



Anhang 1 der Verordnung des Eidgenössischen Justiz- und Polizeidepartements (EJPD) über die Erstellung elektronischer öffentlicher Urkunden und elektronischer Beglaubigungen (EÖBV-EJPD; SR 211.435.11)

Technische Anforderungen an Schnittstellen, über die dem UPRReg Daten über Urkundspersonen aus anderen Systemen geliefert werden

Version: 2

Inkrafttreten: 15.03.2022

Inhaltsverzeichnis

1.	Allgemeines	4
1.1.	Grundlage.....	4
1.2.	Gegenstand.....	4
1.3.	Geltungsbereich	4
1.4.	Gegenstand der Prüfung im Rahmen des Bewilligungsverfahrens.....	4
1.5.	Übersicht über den Ablauf der Datenlieferung	4
1.6.	Referenzen	4
2.	Verantwortung für die Qualität der gelieferten Daten	5
3.	Schnittstelle	5
4.	Datenstruktur des Exports	6
4.1.	Logische Dateneinheit	6
4.2.	Zuordnung der Register zu Kantonen	7
4.3.	Datenstruktur	8
4.4.	Identifikatoren	9
4.5.	Person	9
4.6.	Organisation	9
4.7.	Funktionstypen	10
4.8.	Funktion.....	10
4.8.1.	Verwendungs- und Gültigkeitszeiträume	11
4.9.	Zertifikate.....	12
4.9.1.	Qualifizierte Zertifikate sind persönlich	12
4.9.2.	Verwendungszeiträume von Zertifikaten.....	13
5.	Zu verwendende Typen und globale Einstellungen.....	14
5.1.	Identifikation des Registers.....	14
5.2.	Startdatum für elektronische Ausfertigungen und elektronische Beglaubigungen ..	15
5.3.	Encoding	15
6.	Fachliche Antwort	15
6.1.	Datenstruktur	16
6.2.	Fehlercodes.....	16
7.	Sedex Transport	17
7.1.	Meldungsfluss.....	17
7.2.	Konventionen.....	18
7.2.1.	Konvention für Übermittlung des Exports.....	18
7.2.2.	Konvention für die fachliche Antwort.....	18
8.	Zertifikate für die Signatur der Daten	19
8.1.	Struktur der Certificate Authority	19
9.	Glossar	20
10.	Version 2.0 der Struktur für den Datenaustausch mittels XML (XSD)	21

Abbildungsverzeichnis

Abbildung 1: Datenkapselung innerhalb einer Domäne.....	7
Abbildung 2: Schema für Export.....	8
Abbildung 3: Schema der Person.....	9
Abbildung 4: Schema der Organisation	10
Abbildung 5: Schema des Funktionstyps	10
Abbildung 6: Schema der Funktion	11
Abbildung 7: Datenstruktur für Zertifikat.....	12
Abbildung 8: Zuordnung von Zertifikaten zu Personen	13
Abbildung 9: Verwendungszeiträume von Zertifikaten.....	14
Abbildung 10: Datenstruktur der Antwort	16
Abbildung 11: Meldungsfluss	17

1. Allgemeines

1.1. Grundlage

Die vorliegenden Vorschriften bilden Anhang 1 der EÖBV-EJPD [3]. Sie stützen sich auf Artikel 8 Absatz 4 und Artikel 20 Absatz 1 Buchstabe a und Absatz 2 EÖBV [2] sowie auf Artikel 2 und 3 EÖBV-EJPD [3].

1.2. Gegenstand

Das Schweizerische Register der Urkundspersonen (UPReg) führt zwecks Ausgabe von Zulassungsbestätigungen die dazu erforderlichen Daten der Urkundspersonen (Art. 2 Bst. b, Art. 5 Abs. 1 und Art. 7 Abs. 1 EÖBV [2]). Nach Artikel 8 Absatz 4 EÖBV [2] dürfen die Daten über die Urkundspersonen über eine Eingabemaske des UPReg eingetragen oder, mit Bewilligung des BJ, dem UPReg über eine Schnittstelle aus anderen Systemen geliefert werden.

Das vorliegende Dokument führt die technischen Anforderungen auf, die zur Erteilung der Bewilligung nach Artikel 20 Absatz 1 Buchstabe a EÖBV [2] erfüllt sein müssen. Es spezifiziert insbesondere die Anforderungen an die Schnittstelle für den Export der Daten.

1.3. Geltungsbereich

Die vorliegenden Vorschriften richten sich an die zuständigen Behörden der Kantone, die Daten über die Urkundspersonen aus eigenen Systemen über eine Schnittstelle (*Interface*) an das UPReg zu liefern beabsichtigen. Im Folgenden wird zur Bezeichnung der zuständigen Behörde des Kantons und des Bundes der Einfachheit halber allgemein von «Gesuchstellerin» die Rede sein, zur Bezeichnung des jeweils eigenen Systems der Gesuchstellerin vom «liefernden Register». In den Beispielen wird davon abweichend jedoch auch die Bezeichnung «Kanton» und «kantonales Register» verwendet.

1.4. Gegenstand der Prüfung im Rahmen des Bewilligungsverfahrens

Das Gesuch um Bewilligung, Daten zu liefern, muss den Nachweis enthalten, dass die technischen Anforderungen erfüllt sind (Art. 3 EÖBV-EJPD [3]). Das Bundesamt für Justiz BJ prüft das Vorliegen der technischen Anforderungen theoretisch und praktisch. Neben einer Beschreibung verlangt das BJ von der Gesuchstellerin einen Funktionstest, der aufzeigt, dass rechtsverbindliche Daten korrekt an UPReg übermittelt werden, und dass gestützt darauf gültige Zulassungsbestätigungen ausgestellt werden können (End-to-End-Tests).

1.5. Übersicht über den Ablauf der Datenlieferung

Die Daten aus dem liefernden Register werden als Gesamtexport in der Auszeichnungssprache XML über die Plattform sedex übermittelt und von UPReg als Ganzes unverändert übernommen. Der erfolgreiche Import der Daten ans UPReg wird dem liefernden Register durch eine fachliche Antwort elektronisch quittiert.

1.6. Referenzen

[1] Schweizerisches Zivilgesetzbuch vom 10. Dezember 1907 (ZGB; SR 210)

- [2] Verordnung über die Erstellung elektronischer öffentlicher Urkunden und elektronischer Beglaubigungen (EÖBV; SR 211.435.1)
- [3] Verordnung des EJPD über die Erstellung elektronischer öffentlicher Urkunden und elektronischer Beglaubigungen (EÖBV-EJPD; SR 211.435.11)

2. Verantwortung für die Qualität der gelieferten Daten

Die Verantwortung für die Daten, die Datenführung und für die Datenlieferung richtet sich nach Artikel 8 EÖBV [2]. Das UPReg übernimmt die von der Gesuchstellerin gelieferten Daten unverändert. Vor der tatsächlichen Übernahme der Daten ins UPReg wird der jeweilige Datensatz lediglich auf formale Korrektheit hin überprüft, namentlich auf die Vollständigkeit der Angaben nach Artikel 7 EÖBV [2] sowie auf das Erfüllen der Anforderungen nach dem vorliegenden Anhang.

3. Schnittstelle

Die Daten werden als signiertes XML-Dokument über Sedex angeliefert. Das Dokument muss dem XML-Schema <https://www.upreg.ch/export/2/export-2-0.xsd> entsprechen (vgl. Kapitel 10). Eine Datenlieferung besteht immer aus dem vom liefernden Register signierten Komplettbestand eines kantonalen Registers für eine Domäne. Ein allenfalls bereits bestehender Datenbestand des liefernden Registers im UPReg wird durch den jeweils aktuellsten Datenbestand bzw. durch den aktuellsten Import vollständig überschrieben. Die Historisierung ist Sache der Gesuchstellerin. Der jeweils zuletzt importierte und damit aktuellste Datenbestand dient ab dem Tag nach der Lieferung als Grundlage zur Generierung von Zulassungsbestätigungen anlässlich des Abrufs und der Ausgabe.

Ablauf des Imports der Daten des kantonalen Registers bei UPReg:

1. Überprüfung, ob der angelieferte Export dem XML-Schema entspricht.
Im Fehlerfall: Abbruch und Versand einer negativen fachlichen Antwort an das liefernde Register.
2. Überprüfung, ob die Signatur des angelieferten Exports gültig ist. Die Signatur ist dann gültig, wenn sie formal gültig ist und das verwendete Zertifikat in UPReg für das liefernde Register bekannt ist.
Im Fehlerfall: Abbruch und Versand einer negativen fachlichen Antwort an das liefernde Register.
3. Überprüfung der fachlichen Korrektheit der Daten (Verwendungs- und Gültigkeitszeiträume, eindeutige Zuordnung der Zertifikate zu Personen).
4. Import der Lieferung in UPReg.
 - a. Im Fehlerfall (interner Fehler): Abbruch und Versand einer negativen fachlichen Antwort an das liefernde Register.
 - b. Im Erfolgsfall: Versand einer positiven fachlichen Antwort an das liefernde Register.

Bei einem allfälligen Abbruch bleibt der jeweils letzte erfolgreich importierte Datenbestand gültig und bildet damit die massgebliche Grundlage für die Ausgabe der abgerufenen Zulassungsbestätigungen.

4. Datenstruktur des Exports

Die Daten werden immer als voller Export des kantonalen Registers übermittelt. Dieser muss in sich konsistent sein. Allfällige Überschneidungen oder Zusammenhänge mit den Datenlieferungen anderer kantonalen Register (eine Person ist in mehreren Kanonen registriert) oder anderer Domänen (eine Person ist in mehreren Domänen registriert) werden von UPReg beim Import nicht aufgelöst.

4.1. Logische Dateneinheit

Die Daten sind in einer Sternstruktur um die Funktionen gelegt. Eine Funktion ist immer genau einer Person und einer Organisation zugeordnet. Ausserdem können einer Funktion beliebig viele Zertifikate zugeordnet werden.

Die von einem Kanton eingelierten Datensätze für eine Domäne werden von UPReg beim Import nicht mit Datensätzen eines anderen Kantons oder einer anderen Domäne verknüpft. Dadurch ist es möglich, dass die gleichen Personen, Organisationen und Zertifikate mehrfach erfasst werden.

Zertifikate werden falls nötig mehrmals eingebettet. Sie sind Teil der Funktion. Dubletten werden immer mehrfach geführt.

