

IXSI - Interface for X-Sharing Information Version 4

Autoren:

Markus Beutel, Sevket Gökay, Peter von Grumbkow, Dirk Hillbrecht, Karl-Heinz Krempels, Christian Samsel, Christoph Terwelp, Heike Twele, Gerhard Wagner

Datum: 31. Januar 2017 Dokumentversion: 4.7d

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 aus dem GitHub Repository (<https://github.com/RWTH-i5-IDSG/ixsi/>) heruntergeladen werden.

Gefördert durch:



Bundesministerium
für Wirtschaft
und Technologie



aufgrund eines Beschlusses
des Deutschen Bundestages

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Zusammenfassung | 5 |
| 2 | Rollenmodell | 7 |
| 3 | Hierarchiemodell | 11 |
| 3.1 | Basisdienst A – Sitzungen (session handling) | 11 |
| 3.2 | Basisdienst B – Abonnements (subscription handling) | 12 |
| 3.3 | Basisdienst C – tokens | 12 |
| 3.4 | Dienst 1 – Statische Daten (static data) | 12 |
| 3.5 | Dienst 2 – Verfügbarkeitsauskunft (availability query) | 13 |
| 3.6 | Dienst 3 – Verfügbarkeitsabonnement (availability subscription) | 13 |
| 3.7 | Dienst 4 – Buchung (booking) | 14 |
| 3.8 | Dienst 5 – Buchungsabonnement (booking subscription) | 15 |
| 3.9 | Dienst 6 – Preisauskunft (price information) | 15 |
| 4 | Interaktionsprotokolle | 17 |
| 4.1 | Überblick | 17 |
| 4.2 | Dienst 1 – Statische Daten | 20 |
| 4.3 | Dienst 2 – Verfügbarkeitsauskunft | 21 |
| 4.4 | Dienst 3 – Verfügbarkeitsabonnement | 22 |
| 4.5 | Dienst 4 – Buchung / Buchungsänderung | 24 |
| 4.6 | Dienst 5 – Buchungsabonnement | 25 |
| 4.7 | Dienst 6 – Preisauskunft | 26 |
| 5 | Datenmodell | 27 |
| 5.1 | Basisdatentypen | 27 |
| 5.2 | Basisgruppen | 36 |
| 5.3 | Dienst 1 - Statische Daten | 38 |
| 5.4 | Dienst 2 - Verfügbarkeitsauskunft | 46 |
| 5.5 | Dienst 4 - Buchung | 48 |
| 5.6 | Dienst 5 - Buchungsabonnement | 49 |
| 5.7 | Dienst 6 - Preisauskunft | 51 |
| 5.8 | Authentifizierung | 52 |
| 5.9 | Fehlerbehandlung | 53 |

| | | |
|----------|---|-----------|
| 6 | Technische Realisierung | 55 |
| 6.1 | Nachrichtenkodierung | 55 |
| 6.2 | Kommunikationskanal | 55 |
| 6.3 | Authentifizierung | 55 |
| 6.4 | Verbindungssicherheit | 56 |
| 7 | Nachrichten | 57 |
| 7.1 | Basisnachrichten | 57 |
| 7.2 | Basisgruppen | 61 |
| 7.3 | Dienst A - Sitzungen | 65 |
| 7.4 | Dienst B - Abonnements | 67 |
| 7.5 | Dienst C - Tokens | 67 |
| 7.6 | Dienst 1 - Statische Daten | 68 |
| 7.7 | Dienst 2 - Verfügbarkeitsauskunft | 71 |
| 7.8 | Dienst 3 - Verfügbarkeitsabonnement | 74 |
| 7.9 | Dienst 4 - Buchung / Buchungsänderung | 82 |
| 7.10 | Dienst 5 - Buchungsabonnement | 85 |
| 7.11 | Dienst 6 - Preisauskunft | 88 |
| 8 | Code Tabellen | 91 |
| 8.1 | Fahrzeugklassen | 91 |
| 8.2 | Antriebsklassen | 91 |
| 8.3 | Fahrzeugeigenschaften | 92 |
| 8.4 | Fehlercodes | 93 |
| | Quellcodeverzeichnis | 95 |
| | Stichwortverzeichnis | 99 |

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.

2 Rollenmodell

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

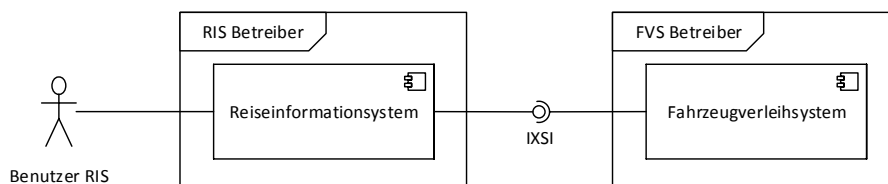


Abbildung 2.1: Überblick Rollenmodell.

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.

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.

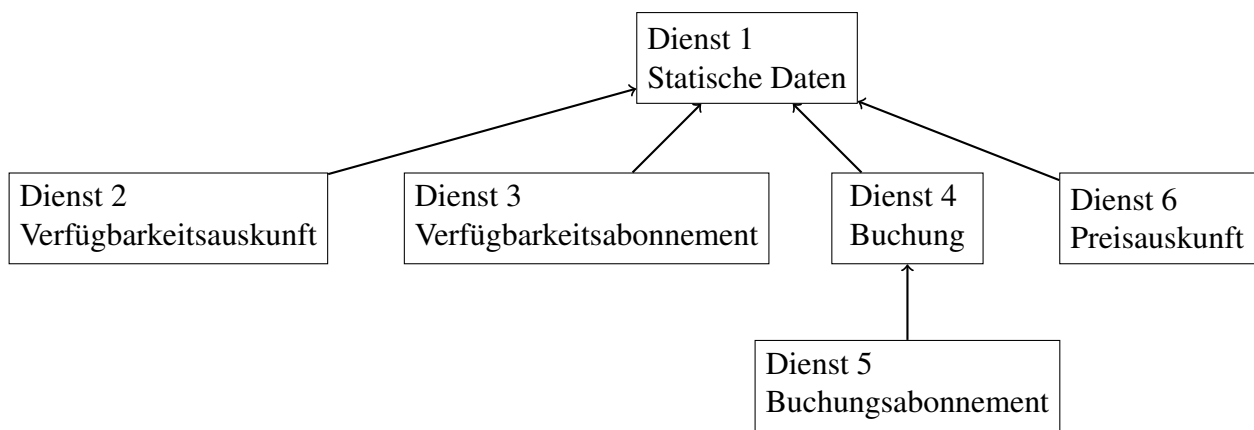


Abbildung 3.1: Dienstabhängigkeiten

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 2 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

- Authentifizierung von Nutzern gegenüber dem FVS.
- Anfrage zur Buchung
- Anfrage zur Buchungsänderung / Stornierung

Abhängigkeiten

- Dienst 1

3.8 Dienst 5 – Buchungsabonnement (booking subscription)

Dienst 5 dient dem Abonnament 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

