

Elektronische Zustellung Message Spezifikation		Konvention
		zusemsg-1.4.0
		Ergebnis der AG
Kurzbeschreibung	Diese Spezifikation beschreibt die Kommunikation zwischen Versenderapplikationen und Zustelldiensten zur Übergabe von Zustellungen und zur Übermittlung von Zustellnachweisen.	
Autor(en):	Arne Tauber	Projektteam / Arbeitsgruppe:
	Thomas Rössler Peter Reichstädter	AG-ZUSE / AG-II
Beiträge von:	Arno Hollosi, Larissa Naber, Michael Liehmann	

Version 1.4.0 : **02.02.2012**

Fristablauf: **TT.MM.JJJJ**

Abgelehnt von:

*(Länderangabe bei ablehnender Stellungnahme)*

Unter-Version ... : **TT.MM.JJJJ**

Fristablauf: **TT.MM.JJJJ**

Abgelehnt von:

*(Länderangabe bei ablehnender Stellungnahme)*

Detail-Version ... : **TT.MM.JJJJ**

Fristablauf: **TT.MM.JJJJ**

Anmerkungen:

*(Detailangaben zur Freigabe)*

## Inhaltsverzeichnis

<b>1 Funktionaler Überblick .....</b>	<b>3</b>
1.1 Einführung.....	3
<b>2 Datenformat der (un-)verschlüsselten Sendung .....</b>	<b>5</b>
2.1 Container für Sendungsteile.....	5
2.2 Verschlüsselung.....	7
<b>3 Datenformat für die Dokumentenanlieferung (DeliveryRequest) .....</b>	<b>9</b>
3.1 Sender .....	11
3.2 Empfänger .....	13
3.3 Zustell-Metainformation .....	13
3.4 Sendungs-Referenz .....	15
3.5 Empfängerverständigungsintervalle.....	16
<b>4 Datenformat für Erfolgs- und Fehlermeldungen in der Dokumentenanlieferung.....</b>	<b>17</b>
<b>5 Datenformat für Nachrichten über Callback-Attachments (asynchrone Kommunikation) .....</b>	<b>22</b>
<b>6 Datenformat für den Zustellnachweis .....</b>	<b>23</b>
6.1 Erfolgsfall (Success) .....	25
6.2 Nichterfallsfall (Error) .....	26
<b>7 Datenformat für die Bestätigung der Benachrichtigung.....</b>	<b>27</b>
<b>8 APPENDIX .....</b>	<b>28</b>
8.1 WSDL Beschreibung des Interfaces Applikation (APP2ZUSE) .....	28
8.2 WSDL Beschreibung des Interfaces ZUSE Applikation (ZUSE2APP).....	28
8.3 Beispiel: Anlieferung eines PDF Zustellstücks.....	28
8.4 Beispiel: Anlieferung einer verschlüsselten Sendung .....	29
8.5 Beispiel: Zustellnachweis .....	30
8.6 Beispiel: PrintParameter .....	32
8.7 Beispiel: Zustellung Fehlgeschlagen.....	33
8.8 Beispiel: Mailbody .....	34
8.9 AuthBlock auf SAML Basis .....	34
8.10 Beispiel PDF Rückschein (RSa) mit Abholung durch einen Bevollmächtigten.....	44
8.11 Beispiel PDF Rückschein (nonRSa) .....	45
<b>A. Abbildungsverzeichnis .....</b>	<b>46</b>
<b>B. Tabellenverzeichnis .....</b>	<b>47</b>
<b>C. Revision History .....</b>	<b>48</b>
<b>D. Referenzen .....</b>	<b>49</b>

# 1 Funktionaler Überblick

## 1.1 Einführung

Der Zustelldienst übernimmt die Sendung von der aufrufenden Applikation (z.B. MOA-ZS). Das dabei zu verwendende Datenformat ist die ZUSE Message `DeliveryRequest` (siehe Abschnitt 3). Die Übermittlung des zuzustellenden Dokumentes basiert auf dem SOAP with Attachments (SwA/MIME) Message Format [SwA] gemäß Abschnitt 2.

Die Nachricht besteht aus einem MIME oder S/MIME Container der ohne weitere Manipulation an den Empfänger weitergereicht werden kann. Die für die Zustellung nötigen Informationen (Metadaten) werden im XML Format innerhalb des SOAP Message Bodys übermittelt. Die Sendung kann als SOAP Attachment übergeben werden oder vom Zustelldienst als Callback Attachment unter einer in den XML Metadaten angegebenen URL abgeholt werden.

Die Sendung wird vom Zustelldienst im ersten Schritt auf seine formale Richtigkeit und Vollständigkeit geprüft. Verfügt die Sendung über ein Attachment das mittels Callback Funktion zu holen ist, wird nur der bereits vorhandene Teil (Zustellmetainformation) der Sendung geprüft. Ist diese Prüfung nicht erfolgreich erhält die Applikation eine entsprechende Fehlermeldung (`DeliveryRequestStatus/Error`).

War die Prüfung erfolgreich erhält die Applikation eine `DeliveryRequestStatus/Success` oder `DeliveryRequestStatus/PartialSuccess` Nachricht.

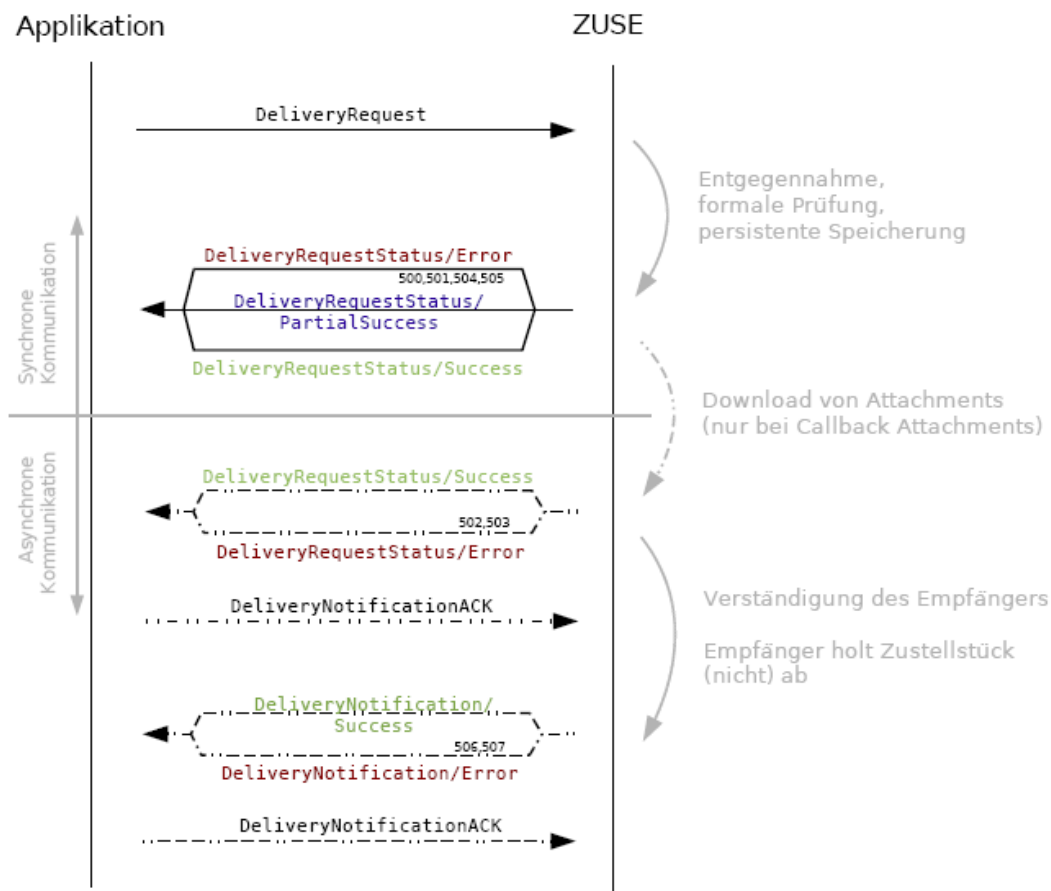
Im zweiten Schritt wird die Sendung persistent gespeichert (Fehlermeldung falls die Speicherung der Sendung nicht möglich ist). Sind die Callback Attachments geladen erhält die Applikation eine Empfangsbestätigung (`DeliveryRequestStatus/Success`). Können die Attachments nicht geladen werden ergeht eine entsprechende Fehlermeldung (`DeliveryRequestStatus/Error`). Erhält die aufrufende Applikation weder eine positive, noch eine negative Antwort ist davon auszugehen, dass die Antwort negativ ist, aber nicht übermittelt werden konnte.

Es ist zu beachten, dass die Kommunikation zwischen Applikation und Zustelldienst aus synchronen und asynchronen Komponenten bestehen kann. Im Falle der asynchronen Kommunikation, die nur beim Einsatz von Callback Attachments auftritt, wechseln SOAP Client und SOAP Server die Rollen: Jetzt ist der Zustelldienst der Client und die Applikation der Server. Welche SOAP-Nachrichten zu welchem Zeitpunkt ausgetauscht werden, ist Abbildung 1 zu entnehmen.

Der Zustelldienst bestätigt die erfolgreiche oder nicht erfolgreiche Übergabe der Sendung an den Empfänger. Es ist zu beachten, dass die Benachrichtigung immer an den Absender gerichtet wird und nicht an die Applikation die die Sendung an den Zustelldienst übergeben hat (z.B. MOA-ZS).

Die Benachrichtigung besteht aus einer `DeliveryNotification` Nachricht (oder alternativ einem PDF Dokument), die an den bei der Zustellung angegebenen Webservice Endpunkt (`NotificationAddress`) oder aber als E-Mail an den Absender übermittelt wird.

Abbildung 1 – Sequenz einer Zustellung



Im Falle der Benachrichtigung via SOAP Webservice bestätigt die Applikation den Erhalt mit einer `DeliveryNotificationACK` Nachricht. Bei der Verständigung via E-Mail entfällt die Bestätigung des Erhalts der Nachricht.

Erhält der Zustelldienst bei der Benachrichtigung via Webservice keine Bestätigung seitens der Applikation, so ist die Applikation gemäß dem Prozess, welcher in [ZUSESPEC] definiert ist, erneut zu verständigen.

Die Authentifizierung und Sicherung des Datenverkehrs zwischen Applikation und Zustelldienst ist nicht Gegenstand dieser Spezifikation. Authentifizierung und Sicherung wird gesondert in [ZUSESPEC] behandelt.

Optional können URLs für die Webservice-Notification oder Email-Notification bei der Entgegennahme der Zustellung auf Plausibilität geprüft werden. Verläuft diese Prüfung negativ, muss die Zustellung abgelehnt werden, da in Folge Probleme mit der Übermittlung der `DeliveryNotification` unvermeidbar wären. In diesem Fall muss eine entsprechende Fehlermeldung zurückgeliefert werden.

## 2 Datenformat der (un-)verschlüsselten Sendung

Hat der Empfänger bei einem Zustelldienst einen Public Key im X.509 DER Format hinterlegt, so MUSS dieser benutzt werden, um die Sendung zu verschlüsseln. Existiert kein Schlüssel, so MUSS die Sendung unverschlüsselt an den Zustelldienst übergeben werden.

### 2.1 Container für Sendungsteile

Bevor die eigentliche Verschlüsselung stattfinden kann, MÜSSEN die Sendungen entsprechend aufbereitet werden. Für unverschlüsselte Sendungen stellt diese Aufbereitung bereits das Endformat dar.

Die Aufbereitung MUSS nach dieser Methode erfolgen:

1. Es MUSS ein multipart-MIME Container erstellt werden (Content-Type multipart/mixed); dieser multipart-MIME Container MUSS einen Mailbody-Part beinhalten. Der Mailbody-Part MUSS den ersten Body-Part darstellen.
2. Zur Einbettung der Sendungsteile selbst (diese sind zum Beispiel XML-Datei inklusive Signatur, weitere XML-Dateien, XSLT-Stylesheet(s), PDF-Dateien, binäre Anhänge etc.) MUSS eine der beiden Varianten a) oder b) gewählt werden:
  - a) Die einzelnen Sendungsteile werden als eigene MIME-Parts im multipart-MIME Container eingebettet. Die MIME-Types der einzelnen Sendungsteile sind entsprechend dem jeweiligen Inhalt anzugeben.
  - b) Die einzelnen Sendungsteile werden im Rahmen eines weiteren, eigenen MIME-Containers gepackt und in den multipart-MIME Container eingebettet. Der zusätzliche MIME-Container der Sendungsteile MUSS als .ZUS-File bezeichnet werden.

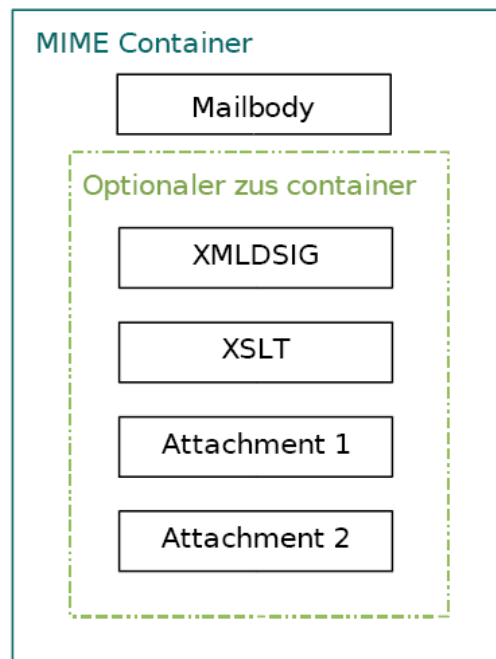
Ein Zustelldienst MUSS beide Varianten der Einbettung von Sendungsteilen verarbeiten können.

Der Mailbody ist ein Text der dem Empfänger im Falle der Übermittlung des Zustellstücks als Mailbody verfügbar gemacht wird und ihn darüber informiert, dass die eigentliche Sendung als E-Mail Attachment vorliegt. Zusätzlich können darin Informationen über den Zustelldienst enthalten sein. Ein Beispiel für einen Mailbody findet sich in Abschnitt 8.8.

Bei XML-basierten Sendungen muss für die Browseransicht ein XSLT Stylesheet mitübergeben werden. (Anmerkung: das Stylesheet für eine gesicherte Ansicht in der Bürgerkartenumgebung ist als Transformation in der eingebetteten XML-DSig Signatur enthalten).

Das .ZUS File MUSS aus einem ZIP-Container [ZIP] mit geänderter Datei Extension (.zus) und eigenem MIME-Type bestehen. Der MIME-Type des .ZUS-Files MUSS auf application/x-zus lauten. Dieser Container dient dazu, ein allfälliges auf dem Endgerät des Empfängers installiertes Tool automatisch ansprechen zu können (via File-Extension .ZUS bzw. über den definierten MIME-Type). Dieser Container wird seinerseits in einem eigenen MIME Body-Part verpackt. Die Zusammenhang zwischen multipart-MIME Container und dem .ZUS-File wird in Abbildung 2 dargestellt.

