Release Notes
Schnittstellen VDV KA

Atos Worldline GmbH
Pascalstraße 19
D - 52076 Aachen

DOKUMENTINFORMATION

Titel	Release Notes
Thema	Schnittstellen VDV KA
Dateiname	Release Notes-Schnittstellen-1.1.07.doc
Anzahl Seiten	23
Version	1.0 / 1.1.07
Datum	Aachen, den 23.09.2011
Status	Final
Vertraulichkeit	Public
Kontaktperson	Wilk Hoffmann
	Wilk.Hoffmann@atos.net
Abteilung	Prepaid & Mobility Solutions

Inhaltsverzeichnis

1 Historie

VERSIONSHISTORIE

Version	Datum	Autor	Grund der Änderung
0.1	17.06.2011	Wilk Hoffmann	Erster Entwurf der Release Notes
0.2	28.07.2011	Wilk Hoffmann	Überarbeitung im Rahmen von RfC0006
0.3	23.09.2011	Wilk Hoffmann	Überarbeitung im Rahmen von RfC0006
1.0	11.04.2013	Wilk Hoffmann	Zusammenführung der Release Notes zu diesem Dokument

1.1 Version 0.1

Schema/WSDL	Version (vor/nach)	Element/Typ/Operation	Beschreibung
XML-Schema_ION.xsd	1	TX_RET_Type -> TXA, TXB	Attribut transOrigSignatur als optionales Attribut eingefügt
XML-Schema_ION.xsd	1	TransSignatur_Type	Neu eingeführt, um bei Signatur in TX_Base und TX_RET_Type den gleichen Datentyp zu verwenden
XML-Schema_ION.xsd	1	TX_SANF_Type	Von XML-Schema_AH.xsd übernommen. Muss dort gelöscht werden
XML-Schema_ION.xsd	1	TX_SMIT_Type	Von XML-Schema_AH.xsd übernommen. Muss dort gelöscht werden
XML-Schema_ION.xsd	1	Mitteilung_CODE_Type	Von XML-Schema_AH_Types.xsd übernommen. Muss dort gelöscht werden
XML-Schema_ION.xsd	1	TXFREIMITA (Typ+Elem.)	Von XML-Schema_AH.xsd übernommen. Muss dort gelöscht werden
XML-Schema_ION.xsd	1	TXSANFSYMK (Typ+Elem.)	Von XML-Schema_AH.xsd übernommen. Muss dort gelöscht werden
XML-Schema_ION.xsd	1	TXSANFASYMK (Typ+Elem.)	Von XML-Schema_AH.xsd übernommen. Muss dort gelöscht werden
XML-Schema_ION.xsd	1	TXSAANFSYMK (Typ+Elem.)	Von XML-Schema_AH.xsd übernommen. Muss dort gelöscht

			werden
XML-Schema_ION.xsd	1	TXSAANFASYMK (Typ+Elem.)	Von XML-Schema_AH.xsd übernommen. Muss dort gelöscht werden
XML-Schema_AH.xsd	1		Elemente und Typen löschen (s.o.), Referenzen umstellen
XML-Schema_AH_Types.xsd	1		Mitteilung_CODE_Type löschen
XML-Schema_KVP_DL_PV.xsd	1		Überarbeitet (bisher nicht veröffentlicht)
KOSE_Services.wsdl	1	TXAMsg->TXAReqMsg TXBMsg->TXBReqMsg	Für TXA und TXB bei Entgegennahme werden eigene Nachrichten verwendet, die Nachricht selbst bleibt im Aufbau gleich (Service Listenstatus mitteilen).
AH_Services.wsdl	1		Referenzen umstellen auf allgemeine Typen/Elemente
DL_Services.wsdl	1		Überarbeitet (bisher nicht veröffentlicht)
KVP_Services.wsdl	1		Überarbeitet (bisher nicht veröffentlicht)
PV_Services.wsdl	1		Überarbeitet (bisher nicht veröffentlicht)

1.2 Version 0.2

Schema/WSDL	Version (vor/nach)	Element/Typ/Operation	Beschreibung
XML-Schema_KA_Common.xsd	1		Keine Änderungen
XML-Schema_Common_Types.xsd	1	DateTimeCompact	Neuer Datentyp DateTimeCompact
XML-Schema_Common_Types.xsd	1	DateCompact	Neuer Datentyp DateCompact
XML-Schema_Common_Types.xsd	1	Datef	Neuer Datentyp Datef
XML-Schema_ION.xsd	1	PrintableString	Hier neu, von XML-Schema_KVP_DL_PV.xsd verschoben
XML-Schema_ION.xsd	1	AuftragsTyp_CODE	Hier neu, von XML-Schema_KVP_DL_PV.xsd

