

## Beschreibung zur Anbindung der ERV-Schnittstelle

Dateiname: Entwicklerhandbuch.pdf  
Version: 1.3 vom 17.04.2020  
Ersteller: Christian Adorjan ([christian.adorjan@brz.gv.at](mailto:christian.adorjan@brz.gv.at))

# 1. Dokumentinformation

## 1.1. Inhaltsverzeichnis

1	Dokumentinformation	2
1.1	Inhaltsverzeichnis	2
1.2	Änderungsverlauf	2
2	Einführung	3
2.1	Code-Beispiele	3
2.2	Beschreibung	4
2.3	Adressierung im ERV	5
2.3.1	Kommunikation mit Justizapplikationen: Zustellqualität ERV	6
2.3.2	Kommunikation mit anderen Teilnehmern: Zustellqualität nonERV	7
2.3.3	Beispiele der Kommunikation	7
2.4	Funktionen	7
3	Webservice-Operationen	8
3.1	Nachricht- und Nachweis-Schnittstelle	8
3.1.1	Nachricht einbringen	8
3.1.2	Nachrichtstatus abfragen	10
3.1.3	Nachrichten abholen	11
3.1.4	Nachweis einbringen	13
3.1.5	Nachweis abholen	14
3.2	Anschritcode-Schnittstelle	15
3.2.1	Informationen zu Anschritcodes abfragen	15
3.2.2	Status aktualisieren	16
3.2.3	Datensichtbarkeit aktualisieren	17
3.2.4	Koppeln und Entkoppeln	18
4	Validierung vor Einbringung an Justizanwendungen	19
4.1	Aufruf der Validierung	20
5	Anhang	21
5.1	Beispiel 1: Zustellqualität ERV	21
5.2	Beispiel 2: Zustellqualität nonERV	21
6	Kontakt	23

## 1.2. Änderungsverlauf

Version	Datum	Ersteller	Kommentar
1.0	06.11.2015	Adorjan	erste veröffentlichte Version
1.1	19.09.2018	Adorjan	Anpassungen Layout BMVRDJ
1.2	06.05.2019	Adorjan	Überarbeitung der Beschreibungen und Verweise
1.3	17.04.2020	Adorjan	Anpassungen aufgrund von Kopplung und Datensichtbarkeit

## 2. Einführung

Dieses Handbuch stellt Entwicklern eine Anleitung für die praktische Verwendung der ERV-Webserviceschnittstelle, insbesondere für die Anbindung als Direkt-Übermittler zur Verfügung. Dabei wird nicht jede Webserviceschnittstelle einzeln beschrieben sondern - sofern sinnvoll - an manchen Stellen mehrere WS-Operationen zusammengefasst erklärt. In den nachfolgenden Abschnitten sind Code-Beispiele (zum Teil verkürzt) angeführt die eine mögliche Verwendung darstellen, der Verweis auf die vollständigen Code-Beispiele ist bei jedem Abschnitt mit angeführt. Die vollständigen Code-Beispiele sind in der ERV-Publikation enthalten und dienen als Orientierung und nicht als Vorgabe wie die Schnittstelle zu bedienen ist. Zusätzlich wird darauf hingewiesen, dass die Verwendung des Client-Zertifikats bzw. die notwendige Freischaltung im Portal für die ERV Services in den Code-Beispielen nicht beschrieben ist.

Die ERV - Publikation veröffentlicht die jeweils gültige Version der Schnittstellenbeschreibung des ERV, sowie die Definitionen aller über den ERV erreichbaren Justizapplikationen (sowie deren Validierungsmodule, sofern vorhanden). Die Publikation kann online unter <http://kundmachungen.justiz.gv.at/edikte/km/kmhlp05.nsf/all/erv!OpenDocument> erreicht werden.

### 2.1. Code-Beispiele

Die in diesem Handbuch angeführten Code-Beispiele wurden in Java (Version 1.8) implementiert. Die vollständigen Code-Beispiele sind in einem Maven-Modul-Projekt svk-erv-samples organisiert und befinden sich in der ERV-Publikation unter dem Punkt "ERV 3.0". Im Projekt-Ordner svk-erv-samples befindet sich eine README.md Datei in der das Projekt und die einzelnen Maven-Module kurz beschrieben sind. Technische Werkzeuge die für die Ausführung der Beispiele benötigt werden sind:

- Java SE Development Kit 8
- Apache Maven 3

Bei der Umsetzung der Beispiele wurde der Fokus auf den Aufruf der ERV-Schnittstelle selbst gelegt, Code der nicht in einem direkten Zusammenhang mit der jeweiligen WS-Operation (Webservice-Operation) steht wie etwa die Einbindung des Client Zertifikats, wurde weggelassen. Die Code-Beispiele sind ausführbar und geben den Request und Response der jeweiligen Operationen in der Standardausgabe (System.out) aus.

## 2.2. Beschreibung

Der ERV (elektronischer Rechtsverkehr) ermöglicht in erster Linie (aber nicht nur) die Kommunikation mit Anwendungen der österreichischen Justiz. In diesem Kontext sind das unter anderem folgende Justizanwendungen:

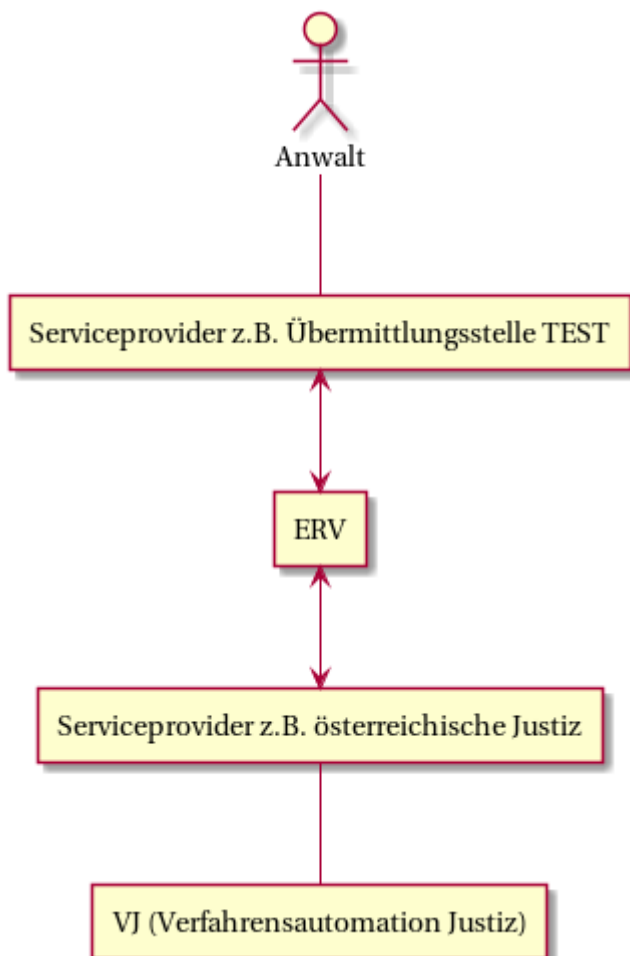
- Verfahrensautomation Justiz (VJ)
- Firmenbuch (FB)
- Grundbuch (GB)
- EliAs (Elektronisch integrierte Assistenz für die Staatsanwaltschaft)
- EUM (Europäisches Mahnverfahren)

Im Rahmen des ERV können jedoch auch Nachrichten zwischen allen anderen Teilnehmern (ohne Bezug zu Justizanwendungen) ausgetauscht werden. Der Nachrichtenaustausch erfolgt über eine HTTP-SOAP-Webserviceschnittstelle (nachfolgend nur mehr Webserviceschnittstelle genannt) die vom ERV zur Verfügung gestellt wird. Nachrichtenaustausch deswegen, weil es nicht nur möglich ist **an** die österreichische Justiz Nachrichten zu schicken, sondern auch Nachrichten von der österreichischen Justiz zu erhalten (bidirektionale Kommunikation).



