



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Bundesamt für Strassen ASTRA

IT-DOKUMENTATION

**FACHAPPLIKATION
BETRIEBS- UND SICHER-
HEITSAUSRÜSTUNGEN
SOFORTLÖSUNG
(FA BSAS)**

Schnittstelle

*Ausgabe 2019 V2.01
ASTRA 63012*

Impressum

Autoren

Fuchs Eugen (ASTRA N-ST, Vorsitz)
Ott Stefan (ASE, Zürich, Erarbeitung)
Tayebi Mirwais (ASE, Zürich, Erarbeitung)

Übersetzung (Originalversion in Deutsch)

Herausgeber

Bundesamt für Strassen ASTRA
Abteilung Strassennetze N
Standards und Sicherheit der Infrastruktur SSI
3003 Bern

Bezugsquelle

Das Dokument kann kostenlos von www.astra.admin.ch herunter geladen werden.

© ASTRA 2019

Abdruck - ausser für kommerzielle Nutzung - unter Angabe der Quelle gestattet.

Inhaltsverzeichnis

	Impressum	2
1	Einleitung	5
1.1	Zweck des Dokuments	5
1.2	Geltungsbereich	5
1.3	Adressaten	5
1.4	Inkrafttreten und Änderungen	5
2	Allgemein	6
3	Funktionen	7
3.1	Fachkatalog Typdefinitionen	7
3.2	Fachkatalog Strukturdefinitionen.....	7
3.3	Verzeichnisse (Inventardaten)	7
3.4	Inventarmutationen	7
3.5	Zustandsdaten.....	7
4	Datenmodell	8
4.1	Inventardaten	8
4.1.1	Katalogdaten	9
4.1.2	Verzeichnisse	14
4.1.3	Eigenschaften.....	18
4.2	Zustandsdaten.....	21
4.2.1	Katalogdaten	21
4.2.2	Zustandsbewertungen.....	22
5	Webservice	23
5.1	Datenübertragungsobjekt.....	23
5.2	Datengültigkeit.....	23
5.3	Authentifikation.....	23
5.4	Konzept & Beschreibung Webservices	23
5.4.1	Inventardaten (LiveInventory-Webservice)	24
5.4.2	Projektdateien (Projectroom-Webservice).....	27
5.4.3	Zustandsdaten (InventoryEvaluation-Webservice)	28
6	Codebeispiele	29
6.1	Authentifizierung	29
6.2	Inventardaten	29
6.2.1	Katalogdaten	29
6.2.2	BSA- und Ortsverzeichnis BSAS	30
6.2.3	Eigenschaften.....	31
	Glossar	32
	Literaturverzeichnis	33
	Auflistung der Änderungen.....	35

1 Einleitung

1.1 Zweck des Dokuments

Ausgangslage

Das Inventar zu den Betriebs- und Sicherheitsausrüstungen der Nationalstrasse wird im ASTRA in der Fachapplikation Betriebs- und Sicherheitsausrüstungen Sofortlösung (FA BSAS) verwaltet. Dieses Inventar wird von allen ASTRA Filialen und der Zentrale für die Erhaltungsplanung und als Schnittstelle zu den Gebietseinheiten genutzt.

Als Grundlage dient die Richtlinie ASTRA 13013 „Struktur und Kennzeichnung der Betriebs- und Sicherheitsausrüstung (AKS-CH)“ [1]. Sie gibt vor, wie die BSA strukturiert werden.

Zweck

Das vorliegende Dokument enthält die funktionellen und technischen Informationen zur Schnittstelle der FA BSAS.

Über diese Schnittstelle können Fremdsysteme die Fachkataloge und Verzeichnisdaten, welche in Zusammenhang mit der Richtlinie AKS-CH und damit mit den Betriebs- und Sicherheitsausrüstungen stehen, beziehen.

Anwendung

Das vorliegende Dokument ist anzuwenden bei der Projektierung und Realisierung von Schnittstellen, über welche ein Abgleich von Orts- und BSA Verzeichnisdaten zwischen der FA BSAS und Fremdsystemen stattfinden soll.

Hinweis:

Zu den Daten, welche abgeglichen werden gehören auch die Fachkataloge. Ein Datenabgleich kann einmalig oder periodisch vorgenommen werden. In den Datenabgleichprozess fallen auch Datenmutationen, welche zwischen der FA BSAS und den Fremdsystemen vereinbart werden.

1.2 Geltungsbereich

Dieses Dokument enthält nur Vorgaben und technische Informationen zur Schnittstelle mit der FA BSAS. Es enthält keinerlei Vorgaben oder Definitionen, welche andere Fachapplikationen des ASTRA betreffen.

Dieses Dokument enthält nur Vorgaben zum Austausch von Fachkatalogen, Orts- und BSA-Verzeichnisdaten sowie der Zustandsdaten. Weitere Daten, welche im Rahmen des Betriebs- und Unterhalt von BSA im ASTRA oder bei Gebietseinheiten verwaltet und über die Schnittstelle übertragen werden sollen, sind nicht enthalten.

1.3 Adressaten

Diese IT-Dokumentation richtet sich an Ersteller von IT-Systemen, die Daten aus der FA BSAS beziehen, weiterverarbeiten und Änderungen an die FA BSAS zurückgeben.

1.4 Inkrafttreten und Änderungen

Die vorliegende IT-Dokumentation tritt am 01.10.2015 in Kraft. Die Auflistung der Änderungen ist auf Seite 29 dokumentiert.

2 Allgemein

Die Schnittstelle besteht aus Standard SOAP Web Services und ist via Internet systemunabhängig verfügbar. Es wird davon ausgegangen, dass zwischen den beteiligten Systemen keine grossen Mengen an Daten sehr schnell in beiden Richtungen ausgetauscht werden müssen.

Die Schnittstelle der FA BSAS kann heute die folgenden Informationen übertragen:

- Fachkataloge BSA-, Orts- und Zugehörigkeitstypen
- Fachkataloge BSA-, Orts- und Zugehörigkeitsstruktur
- Inventardaten (BSA-, Orts- und Zugehörigkeitsverzeichnisdaten)

Zustandsdaten

Ausserdem können folgende Informationen übernommen werden:

- Mutationen & Ergänzungen an den Inventardaten

Mutationen & Ergänzungen an den Zustandsdaten

Jedes Objekt ist durch einen Primärschlüssel identifizierbar und referenzierbar. Das reduziert die Menge der zu übertragenden Daten, weil eine Referenz (Primärschlüssel) immer kleiner ist als ein vollständiges Objekt.

Um die auszuführenden Anfragen an die Schnittstelle möglichst tief zu halten, müssen die übertragenden Objekte so viele Informationen wie möglich enthalten. Allerdings ist zu beachten, dass keine unnötigen Informationen hinzugefügt werden, welche lediglich die Menge der übertragenden Daten erhöhen, aber keinen tatsächlichen Mehrwert generieren.

3 Funktionen

3.1 Fachkatalog Typendefinitionen

Als Basis der Inventardaten werden in der FA BSAS die einzelnen Inventartypen definiert. Die Fachkataloge mit den Typendefinitionen können nur gelesen werden. Der Umfang der definierten Typen ist verhältnismässig klein. Ausserdem sind nur wenige Eigenschaften mit ihnen verbunden. Daher wird eine Funktion zur Verfügung gestellt, welche alle in den Fachkatalogen definierten Typen zurückgibt. Für Objekte mit einem Primärschlüssel ist jedoch eine zweite Funktion nötig, um über den Primärschlüssel nur einzelne Objekte abfragen zu können.

3.2 Fachkatalog Strukturdefinitionen

In den Strukturen ist die Datenstruktur der Inventardaten definiert (hierarchische Beziehung der Typen) und kann ebenfalls nur gelesen werden. Die Fachkataloge der Strukturen sind nur unwesentlich grösser als diejenigen der Typen. Daher wird auch für diese Fachkataloge eine Funktion zur Verfügung gestellt, welche die gesamte Struktur zurückgibt. Auch für die Fachkataloge der Strukturen wird eine zweite Funktion zur Verfügung gestellt, um über den Primärschlüssel einzelne Objekte der Fachkataloge abzufragen.

Weil die Strukturen hierarchisch definiert sind, wird ausserdem eine Funktion zur Verfügung gestellt, um die Wurzel (oberste Ebene der Hierarchie) anzufragen mit der Option, eine oder mehrere Ebenen der Hierarchie zurückzugeben.

3.3 Verzeichnisse (Inventardaten)

Der Umfang der Verzeichnisse bzw. Inventardaten ist wesentlich grösser als derjenige der Fachkataloge. Eine Funktion, welche immer alle Objekte gleichzeitig zurückgibt ist daher nicht optimal. Da die Verzeichnisdaten ebenfalls hierarchisch strukturiert sind, wird eine Funktion zur Verfügung gestellt, über welche die Wurzel der Struktur (oberste Ebene der Hierarchie) angefragt werden kann. Diese Funktion beinhaltet auch die Option eine oder mehrere Ebenen der Hierarchie zurückzugeben.

3.4 Inventarmutationen

Mutationen an bestehenden Inventardaten betreffen nur eine Teilmenge aller verfügbaren Objekte. Die Schnittstelle verfügt daher über eine Funktion, über welche nur die Teilmenge der mutierten Daten abgefragt werden kann.