4 Interaktionsprotokolle

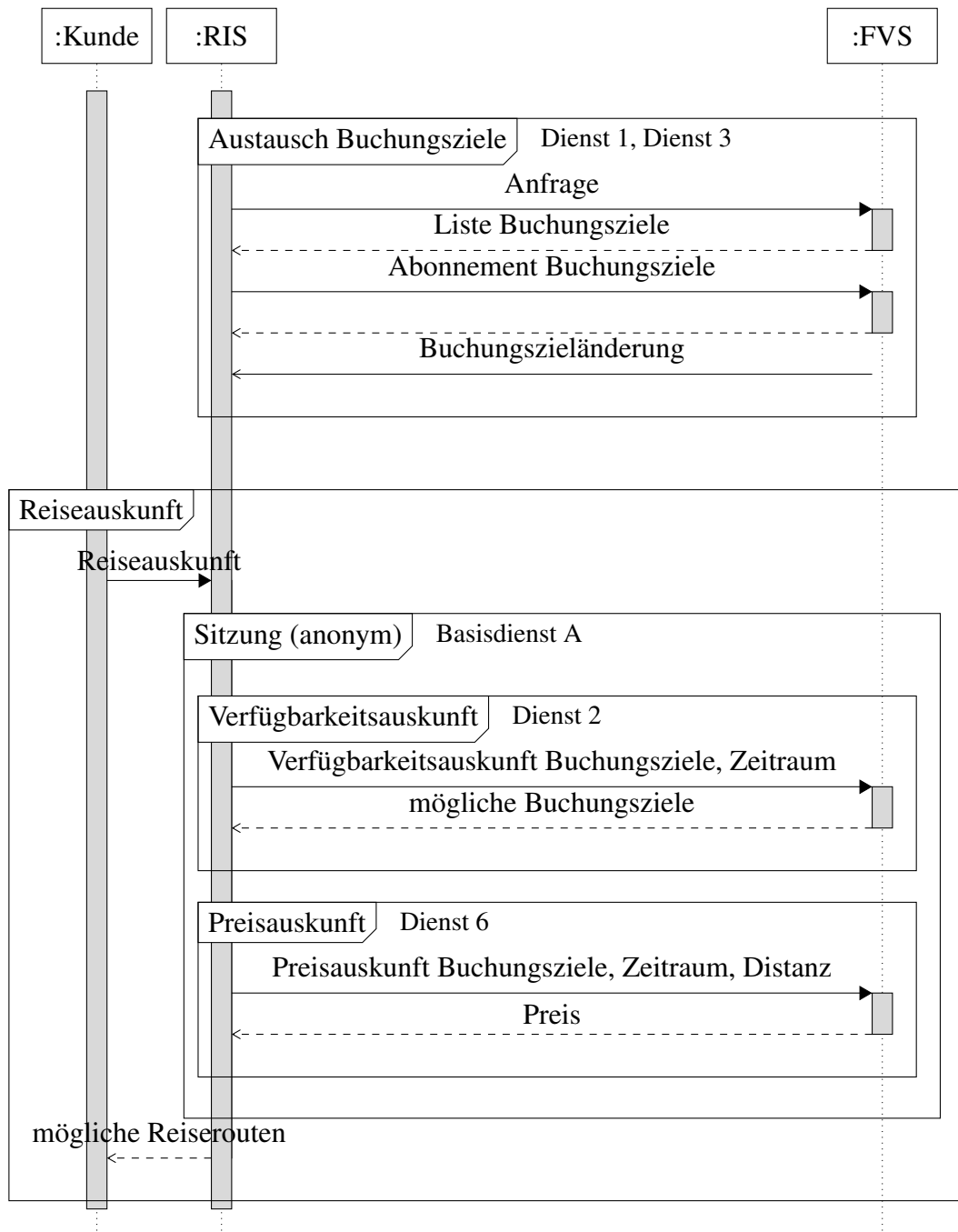
Dieser Abschnitt gibt einen Überblick über die in IXSI verwendeten Interaktionsschemata. Zur Vereinfachung werden die Interaktionsschemata in den Sequenzdiagrammen informell ohne Verwendung der technischen Bezeichnungen der Funktionsaufrufe beschrieben. Grundsätzlich werden zwei Typen von Interaktionen 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.

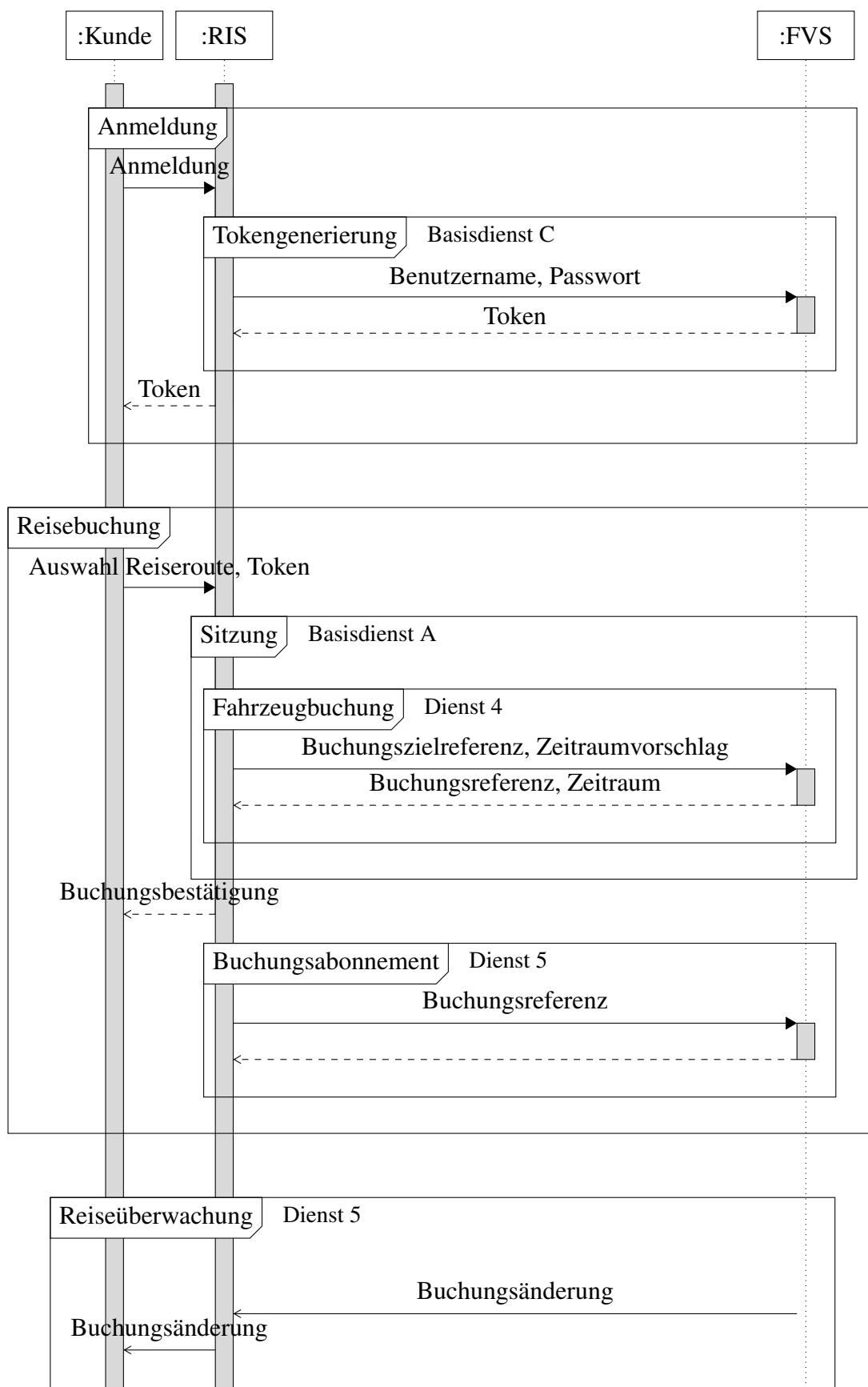
4.1 Überblick

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

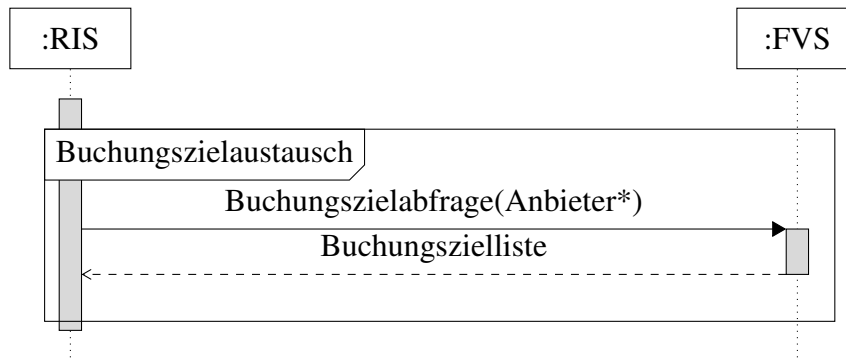
an das FVS. Das sendet eine Buchungsbestätigung, welche vom RIS an den Kunden weitergegeben wird. Zusätzlich abonniert das RIS die entsprechende Buchung beim FVS. Im letzten Block *Reiseüberwachung* wird der Kunde über Änderungen an der Buchung, die das RIS vom FVS erhält, benachrichtigt.