**Abbildung 2 – Container-Struktur - Zustellteile werden innerhalb eines .ZUS-Files übermittelt**



Das Beispiel 1 zeigt einen Ausschnitt aus einer Nachricht. Es handelt sich hier um einen MIME Container, der als ersten Teil den Mailbody enthält. Der Mailbody ist ein generischer Text, der auf das eigentliche Zustellstück im Anhang hinweist. Nach dem zweiten Boundary-String ist die zu sendende Nachricht zu erkennen (im Beispiel ein PDF-Dokument). Beispiel 2 zeigt wiederum eine Nachricht im MIME-Format. Diesmal jedoch befindet sich die Nachricht bzw. die einzelnen Sendungsteile in einem zusätzlichen .ZUS Container, wie man aus dem Content-Type erkennen kann.

```

MIME-Version: 1.0
Content-Type: multipart/mixed;
  boundary="-----070603060700010608080604"

This is a multi-part message in MIME format.

-----070603060700010608080604
Content-Type: text/plain; charset=us-ascii; format=flowed
Content-Transfer-Encoding: 7bit
Content-Disposition: inline;

Das ist eine elektronische Zustellung!

-----070603060700010608080604
Content-Type: application/pdf; name="test.pdf"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="test.pdf"

PHA+TG9yZW0gaXBzdW0gZG9sb3Igc2l0IGFtZXQgY29uc2VjdGV0dWVyIGNvbnNlcX
....
VhdCBjb21tb2RvIE1vcmlEN1cmFiaXR1ciBxdWlzlLiBSaXN1cyBqdXN0byBpZCB1
dWlzbW9kIGFjIGxpYmVybyB1cm5hIE51bGxhIGFkaXBpc2NpbmcgbmV0dXMgb3JjaS
4gTG9vIHRlbGx1cyB1dCBhdWN0b3Igc2l0IGFtZXQgY29uc2VjdGV0dWVyIGNvbnNlcX
bmMgdXQuIFN1bSBYaG9uY3VzIHV0IHRvcnRvcjB1bnVjZW5hcyBwaXZh
-----070603060700010608080604

```

**Beispiel 1: MIME Header für den Transfer von Sendungsteilen ohne .ZUS-Container**

```

MIME-Version: 1.0
Content-Type: multipart/mixed;
    boundary="-----070603060700010608080604"

This is a multi-part message in MIME format.

-----070603060700010608080604
Content-Type: text/plain; charset=us-ascii; format=flowed
Content-Transfer-Encoding: 7bit
Content-Disposition: inline;

Das ist eine elektronische Zustellung mit .zus Container!

-----070603060700010608080604
Content-Type: application/x-zus; name="test.zus"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="test.zus"

ZWxsZW50ZXNxdWUgb3JjaSBwZWRLIGVyb3Mgbm9uIG51bmMgU3VzcGVuZGlzc2UgY2
.....
9uc2VjdGV0dWVyIHZpdGF1IGNvbnZhbGxpcyBmYXVjaWJ1cy4gTmFzY2V0dXIgY3Vy
c3VzIGxvYm9ydGlzIHNlbSBwaGFyZXRYYSBlZ2V0IGxvcmVtIHRpbmNpZHVudCBtYX
R0aXMgYXQgQWxpcXVhbS4gRGFwaWJ1cyBDdXJhYml0dXIuPC9wPg==
-----070603060700010608080604

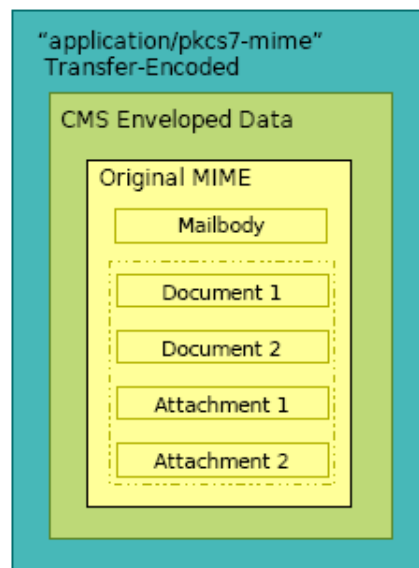
```

**Beispiel 2: MIME Header für den Transfer der Sendungsteile innerhalb eines .ZUS Containers****2.2 Verschlüsselung**

Um die Nachricht zu verschlüsseln, MUSS auf folgende Weise vorgegangen werden:

3. Ausgangspunkt ist der MIME-Container wie in Kapitel 2.1 definiert.
4. Dieser MIME-Container MUSS gemäß RSA [PKCS1] mit dem öffentlichen Schlüssel des Empfängers verschlüsselt werden. Dieser Schlüssel ist aus dem X.509 Zertifikat des Empfängers zu beziehen. Das Resultat dieser Operation ist eine CMS-Datei [CMS].
5. Das Ergebnis der vorigen Operation – die CMS-Datei – MUSS unter Anwendung von S/MIME [SMIME] gekapselt und versendet werden.

Abbildung 3 verdeutlicht den dargestellten Prozess und skizziert das Endresultat.

**Abbildung 3 – Container-Struktur einer verschlüsselten Zustellung**

Beispiel 3 zeigt den verschlüsselten MIME Header. Der Content-Typ ist hier auf `application/x-pkcs7-mime` gesetzt. Folge dessen handelt es sich bei diesem Ausschnitt um einen Teil des CMS-Containers und der darin enthaltenen verschlüsselten Nachricht.

```
MIME-Version: 1.0
Content-Type: application/x-pkcs7-mime; name="smime.p7m"
Content-Transfer-Encoding: base64

Y29uc2VxdWF0IGFjIFZlc3RpYnVsdW0uIFZlc3RpYnVsdW0gdHVycGlzIGR1aSBxdWFTIGdyYXZpZG
EgdmVsIHN1bmVjdHVzIEluIGlkIHV0IGFkaXBpc2NpbmcuIFRpbmNpZHVudCBzZWQgdmVsIGludGVy
ZHVtIGZhY2lsaXNpcyB0dWxsYSBwcmV0aXVtIHRlbXB1cyBmZXJtZW50dW0gbGFjdXMGY29uZ3VlLi
BMiYWNpbm1hIERvbmVjIGNvbmd1ZSBuaWJoIHZ1bCBhYyBzYXBpZW4gZW5pbSBWZXN0aWJlbHVtIHBy
ZXRpdW0gbW9yYmkuIDwvD4NCjxwP1BlbGxlbmRlc3F1ZSBldCBtYWxlc3VhZGEgZHVpIGV0IGNvb
RpbWVudHVtIG51YyBzZWQgdm1iaCBqdXN0byBpbmR1cmRlbS4gU2VtIHV0IE1vcmlpIHZvbHV0cGF0
IG5pYm9gdXJuYSB0dWxsYSB1bm1tIEN1cmFiaXR1ciBlbSBDbXJhYm10dXIuIENvbnNlcXVhdCB0dW
xsYSBjb25kaW11bnR1bSBtb250ZXMGbG9yZW0gc2VtcGVyIGVgUGhhc2VsbHVzIG5pYm9gdXJuZGlt
ZW50dW0gdmVsLiBQZWxsZW50ZXNxdWUgb3JjaSBwZWRLIGVyb3MgYm9uIG51bmMgU3VzcGVuZGlzc2
UgY29uc2VjdGV0dWVzIHZpdGF1IGNvbzZhbGxpcyBmYXVjaWJ1cy4gTmFzY2V0dXIgY3Vyc3VzIGxv
Ym9ydG1zIHN1bSBwaGFyZXRYYSB1Z2V0IGxvcmlpIHRpbmNpZHVudCBtYXR0aXMGYXQgQWxpXVhbS
...
4gRGFwaWJ1cyBDdXJhYm10dXIuPC9wPg==
```

### Beispiel 3: MIME Header für den Transfer von CMS Containern



### 3 Datenformat für die Dokumentenanlieferung (DeliveryRequest)

In den kommenden Kapiteln befinden sich eine Vielzahl von unterschiedlichen Namespaces. Anbei eine kurze Aufstellung:

**Tabelle 1 – Namespace-Deklarationen**

Präfix	Namespace	Erläuterung
dsig	http://www.w3.org/2000/09/xmldsig#	XMLDsig [XMLDSIG]
soap	http://www.w3.org/2001/12/soap	SOAP [SOAP]
saml	urn:oasis:names:tc:SAML:1.0:assertion	SAML Assertion [SAML]
p	http://reference.e-government.gv.at/namespace/persondata/20020228#	PersonData 2.0 [PERSDATA]

Für den Datenaustausch MUSS das Transportprotokoll SOAP 1.1 [SOAP] verwendet werden. Die Zustellmetainformationen werden im SOAP Body Part transportiert. Die eigentliche Sendung bildet ein einziges Attachement oder wird via Callback-Attachment-URL referenziert.

Die Sendung ist gemäß den Vorgaben in Abschnitt 2 definiert entweder eine verschlüsselte Sendung im S/MIME Format oder eine unverschlüsselte Sendung im MIME Format und wird als Attachment gemäß [SwA] transportiert. Das Attachment der Sendung ist so beschaffen, dass es ohne weitere Änderung dem Empfänger, bspw. per E-Mail, übermittelt werden kann.

Die Metainformationen werden gemäß der `DeliveryRequest` Nachricht übermittelt (siehe Abbildung 4). Darin MUSS die sendende Applikation (der Sender) die folgenden Informationen an den Zustelldienst übermitteln:

1. Zustell-Token: dieses Token wird vom Zustellkopf [ZUSEKOPF] generiert und enthält den Identifikationsbegriff des Empfängers. Bei natürlichen Personen ist dies das Zustell-bPK, bei juristischen Personen die Stammzahl. Das Zustell-Token kann in einem der folgenden Element übergeben werden:
  - a) `DeliveryRequest/ZbPK`: dieses Element dient der Abwärtskompatibilität zu bestehenden Versenderapplikation. Neue Applikationen sollten folgendes Adressierungschema verwenden:
  - b) `DeliveryRequest/Identification/Value`: als Typ für das Zustell-Token ist der Wert „urn:publicid:gv:at:cemtoken“ zu verwenden. Dieses Schema erlaubt die zukünftige Unterstützung von weiteren Schemen. Ist der Typ dem Zustelldienst unbekannt, ist die Zustellung mit dem Fehlercode 511 (siehe Kapitel 4) abzulehnen.
2. Benachrichtigungsadresse (`NotificationAddress`) des Senders, an die allfällige Benachrichtigungen (`DeliveryNotification`), d.h. z.B. Zustellnachweise, durch einen Zustelldienst geschickt werden können. Diese MUSS eine der folgenden sein:

- a) Internetadresse eines Webservices (`WebServiceURL`)
- b) E-Mailadresse (`Email`)

Desweiteren kann optional über das Attribut (`Type`) der gewünschte Benachrichtigungstyp angegeben werden. Folgende Werte sind zulässig:

- c) `xml`
- d) `pdf`
- e) `xml+pdf`

wobei `xml` der Standardwert ist, sofern dieses Attribut nicht verwendet wird. „`xml+pdf`“ signalisiert, dass sowohl ein XML als auch ein PDF Zustellnachweis retourniert werden müssen. Der Benachrichtigungstyp „`pdf`“ kann ausschließlich für eine E-mail Benachrichtigungsadresse angegeben werden. Handelt es sich bei der Benachrichtigungsadresse um ein Webservice, so können ausschließlich die Typen „`xml`“ bzw. „`xml+pdf`“ angegeben werden. Andernfalls ist ein Fehler mit dem Code 512 zu retournieren. Beispiele für XML- und PDF-Rückscheine sind im Anhang zu finden.

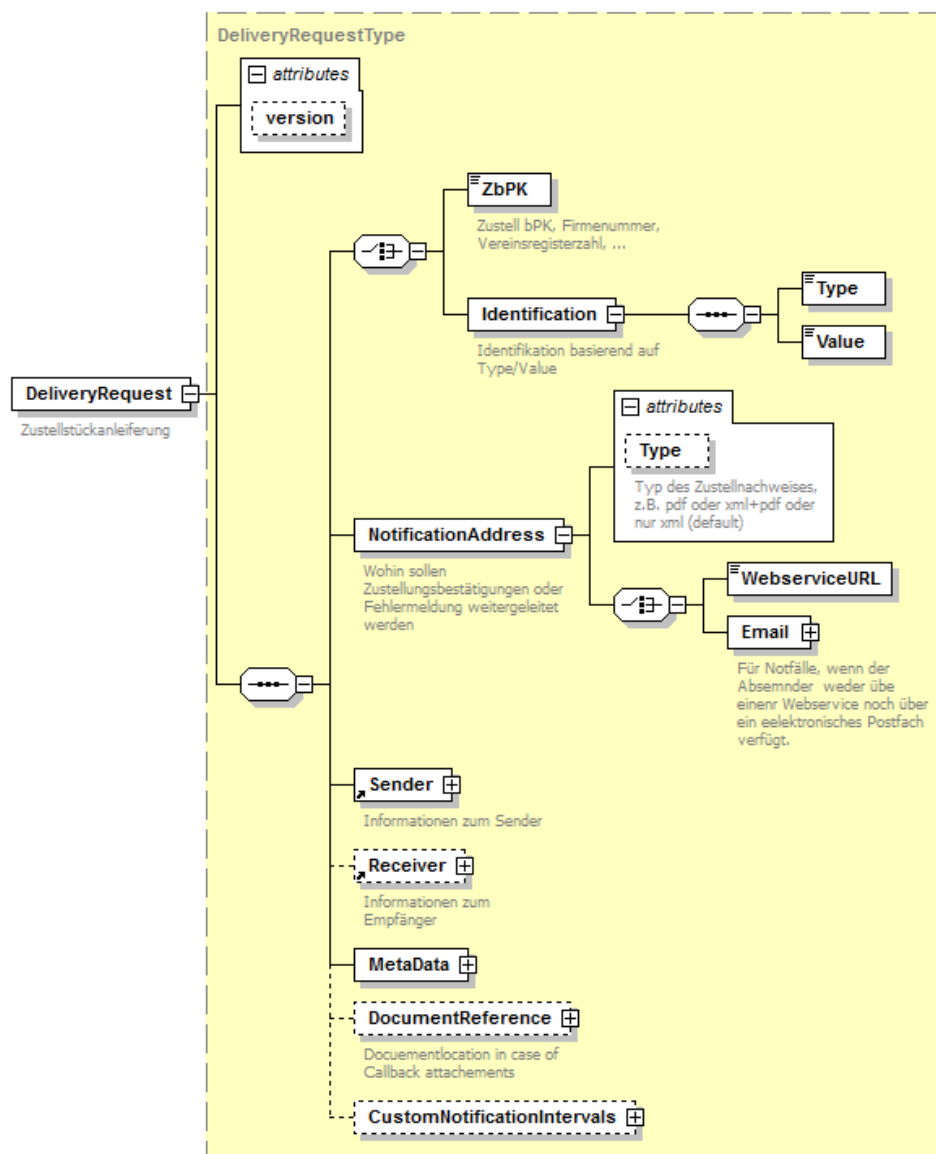
3. Identifikationsdaten des Senders (der Behörde; Anmerkung: dies ist nicht ein allfälliger Dienstleister bzw. Betreiber der zustellenden Applikation). Der Sender – eine natürliche oder juristische Person – wird durch entsprechende Elemente des Personendaten-Schemas [`PERSDATA`] beschrieben. Abschnitt 3.1 beschreibt dieses Element im Detail.
4. Zustell-Metainformation (`MetaData`). Diese MÜSSEN folgende Informationen beinhalten (Abschnitt 3.3 beschreibt dieses Element im Detail):
  - a) ID des Zustellstücks beim Sender (`AppDeliveryID`)
  - b) Zustellqualität (`DeliveryQuality`). Als Zustellqualität MUSS eine der in Abschnitt 3.3 definierten Qualitäten gewählt werden.
  - c) Optional KÖNNEN folgende Zustell-Metainformation angegeben werden:
    - i) Geschäftszahl (`GZ`)
    - ii) ID des Zustellstücks beim Mittler, zum Beispiel bei der zustellenden Applikation MOA-ZS, (`MZSDeliveryID`)
    - iii) Druckparameter für die duale Zustellung (`PrintParameter`)

Zusätzlich KANN die sendende Applikation (der Sender) die folgenden Informationen in einer `DeliveryRequest` Nachricht übermitteln:

5. Zusätzliche Identifikationsdaten des Empfängers. Der Empfänger – ein natürliche oder juristische Person – wird durch entsprechende Elemente des Personendaten-Schemas [`PERSDATA`] beschrieben. Abschnitt 3.2 beschreibt dieses Element im Detail.
6. Information über das Attachment der Sendung im Falle einer Callback-Zustellung (`DocumentReference`). Abschnitt 3.4 beschreibt dieses Element im Detail.