3.5 Zustandsdaten

Die Zustandsdaten in der FA BSAS sind in unterschiedlichen Bewertungssystem gespeichert, welche sich sowohl in der Granularität der Zustandsbewertung als auch in den Bewertungskriterien unterscheiden. Die Schnittstelle bietet den Anwendern die Möglichkeit sowohl die Definitionen der Bewertungssysteme abzufragen als auch die erfassten Zustandsbewertungen zu lesen oder in die FA BSAS zu schreiben.

4 Datenmodell

4.1 Inventardaten

Die Inventardaten lassen sich in vier Bereiche unterteilen:

- Ort (AKSLocation)
- BSA (OSEInventory)
- Zugehörigkeit (AKSComponent)
- Ort BSAS (Location)¹

Die Bereiche sind ähnlich modelliert. Die Daten werden als Typen modelliert und in Fachkatalogen (Typenkatalogen) verwaltet. Die Typen werden in hierarchischen Gliederungen zu Strukturen zusammengefügt. Dabei kann ein spezifischer Typ an mehreren Stellen in der Struktur eingesetzt werden. Die Verzeichnisse (Inventardaten) widerspiegeln die Strukturen, d.h. jede Instanz folgt der vorgegebenen Struktur. Die Kinder einer Instanz können wiederum nur einem spezifischen Strukturpfad zugeordnet werden. Die Strukturen definieren also, welche Typen einer Instanz hierarchisch zugeordnet werden.

In der Abbildung 4.1 ist das Datenmodell bzw. die verschiedenen Entitäten dargestellt. Die Pfeile der Entitäten welche auf sich selber verweisen, dienen der Darstellung, dass diese Daten hierarchisch strukturiert sind. D.h. jede Entität eines Elternknotens kann weitere Entitäten besitzen bzw. selbst das Kind einer anderen Entität sein.

Die Farben der Entitäten geben einen Hinweis auf die Häufigkeit, mit welcher die Entitäten geändert werden. Die rot eingefärbten Entitäten sind fast konstant, die Änderungshäufigkeit ist sehr klein. Diese Daten werden auch „Katalogdaten“ genannt. Die blau eingefärbten Entitäten haben eine mittlere bis hohe Änderungshäufigkeit. Typischerweise werden diese Daten im Rahmen von Projekten mit BSA Bezug oder im laufenden Betrieb durch Reparaturen verändert. Wobei die Verzeichnissdaten (AKSLocation, AKSComponent, OSEInventory und Location) sowie die Geoinformationen (GeoDataFromTo) voraussichtlich in einem tieferen Rhythmus verändert werden, als die technischen Eigenschaften (OSData).

Die Replica-Entitäten sind Kopien und werden erstellt, wenn Daten innerhalb der Projekt-räume bearbeitet werden. Genauere Informationen zu Projekte bzw. deren Verwendungszweck sind im Kapitel 5.4 zu finden.

¹ In der FA BSAS sind aufgrund historischer Entwicklung zwei «Ortungssysteme» vorhanden, welche beide über die Schnittstelle (Webservice) exponiert sind.

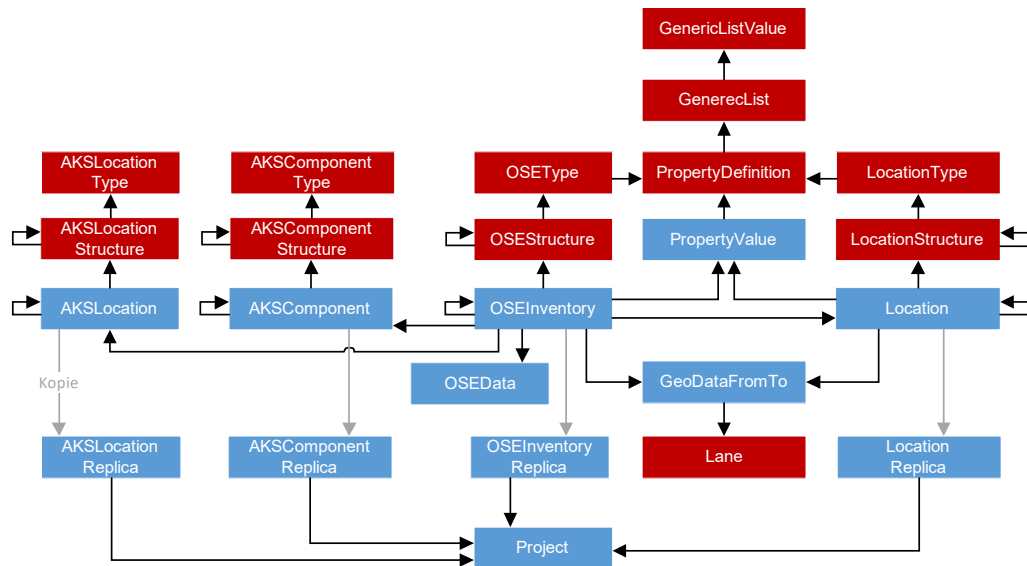


Abb. 4.1 Datenmodell Inventardaten

Jede Instanz von Orte (AKSLocation), BSA (OSEInventory) und Orte BSAS (Location) besitzt geographische Daten. Bei den BSA (OSEInventory) und Orten BSAS (Location) sind diese Informationen in einer getrennten Entität gespeichert. In der obigen Abbildung sind diese mit der Entität „GeoDataFromTo“ dargestellt. Bei den Orten (AKSLocation) ist diese Information direkt in der eigenen Entität vorhanden. Des Weiteren dient als zusätzliche Standortangabe von BSA der Streifen (Lane), welche die Platzierung auf der Fahrbahn beschreibt (z.B. Fahrstreifen 1).

Jede BSA kann einem Installationsort (AKSLocation und Location) zugewiesen werden. Zudem verfügen BSA über eine Liste von technischen Eigenschaften, welche in Abbildung 4.1 als „OSEData“ dargestellt werden. Neben diesen allgemein gültigen Eigenschaften gibt es zusätzlich typenspezifische Eigenschaften (PropertyValues), welche auf den Typen (OSEType, LocationType²) definiert sind (PropertyDefinition).

4.1.1 Katalogdaten

Die Katalogdaten sind grundsätzlich die Konstanten der Applikation. Sie werden normalerweise nur sehr selten geändert werden. Zu den Katalogdaten gehören:

- Ortstypen (AKSLocationType)
- Ortsstruktur (AKSLocationStructure)
- Zugehörigkeitstypen (AKSComponentType)
- Zugehörigkeitsstruktur (AKSComponentStructure)
- BSA Typen (OSEType)
- BSA Struktur (OSEStructure)
- Ortstypen BSAS (LocationType)
- Ortstruktur BSAS (LocationStructure)
- Streifen (Lane)
- Typenspezifische Eigenschaften (PropertyDefinition)
- Auswahllisten (GenericList)

Listeneinträge (GenericListValue)

Nachfolgend sind die einzelnen Data Transfer Objects (DTO) der Katalogdaten genauer beschrieben. Die Struktur-DTO's besitzen eine hierarchische Gliederung. Deshalb haben sie auch einen Verweis auf das Elternobjekt (ParentId), eine Liste der Kinderobjekte (ChildrenId) und die Ebene (Level) auf welchem sie sich in der hierarchischen Struktur befinden.

² Für AKSLocationType und AKSComponentType existieren keine typenspezifischen Eigenschaften

Abb. 4.2 AKSLocationTypeDTO (Fachkatalog Typen Ort)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Guid	Primarschlüssel	Kann nur gelesen werden	Ja
Name	String	Bezeichnung des Ortstyp	Kann nur gelesen werden	Ja
Shortcut	String	Kurzform des Ortstyp	Kann nur gelesen werden	Ja
Code	String	Nr. des Ortstyp	Kann nur gelesen werden	Ja
PredefinedShortcut*	String	Vordefinierter Wert für die Nr. der Codierung der Orte	Kann nur gelesen werden	Nein

*Der «PredefinedShortcut» bestimmt in der FA BSAS die Kurzform für die AKS-Codierung der Instanzen von diesem Typ (Shortcut bei AKSLocationDTO).

Tab. 4.3 AKSLocationStructureDTO (Fachkatalog Struktur Ort)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Guid	Primarschlüssel	Kann nur gelesen werden	Ja
Name	String	Bezeichnung des Ortstyps in der Struktur	Kann nur gelesen werden	Ja
Shortcut	String	Kurzform des Ortstyps in der Struktur	Kann nur gelesen werden	Ja
Code	String	Nr. des Ortstyps in der Struktur	Kann nur gelesen werden	Ja
PredefinedShortcut	String	Vordefinierter Wert für die Nr. der Codierung der Orte	Kann nur gelesen werden	Nein
ChildrenId	Guid[]	Liste der Kinder Id's	Kann nur gelesen werden	Ja
ParentId	Guid	Die Id des Elternobjektes	Kann nur gelesen werden	Ja
TypeId	Guid	Ort Typ Id	Kann nur gelesen werden	Ja
Level	Int	Ebene in der hierarchischen Struktur	Kann nur gelesen werden	Ja

Abb. 4.4 AKSComponentTypeDTO (Fachkatalog Typen Zugehörigkeit)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Guid	Primarschlüssel	Kann nur gelesen werden	Ja
Name	String	Bezeichnung des Zugehörigkeitstyp	Kann nur gelesen werden	Ja
Shortcut	String	Kurzform des Zugehörigkeitstyp	Kann nur gelesen werden	Ja
Code	String	Nr. des Zugehörigkeitstyp	Kann nur gelesen werden	Ja
PredefinedShortcut*	String	Vordefinierter Wert für den Freitext der Codierung der Zugehörigkeiten	Kann nur gelesen werden	Nein

*Der «PredefinedShortcut» bestimmt in der FA BSAS den Freitext für die AKS-Codierung der Instanzen von diesem Typ (Shortcut bei AKSComponentDTO).