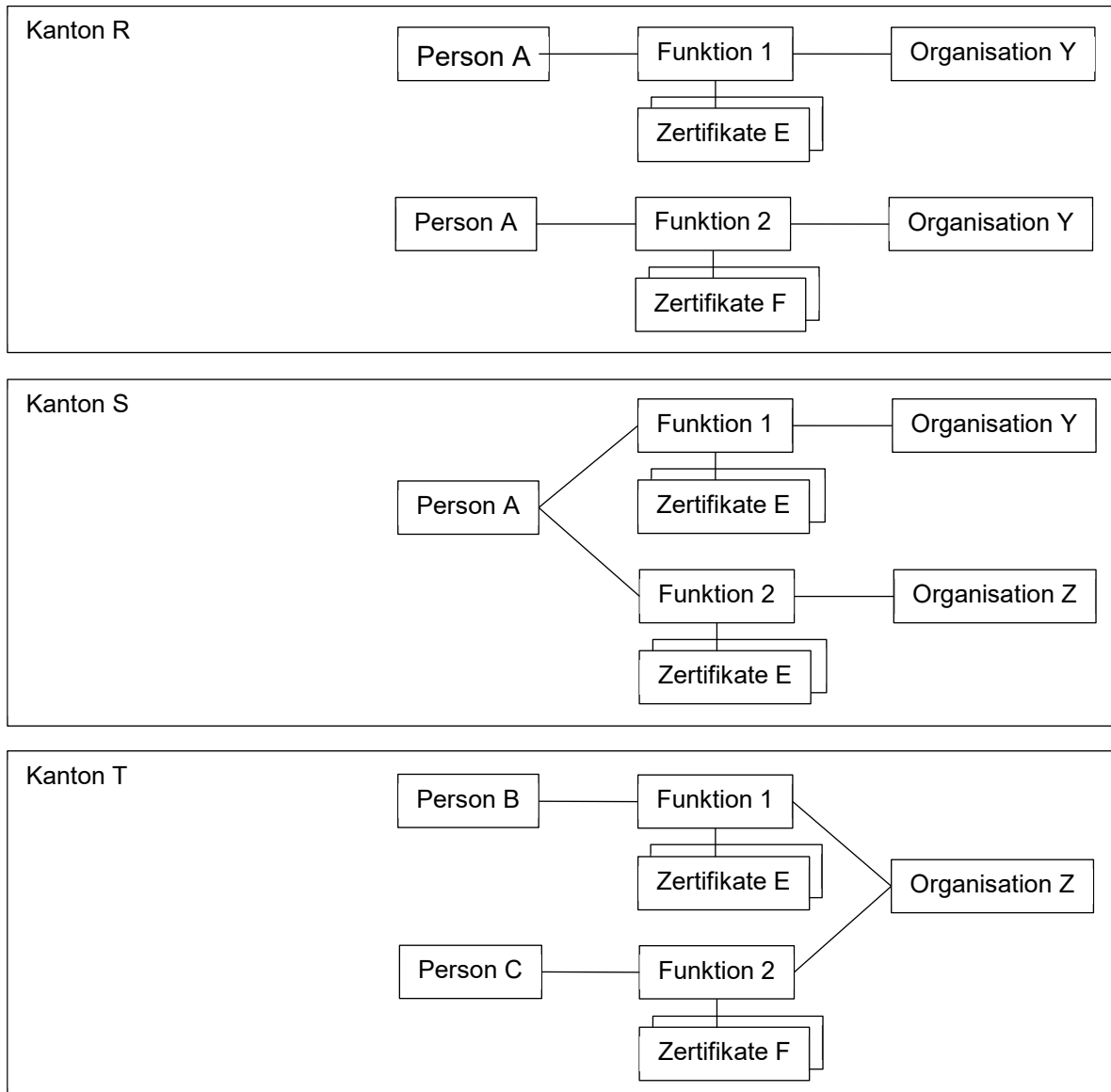


Abbildung 1: Datenkapselung innerhalb einer Domäne

Personen und Organisationen müssen nicht unbedingt zusammengelegt werden. Speichert ein kantonales Register die Personen und/oder Organisationen pro Funktion einzeln, sind diese auch innerhalb des kantonalen Datenbestandes in UPReg mehrfach hinterlegt.

Ein bestimmtes Zertifikat darf zwar mehreren Funktionen (derselben Person), jedoch nicht mehreren Personen zugeordnet werden.

4.2. Zuordnung der Register zu Kantonen

Ein Export gilt immer für ein Register, welches genau einem Kanton und einer Domäne zugeordnet ist.

Im Datenstamm von UPReg wird in der Liste der Kantone die Zuordnung der zuständigen Register innerhalb einer Domäne geführt. Ein Register muss in UPReg vorgängig dem Kanton und der Domäne zugeordnet werden, bevor Datenlieferungen importiert werden können.

4.3. Datenstruktur

Ein Export hat die folgende Struktur:

- Metadaten zum Export
 - Um welches Register handelt es sich (Kanton, Domäne)
 - Exportdatum
 - Applikationsspezifische Referenz: diese wird vom kantonalen Register im Export mitgegeben und wird in der fachlichen Antwort des UPReg zurückgeliefert und dient dem kantonalen Register zur Zuordnung der Antwort zum Export (oder für andere applikationsspezifische Zwecke)
- Personen (persons)
- Organisationen (organisations)
- Funktionen (functions)
 - Zertifikate sind Teil der Funktion
- Funktionstypen (functionTypes)
- Signatur des Registers

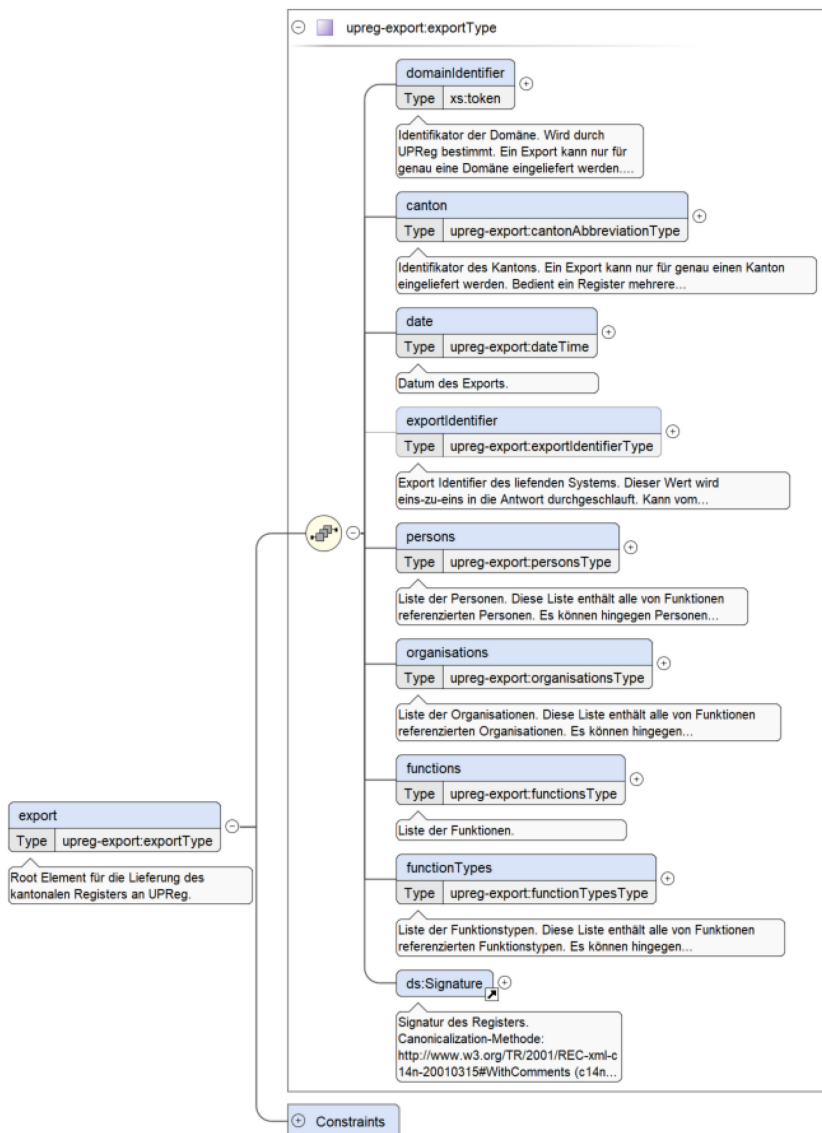


Abbildung 2: Schema für Export

4.4. Identifikatoren

Die Funktion wird einem Funktionstyp, einer Person und einer Organisation über deren Identifikatoren (Attribut *id*) verknüpft. Kann eine solche Referenz nicht aufgelöst werden, wird der ganze Export nicht in den Datenstamm übernommen. Hingegen können Funktionstypen, Organisationen und Personen ohne Funktionen existieren.

Die Attribute *id* der Elemente der Typen *functionType*, *organisationType* und *personType* müssen in deren jeweiligen Liste eindeutig sein. Die Identifikatoren der kantonalen Register werden 1:1 in UPRReg übernommen.

4.5. Person

Für eine Person müssen nur die Daten geliefert werden, die entweder für die Ausstellung der Zulassungsbestätigung erforderlich sind (vgl. Art. 7 und 10 EÖBV) oder vom Webservice Suche zurückgeliefert werden.

Das Attribut *id* wird als Identifikator für die Referenz in der Funktion verwendet. Der Identifikator muss innerhalb der Menge der Personen pro Kanton und Domäne eindeutig sein.

Zusätzlich zu den öffentlichen Informationen, kann eine Telefonnummer und eine E-Mailadresse angegeben werden. Diese dienen nur zur Kontaktaufnahme mit der betreffenden Person im Falle von Problemen bei der Ausstellung der Zulassungsbestätigung. Diese Telefonnummer und E-Mailadresse werden von UPRReg nie veröffentlicht.