Abbildung 4 zeigt die Zustellanfrage (`DeliveryRequest`) gesamt; die nachfolgenden Abschnitte beschreiben die einzelnen Elemente im Detail.

Abbildung 4 – Gesamtstruktur der DeliveryRequest-Nachricht



### 3.1 Sender

Als Sender können sowohl juristische als auch natürliche Personen fungieren. Folgende Daten der absendenden Behörde MÜSSEN angegeben werden:

7. Name der Person (unter Anwendung entsprechender Elemente des Personendaten-Schemas [PERSDATA])
  - a) Im Falle einer juristischen Person (zum Beispiel Behörde, Firma, Verein, etc.) MUSS der Name der juristischen Person (Name, Untereinheit, OID) angeführt werden. Zusätzlich KANN auch die Stammzahl (Ordnungsnummer aus dem jeweiligen Register, d.h. Firmenbuchnummer, Ergänzungsregister für sonstige Betroffene, Vereinsregister) der Person angegeben werden, um Rücksendungen zu erleichtern.
  - b) Im Falle einer natürlichen Person MUSS der Name der Person (Vor- und Familienname) angegeben werden.

8. Es MUSS zumindest eine Adresse des Senders angegeben werden. Als Adressen KÖNNEN die folgenden Adressarten herangezogen werden (unter Anwendung entsprechender Elemente des Personendaten-Schemas [PERSDATA]):

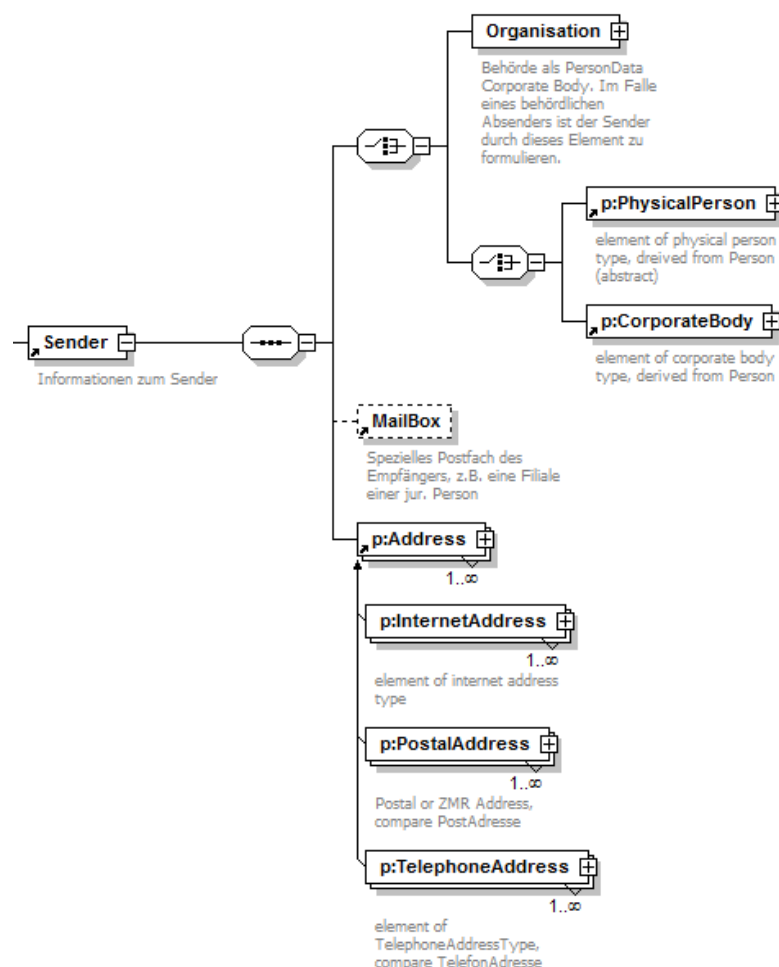
- a) Internetadresse (InternetAddress)
- b) Postanschrift (PostalAddress)
- c) Telefonnummer (TelephoneAddress)

Optional kann auch eine spezielle Mailbox (MailBox) des Senders - sofern vom Zustelldienstbetreiber unterstützt – angegeben werden. Dieses Element ist für Rückantworten sinnvoll, sodass Empfänger eine Antwort an eine spezielle Mailbox des Versenders adressieren können (Details dazu siehe auch MailBox Element des Empfängers).

Abbildung 5 stellt die Angaben zum Sender im Überblick dar.

Im Falle der Papierzustellung (Duale Zustellung) ist zwischen dem Betreiber der Fachapplikation und dem Zustelldienst zu vereinbaren, ob die Adresse des Zustelldienstes oder die Absenderadresse für eine evt. Retourenbearbeitung durch den Zustelldienst auf dem Zustellstück aufscheinen soll. Wurde die Angabe einer Absenderadresse vereinbart und ist diese im Request nicht vorhanden, so ist die Zustellstückannahme mit dem Fehlercode 510 abzulehnen.

**Abbildung 5 – DeliveryRequest / Sender: Sender Information**



## 3.2 Empfänger

Der Empfänger MUSS entweder eine natürliche oder eine juristische Person sein.

Der Empfänger einer Sendung MUSS über ein Zustell-Token adressiert werden, das als erstes Kindelement der `DeliveryRequest`-Nachricht übermittelt werden MUSS. Das Token wird gemäß [ZUSEKOPF] gebildet. Aus diesem Token kann der Zustelldienst den Identifikationsbegriff des Empfängers extrahieren.

Wenn erforderlich, KÖNNEN weitere, diskrete Empfängerinformationen, wie Vorname, Familienname, Geburtsdatum und Adresse, angegeben werden, um eine `DeliveryRequest`-Nachricht am Eingang einer Zustellung menschenlesbar vorliegend zu haben.

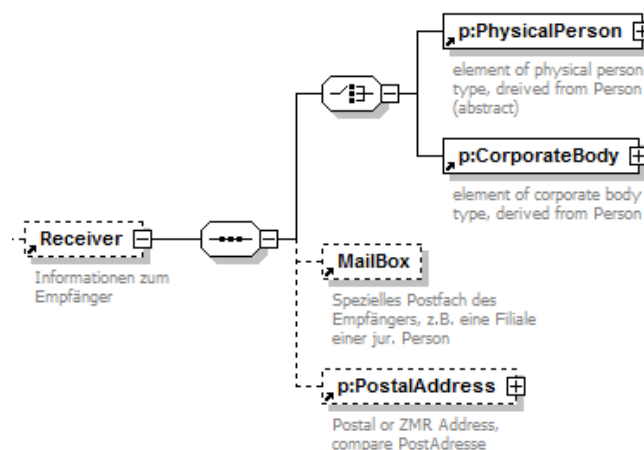
Die Inhalte des den Empfänger beschreibenden Elements folgen der Personendatenstruktur [PERSDATA] und sehen neben der Angabe des Namens auch die optionale Angabe der Anschrift vor.

Zwar kann die Zustelladresse auch im Zustellstück selbst enthalten sein, zum Beispiel bei Zustellung mittels Fensterkuvert, als Postkarte oder bei Zustellung eines elektronischen Bescheides wo die Anschrift bereits am Schriftstück zu finden ist, jedoch hat auf diese Zustellstück-internen Informationen der Zustelldienst keinen Zugriff. Daher KANN die Zustelladresse (`PostalAddress`) auch im Element `Receiver` angegeben werden.

Optional kann auch eine spezielle Mailbox (`MailBox`) des Empfängers - sofern vom Zustelldienstbetreiber unterstützt – angegeben werden. Bspw. kann damit eine spezielle Filiale einer großen Firma oder eine Organisationseinheit adressiert werden. Es obliegt Zustelldienstbetreibern dieses Feature zu unterstützen (Details dazu siehe [ZUSESPEC]).

Abbildung 6 stellt die Angaben zum Empfänger im Überblick dar.

**Abbildung 6 – DeliveryRequest/Receiver: Empfänger Information**



## 3.3 Zustell-Metainformation

Folgende Metainformationen MÜSSEN in einer `DeliveryRequest`-Nachricht enthalten sein:

1. Applikations Zustell ID (`AppDeliveryID`)

Von der absendenden Applikation frei belegbare Identifikationsnummer.

## 2. Zustellungsqualität (`DeliveryQuality`)

Als Zustellqualität MUSS eine der folgenden gewählt werden:

`nonRSa` einfache Zustellung ohne Zustellnachweis

`nonRSa+` einfache Zustellung ohne Zustellnachweis – nicht an Postbevollmächtigte (Abholung durch Dritte unter Verwendung von Vollmachten ist nicht möglich)

`RSb` RSb-Zustellung mit Zustellnachweis

`RSb+` RSb-Zustellung mit Zustellnachweis – nicht an Postbevollmächtigte (Abholung durch Dritte unter Verwendung von Vollmachten ist nicht möglich)

`RSa` Rsa-Zustellung zu eigenen Händen mit Zustellnachweis

`RSa+` Rsa-Zustellung zu eigenen Händen mit Zustellnachweis (Abholung durch Dritte unter Verwendung von Vollmachten ist nicht möglich)

`RSb/RSb+` ist nur für die Papierzustellung relevant. Die Qualität `RSb/RSb+` MUSS in Bezug auf die elektronische Zustellung gleich behandelt werden wie `RSa/RSa+`. In Bezug auf die duale Zustellung ist jedoch die Unterscheidung nach wie vor notwendig.

Zusätzlich KÖNNEN die folgenden, optionalen Metainformationen in einer `DeliveryRequest`-Nachricht enthalten sein:

1. Subject zur Anzeige im Webinterface des Zustelldienstbetreibers bzw. als Message Subject bei Weiterleitung der (verschlüsselten) Zustellung an eine Verständigungsadresse des Empfängers bzw. bei Mailabholung gemäß § 35 (3) (siehe [ZUSEMAIL]).

2. Geschäftszahl (`GZ`) des Zustellstücks

3. Mittler Zustell ID (`MZSDeliveryID`)

Falls die Sendung von einem Mittler (z.B.: MOA-ZS Betreiber) übergeben wird, kann eine vom Mittler frei belegbare Identifikationsnummer angegeben werden.

4. Dokumentenklasse (`DocumentClass`). Siehe Abschnitt 3.4

5. Referenzen (`Relations`) zu anderen Zustellstücken. Absender haben die Möglichkeit, eine Verbindung mit anderen Zustellstücken herzustellen. Dazu kann die eindeutige `AppDeliveryID`, die Geschäftszahl (`GZ`), die Mittler-ID (`MZSDeliveryID`) oder die interne Zustellserver-ID (`ZSDeliveryID`) angegeben werden. Zustelldienstbetreiber können somit z.B. zusätzlich die Möglichkeit einer Gruppierung von Zustellstücken nach diesen Kriterien vornehmen. Bekannt ist diese Gruppierung z.B. von Mail-clients, die Nachrichten nach Kommunikationsthread gruppieren.

6. Letztmöglicher Zeitpunkt der ersten Empfängerverständigung (`StartNotificationBefore`)

Damit ist es beispielsweise möglich, mehrere Zustellungen für einen Empfänger innerhalb eines bestimmten Zeitraums (z.B. 24h) zu einer Sendung zusammenzufassen.

Das Element zur Vorgabe des Zeitpunkts für die erste Empfängerverständigung ist vor dem Hintergrund des aktuellen Zustellgesetzes für behördliche Zustellung obsolet, da dieser Aspekt im Gesetz detailliert geregelt ist. Bei behördlichen Zustellungen DARF dieses Element NICHT verwendet werden. Die Spezifikation behält die Möglichkeit der Vorgabe des Zeitpunkts für die erste Empfängerverständigung aber für künftige Formen der Zustellung bei (zum Beispiel für allf. Formen der Privatzusendung).

#### 7. Letztmöglicher Zeitpunkt der Abholung der Sendung durch den Empfänger (*DeliverBefore*)

Das Element zur Vorgabe von Abholfristen ist vor dem Hintergrund des aktuellen Zustellgesetzes für behördliche Zustellung obsolet, da die Abholfristen im Gesetz detailliert geregelt ist. Bei behördlichen Zustellungen DARF dieses Element NICHT verwendet werden. Die Spezifikation behält diese Möglichkeit der Vorgabe von Abholfristen aber für künftige Formen der Zustellung bei (zum Beispiel für allf. Formen der Privatzusendung).

#### 8. Druckparameter für die Duale Zustellung (*PrintParameter*)

Dieses zusätzliche optionale Element ermöglicht die Angabe von beliebigen Parametern für den Druck. Über das Attribut `profile` vom Typ `xsd:anyURI` wird ein Profil referenziert, das die Verarbeitung des Zustellstücks durch den Zustelldienst und das Format und die Bedeutung zusätzlicher übergebener Parameter regelt. Die im Rahmen des Profils spezifizierten Parameter werden als Kindknoten des Elements *PrintParameter* übergeben.

### 3.4 Dokumentenklasse

Optional kann eine sogenannte Dokumentenklasse für Schriftstücke angegeben werden. Zweck dieser Dokumentenklasse ist die einfachere Verwaltung von empfangenen privaten Schriftstücken am Zustellserver. Schriftstücke können so bspw. visuell nach Dokumentenklassen (auch prioritär) geordnet werden.

Dokumentenklassen müssen gemäß den Vorgaben für MIME-Types (siehe [RFC2048]) angegeben werden. Als Präfix wird „*x-edclass*“ verwendet. Zulässige Dokumentenklassen werden in regelmässigen Abständen aktualisiert auf den Informationsseiten des behördlichen Zustellkopfs publiziert.

Die Dokumentenklasse gilt für die gesamte Zustellung (*DeliveryRequest*) und kann nicht für einzelne in der Zustellung enthaltene Dokumente differenziert werden. Sie kann daher optional in den Metainformationen im *DocumentClass* Element übergeben werden.