Abb. 4.5 AKSComponentStructureDTO (Fachkatalog Struktur Zugehörigkeit)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Guid	Primarschlüssel	Kann nur gelesen werden	Ja
Name	String	Bezeichnung der Zugehörigkeitstypen in der Struktur	Kann nur gelesen werden	Ja
Shortcut	String	Kurzform der Zugehörigkeitstypen in der Struktur	Kann nur gelesen werden	Ja
Code	String	Nr. der Zugehörigkeitstypen in der Struktur	Kann nur gelesen werden	Ja
PredefinedShortcut	String	Vordefinierter Wert für den Freitext der Codierung der Zugehörigkeiten	Kann nur gelesen werden	Nein
ChildrenId	Guid[]	Liste der Kinder Id's	Kann nur gelesen werden	Ja
ParentId	Guid	Die Id des Elternobjektes	Kann nur gelesen werden	Ja
TypeId	Guid	Zugehörigkeitstyp Id	Kann nur gelesen werden	Ja
Level	Int	Ebene in der hierarchischen Struktur	Kann nur gelesen werden	Ja

Abb. 4.6 OSETypeDTO (Fachkatalog BSA Typ)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Guid	Primarschlüssel	Kann nur gelesen werden	Ja
Name	String	Bezeichnung des BSA Typs	Kann nur gelesen werden	Ja
Shortcut	String	Kurzform der Bezeichnung des BSA Typs	Kann nur gelesen werden	Ja
Properties	Objekt[]	Typenspezifische Eigenschaften	PropertyDefinitionDTO[] Kann nur gelesen werden	Ja

Abb. 4.7 OSEStructureDTO (Fachkatalog BSA Struktur)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Guid	Primarschlüssel	Kann nur gelesen werden	Ja
Name	String	Bezeichnung des BSA Typen in der Struktur	Kann nur gelesen werden	Ja
Shortcut	String	Kurzform des BSA Typen in der Struktur	Kann nur gelesen werden	Ja
ChildrenId	Guid[]	Liste der Kinder Id's	Kann nur gelesen werden	Ja
ParentId	Guid	Die Id des Elternobjektes	Kann nur gelesen werden	Ja
TypeId	Guid	BSA Typ Id	Kann nur gelesen werden	Ja
Level	Int	Ebene in der hierarchischen Struktur	Kann nur gelesen werden	Ja
SortIndex	Int	Die Reihenfolge innerhalb der Geschwister	Kann nur gelesen werden	Ja
AverageLifeCycle	Int	Defaultwert für die theoretische Nutzungsdauer in Jahren	Kann nur gelesen werden	Ja
PurchasePrice	Decimal	Defaultwert für den Beschaffungspreis	Kann nur gelesen werden	Ja

Abb. 4.8 LocationTypeDTO (Fachkatalog Ortstyp BSAS)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Guid	Primärschlüssel, Guid (Global Unique Identifier)	Kann nur gelesen werden	Ja
Name	String	Bezeichnung des Ortstyp	Kann nur gelesen werden	Ja
Shortcut	String	Kurzform der Bezeichnung des Ortstyp	Kann nur gelesen werden	Ja
Properties	Objekt[]	Typenspezifische Eigenschaften	PropertyDefinitionDTO[] Kann nur gelesen werden	Ja

Abb. 4.9 LocationStructureDTO (Fachkatalog Ortsstruktur BSAS)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Guid	Primärschlüssel	Kann nur gelesen werden	Ja
Name	String	Bezeichnung des Ort Type in der Struktur	Kann nur gelesen werden	Ja
Shortcut	String	Kurzform des Ort Typen in der Struktur	Kann nur gelesen werden	Ja
ChildrenId	Guid[]	Liste der Kinder Id's	Kann nur gelesen werden	Ja
ParentId	Guid	Id des Elternobjektes	Kann nur gelesen werden	Ja
TypeId	Guid	Ort Typ Id	Kann nur gelesen werden	Ja
Level	Int	Ebene in der hierarchischen Struktur	Kann nur gelesen werden	Ja
SortIndex	Int	Die Reihenfolge innerhalb der Geschwister	Kann nur gelesen werden	Ja

Abb. 4.10 LaneDTO (Fachkatalog Streifen)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Guid	Primärschlüssel	Kann nur gelesen werden	Ja
Name	String	Bezeichnung des Streifens	Kann nur gelesen werden	Ja
Shortcut	String	Kurzform der Bezeichnung des Streifens	Kann nur gelesen werden	Ja

Abb. 4.11 PropertyDefinitionDTO (Fachkatalog typenspezifische Eigenschaften)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Guid	Primärschlüssel	Kann nur gelesen werden	Ja
Shortcut	String	Textueller Schlüssel der typenspezifischen Eigenschaft.	Maximale Länge: 40 Zeichen Kann nur gelesen werden	Ja
Name	String	Der Name bzw. die Bezeichnung der typenspezifischen Eigenschaft	Maximale Länge: 1024 Zeichen Kann nur gelesen werden	Ja
ValueType	{Boolean, String, Integer, Date, Float, List, Multilist, Tri-State}	Typ der Eigenschaft	Kann nur gelesen werden	Ja
Listname	String	Name der Auswahlliste für die gültigen Werte, wenn ValueType = List oder Multilist ist	Kann nur gelesen werden Maximale Länge: 1024 Zeichen	Ja
DefaultValue	String	Standardwert	Bei mehreren Werten (gilt wenn Typ = Multilist) ist es ein kommagetrennter String. Maximale Länge: 40 Zeichen Kann nur gelesen werden	Ja

Abb. 4.12 GenericListDTO (Fachkatalog Auswahlliste)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Guid	Primärschlüssel	Kann nur gelesen werden	Ja
Name	String	Name der Auswahlliste	Maximale Länge: 1024 Zeichen Kann nur gelesen werden	Ja
Values	Objekt	Werte der Auswahlliste	GenericListValueDTO[] Kann nur gelesen werden	Ja

Abb. 4.13 GenericListValueDTO (Fachkatalog Listeneinträge)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Guid	Primärschlüssel	Kann nur gelesen werden	Ja
Name	String	Bezeichnung des Listeneintrages	Maximale Länge: 40 Zeichen Kann nur gelesen werden	Ja
Shortcut	String	Schlüssel bzw. Kurzform des Listeneintrages	Maximale Länge: 1024 Zeichen Kann nur gelesen werden	Ja

4.1.2 Verzeichnisse

Es wird unterschieden zwischen folgenden Verzeichnisdaten (Inventardaten)

- Verzeichnis Ort (AKSLocation)
- Verzeichnis Zugehörigkeit (AKSComponent)
- Verzeichnis BSA (OSEInventory)
- Verzeichnis Ort BSAS (Location)

Nachfolgend sind die einzelnen Data Transfer Objects (DTO) der Verzeichnisdaten (Inventardaten) genauer beschrieben. Die Struktur-DTO's besitzen eine hierarchische Gliederung. Deshalb haben sie auch einen Verweis auf das Elternobjekt (ParentId) und eine Liste der Kinderobjekte (ChildrenId) und die Ebene (Level) auf welchem sie sich in der hierarchischen befinden.

Abb. 4.14 AKSLocationDTO (Verzeichnis Ort)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Nullable Guid	Primärschlüssel	Für neue Orts-Objekte kann der Wert „leer“ sein	Ja
Name	String	Bezeichnung des Orts	Maximale Länge: 255 Zeichen	Ja
Shortcut	String	Nr. für Codierung des Orts	Maximale Länge: Abhängig von Ebene der AKSLocation: <ul style="list-style-type: none"> • Ebene 1: 30 Zeichen³ • Ebene 2: 4 Zeichen • Ebene 3: 5 Zeichen • Ebene 4: 3 Zeichen Nur Zahlen, Grossbuchstaben Wert muss eindeutig sein Kann „leer“ sein	Nein
StructureId	Guid	Id der Ortstruktur	Wert muss ein Kind der Struktur des Elternobjekts sein	Ja
ChildrenId	Guid[]	Liste der Kinder Id's	Kann nur gelesen werden	Nein
ParentId	Guid	Id des Elternobjektes		Ja
TerritorialUnits	String[]	Liste der Standort-Gebiete	{I, II, III, IV, V, VI, VII, VIII, IX, X, XI} Elternobjekt muss den Wert auch besitzen	Ja
LV03	Objekt	Schweizer Landeskoordinaten LV03		Nein
X	Int	X-Achse	Falls LV03 nicht NULL, muss der Wert zwischen 50000 und 350000 liegen.	Ja
Y	Int	Y-Achse	Falls LV03 nicht NULL, muss der Wert zwischen 450000 und 850000 liegen.	Ja
Z	Int	Z-Achse	Kann nur gelesen werden	Nein
ReadOnly	Nullable Boolean	Angabe, ob das Orts-Objekt in der FA BSAS readonly ist oder nicht.		Nein
RBBSCode	String	RBBS Codierung	Maximale Länge: 255 Zeichen, Kann „leer“ sein	Nein

³ Für AKSLocations der Ebene 1 wird in der FA BSAS der Shortcut nicht verwendet/eingeblendet. Eine Vergabe des Shortcuts ist allerdings grundsätzlich möglich.