4.2 Dienst 1 – Statische Daten

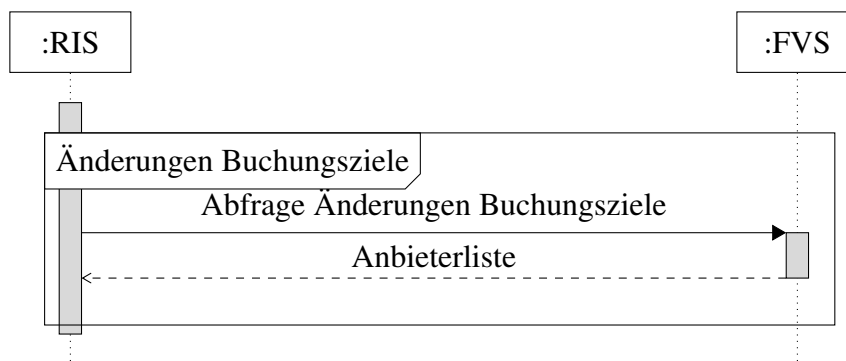
Abfrage Buchungsziele



** 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.

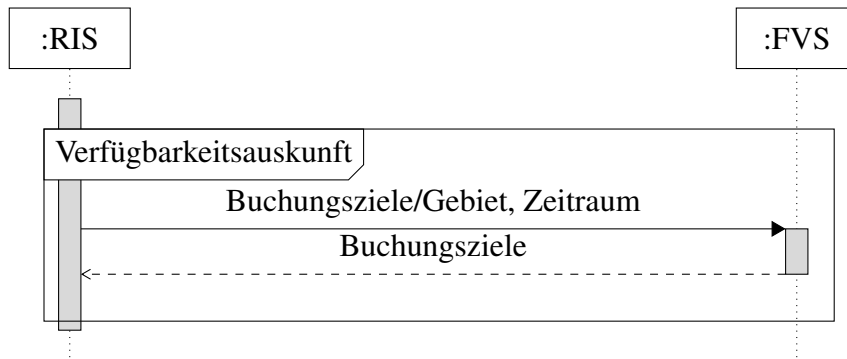
Abfrage Änderungen Buchungsziele



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.

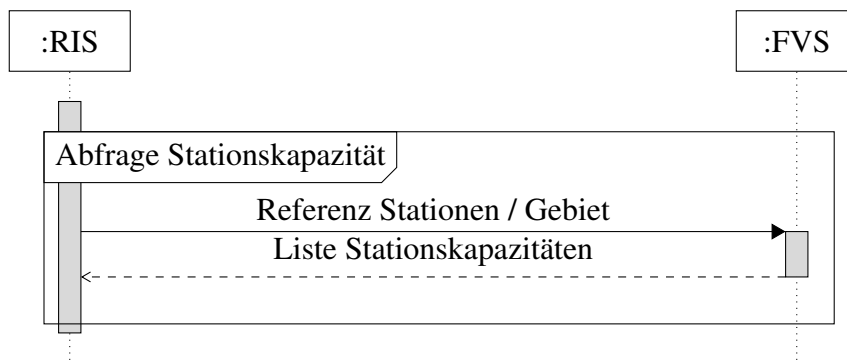
4.3 Dienst 2 – Verfügbarkeitsauskunft

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.

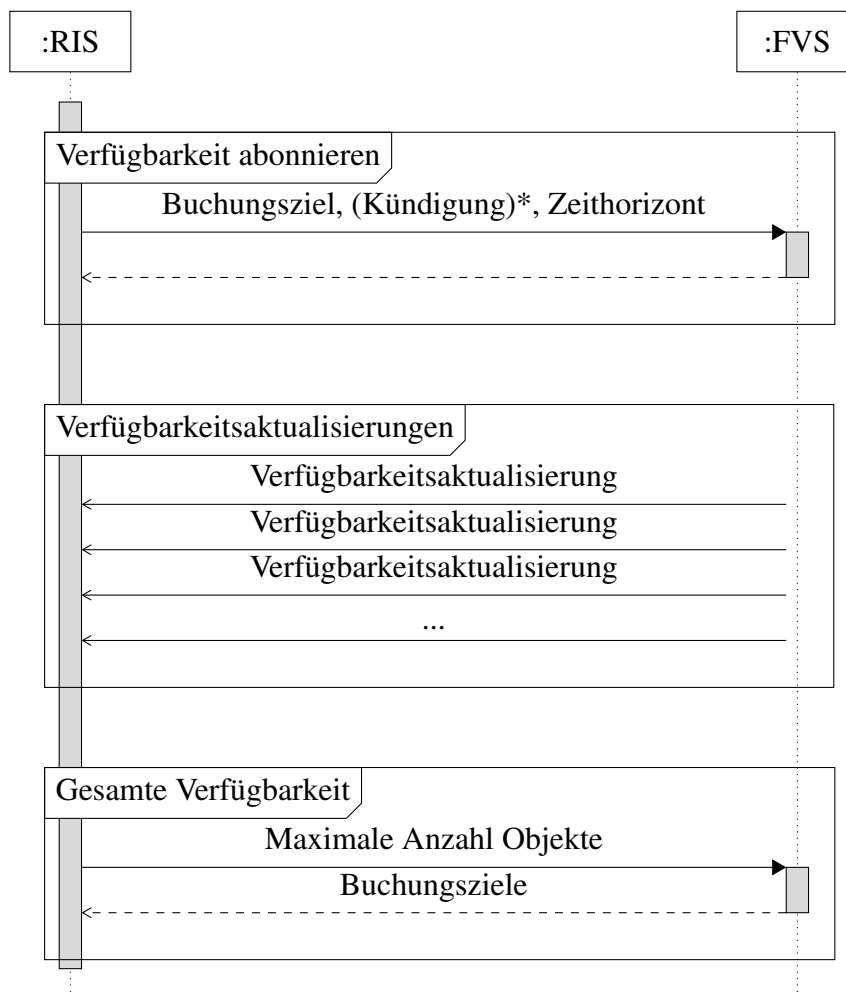
Abfrage aktuelle Stationskapazität (Dienst 2a)



