



Elektronischer Rechtsverkehr (ERV)

Beschreibung der externen Schnittstelle der Services

Dateiname: Services_Beschreibung.doc

Version: 1.5 vom 08.08.2014

Ersteller: Christian Adorjan (christian.adorjan@brz.gv.at)

1 Dokumentinformation

1.1 Inhaltsverzeichnis

1	Dokumentinformation	2
1.1	Inhaltsverzeichnis	2
1.2	Abbildungsverzeichnis	3
1.3	Tabellenverzeichnis	3
1.4	Änderungsverlauf	4
2	Einleitung	5
2.1	Zweck des Dokuments	5
3	Konzepte.....	6
3.1	Eindeutige Identifizierbarkeit.....	6
3.2	Authentifizierung der externen Stellen.....	7
3.3	Autorisierung.....	7
3.4	Nichtbestreitbarkeit	7
3.5	Integrität und Verschlüsselung.....	7
3.6	Asynchrone Prüfung und Verarbeitung.....	7
3.7	Sichere Übertragung.....	8
3.8	Fehlerverhalten	8
3.9	Übermittlung von Anhängen	8
4	Implementierung	9
4.1	SOAP-Message	9
4.1.1	SOAP-Body	10
4.1.2	SOAP-Attachments	10
4.1.3	SOAP-Fault	10
4.2	ERV Nachrichtenschema	11
4.2.1	Nachrichtendaten	12
4.2.2	Nachrichtenkennung	16
4.2.3	Eingangstatus.....	17
4.2.4	Ausgangstatus.....	18
4.2.5	Postkorb	18
4.3	Webservice Definition.....	19

4.3.1	Eingang einbringen	19
4.3.2	Ergebnis abholen.....	20
4.3.3	Status abfragen.....	22
4.3.4	Ausgang abholen.....	23
4.3.5	Ausgang bestätigen	25
4.3.6	Anzahl abfragen	26
4.3.7	Operationen im Überblick.....	27
5	Anhang	29
5.1	Referenzen.....	29
5.1.1	Dokumente	29
5.1.2	Modelle	29
5.1.3	Literatur.....	29

1.2 Abbildungsverzeichnis

Abbildung 1: SOAP Message Struktur	9
Abbildung 2: Schema der Nachrichtendaten	12
Abbildung 3: Schema der Nachrichtenkennung	16
Abbildung 4: Schema des Eingangsstatus	17
Abbildung 5: Schema des Ausgangsstatus.....	18
Abbildung 6: Schema der Anzahlinformation	18
Abbildung 7: Request "EingangEinbringen"	19
Abbildung 8: Response "EingangEinbringen"	20
Abbildung 9: Request "ErgebnisAbholen"	21
Abbildung 10: Response "ErgebnisAbholen".....	21
Abbildung 11: Request "StatusAbfragen"	22
Abbildung 12: Response "StatusAbfragen".....	23
Abbildung 13: Request "AusgangAbholen"	24
Abbildung 14: Response "AusgangAbholen".....	24
Abbildung 15: Request "AusgangBestaetigen"	25
Abbildung 16: Response "AusgangBestaetigen".....	25
Abbildung 17: Request "AnzahlAbfragen"	26
Abbildung 18: Response "AnzahlAbfragen"	27

1.3 Tabellenverzeichnis

Tabelle 1: Überblick Webservice Operationen	28
Tabelle 2: Referenzen auf externe Dokumente	29
Tabelle 3: Referenzen auf Modelle	29

1.4 Änderungsverlauf

Version	Datum	Ersteller	Kommentar
1.0	29.06.2006	Tauber	erste veröffentlichte Version
1.1	29.09.2006	Tauber	Erweiterung der Webservice-Schnittstelle (CR-1118), Änderung des Verhaltens bei „Eingabe einbringen“, Anpassungen im Fehlerverhalten
1.2	27.02.2009	Winter	Änderung der Struktur der Message-ID im Adressteil (JJJMMTT eingefügt), Anpassung an neue Webservice-Schnittstelle
1.3	02.09.2009	Winter	Anpassung für neuen ERV
1.4	26.08.2010	Winter	Review und Update
1.5	08.08.2014	Adorjan	XLMDOKUMENT und JUSTIZARCHIVREF bei Anhang ergänzt

2 Einleitung

2.1 Zweck des Dokuments

Dieses Dokument beschreibt die Funktionalität der Services anhand der Schnittstelle zu den externen Stellen. Externe Stellen sind in diesem Zusammenhang Übermittlungsstellen und Direktteilnehmer-Institutionen, die im Folgenden auch Konzentratoren genannt werden. Beschrieben wird die Struktur und das Verhalten der Webservice Schnittstelle für den Eingang und Ausgang. Nicht Teil der Beschreibung ist die Schnittstelle zwischen Teilnehmern und Übermittlungsstellen.

3 Konzepte

Folgende Konzepte wurden bei der Definition der externen Service-Schnittstelle verfolgt.

3.1 Eindeutige Identifizierbarkeit

Jede Nachricht im Eingang und Ausgang ist über eine MessageId eindeutig identifizierbar. Diese wird vom Ersteller der Nachricht nach einer definierten Struktur vergeben und von den Services auf Eindeutigkeit geprüft.

Die Struktur der MessageId ist wie folgt definiert:

```
message-id = "mid://" address-teil "@" domain-name
```

Der domain-name wird für jeden Konzentrador sowie für die Services selbst vom Betreiber der Services vergeben und ist in den Services konfiguriert. Der Domain-Name wird case-sensitiv überprüft.

Für den Adressteil ist der jeweilige Absender verantwortlich. Die Struktur dafür ist folgende:

```
address-teil = JJJJMMTT "." freier-teil "." teilnehmer  
              "." anwendung
```

Das Element JJJJTTMM muss einem gültigen Datum entsprechen, wobei "JJJJ" für das Jahr steht, "MM" für den Monat und "TT" für den Tag.

Das Element freier-teil kann eine für die erzeugende Applikation eindeutige Nummer des Eingangs oder Ausgangs sein. Was das Element genau enthält, ist nicht vorgegeben, es könnte sich zB auch um einen Zeitstempel handeln. Das Element dient nur zur Unterscheidung von Nachrichten ein und desselben Teilnehmers an die gleiche bzw von der gleichen Anwendung über dieselbe Übermittlungsstelle.

Das Element teilnehmer steht für die Kennung des registrierten Teilnehmers (Anschriftcode), der Absender bzw Empfänger ist.

Das Element anwendung steht für die Justizanwendung, welche Absender (Ausgang) oder Empfänger (Eingang) der Nachricht ist, zB "FB" für Firmenbuch und "VJ" für Verfahrensautomation.

Beispiele:

```
mid://20081020.110.R123456.VJ@telekom.at  
mid://20081020.220.R123456.FB@imd.at  
mid://20081020.330.R123456.VJ@jusline.at  
mid://20081020.2233.N123456.FB@advokat.at  
mid://20081020.667.U123456.VJ@bmi.gv.at  
mid://20081020.14455.R123456.VJ@vj.erv.justiz.gv.at
```

3.2 Authentifizierung der externen Stellen

Jede Übermittlungsstelle und Direktteilnehmer-Institution erhält von der BRZG als Betreiber der Services ein Client-Zertifikat mit welchem Eingänge eingebracht und Ausgänge abgeholt werden können.

3.3 Autorisierung

Jede Übermittlungsstelle und Direktteilnehmer-Institution, die von den Services authentifiziert werden kann, ist autorisiert, den vollen Umfang der externen Schnittstelle zu nutzen. Die Services schränken die Berechtigung Eingänge einzubringen oder Ausgänge abzuholen nicht auf bestimmte Inhalte ein.

3.4 Nichtbestreitbarkeit

Nichtbestreitbarkeit, engl. *non-repudiation*, bedeutet, dass ein Empfänger immer nachweisen kann, einen bestimmten Inhalt von einem bestimmten Absender erhalten zu haben. Der Absender kann nicht behaupten, der Inhalt der Nachricht sei im Nachhinein verfälscht worden.

Die Nichtbestreitbarkeit eines Eingangs kann durch XML-Signatur (XML Dsig) erzielt werden. Dazu muss der Teilnehmer über einen offiziell zertifizierten Public Key verfügen, dessen Korrektheit durch die jeweilige Justizanwendung bei Empfang des Eingangs überprüft werden muss. Die XML-Signatur (vgl. [W3C-2002]) wird durch die Software des Teilnehmers hergestellt. In der Release 2.0 des elektronischen Rechtsverkehrs der Justiz wird die Nichtbestreitbarkeit nicht durch die Services unterstützt, sondern gegebenenfalls von der jeweiligen Justizanwendung implementiert.

3.5 Integrität und Verschlüsselung

Die Integrität einer Übersendung, also der Beweis, dass sie nicht verändert worden ist, und auch die Verschlüsselung, werden in der vorliegenden Version des elektronischen Rechtsverkehrs der Justiz ausschließlich über die SSL-Verbindungen zwischen Teilnehmer und Übermittlungsstellen und den Services gewährleistet.

Ähnlich wie die Nichtbestreitbarkeit kann auch die Integrität und Verschlüsselung über XML-Signatur und XML-Verschlüsselung erreicht werden. Der Vorteil einer solchen Lösung liegt darin, dass auch Verbindungsstrecken, die nicht mit SSL abgesichert sind, toleriert werden können. In der Release 2.0 des elektronischen Rechtsverkehrs wird die XML-Signatur und -Verschlüsselung jedoch nicht durch die Services unterstützt.

3.6 Asynchrone Prüfung und Verarbeitung

Die Services nehmen Nachrichten im Eingang entgegen und reichen diese nach Prüfung entsprechend dem ERV Nachrichtenschema und der spezifizierten Regeln für die Services an die prüfende und verarbeitende Justizanwendung weiter. Das Ergebnis der Prüfung und Verarbeitung muss später durch den Konzentrador explizit von den Services abgeholt werden. Nur ein positives Prüfergebnis bedeutet, dass der Eingang wirklich angenommen wurde. Dann gilt der bereits bei der Übermittlungsstelle vergebene Einbringungszeitpunkt. Bei einer Übermittlung einer Nachricht von einer Direktteilnehmer-Institution wird der Einbringungszeitpunkt immer von den Services vergeben. Bei einem

negativen Prüfergebnis muss der Eingang korrigiert und in einer neuen Nachricht mit anderer MessageId erneut eingebracht werden.