Besteht der Anhang des Zustellstücks aus einem XML Dokument kann anstelle bzw. zusätzlich zur Dokumentenklasse das Attribut „*Namespace*“ verwendet werden um den XML Namespace des Zustellstücks anzugeben. Damit können Workflowsysteme das Dokument automatisch klassifizieren ohne es vorher parsen zu müssen.

### 3.5 Sendungs-Referenz

Die Sendung (Zustell-Request) MUSS entweder ein SOAP-Attachment oder ein sogenanntes Callback-Attachment enthalten. Eine Zustellung ohne jeglichem Attachment ist nicht gültig und MUSS vom Zustelldienst zurückgewiesen werden.

Im Falle eines SOAP-Attachments MUSS der Dateiname des Attachments im SwA-MIME Header [SwA] angegeben werden. Optional KANN zusätzlich der MIME-Type angegeben werden.

Im Falle eines Callback-Attachments MUSS vom Zustelldienst dieses unter dem in der `DeliveryRequest`-Nachricht genannten URL, per HTTP(S)-GET, selbst abgeholt werden. Diese URL KANN im Rahmen der `DeliveryRequest`-Nachricht im `DocumentReference`-Element angegeben werden.

Wird dadurch ein Callback eingesetzt, so MUSS der dazu vom Zustelldienst eingesetzte http-Client sowohl HTTPS- als auch HTTP 1.1-Range Header Fields [HTTP11] unterstützen. Für Callback-Attachments KANN neben der URL in der `DeliveryRequest`-Nachricht zusätzlich auch eine MD5 Prüfsumme [MD5] angegeben werden. Ist diese MD5-Prüfsumme angegeben, so MUSS der Zustelldienst, nach Bezug des Callback-Attachments, die vollständige bzw. korrekte Übertragung unter Zuhilfenahme der MD5-Prüfsumme verifizieren. Bei negativem Prüfergebnis MUSS der Zustelldienst mit einem Fehler antworten.

### 3.6 Empfängerverständigungsintervalle

Das Element zur Vorgabe von Empfängerverständigungsintervallen ist vor dem Hintergrund des aktuellen Zustellgesetzes für behördliche Zustellung obsolet, da die Verständigungsintervalle und -arten im Gesetz detailliert geregelt sind. Bei behördlichen Zustellungen DARF dieses Element NICHT verwendet werden. Die Spezifikation behält diese Möglichkeit der Vorgabe von Empfängerverständigungsintervallen aber für künftige Formen der Zustellung bei (zum Beispiel für allf. Formen der Privatzusendung).

Die Vorgabe von Empfängerverständigungsintervallen geschieht über `RecipientNotification`-Elemente, die eine Methode (`NotificationMethode`: hier ist die Auswahl zwischen elektronischer Verständigung - `electronic` - und papierbasierter Verständigung - `paperbased` - möglich) und einen Zeitabstand – relativ zum Zeitpunkt der Übergabe an den Zustelldienst (`TimeSinceDisposal`) im `xs:duration` Format – enthalten KÖNNEN.

Die Festlegung einer Verständigungsmethode (`NotificationMethode`) ist nur ein Hinweis für den Zustelldienst und entbindet diesen nicht von seinen Verständigungspflichten gemäß Zustellgesetz [ZUSTG]. Der Zustelldienst KANN die Vorgabe einer Verständigungsmethode im Rahmen seiner Zusatzleistungen unterstützen.

Das `TimeSinceDisposal`-Element (vom Typ `xs:duration`) enthält Zeitintervalle als Zeichenkette (String). Zum Beispiel steht `P3D` für drei Tage, `PT12H` für 12 Stunden und `P2DT12H` für 2 Tage und 12 Stunden.

Es ist nicht sinnvoll, Benachrichtigungsintervalle in etwas anderem als ganze Tage oder Stunden zu definieren. Werden keine Intervalle angegeben, so gelten die vom Zustell-Gesetz [ZUSTG] vorgegebenen Standardintervalle für Zustellungen.



## 4 Datenformat für Erfolgs- und Fehlermeldungen in der Dokumentenanolieferung

Es gibt drei Arten von Erfolgs- bzw. Fehlermeldungen:

1. Erfolgsmeldungen (`DeliveryResponse/Success`)
2. Partielle Übernahmebestätigungen (`DeliveryResponse/PartialSuccess`)

Diese finden nur bei Callback-Attachments ihre Anwendung und werden übermittelt bevor die Sendung abgeholt wird.

3. Fehlermeldungen (`DeliveryResponse/Error`)

Diese Fehlermeldungen werden durch Errorcodes und Errormessages näher spezifiziert. Details dazu sind der Abbildung 7, in Beispiel 4, Beispiel 5 und Beispiel 6, sowie der Tabelle 2 zu entnehmen. Außer den in diesem Dokument spezifizierten Fehlermeldungen gibt es noch automatisch generierte Fehlermeldungen des HTTP Servers (z.B. 500 – internal Server Error) und des SOAP-Servers (SOAP-Faults über falsches Datenformat etc.).

Tabelle 2 zeigt die definierten Fehlercodes im Zuge der Annahme von Zustellstücken. Beispiel 4 zeigt eine exemplarische Erfolgsmeldung.

**Tabelle 2 – Fehlercodes im Zuge der Sendungsannahme**

<i>Server und formale Fehler bei Anlieferung von Zustellstücken</i>		
<b>Code</b>	<b>Error-Message</b>	<b>File</b>
500	Queue congestion	–
501	Missing metainformation	–
502	Attachment could not be loaded	attachment
503	Attachment MD5 verification failed	attachment
504	Unresolvable ZbPK	–
505	Delivery arrived too late to send first notification	–
506	Electronic notification of the recipient failed	–
507	Receiver did not pick up delivery	–
508	Illegal DeliveryRequest	–
509	Internal Error	–
510	Missing sender data	–

511	Unknown ID type	
512	Illegal notification type	

Die Handhabung des Fehlers 500 (Queue congestion<sup>1</sup>) liegt im Ermessen der absendenden Applikation.

Die Fehler 501 (Missing Meta Information) und 504 (Unresolvable ZbPK) erfordern manuelles Zutun durch den Anwender auf der Senderseite und führen in jedem Fall zu einem Abbruch der Zustellung.

Bei Fehlern im der Handhabung von Callback-Attachments obliegt es dem Zustelldienst wiederholte Anläufe zu unternehmen, um das Attachment vollständig zu laden. Es MÜSSEN jedoch zumindest 3 Versuche binnen 3 Stunden unternommen werden. Die Fehler 502 und 503 sind in Verbindung mit Fehlern im Zuge des Attachment-Handlings anzuwenden.

Der Fehler 505 (Delivery arrived too late to send first notification) MUSS in Verbindung mit der Angabe einer StartNotificationBefore-Angabe im Rahmen der DeliveryRequest-Nachricht verwendet werden, nämlich dann, wenn diese Zeitangabe zu knapp gewählt wurde und/oder der Zustelldienst die Zustellanfrage (oder auch das Attachment im Falle eines Callback-Attachments) zu spät erhalten hat.

Der Fehler 506 und 507 sind dann zu retournieren, wenn der Empfänger gar nicht benachrichtigt werden konnte (506) oder dieser das Zustellstück nie abgeholt hat (507). Der Zustelldienst MUSS den Fehler 507 nach Ablauf einer optional in der Zustellanfrage mit angegeben oder der gesetzmäßig anzunehmenden Abholfrist retournieren.

Der Fehler Illegal DeliveryRequest (Code 508) MUSS vom Zustellserver dann retourniert werden, falls eine DeliveryRequest-Nachricht technisch oder fachlich ungültig ist, und keine der in Tabelle 2 spezifizierten Fehlermeldungen angewendet werden kann. Der Fehler Internal Error (Code 509) beschreibt interne Fehler des Zustellservers im Zuge der Zustellstückannahme. Die Fehler 508 sowie 509 bedürfen daher der manuellen Kontrolle/Nachbearbeitung. Die hierzu notwendigen Vorgangsweisen MÜSSEN im Betriebshandbuch des jeweiligen Zustelldienstes beschrieben werden.

Der Fehler 510 MUSS zur Signalisierung von fehlenden Angaben zum Sender (in der DeliveryRequest-Nachricht) verwendet werden.

---

<sup>1</sup> Dieser Fehlercode wird geliefert, wenn eine Warteschlange weit über das Normalmaß ausgelastet bzw. überfüllt ist. Eine weitere Abarbeitung der anliegenden Operationen kann nicht in ausreichendem Maße gesichert werden.

```

<SOAP-ENV:Envelope
xmlns:SOAP-ENV=http://schemas.xmlsoap.org/soap/envelope
xmlns:zs="http://reference.e-
government.gv.at/namespace/zuse/20040301/">
  <SOAP-ENV:Body>
    <zs:DeliveryResponse>
      <zs:Success>
        <zs:DeliveryService>
          http://www.zustellung.gv.at
        </zs:DeliveryService>
        <zs:AppDeliveryID>abc1234x</zs:AppDeliveryID>
        <zs:MZSDeliveryID>0503214567</zs:MZSDeliveryID>
        <zs:ZSDeliveryID>a753456789</zs:ZSDeliveryID>
      </zs:Success>
    </zs:DeliveryResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

#### Beispiel 4: Nachrichtenformat für Erfolgsmeldungen - Übernahmebestätigung

```

<ENV:Envelope
xmlns:SOAP-ENV=http://schemas.xmlsoap.org/soap/envelope
xmlns:zs="http://reference.e-
government.gv.at/namespace/zuse/20040301/">
  <SOAP-ENV:Body>
    <zs:DeliveryResponse>
      <zs:PartialSuccess>
        <zs:DeliveryService>
          http://www.zustellung.gv.at
        </zs:DeliveryService>
        <zs:AppDeliveryID>abc1234x</zs:AppDeliveryID>
        <zs:MZSDeliveryID>0503214567</zs:MZSDeliveryID>
        <zs:ZSDeliveryID>a753456789</zs:ZSDeliveryID>
      </zs:PartialSuccess>
    </zs:DeliveryResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

