dienst A - Sitzungen	88
7.4	Dienst B - Abonnements	90
7.5	Dienst C - Tokens	90
7.6	Dienst 1 - Statische Daten	91
7.7	Dienst 2 - Verfügbarkeitsauskunft	94
7.8	Dienst 3 - Verfügbarkeitsabonnement	97
7.9	Dienst 4 - Buchung / Buchungsänderung	105
7.10	Dienst 5 - Buchungsabonnement	111
7.11	Dienst 6 - Preisauskunft	115
7.12	Dienst 7 - Verbrauchsdatenabonnement	116
7.13	Dienst 8 - Buchungsstatus ändern	119
7.14	Dienst 9 - User Management	120
7.15	Dienst 10 - Fahrzeugseinstellungsmanagement	122
7.16	Dienst 11 - Fernkonfiguration Navigationssystem und Fahrtverlauf überwachen	126
7.17	Dienst 12 - Dialog Intermodale Alternative	129

8 Code Tabellen	135
8.1 Fahrzeugklassen	135
8.2 Antriebsklassen	135
8.3 Fahrzeugeigenschaften	136
8.4 Verbrauchsdaten	137
8.5 Nutzerzustände	137
8.6 Fehlercodes	137
Quellcodeverzeichnis	139
Stichwortverzeichnis	143

1 Zusammenfassung

Ziel dieser Schnittstellenspezifikation ist die Kopplung von Verleihsystemen für Fahrzeuge des Individualverkehrs mit Reiseinformationssystemen. Der Grund für die Kopplung ist die Bedienung der trendgetriebenen Anforderung an intermodale Reiseketten, hinsichtlich der Integration von Verleihsystemen.

Die Schnittstellenspezifikation besteht aus:

- einem Rollenmodell der beteiligten Akteure,
- einer Empfehlung für eine Dienstebenenhierarchie zur Bereitstellung unterschiedlicher Qualitäten der Informationskopplung,
- der Interaktionsprotokolle zur Beschreibung der Nachrichtenabfolgen zwischen den beteiligten Akteuren zur Bereitstellung der Informationskopplung entsprechend der spezifizierten Dienstebenen,
- der Beschreibung der zugrundeliegenden Datentypen für die Nachrichten der Interaktionsprotokolle,
- der Spezifikation von geeigneten Technologien zur Darstellung der Daten, der Kommunikation zwischen den zu koppelnden Informationssystemen, der Implementierung der Interaktionsprotokolle und der abschließenden Verarbeitung der Informationen und
- einer Reihe von Codetabellen, die erlaubte Werte für Aufzählungen enthält.

Diese Version von IXSI enthält die folgenden MoblityBroker-spezifischen Erweiterungen:

- Fahrzeugfreischaltung
- Verbrauchsdatenaustausch
- Abonnement von externen Buchungen
- Anlegen und Sperren von FVS (Fahrzeugverleihsystem)-Nutzern

Diese Version von IXSI enthält die folgenden Smartcar-spezifischen Erweiterungen:

- Synchronisieren von Fahrzeugeinstellungen
- Fernkonfiguration Navigationssystem und Fahrtverlauf überwachen

1 Zusammenfassung

- Dialog Intermodale Alternative

2 Rollenmodell

Das Rollenmodell beschreibt die vorkommenden Rollen des Integrationsszenarios eines Fahrzeugverleihsystems (z. B. CarSharing) in ein Reiseinformationssystem.

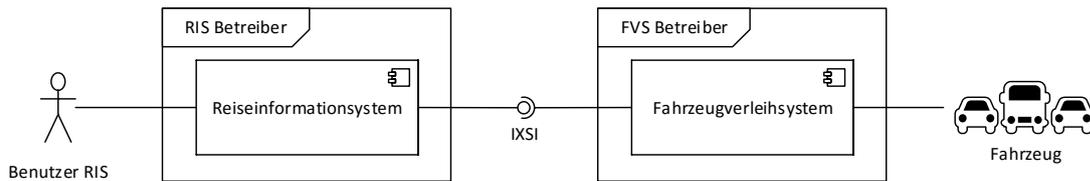


Abbildung 2.1: Überblick Rollenmodell.

Reiseinformationssystem

Das Reiseinformationssystem (RIS) stellt ein Informationssystem für Reiseauskünfte dar und umfasst die Zusammenführung der Reiseangebote, die Konstruktion von Reiseketten, die Berechnung des Gesamtpreises für eine Reisekette, die Reservierung der Mobilitätsangebote für die einzelnen Elemente einer Reisekette, die Aufbereitung der Daten für die Darstellung in Benutzerschnittstellen und abschließend die Darstellung der Informationen.

Anwendungsfälle

- Kunde fragt mit Suchbedingungen und Präferenzen Auskünfte über Mobilitätsangebote beim RIS an. Suchbedingungen und Präferenzen sind beispielsweise Start- und Zielort, Abfahrt- und Ankunftszeitpunkt, zu berücksichtigende Verkehrsmittel, Anzahl der Umstiege, Preisspanne, etc. Die Ergebnisse werden dem Kunden über die Benutzerschnittstelle des RIS in Form von Reiseketten dargestellt.

Fahrzeugverleihsystem

Das Fahrzeugverleihsystem (FVS) stellt ein Informationssystem zur Verwaltung und Buchung von Leihfahrzeugen und Kunden dar. Fahrzeuge können sowohl unterschiedlichen Typs, als auch stationsgebunden oder -ungebunden sein.

Anwendungsfälle

- Ein Kunde bucht (leiht) über das FVS ein Fahrzeug zu vertraglich geregelten Preisen, Zeiten und Stationen aus und nutzt dieses.
- Ein Kunde fragt über das FVS die Verfügbarkeit eines Fahrzeugs an.

Benutzer RIS

Kunde RIS – stellt eine juristische Person dar, die befugt ist eine Reise zu buchen und unter Nutzung der gewählten Verkehrsmittel anzutreten.

Anwendungsfälle

- Benutzer stellt Auskunftsanfrage an das RIS.
- Benutzer bucht Reise über das RIS.

Benutzer FVS

Kunde FVS - stellt eine juristische Person dar, die befugt ist ein Fahrzeug zu leihen und zu nutzen.

Anwendungsfälle

- Benutzer stellt Auskunftsanfrage an das FVS.
- Benutzer bucht ein Fahrzeug über das FVS.

Betreiber RIS

Betreiber RIS – stellt das RIS als Informationssystem für einen Mobilitätsanbieter als Dienstleistung bereit.

Betreiber FVS

Betreiber FVS - stellt ein Informationssystem für FVS für einen Fahrzeugverleiher als Dienstleistung bereit.

Fahrzeug

Fahrzeug – wird vom Mobilitätsanbieter über das FVS an einer Station als Mobilitätsangebot für den Kunden angeboten.

3 Hierarchiemodell

Das Hierarchiemodell beschreibt unterschiedliche Qualitäten der Informationskopplung, basierend auf Dienstgruppen und dient als Empfehlung für unterschiedliche Ausbaustufen der Implementierung. Um eine Kopplung zwischen FVS und RIS zu realisieren, muss mindestens der Dienst 1: Statische Daten (static data), von beiden Interaktionspartnern unterstützt werden. In Abb. 3.1 sind die Abhängigkeiten zwischen den Diensten dargestellt.

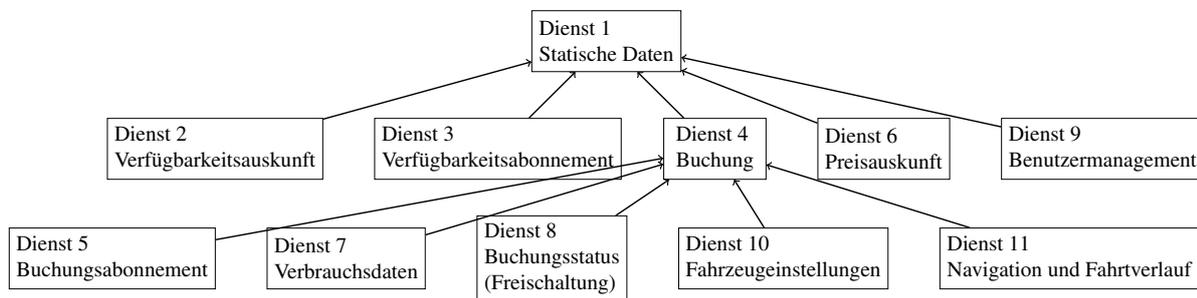


Abbildung 3.1: Dienstabhängigkeiten

3.1 Basisdienst A – Sitzungen (session handling)

Dienst A dient der Authentifizierung von Endkunden gegenüber dem FVS.

Funktionen

- Sitzung öffnen / schließen

Abhängigkeiten

keine

3.2 Basisdienst B – Abonnements (subscription handling)

Dienst B enthält eine Funktion zur Überprüfung des Status einer Abonnementverbindung (heartbeat)

Funktionen

- Heartbeat

Abhängigkeiten

keine

3.3 Basisdienst C – tokens

Dienst C enthält eine Funktion zur Erstellung von Authentifizierungstokens für Endkunden, die an Stelle von Klartextpasswörtern gespeichert / übertragen werden können.

Funktionen

- Erstellung von Tokens

Abhängigkeiten

keine

3.4 Dienst 1 – Statische Daten (static data)

Dienst 1 dient dem Austausch von Informationen über Fahrzeugverleihanbietern und statischen Daten von Buchungszielen. Hierzu gehören Anbieter, Standort- und Fahrzeugdaten. Dienst 1 kann beispielsweise dafür verwendet werden, um nur Hinweise zu vorhandenen Standorten eines FVS-Betreibers in einem RIS anzuzeigen.

Funktionen

- Abruf von Buchungsziel- und Betreiberinformationen

Abhängigkeiten

keine

3.5 Dienst 2 – Verfügbarkeitsauskunft (availability query)

Dienst 2 dient dem synchronen Abruf von Verfügbarkeitsinformationen. Die tatsächlichen Verfügbarkeitszeiten von Buchungszielen werden während der Reiseauskunft durch das RIS beim FVS abgerufen.

Funktionen

- Abruf von Verfügbarkeiten von Buchungszielen
- Abruf von Standortkapazitäten (Dienst 2a)

Abhängigkeiten

- Dienst 1

3.6 Dienst 3 – Verfügbarkeitsabonnement (availability subscription)

Dienst 3 dient dem asynchronen Austausch von Verfügbarkeitsinformationen. Um die Reiseauskunft zu beschleunigen, kann das RIS Verfügbarkeitszeiträume von Buchungszielen abonnieren um eine Abfrage während der Reiseauskunft zu vermeiden. Nach dem Abonnement einer Menge von Buchungszielen, informiert das FVS fortlaufend über Änderungen an Verfügbarkeitszeiträumen.

Funktionen

- Abonnement von Verfügbarkeiten
- Verfügbarkeitsinformation (push)
- Abonnement von Standortkapazitäten (Dienst 3a)

Abhängigkeiten

- Dienst 1
- Basisdienst B

3.7 Dienst 4 – Buchung (booking)

Dienst 4 dient der Buchung, Umbuchung und Stornierung von Fahrzeugen durch das RIS im Auftrag eines Kunden des FVS. Für die Buchung eines Fahrzeuges ist es notwendig, dass der Kunde sich gegenüber dem FVS authentifiziert. Dazu werden die Authentifizierungsinformationen vom RIS an das FVS weitergeleitet. Das Geheimnis (Passwort, PIN, etc.) des Kunden darf vom RIS aus Sicherheitsgründen nicht gespeichert werden. Das RIS erhält bei erfolgreicher Authentifizierung eines Kunden für diesen ein Authentifizierungs-Token. Dieser kann entweder im RIS oder auf dem Endgerät des Kunden gespeichert werden. Unter Verwendung des Authentifizierungs-Tokens kann das RIS Anfragen zur Buchung und Stornierung von Buchungen durchführen. Um dem Nutzer die Änderung einer Buchung zu ermöglichen, kann eine Buchung durch eine Umbuchungsanfrage des RIS durch eine neue Buchung ersetzt werden. Dabei muss bei neuen Buchungen sichergestellt werden, dass eine alte Buchungen ihre Gültigkeit behält, wenn keine neue Buchung möglich ist.

Funktionen

- Anfrage zur Buchung
- Anfrage zur Buchungsänderung / Stornierung
- Abonnement von externen Buchungen (Dienst 4a)

Abhängigkeiten

- Dienst 1

3.8 Dienst 5 – Buchungsabonnement (booking subscription)

Dienst 5 dient dem Abonnement von Buchungsänderungen. Das RIS kann durchgeführte Buchungen beim FVS abonnieren, um den Benutzer bei Änderungen (z. B. durch ein defektes Fahrzeug) informieren zu können.

Funktionen

- Buchungsabonnement
- Buchungsalarm (Push)

Abhängigkeiten

- Dienst 4
- Basisdienst B

3.9 Dienst 6 – Preisauskunft (price information)

Dienst 6 dient der Preisauskunft von Verleihdienstleistungen. Durch die Übermittlung von Start-, Zielort und Startzeitpunkt und Endzeitpunkt der Fahrt kann das RIS beim FVS eine unverbindliche Preisauskunft zur Information des Benutzers einholen. Das FVS antwortet mit einem Preis und ggfs. Einzelposten.

Funktionen

- Anfrage eines Preises

Abhängigkeiten

- Dienst 1

3.10 Dienst 7 – Abonnement Verbrauchsdaten / Abrechnung (consumption data)

Dienst 7 dient dem Austausch von Verbrauchsdaten wie z. B. Nutzungsdauer und Entfernung. Durch die Übermittlung der Verbrauchsdaten kann das RIS Rechnungen für Endkunden erstellen. Das RIS abonniert hierzu die Verbrauchsdaten einer durchgeführten Buchung beim FVS und erhält dann automatisch die zu dieser Buchung gehörigen Verbrauchsdaten.

Funktionen

- Abonnement Verbrauchsdaten
- Austausch Verbrauchsdaten (Push)

Abhängigkeiten

- Dienst 4
- Basisdienst B

3.11 Dienst 8 – Buchungsstatus ändern (freischalten / pausieren / abschließen) (booking state change)

Dienst 8 dient dem Freischalten von Buchungen respektive Fahrzeugen (ggfs. Schlüsselkasten) durch das RIS. Dies erlaubt sowohl die Öffnung bzw. Freischaltung als auch die Rückgabe des Fahrzeug innerhalb der Endanwenderanwendung des RIS, z. B. innerhalb einer mobilen App. Weiterhin kann eine Buchung pausiert werden, dies schließt das Fahrzeug physikalisch ab, beendet aber nicht die Buchung. Der Status bezieht sich immer auf eine vorausgegangene Buchung und eine Authentifizierung ist erforderlich.

Funktionen

- Buchung freischalten
- Buchung pausieren
- Buchung abschließen

Abhängigkeiten

- Dienst 4
- Basisdienst A

3.12 Dienst 9 – Anlegen und Sperren von Nutzern (user management)

Dienst 9 dient dem Anlegen und Sperren von FVS Nutzern durch das RIS. Dies erlaubt, dass sich Nutzer nur beim RIS Betreiber registrieren müssen und diese Registrierung auch für das FVS gilt.

Funktionen

- Benutzer anlegen
- Benutzer sperren

Abhängigkeiten

- Dienst 1
- Basisdienst A

3.13 Dienst 10 – Fahrzeug- bzw. Buchungseinstellungsmanagement (booking settings)

Dienst 10 dient dem Management von Fahrzeugeinstellungen. Das RIS kann Fahrzeugeinstellungen (Temperatur der Klimaanlage, eingestellter Radiosender etc) an das FVS übertragen und Einstellung für bestehende Buchungen abonnieren.

Funktionen

- Fahrzeugeinstellungen setzen
- Fahrzeugeinstellungen abonnieren

Abhängigkeiten

- Dienst 4
- Basisdienst A

3.14 Dienst 11 – Fernkonfiguration Navigationssystem und Fahrtverlauf überwachen

Dienst 11 dient der Integration des Fahrzeugnavigationssystems in eine intermodale Reise. Das RIS kann nach der Buchung dem Fahrzeug den Zielort übermitteln, so dass der Reisende den Ort nicht selbst eingeben muss. Zusätzlich kann das RIS den Fortschritt der Fahrt überwachen um ggfs. auf Verzögerungen zu reagieren.

Funktionen

- Navigationsziel des Fahrzeugs setzen
- Fahrtverlauf abonnieren

Abhängigkeiten

- Dienst 4
- Basisdienst A

3.15 Dienst 12 – Dialog Intermodale Alternative

Dienst 12 dient der Integration des Fahrzeugnavigationssystems in eine intermodale Reise. Falls sich durch Verzögerungen alternative Routen ergeben, kann das RIS dem Fahrzeug diese übermitteln. Der Reisende kann eine dieser Alternativen wählen und entscheiden, ob reservierungs- oder kostenpflichtige Teile der Route direkt vom RIS gebucht werden sollen.

Funktionen

- Reisenden über Alternative Routen informieren
- Wahl der Alternative und Buchungswunsch abonnieren

Abhängigkeiten

- Dienst 4
- Dienst 11
- Basisdienst A

4 Interaktionsprotokolle

Dieser Abschnitt gibt einen Überblick über die in IXSI verwendeten Interaktionsschemata. Zur Vereinfachung werden die Interaktionsschemata in den Sequenzdiagrammen informell ohne Verwendung der technischen Bezeichnungen der Funktionsaufrufe beschrieben. Grundsätzlich werden zwei Typen von Interaktionen verwendet: Das einfache und wohlbekannte Request/Response- bzw. Query-Interaktionsschema bei dem auf jede Anfrage des Clients (in diesem Fall das RIS) genau eine Antwort des Servers folgt (FVS). Weiterhin das Subscription-Schema bei dem einmalig ein Objekt durch den Client abonniert wird und dann Aktualisierungen an diesem Objekt laufend durch den Server geliefert werden. Hierbei bleibt der Kommunikationskanal die ganze Zeit über geöffnet.

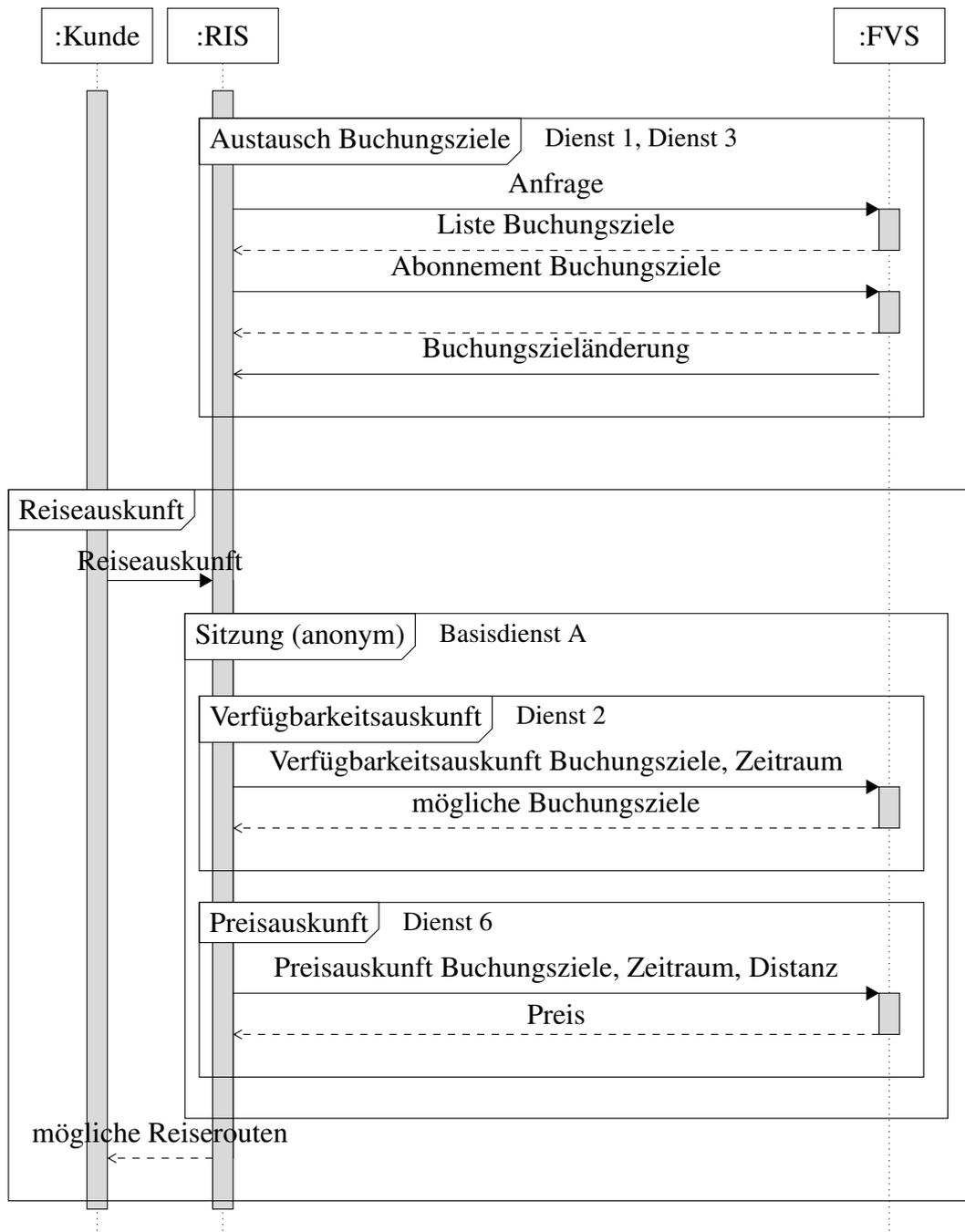
4.1 Überblick

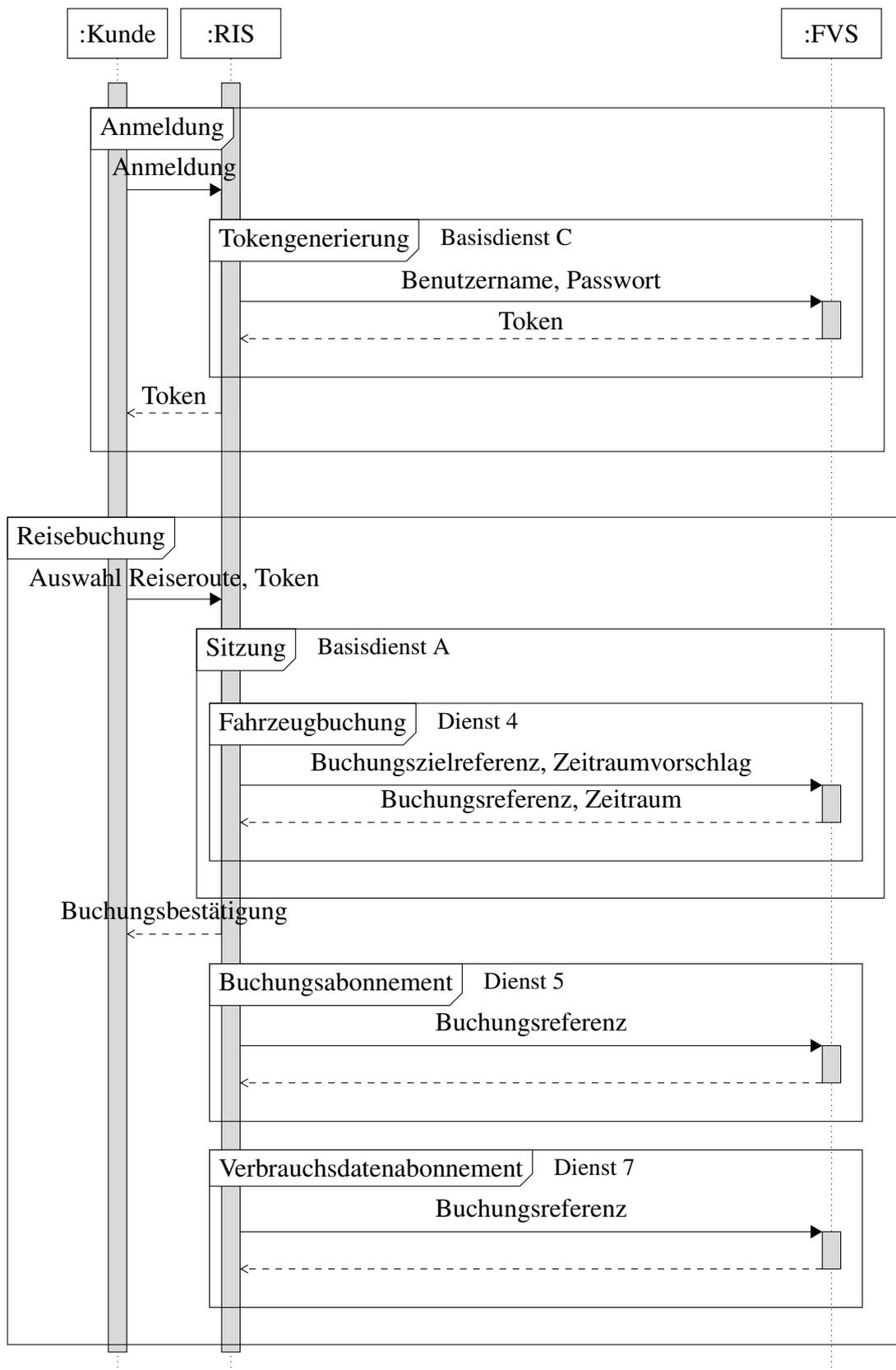
Das folgende Sequenzdiagramm gibt einen Überblick über eine beispielhafte Informationskopplung wie sie durch IXSI realisiert werden kann. Die im einzelnen verwendeten Dienste werden in den folgenden Abschnitten genauer beschrieben. In diesem Anwendungsfall holt ein Kunde beim RIS eine Reiseauskunft ein, bucht eine entsprechende Reise und lässt sich über Änderungen an der Buchung benachrichtigen.

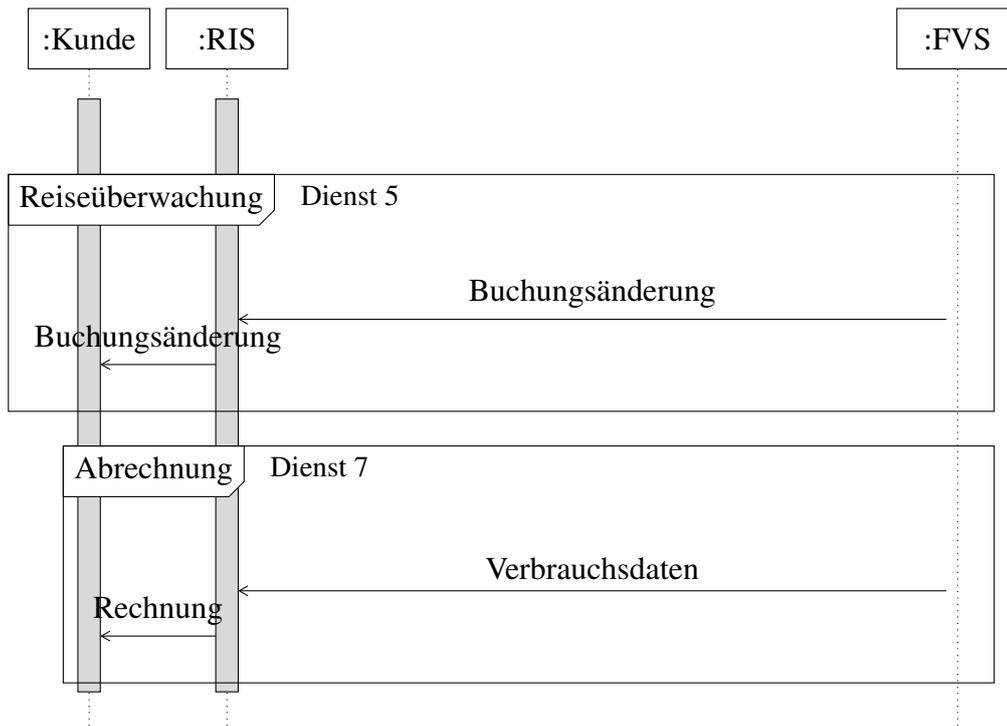
Im ersten Block *Austausch Buchungsziele* werden die vom FVS zur Verfügung gestellten Buchungsziele mit dem RIS ausgetauscht und relevante Buchungsziele abonniert (vgl. Abschnitte 4.2 und 4.4). Dies geschieht proaktiv ohne Involvierung eines Kunden. Im Block *Reiseauskunft* führt ein Kunde, z. B. mit einem Mobilgerät, eine Reiseauskunft beim RIS durch. Hierbei kommen mehrere Leihfahrzeuge in Frage, deren Verfügbarkeit dann synchron beim FVS abgefragt wird. Für die verfügbaren Fahrzeuge fragt das RIS zusätzlich eine Preisauskunft an. Als Ergebnis gibt das RIS eine Auswahl an möglichen Reiserouten / Verbindungen an den Kunden zurück. Da es sich um durch einen Kunden ausgelöste Kommunikation handelt, wird implizit eine Sitzung erstellt, in deren Kontext die Abfragen ausgeführt werden. Da sich der Kunde nicht auf seinem Gerät eingeloggt hat, wird eine anonyme Sitzung verwendet. Im Block *Reisebuchung* hat sich der Kunde für eine Reiseroute entschieden und möchte diese buchen. Hierzu loggt er sich auf seinem Mobilgerät ein, wodurch ein Token generiert wird (Block *Anmeldung*). Mit diesem Token wird eine (nicht anonyme) Sitzung erstellt, in welcher der Buchungsvorgang durchgeführt wird. Hierzu übergibt das RIS die Buchungszielreferenz und einen Zeitvorschlag

4 Interaktionsprotokolle

an das FVS. Das sendet eine Buchungsbestätigung, welche vom RIS an den Kunden weitergegeben wird. Zusätzlich abonniert das RIS die entsprechende Buchung beim FVS. Im letzten Block *Reiseüberwachung* wird der Kunde über Änderungen an der Buchung, die das RIS vom FVS erhält, benachrichtigt.

