



sedex-Entwicklung

---

# **sedex im Projekt eUmzugCH/eUmzugZH**

## **Integrationsanleitung für Nutzer und Anbieter synchroner SOAP-Webservices**

Version 1.2 vom 15.08.2016

---

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Referenzierte Dokumente</b>	<b>3</b>
<b>3</b>	<b>Die Vorhaben „eUmzugCH“ und „eUmzugZH“</b>	<b>3</b>
3.1	Das Projekt im Allgemeinen .....	3
3.2	Prozess und Service-Integration .....	3
3.2.1	Zu integrierende SOAP-Webservices .....	4
3.3	Anforderungen an die Service-Integration .....	4
<b>4</b>	<b>Einführung in die Services von sedex für synchrone SOAP-Webservices</b>	<b>5</b>
4.1	System sedex .....	5
4.2	sedex-Client .....	5
4.3	sedex-Webservice-Proxy .....	6
4.4	sedex-externalAuthorization-Service .....	7
4.5	Anforderungsabdeckung durch sedex .....	8
<b>5</b>	<b>sedex-Lösung für SOAP-Kommunikation in eUmzugCH/ZH</b>	<b>8</b>
5.1	Herausforderungen beim Einsatz von sedex für eUmzugCH .....	8
5.1.1	Service zur Personenidentifikation erfordert Content Based Routing .....	8
5.1.2	Authentifizierung und Autorisierung seitens Dienstleister .....	9
5.2	Lösungsüberblick .....	9
<b>6</b>	<b>Integration auf Seite Dienstverwender</b>	<b>11</b>
6.1	Vorbereitende Arbeiten/Organisatorisches .....	11

6.1.1	sedex-Teilnehmer bestellen und einrichten .....	11
<b>6.2</b>	<b>Installation sedex-Webservice-Proxy .....</b>	<b>12</b>
6.2.1	Besondere Hinweise im Kontext eUmzugCH .....	12
<b>6.3</b>	<b>Installation AAR-Konfigurationspakete der angebotenen Webservices .....</b>	<b>13</b>
6.3.1	Webservice-Definitionen/AAR-Dateien .....	13
6.3.2	Installation einer neuen AAR-Datei:.....	13
6.3.3	Installation eines neuen SSL-Truststore .....	14
<b>6.4</b>	<b>Content Based Routing für Personenidentifikation .....</b>	<b>14</b>
<b>6.5</b>	<b>Weitere technische Anforderungen .....</b>	<b>15</b>
<b>6.6</b>	<b>Nutzung der Webservices via Webservice-Proxy .....</b>	<b>15</b>
<b>7</b>	<b>Integration auf Seite Dienstanbieter .....</b>	<b>17</b>
<b>7.1</b>	<b>Szenario 1: „Nur bestehender Security Reverse Proxy“ .....</b>	<b>18</b>
<b>7.2</b>	<b>Szenario 2: „Bestehender Security Reverse Proxy und Membrane“ .....</b>	<b>19</b>
7.2.1	Konfigurieren des Security Reverse Proxy .....	21
7.2.2	Konfigurieren des Membrane Service Proxy .....	21
<b>7.3</b>	<b>Szenario 3: „Nur Membrane“ .....</b>	<b>23</b>
7.3.1	Konfiguration des Membrane .....	24
<b>7.4</b>	<b>Vorbereitende Arbeiten/Organisatorisches .....</b>	<b>26</b>
7.4.1	sedex-Teilnehmer bestellen und einrichten .....	26
7.4.2	SSL-Zertifikat beschaffen.....	26
<b>7.5</b>	<b>Weitere technische Anforderungen .....</b>	<b>26</b>
<b>7.6</b>	<b>Besondere Anmerkungen .....</b>	<b>26</b>
7.6.1	Einfügen des SSL-Zertifikats in den HTTP-Header .....	26
7.6.2	Einfügen der Client-sedexId in den HTTP-Header .....	26
7.6.3	Certificate Authority (CA) für sedex-Zertifikate .....	27
7.6.4	Benutzung des sedex-Webservice externalAuthorization .....	27
7.6.5	Erstellen von Keystore-/Truststore-Dateien .....	29
<b>8</b>	<b>Membrane Service Proxy .....</b>	<b>30</b>
<b>8.1</b>	<b>Einführung .....</b>	<b>30</b>
<b>8.2</b>	<b>Download .....</b>	<b>30</b>
<b>8.3</b>	<b>Installation .....</b>	<b>30</b>
<b>8.4</b>	<b>Konfiguration .....</b>	<b>30</b>
8.4.1	Logging .....	31
8.4.2	Option: SSL-Verbindung von Membrane zur Service-Implementation .....	32
8.4.3	Option: Membrane-Administrationskonsole .....	32
8.4.4	Option: Schutz gegen XML- und DoS-Attacken .....	33
<b>8.5</b>	<b>Plugins/Interceptoren .....</b>	<b>34</b>
8.5.1	Installation eines Plugins/Interceptors .....	34
8.5.2	sedex Client Certificate Interceptor .....	34
8.5.3	sedex External Authorization Interceptor .....	35
8.5.4	sedex Whitelist Interceptor .....	36
8.5.5	Konfiguration der Interceptoren in der Membrane-Administrationskonsole .....	38
<b>8.6</b>	<b>Kommentierte Muster-Konfigurationsdatei .....</b>	<b>38</b>
<b>9</b>	<b>Fehlereingrenzung und -suche .....</b>	<b>41</b>
<b>10</b>	<b>Typische Fragen (FAQ) .....</b>	<b>44</b>

# 1 Einleitung

Das vorliegende Dokument zeigt Anbietern und Verwendern von synchronen SOAP-Webservices innerhalb der Projekte eUmzugCH und eUmzugZH auf, wie sie ihre Services über sedex integrieren können.

## 2 Referenzierte Dokumente

Abkürzung	Titel, Quelle
[sdxBHB]	sedex-Betriebs-/Integrationshandbuch V5.0.0
[sdxIHB]	Installation instructions for sedex Client V5.0.0
[sdxWSP]	Benutzerhandbuch des Webservice-Proxy V5.0.0
[sdxExtAuth]	Technische Spezifikation - Schnittstelle sedex-Autorisierungs-Dienst
[sdxAARPersIdent]	sedex-Webservice-Proxy: Benutzerhandbuch zum Personenidentifikations-Service (eUmzugCH)

## 3 Die Vorhaben „eUmzugCH“ und „eUmzugZH“

### 3.1 Das Projekt im Allgemeinen

Zusammen mit E-Government Schweiz hat der Verband Schweizerischer Einwohnerdienste (VSED) im Rahmen von „eUmzugCH“ (Projekt A1.12) ein Fach- und ein Lösungskonzept zur elektronischen Meldung und Abwicklung der Ereignisse *Umzug*, *Wegzug* und *Zuzug* erarbeitet.

Der Kanton Zürich realisiert nun im Projekt „eUmzugZH“ als Pilotanwender die Konzepte von „eUmzugCH“. Die Projektplanung sieht vor, dass ab Mai 2015 alle nötigen E-Services im Verbund erreichbar sind und bis Oktober 2015 Testsequenzen durchgeführt werden. Per Q4 2015 sollen dann erste echte Ereignisse produktiv über „eUmzugZH“ abgewickelt werden können.

### 3.2 Prozess und Service-Integration

Die Konzepte von „eUmzugCH“ bzw. „eUmzugZH“ sehen eine verteilte serviceorientierte Architektur (SOA) für das System vor: Die Frontend-Portale kommunizieren während des Prozesses mit mehreren benötigten E-Services zwecks Personenidentifikation, Gebäude- und Wohnungsidentifikation und Grundversicherungsprüfung (vgl. Abbildung 1). Diese E-Services sollen technisch als SOAP-basierte Webservices (gemäss eCH-Spezifikationen) realisiert werden. Besonderheit: Die Realisierung der Personenidentifikation erfolgt gefördert durch mehrere Einwohnerkontrolle-Systeme (EK-Systeme). Der Zugriff von den Portalen auf die verschiedenen Webservices erfolgt über das öffentliche Internet.

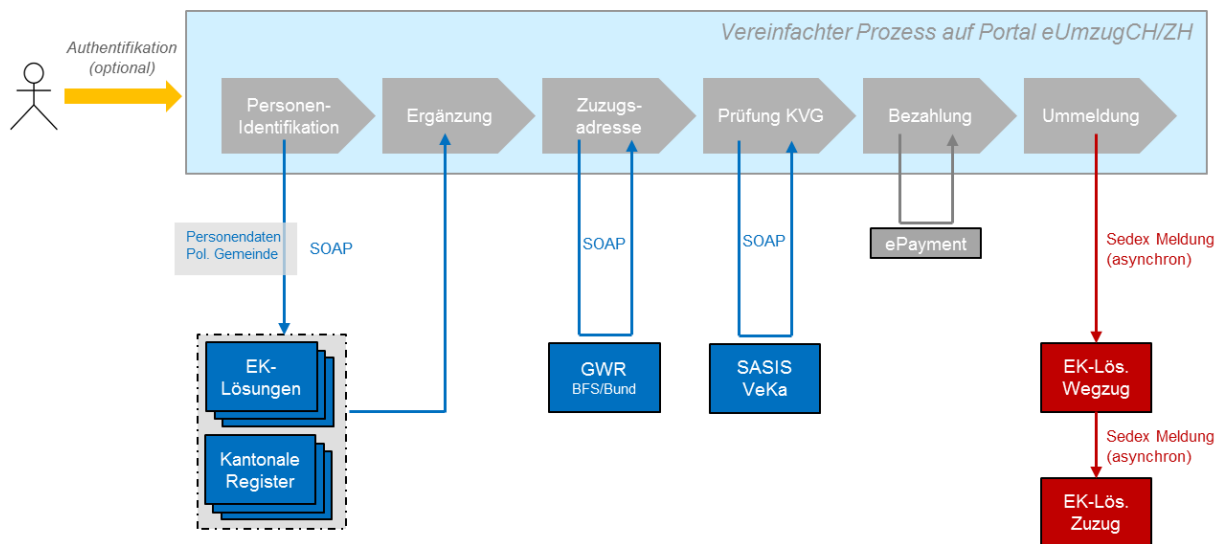


Abbildung 1: Prozess eUmzugCH/ZH und SOA-Realisierung

### 3.2.1 Zu integrierende SOAP-Webservices

#### 3.2.1.1 Gebäude- und Wohnungsregister (GWR)

Webservice des Bundesamts für Statistik (BFS), der das Abfragen des Gebäude- und Wohnungsregisters erlaubt. Damit lässt sich beispielsweise die politische Gemeindenummer einer bestimmten Wohnadresse eruieren.

#### 3.2.1.2 Krankenkassen-Versichertenkarte (SASIS)

Webservice der SASIS AG, welche die Services zur Versichertenkarte anbietet. Damit lässt sich beispielsweise abklären, ob eine Krankenkassen-Grundversicherung vorhanden ist.

#### 3.2.1.3 Personenidentifikation (PersonIdentification)

Webservices der Gemeinden und/oder Kantone, welche die Identifikation von Personen aufgrund bestimmter Merkmale (Attribute) erlauben.

## 3.3 Anforderungen an die Service-Integration

Die technische Integration der Umzugsportale mit den Webservices erfordert eine Kommunikation über das Internet und muss unter anderem folgende Anforderungen erfüllen:

- *Synchrone Kommunikation:* eUmzug-Portale (=Dienstverwender, Client) und Webservices (=Dienstbringer, Server) kommunizieren synchron (zeitgleich) unter Verwendung von SOAP über HTTPS
- *Echtzeit:* Kommunikation erfolgt interaktiv und in Echtzeit (d.h. Round-Trip-Time in der Größenordnung von maximal wenigen Sekunden ist tolerierbar)
- *Abhörsicher:* Kein Dritter darf die Kommunikation einsehen können
- *Unveränderbarkeit:* Kein Dritter darf die kommunizierten Daten verändern können
- *Authentizität Server:* Portale (Clients) müssen die Authentizität der Webservices (Server) überprüfen können
- *Authentizität Client:* Webservices (Server) müssen die Authentizität der Portale (Clients) überprüfen können
- *Autorisierung:* Webservices (Server) müssen die Autorisierung zur Nutzung durch ein Portal (Client) prüfen können
- *Ausrollung:* Infrastruktur, Credentials und Konfiguration müssen sich effizient ausrollen

- lassen
- *Aktualisierung*: Infrastruktur, Credentials und Konfiguration müssen sich effizient aktualisieren lassen
- *Vereinfachend*: Content Based Routing für föderierte Webservices soll unterstützt werden
- *Datenschutz*: Geltende Bestimmungen müssen eingehalten werden (Zugriffseinschränkungen, Aufbewahrungszeiten)

## 4 Einführung in die Services von sedex für synchrone SOAP-Webservices

Für **asynchrone** Meldungs-basierte Kommunikation mittels sedex-Meldungen verwenden die Einwohner-Kontrollregister bereits heute die Plattform sedex und haben im Normalfall bereits sedex-Clients installiert. Da sedex neben der Hauptfunktionalität des asynchronen Messagings zusätzlich **gewisse<sup>1</sup> synchrone** Kommunikationsservices unterstützt, bietet sich der Einsatz von sedex für die angestrebte Kommunikation zwischen den Portalen und den Webservices an.