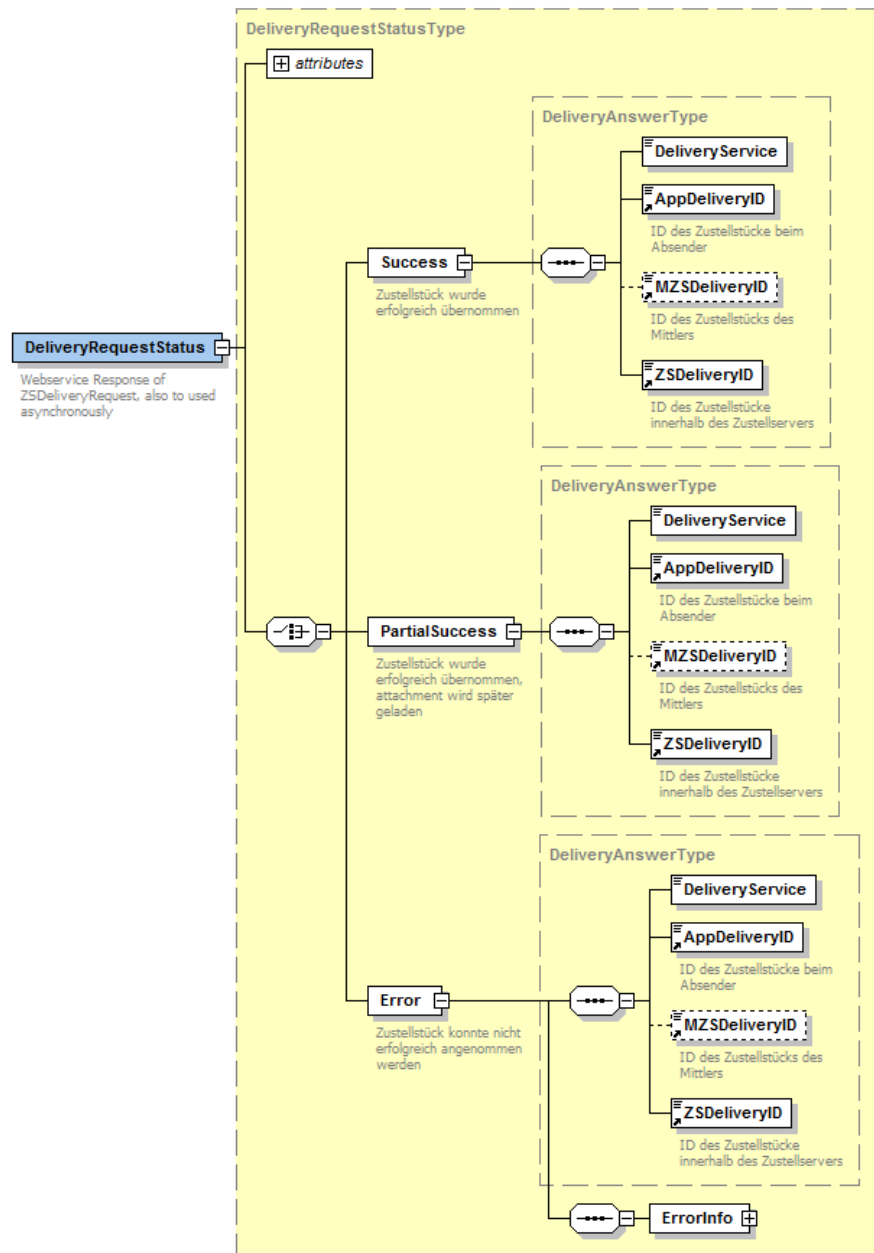
```

**Beispiel 5: Nachrichtenformat für Erfolgsmeldungen**  
**partielle Übernahmenbestätigung im Callback Fall**

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope"
xmlns:zs="http://reference.e-
government.gv.at/namespace/zuse/20040301/">
  <SOAP-ENV:Body>
    <zs:DeliveryResponse>
      <zs:Error>
        <zs:DeliveryService>
          http://www.zustellung.gv.at
        </zs:DeliveryService>
        <zs:AppDeliveryID>abc1234x</zs:AppDeliveryID>
        <zs:MZSDeliveryID>0503214567</zs:MZSDeliveryID>
        <zs:ZSDeliveryID>753456789</zs:ZSDeliveryID>
        <zs:ErrorInfo>
          <zs:Code>errorcode</zs:Code>
          <zs:Text>errormessage</zs:Text>
        </zs:ErrorInfo>
      </zs:Error>
    </zs:DeliveryResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**Beispiel 6: Nachrichtenformat für Fehlermeldungen**

**Abbildung 7 – XML-Datenformat für die Antwort auf Datenanlieferung (synchrone Kommunikation): Success, PartialSuccess und Error**



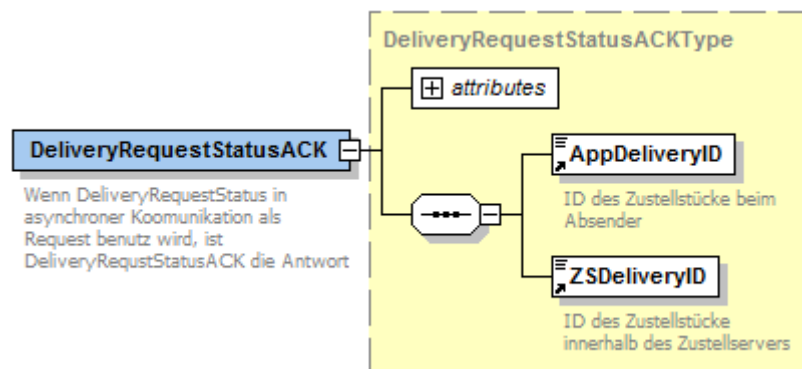
## 5 Datenformat für Nachrichten über Callback-Attachments (asynchrone Kommunikation)

In der asynchronen Rückwärts-Kommunikation im Falle von Callback-Attachments (zwischen einem Zustelldienst und einer zustellenden Applikation; hier fragt der Zustelldienst bei der zustellenden Applikation um das Callback-Attachment an) MÜSSEN die Nachrichten `DeliveryRequestStatus` (siehe Abbildung 7) als Anfrage und `DeliveryRequestStatusACK` als Antwort verwendet werden (siehe Abbildung 8).

`DeliveryRequestStatus` entspricht genau der Nachricht, die in der synchronen Kommunikation als Antwort auf eine Anfrage geschickt wird. Es dürfen aber nur die Unterelemente `Success` und `Error` benutzt werden. `PartialSuccess` macht bei der asynchronen Kommunikation keinen Sinn und DARF NICHT benutzt werden. Die Fehlermeldungen decken sich ebenfalls mit denen aus der synchronen Kommunikation und sind aus Tabelle 2 ersichtlich.

Die Nachricht `DeliveryRequestStatusACK` dient nur dazu, den Erhalt der `DeliveryRequestStatus` Nachrichten zu bestätigen und enthält keine neue Information.

**Abbildung 8 – DeliveryRequestStatusACK-Element**



## 6 Datenformat für den Zustellnachweis

Der Zustellnachweis (`DeliveryNotification`) beinhaltet entweder ein `Success` oder ein `Error` Element (Abbildung 10 und Abbildung 11).

In jedem Fall MÜSSEN im Element `DeliveryNotification` folgende Informationen übermittelt werden:

1. Zustelldienst (`DeliveryService`)
2. IDs der Zustellung auf allen beteiligten Systemen:
  - a) `AppDeliveryID`
  - b) `ZSDeliveryID`<sup>2</sup>
3. Informationen über erfolgte Verständigungen (`NotificationsPerformed`)
4. Angaben zum Sender (in diesem Falle immer der Sender des Zustellstücks, zum Beispiel die Behörde – nicht jedoch ein Dienstleister, wie zum Beispiel der MOA-ZS-Betreiber) (`Sender`)
5. Angaben zum Empfänger (`Receiver`)
6. Die `DeliveryNotification`-Nachricht MUSS mit der elektronischen Signatur des Empfängers versehen werden. Das XML-Document ist enveloped nach XMLDSIG [XMLDSIG] zu signieren.

Sender- und Empfängerinformationen werden direkt aus den korrespondierenden Elementen der `DeliveryRequest`-Nachricht (der Schnittstelle zwischen zustellender Applikation und Zustelldienst übernommen). Es MÜSSEN dieselben Definitionen beachtet werden (siehe auch Abschnitt 3). Falls die Empfängerinformationen nicht in der `DeliveryRequest` Nachricht vorhanden waren, können diese auch aus den internen Stammdaten des Zustelldienstbetreibers übernommen werden.

Wurde im `DeliveryRequest` der Benachrichtigungstyp „xml+pdf“ angegeben, so muss der Zustelldienst einen PDF Zustellnachweis gemäß der Vorlage in Anhang 8.10 bzw. 8.11 zu erstellen und diesen zusätzlich im Element `AdditionalFormat` Base64-kodiert einzubetten. Als Type ist „application/pdf“ zu verwenden.

---

<sup>2</sup> Vom Zustelldienst frei wahlbare Identifikationsnummer der einzelnen Sendung.

Abbildung 9 – XML Schema für weitere Typen von Zustellnachweisen

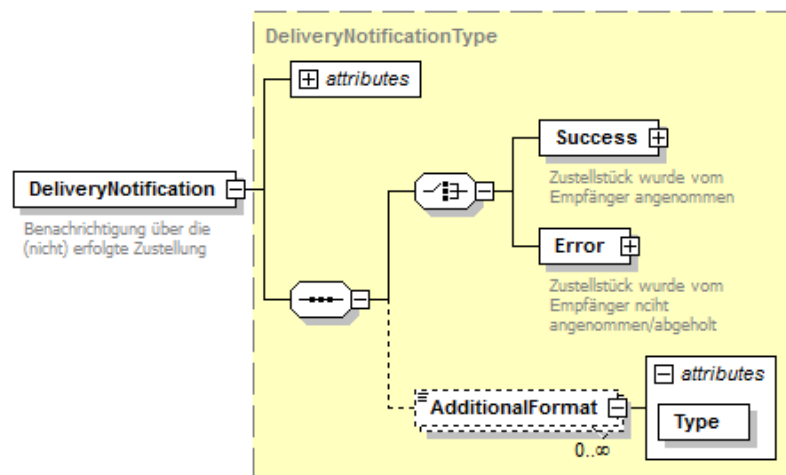


Abbildung 10 – Benachrichtigung über eine erfolgreiche Zustellung

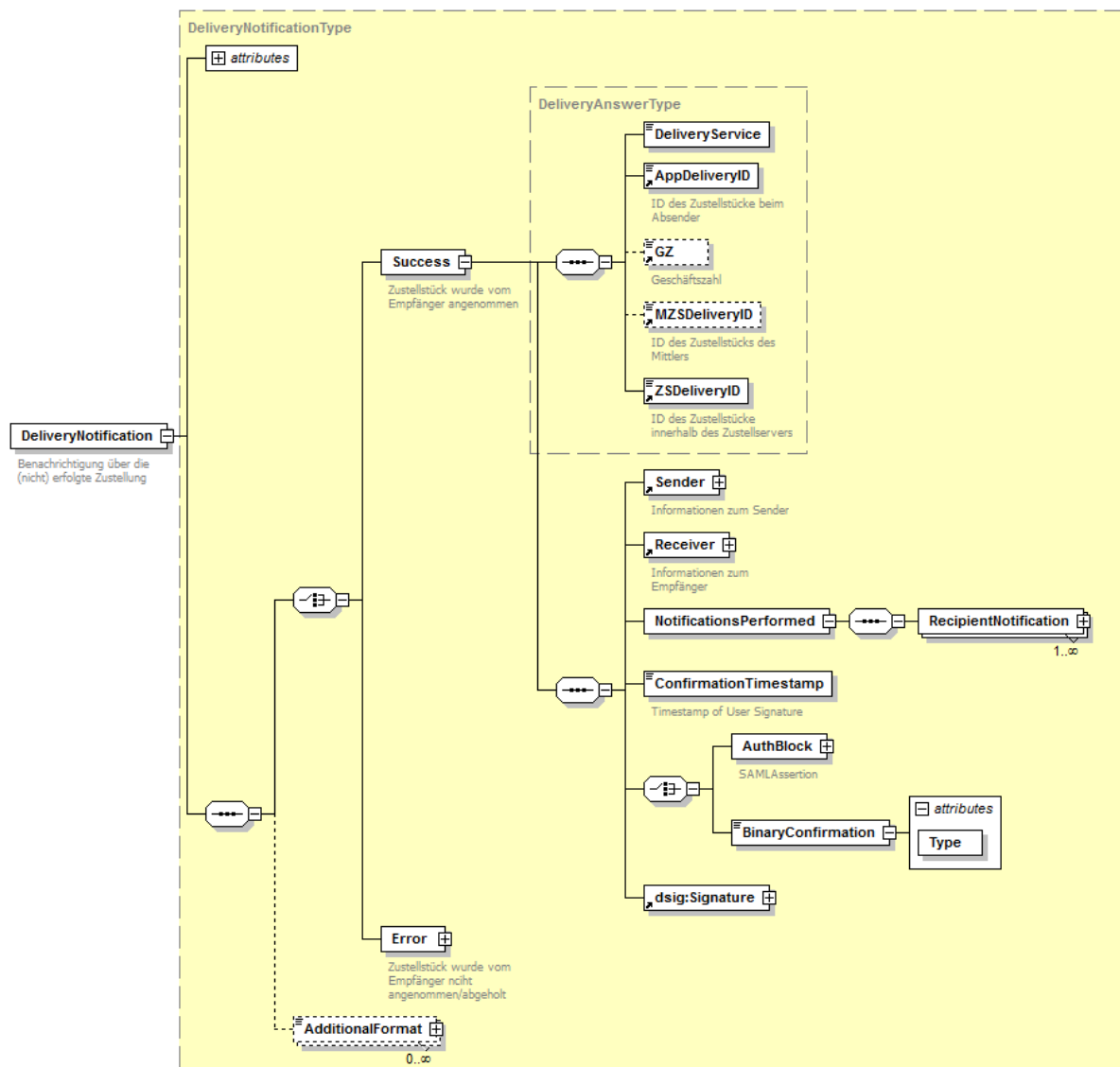
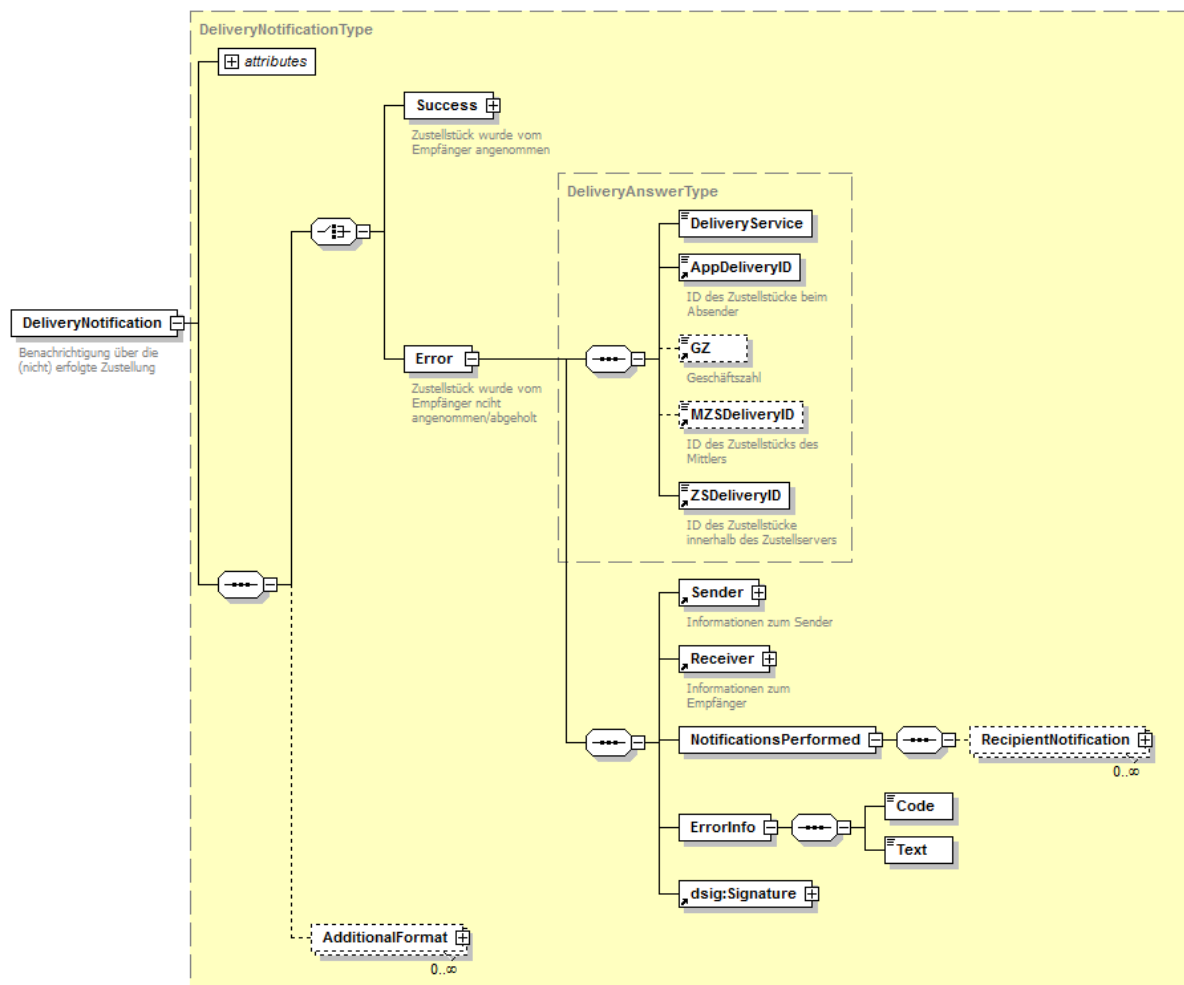




Abbildung 11 – Benachrichtigung über eine nicht erfolgreiche Zustellung



Die nachfolgenden Abschnitte erläutern was im Erfolgs- und Fehlerfall im Rahmen einer **DeliveryNotification**-Nachricht zu übermitteln ist.

## 6.1 Erfolgsfall (Success)

Im Erfolgsfall MÜSSEN der Zeitpunkt der Übernahme durch den Empfänger (**ConfirmationTimestamp**) und die Signatur des Empfängers (**AuthBlock**) in der **DeliveryNotification**-Nachricht übermittelt werden.

Um die Akzeptanz des Verfahrens zu erhöhen, muss der Empfänger nicht den Empfang jedes einzelnen Schriftstücks bei der Übernahme durch dessen Signatur gesondert bestätigen, sondern jede an ihn gerichtete Zustellung, die sich zum Zeitpunkt der Anmeldung (Login) des Empfängers beim Zustelldienst in dessen Zustellpostfach befindet, gilt mit dessen einmaliger Signatur (Signatur am **AuthBlock**) als angenommen.

Der Empfänger signiert eine textuelle Übernahmebestätigung, welche die erforderlichen Daten gemäß [ZUSESPEC] beinhalten MUSS.

Aus diesen Daten erstellt der Zustelldienst eine XML-Struktur – den sogenannten **AuthBlock** – der die obengenannten Daten in XML-Form sowie die elektronische Signatur des Empfängers (als XMLDSig [XMLDSIG]; enveloped Signature) beinhaltet. Das XSLT-

Stylesheet, das zur Erzeugung der lesbaren, textuellen Repräsentation des `AuthBlock` benutzt worden ist, MUSS in der XMLDSig-Signatur des Empfängers im Zuge der XSLT-Transformation enthalten sein. Zustellnachweise (Beispiel siehe Abschnitt 8.5) MÜSSEN durch den Zustelldienst ehest möglich an den Sender übermittelt werden.

Die technische Umsetzung des `AuthBlock` steht dem Betreiber des Zustelldienstes frei. Der Zustelldienst trägt aber die Verantwortung dafür, dass der von ihm ausgegebene `AuthBlock` bei Bedarf (im Streitfall) verifiziert werden kann.

Im Falle der dualen Zustellung kann an Stelle des `AuthBlock` ein digitalisierter RSa-Rückschein (z.B. Scan) verwendet werden. Für diesen Zweck ist das Element `BinaryConfirmation` vorgesehen, dessen Inhalt den Base64-kodierten Rückschein enthält. Das obligatorische Attribut `Type` gibt den Dokumententyp des digitalisierten Rückscheins an.

Falls die im `DeliveryRequest` optionalen Empfänger-Informationen (das `Receiver` Element) nicht vorhanden ist, MÜSSEN die Empfänger-Informationen direkt aus dem Datenbestand des Zustelldienstes übernommen werden, um so das `Receiver`-Element der `DeliveryNotification`-Nachricht zu erzeugen.

## 6.2 Nichterfolgsfall (Error)

Im Fehlerfall wird ein `ErrorInfo`-Element übermittelt, das den Grund des Misserfolges angibt. Die anwendbaren Fehler-Codes sind in Tabelle 3 angeführt; nähere Definitionen dazu siehe Abschnitt 4.

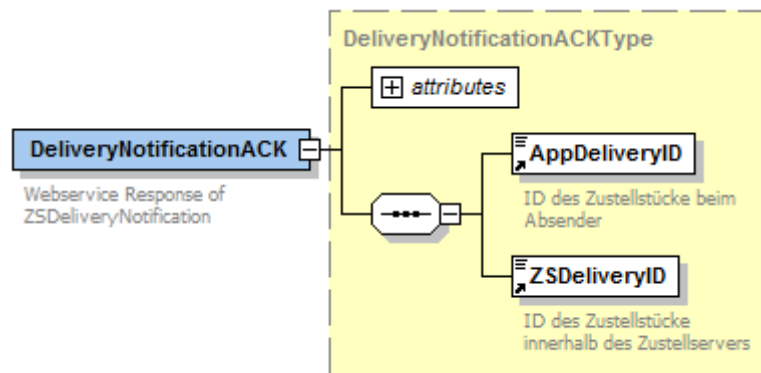
**Tabelle 3 – Fehler-Codes im Nichterfolgsfall**

<i>Server und formale Fehler bei Anlieferung von Zustellstücken</i>		
<b>Code</b>	<b>Error-Message</b>	<b>File</b>
506	Electronic notification of the recipient failed	–
507	Receiver did not pick up delivery	–

## 7 Datenformat für die Bestätigung der Benachrichtigung

Mit dieser Nachricht bestätigt die Applikation den Erhalt einer `DeliveryNotification`-Nachricht (siehe Abbildung 12).

Abbildung 12 – XML Datenformat für die Antwort auf die Zustellbenachrichtigung



## 8 APPENDIX

Der Spezifikation sind eine Reihe von technischen XML-Beschreibungen (Schemata) beigelegt. Diese Schemata sind normativer Bestandteil der Spezifikation. Besonders Feldlängen und Datentypen werden ausschließlich darin festgelegt und MÜSSEN beachtet werden.

Die Schemata werden parallel zu den vorliegenden textuellen Spezifikationen veröffentlicht.

In den XML-Schemata wurde ab der Version 1.3.0 ein optionales Version-Attribut (`version`) eingeführt, das jede Nachricht (jeden Request oder Response) optional mit der angewandten Version kennzeichnet. Aus Gründen der Abwärtskompatibilität ist dieses Attribut optional. Fehlt dieses Attribut, so wird von einer Nachricht gem. der Vorgängerspezifikation bzw. Schema-Version (1.2.0) ausgegangen. Somit wird ein Maximum an Abwärtskompatibilität erreicht.

Dieser Abschnitt listet ergänzend einige zusätzliche, für die Schnittstelle relevante Beschreibungen und Beispiele auf.

### 8.1 WSDL Beschreibung des Interfaces Applikation (APP2ZUSE)

Die entsprechende WSDL-Beschreibung ist als gesonderte Datei der Spezifikation beigelegt.

- `app2zuse.wsdl`

### 8.2 WSDL Beschreibung des Interfaces ZUSE Applikation (ZUSE2APP)

Die entsprechende WSDL-Beschreibung ist als gesonderte Datei der Spezifikation beigelegt.

- `zuse2app.wsdl`

### 8.3 Beispiel: Anlieferung eines PDF Zustellstücks