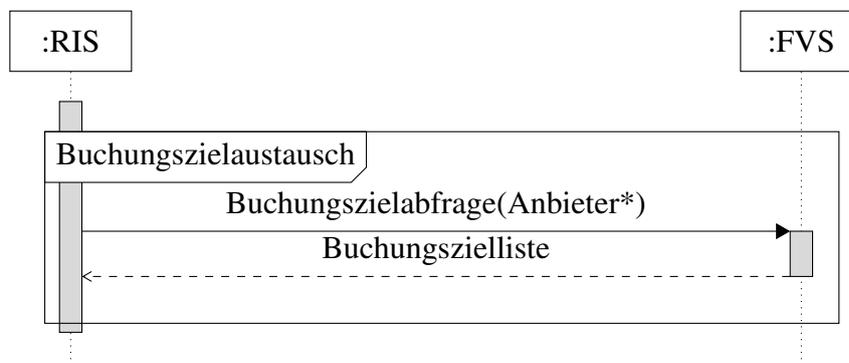






4.2 Dienst 1 – Statische Daten

Abfrage Buchungsziele

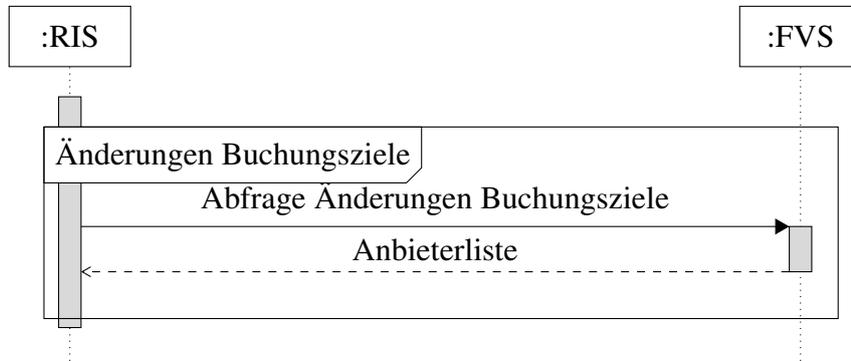


** optional*

Als Basis für die Informationskopplung dient der Austausch von sogenannten Buchungszielen. Buchungsziele sind eine logische Repräsentation von einem oder mehreren Fahrzeugen mit gemeinsamen Eigenschaften, wie z. B. vom gleichen Anbieter bereitgestellt, gleicher Fahrzeugtyp und gleiche Verleihstation. Diese Eigenschaften sind statisch. Um nur Informationen

zu Buchungszielen eines bestimmten Anbieters zu erhalten, kann nach Provider gefiltert werden. Die Übertragung wird vom RIS ausgelöst.

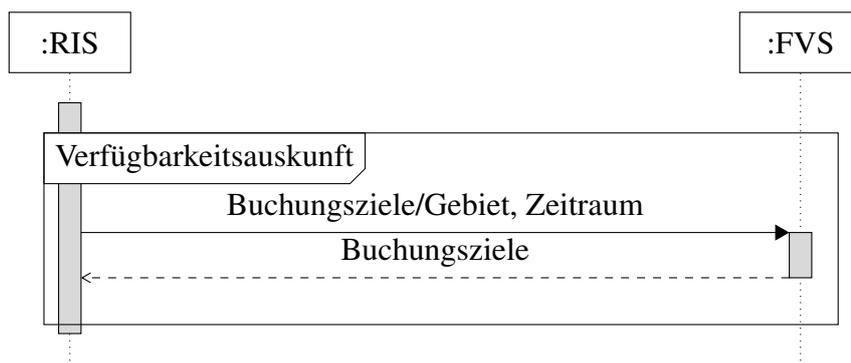
Abfrage Änderungen Buchungsziele



Um nicht intervallweise alle Buchungszielinformationen übertragen zu müssen, kann mit dem Aufruf `ChangedProviders` angefragt werden, bei welchem Anbieter sich Änderungen seit einem bestimmten Zeitpunkt, vorgegeben durch den Parameter `timestamp`, ergeben haben. Zurückgegeben wird eine Providerreferenz, die wiederum bei einem Funktionsaufruf `BookingTargetsInfo` als Parameter übergeben werden kann.

4.3 Dienst 2 – Verfügbarkeitsauskunft

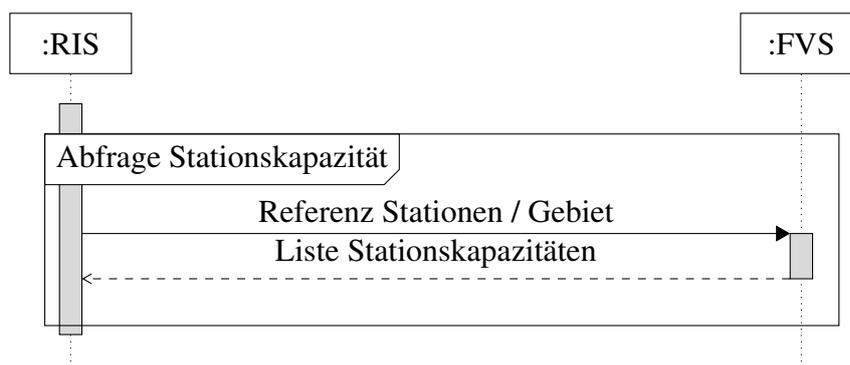
Abfrage Verfügbarkeit



Um die konkreten Verfügbarkeiten abzufragen, sendet das RIS eine Anfrage die entweder eine Liste mit Buchungszielen oder ein geographisches Gebiet in Form einer Umgebungssuche

oder als Rechteck und eine gewünschte Zeitperiode enthält. Ohne Angabe wird die Verfügbarkeit von allen Buchungszielen zurückgegeben. Als Antwort sendet das FVS eine Liste mit Buchungszielen und deren Verfügbarkeiten zurück.

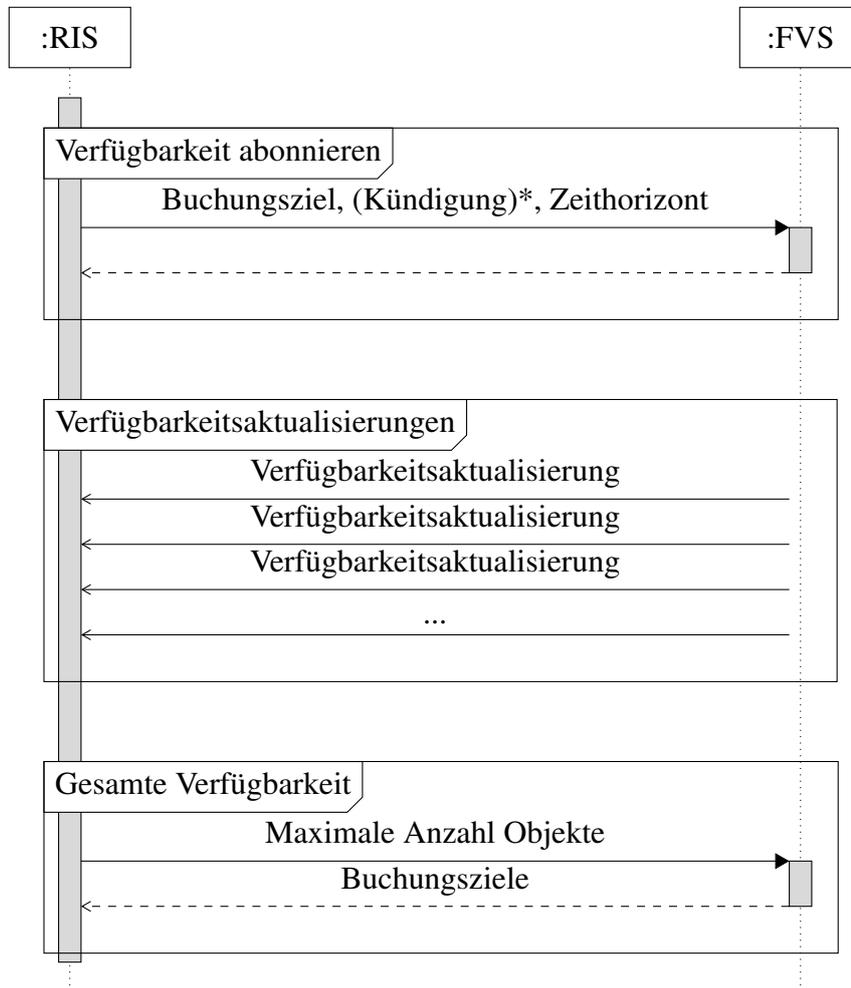
Abfrage aktuelle Stationskapazität (Dienst 2a)



Das RIS kann die aktuellen Kapazitäten, bspw. zur Kartendarstellung, von Verleihstation anfragen. Hierzu wird eine Liste mit Standort IDs oder ein Gebiet übermittelt und eine Liste mit Standorten und deren aktueller Anzahl verfügbarer Fahrzeuge zurückgegeben.

4.4 Dienst 3 – Verfügbarkeitsabonnement

Verfügbarkeitsabonnement



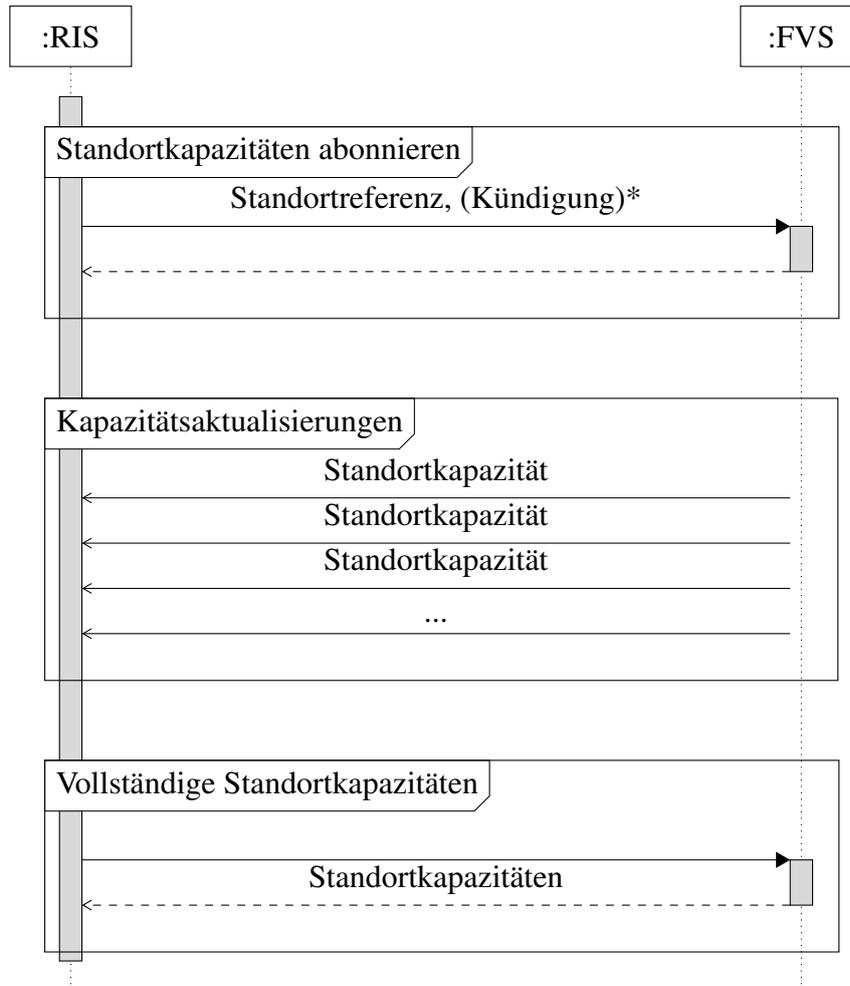
Das RIS kann Informationen zu Buchungszielen abonnieren, um unmittelbar über Änderungen von Verfügbarkeiten informiert zu werden. Dies dient im Wesentlichen dazu, Reiseauskünfte ohne zusätzliche (synchrone) Anfrage an das FVS beantworten zu können.

Durch die initiale Anfrage `AvailabilitySubscriptionRequest` wird ein Abonnement (subscription) begonnen. Hierzu übergibt das RIS die entsprechende Buchungszielreferenz. Durch das Setzen des Flags `Kündigung` kann ein Abonnement storniert werden. Bei Änderungen an Verfügbarkeiten überträgt das FVS asynchron `AvailabilityPushMessages`. Diese werden über den gleichen Kommunikationskanal geliefert, über den das Abonnement erstellt wurde. Beim Beenden des Kommunikationskanals werden alle Abonnements hinfällig.

4 Interaktionsprotokolle

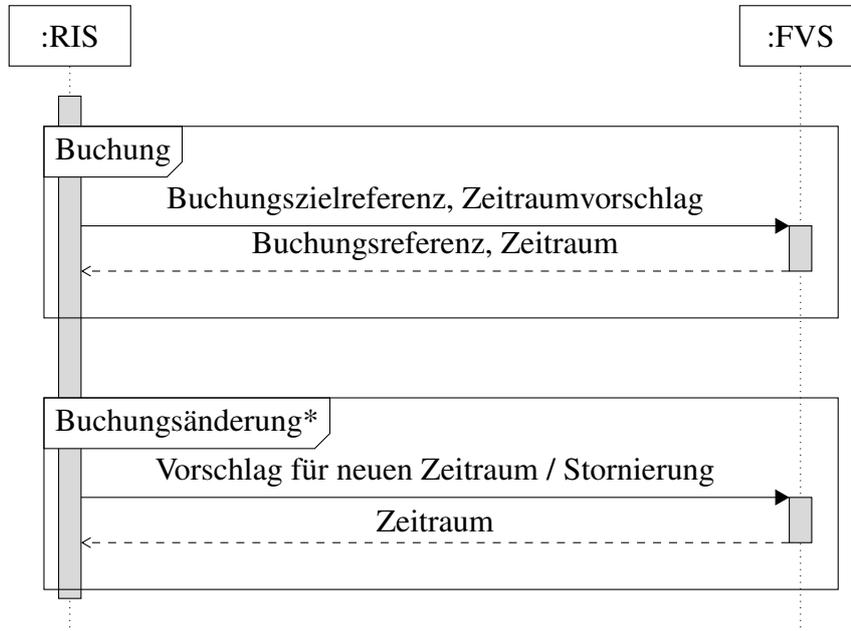
Zur anfänglichen Synchronisierung aller Verfügbarkeiten kann das RIS die Funktion CompleteAvailabilityRequest aufrufen.

Standortkapazitätsabonnement (Dienst 3a)



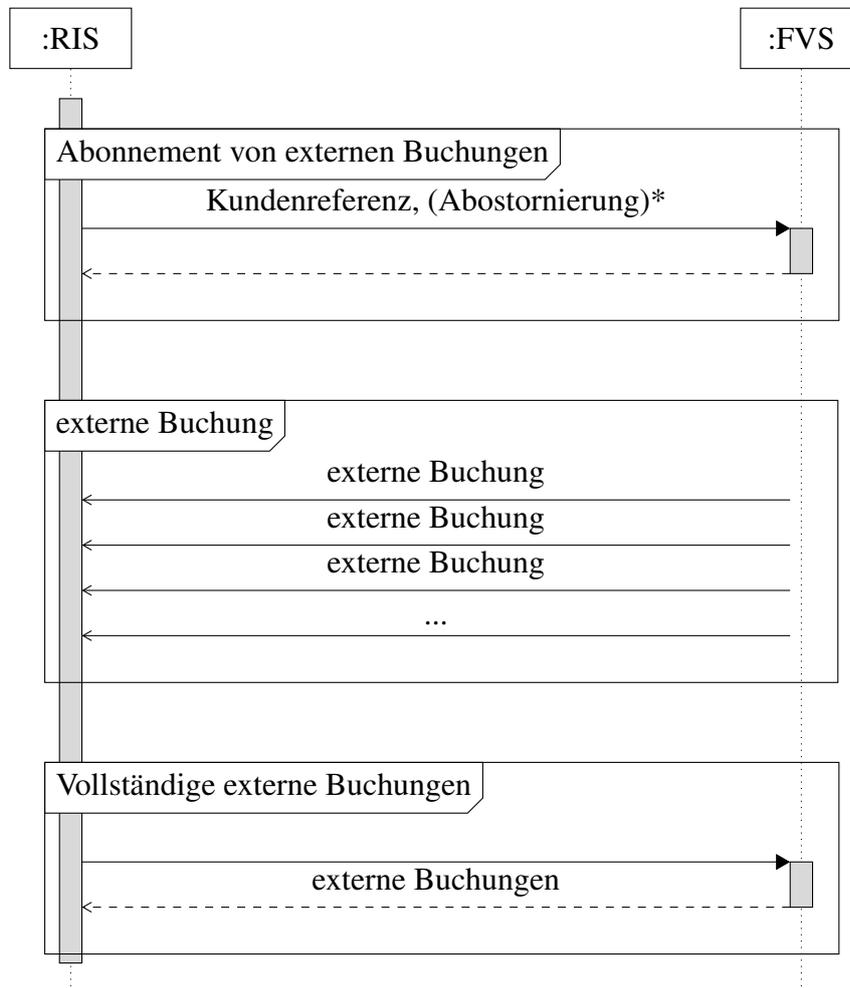
Das RIS kann die Kapazitätsinformation von Standorten abonnieren. Der Interaktionsablauf ist analog zu Abschnitt 4.4.

4.5 Dienst 4 – Buchung / Buchungsänderung



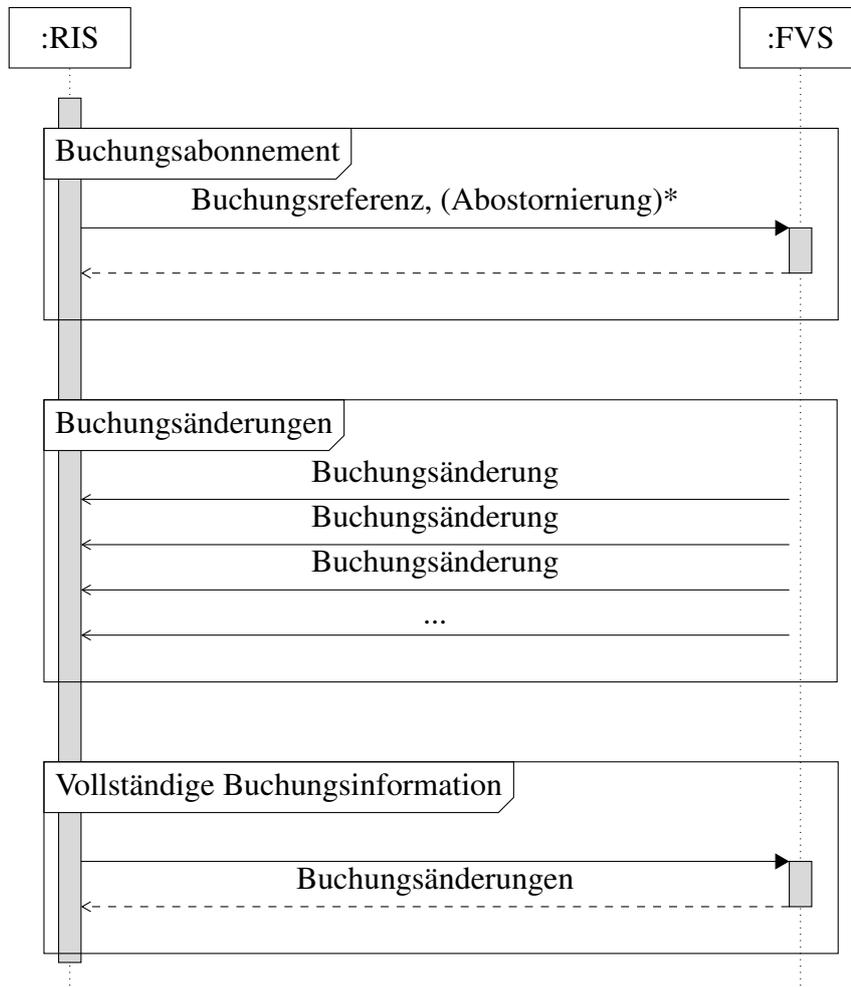
Um im Kundenauftrag ein Fahrzeug zu buchen, ist es erforderlich, dass das RIS den Kunden gegenüber dem FVS authentifiziert. Hierzu gibt es drei Möglichkeiten, die in Abschnitt 5.14 genauer dargestellt sind. In diesem Beispiel wird explizit eine Sitzung geöffnet und im Anschluss an die Transaktion wieder geschlossen. Danach kann eine Buchung durch den Aufruf von `Booking` mit Angabe der entsprechenden Buchungsziel ID und einem Vorschlag für einen Zeitraum durchgeführt werden. „Vorschlag“ deshalb, da das FVS z. B. den Zeitraum auf das verwendete Buchungsraster ändern kann. Als Antwort wird die verwendete Buchungsreferenz und der tatsächliche Buchungszeitraum zurückgegeben. Die Buchungsreferenz kann zur Überwachung der Buchung verwendet werden (vgl. Abschnitt 4.6). Zur Änderung des Buchungszeitraums oder zur Stornierung kann `ChangeBooking` aufgerufen werden. Bei Änderung des Buchungsziels ist eine Stornierung und Neubuchung erforderlich.

Abonnement externe Buchungen (Dienst 4a)



Das RIS kann extern durchgeführte Buchungen (z. B. direkt an der Verleihstation ohne Mitwirken des RIS) beim FVS abonnieren, um Änderungen an diesen zu abonnieren oder entsprechende Verbrauchsdaten abzufragen. Hierzu wird eine oder mehrere Kundenreferenzen übergeben.

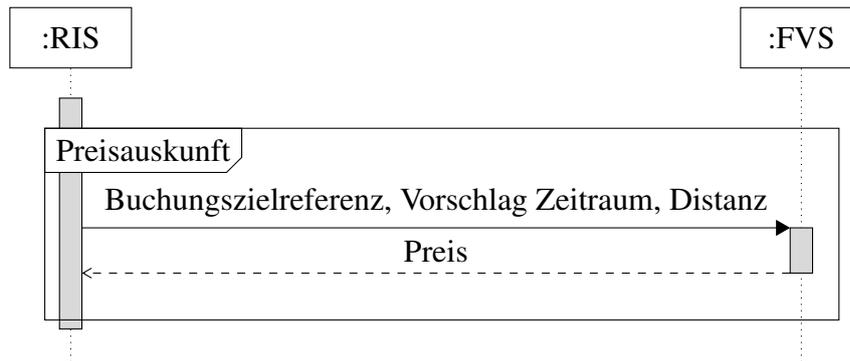
4.6 Dienst 5 – Buchungsabonnement



Das RIS kann Änderungen an Buchungen abonnieren, um diese Informationen dem Kunden weiterzugeben und ggfs. Alternativen anzubieten. Beispielsweise im Falle eines technischen Defekts an einem Fahrzeug kann das FVS das RIS darüber informieren, dass die Buchung nicht mehr möglich ist. Ebenfalls ist es möglich, eine Buchung als „wieder möglich“ festzulegen. Endgültig storniert werden kann eine Buchung nur vom Endkunden.

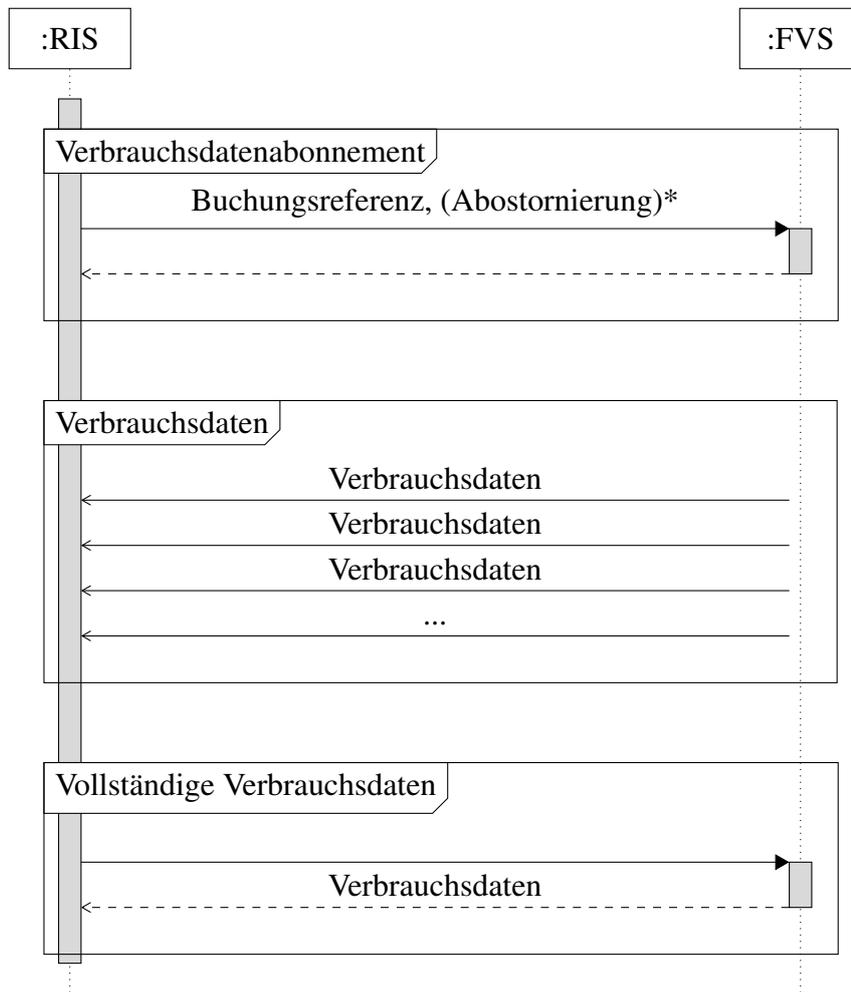
Der Interaktionsablauf ist analog zu Abschnitt 4.4.

4.7 Dienst 6 – Preisauskunft



Mit einer Anfrage `PriceInformationRequest` kann das RIS beim FVS eine Preisauskunft auf Basis von Buchungsziel ID, Zeitraum und zurückzulegende Distanz anfragen. Falls vorher eine Authentifizierung des Endkunden z. B. durch `OpenSession` stattgefunden hat, ist die Preisanfrage entsprechend des Kundenvertrags zu beantworten.

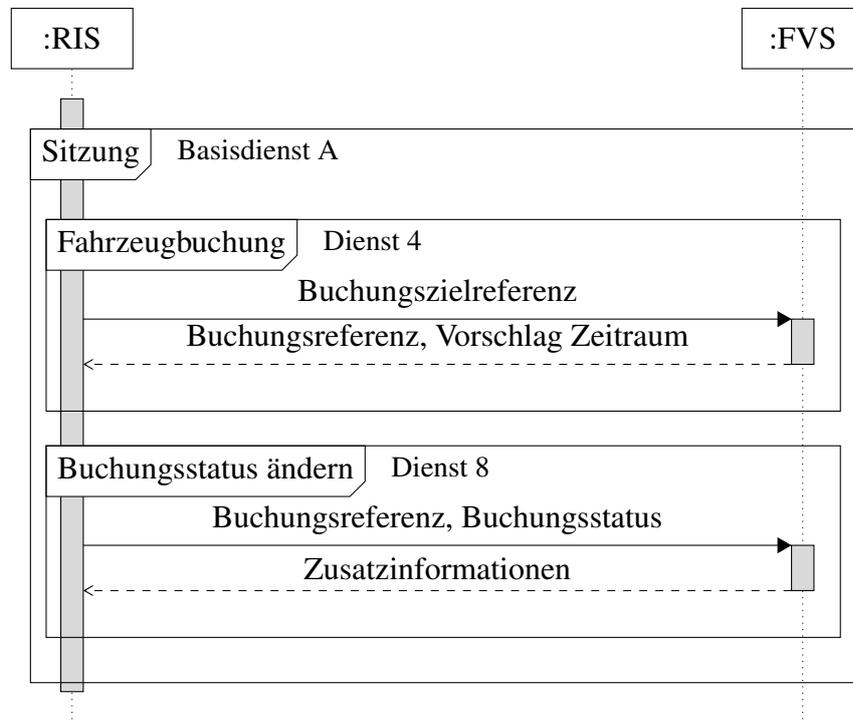
4.8 Dienst 7 – Verbrauchsdatenabonnement



Das RIS kann Verbrauchsdaten (Nutzungsdauer, Entfernung, ggfs. andere) für eine Buchung abonnieren. Sobald neue Verbrauchsdaten vorliegen (z. B. bei Rückgabe eines Fahrzeuges) informiert das FVS das RIS über den vorliegenden Verbrauch. Zu einer Buchung können mehrere Informationen zu Verbrauchsdaten versendet werden. Neue Verbrauchsdaten zu einer Buchung invalidieren alle vorherigen und müssen entsprechend vollständig sein.

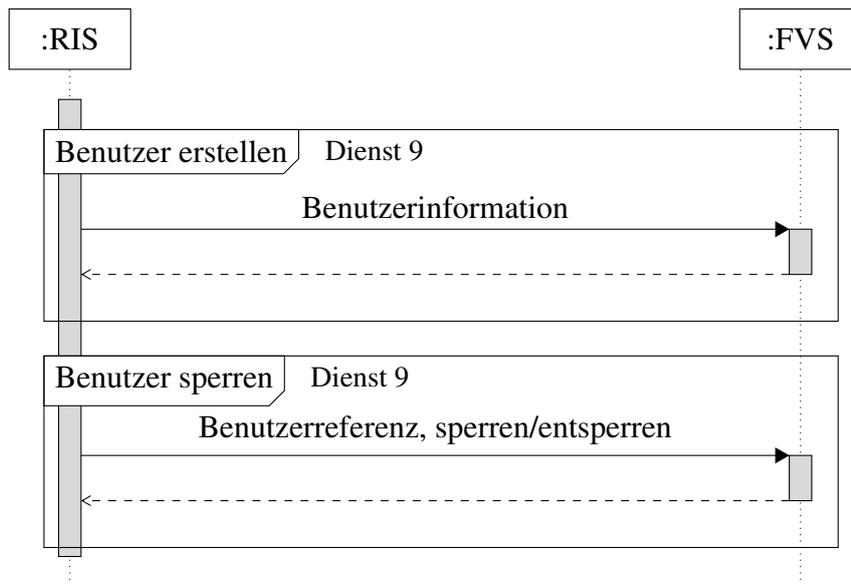
Der Interaktionsablauf ist analog zu Abschnitt 4.4.