			verschoben
XML-Schema_ION.xsd	1	Abrechnungsverfahren_CODE	Hier neu, von XML-Schema_KVP_DL_PV.xsd verschoben
XML-Schema_ION.xsd	1	Bezahlart_CODE	Hier neu, von XML-Schema_KVP_DL_PV.xsd verschoben
XML-Schema_ION.xsd	1	TerminalTyp_CODE	Hier neu, von XML-Schema_KVP_DL_PV.xsd verschoben
XML-Schema_ION.xsd	1	Terminal_ID_Type	Hier neu, von XML-Schema_KVP_DL_PV.xsd verschoben
XML-Schema_ION.xsd	1	TXAS_Type	Hier neu, von XML-Schema_KVP_DL_PV.xsd verschoben
XML-Schema_ION.xsd	1	TX_BER_Type	Hier neu, von XML-Schema_KVP_DL_PV.xsd verschoben
XML-Schema_ION.xsd	1	Ort_ID_Type und OrtsTyp_CODE_Type	Hier neu, von XML-Schema_KVP_DL_PV.xsd verschoben
XML-Schema_ALISE.xsd	1		Neu eingeführt
XML-Schema_AH.xsd	1		Elemente und Typen löschen (s.o.), Referenzen umstellen für ALISE
XML-Schema_AH_Types.xsd	1		Terminal_ID_Type löschen, DateTimeCompact löschen, PrintableString löschen, TXAS_Type löschen, Ort_ID_type löschen. Dann auf richtige Referenz setzen
XML-Schema_KOSE.xsd	1	TXAUFA, TXAUFB	sperrLoeschDatum ist nun Date statt DateTime
XML-Schema_KVP_DL_PV.xsd	1		Überarbeitet (bisher nicht veröffentlicht)
KOSE_Services.wsdl	1		Keine Änderungen

AH_Services.wsdl	1		Referenzen umstellen auf allgemeine Typen/Elemente
DL_Services.wsdl	1		Keine Änderungen (bisher nicht veröffentlicht)
KVP_Services.wsdl	1		Überarbeitet (bisher nicht veröffentlicht)
PV_Services.wsdl	1		Überarbeitet (bisher nicht veröffentlicht)
KVP_ALISE_Services.wsdl	1		Neu eingeführt.
PV_ALISE_Services.wsdl	1		Neu eingeführt

1.3 Version 0.3

1.3.1 Änderungen XSD

Schema	Version (vor/nach)	Element/Typ	Beschreibung
XML-Schema_KA_Common.xsd	1	KA_TX_HEADER_TYPE -> TX_BASE	KA_TX_HEADER_TYPE entfällt, in Zukunft wird ebenfalls txbase verwendet. Betroffen sind nur die Transaktionen „diensteAnmelden“ und „diensteAbmelden“ in der WSDL der ZVM
XML-Schema_ZVM_Common.xsd	1	diensteAnmelden, diensteAbmelden	Nutzen jetzt als Header TX_BASE. Betroffen sind nur die Transaktionen „diensteAnmelden“ und „diensteAbmelden“ in der WSDL der ZVM
XML-Schema_Common_Types.xsd	1	Keine Änderungen	
XML-Schema_ION.xsd	1	TXSFREIMITSYMK	Wurde aus dem Schema XML-Schema_KVP_DL_PV.xsd hierher verschoben (Typ & Element)
XML-Schema_ION.xsd	1	TXSFREIMITASYMK	Wurde aus dem Schema XML-Schema_KVP_DL_PV.xsd hierher verschoben (Typ & Element)
XML-Schema_ION.xsd	1	TXSMITSYMK	Wurde aus dem Schema XML-Schema_KVP_DL_PV.xsd hierher verschoben (Typ & Element)
XML-Schema_ION.xsd	1	TXSMITASYMK	Wurde aus dem Schema XML-Schema_KVP_DL_PV.xsd hierher verschoben (Typ & Element)

Schema	Version (vor/nach)	Element/Typ	Beschreibung
XML-Schema_ALISE.xsd	1	Keine Änderungen	
XML-Schema_PE.xsd	1	Keine Änderungen	
XML-Schema_RVS.xsd	1	Keine Änderungen	
XML-Schema_AH.xsd	1	TXSANFA	Wurde entfernt (Element & Typ)
XML-Schema_AH.xsd	1	TXSAANFA	Wurde entfernt (Element & Typ)
XML-Schema_AH_Types.xsd	1	AMErrorCode	Neu eingeführt.
XML-Schema_KOSE.xsd	1	Keine Änderungen	
XML-Schema_KOSE_Attachments.xsd	1	Keine Änderungen	
XML-Schema_KOSE_Types.xsd	1	Keine Änderungen	
XML-Schema_KVP_DL_PV.xsd	1	TXSFREIMITSYMK	Wurde aus dem Schema XML-Schema_KVP_DL_PV.xsd entfernt und nach XML-Schema_ION.xsd verschoben (Typ & Element)
XML-Schema_KVP_DL_PV.xsd	1	TXSFREIMITASYMK	Wurde aus dem Schema XML-Schema_KVP_DL_PV.xsd entfernt und nach XML-Schema_ION.xsd verschoben (Typ & Element)
XML-Schema_KVP_DL_PV.xsd	1	TXSMITSYMK	Wurde aus dem Schema XML-Schema_KVP_DL_PV.xsd entfernt und nach XML-Schema_ION.xsd verschoben (Typ & Element)
XML-Schema_KVP_DL_PV.xsd	1	TXSMITASYMK	Wurde aus dem Schema XML-Schema_KVP_DL_PV.xsd entfernt und nach XML-Schema_ION.xsd verschoben (Typ & Element)