```
MIME-Version: 1.0
Content-Type: Multipart/Related; boundary=__SwA__MIME__boundary;

__SwA__MIME__boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <application@appl.com>

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding" xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
xmlns="http://reference.e-
government.gv.at/namespace/moazs/zuse11#"
xmlns:p="http://reference.e-
government.gv.at/namespace/persondata/20020228#">
  <soap:Body>
    <DeliveryRequest>
      <ZbPK>8238748934987asdkj12398urqdjgoi==</ZbPK>
      <NotificationAddress>
        <WebserviceURL>://delivery.somewhere.at/delSoap
```

```

</WebserviceURL>
  </NotificationAddress>
  <Sender>
    <Organisation>
      <FullName>Bundeskanzleramt</FullName>
    </Organisation>
    <p:PostalAddress>
      <p:PostalCode>1014</p:PostalCode>
      <p:Municipality>Wien</p:Municipality>
      <p:DeliveryAddress>

<p:StreetName>Ballhausplatz</p:StreetName>

<p:BuildingNumber>2</p:BuildingNumber>
      </p:DeliveryAddress>
    </p:PostalAddress>
  </Sender>
  <MetaData>
    <AppDeliveryID>123456</AppDeliveryID>
    <GZ>GZ/12345</GZ>
    <DeliveryQuality>RSa</DeliveryQuality>
  </MetaData>
</DeliveryRequest>
</soap:Body>
</soap:Envelope>
__SwA__MIME__boundary
MIME-Version: 1.0
Content-Type:Multipart/mixed; boundary=__Sendung__MIME__boundary

This is a multi-part message in MIME format.
__Sendung__MIME__boundary
Content-Type: text/plain; charset=us-ascii; format=flowed
Content-Transfer-Encoding: 7bit
Content-Disposition:inline;

Das ist eine elektronische Zustellung!
__Sendung__MIME__boundary
Content-Type: application/pdf; name="test.pdf"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="test.pdf"

JVBERi0xLjQNCiXk9tzfdQoxIDAgb2JqDQo8PCAvTG VuZ3RoIDIgM CBSDQogICAvRm
lsdGVyIC9GbGF0ZURlY29kZQ0KPj4NCnN0cmVhbQ0KeJyVVNtqwzAMfQ/4H/RcmGf5
GkMZNF37XgjsB7ZujG3Q...
...
__Sendung__MIME__boundary
__SwA__MIME__boundary

```

## 8.4 Beispiel: Anlieferung einer verschlüsselten Sendung

```

MIME-Version: 1.0
Content-Type: Multipart/Related;boundary=__SwA__MIME__boundary;

__SwA__MIME__boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <application@appl.com>

<?xml version="1.0" encoding="UTF-8"?>

```

```

<soap:Envelope soap:encodingStyle="http://www.w3.org/2001/12/soap-
encoding" xmlns="http://reference.e-
government.gv.at/namespace/moazs/zuse11#"
xmlns:p="http://reference.e-
government.gv.at/namespace/persondata/20020228#"
xmlns:soap="http://www.w3.org/2001/12/soap-envelope">
  <soap:Body>
    <DeliveryRequest>
      <ZbPK>8238748934987asdkj12398urqdjqoi==</ZbPK>
      <NotificationAddress>
        <WebServiceURL>http://delivery.somewhere.at/delSoap</WebServiceU
RL>
        </NotificationAddress>
        <Sender>
          <Organisation>
            <FullName>Bundeskanzleramt</FullName>
          </Organisation>
          <p:PostalAddress>
            <p:PostalCode>1014</p:PostalCode>
            <p:Municipality>Wien</p:Municipality>
            <p:DeliveryAddress>

<p:StreetName>Ballhausplatz</p:StreetName>

<p:BuildingNumber>2</p:BuildingNumber>
          </p:DeliveryAddress>
          </p:PostalAddress>
        </Sender>
        <MetaData>
          <AppDeliveryID>123456</AppDeliveryID>
          <GZ>GZ/12345</GZ>
          <DeliveryQuality>RSa</DeliveryQuality>
        </MetaData>
      </DeliveryRequest>
    </soap:Body>
  </soap:Envelope>
__SwA__MIME__boundary
MIME-Version: 1.0
Content-Type: application/x-pkcs7-mime; name="smime.p7m"
Content-Transfer-Encoding: base64

MIMC7VMGCSqGSIB3DQEHA6CDAu1DMIMC7T4CAQAxggEuMIIBKgIBADCBkjCBizEL
MAkGA1UEBhMCQVQxSDBGBGgNVBAoTP0EtVHJlc3QgR2VzLiBmLiBTaWNNoZXJoZWl0
iBTaWNNoZXJoZWl0.....
...
__SwA__MIME__boundary

```

## 8.5 Beispiel: Zustellnachweis

```

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope soap:encodingStyle="http://www.w3.org/2001/12/soap-
encoding" xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
xmlns:p="http://reference.e-
government.gv.at/namespace/persondata/20020228#"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
  <soap:Body>
    <z:DeliveryNotification
xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#"
xmlns:zs="http://reference.e-
government.gv.at/namespace/zustellung/msg"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

```

```

    <zs:Success>
    <zs:DeliveryService>https://www.zustellung.gv.at</zs:DeliveryService>

    <zs:AppDeliveryID>Musterbescheid_47110815</zs:AppDeliveryID>
      <zs:MZSDeliveryID>118532</zs:MZSDeliveryID>
      <zs:ZSDeliveryID>11687</zs:ZSDeliveryID>
      <zs:Sender>
        <Organisation xmlns="http://reference.e-government.gv.at/namespace/zustellung/msg">
          <p:FullName>Musteramt</p:FullName>
        </Organisation>
        <p:Address type="undefined"
xsi:type="p:PostalAddressType">
          <p:CountryCode>AT</p:CountryCode>
          <p:PostalCode>0000</p:PostalCode>

          <p:Municipality>Graz</p:Municipality>
          <p:DeliveryAddress>

          <p:StreetName>Musterstrasse</p:StreetName>

          <p:BuildingNumber>5</p:BuildingNumber>
          <p:Unit>1</p:Unit>
        </p:DeliveryAddress>
        </p:Address>
        <p:Address
xsi:type="p:InternetAddressType">
          <p:Address>musteramt@musteramt.gv.at</p:Address>
        </p:Address>
      </zs:Sender>
      <zs:Receiver>
        <p:PhysicalPerson>
          <p:Name>

          <p:GivenName>Max</p:GivenName>

          <p:FamilyName>Mustermann</p:FamilyName>
          </p:Name>
          <p:DateOfBirth>1972-02-
22</p:DateOfBirth>
          </p:PhysicalPerson>
          <p:PostalAddress>
            <p:PostalCode>1200</p:PostalCode>

            <p:Municipality>Wien</p:Municipality>
            <p:DeliveryAddress>
              <p:StreetName>Mustermannstrasse</p:StreetName>

              <p:BuildingNumber>5</p:BuildingNumber>
            </p:DeliveryAddress>
          </p:PostalAddress>
        </zs:Receiver>
        <zs:NotificationsPerformed>
          <zs:RecipientNotification>
            <zs:NotificationMethod>electronic</zs:NotificationMethod>
            <zs:Timestamp>2012-11-
30T22:53:07+01:00</zs:Timestamp>
          </zs:RecipientNotification>
        </zs:NotificationsPerformed>
        <zs:ConfirmationTimestamp>
2012-11-30T22:55:42+01:00

```

```

</zs:ConfirmationTimestamp>
    <zs:AuthBlock AssertionID="any"
IssueInstant="2006-11-30T22:55:27+01:00" Issuer="Max Mustermann"
MajorVersion="1" MinorVersion="0">
        <saml:AttributeStatement>
            <saml:Subject>
                <saml:NameIdentifier>://www.zustellung.gv.at:443/moa-
id-auth/
</saml:NameIdentifier>
                                </saml:Subject>
                                <saml:Attribute
AttributeNames="Geschaeftsbereich" AttributeNamespace="...">
                <saml:AttributeValue>:publicid:gv.at:cdid+ZU
</saml:AttributeValue>
                                </saml:Attribute>
                                <saml:Attribute AttributeName="OA"
AttributeNamespace="...">
                <saml:AttributeValue>://www.zustellung.gv.at/Zustellservice
</saml:AttributeValue>
                                </saml:Attribute>
                                <saml:Attribute
AttributeNames="Geburtsdatum" AttributeNamespace="...">
                <saml:AttributeValue>1979-09-
07</saml:AttributeValue>
                                </saml:Attribute>
            </saml:AttributeStatement>
            <dsig:Signature Id="signature-1164923496-
175982843-14580">
                <!-- enveloped DSig signiert gesamte Assertion
-->
                                </dsig:Signature>
        </zs:AuthBlock>
        <dsig:Signature Id="signature-1-1"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
            <!-- Daten der Signatur des Zustelldienstes
über den SOAP-Body -->
        </dsig:Signature>
    </zs:Success>
</zs:DeliveryNotification>
</soap:Body>
</soap:Envelope>

```

## 8.6 Beispiel: PrintParameter

```

<?xml version="1.0" encoding="UTF-8"?>
<PrintParameter
profile="http://example.gv.at/namespace/zustellung/printprofile/ex
ample-20051128#"
xmlns:pp="http://example.gv.at/namespace/zustellung/printprofile/e
xample-20051128">
    <xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
        <!-- An XSLT stylesheet for transforming the delivered XML
document into andocument, which is then transformed into a
printable PDF using anFO stylesheet provided by the delivery
service. -->
        <xsl:include
href="http://example.gv.at/zustellung/xslt/xml-xhtml.xsl"/>
    </xsl:stylesheet>
    <pp:pageformat>
        <pp:pagesize>A4</pp:pagesize>
    </pp:pageformat>
</PrintParameter>

```



```

    <pp:orientation>portrait</pp:orientation>
    <pp:pagemargins>
      <pp:top>2.0cm</pp:top>
      <pp:bottom>2.0cm</pp:bottom>
      <pp:left>1.5cm</pp:left>
      <pp:right>1.5cm</pp:right>
      <pp:innermargin>1.0cm</pp:innermargin>
    </pp:pagemargins>
    <pp:pagespread>double</pp:pagespread>
  </pp:pageformat>
  <pp:enveloping>
    <pp:envelope>C5</pp:envelope>
  </pp:enveloping>
</PrintParameter>

```

## 8.7 Beispiel: Zustellung Fehlgeschlagen

```

<?xml version="1.0" encoding="UTF-8"?>
<zs:DeliveryNotification
  xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#"
  xmlns:zs="http://reference.e-
government.gv.at/namespace/zustellung/msg"
  xmlns:p="http://reference.e-
government.gv.at/namespace/persondata/20020228#"
  xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
  xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
  <zs:Error>