### 4.1 System sedex

sedex<sup>2</sup> steht für secure data exchange und ist eine Dienstleistung des Bundesamts für Statistik BFS.

Die Plattform ist für den sicheren asynchronen Datenaustausch zwischen Organisationseinheiten konzipiert. In spezifischen Fällen erfolgt auch ein synchroner Datenaustausch. Die Plattform ist hochverfügbar (24/7).

sedex wurde im Rahmen der Modernisierung der Volkszählung ab 2010 aufgebaut, um die Statistiklieferungen der kommunalen Einwohnerdienste und der Personenregister des Bundes an das BFS sicherzustellen. Da sensitive Daten ausgetauscht werden, musste die Plattform von Beginn an hohen Anforderungen an die Sicherheit sowie Nachvollziehbarkeit genügen. Dazu setzt sedex moderne Verschlüsselungsverfahren sowie Sicherheitszertifikate der [Swiss Government PKI](#) ein.

Seit Inbetriebnahme Mitte 2008 hat sich sedex auch Teilnehmern ausserhalb der Registerharmonisierung und der Statistik geöffnet. Heute wird sedex von über 4600 Organisationseinheiten in über 40 Domänen eingesetzt. Im Jahr 2014 wurden ca. 10 Millionen Meldungen via sedex übermittelt.

sedex fungiert als Postbote und ist vergleichbar mit einem eingeschriebenen Brief.

### 4.2 sedex-Client

Bei sedex handelt es sich grundsätzlich um ein Client-Server-System. D.h. es gibt einen zentralen sedex-Server (in den RZ des BIT betrieben) und dezentrale sedex-Clients, die bei den teilnehmenden Fachapplikationen installiert sind (vgl. Abbildung 2).

Der sedex-Client ist detailliert im Handbuch [sdxBHB] beschrieben.

<sup>1</sup> Es besteht die Absicht, die Funktionen zur Unterstützung synchroner Kommunikation über sedex in Zukunft deutlich zu erweitern.

<sup>2</sup> Siehe auch [www.sedex.ch](http://www.sedex.ch), das Informations-Portal zu sedex

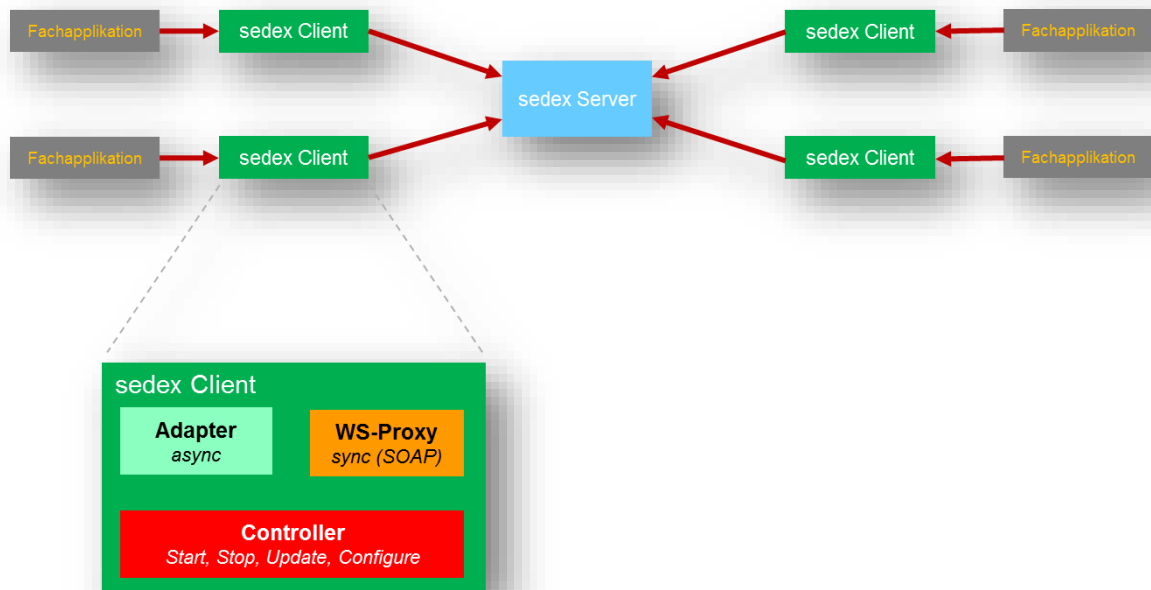


Abbildung 2: sedex-Client

Der sedex-Client umfasst heute die folgenden **drei Komponenten**:

1. *sedex-Controller*: Erlaubt Steuerung, Überwachung und Aktualisierung des sedex-Clients.
2. *sedex-Adapter*: Bildet die dateibasierte Schnittstelle für die asynchrone Meldungs-basierte Kommunikation über sedex.
3. *sedex-Webservice-Proxy*: Erlaubt den Zugriff auf Webservices (SOAP-over-HTTPS-Protokoll) unter Verwendung des sedex-Zertifikats für die SSL-/TLS-Kanalverschlüsselung.

Für die Integration der synchronen Webservices in „eUmzugCH“ kommt somit der *sedex-Webservice-proxy* (kurz WS-Proxy) in Frage.

## 4.3 sedex-Webservice-Proxy

Die Abbildung 3 zeigt das grundlegende Konzept des Webservice-Proxy. Für eine detaillierte Beschreibung des Webservice-Proxy (kurz WS-Proxy) sei auf das entsprechende Handbuch [sdxWSP] verwiesen.

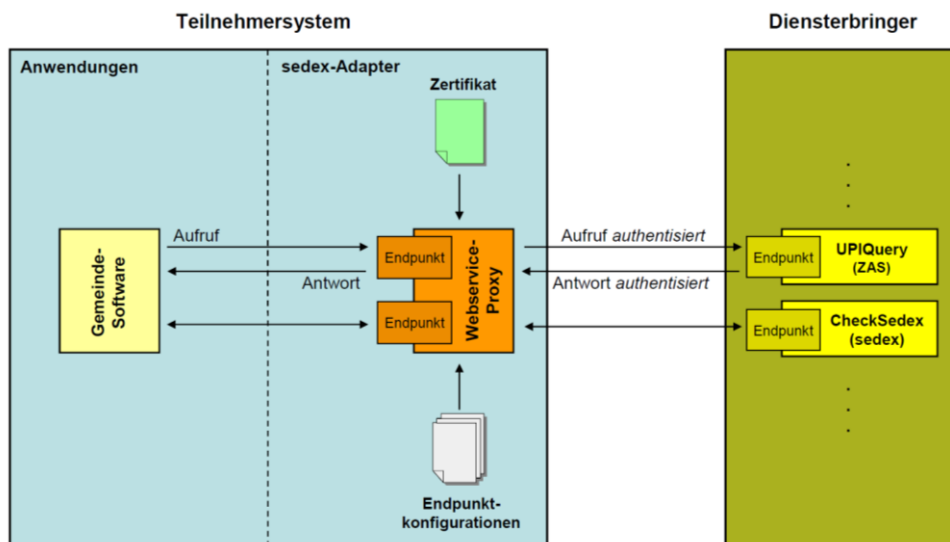


Abbildung 3: Konzept sedex-Webservice-Proxy

Die Kommunikation via Webservice-Proxy weist folgende Eigenschaften auf:

- **Proxy-Pattern:** Der geschützte Remote-Webservice (SOAP über HTTPS) wird dem Dienstverwender vom sedex-WS-Proxy lokal offen/ungeschützt (SOAP über HTTP) noch einmal angeboten
- **Authentisierung:** Der Webservice-Anbieter kann am serverseitigen Terminationspunkt der SSL-/TLS-Verbindung das für die Kommunikation verwendete sedex-Zertifikat des Clients feststellen (z.B. Attribute *Issuer*, *Serialnumber*, *Subject* aus dem Zertifikat extrahieren oder das Zertifikat als Ganzes verwenden)
- **Autorisierung:** Der Webservice-Anbieter hat für die Autorisierung basierend auf dem sedex-Zertifikat (bzw. den extrahierten Attributen) verschiedene Möglichkeiten:
  1. Implizite Autorisierung
    - a. Einschränkung auf sedex-Benutzer → ein gültiges sedex-Zertifikat gilt als implizit autorisiert
    - b. Einschränkung auf bestimmte sedex-Zertifikate → Attribut des Zertifikats muss definierten Wert aufweisen (z.B. Substring „XY“ im *Subject* des Zertifikats)
  2. Explizite Autorisierung
    - a. Manuelle Zuordnung der autorisierten sedex-Identitäten lokal beim Dienstanbieter („Whitelisting“)
    - b. Führen der Berechtigungen im System sedex (als „Authorization Rule“, welche die Rechte eines Teilnehmers auf einem Meldungstyp definiert) und Abfragen der Berechtigung eines sedex-Zertifikats über den bestehenden Webservice „sedex externalAuthorization“

## 4.4 sedex-externalAuthorization-Service

Die sichere Meldungs- und Daten-Austauschplattform sedex verfügt über ein *Teilnehmerverzeichnis* der angeschlossenen Benutzer (Amtsstellen usw.) sowie über die nötigen Informationen zu deren zertifikatsbasierter Authentisierung und Autorisierung.

Mittels der **sedex-Webservice-Proxy-Funktionalität** können IT-Systeme der sedex-Benutzer indirekt auch auf definierte Webservices Dritter zugreifen. Der sedex-Webservice-Proxy sorgt dabei dafür, dass der Benutzer mittels seines sedex-Zertifikats identifizierbar wird. Die beiden Aufgaben Authentisierung und Autorisierung bleiben hingegen Aufgabe des Dienstanbieters.

Da die für die Aufgaben Authentisierung und Autorisierung nötigen Informationen in vielen Fällen in der Plattform sedex bereits vorliegen, kann sedex diese Aufgaben der Dienstanbieter direkt unterstützen, indem es eine Zugriffsmöglichkeit auf seine Autorisierungsinformationen anbietet. Hierzu bietet sedex den SOAP-Webservice externalAuthorization an.

Der sedex-Webservice externalAuthorization ist in [sdxExtAuth] beschrieben.

## 4.5 Anforderungsabdeckung durch sedex

Erfolgt die Integration der Portale mit den Webservices unter Verwendung des sedex-Webservice-Proxy, lassen sich grundsätzlich alle in Abschnitt 3.2.1 aufgeführten Anforderungen erfüllen:

Anforderung	Sedex-Mittel zur Erfüllung
Abhörsicherheit	SSL-/TLS-Kanalverschlüsselung + bekannte Zertifikate
Unveränderbarkeit	SSL-/TLS-Kanalverschlüsselung + bekannte Zertifikate
Authentisierbarkeit Dienstverwender	Benutzung des sedex-Zertifikats des Dienstverwenders für die SSL-/TLS-Kommunikation
Authentisierbarkeit Dienstbringer	SSL-Zertifikat des Dienstbringers im Truststore des WS-Proxys des Clients
Autorisierbarkeit	sedex-Autorisierung basierend auf Teilnehmer + Meldungstyp
Ausrollbarkeit	Installationsanleitung, Support, automatisierte Zertifikatserstinstallation
Aktualisierbarkeit	Automatisierte Software-Updates, automatisierte Zertifikatserneuerung
Vereinfachung	Content Based Routing innerhalb des WS-Proxy

## 5 sedex-Lösung für SOAP-Kommunikation in eUmzugCH/ZH

### 5.1 Herausforderungen beim Einsatz von sedex für eUmzugCH

#### 5.1.1 Service zur Personenidentifikation erfordert Content Based Routing

Der Webservice zur Personenidentifikation soll nicht zentral, sondern von einer relativ grossen Anzahl (dutzenden/hundert) heterogener und dezentral organisierter EK-Systeme realisiert werden. Je nach politischer Gemeinde ist somit ein anderer bestimmter Webservice für die Personenidentifikation zuständig.