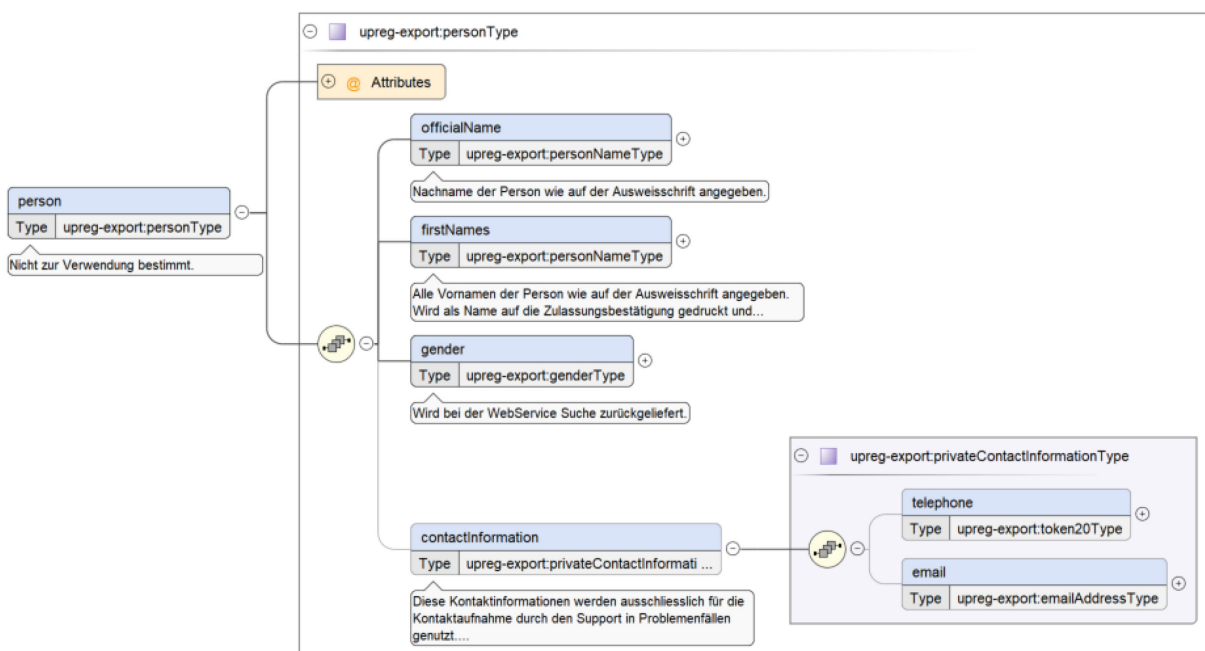


Abbildung 3: Schema der Person

4.6. Organisation

In der Datenstruktur zur Organisation müssen die UID-Nummer und der Name der Organisation geliefert werden. Wird die fakultative E-Mailadresse nicht geliefert, wird die zugehörige Funktion vom Webservice Suche nicht gefunden.

Das Attribut *id* wird auch hier als Identifikator für die Referenz in der Funktion verwendet. Der Identifikator muss innerhalb der Menge der Organisationen pro Kanton und Domäne eindeutig sein.

Bei der Organisation handelt es sich bei Notaren um die persönliche UID des Notars und nicht um die einer Kanzlei oder ähnliches, auch wenn ein Anstellungsverhältnis besteht.

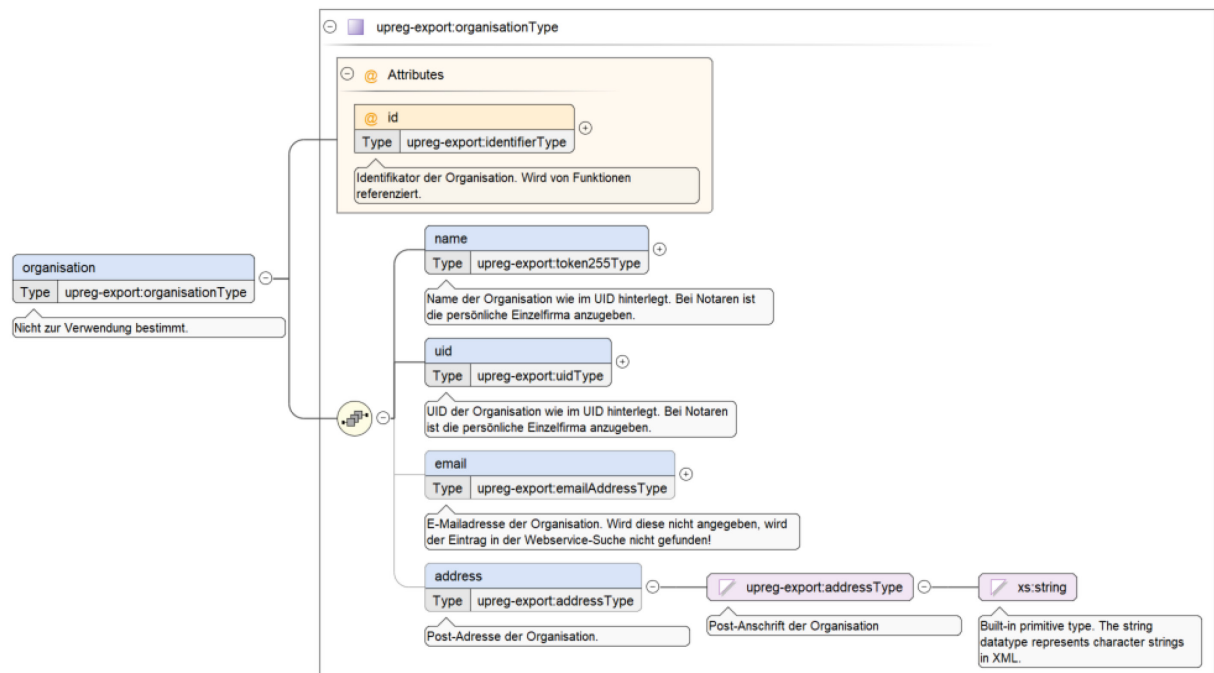


Abbildung 4: Schema der Organisation

4.7. Funktionstypen

Alle Typen von Funktionen, werden bei jedem Export mitgegeben. Diese erlauben die Anbringung einer Zulassungsbestätigung auf einem Dokument.

Das Attribut `id` wird als Identifikator für die Referenz in der Funktion verwendet. Der Identifikator muss innerhalb der Menge der Funktionstypen pro Kanton und Domäne eindeutig sein.

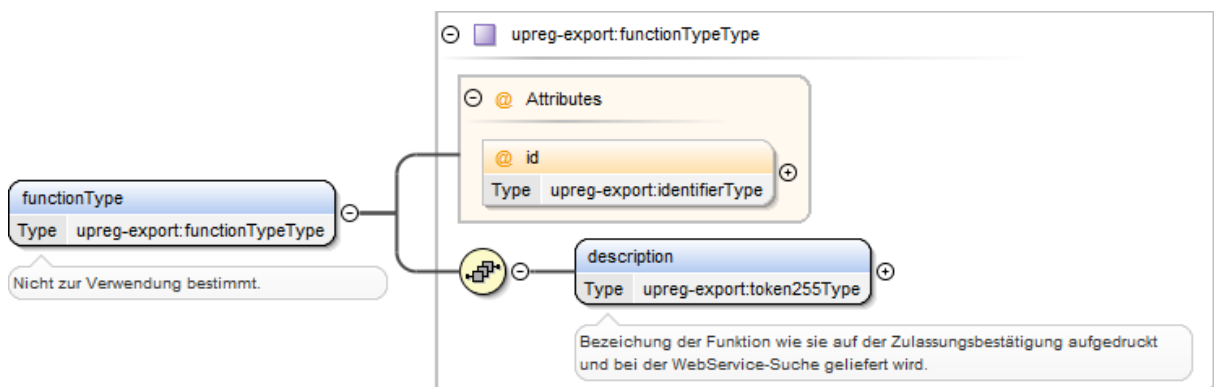


Abbildung 5: Schema des Funktionstyps

Die Bezeichnung des Funktionstyps (Element `description`) ist in der Zulassungsbestätigung und der Antwort des WebServices Suche enthalten und bezeichnet dort den Namen der *Funktion*.

4.8. Funktion

Die Funktion verknüpft eine Person mit einer Organisation und ist von einem bestimmten Funktionstyp. Der Funktion können beliebig viele Zertifikate beigefügt werden.

Der Identifikator (*id*) muss innerhalb der Menge der Funktionen pro Kanton und Domäne eindeutig sein. Er wird in den Metadaten der Zulassungsbestätigung ausgewiesen.

Die referenzierten Personen, Organisationen und Funktionstypen müssen gemäss XML-Schema im Export enthalten sein. Ansonsten wird die ganze Datenlieferung abgelehnt.

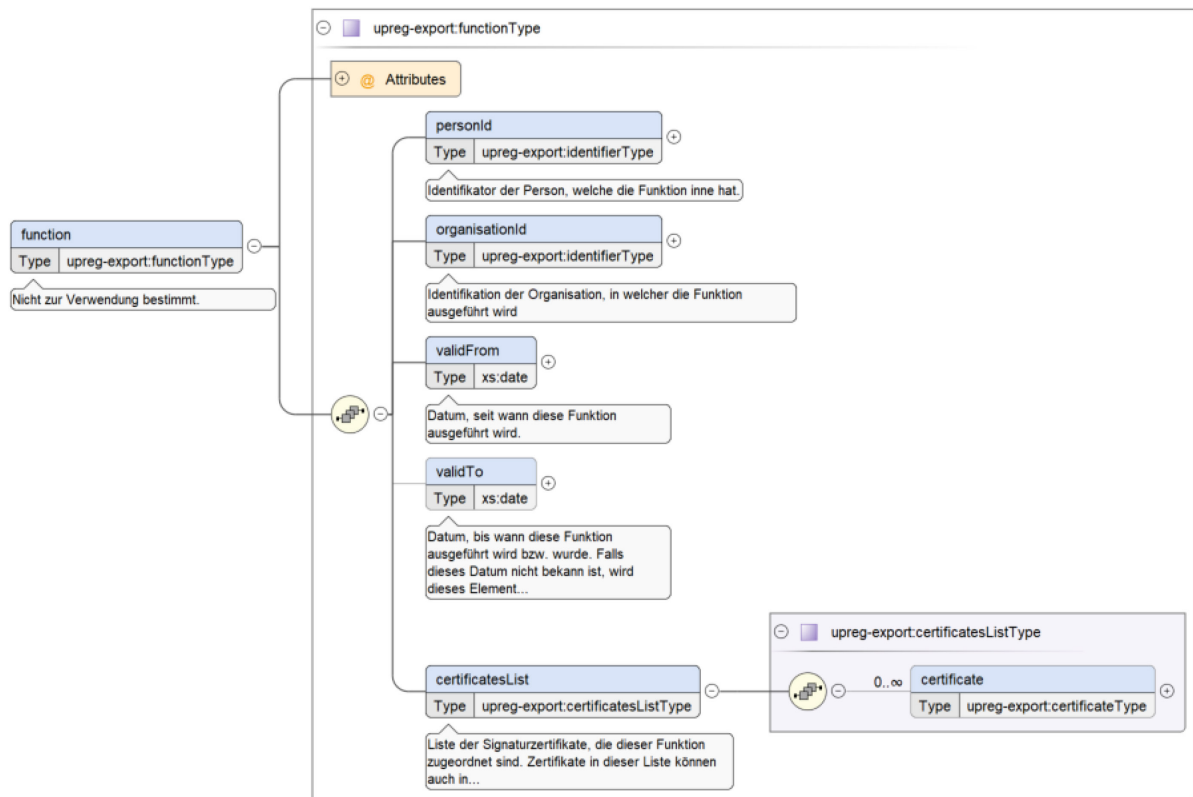


Abbildung 6: Schema der Funktion

Der Gültigkeitszeitraum der Funktion bezeichnet zum einen das Datum, seit welchem eine Funktion ausgeübt wird (*validFrom*) und zum anderen das Datum bis zu welchem (*validTo*), im Namen dieser Funktion, elektronische Ausfertigungen und elektronische Beglaubigungen ausgestellt werden dürfen. Ist das Enddatum noch nicht bekannt, muss keines angegeben werden.