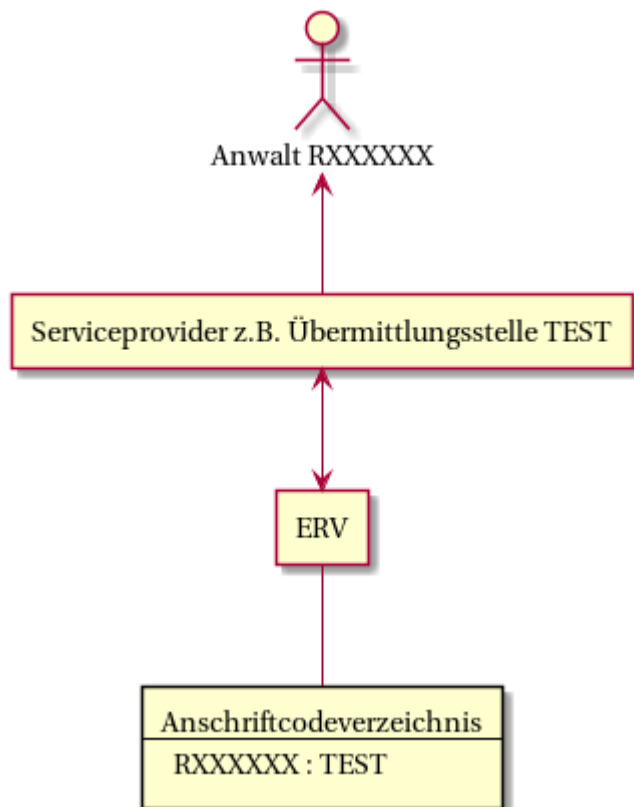
### Kommunikationswege

Die Kommunikation über den ERV ist technisch nicht nur auf die österreichische Justiz beschränkt, sondern ist mittlerweile eine zentrale Drehscheibe für einen sicheren, zuverlässigen und nachvollziehbaren Nachrichtenaustausch zwischen zwei Kommunikationspartnern. Der sichere Nachrichtenaustausch wird durch eine verschlüsselte Kommunikation und die Ausführung von Validierungen aller gesendeter Nachrichten gewährleistet.



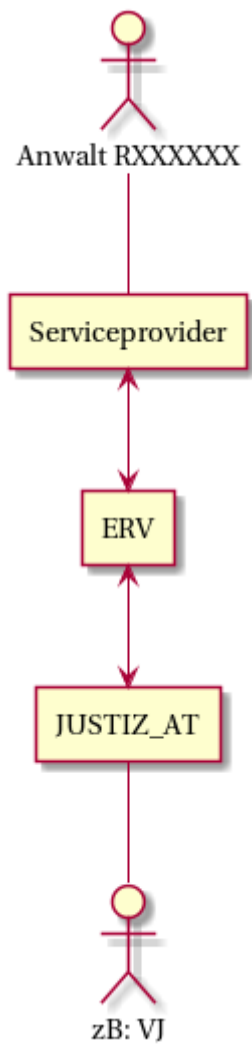
## 2.3. Adressierung im ERV

In der technischen Kommunikation kennt der ERV als Kommunikationspartner in erster Linie sogenannte Serviceprovider. Ein Serviceprovider ist eine technische Instanz dem Empfänger zugeteilt sind. Die übliche Art von Service Providern sind Übermittlungsstellen. Jede Übermittlungsstelle die an den ERV angebunden ist hat eine Menge von Teilnehmern mit eindeutigen Anschriftcodes (Empfänger der eigentlichen Nachricht). Die Anschriftcodes einer Übermittlungsstelle sind im ERV in einem zentralen Teilnehmerverzeichnis hinterlegt. Analog den Anschriftcodes einer Übermittlungsstelle sind auch die Anwendungen der österreichischen Justiz im ERV hinterlegt. Wird nun beispielsweise über die Justizanwendung Firmenbuch eine Nachricht an einen Rechtsanwalt versendet so ist dessen Anschriftcode als Empfänger in der ERV-Nachricht eingetragen. Über die Zuordnung Übermittlungstelle und Anschriftcodes im ERV (Teilnehmerverzeichnis), wird der entsprechenden Übermittlungsstelle (Serviceprovider) die Nachricht zur Abholung bereitgestellt.



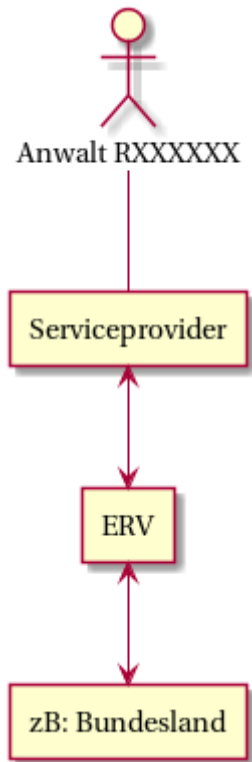
### 2.3.1. Kommunikation mit Justizapplikationen: Zustellqualität ERV

Als Kommunikation mit Justizanwendungen können alle Sendungen angesehen werden, die entweder an eine Justizanwendung gerichtet sind, oder von einer versandt werden. Da in diesem Fall rechtliche Grundlagen bestehen (GOG §89, ERV Gesetz) ist diese Zustellqualität für keine andere Kommunikation zulässig.



### 2.3.2. Kommunikation mit anderen Teilnehmern: Zustellqualität nonERV

Als Kommunikation mit anderen Teilnehmern wird generell jegliche Kommunikation über den ERV angesehen, an denen keine Justizanwendung teilnimmt (weder als Sender, noch als Empfänger). Diese Zustellqualität erlaubt es, die Einschränkungen, die für die Kommunikation mit Justizanwendungen gelten freier zu gestalten. Bei der Kommunikation mittels Zustellqualität 'nonERV' müssen die Nachweise immer vom Empfänger der Nachricht erstellt werden, falls der Sender einen Nachweis anfordert.



### 2.3.3. Beispiele der Kommunikation

Um dem weiteren Dokument nicht vorzugreifen wurden Beispiele der beiden Aufrufe im XML - Format als Anhang beigefügt.