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

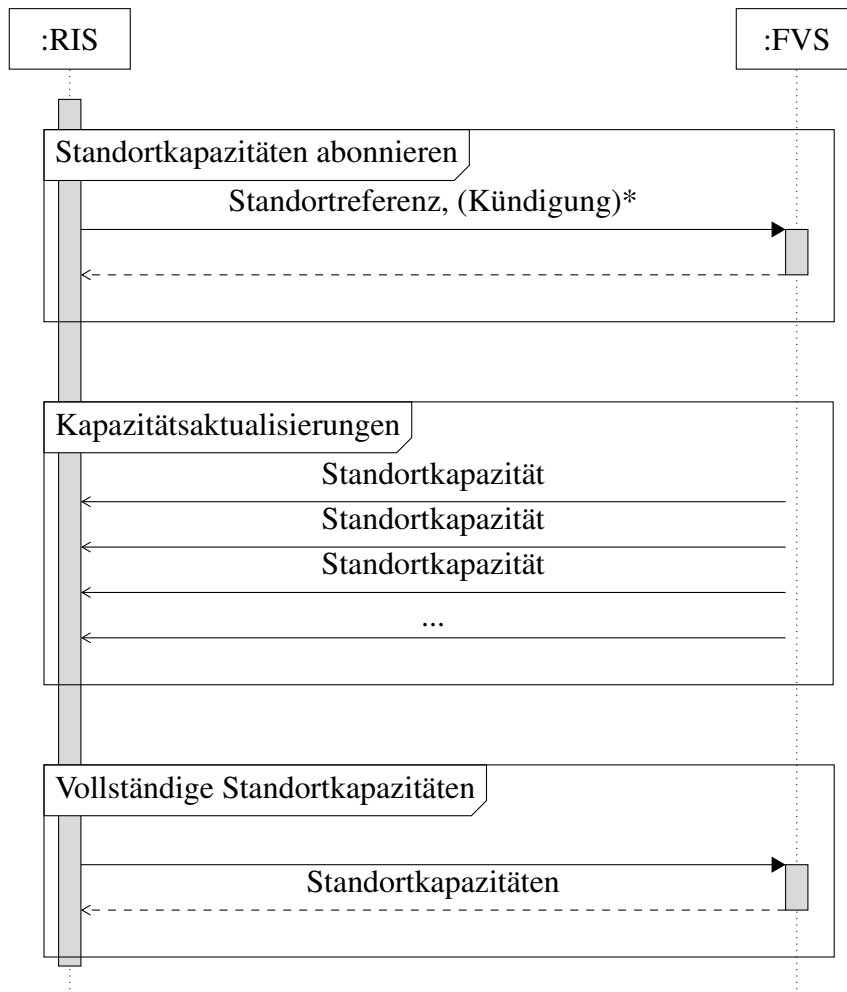


Das RIS kann Informationen zu Buchungszielen abonnieren, um unmittelbar über Änderungen von Verfügbarkeiten informiert zu werden. Dies dient im Wesentlichen dazu, Reiseauskünfte ohne zusätzliche (synchrone) Anfrage an das FVS beantworten zu können.

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.

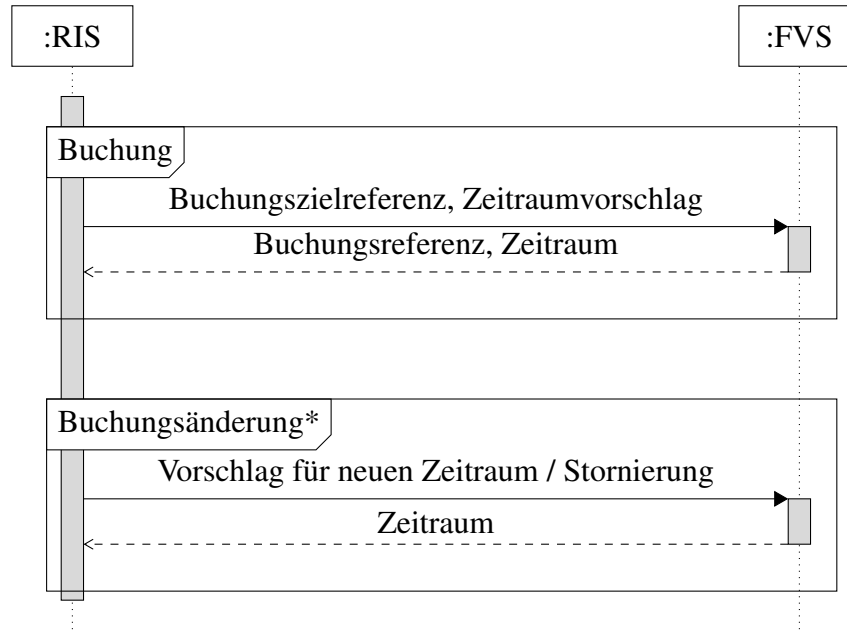
Zur anfänglichen Synchronisierung aller Verfügbarkeiten kann das RIS die Funktion Complete-AvailabilityRequest aufrufen.

Standortkapazitätsabonnement (Dienst 3a)



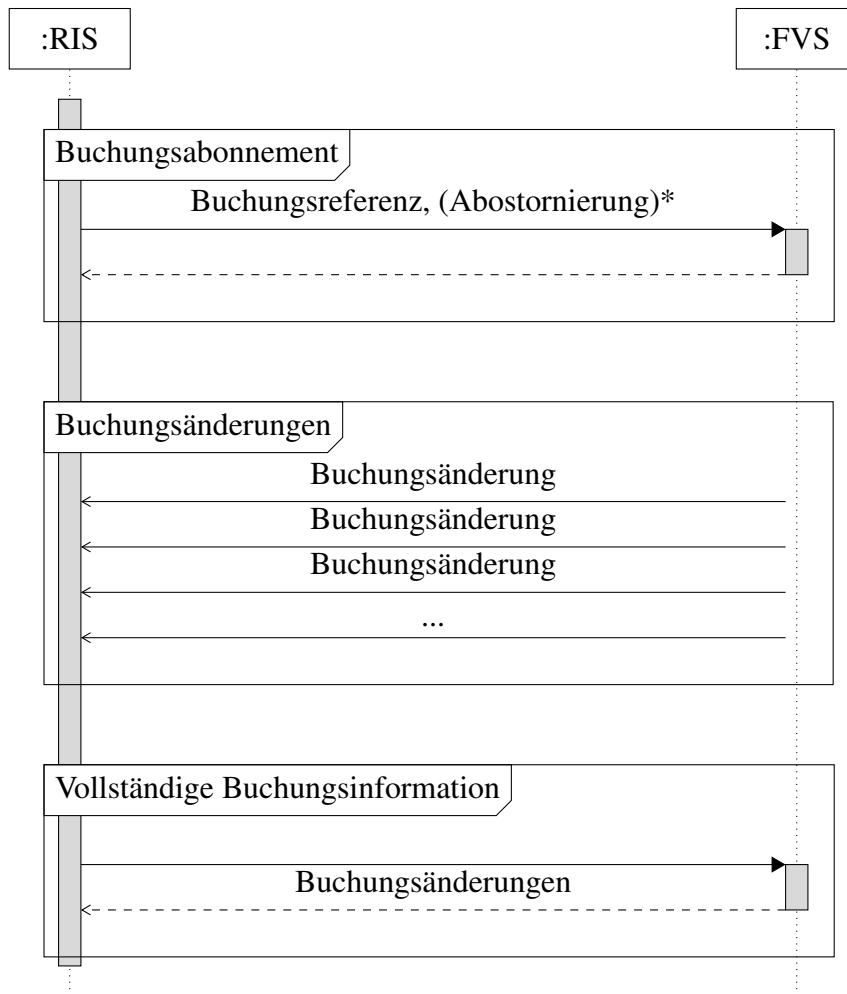
Das RIS kann die Kapazitätsinformation von Standorten abonnieren. Der Interaktionsablauf ist analog zu Abschnitt 4.4.

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 5.8 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.

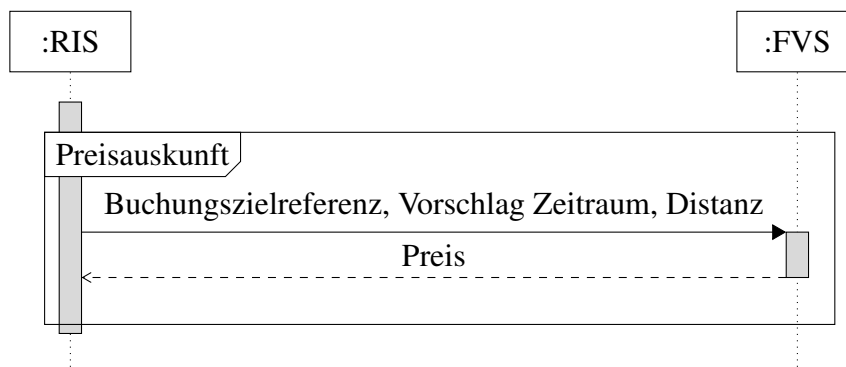
4.6 Dienst 5 – Buchungsabonnement



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 Buchungziel 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.