1.3.2 Änderungen WSDLs

WSDL	Version (vor/nach)	Operation	Beschreibung
ZVM_Services.wsdl	1	Keine Änderungen	Interner Aufbau der Nachrichten geändert (siehe oben)

WSDL	Version (vor/nach)	Operation	Beschreibung
KOSE_Services.wsdl	1	Keine Änderungen	
AH_Services.wsdl	1	TXSFREIMITSYMK	Neue Operation
AH_Services.wsdl	1	TXSFREIMITASYMK	Neue Operation
AH_Services.wsdl	1	TXSMITSYMK	Neue Operation
AH_Services.wsdl	1	TXSMITASYMK	Neue Operation
AH_Services.wsdl	1	Policy	Für WS Security eingeführt
AH_Services.wsdl	1	Soap Action	Alle Operationen haben nun auch eine namentliche SOAP Action
DL_Services.wsdl	1	TXSFREIMITSYMK, TXSFREIMITASYMK, TXSMITSYMK, TXSMITASYMK	Referenz auf XML-Schema_ION.xsd statt XML-Schema_KVP_DL_PV.xsd
KVP_Services.wsdl	1	TXSFREIMITSYMK, TXSFREIMITASYMK, TXSMITSYMK, TXSMITASYMK	Referenz auf XML-Schema_ION.xsd statt XML-Schema_KVP_DL_PV.xsd
PV_Services.wsdl	1	TXSFREIMITSYMK, TXSFREIMITASYMK, TXSMITSYMK, TXSMITASYMK	Referenz auf XML-Schema_ION.xsd statt XML-Schema_KVP_DL_PV.xsd
KVP_ALISE_Services.wsdl	1	Keine Änderungen	
PV_ALISE_Services.wsdl	1	Keine Änderungen	

2 Release Notes

Hinweis: diese allererste Version der technischen Umsetzung der Spezifikationen ist unvollständig und enthält noch sowohl fachliche als auch technische Fehler.

Es wird empfohlen, direkt mit der neuen Version 1.1.09 zu arbeiten.

Basis für die Änderungen ist die letzte Version 1.1.06 (technische Version) der Schnittstellen vom 23.06.2011 als monolithisches XML Schema (4014 Zeilen XSD Code).

Services im Rahmen von WSDLs waren in 1.1.06 nur rudimentär vorhanden.

Ergebnis ist die Version 1.1.07 welche die technische Version 1 der Schnittstelle bedingt. Erklärungen zu dieser technischen Version finden sich in den nachfolgenden Kapiteln.

Die folgende Übersicht zeigt, welche technische Version aus welchem Teilbereich zur KA Spezifikation 1.1.07 gehört. Allgemeine Anteile haben kein konkretes Pendant auf Ebene der Spezifikationen.

Spezifikation	Teilbereich	Version	XSD oder WSDL
1.1.07	Kommunikation	1	XML-Schema_KA_Common.xsd
1.1.07	Allgemein, Datentypen	1	XML-Schema_Common_Types.xsd
1.1.07	Übergreifende Nachrichten und abstrakte Nachrichtenanteile	1	XML-Schema_ION.xsd
1.1.07	ASM / AH	1	AH_Services.wsdl
1.1.07	ASM / AH	1	XML-Schema_AH.xsd
1.1.07	ASM / AH	1	XML-Schema_AH_Attachments.xsd
1.1.07	ASM / AH	1	XML-Schema_AH_Types.xsd
Spec KOSES 1.07	KOSES	1	KOSE_Services.wsdl
Spec KOSES 1.07	KOSES	1	XML-Schema_KOSE.xsd
Spec KOSES 1.07	KOSES	1	XML-Schema_KOSE_Attachments.xsd
Spec KOSES 1.07	KOSES	1	XML-Schema_KOSE_Types.xsd
1.1.07	Schema für DL, KVP	1	XML-Schema_KVP_DL_PV.xsd

Spezifikation	Teilbereich	Version	XSD oder WSDL
	und PV Nachrichten		
1.1.07	DLS	1	DL_Services.wsdl
Spec AktM 1.1.07	Aktionsmanagement ALISE	1	XML-Schema_ALISE.xsd
Spec AktM 1.1.07	PV ALISE Erweiterung	1	PV_ALISE_Services.wsdl
Spec AktM 1.1.07	KVP ALISE Erweiterung	1	KVP_ALISE_Services.wsdl
Spec PE 1.1.07		1	XML-Schema_PE.xsd
Spec RVS 1.1.07		1	XML-Schema_RVS.xsd
1.1.07	NM Lieferliste	1	XML-Schema_NMLieferliste.xsd
Spec ION 1.1.07	ZVM	1	XML-Schema_ZVM_Common.xsd
Spec ION 1.1.07	ZVM	1	ZVM_Services.wsdl

3 Allgemeine Änderungen im Schnittstellendesign

In diesem Kapitel werden die Abweichungen zur monolithischen XSD-Version beschrieben, welche sich durch Erkenntnisgewinn während der Spezifikations- und Designphase ergeben haben.