## 2.4. Funktionen

Die Webserviceschnittstelle vom ERV bietet die nachfolgend aufgelisteten Operationen:

- NachrichtEinbringen - Eine Nachricht an einen Kommunikationspartner senden
- NachrichtStatus - Den Status einer Nachricht abfragen
- NachrichtAbholen - Eine Nachricht abholen die von einem anderen Kommunikationspartner versendet wurde
- NachrichtBestaetigen - Eine abgeholte Nachricht bestätigen
- NachrichtAnzahl - Die Anzahl der abzuholenden Nachrichten abfragen
- NachweisEinbringen - Einen Nachweis für eine zuvor abgeholte Nachricht senden
- NachweisAbholen - Einen Nachweis für eine zuvor versendete Nachricht abholen

Nachfolgendes Beispiel demonstriert die Verwendung der WS-Operationen NachrichtAnzahl

Jeder Teilnehmer (z.B. Rechtsanwalt oder Notar) der im ERV im Wege einer Übermittlungsstelle Nachrichten senden und empfangen möchte, muss zumindest einen eindeutigen Anschriftcode besitzen. Dieser wird im ERV-Teilnehmerverzeichnis hinterlegt und kann von der jeweiligen Übermittlungsstelle über eine eigene Webserviceschnittstelle verwaltet werden. Die Anschriftcode-Webserviceschnittstelle bietet die nachfolgend aufgelisteten Operationen:

- AnschriftcodeAbfragen
- StatusSetzen
- SichtbarkeitsstufeSetzen
- KopplungsstatusSetzen (koppeln und entkoppeln mit dem Anzeigemodul)

## 3. Webservice-Operationen

Im nachfolgenden werden die Aufrufe der ERV-Webserviceschnittstellen anhand von Code-Beispielen erläutert. Zur jeder Operation gibt es eine kurze Beschreibung mit Hinweis auf mögliche Fehlerquellen und Tipps die entsprechenden Fehler zu vermeiden.

### 3.1. Nachricht- und Nachweis-Schnittstelle

Die Schnittstellenbeschreibung ERVServiceExtern.wsdl enthält die Operationen für die Verarbeitung von Nachrichten und Nachweisen.

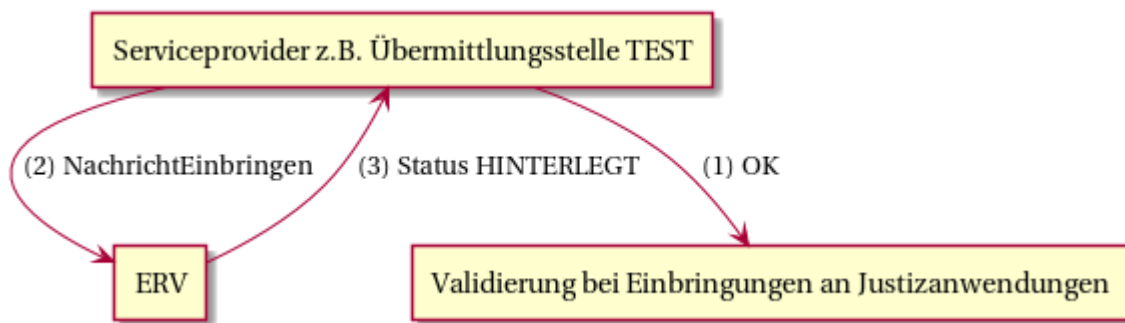
#### 3.1.1. Nachricht einbringen

WS-Operation: NachrichtEinbringen

Code-Beispiel: at.gv.brz.bjujq.erv.samples.nachricht\_nachweis.NachrichtEinbringen

Beschreibung: Diese Operation erlaubt es eine Nachricht über den ERV an einen Empfänger zu senden, in diesem Beispiel an die Justizanwendung VJ

Bei Nachrichtenversand an eine Justizanwendung muss vor der Einbringung eine Vorabvalidierung erfolgen (Punkt 1 in der Abbildung), erst danach darf die ERV-Nachricht eingebracht werden. In einem nachfolgenden Abschnitt ist die Validierung für Justizanwendungen gesondert beschrieben.



```
// Request fuer die Einbringung vorbereiten
// Die Nachricht soll mit dem Anschriftcode RXXXXXX an die Justizanwendung VJ geschickt werden
NachrichtEinbringenRequest request = SchemaUtil.OF_SERVICE.createNachrichtEinbringenRequest();
ERVNachrichtTyp ervNachricht = SchemaUtil.createErvNachricht("RXXXXXX", "VJ");
request.setERVNachricht(ervNachricht);

// ERV-Aufruf
NachrichtEinbringenResponse response = ervNachrichtPortType.nachrichtEinbringen(request);

// Fehlerbehandlung, siehe Fehlercodes in der ERV-Publikation
if (response.getFehler() != null) {

} else {
    StatusTyp status = response.getNachrichtstatus();
}
}
```

Das Element NachrichtEinbringenRequest erwartet lediglich eine ERV-Nachricht. Die Antwort auf eine Einbringung beinhaltet entweder ein Fehler-Element oder ein Status-Element welches den aktuellen Status der Nachricht im ERV anzeigt. Im Fehler-Fall sollte durch ausführliches Logging eine spätere Fehleranalyse erleichtert werden. Manche Fehler wie ERV\_0004 (ERV-ID ist nicht eindeutig) können automatisch korrigiert werden, im Fall von ERV\_0004 durch eine neu erstellte ERV-ID. Im positiven Fall wird der Status HINTERLEGT retourniert werden. Der Status HINTERLEGT gibt an, dass die Nachricht alle ERV-Regeln positiv durchlaufen hat und für die Abholung bereit steht. Sobald die ERV-Nachricht abgeholt und bestätigt wurde, wird der Status auf ZUGESTELLT gesetzt.





#### **Nachweisabholung**

Der Nachweis für eine zuvor eingebrachte Nachricht kann im Status HINTERLEGT als auch im Status ZUGESTELLT vorliegen. Daher sollte mit der WS-Operation NachrichtStatus abgefragt werden ob ein Nachweis bereits vorliegt (siehe weiter unten), diese Operation retourniert das Flag "nachweisVorhanden" das angibt ob ein Nachweis für eine zuvor eingebrachte Nachricht vorliegt und vereinfacht somit die Abholung von Nachweisen.



#### **ERV - Regeln und ERV - Fehler**

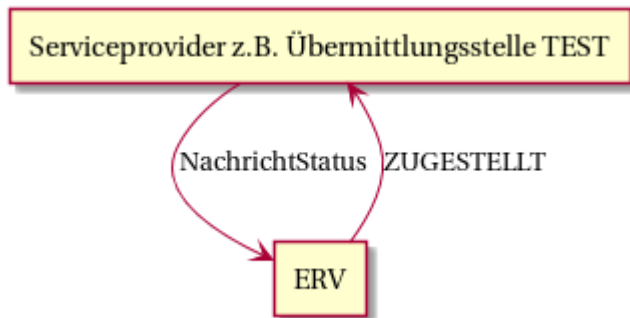
Die gesamte Liste aller ERV - Regeln und den entsprechenden Fehlermeldungen kann in den ERV - Publikationen (Siehe Einleitung) eingesehen werden.