Die Kommunikation zur Personenidentifikation zwischen Client und Server erfordert also ein inhaltsbasiertes Umleiten der Anfragen (sog. *Content Based Routing*<sup>3</sup>, *CBR*).

Damit Content Based Routing angewendet werden kann, müssen folgende Voraussetzungen erfüllt sein:

1. **Content muss Routing-Kriterium enthalten**  
Eine Anfrage zur Personenidentifikation muss den Identifikator der betroffenen politischen Gemeinde enthalten. Für die Personenidentifikation wird hierzu das Feld municipalityId im SOAP-Request searchPersonIdentification verwendet.
2. **Kommunikation erfordert Service-Lookup**  
Der Webservice-Proxy muss einen Service-Lookup vornehmen, um aufgrund der im Content enthaltenen politischen Gemeinde den zuständigen technischen Ziel-Webservice bestimmen zu können.

<sup>3</sup> CBR: <http://www.itwissen.info/definition/lexikon/CBR-content-based-routing.html>



### 3. Service-Directory muss verfügbar sein

Für den Service-Lookup muss in einem Directory die für eine politische Gemeinde zuständige Webservice-Implementation definiert sein. Geführt werden im Directory konkret mindestens der Gemeinde-Identifikator („Gemeinde-Nr.“) und der zugehörige Webservice-Endpunkt (eine URI<sup>4</sup>). Aus Sicherheitsgründen sind zusätzlich noch die zu akzeptierenden <sup>5</sup>SSL-Zertifikate des Webservices (zwecks Authentifikation desselben) abgelegt.

### 4. Routing der Service-Anfrage erforderlich

Die Anfrage des Portals muss letztlich an den aus dem Service-Lookup resultierenden Webservice adressiert werden (konkret an dessen URI). Der Webservice-Proxy muss dieses Routing vornehmen.

## 5.1.2 Authentifizierung und Autorisierung seitens Dienstbringer

Wird ein Webservice via Webservice-Proxy aufgerufen, erfolgt die Kommunikation über einen mit dem sedex-Zertifikat abgesicherten SSL-/TLS-Kanal. Damit der Dienstbringer daraus einen Nutzen ziehen kann, muss er am Punkt der SSL-/TLS-Terminierung folgendes tun:

Schritt	Prüfung	deckt sedex-externalAuth dies ab?
1	Prüfen, dass das Zertifikat von der AdminPKI-CA ausgestellt (signiert) ist	Ja
2	Prüfen, dass das Zertifikat gültig ist:  a. Prüfzeitpunkt liegt innerhalb der Gültigkeitsperiode des Zertifikats b. Zertifikat wurde von der CA nicht zurückgezogen (Test mittels CRL <sup>6</sup> oder OCSP <sup>7</sup> )	Ja
3	Den Aufrufer anhand des Zertifikats identifizieren	Ja
4	Prüfen, ob der identifizierte Aufrufer zur Nutzung des Dienstes autorisiert ist	Ja

### Unterstützung durch sedex

Wird der Webservice externalAuthorization von sedex verwendet, werden alle Schritte 1 bis 4 von sedex ausgeführt. Der Dienstanbieter muss einzig das Zertifikat des Aufrufers extrahieren und den SOAP-Webservice externalAuthorization damit abfragen.

## 5.2 Lösungsüberblick

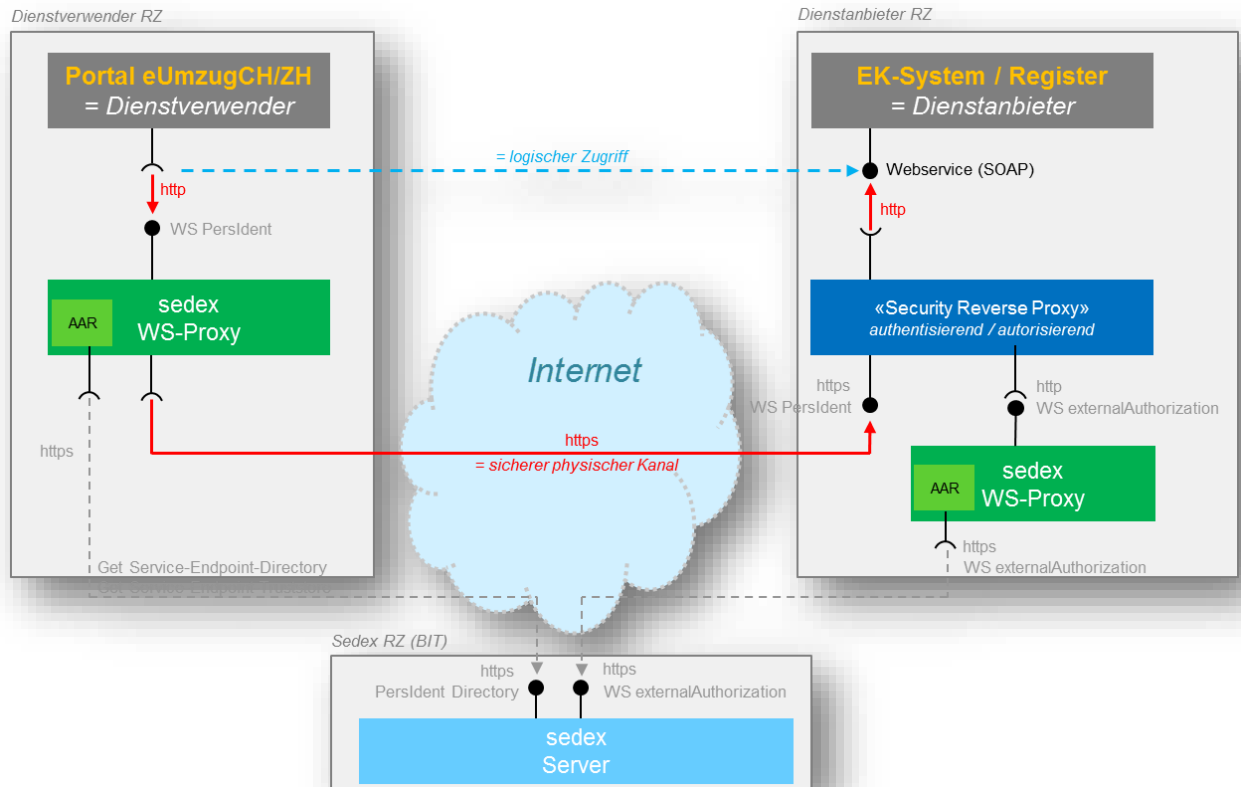
Die Abbildung 4 zeigt die selektierte sedex-Lösung im Überblick. Die einzelnen Lösungsbausteine werden in den nachfolgenden Kapiteln genauer beschrieben.

<sup>4</sup> URI: [http://de.wikipedia.org/wiki/Uniform\\_Resource\\_Identifier](http://de.wikipedia.org/wiki/Uniform_Resource_Identifier)

<sup>5</sup> Damit auch ein Zertifikatswechsel unterbrechungsfrei erfolgen kann, sollten mindestens zwei Zertifikate pro Webservice-Endpunkt geführt werden können

<sup>6</sup> Certificate Revocation List (CRL): Zertifikatssperrliste, <http://de.wikipedia.org/wiki/Zertifikatssperrliste>

<sup>7</sup> Online Certificate Status Protocol (OCSP): Online-Gültigkeitsabfrage, [http://de.wikipedia.org/wiki/Online\\_Certificate\\_Status\\_Protocol](http://de.wikipedia.org/wiki/Online_Certificate_Status_Protocol)



**Abbildung 4: Überblick synchrone Kommunikation über sedex**

*Logisch* erfolgt ein Zugriff des Dienstverwenders (hier konkret des Portals eUmzugCH/ZH) auf den Webservice des Dienstanbieters (hier konkret des EK-Systems einer Gemeinde). *Physisch* erfolgt der Zugriff aber via den lokal beim Portal installierten sedex-WS-Proxy. Dazu spiegelt dieser exakt den gleichen Webservice lokal und macht diesen dem Portal offen über das Protokoll HTTP zugänglich. Analog spiegelt er auch die weiteren Webservices des GWR und der SASIS.

Im sedex-WS-Proxy ist der Webservice als sog. CBR<sup>8</sup>-AAR<sup>9</sup> realisiert, d.h. die eingehende Anfrage des Portals wird analysiert, um basierend auf dem Inhalt den zuständigen Webservice-Endpunkt zu bestimmen. Konkret kann hierzu das im Aufruf enthaltene Merkmal `municipalityId` verwendet werden, welches die ID der Wegzugsgemeinde (politische Gemeindenummer) enthält. Die Zuordnung von dieser ID auf den zuständigen Endpunkt (eine URI<sup>10</sup>) entnimmt der WS-Proxy einem publizierten Service-Endpoint-Directory, welches auf dem sedex-Update-Server bereitgestellt wird.

Der Webservice-Proxy sendet die Anfrage an den Endpunkt des zuständigen Webservices mittels des HTTPS-Protokolls, also nach dem SSL-/TLS<sup>11</sup>-Verfahren, über einen sicheren physischen Kanal. Als SSL-Client-Zertifikat verwendet der WS-Proxy das normale sedex-Zertifikat des Teilnehmers, wodurch er sich eindeutig elektronisch ausweisen kann.

Grundsätzlich muss auf der Dienstanbieterseite der Webservice ganz normal und völlig unabhängig von sedex über SSL/TLS zugänglich gemacht werden. D.h. der Dienstanbieter besorgt sich bei einem Zertifikatsaussteller ein für seinen Endpunkt gültiges SSL-Zertifikat. Die Aufgabe des Dienstanbieters ist nun die Terminierung der SSL-Verbindung und der Authentifizierung sowie Autorisierung der aufrufenden Teilnehmer. Üblicherweise übernimmt diese Aufgabe ein beliebiger vorgeschalteter Reverse-Proxy, die Autorisierungsantwort kann er sich dabei beim sedex-Server abholen. Der Reverse-Proxy

<sup>8</sup> Content Based Routing (CBR)

<sup>9</sup> Apache Axis Archive (AAR) – Definitionspaket für einen SOAP-Webservice in Apache Axis

<sup>10</sup> Uniform Resource Indicator ([http://de.wikipedia.org/wiki/Uniform\\_Resource\\_Identifier](http://de.wikipedia.org/wiki/Uniform_Resource_Identifier))

<sup>11</sup> Secure Socket Layer / Transport Level Security ([http://de.wikipedia.org/wiki/Transport\\_Layer\\_Security](http://de.wikipedia.org/wiki/Transport_Layer_Security))

leitet nur authentifizierte und autorisierte Anfragen an den effektiven Webservice weiter – alle anderen blockiert er und sorgt so für Sicherheit beim Zugriff auf den Service.

## 6 Integration auf Seite Dienstverwender

Dieses Kapitel zeigt die nötigen Schritte auf, damit die SOAP-Webservices auf Seite Dienstverwender (konkret also von den Umzugsportalen) indirekt über den sedex-Webservice-Proxy verwendet werden können.

Die Abbildung 5 zeigt die wesentlichen Bausteine der Lösung auf Seite Dienstverwender im Überblick auf.