Abb. 4.15 AKSComponentDTO (Verzeichnis Zugehörigkeit)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Nullable Guid	Primärschlüssel	Für neue Zugehörigkeits-Objekte kann der Wert „leer“ sein	Ja
Name	String	Bezeichnung der Zugehörigkeit	Maximale Länge: 255 Zeichen	Ja
Shortcut	String	Freitext für Codierung der Zugehörigkeit	Maximale Länge: Abhängig von Ebene der AKSComponent: <ul style="list-style-type: none"> • Ebene 1: 11 Zeichen • Ebene 2: 11 Zeichen • Ebene 3: 7 Zeichen Nur Zahlen, Grossbuchstaben, Bindestrich, Unterstrich Wert muss eindeutig sein	Ja
StructureId	Guid	Id der AKS Zugehörigkeitsstruktur	Wert muss ein Kind der Struktur des Elternobjekts sein	Ja
ChildrenId	Guid[]	Liste der Kinder Id's	Kann nur gelesen werden	Nein
ParentId	Guid	Id des Elternobjektes		Ja
TerritorialUnits	String[]	Liste der Standort-Gebiete	{I, II, III, IV, V, VI, VII, VIII, IX, X, XI} Elternobjekt muss den Wert auch besitzen	
ReadOnly	Nullable Boolean	Angabe, ob das Zugehörigkeits-Objekt in der FA BSAS readonly ist oder nicht.		Nein

Abb. 4.16 OSEInventoryDTO (Verzeichnis BSA)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Nullable Guid	Primärschlüssel	Für neue BSA-Objekte kann der Wert „leer“ sein	Ja
Name	String	Bezeichnung der BSA	Maximale Länge: 255 Zeichen	Ja
Shortcut	String	Kurzform der Bezeichnung der BSA	Maximale Länge: 30 Zeichen Wert muss eindeutig sein Kann „leer“ sein	Ja
CodeNumber	String	Nr. für Codierung der BSA	Maximale Länge: 10 Zeichen OSEInventory muss Level = 3 sein.	Nein
AKSCode	String	Bestehender, kantonaler AKS Code, falls vorhanden	Maximale Länge: 255 Zeichen	Nein
StructureId	Guid	Id der BSA Struktur	Wert muss ein Kind der Struktur des Elternobjekts sein	Ja
AKSLocationId	Guid	ID des Installationsortes	AKSLocation muss Level = 4 sein. OSEInventory muss Level = 3 sein.	Nein
AKSComponentId	Guid	ID, zu welcher Zugehörigkeit die BSA gehört	AKSComponent muss Level = 3 sein. OSEInventory muss Level = 3 sein.	Nein
LocationId	Guid	ID des Installationsortes BSAS		Nein
ChildrenId	Guid[]	Liste der Kinder Id's	Kann nur gelesen werden	Nein
ParentId	Guid	Id des Elternobjektes	Die Id des Elternobjekts	Ja
Level	Int	Ebene in der hierarchischen Struktur	Kann nur gelesen werden	Nein
SortIndex	Int	Reihenfolge innerhalb der Geschwister	Kann nur gelesen werden	Nein
Cantons	String[]	Liste der Kantone	{AG, AI, AR, BE, BL, BS, FR, GL ...} Wert darf nicht „leer“ sein Elternobjekt muss den Wert auch besitzen	Ja
TerritorialUnits	String[]	Liste der Standort-Gebiete	{I, II, III, IV, V, VI, VII, VIII, IX, X, XI} Wert darf nicht „leer“ sein Elternobjekt muss den Wert auch besitzen	Ja
GeoDataFromTo	Objekt	Geographische Informationen der BSA	GeoDataFromToDTO	Ja
Data	Objekt	Technischen Daten der BSA	OSDataDTO	Ja
ReadOnly	Nullable Boolean	Angabe, ob das BSA-Objekt in der FA BSAS readonly ist oder nicht.		Nein
Properties	Objekt[]	Alle typenspezifischen Eigenschaften eines BSA-Objekts	PropertyValueDTO[]	Nein
RBBSCode	String	RBBS Codierung	Maximale Länge: 255 Zeichen, Kann „leer“ sein OSEInventory muss Level = 3 sein.	Nein

Abb. 4.17 LocationDTO (Verzeichnis Ort BSAS)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Nullable Guid	Primärschlüssel	Für neue Orts-Objekte kann der Wert „leer“ sein	Ja
Name	String	Bezeichnung des Orts BSAS	Maximale Länge: 255 Zeichen	Ja
Shortcut	String	Kurzform der Bezeichnung des Orts BSAS	Maximale Länge: 15 Zeichen Wert muss eindeutig sein Kann „leer“ sein	Ja
AKSCode	String	Bestehender, kantonaler AKS Code, falls vorhanden	Maximale Länge: 255 Zeichen	Nein
StructureId	Guid	Id der Ortstruktur BSAS	Wert muss ein Kind der Struktur des Elternobjekts sein	Ja
ChildrenId	Guid[]	Liste der Kinder Id's	Kann nur gelesen werden	Nein
ParentId	Guid	Id des Elternobjektes		Ja
Level	Int	Ebene in der hierarchischen Struktur	Kann nur gelesen werden	Nein
SortIndex	Int	Reihenfolge innerhalb der Geschwister	Kann nur gelesen werden	Nein
Cantons	String[]	Liste der Kantone	{AG, AI, AR, BE, BL, BS, FR, GL ...} Elternobjekt muss den Wert auch besitzen	Ja
TerritorialUnits	String[]	Liste der Standort-Gebiete	{I, II, III, IV, V, VI, VII, VIII, IX, X, XI} Elternobjekt muss den Wert auch besitzen	Ja
GeoDataFromTo	Objekt	Geographischen Informationen des Orts BSAS	GeoDataFromToDTO	Ja
ReadOnly	Nullable Boolean	Angabe, ob das Orts-Objekt in der FA BSAS readonly ist oder nicht.		Nein
Properties	Objekt[]	Alle typenspezifischen Eigenschaften eines Orts-Objekts	PropertyValueDTO[]	Nein

4.1.3 Eigenschaften

Es gibt drei Gruppen von Eigenschaften. Die typenspezifischen und geographischen Informationen, die sowohl bei Orte BSAS als auch bei BSA vorkommen und die technischen Daten, die nur bei BSA vorkommen.

Abb. 4.18 Verwendung der Eigenschaften

	Typenspezifische Informationen	Geographische Informationen	Technische Daten
Ort BSAS-Entität	x	x	
BSA-Entität	x	x	x

Im Folgenden werden die typenspezifischen Informationen (PropertyValue), geographischen Informationen (GeoDataFromTo) und technischen Daten (OSEData) genauer beschrieben:

Abb. 4.19 PropertyValueDTO (Wert der Eigenschaft)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Key	String	Der Name bzw. die Bezeichnung der typenspezifischen Eigenschaft	Maximale Länge: 1024 Zeichen	Ja
Value	String	Wert der Eigenschaft. Bei mehreren Werten (z.B. Multilist) ist es ein kommagetrennter String.	Maximale Länge: 255 Zeichen Kann „leer“ sein	Ja

Abb. 4.20 GeoDataFromToDTO (Geografische Informationen)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
InventoryId	Guid	BSA oder Ort Id	Entweder BSA Id oder Ort Id	Ja
LaneId	Guid	Streifen Id	Streifen Id Kann NULL sein	Nein
From	Objekt	Geographische Informationen „Von“		Nein
CrossSection	Int	Querschnitt (Kilometrierung)	Zwischen 0 und 999999	Nein
LV03	Objekt	Schweizer Landeskoordinaten LV03		Nein
X	Int	X-Achse	Falls LV03 nicht NULL, muss der Wert zwischen 50000 und 350000 liegen.	Ja
Y	Int	Y-Achse	Falls LV03 nicht NULL, muss der Wert zwischen 450000 und 850000 liegen.	Ja
Z	Int		Kann nur gelesen werden	Nein
RBBS*	Objekt	RBBS Koordinaten		Nein
Id	String	Id des RBBS Referenzpunktes	Maximale Länge: 255 Zeichen	Nein
U	Float	Meter in Richtung RBBS Achse (aufsteigende Kilometrierung)		Nein
V	Float	Meter, rechtwinklig zur RBB Achse		Nein
W	Float	Meter ab Fahrbahnebene		Nein
To	Objekt	Geographische Informationen „Bis“		Nein
CrossSection	Int	Querschnitt (Kilometrierung)	Zwischen 0 und 999999	Nein
LV03	Objekt	Schweizer Landeskoordinaten LV03		Nein
X	Int	X-Achse	Falls LV03 nicht NULL, muss der Wert zwischen 50000 und 350000 liegen.	Ja
Y	Int	Y-Achse	Falls LV03 nicht NULL, muss der Wert zwischen 450000 und 850000 liegen.	Ja
Z	Int		Kann nur gelesen werden	Nein
RBBS*	Objekt	RBBS Koordinaten		Nein
Id	String	Id des RBBS Referenzpunktes	Maximale Länge: 255 Zeichen	Nein
U	Float	Meter in Richtung RBBS Achse (aufsteigende Kilometrierung)		Nein
V	Float	Meter, rechtwinklig zur RBB Achse		Nein
W	Float	Meter ab Fahrbahnebene		Nein

*Diese RBBS-Informationen sind in der FA BSAS zwar noch vorhanden, sind jedoch veraltet und werden nicht verwendet. Der RBBS werden in AKSLocationDTO.RBBSCode und OSEInventoryDTO.RBBSCode exponiert.