3.7 Sichere Übertragung

SOAP über HTTP ist an sich ein unverlässliches Protokoll. Ohne weitere Maßnahmen ist man nicht vor dem Verlust von Nachrichten geschützt. Es gibt zahlreiche Versuche, diesen Nachteil durch zusätzliche Protokolle zu beheben. Man vergleiche dazu [BEA-IBM-MS-Tibco-2004], [OASIS-2002] und [OASIS-2004].

Da noch kein öffentlicher Standard für ein Protokoll zur Erzielung von Reliable Messaging existiert, wird im elektronischen Rechtsverkehr der Justiz ein eigenes Protokoll umgesetzt. Dieses basiert auf eindeutigen Ids und Bestätigungen.

Das Einbringen von Nachrichten im Eingang und das Abholen vom Postkorb des Ausganges erfolgt über eine sichere Verbindung (SSL mit Server- und Client-Zertifikaten) und gewährleistet, dass Nachrichten weder im Eingang noch im Ausgang verloren gehen können.

Für einen Eingang wird eine Annahmebestätigung in Form einer Statusnachricht retourniert. Der Absender eines Eingangs muss diesen solange aufheben bis eine Annahmebestätigung zurückgeschickt wurde.

Die Abholung eines Ausganges muss mit einem Aufruf bestätigt werden, der innerhalb einer definierten Zeitspanne erfolgen muss. Ist das nicht der Fall, so muss davon ausgegangen werden, dass der Ausgang verlorengegangen ist und erneut zur Abholung bereitgestellt werden muss.

3.8 Fehlerverhalten

Wenn ein Aufruf der Service-Schnittstelle an einem Systemfehler scheitert, wird an den Aufrufer des Webservices ein SOAP-Fault mit einer `ERVException - Internal Server Error` - zurückgegeben. Der Aufrufer muss nach einigen Fehlern dieser Art annehmen, dass die Verarbeitung durch die Services zu diesem Zeitpunkt gar nicht möglich ist und sollte von weiteren Übermittlungen absehen. Solche technische Fehler werden intern als Alert an den Helpdesk und den Betriebsverantwortlichen geschickt, damit darauf reagiert werden kann.

Bei strukturellen Fehlern in der übermittelten Nachricht (zB Anhänge, die nicht in den Nachrichtendaten referenziert werden, Nachricht ist zu groß, Justizanwendung existiert nicht, usw) wird der Aufruf abgebrochen und ein SOAP-Fault mit der entsprechenden Fehlermeldung zurückgegeben. In diesem Fall gilt die Nachricht als nicht angenommen und kann sogar korrigiert mit derselben MessageId nochmals geschickt werden. Die Nachricht wird nicht an die Justizanwendung weitergereicht.

3.9 Übermittlung von Anhängen

Alle fachlichen Inhalte, die für die Services nicht, hingegen für die verarbeitende Justizanwendung schon relevant sind, werden als Anhang der Nachricht übermittelt. Dies betrifft die Payload mit dem fachlichen Inhalt der Eingabe, sowie die beigelegten Dokumente und deren Metadaten, wie auch Referenzen auf Dokumente in einem externen Archiv (sogenannte Archivreferenzen).

4 Implementierung

Die externe Services-Schnittstelle ist als Webservice implementiert, das SOAP Messages im Document/Literal Style mit Attachments empfangen kann.

4.1 SOAP-Message

Eine SOAP-Message im elektronischen Rechtsverkehr ist wie in der folgenden Abbildung gezeigt aufgebaut. In diesem Beispiel ist eine Message mit zwei Attachments dargestellt.

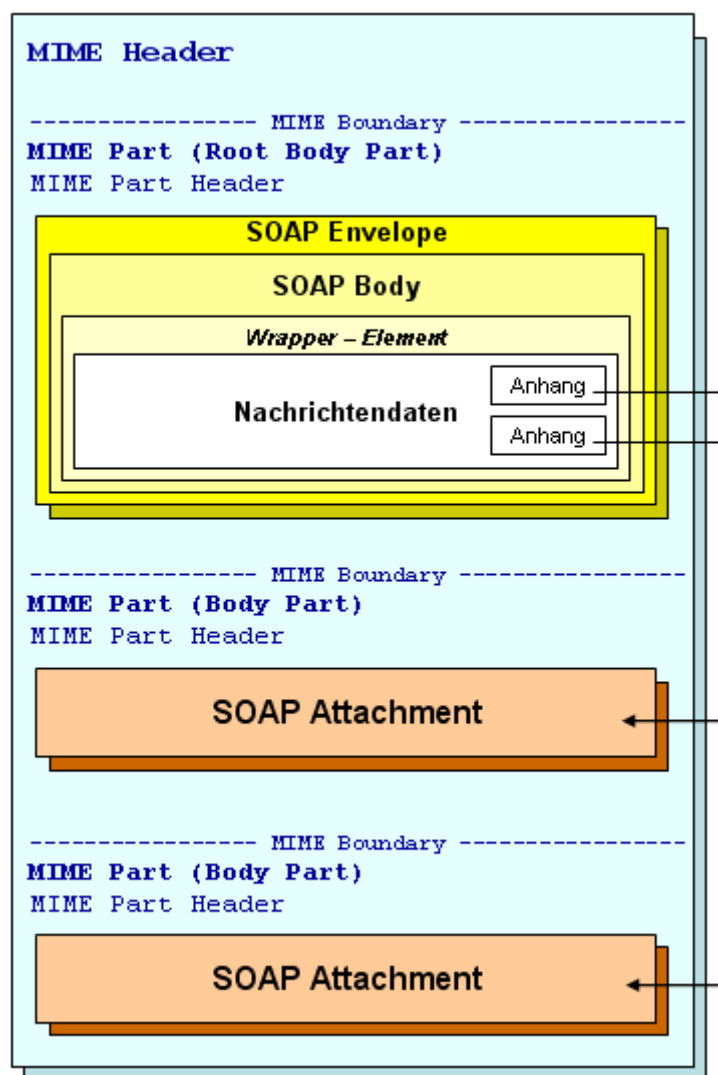


Abbildung 1: SOAP Message Struktur

Der SOAP-Envelope folgt dem Document/Literal-Style [Butek-2003], um die größtmögliche Kompatibilität zu Microsoft-SOAP-Clients zu gewährleisten. Der SOAP-Envelope enthält keinen SOAP-Header, sondern nur den SOAP-Body.

Dieser enthält ein „Wrapper“-Element, das erst die eigentliche XML-Struktur mit den Nachrichtendaten beinhaltet. Neben dem SOAP-Envelope kann die SOAP-Message mehrere SOAP-Attachments enthalten.

4.1.1 SOAP-Body

Der SOAP-Body besteht aus einem einzigen Element, dem „Wrapper“-Element, das die eigentlichen Request- bzw. Responsedaten kapselt. Für jeden Request bzw. Response gibt es ein definiertes „Wrapper“-Element. Alle innerhalb eines „Wrappers“ übertragenen Datenstrukturen entsprechen dem **ERV Nachrichtenschema**.

4.1.2 SOAP-Attachments

Jede Übersendung kann beliebig viele Anhänge beinhalten. Die Anhänge werden als MIME-Attachment-Parts gemäß der SOAP-With-Attachments Spezifikation gesendet (Vgl. dazu [W3C-2000]).

Auf jeden einzelnen Anhang wird von der im SOAP-Body übertragenen ERV Nachricht durch ein Element *Anhang* referenziert. Ein solches Element enthält neben dem Verweis auf den SOAP-Attachment-Part, in dem der zugehörige Anhang steht, auch noch die Art des Anhangs, welche Aufschluss über den fachlichen Inhalt gibt.

Ein Attachment entspricht hierbei den fachlichen Daten¹, die an eine Justizapplikation bzw. von einer Justizapplikation an einen Teilnehmer im elektronischen Rechtsverkehr übermittelt werden. Diese Daten werden im Binärformat übermittelt und vom ERV-Service im selben Format an das jeweilige Ziel weitergegeben.

Ein Attachment kann eine PAYLOAD (XML), ein DOKUMENT (PDF), eine ARCHIVREFERENZ (XML), METADATEN (XML) oder eine ZUSATZINFO (XML) darstellen.

Ein Attachment repräsentiert somit die eigentlichen Daten die zwischen einer Justizapplikation und einem Teilnehmer am elektronischen Rechtsverkehr ausgetauscht werden. Die ERV Nachrichtendaten stellen nur den Umschlag dar, innerhalb dessen diese Daten gekapselt sind. Innerhalb dieses Umschlags werden die Attachments referenziert (siehe Kapitel [4.2.1](#)).

4.1.3 SOAP-Fault

Bei Auftreten eines Fehlers oder einer Ausnahmesituation wird ein SOAP-Fault mit entsprechender Fehlermeldung und Fehlercode zurückgegeben. Genauerer siehe Kapitel 3.8 und im Dokument **Services Fehlermeldungen**.

Im Falle eines SOAP-Faults wird eine *ERVException*, welche die Elemente *Code* und *Message* zur Beschreibung des aufgetretenen Fehlers zurückgegeben. Die *ERVException* ist in der **ERV Webservicedefinition** definiert.

¹ Für Details zu den eigentlichen fachlichen Daten (z.B.: Mahnklage oder Firmenbuchantrag) siehe die entsprechende Schnittstellenbeschreibung der jeweiligen Anwendung unter <http://www.edikte.justiz.gv.at/erv>.

4.2 ERV Nachrichtenschema

Die Struktur der zu übertragenden Nachricht wird im *ERV Nachrichtenschema* definiert. Eine ERV Nachricht wird immer innerhalb des SOAP-Bodys in einem „Wrapper“-Element transportiert, das Aufschluss über die aufgerufene Operation gibt. Das „Wrapper“-Element selbst weist den Namespace des Services auf. Für alle Strukturen der ERV Nachricht ist ein eigener Namespace definiert, der eine Versionskennung enthält. Er hat folgende Struktur:

```
xmlns:erv="ns://erv.justiz.gv.at/Nachricht/V<Major>_<Minor>"
```

Major und *Minor* stellen die Versionskennung dar, die sich immer dann ändert, wenn die Datenstrukturen so erweitert werden, dass die Verarbeitung der Nachrichtendaten angepasst werden muss (nicht rückwärts kompatible Änderung).

Für kleinere Änderungen enthält das Schema ein `xsd:schema`-Attribut mit einer Versionskennung der Form

```
version="<Major>.<Minor>.<Fix>"
```

wobei *Major* und *Minor* mit den Versionsnummern im Namespace übereinstimmen und *Fix* für jede Änderung weitergezählt wird.