Auch wenn die Funktion nicht mehr aktiv ist, wird empfohlen diese noch während eines Jahres nach Deaktivierung zu exportieren, da Zulassungsbestätigungen auch nachträglich angebracht werden können, sofern der Signaturzeitpunkt innerhalb des Gültigkeitsbereiches der Funktion und des verwendeten Zertifikats liegt.

Der Gültigkeitsbereich der Funktion wirkt sich direkt auf den Verwendungszeitraum der untergeordneten Zertifikate aus, da diese nicht vorher oder länger verwendet werden dürfen als die Funktion gültig ist.

4.8.1. Verwendungs- und Gültigkeitszeiträume

Die Gültigkeitszeiträume von Funktionen können problemlos vor dem Inkrafttreten der Möglichkeit zur elektronischen Signatur von elektronischen Ausfertigungen und elektronischen Beglaubigungen im jeweiligen Kanton einer Domäne liegen. Es werden jedoch nur Zulassungsbestätigungen für Signaturen ausgegeben, welche nach dem Inkrafttreten der nötigen gesetzlichen Rahmenbedingungen angebracht wurden.

Bei Verstoß gegen diese Regeln wird zur Laufzeit beim Anbringen der Zulassungsbestätigung ein Fehler zurückgeliefert.

4.9. Zertifikate

Zertifikate werden direkt einer Funktion zugeordnet. Dabei können beliebig viele Zertifikate zugeordnet werden. Auch wird empfohlen, abgelaufene oder gesperrte Zertifikate zu exportieren, da eine Zulassungsbestätigung auch nachträglich auf einer elektronischen Ausfertigung oder Beglaubigung angebracht werden kann, solange der Zeitpunkt der Signatur innerhalb der Gültigkeit der Zertifikate und der Funktion liegt.

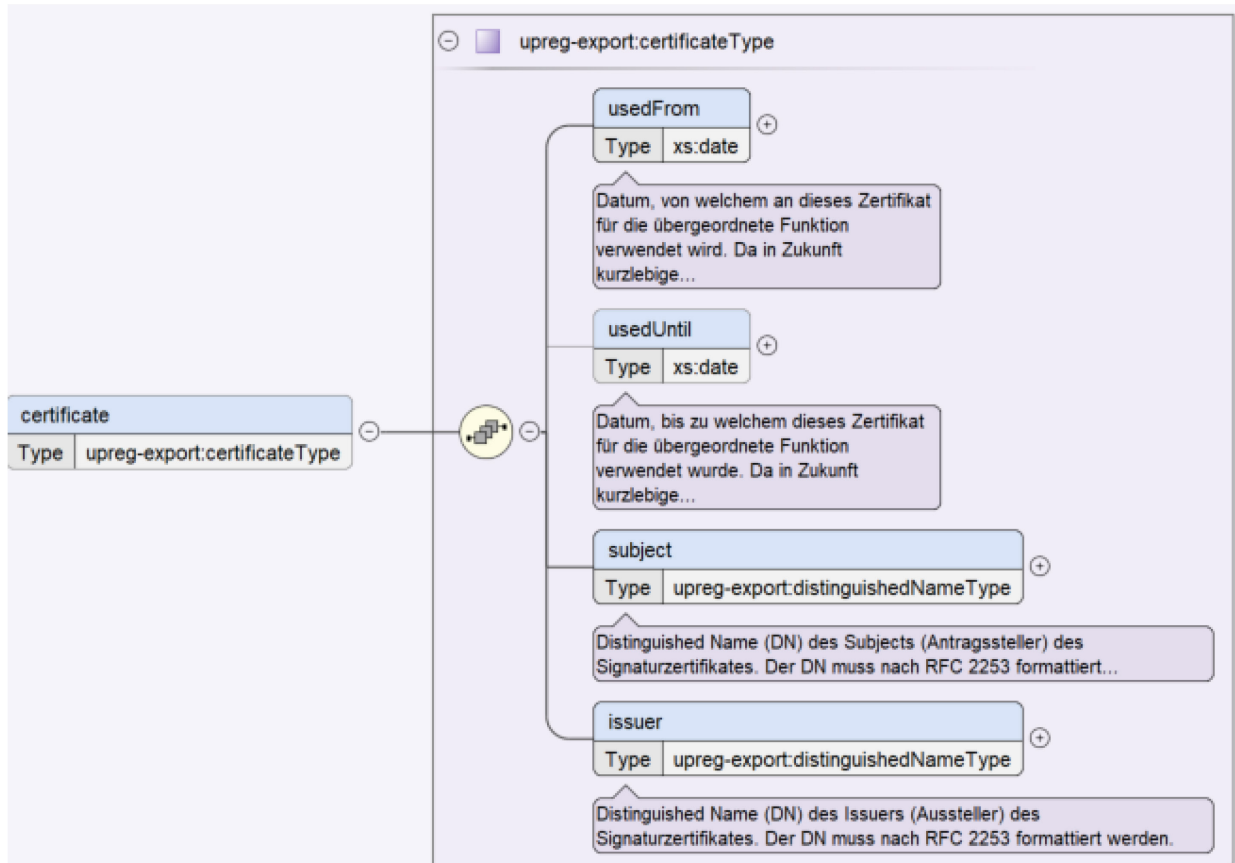


Abbildung 7: Datenstruktur für Zertifikat

4.9.1. Qualifizierte Zertifikate sind persönlich

Da es sich bei diesen Zertifikaten um qualifizierte Zertifikate von Personen handelt, dürfen diese nur einer bestimmten natürlichen Person zugeordnet sein.

Es ist möglich, ein Zertifikat mehreren Funktionen zuzuordnen, solange diese wiederum derselben Person zugeordnet sind.

In Abbildung 8: *Zuordnung von Zertifikaten zu Personen* ist die unterste, rot umrahmte Zuordnung nicht möglich, da das Zertifikat `Zertifikatsig A` bereits der Person A zugeordnet ist. Dies wäre möglich, wenn die Funktion nicht der Person A sondern einer anderen Person A' zugeordnet wäre. Dies entspräche dann dem Fall wie bei Person C.

Werden im Export Zertifikate mehreren Personen zugeordnet, wird der Import mit dem entsprechenden Fehler abgelehnt. Die Fehler-Codes sind unter 6.2 *Fehlercodes* auf der Seite 16 beschrieben.

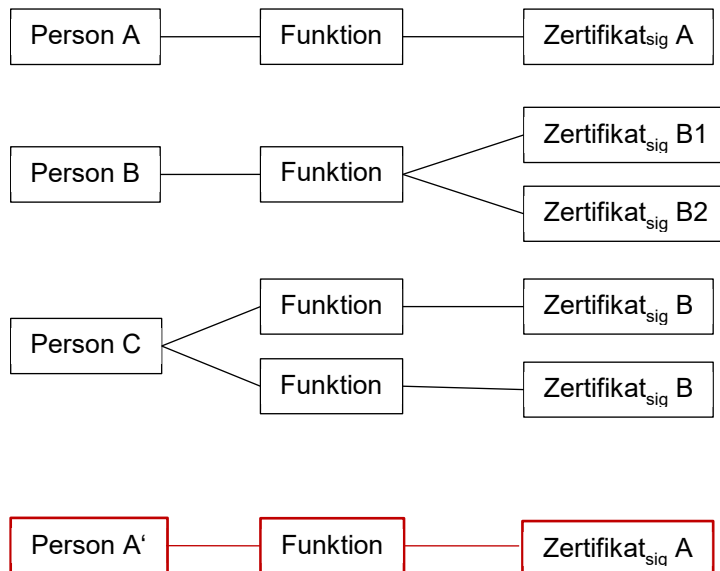


Abbildung 8: Zuordnung von Zertifikaten zu Personen

4.9.2. Verwendungszeiträume von Zertifikaten

In früheren Versionen von UPReg wurden die Verwendungszeiträume von Zertifikaten gegenüber den zugeordneten Funktionen während des Imports überprüft. Diese Regeln wurden jedoch entfernt, da sie mit dem neuen Datenmodell keinen Sinn mehr machen. So ist es nun durchaus möglich, dass eine Person für eine Funktion ein Zertifikat verwendet, das bereits vor Beginn der Funktion gültig war.

Konkret wurden somit folgende Regeln gelöscht:

Das *usedFrom* Datum des Zertifikats darf nicht vor dem *validFrom* Datum der Funktion liegen.

Das *usedUntil* Datum des Zertifikats hingegen darf nicht nach dem *validTo* Datum der Funktion liegen. Hierbei ist zu beachten, dass eine Funktion nicht unbedingt terminiert sein muss. In diesem Fall spielt das *validTo* Datum der Funktion keine Rolle.