4.9 Dienst 8 – Buchungsstatus ändern



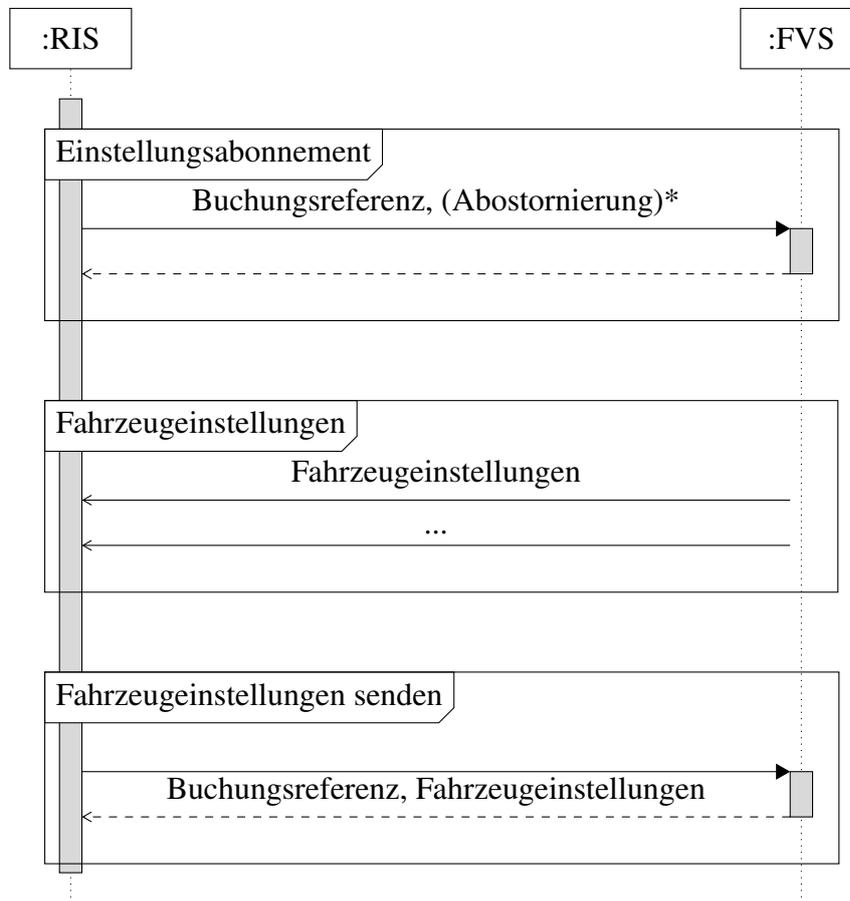
Um ein Buchungsziel (Fahrzeug bzw. Schlüsselkasten) durch das RIS freischalten zu lassen, muss eine Buchung für dieses Buchungsziel bereits vorliegen (diese muss ggfs. für den Nutzer transparent durchgeführt werden). Mit einer Referenz auf diese Buchung kann das Buchungsziel dann freigeschaltet werden. Eine Authentifizierung ist hierfür erforderlich. Pausieren und Abschließen einer Buchung passiert analog.

4.10 Dienst 9 – Benutzermanagement



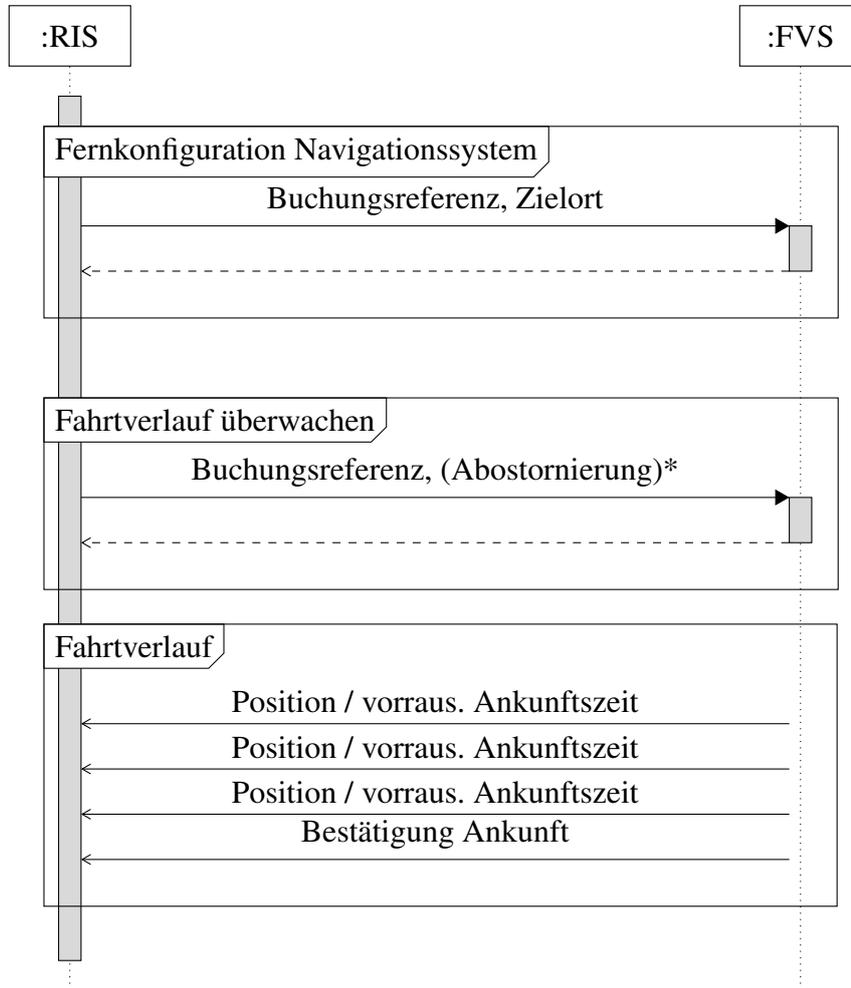
Das RIS legt Benutzer im FVS an, um eine doppelte Anmeldung bei beiden Diensten überflüssig zu machen. Optional können Benutzer auch gesperrt/entsperrt werden.

4.11 Dienst 10 – Fahrzeugeinstellungsmanagement



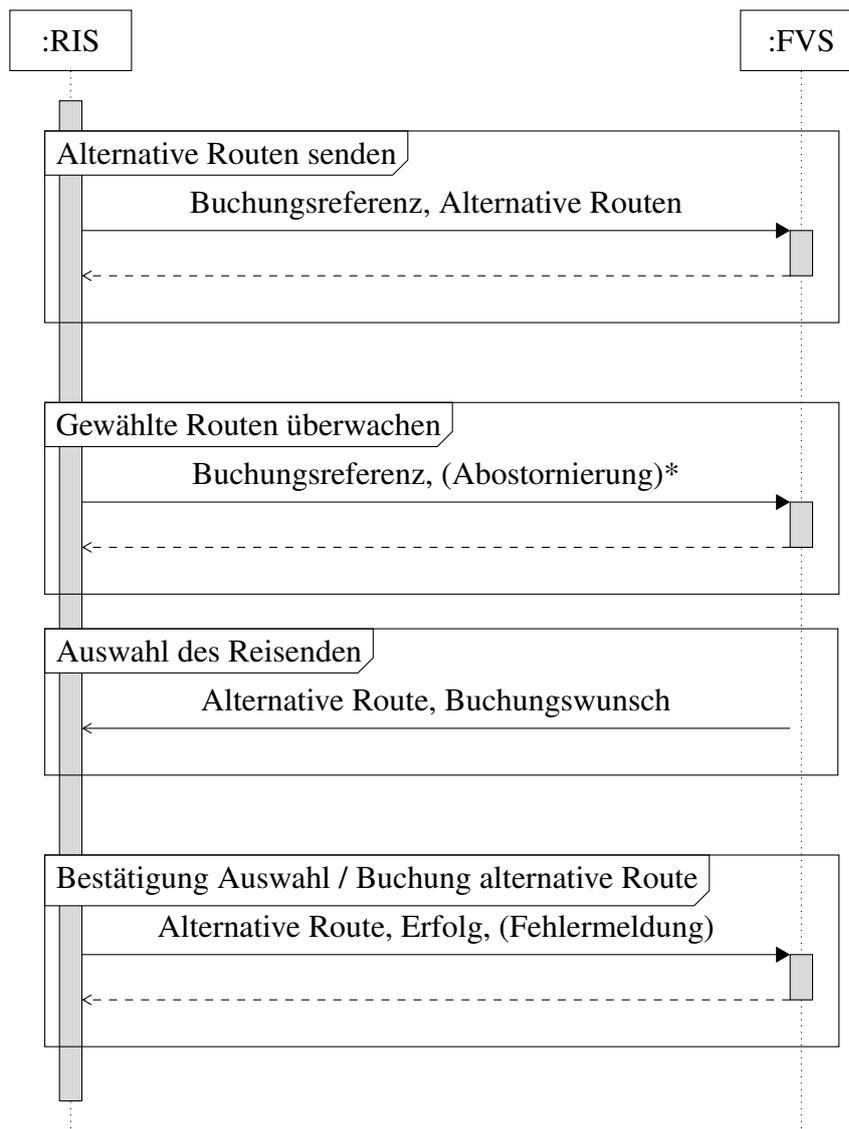
Das RIS kann Buchungen bzw. Fahrzeugeinstellungen beim FVS abonnieren. Nach einer abgeschlossenen Buchung (oder in Intervallen) übermittelt das FVS die durch den Benutzer durchgeführten Fahrzeugeinstellungen wie Sitzposition, Zieltemperatur der Klimaanlage, eingestellter Radiosender usw. Kurz vor dem Start einer Folgebuchung kann das RIS dann wiederum Fahrzeugeinstellung übermitteln damit der Benutzer bei betreten des Fahrzeugs die gleiche Konfiguration wie vorher wiederfindet.

4.12 Dienst 11 – Fernkonfiguration Navigationssystem und Fahrtverlauf überwachen



Das RIS kann nach der Buchung dem Fahrzeug den Zielort übermitteln (Fernkonfiguration Navigationssystem), so dass der Reisende den Zielort nicht selbst eingeben muss. Außerdem kann das RIS den Fortschritt der Fahrt abonnieren, und wird dann fortlaufend vom FVS über Änderungen der Position und der voraussichtlichen Ankunftszeit informiert.

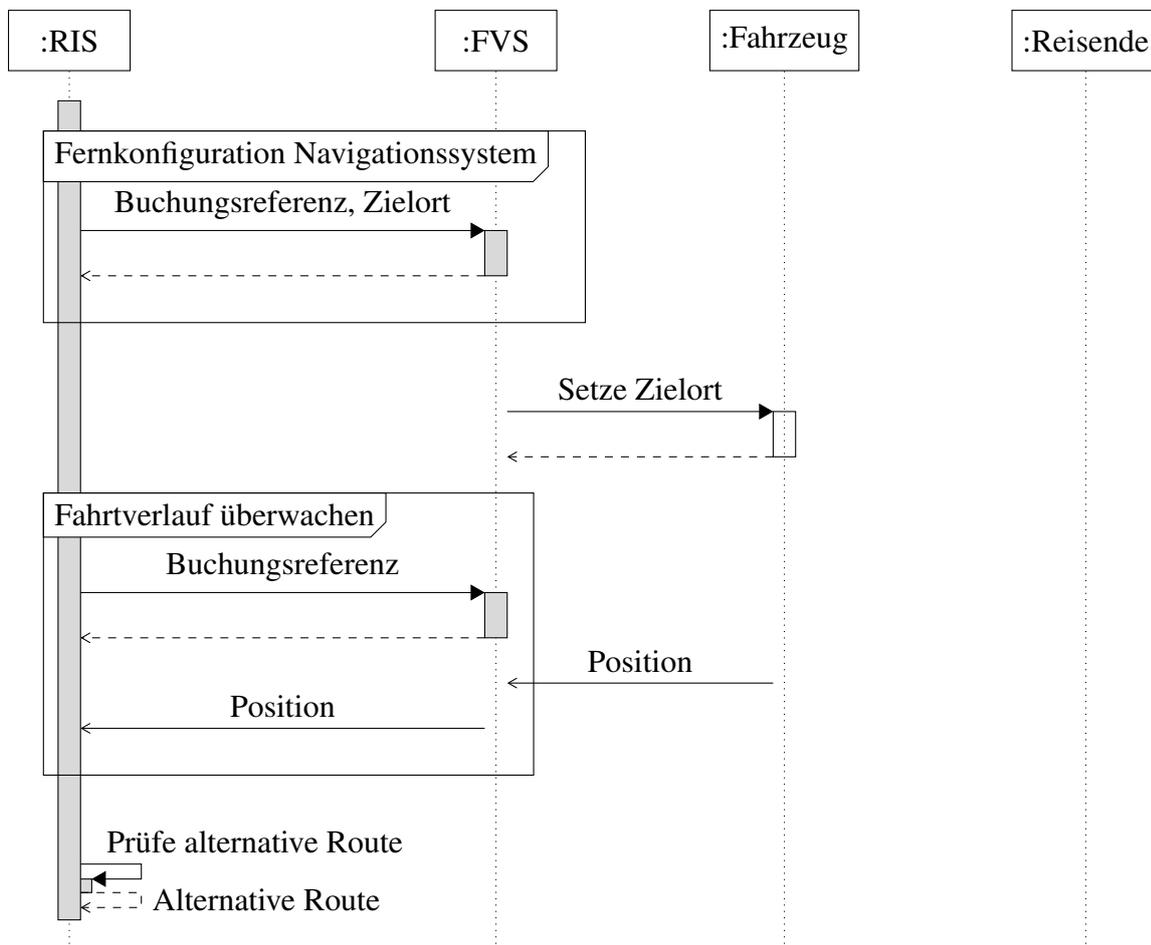
4.13 Dienst 12 – Dialog Intermodale Alternative



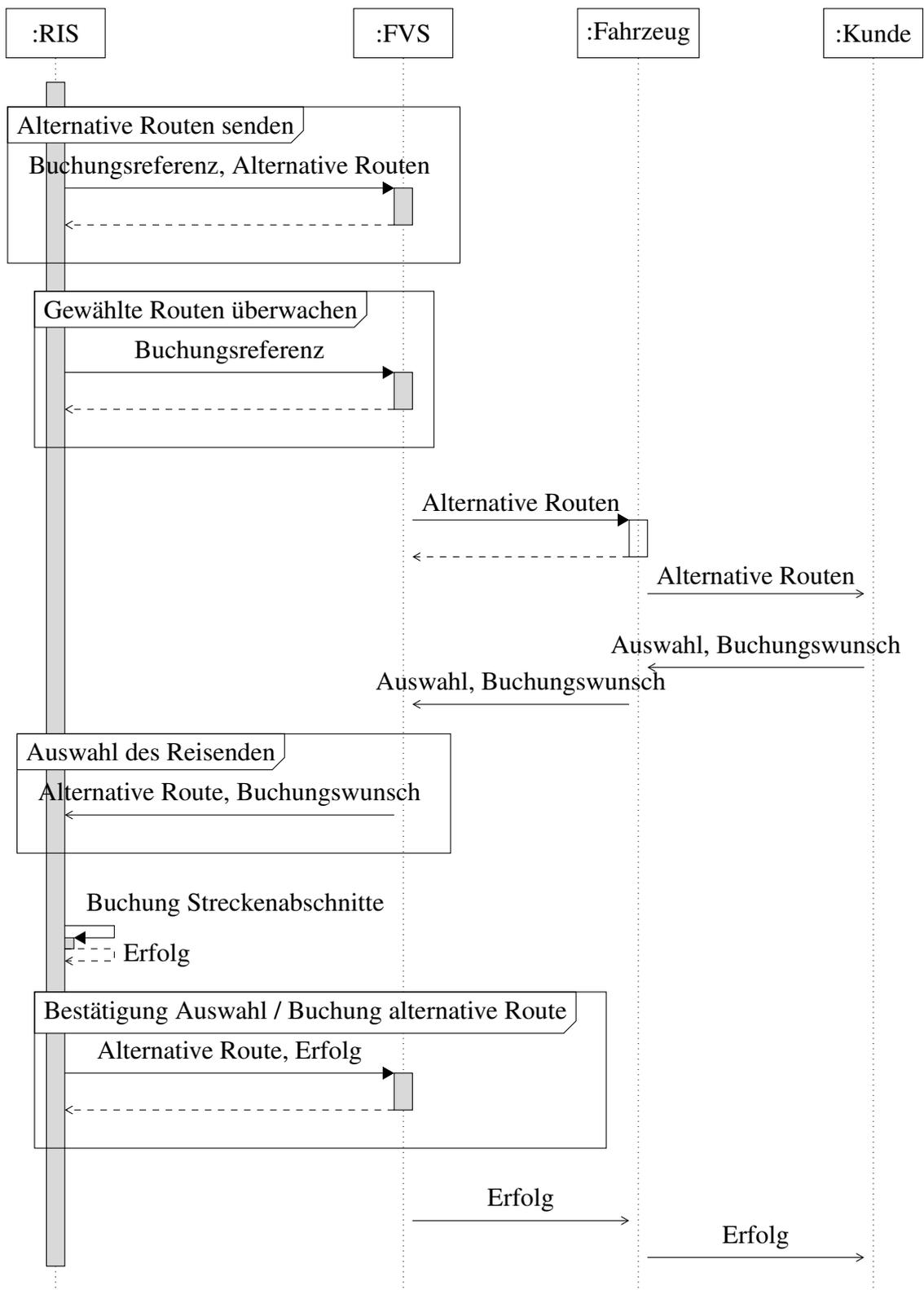
Das RIS kann, falls alternative Routen erkannt werden, diese an das Fahrzeug übermitteln (Alternative Routen senden). Dem Nutzer werden diese Alternativen dargestellt. Er kann eine von diesen Wählen und zusätzlich angeben, dass alle Teile der Route gebucht werden sollen. Um über die Wahl des Nutzers benachrichtigt zu werden kann das RIS diese beim FVS abonnieren. Nachdem die Route des Reisenden erfolgreich aktualisiert und gewünschte Teile gebucht wurden, schickt das RIS eine Bestätigung an das FVS oder benachrichtigt über mögliche Fehler.

Das folgende Sequenzdiagramm gibt einen Überblick über eine beispielhafte Interaktion bestehend aus Dienst 11 und 12. In diesem Anwendungsfall konfiguriert zunächst das RIS das Navigationssystem des vom Kunden gebuchten Fahrzeugs und startet eine Überwachung des

Fahrtverlaufs. Nachdem das RIS über eine Positionsänderung informiert wird, findet es eine alternative intermodale Route. Diese wird über das FVS an das Fahrzeug übertragen, welches diese dem Reisenden über einen Dialog darstellt. Dieser kann dann eine der alternativen Route wählen und falls gewünscht die gesamte Route buchen. Gleichzeitig erstellt das RIS ein Abonnement um über die Auswahl des Reisenden benachrichtigt zu werden. Sobald es über eine Auswahl benachrichtigt wird, bucht das RIS die entsprechenden Teile der Route und informiert das FVS.



4 Interaktionsprotokolle



5 Datenmodell

Dieser Abschnitt beschreibt das der Schnittstelle zugrunde liegende Datenmodell.

Symbollegende:

- Auswahl (`xs:choice`)
- Optional (`minOccurs=0`)
- Mehrwertig (`maxOccurs=0`)

5.1 Basisdatentypen

Einfache Basistypen

Einfache Basistypen sind Aliasnamen für vorhandene Datentypen, um eine semantische Unterscheidung zu erlauben.

Name	Basistyp	Kommentar
<i>AreaIDType</i>	xs:token	Typ für Gebiet-IDs.
<i>AttributeClassType</i>	xs:token	Aufzählung von Attributklassen.
<i>AttributeIDType</i>	xs:token	Aufzählung von Attributcodes.
<i>BookeeIDType</i>	xs:token	Typ für bookee IDs.
<i>BookingIDType</i>	xs:token	Typ für Buchungs-IDs.
<i>BookingSettingsClassType</i>	xs:token	Aufzählung von Buchungseinstellungsklassen.
<i>BookingStateType</i>	xs:token	Aufzählung von verschiedenen Buchungszuständen.
<i>ClassType</i>	xs:token	Aufzählung von Autoklassen.
<i>ConsumptionClassType</i>	xs:token	Aufzählung von Verbrauchsdatentypen.
<i>EngineType</i>	xs:token	Aufzählung von Motoren.
<i>ErrorCodeType</i>	xs:token	Aufzählung von Fehlercodes.
<i>EuroPriceType</i>	xs:nonNegativeInteger	Typ für Preise in Euro-Cents.
<i>GlobalIDType</i>	xs:token	Type for globally unique IDs for identifying clone entities.
<i>ItineraryIDType</i>	xs:token	Typ für intermodale Routen-IDs.
<i>PercentType</i>	xs:nonNegativeInteger	Typ für Prozent zwischen 0 und 100.
<i>PlaceGroupIDType</i>	xs:token	Typ für Standortgruppen-IDs.
<i>PlaceIDType</i>	xs:token	Typ für Standort-IDs.
<i>ProviderIDType</i>	xs:token	Typ für Anbieter-IDs.
<i>SessionIDType</i>	xs:token	Typ für Session-IDs.
<i>SystemIDType</i>	xs:token	Typ für System-IDs.
<i>UserFeatureClassType</i>	xs:token	Aufzählung für Eigenschaften von Benutzern.
<i>UserIDType</i>	xs:token	Typ für User-IDs.
<i>UserStateType</i>	xs:token	Aufzählung für Benutzerzustand.

```

1 <xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="AreaIDType">
2   <xs:restriction base="xs:token"/>
3 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
  ↳ /2001/XMLSchema" name="PlaceIDType">
4   <xs:restriction base="xs:token"/>
5 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
  ↳ /2001/XMLSchema" name="PlaceGroupIDType">
6   <xs:restriction base="xs:token"/>
7 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
  ↳ /2001/XMLSchema" name="BookeeIDType">
8   <xs:restriction base="xs:token"/>
9 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
  ↳ /2001/XMLSchema" name="ProviderIDType">
10  <xs:restriction base="xs:token"/>

```

```

11 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="SessionIDType">
12   <xs:restriction base="xs:token"/>
13 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="BookingIDType">
14   <xs:restriction base="xs:token"/>
15 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="ItineraryIDType">
16   <xs:restriction base="xs:token"/>
17 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="UserIDType">
18   <xs:restriction base="xs:token"/>
19 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="SystemIDType">
20   <xs:restriction base="xs:token"/>
21 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="GlobalIDType">
22   <xs:restriction base="xs:token"/>
23 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="EuroPriceType">
24   <xs:restriction base="xs:nonNegativeInteger"/>
25 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="PercentType">
26   <xs:restriction base="xs:nonNegativeInteger">
27     <xs:maxInclusive value="100"/>
28   </xs:restriction>
29 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="ErrorCodeType">
30   <xs:restriction base="xs:token"/>
31 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="ClassType">
32   <xs:restriction base="xs:token"/>
33 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="EngineType">
34   <xs:restriction base="xs:token"/>
35 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="AttributeClassType">
36   <xs:restriction base="xs:token"/>
37 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="ConsumptionClassType">
38   <xs:restriction base="xs:token"/>
39 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="BookingSettingsClassType">
40   <xs:restriction base="xs:token"/>
41 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="AttributeIDType">
42   <xs:restriction base="xs:token"/>
43 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="UserStateType">
44   <xs:restriction base="xs:token"/>
45 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="UserFeatureClassType">
46   <xs:restriction base="xs:token"/>
47 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="BookingStateType">
48   <xs:restriction base="xs:token">
49     <xs:enumeration value="open"/>
50     <xs:enumeration value="suspended"/>
51     <xs:enumeration value="closed"/>
52   </xs:restriction>
53 </xs:simpleType>

```

XML Quellcode 5.1: Baisdatentypen

Erlaubte Werte für die Aufzählungen *ClassType*, *EngineType*, *AttributeClassType* und *ErrorCodeType* sind in Kapitel 8 dargestellt.

Text

TextType: Typ für einen Text.

Element	Typ	Kommentar
<i>Text</i>	xs:string	
□ <i>Language</i>	xs:language	Sprach-Identifer entsprechend BCP 47, welcher auf ISO 639 basiert ist.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="TextType">
2   <xs:sequence>
3     <xs:element name="Text" type="xs:string"/>
4     <xs:element name="Language" type="xs:language" minOccurs="0"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 5.2: TextType

Geografische Koordinaten

CoordType: Typ für geodezimale Koordinaten. Das Referenz-Ellipsoid ist immer WGS84 (GPS).

Element	Typ	Kommentar
<i>Longitude</i>	simpleType	
<i>Latitude</i>	simpleType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CoordType">
2   <xs:sequence>
3     <xs:element name="Longitude">
4       <xs:simpleType>
5         <xs:restriction base="xs:decimal">
6           <xs:minInclusive value="-180.0"/>
7           <xs:maxInclusive value="180.0"/>
8         </xs:restriction>
9       </xs:simpleType>
10    </xs:element>

```

```

11 <xs:element name="Latitude">
12   <xs:simpleType>
13     <xs:restriction base="xs:decimal">
14       <xs:minInclusive value="-90.0"/>
15       <xs:maxInclusive value="90.0"/>
16     </xs:restriction>
17   </xs:simpleType>
18 </xs:element>
19 </xs:sequence>
20 </xs:complexType>

```

XML Quellcode 5.3: CoordType

Adresse

AddressType: Typ um eine Adresse zu speichern.

Element	Typ	Kommentar
<input type="checkbox"/> <i>Country</i>	xs:string	Name des Landes.
<input type="checkbox"/> <i>PostalCode</i>	xs:string	Postleitzahl / ZIP-Code.
<i>City</i>	xs:string	Name der Stadt.
<input type="checkbox"/> <i>StreetHouseNr</i>	xs:string	Straße und Hausnummer.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   → name="AddressType">
2   <xs:sequence>
3     <xs:element name="Country" type="xs:string" minOccurs="0"/>
4     <xs:element name="PostalCode" type="xs:string" minOccurs="0"/>
5     <xs:element name="City" type="xs:string"/>
6     <xs:element name="StreetHouseNr" type="xs:string" minOccurs="0"/>
7   </xs:sequence>
8 </xs:complexType>

```

XML Quellcode 5.4: AddressType

Ort

GeoPositionType: Typ für Geografische Ortsangabe.

Element	Typ	Kommentar
<i>Coord</i>	CoordType	Geographische Koordinaten.
<input type="checkbox"/> <i>Address</i>	AddressType	Adresse der geographischen Position (optional).

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="GeoPositionType">
2   <xs:sequence>
3     <xs:element name="Coord" type="CoordType"/>
4     <xs:element name="Address" type="AddressType" minOccurs="0"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 5.5: GeoPositionType

Kreis

GeoCircleType: Typ für einen geographischen Kreis.

Element	Typ	Kommentar
<i>Center</i>	CoordType	Koordinaten des Kreis zentrums.
<i>Radius</i>	xs:nonNegativeInteger	Radius des Kreises in Metern.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="GeoCircleType">
2   <xs:sequence>
3     <xs:element name="Center" type="CoordType"/>
4     <xs:element name="Radius" type="xs:nonNegativeInteger"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 5.6: GeoCircleType

Rechteck

GeoRectangleType: Typ für ein geographisches Rechteck.

Element	Typ	Kommentar
<i>UpperLeft</i>	CoordType	Obere linke Ecke des geographischen Rechtecks.
<i>LowerRight</i>	CoordType	Untere rechte Ecke des geographischen Rechtecks.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="GeoRectangleType">
2   <xs:sequence>
3     <xs:element name="UpperLeft" type="CoordType"/>
4     <xs:element name="LowerRight" type="CoordType"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 5.7: GeoRectangleType

Gebiet

GeoAreaType: Typ für geographische Gebiete.

Element	Typ	Kommentar
<input type="checkbox"/> <i>PolyPoint</i>	CoordType	Geopositionen der nächstgelegenen Polygongrenze des Gebiets.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="GeoAreaType">
2   <xs:sequence>
3     <xs:element name="PolyPoint" type="CoordType" minOccurs="3" maxOccurs="unbounded"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 5.8: GeoAreaType

Gebiet (Einschluss / Ausschluss)

IncExcGeoAreaType: Typ für geographische Gebiete mit Ausschlusswahrscheinlichkeit.

Basistyp: *GeoAreaType*

Element	Typ	Kommentar
<input type="checkbox"/> <i>Exclude</i>	xs:boolean	Nur im Falle von "wahr" definiert diese Gebietdefinition einen Ausschluss.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="IncExcGeoAreaType">
2   <xs:complexContent>
3     <xs:extension base="GeoAreaType">
4       <xs:sequence>
5         <xs:element name="Exclude" type="xs:boolean" default="false" minOccurs="0"/>
6       </xs:sequence>
7     </xs:extension>
8   </xs:complexContent>
9 </xs:complexType>

```

XML Quellcode 5.9: IncExcGeoAreaType

Zeitperiode

TimePeriodType: Typ für eine Zeitperiode.

Element	Typ	Kommentar
<i>Begin</i>	xs:dateTime	
<i>End</i>	xs:dateTime	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="TimePeriodType">
2   <xs:sequence>
3     <xs:element name="Begin" type="xs:dateTime"/>
4     <xs:element name="End" type="xs:dateTime"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 5.10: TimePeriodType

Zeitperiode (Vorschlag)

TimePeriodProposalType: Typ für ein Zeitperioden-Vorschlag.

Basistyp: *TimePeriodType*

Element	Typ	Kommentar
<input type="checkbox"/> <i>MaxWait</i>	xs:duration	Maximale Wartezeit bis die Buchung tatsächlich beginnen kann unter Berücksichtigung des Buchungsterminrasters.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="TimePeriodProposalType">
2   <xs:complexContent>
3     <xs:extension base="TimePeriodType">
4       <xs:sequence>
5         <xs:element name="MaxWait" type="xs:duration" minOccurs="0"/>
6       </xs:sequence>
7     </xs:extension>
8   </xs:complexContent>
9 </xs:complexType>

```

XML Quellcode 5.11: TimePeriodProposalType

Herkunft/Ziel

OriginDestType: Typ für den Ausgangspunkt oder den Reisezielort.