### 3.1.2. Nachrichtstatus abfragen

WS-Operation: NachrichtStatus

Code-Beispiel: at.gv.brz.bjujq.erv.samples.nachricht\_nachweis.NachrichtStatus

Beschreibung: Diese Operation erlaubt es den aktuellen Status einer Nachricht abzufragen und damit die Nachweis-Abholung zu steuern (falls ein Nachweis erwartet wird)



```
// Request fuer den Statusaufruf vorbereiten
NachrichtStatusRequest request = SchemaUtil.OF_SERVICE.createNachrichtStatusRequest();
request.setNachrichtenkennung(SchemaUtil.OF_NR.createNachrichtenkennungTyp());
request.getNachrichtenkennung().setNachrichtId(SchemaUtil.generateRandomId());

// ERV-Aufruf
NachrichtStatusResponse response = ervNachrichtPortType.nachrichtStatus(request);

// Fehlercodes siehe ERV-Publikation
if (response.getFehler() != null) {

} else {
    StatusTyp status = response.getNachrichtstatus();

    if (status.getCode() == NachrichtStatuscode.ZUGESTELLT) {
        if (response.isNachweisVorhanden()) {
            // Nachweis abholen, falls ein Nachweis erwartet wird
        }
    }
}
}
```

Die Statusabfrage erfolgt nur mit der ERV-ID. In der Regel möchte man den Status einer zuvor eingebrachten Nachricht abfragen. Falls die Statusabfrage nicht erfolgreich aufgerufen werden kann, so wird im Response das Fehler-Element befüllt. Bei einem technischen Fehler wird eine Exception geworfen. Handelt es sich um eine Nachricht die bei der Einbringung vom ERV mit einem Fehler abgewiesen wurde, so wird der Status ABGEWIESEN geliefert, ansonsten ist der Status der Nachricht HINTERLEGT, AUSGEFOLGT oder ZUGESTELLT.

Folgende Statusinformationen existieren im ERV:

- EINGELANGT - Hierbei handelt es sich um einen internen Status
- ABGEWIESEN - Verstoß gegen ERV-Regel
- HINTERLEGT - Die ERV-Regeln wurden positiv durchlaufen und die Nachricht kann abgeholt werden
- AUSGEFOLGT - Die Nachricht wurde abgeholt jedoch noch nicht bestätigt
- ZUGESTELLT - Die Nachricht wurde abgeholt und bestätigt



#### Status-Info für die Nachweisabholung

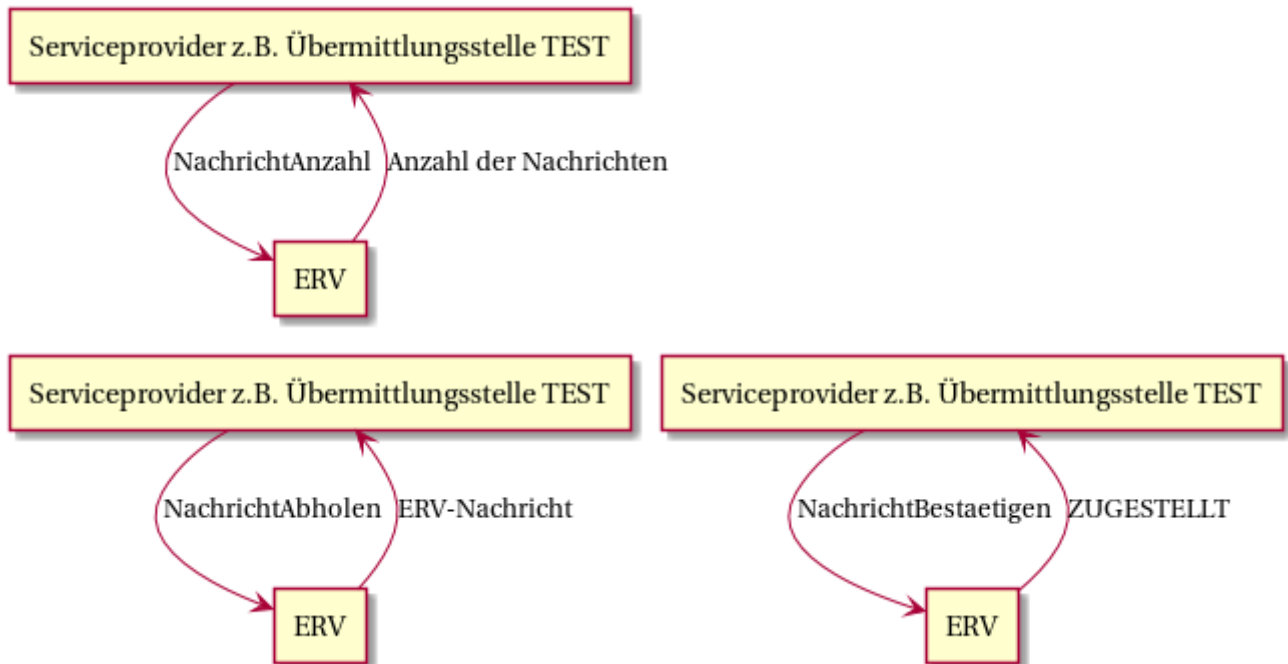
Ein Nachweis für eine zuvor eingebrachte Nachricht liegt dann vor wenn das Flag "nachweisVorhanden" auf true gesetzt ist. Das Polling-Intervall sollte immer einige Sekunden sein und nie im Millisekunden-Bereich liegen.

### 3.1.3. Nachrichten abholen

WS-Operationen: NachrichtAnzahl, NachrichtAbholen, NachrichtBestaetigen

Code-Beispiel: `at.gv.brz.bjujq.erv.samples.nachricht_nachweis.NachrichtAbholen`

Beschreibung: Bei der Abholung von Nachrichten können drei WS-Operationen kombiniert werden damit eine effiziente Abholung von Nachrichten stattfindet



Nachfolgendes Beispiel demonstriert die Verwendung der WS-Operationen NachrichtAnzahl, NachrichtAbholen und NachrichtBestaetigen.

```
// Ermittlung der Nachrichten die grundsätzlich zur Abholung bereitstehen
NachrichtAnzahlResponse nachrichtAnzahlResponse = ervNachrichtPortType.nachrichtAnzahl(SchemaUtil.
createEmptyElement());

// Abholen der Nachrichten
for (int i = 0; i < nachrichtAnzahlResponse.getAnzahl(); i++) {

    NachrichtAbholenResponse response = ervNachrichtPortType.nachrichtAbholen(SchemaUtil.
createEmptyElement());

    // Fehlercodes siehe ERV-Publikation
    if (response.getFehler() != null) {
        if ("ERV_0043".equals(response.getFehler().getFehlercode())) {
            // Keine Nachrichten mehr zum Abholen
            break;
        }
    }

    ...

    NachrichtBestaetigenResponse bestaetigenResponse = ervNachrichtPortType.nachrichtBestaetigen
(bestaetigenRequest);
}
```

Bei der Abholung sollte zuerst die Anzahl der zur Abholung bereitstehenden Nachrichten abgefragt werden, danach kann in einer Schleife die Abholung durchgeführt werden, wobei die Iterationen der Nachrichtenanzahl entsprechen. Eine vom ERV abgeholte Nachricht hat den Status AUSGEFOLGT, sobald die Nachricht bestätigt wird erhält sie den Status ZUGESTELLT. Daher sollte vor dem Bestätigen unbedingt gewährleistet sein, dass die abgeholte Nachricht auch verarbeitet werden kann. Sobald die Nachricht bestätigt wird steht sie zur Abholung nicht mehr bereit. Bleibt die Bestätigung aus so wird der Status der Nachricht wieder auf HINTERLEGT gesetzt (dies kann bis zu 2h dauern). Im Status AUSGEFOLGT kann eine Nachricht nicht wieder abgeholt werden, sondern erst wieder im Status HINTERLEGT.