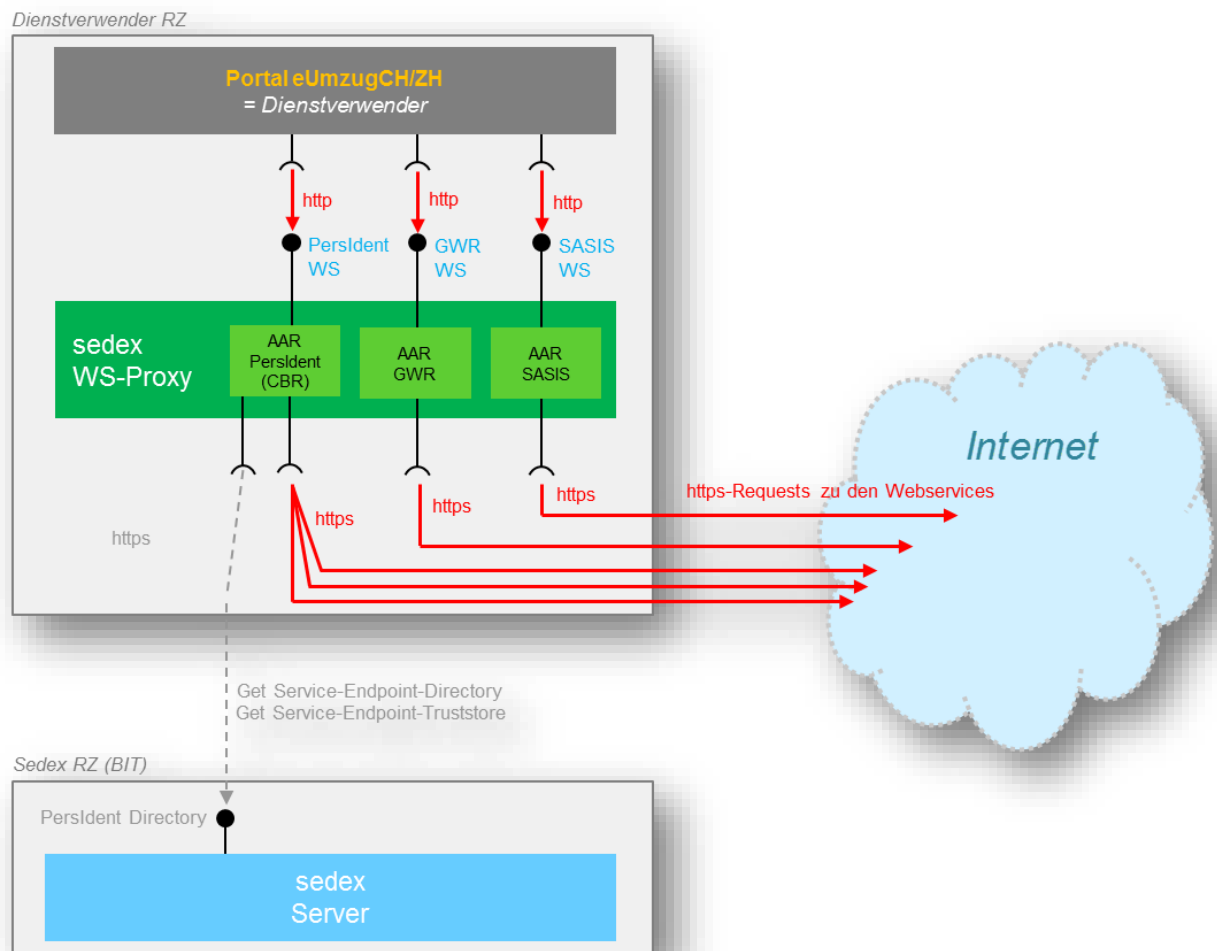


Abbildung 5: Bausteine auf Seite Dienstverwender

### 6.1 Vorbereitende Arbeiten/Organisatorisches

#### 6.1.1 sedex-Teilnehmer bestellen und einrichten

Damit der Dienstverwender sedex überhaupt nutzen kann, muss er im System sedex erst mal als *Teilnehmer* bekannt sein und eine *Teilnehmerkennung*, eine sogenannte „*sedexId*“ zugewiesen erhalten. Ebenfalls müssen in sedex die nötigen Berechtigungen für den Teilnehmer so gesetzt werden, dass er die Webservices nutzen darf.

Neue Teilnehmer kann das Bundesamt für Statistik (BFS) als fachlicher Betreiber des Dienstes sedex einrichten und konfigurieren. Im Kontext von eUmzugCH erfolgt die Anmeldung für den Dienst sedex aber indirekt über den sedex-Domänenverantwortlichen von eUmzugCH. Dieser bestellt den Teilnehmer dann seinerseits beim sedex-Kundendienst des BFS.

Das System sedex erstellt für einen neuen Teilnehmer bzw. beim Bestellen eines neuen Teilnehmerzertifikats Credentials, die aus folgenden zwei Komponenten bestehen:

1. **Certificate Request ID (CRID)**

Die Certificate Request ID (CRID) ist die eindeutige ID des Prozesses zur Erstellung eines neuen Teilnehmerzertifikats. Diese wird via Domänenverantwortlichen an den neuen sedex-Teilnehmer weitergeleitet.

2. **One Time Password (OTP)**

Das One Time Password (OTP) ist ein Einmalpasswort zum Starten des Prozesses zur Erstellung eines neuen Teilnehmerzertifikats. Dieses wird vom System sedex als E-Mail direkt an die E-Mail-Adresse des neuen Teilnehmers übermittelt.

Hinweis: Das OTP sollte niemandem ausser dem neuen Teilnehmer bekannt sein. Nach der ersten Installation ist es verwirkt.

## 6.2 Installation sedex-Webservice-Proxy

Damit der sedex-Webservice-Proxy auf Seite Dienstverwender genutzt werden kann, muss dieser als Teil des sedex-Clients auf einem Host innerhalb einer geschützten Netzwerkzone installiert sein, so dass das Umzugsportal auf den Webservice-Proxy zugreifen kann.

Der sedex-Webservice-Proxy ist während der Installation des sedex-Clients optional. Da er hier genutzt werden soll, muss diese Option zwingend selektiert werden. Soll ein bereits bestehender sedex-Client verwendet werden, muss bei diesem überprüft werden, ob die Option Webservice-Proxy installiert ist.

Die Installationsanleitung [sdxIHB] für den sedex-Client, die Benutzerhandbücher [sdxBHB] und [sdxWSP] sowie die Installationspakete des sedex-Clients sind auf dem sedex-Portal [www.sedex.ch](http://www.sedex.ch) verfügbar.

Um die Installation durchführen zu können, braucht der zuständige Techniker:

- |  |  |
|--|--|
| 1. Eines der sedex-Installationspakete | von <a href="http://www.sedex.ch">www.sedex.ch</a> |
| 2. sedexId (Teilnehmerkennung)         | vom Domänenadministrator                           |
| 3. Certificate Request ID (CRID)       | vom Domänenadministrator                           |
| 4. One Time Password (OTP)             | per E-Mail   |

Für die konkrete Installation des sedex-Webservice-Proxy verweisen wir auf die separate Installationsanleitung [sdxWSP] auf [www.sedex.ch](http://www.sedex.ch).

### 6.2.1 Besondere Hinweise im Kontext eUmzugCH

Die separate Installationsanleitung [sdxWSP] enthält bereits viele Informationen und Hinweise zum Webservice-Proxy. Wir empfehlen eine detaillierte Lektüre. Die folgenden Punkte enthalten darüber hinaus spezifische Hinweise für den Einsatz im Kontext eUmzugCH auf Dienstverwenderseite:

- [zur Zeit keine]

## 6.3 Installation AAR-Konfigurationspakete der angebotenen Webservices

Die Installation eines (neuen) Webservices in den Webservice-Proxy erfordert in der Regel die Installation einer neuen *AAR-Datei* und eines neuen *SSL-Truststores*. Die dazu nötigen Schritte sind den spezifischen Anleitungen (README-Dateien) zu jedem AAR zu entnehmen, sind aber nachfolgend auch noch einmal kurz zusammengefasst.

### 6.3.1 Webservice-Definitionen/AAR-Dateien

Für jeden SOAP-Webservice, welcher via den sedex-Webservice-Proxy genutzt werden soll, muss im Installationsverzeichnis des WS-Proxy je eine sogenannte AAR-Datei (Apache Axis Archive) vorhanden sein. AAR-Dateien enthalten die Definition des Webservice (WSDL) und die zugehörige Weiterleitungslogik.

Für die folgenden im Prozess eUmzugCH eingesetzten SOAP-Webservices wird je eine AAR-Datei benötigt, die installiert werden muss:

- **AAR zum Webservice Personenidentifikation**  
Dieses Modul stellt einen Webservice zur Personenidentifikation zur Verfügung und realisiert das Content Based Routing auf die verschiedenen Endpunkte zur Personenidentifikation. Zu diesem AAR gibt es ein dediziertes Benutzerhandbuch [sdxAARPersIdent].
- **AAR zum Webservice Gebäude und Wohnungsregister**  
Dieses Modul stellt einen Webservice zur Abfrage des Gebäude- und Wohnungsregisters zur Verfügung. Dieser zentrale Webservice wird vom BFS angeboten.
- **AAR zum Webservice SASIS / Versichertenkarte**  
Dieses Modul stellt einen Webservice zur Abfrage der Krankassen-Grunddeckung zur Verfügung. Dieser zentrale Webservice wird von SASIS/Versichertenkarte angeboten.



#### Hinweis:

Das Benutzerhandbuch des Webservice-Proxy [sdxWSP] enthält zu jedem AAR spezifische Informationen bezüglich seiner Verwendung (URL, Beispielrequests usw.).

### 6.3.2 Installation einer neuen AAR-Datei:

Einige der AAR-Dateien werden bereits mit dem sedex-Client ausgeliefert und installiert. Es kann aber sein, dass auf [www.sedex.ch](http://www.sedex.ch) neuere Versionen existieren. Es ist empfohlen, die jeweils neueste Version zu verwenden.

1. **AAR-Datei zum gewünschten Webservice von [www.sedex.ch](http://www.sedex.ch) herunterladen**  
Die jeweils aktuelle AAR-Datei kann zusammen mit einer spezifischen Kurzanleitung (README) zur Installation und Benutzung des Webservice unter folgender URL heruntergeladen werden: [www.sedex.ch](http://www.sedex.ch) > „Synchron“
2. **AAR-Datei des Webservice in den Webservice-Proxy kopieren**  
Gemäss der spezifischen Anleitung (README) muss die-AAR Datei in die sedex-Webservice-Proxy-Installation kopiert werden. Der Pfad lautet i.d.R. <sedexClient>/adapter/axis2/repository /services.

### 6.3.3 Installation eines neuen SSL-Truststore

Damit „Man-in-the-Middle-Attacks“<sup>12</sup> ausgeschlossen werden können, traut der sedex-Client nur Webservices, deren SSL-Zertifikate explizit in seinem Truststore aufgeführt sind. Wenn der Endpunkt des implementierenden Webservices für sedex neu ist, kann es somit erforderlich sein, ebenfalls einen neuen (um das neue SSL-Zertifikat ergänzten) Truststore herunterzuladen und in den sedex-Client zu installieren.

Das Thema Truststore ist ebenfalls in der spezifischen Anleitung (README) zu einer AAR-Datei beschrieben. Nachfolgend aber das grundsätzliche Vorgehen zur Erneuerung des Truststores.

Der Truststore besteht aus zwei Dateien (beide müssen erneuert werden):

- wsproxytrust.jks
- wsproxytrust.jks.properties

Installation eines neuen Truststores:

1. **Truststore von [www.sedex.ch](http://www.sedex.ch) herunterladen**

Die Dateien des Truststores können i.d.R. hier heruntergeladen werden:  
[www.sedex.ch](http://www.sedex.ch) > Downloads > Truststores

2. Die beiden Truststore-Dateien sind i.d.R. hier in den sedex-Client zu kopieren:  
<sedexClient>/adapter/certificate/prod-bit

## 6.4 Content Based Routing für Personenidentifikation

Das AAR für die Personenidentifikation unterscheidet sich grundsätzlich von den anderen AAR. Normalerweise leitet ein AAR alle eintreffenden Webservice-Requests an einen einzigen vordefinierten Endpunkt weiter, an den „richtigen“ Webservice des Diensteanbieters. Bei der Personenidentifikation ist das anders: Je nach politischer Wohngemeinde ist eine andere Instanz des Webservices für die Beantwortung eines Requests zuständig. Hierzu findet im AAR für die Personenidentifikation ein sogenanntes Content Based Routing (CBR) statt.

In diesem Abschnitt sind die Besonderheiten des Content Based Routing festgehalten.