3.1 Namensräume

Prinzipiell dienen die Namensräume einer sauberen Strukturierung. Des Weiteren sollen sie für die spätere Versionierung verwendet werden. Hierbei ist zu beachten, dass nicht das Kürzel für den Spezifikationsstand (z.B. „1.107“) für die Namensräume verwendet wurde, da nicht jede Änderung in der Spezifikation Einfluss auf alle Nachrichten bzw. Services haben muss. Daher wurde hier die abstrakte Nummer „1“ vergeben.

Ferner wurde eine Versionsnummer in den Namespace integriert. Die Versionsnummer im Namespace der WSDL definiert eindeutig die Schnittstelle. Es versteht sich, dass eine Änderung in der WSDL oder eines die Schnittstelle beschreibenden inkludierten Schemata eine Erhöhung der Versionsnummer im WSDL Namensraum nach sich zieht.

Von der sauberen Modellierung abgesehen werden die Namensräume im WS Security Umfeld zwingend benötigt.

3.2 Modularisierung der XML Schemata

Zum Zwecke von Schnittstellen Wartbarkeit, zur Modularisierung der unterschiedlich benötigten Anteile und zum einfacheren Versionsmanagement verschiedener Anteile wurde das XML Schema („XSD“) aufgeteilt.

Die Aufteilung geschieht auf folgende Art und Weise:

- Pro KA Rolle werden eigene XSDs und eine eigene WSDLs verwendet. Hierbei wird der Serviceorientierte Ansatz gewählt („Wer den Service anbietet bzw. die Nachricht empfängt, der muss die Definitionen hierfür bereitstellen“)
- Die XSDs sind pro Namensraum in eigenen Dateien gehalten (siehe 3.1)
- Es gibt eine Hierarchie, bei der allgemeine Anteile, welche für alle KA Rollen und Services verwendet werden, in eigene Schemata ausgelagert werden. Spezifische Nachrichten und Typen bauen auf diesen allgemeinen Anteilen auf.

3.3 Versionsverwaltung

Einer der wichtigen Aspekte in der Kernapplikation ist der Umgang mit verschiedenen Versionen der Nachrichten. Diese (und weitere) nicht-funktionalen Anforderungen sollten keinesfalls innerhalb der Nachricht selbst abgebildet werden. Daher wird eine Versionsverwaltung deklarativ von außen über unterschiedliche URLs abgebildet, ohne Einfluss auf die Nachrichteninhalte selbst zu haben.

Die Version einer Nachricht oder vielmehr die Version einer Schnittstelle muss „von außen“ deklarativ per Namensraum erfolgen. Im unten aufgeführten Beispiel wird das Kürzel „ka“ der abstrakten URL inklusive Versionsnummer (fett markiert) zugeordnet, wobei es sich hier nur um ein Beispiel handelt. Das Format der Versionsnummer kann auch anders sein.

```
xmlns:ka=http://vdv/ka/common/1
```

Für die Routing Information wird die Versionierung ebenfalls benötigt, da eine neue Version der Schnittstelle eine neue WSDL und eine neue Ziel-URL bedeutet, sofern keine Schnittstellenkonverter eingesetzt werden. Daher ist die Information über die Version ebenfalls im SOAP Header lokalisiert. Eine genaue Spezifikation des SOAP Headers findet sich in [1], Kapitel 7.

3.4 Ausnahmen Behandlung

Die sogenannten „TXA“ Nachrichten werden im Fehlerfall versendet. Im Falle eines SOAP-Faults ist es dann möglich, eine definierte Struktur zu senden. Allerdings gibt es hierbei folgendes zu beachten:

- Bei den sicherheitsbedingten Fehlern (WS-Security) wird kein TXA gesendet. Hier gibt das System grundsätzliche keine Einzelheiten zurück, da es sich um einen Angriff handeln könnte. Dies gilt für Entschlüsselung aber auch für fehlerhafte Signaturen. Ohnehin könnte man bei nicht möglicher Entschlüsselung die TXA Struktur nicht sinnvoll füllen, da Pflichtanteile gefüllt werden müssen, die einen lesbaren Request voraussetzen.
- Bei syntaxbedingten Fehlern könnte es sich ebenfalls um einen Angriff handeln. Darüber hinaus kann die Struktur TXA nicht sinnvoll gefüllt werden, wenn die Nachricht nicht validiert werden kann. Auch bei diesen Fehlern wird kein kein TXA geschickt.
- Die einzige sinnvolle Struktur innerhalb von TXA ist ein Fehlercode, der aber dann nur bei semantischen Fehlern geliefert werden darf (nicht bei Sicherheits- oder Format-Problemen). Dieser Code kann dann auf dem anfragenden System zur Weiterverarbeitung genutzt werden. Textuelle Felder machen hierbei keinen Sinn. Somit wurde das textuelle Feld in TXA durch einen dedizierten Fehlercode ersetzt, welcher als Aufzählungstyp in einer XSD definiert ist.