Element	Typ	Kommentar
○ <i>PlaceID</i>	PlaceIDType	Standort-ID. Falls gegeben, bezieht sich die Lokalität auf den Standort mit dieser ID. Verwendet für standort-basierte Buchungsziele.
○ <i>GeoPosition</i>	CoordType	Falls gegeben, bezieht sich die Lokalität auf diese Geoposition. Verwendet für Free-Floating-Buchungsziele.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="OriginDestType">
2   <xs:choice>
3     <xs:element name="PlaceID" type="PlaceIDType"/>
4     <xs:element name="GeoPosition" type="CoordType"/>
5   </xs:choice>
6 </xs:complexType>

```

XML Quellcode 5.12: OriginDestType

Strukturierte Information

InfoType: Typ für struktuierte Information und Text.

	Element	Typ	Kommentar
☐ ☐	<i>Text</i>	xs:string	Informationstext für den Benutzer.
	<i>WithText</i>	xs:boolean	Falls gesetzt enthält die Information Text der dem Benutzer angezeigt werden soll.
☐	<i>Class</i>	AttributeClassType	Informationsklasse.
☐	<i>Mandatory</i>	xs:boolean	Falls gesetzt ist es obligatorisch dem Benutzer die Information anzuzeigen.
☐	<i>Importance</i>	PercentType	Wichtigkeit der Information in Prozent.
☐	<i>URL</i>	xs:anyURI	URL mit weiteren Information. Falls möglich, soll der der Text als Link verwendet werden.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  → name="InfoType">
2   <xs:sequence>
3     <xs:element name="Text" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
4     <xs:element name="WithText" type="xs:boolean"/>
5     <xs:element name="Class" type="AttributeClassType" minOccurs="0"/>
6     <xs:element name="Mandatory" type="xs:boolean" default="false" minOccurs="0"/>
7     <xs:element name="Importance" type="PercentType" minOccurs="0"/>
8     <xs:element name="URL" type="xs:anyURI" minOccurs="0"/>
9   </xs:sequence>
10 </xs:complexType>

```

XML Quellcode 5.13: InfoType

Verknüpfung zum ÖPNV

StopLinkType: Typ für eine Verbindung von einem Verleihstandort zu einer Haltestelle.

Gruppenzugehörigkeit: *DurationGroup*

	Element	Typ	Kommentar
	<i>StopID</i>	xs:token	
☐	<i>Distance</i>	xs:nonNegativeInteger	Distanz des Fußweges für diese Verbindung in Metern. Für den Fall von Gebieten ist dies die wahrscheinliche Distanz des Fußweges um ein Fahrzeug zu erreichen.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  → name="StopLinkType">
2   <xs:sequence>
3     <xs:element name="StopID" type="xs:token"/>
4     <xs:group ref="DurationGroup"/>

```

```

5     <xs:element name="Distance" type="xs:nonNegativeInteger" minOccurs="0"/>
6   </xs:sequence>
7 </xs:complexType>

```

XML Quellcode 5.14: StopLinkType

5.2 Basisgruppen

Standort oder Gebiet

PlaceOrAreaGroup: Gruppe für Informationen, welche für Standorte und Free-Floating-Gebiete verwendet werden.

	Element	Typ	Kommentar
<input checked="" type="checkbox"/>	<i>Name</i>	TextType	Druckbare Namen des Standortes oder Gebiets.
	<i>ProviderID</i>	ProviderIDType	Provider-ID, zu welcher der Standort oder das Gebiet gehört.
<input type="checkbox"/>	<i>Description</i>	TextType	
<input type="checkbox"/>	<i>StopLink</i>	StopLinkType	Fußwegverbindungen zu Stops.
<input type="checkbox"/>	<i>AttributeID</i>	AttributeIDType	Zusätzliche Informationen zu diesem Standort oder Gebiet.

```

1 <xs:group xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema" name="
   → PlaceOrAreaGroup">
2   <xs:sequence>
3     <xs:element name="Name" type="TextType" maxOccurs="unbounded"/>
4     <xs:element name="ProviderID" type="ProviderIDType"/>
5     <xs:element name="Description" type="TextType" minOccurs="0" maxOccurs="unbounded"/>
6     <xs:element name="StopLink" type="StopLinkType" minOccurs="0" maxOccurs="unbounded"/>
7     <xs:element name="AttributeID" type="AttributeIDType" minOccurs="0" maxOccurs="unbounded"/>
8   </xs:sequence>
9 </xs:group>

```

XML Quellcode 5.15: PlaceOrAreaGroup

Dauer

DurationGroup: Gruppe für die wahrscheinliche Dauer und ihrer Varianz um ein Buchungsziel zu erhalten.

Element	Typ	Kommentar
<i>Duration</i>	xs:duration	Wahrscheinliche Dauer um ein Buchungsziel zu erhalten.
□ <i>Variance</i>	xs:duration	Varianz der wahrscheinlichen Dauer um ein Buchungsziel zu erhalten. Die wahrscheinliche Dauer kann um diesen Wert nach oben oder unten variieren.

```

1 <xs:group xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema" name="
  ↳ DurationGroup">
2   <xs:sequence>
3     <xs:element name="Duration" type="xs:duration"/>
4     <xs:element name="Variance" type="xs:duration" minOccurs="0"/>
5   </xs:sequence>
6 </xs:group>

```

XML Quellcode 5.16: DurationGroup

5.3 Dienst 1 - Statische Daten

Buchungsziel ID

BookingTargetIDType: Typ für Buchungsziel-IDs.

Element	Typ	Kommentar
<i>BookeeID</i>	BookeeIDType	
<i>ProviderID</i>	ProviderIDType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="BookingTargetIDType">
2   <xs:sequence>
3     <xs:element name="BookeeID" type="BookeeIDType"/>
4     <xs:element name="ProviderID" type="ProviderIDType"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 5.17: BookingTargetIDType

Verleihstations ID

ProviderPlaceIDType: Typ für Anbieter/Ort-Kombinationen.

Element	Typ	Kommentar
<i>ProviderID</i>	ProviderIDType	
<i>PlaceID</i>	PlaceIDType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ↳ name="ProviderPlaceIDType">
2   <xs:sequence>
3     <xs:element name="ProviderID" type="ProviderIDType"/>
4     <xs:element name="PlaceID" type="PlaceIDType"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 5.18: ProviderPlaceIDType

Fahrzeugattribut

AttributeType: Typ für strukturierte Attributklassifikation mit assoziiertem Text.

	Element	Typ	Kommentar
<input type="checkbox"/> 	<i>Text</i>	TextType	Text des Attributs, welcher dem User angezeigt wird.
	<i>WithText</i>	xs:boolean	Definiert, ob dieses Attribut einen Text hat, welcher für den User bestimmt ist, oder nicht.
	<i>ID</i>	AttributeIDType	Interner Code des Attributs zur Referenzierung
<input type="checkbox"/>	<i>Class</i>	AttributeClassType	Klasse von Attributen, entnommen aus der Liste für erlaubte Werte.
<input type="checkbox"/>	<i>Separate</i>	xs:boolean	Falls auf "wahr" gesetzt, beschreibt dieses Attribut Zubehörequipment, welches separat gebucht werden muss.
<input type="checkbox"/>	<i>Mandatory</i>	xs:boolean	Definiert, ob das Attribut dem Nutzer in irgendeiner Form angezeigt wird.
<input type="checkbox"/>	<i>Importance</i>	PercentType	Wichtigkeit des Attributs in Prozent.
<input type="checkbox"/>	<i>URL</i>	xs:anyURI	URL für zusätzliche Informationen zu diesem generellen Attribut. Falls verfügbar, muss der gesamte Attributtext als markierter Link verwendet werden.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↪ name="AttributeType">
2   <xs:sequence>
3     <xs:element name="Text" type="TextType" minOccurs="0" maxOccurs="unbounded"/>
4     <xs:element name="WithText" type="xs:boolean"/>
5     <xs:element name="ID" type="AttributeIDType"/>
6     <xs:element name="Class" type="AttributeClassType" minOccurs="0"/>
7     <xs:element name="Separate" type="xs:boolean" default="false" minOccurs="0"/>
8     <xs:element name="Mandatory" type="xs:boolean" default="false" minOccurs="0"/>
9     <xs:element name="Importance" type="PercentType" minOccurs="0"/>
10    <xs:element name="URL" type="xs:anyURI" minOccurs="0"/>
11  </xs:sequence>
12 </xs:complexType>

```

XML Quellcode 5.19: AttributeType

Verleihstation

PlaceType: Typ für Standortinformationen.

Gruppenzugehoerigkeit: *PlaceOrAreaGroup*

Element	Typ	Kommentar
<i>ID</i>	PlaceIDType	Standort-ID. Für jeden physischen Standard muss diese ID einheitlich sein.
□ <i>GlobalID</i>	GlobalIDType	Global eindeutige ID. Wird benutzt um Duplikate eines Objektes das von verschiedenen Systemen verwaltet wird zu erkennen.
<i>Coord</i>	GeoPositionType	Geoposition des Standortes.
□ <i>Capacity</i>	xs:nonNegativeInteger	Kapazität des Standortes zum Zurückweisen von Buchungszielen.
□ <i>OnPremisesTime</i>	xs:duration	Zusätzliche Zeit um das Buchungsziel am Abstellort zu finden, z.B. in großen Parkhäusern. Standardmäßig 0.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="PlaceType">
2   <xs:sequence>
3     <xs:element name="ID" type="PlaceIDType"/>
4     <xs:element name="GlobalID" type="GlobalIDType" minOccurs="0"/>
5     <xs:element name="Coord" type="GeoPositionType"/>
6     <xs:element name="Capacity" type="xs:nonNegativeInteger" minOccurs="0"/>
7     <xs:group ref="PlaceOrAreaGroup"/>
8     <xs:element name="OnPremisesTime" type="xs:duration" minOccurs="0"/>
9   </xs:sequence>
10 </xs:complexType>

```

XML Quellcode 5.20: PlaceType

Gebiet mit Dichteangabe

DensityAreaType: Typ für geographische Gebiete mit wahrscheinlicher Zeit um ein Fahrzeug zu erhalten.

Gruppenzugehörigkeit: *DurationGroup*

Element	Typ	Kommentar
<i>Area</i>	GeoAreaType	Definition des geographischen Gebiets.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="DensityAreaType">
2   <xs:sequence>
3     <xs:element name="Area" type="GeoAreaType"/>
4     <xs:group ref="DurationGroup"/>

```

5 </xs:sequence>
6 </xs:complexType>

XML Quellcode 5.21: DensityAreaType

Freefloating Gebiet

FloatingAreaType: Typ für Free-Floating-Gebiete.

Gruppenzugehoerigkeit: *PlaceOrAreaGroup* *DurationGroup*

Element	Typ	Kommentar
<i>ID</i>	AreaIDType	Gebiet-ID.
☐ <i>Area</i>	IncExcGeoAreaType	Geographische Gebiete, abgedeckt durch Free-Floater. Ein Punkt gehört zum Free-Floating-Gebiet, falls dieser zumindest einer der gegebenen Inklusionsareale und nicht einer der Exklusionsareale ist.
☐ ☐ <i>SubArea</i>	DensityAreaType	Sub-Gebiete des Free-Floating-Gebiets. Für jedes Sub-Gebiet ist eine geschätzte Zeit (und dessen Varianz) gegeben, um ein Free-Floating-Buchungsziel zu erhalten. Für Stops, die mehr als ein Sub-Gebiet abdecken, wird die Mindestdauer verwendet.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="FloatingAreaType">
2   <xs:sequence>
3     <xs:element name="ID" type="AreaIDType"/>
4     <xs:element name="Area" type="IncExcGeoAreaType" maxOccurs="unbounded"/>
5     <xs:group ref="PlaceOrAreaGroup"/>
6     <xs:group ref="DurationGroup"/>
7     <xs:element name="SubArea" type="DensityAreaType" minOccurs="0" maxOccurs="unbounded"/>
8   </xs:sequence>
9 </xs:complexType>

```

XML Quellcode 5.22: FloatingAreaType

Gruppe von Verleihstationen

PlaceGroupType: Typ für Standort-Gruppen.

Element	Typ	Kommentar
<i>ID</i>	PlaceGroupIDType	Standort-Gruppen-ID
<input checked="" type="checkbox"/> <i>PlaceID</i>	ProbabilityPlaceIDType	
<i>Probability</i>	PercentType	Durchschnittliche Wahrscheinlichkeit, ein Buchungsziel auf einem spezifischen Standort, innerhalb dieser Standort-Gruppe, zu erhalten. Gilt nur für solche Standorte, für die ein spezieller Wert hinterlegt ist.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="PlaceGroupType">
2   <xs:sequence>
3     <xs:element name="ID" type="PlaceGroupIDType"/>
4     <xs:element name="PlaceID" type="ProbabilityPlaceIDType" maxOccurs="unbounded"/>
5     <xs:element name="Probability" type="PercentType"/>
6   </xs:sequence>
7 </xs:complexType>

```

XML Quellcode 5.23: PlaceGroupType

Provider

ProviderType: Type for Anbieter Informationen.

Element	Typ	Kommentar
<i>ID</i>	ProviderIDType	Anbieter ID.
<i>Name</i>	xs:normalizedString	Druckbarer, vollständiger Name des Anbieters.
<i>CustomerChoice</i>	xs:boolean	Falls gesetzt, wird der Anbieter zum Heimat/Standardanbieter des Kunden.
<input type="checkbox"/> <i>ShortName</i>	xs:normalizedString	Kurzname des Anbieters.
<input checked="" type="checkbox"/> <i>AttributeID</i>	AttributeIDType	Weiter Informationen zu dem Anbieter, z.B. URLs.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ProviderType">
2   <xs:sequence>
3     <xs:element name="ID" type="ProviderIDType"/>
4     <xs:element name="Name" type="xs:normalizedString"/>
5     <xs:element name="CustomerChoice" type="xs:boolean"/>
6     <xs:element name="ShortName" type="xs:normalizedString" minOccurs="0"/>

```

5 Datenmodell

```
7   <xs:element name="AttributeID" type="AttributeIDType" minOccurs="0" maxOccurs="unbounded"/>
8 </xs:sequence>
9 </xs:complexType>
```

XML Quellcode 5.24: ProviderType

Buchungsziel

BookingTargetType: Typ für Buchungszielinformationen.

Element	Typ	Kommentar
<i>ID</i>	BookingTargetIDType	BookingTargetID.
<input type="checkbox"/> <i>GlobalID</i>	GlobalIDType	Global eindeutige ID. Wird benutzt um Duplikate eines Objektes das von verschiedenen Systemen verwaltet wird zu erkennen.
<input checked="" type="checkbox"/> <i>Name</i>	TextType	Druckbarer Name des Bookee.
<input type="checkbox"/> <i>PlaceID</i>	PlaceIDType	Standard-ID, auf dem der Bookee platziert ist. Muss mit stationsbasierten Buchungszielen befüllt werden.
<input type="checkbox"/> <i>PlaceGroupID</i>	PlaceGroupIDType	Standort-Gruppen-ID dieses Bookees, falls er nicht zu einem festen Standort gehört.
<input type="checkbox"/> <i>AreaID</i>	AreaIDType	Gebiet-ID dieses Bookees. Muss mit Free-Floating-Buchungszielen befüllt werden.
<i>Class</i>	ClassType	Klasse des Bookees.
<input type="checkbox"/> <i>BookingHorizon</i>	xs:duration	Maximale Dauer in der Zukunft, wenn dieses Ziel gebucht werden kann. Ist relativ zur aktuellen Zeit.
<input type="checkbox"/> <i>BookingGrid</i>	xs:nonNegativeInteger	Netz für Start- und Endzeit der Buchung in Minuten. Sollte ein Divisor von 60 sein. Beispiele: 1, 5, 10, 15, 30, 60.
<input type="checkbox"/> <i>OpeningTime</i>	xs:duration	Benötigte Zeit um das Fahrzeug aufzuschließen..
<input type="checkbox"/> <i>Engine</i>	EngineType	
<input type="checkbox"/> <i>CO2Factor</i>	xs:nonNegativeInteger	CO2-Emissionen in Gramm pro Kilometer.
<input type="checkbox"/> <i>MaxDistance</i>	xs:nonNegativeInteger	Maximale Distanz in Metern, die ein Bookee ohne längere Stops zurücklegen kann. Falls nicht gegeben, ist kein Limit angenommen.
<input type="checkbox"/> <input checked="" type="checkbox"/> <i>AttributeID</i>	AttributeIDType	Zusätzliche Informationen zu diesem Bookee, z.B. Klimaanlage, Navigationssystem, etc..

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↪ name="BookingTargetType">
2 <xs:sequence>

```

```

3 <xs:element name="ID" type="BookingTargetIDType"/>
4 <xs:element name="GlobalID" type="GlobalIDType" minOccurs="0"/>
5 <xs:element name="Name" type="TextType" maxOccurs="unbounded"/>
6 <xs:choice>
7   <xs:element name="PlaceID" type="PlaceIDType"/>
8   <xs:element name="PlaceGroupID" type="PlaceGroupIDType"/>
9   <xs:element name="AreaID" type="AreaIDType"/>
10 </xs:choice>
11 <xs:element name="Class" type="ClassType"/>
12 <xs:element name="BookingHorizon" type="xs:duration" minOccurs="0"/>
13 <xs:element name="BookingGrid" type="xs:nonNegativeInteger" minOccurs="0"/>
14 <xs:element name="OpeningTime" type="xs:duration" minOccurs="0"/>
15 <xs:element name="Engine" type="EngineType" minOccurs="0"/>
16 <xs:element name="CO2Factor" type="xs:nonNegativeInteger" minOccurs="0"/>
17 <xs:element name="MaxDistance" type="xs:nonNegativeInteger" minOccurs="0"/>
18 <xs:element name="AttributeID" type="AttributeIDType" minOccurs="0" maxOccurs="unbounded"/>
19 </xs:sequence>
20 </xs:complexType>

```

XML Quellcode 5.25: BookingTargetType

Buchung

BookingType: Typ für eine Buchung.

Element	Typ	Kommentar
<i>ID</i>	BookingIDType	
<input type="checkbox"/> <i>TimePeriod</i>	TimePeriodType	Tatsächlicher Zeitraum für die Buchung, wurde ggfs. an das Buchungsraster angepasst.
<input type="checkbox"/> <input type="checkbox"/> <i>Info</i>	InfoType	Weitere Informationen zur Buchung.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   → name="BookingType">
2 <xs:sequence>
3   <xs:element name="ID" type="BookingIDType"/>
4   <xs:element name="TimePeriod" type="TimePeriodType" minOccurs="0"/>
5   <xs:element name="Info" type="InfoType" minOccurs="0" maxOccurs="unbounded"/>
6 </xs:sequence>
7 </xs:complexType>

```

XML Quellcode 5.26: BookingType

5.4 Dienst 2 - Verfügbarkeitsauskunft

Buchungseigenschaften

BookingTargetPropertiesType: Typ für benötigte Eigenschaften eines Buchungsziels für synchrone Verfügbarkeit.

Element	Typ	Kommentar
<i>ID</i>	BookingTargetIDType	Buchungsziel-ID.
□ <i>Distance</i>	xs:nonNegativeInteger	Distanz zu Fahren in Metern. Das Car-Sharing-System sollte nur die verfügbaren Buchungsziele zurückgeben, welche die Distanz zurücklegen können (gemäß Ladestatus).
□ <i>SeverityFactor</i>	xs:nonNegativeInteger	Severity-Faktor der vorgeschlagenen Route in Prozent. 100 repräsentieren eine nahezu vollkommen flache Route. Je höher der Wert, desto mehr Energie wird für das Fahren der Route verbraucht. Beispielsweise ein Wert von 200 indiziert, dass doppelt so viel Energie verbraucht wird, wie es bei einer vollkommen flachen Route der Fall wäre.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="BookingTargetPropertiesType">
2   <xs:sequence>
3     <xs:element name="ID" type="BookingTargetIDType"/>
4     <xs:element name="Distance" type="xs:nonNegativeInteger" minOccurs="0"/>
5     <xs:element name="SeverityFactor" type="xs:nonNegativeInteger" minOccurs="0"/>
6   </xs:sequence>
7 </xs:complexType>

```

XML Quellcode 5.27: BookingTargetPropertiesType

Buchungszielverfügbarkeit

BookingTargetAvailabilityType: Typ für Verfügbarkeit von Buchungszielen.

Element	Typ	Kommentar
<i>ID</i>	BookingTargetIDType	
□ <i>PlaceID</i>	PlaceIDType	Für aktuell verfügbare Buchungsziele, die nicht an einen spezifischen Standort gebunden sind, sollte der aktuelle Standort zurückgegeben werden.
□ <i>GeoPosition</i>	GeoPositionType	Für aktuell verfügbare Buchungsziele, welche nicht an einen spezifischen Standort gebunden sind, sollte die aktuelle Position zurückgegeben werden.
□ ⊠ <i>Inavailability</i>	TimePeriodType	Perioden der Nicht-Verfügbarkeit von Buchungszielen.
□ <i>CurrentStateOfCharge</i>	PercentType	Aktueller Ladestatus der Buchungsziels in Prozent.
□ <i>CurrentDrivingRange</i>	xs:nonNegativeInteger	Aktuelle Reichweite des Buchungsziels in Metern, gegeben einer nahezu flachen Route. Falls nicht gegeben, wird für das Fahrzeug einen unbegrenzte Reichweite angenommen.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ↳ name="BookingTargetAvailabilityType">
2   <xs:sequence>
3     <xs:element name="ID" type="BookingTargetIDType"/>
4     <xs:element name="PlaceID" type="PlaceIDType" minOccurs="0"/>
5     <xs:element name="GeoPosition" type="GeoPositionType" minOccurs="0"/>
6     <xs:element name="Inavailability" type="TimePeriodType" minOccurs="0" maxOccurs="unbounded"/>
7     <xs:element name="CurrentStateOfCharge" type="PercentType" minOccurs="0"/>
8     <xs:element name="CurrentDrivingRange" type="xs:nonNegativeInteger" minOccurs="0"/>
9   </xs:sequence>
10 </xs:complexType>

```

XML Quellcode 5.28: BookingTargetAvailabilityType

5.5 Dienst 4 - Buchung

Fahrzeugverfügbarkeit

BookingTargetChangeAvailabilityType: Typ für Änderungen der Verfügbarkeit von Buchungszielen.

Element	Typ	Kommentar
<i>ID</i>	BookingTargetIDType	Buchungsziel-ID.
<input type="checkbox"/> <i>PlaceID</i>	PlaceIDType	Für aktuell verfügbare Buchungsziele, welche nicht an einen spezifischen Standort gebunden sind, sollte der aktuelle Standort zurückgegeben werden.
<input type="checkbox"/> <i>GeoPosition</i>	GeoPositionType	Für neu verfügbare Fahrzeuge, welche nicht an einen spezifischen Standort gebunden sind, sollte die Position zurückgegeben werden.
<input type="radio"/> <i>Inavailability</i>	TimePeriodType	Neue Nicht-Verfügbarkeitsperiode.
<input type="radio"/> <i>Availability</i>	TimePeriodType	Neue Verfügbarkeitsperiode.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="BookingTargetChangeAvailabilityType">
2   <xs:sequence>
3     <xs:element name="ID" type="BookingTargetIDType"/>
4     <xs:element name="PlaceID" type="PlaceIDType" minOccurs="0"/>
5     <xs:element name="GeoPosition" type="GeoPositionType" minOccurs="0"/>
6     <xs:choice>
7       <xs:element name="Inavailability" type="TimePeriodType"/>
8       <xs:element name="Availability" type="TimePeriodType"/>
9     </xs:choice>
10  </xs:sequence>
11 </xs:complexType>

```

XML Quellcode 5.29: BookingTargetChangeAvailabilityType

Standortkapazität

PlaceAvailabilityType: Typ für die verfügbare Kapazität eines Standortes.

Element	Typ	Kommentar
<i>ID</i>	ProviderPlaceIDType	
<i>Availability</i>	xs:nonNegativeInteger	Aktuell verfügbare Kapazität um Buchungsziele zurückzugeben.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="PlaceAvailabilityType">
2   <xs:sequence>
3     <xs:element name="ID" type="ProviderPlaceIDType"/>
4     <xs:element name="Availability" type="xs:nonNegativeInteger"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 5.30: PlaceAvailabilityType

Dienst 4a - externe Buchungen

ExternalBookingType: Type für Informationen über externe Buchungen.

Element	Typ	Kommentar
<i>BookingID</i>	BookingIDType	BookingID der externen hinzugefügten Buchungen.
<i>BookingTargetID</i>	BookingTargetIDType	BookingTargetID des gebuchten Fahrzeuges.
<i>UserInfo</i>	UserInfoType	UserInfo des Benutzers dem diese Buchung gehört.
<i>TimePeriod</i>	TimePeriodType	Zeitraum der Buchung.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ExternalBookingType">
2   <xs:sequence>
3     <xs:element name="BookingID" type="BookingIDType"/>
4     <xs:element name="BookingTargetID" type="BookingTargetIDType"/>
5     <xs:element name="UserInfo" type="UserInfoType"/>
6     <xs:element name="TimePeriod" type="TimePeriodType"/>
7   </xs:sequence>
8 </xs:complexType>

```

XML Quellcode 5.31: ExternalBookingType

5.6 Dienst 5 - Buchungsabonnement

Buchungsverfügbarkeit

BookingChangeType: Typ für Änderungen einer Buchung.

Element	Typ	Kommentar
<i>BookingID</i>	BookingIDType	
○ <i>Cancelled</i>	xs:boolean	Buchung wurde vom User abgebrochen. Eventuell über verschiedene Kanäle (z.B. native App, Telefon, etc.). Anschließend ist die Buchungs-ID nicht mehr valide.
○ <i>NewPeriod</i>	TimePeriodType	Der Buchung wurde eine neue Periode vom User zugewiesen. Eventuell über verschiedene Kanäle (z.B. native App, Telefon, etc.).
○ <i>Notification</i>	xs:boolean	Die Buchung ist noch valide, allerdings gibt es Informationen für den User, betreffend der Buchung.
○ <i>Impossible</i>	xs:boolean	Es war nicht möglich die Buchung auszuführen von Seiten des Car-Sharing-Systems (z.B. durch Fahrzeugversagen). Trotzdem ist die Buchung weiter valide, solange die Ausführung später ermöglicht wird.
○ <i>Repossiblized</i>	xs:boolean	Es war möglich, die Buchung erneut auszuführen (z.B. durch das Ersetzen eines Fahrzeugs).
□ □ <i>Reason</i>	TextType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  → name="BookingChangeType">
2   <xs:sequence>
3     <xs:element name="BookingID" type="BookingIDType"/>
4     <xs:choice>
5       <xs:element name="Cancelled" type="xs:boolean"/>
6       <xs:element name="NewPeriod" type="TimePeriodType"/>
7       <xs:element name="Notification" type="xs:boolean"/>
8       <xs:element name="Impossible" type="xs:boolean"/>
9       <xs:element name="Repossiblized" type="xs:boolean"/>
10    </xs:choice>
11    <xs:element name="Reason" type="TextType" minOccurs="0" maxOccurs="unbounded"/>
12  </xs:sequence>
13 </xs:complexType>

```

XML Quellcode 5.32: BookingChangeType

5.7 Dienst 6 - Preisauskunft

Preisinformation (Details)

TariffDetailType: Typ für eine Tarifkomponente einschließlich des Preises.

Element	Typ	Kommentar
<i>Name</i>	xs:string	Druckbarer Name des Tarifdetails.
<i>Price</i>	EuroPriceType	Preis in Euro-Cent.
<input type="checkbox"/> <i>Description</i>	xs:string	Druckbare Beschreibung des Tarifdetails.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="TariffDetailType">
2   <xs:sequence>
3     <xs:element name="Name" type="xs:string"/>
4     <xs:element name="Price" type="EuroPriceType"/>
5     <xs:element name="Description" type="xs:string" minOccurs="0"/>
6   </xs:sequence>
7 </xs:complexType>

```

XML Quellcode 5.33: TariffDetailType

Preisinformation

TariffType: Typ für einen Tarif einschließlich des Preises.

Element	Typ	Kommentar
<i>Name</i>	xs:string	Druckbarer Name des Tarifs.
<i>Price</i>	EuroPriceType	Preis in Euro-Cent.
<input type="checkbox"/> <i>Description</i>	xs:string	Druckbare Beschreibung des Tarifs.
<input type="checkbox"/> <input type="checkbox"/> <i>Detail</i>	TariffDetailType	Details des Endpreises. Falls bereitgestellt, sollten diese einen Satz Preiskomponenten beinhalten, welche den Gesamtpreis bilden.
<input type="checkbox"/> <input type="checkbox"/> <i>Info</i>	InfoType	Zusatzinformationen für den Tarif.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="TariffType">
2   <xs:sequence>
3     <xs:element name="Name" type="xs:string"/>
4     <xs:element name="Price" type="EuroPriceType"/>
5     <xs:element name="Description" type="xs:string" minOccurs="0"/>
6     <xs:element name="Detail" type="TariffDetailType" minOccurs="0" maxOccurs="unbounded"/>
7     <xs:element name="Info" type="InfoType" minOccurs="0" maxOccurs="unbounded"/>
8   </xs:sequence>
9 </xs:complexType>

```

XML Quellcode 5.34: TariffType

5.8 Dienst 7 - Verbrauchsdatenabonnement

Verbrauchsdaten

ConsumptionType: Typ für einen Rechnungsposten der Verbrauchsdaten.