Im Folgenden werden die Datenstrukturen des Schemas beschrieben.

4.2.1 Nachrichtendaten

Die *Nachrichtendaten* definieren jene Struktur, die dazu dient, eine fachliche Nachricht, wie einen *Eingang*, *Ausgang*, *Nachweis* oder ein *Ergebnis* zu transportieren. Sie bilden das Konzept der eindeutigen Identifizierbarkeit durch die *MessageId* ab und enthalten als Nachrichtenstück eine konkrete Ausprägung von *Eingang*, *Ausgang*, *Nachweis* oder *Ergebnis*.

Überblick

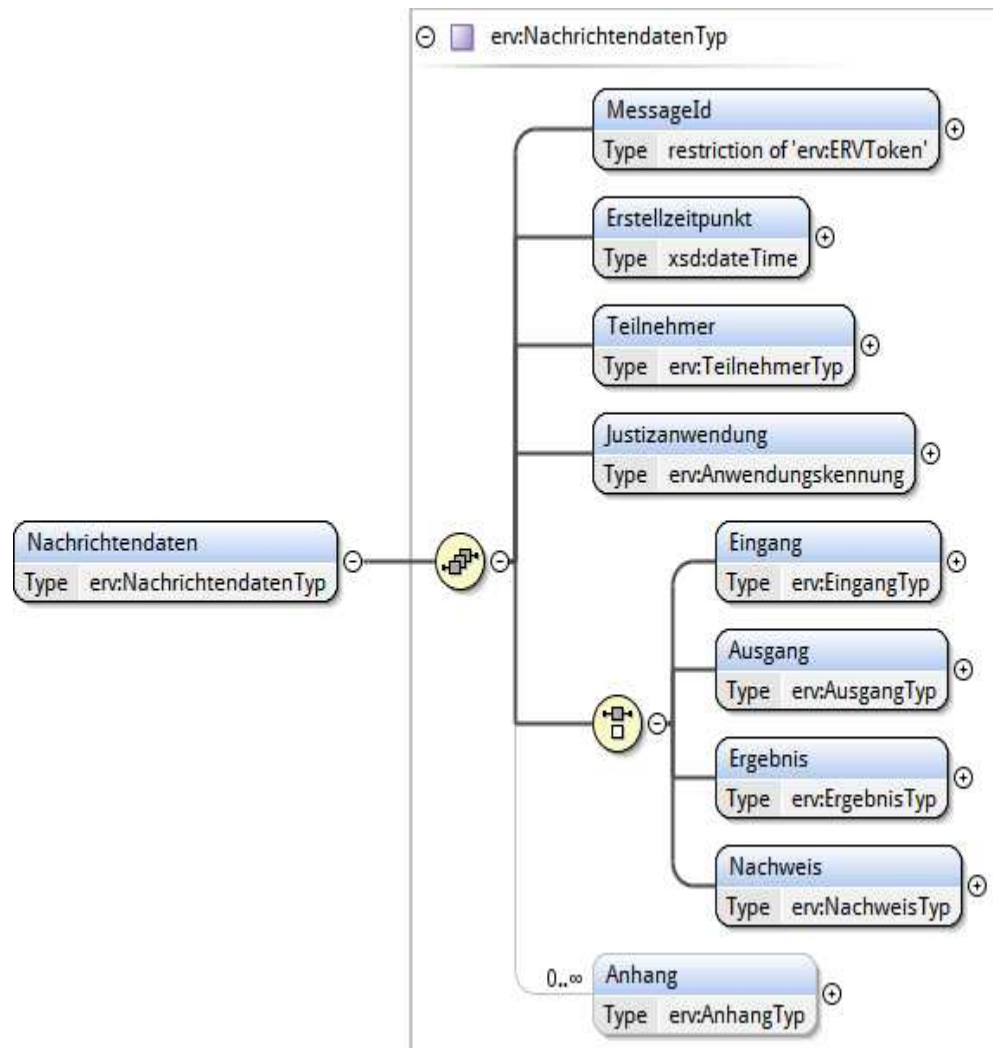
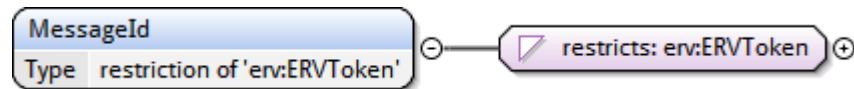


Abbildung 2: Schema der Nachrichtendaten

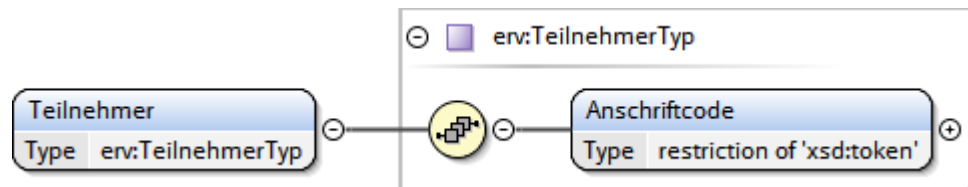
MessageId Eindeutige Kennung der Nachricht, siehe Kapitel 3.1



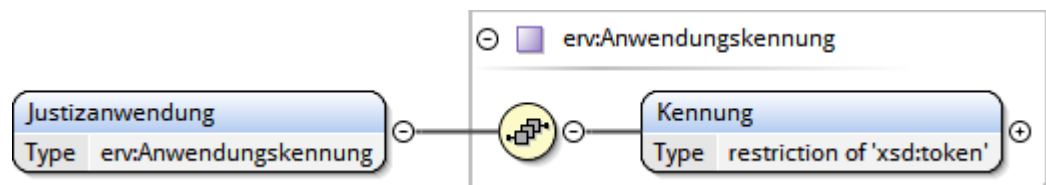
Erstellzeitpunkt Zeitpunkt, zu dem die Nachricht erstellt worden ist



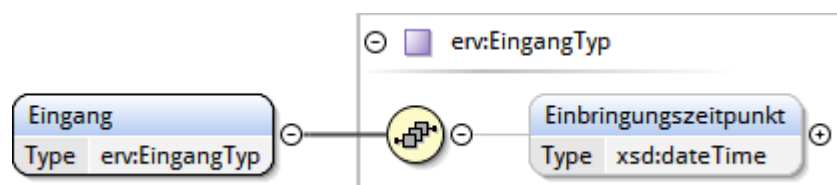
Teilnehmer Enthält im Element *Anschriftcode* die eindeutige Kennung des Teilnehmers



Justizanwendung Enthält im Element *Kennung* die eindeutige Bezeichnung für die adressierte Justizanwendung

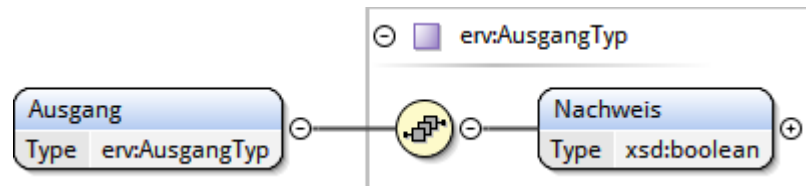


Eingang Nachrichtenstück für das Einbringen eines Einganges. Enthält beim Einbringen von einer Übermittlungsstelle aus den *Einbringungszeitpunkt*, welcher den Zeitpunkt der positiven Annahme der Nachricht bei der Übermittlungsstelle darstellt. Dieser Zeitpunkt ist dann gültig, wenn die Nachricht anschließend durch die Justizanwendung auch als gültig geprüft wird. (vgl. Kapitel 3.6)

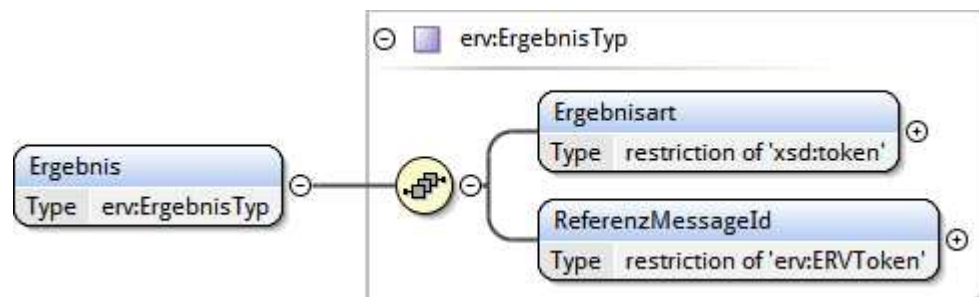


Ausgang

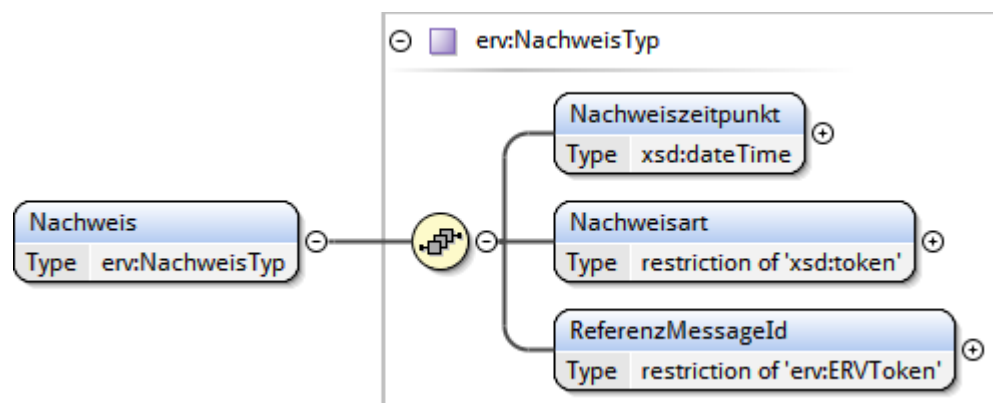
Nachrichtenstück für die Übertragung eines Ausgangs. Enthält das Element *Nachweis*, welches ausdrückt, ob für den Ausgang von der Übermittlungsstelle ein Nachweis erstellt und an die Services geschickt werden muss. Bei einer Direktteilnehmer-Institution ist nie ein Nachweis zu schicken.

**Ergebnis**

Nachrichtenstück für die Übertragung eines Prüfungsergebnisses eines Einganges durch eine Justizanwendung. Enthält die Elemente *Ergebnisart* (,OK' oder ,FEHLER') und *ReferenzMessageId*, welches die MessageId des geprüften Eingangs enthält.