3.5 Wiederhol-Szenarien

Wiederhol-Szenarien werden möglichst gradlinig gehandhabt. In der Nachricht selbst sollten keine Informationen darüber gespeichert sein (siehe auch nachfolgendes Kapitel)

Wird die Annahme der Nachricht aus technischen Gründen verweigert, so wiederholt der Client das Senden derselben Nachricht, bis eine Annahme erfolgt ist.

Geht die Antwort auf dem Weg zum Client verloren, so sendet der Client dieselbe Nachricht mit derselben Transaktions-ID erneut. Es wird dann das Duplikat festgestellt und der entsprechend Code als TXA zurückgesendet. In dem Fall muss der Client das nur registrieren und braucht keine weiteren Aktivitäten, da die Nachricht ja im Endsystem eingetroffen ist und verarbeitet wurde.

Wurde die Nachricht aus semantischen Gründen abgelehnt, so bedeutet dies in der Regel eine notwendige Anpassung des Inhalts. In dem Moment muss mit einer neuen Transaktions-ID gearbeitet werden.

3.6 Optionale Felder in TX_BASE

Der abstrakte Basis-Transaktionsdatensatz enthält eine Reihe Attribute, welche aufgrund der oben genannten Abweichungen / Festlegungen nicht mehr benötigt werden. Diese Felder sind aus Kompatibilitätsgründen weiterhin in TX_BASE enthalten, werden aber als optional definiert und können unabhängig von deren Inhalt ignoriert werden. Dadurch wird Redundanz eliminiert und ein sauberes Systemverhalten ermöglicht.

Nachfolgend werden diese Felder gezeigt (rot markiert):

```
<txbase>
  <transAuftrag>transAuftrag</transAuftrag>
  <transEmpfaenger_ID>0</transEmpfaenger_ID>
  <transEmpfaengerRolle>0</transEmpfaengerRolle>
  <transStatus>0</transStatus>
  <transTransaktion_ID>
    <transSequenznummer>0</transSequenznummer>
    <transSenderRolle>0</transSenderRolle>
    <transSender_ID>0</transSender_ID>
  </transTransaktion_ID>
  <transTransaktionsTyp>0</transTransaktionsTyp>
  <transTransaktionsZeitpunkt>2001-12-31T12:00:00</transTransaktionsZeitpunkt>
  <transVersion>transVersion</transVersion>
  <transWiederholungsZaehler>0</transWiederholungsZaehler>
   <transSignaturTyp>0</transSignaturTyp>
  <transSignaturZertifikat>0F00</transSignaturZertifikat>
  <transSignatur>0F00</transSignatur>
</txbase>
```

Nachfolgend wird auf die optionalen Felder nochmal einzeln eingegangen.

Nicht benötigtes Feld	Begründung
<transAuftrag>	TX_Base ist abstrakt, d.h. eine konkrete Transaktion ist immer umschließend (z.B. TXSAUFA = Sperrauftrag Applikation). Dies spezifiziert ausreichend den Anwendungsfall. Textuelle Felder können grundsätzlich nicht ausgewertet werden. Es ergibt sich somit kein Mehrwert für dieses Feld
<transStatus>	Das Tag <transStatus> entfällt, da Wiederholungen über eine gleiche Transaktions-ID abgebildet werden sollen.
<transTransaktionsTyp>	Das Tag <transTransaktionsTyp> entfällt, die abgeleitete Transaktion selber die Transaktion spezifiziert (z.B. TXSLNMREQ = Holen einer Gesamtsperrliste NM)
<transWiederholungsZaehler>	Das Tag <transWiederholungsZaehler> entfällt, da Wiederholungen über eine gleiche Transaktions-ID abgebildet werden sollen und semantisch kein Nutzen entsteht wenn die Anzahl der Wiederholversuche bekannt ist (Wer soll dies auswerten, was wären die Folgen?)
<transVersion>	Das Tag <transVersion> entfällt, da die Versionierung über Namensräume und eigenen URLs im ION abgebildet werden muss
<transSignaturTyp> <transSignaturZertifikat> <transSignatur>	Diese Felder entfallen im Rahmen des CR100, da nun WS-Security eingesetzt wird.

Somit ergibt sich folgender noch relevanter Anteil (nach Weglassen der nun optionalen Felder):

```
<txbase>
  <transEmpfaenger_ID>0</transEmpfaenger_ID>
  <transEmpfaengerRolle>0</transEmpfaengerRolle>
  <transTransaktion_ID>
    <transSequenznummer>0</transSequenznummer>
    <transSenderRolle>0</transSenderRolle>
    <transSender_ID>0</transSender_ID>
  </transTransaktion_ID>
  <transTransaktionsZeitpunkt>2001-12-31T12:00:00</transTransaktionsZeitpunkt>
</txbase>
```

3.7 TXAS

Lastenheft 1.107	Umsetzung in KOSES Spez.
Allgemeiner Request mit Auftragstyp_Code verwendet bzw. auch erweitert (mit neuem Namen), wenn andere Datenelemente verwendet werden müssen Historisch bedingt ist kein SOA-Ansatz verwendet worden	• SOA Ansatz umgesetzt • werden umgesetzt als spezielle Requests • Für unterschiedliche Anfragen werden unterschiedliche Daten benötigt, die im Request eindeutig zugeordnet sind • Vermeidung von optionalen Feldern und Abfragen auf Auftragstyp_Code • Für Routing in Kombination mit WS Security (ZVM) müsste ohnehin für jeden Auftragstyp_Code in der WSDL eine eigene Operation und ein eigenes Binding verwendet werden.