#### Nachrichtenanzahl

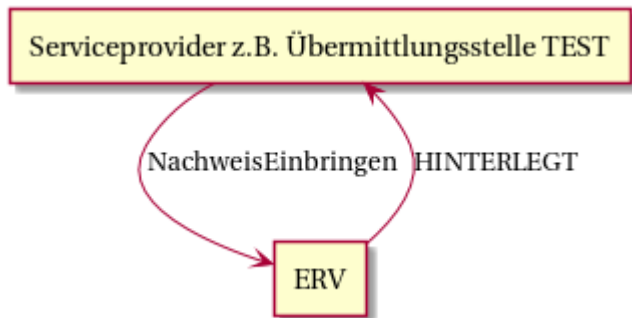
Die Nachrichtanzahl liefert die Anzahl jener Nachrichten die im Status HINTERLEGT und AUSGEFOLGT sind, dh. auch jene Nachrichten sind umfasst, die abgeholt aber noch nicht bestätigt wurden. Daher kann es sein dass die tatsächlich abzuholende Anzahl an Nachrichten kleiner ist als die retournierte Anzahl. Falls keine Nachricht für die Abholung bereit steht liefert die WS-Operation NachrichtAbholen den Fehlercode ERV\_0043 (siehe Code-Beispiel) Dieser Fehlercode kann abgefragt werden um die Schleife abzubrechen.

### 3.1.4. Nachweis einbringen

WS-Operationen: NachweisEinbringen

Code-Beispiel: at.gv.brz.bjujq.erv.samples.nachricht\_nachweis.NachweisEinbringen

Beschreibung: Diese Operation erlaubt es einen Nachweis über den ERV an einen Empfänger zu senden, in diesem Beispiel an die Justizanwendung "VJ".



```
// ERV-ID einer zuvor abgeholten und bestaetigten ERV-Nachricht, in diesem Beispiel erstellen wir der
einfachheitshalber eine beliebige ID
final String nachrichtId = SchemaUtil.generateRandomId();
ERVNachrichtTyp ervNachrichtTyp = SchemaUtil.createErvNachweisToJustizAT("RXXXXXX", "VJ", nachrichtId);
NachweisEinbringenRequest request = SchemaUtil.OF_SERVICE.createNachweisEinbringenRequest();
request.setERVNachricht(ervNachrichtTyp);

NachweisEinbringenResponse response = ervNachrichtPortType.nachweisEinbringen(request);

// Fehlercodes siehe ERV-Publikation
if (response.getFehler() != null) {

} else {
    StatusTyp statusTyp = response.getNachrichtstatus();
    // Der eingebrachte Nachweis befindet sich im Status HINTERLEGT
}
```

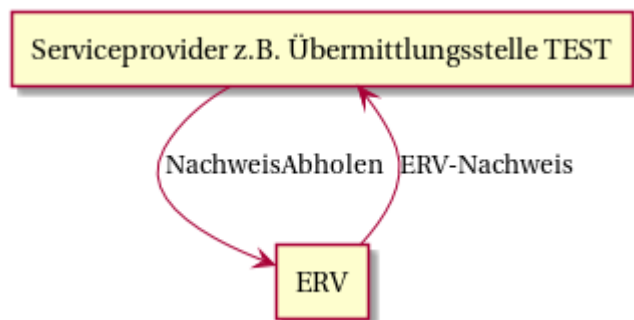
ERV-Nachweise werden als Antworten auf ERV-Nachrichten verschickt, daher beinhalten auch Nachweise die ERV-ID der referenzierten Nachricht. Die erfolgreiche Einbringung eines Nachweises in den ERV wird durch das Vorhandensein des Status-Elements in der Response gekennzeichnet. Ein Nachweis befindet sich nach Einbringung im Status HINTERLEGT, dieser Status bleibt unverändert im Gegensatz zum Status einer Nachricht der sich bei einer Abholung und Bestätigung verändert. Der ERV akzeptiert nur **einen** Nachweis für eine Nachricht, weitere Nachweise zu einer Nachricht werden mit dem Fehlercode ERV\_0035 abgelehnt. Bei der Zustellqualität 'ERV' wird der Nachweis für Direkt-Übermittler vom ERV erstellt, falls ein Nachweis vom Sender angefordert wurde. Bei der Zustellqualität 'nonERV' muss der Nachweis vom Empfänger selbst erstellt werden, falls ein Nachweis vom Sender angefordert wird. Welche Serviceprovider als Direkt-Übermittler registriert sind, geht aus dem Dokument [Services\\_Configuration.pdf](#) hervor.

### 3.1.5. Nachweis abholen

WS-Operationen: NachweisAbholen

Code-Beispiel: at.gv.brz.bjujq.erv.samples.nachricht\_nachweis.NachweisAbholen

Beschreibung: Der Nachweis einer zuvor eingebrachten Nachricht wird abgeholt



```
// ERV-ID einer zuvor eingebrachten Nachricht
final String nachrichtId = SchemaUtil.generateRandomId();
NachweisAbholenRequest request = SchemaUtil.OF_SERVICE.createNachweisAbholenRequest();
request.setNachrichtenkennung(SchemaUtil.OF_NR.createNachrichtenkennungTyp());
request.getNachrichtenkennung().setNachrichtId(nachrichtId);

NachweisAbholenResponse response = ervNachrichtPortType.nachweisAbholen(request);

// Fehlercodes siehe ERV-Publikation
if (response.getFehler() != null) {
    if ("ERV_0039".equals(response.getFehler().getFehlercode())) {
        // Kein Nachweis vorhanden, spaeter nochmal abfragen
    }
} else {
    ERVNachrichtTyp ervNachricht = response.getERVNachricht();
}
}
```

Ein Nachweis kann mit der ERV-ID einer zuvor eingebrachten Nachricht abgeholt werden. Ein Nachweis kann beliebig oft abgeholt werden, der Status eines Nachweises bleibt auf HINTERLEGT. Liegt ein Nachweis noch nicht vor, so wird der Fehlercode ERV\_0039 retourniert.

## 3.2. Anschriftcode-Schnittstelle

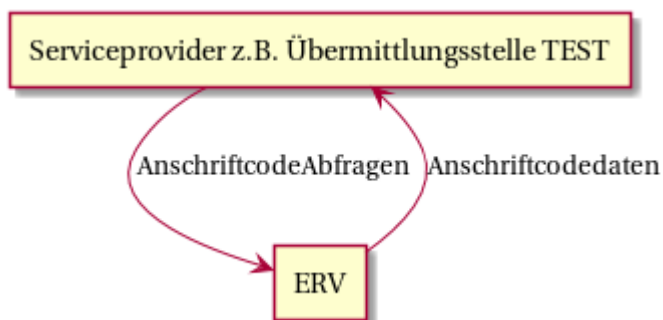
Die Schnittstellenbeschreibung ERVAnschriftcodeServiceExtern.wsdl enthält die Operationen für die Verarbeitung von Anschriftcodes. Sie ermöglicht das Aktivieren / deaktivieren von Anschriftcodes, das Übernehmen von Anschriftcodes anderer Übermittlungsstellen, sowie die Koppelung oder Entkoppelung am Anzeigemodul