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>ClassType</i> | xs:token | Aufzählung von Autoklassen. |
| <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>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>UserIDType</i> | xs:token | Typ für User-IDs. |

```

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="UserIDType">
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="SystemIDType">
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="GlobalIDType">
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="EuroPriceType">
22  <xs:restriction base="xs:nonNegativeInteger"/>

```

```

23 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="PercentType">
24   <xs:restriction base="xs:nonNegativeInteger">
25     <xs:maxInclusive value="100"/>
26   </xs:restriction>
27 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="ErrorCodeType">
28   <xs:restriction base="xs:token"/>
29 </xs:simpleType><xs:simpleType xmlns="http://www.ixsi-schnittstelle.de/" xmlns:xs="http://www.w3.org
    ↳ /2001/XMLSchema" name="ClassType">
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="EngineType">
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="AttributeClassType">
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="AttributeIDType">
36   <xs:restriction base="xs:token"/>
37 </xs:simpleType>

```

XML Quellcode 5.1: Basisdatentypen

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-Identifizier 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 Kreiscentrums. |
| <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 |
|--------------------|-----------|--|
| ☐ <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 |
|---|----------------|-------------|--|
| □ | <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 |
|---|--------------------|---|
| <input type="checkbox"/> <input type="checkbox"/> <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. |
| <input type="checkbox"/> <i>Class</i> | AttributeClassType | Informationsklasse. |
| <input type="checkbox"/> <i>Mandatory</i> | xs:boolean | Falls gesetzt ist es obligatorisch dem Benutzer die Information anzuzeigen. |
| <input type="checkbox"/> <i>Importance</i> | PercentType | Wichtigkeit der Information in Prozent. |
| <input type="checkbox"/> <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"/> <input checked="" type="checkbox"/> <i>Description</i> | TextType | |
| <input type="checkbox"/> <input checked="" type="checkbox"/> <i>StopLink</i> | StopLinkType | Fußwegverbindungen zu Stops. |
| <input type="checkbox"/> <input checked="" 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. |
| <input type="checkbox"/> <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"/> <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.

Gruppenzugehörigkeit: *PlaceOrAreaGroup*

| Element | Typ | Kommentar |
|--|-----------------------|---|
| <i>ID</i> | PlaceIDType | Standort-ID. Für jeden physischen Standort muss diese ID einheitlich sein. |
| <input type="checkbox"/> <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. |
| <input type="checkbox"/> <i>Capacity</i> | xs:nonNegativeInteger | Kapazität des Standortes zum Zurückweisen von Buchungszielen. |
| <input type="checkbox"/> <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.

Gruppenzugehoerigkeit: *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.

Gruppenzugehörigkeit: *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 type="checkbox"/> <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"/>

```

```
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. |
| □ <i>GlobalID</i> | GlobalIDType | Global eindeutige ID. Wird benutzt um Duplikate eines Objektes das von verschiedenen Systemen verwaltet wird zu erkennen. |
| ▣ <i>Name</i> | TextType | Druckbarer Name des Bookee. |
| ○ <i>PlaceID</i> | PlaceIDType | Standard-ID, auf dem der Bookee platziert ist. Muss mit stationsbasierten Buchungszielen befüllt werden. |
| ○ <i>PlaceGroupID</i> | PlaceGroupIDType | Standort-Gruppen-ID dieses Bookees, falls er nicht zu einem festen Standort gehört. |
| ○ <i>AreaID</i> | AreaIDType | Gebiet-ID dieses Bookees. Muss mit Free-Floating-Buchungszielen befüllt werden. |
| <i>Class</i> | ClassType | Klasse des Bookees. |
| □ <i>BookingHorizon</i> | xs:duration | Maximale Dauer in der Zukunft, wenn dieses Ziel gebucht werden kann. Ist relativ zur aktuellen Zeit. |
| □ <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. |
| □ <i>OpeningTime</i> | xs:duration | Benötigte Zeit um das Fahrzeug aufzuschließen.. |
| □ <i>Engine</i> | EngineType | |
| □ <i>CO2Factor</i> | xs:nonNegativeInteger | CO2-Emissionen in Gramm pro Kilometer. |
| □ <i>MaxDistance</i> | xs:nonNegativeInteger | Maximale Distanz in Metern, die ein Bookee ohne längere Stops zurücklegen kann. Falls nicht gegeben, ist kein Limit angenommen. |
| □ ▣ <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 | |
| □ <i>TimePeriod</i> | TimePeriodType | Tatsächlicher Zeitraum für die Buchung, wurde ggfs. an das Buchungsraster angepasst. |
| □ □ <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. |
| □ <i>PlaceID</i> | PlaceIDType | Für aktuell verfügbare Buchungsziele, welche nicht an einen spezifischen Standort gebunden sind, sollte der aktuelle Standort zurückgegeben werden. |
| □ <i>GeoPosition</i> | GeoPositionType | Für neu verfügbare Fahrzeuge, welche nicht an einen spezifischen Standort gebunden sind, sollte die Position zurückgegeben werden. |
| ○ <i>Inavailability</i> | TimePeriodType | Neue Nicht-Verfügbarkeitsperiode. |
| ○ <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

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 5.31: 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 Tarifiedetails. |
| <i>Price</i> | EuroPriceType | Preis in Euro-Cent. |
| <input type="checkbox"/> <i>Description</i> | xs:string | Druckbare Beschreibung des Tarifiedetails. |

```

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 5.32: 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 5.33: TariffType

5.8 Authentifizierung

Benutzeridentifikation

UserInfoType: Typ für eine User-Identifizierung.

| 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. |
| ○ <i>Password</i> | xs:normalizedString | Passwort des Users. |
| ○ <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>
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 5.34: UserInfoType

Authentifizierung

AuthType: Typ für eine Authentifizierung innerhalb einer Anfrage.

| Element | Typ | Kommentar |
|-------------------------|---------------|--|
| ○ <i>SessionID</i> | SessionIDType | Valide Session-ID. Falls eine Session wiederverwendet wird, diese ausfüllen. |
| ○ □ <i>UserInfo</i> | UserInfoType | Falls keine Session verfügbar ist, muss der User authentisieren. |
| ○ <i>Anonymous</i> | xs:boolean | Falls weder eine existierende Session, noch eine Nutzerinformation verfügbar ist, kann eine anonyme Session geöffnet werden. |
| □ <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 5.35: AuthType

5.9 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 5.36: ErrorType

6 Technische Realisierung

6.1 Nachrichtenkodierung

Die Nachrichten zwischen den beiden System 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, entsteht 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.

Für einige Nachrichten sind Beispiele zur Illustration vorhanden, siehe z.B. Listing 7.24.

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="IxsiMessageType">
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: IxsiMessageType

Basisantwort

AbstractBaseResponseType: Abstrakter Typ als eine Basis für Antworten.

| Element | Typ | Kommentar |
|------------------|-----------|-----------|
| □ □ <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.

Gruppenzugehörigkeit: *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.

Gruppenzugehörigkeit: *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.

Gruppenzugehörigkeit: *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 |
|----------------------------|------------------------------|---|
| □ <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 | Authentifikationsinformationen. |
| ○ <i>OpenSession</i> | OpenSessionRequestType | Öffnen einer Session für einen User. |
| ○ <i>CloseSession</i> | CloseSessionRequestType | Explizites Schließen einer Session. |
| ○ <i>TokenGeneration</i> | TokenGenerationRequestType | Generieren eines Tokens für einen User, der sein Passwort ersetzt. |
| ○ <i>Availability</i> | AvailabilityRequestType | Die Verfügbarkeit für gegebene Buchungsziele erhalten. |
| ○ <i>PlaceAvailability</i> | PlaceAvailabilityRequestType | Die verfügbare Kapazität für gegebene Standorte erhalten. |
| ○ <i>PriceInformation</i> | PriceInformationRequestType | Beziehen von Preisinformationen. |
| ○ <i>Booking</i> | BookingRequestType | Buchung eines Buchungsziels. |
| ○ <i>ChangeBooking</i> | ChangeBookingRequestType | Ändern einer Buchung. |