3.8 Vererbungsanteile in den Nachrichten

Vererbung von Nachrichtenanteilen macht nur in wenigen Fällen Sinn. Insbesondere dürfen abstrakte Nachrichtenanteile nicht mit eigenem Namen versehen werden, da dann unnötige Gruppierungen innerhalb der Nachricht entstehen, die nicht nötig sind und die Länge der Nachricht vergrößern (Beispiel TXSAUF für alle Sperraufträge).

In der vorliegenden XSD ist dies nicht einheitlich umgesetzt.

Vererbung wird noch nur an sinnvollen Stellen verwendet und ohne eigene Namensgebung.

Insbesondere entfällt die Vererbung vom leeren Typ TX_TYPE. Elemente sind direkt von TX_BASE abgeleitet, welcher selber seine Elemente als „Header“ besitzt. Im Layout und in der Kompatibilität der Nachrichten ändert sich hierdurch nichts. Nur im aus der WSDL generierten Source Code werden Unterschiede auftreten.

3.9 Listen als Massendatenübertragung

Da SOAP Nachrichten, in denen viele (> 1000) XML-Elemente eingebettet sind, informationstechnisch nicht effizient verarbeitet werden können, werden potentiell große Listen

mit dem ZIP Algorithmus gepackt.

Durch das Packen der XML Liste lässt sich die spätere Größe des Base64 Elements effizient verringern da XML sehr viele Wiederholungen enthält.

Die ZIP codierten XML Daten werden dann in Base64 umgewandelt und in diesem Format in die SOAP Nachricht eingebettet.

Unter Verwendung des im WS Security Umfeld üblichen standardisierten MTOM Verfahrens wird dann Liste und Nachricht an den Dienstaufrufer übertragen. Dadurch erfolgt die Verschlüsselung des „Anhangs“ (das base64 codierte Datenfeld) implizit über den für die Nachrichten verwendeten Schlüssel.

MTOM sorgt schließlich transparent dafür, dass zu große Anteile im base64 Feld automatisch vor der Übertragung effizient im Binärformat in die http-Nachricht eingebettet werden.

Durch das Auslagern von Nachrichtenanteilen ins Binärformat auf der Transportschicht erzielt MTOM gegenüber der Übertragung der Daten im base64 Format innerhalb der Nachricht eine weitere Bandbreiteneinsparung um ca. 33 Prozent.

4 Änderungen der Schnittstellen

4.1 Änderungen Common

4.1.1 Änderungen XML-Schema_KA_Common.xsd

Neueinführung in der technischen Version 1.

4.1.2 Änderungen XML-Schema_Common_Types.xsd

Neueinführung in der technischen Version 1, extrahiert alle allgemeinen Typen aus dem Ursprungsschema.

Fachliche Transaktion	Element/Typ	Beschreibung

4.1.3 Änderungen XML-Schema_ION.xsd

Neueinführung in der technischen Version 1, extrahiert alle allgemeinen Nachrichtenanteile aus dem Ursprungsschema.

4.2 Änderungen ASM Tool (AH-System)

Neueinführung in der technischen Version 1.

4.2.1 Änderungen AH_Services.wsdl

Neueinführung in der technischen Version 1.

4.2.2 Änderungen XML-Schema_AH.xsd

Neueinführung in der technischen Version 1, extrahiert alle AH relevanten Nachrichtenanteile, Elemente und Typen aus dem Ursprungsschema und ergänzt neue benötigte Elemente.

4.2.3 Änderungen XML-Schema_AH_Types.xsd

Neueinführung in der technischen Version 1, extrahiert alle AH relevanten Typen aus dem Ursprungsschema und ergänzt neue benötigte Typen.

4.2.4 Änderungen XML-Schema_AH_Attachments.xsd

Neueinführung in der technischen Version 1, extrahiert alle AH relevanten Nachrichtenanteile,

Elemente und Typen für Antwortlisten aus dem Ursprungsschema und ergänzt neue benötigte Typen und Elemente.