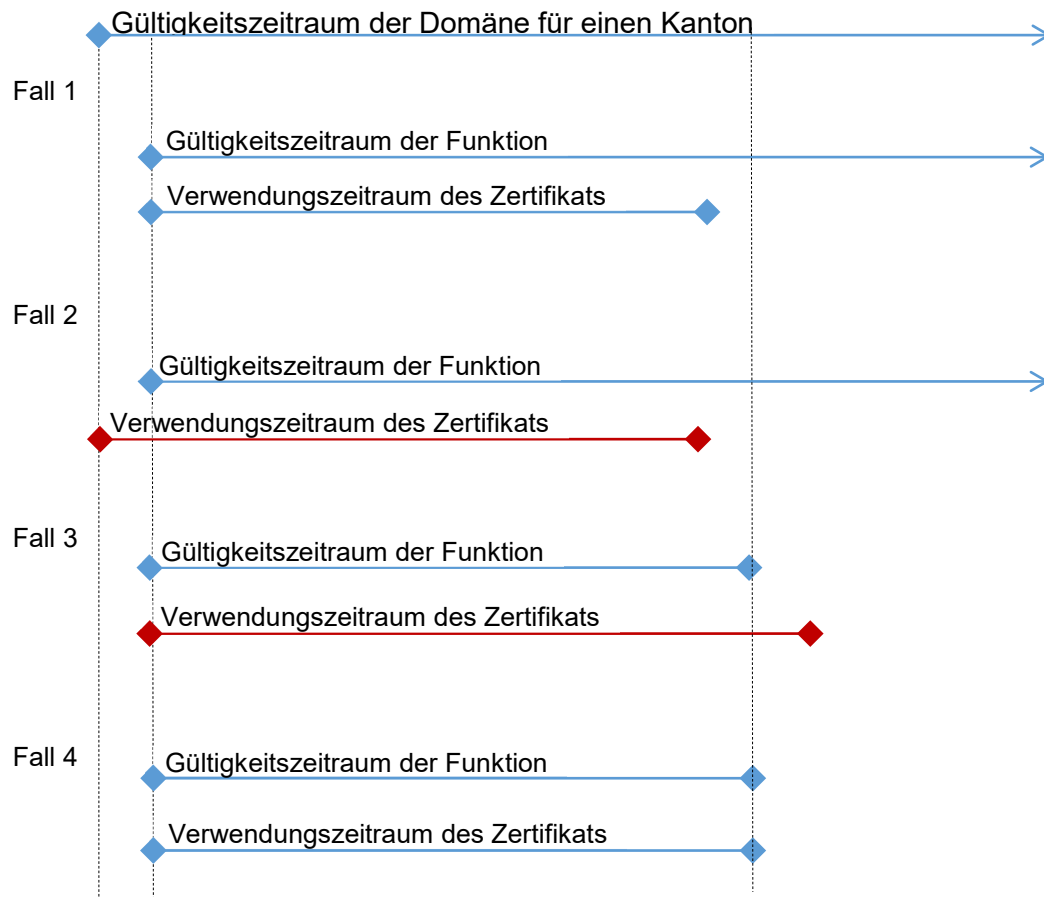


Abbildung 9: Verwendungszeiträume von Zertifikaten

Fall 1: Alle Zeiträume sind gültig.

Fall 2: Der Verwendungszeitraum des Zertifikats darf nicht ausserhalb des Gültigkeitsbereiches der Funktion liegen.

Fall 3: Der Verwendungszeitraum des Zertifikats darf nicht ausserhalb des Gültigkeitszeitraumes der Funktion liegen.

Fall 4: Alle Zeiträume sind gültig.

5. Zu verwendende Typen und globale Einstellungen

Gewisse Typen sind für einen reibungslosen Ablauf von den kantonalen Registern so zu verwenden, wie diese im UPRReg konfiguriert sind.

5.1. Identifikation des Registers

Ein Export wird aufgrund der gelieferten Metadaten einem Kanton (`/export/canton`) und einer Domäne (`/export/domainIdentifizier`) zugeordnet.

Der Domänenidentifikator wird vom Bundesamt von Justiz vergeben und ist so zu verwenden.

Bei der Eintragung eines kantonalen Registers in UPReg wird das Zertifikat, mit welchem die Exporte signiert werden, hinterlegt. Das für die Signatur verwendete Zertifikat muss nicht zwingend ein von einem anerkannten Zertifikatsanbieter ausgestelltes Zertifikat sein.

5.2. Startdatum für elektronische Ausfertigungen und elektronische Beglaubigungen

Ab welchem Datum in einem Kanton elektronische Ausfertigungen und elektronische Beglaubigungen gemäss EÖBV elektronisch signiert werden können, ist abhängig von der jeweiligen kantonalen Gesetzgebung. Dieses Datum wird für Kantone in den jeweiligen Domänen im UPReg konfiguriert. In einem Kanton werden nur Zulassungsbestätigungen für eine Domäne ausgegeben, wenn der Signaturzeitpunkt des jeweiligen Dokumentes nach dem Inkrafttreten des kantonalen Gesetzes liegt.

5.3. Encoding

Die XML-Dateien für den Export und die fachliche Antwort werden immer UTF-8 kodiert.

6. Fachliche Antwort

Die fachliche Antwort bestätigt entweder den erfolgreichen Import oder weist auf ein Problem beim Import hin. Um eine Zuordnung der Antwort zu einem Export zu erlauben, wird die im Export angegebene applikationsspezifische Referenz (*exportIdentifier*) zurückgeliefert.

Im Erfolgsfall enthält die Antwort eine Zusammenfassung des Datenimports (die Anzahl importierter Personen, Organisationen, Funktionstypen und Funktionen).

Im Fehlerfall enthält die Antwort einen Fehlercode und eine textuelle Beschreibung des aufgetretenen Fehlers. Im Fehlerfall werden von UPReg alle angelieferten Daten verworfen und der letzte erfolgreich importierte Export behält seine Gültigkeit.

6.1. Datenstruktur

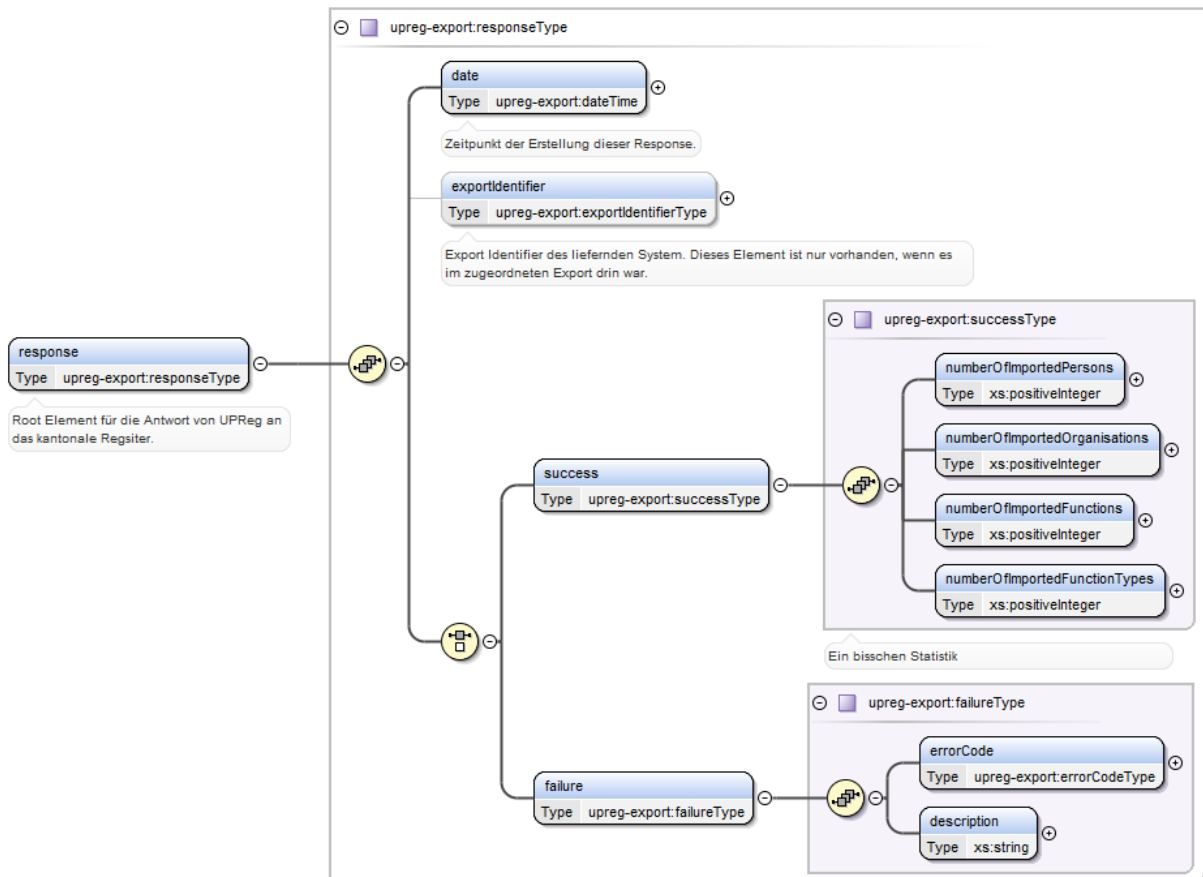


Abbildung 10: Datenstruktur der Antwort

Die Datenintegrität wird vor dem Import zum einen durch eine Validation gegen das XML-Schema geprüft. Danach werden die Signatur und die fachliche Korrektheit der Daten geprüft. Sind diese Prüfungen erfolgreich, werden die Daten übernommen. Die Verantwortung für die Korrektheit der Daten liegt ausschliesslich beim liefernden Register.

Das Element Export Identifier der Antwort (/response/exportIdentifier) ist gemäss XML-Schema fakultativ. UPReg garantiert aber, dass ein in einem Export gelieferter Export Identifier (/export/exportIdentifier) in der zugehörigen Antwort zurückgeliefert wird.

6.2. Fehlercodes

Wird während dem Importieren der Daten ein Problem festgestellt, wird einer der folgenden Fehlercodes in der Antwort mitgegeben.

Code	Fehler	Beschreibung
100	Export ist nicht schemakonform	Das Format des Exports validiert nicht gegen das XML-Schema. Die Fehlermeldung des XML-Parsers wird mitgegeben.
101	Signatur ungültig	Die Signatur des Exports ist nicht gültig.
102	Falsches Signaturzertifikat	Das für die Signatur des Exports verwendete Zertifikat ist nicht dem liefernden Register zugeordnet.

Code	Fehler	Beschreibung
103	Register nicht konfiguriert	Das Register (anhand des Kantons- und Domänenidentifikators) ist nicht oder nicht vollständig konfiguriert.
200	Zertifikat nicht korrekt	Ein Zertifikat (Elemente /export/functions/function/certificatesList/certificate) ist nicht korrekt. Mögliche Fehlerursachen: <ul style="list-style-type: none"> • DN des Antragstellers (subject) ist nicht gemäss RFC 2235 formatiert • DN des Ausstellers (issuer) ist nicht gemäss RFC 2235 formatiert • usedFrom > usedUntil
201	Gleiches Zertifikat ist mehreren Personen zugeordnet	Mindestens ein Zertifikat ist mehreren Personen zugeordnet.
202	Falsche Verwendungszeiträume von Zertifikaten	Überprüfung der in Kap. 4.9.2 definierten Regeln hat einen Fehler ergeben.
300	Interner UPReg Fehler	Es ist ein unvorhergesehener Fehler aufgetreten. Dies ist ein Problem auf der Seite von UPReg. Der Export konnte nicht importiert werden.

7. Sedex Transport

7.1. Meldungsfluss

Die Abbildung 11 zeigt den Meldungsfluss zwischen dem kantonalen Register und UPReg. Der Export wird von der Anwendung des kantonalen Registers erstellt und an den Sedex-Client für die Übermittlung weitergereicht. Auf der Seite von UPReg wird der Export von Sedex-Client an die Anwendung zur Verarbeitung weitergereicht. Der Sedex-Client erstellt eine Transportquittung (/receipt), die von der Anwendung des kantonalen Registers verarbeitet werden sollte, um sicher zu sein, dass der Export an UPReg zugestellt werden konnte.