```

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:choice>
15  </xs:sequence>
16 </xs:group>

```

XML Quellcode 7.8: UserTriggeredRequestGroup

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

| Element | Typ | Kommentar |
|--|-------------------------------|---|
| <input type="checkbox"/> <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. |
| <input type="checkbox"/> <i>SessionTimeout</i> | xs:duration | Inaktivitäts-Timeout einer Session. Wenn eine Zeitspanne von Inaktivität für diese Session diesen Wert übersteigt, läuft die Session ab. |
| <input type="radio"/> <i>OpenSession</i> | OpenSessionResponseType | Öffnen einer Session für den User. |
| <input type="radio"/> <i>CloseSession</i> | CloseSessionResponseType | Explizites Schließen einer Session. |
| <input type="radio"/> <i>TokenGeneration</i> | TokenGenerationResponseType | Generieren eines Tokens für einen User, der sein Passwort ersetzt. |
| <input type="radio"/> <i>Availability</i> | AvailabilityResponseType | Beziehen der Verfügbarkeit der gegebenen Buchungsziele. |
| <input type="radio"/> <i>PlaceAvailability</i> | PlaceAvailabilityResponseType | Beziehen der verfügbaren Kapazitäten für gegebene Standorte. |
| <input type="radio"/> <i>PriceInformation</i> | PriceInformationResponseType | Beziehen einer Preisinformation. |
| <input type="radio"/> <i>Booking</i> | BookingResponseType | Buchen eines Buchungsziels. |
| <input type="radio"/> <i>ChangeBooking</i> | ChangeBookingResponseType | Ändern einer Buchung. |

```

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:choice>
15 </xs:sequence>
16 </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.

Gruppenzugehoerigkeit: *UserTriggeredRequestGroup*

| Element | Typ | Kommentar |
|---------|-----|-----------|
| (leer) | | |

```
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*

Gruppenzugehoerigkeit: *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 |
|---------|-----|-----------|
| (leer) | | |

```
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.

Gruppzugehörigkeit: *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: ChangedProvidersResponseType

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

Basistyp: *AbstractBaseResponseType*

Gruppzugehörigkeit: *StaticDataResponseGroup*

| Element | Typ | Kommentar |
|---------------------|----------------|---|
| □ □ <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: Synchrone Anfrage von Verfügbarkeiten von Buchungszielen.
Gruppenzugehörigkeit: *UserTriggeredRequestGroup*

| | Element | Typ | Kommentar |
|-----|----------------------|-----------------------------|--|
| ○ □ | BookingTarget | BookingTargetPropertiesType | |
| ○ | Circle | GeoCircleType | |
| ○ | GeoRectangle | GeoRectangleType | |
| □ | TimePeriod | 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: Synchrone Antwort für die Verfügbarkeit von Buchungszielen.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehörigkeit: *UserTriggeredResponseGroup*

| Element | Typ | Kommentar |
|--|-------------------------------|--|
|   <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>

```



```

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

| Element | Typ | Kommentar |
|------------------|-----------------------|--|
| □ □ <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 |
|---------|-----|-----------|
| (leer) | | |

```

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 |
|----------------------------|---------------------|---------------------------|
| ☐ ☐ <i>BookingTargetID</i> | BookingTargetIDType | Abonnierte Buchungsziele. |

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

Gruppenzugehoerigkeit: *PushMessageGroup*

| Element | Typ | Kommentar |
|-----------------------------|-------------------------------------|-----------|
| ☐ <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 Verfügbarkeitsabonnementinformation

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

Gruppenzugehörigkeit: *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 </Ixsi>

```

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*

Gruppzugehörigkeit: *SubscriptionResponseGroup*

| Element | Typ | Kommentar |
|---------|-----|-----------|
| (leer) | | |

```

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.

Gruppzugehörigkeit: *SubscriptionRequestGroup*

| Element | Typ | Kommentar |
|---------|-----|-----------|
| (leer) | | |

PlaceAvailabilitySubscriptionStatusResponseType: Antwort um alle abonnierten Standorte zu erhalten


Basistyp: *AbstractBaseResponseType*

Gruppzugehörigkeit: *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.

Gruppzugehörigkeit: *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.

Gruppenzugehörigkeit: *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*

Gruppenzugehörigkeit: *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.

Gruppenzugehörigkeit: *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 |
|---|-------------|---|
| <input type="checkbox"/> <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 <Icsi>
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 </Icsi>

```

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. |
| ○ <i>NewTimePeriodProposal</i> | TimePeriodProposalType | Neuer Zeitperioden-Vorschlag um die Buchung zu ändern. |
| ○ <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: *AbstractResponseType*

Gruppenzugehoerigkeit: *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="AbstractResponseType">
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

7.10 Dienst 5 - Buchungsabonnement

Buchungsabonnement

BookingAlertSubscriptionRequestType: Anfrage um Alarmmeldungen für gegebene Buchungen zu abonnieren.

Gruppenzugehoerigkeit: *SubscriptionRequestGroup*

| Element | Typ | Kommentar |
|--|---------------|--|
| <input checked="" type="checkbox"/> <i>BookingID</i> | BookingIDType | Buchungen zu abonnieren / kündigen. |
| <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 7.49: 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 7.50: 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 |
|--|---------------|-----------------------|
| <input type="checkbox"/> <input type="checkbox"/> <i>BookingID</i> | BookingIDType | Abonnierte Buchungen. |

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

Gruppenzugehoerigkeit: *PushMessageGroup*

| Element | Typ | Kommentar |
|---|-------------------|-----------|
| <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="BookingAlertPushMessageType">
2   <xs:sequence>
3     <xs:element name="BookingChange" type="BookingChangeType" maxOccurs="unbounded"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.51: BookingAlertPushMessageType