**Nachweis**

Nachrichtenstück für die Übertragung eines Nachweises für die Zustellung eines Ausgangs. Enthält die Elemente *Nachweiszeitpunkt* (Zeitpunkt der Zurverfügungstellung des Ausgangs für den Teilnehmer), *Nachweisart* (,POSITIV' oder ,NEGATIV') und eine *ReferenzMessageId*, welche die MessageId des Ausgangs enthält, für den der Nachweis geschickt wird.



Anhang Für jeden als Attachment geschickten Anhang wird ein Element *Anhang* in den *Nachrichtendaten* übertragen. Es enthält die *AnhangId* (Content-ID des SOAP-Attachments, eindeutig innerhalb der SOAP-Message), die *ReferenzId*, mit welcher auf einen andere *AnhangId* verwiesen werden kann (zB Metadaten verweisen auf das Dokument zu dem sie gehören) und die *Art*, welche folgende Werte haben kann:

PAYLOAD: Anhang, der die eigentlichen fachlichen, strukturierten Daten enthält, also zum Beispiel die Eingabe oder Erledigung,

DOKUMENT: Anhang, der ein Dokument in binärer Form enthält,

XMLDOKUMENT: Anhang, der ein Dokument im XML Format enthält,

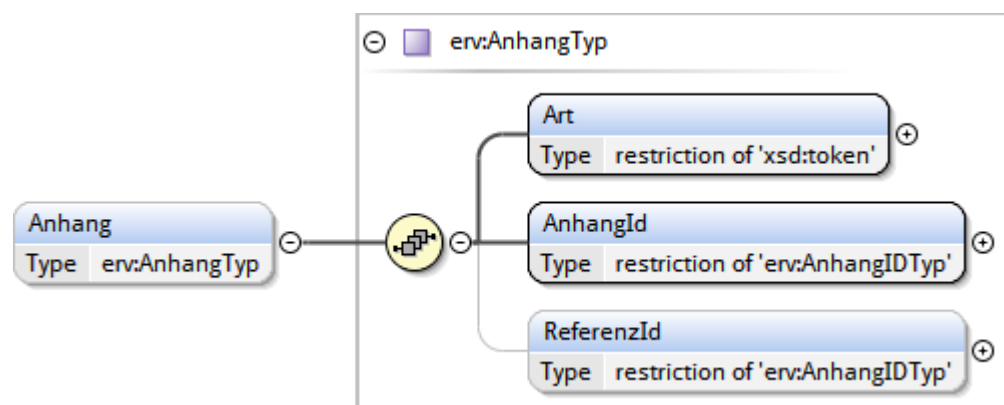
ARCHIVREFERENZ: Anhang mit einem Verweis auf ein Dokument in einem externen Archiv (wie zB CYBERDOC),

JUSTIZARCHIVREF: Anhang mit einem Verweis auf ein Dokument im Justizarchiv,

METADATEN: Anhang mit den zum Dokument-Anhang oder Archivreferenz-Anhang gehörenden Metadaten,

ZUSATZINFO: Frei definierbarer strukturierter Anhang zur Übermittlung von Daten, die weder direkt den fachlichen Inhalt betreffen, noch technische Relevanz haben (zB Abrechnungsdaten).

Die Services prüfen, ob die in den *Nachrichtendaten* beschriebenen Anhänge mitübertragen wurden und das erforderliche Format haben, kümmern sich aber nicht um den in den SOAP-Attachments enthaltenen Inhalt.



Der Typ *Anhang* innerhalb der ERV-Nachrichtendaten verweist also auf jeweils genau ein SOAP-Attachment und beschreibt die Eigenschaften dieses Attachments. Ausserdem dient es der Referenzierung zwischen Umschlag (ERV-Nachrichtendaten) und den tatsächlichen fachlichen Daten.

Die *AnhangId* innerhalb des Typs *Anhang* verweist dabei direkt auf die *content_id* eines Attachments in der SOAP-Nachricht und stellt über diesen Verweis den Konnex zwischen einem Element *Anhang* und dem zugehörigen SOAP-Attachment in einer übermittelten Nachricht her.

Das Datenfeld *Art* des Typs *Anhang* gibt dabei Aufschluss über den Inhalt eines referenzierten SOAP-Attachments.

4.2.2 Nachrichtenkennung

Die *Nachrichtenkennung* wird dann geschickt, wenn ein Ergebnis für einen bestimmten Eingang abgeholt oder die Abholung eines bestimmten Ausgangs bestätigt werden soll. Sie referenziert also den entsprechenden *Eingang* bzw. *Ausgang* mithilfe der eindeutigen *MessageId*.

Überblick

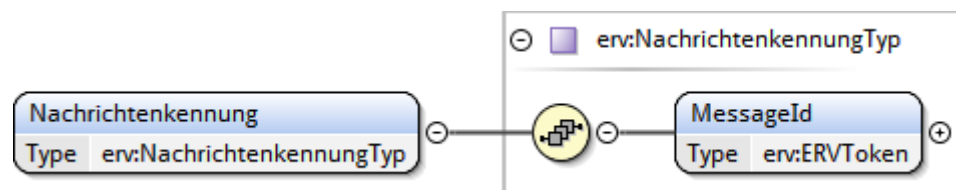


Abbildung 3: Schema der Nachrichtenkennung

MessageId

Die eindeutige Kennung des Eingangs für welchen ein Ergebnis abgeholt werden soll bzw des zu bestätigenden Ausgangs.

4.2.3 Eingangsstatus

Der *Eingangsstatus* wird als Antwort auf das Einbringen eines Eingangs oder Nachweises zurückgegeben. Dies erfüllt die Funktion einer Annahmebestätigung (vgl. Kapitel 3.7).

Überblick

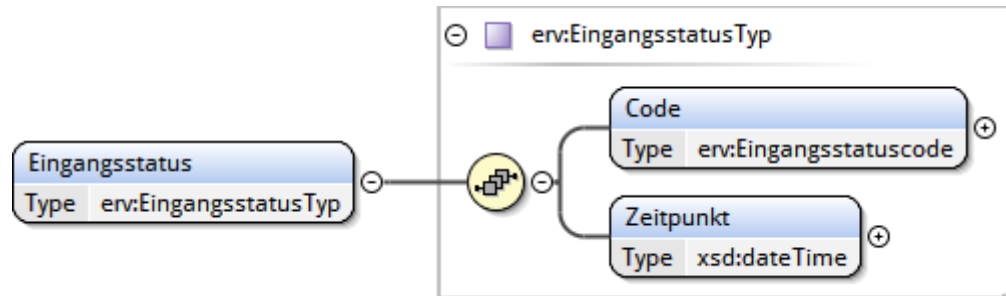


Abbildung 4: Schema des Eingangsstatus

Code

Enthält den Statuscode des Einganges.

ANGENOMMEN für von den Services akzeptierte Eingänge, die an die Justizanwendung zur fachlichen Prüfung und Verarbeitung weitergereicht wurden.

VALIDIERT für bereits von der Justizanwendung geprüfte Eingänge, für welche ein Ergebnis zur Abholung bereitliegt.

Zeitpunkt

Der Annahmezeitpunkt bei Code ‚ANGENOMMEN‘ bzw Validierungszeitpunkt bei Code ‚VALIDIERT‘.

4.2.4 Ausgangsstatus

Der *Ausgangsstatus* wird als Antwort auf das Bestätigen eines abgeholten Ausgangs zurückgegeben.

Überblick

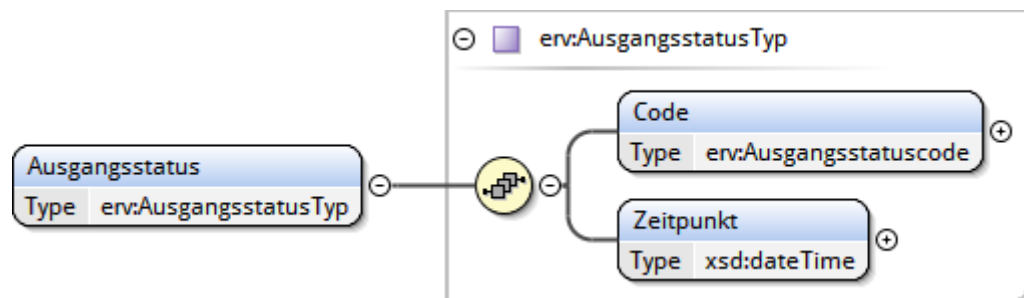


Abbildung 5: Schema des Ausgangsstatus

Code

Enthält den Statuscode des Ausgangs.

BEREITGESTELLT für Ausgänge, die noch nicht abgeholt oder bereits abgeholt, aber nicht im definierten Zeitintervall (,AUSGANG_BESTAETIGUNG_TIMEOUT') bestätigt wurden.

BESTAETIGT für Ausgänge, die abgeholt und bestätigt wurden.

Zeitpunkt

Der Zeitpunkt der Bereitstellung in den Services zur Abholung bei Code ,BEREITGESTELLT' bzw Bestätigungszeitpunkt bei Code ,BESTAETIGT'

4.2.5 Postkorb

Das Element *Postkorb* wird als Antwort auf die Abfrage der im Ausgang zur Abholung bereitgestellten Ausgangsstücke zurückgegeben.

Überblick

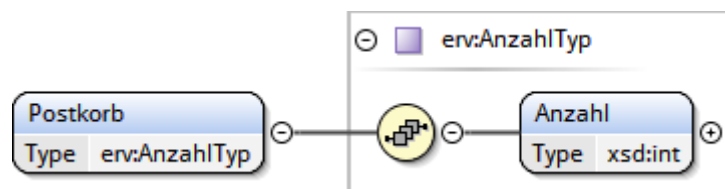


Abbildung 6: Schema der Anzahlinformation

Anzahl

Enthält als xsd:int die Anzahl der im Ausgang zur Abholung bereitgestellten Ausgangsstücke.

4.3 Webservice Definition

Das Webservice stellt sechs Operationen zur Verfügung:

- Eingang einbringen
- Ergebnis abholen
- Status abfragen
- Ausgang abholen
- Ausgang bestätigen
- Anzahl abfragen

Diese sechs Operationen werden durch die **ERV Webservedefinition** definiert und nutzen das **ERV Nachrichtenschema**.