- Die Logik innerhalb des AAR für die Personenidentifikation realisiert das Content Based Routing. D.h. der Webservice-Proxy muss direkt auf alle Instanzen des Webservice Zugriff haben.
- Das AAR holt sich automatisch vom sedex- Server die für das CBR notwendigen Daten:
  - *Service-Endpoint-Directory*  
Hierbei handelt es sich um eine einfache Liste, welche das Mapping vom Routing-Merkmal (z. B. Politische-Gemeinde-ID) auf den Endpunkt des zuständigen Webservices (z. B. eine URI) definiert.
  - *Service-Endpoint-Truststore*  
Hierbei handelt es sich um eine Truststore-Datei im Java-Keystore-Format (JKS). Diese Datei enthält die Zertifikate der zugelassenen SSL-Endpunkte.
- Die CBR-Daten (Routing-Tabelle, Truststore) werden automatisch vom sedex-Server heruntergeladen und aktualisiert:
  - Beim Neustart des sedex-Webservice-Proxy
  - Alle 24 Stunden im laufenden Betrieb

---

<sup>12</sup> <https://de.wikipedia.org/wiki/Man-in-the-Middle-Angriff>

Weitere Informationen können [sdxAARPersIdent] entnommen werden.

## 6.5 Weitere technische Anforderungen

Auf der Seite des Dienstverwenders müssen folgende technischen Anforderungen besonders beachtet werden:

- **Ermöglichung folgender Netzwerkzugriffe**
  - Das Umzugsportal muss für die Nutzung der Webservices auf den sedex-Webservice-Proxy zugreifen können. D.h. eine Kommunikation mit dem Protokoll HTTP auf den Endpunkt des Webservice-Proxy (Default-Einstellung ist 8080) muss möglich sein.
  - Der Webservice-Proxy muss Zugriff auf alle Endpunkte der unterstützten Webservices erhalten. D.h. eine Kommunikation mit dem Protokoll HTTPS auf die Endpunkte der Webservices (meist Port 443) muss möglich sein.
  - Der Webservice-Proxy muss Zugriff auf die Endpunkte des sedex-Servers erhalten. D.h. eine Kommunikation mit dem Protokoll HTTPS auf die Endpunkte von sedex (Port 443) muss möglich sein. Konkret sind dies:
    - Sedex-Client bis Version 4.0.4 (5 Einträge):
      - `www.sedex-gw.admin.ch`, Port 443, Protokoll HTTPS
      - `www.oscitr-gw.admin.ch`, Port 443, Protokoll HTTPS
      - `www.osciseservices-gw.admin.ch`, Port 443, Protokoll HTTPS
      - `www.governikus.admin.ch`, Port 80, Protokoll HTTP
      - `sedex-service.admin.ch`, Port 443, Protokoll HTTPS
    - sedex-Client ab Version 5.0 (nur ein einziger Eintrag):
      - `sedex-service.admin.ch`, Port 443, Protokoll HTTPS

## 6.6 Nutzung der Webservices via Webservice-Proxy

Ist der sedex-Webservice-Proxy installiert und konfiguriert, kann das Umzugsportal die Webservices via Webservice-Proxy nutzen. Natürlich kann der Proxy auch mit einem geeigneten Testwerkzeug wie z. B. SoapUI überprüft werden.

Der Webservice-Client muss auf die lokalen Adressen der Services auf dem Webservice-Proxy zugreifen. Die Endpunkte können der jeweiligen Installationsanleitung zu einem AAR (README) entnommen werden.

Normalerweise sind dies die nachfolgenden lokalen URLs:

- **Webservice Personen-Identifikation:**  
Endpunkt-URL: `http://<wsproxyhost>:<wsproxyport>/wsproxy/services/PersonIdentificationService`  
WSDL-URL: `http://<wsproxyhost>:<wsproxyport>/wsproxy/services/PersonIdentificationService?wsdl`
- **Webservice Gebäude und Wohnungsregister:**  
Endpunkt-URL: `http://<wsproxyhost>:<wsproxyport>/wsproxy/services/thirdPartyNotificationService`  
WSDL-URL: `http://<wsproxyhost>:<wsproxyport>/wsproxy/services/thirdPartyNotificationService?wsdl`
- **Webservice SASIS / Versichertenkarte:**  
Endpunkt-URL: `http://<wsproxyhost>:<wsproxyport>/wsproxy/services/VeKa_Query`

WSDL-URL: `http://<wsproxyhost>:<wsproxyport>/wsproxy/services/VeKa_Query?wsdl`



**Hinweise:**

Das Benutzerhandbuch des Webservice-Proxy [sdxWSP] enthält Informationen zur Nutzung aller Webservices via Webservice-Proxy (URL, Beispiel-Requests usw.)

Für den Webservice Personenidentifikation besteht zudem ein spezifisches Benutzerhandbuch [sdxAARPersIdent] mit weitergehenden Informationen.



## 7 Integration auf Seite Dienstanbieter

Dieses Kapitel führt die Schritte auf, die nötig sind, damit die SOAP-Webservices auf Seite Dienstanbieter (konkret also bei den Gemeinde- oder Kantonsregistern) für die clientseitige Nutzung durch den sedex-Webservice-Proxy zur Verfügung gestellt werden können.

Die Abbildung 6 zeigt die Bausteine auf Seite Dienstanbieter allgemein auf hohem Abstraktionsniveau. Später werden diese Bausteine in drei konkreten Szenarien verfeinert.

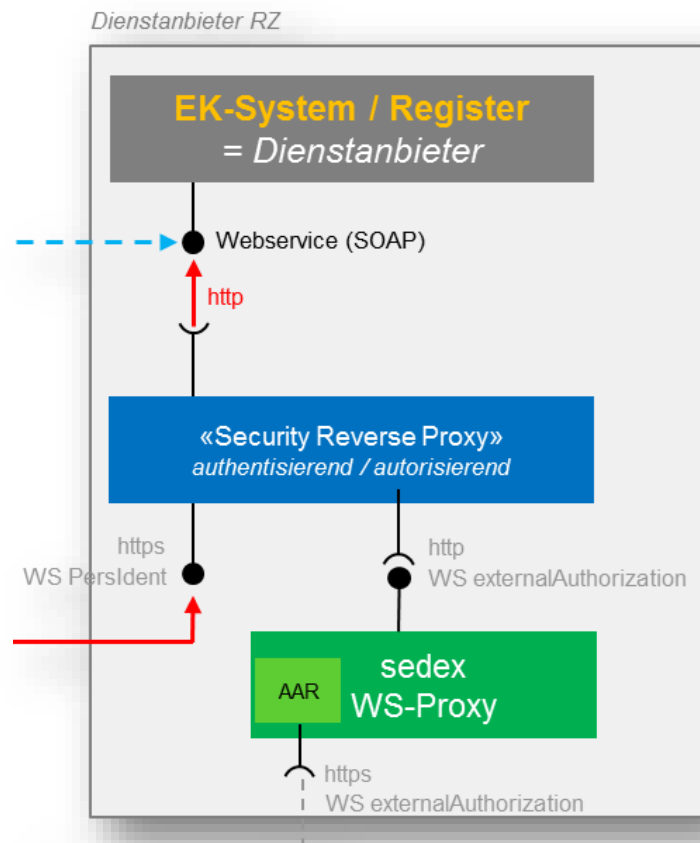


Abbildung 6: Bausteine auf Seite Dienstanbieter (abstrahiert)

Eintreffende SOAP-Requests werden mittels SSL/TLS verschlüsselt und über das HTTPS-Protokoll übertragen. Hierzu muss der Dienstanbieter wie üblich selber einen SSL-Endpunkt für seinen Webservice einrichten und sich von einem Zertifikatsaussteller für seinen Endpunkt gültige SSL-Zertifikate ausstellen lassen und diese auf seinem SSL-Endpunkt installieren.

Hinweis: sedex-Zertifikate spielen bei der SSL-Terminierung auf der Dienstanbieterseite keine Rolle. Grundsätzlich sind sedex-Zertifikate nicht für den Einsatz als SSL-Zertifikate geeignet, denn es fehlen ihnen hierzu wichtige Elemente. SSL-Zertifikate muss der Dienstanbieter also immer selber beschaffen und installieren.

Beim Dienstanbieter muss diese SSL-/TLS-Verbindung terminiert (ausgepackt) werden. Üblicherweise passiert das auf einer Komponente mit der Funktion eines „Security Reverse Proxy“. Konkret kann dies z.B. ein Loadbalancer, eine Web Application Firewall (WAF), ein Entry Server oder ein Web Service Gateway (WSG) sein. Wie die genaue Topologie im Eingangsbereich aussieht, variiert von Anbieter zu Anbieter.

Zentral für die Funktionsweise ist, dass am SSL-Terminationspunkt das Client-Zertifikat gewonnen und als zusätzliches HTTP-Header-Feld für die nachfolgenden Verarbeitungsschritte an den HTTP-

Request angefügt wird.

Entweder erfolgen Authentisierung und Autorisierung direkt auf dem SSL-Endpunkt (vgl. oben) oder in einer nachgelagerten Komponente. In Abhängigkeit von Infrastruktur, Richtlinien und Möglichkeiten eines Diensteanbieters ergeben sich somit verschiedene Szenarien, von denen drei typische als Musterlösungen nachfolgend vorgestellt werden.

In allen Realisierungs-Szenarien bietet sedex für die Authentisierung und Autorisierung einen SOAP-Webservice „External Authorization“ an. Dieser bestimmt aufgrund des nachfolgenden Datentripels, ob der Zugriff auf den Service berechtigt ist oder nicht (boolesche Antwort):

- sedex-Teilnehmer auf Seite Service-Anbieter (dessen sedexID, konstant)
- sedex-Teilnehmer auf Seite Service-Aufrufer (dessen Zertifikat aus dem SSL-Request, variabel)
- Service-Typ (d.h. die ID des zugeordneten sedex-Meldungstyps, konstant)

Der sedex-Webservice External Authorization ist selber auch wiederum mit sedex-Mitteln abgesichert, so dass Nutzer via ihren lokalen Webservice-Proxy darauf zugreifen müssen.

## 7.1 Szenario 1: „Nur bestehender Security Reverse Proxy“

In diesem Szenario erfolgen die Schritte *Authentisierung* und *Autorisierung* eines eintreffenden Requests in einem beim Diensteanbieter bereits bestehenden Netzwerkbaustein mit der Aufgabe eines Security Reverse Proxys. Typischerweise handelt es sich hierbei um sog. Web Application Firewalls (WAF) oder Webservice Gateways (WSG).

Die Abbildung 7 zeigt das Szenario 1 schematisch auf. In gelb-roter Markierung sind die Elemente dargestellt, welche durch den Diensteanbieter im bestehenden Security Reverse Proxy realisiert werden müssen. Die nötigen Schritte sind nachfolgend genauer beschrieben.