### 3.2.1. Informationen zu Anschriftcodes abfragen

WS-Operationen: AnschriftcodeAbfragen

Code-Beispiel: at.gv.brz.bjujq.erv.samples.anschriftcode.AnschriftcodeAbfragen

Beschreibung: Abfragen von Status-, Datensichtbarkeit-, Kopplungs- und Übermittlerinformationen zu Anschriftencodes



```
AnschriftcodeAbfragenRequest request = SchemaUtil.OF_AN_SERVICE.createAnschriftcodeAbfragenRequest();
request.getAnschriftcode().add("R000001");
request.getAnschriftcode().add("R000002");
request.getAnschriftcode().add("R000003");

AnschriftcodeAbfragenResponse response = ervAnschriftcodePortType.anschriftcodeAbfragen(request);

if (!response.getAnschriftcodedaten().isEmpty()) {
    // Verarbeite Antwort
}
```

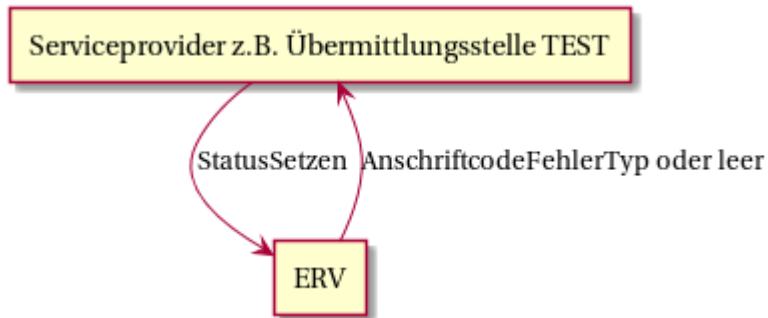
Die Abfrage von Anschriftcode-Informationen wird mit einer Liste von Anschriftcodes durchgeführt, zu jedem übergebenen Anschriftcode wird ein Element Anschriftcodedaten retourniert in dem Status-, Datensichtbarkeit-, Kopplungs- und Übermittlerinformationen enthalten sind.

### 3.2.2. Status aktualisieren

WS-Operationen: StatusSetzen

Code-Beispiel: at.gv.brz.bjujq.erv.samples.anschriftcode.StatusSetzen

Beschreibung: Daten (Status, Übermittlungsstelle, ...) zu einem Anschriftcode aktualisieren



Der Aufruf kann ebenfalls wie bei den Abfrage-Operationen mit einer Liste von Daten erfolgen. Beim StatusSetzen werden die aktuellen Anschriftcode-Informationen (Status, Übermittler,...) übergeben. Ein Anschriftcode kann in folgenden drei Zuständen sein:

E - Dieser Anschriftcode darf nur Nachrichten senden, aber keine Nachrichten empfangen

R- Dieser Anschriftcode darf Nachrichten senden **und** empfangen

N - Dieser Anschriftcode darf weder Nachrichten senden noch empfangen (temporäre) Deaktivierung des Teilnehmers

```
StatusSetzenRequest request = SchemaUtil.OF_AN_SERVICE.createStatusSetzenRequest();
AnschriftcodeStatusTyp anschriftcodeStatusTyp = SchemaUtil.OF_AN_NR.createAnschriftcodeStatusTyp();
anschriftcodeStatusTyp.setAnschriftcode("R000001");
anschriftcodeStatusTyp.setService(SchemaUtil.OF_TYP.createServiceTyp());
anschriftcodeStatusTyp.getService().setServiceProvider("TEST");
anschriftcodeStatusTyp.getService().setServiceArt(Service.UEBERMITTLUNGSSTELLE);
anschriftcodeStatusTyp.setStatus(AnschriftcodeStatus.E);
// Beauftragungszeitpunkt, im Beispiel wird die aktuelle Zeit gewählt
anschriftcodeStatusTyp.setZeitpunkt(SchemaUtil.createCurrentDateTime());
request.getAnschriftcodeStatus().add(anschriftcodeStatusTyp);

StatusSetzenResponse response = ervAnschriftcodePortType.statusSetzen(request);

// Fehlercodes siehe ERV-Publikation
if (response.getFehler() != null) {

} else {
    for (AnschriftcodeFehlerTyp anschriftcodeFehlerTyp : response.getAnschriftcodeFehler()) {
        // Fehlerbehandlung pro Anschriftcode
    }
}
```

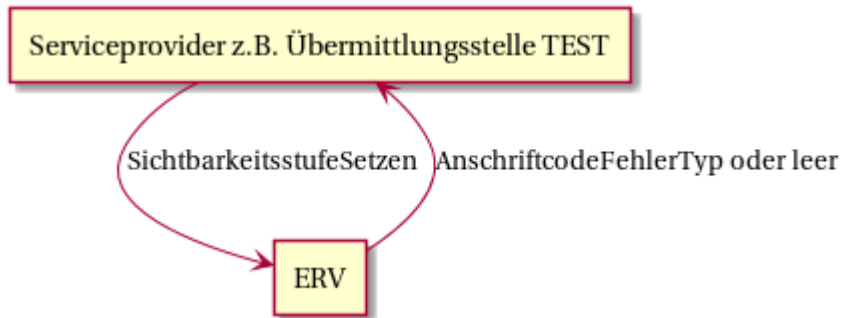


### 3.2.3. Datensichtbarkeit aktualisieren

WS-Operationen: SichtbarkeitsstufeSetzen

Code-Beispiel: at.gv.brz.bjujq.erv.samples.anschriftcode.SichtbarkeitsstufeSetzen

Beschreibung: Sichtbarkeitsstufe der Anschriftcodedaten für Aufrufer setzen



Steuert die Sichtbarkeit von Daten (Name, Ordnungsbegriff, Kontaktdaten,...) für Aufrufer.

```
SichtbarkeitsstufeSetzenRequest request = SchemaUtil.OF_AN_SERVICE.createSichtbarkeitsstufeSetzenRequest();
AnschriftcodeSichtbarkeitTyp anschriftcodeSichtbarkeitTyp = SchemaUtil.OF_AN_NR.
createAnschriftcodeSichtbarkeitTyp();
anschriftcodeSichtbarkeitTyp.setAnschriftcode("R000001");
anschriftcodeSichtbarkeitTyp.setSichtbarkeitsstufe(Sichtbarkeitsstufe.LEVEL_2);
request.getAnschriftcodeSichtbarkeit().add(anschriftcodeSichtbarkeitTyp);

// Fehlercodes siehe ERV-Publikation
SichtbarkeitsstufeSetzenResponse response = ervAnschriftcodePortType.sichtbarkeitsstufeSetzen(request);
if (response.getFehler() != null) {

} else {
    for (AnschriftcodeFehlerTyp anschriftcodeFehlerTyp : response.getAnschriftcodeFehler()) {
        // Fehlerbehandlung pro Anschriftcode
    }
}
```

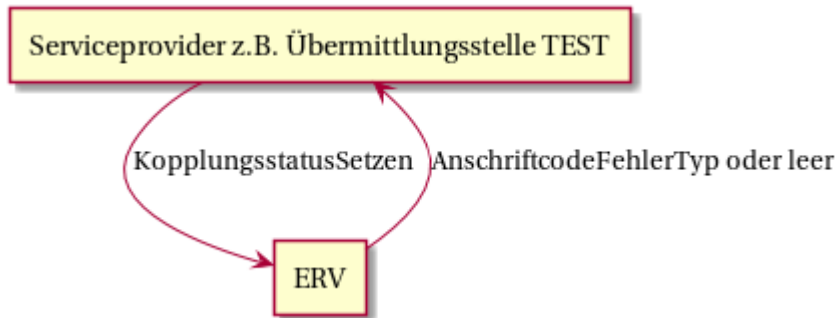
Dieser Beispiel-Aufruf setzt die Sichtbarkeitsstufe vom Anschriftcode auf LEVEL\_2 (siehe ERV-Dokumentation).