Abb. 4.21 OSEDataDTO (Technische Eigenschaften von BSA)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
InventoryId	Guid	BSA Id (Guid)	BSA Id	Ja
AvarageLifeCycle	Int	Bewertete Nutzungsdauer (Jahre)	Zwischen 0 und 100	Nein
CommissioningDate	Datum	Beginn der Garantie (Datum)	Zwischen 01.01.1753 und 31.12.2100	Nein
Contractor	String	Werkhersteller	Maximale Länge: 255 Zeichen Kann „leer“ sein	Ja
Details	String	Kommentar (Freitext)	Maximale Länge: 1024 Zeichen Kann „leer“ sein	Ja
GuaranteeExpiryDate	Datum	Ende der Garantie (Datum)	Zwischen 01.01.1753 und 31.12.2100	Read-Only
InventoryPlannumber	String	Inventarplannummer	Maximale Länge: 255 Zeichen Kann „leer“ sein	Nein
Manufacturer	String	Komponenten Hersteller	Maximale Länge: 255 Zeichen Kann „leer“ sein	Ja
MaintenanceCompany	String	Wartungsfirma	Maximale Länge: 255 Zeichen Kann „leer“ sein Existiert in FA BSAS nicht mehr. Nur zur Rückwärtskompatibilität vorhanden	Nein
Multiplicity	Int	Anzahl	Zwischen 0 und 999999	Ja
ProductNumber	String	Produkttyp	Maximale Länge: 255 Zeichen	Nein
PurchasePrice	Float	Beschaffungspreis (CHF/Stk.)	Zwischen 0 und 999999999999999	Nein
SerialNumber	String	Seriennummer	Maximale Länge: 255 Zeichen Kann „leer“ sein Existiert in FA BSAS nicht mehr. Nur zur Rückwärtskompatibilität vorhanden	Nein
YearOfConstruction	Int	Inbetriebnahmejahr	Zwischen 1753 und 2100	Nein

Projekte

In der FA BSAS können Projekte bzw. Projekträume aufgesetzt und Inventardaten zugeordnet werden. Dadurch werden die Inventardaten in der FA BSAS zu Bearbeitung gesperrt. Diese Funktionalität wird in erster Linie verwendet, um bei bidirektionaler Schnittstellennutzung die Datenhoheit sicherzustellen (vergl. Kapitel 5.4). Um die Projekte bzw. die Projektdaten abzufragen, sind folgende DTO's vorhanden.

Abb. 4.22 ProjectDTO (Projektraum)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
ID	Guid	Primärschlüssel	Kann nur gelesen werden	Ja
OSReplicas	Guid[]	Liste aller Ids von dem Projekt zugewiesenen BSA	Kann nur gelesen werden	Nein
LocationReplicas	Guid[]	Liste aller Ids von dem Projekt zugewiesenen Orte BSAS	Kann nur gelesen werden	Nein
AKSLocationReplicas	Guid[]	Liste aller Ids von dem Projekt zugewiesenen Orte	Kann nur gelesen werden	Nein
AKSComponentReplicas	Guid[]	Liste aller Ids von dem Projekt zugewiesenen Zugehörigkeiten	Kann nur gelesen werden	Nein
StartDate	DateTime	Startdatum des Projektes	Kann nur gelesen werden	Nein
EndDate	DateTime	Enddatum des Projektes	Kann nur gelesen werden	Nein
State	Objekt	Zustand des Projektes	ProjectStateDTO	Ja

Abb. 4.23 ProjectStateDTO (Projektstatus)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
State	Enum	Zustand des Projektes	{Preparation = 0, Started = 1, Stopped = 2, Terminated = 3, Archived = 4, Cancelled = 5, Deleted = 6}	Ja

4.2 Zustandsdaten

Die Zustandsbewertung von BSA erfolgt in der FA BSAS nach der vom ASTRA definierten Methodologie [3]. Das Datenmodell ist in Abbildung 4.24 dargestellt. Für jede BSA (OSEInventory) können grundsätzlich Bewertungen (Evaluation) innerhalb der Bewertungssysteme (EvaluationSystem) erstellt werden.

Wie im Datenmodell der Inventardaten geben die Farben der Entitäten in der Abbildung 4.24 Auskunft über die erwartete Änderungshäufigkeit der Daten. Die rot eingefärbten Entitäten sind fast konstant, die Änderungshäufigkeit ist sehr klein. Diese Daten werden auch „Katalogdaten“ genannt. Die blau eingefärbten Entitäten haben eine mittlere bis hohe Änderungshäufigkeit.



Abb. 4.24 Datenmodell Zustandsdaten.

Jede Bewertung (Evaluation) gehört genau zu einem Bewertungssystem (EvaluationSystem) und besitzt ein oder mehrere Zustandswerte (Score), welche in einer getrennten Entität gespeichert werden.

4.2.1 Katalogdaten

Als Katalogdaten gelten die Bewertungssysteme (EvaluationSystem). Ein Bewertungssystem definiert, in welcher Granularität die Zustandsbewertung erfolgt und welche Bewertungskriterien dabei berücksichtigt werden müssen. Die Bewertungssysteme bilden sozusagen die Vorgaben der Zustandsbewertungen ab. Es ist allerdings zu beachten, dass die einzelnen Bewertungssysteme sowie die darin enthaltenen Angaben hinsichtlich Granularität und Bewertungskriterien nicht als statisch angesehen können. Die Änderungshäufigkeit von diesen Daten ist zwar gering, jedoch lassen sich die Bewertungssysteme in der FA BSAS nach Bedarf konfigurieren.

Abb. 4.25 EvaluationSystemDTO (Fachkatalog Bewertungssysteme)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Guid	Id des Bewertungssystem in BSAS	Kann nur gelesen werden	Ja
Name	String	Name des Bewertungssystems	Kann nur gelesen werden	Ja
InventoryLevel	String	Angabe auf welchen Ebenen (Anlage, Teilanlage, Aggregat) innerhalb des Bewertungssystems Bewertungen zulässig sind	Kann nur gelesen werden 1 = Anlage 2 = Teilanlage 3 = Aggregat	Ja
Criteria	String[]	Liste der Beurteilungskriterien, die innerhalb des Bewertungssystems bewertet werden können.	Kann nur gelesen werden	Ja

4.2.2 Zustandsbewertungen

Die Zustandsbewertungen teilen sich in zwei Entitäten: Evaluations und Scores. Die Bewertung (Evaluation) bildet sozusagen den Rahmen für die effektiven Zustandswerte (Score). Für eine BSA können in der FA BSAS mehrere Bewertungen innerhalb eines Bewertungssystems gespeichert werden, die Voraussetzung ist die Angabe eines eindeutigen Datums.

Abb. 4.26 EvaluationDTO (Bewertung)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
Id	Guid	Id der Bewertung	Für neue Bewertungen kann der Wert eine leere GUID (GUID.Empty) sein	Ja
InventoryId	Guid	Id der BSA (OSEInventory)	Level der BSA muss mit InventoryLevel des Bewertungssystems übereinstimmen.	Ja
EvaluationSystemId	Guid	Id des Bewertungssystem		Ja
InspectionDate	Date	Datum der Zustandserfassung/Bewertung	Muss für jede BSA innerhalb des Bewertungssystems eindeutig (unique) sein.	Ja
Scores	Objekt[]	Eine Liste von allen Kriterien samt Note	ScoreDTO[] Kann „leer“ sein	Ja

Abb. 4.27 ScoreDTO (Zustandswert)

Eigenschaft	Typ	Beschreibung	Gültigkeitsbereich	Nötig
CriterionName	String	Die Bezeichnung des Beurteilungskriteriums	Maximale Länge: 1024 Zeichen	Ja
Value	Float?	Die vergebene Zustandsnote	Wert darf NULL sein. Wert muss zwischen 1 und 5 liegen. ⁴	Ja
Comment	String	Kommentar/Begründung der Zustandsnotenvergabe	Maximale Länge: 512 Zeichen	Ja ⁵

⁴ Die Min- und Max-Werte können in der FA BSAS konfiguriert werden. Daher ist es möglich, dass die Angaben in diesem Dokument nicht aktuell sind. Die Benotung zwischen 1 und 5 entspricht allerdings den Vorgaben des ASTRA (siehe Methodologie [3]).

⁵ Ob ein Kommentar/Begründung zur Note notwendig ist, ist abhängig von der Definition des Bewertungssystems in der FA BSAS.