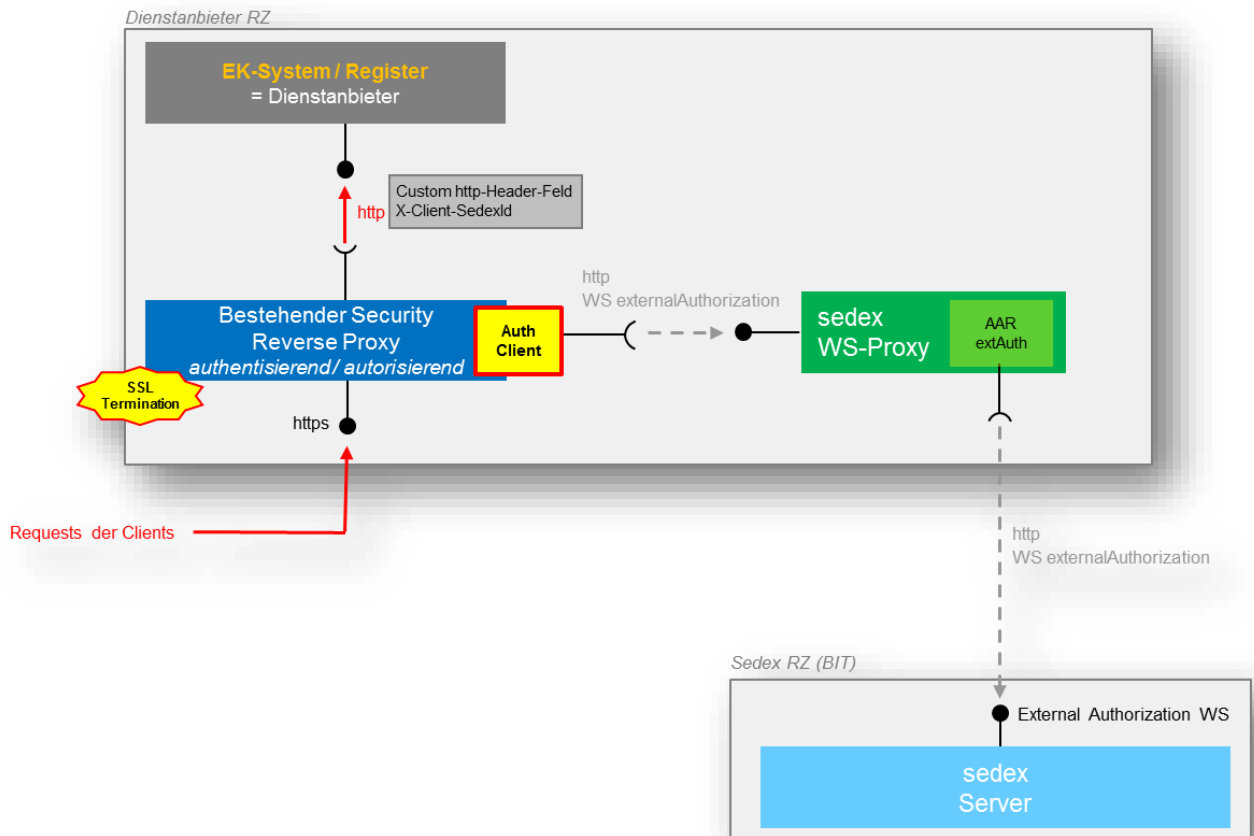


Abbildung 7: Szenario "Nur bestehender Security Reverse Proxy"

Der bestehende **Security Reverse Proxy** muss so erweitert bzw. konfiguriert werden, dass er folgende Arbeitsschritte durchführen kann

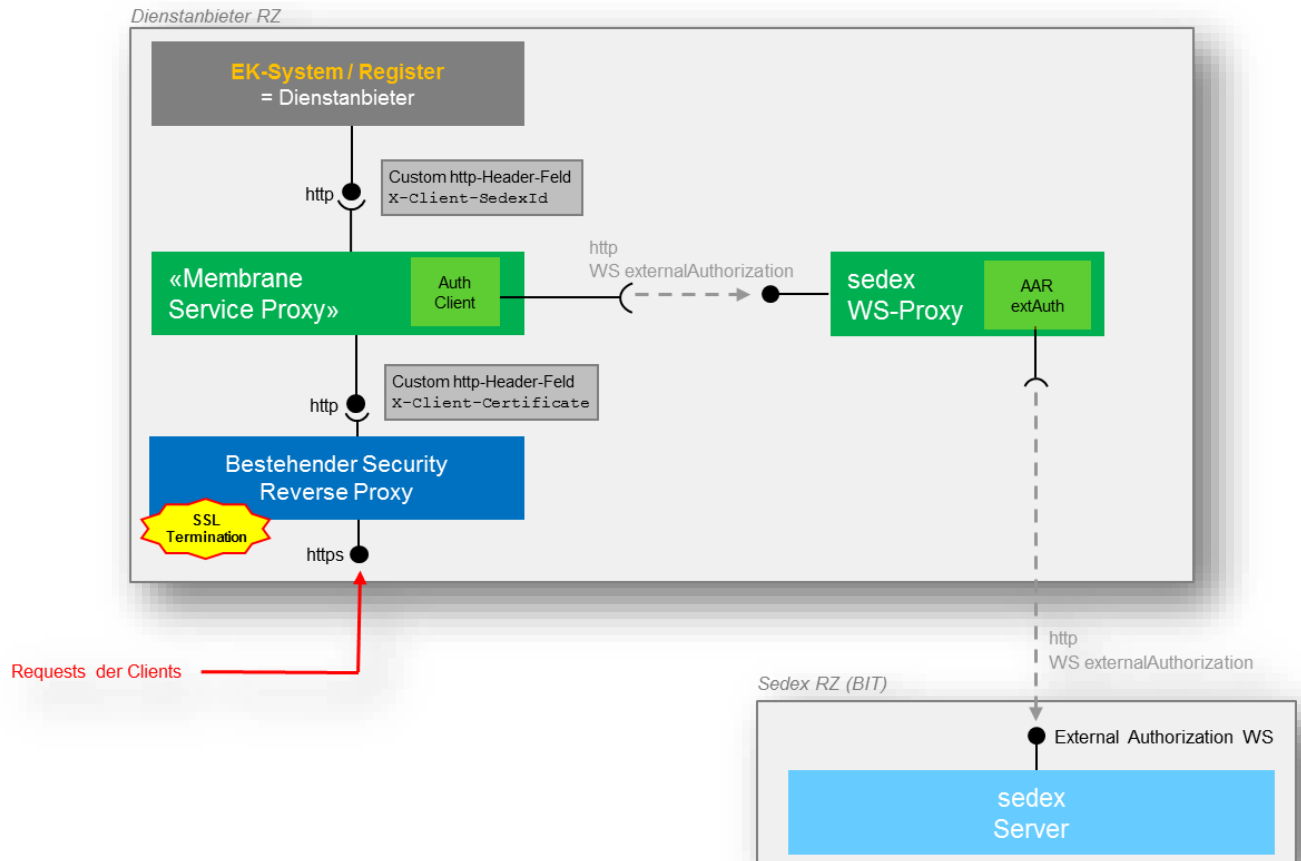
1. Terminieren der eingehenden SSL-Verbindung. Hierzu ist vom Dienstleister selber ein für den Endpunkt gültiges SSL-Zertifikat zu beschaffen und zu installieren.
2. Extrahieren des Client-SSL-Zertifikats
3. Aufrufen des Webservices externalAuthorisation (via lokal installierten sedex-Webservice-Proxy) unter Verwendung des extrahierten Client-Zertifikats als einem der Autorisierungsparameter (siehe 7.6.4)
4. Optional:  
Zusätzliches Überprüfen des Aufrufers gegen eine lokal definierte Whitelist mit vom Betreiber spezifisch zugelassenen sedexIds
5. Optional:  
Die ermittelte sedexId des Clients kann als zusätzliches HTTP-Headerfeld an die Webservice-Implementation mitgegeben werden (siehe 7.6.2)
6. Berechtigte Requests werden an die Webservice-Implementation weitergereicht

## 7.2 Szenario 2: „Bestehender Security Reverse Proxy und Membrane“

In diesem Szenario erfolgen die Schritte *Authentisierung* und *Autorisierung* eines eintreffenden Requests im zusätzlichen Netzwerkbaustein Membrane Service Proxy, der eine frei verfügbare Open-Source-Lösung ist (vgl. Kapitel 8). Ein spezifisches von sedex zur Verfügung gestelltes Plugin realisiert zusammen mit dem sedex-Webservice-Proxy die Aufgaben Authentisierung und Autorisierung,

so dass der vorhandene Security Reverse Proxy bzw. Entry Server nur noch die SSL-Verbindung terminieren und das Client-SSL-Zertifikat als HTTP-Headerfeld an Membrane weiterreichen muss.

Die Abbildung 8 zeigt das Szenario 2 schematisch auf. In gelb-roter Markierung sind die Elemente dargestellt, welche im bestehenden Security Reverse Proxy durch den Dienstanbieter realisiert werden müssen.



**Abbildung 8: Szenario „Bestehender Security Reverse Proxy und Membrane“**

Der bestehende **Security Reverse Proxy** muss noch so erweitert bzw. konfiguriert werden, dass er folgende Arbeitsschritte durchführen kann:

1. Terminieren der eingehenden SSL-Verbindung. Hierzu ist vom Dienstanbieter selber ein für den Endpunkt gültiges SSL-Zertifikat zu beschaffen und zu installieren.
2. Extrahieren des Client-SSL-Zertifikats
3. Einfügen des extrahierten Zertifikats als zusätzliches HTTP-Headerfeld des Requests (siehe 7.6.1)

Der **Membrane Service Proxy** kann folgende Arbeitsschritte durchführen:

1. Extrahieren des Client-SSL-Zertifikats aus dem HTTP-Header
2. Aufrufen des Webservices externalAuthorisation (via lokal installierten sedex Webservice-Proxy) unter Verwendung des extrahierten Client-SSL-Zertifikats als einen der Autorisierungsparameter. Details zum Webservice externalAuthorization sind [sdxEAuth] zu entnehmen. Das Ergebnis enthält u. a. die sedexId des Aufrufers
3. Optional:  
Zusätzliche Überprüfen des Aufrufers gegen eine lokal definierte Whitelist mit vom Betreiber spezifisch zugelassenen sedexIds
4. Die in Schritt 2 ermittelte sedexId des Clients wird als zusätzliches HTTP-Headerfeld an die Webservice-Implementation mitgegeben (siehe 7.6.2)

5. Berechtigte Requests werden an die Webservice-Implementation weitergereicht

### 7.2.1 Konfigurieren des Security Reverse Proxy

Das Szenario 2 geht davon aus, dass seitens Dienstanbieter ein Security Reverse Proxy zur Verfügung steht, welcher die vom Internet her eintreffenden HTTPS-Requests entgegen nimmt und als http-Headerfeld auf die interne Serverinfrastruktur weiterreicht.

Die Konfiguration des konkret eingesetzten Produkts ist gemäss Anleitung des jeweiligen Herstellers vorzunehmen. Beispielhaft ist in Tabelle 1 die Konfiguration mit einem vorgeschalteten Citrix Netscaler aufgezeigt. Produkte anderer Hersteller dürften sich in ähnlicher Weise konfigurieren lassen.

Parameter	Beschreibung
Operation	insert_http_header
Target	X-Client-Certificate
Value	CLIENT.SSL.CLIENT_CERT.TO_PEM

Tabelle 1: Rewrite Action für Citrix Netscaler

### 7.2.2 Konfigurieren des Membrane Service Proxy

Im Szenario 2 muss der Membrane Service Proxy installiert und konfiguriert werden. Details sind dem Kapitel 8 zu entnehmen, das spezifisch auf Membrane eingeht.

Nachfolgend sind darüber hinaus wichtige Elemente einer entsprechenden Membrane-Konfiguration für Szenario 2 aufgezeigt und erläutert.

Das nachfolgende Beispiel zeigt die Konfiguration für einen Membrane Service Proxy, welcher:

- Auf Port 80 auf eingehende HTTP-Verbindungen wartet
- Nur Verbindungen des Pfades /PersonIdentificationService/PersonIdentification akzeptiert
- Das WSDL umschreibt auf die externe URL „https://personenident.bern.ch“
- Den Aufruf weiterleitet an http://vmpersiden01 auf Port 8080

```

<spring:beans xmlns="http://membrane-soa.org/proxies/1/"
  xmlns:spring="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
  http://membrane-soa.org/proxies/1/ http://membrane-soa.org/schemas/proxies-1.xsd">

  <spring:bean id="extAuthInterceptor" class="ch.sedex.membrane.interceptor.externalauthorization.ExternalAuthInterceptor">
    <spring:property name="sedexId" value="a-bbbbbbb-c" />
    <spring:property name="messageType" value="1006" />
    <spring:property name="url" value="HTTP://localhost:8080/wsproxy/services/sedexExternalAuthorisationService/" />
    <spring:property name="cacheTimeInMinutes" value="30" />
  </spring:bean>

  <router>
    <serviceProxy port="80">
      <path>/PersonIdentificationService/PersonIdentification</path>
      <wsdlRewriter protocol="https" host="personenident.bern.ch" />
      <interceptor refid="extAuthInterceptor" />
      <target host="vmpersident01" port="8080"/>
    </serviceProxy>
  </router>
</spring:beans>

```

### Bedeutung der Parameter:

Parameter	Beschreibung
<spring:property name="sedexId">	Die sedexId des SOAP-Webservice-Providers. Üblicherweise ist dies die sedexId, welche im sedex-Client des Dienstanbieters konfiguriert ist.
<spring:property name="messageType">	Die sedex-Meldungstyp-ID, welche dem abzusichernden Webservice zugeordnet ist. Für PersonIdentification ist dies der Wert 1006.
<spring:property name="url">	URL des sedex-externalAuthorization-Webservice. Da via sedex-WS-Proxy darauf zugegriffen wird, ist dies normalerweise der entsprechende Endpunkt auf dem lokalen sedex-WS-Proxy.
<spring:property name="cacheTimeInMinutes">	Anzahl Minuten, für welche ein Autorisierungsergebnis aus dem Cache des Interceptors verwendet wird, bevor erneut eine Autorisierungs-Anfrage gestellt wird.
<serviceProxy port>	Der Port, unter welchem Membrane auf eingehende Verbindungen hört.
<path>	Wenn dieser Parameter gesetzt wird, werden nur auf diesem Pfad eingehende Requests akzeptiert. Der Wert sollte dem Pfad der Implementierung des Personenidentifikations-Service entsprechen.
<wsdlRewriter protocol>	Protokoll, unter dem der Service im Internet exponiert ist.
<wsdlRewriter host>	Hostname nach aussen, unter welchem Membrane vom KTV / Internet her verfügbar ist. Entspricht dem Hostname des bestehenden Security Reverse Proxy. Dieser Parameter wird für das WSDL-Rewriting verwendet.
<target host>	Hostname, auf dem die Implementierung des Personenidentifikations-Services läuft.