Sobald UPReg den Export verarbeitet hat, wird eine fachliche Quittung (/response) ausgestellt und dem kantonalen Register via Sedex zugestellt. Das kantonale Register sollte die fachliche Quittung verarbeiten, um sicherzustellen, dass die Daten in UPReg importiert wurden. UPReg seinerseits verarbeitet die Transportquittung, um sicher zu sein, dass die fachliche Antwort an das kantonale Register ausgeliefert wurde.

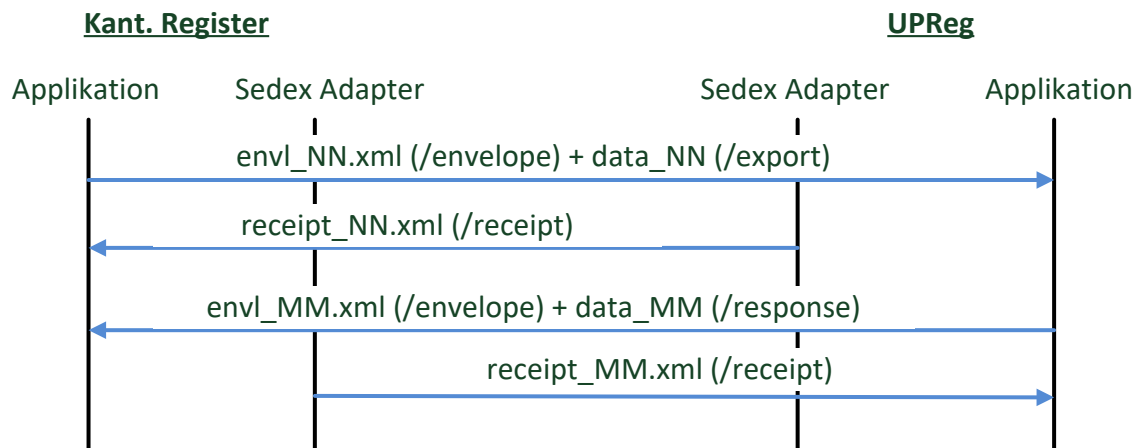


Abbildung 11: Meldungsfluss

7.2. Konventionen

Die folgende Konvention dient dazu, dem Betrieb der Register (sowohl beim Kanton als auch bei UPReg) das Leben bei Nachforschungen betreffend Lieferungen über den Transportkanal Sedex zu vereinfachen.

7.2.1. Konvention für Übermittlung des Exports

Für einen Export mit Sedex Meldungsnummer NNN, der zum Zeitpunkt (/export/date) T1 erstellt und zum Zeitpunkt T2 an Sedex zum Versand übergeben wurde¹, sollen Umschlag und Nutzdatenfile von Sedex wie folgt benannt werden: envl_NNN.xml, data_NNN.xml. Im Umschlag sollen die Elemente wie folgt befüllt werden:

Element im Umschlag	Wert
/envelope/messageId	NNN
/envelope/messageType	Muss noch definiert werden
/envelope/messageClass	0
/envelope/senderId	Sedex ID des Register
/envelope/recipientId	Sedex ID von UPReg, muss noch definiert werden.
/envelope/eventDate	T1, in UTC im Format CCYY-MM-DDThh:mm:ssZ
/envelope/messageDate	T2, in UTC im Format CCYY-MM-DDThh:mm:ssZ

Beispiel eines Umschlags für einen Export mit Meldungsnummer f81d4fae-7dec-11d0-a765-00a0c91e6bf6, der zum Zeitpunkt 2016-08-10T18:13:51Z (10. August 2016, 16:13:51 METS) erstellt und zum Zeitpunkt 2016-08-10T18:20:11Z (10. August 2016, 16:20:11 METS) an Sedex zum Versand übergeben wurde:

```
<?xml version="1.0" encoding="UTF-8"?>
<envelope xmlns="http://www.ech.ch/xmlns/eCH-0090/2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <messageId>f81d4fae-7dec-11d0-a765-00a0c91e6bf6</messageId>
  <messageType>TDB</messageType>
  <messageClass>0</messageClass>
  <senderId>7-4-2</senderId>
  <recipientId>7-4-13</recipientId>
  <eventDate>2016-08-10T18:13:51Z</eventDate>
  <messageDate>2016-08-10T18:20:11Z </messageDate>
</envelope>
```

7.2.2. Konvention für die fachliche Antwort

Für einen Export eines kantonalen Registers mit Sedex Meldungsnummer NNN, wird UPReg eine fachliche Antwort senden. Im Umschlag für die Antwort werden die Elemente wie folgt befüllt:

Element im Umschlag	Wert
/envelope/messageId	MMM

¹ Je nach Implementierung des Export-Prozesses beim kantonalen Register kann natürlich T1 = T2 sein.

/envelope/messageType	Muss noch definiert werden
/envelope/messageClass	1
/envelope/referenceMessageId	NNN (messageld der Sedex Meldung mit dem Export, auf den sich diese Antwort bezieht)
/envelope/senderId	Sedex ID von UPReg, muss noch definiert werden
/envelope/recipientId	Sedex Id des Registers
/envelope/eventDate	In UTC im Format CCYY-MM-DDThh:mm:ssZ
/envelope/messageDate	Gleicher Wert wie /envelope/eventDate

Beispiel des Umschlags für die fachliche Antwort auf einen Export mit sedex Meldungsnummer f81d4fae-7dec-11d0-a765-00a0c91e6bf6:

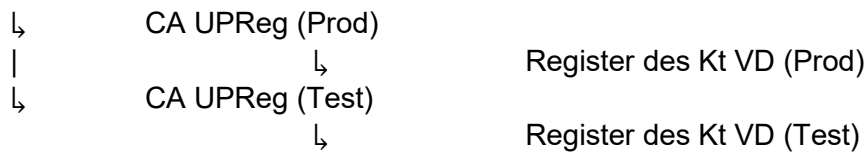
```
<?xml version="1.0" encoding="UTF-8"?>
<envelope xmlns="http://www.ech.ch/xmlns/eCH-0090/2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <messageId>7f707f11-961f-4e5f-84f0-665f279c6965</messageId>
  <messageType>TDB</messageType>
  <messageClass>0</messageClass>
  <referenceMessageId>f81d4fae-7dec-11d0-a765-00a0c91e6bf6</reference-
MessageId>
  <senderId>7-4-13</senderId>
  <recipientId>7-4-2</recipientId>
  <eventDate>2016-08-10T18:13:51Z</eventDate>
  <messageDate>2016-08-10T18:13:51Z</messageDate>
</envelope>
```

8. Zertifikate für die Signatur der Daten

UPReg betreibt eine eigene CA und stellt für die angeschlossenen kantonalen Register Zertifikate aus, mit denen die Lieferungen signiert werden müssen. Die Kantone erstellen einen CSR, der von der entsprechenden CA (Produktion oder Test) signiert wird.

8.1. Struktur der Certificate Authority

Root CA UP-
Reg



9. Glossar

Begriff	Bedeutung
Domäne	Eine Domäne im UPReg kann als eine Art Mandant verstanden werden. UPReg ist fähig, mehrere Domänen (wie Zivilstandswesen, Notariatswesen, Grundbuch etc.) in einer einzigen Instanz zu verwalten.
Kantonale Register	Auch wenn UPReg dazu entwickelt wurde, ein einheitliches Register für alle Kantone zu sein, haben sich gewisse für andere Lösungen entschieden. Diese werden in diesem Kontext „kantonale Register“ genannt. Diese werden die hier beschriebene Schnittstelle dazu verwenden, ihre Daten in das UPReg zu exportieren und diese somit für die Ausstellung der Zulassungsbestätigung zugänglich zu machen.
UPReg	<p>Urkundspersonenregister – www.upreg.ch</p> <p>Das UPReg ist ein Verzeichnis für Urkundspersonen. Diese können sich registrieren, ihre Daten selbständig verwalten und mit geeigneter Software eine sogenannte Zulassungsbestätigung auf einer elektronisch signierten Ausfertigung oder Beglaubigung anbringen.</p> <p>Ob das UPReg von einer Urkundsperson verwendet werden kann, eine andere Registerlösung eingesetzt werden muss oder dies rechtlich nicht zulässig ist, kommt auf die Gesetzgebung des jeweiligen Sitzkantons der juristischen Person, welcher die Urkundsperson im UID-Register der Schweiz zugeordnet ist, an.</p>
Sedex	<p>secure data exchange – www.sedex.ch</p> <p>Eine Plattform für den sicheren asynchronen Datenaustausch zwischen Organisationseinheiten. Die Daten werden hierbei verschlüsselt und nachvollziehbar an registrierte Teilnehmer übermittelt.</p>

10. Version 2.0 der Struktur für den Datenaustausch mittels XML (XSD)

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"

    xmlns:upreg-export="http://www.upreg.ch/export/2"
    targetNamespace="http://www.upreg.ch/export/2" elementFormDefault="qualified"

    xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
    schemaLocation="https://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd"/>

  <xs:element name="export" type="upreg-export:exportType">
    <xs:annotation>
      <xs:documentation xml:lang="de">
        Root Element für die Lieferung des kantonalen Registers an UPReg.
      </xs:documentation>
    </xs:annotation>
    <!-- a function refers to a person -->
    <xs:key name="personKey">
      <xs:selector xpath="./upreg-export:persons/upreg-export:person"/>
      <xs:field xpath="@id"/>
    </xs:key>
    <xs:keyref name="personKeyref" refer="upreg-export:personKey">
      <xs:selector xpath="./upreg-export:functions/upreg-export:function"/>
      <xs:field xpath="upreg-export:personId"/>
    </xs:keyref>

    <!-- a function refers to an organisation -->
    <xs:key name="organisationKey">

```

```

        <xs:selector xpath="./upreg-export:organisations/upreg-export:organisation"/>
        <xs:field xpath="@id"/>
    </xs:key>
    <xs:keyref name="organisationKeyref" refer="upreg-export:organisationKey">
        <xs:selector xpath="./upreg-export:functions/upreg-export:function"/>
        <xs:field xpath="upreg-export:organisationId"/>
    </xs:keyref>

    <!-- a function refers to a function type -->
    <xs:key name="functionTypeKey">
        <xs:selector xpath="./upreg-export:functionTypes/upreg-export:functionType"/>
        <xs:field xpath="@id"/>
    </xs:key>
    <xs:keyref name="functionTyperef" refer="upreg-export:functionTypeKey">
        <xs:selector xpath="./upreg-export:functions/upreg-export:function"/>
        <xs:field xpath="@functionTypeId"/>
    </xs:keyref>