    <zs:DeliveryService>https://www.zustellung.gv.at</zs:DeliverySer-
vice>

    <zs:AppDeliveryID>Test GZ 1-TA-0815-4711</zs:AppDeliveryID>
    <zs:MZSDeliveryID>1161183</zs:MZSDeliveryID>
    <zs:ZSDeliveryID>12697</zs:ZSDeliveryID>
    <zs:Sender>
      <Organisation xmlns="http://reference.e-
government.gv.at/namespace/zustellung/msg">
        <p:FullName>Musteramt</p:FullName>
      </Organisation>
      <p:Address type="undefined"
xsi:type="p:PostalAddressType">
        <p:CountryCode>AT</p:CountryCode>
        <p:PostalCode>0000</p:PostalCode>
        <p:Municipality>Graz</p:Municipality>
        <p:DeliveryAddress>

          <p:StreetName>Musterstrasse</p:StreetName>
          <p:BuildingNumber>5</p:BuildingNumber>
          <p:Unit>1</p:Unit>
        </p:DeliveryAddress>
      </p:Address>
      <p:Address xsi:type="p:InternetAddressType">

        <p:Address>musteramt@musteramt.gv.at</p:Address>
      </p:Address>
    </zs:Sender>
    <zs:Receiver>
      <p:PhysicalPerson>
        <p:Name>
          <p:GivenName>Max</p:GivenName>
          <p:FamilyName>Mustermann</p:FamilyName>
        </p:Name>
        <p:DateOfBirth>1972-02-22</p:DateOfBirth>
      </p:PhysicalPerson>
    </zs:Receiver>
  </zs:Error>
</zs:DeliveryNotification>

```

```

        </p:PhysicalPerson>
        <p:PostalAddress>
            <p:PostalCode>1200</p:PostalCode>
            <p:Municipality>Wien</p:Municipality>
            <p:DeliveryAddress>

        <p:StreetName>Mustermannstrasse</p:StreetName>
            <p:BuildingNumber>5</p:BuildingNumber>
            </p:DeliveryAddress>
        </p:PostalAddress>
    </zs:Receiver>
    <zs:NotificationsPerformed>
        <zs:RecipientNotification>

    <zs:NotificationMethod>electronic</zs:NotificationMethod>
        <zs:Timestamp>2006-12-
01T17:21:07+01:00</zs:Timestamp>
        </zs:RecipientNotification>
    </zs:NotificationsPerformed>
    <zs:ErrorInfo>
        <zs:Code>507</zs:Code>
        <zs:Text>Receiver did not pick up delivery</zs:Text>
    </zs:ErrorInfo>
    <dsig:Signature Id="signature-1-1"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
        <!-- Daten der Signatur des Zustelldienstes über den
SOAP-Body -->
    </dsig:Signature>
</zs:Error>
</zs:DeliveryNotification>

```

## 8.8 Beispiel: Mailbody

Sehr geehrte Damen und Herren,

bei dieser Nachricht handelt sich um eine elektronische Zustellung, die auf Ihren Wunsch an Sie per E-Mail übermittelt wurde.

Alle weiteren Informationen finden Sie in der Anlage (Attachment). Bei weiteren Fragen wenden Sie sich bitte an Ihren Zustelldienst.

Vielen Dank

## 8.9 AuthBlock auf SAML Basis

Diese AuthBlock-Lösung lehnt sich stark an die SAML-Struktur [SAML] an, die auch bei MOA-ID zum Einsatz kommt. Abbildung 15 und Beispiel 7 zeigen die interne Struktur des AuthBlock Elements. Mit der Assertion (Behauptung) innerhalb des AuthBlock bestätigt der Empfänger (@Issuer) durch seine elektronische Signatur (dsig:Signature), zu einem bestimmten Zeitpunkt (@IssuerInstant), dass die Identifizierungs-/Authentifizierungskomponente des Zustelldienstes (saml:Subject) berechtigt ist, ihn (den Empfänger) gegenüber einer Applikation (saml:Attribute AttributeName="OA") eines bestimmten Geschäftsbereichs zu authentifizieren.

Die AuthBlock Struktur ist identisch jener, die auch bei der Signatur mittels MOA-ID [MOAID] zum Einsatz kommt. Dadurch ist es möglich, dass die von MOA-ID im Zuge der Anmeldung am Zustelldienst erstellte und vom Anwender signierte AuthBlock sofort und unmittelbar im Rahmen einer Zustellbestätigung verwendet werden kann.

Die SAML-Assertion enthält die Attribute Issuer und IssuerInstant, deren Werte gemeinsam mit den Werten der saml:Attribute-Elemente für Geschäftsbereich und Applikation, nach Aufbereitung durch ein XSLT-Stylesheet, dem Anwender in Form einer HTML-Darstellung zur Signatur vorgelegt werden. Zu beispielhaften HTML-Darstellungen siehe Abbildung 13 und Abbildung 14.

```
<sl:CreateXMLSignatureRequest
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:sl="http://www.buergerkarte.at/namespaces/securitylayer/1.2#"
">
  <sl:KeyboxIdentifier>SecureSignatureKeypair</sl:KeyboxIdentifier
>
  <sl:DataObjectInfo Structure="detached">
    <sl:DataObject Reference=""/>
    <sl:TransformsInfo>
      <dsig:Transforms
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
        <dsig:Transform
Algorithm="http://www.w3.org/TR/1999/REC-xslt-19991116">
          <xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:pr="http://reference.e-
government.gv.at/namespace/persondata/20020228#" exclude-result-
prefixes="pr saml">
                                <xsl:output method="xml"
xml:space="default"/>
                                <xsl:template match="/"
xmlns="http://www.w3.org/1999/xhtml">
                                  <html>
                                    <head>
                                      <title>Signatur
der Anmeldedaten</title>
                                      <style
type="text/css" media="screen">
                                          .normalstyle { font-
size: medium; }
                                          .italicstyle { font-
size: medium; font-style: italic; }
                                          .titlestyle{ text-
decoration:underline; font-weight:bold; font-size: medium; }
                                          .h4style{ font-size:
large; }
                                  </style>
                                    </head>
                                    <body>
                                      <h4
class="h4style">Anmeldedaten:</h4>
                                      <p
class="titlestyle">Daten zur Person</p>
                                      <table
class="parameters">
                                          <xsl:if
test="normalize-space(//@Issuer)">
                                                                <tr>
```

```

<td class="italicstyle">Name:</td>

<td class="normalstyle">

<xsl:value-of select="//@Issuer"/>

</td>

</tr>
</xsl:if>
<xsl:if
test="string(//saml:Attribute[@AttributeName='Geburtsdatum']/saml:
AttributeValue)">
<tr>

<td class="italicstyle">Geburtsdatum:</td>

<td class="normalstyle">

<xsl:value-of
select="substring(//saml:Attribute[@AttributeName='Geburtsdatum']/
saml:AttributeValue,9,2)"/>

<xsl:text>.</xsl:text>

<xsl:value-of
select="substring(//saml:Attribute[@AttributeName='Geburtsdatum']/
saml:AttributeValue,6,2)"/>

<xsl:text>.</xsl:text>

<xsl:value-of
select="substring(//saml:Attribute[@AttributeName='Geburtsdatum']/
saml:AttributeValue,1,4)"/>

</td>

</tr>
</xsl:if>
<xsl:if
test="//saml:Attribute[@AttributeName='OIDTextualDescription']">
<tr>

<td class="italicstyle">Rolle:</td>

<td class="normalstyle">

<xsl:value-of
select="//saml:Attribute[@AttributeName='OIDTextualDescription']/s
aml:AttributeValue"/>

</td>

</tr>
</xsl:if>
<xsl:if
test="//saml:Attribute[@AttributeName='mandateReferenceValue']">
<tr>

<td class="italicstyle">Vollmacht:</td>

<td class="normalstyle">

```

```

    <xsl:text>Ich melde mich in Vertretung an. Im nächsten Schritt
    wird mir eine Liste der für mich verfügbaren
    Vertretungsverhältnisse angezeigt, aus denen ich eines auswählen
    werde.</xsl:text>

```

```

    </td>

```

```

    </tr>

```

```

    </xsl:if>

```

```

</table>

```

```

<p>

```

```

class="titlestyle">Daten zur Anwendung</p>

```

```

<table>

```

```

class="parameters">

```

```

    <tr>

```

```

        <td>

```

```

class="italicstyle">Name:</td>

```

```

        <td>

```

```

class="normalstyle">

```

```

    <xsl:value-of
    select="//saml:Attribute[@AttributeName='oaFriendlyName']/saml:Att
    ributeValue"/>

```

```

    </td>

```

```

    </tr>

```

```

    <tr>

```

```

        <td>

```

```

class="italicstyle">Staat:</td>

```

```

        <td>

```

```

class="normalstyle">Österreich</td>

```

```

    </tr>

```

```

</table>

```

```

<p>

```

```

class="titlestyle">Technische Parameter</p>

```

```

<table>

```

```

class="parameters">

```

```

    <tr>

```

```

        <td>

```

```

class="italicstyle">URL:</td>

```

```

        <td>

```

```

class="normalstyle">

```

```

    <xsl:value-of
    select="//saml:Attribute[@AttributeName='OA']/saml:AttributeValue"
    />

```

```

    </td>

```

```

    </tr>

```

```

    <xsl:if

```

```

test="//saml:Attribute[@AttributeName='Geschaeftsbereich']">

```

```

    <tr>

```

```

    <td class="italicstyle">Bereich:</td>

```

```

    <td class="normalstyle">

```

```

    <xsl:value-of
    select="//saml:Attribute[@AttributeName='Geschaeftsbereich']/saml:
    AttributeValue"/>

```

```

    </td>

```

```

    </tr>

```

```

    </xsl:if>

```

```

                                <xsl:if
test="//saml:Attribute[@AttributeName='mandateReferenceValue']">
                                <tr>

        <td class="italicstyle">

Referenz:</td>

        <td class="normalstyle">

        <xsl:value-of
select="//saml:Attribute[@AttributeName='mandateReferenceValue']"/
>

        </td>

                                </tr>
                                </xsl:if>
                                <xsl:if
test="//saml:Attribute[@AttributeName='IdentityLinkDomainIdentifie
rType']">

                                <tr>

        <td class="italicstyle">

        <xsl:value-of
select="//saml:Attribute[@AttributeName='IdentityLinkDomainIdentif
ierType']"/></td>

        <td class="normalstyle">

        <xsl:value-of
select="//saml:Attribute[@AttributeName='wbPK']/saml:AttributeValu
e/pr:Identification/pr:Type"/>

        </td>

                                </tr>
                                </xsl:if>
                                <xsl:if
test="//saml:Attribute[@AttributeName='bPK'] or
//saml:Attribute[@AttributeName='wbPK']">

                                <tr>

        <td class="italicstyle">Identifikator:</td>

        <td class="normalstyle">

        <xsl:value-of
select="//saml:Attribute[@AttributeName='bPK']/saml:AttributeValue
/pr:Identification/pr:Value"/>

        <xsl:value-of
select="//saml:Attribute[@AttributeName='wbPK']/saml:AttributeValu
e/pr:Identification/pr:Value"/>

        </td>

                                </tr>
                                </xsl:if>
                                <xsl:if
test="//saml:Attribute[@AttributeName='OIDTextualDescription']">

                                <tr>

        <td class="italicstyle">OID:</td>

```

```

<td class="normalstyle">

  <xsl:value-of
select="//saml:Attribute[@AttributeName='OID']/saml:AttributeValue
"/>

</td>

</tr>
</xsl:if>
<xsl:if
test="//saml:Attribute[@AttributeName='HPI']">
  <tr>

    <td class="italicstyle">HPI:</td>

    <td class="normalstyle">

      <xsl:value-of
select="//saml:Attribute[@AttributeName='HPI']/saml:AttributeValue
"/>

    </td>

    </tr>
  </xsl:if>
</tr>

class="italicstyle">Datum:</td>
  <td
class="normalstyle">

    <xsl:value-of select="substring(//@IssueInstant,9,2)"/>

    <xsl:text>.</xsl:text>

    <xsl:value-of select="substring(//@IssueInstant,6,2)"/>

    <xsl:text>.</xsl:text>

    <xsl:value-of select="substring(//@IssueInstant,1,4)"/>

  </td>
</tr>
<tr>
  <td
class="italicstyle">Uhrzeit:</td>
  <td
class="normalstyle">

    <xsl:value-of select="substring(//@IssueInstant,12,2)"/>

    <xsl:text>:</xsl:text>

    <xsl:value-of select="substring(//@IssueInstant,15,2)"/>

    <xsl:text>:</xsl:text>

    <xsl:value-of select="substring(//@IssueInstant,18,2)"/>

  </td>
</tr>
</table>
</body>
</html>

```

```

        </xsl:template>
    </xsl:stylesheet>
    </dsig:Transform>
    <dsig:Transform
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315#WithComments"/>
        </dsig:Transforms>
    <sl:FinalDataMetaInfo>

    <sl:MimeType>application/xhtml+xml</sl:MimeType>
        </sl:FinalDataMetaInfo>
    </sl:TransformsInfo>
</sl:DataObjectInfo>
<sl:SignatureInfo>
    <sl:SignatureEnvironment>
        <sl:XMLContent>
            <saml:Assertion
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:pr="http://reference.e-
government.gv.at/namespace/persondata/20020228#" MajorVersion="1"
MinorVersion="0" AssertionID="any" Issuer="Max Mustermann"
IssueInstant="2012-01-27T13:23:56+01:00">
                <saml:AttributeStatement>
                    <saml:Subject>

                    <saml:NameIdentifier>https://www.mein-zustelldienst.at/moa-id-
auth/</saml:NameIdentifier>

                    </saml:Subject>
                    <saml:Attribute
AttributeName="Geschaeftsbereich"
AttributeNamespace="http://reference.e-
government.gv.at/namespace/moa/20020822#">
                        <saml:AttributeValue>ZU
(Zustellungen)</saml:AttributeValue>
                    </saml:Attribute>
                    <saml:Attribute AttributeName="OA"
AttributeNamespace="http://reference.e-
government.gv.at/namespace/moa/20020822#">
                        <saml:AttributeValue>https://www.mein-
zustelldienst.at/Zustellservice/</saml:AttributeValue>
                    </saml:Attribute>
                    <saml:Attribute
AttributeName="Geburtsdatum"
AttributeNamespace="http://reference.e-
government.gv.at/namespace/moa/20020822#">
                        <saml:AttributeValue>1979-08-
21</saml:AttributeValue>
                    </saml:Attribute>
                    <saml:Attribute AttributeName="bPK"
AttributeNamespace="http://reference.e-
government.gv.at/namespace/moa/20020822#">
                        <saml:AttributeValue>
<pr:Identification
xmlns:pr="http://reference.e-
government.gv.at/namespace/persondata/20020228#">

                            <pr:Value>arTa6sZRiGVK0btb8FLbparv43w=</pr:Value>

                            <pr:Type>urn:publicid:gv.at:cdid+bpk</pr:Type>
                        </pr:Identification>
                    </saml:AttributeValue>

```



```

        </saml:Attribute>
        <saml:Attribute
AttributeName="mandateReferenceValue"
AttributeNamespace="http://reference.e-
government.gv.at/namespace/moa/20020822#">