5 Webservice

5.1 Datenübertragungsobjekt

Die Webservices benutzen Datenübertragungsobjekte. Deswegen besitzen alle Entitäten der Webservices den Suffix DTO (Data Transfer Object). Die Entitäten werden gemäss der Spracheinstellung des Benutzers lokalisiert, d.h. die Namen und Abkürzungen werden in der richtigen Sprache (D, F, I) zurückgegeben.

5.2 Datengültigkeit

Fast alle Attribute sind eingeschränkt, bzw. haben einen Gültigkeitsbereich (siehe einzelne DTO's). Das System prüft die Gültigkeit jedes Attributs und wirft eine Ausnahme (Exception), wenn der Wert ungültig ist.

Die Prüfungen der Datengültigkeitsbereiche betreffen unter anderem folgende Bereiche:

- Bestimmte „Länge“ oder „Spanne“ des Werts
- Eindeutigkeit der Abkürzungen (Shortcut), sie müssen unter allen Geschwistern mit dem gleichen Elternobjekt und derselben StructureId eindeutig sein

Zu gewissen Attributen müssen zudem zwingend Werte angegeben werden. Das heisst, dass der Wert des Attributs nicht „NULL“ oder „leer“ sein darf, wenn nicht anders spezifiziert ist.

5.3 Authentifikation

Um auf den Webservice zugreifen zu können wird ein Account bzw. Benutzer mit der Rolle „Schnittstellennutzer“ benötigt. Zudem werden je nach gewünschter Funktion weitere Rollen benötigt:

- Inventardaten lesen: BSA Betrachter
- Inventardaten lesen & schreiben: BSA Erfasser
- Projektdaten lesen & schreiben: Projekt Bearbeiter

Zustandsdaten lesen & schreiben: BSA Inspektor

Die Authentifikationsangaben (der Benutzer und das Passwort) werden bei jeder Anfrage in dem SOAP Header mitgeschickt.

5.4 Konzept & Beschreibung Webservices

Um die weiteren Kapitel besser zu verstehen wird an dieser Stelle das Konzept der Schnittstelle kurz beschrieben.

Für das Lesen und Schreiben von Daten stehen heute drei Webservices zur Verfügung:

- **LiveInventory-Webservice**
Der LiveInventory-Webservice erlaubt das Lesen & Bearbeiten der Live-Inventardaten. Dieser Service wird grundsätzlich bei einer unidirektionalen Schnittstelle verwendet. Sprich: Die Inventardaten werden entweder nur bezogen oder nur in die FA BSAS geschrieben.
- **Projectroom-Webservice**
Für den Fall, dass zwischen der FA BSAS und einem Schnittstellennutzer sowohl Inventardaten bezogen, bearbeitet und zurückgeschrieben werden (bidirektionale Schnittstelle), gibt es den Projectroom-Webservice. Über diesen Service werden nicht direkt die Live-Inventardaten, sondern eine Kopie (Replica) bearbeitet.
Beim bidirektionalen Austausch der Inventardaten muss zwingend die Datenhoheit geregelt werden, da sonst allfällige Änderungen in der FA BSAS über die Schnittstelle

unkoordiniert überschrieben werden. Dafür existieren sogenannte Projekte bzw. Projekträume. Die Live-Inventardaten werden einzelnen Projekträumen zugeordnet, wodurch die Datenbearbeitung in der FA BSAS gesperrt und die Projektdaten (Replicas) im Projektraum angelegt werden. Der Schnittstellennutzer bearbeitet danach die Projektdaten im Projektraum über die Schnittstelle. Sobald die Bearbeitung abgeschlossen ist, wird der Projektraum geschlossen. Dabei werden die Informationen von den Projektdaten auf die Live-Inventardaten übertragen.

- **InventoryEvaluation-Webservice**
- Der InventoryEvaluation erlaubt das Lesen & Bearbeiten der Zustandsdaten zu den vorhandenen BSA.

In der folgenden Abbildung ist das Konzept vereinfacht dargestellt:

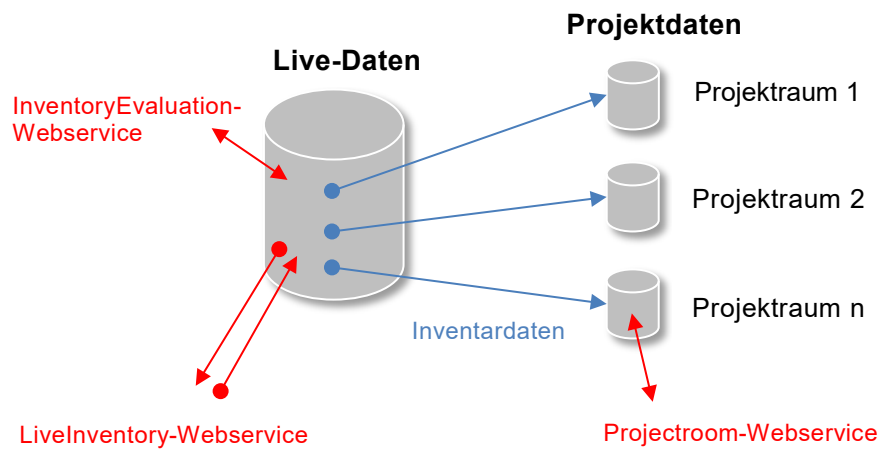


Abb. 5.1 Konzept Schnittstelle / Webservices

5.4.1 Inventardaten (LiveInventory-Webservice)

Die URL des Webservices lautet:

- **Live-Umgebung**
<http://www.bsa-ch.ch/webservices/GEVIII/soap/LiveInventory.svc>
- **Test-Umgebung**
<http://test.bsa-ch.ch/webservices/GEVIII/soap/LiveInventory.svc>

Die WSDL Datei, die den Service beschreibt, ist erreichbar über die URL <http://www.bsa-ch.ch/webservices/GEVIII/soap/LiveInventory.svc?wsdl>

5.4.1.1 Methoden für Inventardaten

Die ersten 8 aufgeführten Methoden im SOAP Format in der Abbildung 5.2 sind für das Lesen der Katalogdaten. Die restlichen Methoden können für das Lesen, Bearbeiten und Ergänzen der Verzeichnisdaten (BSA und Orte) verwendet werden.

Abb. 5.2 SOAP LiveInventory Webservice

Nr.	Methode	Antwort
1	GetLane(Guid id)	LaneDTO[]
2	GetOSEType(Guid id)	OSETypeDTO[]
3	GetLocationType(Guid id)	LocationTypeDTO[]
4	GetOSEStructure(Guid id, int depth)	OSEStructureDTO
5	GetLocationStructure(Guid id, int depth)	LocationStructureDTO
6	GetListValues(String listname)	GenericListValueDTO[]
7	GetAllLists()	GenericListDTO[]
8	GetPropertyTypes ()	String[]
9	GetOSEInventory(Guid id, int depth)	OSEInventoryDTO
10	GetLocation(Guid id, int depth)	LocationDTO
11	UpdateOSData(OSEDataDTO data)	OSEDataDTO
12	UpdateGeoDataFromTo(GeoDataFromToDTO geoData)	GeoDataFromToDTO
13	CreateOSEInventory(OSEInventoryDTO ose)	OSEInventoryDTO
14	UpdateOSEInventory(OSEInventoryDTO ose)	OSEInventoryDTO
15	DeleteOSEInventory (Guid oseld)	bool
16	CreateLocation (LocationDTO location)	LocationDTO
17	UpdateLocation (LocationDTO location)	LocationDTO
18	DeleteLocation (Guid locationId)	bool
19	GetOSEShortcuts(Guid parentOseld, Guid childStructureId)	String[]
20	GetLocationShortcuts(Guid parentLocationId, Guid childStructureId)	String[]
21	GetOSENames(Guid parentOseld, Guid childStructureId)	String[]
22	GetLocationNames(Guid parentLocationId, Guid childStructureId)	String[]
23	GetAKSLocationType(Guid id)	AKSLocationTypeDTO[]
24	GetAKSComponentType(Guid id)	AKSComponentTypeDTO[]
25	GetAKSLocationStructure(Guid id, int depth)	AKSLocationStructureDTO
26	GetAKSComponentStructure(Guid id, int depth)	AKSComponentStructureDTO
27	GetAKSLocation(Guid id)	AKSLocationDTO
28	CreateAKSLocation(AKSLocationDTO AKSLocation)	AKSLocationDTO
29	UpdateAKSLocation(AKSLocationDTO AKSLocation)	AKSLocationDTO
30	DeleteAKSLocation(Guid AKSLocationId)	bool
31	GetAKSComponent(Guid id)	AKSComponentDTO
32	CreateAKSComponent(AKSComponentDTO AKSComponent)	AKSComponentDTO
33	UpdateAKSComponent(AKSComponentDTO AKSComponent)	AKSComponentDTO
34	DeleteAKSComponent(Guid AKSComponentId)	bool

Zu Methoden 1 – 3, 23, 24:

Bei diesen Methoden kann als Parameter eine leere Id (Guid 00000000-0000-0000-0000-000000000000) eingegeben werden. In diesem Fall wird eine Liste aller vorhandenen Entitäten zurückgegeben. Wenn eine spezifische Id eingegeben wird, wird nur die Entität der entsprechenden Id zurückgegeben.