    <!-- identifiers functions must be unique -->
    <xs:unique name="functionUnique">
        <xs:selector xpath="./upreg-export:functions/upreg-export:function"/>
        <xs:field xpath="@id"/>
    </xs:unique>
</xs:element>

<xs:simpleType name="exportIdentifierType">
    <xs:annotation>
        <xs:documentation xml:lang="de">
            Datentyp für den Export Identifier des liefernden Systems.
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">

```

```
        <xs:maxLength value="128"/>
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="exportType">
    <xs:sequence>
        <xs:element name="domainIdentifrier" type="xs:token">
            <xs:annotation>
                <xs:documentation xml:lang="de">
                    Identifikator der Domäne. Wird durch UPReg bestimmt. Ein Export kann
                    nur für genau eine Domäne eingeliefert werden. Bedient ein Register
                    mehrere Domänen, muss für jede Domäne ein eigener Export geliefert
                    werden.
                </xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="canton" type="upreg-export:cantonAbbreviationType">
            <xs:annotation>
                <xs:documentation xml:lang="de">
                    Identifikator des Kantons. Ein Export kann nur für genau einen Kanton
                    eingeliefert werden. Bedient ein Register mehrere Kantone, muss für
                    jeden Kanton ein eigener Export geliefert werden.
                </xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="date" type="upreg-export:dateTime">
            <xs:annotation>
                <xs:documentation xml:lang="de">
                    Datum des Exports.
                </xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

```
<xs:element name="exportIdentifizier" type="upreg-export:exportIdentifizierType" minOccurs="0">
  <xs:annotation>
    <xs:documentation xml:lang="de">
      Export Identifizier des liefernden Systems. Dieser Wert wird eins-zu-eins
      in die Antwort durchgeschlauft. Kann vom liefernden System für die
      Zuordnung der Antwort von UPReg zum gesendeten Export oder für andere
      applikationsspezifische Zwecke verwendet werden.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="persons" type="upreg-export:personsType">
  <xs:annotation>
    <xs:documentation xml:lang="de">
      Liste der Personen. Diese Liste enthält alle von Funktionen referenzierten
      Personen. Es können hingegen Personen enthalten sein, welche nicht referenziert
      werden.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="organisations" type="upreg-export:organisationsType">
  <xs:annotation>
    <xs:documentation xml:lang="de">
      Liste der Organisationen. Diese Liste enthält alle von Funktionen referenzierten
      Organisationen. Es können hingegen Organisationen enthalten sein, welche nicht
      referenziert werden.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="functions" type="upreg-export:functionsType">
  <xs:annotation>
    <xs:documentation xml:lang="de">
```



```

        Liste der Funktionen.
    </xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="functionTypes" type="upreg-export:functionTypesType">
    <xs:annotation>
        <xs:documentation xml:lang="de">
            Liste der Funktionstypen. Diese Liste enthält alle von Funktionen referenzierten
            Funktionstypen. Es können hingegen Funktionstypen enthalten sein, welche nicht
            referenziert werden.
        </xs:documentation>
    </xs:annotation>
</xs:element>

<xs:element ref="ds:Signature">
    <xs:annotation>
        <xs:documentation xml:lang="de">
            Signatur des Registers.

            Canonicalization-Methode:
            http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments (c14n mit Kommentar).
            Signaturalgorithmus:
            RSA mit SHA256.
        </xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>

<!--
    Hinweis: die Typen personsType, organisationsType, functionsType und functionTypesType verwenden bewusst eine
    Referenz auf Elemente (ref="upreg-export:organisation"). Wenn die Elemente person, organisation, function und

```

functionType direkt im complexType definiert werden (mit name="person type="upreg-export:personType"), funktioniert der Key-Mechanismus nicht mehr. Das ist wahrscheinlich ein Xalan Problem.

```
-->
```

```
<xs:complexType name="personsType">
  <xs:sequence>
    <xs:element ref="upreg-export:person" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="organisationsType">
  <xs:sequence>
    <xs:element ref="upreg-export:organisation" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="functionsType">
  <xs:sequence>
    <xs:element ref="upreg-export:function" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="functionTypesType">
  <xs:sequence>
    <xs:element ref="upreg-export:functionType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!--
  Elemente zur Adressierung für Key/Keyref-Mechanismus. Diese Elemente dürfen nicht standalone verwendet werden.
-->
<xs:element name="person" type="upreg-export:personType">
  <xs:annotation>
```

```

        <xs:documentation xml:lang="de">Nicht zur Verwendung bestimmt.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="organisation" type="upreg-export:organisationType">
    <xs:annotation>
        <xs:documentation xml:lang="de">Nicht zur Verwendung bestimmt.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="function" type="upreg-export:functionType">
    <xs:annotation>
        <xs:documentation xml:lang="de">Nicht zur Verwendung bestimmt.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="functionType" type="upreg-export:functionTypeType">
    <xs:annotation>
        <xs:documentation xml:lang="de">Nicht zur Verwendung bestimmt.</xs:documentation>
    </xs:annotation>
</xs:element>

<xs:complexType name="functionType">
    <xs:sequence>
        <xs:element name="personId" type="upreg-export:identifizierType">
            <xs:annotation>
                <xs:documentation xml:lang="de">
                    Identifikator der Person, welche die Funktion inne hat.
                </xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="organisationId" type="upreg-export:identifizierType">
            <xs:annotation>
                <xs:documentation xml:lang="de">
                    Identifikation der Organisation, in welcher die Funktion

```

```
        ausgeführt wird
    </xs:documentation>
</xs:annotation>
</xs:element>

<xs:element name="validFrom" type="xs:date">
    <xs:annotation>
        <xs:documentation xml:lang="de">
            Datum, seit wann diese Funktion ausgeführt wird.
        </xs:documentation>
    </xs:annotation>
</xs:element>

<xs:element name="validTo" type="xs:date" minOccurs="0">
    <xs:annotation>
        <xs:documentation xml:lang="de">
            Datum, bis wann diese Funktion ausgeführt wird bzw. wurde.
            Falls dieses Datum nicht bekannt ist, wird dieses Element
            nicht angegeben. Das Datum kann in der Zukunft liegen, wenn
            der Zeitraum terminiert ist.
        </xs:documentation>
    </xs:annotation>
</xs:element>

<xs:element name="certificatesList" type="upreg-export:certificatesListType">
    <xs:annotation>
        <xs:documentation xml:lang="de">
            Liste der Signaturzertifikate, die dieser Funktion zugeordnet sind.
            Zertifikate in dieser Liste können auch in Zertifikatslisten
            anderer Funktionen vorkommen, falls dieselben Zertifikate für
            mehrere Funktionen benutzt werden.
        </xs:documentation>
    </xs:annotation>
</xs:element>
```

```

        </xs:annotation>
    </xs:element>
</xs:sequence>

<xs:attribute name="id" use="required" type="upreg-export:identifizierType">
    <xs:annotation>
        <xs:documentation xml:lang="de">
            Identifikator dieser Funktion.
        </xs:documentation>
    </xs:annotation>
</xs:attribute>
<xs:attribute name="functionTypeId" use="required" type="upreg-export:identifizierType">
    <xs:annotation>
        <xs:documentation xml:lang="de">
            Identifikator des Typs der Funktion.
        </xs:documentation>
    </xs:annotation>
</xs:attribute>
</xs:complexType>

<xs:complexType name="certificatesListType">
    <xs:sequence>
        <xs:element name="certificate" type="upreg-export:certificateType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<!-- function type related types -->
<xs:complexType name="functionTypeType">
    <xs:sequence>
        <xs:element name="description" type="upreg-export:token255Type">
            <xs:annotation>
                <xs:documentation xml:lang="de">

```

Bezeichnung der Funktion wie sie auf der Zulassungsbestätigung aufgedruckt und bei der Webservice-Suche geliefert wird.

```
</xs:documentation>
```

```
</xs:annotation>
```

```
</xs:element>
```

```
</xs:sequence>
```

```
<xs:attribute name="id" type="upreg-export:identifizierType"/>
```

```
</xs:complexType>
```

```
<!-- certificate related types -->
```

```
<xs:complexType name="certificateType">
```

```
<xs:sequence>
```

```
<xs:element name="usedFrom" type="xs:date">
```

```
<xs:annotation>
```

```
<xs:documentation xml:lang="de">
```

Datum, von welchem an dieses Zertifikat für die übergeordnete Funktion verwendet wird.

Da in Zukunft kurzlebige Zertifikate von Signaturdiensten verwendet werden, ist kein direkter Zusammenhang zwischen der Gültigkeitsdauer des für eine Signatur eingesetzten (kurzlebigen) Zertifikats und diesem Element gegeben.

Für langlebige Zertifikate muss dieses Datum innerhalb der Gültigkeitszeitspanne des Zertifikats (valid-not-before bis valid-not-after) und muss vor dem usedUntil Datum liegen (max(valid-not-before, function:valid-from) <= used-from)

```
</xs:documentation>
```

```
</xs:annotation>
```

```
</xs:element>
```

```
<xs:element name="usedUntil" type="xs:date" minOccurs="0">
```

```
<xs:annotation>
```

```
<xs:documentation xml:lang="de">
```

Datum, bis zu welchem dieses Zertifikat für die übergeordnete Funktion

verwendet wurde.