4.3.1 Eingang einbringen

Mit der Operation *EingangEinbringen* kann der Konzentrator mit seinem Zertifikat einen Eingang bzw einen Nachweis einbringen, welcher nach einer strukturellen Prüfung an die in den Nachrichtendaten angegebene Justizanwendung weitergereicht wird. Es wird eine Annahmebestätigung zurückgegeben.

4.3.1.1 Request

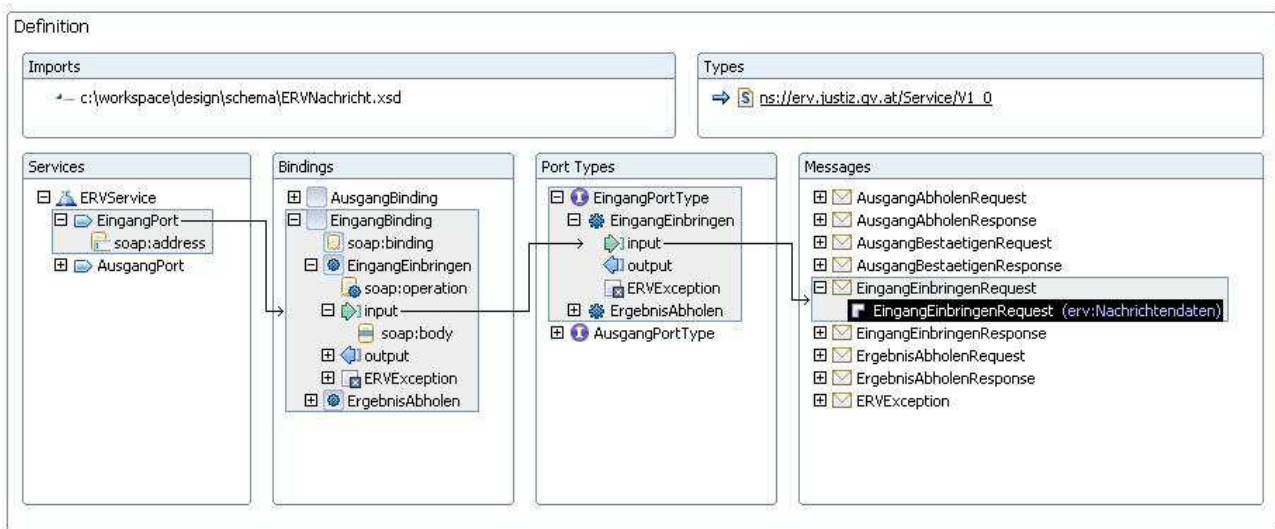


Abbildung 7: Request "EingangEinbringen"

Wrapper-Element: EingangEinbringenRequest

SOAP-Body-Content: *Nachrichtendaten* mit der Ausprägung *Eingang* oder *Nachweis*

SOAP-Attachments: Bei einem *Eingang* gibt es genau einen Anhang der Art PAYLOAD und beliebig viele Anhänge mit Art DOKUMENT bzw ARCHIVREFERENZ und entsprechend genauso viele Anhänge mit Art METADATEN, die darauf referenzieren. Optional kann ein Anhang mit der Art ZUSATZINFO geschickt werden. Bei einem *Nachweis* sind alle Informationen in den Nachrichtendaten enthalten, weshalb kein Anhang übertragen werden kann.

4.3.1.2 Response

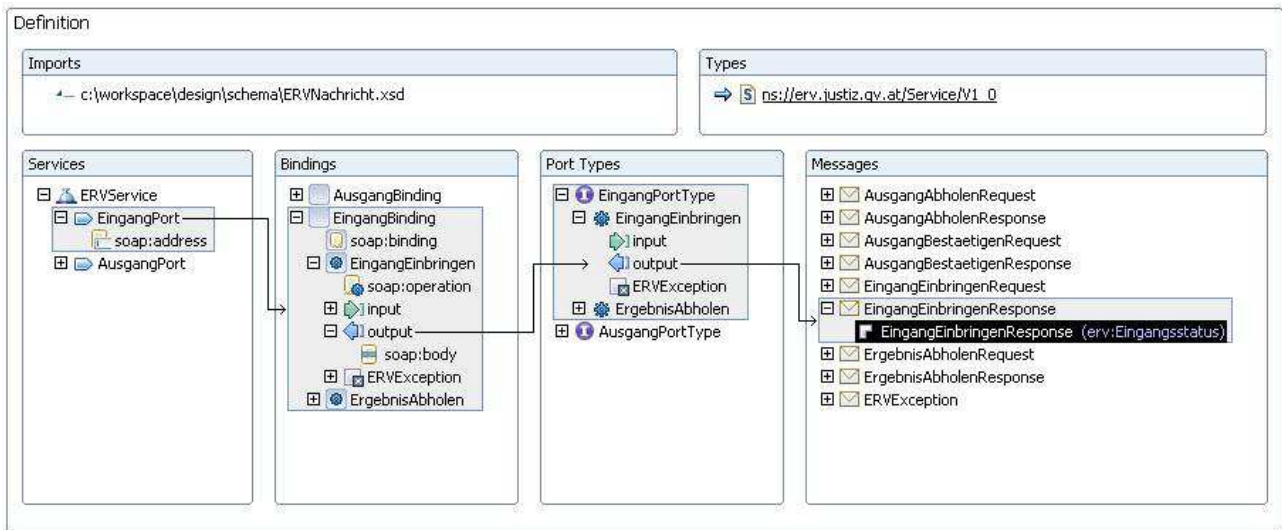


Abbildung 8: Response "EingangEinbringen"

Wrapper-Element: EingangEinbringenResponse

SOAP-Body-Content: *Eingangsstatus*

Beim ersten Einbringen eines *Eingangs* oder eines *Nachweises* wird ein *Eingangsstatus* mit ‚ANGENOMMEN‘ und dem Annahmezeitpunkt zurückgegeben.

Ist inzwischen ein Ergebnis der Prüfung des *Eingangs* von der Justizanwendung in den Services bereitgestellt, so wird ein *Eingangsstatus* mit ‚VALIDIERT‘ und dem Validierungszeitpunkt zurückgegeben.

SOAP-Attachments: keine

4.3.1.3 Verhalten

Wird ein und derselbe *Eingang* bzw. *Nachweis* mehrmals eingebracht (erkennbar an der gleichen *MessageId*), so wird nur beim ersten Mal, wenn der *Eingang* bzw. *Nachweis* verarbeitet wurde, der *Eingangsstatus* zurückgegeben. Alle weiteren Einbringungen werden mit einem Fehler zurückgewiesen.

Das Fehlerverhalten ist wie in Kapitel 3.8 beschrieben.

4.3.2 Ergebnis abholen

Mit der Operation *ErgebnisAbholen* holt ein Konzentrator mit seinem Zertifikat ein von einer Justizanwendung bei Prüfung eines Einganges erstelltes Ergebnis von den Services ab. Es wird für die in der *Nachrichtenkennung* übergebene *MessageId* des Einganges das entsprechende Ergebnis zurückgegeben.

4.3.2.1 Request

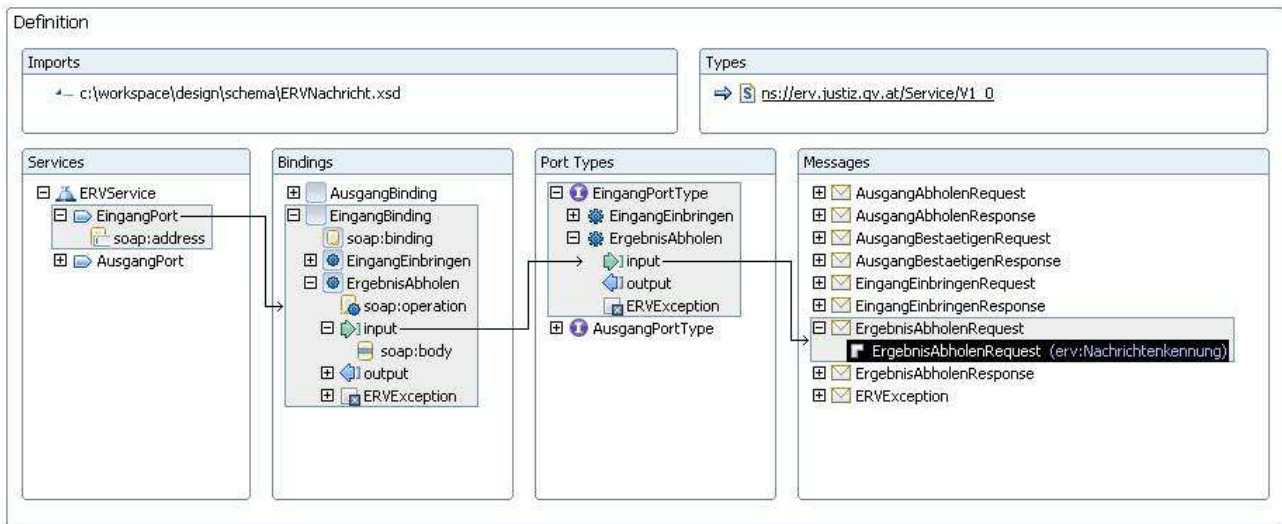


Abbildung 9: Request "ErgebnisAbholen"

Wrapper-Element: ErgebnisAbholenRequest

SOAP-Body-Content: *Nachrichtenkennung*, die als *MessageId* die *MessageId* des zuvor eingebrachten *Einganges* enthält, für welchen das erstellte Ergebnis abgeholt werden soll.