Vollständige Buchungsabonnementinformation

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

Gruppenzugehoerigkeit: *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="CompleteBookingAlertRequestType">
2   <xs:sequence>
3     <xs:element name="MaxResults" type="xs:integer" minOccurs="0"/>
4   </xs:sequence>
5 </xs:complexType>

```

XML Quellcode 7.52: CompleteBookingAlertRequestType

CompleteBookingAlertResponseType: Nachricht um den vollständigen aktuellen Status abon-
nierter Buchungen zu erhalten. Verschiedene solcher Antworten können zu einer einzigen An-
frage gehören.

Basistyp: *AbstractBaseResponseType*

Gruppzugehoerigkeit: *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 7.53: CompleteBookingAlertResponseType

7.11 Dienst 6 - Preisauskunft

Preisinformation

PriceInformationRequestType: Anfrage um einen Preis zu erhalten.

Gruppzugehoerigkeit: *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 7.54: PriceInformationRequestType

PriceInformationResponseType: Antwort um einen Preis zu erhalten.

Basistyp: *AbstractBaseResponseType*

Gruppenzugehoerigkeit: *UserTriggeredResponseGroup*

| | Element | Typ | Kommentar |
|---|---------------|------------|-----------|
| □ | <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 7.55: PriceInformationResponseType

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 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.

| 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. |
| language_not_supported | Angefragte Sprache nicht vollständig unterstützt, andere Sprache geliefert. |

Quellcodeverzeichnis

| | | |
|------|---|----|
| 5.1 | Baisdatentypen | 28 |
| 5.2 | TextType | 29 |
| 5.3 | CoordType | 30 |
| 5.4 | AddressType | 31 |
| 5.5 | GeoPositionType | 31 |
| 5.6 | GeoCircleType | 32 |
| 5.7 | GeoRectangleType | 32 |
| 5.8 | GeoAreaType | 33 |
| 5.9 | IncExcGeoAreaType | 33 |
| 5.10 | TimePeriodType | 34 |
| 5.11 | TimePeriodProposalType | 34 |
| 5.12 | OriginDestType | 35 |
| 5.13 | InfoType | 35 |
| 5.14 | StopLinkType | 36 |
| 5.15 | PlaceOrAreaGroup | 37 |
| 5.16 | DurationGroup | 37 |
| 5.17 | BookingTargetIDType | 38 |
| 5.18 | ProviderPlaceIDType | 38 |
| 5.19 | AttributeType | 39 |
| 5.20 | PlaceType | 40 |
| 5.21 | DensityAreaType | 40 |
| 5.22 | FloatingAreaType | 41 |
| 5.23 | PlaceGroupType | 42 |
| 5.24 | ProviderType | 42 |
| 5.25 | BookingTargetType | 44 |
| 5.26 | BookingType | 45 |
| 5.27 | BookingTargetPropertiesType | 46 |
| 5.28 | BookingTargetAvailabilityType | 47 |
| 5.29 | BookingTargetChangeAvailabilityType | 48 |
| 5.30 | PlaceAvailabilityType | 49 |
| 5.31 | BookingChangeType | 50 |
| 5.32 | TariffDetailType | 51 |
| 5.33 | TariffType | 52 |
| 5.34 | UserInfoType | 52 |

| | | |
|------|---|----|
| 5.35 | AuthType | 53 |
| 5.36 | ErrorType | 54 |
| 7.1 | IxsiMessageType | 58 |
| 7.2 | AbstractBaseResponseType | 58 |
| 7.3 | TransactionType | 59 |
| 7.4 | QueryRequestType | 59 |
| 7.5 | QueryResponseType | 60 |
| 7.6 | SubscriptionRequestType | 60 |
| 7.7 | SubscriptionResponseType | 61 |
| 7.8 | UserTriggeredRequestGroup | 62 |
| 7.9 | UserTriggeredResponseGroup | 63 |
| 7.10 | OriginDestGroup | 64 |
| 7.11 | OpenSessionRequestType | 65 |
| 7.12 | OpenSessionResponseType | 66 |
| 7.13 | CloseSessionRequestType | 66 |
| 7.14 | CloseSessionResponseType | 66 |
| 7.15 | HeartBeatRequestType | 67 |
| 7.16 | HeartBeatResponseType | 67 |
| 7.17 | TokenGenerationRequestType | 68 |
| 7.18 | TokenGenerationResponseType | 68 |
| 7.19 | BookingTargetsInfoRequestType | 69 |
| 7.20 | BookingTargetsInfoResponseType | 69 |
| 7.21 | ChangedProvidersRequestType | 70 |
| 7.22 | ChangedProvidersResponseType | 70 |
| 7.23 | AvailabilityRequestType | 71 |
| 7.24 | AvailabilityRequest Beispiel | 72 |
| 7.25 | AvailabilityResponseType | 72 |
| 7.26 | AvailabilityResponse Beispiel | 73 |
| 7.27 | PlaceAvailabilityRequestType | 73 |
| 7.28 | PlaceAvailabilityResponseType | 74 |
| 7.29 | AvailabilitySubscriptionRequestType | 75 |
| 7.30 | AvailabilitySubscriptionRequest Beispiel | 75 |
| 7.31 | AvailabilitySubscriptionResponseType | 76 |
| 7.32 | AvailabilityPushMessageType | 76 |
| 7.33 | AvailabilityPush Beispiel | 77 |
| 7.34 | CompleteAvailabilityRequestType | 77 |
| 7.35 | CompleteAvailabilityRequest Beispiel | 77 |
| 7.36 | CompleteAvailabilityResponseType | 78 |
| 7.37 | CompleteAvailabilityResponse Beispiel | 78 |
| 7.38 | PlaceAvailabilitySubscriptionRequestType | 79 |
| 7.39 | PlaceAvailabilitySubscriptionResponseType | 80 |
| 7.40 | PlaceAvailabilityPushMessageType | 81 |

| | | |
|------|---|----|
| 7.41 | CompletePlaceAvailabilityRequestType | 81 |
| 7.42 | CompletePlaceAvailabilityResponseType | 82 |
| 7.43 | BookingRequestType | 83 |
| 7.44 | BookingRequest Beispiel | 83 |
| 7.45 | BookingResponseType | 83 |
| 7.46 | BookingResponse Beispiel | 84 |
| 7.47 | ChangeBookingRequestType | 84 |
| 7.48 | ChangeBookingResponseType | 85 |
| 7.49 | BookingAlertSubscriptionRequestType | 86 |
| 7.50 | BookingAlertSubscriptionResponseType | 86 |
| 7.51 | BookingAlertPushMessageType | 87 |
| 7.52 | CompleteBookingAlertRequestType | 87 |
| 7.53 | CompleteBookingAlertResponseType | 88 |
| 7.54 | PriceInformationRequestType | 89 |
| 7.55 | PriceInformationResponseType | 89 |

Stichwortverzeichnis

- AbstractBaseResponseType, 58, 65–70, 72, 74–76, 78, 80, 81, 83, 85, 86, 88, 89
- Address, 31
- AddressType, 30, 31
- Anmeldung, 17
- Anonymous, 53
- Area, 40, 41
- AreaID, 44
- AreaIDType, 28, 41, 44
- AttributeClassType, 28, 29, 35, 39, 92
- AttributeID, 37, 42, 44
- AttributeIDType, 28, 37, 39, 42, 44
- Attributes, 69
- AttributeType, 39, 69
- Auth, 62
- AuthType, 53, 62
- Availability, 48, 49, 62, 63
- AvailabilityChange, 76
- AvailabilityPushMessage, 22
- AvailabilityPushMessageType, 76
- AvailabilityRequestType, 62, 71
- AvailabilityResponseType, 63, 72
- AvailabilitySubscriptionRequest, 22
- AvailabilitySubscriptionRequestType, 74, 75
- AvailabilitySubscriptionResponseType, 75, 76
- AvailabilitySubscriptionStatusRequestType, 76
- AvailabilitySubscriptionStatusResponseType, 76
- Begin, 33
- Benutzer
 - FVS, 8
 - RIS, 8
- Betreiber
 - FVS, 9
 - RIS, 9
- Bookee, 69
- BookeeID, 38
- BookeeIDType, 28, 38
- Booking, 24, 62, 63, 83, 85