Element	Typ	Kommentar
<i>BookingID</i>	BookingIDType	BookingID Referenz für diesen Verbrauchsdatenposten.
<i>Type</i>	ConsumptionClassType	Typ des Verbrauchs.
<input checked="" type="checkbox"/> <i>Description</i>	TextType	Druckbare Beschreibung des Rechnungsposten.
<input type="checkbox"/> <i>Value</i>	xs:decimal	Dezimaler Verbrauchswert (z.B., Distanz).
<input type="checkbox"/> <i>TimePeriod</i>	TimePeriodType	Verbrauchswert als Zeitperiode.
<input type="checkbox"/> <i>Unit</i>	xs:string	Druckbarer Name der Einheit.
<input type="checkbox"/> <i>Price</i>	EuroPriceType	Preis in Euro-Cent.
<input type="checkbox"/> <input checked="" type="checkbox"/> <i>Annotation</i>	TextType	Druckbare Erläuterung zu dem Rechnungsposten.
<input type="checkbox"/> <i>Final</i>	xs:boolean	Verbrauchsdaten für diese Buchung sind final, weitere Verbrauchsdaten werden nicht erwartet.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ConsumptionType">
2   <xs:sequence>
3     <xs:element name="BookingID" type="BookingIDType"/>
4     <xs:element name="Type" type="ConsumptionClassType"/>
5     <xs:element name="Description" type="TextType" maxOccurs="unbounded"/>
6     <xs:choice>
7       <xs:element name="Value" type="xs:decimal"/>

```

```
8     <xs:element name="TimePeriod" type="TimePeriodType"/>
9   </xs:choice>
10  <xs:element name="Unit" type="xs:string" minOccurs="0"/>
11  <xs:element name="Price" type="EuroPriceType" minOccurs="0"/>
12  <xs:element name="Annotation" type="TextType" minOccurs="0" maxOccurs="unbounded"/>
13  <xs:element name="Final" type="xs:boolean" default="false" minOccurs="0"/>
14 </xs:sequence>
15 </xs:complexType>
```

XML Quellcode 5.35: ConsumptionType

5.9 Dienst 8 - Fahrzeugfreischaltung

Für Dienst 8 sind keine weiteren Datentypen erforderlich.

5.10 Dienst 9 - Usermanagement

Benutzereigenschaften

UserFeatureType: Typ für Eigenschaft von Benutzern.

Element	Typ	Kommentar
<i>Value</i>	xs:string	Wert der Klasse.
<i>Class</i>	UserFeatureClassType	Eigenschaftensklasse.

```
1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ↳ name="UserFeatureType">
2   <xs:sequence>
3     <xs:element name="Value" type="xs:string"/>
4     <xs:element name="Class" type="UserFeatureClassType"/>
5   </xs:sequence>
6 </xs:complexType>
```

XML Quellcode 5.36: UserFeatureType

Benutzertyp

UserType: Typ für einen Benutzer (Für Benutzerverwaltung)).

Element	Typ	Kommentar
<i>ID</i>	UserInfoType	UserID und ProviderID zur (eindeutigen) Identifikation.
<i>State</i>	UserStateType	State of the (e.g., default, banned etc).
<input type="checkbox"/> <i>Info</i>	UserFeatureType	Zusätzliche Benutzerinformationen.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  → name="UserType">
2   <xs:sequence>
3     <xs:element name="ID" type="UserInfoType"/>
4     <xs:element name="State" type="UserStateType"/>
5     <xs:element name="Info" type="UserFeatureType" minOccurs="0" maxOccurs="unbounded"/>
6   </xs:sequence>
7 </xs:complexType>

```

XML Quellcode 5.37: UserType

5.11 Dienst 10 - Fahrzeugeinstellungsmanagement

Fahrzeugeinstellungen

BookingSettingsType: Typ für eine einzelne Buchungseinstellung.

Element	Typ	Kommentar
<i>BookingID</i>	BookingIDType	BookingID Referenz für diese Buchungseinstellung.
<input type="checkbox"/> <i>BookingTargetSpecific</i>	xs:boolean	Falls gesetzt, ist die Einstellung nur für das gleiche Buchungsziel anwendbar (fahrzeugspezifisch).
<i>Type</i>	BookingSettingsClassType	Klasse der Buchungseinstellung.
<input type="radio"/> <i>Value</i>	xs:decimal	Dezimale Wert (z.B. Temperatur in C).
<input type="radio"/> <i>Value</i>	xs:string	Wert als String.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="BookingSettingsType">
2   <xs:sequence>
3     <xs:element name="BookingID" type="BookingIDType"/>
4     <xs:element name="BookingTargetSpecific" type="xs:boolean" default="false" minOccurs="0"/>
5     <xs:element name="Type" type="BookingSettingsClassType"/>
6     <xs:choice>
7       <xs:element name="Value" type="xs:decimal"/>
8       <xs:element name="Value" type="xs:string"/>
9     </xs:choice>
10  </xs:sequence>
11 </xs:complexType>

```

XML Quellcode 5.38: BookingSettingsType

5.12 Dienst 11 - Fernkonfiguration Navigationssystem und Fahrtverlauf überwachen

Buchungsfortschritt

BookingProgressType: Typ für einzelnen Buchungsfortschritt (Reisefortschritt).

Element	Typ	Kommentar
<i>BookingID</i>	BookingIDType	Buchungsreferenz.
<i>Timestamp</i>	xs:dateTime	Zeitstempel der Geopositionierung / ETA Berechnung.
<i>ETA</i>	xs:dateTime	Geschätzte Ankunftszeit des Buchungsziel am Zielort.
<input type="checkbox"/> <i>Position</i>	CoordType	Aktuelle Position des Buchungsziel. Optional wegen möglicher Datenschutzbedenken.
<input type="checkbox"/> <i>Progress</i>	PercentType	Prozentuale Schätzung des Reisefortschritt zur Visualisierung.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="BookingProgressType">
2   <xs:sequence>
3     <xs:element name="BookingID" type="BookingIDType"/>
4     <xs:element name="Timestamp" type="xs:dateTime"/>
5     <xs:element name="ETA" type="xs:dateTime"/>
6     <xs:element name="Position" type="CoordType" minOccurs="0"/>
7     <xs:element name="Progress" type="PercentType" minOccurs="0"/>
8   </xs:sequence>
9 </xs:complexType>

```

XML Quellcode 5.39: BookingProgressType

5.13 Dienst 12 - Dialog Intermodale Alternative

Routenvorschlag

ItinerarySuggestionType: Typ für einen einzelnen intermodalen Routenvorschlag.

Element	Typ	Kommentar
<i>ItineraryID</i>	ItineraryIDType	Eindeutige ID der intermodalen Route. Wird genutzt um gewählte Routen zu identifizieren.
<i>NextDestination</i>	OriginDestType	Zielort des aktuellen Teils der intermodalen Route. An diesem Ort findet der Wechsel auf ein anderes Verkehrsmittel statt.
<i>ExpectedArrivalTime</i>	xs:dateTime	Die erwartete Ankunftszeit der gesamte intermodalen Route.
<i>Price</i>	EuroPriceType	Der Preis in Euro Cent um alle Teile dieser Route zu buchen.
<i>ShortDescription</i>	TextType	Eine kurze Beschreibung der Route.
<i>LongDescription</i>	TextType	Eine detaillierte Beschreibung der Route.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ItinerarySuggestionType">
2   <xs:sequence>
3     <xs:element name="ItineraryID" type="ItineraryIDType"/>
4     <xs:element name="NextDestination" type="OriginDestType"/>
5     <xs:element name="ExpectedArrivalTime" type="xs:dateTime"/>
6     <xs:element name="Price" type="EuroPriceType"/>
7     <xs:element name="ShortDescription" type="TextType"/>
8     <xs:element name="LongDescription" type="TextType"/>
9   </xs:sequence>
10 </xs:complexType>

```

XML Quellcode 5.40: ItinerarySuggestionType

5.14 Authentifizierung

Benutzeridentifikation

UserInfoType: Typ für eine User-Identifizierung. Passwort/Token sind optional.

Element	Typ	Kommentar
<i>ProviderID</i>	ProviderIDType	ID des Bereitstellers des Users. Ohne diese ist die User-ID nicht einzigartig.
<i>UserID</i>	UserIDType	User-ID.
<input type="radio"/> <i>Password</i>	xs:normalizedString	Passwort des Users.
<input type="radio"/> <i>Token</i>	xs:token	Authentifizierungs-Token für den User, welcher das Passwort aus Sicherheitsgründen ersetzt.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="UserInfoType">
2   <xs:sequence>
3     <xs:element name="ProviderID" type="ProviderIDType"/>
4     <xs:element name="UserID" type="UserIDType"/>
5     <xs:choice minOccurs="0">
6       <xs:element name="Password" type="xs:normalizedString"/>
7       <xs:element name="Token" type="xs:token"/>
8     </xs:choice>
9   </xs:sequence>
10 </xs:complexType>

```

XML Quellcode 5.41: UserInfoType

Authentifizierung

AuthType: Typ für eine Authentifizierung innerhalb einer Anfrage.

Element	Typ	Kommentar
<input type="radio"/> <i>SessionID</i>	SessionIDType	Valide Session-ID. Falls eine Session wiederverwendet wird, diese ausfüllen.
<input type="radio"/> <input type="checkbox"/> <i>UserInfo</i>	UserInfoType	Falls keine Session verfügbar ist, muss der User authentisieren.
<input type="radio"/> <i>Anonymous</i>	xs:boolean	Falls weder eine existierende Session, noch eine Nutzerinformation verfügbar ist, kann eine anonyme Session geöffnet werden.
<input type="checkbox"/> <i>SessionTimeout</i>	xs:duration	Falls gesetzt, wünscht der Client eine Session mit gegebenem Inaktivitätstimeout.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="AuthType">
2   <xs:sequence>
3     <xs:choice>

```

```

4     <xs:element name="SessionID" type="SessionIDType"/>
5     <xs:element name="UserInfo" type="UserInfoType" maxOccurs="unbounded"/>
6     <xs:element name="Anonymous" type="xs:boolean"/>
7   </xs:choice>
8   <xs:element name="SessionTimeout" type="xs:duration" minOccurs="0"/>
9 </xs:sequence>
10 </xs:complexType>

```

XML Quellcode 5.42: AuthType

5.15 Fehlerbehandlung

Fehler

ErrorType: Typ für Fehler.

Element	Typ	Kommentar
<i>Code</i>	ErrorCodeType	Fehlercode, entnommen aus der Liste für erlaubte Fehlercodes.
<input type="checkbox"/> <i>NonFatal</i>	xs:boolean	Falls gesetzt, war die korrespondierende Operation generell erfolgreich und Operationen können fortlaufen.
<input type="checkbox"/> <i>SystemMessage</i>	xs:string	Text für Systemlogging, welcher den Fehler detaillierter erläutert. Sollte in englischer Sprache sein.
<input type="checkbox"/> <i>UserMessage</i>	xs:string	Text für Userinformation in angeforderter Sprache.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ↪ name="ErrorType">
2   <xs:sequence>
3     <xs:element name="Code" type="ErrorCodeType"/>
4     <xs:element name="NonFatal" type="xs:boolean" default="false" minOccurs="0"/>
5     <xs:element name="SystemMessage" type="xs:string" minOccurs="0"/>
6     <xs:element name="UserMessage" type="xs:string" minOccurs="0"/>
7   </xs:sequence>
8 </xs:complexType>

```

XML Quellcode 5.43: ErrorType

6 Technische Realisierung

6.1 Nachrichtenkodierung

Die Nachrichten zwischen den beiden Systemen werden als XML übertragen. Eine präzise Typdefinition wird durch das zur Schnittstelle gehörige XML Schema vorgegeben.

Falls sich der Overhead, der durch die Einführung von XML, als problematisch herausstellen sollte, besteht die Möglichkeit das Efficient XML Interchange (EXI) Protokoll einzusetzen. Der Einsatz von EXI würde die Größe der Nachrichten erheblich verringern, ohne die Vorteile der Verwendung von XML zu verlieren.

6.2 Kommunikationskanal

Da die Schnittstelle neben dem Anfrage- und Antwort-Schema auch ein asynchrones Abonnementmodell vorsieht und eine geringe Antwortzeit wünschenswert ist, wird für den Kommunikationskanal das WebSocket-Protokoll empfohlen. WebSockets erlauben es, eine bestehende Verbindung der beiden Systeme herzustellen und über diese bidirektional Nachrichten auszutauschen. Das FVS stellt den Server und das RIS den Client (im HTTP Kontext) dar. Es können im Prinzip beliebig viele Kommunikationskanäle geöffnet werden. Aktualisierungen von abonnierten Objekten werden über die gleiche Verbindung geliefert, über die sie abonniert wurden. Bei Unterbrechung der Verbindung endet das Abonnement.

6.3 Authentifizierung

IXSI ist als B2B-Schnittstelle konzipiert und enthält deswegen keinen eigenen Authentifizierungsmechanismus. Falls erforderlich können die Kommunikationspartner vorhandene Mechanismen wie eine SSL-Zertifikatsauthentifizierung (empfohlen), VPN oder HTTP Authentifizierung verwenden.

Endkunde

Da per IXSI auch nutzergesteuerte Requests von System zu System geschickt werden, ist es nötig, dass sich ein Nutzer gegenüber dem FVS authentifizieren kann. Dies geschieht normalerweise über das Tripel Anbieter Referenz/Nutzer Referenz/Passwort. Um das Passwort nicht im Klartext speichern zu müssen, kann alternativ zum Passwort ein Token verwendet werden, welches über die Funktion TokenGeneration (vgl. Abschnitt 7.5) vom FVS generiert werden kann. So muss der Nutzer zwar initial sein Passwort einmalig eingeben, anschließend kann er sich aber über das daraus generierte, gespeicherte Token authentifizieren. Dieses kann z. B. auf dem Endgerät des Nutzers gespeichert werden.

Bei aufeinanderfolgenden Anfragen, die vom selben Nutzer ausgelöst werden, soll nicht in jedem Request eine Authentifizierung durchgeführt werden. Daher wird mit dem ersten Request eines Nutzers eine Session eröffnet (explizit durch OpenSession oder implizit). Für Folge-requests desselben Nutzers kann dann anstatt einer Authentifizierung die ID der eröffneten Session mitgeschickt werden. Diese Sessions besitzen nur eine beschränkte zeitliche Gültigkeit. Nach Ablauf dieser Gültigkeit muss der Nutzer neu identifiziert und eine neue Session eröffnet werden. Eine Session kann auch explizit durch CloseSession geschlossen werden.

6.4 Verbindungssicherheit

Um die Sicherheit der übermittelten Daten zu gewährleisten ist eine Verschlüsselung der Verbindung notwendig. Hierfür eignet sich SSL/TLS-Protokoll. Dieses sollte verwendet werden, wenn die Verbindung nicht bereits durch andere entsprechende Maßnahmen (z. B. durch die Verwendung von Virtuellen Privaten Netzwerken (VPN)) gesichert ist.

7 Nachrichten

Die zwischen den Interaktionspartnern ausgetauschten Nachrichten basieren auf fünf Nachrichtentypen der Art Request / Response und Push. Alle Nachrichten vom RIS an das FVS stellen Anfragen dar und sind daher vom Typ Request. Der Typ Response wird für direkte Antworten des FVS auf Anfragen des RIS verwendet. Um eine Zuordnung der Response-Nachrichten zu den zugehörigen Request-Nachrichten zu erlauben, wird jede Request-Nachricht mit einer eindeutigen Transaction-ID markiert, die in der zugehörigen Response-Nachricht wieder mitgegeben werden muss. Eine Response-Nachricht kann anstelle ihres normalen Inhalts auch einen Fehler enthalten. Der letzte Nachrichtentyp Push findet in den Fällen Verwendung, in denen das RIS fortlaufende Aktualisierungen (Abonnement) vom FVS angefordert hat.

7.1 Basisnachrichten

Basisklasse

IxsiMessageType: Typ für all IXSI-Nachrichten.

Element	Typ	Kommentar
○ □ <i>Request</i>	QueryRequestType	Unabhängige Resultate im Anfrageschema.
○ □ <i>Response</i>	QueryResponseType	Unabhängige Antworten in Anfrageschema.
○ <i>SubscriptionRequest</i>	SubscriptionRequestType	Anfrage in Abonnementschema. Jede Anfrage führt zu mindestens einer Anfrage.
○ <i>SubscriptionResponse</i>	SubscriptionResponseType	Antwort im Abonnementschema. Jede Antwort korrespondiert exakt mit einer Anfrage.
○ <i>SubscriptionMessage</i>	SubscriptionMessageType	Push-Nachricht in Abonnementschema.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="IcsiMessageType">
2   <xs:choice>
3     <xs:element name="Request" type="QueryRequestType" maxOccurs="unbounded"/>
4     <xs:element name="Response" type="QueryResponseType" maxOccurs="unbounded"/>
5     <xs:element name="SubscriptionRequest" type="SubscriptionRequestType"/>
6     <xs:element name="SubscriptionResponse" type="SubscriptionResponseType"/>
7     <xs:element name="SubscriptionMessage" type="SubscriptionMessageType"/>
8   </xs:choice>
9 </xs:complexType>

```

XML Quellcode 7.1: IcsiMessageType

Basisantwort

AbstractBaseResponseType: Abstrakter Typ als eine Basis für Antworten.

Element	Typ	Kommentar
<input type="checkbox"/> <input type="checkbox"/> <i>Error</i>	ErrorType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="AbstractBaseResponseType" abstract="true">
2   <xs:sequence>
3     <xs:element name="Error" type="ErrorType" minOccurs="0" maxOccurs="unbounded"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.2: AbstractBaseResponseType

Eine abstrakte Basisklasse für einen Request ist nicht vorhanden, da nicht erforderlich.

Transaktionsdaten

TransactionType: Typ für eine Transaktions-ID innerhalb von Anfragen und Antworten.

Element	Typ	Kommentar
<i>TimeStamp</i>	xs:dateTime	
<i>MessageID</i>	xs:nonNegativeInteger	Eindeutige ID.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  → name="TransactionType">
2   <xs:sequence>
3     <xs:element name="TimeStamp" type="xs:dateTime"/>
4     <xs:element name="MessageID" type="xs:nonNegativeInteger"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 7.3: TransactionType

Query Request

QueryRequestType: Anfragenachrichten in Anfrageschema.

Gruppenzugehoerigkeit: *StaticDataRequestGroup* *UserTriggeredRequestGroup*

Element	Typ	Kommentar
<i>Transaction</i>	TransactionType	Transaktions-Identifizier, gesetzt vom Client.
<i>SystemID</i>	SystemIDType	ID des Clientsystems.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  → name="QueryRequestType">
2   <xs:sequence>
3     <xs:element name="Transaction" type="TransactionType"/>
4     <xs:element name="SystemID" type="SystemIDType"/>
5     <xs:choice>
6       <xs:group ref="StaticDataRequestGroup"/>
7       <xs:group ref="UserTriggeredRequestGroup"/>
8     </xs:choice>
9   </xs:sequence>
10 </xs:complexType>

```

XML Quellcode 7.4: QueryRequestType

Query Response

QueryResponseType: Antwortnachrichten in Anfrageschema.

Gruppenzugehoerigkeit: *StaticDataResponseGroup* *UserTriggeredResponseGroup*

Element	Typ	Kommentar
<i>Transaction</i>	TransactionType	Transaktions-Identifizier, kopiert von der korrespondierenden Anfrage.
<i>CalcTime</i>	xs:duration	Benötigte Zeit um die beinhaltete Resultate zu berechnen.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="QueryResponseType">
2   <xs:sequence>
3     <xs:element name="Transaction" type="TransactionType"/>
4     <xs:element name="CalcTime" type="xs:duration"/>
5     <xs:choice>
6       <xs:group ref="StaticDataResponseGroup"/>
7       <xs:group ref="UserTriggeredResponseGroup"/>
8     </xs:choice>
9   </xs:sequence>
10 </xs:complexType>

```

XML Quellcode 7.5: QueryResponseType

Subscription Request

SubscriptionRequestType: Anfragenachricht in Abonnementschema.

Gruppenzugehoerigkeit: *SubscriptionAdministrationRequestGroup* *SubscriptionRequestGroup* *RequestMessageGroup*

Element	Typ	Kommentar
<i>Transaction</i>	TransactionType	Transaktions-Identifizier, gesetzt vom Client.
<i>SystemID</i>	SystemIDType	ID des Clientsystems.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="SubscriptionRequestType">
2   <xs:sequence>
3     <xs:element name="Transaction" type="TransactionType"/>
4     <xs:element name="SystemID" type="SystemIDType"/>
5     <xs:choice>
6       <xs:group ref="SubscriptionAdministrationRequestGroup"/>
7       <xs:group ref="SubscriptionRequestGroup"/>
8       <xs:group ref="RequestMessageGroup"/>
9     </xs:choice>
10  </xs:sequence>
11 </xs:complexType>

```

XML Quellcode 7.6: SubscriptionRequestType

Subscription Response

SubscriptionResponseType: Anfragenachrichten in Abonnementschema.

Gruppenzugehoerigkeit: *SubscriptionAdministrationResponseGroup* *SubscriptionResponseGroup* *ResponseMessageGroup*

Element	Typ	Kommentar
<i>Transaction</i>	TransactionType	Transaktions-Identifizier, kopiert von der korrespondierenden Anfrage.
<i>CalcTime</i>	xs:duration	Benötigte Zeit um die beinhalteten Resultate zu berechnen.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  → name="SubscriptionResponseType">
2   <xs:sequence>
3     <xs:element name="Transaction" type="TransactionType"/>
4     <xs:element name="CalcTime" type="xs:duration"/>
5     <xs:choice>
6       <xs:group ref="SubscriptionAdministrationResponseGroup"/>
7       <xs:group ref="SubscriptionResponseGroup"/>
8       <xs:group ref="ResponseMessageGroup"/>
9     </xs:choice>
10  </xs:sequence>
11 </xs:complexType>

```

XML Quellcode 7.7: SubscriptionResponseType

7.2 Basisgruppen

Benutzer-initiiert

UserTriggeredRequestGroup: Anfragen, die direkt durch Userinteraktion ausgelöst werden.

Element	Typ	Kommentar
<input type="checkbox"/> <i>Language</i>	xs:language	Angefragte Sprache für die Resultate. Alle Texte zurückgegeben in den betreffenden Antworten, sollten in dieser Sprache sein.
<i>Auth</i>	AuthType	Authentifizierungsinformation des entsprechenden Benutzers.
<input type="radio"/> <i>OpenSession</i>	OpenSessionRequestType	Öffnen einer Session für einen User.
<input type="radio"/> <i>CloseSession</i>	CloseSessionRequestType	Explizites Schließen einer Session.
<input type="radio"/> <i>TokenGeneration</i>	TokenGenerationRequestType	Generieren eines Tokens für einen User, der sein Passwort ersetzt.
<input type="radio"/> <i>Availability</i>	AvailabilityRequestType	Die Verfügbarkeit für gegebene Buchungsziele erhalten.
<input type="radio"/> <i>PlaceAvailability</i>	PlaceAvailabilityRequestType	Die verfügbare Kapazität für gegebene Standorte erhalten.
<input type="radio"/> <i>PriceInformation</i>	PriceInformationRequestType	Beziehen von Preisinformationen.
<input type="radio"/> <i>Booking</i>	BookingRequestType	Buchung eines

```
1 <xs:group xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema" name="
  ↳ UserTriggeredRequestGroup">
2   <xs:sequence>
3     <xs:element name="Language" type="xs:language" minOccurs="0"/>
4     <xs:element name="Auth" type="AuthType"/>
5     <xs:choice>
6       <xs:element name="OpenSession" type="OpenSessionRequestType"/>
7       <xs:element name="CloseSession" type="CloseSessionRequestType"/>
8       <xs:element name="TokenGeneration" type="TokenGenerationRequestType"/>
9       <xs:element name="Availability" type="AvailabilityRequestType"/>
10      <xs:element name="PlaceAvailability" type="PlaceAvailabilityRequestType"/>
11      <xs:element name="PriceInformation" type="PriceInformationRequestType"/>
12      <xs:element name="Booking" type="BookingRequestType"/>
13      <xs:element name="ChangeBooking" type="ChangeBookingRequestType"/>
14      <xs:element name="ChangeBookingState" type="ChangeBookingStateRequestType"/>
15      <xs:element name="CreateUser" type="CreateUserRequestType"/>
16      <xs:element name="ChangeUser" type="ChangeUserRequestType"/>
17      <xs:element name="SetBookingSettings" type="SetBookingSettingsRequestType"/>
18      <xs:element name="SetNavigationDestination" type="SetNavigationDestinationRequestType"/>
19    </xs:choice>
20  </xs:sequence>
21 </xs:group>
```

XML Quellcode 7.8: UserTriggeredRequestGroup

UserTriggeredResponseGroup: Antworten, die direkt durch einer Userinteraktion ausgelöst werden.

Element	Typ	Kommentar
□ <i>SessionID</i>	SessionIDType	Falls zurückgegeben wurde eine Session für einen User geöffnet. Diese Session-ID kann für Authentifizierung in der nächsten Anfrage verwendet werden.
□ <i>SessionTimeout</i>	xs:duration	Inaktivitäts-Timeout einer Session. Wenn eine Zeitspanne von Inaktivität für diese Session diesen Wert übersteigt, läuft die Session ab.
○ <i>OpenSession</i>	OpenSessionResponseType	Öffnen einer Session für den User.
○ <i>CloseSession</i>	CloseSessionResponseType	Explizites Schließen einer Session.
○ <i>TokenGeneration</i>	TokenGenerationResponseType	Generieren eines Tokens für einen User, der sein Passwort

```

1 <xs:group xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema" name="
  ↳ UserTriggeredResponseGroup">
2   <xs:sequence>
3     <xs:element name="SessionID" type="SessionIDType" minOccurs="0"/>
4     <xs:element name="SessionTimeout" type="xs:duration" minOccurs="0"/>
5     <xs:choice>
6       <xs:element name="OpenSession" type="OpenSessionResponseType"/>
7       <xs:element name="CloseSession" type="CloseSessionResponseType"/>
8       <xs:element name="TokenGeneration" type="TokenGenerationResponseType"/>
9       <xs:element name="Availability" type="AvailabilityResponseType"/>
10      <xs:element name="PlaceAvailability" type="PlaceAvailabilityResponseType"/>
11      <xs:element name="PriceInformation" type="PriceInformationResponseType"/>
12      <xs:element name="Booking" type="BookingResponseType"/>
13      <xs:element name="ChangeBooking" type="ChangeBookingResponseType"/>
14      <xs:element name="ChangeBookingState" type="ChangeBookingStateResponseType"/>
15      <xs:element name="CreateUser" type="CreateUserResponseType"/>
16      <xs:element name="ChangeUser" type="ChangeUserResponseType"/>
17      <xs:element name="SetBookingSettings" type="SetBookingSettingsResponseType"/>
18      <xs:element name="SetNavigationDestination" type="SetNavigationDestinationResponseType"/>
19    </xs:choice>
20  </xs:sequence>
21 </xs:group>

```

XML Quellcode 7.9: UserTriggeredResponseGroup

Herkunft/Zielangabe

OriginDestGroup: Gruppe für Ausgangspunkt und Reisezielort.

	Element	Typ	Kommentar
<input type="checkbox"/>	<i>Origin</i>	OriginDestType	Lokalität, wo Buchungsziel aufgenommen wird.
<input type="checkbox"/>	<i>Dest</i>	OriginDestType	Lokalität, wo Buchungsziel zurückgegeben wird.

```

1 <xs:group xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema" name="
  ↳ OriginDestGroup">
2   <xs:sequence>
3     <xs:element name="Origin" type="OriginDestType" minOccurs="0"/>
4     <xs:element name="Dest" type="OriginDestType" minOccurs="0"/>
5   </xs:sequence>
6 </xs:group>

```

XML Quellcode 7.10: OriginDestGroup

Gruppen rein zur Organisation

Anmerkung: Im XML Schema sind weitere Gruppen definiert, die jedoch nur der Organisation dienen und keinen Einfluss auf das Nachrichtenformat haben und deshalb hier nicht aufgelistet sind.

Name	Kommentar
StaticDataRequestGroup	
StaticDataResponseGroup	
SubscriptionAdministrationRequestGroup	
SubscriptionAdministrationResponseGroup	
SubscriptionRequestGroup	
SubscriptionResponseGroup	
RequestMessageGroup	
ResponseMessageGroup	
PushMessageGroup	

7.3 Dienst A - Sitzungen

Session öffnen

OpenSessionRequestType: Anfrage für das Öffnen einer Session für den User. Kann für Funktionen 2, 4, und 6 verwendet werden.

Gruppenzugehörigkeit: *UserTriggeredRequestGroup*

Element	Typ	Kommentar
(<i>leer</i>)		

```
1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ↪ name="OpenSessionRequestType"/>
```

XML Quellcode 7.11: OpenSessionRequestType

OpenSessionResponseType: Antwort für das Öffnen einer Session für einen User.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehörigkeit: *UserTriggeredResponseGroup*

Element	Typ	Kommentar
(leer)		

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="OpenSessionResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType"/>
4   </xs:complexContent>
5 </xs:complexType>

```

XML Quellcode 7.12: OpenSessionResponseType

Session schließen

CloseSessionRequestType: Anfrage für das Schließen einer existierenden Session.
 Gruppenzugehörigkeit: *UserTriggeredRequestGroup*

Element	Typ	Kommentar
(leer)		

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CloseSessionRequestType"/>

```

XML Quellcode 7.13: CloseSessionRequestType

CloseSessionResponseType: Antwort für das Schließen einer existierenden Session.
 Basistyp: *AbstractBaseResponseType*
 Gruppenzugehörigkeit: *UserTriggeredResponseGroup*

Element	Typ	Kommentar
(leer)		

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CloseSessionResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType"/>
4   </xs:complexContent>
5 </xs:complexType>

```

XML Quellcode 7.14: CloseSessionResponseType

7.4 Dienst B - Abonnements

Heartbeat

HeartBeatRequestType: Anfrage für das Prüfen einer Verbindung.
Gruppenzugehörigkeit: *SubscriptionAdministrationRequestGroup*

Element	Typ	Kommentar
(leer)		

```
1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ↳ name="HeartBeatRequestType"/>
```

XML Quellcode 7.15: HeartBeatRequestType

HeartBeatResponseType: Antwort für das Prüfen einer Verbindung.
Basistyp: *AbstractBaseResponseType*
Gruppenzugehörigkeit: *SubscriptionAdministrationResponseGroup*

Element	Typ	Kommentar
(leer)		

```
1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ↳ name="HeartBeatResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType"/>
4   </xs:complexContent>
5 </xs:complexType>
```

XML Quellcode 7.16: HeartBeatResponseType

7.5 Dienst C - Tokens

Tokengenerierung

TokenGenerationRequestType: Anfrage für eine Tokengenerierung für einen User.
Gruppenzugehörigkeit: *UserTriggeredRequestGroup*

Element	Typ	Kommentar
<i>(leer)</i>		

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="TokenGenerationRequestType"/>

```

XML Quellcode 7.17: TokenGenerationRequestType

TokenGenerationResponseType: Antwort für eine Tokengenerierung für einen User.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *UserTriggeredResponseGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <i>Token</i>	xs:token	Token für einen User. Kann zukünftig anstelle eines Passwortes für die Authentifizierung verwendet werden.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="TokenGenerationResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="Token" type="xs:token" minOccurs="0"/>
6       </xs:sequence>
7     </xs:extension>
8   </xs:complexContent>
9 </xs:complexType>

```

XML Quellcode 7.18: TokenGenerationResponseType

7.6 Dienst 1 - Statische Daten

Abfrage Buchungsziele

BookingTargetsInfoRequestType: Anfrage um alle Informationen zu allen Buchungszielen zu erhalten.