4.3 Änderungen KOSES

4.3.1 Zyklusbasierte Listen

Anders als im Lastenheft vorgesehen (Sperrlistennummern), arbeitet KOSES mit Zyklen für Sperrlisten. Dies bietet mehrere Vorteile:

- Alle Listen bekommen die gleiche Zyklusnummer und brauchen keine eigenen Nummernkreise
- Der Zyklus kann beliebig verkürzt (z.B. von täglich auf ½ täglich oder stündlich) oder verlängert werden, je nach Anforderung.
- Innerhalb eines Zyklus ist der nach außen gereichte Sperrbestand konstant, d.h. alle Partner haben den gleichen Sperrbestand innerhalb eines Zyklus.
- Der Zyklus stellt den „Snapshot“ aller Sperreinträge im System im Moment der Zykluserstellung dar. Nur zu diesem Zeitpunkt ist eine Synchronisation hinsichtlich eingehender Listenabfragen nötig, während der übrigen Zeit können Listen vom KOSES abgefragt werden.
- Die Unterscheidung der Listen ist hinsichtlich ihres Typs ohnehin gegeben, Zyklus-ID und Listentyp identifizieren somit eindeutig jede Liste. Somit können jederzeit neue Listen ins System aufgenommen werden, ohne dass Nummernkreise beachtet werden müssen.
- Das Loggen hinsichtlich abgeholter Listen wird vereinfacht und verallgemeinert, da nur Zyklus und Listentyp beachtet werden müssen.

Definition:

- Die (Sperr-) Listenzyklusnummer ist eine streng Monoton aufsteigende Sequenznummer.
- Sie wird in einem festen konfigurierbaren Turnus zu einem bestimmten Zeitpunkt inkrementiert (z.B. einmal täglich um 1:00 Uhr).
- Zu einer (Sperr-) Listenzyklusnummer werden alle Sperrobjektzustände, die zu diesem Zeitpunkt im Kose System erfasst wurden, assoziiert. Sie gilt Systemweit und ist nicht an Organisationen oder an Rollen gebunden und auch nicht an den Listentyp

In allen Dienstantworten, in denen Listen (Gesamt oder Differenzsperrlisten, Sperrnachweislisten) übertragen werden, wird die (Sperr-) Listenzyklusnummer (zusammen mit dem Datum der Zykluszeitpunkts) übertragen.

Gesamtsperrlisten und Sperrnachweislisten:

Hier wird die Zyklusnummer („bis“) übertragen. Diese Entität definiert eindeutig den Zyklus und damit Zeitpunkt der dieser Sperrliste zugrundeliegenden Sperrobjekte.

Differenzsperrlisten:

In Differenzsperrlisten wird mit den Entitäten „Sperrlisten Zyklus von“, „Sperrlisten Zyklus bis“ eindeutig der zeitliche Bereich definiert der Grundlage für die

Differenzsperrlistenerzeugung ist.

Wird eine Differenzsperrliste unter Angabe der Sperrlistenzyklusnummer angefordert, so werden alle Sperrobjektänderungen zwischen dieser Zyklusnummer und der aktuellen Zyklusnummer übertragen.

Eine Differenzsperrlistenverarbeitung (auf initialer Basis einer Gesamtliste) beruht auf der kontinuierlichen Einspielung von Differenzsperrlisten. Dies ist der Fall solange man lückenlos Differenzsperrlisten unter der Angabe der letzten erfolgreichen „Sperrlisten Zyklus bis“ Nummer anfordert.

Die Verwendung der Zyklusnummer erlaubt es ebenfalls, das Sperrlistenverarbeitende Systeme in Ausnahmefällen (z.B. bei einem Systemausfall eines Sperrlistenanfordernden VU) eine Sperrlistenanforderung abzusetzen, deren Zyklusnummer z.B. 2 Tage in der Vergangenheit liegt. KOSES würde in diesem Fall alle Sperrobjektänderungen zwischen dieser Zyklusnummer und der aktuellen Zyklusnummer übertragen. Sowohl das hieraus resultierende Sperrlistenvolumen, als auch die Verarbeitung der Sperreinträge ist weniger aufwendig als die zweier einzelner Differenzsperrlisten.

Wird ein bestimmter Zeitraum überschritten (Default 7 Tage), so muss eine Gesamtsperrliste für Nutzermedien angefordert werden.

4.3.2 Trennen der ausgegebenen Listen nach Fachlichkeit

Im Lastenheft werden Listen aus fachlichen Elementen zusammengefügt, die unterschiedlichen Ursprungs sind (z.B. Liste mit SAM- und Organisationssperren).

Da die Datenelemente im Inhalt komplett unterschiedlich sind, bedeutet das Trennen der Listen eine deutliche Vereinfachung der Verarbeitung im anfragenden System.

Neben der Sperrliste „ORG SAM“ wurde zusätzlich die Liste für die symmetrischen und asymmetrischen Schlüssel aufgeteilt.

4.3.3 Sperrnachweislisten

Statt einer wiederholten Einzelabfrage von Sperrnachweisen mit TXAS/TXSNAWA, TXSNAWB wird pro Zyklus eine neue Liste der Sperrnachweise bereitgestellt. Damit verhalten sich die Sperrnachweise im Listenformat wie die Sperrlisten. Das gilt auch für die Abholregistrierung. Das Einführen der Liste geschah aus folgenden Gründen:

- Performanz- und Bandbreitenprobleme bei einem Deutschlandweiten System
- Nachvollziehbarkeit der Abholung der Sperrnachweise ist im Einzelmodus nicht gegeben
- War bereits Anforderung vieler Systemrealisierer, Einzeltransaktionen in eine Liste zu verpacken
- Kein sinnvoller und robuster Algorithmus möglich für Einzelabfragen: „Abfrage-Polling bis kein Nachweis mehr da ist“, „Liefere bei Anfrage irgendeinen Nachweis“
- Monitoring für Teilnehmer bei Einzelabfragen nicht sinnvoll umsetzbar.