- BookingTargetsInfo, 20
- BookingAlertPushMessageType, 87
- BookingAlertSubscriptionRequestType, 85
- BookingAlertSubscriptionResponseType, 86
- BookingAlertSubscriptionStatusRequestType, 86
- BookingAlertSubscriptionStatusResponseType, 86
- BookingChange, 87, 88
- BookingChangeType, 49, 50, 87, 88
- BookingGrid, 44
- BookingHorizon, 44
- BookingID, 50, 84, 85, 87
- BookingIDType, 28, 45, 50, 84, 85, 87
- BookingRequestType, 62, 82
- BookingResponseType, 63, 83
- BookingTarget, 71, 72, 78
- BookingTargetAvailabilityType, 46, 47, 72, 78
- BookingTargetChangeAvailabilityType, 48, 76
- BookingTargetID, 75, 76, 82, 89
- BookingTargetIDType, 38, 44, 46–48, 75, 76, 82, 89
- BookingTargetPropertiesType, 46, 71

BookingTargetsInfoRequestType, 68, 69
BookingTargetsInfoResponseType, 69
BookingTargetType, 43, 44, 69
BookingType, 45, 83, 85

CalcTime, 60, 61
Cancel, 84
Cancelled, 50
Capacity, 40
Center, 31
ChangeBooking, 24, 62, 63
ChangeBookingRequestType, 62, 84
ChangeBookingResponseType, 63, 85
ChangedProviders, 20
ChangedProvidersRequestType, 70
ChangedProvidersResponseType, 70
Circle, 71, 73
City, 30
Class, 35, 39, 44
ClassType, 28, 29, 44, 91
CloseSession, 62, 63
CloseSessionRequestType, 62, 66
CloseSessionResponseType, 63, 66
CO2Factor, 44
Code, 54
CompleteAvailabilityRequest, 23
CompleteAvailabilityRequestType, 77
CompleteAvailabilityResponseType, 78
CompleteBookingAlertRequestType, 87
CompleteBookingAlertResponseType, 88
CompletePlaceAvailabilityRequestType, 81
CompletePlaceAvailabilityResponseType, 81, 82
Coord, 31, 40
CoordType, 30–32, 35
Country, 30
CurrentDrivingRange, 47
CurrentStateOfCharge, 47
CustomerChoice, 42

DensityAreaType, 40, 41
Description, 37, 51

Dest, 64
Detail, 51
Distance, 36, 46, 89
Duration, 37
DurationGroup, 36, 37, 40, 41

End, 33
Engine, 44
EngineType, 28, 29, 44, 91
Error, 58
ErrorCodeType, 28, 29, 54, 93
ErrorType, 53, 54, 58
EuroPriceType, 28, 51
EventHorizon, 75
Exclude, 33

Fahrzeugverleihsystem, 8
FloatingAreaType, 41, 69
FreeFloatingArea, 69
FVS, 8, 17

GeoAreaType, 32, 33, 40
GeoCircleType, 31, 71, 73
GeoPosition, 35, 47, 48
GeoPositionType, 31, 40, 47, 48
GeoRectangle, 71, 73
GeoRectangleType, 32, 71, 73
GlobalID, 40, 44
GlobalIDType, 28, 40, 44

HeartBeatRequestType, 67
HeartBeatResponseType, 67

ID, 39–42, 44–49
Importance, 35, 39
Impossible, 50
Inavailability, 47, 48
IncExcGeoAreaType, 33, 41
Info, 45, 51
InfoType, 35, 45, 51
IxsiMessageType, 57, 58

Language, 29, 62
Last, 78, 82, 88
Latitude, 30

Longitude, 30
 LowerRight, 32

 Mandatory, 35, 39
 MaxDistance, 44
 MaxPlaces, 81
 MaxResults, 87
 MaxTargets, 77
 MaxWait, 34
 MessageBlockID, 78, 82, 88
 MessageID, 59

 Name, 37, 42, 44, 51
 NewPeriod, 50
 NewTimePeriodProposal, 84
 NonFatal, 54
 Notification, 50

 OnPremisesTime, 40
 OpeningTime, 44
 OpenSession, 26, 62, 63
 OpenSessionRequestType, 62, 65
 OpenSessionResponseType, 63, 65, 66
 Origin, 64
 OriginDestGroup, 64, 82, 88
 OriginDestType, 34, 35, 64

 Password, 52
 PercentType, 28, 35, 39, 42, 47
 Place, 69, 74
 PlaceAvailability, 62, 63, 81, 82
 PlaceAvailabilityPushMessageType, 80, 81
 PlaceAvailabilityRequestType, 62, 73
 PlaceAvailabilityResponseType, 63, 74
 PlaceAvailabilitySubscriptionRequestType, 79
 PlaceAvailabilitySubscriptionResponseType, 80
 PlaceAvailabilitySubscriptionStatusRequestType, 80
 PlaceAvailabilitySubscriptionStatusResponseType, 80
 PlaceAvailabilityType, 48, 49, 74, 81, 82
 PlaceGroup, 69
 PlaceGroupID, 44
 PlaceGroupIDType, 28, 42, 44
 PlaceGroupType, 41, 42, 69
 PlaceID, 35, 38, 42, 44, 47, 48, 73, 79, 80
 PlaceIDType, 28, 35, 38, 40, 44, 47, 48
 PlaceOrAreaGroup, 36, 37, 39, 41
 PlaceType, 39, 40, 69
 PolyPoint, 32
 PostalCode, 30
 Price, 51
 PriceInformation, 62, 63
 PriceInformationRequest, 26
 PriceInformationRequestType, 62, 88, 89
 PriceInformationResponseType, 63, 89
 Probability, 42
 ProbabilityPlaceIDType, 42
 Provider, 69, 70
 ProviderFilter, 69
 ProviderID, 37, 38, 52
 ProviderIDType, 28, 37, 38, 42, 52, 69, 70
 ProviderPlaceIDType, 38, 49, 73, 79, 80
 ProviderType, 42, 69
 PushMessageGroup, 65, 76, 80, 87

 QueryRequestType, 58, 59
 QueryResponseType, 58, 60

 Radius, 31
 Reason, 50
 Reiseinformationssystem, 7
 Repossibilized, 50
 Request, 58
 RequestMessageGroup, 60, 65, 77, 81, 87
 Response, 58
 ResponseMessageGroup, 61, 65, 78, 81, 88
 RIS, 7, 17

 Separate, 39
 SessionID, 53, 63
 SessionIDType, 28, 53, 63
 SessionTimeout, 53, 63
 SeverityFactor, 46
 ShortName, 42
 simpleType, 30

- StaticDataRequestGroup, 59, 65, 68, 70
- StaticDataResponseGroup, 60, 65, 69, 70
- StopID, 36
- StopLink, 37
- StopLinkType, 36, 37
- StreetHouseNr, 30
- SubArea, 41
- SubscriptionAdministrationRequestGroup, 60, 65, 67
- SubscriptionAdministrationResponseGroup, 61, 65, 67
- SubscriptionMessage, 58
- SubscriptionMessageType, 58
- SubscriptionRequest, 58
- SubscriptionRequestGroup, 60, 65, 74, 76, 79, 80, 85, 86
- SubscriptionRequestType, 58, 60
- SubscriptionResponse, 58
- SubscriptionResponseGroup, 61, 65, 75, 76, 80, 86
- SubscriptionResponseType, 58, 61
- SystemID, 59, 60
- SystemIDType, 28, 59, 60
- SystemMessage, 54
- Tariff, 89
- TariffDetailType, 51
- TariffType, 51, 89
- Text, 29, 35, 39
- TextType, 29, 37, 39, 44, 50
- TimePeriod, 45, 71
- TimePeriodProposal, 82, 89
- TimePeriodProposalType, 34, 82, 84, 89
- TimePeriodType, 33, 34, 45, 47, 48, 50, 71
- TimeStamp, 59
- Timestamp, 69, 70
- timestamp, 20
- Token, 52, 68
- TokenGeneration, 62, 63
- TokenGenerationRequestType, 62, 67, 68
- TokenGenerationResponseType, 63, 68
- Transaction, 59–61
- TransactionType, 59–61
- Unsubscription, 75, 79, 85
- UpperLeft, 32
- URL, 35, 39
- UserID, 52
- UserIDType, 28, 52
- UserInfo, 53
- UserInfoType, 52, 53
- UserMessage, 54
- UserTriggeredRequestGroup, 59, 61, 62, 65–67, 71, 73, 82, 84, 88
- UserTriggeredResponseGroup, 60, 63, 65, 66, 68, 72, 74, 83, 85, 89
- Variance, 37
- WithText, 35, 39
- xs:anyURI, 35, 39
- xs:boolean, 33, 35, 39, 42, 50, 53, 54, 75, 78, 79, 82, 84, 85, 88
- xs:dateTime, 33, 59, 69, 70
- xs:duration, 34, 37, 40, 44, 53, 60, 61, 63, 75
- xs:integer, 77, 81, 87
- xs:language, 29, 62
- xs:nonNegativeInteger, 28, 31, 36, 40, 44, 46, 47, 49, 59, 89
- xs:normalizedString, 42, 52
- xs:string, 29, 30, 35, 51, 54
- xs:token, 28, 36, 52, 68, 78, 82, 88