Gruppenzugehoerigkeit: *StaticDataRequestGroup*

Element	Typ	Kommentar
☐ ☐ <i>ProviderFilter</i>	ProviderIDType	Falls gesetzt, beinhaltet der Filter jene Provider, für die Daten angefragt sind. Falls nicht gesetzt, sind Daten für alle Provider angefragt.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="BookingTargetsInfoRequestType">
2   <xs:sequence>
3     <xs:element name="ProviderFilter" type="ProviderIDType" minOccurs="0" maxOccurs="unbounded"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.19: BookingTargetsInfoRequestType

BookingTargetsInfoResponseType: Antwort um alle Informationen für alle Buchungsziele zu erhalten.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *StaticDataResponseGroup*

Element	Typ	Kommentar
☐ <i>Timestamp</i>	xs:dateTime	Zeitstempel dieser Datenlieferung. Das liefernde System sollte in der Lage sein, Änderungen durch diesen Zeitstempel zu identifizieren.
☐ ☐ <i>Bookee</i>	BookingTargetType	
☐ ☐ <i>Place</i>	PlaceType	
☐ ☐ <i>PlaceGroup</i>	PlaceGroupType	
☐ ☐ <i>FreeFloatingArea</i>	FloatingAreaType	
☐ ☐ <i>Provider</i>	ProviderType	
☐ ☐ <i>Attributes</i>	AttributeType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="BookingTargetsInfoResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="Timestamp" type="xs:dateTime" minOccurs="0"/>
6         <xs:element name="Bookee" type="BookingTargetType" minOccurs="0" maxOccurs="unbounded"/>
7         <xs:element name="Place" type="PlaceType" minOccurs="0" maxOccurs="unbounded"/>
8         <xs:element name="PlaceGroup" type="PlaceGroupType" minOccurs="0" maxOccurs="unbounded"/>
9         <xs:element name="FreeFloatingArea" type="FloatingAreaType" minOccurs="0" maxOccurs="unbounded"
  ↳ "/>
10        <xs:element name="Provider" type="ProviderType" minOccurs="0" maxOccurs="unbounded"/>
11        <xs:element name="Attributes" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/>
12      </xs:sequence>

```

```

13 </xs:extension>
14 </xs:complexContent>
15 </xs:complexType>

```

XML Quellcode 7.20: BookingTargetsInfoResponseType

Abfrage Änderungen Buchungsziele

ChangedProvidersRequestType: Anfrage um jene Anbieter mit veränderten statischen Daten zu erhalten.

Gruppenzugehoerigkeit: *StaticDataRequestGroup*

Element	Typ	Kommentar
<i>Timestamp</i>	xs:dateTime	Zeitstempel der statischen Datenlieferung, für welche die Änderungen angefordert wurden.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ChangedProvidersRequestType">
2   <xs:sequence>
3     <xs:element name="Timestamp" type="xs:dateTime"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.21: ChangedProvidersRequestType

ChangedProvidersResponseType: Antwort um Anbieter mit veränderten statischen Daten zu erhalten.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *StaticDataResponseGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <input type="checkbox"/> <i>Provider</i>	ProviderIDType	Liste von Anbietern mit Änderungen in deren statischen Daten.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ChangedProvidersResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="Provider" type="ProviderIDType" minOccurs="0" maxOccurs="unbounded"/>
6       </xs:sequence>

```

```

7     </xs:extension>
8 </xs:complexContent>
9 </xs:complexType>

```

XML Quellcode 7.22: ChangedProvidersResponseType

7.7 Dienst 2 - Verfügbarkeitsauskunft

Abfrage Verfügbarkeiten

AvailabilityRequestType: Synchroner Anfrage von Verfügbarkeiten von Buchungszielen.
 Gruppenzugehörigkeit: *UserTriggeredRequestGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <input type="checkbox"/> <i>BookingTarget</i>	BookingTargetPropertiesType	
<input type="checkbox"/> <i>Circle</i>	GeoCircleType	
<input type="checkbox"/> <i>GeoRectangle</i>	GeoRectangleType	
<input type="checkbox"/> <i>TimePeriod</i>	TimePeriodType	Relevante Zeitperiode, für diese Verfügbarkeiten der gegebenen Buchungsziele zurückgegeben werden sollen. Falls nicht gegeben, sollten aktuell verfügbare Buchungsziele ausgegeben werden.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   → name="AvailabilityRequestType">
2   <xs:sequence>
3     <xs:choice>
4       <xs:element name="BookingTarget" type="BookingTargetPropertiesType" maxOccurs="unbounded"/>
5       <xs:element name="Circle" type="GeoCircleType"/>
6       <xs:element name="GeoRectangle" type="GeoRectangleType"/>
7     </xs:choice>
8     <xs:element name="TimePeriod" type="TimePeriodType" minOccurs="0"/>
9   </xs:sequence>
10 </xs:complexType>

```

XML Quellcode 7.23: AvailabilityRequestType

```

1 <Ixsi>
2   <Request>
3     <Transaction>
4       <TimeStamp>2014-11-03T11:19:01.976+01:00</TimeStamp>
5       <MessageID>100</MessageID>
6     </Transaction>
7     <Auth>
8       <UserInfo>
9         <ProviderID>2</ProviderID>
10        <UserID>40</UserID>
11        <Password>x</Password>
12      </UserInfo>
13    </Auth>
14    <Availability>
15      <BookingTarget>
16        <ID>
17          <BookeeID>10</BookeeID>
18          <ProviderID>2</ProviderID>
19        </ID>
20      </BookingTarget>
21      <TimePeriod>
22        <Begin>2014-11-04T15:01:00.000+01:00</Begin>
23        <End>2014-11-09T17:14:00.000+01:00</End>
24      </TimePeriod>
25    </Availability>
26  </Request>
27 </Ixsi>

```

XML Quellcode 7.24: AvailabilityRequest Beispiel

AvailabilityResponseType: Synchroner Antwort für die Verfügbarkeit von Buchungszielen.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehörigkeit: *UserTriggeredResponseGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <input type="checkbox"/> <i>BookingTarget</i>	BookingTargetAvailabilityType	Liste von Buchungszielen und deren Verfügbarkeiten. Nicht gelistete Buchungsziele sind dem Car-Sharing-System entweder nicht bekannt, oder nicht in der angefragten Zeitperiode verfügbar.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
2   ↪ name="AvailabilityResponseType">
3   <xs:complexContent>
4     <xs:extension base="AbstractBaseResponseType">
5       <xs:sequence>

```

7 Nachrichten

```
5     <xs:element name="BookingTarget" type="BookingTargetAvailabilityType" minOccurs="0" maxOccurs=
      ↳ "unbounded"/>
6   </xs:sequence>
7 </xs:extension>
8 </xs:complexContent>
9 </xs:complexType>
```

XML Quellcode 7.25: AvailabilityResponseType

```
1 <Ixsi>
2   <Response>
3     <Transaction>
4       <TimeStamp>2014-11-03T11:19:01.976+01:00</TimeStamp>
5       <MessageID>100</MessageID>
6     </Transaction>
7     <CalcTime>PT0.090S</CalcTime>
8     <Availability>
9       <BookingTarget>
10        <ID>
11          <BookeeID>10</BookeeID>
12          <ProviderID>2</ProviderID>
13        </ID>
14        <Inavailability>
15          <Begin>2014-11-09T17:00:00.000+01:00</Begin>
16          <End>2014-11-09T20:00:00.000+01:00</End>
17        </Inavailability>
18        <CurrentStateOfCharge>57</CurrentStateOfCharge>
19      </BookingTarget>
20    </Availability>
21  </Response>
22 </Ixsi>
```

XML Quellcode 7.26: AvailabilityResponse Beispiel

Abfrage Standortkapazitäten (Dienst 2a)

PlaceAvailabilityRequestType: Synchrone Anfrage für verfügbare Kapazität der Standorte.
Gruppenzugehörigkeit: *UserTriggeredRequestGroup*

	Element	Typ	Kommentar
○ □	<i>PlaceID</i>	ProviderPlaceIDType	
○	<i>Circle</i>	GeoCircleType	
○	<i>GeoRectangle</i>	GeoRectangleType	

```
1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
      ↳ name="PlaceAvailabilityRequestType">
2   <xs:sequence>
3     <xs:choice>
4       <xs:element name="PlaceID" type="ProviderPlaceIDType" maxOccurs="unbounded"/>
```

```

5     <xs:element name="Circle" type="GeoCircleType"/>
6     <xs:element name="GeoRectangle" type="GeoRectangleType"/>
7   </xs:choice>
8 </xs:sequence>
9 </xs:complexType>

```

XML Quellcode 7.27: PlaceAvailabilityRequestType

PlaceAvailabilityResponseType: Synchroner Antwort für verfügbare Kapazität der Standorte.
 Basistyp: *AbstractBaseResponseType*
 Gruppenzugehörigkeit: *UserTriggeredResponseGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <input type="checkbox"/> <i>Place</i>	PlaceAvailabilityType	Liste der Standorte und deren verfügbare Kapazität. Nicht gelistete Standorte sind dem Car-Sharing-System entweder nicht bekannt, oder nicht in der angefragten Zeitperiode verfügbar.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   → name="PlaceAvailabilityResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="Place" type="PlaceAvailabilityType" minOccurs="0" maxOccurs="unbounded"/>
6       </xs:sequence>
7     </xs:extension>
8   </xs:complexContent>
9 </xs:complexType>

```

XML Quellcode 7.28: PlaceAvailabilityResponseType

7.8 Dienst 3 - Verfügbarkeitsabonnement

Verfügbarkeitsabonnement

AvailabilitySubscriptionRequestType: Anfrage zum Abonnement von Verfügbarkeiten von gegebenen Buchungszielen.
 Gruppenzugehörigkeit: *SubscriptionRequestGroup*

Element	Typ	Kommentar
<input checked="" type="checkbox"/> <i>BookingTargetID</i>	BookingTargetIDType	Buchungsziele abonnieren/ Abo kündigen.
<input type="checkbox"/> <i>Unsubscription</i>	xs:boolean	Falls gesetzt, Abo von gegebenen Buchungszielen kündigen. Andernfalls abonnieren.
<input type="checkbox"/> <i>EventHorizon</i>	xs:duration	Falls gesetzt, beschreibt der Ereignishorizont die Zeit in der Zukunft, für diese das Abo valide ist. Der Wert wird für alle abonnierten Buchungsziele verwendet. Diese Dauer ist relativ zur aktuellen Zeit.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ↳ name="AvailabilitySubscriptionRequestType">
2   <xs:sequence>
3     <xs:element name="BookingTargetID" type="BookingTargetIDType" maxOccurs="unbounded"/>
4     <xs:element name="Unsubscription" type="xs:boolean" default="false" minOccurs="0"/>
5     <xs:element name="EventHorizon" type="xs:duration" minOccurs="0"/>
6   </xs:sequence>
7 </xs:complexType>

```

XML Quellcode 7.29: AvailabilitySubscriptionRequestType

```

1 <Ixsi>
2   <SubscriptionRequest>
3     <Transaction>
4       <TimeStamp>2014-11-03T11:23:47.309+01:00</TimeStamp>
5       <MessageID>100</MessageID>
6     </Transaction>
7     <AvailabilitySubscription>
8       <BookingTargetID>
9         <BookeeID>7</BookeeID>
10        <ProviderID>2</ProviderID>
11      </BookingTargetID>
12    </AvailabilitySubscription>
13  </SubscriptionRequest>
14 </Ixsi>

```

XML Quellcode 7.30: AvailabilitySubscriptionRequest Beispiel

AvailabilitySubscriptionResponseType: Antwort für Abo von Verfügbarkeiten für ein gegebenes Buchungsziel.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehörigkeit: *SubscriptionResponseGroup*

Element	Typ	Kommentar
<i>(leer)</i>		

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="AvailabilitySubscriptionResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType"/>
4   </xs:complexContent>
5 </xs:complexType>

```

XML Quellcode 7.31: AvailabilitySubscriptionResponseType

AvailabilitySubscriptionStatusRequestType: Anfrage um alle abonnierten Buchungsziele zu erhalten.

Gruppenzugehoerigkeit: *SubscriptionRequestGroup*

Element	Typ	Kommentar
(leer)		

AvailabilitySubscriptionStatusResponseType: Antwort um alle abonnierten Buchungsziele zu erhalten.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *SubscriptionResponseGroup*

Element	Typ	Kommentar	
<input type="checkbox"/> <input type="checkbox"/>	<i>BookingTargetID</i>	BookingTargetIDType	Abonnierte Buchungsziele.

AvailabilityPushMessageType: Push-Nachricht, die Änderungen in der Verfügbarkeit von abonnierten Buchungszielen beinhaltet.

Gruppenzugehoerigkeit: *PushMessageGroup*

Element	Typ	Kommentar	
<input type="checkbox"/>	<i>AvailabilityChange</i>	BookingTargetChangeAvailabilityType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="AvailabilityPushMessageType">
2   <xs:sequence>
3     <xs:element name="AvailabilityChange" type="BookingTargetChangeAvailabilityType" maxOccurs="
  ↳ unbounded"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.32: AvailabilityPushMessageType

```

1 <Ixsi>
2   <SubscriptionMessage>
3     <AvailabilityPush>
4       <AvailabilityChange>
5         <ID>
6           <BookeeID>7</BookeeID>
7           <ProviderID>2</ProviderID>
8         </ID>
9         <Inavailability>
10          <Begin>2014-11-04T15:00:00.000+01:00</Begin>
11          <End>2014-11-04T17:30:00.000+01:00</End>
12        </Inavailability>
13      </AvailabilityChange>
14    </AvailabilityPush>
15  </SubscriptionMessage>
16 </Ixsi>

```

XML Quellcode 7.33: AvailabilityPush Beispiel

Vollständige Verfügbarkeitsabonnementsinformation

CompleteAvailabilityRequestType: Anfrage um den vollständigen aktuellen Status von abonnierten Verfügbarkeiten zu erhalten.

Gruppzugehörigkeit: *RequestMessageGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <i>MaxTargets</i>	xs:integer	Definiert die maximale Anzahl von Zielen, die in einer Nachricht zurückgegeben werden.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   < name="CompleteAvailabilityRequestType">
2   <xs:sequence>
3     <xs:element name="MaxTargets" type="xs:integer" minOccurs="0"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.34: CompleteAvailabilityRequestType

```

1 <Ixsi>
2   <SubscriptionRequest>
3     <Transaction>
4       <TimeStamp>2014-11-03T11:23:30.059+01:00</TimeStamp>
5       <MessageID>100</MessageID>
6     </Transaction>
7     <CompleteAvailability>
8       <MaxTargets>5</MaxTargets>
9     </CompleteAvailability>
10  </SubscriptionRequest>
11 </Ixsi>

```

XML Quellcode 7.35: CompleteAvailabilityRequest Beispiel

CompleteAvailabilityResponseType: Antwort um den vollständigen aktuellen Status der Verfügbarkeiten zu erhalten. Einige solcher Antworten können zu einer einzigen Anfrage gehören.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehörigkeit: *ResponseMessageGroup*

Element	Typ	Kommentar
<i>MessageBlockID</i>	xs:token	ID des Blocks, zu dem diese Nachricht gehört. Alle Nachrichten des Blocks haben dieselbe ID.
<input type="checkbox"/> <i>Last</i>	xs:boolean	Falls gesetzt, ist die Nachricht die letzte des korrespondierenden Blocks. Andernfalls werden mehr Nachrichten desselben Blocks folgen.
<input type="checkbox"/> <input type="checkbox"/> <i>BookingTarget</i>	BookingTargetAvailabilityType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CompleteAvailabilityResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="MessageBlockID" type="xs:token"/>
6         <xs:element name="Last" type="xs:boolean" default="false" minOccurs="0"/>
7         <xs:element name="BookingTarget" type="BookingTargetAvailabilityType" minOccurs="0" maxOccurs=
  ↳ "unbounded"/>
8       </xs:sequence>
9     </xs:extension>
10  </xs:complexContent>
11 </xs:complexType>

```

XML Quellcode 7.36: CompleteAvailabilityResponseType

```

1 <Ixsi>
2   <SubscriptionResponse>
3     <Transaction>
4       <TimeStamp>2014-11-03T11:23:30.059+01:00</TimeStamp>
5       <MessageID>100</MessageID>
6     </Transaction>
7     <CalcTime>PT0.000S</CalcTime>
8     <CompleteAvailability>

```

```

9      <MessageBlockID>100</MessageBlockID>
10     <Last>true</Last>
11     <BookingTarget>
12       <ID>
13         <BookeeID>14</BookeeID>
14         <ProviderID>2</ProviderID>
15       </ID>
16       <Inavailability>
17         <Begin>2014-11-04T15:00:00.000+01:00</Begin>
18         <End>2014-11-04T18:00:00.000+01:00</End>
19       </Inavailability>
20     </BookingTarget>
21     <BookingTarget>
22       <ID>
23         <BookeeID>15</BookeeID>
24         <ProviderID>2</ProviderID>
25       </ID>
26     </BookingTarget>
27     <BookingTarget>
28       <ID>
29         <BookeeID>16</BookeeID>
30         <ProviderID>2</ProviderID>
31       </ID>
32     </BookingTarget>
33   </CompleteAvailability>
34 </SubscriptionResponse>
35 </Ixsii>

```

XML Quellcode 7.37: CompleteAvailabilityResponse Beispiel

Standortkapazitätsabonnement (Dienst 3a)

PlaceAvailabilitySubscriptionRequestType: Anfrage zum Abonnieren von verfügbaren Kapazitäten von Standorten.

Gruppenzugehörigkeit: *SubscriptionRequestGroup*

Element	Typ	Kommentar
<input checked="" type="checkbox"/> <i>PlaceID</i>	ProviderPlaceIDType	Standorte zu abonnieren / Abo kündigen.
<input type="checkbox"/> <i>Unsubscription</i>	xs:boolean	Falls gesetzt, Kündigen der Abos gegebener Buchungsziele. Andernfalls abonnieren.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   name="PlaceAvailabilitySubscriptionRequestType">
2   <xs:sequence>
3     <xs:element name="PlaceID" type="ProviderPlaceIDType" maxOccurs="unbounded"/>
4     <xs:element name="Unsubscription" type="xs:boolean" default="false" minOccurs="0"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 7.38: PlaceAvailabilitySubscriptionRequestType

PlaceAvailabilitySubscriptionResponseType: Antwort zum Abonnieren der verfügbaren Kapazität von Standorten.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehörigkeit: *SubscriptionResponseGroup*

Element	Typ	Kommentar
<i>(leer)</i>		

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ↳ name="PlaceAvailabilitySubscriptionResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType"/>
4   </xs:complexContent>
5 </xs:complexType>

```

XML Quellcode 7.39: PlaceAvailabilitySubscriptionResponseType

PlaceAvailabilitySubscriptionStatusRequestType: Anfrage um alle abonnierten Standorte zu erhalten.

Gruppenzugehörigkeit: *SubscriptionRequestGroup*

Element	Typ	Kommentar
<i>(leer)</i>		

PlaceAvailabilitySubscriptionStatusResponseType: Antwort um alle abonnierten Standorte zu erhalten

Basistyp: *AbstractBaseResponseType*

Gruppenzugehörigkeit: *SubscriptionResponseGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <input type="checkbox"/>	<i>PlaceID</i>	ProviderPlaceIDType
		Abonnierte Standorte.

PlaceAvailabilityPushMessageType: Push-Nachricht, die Änderungen in der Verfügbarkeit von abonnierten Standorten enthält.

Gruppenzugehörigkeit: *PushMessageGroup*

Element	Typ	Kommentar
☐ <i>PlaceAvailability</i>	PlaceAvailabilityType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="PlaceAvailabilityPushMessageType">
2   <xs:sequence>
3     <xs:element name="PlaceAvailability" type="PlaceAvailabilityType" maxOccurs="unbounded"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.40: PlaceAvailabilityPushMessageType

Vollständige Standortkapazitätsinformation (Dienst 3a)

CompletePlaceAvailabilityRequestType: Anfrage um den vollständigen aktuellen Status abon-
nierter Standorte zu erhalten.

Gruppenzugehoerigkeit: *RequestMessageGroup*

Element	Typ	Kommentar
☐ <i>MaxPlaces</i>	xs:integer	Definiert die maximale Anzahl an Standorten, die in einer Nachricht zurückgegeben werden.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CompletePlaceAvailabilityRequestType">
2   <xs:sequence>
3     <xs:element name="MaxPlaces" type="xs:integer" minOccurs="0"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.41: CompletePlaceAvailabilityRequestType

CompletePlaceAvailabilityResponseType: Antwort um den vollständigen aktuellen Status abon-
nierter Standorte zu erhalten. Verschiedene solcher Antworten können zu einer einzelnen An-
frage gehören.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *ResponseMessageGroup*

Element	Typ	Kommentar
<i>MessageBlockID</i>	xs:token	ID des Blocks, zu dem die Nachricht gehört. Alle Nachrichten eines Blocks haben dieselbe ID.
<input type="checkbox"/> <i>Last</i>	xs:boolean	Falls gesetzt, ist diese Nachricht die Letzte des korrespondierenden Blocks. Andernfalls werden mehr Nachrichten desselben Blocks folgen.
<input type="checkbox"/> <input type="checkbox"/> <i>PlaceAvailability</i>	PlaceAvailabilityType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CompletePlaceAvailabilityResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="MessageBlockID" type="xs:token"/>
6         <xs:element name="Last" type="xs:boolean" default="false" minOccurs="0"/>
7         <xs:element name="PlaceAvailability" type="PlaceAvailabilityType" minOccurs="0" maxOccurs="
  ↳ unbounded"/>
8       </xs:sequence>
9     </xs:extension>
10  </xs:complexContent>
11 </xs:complexType>

```

XML Quellcode 7.42: CompletePlaceAvailabilityResponseType

7.9 Dienst 4 - Buchung / Buchungsänderung

Buchung

BookingRequestType: Anfrage zum Buchen eines Buchungsziels. Eine Buchung impliziert nicht eine Nicht-Verfügbarkeit.

Gruppenzugehoerigkeit: *OriginDestGroup UserTriggeredRequestGroup*

Element	Typ	Kommentar
<i>BookingTargetID</i>	BookingTargetIDType	
<i>TimePeriodProposal</i>	TimePeriodProposalType	Vorgeschlagene Zeitperiode für die Nutzung eines Buchungsziels.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="BookingRequestType">
2   <xs:sequence>
3     <xs:element name="BookingTargetID" type="BookingTargetIDType"/>
4     <xs:element name="TimePeriodProposal" type="TimePeriodProposalType"/>
5     <xs:group ref="OriginDestGroup"/>
6   </xs:sequence>
7 </xs:complexType>

```

XML Quellcode 7.43: BookingRequestType

```

1 <Ixsi>
2   <Request>
3     <Transaction>
4       <TimeStamp>2014-11-03T11:19:02.258+01:00</TimeStamp>
5       <MessageID>100</MessageID>
6     </Transaction>
7     <Auth>
8       <UserInfo>
9         <ProviderID>2</ProviderID>
10        <UserID>40</UserID>
11        <Password>x</Password>
12      </UserInfo>
13    </Auth>
14    <Booking>
15      <BookingTargetID>
16        <BookeeID>14</BookeeID>
17        <ProviderID>2</ProviderID>
18      </BookingTargetID>
19      <TimePeriodProposal>
20        <Begin>2014-11-04T15:21:00.000+01:00</Begin>
21        <End>2014-11-04T17:18:00.000+01:00</End>
22      </TimePeriodProposal>
23    </Booking>
24  </Request>
25 </Ixsi>

```

XML Quellcode 7.44: BookingRequest Beispiel

BookingResponseType: Antwort zur Buchung eines Buchungsziels.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *UserTriggeredResponseGroup*

Element	Typ	Kommentar
☐ <i>Booking</i>	BookingType	Information über eine erfolgreiche Buchung.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="BookingResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="Booking" type="BookingType" minOccurs="0"/>
6       </xs:sequence>

```

```

7   </xs:extension>
8 </xs:complexContent>
9 </xs:complexType>

```

XML Quellcode 7.45: BookingResponseType

```

1 <Ixsi>
2   <Response>
3     <Transaction>
4       <TimeStamp>2014-11-03T11:19:02.258+01:00</TimeStamp>
5       <MessageID>100</MessageID>
6     </Transaction>
7     <CalcTime>PT0.325S</CalcTime>
8     <Booking>
9       <BookingID>2-9</BookingID>
10      <TimePeriod>
11        <Begin>2014-11-04T15:00:00.000+01:00</Begin>
12        <End>2014-11-04T17:30:00.000+01:00</End>
13      </TimePeriod>
14    </Booking>
15  </Response>
16 </Ixsi>

```

XML Quellcode 7.46: BookingResponse Beispiel

Buchungsänderung

ChangeBookingRequestType: Anfrage zur Änderung einer Buchung.
 Gruppenzugehörigkeit: *UserTriggeredRequestGroup*

Element	Typ	Kommentar
<i>BookingID</i>	BookingIDType	Zu ändernde Buchung.
<input type="radio"/> <i>NewTimePeriodProposal</i>	TimePeriodProposalType	Neuer Zeitperioden-Vorschlag um die Buchung zu ändern.
<input type="radio"/> <i>Cancel</i>	xs:boolean	Die Buchung soll abgebrochen werden.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   → name="ChangeBookingRequestType">
2   <xs:sequence>
3     <xs:element name="BookingID" type="BookingIDType"/>
4     <xs:choice>
5       <xs:element name="NewTimePeriodProposal" type="TimePeriodProposalType"/>
6       <xs:element name="Cancel" type="xs:boolean"/>
7     </xs:choice>
8   </xs:sequence>
9 </xs:complexType>

```

XML Quellcode 7.47: ChangeBookingRequestType

ChangeBookingResponseType: Antwort für eine Buchungsänderung.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehörigkeit: *UserTriggeredResponseGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <i>Booking</i>	BookingType	Information über eine erfolgreiche Umbuchung.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ChangeBookingResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="Booking" type="BookingType" minOccurs="0"/>
6       </xs:sequence>
7     </xs:extension>
8   </xs:complexContent>
9 </xs:complexType>

```

XML Quellcode 7.48: ChangeBookingResponseType

Externe Buchungen abfragen (Dienst 4a)

ExternalBookingSubscriptionRequestType: Anfrage um BuchungsID abzufragen die extern erstellt wurden. Kann zur Abfrage von Verbrauchsdaten verwendet werden.

Gruppenzugehörigkeit: *SubscriptionRequestGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <input type="checkbox"/> <i>UserInfo</i>	UserInfoType	Buchungen nur für die entsprechenden Benutzer abrufen (falls weggelassen alle erlaubten Benutzer abonnieren).
<input type="checkbox"/> <i>Unsubscription</i>	xs:boolean	Falls gesetzt, gegebene Abos der Buchungen kündigen. Andernfalls abonnieren.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ExternalBookingSubscriptionRequestType">
2   <xs:sequence>
3     <xs:element name="UserInfo" type="UserInfoType" minOccurs="0" maxOccurs="unbounded"/>
4     <xs:element name="Unsubscription" type="xs:boolean" default="false" minOccurs="0"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 7.49: ExternalBookingSubscriptionRequestType

ExternalBookingSubscriptionResponseType: Antwort für Abonnement von externen Buchungen.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *SubscriptionResponseGroup*

Element	Typ	Kommentar
---------	-----	-----------

(*leer*)

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ExternalBookingSubscriptionResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType"/>
4   </xs:complexContent>
5 </xs:complexType>

```

XML Quellcode 7.50: ExternalBookingSubscriptionResponseType

ExternalBookingSubscriptionStatusRequestType: Anfrage um für externe Buchungen abonnierte Benutzer zu erhalten.

Gruppenzugehoerigkeit: *SubscriptionRequestGroup*

Element	Typ	Kommentar
---------	-----	-----------

(*leer*)

ExternalBookingSubscriptionStatusResponseType: Antwort um für externe Buchungen abonnierte Benutzer zu erhalten.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *SubscriptionResponseGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <input type="checkbox"/> <i>UserInfo</i>	UserInfoType	Abonnierte Benutzer.

ExternalBookingPushMessageType: Antwort für externe BuchungsIDs.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *PushMessageGroup*

Element	Typ	Kommentar
<i>externalBooking</i>	ExternalBookingType	Information über externe Buchungen.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ExternalBookingPushMessageType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="externalBooking" type="ExternalBookingType"/>
6       </xs:sequence>
7     </xs:extension>
8   </xs:complexContent>
9 </xs:complexType>

```

XML Quellcode 7.51: ExternalBookingPushMessageType

CompleteExternalBookingRequestType: Anfrage um alle abonnierten externen Buchungen zu erhalten.