        <saml:AttributeValue>6451328752401945321301</saml:AttributeValue
        >
        </saml:Attribute>
        <saml:Attribute
AttributeName="oaFriendlyName"
AttributeNamespace="http://reference.e-
government.gv.at/namespace/moa/20020822#">
        <saml:AttributeValue>Mein
Zustelldienst</saml:AttributeValue>
        </saml:Attribute>
        </saml:AttributeStatement>
    </saml:Assertion>
</sl:XMLContent>
</sl:SignatureEnvironment>
<sl:SignatureLocation
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
Index="2">/saml:Assertion</sl:SignatureLocation>
</sl:SignatureInfo>
</sl:CreateXMLSignatureRequest>

```

### Beispiel 7: XML Signatur des Authblocks mittels Bürgerkarte

**Abbildung 13 – Beispiel der HTML-Darstellung eines AuthBlock; diese wird durch den Empfänger elektronisch signiert.**

#### Anmeldedaten:

##### Daten zur Person:

Name: Arne Tauber

Geburtsdatum: 21.08.1979

##### Daten zur Anwendung:

Name: BRZ Zustelldienst

Staat: Österreich

##### Technische Parameter:

URL: <https://www.brz-zustelldienst.at/Zustellservice/>

Bereich: ZU (Zustellungen)

Identifikator: uTvh8sZRiOVK0mtb8FLbparv43w=

Datum: 27.01.2012

Uhrzeit: 13:37:44

**Abbildung 14 – Beispiel der HTML-Darstellung eines AuthBlock; hier signiert der Vertreter im Namen einer Firma (die Firma ist der Empfänger der Zustellung).**

**Anmeldedaten:**

Daten zur Person:

Name: Arne Tauber

Geburtsdatum: 21.08.1979

Vollmacht: Ich melde mich in Vertretung an. Im nächsten Schritt wird mir eine Liste der für mich verfügbaren Vertretungsverhältnisse angezeigt, aus denen ich eines auswählen werde.

Daten zur Anwendung:

Name: BRZ Zustelldienst

Staat: Österreich

Technische Parameter:

URL: <https://www.brz-zustelldienst.at/Zustellservice/>

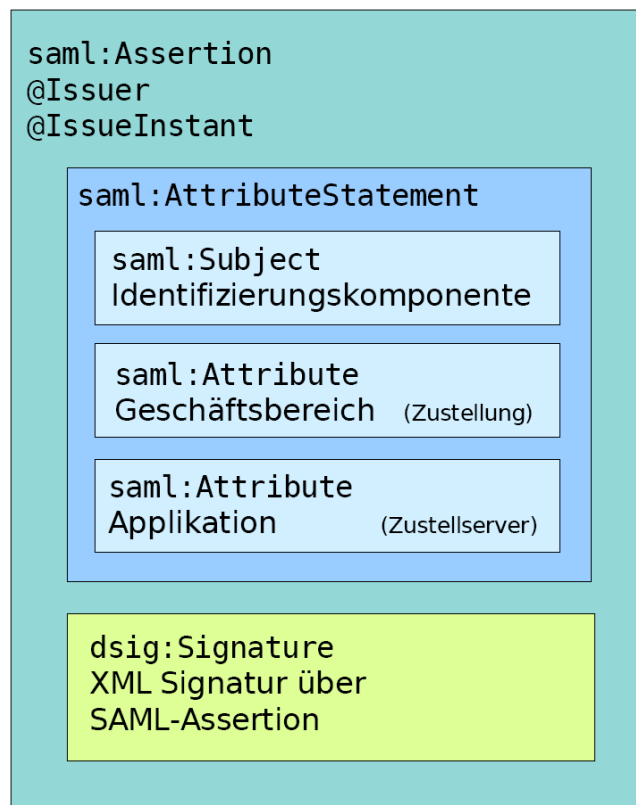
Bereich: ZU (Zustellungen)

Vollmachten-Referenz: 6082652938987242307

Identifikator: uTvh8sZRiOVK0mtb8FLbparv43w=

Datum: 27.01.2012

Uhrzeit: 13:35:44

**Abbildung 15 – Interne Struktur eines AuthBlock (auf Basis SAML-Assertion).**

## 8.10 Beispiel PDF Rückschein (RSa) mit Abholung durch einen Bevollmächtigten

### Elektronische Zustellung mit Zustellnachweis

GZ 1234567890

Empfänger

Max Mustermann

Verständigungen über die Bereithaltung zur Abholung

Absender

Erste elektronische Verständigung 01.01.2020 10:50

Musterbehörde, Musterplatz 1, 1010 Wien

Zweite elektronische Verständigung 03.01.2020 10:50

Dritte postalische Verständigung 04.01.2020 10:50

☐ Nicht an Bevollmächtigten

Bereithaltung zur Abholung

bei

Beginn der Abholfrist

Zustelldienst.at

01.01.2020


Übernahmebestätigung

Übernommen am 10.01.2020 10:50

☐ Empfänger

☒ Bevollmächtigter

Max Vertretermann

Signaturwert	AW2hydumViCg81e1cnSUwWlfwzcDHPciP/a40CnJu2eJO6SoRamTF861WQ9Vkf9	
	Unterzeichner	EGIZ DEMO Amtssignatur
	Datum/Zeit-UTC	2010-10-21T11:16:13+02:00
	Aussteller-Zertifikat	CN=MOA Test CA,OU=EGIZ,O=TU Graz,C=AT
	Serien-Nr.	19
	Methode	urn:pdfsigfilter:bka.gv.at:text:v1.2.0
	Parameter	etsi-bka-moa-1.0
Prüfinformation	Informationen zur Prüfung der elektronischen Signatur und des Ausdrucks finden Sie unter: <a href="http://www.signaturpruefung.gv.at">http://www.signaturpruefung.gv.at</a>	
Hinweis	Dieses Dokument wurde amtssigniert. Auch ein Ausdruck dieses Dokuments hat gemäß § 20 E-Government-Gesetz die Beweiskraft einer öffentlichen Urkunde.	

## 8.11 Beispiel PDF Rückschein (nonRSa)

### Elektronische Zustellung ohne Zustellnachweis

GZ 1234567890

Empfänger

Max Mustermann

Verständigungen über die Bereithaltung zur Abholung

Absender

Erste elektronische Verständigung 01.01.2020 10:50

Musterbehörde, Musterplatz 1, 1010 Wien

Zweite elektronische Verständigung 03.01.2020 10:50

☐ Nicht an Bevollmächtigten

Bereithaltung zur Abholung


bei  
Zustelldienst.at

Beginn der Abholfrist  
01.01.2020

Nichtabholung

Ende der Abholfrist

15.01.2020 10:50

	Signaturwert	AW2hyduViCq81e1cnSUwWlfwzcDHpCiP/a40CnJu2eJO6SoRamTF86iWQ9Vkf9	
	Unterzeichner	EGIZ DEMO Amtssignatur	
	Datum/Zeit-UTC	2010-10-21T11:16:13+02:00	
	Aussteller-Zertifikat	CN=MOA Test CA,OU=EGIZ,O=TU Graz,C=AT	
	Serien-Nr.	19	
	Methode	urn:pdfsigfilter:bka.gv.at:text:v1.2.0	
	Parameter	etsi-bka-moa-1.0	
Prüfinformation	Informationen zur Prüfung der elektronischen Signatur und des Ausdrucks finden Sie unter: <a href="http://www.signaturpruefung.gv.at">http://www.signaturpruefung.gv.at</a>		
Hinweis	Dieses Dokument wurde amtssigniert. Auch ein Ausdruck dieses Dokuments hat gemäß § 20 E-Government-Gesetz die Beweiskraft einer öffentlichen Urkunde.		

## A. Abbildungsverzeichnis

Abbildung 1 – Sequenz einer Zustellung.....	4
Abbildung 2 – Container-Struktur - Zustellteile werden innerhalb eines .ZUS-Files übermittelt .....	6
Abbildung 3 – Container-Struktur einer verschlüsselten Zustellung.....	8
Abbildung 4 – Gesamtstruktur der <code>DeliveryRequest</code> -Nachricht .....	11
Abbildung 5 – <code>DeliveryRequest/Sender</code> : Sender Information .....	12
Abbildung 6 – <code>DeliveryRequest/Receiver</code> : Empfänger Information .....	13
Abbildung 7 – XML-Datenformat für die Antwort auf Datenanlieferung (synchrone Kommunikation): <code>Success</code> , <code>PartialSuccess</code> und <code>Error</code> .....	21
Abbildung 8 – <code>DeliveryRequestStatusACK-Element</code> .....	22
Abbildung 9 – XML Schema für weitere Typen von Zustellnachweisen .....	24
Abbildung 10 – Benachrichtigung über eine erfolgreiche Zustellung .....	24
Abbildung 11 – Benachrichtigung über eine nicht erfolgreiche Zustellung.....	25
Abbildung 12 – XML Datenformat für die Antwort auf die Zustellbenachrichtigung .....	27
Abbildung 13 – Beispiel der HTML-Darstellung eines <code>AuthBlock</code> ; diese wird durch den Empfänger elektronisch signiert. ....	41
Abbildung 14 – Beispiel der HTML-Darstellung eines <code>AuthBlock</code> ; hier signiert der Vertreter im Namen einer Firma (die Firma ist der Empfänger der Zustellung). ....	42
Abbildung 15 – Interne Struktur eines <code>AuthBlock</code> (auf Basis SAML-Assertion). ....	43

## B. Tabellenverzeichnis

Tabelle 1 – Namespace-Deklarationen .....	9
Tabelle 2 – Fehlercodes im Zuge der Sendungsannahme .....	17
Tabelle 3 – Fehler-Codes im Nichterfolgsfall .....	26

## C. Revision History

Version	Datum	Autor(en)	
1.1.0	31.03.2004	IKT-Stabsstelle	<p>Anpassungen an das EGovG</p> <p>Verbesserte Unterstützung für automatische Codegenerierung</p> <p>Schemaabbildungen für bessere Übersicht hinzugefügt</p>
1.2.0	25.04.2007	Arne Tauber (EGIZ)	<p>Editorielle Korrekturen</p> <p>Schemaerweiterungen</p>
1.3.0	17.03.2008	Thomas Rössler (EGIZ)	Anpassung ZustG-Novelle (2008)
1.4.0	27.01.2012	<p>Arne Tauber (EGIZ)</p> <p>Peter Reichstädter (BKA)</p>	<p>Editorielle Änderungen</p> <p>Identification Element</p> <p>MZSDeliveryID optional</p> <p>PDF Zustellnachweis</p> <p>Update Beispiele</p> <p>Sub-postfächer</p> <p>Anpassung an ZUSE Version 1.4.0</p>



## D. Referenzen

[SwA]	W3C. SOAP with Attachments (SwA). <a href="http://www.w3.org/TR/2000/NOTE-SOAP-attachments-20001211">http://www.w3.org/TR/2000/NOTE-SOAP-attachments-20001211</a> , 2000.
[MOAID]	Rudolf Schamberger and Gregor Karlinger. MOA-ID Spezifikation. Version 1.1. Beziehbar unter <a href="http://egovlabs.gv.at">http://egovlabs.gv.at</a> .
[MD5]	R. Rivest. RFC 1321 The MD5 Message-Digest Algorithm, 1992.
[SMIME]	B. Ramsdell. RFC 2633 s/mime version 3 message specification. <a href="http://www.ietf.org/rfc/rfc2633.txt">http://www.ietf.org/rfc/rfc2633.txt</a> , 1999.
[ZIP]	PKWARE Inc. .zip le format Specification. <a href="http://www.pkware.com/products/enterprise/whitepapers/appnote.html">http://www.pkware.com/products/enterprise/whitepapers/appnote.html</a> , 2004.
[SAML]	OASIS. SAML 1.0 Specification. <a href="http://www.oasis-open.org/committees/download.php/2290/oasis-saml-1.0.zip">http://www.oasis-open.org/committees/download.php/2290/oasis-saml-1.0.zip</a> , 2002.
[ZUSTG]	Bundesgesetz: Zustellgesetz, BGBl. Nr. 200/1982 i.d.F. BGBl. I Nr. 5/2008.
[PERSDATA]	Larissa Naber: PersonData Struktur: XML Spezifikation – Dokumentation, E-Government Konvention, Version 2.0.0.
[PKCS1]	B. Kaliski. RFC2437 PKCS #1: RSA Cryptography Specifications. <a href="http://www.ietf.org/rfc/rfc2437.txt">http://www.ietf.org/rfc/rfc2437.txt</a> , 1998.
[CMS]	R. Housley. RFC 3369 Cryptographic Message Syntax (CMS), 2002.
[HTTP11]	R. Fielding and et. al. RFC 2616 Hypertext Transfer Protocol HTTP/1.1 . <a href="http://www.ietf.org/rfc/rfc2616.txt">http://www.ietf.org/rfc/rfc2616.txt</a> , 1999.
[RFC2015]	M. Elkins. RFC 2015 MIME Security with Pretty Good Privacy (PGP). <a href="http://www.ietf.org/rfc/rfc2015.txt">http://www.ietf.org/rfc/rfc2015.txt</a> , 1996.
[RFC2048]	Multipurpose Internet Mail Extensions (MIME), Nov. 1996; Freed. et. al. <a href="http://www.ietf.org/rfc/rfc2048.txt">http://www.ietf.org/rfc/rfc2048.txt</a>
[XMLDSIG]	Mark Bartel. XML Signature Syntax and Processing. <a href="http://www.w3.org/TR/xmlsig-core/">http://www.w3.org/TR/xmlsig-core/</a> , 2002.
[SOAP]	W3C. Simple Object Access Protocol (SOAP) 1.1. <a href="http://www.w3c.org/TR/2000/NOTE-SOAP-20000508/">http://www.w3c.org/TR/2000/NOTE-SOAP-20000508/</a> , 2000.
[WSDL]	W3C. Web Services Description Language (WSDL) 1.1. <a href="http://www.w3.org/TR/wsdl">http://www.w3.org/TR/wsdl</a>

[URL]	T. Berners Lee. RFC 1738 – Uniform Resource Locators (URL), 1994.
[X509]	S. Chokhani. RFC 3647 –Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework.  <a href="http://www.ietf.org/rfc/rfc3647.txt">http://www.ietf.org/rfc/rfc3647.txt</a> , 2003.
[XMLDSIG]	Mark Bartel, XML Signature Syntax and Processing. <a href="http://www.w3.org/TR/xmlsig-core/">http://www.w3.org/TR/xmlsig-core/</a> , 2002.
[ZUSEKOPF]	Tauber A., Rössler T., Elektronische Zustellung - Zustellkopf Schnittstellenspezifikation.
[ZUSELDAP]	A. Tauber, P. Reichstädter, Zustellverzeichnis – LDAP Schema-Beschreibung, 1.4.0
[ZUSEMSG]	Tauber A., Rössler T., Elektronische Zustellung – Message Spezifikation 1.4.0.
[ZUSEMOD]	Reichstädter P., Modell und Prozesse der elektronischen Zustellung, 1.4.0
[ZUSEMAIL]	Rössler T., Posch. R., Abholung von Zustellung über Mailprotokolle, 1.4.0
[ZUSERECH]	Tauber A., Reichstädter P., Modell und Prozesse der Zustellungs-Verrechnung, 1.4.0.
[ZUSESPEC]	A. Tauber, T. Rössler, P. Reichstädter, Elektronische Zustellung – Technische Spezifikation, 1.4.0