4.3.4 Prüfsummen bei Differenzsperrlisten

Die Prüfsummenfunktionalität für Differenzsperrlisten wird zurückgestellt.

Der Algorithmus hierzu hängt stark von der verwendeten Programmiersprache, daher würde auch die Veröffentlichung des Codes nur bedingt für die Verkehrsunternehmen die Integration der Funktionalität vereinfachen. Das gesamte Thema wird als sehr komplex und problematisch im Fehlerfäll (abweichende Prüfsumme) angesehen.

4.3.5 Informationen über Abholverhalten und Berechtigungssperren

Lastenheft 1.107	**Umsetzung in KOSES Spez.**
TXSINF undifferenziert nach Sperrlisten und Sperrnachweise vom KOSE, angefragt mittels TXAS (Code=17) und Anzahl und Verweildauer von Berechtigungen (Code=18)	Listen zur Abfrage spezifiziert, getrennt nach TXSINFBER und TXSINFORG • Daten disjunkt • Trennung in sinnvolle Monitoringinformationen AH und KA-Teilnehmer (unterschiedliche Empfänger) • Lastenheft war zu ungenau, nicht für beide Typen ausgeführt • Erweiterungswunsch KA KG zyklusorientiert, unternehmensorientiert (deshalb Feld „ergebnistyp“ in TXSINFORGREQ eingeführt) • Verweildauer war bezüglich der Informationsgehalte gar nicht ausspezifiziert • WebService Security erfordert eine eigene Operation, deswegen muss TXAS je Abfragetyp separiert werden und mit eindeutigem Bezeichner versehen werden TXAS für Code 17 und 18 wird in der WSDL in TXSINFBERREQ bzw. TXSINFORGREQ übergehen.

4.3.6 Änderungen KOSE_Services.wsdl

Neueinführung in der technischen Version 1.

4.3.7 Änderungen XML-Schema_KOSE.xsd

Neueinführung in der technischen Version 1, extrahiert alle KOSE relevanten Nachrichtenanteile, Elemente und Typen aus dem Ursprungsschema und ergänzt neue benötigte Typen und Elemente.

4.3.8 Änderungen XML-Schema_KOSE_Types.xsd

Neueinführung in der technischen Version 1, extrahiert alle KOSE relevanten Typen aus dem Ursprungsschema und ergänzt neue benötigte Typen.

4.3.9 Änderungen XML-Schema_KOSE_Attachments.xsd

Neueinführung in der technischen Version 1, extrahiert alle KOSE relevanten Nachrichtenanteile, Elemente und Typen für Antwortlisten aus dem Ursprungsschema und ergänzt neue benötigte Typen und Elemente.

4.4 Änderungen ALISE

4.4.1 Änderungen XML-Schema_ALISE.xsd

Neueinführung in der technischen Version 1.

4.5 Änderungen Webservices DL, DL_Services.wsdl

Neueinführung in der technischen Version 1.

4.6 Änderungen Webservices KVP, KVP_Services.wsdl

Neueinführung in der technischen Version 1.

4.7 Änderungen Webservices PV, PV_Services.wsdl

Neueinführung in der technischen Version 1.

4.8 Änderungen ALISE Service Erweiterung für KVP und PV

4.8.1 Änderungen KVP_ALISE_Services.wsdl

Neueinführung in der technischen Version 1.

4.8.2 Änderungen PV_ALISE_Services.wsdl

Neueinführung in der technischen Version 1.

4.9 Änderungen für gemeinsames KVP, DL und PV Schema

4.9.1 Änderungen XML-Schema_KVP_DL_PV.xsd

Neueinführung in der technischen Version 1, extrahiert alle relevanten Nachrichtenanteile, Elemente und Typen für KVP, DL und PV aus dem Ursprungsschema und ergänzt neue benötigte Typen und Elemente.

4.10 Änderungen für Schema XML-Schema_NMLieferliste.xsd

Neueinführung in der technischen Version 1.

4.11 Änderungen für Schema XML-Schema_RVS.xsd

Neueinführung in der technischen Version 1, extrahiert alle verbliebenen Nachrichtenanteile, Elemente und Typen für aus dem Ursprungsschema.

4.12 Änderungen XML-Schema_PE.xsd

Neueinführung in der technischen Version 1, extrahiert alle relevanten Nachrichtenanteile, Elemente und Typen aus dem Ursprungsschema und ergänzt neue benötigte Typen und Elemente.

4.13 Änderungen im Bereich ZVM

4.13.1 Änderung für WSDL ZVM_Services.wsdl

Neueinführung in der technischen Version 1. Enthält Services zum An- und Abmelden von der ZVM.

4.13.2 Änderung XML-Schema_ZVM_Common.xsd

Neueinführung in der technischen Version 1. Enthält Nachrichten zum An- und Abmelden von der ZVM.

5 Weiterführende Dokumente

1. Spec-ION_V1107 - 1.07