Gruppenzugehoerigkeit: *RequestMessageGroup*

ResponseMessageGroup

Element	Typ	Kommentar
<input type="checkbox"/> <i>MaxResults</i>	xs:integer	Definiert die maximale Anzahl Resultate, die in einer Nachricht zurückgegeben werden.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CompleteExternalBookingRequestType">
2   <xs:sequence>
3     <xs:element name="MaxResults" type="xs:integer" minOccurs="0"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.52: CompleteExternalBookingRequestType

CompleteExternalBookingResponseType: Nachricht um den vollständigen aktuellen Status abon-
nierter externer Buchungen zu erhalten. Mehrere solcher Antworten können zu einer einzigen
Anfrage gehören.

Basistyp: *AbstractBaseResponseType*

Element	Typ	Kommentar
<i>MessageBlockID</i>	xs:token	ID des Blocks, zu dem diese Nachricht gehört. Alle Nachrichten eines Blockes haben dieselbe ID.
<input type="checkbox"/> <i>Last</i>	xs:boolean	Falls gesetzt, ist diese Nachricht die Letzte im korrespondierenden Block. Andernfalls folgen mehr Nachrichten desselben Blocks.
<input type="checkbox"/> <input type="checkbox"/> <i>externalBooking</i>	ExternalBookingType	Information über externe Buchungen.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CompleteExternalBookingResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="MessageBlockID" type="xs:token"/>
6         <xs:element name="Last" type="xs:boolean" default="false" minOccurs="0"/>
7         <xs:element name="externalBooking" type="ExternalBookingType" minOccurs="0" maxOccurs="
  ↳ unbounded"/>
8       </xs:sequence>
9     </xs:extension>
10  </xs:complexContent>
11 </xs:complexType>

```

XML Quellcode 7.53: CompleteExternalBookingResponseType

7.10 Dienst 5 - Buchungsabonnement

Buchungsabonnement

BookingAlertSubscriptionRequestType: Anfrage um Alarmmeldungen für gegebene Buchun-
gen zu abonnieren oder zu kündigen.

Gruppenzugehoerigkeit: *SubscriptionRequestGroup*

Element	Typ	Kommentar
<input checked="" type="checkbox"/> <i>BookingID</i>	BookingIDType	zu abonnierende Buchungen.
<input type="checkbox"/> <i>Unsubscription</i>	xs:boolean	Falls gesetzt, gegebene Abos der Buchungen kündigen. Andernfalls abonnieren.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="BookingAlertSubscriptionRequestType">
2   <xs:sequence>
3     <xs:element name="BookingID" type="BookingIDType" maxOccurs="unbounded"/>
4     <xs:element name="Unsubscription" type="xs:boolean" default="false" minOccurs="0"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 7.54: BookingAlertSubscriptionRequestType

BookingAlertSubscriptionResponseType: Antwort um Alarmmeldungen für gegebene Buchungen zu abonnieren.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *SubscriptionResponseGroup*

Element	Typ	Kommentar
<i>(leer)</i>		

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="BookingAlertSubscriptionResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType"/>
4   </xs:complexContent>
5 </xs:complexType>

```

XML Quellcode 7.55: BookingAlertSubscriptionResponseType

BookingAlertSubscriptionStatusRequestType: Anfrage um alle abonnierten Buchungen zu erhalten.

Gruppenzugehoerigkeit: *SubscriptionRequestGroup*

Element	Typ	Kommentar
<i>(leer)</i>		

BookingAlertSubscriptionStatusResponseType: Antwort um alle abonnierten Buchungen zu erhalten.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *SubscriptionResponseGroup*

Element	Typ	Kommentar
☐ ☐ <i>BookingID</i>	BookingIDType	Abonnierte Buchungen.

BookingAlertPushMessageType: Push-Nachricht, die Alarmmeldungen für abonnierte Buchungen enthält.

Gruppenzugehoerigkeit: *PushMessageGroup*

Element	Typ	Kommentar
☐ <i>BookingChange</i>	BookingChangeType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  → name="BookingAlertPushMessageType">
2   <xs:sequence>
3     <xs:element name="BookingChange" type="BookingChangeType" maxOccurs="unbounded"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.56: BookingAlertPushMessageType

Vollständige Buchungsabonnementsinformation

CompleteBookingAlertRequestType: Anfrage um den vollständigen aktuellen Status abonniert Buchungsalarmmeldungen zu erhalten.

Gruppenzugehoerigkeit: *RequestMessageGroup*

Element	Typ	Kommentar
☐ <i>MaxResults</i>	xs:integer	Definiert die maximale Anzahl Resultate, die in einer Nachricht zurückgegeben werden.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CompleteBookingAlertRequestType">
2   <xs:sequence>
3     <xs:element name="MaxResults" type="xs:integer" minOccurs="0"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.57: CompleteBookingAlertRequestType

CompleteBookingAlertResponseType: Nachricht um den vollständigen aktuellen Status abonierter Buchungen zu erhalten. Mehrere solcher Antworten können zu einer einzigen Anfrage gehören.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehörigkeit: *ResponseMessageGroup*

Element	Typ	Kommentar
<i>MessageBlockID</i>	xs:token	ID des Blocks, zu dem diese Nachricht gehört. Alle Nachrichten eines Blockes haben dieselbe ID.
<input type="checkbox"/> <i>Last</i>	xs:boolean	Falls gesetzt, ist diese Nachricht die Letzte im korrespondierenden Block. Andernfalls folgen mehr Nachrichten desselben Blocks.
<input type="checkbox"/> <input type="checkbox"/> <i>BookingChange</i>	BookingChangeType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CompleteBookingAlertResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="MessageBlockID" type="xs:token"/>
6         <xs:element name="Last" type="xs:boolean" default="false" minOccurs="0"/>
7         <xs:element name="BookingChange" type="BookingChangeType" minOccurs="0" maxOccurs="unbounded"
  ↳ >
8       </xs:sequence>
9     </xs:extension>
10  </xs:complexContent>
11 </xs:complexType>

```

XML Quellcode 7.58: CompleteBookingAlertResponseType

7.11 Dienst 6 - Preisauskunft

Preisinformation

PriceInformationRequestType: Anfrage um einen Preis zu erhalten.

Gruppenzugehoerigkeit: *OriginDestGroup UserTriggeredRequestGroup*

Element	Typ	Kommentar
<i>BookingTargetID</i>	BookingTargetIDType	
<i>TimePeriodProposal</i>	TimePeriodProposalType	Vorgeschlagene Zeitperiode für die Nutzung des Buchungsziels.
<i>Distance</i>	xs:nonNegativeInteger	Distanz zu Fahren in Metern.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="PriceInformationRequestType">
2   <xs:sequence>
3     <xs:element name="BookingTargetID" type="BookingTargetIDType"/>
4     <xs:element name="TimePeriodProposal" type="TimePeriodProposalType"/>
5     <xs:element name="Distance" type="xs:nonNegativeInteger"/>
6     <xs:group ref="OriginDestGroup"/>
7   </xs:sequence>
8 </xs:complexType>

```

XML Quellcode 7.59: PriceInformationRequestType

PriceInformationResponseType: Antwort um einen Preis zu erhalten.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *UserTriggeredResponseGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <i>Tariff</i>	TariffType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="PriceInformationResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="Tariff" type="TariffType" minOccurs="0"/>
6       </xs:sequence>
7     </xs:extension>
8   </xs:complexContent>
9 </xs:complexType>

```

XML Quellcode 7.60: PriceInformationResponseType

7.12 Dienst 7 - Verbrauchsdatenabonnement

Nutzungsdatenabonnement

ConsumptionSubscriptionRequestType: Anfrage um Verbrauchsdaten für gegebene Buchungen zu abonnieren oder zu kündigen.

Gruppenzugehörigkeit: *SubscriptionRequestGroup*

Element	Typ	Kommentar
<input checked="" type="checkbox"/> <i>BookingID</i>	BookingIDType	Verbrauchsdaten für Buchungen abonnieren.
<input type="checkbox"/> <i>Unsubscription</i>	xs:boolean	Falls gesetzt, gegebene Abos der Buchungen kündigen. Andernfalls abonnieren.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ↳ name="ConsumptionSubscriptionRequestType">
2   <xs:sequence>
3     <xs:element name="BookingID" type="BookingIDType" maxOccurs="unbounded"/>
4     <xs:element name="Unsubscription" type="xs:boolean" default="false" minOccurs="0"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 7.61: ConsumptionSubscriptionRequestType

ConsumptionSubscriptionResponseType: Antwort um Verbrauchsdaten für gegebene Buchungen zu abonnieren.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehörigkeit: *SubscriptionResponseGroup*

Element	Typ	Kommentar
<i>(leer)</i>		

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ↳ name="ConsumptionSubscriptionResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType"/>
4   </xs:complexContent>
5 </xs:complexType>

```

XML Quellcode 7.62: ConsumptionSubscriptionResponseType

ConsumptionSubscriptionStatusRequestType: Anfrage um alle abonnierten Verbrauchsdaten zu erhalten.

Gruppenzugehoerigkeit: *SubscriptionRequestGroup*

Element	Typ	Kommentar
<i>(leer)</i>		

ConsumptionSubscriptionStatusResponseType: Antwort um alle abonnierten Verbrauchsdaten zu erhalten.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *SubscriptionResponseGroup*

Element	Typ	Kommentar	
<input type="checkbox"/> <input type="checkbox"/>	<i>BookingID</i>	BookingIDType	Abonnierte Buchungen.

ConsumptionPushMessageType: Push-Nachricht, die Verbrauchsdaten für abonnierte Buchungen enthält.

Gruppenzugehoerigkeit: *PushMessageGroup*

Element	Typ	Kommentar
<input type="checkbox"/>	<i>Consumption</i>	ConsumptionType

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ConsumptionPushMessageType">
2   <xs:sequence>
3     <xs:element name="Consumption" type="ConsumptionType" maxOccurs="unbounded"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.63: ConsumptionPushMessageType

Vollständige Nutzungsdaten

CompleteConsumptionRequestType: Anfrage um den vollständigen aktuellen Status abonniertes Verbrauchsdaten zu erhalten.

Gruppenzugehörigkeit: *RequestMessageGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <i>MaxResults</i>	xs:integer	Definiert die maximale Anzahl Resultate, die in einer Nachricht zurückgegeben werden.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ↳ name="CompleteConsumptionRequestType">
2   <xs:sequence>
3     <xs:element name="MaxResults" type="xs:integer" minOccurs="0"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.64: CompleteConsumptionRequestType

CompleteConsumptionResponseType: Nachricht um den vollständigen aktuellen Status abonniertes Verbrauchsdaten zu erhalten. Verschiedene solcher Antworten können zu einer einzigen Anfrage gehören.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehörigkeit: *ResponseMessageGroup*

Element	Typ	Kommentar
<i>MessageBlockID</i>	xs:token	ID des Blocks, zu dem diese Nachricht gehört. Alle Nachrichten eines Blockes haben dieselbe ID.
<input type="checkbox"/> <i>Last</i>	xs:boolean	Falls gesetzt, ist diese Nachricht die Letzte im korrespondierenden Block. Andernfalls folgen mehr Nachrichten desselben Blocks.
<input type="checkbox"/> <input type="checkbox"/> <i>Consumption</i>	ConsumptionType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ↳ name="CompleteConsumptionResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="MessageBlockID" type="xs:token"/>
6         <xs:element name="Last" type="xs:boolean" default="false" minOccurs="0"/>
7         <xs:element name="Consumption" type="ConsumptionType" minOccurs="0" maxOccurs="unbounded"/>

```

```

8     </xs:sequence>
9   </xs:extension>
10 </xs:complexContent>
11 </xs:complexType>

```

XML Quellcode 7.65: CompleteConsumptionResponseType

7.13 Dienst 8 - Buchungsstatus ändern

ChangeBookingStateRequestType: Anfrage um den Buchungszustand (Fahrzeug) zu ändern.
Gruppenzugehörigkeit: *UserTriggeredRequestGroup*

Element	Typ	Kommentar
<i>BookingID</i>	BookingIDType	Die entsprechende Buchung.
<i>BookingState</i>	BookingStateType	Der neue Zustand der Buchung (open = Fahrzeug aufschließen, suspended = Fahrzeug abschließen aber Buchung nicht beenden, close = Fahrzeug abschließen und Buchung beenden).

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  → name="ChangeBookingStateRequestType">
2   <xs:sequence>
3     <xs:element name="BookingID" type="BookingIDType"/>
4     <xs:element name="BookingState" type="BookingStateType"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 7.66: ChangeBookingStateRequestType

ChangeBookingStateResponseType: Antwort auf Buchungszustandsänderung. Enthält zusätzliche Attribute die dem Nutzer übermittelt werden sollen.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehörigkeit: *UserTriggeredResponseGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <input type="checkbox"/> <i>Attributes</i>	AttributeType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ChangeBookingStateResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="Attributes" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/>
6       </xs:sequence>
7     </xs:extension>
8   </xs:complexContent>
9 </xs:complexType>

```

XML Quellcode 7.67: ChangeBookingStateResponseType

7.14 Dienst 9 - User Management

Benutzer erstellen

CreateUserRequestType: Anfrage um einen oder mehrere Benutzer zu erstellen (Löschen wird bewusst nicht unterstützt).

Gruppenzugehoerigkeit: *UserTriggeredRequestGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <i>User</i>	UserType	Vorgeschlagene Benutzeridentifikation.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CreateUserRequestType">
2   <xs:sequence>
3     <xs:element name="User" type="UserType" maxOccurs="unbounded"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.68: CreateUserRequestType

CreateUserResponseType: Antwort um Benutzer zu erstellen.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *UserTriggeredResponseGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <input type="checkbox"/> <i>User</i>	UserType	Informationen über den erstellten Benutzer (kann von der Anfrage abweichen, fehlgeschlagene werden nicht zurückgegeben).

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CreateUserResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="User" type="UserType" minOccurs="0" maxOccurs="unbounded"/>
6       </xs:sequence>
7     </xs:extension>
8   </xs:complexContent>
9 </xs:complexType>

```

XML Quellcode 7.69: CreateUserResponseType

Benutzer ändern

ChangeUserRequestType: Anfrage um einen Benutzer zu ändern..

Gruppenzugehoerigkeit: *UserTriggeredRequestGroup*

Element	Typ	Kommentar
<input checked="" type="checkbox"/> <i>User</i>	UserType	geänderter Benutzer.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ChangeUserRequestType">
2   <xs:sequence>
3     <xs:element name="User" type="UserType" maxOccurs="unbounded"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.70: ChangeUserRequestType

ChangeUserResponseType: Antwort Benutzeränderung.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *UserTriggeredResponseGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <input checked="" type="checkbox"/> <i>User</i>	UserType	Informationen über den geänderten Benutzer (fehlgeschlagene Änderungen werden nicht zurückgegeben).

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ChangeUserResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="User" type="UserType" minOccurs="0" maxOccurs="unbounded"/>
6       </xs:sequence>
7     </xs:extension>
8   </xs:complexContent>
9 </xs:complexType>

```

XML Quellcode 7.71: ChangeUserResponseType

7.15 Dienst 10 - Fahrzeugseinstellungsmanagement

Fahrzeugeinstellung setzen

SetBookingSettingsRequestType: Anfrage um Buchungseinstellungen zu setzen.

Gruppenzugehoerigkeit: *UserTriggeredRequestGroup*

Element	Typ	Kommentar
☐ <i>BookingSettings</i>	BookingSettingsType	Menge von anzuwendenden Einstellungen (Sitzposition, Temperatur).

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="SetBookingSettingsRequestType">
2   <xs:sequence>
3     <xs:element name="BookingSettings" type="BookingSettingsType" maxOccurs="unbounded"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.72: SetBookingSettingsRequestType

SetBookingSettingsResponseType: Antwort um Buchungseinstellungen zu setzen.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *UserTriggeredResponseGroup*

Element	Typ	Kommentar
<i>(leer)</i>		

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  → name="SetBookingSettingsResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType"/>
4   </xs:complexContent>
5 </xs:complexType>

```

XML Quellcode 7.73: SetBookingSettingsResponseType

Fahrzeugeinstellung abonnieren

BookingSettingsSubscriptionRequestType: Anfrage um Buchungseinstellungen zu abonnieren oder zu stornieren.

Gruppenzugehoerigkeit: *SubscriptionRequestGroup*

Element	Typ	Kommentar
<input checked="" type="checkbox"/> <i>BookingID</i>	BookingIDType	zu abonnierende BuchungsID.
<input type="checkbox"/> <i>Unsubscription</i>	xs:boolean	Falls gesetzt, abo stornieren.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  → name="BookingSettingsSubscriptionRequestType">
2   <xs:sequence>
3     <xs:element name="BookingID" type="BookingIDType" maxOccurs="unbounded"/>
4     <xs:element name="Unsubscription" type="xs:boolean" default="false" minOccurs="0"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 7.74: BookingSettingsSubscriptionRequestType

BookingSettingsSubscriptionResponseType: Antwort auf Buchungseinstellungsabo.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *SubscriptionResponseGroup*

Element	Typ	Kommentar
<i>(leer)</i>		

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="BookingSettingsSubscriptionResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType"/>
4   </xs:complexContent>
5 </xs:complexType>

```

XML Quellcode 7.75: BookingSettingsSubscriptionResponseType

BookingSettingsSubscriptionStatusRequestType: Anfrage um den Status von abonnierten Buchungseinstellungen abzufragen.

Gruppenzugehoerigkeit: *SubscriptionRequestGroup*

Element	Typ	Kommentar
(leer)		

BookingSettingsSubscriptionStatusResponseType: Antwort um den Status von abonnierten Buchungseinstellungen abzufragen.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *SubscriptionResponseGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <input type="checkbox"/> <i>BookingID</i>	BookingIDType	Abonnierte Buchungen.

CompleteBookingSettingsRequestType: Anfrage um alle abonnierten Buchungseinstellungen abzufragen.

Gruppenzugehoerigkeit: *RequestMessageGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <i>MaxResults</i>	xs:integer	Maximale Anzahl Ergebnisse..

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CompleteBookingSettingsRequestType">
2   <xs:sequence>
3     <xs:element name="MaxResults" type="xs:integer" minOccurs="0"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.76: CompleteBookingSettingsRequestType

CompleteBookingSettingsResponseType: Antwort um alle abonnierten Buchungseinstellungen abzufragen.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *ResponseMessageGroup*

Element	Typ	Kommentar
<i>MessageBlockID</i>	xs:token	Block ID, alle Nachrichten eines Blocks enthalten die gleiche ID.
<input type="checkbox"/> <i>Last</i>	xs:boolean	Falls gesetzt ist dies die letzte Nachricht des Blocks.
<input type="checkbox"/> <input checked="" type="checkbox"/> <i>BookingSettings</i>	BookingSettingsType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CompleteBookingSettingsResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="MessageBlockID" type="xs:token"/>
6         <xs:element name="Last" type="xs:boolean" default="false" minOccurs="0"/>
7         <xs:element name="BookingSettings" type="BookingSettingsType" minOccurs="0" maxOccurs="
  ↳ unbounded"/>
8       </xs:sequence>
9     </xs:extension>
10  </xs:complexContent>
11 </xs:complexType>

```

XML Quellcode 7.77: CompleteBookingSettingsResponseType

BookingSettingsPushMessageType: Pushnachricht mit Buchungseinstellungen.

Gruppenzugehoerigkeit: *PushMessageGroup*

Element	Typ	Kommentar
<input checked="" type="checkbox"/> <i>BookingSettings</i>	BookingSettingsType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="BookingSettingsPushMessageType">
2   <xs:sequence>
3     <xs:element name="BookingSettings" type="BookingSettingsType" maxOccurs="unbounded"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.78: BookingSettingsPushMessageType

7.16 Dienst 11 - Fernkonfiguration Navigationssystem und Fahrtverlauf überwachen

Zielort des Navigationssystems setzen

SetNavigationDestinationRequestType: Anfrage um den Zielort des Navigationssystems zu setzen.

Gruppenzugehoerigkeit: *UserTriggeredRequestGroup*

Element	Typ	Kommentar
<i>BookingID</i>	BookingIDType	Die entsprechende Buchung.
<i>Destination</i>	GeoPositionType	Der Zielort der Buchung.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="SetNavigationDestinationRequestType">
2   <xs:sequence>
3     <xs:element name="BookingID" type="BookingIDType"/>
4     <xs:element name="Destination" type="GeoPositionType"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 7.79: SetNavigationDestinationRequestType

SetNavigationDestinationResponseType: Antwort für setzen des Zielorts.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *UserTriggeredResponseGroup*

Element	Typ	Kommentar
(<i>leer</i>)		

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="SetNavigationDestinationResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType"/>
4   </xs:complexContent>
5 </xs:complexType>

```

XML Quellcode 7.80: SetNavigationDestinationResponseType

Fahrtverlauf (Buchungsfortschritt) überwachen

BookingProgressSubscriptionRequestType: Anfrage um einen Buchungsfortschritt zu abonnieren oder zu stornieren.

Gruppenzugehoerigkeit: *SubscriptionRequestGroup*

Element	Typ	Kommentar
<input checked="" type="checkbox"/> <i>BookingID</i>	BookingIDType	Buchungsreferenz.
<input type="checkbox"/> <i>Unsubscription</i>	xs:boolean	Falls gesetzt stornieren.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="BookingProgressSubscriptionRequestType">
2   <xs:sequence>
3     <xs:element name="BookingID" type="BookingIDType" maxOccurs="unbounded"/>
4     <xs:element name="Unsubscription" type="xs:boolean" default="false" minOccurs="0"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 7.81: BookingProgressSubscriptionRequestType

BookingProgressSubscriptionResponseType: Antwort um einen Buchungsfortschritt zu abonnieren oder zu stornieren.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *SubscriptionResponseGroup*

Element	Typ	Kommentar
<i>(leer)</i>		

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="BookingProgressSubscriptionResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType"/>
4   </xs:complexContent>
5 </xs:complexType>

```

XML Quellcode 7.82: BookingProgressSubscriptionResponseType

BookingProgressSubscriptionStatusRequestType: Anfrage um den Abonnementstatus von Buchungsfortschritten zu erhalten.

Gruppenzugehoerigkeit: *SubscriptionRequestGroup*

Element	Typ	Kommentar
(leer)		

BookingProgressSubscriptionStatusResponseType: Antwort um den Abonnementstatus von Buchungsfortschritten zu erhalten.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *SubscriptionResponseGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <input type="checkbox"/> <i>BookingID</i>	BookingIDType	Abonnierte Buchungen.

CompleteBookingProgressRequestType: Anfrage um alle abonnierten Buchungsfortschritte zu erhalten.

Gruppenzugehoerigkeit: *RequestMessageGroup*

Element	Typ	Kommentar
<input type="checkbox"/> <i>MaxResults</i>	xs:integer	Maximale Anzahl Ergebnisse pro Nachricht.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CompleteBookingProgressRequestType">
2   <xs:sequence>
3     <xs:element name="MaxResults" type="xs:integer" minOccurs="0"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.83: CompleteBookingProgressRequestType

CompleteBookingProgressResponseType: Antwort um alle abonnierten Buchungsfortschritte zu erhalten

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *ResponseMessageGroup*

Element	Typ	Kommentar
<i>MessageBlockID</i>	xs:token	Block ID, alle Nachrichten eines Blocks enthalten die gleiche ID.
<input type="checkbox"/> <i>Last</i>	xs:boolean	Falls gesetzt ist dies die letzte Nachricht des Blocks.
<input type="checkbox"/> <input type="checkbox"/> <i>BookingProgress</i>	BookingProgressType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  → name="CompleteBookingProgressResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="MessageBlockID" type="xs:token"/>
6         <xs:element name="Last" type="xs:boolean" default="false" minOccurs="0"/>
7         <xs:element name="BookingProgress" type="BookingProgressType" minOccurs="0" maxOccurs="
  → unbounded"/>
8       </xs:sequence>
9     </xs:extension>
10  </xs:complexContent>
11 </xs:complexType>

```

XML Quellcode 7.84: CompleteBookingProgressResponseType

BookingProgressPushMessageType: Pushnachricht mit Buchungsfortschritten.
Gruppenzugehoerigkeit: *PushMessageGroup*

Element	Typ	Kommentar
☐ <i>BookingProgress</i>	BookingProgressType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  → name="BookingProgressPushMessageType">
2   <xs:sequence>
3     <xs:element name="BookingProgress" type="BookingProgressType" maxOccurs="unbounded"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.85: BookingProgressPushMessageType

7.17 Dienst 12 - Dialog Intermodale Alternative

Routenvorschlag senden

ItinerarySuggestionsRequestType: Vorschlag einer intermodalen Alternative zur aktuellen Route.

Element	Typ	Kommentar
<i>BookingID</i>	BookingIDType	Buchungsreferenz.
□ <i>CurrentItineraryArrivalTime</i>	xs:dateTime	Erwartete Ankunftszeit der aktuellen Route.
□ <i>DelayExplanation</i>	TextType	Begründung der Verspätung der aktuellen Route.
▣ <i>ItinerarySuggestions</i>	ItinerarySuggestionType	Menge an alternativen intermodalen Routenvorschlägen.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ↳ name="ItinerarySuggestionsRequestType">
2   <xs:sequence>
3     <xs:element name="BookingID" type="BookingIDType"/>
4     <xs:element name="CurrentItineraryArrivalTime" type="xs:dateTime" minOccurs="0"/>
5     <xs:element name="DelayExplanation" type="TextType" minOccurs="0"/>
6     <xs:element name="ItinerarySuggestions" type="ItinerarySuggestionType" maxOccurs="unbounded"/>
7   </xs:sequence>
8 </xs:complexType>

```

XML Quellcode 7.86: ItinerarySuggestionsRequestType

ItinerarySuggestionsResponseType: Antwort auf den Vorschlag einer intermodalen Alternative zur aktuellen Route.

Basistyp: *AbstractBaseResponseType*

Element	Typ	Kommentar
(<i>leer</i>)		

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ↳ name="ItinerarySuggestionsResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType"/>
4   </xs:complexContent>
5 </xs:complexType>

```

XML Quellcode 7.87: ItinerarySuggestionsResponseType

Routenauswahl überwachen

ItinerarySuggestionsSelectionSubscriptionRequestType: Anfrage um Antworten auf einen Routenvorschlag zu abonnieren oder zu stornieren.

Element	Typ	Kommentar
<input checked="" type="checkbox"/> <i>BookingID</i>	BookingIDType	Zu abonnierende BuchungsID.
<input type="checkbox"/> <i>Unsubscription</i>	xs:boolean	Falls gesetzt, Abo stornieren.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ItinerarySuggestionsSelectionSubscriptionRequestType">
2   <xs:sequence>
3     <xs:element name="BookingID" type="BookingIDType" maxOccurs="unbounded"/>
4     <xs:element name="Unsubscription" type="xs:boolean" default="false" minOccurs="0"/>
5   </xs:sequence>
6 </xs:complexType>

```

XML Quellcode 7.88: ItinerarySuggestionsSelectionSubscriptionRequestType

ItinerarySuggestionsSelectionSubscriptionResponseType: Antwort auf Routenvorschlagsauswahl-
abo.

Basistyp: *AbstractBaseResponseType*

Element	Typ	Kommentar
<i>(leer)</i>		

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ItinerarySuggestionsSelectionSubscriptionResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType"/>
4   </xs:complexContent>
5 </xs:complexType>

```

XML Quellcode 7.89: ItinerarySuggestionsSelectionSubscriptionResponseType

ItinerarySuggestionsSelectionSubscriptionStatusRequestType: Anfrage um den Abonnement-
status der Auswahl von Routenvorschlägen zu erhalten.

Element	Typ	Kommentar
<i>(leer)</i>		

ItinerarySuggestionsSelectionSubscriptionStatusResponseType: Antwort um den Abonnement-
status der Auswahl von Routenvorschlägen zu erhalten.

Basistyp: *AbstractBaseResponseType*

Element	Typ	Kommentar
<input type="checkbox"/> <input type="checkbox"/> <i>BookingID</i>	BookingIDType	Abonnierte Buchungen.

CompleteItinerarySuggestionsSelectionRequestType: Anfrage um alle abonnierten Auswahlen von Routenvorschlägen zu erhalten.

Element	Typ	Kommentar
<input type="checkbox"/> <i>MaxResults</i>	xs:integer	Maximale Anzahl Ergebnisse pro Nachricht.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CompleteItinerarySuggestionsSelectionRequestType">
2   <xs:sequence>
3     <xs:element name="MaxResults" type="xs:integer" minOccurs="0"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.90: CompleteItinerarySuggestionsSelectionRequestType

CompleteItinerarySuggestionsSelectionResponseType: Antwort um alle abonnierten Auswahlen von Routenvorschlägen zu erhalten.