SOAP-Attachments: keine

4.3.2.2 Response

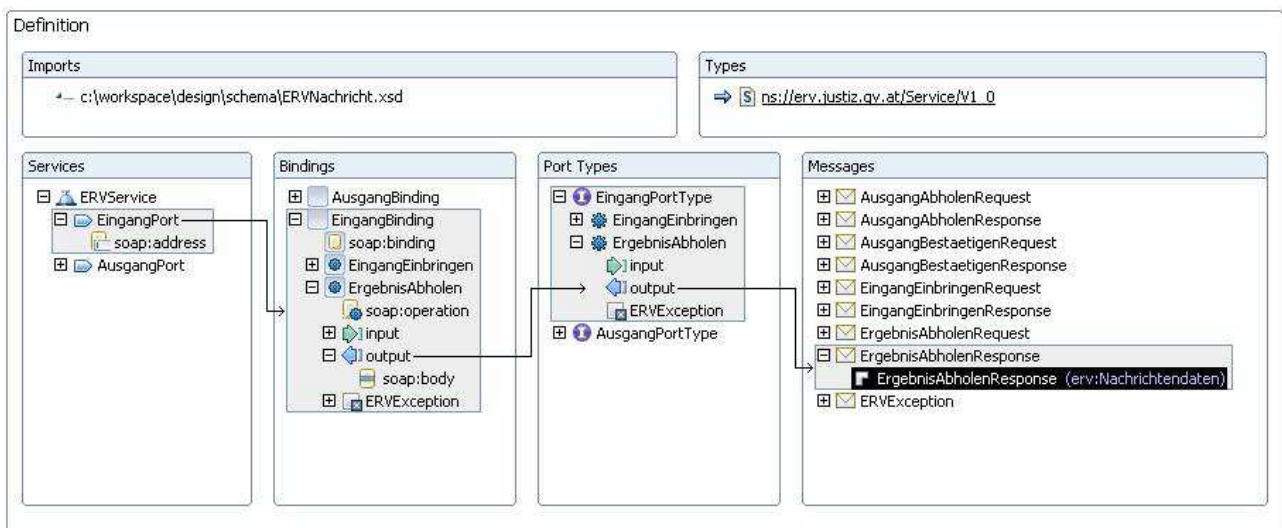


Abbildung 10: Response "ErgebnisAbholen"

Wrapper-Element: ErgebnisAbholenResponse

SOAP-Body-Content: *Nachrichtendaten* mit der Ausprägung *Ergebnis*. Hat das Element *Ergebnisart* den Wert FEHLER, so sind im Anhang mit der Payload die Fehlermeldungen zu finden.

SOAP-Attachments: Mit einem Ergebnis wird im Fall von aufgetretenen Fehlern genau ein Anhang mit der Art PAYLOAD übertragen, welcher die von der

Justizanwendung definierten Fehlermeldungen in strukturierter Form als XML enthält.

4.3.2.3 Verhalten

Ergebnisse werden nur für eingebrachte Nachrichten vom Typ *Eingang* erstellt. Für einen *Nachweis* gibt es kein *Ergebnis* zur Abholung.

Ein *Ergebnis* kann beliebig oft abgeholt werden. Wird versucht für eine *MessageId* ein Ergebnis abzuholen, das noch nicht von der Anwendung erstellt wurde oder, wo es für die *MessageId* gar keinen *Eingang* gibt, so wird eine entsprechende Fehlermeldung in einem SOAP-Fault zurückgegeben.

Das Fehlerverhalten ist wie in Kapitel 3.8 beschrieben.

4.3.3 Status abfragen

Mit der Operation *StatusAbfragen* kann der Status einer eingebrachten Nachricht abgefragt werden. Es wird für die in der *Nachrichtenkennung* übergebene *MessageId* des Einganges der entsprechende *Eingangsstatus* zurückgegeben.

4.3.3.1 Request

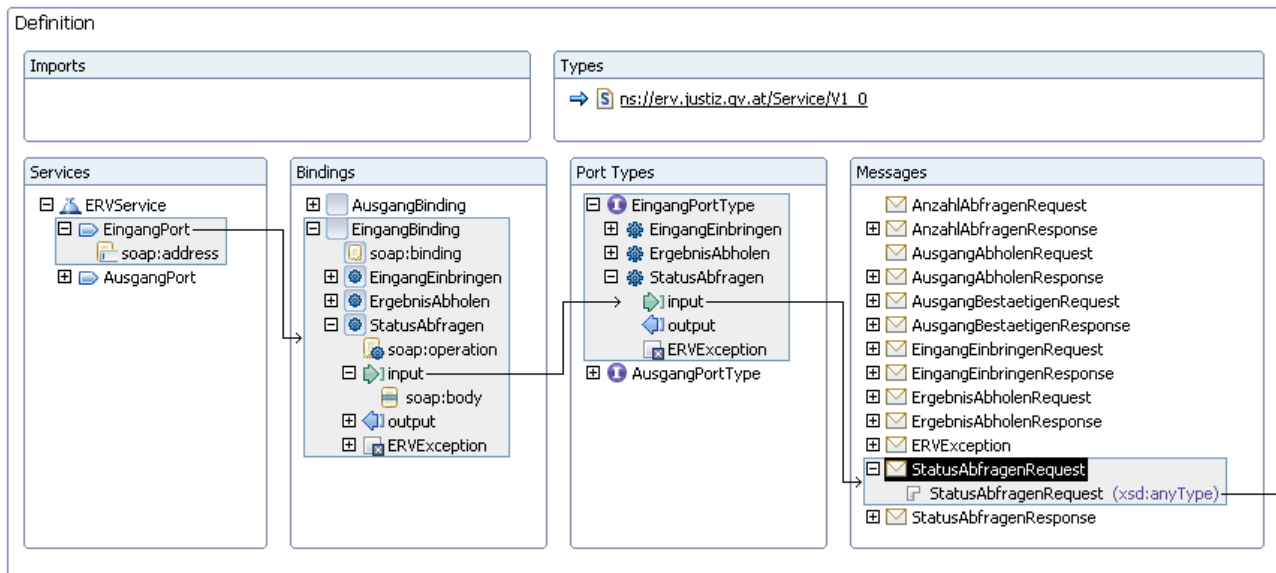


Abbildung 11: Request "StatusAbfragen"

Wrapper-Element: StatusAbfragenRequest

SOAP-Body-Content: *Nachrichtenkennung*, die als *MessageId* die *MessageId* des zuvor eingebrachten *Einganges* enthält, für welchen der Status abgefragt werden soll.

SOAP-Attachments: keine

4.3.3.2 Response

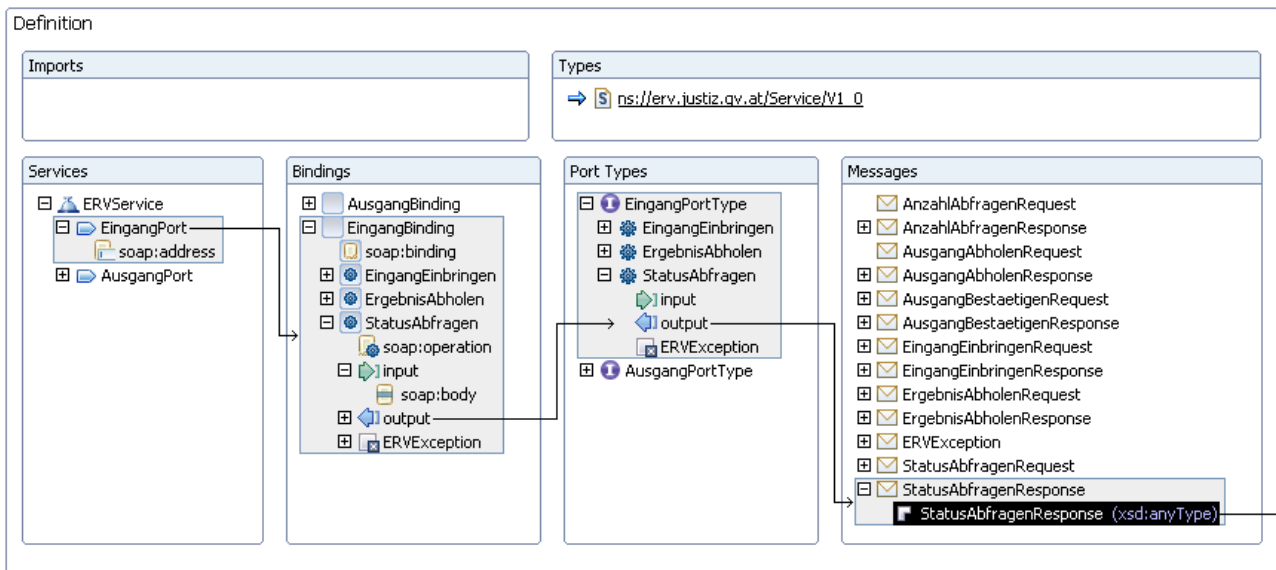


Abbildung 12: Response "StatusAbfragen"

Wrapper-Element: StatusAbfragenResponse

SOAP-Body-Content: *Eingangsstatus*, welcher den Statuscode („ANGENOMMEN“ oder „VALIDIERT“) und den Zeitpunkt der letzten Statusänderung für den abgefragten Eingang enthält..

SOAP-Attachments: keine

4.3.3.3 Verhalten

Die Statusabfrage funktioniert nur für im Eingang eingebrachte Nachrichten und kann beliebig oft durchgeführt werden. Wird versucht eine Statusabfrage für eine MessageId, für welche es gar keinen *Eingang* gibt, durchzuführen, so wird eine entsprechende Fehlermeldung in einem SOAP-Fault zurückgegeben.

Das Fehlerverhalten ist wie in Kapitel 3.8 beschrieben.

4.3.4 Ausgang abholen

Mit der Operation *AusgangAbholen* holt ein Konzentrator mit seinem Zertifikat einen von einer Justizanwendung erstellten Ausgang ab. Es wird der nächste bereitgestellte Ausgang für den Konzentrator zurückgegeben.

4.3.4.1 Request

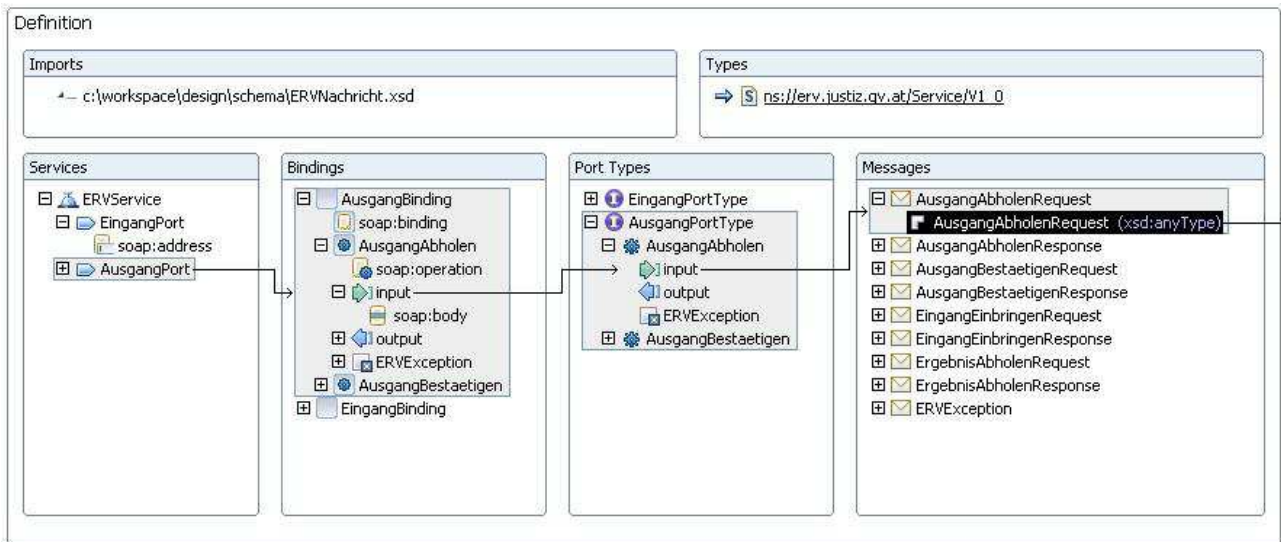


Abbildung 13: Request "AusgangAbholen"

Wrapper-Element: AusgangAbholenRequest

SOAP-Body-Content: leer.