<target port>	Port, auf dem die Implementierung des Personenidentifikations-Services läuft
---------------	--

### 7.3 Szenario 3: „Nur Membrane“

Dieses Szenario setzt einzig einen Membrane Service Proxy ein. Das Szenario verändert das vorangehende Szenario 2 dahingehend, dass hier der Membrane Service Proxy zusätzlich auch noch Endpunkt der SSL-Verbindung ist und diese terminiert.

Wenn immer möglich, sollte ein spezieller und gehärteter Security Reverse Proxy vorgeschaltet sein (Szenarien 1 und 2). Wenn es nicht anders geht, kann aber dieses Szenario (allenfalls auch nur temporär) gewählt und umgesetzt werden.

Die Abbildung 9 zeigt das Szenario 3 schematisch auf. Die nötigen Schritte sind nachfolgend genauer beschrieben.

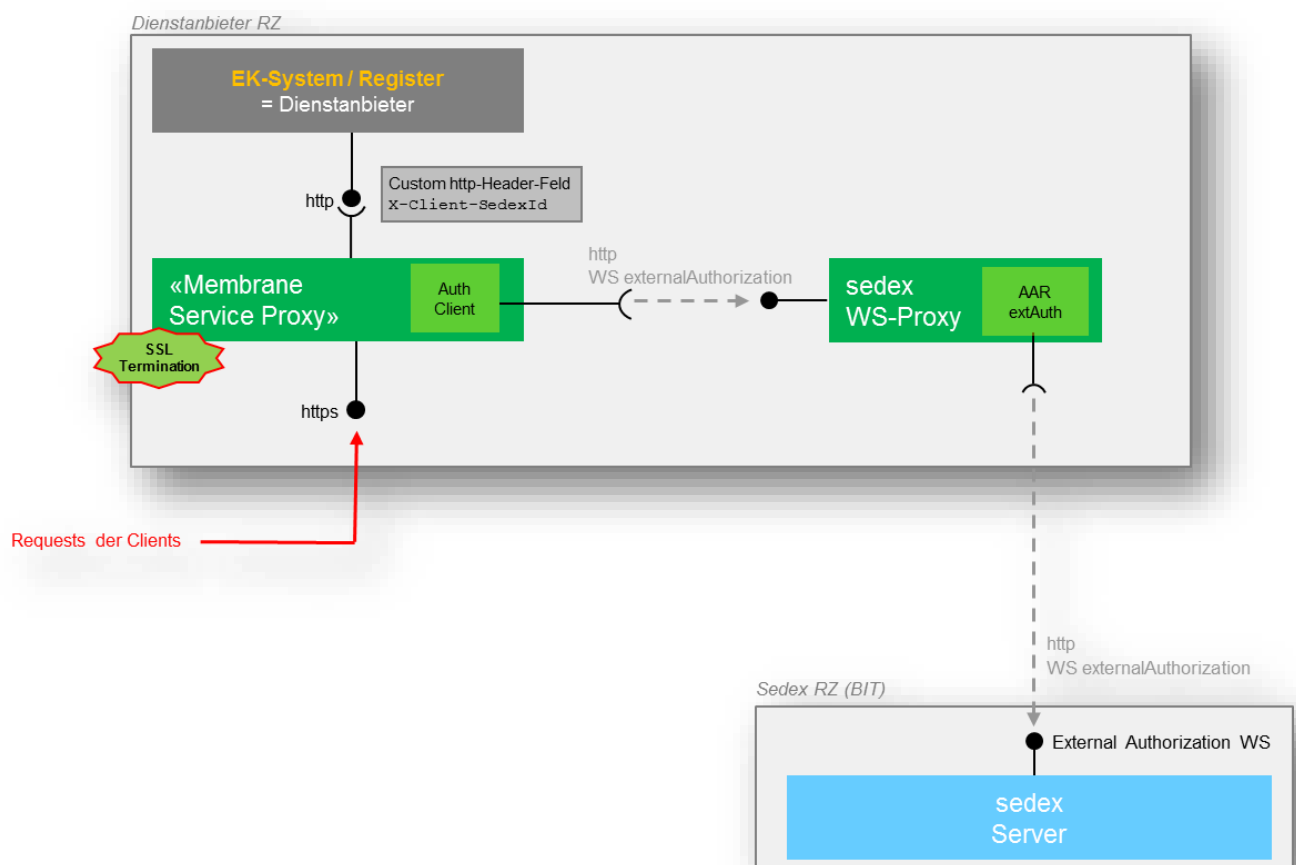


Abbildung 9: Szenario „Nur Membrane“

Der **Membrane Service Proxy** kann folgende Arbeitsschritte durchführen:

1. Terminieren der eingehenden SSL-Verbindung. Hierzu ist vom Dienstanbieter selber ein für den Endpunkt gültiges SSL-Zertifikat zu beschaffen und zu installieren.
2. Extrahieren des Client-SSL-Zertifikats
3. Einfügen des extrahierten Zertifikats als zusätzliches HTTP-Headerfeld des Requests (siehe 7.6.1)
4. Extrahieren des Client-SSL-Zertifikats aus dem HTTP-Header

5. Aufrufen des sedex-Webservices externalAuthorisation (via lokal installierten sedex-Webser-vice-Proxy) unter Verwendung des extrahierten Client-Zertifikats als einen der Autorisierungsparameter. Details zum Webservice externalAuthorization sind Abschnitt 7.6.4 zu entnehmen
6. Optional:  
Zusätzliches Überprüfen des Aufrufers gegen eine lokal definierte Whitelist mit vom Betreiber spezifisch zugelassenen sedexIds
7. Die ermittelte sedexId des Clients wird als zusätzliches HTTP-Headerfeld an die Webservice-Implementation mitgegeben (siehe 7.6.2)
8. Berechtigte Requests werden an die Webservice-Implementation weitergereicht

### 7.3.1 Konfiguration des Membrane

Im Szenario 3 muss der Membrane Service Proxy installiert und konfiguriert werden. Details sind dem Kapitel 8 zu entnehmen, das spezifisch auf Membrane eingeht.

Nachfolgend ist darüber hinaus exemplarisch eine spezifische Membrane-Konfiguration für Szenario 3 aufgezeigt und erläutert.

Das nachfolgende Beispiel zeigt die Konfiguration für einen Membrane Proxy, welcher:

- Auf Port 443 auf eingehende HTTPS-Verbindungen wartet
- Nur Verbindungen des Pfades /PersonIdentificationService/PersonIdentification akzeptiert
- Das WSDL umschreibt auf die externe URL „https://personenident.bern.ch“
- Den Aufruf weiterleitet an http://vmpersiden01 auf Port 8080

```
<spring:beans xmlns="http://membrane-soa.org/proxies/1/"
  xmlns:spring="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
  http://membrane-soa.org/proxies/1/ http://membrane-soa.org/schemas/proxies-1.xsd">

  <spring:bean id="clientCertificateInterceptor" class="ch.sedex.membrane.interceptor.clientcert.ClientCertificateInterceptor" />


  <spring:bean id="extAuthInterceptor" class="ch.sedex.membrane.interceptor.externalauthorization.ExternalAuthInterceptor">
    <spring:property name="sedexId" value="a-bbbbbbb-c" />
    <spring:property name="messageType" value="1006" />
    <spring:property name="url" value="HTTP://localhost:8080/wsproxy/services/sedexExternalAuthorisationService/" />
    <spring:property name="cacheTimeInMinutes" value="30" />
  </spring:bean>

  <router>

  <serviceProxy port="443">
    <path>/PersonIdentificationService/PersonIdentification</path>
    <ssl protocol="TLS" clientAuth="need">
      <keystore location="C:/credentials/server.p12"
        password="pass" keyPassword="pass" type="PKCS12" />
      <truststore location="sedexTrust.jks" password="trustme" />
    </ssl>
    <wsdlRewriter protocol="https" host="personenident.bern.ch" />
    <target host="vmpersident01" port="8080"/>
    <interceptor refid="clientCertificateInterceptor" />
    <interceptor refid="extAuthInterceptor" />
  </serviceProxy>
</router>
</spring:beans>
```

#### Bedeutung der Parameter:



Parameter	Beschreibung
<code>&lt;spring:property name="sedexId"&gt;</code>	Die sedexId des SOAP-Webservice-Providers. Üblicherweise ist dies die sedexId, welche im sedex-Client des Diensteanbieters konfiguriert ist.
<code>&lt;spring:property name="messageType"&gt;</code>	Die sedex-MessageType-ID, welche dem abzusichernden Webservice zugeordnet ist. Für PersonIdentification ist dies der Wert 1006.
<code>&lt;spring:property name="url"&gt;</code>	URL des sedex-externalAuthorization-Webservice. Da via sedex-WS-Proxy darauf zugegriffen wird, ist dies normalerweise der entsprechende Endpunkt auf dem lokalen sedex-WS-Proxy.
<code>&lt;spring:property name="cacheTimeInMinutes"&gt;</code>	Anzahl Minuten, für welche ein Autorisierungsergebnis aus dem Cache des Interceptors verwendet wird, bevor erneut eine Autorisierungs-Anfrage gestellt wird.
<code>&lt;serviceProxy port&gt;</code>	Der Port, unter welchem Membrane auf eingehende Verbindungen hört.
<code>&lt;path&gt;</code>	Wenn dieser Parameter gesetzt wird, werden nur auf diesem Pfad eingehende Requests akzeptiert. Der Wert sollte dem Pfad der Implementierung des Personenidentifikations-Service entsprechen.
<code>&lt;keystore location&gt;</code>	<p>Pfad zu einer Keystore-Datei, welche das SSL-Zertifikat für den SSL-Endpunkt enthält.</p> <div>  <p><b>Achtung:</b> Als Pfad-Trennzeichen immer „/“ verwenden, <b>auch auf Windows!</b></p> </div>
<code>&lt;keystore password&gt;</code>	Passwort zum Keystore.
<code>&lt;keystore keyPassword&gt;</code>	Passwort zum Private Key im Keystore. Dies ist sehr oft identisch mit dem Passwort des Keystores.
<code>&lt;keystore type&gt;</code>	Typ des Keystores. Es werden die beiden Typen «PKCS12» für P12-Dateien und «JKS» für Java-Keystore unterstützt.
<code>&lt;wsdlRewriter protocol&gt;</code>	Protokoll, unter dem der Service im Internet exponiert ist.
<code>&lt;wsdlRewriter host&gt;</code>	Hostname nach aussen, unter welchem Membrane vom KTV/Internet her verfügbar ist. Entspricht dem Hostname des bestehenden Security Reverse Proxy. Dieser Parameter wird für das WSDL-Rewriting verwendet.
<code>&lt;target host&gt;</code>	Hostname, auf dem die Implementierung des Personenidentifikations-Services läuft.
<code>&lt;target port&gt;</code>	Port, auf dem die Implementierung des Personenidentifikations-Services läuft

## 7.4 Vorbereitende Arbeiten/Organisatorisches

### 7.4.1 sedex-Teilnehmer bestellen und einrichten

Der Zugriff auf den von sedex angebotenen Webservice externalAuthorization erfolgt für (bundesexterne) Dienstanbieter in jedem der Szenarien indirekt via einen lokalen sedex-Webservice-Proxy. D.h., damit der Dienstbringer den Service externalAuthorization verwenden kann, muss er über einen eigenen sedex-Client mit installiertem Webservice-Proxy-AAR verfügen.