Zu Methoden 4, 5, 25, 26:

Die Struktur für BSA und Orte sind hierarchisch organisiert, deswegen wird keine Liste zurückgegeben, sondern nur eine Entität. Wenn eine leere Id (Guid 00000000-0000-0000-0000-000000000000) eingegeben wird, wird die Wurzel (oberste Ebene) der hierarchischen Struktur zurückgegeben. Wenn eine spezifische Id eingegeben wird, wird nur die Entität der entsprechenden Id zurückgegeben.

Zu Methode 6:

Für typenspezifische Eigenschaften vom Typ „List“ und „MultiList“ sind in der FA BSAS Auswahllisten hinterlegt. Der Listenname wird auf Deutsch angegeben. Wenn diese Liste gefunden wird, werden die Elemente der Liste zurückgegeben.

Zu Methode 7:

Gibt alle vorhandene Listen zurück. Das ist vorteilhaft, wenn periodisch alle Listen abgefragt werden möchten.

Zu Methode 8:

Gibt alle vorhandenen Datentypen für die Eigenschaften zurück (z.B. Boolean, Float, String,...).

Nebenbedingung: Gross- Kleinschreibung bei Boolean/Tristate

- Boolean: True, False.
- Tristate: True, False, Unassigned

Zu Methoden 4,5,9,10:

Der Parameter „depth“ kontrolliert die Ebenen, welche zusammen mit der Entität zurückgegeben werden sollen.

Abb. 5.3 Parameterwert "depth" und Rückgabe der Methode

Parameterwert « depth »	Rückgabe
0	Nur die Entität selbst
1	Entität selbst + alle Kinder der 1. Ebene unterhalb der Entität
2	Entität selbst + alle Kinder der 2. Ebene unterhalb der Entität
...	
n	Entität selbst + alle Kinder der n-ten Ebene unterhalb der Entität
-1	Entität selbst + alle Kinder der 1. Ebene unterhalb bis und mit n-ten Ebene unterhalb der Entität
-2	Entität selbst + alle Kinder der 2. Ebene unterhalb bis und mit n-ten Ebene unterhalb der Entität
...	
-n	Entität + alle Kinder der n-ten Ebene unterhalb der Entität

Zu Methoden 11 und 12:

Die beiden Methoden sind für das Speichern der Mutationen von Eigenschaften vorhanden. Die übermittelten Daten werden validiert und im Fall eines Fehlers eine Ausnahme (Exception) geworfen.

Zu Methoden 13, 16, 28, 32

Legt eine neue Instanz an. Die übermittelten Daten werden validiert (Datengültigkeitsbereiche) und im Fall eines Fehlers eine Ausnahme (Exception) geworfen.

Zu Methoden 14, 17, 29, 33

Aktualisiert eine bestehende Instanz. Die übermittelten Daten werden validiert (Datengültigkeitsbereiche) und im Fall eines Fehlers eine Ausnahme (Exception) geworfen.

Zu Methoden 15, 18, 30, 34

Löscht die angegebene Instanz und im Fall eines Fehlers wird eine Ausnahme (Exception) geworfen.

Zu Methoden 19, 20

Gibt eine Liste von **Kurzformen** von allen Kindern mit dieser Struktur zurück, die schon verwendet worden sind.

Zu Methoden 21, 22

Gibt eine Liste von **Namen** von allen Kindern mit dieser Struktur zurück, die schon verwendet worden sind.

Zu Methoden 27, 31

Gibt den Ort (AKSLocationDTO) bzw. die Zugehörigkeit (AKSComponentDTO) zurück.

5.4.2 Projektdaten (Projectroom-Webservice)

Die URL des Webservices lautet:

- **Live-Umgebung**
http://www.bsa-ch.ch/webservices/GEVIII/soap/Projectroom.svc
- **Test-Umgebung**
http://test.bsa-ch.ch/webservices/GEVIII/soap/Projectroom.svc

Die WSDL Datei, die den Service beschreibt, ist erreichbar über die URL /soap/Projectroom.svc?wsdl

Der „ProjectRoom Webservice“ ist für das Lesen und die Bearbeitung der Projektdaten (vergl. Kapitel 5.4). Der „ProjectRoom Webservice“ und der „LiveInventory Webservice“ sind nicht vollständig getrennt. Einige Informationen in den Entitäten des einen Webservice referenzieren die Entitäten des anderen Webservice. Zum Beispiel referenziert die ReplicaId der OSEInventory und die Location Entitäten des „LiveInventory Webservice“ die Id derselben Entitäten in dem ProjectRoom Webservice.

Die OSEReplica und die LocationReplica Entitäten des ProjectRoom Webservice referenzieren die OSEStructureId beziehungsweise die LocationStructureId des LiveInventory Webservice.

5.4.2.1 Methoden für Projektdaten

Abb. 5.4 SOAP Projectroom Webservice

Nr.	Methode	Antwort
1	GetUserProject(Guid id)	ProjectDTO[]
2	SetProjectState(Guid id, ProjectStateDTO state)	bool
3	GetOSEReplica(Guid id, int depth)	OSEReplicaDTO
4	CreateOSEReplica(OSEReplicaDTO replica)	OSEReplicaDTO
5	UpdateOSEReplica(OSEReplicaDTO replica)	OSEReplicaDTO
6	DeleteOSEReplica(Guid replicaId)	bool
7	GetLocationReplica(Guid id, int depth)	LocationReplicaDTO
8	CreateLocationReplica(LocationReplicaDTO replica)	LocationReplicaDTO
9	UpdateLocationReplica(LocationReplicaDTO replica)	LocationReplicaDTO
10	DeleteLocationReplica(Guid id)	Bool
11	GetAKSLocationReplica(Guid id)	AKSLocationReplicaDTO
12	CreateAKSLocationReplica(AKSLocationReplicaDTO AKSLocation)	AKSLocationReplicaDTO
13	UpdateAKSLocationReplica(AKSLocationReplicaDTO AKSLocation)	AKSLocationReplicaDTO
14	DeleteAKSLocationReplica(Guid AKSLocationReplicaId)	bool
15	GetAKSComponentReplica (Guid id)	AKSComponentReplicaDTO
16	CreateAKSComponentReplica(AKSComponentReplicaDTO AKSComponent)	AKSComponentReplicaDTO
17	UpdateAKSComponentReplica (AKSComponentReplicaDTO AKSComponent)	AKSComponentReplicaDTO
18	DeleteAKSComponentReplica (Guid AKSComponentReplicaId)	bool

Zu Methode 1:

Gibt das Projekt mit der entsprechenden Id zurück. Wenn eine leere Id (Guid 00000000-0000-0000-0000-000000000000) als Parameter mitgegeben wird, dann werden alle Projekte des angemeldeten Users zurückgegeben.

Zu Methode 2:

Setzt den Projekt Status.

Zu Methode 3, 7, 11, 15:
Gibt das ReplicaDTO zurück.

Zu Methode 4, 8, 12, 16:
Legt ein neues ReplicaDTO an.

Zu Methode 5, 9, 13, 17:
Aktualisiert das bestehende ReplicaDTO.

Zu Methode 6, 10, 14, 18:
Löscht das ReplicaDTO und gibt an, ob es erfolgreich verlief.

5.4.3 Zustandsdaten (InventoryEvaluation-Webservice)

Die URL des Webservices lautet:

- **Live-Umgebung**
http://www.bsa-ch.ch/webservices/GEVIII/soap/ InventoryEvaluation.svc
- **Test-Umgebung**
http://test.bsa-ch.ch/webservices/GEVIII/soap/ InventoryEvaluation.svc

Die WSDL Datei, die den Service beschreibt, ist erreichbar über die URL /soap/InventoryEvaluation.svc?wsdl

5.4.3.1 Methoden für Zustandsdaten

Abb. 5.5 SOAP InventoryEvaluation Webservice

Nr.	Methode	Antwort
1	GetAllEvaluationSystems()	EvaluationSystemDTO[]
2	GetEvaluations(Guid inventoryId, Guid evaluationSystemId)	EvaluationDTO[]
3	GetEvaluation(Guid evaluationId)	EvaluationDTO
4	UpdateEvaluation(EvaluationDTO eval)	EvaluationDTO
5	AddEvaluation(EvaluationDTO eval)	EvaluationDTO

Das Löschen von Zustandsbewertungen ist in der FA BSAS nicht vorgesehen, weshalb es auch keine entsprechende Methode dazu gibt.

Zu Methode 1:
Es wird eine Liste der vorhandenen Bewertungssysteme zurückgegeben.

Zu Methode 2:
Es wird eine Liste der vorhandenen Bewertungen einer BSA zurückgegeben.

Wenn EvaluationSystemId NULL ist bzw. fehlt, werden alle Bewertungen einer BSA angezeigt, sonst werden nur die Bewertungen des eingegebenen Bewertungssystems angezeigt.

Zu Methode 3:
Es wird die vorhandene Bewertung zurückgegeben.

Zu Methode 4:
Eine bestehende Bewertung wird überschrieben. Es werden nur das Inspektionsdatum und die Bewertungen (Values) überschrieben, wenn die mitgelieferten Kriterien existieren und für das Bewertungssystem zulässig sind. Allgemeine Restriktionen beim Überschreiben von Bewertungen in der FA-BSAS sind auch hier gültig.