### 3.2.4. Koppeln und Entkoppeln

WS-Operationen: KopplungsstatusSetzen

Code-Beispiel: at.gv.brz.bjujq.erv.samples.anschriftcode.KopplungsstatusSetzen

Beschreibung: Koppeln und Entkoppeln am Anzeigemodul



Der Aufruf kann ebenfalls wie bei den Abfrage-Operationen mit einer Liste von Daten erfolgen. Beim KopplungsstatusSetzen wird der Anschriftcode und der Kopplungsstatus übergeben.

```
KopplungsstatusSetzenRequest request = SchemaUtil.OF_AN_SERVICE.createKopplungsstatusSetzenRequest();
AnschriftcodeKopplungTyp anschriftcodeKopplungTyp = SchemaUtil.OF_AN_NR.createAnschriftcodeKopplungTyp();
anschriftcodeKopplungTyp.setAnschriftcode("R000001");
anschriftcodeKopplungTyp.setKopplungsStatus(KopplungsStatus.K);
request.getAnschriftcodeKopplung().add(anschriftcodeKopplungTyp);

KopplungsstatusSetzenResponse response = ervAnschriftcodePortType.kopplungsstatusSetzen(request);

// Fehlercodes siehe ERV-Publikation
if (response.getFehler() != null) {

} else {
    for (AnschriftcodeFehlerTyp anschriftcodeFehlerTyp : response.getAnschriftcodeFehler()) {
        // Fehlerbehandlung pro Anschriftcode
    }
}
```

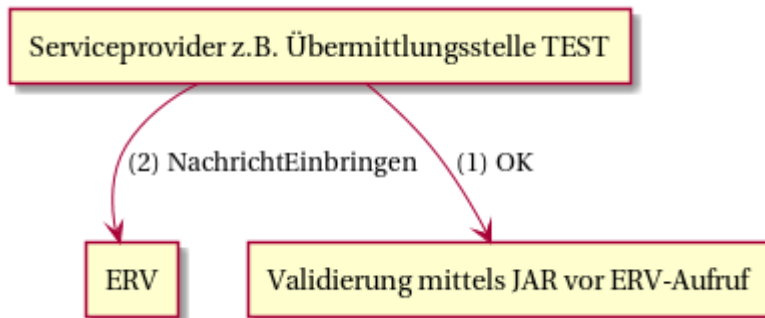
Dieser Beispiel-Aufruf koppelt den Anschriftcode am Anzeigemodul und ermöglicht so das Empfangen von elektronischen Zustellungen (verschickt an das elektronische Postfach bei einem Zustelldienst) über den ERV.

Im Gutfall wird ein leerer Response retourniert. Im Fehlerfall wird bei anschriftcodebezogenen Fehlern ein AnschriftcodeFehlerTyp-Element retourniert, ansonsten ein Fehler-Element.

## 4. Validierung vor Einbringung an Justizanwendungen

Nachrichten die an Justizanwendungen verschickt werden sollen, müssen vor der Einbringung in den ERV vorab validiert werden. Alle Justizanwendungen stellen für diesen Fall eine entsprechende Validierungsbibliothek - in der Regel als JAR (Java Archive) samt Dokumentation - zur Verfügung.

Im nachfolgenden Bild validiert die Übermittlungsstelle TEST die Nachricht mit der Validierungsbibliothek der Justizanwendung an den die Nachricht verschickt werden soll, erst nach erfolgreicher Vorabvalidierung erfolgt der Aufruf der WS-Operation NachrichtEinbringen.



Die Bibliotheken und Dokumentationen der Justizanwendungen können auf den nachfolgenden Seiten heruntergeladen werden:

Vorabinformationen der Justiz: <http://kundmachungen.justiz.gv.at/edikte/km/kmhlp05.nsf/all/voraberv!OpenDocument>

Kundmachungen der Justiz: <http://kundmachungen.justiz.gv.at/edikte/km/kmhlp05.nsf/all/erv!OpenDocument>

Da jede Justizanwendung eine eigene Dokumentation zur Verfügung stellt wird hier nur der wesentliche Teil des Aufrufes der Validierungsbibliothek erläutert, Details zu den Klassen bzw. zur Validierung muss aus der entsprechenden Dokumentation entnommen werden.

## 4.1. Aufruf der Validierung

Der weiter unten beschriebene Aufruf der Validierung gestaltet sich für die nachfolgenden Justizanwendung ähnlich:

- VJ (Verfahrensautomation Justiz)
- FB (Firmenbuch)
- GT (Grundbuch)
- EliAs (Elektronisch integrierte Assistenz für die Staatsanwaltschaft)

Wesentlich sind die Schnittstellen Message, Part und die statische Methode validate der Klasse ErsteValidierung. Die Implementierung der Schnittstelle Message enthält ERV Informationen wie z.B. den Einbringungszeitpunkt und die ERV-ID sowie die Liste der Anhänge den sogenannten Parts, hierfür muss die Schnittstelle Part implementiert werden. In der Liste der Parts muss sich auch die fachliche Payload (XML-Dokument) befinden. Sobald ein Message-Objekt erzeugt wurde kann die statische Methode ErsteValidierung.validate() aufgerufen werden. Diese liefert ein Objekt vom Typ ValidationResult zurück. Der nachfolgende Code zeigt einen beispielhaften Aufruf der VJ-Validierung.

```
// Das Objekt message beinhaltet in der Liste der Part-Objekte unter anderem die fachliche Payload
Message message = ...

// Aufruf der Validierung
ValidationResult validationResult = ErsteValidierung.validate(message);
if (validationResult.isOk()) {
    // Validierung OK
    ...
} else {
    // Validierung negativ, das Objekt validationResult enthält die Validierungsfehler, siehe
    anwendungsspezifische Dokumentation
    ...
}
```

Weitere Informationen sind aus der anwendungsspezifischen Dokumentation des jeweiligen Validierungsmoduls zu entnehmen.

### Validierung für EUM (Europäisches Mahnverfahren)

Die Verwendung der EUM-Validierung ist in dem Dokument EUM\_Validierungsmodul\_Spezifikation.pdf anhand von Code-Beispielen beschrieben, daher wird an dieser Stelle auf die EUM-Publikation verwiesen.