SOAP-Attachments: keine

4.3.4.2 Response

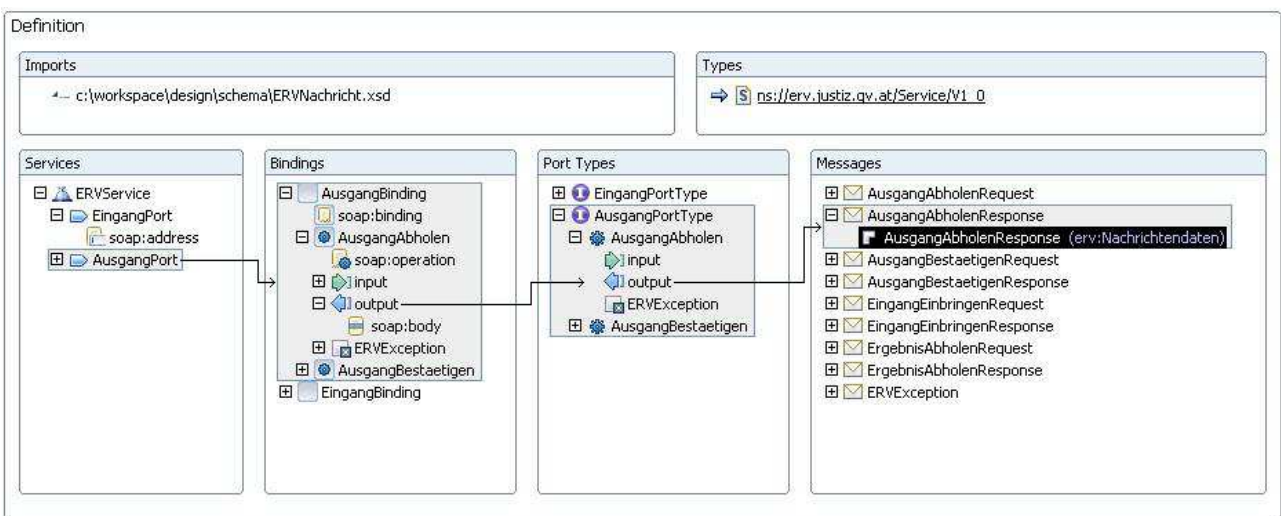


Abbildung 14: Response "AusgangAbholen"

Wrapper-Element: AusgangAbholenResponse

SOAP-Body-Content: *Nachrichtendaten* mit der Ausprägung *Ausgang*.

SOAP-Attachments: Bei einem *Ausgang* gibt es genau einen Anhang der Art PAYLOAD und beliebig viele Anhänge mit Art DOKUMENT.

4.3.4.3 Verhalten

Auf den Request wird der nächste *Ausgang* für die im Zertifikat enthaltene Kennung des Konzentrators zurückgegeben. Dabei wird der älteste im Postkorb verfügbare *Ausgang* zurückgegeben. Die Abholung muss im konfigurierten

Zeitintervall ‚AUSGANG_BESTAETIGUNG_TIMEOUT‘ durch die Operation *AusgangBestaetigen* bestätigt werden.

Das Fehlerverhalten ist wie in Kapitel 3.8 beschrieben.

4.3.5 Ausgang bestätigen

Mit der Operation *AusgangBestaetigen* bestätigt ein Konzentrator mit seinem Zertifikat den Erhalt eines abgeholten Ausgangs. Es wird für die übergebene MessageId des Ausgangs eine Abholbestätigung zurückgegeben.

4.3.5.1 Request

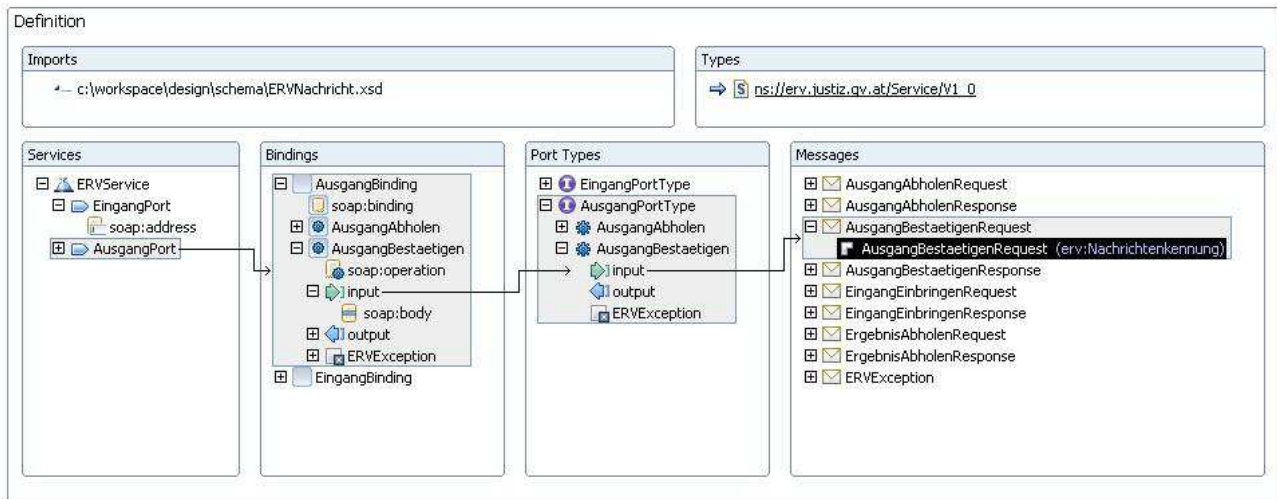


Abbildung 15: Request "AusgangBestaetigen"

Wrapper-Element: AusgangBestaetigenRequest

SOAP-Body-Content: *Nachrichtenkennung*, die als *MessageId* die MessageId des zuvor abgeholten *Ausgangs* enthält, dessen Abholung bestätigt werden soll.

SOAP-Attachments: keine

4.3.5.2 Response

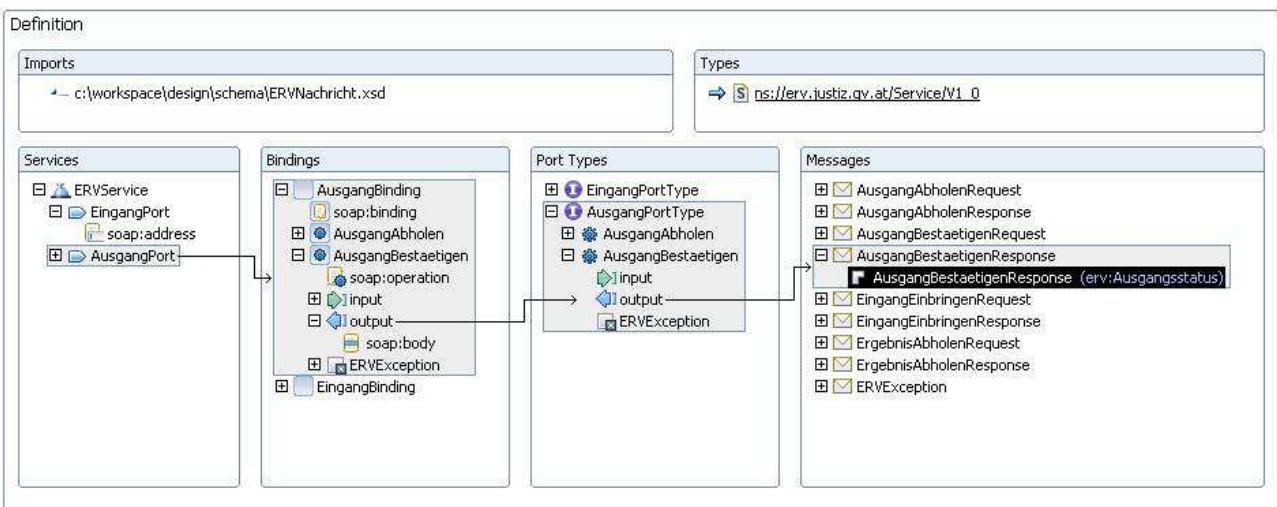


Abbildung 16: Response "AusgangBestaetigen"

Wrapper-Element: AusgangBestaetigenResponse**SOAP-Body-Content:** *Ausgangsstatus*

Beim erfolgreichen Bestätigen der Abholung eines *Ausgangs* wird ein *Ausgangsstatus* mit ‚BESTAETIGT‘ und dem Bestätigungszeitpunkt zurückgegeben.

Bei abgelaufenem Abhol-Timeout wird ein *Ausgangsstatus* mit ‚BEREITGESTELLT‘ und dem Zeitpunkt der Bereitstellung zurückgegeben.

SOAP-Attachments: keine**4.3.5.3 Verhalten**

Bei abgelaufenem Abhol-Timeout (‚AUSGANG_BESTAETIGUNG_TIMEOUT‘) wird der *Ausgang* erneut durch die Services zur Abholung bereitgestellt und muss mit der Operation *AusgangAbholen* abgeholt werden. Erst dann kann er wieder bestätigt werden.

Bei Bestätigung einer MessageId, für welche kein *Ausgang* im Postkorb bereitgestellt ist, wird ein entsprechender Fehler in einem SOAP-Fault zurückgegeben.

Das Fehlerverhalten ist wie in Kapitel 3.8 beschrieben.

4.3.6 Anzahl abfragen

Mit der Operation *AnzahlAbfragen* ist es möglich, die Anzahl der im Postkorb zur Abholung bereitgestellten Ausgangsstücke abzufragen.