Zu Methode 5:
Eine neue Bewertung wird erzeugt und zurückgegeben. Eine neue Bewertung kann nur angelegt werden, wenn die BSA, das Bewertungssystem und die mitgelieferten Kriterien existieren und zulässig sind. Allgemeine Restriktionen beim Erzeugen von Bewertungen in der FA BSAS sind auch hier gültig.

6 Codebeispiele

6.1 Authentifizierung

```
var client = new LiveInventoryClient();  
var authentication = client.ClientCredentials.ServiceCertificate.Authentication;  
authentication.CertificateValidationMode = X509CertificateValidationMode.None;  
client.ClientCredentials.UserName.UserName = "username";  
client.ClientCredentials.UserName.Password = "passwort";
```

6.2 Inventardaten

6.2.1 Katalogdaten

Da es wenige Katalogdaten sind, ist es am einfachsten diese in einem Stück zu lesen:

```
var streifen = client.GetLane(Guid.Empty);  
var bsatypen = client.GetOSEType(Guid.Empty);  
var ortstypen = client.GetLocationType(Guid.Empty);
```

Die gesamte BSA- und Ortsstruktur können auch in einmal gelesen werden:

```
var osestructurewurzel = client.GetOSEStructure(Guid.Empty, -1);  
var ortstructurewurzel = client.GetLocationStructure(Guid.Empty, -1);
```

6.2.2 BSA- und Ortsverzeichnis BSAS

Die Wurzeln (oberste Ebene der Hierarchie) der Verzeichnisse können mit einer leeren Guid gelesen werden:

```
var bsawurzel = client.GetOSEInventory(Guid.Empty, 0);  
var ortswuzel = client.GetLocation(Guid.Empty, 0);
```

Die Kindern können geladen werden, indem man direkt den Parameter depth=1 eingibt oder die Kinder einzeln lädt:

```
var bsawurzel = client.GetOSEInventory(Guid.Empty, 1);  
var ortswuzel = client.GetLocation(Guid.Empty, 1);
```

oder

```
var bsawurzel = client.GetOSEInventory(Guid.Empty, 0);  
foreach (var id in bsawurzel.ChildrenId)  
    bsawurzel.Children.Add(client.GetOSEInventory(id, 0));
```

```
var ortswuzel = client.GetLocation(Guid.Empty, 0);  
foreach (var id in ortswuzel.ChildrenId)  
    ortswuzel.Children.Add(client.GetLocation(id, 0));
```

6.2.3 Eigenschaften

Die Eigenschaften müssen nicht getrennt gelesen werden, sie sind immer mit dem Objekt zusammen vorhanden. Wenn nur einige Eigenschaftswerte angepasst werden sollen, werden dem Webservice nur die geänderten Werte gesendet und nicht das ganze Objekt.

```
var bsaid = Guid.Parse("abcdefgh-1234-1234-1234-abcdefghijkl");
var bsa = client.GetOSEInventory(bsaid, 0);
bsa.Data.Details = "This is a demo";
bsa.Data.Multiplicity = 10;
client.UpdateOSEData(bsa.Data);
```

oder

```
var bsaid = Guid.Parse("abcdefgh-1234-1234-1234-abcdefghijkl");
var data = new OSEDataDTO();
data.InventoryId = bsaid;
data.Details = "This is a demo";
data.Multiplicity = 10;
client.UpdateOSEData(data);
```

Im ersten Beispiel werden die BSA Eingeschalten zuerst gelesen, danach geändert und am Ende gespeichert. Im folgenden Beispiel wird ein neues Objekt mit BSA Eigenschaften erstellt und gespeichert. Die Eigenschaftswerte, welche nicht angegeben sind, werden mit dem Default Wert gespeichert.

```
var ortid = Guid.Parse("12345678-abcd-efgh-ijkl-123456789012");
var ort = client.GetLocation(ortid, 0);
ort.GeoDataFromTo.From.CrossSection = 5000;
ort.GeoDataFromTo.To.CrossSection = 5500;
client.UpdateGeoDataFromTo(ort.GeoDataFromTo);
```

oder

```
var ortid = Guid.Parse("12345678-abcd-efgh-ijkl-123456789012");
var geo = new GeoDataFromToDTO();
geo.From = new GeoDataDTO();
geo.To = new GeoDataDTO();
geo.From.CrossSection = 5000;
geo.To.CrossSection = 5500;
client.UpdateGeoDataFromTo(geo);
```

Glossar

Begriff	Bedeutung
AKS-CH	Struktur und Kennzeichnung der Betriebs- und Sicherheitsausrüstungen (AKS-CH) Dient der schweizweit einheitlichen Kennzeichnung von Betriebs- Sicherheitsausrüstungen (BSA) von Nationalstrassen.
BSA EES	Betriebs- und Sicherheitsausrüstungen (BSA) Technische Ausrüstung der Nationalstrassen zur Steuerung und Überwachung des Betriebs und Gewährleistung der Verkehrssicherheit. Wird in der Richtlinie AKS-CH [1] auch als «Produkt» bezeichnet.
Entität	Informationsobjekt
FA BSAS	Fachapplikation Betriebs und Sicherheitsausrüstungen Sofortlösung
Fachkatalog	Ein Fachkatalog enthält Vorgaben zur Typendifferenzierung, welche in der FA BSAS angewendet werden.
GeoDataFromTo	Objektklasse / Entität, mit welcher die geografisch bezogenen Attribute von BSA wie Querschnitt, Streifen, Koordinaten etc. repräsentiert werden.
Lane	Objektklasse / Entität, mit welcher die Typen der Streifung (Fahrspur) repräsentiert werden.
Location	Objektklasse / Entität, mit welcher das Ortsverzeichnis BSAS repräsentiert wird.
LocationStructure	Objektklasse / Entität, mit welcher der Fachkatalog der Ortsstruktur BSAS repräsentiert wird.
LocationType	Objektklasse / Entität, mit welcher der Fachkatalog der Ortstypen BSAS repräsentiert wird.
OSEStructure	Objektklasse / Entität, mit welcher der Fachkatalog der BSA Struktur repräsentiert wird.
OSEType	Objektklasse / Entität, mit welcher der Fachkatalog der BSA Typen repräsentiert wird.
OSData	Objektklasse / Entität, mit welcher die technischen Attribute von BSA wie Hersteller, Seriennummer, Leuchtmittelart etc. repräsentiert werden.
OSEInventory	Objektklasse / Entität, mit welcher die BSA Verzeichnisse repräsentiert werden.
GUID	Weltweit eindeutiger Datenschlüssel (GUID) Er wird mit Hilfe einer MAC (Media Access Control) Adresse und einem Zeitstempel konstruiert. Globally Unique Identifier (GUID)
WSDL	Web Services Description Language Beschreibung des Webservice in XML

Literaturverzeichnis

Richtlinie des ASTRA

- [1] Bundesamt für Strassen ASTRA (2014), „**Struktur und Kennzeichnung der Betriebs- und Sicherheitsausrüstungen (AKS-CH)**“, *Richtlinien ASTRA 13013, V2.52*, www.astra.admin.ch.
-

IT-Dokumentation

- [2] Bundesamt für Strassen ASTRA (2019), „**Fachapplikation Betriebs- und Sicherheitsausrüstungen Sofortlösung (FA BSAS) - Anwendungshandbuch**“, *IT-Dokumentation ASTRA 63011, V2.00*, www.astra.admin.ch.
-

Dokumentation

- [3] Bundesamt für Strassen ASTRA (2016), „**Methodologie der Bewertung für die Zustandserfassung BSA – Erhaltungsplanung BSA**“, *Dokumentation ASTRA 8B310, V1.00*, www.astra.admin.ch.
-

Auflistung der Änderungen

Ausgabe	Version	Datum	Änderungen
2019	2.01	01.02.2021	<ul style="list-style-type: none"> • Kapitel 4.1.2 «AKSLocationDTO.Shortcut», «OSEInventoryDTO.Cantons», «OSEInventory.TerritorialUnits» Gültigkeitsbereich angepasst, «AKSLocationDTO.RBBSCode» ergänzt • Kapitel 5.4.1 URL Test-Umgebung angepasst • Kapitel 5.4.2 URL Test-Umgebung angepasst • Kapitel 5.4.3 URL Test-Umgebung angepasst
2019	2.00	10.10.2019	<p>Diverse Ergänzungen/Anpassungen betreffend Inventardaten Ort (AKSLocation) und Zugehörigkeit (AKSComponent). Diverse Anpassungen und Aktualisierungen der Schnittstelle. Ergänzungen betreffend «typenspezifischen Eigenschaften» und «Zustandsdaten».</p> <ul style="list-style-type: none"> • Literaturverzeichnis aktualisiert • Kapitel 2 angepasst • Kapitel 4 diverse Anpassungen / Ergänzungen • Abb. 4.17 LocationDTO (Verzeichnis Ort BSAS) erweitert • Abb. 4.16 OSEInventoryDTO (Verzeichnis BSA) erweitert • Kapitel 5.2 angepasst • Kapitel 5.4 mit Konzept erweitert sowie Methoden ergänzt • Kapitel 5.4.1 angepasst
2015	1.10	06.05.2016	Anpassung Abb. 4.21 OSEDataDTO (Technische Eigenschaften von BSA).
2015	1.00	01.10.2015	Inkrafttreten Ausgabe 2015 (original Version in Deutsch).