Basistyp: *AbstractBaseResponseType*

Element	Typ	Kommentar
<i>MessageBlockID</i>	xs:token	Block ID, alle Nachrichten eines Blocks enthalten die gleiche ID.
<input type="checkbox"/> <i>Last</i>	xs:boolean	Falls gesetzt ist dies die letzte Nachricht des Blocks.
<input type="checkbox"/> <input type="checkbox"/> <i>BookingProgress</i>	BookingProgressType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="CompleteItinerarySuggestionsSelectionResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType">
4       <xs:sequence>
5         <xs:element name="MessageBlockID" type="xs:token"/>
6         <xs:element name="Last" type="xs:boolean" default="false" minOccurs="0"/>
7         <xs:element name="BookingProgress" type="BookingProgressType" minOccurs="0" maxOccurs="
  ↳ unbounded"/>
8       </xs:sequence>
9     </xs:extension>
10  </xs:complexContent>
11 </xs:complexType>

```

XML Quellcode 7.91: CompleteItinerarySuggestionsSelectionResponseType

ItinerarySuggestionsSelectionPushMessageType: Pushnachricht mit dem gewählten Routenvorschlag.

Element	Typ	Kommentar
<i>ItinerarySuggestionsSelection</i>	ItinerarySuggestionsSelectionType	

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ItinerarySuggestionsSelectionPushMessageType">
2   <xs:sequence>
3     <xs:element name="ItinerarySuggestionsSelection" type="ItinerarySuggestionsSelectionType"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.92: ItinerarySuggestionsSelectionPushMessageType

Routenauswahl und Buchung bestätigen

ItinerarySelectionConfirmationRequestType: Bestätigung, dass eine gewählte Route erfolgreich aktualisiert oder gebucht wurde oder Benachrichtigung über mögliche Fehler.

Element	Typ	Kommentar
<i>BookingID</i>	BookingIDType	Buchungsreferenz.
<i>SuggestedItineraryId</i>	ItineraryIDType	ID des zugehörigen Routenvorschlags.
<i>Success</i>	xs:boolean	Zeigt an ob die Aktualisierung und die Buchung der gewählten Route erfolgreich waren.
<input type="checkbox"/> <i>ErrorMessage</i>	TextType	Fehlermeldung im Falle eines Misserfolges.

```

1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ↳ name="ItinerarySelectionConfirmationRequestType">
2   <xs:sequence>
3     <xs:element name="BookingID" type="BookingIDType"/>
4     <xs:element name="SuggestedItineraryId" type="ItineraryIDType"/>
5     <xs:element name="Success" type="xs:boolean"/>
6     <xs:element name="ErrorMessage" type="TextType" minOccurs="0"/>
7   </xs:sequence>
8 </xs:complexType>

```

XML Quellcode 7.93: ItinerarySelectionConfirmationRequestType

ItinerarySelectionConfirmationResponseType: Antwort auf die Routenwahlbestätigung.
Basistyp: *AbstractBaseResponseType*

Element	Typ	Kommentar
<i>(leer)</i>		

```
1 <xs:complexType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ↳ name="ItinerarySelectionConfirmationResponseType">
2   <xs:complexContent>
3     <xs:extension base="AbstractBaseResponseType"/>
4   </xs:complexContent>
5 </xs:complexType>
```

XML Quellcode 7.94: *ItinerarySelectionConfirmationResponseType*

8 Code Tabellen

Die syntaktische Definition für IXSI enthält keine konkreten Werte (Enumerations) für beispielsweise Fahrzeugtypen oder Fehlercodes. Stattdessen werden diese Werte in den nachfolgenden Codetabellen festgelegt. Nur die hier festgelegten Werte dürfen innerhalb von IXSI verwendet werden.

8.1 Fahrzeugklassen

Für die Auswahl *ClassType* können folgende Werte verwendet werden:

Wert	Bedeutung
bike	Fahrrad
motorcycle	Motorrad
micro	Kleinstwagen (z. B. Smart4two)
mini	Kleinwagen (z. B. Opel Corsa)
small	Kompaktwagen (z. B. VW Golf)
medium	Mittelklassewagen (z. B. Audi A4)
large	Oberklassewagen (z. B. BMW 7er)
van	Van (z. B. VW T5 Multivan)
transporter	Transporter (z. B. Ford Transit)

8.2 Antriebsklassen

Für die Auswahl *EngineType* können folgende Werte verwendet werden:

Wert	Bedeutung
none	Kein Kraftantrieb (Muskelkraft)
diesel	Dieselmotor
gasoline	Ottomotor
electric	Elektromotor
liquidgas	Flüssiggas (LPG)
naturalgas	Erdgas (CNG)
hydrogen	Wasserstoffantrieb
hybrid	Hybridantrieb mit Elektro- und Verbrennungsmotor

Hinweis: Ein Pedelec kann als Kombination Fahrzeugklasse `bike` und Antriebsklasse `electric` dargestellt werden.

8.3 Fahrzeugeigenschaften

Attribute und Eigenschaften eines Buchungsziels oder eines Standorts können klassifiziert werden, um sie automatisch interpretieren zu können. Dazu werden Attributsklassen verwendet. Attributsklassen werden in IXSI im Datentyp *AttributeClassType* gespeichert. Folgende Werte sind dabei erlaubt:

Wert	Bedeutung
trailer_hitch	Anhängerkupplung
automatic	Automatikgetriebe
convertible	Cabriolet
air_condition	Klimaanlage
navigation	Navigationssystem
cruise_control	Tempomat
winter_tyres	Winter- bzw. Ganzjahresreifen
child_seat_0	Babyschale
child_seat_1	Kindersitz (9-18kg)
child_seat_4	Kindersitz (15-36kg)
utility	Kombi
doors_4	4/5-Türer
seats_9	Mindestens 9 Sitze
seats_7	Mindestens 7 Sitze
seats_5	Mindestens 5 Sitze
seats_4	Mindestens 4 Sitze

8.4 Verbrauchsdaten

Klassen von Verbrauchsdaten zur Abrechnung von Leistungen, wie z. B. Entfernung, Dauer oder Zusatzleistungen wie ein Kindersitz. Wird in *ConsumptionClassType* gespeichert. Folgende Werte sind dabei erlaubt:

Wert	Bedeutung
distance	Entfernung
duration	Dauer

8.5 Nutzerzustände

Zustände, die ein Benutzer im System haben kann, wird in *UserStateType* gespeichert. Folgende Werte sind dabei erlaubt:

Wert	Bedeutung
operative	Benutzer ist aktiviert und darf Fahrzeuge buchen.
nonoperative	Benutzer ist gesperrt und darf keine Fahrzeuge buchen (kann entsperrt werden).
deleted	Benutzer ist (endgültig) gelöscht.

8.6 Fehlercodes

Das FVS kann alternativ zu den normalen Response-, Update- und Handshake-Nachrichten auch im Fall eines Fehlers Fehlernachrichten an das RIS schicken. Fehlercodes werden im Datentyp *ErrorCodeType* gespeichert.

8 Code Tabellen

Wert	Bedeutung
auth_provider_unknown	Authentifizierung: Unbekannte Provider-ID
auth_invalid_password	Authentifizierung: User-Passwort-Kombination ungültig
auth_invalid_token	Authentifizierung: User-Token-Kombination ungültig
auth_session_invalid	Authentifizierung: Session ist ungültig/abgelaufen
auth_anon_not_allowed	Authentifizierung: Anonymer User nicht erlaubt
auth_not_authorized	Autorisation: Nutzer ist zu dieser Anfrage nicht berechtigt
sys_backend_failed	System: Hintergrundsystem antwortet nicht
sys_unknown_failure	System: unbekannter Fehler
sys_not_implemented	System: Request nicht implementiert
sys_request_not_plausible	System: Request ist nicht plausibel. Dieser Wert sollte stets verwendet werden, wenn inhaltlich Fehler im Request enthalten sind.
booking_target_unknown	Buchungsziel unbekannt
price_info_not_available	Preisinformationen nicht verfügbar
booking_too_short	Buchungsdauer zu kurz
booking_too_long	Buchungsdauer zu lang
booking_target_not_available	Buchungsziel im gegebenen Zeitraum nicht buchbar
booking_change_not_possible	Buchungsänderung kann nicht durchgeführt werden
booking_id_unknown	Unbekannte Buchungs-ID. Dieser Wert sollte auch verwendet werden, wenn die Buchungs-ID einem anderen User zugeordnet ist.
booking_state_change_failed	Änderung des Buchungsstatus fehlgeschlagen.
booking_settings_not_understood	Buchungseinstellung unbekannt.
booking_navigation_set_failed	Setzen des Reiseziels fehlgeschlagen.
language_not_supported	Angefragte Sprache nicht vollständig unterstützt, andere Sprache geliefert.

Quellcodeverzeichnis

5.1	Baisdatentypen	44
5.2	TextType	46
5.3	CoordType	46
5.4	AddressType	47
5.5	GeoPositionType	48
5.6	GeoCircleType	48
5.7	GeoRectangleType	49
5.8	GeoAreaType	49
5.9	IncExcGeoAreaType	50
5.10	TimePeriodType	50
5.11	TimePeriodProposalType	51
5.12	OriginDestType	51
5.13	InfoType	52
5.14	StopLinkType	52
5.15	PlaceOrAreaGroup	53
5.16	DurationGroup	54
5.17	BookingTargetIDType	54
5.18	ProviderPlaceIDType	55
5.19	AttributeType	56
5.20	PlaceType	57
5.21	DensityAreaType	57
5.22	FloatingAreaType	58
5.23	PlaceGroupType	59
5.24	ProviderType	59
5.25	BookingTargetType	61
5.26	BookingType	62
5.27	BookingTargetPropertiesType	63
5.28	BookingTargetAvailabilityType	64
5.29	BookingTargetChangeAvailabilityType	65
5.30	PlaceAvailabilityType	66
5.31	ExternalBookingType	66
5.32	BookingChangeType	67
5.33	TariffDetailType	68
5.34	TariffType	69

5.35	ConsumptionType	69
5.36	UserFeatureType	70
5.37	UserType	71
5.38	BookingSettingsType	72
5.39	BookingProgressType	72
5.40	ItinerarySuggestionType	73
5.41	UserInfoType	74
5.42	AuthType	74
5.43	ErrorType	75
7.1	IxsiMessageType	80
7.2	AbstractBaseResponseType	80
7.3	TransactionType	81
7.4	QueryRequestType	81
7.5	QueryResponseType	82
7.6	SubscriptionRequestType	82
7.7	SubscriptionResponseType	83
7.8	UserTriggeredRequestGroup	85
7.9	UserTriggeredResponseGroup	87
7.10	OriginDestGroup	87
7.11	OpenSessionRequestType	88
7.12	OpenSessionResponseType	89
7.13	CloseSessionRequestType	89
7.14	CloseSessionResponseType	89
7.15	HeartBeatRequestType	90
7.16	HeartBeatResponseType	90
7.17	TokenGenerationRequestType	91
7.18	TokenGenerationResponseType	91
7.19	BookingTargetsInfoRequestType	92
7.20	BookingTargetsInfoResponseType	92
7.21	ChangedProvidersRequestType	93
7.22	ChangedProvidersResponseType	93
7.23	AvailabilityRequestType	94
7.24	AvailabilityRequest Beispiel	95
7.25	AvailabilityResponseType	95
7.26	AvailabilityResponse Beispiel	96
7.27	PlaceAvailabilityRequestType	96
7.28	PlaceAvailabilityResponseType	97
7.29	AvailabilitySubscriptionRequestType	98
7.30	AvailabilitySubscriptionRequest Beispiel	98
7.31	AvailabilitySubscriptionResponseType	99
7.32	AvailabilityPushMessageType	99
7.33	AvailabilityPush Beispiel	100

7.34	CompleteAvailabilityRequestType	100
7.35	CompleteAvailabilityRequest Beispiel	100
7.36	CompleteAvailabilityResponseType	101
7.37	CompleteAvailabilityResponse Beispiel	101
7.38	PlaceAvailabilitySubscriptionRequestType	102
7.39	PlaceAvailabilitySubscriptionResponseType	103
7.40	PlaceAvailabilityPushMessageType	104
7.41	CompletePlaceAvailabilityRequestType	104
7.42	CompletePlaceAvailabilityResponseType	105
7.43	BookingRequestType	106
7.44	BookingRequest Beispiel	106
7.45	BookingResponseType	106
7.46	BookingResponse Beispiel	107
7.47	ChangeBookingRequestType	107
7.48	ChangeBookingResponseType	108
7.49	ExternalBookingSubscriptionRequestType	109
7.50	ExternalBookingSubscriptionResponseType	109
7.51	ExternalBookingPushMessageType	110
7.52	CompleteExternalBookingRequestType	110
7.53	CompleteExternalBookingResponseType	111
7.54	BookingAlertSubscriptionRequestType	112
7.55	BookingAlertSubscriptionResponseType	112
7.56	BookingAlertPushMessageType	113
7.57	CompleteBookingAlertRequestType	114
7.58	CompleteBookingAlertResponseType	114
7.59	PriceInformationRequestType	115
7.60	PriceInformationResponseType	115
7.61	ConsumptionSubscriptionRequestType	116
7.62	ConsumptionSubscriptionResponseType	116
7.63	ConsumptionPushMessageType	117
7.64	CompleteConsumptionRequestType	118
7.65	CompleteConsumptionResponseType	118
7.66	ChangeBookingStateRequestType	119
7.67	ChangeBookingStateResponseType	120
7.68	CreateUserRequestType	120
7.69	CreateUserResponseType	121
7.70	ChangeUserRequestType	121
7.71	ChangeUserResponseType	122
7.72	SetBookingSettingsRequestType	122
7.73	SetBookingSettingsResponseType	123
7.74	BookingSettingsSubscriptionRequestType	123
7.75	BookingSettingsSubscriptionResponseType	124
7.76	CompleteBookingSettingsRequestType	124

7.77	CompleteBookingSettingsResponseType	125
7.78	BookingSettingsPushMessageType	125
7.79	SetNavigationDestinationRequestType	126
7.80	SetNavigationDestinationResponseType	126
7.81	BookingProgressSubscriptionRequestType	127
7.82	BookingProgressSubscriptionResponseType	127
7.83	CompleteBookingProgressRequestType	128
7.84	CompleteBookingProgressResponseType	129
7.85	BookingProgressPushMessageType	129
7.86	ItinerarySuggestionsRequestType	130
7.87	ItinerarySuggestionsResponseType	130
7.88	ItinerarySuggestionsSelectionSubscriptionRequestType	131
7.89	ItinerarySuggestionsSelectionSubscriptionResponseType	131
7.90	CompleteItinerarySuggestionsSelectionRequestType	132
7.91	CompleteItinerarySuggestionsSelectionResponseType	132
7.92	ItinerarySuggestionsSelectionPushMessageType	133
7.93	ItinerarySelectionConfirmationRequestType	133
7.94	ItinerarySelectionConfirmationResponseType	134

Stichwortverzeichnis

- AbstractBaseResponseType, 78, 86–91, 93, 95–97, 99, 101, 102, 104, 106–126, 128–130, 132
- Address, 45
- AddressType, 45
- Anmeldung, 21
- Annotation, 67
- Anonymous, 72
- Area, 55, 56
- AreaID, 59
- AreaIDType, 42, 56, 59
- AttributeClassType, 42, 44, 50, 54, 134
- AttributeID, 51, 57, 59
- AttributeIDType, 42, 51, 54, 57, 59
- Attributes, 90, 117
- AttributeType, 53, 54, 90, 117
- Auth, 82
- AuthType, 72, 82
- Availability, 63, 64, 82, 84
- AvailabilityChange, 97
- AvailabilityPushMessage, 27
- AvailabilityPushMessageType, 97
- AvailabilityRequestType, 82, 92
- AvailabilityResponseType, 84, 93
- AvailabilitySubscriptionRequest, 27
- AvailabilitySubscriptionRequestType, 95, 96
- AvailabilitySubscriptionResponseType, 96, 97
- AvailabilitySubscriptionStatusRequestType, 97
- AvailabilitySubscriptionStatusResponseType, 97
- Begin, 48
- Benutzer
 - FVS, 8
 - RIS, 8
- Betreiber
 - FVS, 9
 - RIS, 9
- Bookee, 90
- BookeeID, 52
- BookeeIDType, 42, 52
- Booking, 29, 82, 84, 104, 106
- BookingTargetsInfo, 25
- BookingAlertPushMessageType, 111
- BookingAlertSubscriptionRequestType, 109, 110
- BookingAlertSubscriptionResponseType, 110
- BookingAlertSubscriptionStatusRequestType, 110
- BookingAlertSubscriptionStatusResponseType, 111
- BookingChange, 111, 112
- BookingChangeType, 65, 111, 112
- BookingGrid, 59
- BookingHorizon, 59
- BookingID, 64, 65, 67, 69, 70, 105, 110, 111, 114, 115, 117, 121, 122, 124–126, 128–131
- BookingIDType, 42, 60, 64, 65, 67, 69, 70, 105, 110, 111, 114, 115, 117, 121, 122, 124–126, 128–131
- BookingProgress, 126, 127, 130
- BookingProgressPushMessageType, 127
- BookingProgressSubscriptionRequestType, 125
- BookingProgressSubscriptionResponseType, 125

- 125
- BookingProgressSubscriptionStatusRequestType, 125
- BookingProgressSubscriptionStatusResponseType, 126
- BookingProgressType, 70, 126, 127, 130
- BookingRequestType, 82, 103
- BookingResponseType, 84, 104
- BookingSettings, 120, 123
- BookingSettingsClassType, 42, 69
- BookingSettingsPushMessageType, 123
- BookingSettingsSubscriptionRequestType, 121
- BookingSettingsSubscriptionResponseType, 121
- BookingSettingsSubscriptionStatusRequestType, 122
- BookingSettingsSubscriptionStatusResponseType, 122
- BookingSettingsType, 69, 70, 120, 123
- BookingState, 117
- BookingStateType, 42, 117
- BookingTarget, 92, 93, 99
- BookingTargetAvailabilityType, 61, 62, 93, 99
- BookingTargetChangeAvailabilityType, 63, 97
- BookingTargetID, 64, 96, 97, 103, 113
- BookingTargetIDType, 52, 59, 61–64, 96, 97, 103, 113
- BookingTargetPropertiesType, 61, 92
- BookingTargetsInfoRequestType, 89, 90
- BookingTargetsInfoResponseType, 90
- BookingTargetSpecific, 69
- BookingTargetType, 58, 59, 90
- BookingType, 60, 104, 106
- CalcTime, 79, 81
- Cancel, 105
- Cancelled, 65
- Capacity, 55
- Center, 46
- ChangeBooking, 29, 82, 84
- ChangeBookingRequestType, 82, 105
- ChangeBookingResponseType, 84, 106
- ChangeBookingState, 82, 84
- ChangeBookingStateRequestType, 82, 117
- ChangeBookingStateResponseType, 84, 117
- ChangedProviders, 25
- ChangedProvidersRequestType, 91
- ChangedProvidersResponseType, 91
- ChangeUser, 82, 84
- ChangeUserRequestType, 82, 119
- ChangeUserResponseType, 84, 119
- Circle, 92, 94
- City, 45
- Class, 50, 54, 59, 68
- ClassType, 42, 44, 59, 133
- CloseSession, 82, 84
- CloseSessionRequestType, 82, 87
- CloseSessionResponseType, 84, 87
- CO2Factor, 59
- Code, 73
- CompleteAvailabilityRequest, 28
- CompleteAvailabilityRequestType, 98
- CompleteAvailabilityResponseType, 99
- CompleteBookingAlertRequestType, 111
- CompleteBookingAlertResponseType, 112
- CompleteBookingProgressRequestType, 126
- CompleteBookingProgressResponseType, 126, 127
- CompleteBookingSettingsRequestType, 122
- CompleteBookingSettingsResponseType, 123
- CompleteConsumptionRequestType, 116
- CompleteConsumptionResponseType, 116
- CompleteExternalBookingRequestType, 108
- CompleteExternalBookingResponseType, 109
- CompleteItinerarySuggestionsSelectionRequestType, 130
- CompleteItinerarySuggestionsSelectionResponseType, 130

- 130
 CompletePlaceAvailabilityRequestType, 102
 CompletePlaceAvailabilityResponseType, 102, 103
 Consumption, 115, 116
 ConsumptionClassType, 42, 67, 135
 ConsumptionPushMessageType, 115
 ConsumptionSubscriptionRequestType, 114
 ConsumptionSubscriptionResponseType, 114
 ConsumptionSubscriptionStatusRequestType, 115
 ConsumptionSubscriptionStatusResponseType, 115
 ConsumptionType, 67, 115, 116
 Coord, 45, 55
 CoordType, 44–47, 49, 70
 Country, 45
 CreateUser, 82, 84
 CreateUserRequestType, 82, 118
 CreateUserResponseType, 84, 118, 119
 CurrentDrivingRange, 62
 CurrentItineraryArrivalTime, 128
 CurrentStateOfCharge, 62
 CustomerChoice, 57
- DelayExplanation, 128
 DensityAreaType, 55, 56
 Description, 51, 66, 67
 Dest, 85
 Destination, 124
 Detail, 66
 Distance, 50, 61, 113
 Duration, 52
 DurationGroup, 50–52, 55, 56
- End, 48
 Engine, 59
 EngineType, 42, 44, 59, 133
 Error, 78
 ErrorCodeType, 42, 44, 73, 135
 ErrorMessage, 131
 ErrorType, 73, 78
 ETA, 70
 EuroPriceType, 42, 66, 67, 71
 EventHorizon, 96
 Exclude, 47
 ExpectedArrivalTime, 71
 externalBooking, 108, 109
 ExternalBookingPushMessageType, 108
 ExternalBookingSubscriptionRequestType, 106
 ExternalBookingSubscriptionResponseType, 107
 ExternalBookingSubscriptionStatusRequestType, 107
 ExternalBookingSubscriptionStatusResponseType, 107
 ExternalBookingType, 64, 108, 109
- Fahrzeugverleihsystem, 8
 Final, 67
 FloatingAreaType, 56, 90
 FreeFloatingArea, 90
 FVS, 8, 21
- GeoAreaType, 47, 55
 GeoCircleType, 46, 92, 94
 GeoPosition, 49, 62, 63
 GeoPositionType, 45, 46, 55, 62, 63, 124
 GeoRectangle, 92, 94
 GeoRectangleType, 46, 92, 94
 GlobalID, 55, 59
 GlobalIDType, 42, 55, 59
- HeartBeatRequestType, 88
 HeartBeatResponseType, 88
- ID, 54–57, 59–64, 69
 Importance, 50, 54
 Impossible, 65
 Inavailability, 62, 63
 IncExcGeoAreaType, 47, 56
 Info, 60, 66, 69
 InfoType, 49, 50, 60, 66

Stichwortverzeichnis

ItineraryID, 71
ItineraryIDType, 42, 71, 131
ItinerarySelectionConfirmationRequestType, 131
ItinerarySelectionConfirmationResponseType, 132
ItinerarySuggestions, 128
ItinerarySuggestionsRequestType, 127, 128
ItinerarySuggestionsResponseType, 128
ItinerarySuggestionsSelection, 131
ItinerarySuggestionsSelectionPushMessageType, 131
ItinerarySuggestionsSelectionSubscriptionRequestType, 128, 129
ItinerarySuggestionsSelectionSubscriptionResponseType, 129
ItinerarySuggestionsSelectionSubscriptionStatusRequestType, 129
ItinerarySuggestionsSelectionSubscriptionStatusResponseType, 129
ItinerarySuggestionsSelectionType, 131
ItinerarySuggestionType, 71, 128
IxsiMessageType, 77, 78

Language, 44, 82
Last, 99, 103, 109, 112, 116, 123, 126, 130
Latitude, 44
LongDescription, 71
Longitude, 44
LowerRight, 46

Mandatory, 50, 54
MaxDistance, 59
MaxPlaces, 102
MaxResults, 108, 111, 116, 122, 126, 130
MaxTargets, 98
MaxWait, 48
MessageBlockID, 99, 103, 109, 112, 116, 123, 126, 130
MessageID, 78

Name, 51, 57, 59, 66
NewPeriod, 65
NewTimePeriodProposal, 105

NextDestination, 71
NonFatal, 73
Notification, 65

OnPremisesTime, 55
OpeningTime, 59
OpenSession, 32, 82, 84
OpenSessionRequestType, 82, 86
OpenSessionResponseType, 84, 86, 87
Origin, 85
OriginDestGroup, 85, 103, 113
OriginDestType, 49, 71, 85

Password, 72
PercentType, 42, 50, 54, 57, 62, 70
Place, 90, 95
PlaceAvailability, 82, 84, 102, 103
PlaceAvailabilityPushMessageType, 101, 102
PlaceAvailabilityRequestType, 82, 94
PlaceAvailabilityResponseType, 84, 95
PlaceAvailabilitySubscriptionRequestType, 100
PlaceAvailabilitySubscriptionResponseType, 101
PlaceAvailabilitySubscriptionStatusRequestType, 101
PlaceAvailabilitySubscriptionStatusResponseType, 101
PlaceAvailabilityType, 63, 64, 95, 102, 103
PlaceGroup, 90
PlaceGroupID, 59
PlaceGroupIDType, 42, 57, 59
PlaceGroupType, 56, 57, 90
PlaceID, 49, 53, 57, 59, 62, 63, 94, 100, 101
PlaceIDType, 42, 49, 53, 55, 59, 62, 63
PlaceOrAreaGroup, 51, 54, 56
PlaceType, 54, 55, 90
PolyPoint, 47
Position, 70
PostalCode, 45
Price, 66, 67, 71
PriceInformation, 82, 84

- PriceInformationRequest, 32
- PriceInformationRequestType, 82, 113
- PriceInformationResponseType, 84, 113
- Probability, 57
- ProbabilityPlaceIDType, 57
- Progress, 70
- Provider, 90, 91
- ProviderFilter, 90
- ProviderID, 51–53, 72
- ProviderIDType, 42, 51–53, 57, 72, 90, 91
- ProviderPlaceIDType, 53, 64, 94, 100, 101
- ProviderType, 57, 90
- PushMessageGroup, 86, 97, 101, 108, 111, 115, 123, 127

- QueryRequestType, 77, 79
- QueryResponseType, 77, 79, 80

- Radius, 46
- Reason, 65
- Reiseinformationssystem, 7
- Repossiblized, 65
- Request, 77
- RequestMessageGroup, 80, 86, 98, 102, 108, 111, 116, 122, 126
- Response, 77
- ResponseMessageGroup, 80, 86, 99, 102, 108, 112, 116, 123, 126
- RIS, 7, 21

- Separate, 54
- SessionID, 72, 84
- SessionIDType, 42, 72, 84
- SessionTimeout, 72, 84
- SetBookingSettings, 82, 84
- SetBookingSettingsRequestType, 82, 120
- SetBookingSettingsResponseType, 84, 120
- SetNavigationDestination, 82, 84
- SetNavigationDestinationRequestType, 82, 124
- SetNavigationDestinationResponseType, 84, 124
- SeverityFactor, 61
- ShortDescription, 71
- ShortName, 57
- simpleType, 44
- State, 69
- StaticDataRequestGroup, 79, 86, 89, 91
- StaticDataResponseGroup, 79, 86, 90, 91
- StopID, 50
- StopLink, 51
- StopLinkType, 50, 51
- StreetHouseNr, 45
- SubArea, 56
- SubscriptionAdministrationRequestGroup, 80, 86, 88
- SubscriptionAdministrationResponseGroup, 80, 86, 88
- SubscriptionMessage, 77
- SubscriptionMessageType, 77
- SubscriptionRequest, 77
- SubscriptionRequestGroup, 80, 86, 95, 97, 100, 101, 106, 107, 109, 110, 114, 115, 121, 122, 125
- SubscriptionRequestType, 77, 80
- SubscriptionResponse, 77
- SubscriptionResponseGroup, 80, 86, 96, 97, 101, 107, 110, 111, 114, 115, 121, 122, 125, 126
- SubscriptionResponseType, 77, 80, 81
- Success, 131
- SuggestedItineraryId, 131
- SystemID, 79, 80
- SystemIDType, 42, 79, 80
- SystemMessage, 73

- Tariff, 113
- TariffDetailType, 66
- TariffType, 66, 113
- Text, 44, 50, 54
- TextType, 44, 51, 54, 59, 65, 67, 71, 128, 131
- TimePeriod, 60, 64, 67, 92
- TimePeriodProposal, 103, 113
- TimePeriodProposalType, 48, 49, 103, 105, 113
- TimePeriodType, 48, 60, 62–65, 67, 92

- TimeStamp, 78
- Timestamp, 70, 90, 91
- timestamp, 25
- Token, 72, 89
- TokenGeneration, 82, 84
- TokenGenerationRequestType, 82, 88, 89
- TokenGenerationResponseType, 84, 89
- Transaction, 79–81
- TransactionType, 78–81
- Type, 67, 69

- Unit, 67
- Unsubscription, 96, 100, 106, 110, 114, 121, 125, 129
- UpperLeft, 46
- URL, 50, 54
- User, 118, 119
- UserFeatureClassType, 42, 68
- UserFeatureType, 68, 69
- UserID, 72
- UserIDType, 42, 72
- UserInfo, 64, 72, 106, 108
- UserInfoType, 64, 69, 71, 72, 106, 108
- UserMessage, 73
- UserStateType, 42, 69, 135
- UserTriggeredRequestGroup, 79, 81, 83, 86–88, 92, 94, 103, 105, 113, 117–120, 124

- UserTriggeredResponseGroup, 79, 83, 85–87, 89, 93, 95, 104, 106, 113, 117–120, 124
- UserType, 69, 118, 119

- Value, 67–69
- Variance, 52

- WithText, 50, 54

- xs:anyURI, 50, 54
- xs:boolean, 47, 50, 54, 57, 65, 67, 69, 72, 73, 96, 99, 100, 103, 105, 106, 109, 110, 112, 114, 116, 121, 123, 125, 126, 129–131
- xs:dateTime, 48, 70, 71, 78, 90, 91, 128
- xs:decimal, 67, 69
- xs:duration, 48, 52, 55, 59, 72, 79, 81, 84, 96
- xs:integer, 98, 102, 108, 111, 116, 122, 126, 130
- xs:language, 44, 82
- xs:nonNegativeInteger, 42, 46, 50, 55, 59, 61, 62, 64, 78, 113
- xs:normalizedString, 57, 72
- xs:string, 44, 45, 50, 66–69, 73
- xs:token, 42, 50, 72, 89, 99, 103, 109, 112, 116, 123, 126, 130