## 5. Anhang

### 5.1. Beispiel 1: Zustellqualität ERV

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:NachrichtEinbringenRequest xmlns="http://erv.justiz.gv.at/ERNachrichtExtern/V3_0" xmlns:ns2="
http://erv.justiz.gv.at/ERVTypenExtern/V3_0" xmlns:ns3="http://erv.justiz.gv.at/ERNachrichtServiceExtern
/V3_0">
  <ERNachricht>
    <NachrichtId>mid://20180903.c8503728-72fa-4b68-9e95-59768a9d72e5@test.at</NachrichtId>
    <Erstellzeitpunkt>2018-09-03T09:33:21.055+02:00</Erstellzeitpunkt>
    <Empfaenger>
      <Teilnehmer>
        <ns2:TeilnehmerKennung>
          <ns2:Justizanwendung>VJ</ns2:Justizanwendung>
        </ns2:TeilnehmerKennung>
      </Teilnehmer>
      <ServiceArt>JUSTIZ_AT</ServiceArt>
    </Empfaenger>
    <Sender>
      <Teilnehmer>
        <ns2:TeilnehmerKennung>
          <ns2:Anschriftcode>RXXXXXX</ns2:Anschriftcode>
        </ns2:TeilnehmerKennung>
      </Teilnehmer>
      <Service>
        <ns2:ServiceArt>UEBERMITTLUNGSSTELLE</ns2:ServiceArt>
        <ns2:ServiceProvider>TEST</ns2:ServiceProvider>
      </Service>
    </Sender>
    <Zustellqualitaet>ERV</Zustellqualitaet>
    <Nachricht>
      <Einbringungszeitpunkt>2018-09-03T09:33:21.056+02:00</Einbringungszeitpunkt>
      <Nachweis>true</Nachweis>
    </Nachricht>
    <Anhang>
      <ns2:Art>DOKUMENT</ns2:Art>
      <ns2:AnhangId>1</ns2:AnhangId>
      <ns2:Reihung>1</ns2:Reihung>
      <!-- Bei MTOM-Nachrichten steht hier eine Referenz auf den jeweiligen Mimepart -->
      <ns2:Content>UERGIEluaGFsdA==</ns2:Content>
      <ns2:ContentType>application/pdf</ns2:ContentType>
      <ns2:Metadaten>TWV0YWRhdGVuIGRlcyBQREYtRG9rdW11bnRz</ns2:Metadaten>
      <ns2:Hash>
        <ns2:HashAlgorithmus>MD5</ns2:HashAlgorithmus>
        <ns2:HashWert>43694a2af135a6bede98b18eb6efa93b</ns2:HashWert>
      </ns2:Hash>
    </Anhang>
    <Anhang>
      <ns2:Art>DOKUMENT</ns2:Art>
      <ns2:AnhangId>2</ns2:AnhangId>
      <ns2:Reihung>2</ns2:Reihung>
      <!-- Bei MTOM-Nachrichten steht hier eine Referenz auf den jeweiligen Mimepart -->
      <ns2:Content>UERGIEluaGFsdA==</ns2:Content>
      <ns2:ContentType>application/pdf</ns2:ContentType>
      <ns2:Metadaten>TWV0YWRhdGVuIGRlcyBQREYtRG9rdW11bnRz</ns2:Metadaten>
      <ns2:Hash>
        <ns2:HashAlgorithmus>MD5</ns2:HashAlgorithmus>
        <ns2:HashWert>43694a2af135a6bede98b18eb6efa93b</ns2:HashWert>
      </ns2:Hash>
    </Anhang>
  </ERNachricht>
</ns3:NachrichtEinbringenRequest>
```

### 5.2. Beispiel 2: Zustellqualität nonERV

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:NachrichtEinbringenRequest xmlns="http://erv.justiz.gv.at/ERNachrichtExtern/V3_0" xmlns:ns2="
http://erv.justiz.gv.at/ERVTypenExtern/V3_0" xmlns:ns3="http://erv.justiz.gv.at/ERNachrichtServiceExtern
/V3_0">
  <ERNachricht>
    <NachrichtId>mid://20180903.69b2d902-b8ae-4cca-b6af-990213f99e5a@test.at</NachrichtId>
    <Erstellzeitpunkt>2018-09-03T09:35:46.632+02:00</Erstellzeitpunkt>
    <!-- Falls Empfaenger keine Justizanwendung ist muss die Zustellqualitaet nonERV gewaehlt werden --
  >

    <Empfaenger>
      <Teilnehmer>
        <ns2:TeilnehmerKennung>
          <ns2:Anschriftcode>RYYYYYY</ns2:Anschriftcode>
        </ns2:TeilnehmerKennung>
      </Teilnehmer>
      <ServiceArt>UEBERMITTLUNGSSTELLE</ServiceArt>
    </Empfaenger>
    <Sender>
      <Teilnehmer>
        <ns2:TeilnehmerKennung>
          <ns2:Anschriftcode>RXXXXXX</ns2:Anschriftcode>
        </ns2:TeilnehmerKennung>
      </Teilnehmer>
      <Service>
        <ns2:ServiceArt>UEBERMITTLUNGSSTELLE</ns2:ServiceArt>
        <ns2:ServiceProvider>TEST</ns2:ServiceProvider>
      </Service>
    </Sender>
    <!-- Bein nonERV darf nicht an die Justiz adressiert werden -->
    <Zustellqualitaet>nonERV</Zustellqualitaet>
    <Nachricht>
      <Einbringungszeitpunkt>2018-09-03T09:35:46.632+02:00</Einbringungszeitpunkt>
      <Nachweis>true</Nachweis>
    </Nachricht>
    <Anhang>
      <ns2:Art>DOKUMENT</ns2:Art>
      <ns2:AnhangId>1</ns2:AnhangId>
      <ns2:Reihung>1</ns2:Reihung>
      <!-- Bei MTOM-Nachrichten steht hier eine Referenz auf den jeweiligen Mimepart -->
      <ns2:Content>UERGIEluaGFsdA==</ns2:Content>
      <ns2:ContentType>application/pdf</ns2:ContentType>
      <ns2:Metadaten>TWV0YWRhdGVuIGRlcyBQREYtRG9rdWllbnRz</ns2:Metadaten>
      <ns2:Hash>
        <ns2:HashAlgorithmus>MD5</ns2:HashAlgorithmus>
        <ns2:HashWert>43694a2af135a6bede98b18eb6efa93b</ns2:HashWert>
      </ns2:Hash>
    </Anhang>
    <Anhang>
      <ns2:Art>DOKUMENT</ns2:Art>
      <ns2:AnhangId>2</ns2:AnhangId>
      <ns2:Reihung>2</ns2:Reihung>
      <!-- Bei MTOM-Nachrichten steht hier eine Referenz auf den jeweiligen Mimepart -->
      <ns2:Content>UERGIEluaGFsdA==</ns2:Content>
      <ns2:ContentType>application/pdf</ns2:ContentType>
      <ns2:Metadaten>TWV0YWRhdGVuIGRlcyBQREYtRG9rdWllbnRz</ns2:Metadaten>
      <ns2:Hash>
        <ns2:HashAlgorithmus>MD5</ns2:HashAlgorithmus>
        <ns2:HashWert>43694a2af135a6bede98b18eb6efa93b</ns2:HashWert>
      </ns2:Hash>
    </Anhang>
  </ERNachricht>
</ns3:NachrichtEinbringenRequest>

```

## 6. Kontakt

Email-Adresse bei technischen Fragen: [betrieb-services.erv@brz.gv.at](mailto:betrieb-services.erv@brz.gv.at)

Email-Adresse bei nicht technischen Fragen: [christian.adorjan@brz.gv.at](mailto:christian.adorjan@brz.gv.at)