Da in Zukunft kurzlebige Zertifikate von Signaturdiensten verwendet werden, ist kein direkter Zusammenhang zwischen der Gültigkeitsdauer des für eine Signatur eingesetzten (kurzlebigen) Zertifikats und diesem Element gegeben. Für langlebige Zertifikate muss dieses Datum innerhalb der Gültigkeitszeitspanne des Zertifikats (valid-not-before bis valid-not-after) und muss nach dem usedFrom Datum liegen (min(valid-not-after, function:valid-to) >= used-until).

```

    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="subject" type="upreg-export:distinguishedNameType">
  <xs:annotation>
    <xs:documentation xml:lang="de">
      Distinguished Name (DN) des Subjects (Antragssteller) des Signaturzertifikates.
      Der DN muss nach RFC 2253 formatiert werden.
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="issuer" type="upreg-export:distinguishedNameType">
  <xs:annotation>
    <xs:documentation xml:lang="de">
      Distinguished Name (DN) des Issuers (Aussteller) des Signaturzertifikates.
      Der DN muss nach RFC 2253 formatiert werden.
    </xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>

<!-- person related types -->
<xs:complexType name="personType">
  <xs:sequence>

```

```
<xs:element name="officialName" type="upreg-export:personNameType">
  <xs:annotation>
    <xs:documentation xml:lang="de">
      Nachname der Person wie auf der Ausweisschrift angegeben.
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="firstNames" type="upreg-export:personNameType">
  <xs:annotation>
    <xs:documentation xml:lang="de">
      Alle Vornamen der Person wie auf der Ausweisschrift angegeben.
      Wird als Name auf die Zulassungsbestätigung gedruckt und in die
      Meta-Informationen der Signatur eingebettet.
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="gender" type="upreg-export:genderType">
  <xs:annotation>
    <xs:documentation xml:lang="de">
      Wird bei der Webservice Suche zurückgeliefert.
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="contactInformation" type="upreg-export:privateContactInformationType" minOccurs="0">
  <xs:annotation>
    <xs:documentation xml:lang="de">
      Diese Kontaktinformationen werden ausschliesslich für die
      Kontaktaufnahme durch den Support in Problemenfällen genutzt.
      Diese werden nie veröffentlicht.
    </xs:documentation>
  </xs:annotation>
</xs:element>
```



```

</xs:sequence>

<xs:attribute name="id" use="required" type="upreg-export:identifizierType">
  <xs:annotation>
    <xs:documentation xml:lang="de">
      Identifikator der Person. Wird von Funktionen referenziert.
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
</xs:complexType>

<xs:complexType name="privateContactInformationType">
  <xs:sequence>
    <xs:element name="telephone" type="upreg-export:token20Type" minOccurs="0"/>
    <xs:element name="email" type="upreg-export:emailAddressType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<!-- organisation related types -->
<xs:complexType name="organisationType">
  <xs:sequence>
    <xs:element name="name" type="upreg-export:token255Type">
      <xs:annotation>
        <xs:documentation xml:lang="de">
          Name der Organisation wie im UID hinterlegt. Bei Notaren
          ist die persönliche Einzelfirma anzugeben.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="uid" type="upreg-export:uidType">
      <xs:annotation>
        <xs:documentation xml:lang="de">

```

UID der Organisation wie im UID hinterlegt. Bei Notaren ist die persönliche Einzelfirma anzugeben.

```

    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="email" type="upreg-export:emailAddressType" minOccurs="0">
  <xs:annotation>
    <xs:documentation xml:lang="de">
      E-Mailadresse der Organisation. Wird diese nicht angegeben,
      wird der Eintrag in der Webservice-Suche nicht gefunden!
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="address" type="upreg-export:addressType" minOccurs="0">
  <xs:annotation>
    <xs:documentation xml:lang="de">
      Post-Adresse der Organisation.
    </xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>

<xs:attribute name="id" use="required" type="upreg-export:identifizierType">
  <xs:annotation>
    <xs:documentation xml:lang="de">
      Identifikator der Organisation. Wird von Funktionen
      referenziert.
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
</xs:complexType>

```

```
<!-- common types -->
<xs:simpleType name="identfierType">
  <xs:annotation>
    <xs:documentation xml:lang="de">
      Identifikator in Form einer UUID-v4.
    </xs:documentation>
  </xs:annotation>

  <xs:restriction base="xs:token">
    <xs:maxLength value="36"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="personNameType">
  <xs:restriction base="xs:token">
    <xs:maxLength value="100"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="genderType">
  <xs:restriction base="xs:token">
    <xs:enumeration value="male"/>
    <xs:enumeration value="female"/>
    <xs:enumeration value="other"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="distinguishedNameType">
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="preserve"/>
    <xs:maxLength value="1024"/>
  </xs:restriction>
</xs:simpleType>
```

```
</xs:simpleType>

<xs:simpleType name="token255Type">
  <xs:restriction base="xs:token">
    <xs:maxLength value="255"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="token20Type">
  <xs:restriction base="xs:token">
    <xs:maxLength value="20"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="uidType">
  <xs:restriction base="xs:string">
    <xs:length value="15" fixed="true"/>
    <xs:pattern value="(CHE|ADM)-\d{3}\.\d{3}\.\d{3}"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="dateTime">
  <xs:annotation>
    <xs:documentation>
      Angabe des Zeitpunkt in UTC im Format CCYY-MM-DDThh:mm:ssZ
    </xs:documentation>
  </xs:annotation>

  <xs:restriction base="xs:dateTime">
    <xs:pattern value="\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}Z"/>
  </xs:restriction>
</xs:simpleType>
```

```

<xs:simpleType name="emailAddressType">
  <xs:annotation>
    <xs:documentation>
      Die formale Definition dessen, wie eine e-Mail Adresse aussieht, ist in diesen
      RFC ersichtlich:
      - RFC 5322 (sections 3.2.3 and 3.4.1, obsoletes RFC 2822), RFC 5321, RFC 3696,
      - RFC 6531 (permitted characters).
      Weil eine korrekte Prüfung der erlaubten Zeichen mittels Regex sehr kompliziert
      wird, beschränken wird uns hier bei der Prüfung auf ein Minimum. In der E-Mail
      Adresse es muss ein @ Zeichen vorhanden sein und der String ist so aufgebaut: max.
      64@255 Zeichen, nicht mehr als 256 total.
      Siehe auch:
      http://stackoverflow.com/questions/2049502/what-characters-are-allowed-in-email-address
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:token">
    <xs:maxLength value="256"/>
    <xs:pattern value=".{1,64}@.{1,255}"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="addressType">
  <xs:annotation>
    <xs:documentation>
      Post-Anschrift der Organisation
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:maxLength value="1024"/>
  </xs:restriction>
</xs:simpleType>

```

```
<xs:simpleType name="cantonAbbreviationType">
```

```
  <xs:annotation>
```

```
    <xs:documentation xml:lang="de">
```

Abgeleitet von eCH-0071:cantonAbbreviationType (eCH-0071 Datenstandard historisiertes Gemeindeverzeichnis der Schweiz).

Zusätzlich ist hier der Kanton TS für Testlieferung und der "Kanton" CH (=schweizweit) aufgenommen worden.

```
    </xs:documentation>
```

```
  </xs:annotation>
```

```
  <xs:restriction base="xs:token">
```

```
    <xs:maxLength value="2"/>
```

```
    <xs:enumeration value="ZH"/>
```

```
    <xs:enumeration value="BE"/>
```

```
    <xs:enumeration value="LU"/>
```

```
    <xs:enumeration value="UR"/>
```

```
    <xs:enumeration value="SZ"/>
```

```
    <xs:enumeration value="OW"/>
```

```
    <xs:enumeration value="NW"/>
```

```
    <xs:enumeration value="GL"/>
```

```
    <xs:enumeration value="ZG"/>
```

```
    <xs:enumeration value="FR"/>
```

```
    <xs:enumeration value="SO"/>
```

```
    <xs:enumeration value="BS"/>
```

```
    <xs:enumeration value="BL"/>
```

```
    <xs:enumeration value="SH"/>
```

```
    <xs:enumeration value="AR"/>
```

```
    <xs:enumeration value="AI"/>
```

```
    <xs:enumeration value="SG"/>
```

```
    <xs:enumeration value="GR"/>
```

```
    <xs:enumeration value="AG"/>
```

```
    <xs:enumeration value="TG"/>
```

```
<xs:enumeration value="TI"/>
<xs:enumeration value="VD"/>
<xs:enumeration value="VS"/>
<xs:enumeration value="NE"/>
<xs:enumeration value="GE"/>
<xs:enumeration value="JU"/>
<xs:enumeration value="TS"/>
<xs:enumeration value="CH"/>
</xs:restriction>
</xs:simpleType>

<!-- root elements for response -->
<xs:element name="response" type="upreg-export:responseType">
  <xs:annotation>
    <xs:documentation xml:lang="de">
      Root Element für die Antwort von UPReg an das kantonale Regsiter.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<!-- response related types -->
<xs:complexType name="responseType">
  <xs:sequence>
    <xs:element name="date" type="upreg-export:dateTime">
      <xs:annotation>
        <xs:documentation xml:lang="de">
          Zeitpunkt der Erstellung dieser Response.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="exportIdentifier" type="upreg-export:exportIdentifierType" minOccurs="0">
      <xs:annotation>
```

```

    <xs:documentation xml:lang="de">
      Export Identifier des liefernden System. Dieses Element ist nur vorhanden,
      wenn es im zugeordneten Export drin war.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:choice>
  <xs:element name="success" type="upreg-export:successType">
    <xs:annotation>
      <xs:documentation xml:lang="de">
        Dieses Element ist enthalten, wenn der Import
        erfolgreich durchgeführt werden konnte.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="failure" type="upreg-export:failureType">
    <xs:annotation>
      <xs:documentation xml:lang="de">
        Ist beim Import ein Problem aufgetreten, wird
        dieses Element abgefüllt. An den Stammdaten
        wurde noch keine Änderung vorgenommen.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:choice>

</xs:sequence>
</xs:complexType>

<xs:complexType name="successType">
  <xs:annotation>

```



```
<xs:documentation xml:lang="de">Ein bisschen Statistik</xs:documentation>
</xs:annotation>
<xs:sequence>
  <xs:element name="numberOfImportedPersons" type="xs:positiveInteger">
    <xs:annotation>
      <xs:documentation xml:lang="de">
        Anzahl der importierten Personen. Kann vom Liefersystem
        ausgewertet werden falls gewünscht.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="numberOfImportedOrganisations" type="xs:positiveInteger">
    <xs:annotation>
      <xs:documentation xml:lang="de">
        Anzahl der importierten Organisationen. Kann vom Liefersystem
        ausgewertet werden falls gewünscht.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="numberOfImportedFunctions" type="xs:positiveInteger">
    <xs:annotation>
      <xs:documentation xml:lang="de">
        Anzahl der importierten Funktionen. Kann vom Liefersystem
        ausgewertet werden falls gewünscht.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="numberOfImportedFunctionTypes" type="xs:positiveInteger">
    <xs:annotation>
      <xs:documentation xml:lang="de">
        Anzahl der importierten Funktionstypen. Kann vom
        Liefersystem ausgewertet werden falls gewünscht.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

```
        </xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="failureType">
  <xs:annotation>
    <xs:documentation xml:lang="de">
      Beschreibt den aufgetretenen Fehler.
    </xs:documentation>
  </xs:annotation>

  <xs:sequence>
    <xs:element name="errorCode" type="upreg-export:errorCodeType">
      <xs:annotation>
        <xs:documentation xml:lang="de">
          Fehler-Code gemäss Tabelle.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="description" type="xs:string">
      <xs:annotation>
        <xs:documentation xml:lang="de">
          Beschreibung des Fehlers. Kann technische
          Details enthalten und sollte nicht dem
          Benutzer angezeigt werden.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

```
<xs:simpleType name="errorCodeType">  
  <xs:restriction base="xs:token">  
    <xs:length value="4"/>  
    <xs:pattern value="\d{4}"/>  
  </xs:restriction>  
</xs:simpleType>  
</xs:schema>
```