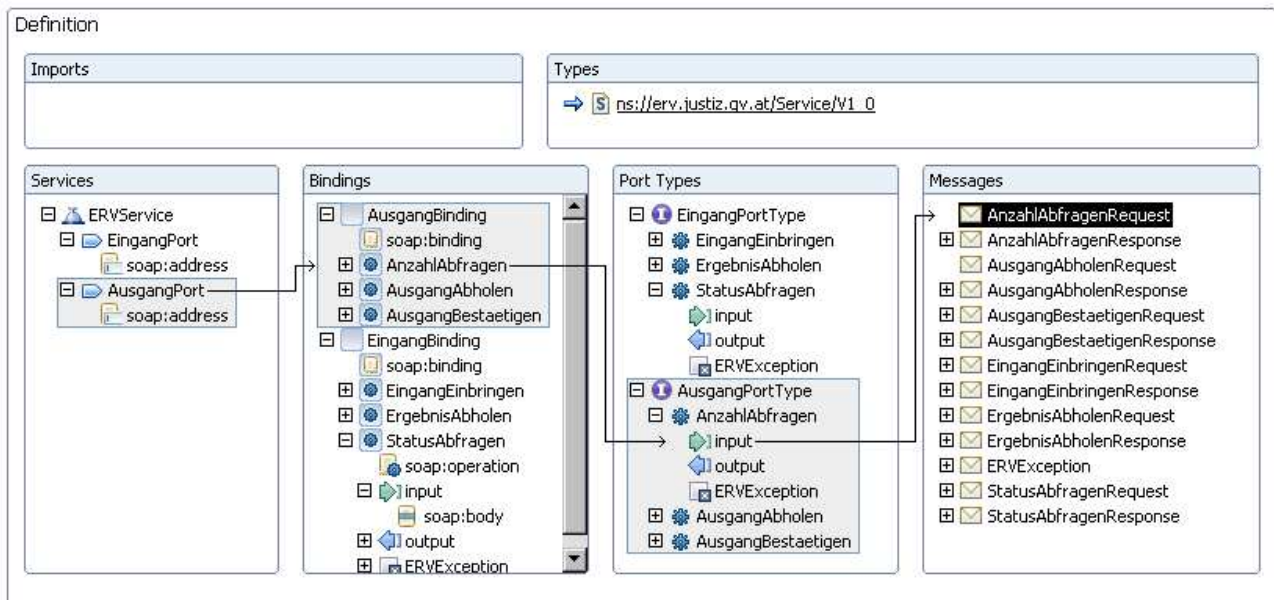
4.3.6.1 Request

Abbildung 17: Request "AnzahlAbfragen"

Wrapper-Element: AnzahlAbfragenRequest**SOAP-Body-Content:** leer.**SOAP-Attachments:** keine

4.3.6.2 Response

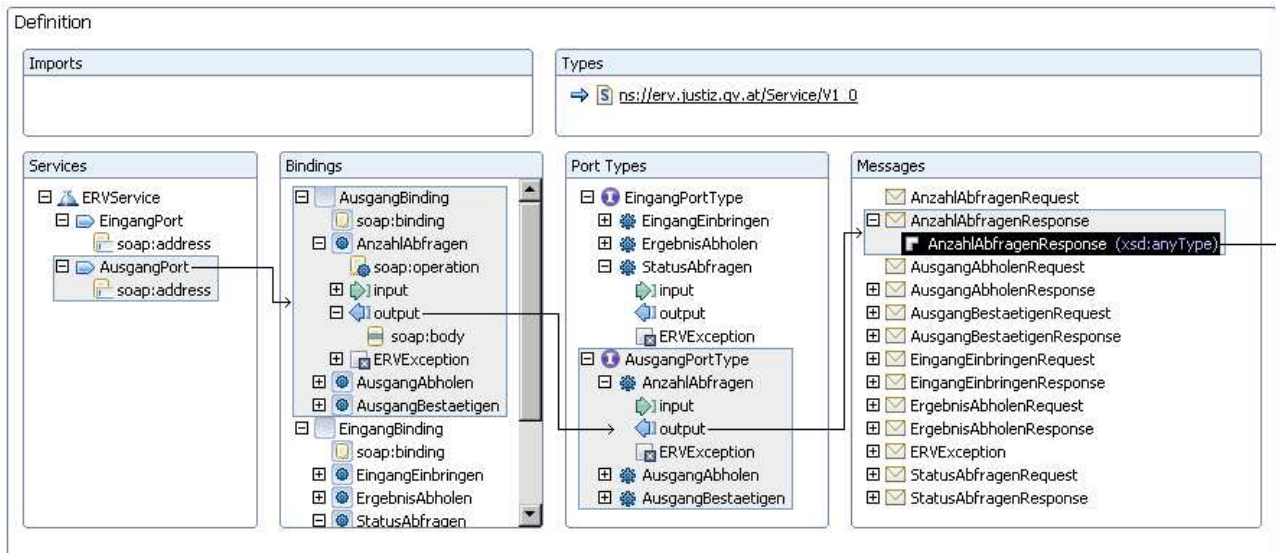


Abbildung 18: Response "AnzahlAbfragen"

Wrapper-Element: `AnzahlAbfragenResponse`

SOAP-Body-Content: *Postkorb* mit einem Element *Anzahl*.

SOAP-Attachments: keine

4.3.6.3 Verhalten

Auf den Request wird ein XML Dokument, welches die Anzahl der für den Konzentrador bereitgestellten Ausgangsstücke enthält zurückgegeben.

Das Fehlerverhalten ist wie in Kapitel 3.8 beschrieben.

4.3.7 Operationen im Überblick

Die folgende Tabelle listet alle durch das Webservice zur Verfügung gestellten Operationen mit deren Signatur auf.

Operation	Message-Type	Message/Wrapper-Element	Parameter-Type	Parameter	Kardinalität
EingangEinbringen	Input	EingangEinbringenRequest	Body	Nachrichtendaten (Ausprägung <i>Eingang</i> oder <i>Nachweis</i>)	1
			Attachment	Payload (Eingabe)	1
			Attachment	Dokument	n
			Attachment	Archivreferenz	n
			Attachment	Metadaten	n
	Output	EingangEinbringenResponse	Body	Eingangsstatus	1
ErgebnisAbholen	Input	ErgebnisAbholenRequest	Body	Nachrichtenennung	1
			Body	Nachrichtendaten (Ausprägung <i>Ergebnis</i>)	1
			Attachment	Payload (Ergebnismeldungen)	1
StatusAbfragen	Input	StatusAbfragenRequest	Body	Nachrichtenennung	1
	Output	StatusAbfragenResponse	Body	Eingangsstatus	1

AusgangAbholen	Input	AusgangAbholenRequest	Body		1
	Output	AusgangAbholenResponse	Body	Nachrichtendaten (Ausprägung <i>Ausgang</i>)	1
			Attachment	Payload (Erledigung)	1
			Attachment	Dokument	n
			Attachment	Archivreferenz	n
			Attachment	Metadaten	n
AusgangBestaetigen	Input	AusgangBestaetigenRequest	Body	Nachrichtenkennung	1
	Output	AusgangBestaetigenResponse	Body	Ausgangsstatus	1
AnzahlAbfragen	Input	AnzahlAbfragenRequest	Body		1
	Output	AnzahlAbfragenResponse	Body	Anzahlinformation	1

Tabelle 1: Überblick Webservice Operationen

5 Anhang

5.1 Referenzen

5.1.1 Dokumente

Referenz	Dateiname	Beschreibung des Inhalts
ERV Nachrichtenschema	ERVNachricht.xsd	XSD Schema von Nachrichtenennung, Nachrichtendaten, Eingangsstatus und Ausgangsstatus inklusive der verwendeten Typen
ERV Webservicedefinition	ERVService.wsdl	WSDL des ERV Webservices, das die Ports, Bindings und Messages für die Operationen definiert. Verwendet das ERV Nachrichtenschema .
Services Regeln	Services_Regeln.pdf	Enthält alle Regeln, die in den Services geprüft werden.
Services Fehlermeldungen	Services_Fehlermeldungen.pdf	Enthält alle durch eine Verletzung einer Regel auftretenden Fehlermeldungen, die von den Services an den Aufrufer zurückgegeben werden können.
Services Konfiguration	Services_Configuration.xls	Tabelle mit allen konfigurierbaren Werten der Services, wie Domain-Namen, Anwendungskennungen, Timeouts, usw

Tabelle 2: Referenzen auf externe Dokumente

5.1.2 Modelle

Modell	Dateiname	Beschreibung des Inhalts
ERV Nachrichtenschema	Schema dokumentation.zip	Modell des ERV Nachrichtenschemas
TNVZ Nachrichtenschema	Schema dokumentation.zip	Modell des Teilnehmerverzeichnis Nachrichtenschemas

Tabelle 3: Referenzen auf Modelle

5.1.3 Literatur

[BEA-IBM-MS-Tibco-2004] BEA, IBM, Microsoft, Tibco: *Web Services Reliable Messaging Protocol*, March 2004, Internet: <ftp://www6.software.ibm.com/software/developer/library/ws-reliablemessaging200403.pdf>

[Butek-2003] Butek, Russel: "Which style of WSDL should I use?", IBM Developerworks, 31.Oct 2003, Internet: <http://www-128.ibm.com/developerworks/webservices/library/ws-whichwsdl>

[Nielsen-2002] Nielsen, Sanders, Butek, Nash: *Direct Internet Message Encapsulation (DIME)*, Internet Draft, Microsoft/IBM, 2003. Internet: <http://msdn.microsoft.com/library/en-us/dnglobspec/html/draft-nielsen-dime-02.txt>.

[OASIS-2002] OASIS, *Message Service Specification V0.2*, April 2002, Internet: http://www.oasis-open.org/committees/ebxml-msg/documents/ebMS_v2_0.pdf

[OASIS-2004] OASIS, *Web Services Reliability Specification 1.1*, 2004, Internet: <http://www.oasis-open.org/committees/download.php/9330/WS-Reliability-CD1.086.zip>

[W3C-2000] W3C, *SOAP Messages with Attachments*, December 2000. Internet: <http://www.w3.org/TR/SOAP-attachments>

[W3C-2002] W3C, *XML-Signature Syntax and Processing*, February 2002, Internet:
<http://www.w3.org/TR/xmlsig-core/>

[W3C-2004] W3C, *Web Services Architecture Usage Scenarios*, February 2004, Internet:
<http://www.w3.org/TR/2004/NOTE-ws-arch-scenarios-20040211/>