Das Vorgehen für Bestellung und Installation des sedex-Clients mit Webservice-Proxy erfolgt analog wie auf Seite Dienstverwender. Details sind in 6.1.1 (Bestellung) und 6.2 (Installation) beschrieben.

### 7.4.2 SSL-Zertifikat beschaffen

Der Dienstanbieter muss das für den Betrieb eines SSL-Endpunkts nötige SSL-Zertifikat selber bei einem Zertifikatsaussteller beschaffen. Ist für den vorgesehenen Service-Endpunkt bereits ein SSL-Zertifikat vorhanden, so kann dies selbstverständlich mitgenutzt werden.

## 7.5 Weitere technische Anforderungen

Auf der Seite des Dienstanbieters müssen folgende technischen Anforderungen besonders beachtet werden:

- Ermöglichung folgender Netzwerkzugriffe
  - Der Autorisierungsclient muss für die Nutzung des Webservices externalAuthorisation auf den sedex-Webservice-Proxy zugreifen können. D.h. eine Kommunikation mit dem Protokoll HTTP auf den Endpunkt des Webservice-Proxy (Default-Einstellung ist Port 8080) muss möglich sein.
  - Der Webservice-Proxy muss Zugriff auf die Endpunkte des sedex-Servers erhalten. D.h. eine Kommunikation mit dem Protokoll HTTPS auf die Endpunkte von sedex (Port 443) muss möglich sein.

## 7.6 Besondere Anmerkungen

### 7.6.1 Einfügen des SSL-Zertifikats in den HTTP-Header

Der bestehende Reverse-Proxy muss in Szenario 2 die eingehende SSL-/TLS-Verbindung terminieren und das ganze Zertifikat des Clients unter folgendem Schlüssel in den HTTP-Header des Requests schreiben, so dass der nachgelagerte Membrane Service Proxy das Zertifikat wieder extrahieren kann:

```
X-Client-Certificate
```



#### **Vorsicht vor Security Issue!**

Es muss zwingend dafür gesorgt werden, dass ein allenfalls vom Client eintreffender Request dieses HTTP-Headerfeld nicht bereits gesetzt hat bzw. dass das Feld dann zwingend mit dem effektiven Zertifikat als Wert überschrieben wird.

### 7.6.2 Einfügen der Client-sedexId in den HTTP-Header

Der bestehende Security Reverse Proxy (Szenario 1) oder der Membrane Service Proxy (Szenarien 2

und 3) kann die vom sedex-Webservice externalAuthorization festgestellte und zurückgegebene sedexId des aufrufenden Clients unter einem beliebigen Schlüssel in den HTTP-Header des Requests schreiben, so dass die nachgelagerte Webservice-Implementation diese (z.B. zwecks Logging) wieder extrahieren kann. Folgendes Feld wird verwendet:

X-Client-SedexId



#### Vorsicht vor Security Issue!

Es muss zwingend dafür gesorgt werden, dass ein allenfalls vom Client eintreffender Request dieses HTTP-Headerfeld nicht bereits gesetzt hat bzw. dass das Feld dann zwingend mit der effektiv festgestellten sedexId überschrieben wird.

### 7.6.3 Certificate Authority (CA) für sedex-Zertifikate

Bei sedex-Zertifikaten handelt es sich um spezifische für das System sedex von der SwissGov-PKI erstellte Zertifikate. Die Zertifikate sind von der folgenden CA ausgestellt:

#### Swiss Government Regular CA 01

Fingerprint (SHA-1): bb 85 9f 5c 90 7b 4a c9 6a 9e 2a 35 c1 3c f7 57 5a 06 0a ce

Link [https://www.bit.admin.ch/adminpki/00247/05329/index.html?lang=de&download=NHZLp-Zeg7t,lnp6l0NTU042l2Z6ln1acy4Zn4Z2qZpnO2Yuq2Z6gpJCDeoF7gWym162epYbg2c\\_JjKbNoKSn6A--](https://www.bit.admin.ch/adminpki/00247/05329/index.html?lang=de&download=NHZLp-Zeg7t,lnp6l0NTU042l2Z6ln1acy4Zn4Z2qZpnO2Yuq2Z6gpJCDeoF7gWym162epYbg2c_JjKbNoKSn6A--)

### 7.6.4 Benutzung des sedex-Webservice externalAuthorization

Details zur Benutzung des Webservice externalAuthorization sind [sdxEAuth] zu entnehmen.

Die Parameter für die Autorisierung mittels Operation **checkAuthorisation** sind jeweils wie folgt zu setzen.

Parameter	Beschreibung
messageTypeId	Jeder abzusichernde Webservice ist einem sedex Meldungstyp zugeordnet. Je nach zu autorisierendem Service ist hier eine andere Id zu verwenden: <ul style="list-style-type: none"><li>• PersonenIdentifikation 1006</li><li>• GWR 1009</li><li>• SASIS/VEKA 1010</li></ul>
sender	Das sedex-Zertifikat des Clients, welches aus dem SSL-Request extrahiert wurde.
recipient	Die sedexId, welche dem Teilnehmer auf Dienstanbieterseite (fix) zugeordnet ist.



#### Hinweis:

Auf den Webservice externalAuthorization wird im Normalfall via sedex-Webservice-Proxy zugegriffen. Über den WS-Proxy ist auch das WSDL abrufbar. Details siehe [sdxEAuth].

#### 7.6.4.1 Beispiel SOAP-Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
```

```
xmlns:ns="http://sedex.admin.ch/sedexExternalAuthorisationSchema/1">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:sedexExternalAuthorisationRequest>
      <ns:messageType>1006</ns:messageType>
      <ns:sender>
        <ns:participantX509Certificate>
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tDQpNSU1IUVRDQ0JTbWdBd0lCQWdJUVZyM1Y3Zzc2V0dw
eHp3bXFXZeUxWVkrBTkJna3Foa2lHOXcwQkFrc0ZBREI5DQpNUXN3Q1FZRFZRUUdFd0pEU0RFT01Bd0dB
MVVFQ2hNRlFXUnRhVzR4RVRBUEJnTlZCQXNUQ0Z0bGnuWnBZMlZ6DQpNU013SUFZRFZRUUxFeGxEWlhK
...MGFXWnBZMkYwYVc5dU1FRjFkR2h2Y21sMGFXVnpNU2N3S1FZRFZRUURFeDVUDQpkMmx6Y3lCSGIzWmxj
bTV0WlclMElGSmxaM1ZzWVhJZlEwRWdNREV3SGhjTk1UVXdOaF4TURnME1UQTVXaGNODQpNVGd3TmpB
eE1EZzBNVEV6V2pDQmp6RUxNQWtHQTFVRUJoTUNRMGd4T3pBNUJnTlZCQW9NTWxSb1pTQkdaV1JsDQpj
bUZzSUVGMWRHaHJjbWwwYVdWek1HOW1JSFJvW1NCVGYybHljeUJEYjI1bVpXUmxbUyYwYVc5dU1SUXdF
Z11EDQpWUWFMREF0QmJuZGxibVIxYm1kbGJqRU9NQXdhQTFVRUN3d0ZVMFZFU1ZneEhUQWJCZ05WQkFN
TUZETXRRMGd0DQpVPGs0TkNCVpYtYjBjRUUpKVkNCSE1JSUJJakFOQmdrcWhraUc5dzBCQVFFRkFBT0NB
UThBTU1JQkNnS0NBUEVBDQptUGtjSXVZa1dQN2srNEZnNHBocjRBVW9tN1BHV0JuRkVvVH1sazRzUm9r
ZEZmeWtaK3NXdG1Cbkk0V0xhS1hFDQpGRE92SDRLanbBMGNVeEhoZxGzZWNrM1JlTldjcnk1Q1dSeTFR
aVY2MHhBNWQwTUtBNU4veldtRDdUNzBMQS90DQpERm1LeDY5czZlTTRQYUluZitHbzhZaGlqdDZxaUJk
ZFF1L2FNY05zVV1qdzI4YkV4cWd1UTA4bzK0Z3JUdndwDQpZbDhJUFVZYXE4ZkErVW0rM1Q2MDZoa2ZB
Qkl1N0ZwTGt1VjdJcGZtQXlhRzNMb3d2bmJrakdGZFBVZVFEEUpqDQpraFk2ZGtBbVRKc3FwVTRHTi8y
WHAVSXg2cmoxNjZKeKpNODhMR3krRFM1bDZOY1VsM3krT0FHVVG5aeFRsdEtJDQp3OEZ2YTdMQVVOmNFT
UGdpVHRxdXJ3SURBUUFChzRJQ3FEQ0NBcVF3SndZSV1JVjBBUkVERndnRUd6QVpCZ2xnDQpOWFFCFRVFN
WENBb0JBZjJhNQ1RNdFEWz3RPVGs0TkRDQjJRWURUWjBnQkl1Uk1JSE9NSUhmQmdoZ2hYUUFU1XDQpG
RENCdmpCRUJnZ3JCZ0VGQ1FjQ0FSWTRhSFwY0RvdKwzZDNkeTV3YTJrdVlXUnRhVzR1WTJnd1kzQnpM
ME5RDQpVMTh5WHpFM1h6YzFObDh4WHpFM1h6TmZNaKZmTVM1d1pHwXdkZ11JS3dZQkJRvUhbZ013YWhw
b1ZHaHBjeUJwDQpjeUwYUdVZ1UzZHBjM01nUjI5M1pYSnViV1Z1ZENCUpXZDFiR0Z5SUVOQklEQXhJ
RU5RVX1CbW1zSWdVMFZFQDQpSVmdnYzNsemRHVnRjeTRnUTFCVELHwnZjaUJU1VSRldDQmKhWFJvW1c1
MGFXTMhkR2x2Ym1Cd2RYSnDiM05sDQpjeTR3RGdZRFZSMFBBUgVqKFRREFnU3dNSFVHQ0NzR0FRVUZC
d0VCQkdRdlp6QTNcZ2dyQmdFRkJRY3dBb1lyDQpSFwY0RvdKwzZDNkeTV3YTJrdVlXUnRhVzR1WTJn
d1lXbGhMMUpswJnNwcl1YSkRRVEF4TG1OeWREXNcZ2dyDQpCZ0VGQ1Fjd0FZWWhSFwY0RvdKwzZDNk
eTV3YTJrdVlXUnRhVzR1WTJnd1lXbGhMMj1qYzNBd2djY0dBMVVKDQpId1NCdnpDQnZEQXhVqYtnTF1Z
cmFIUjBjRG92TDNM2R5NXdhMmt1WVdSdGFxNHVZMmd2WTNKC0wxSmxaM1ZzDQpZWEPeUVRBeExtN1i
RENCaHFDQmc2Q0JnSVorYkdSaGNEb3ZMMkZrY1dsdVpHbH1MbUzrY1dsdUxtTm9Pak00DQpPuz1qYmox
VGQybHpjeV5V2UUVKdmRtVnlibTFsYm5RbE1qQ1NaV2QxYkdGeUpUSXdrMEVsTWpBd01TeHzkVDFEDQpa
WEowYVdacFkyRjBhVz11S1RjdlFYVjBhRz15YVhScFpYTXNiM1U5VTJWeWRtbGpaWE1zYnoxQ1pHMXBi
aXhqDQpQVU5JTUI4R0ExVWRJdlFZTUJhQUZFMtN0ZVR2Y1p6RG02QTZoK0dtN2dpbk9lZUxNQjBHQTfV
ZERnUVdCQ1QyDQpBMMJGV0JKdmY5Z0twZ3orVmtHcVazVndqVEFNQmdOVkhSTUJBZjhFQWpBQU1BMEdd
U3FHU01iM0RRRUJdd1VBDQpBNE1DQVFDcmNFV1FrOTBCZ1V2K3M0cV1vbnZ5M2MxNE11N2M3NHf1ck41
RGVJaEpNdEhqamIwOW9TMHd2U1JPDQpGNC82bjMvT3Y2MmZtR0VZd0gxZnJ3NTUwbkVMAVZieXh1WHM3
SFp1SkNjM19VWVwWnKc5cjhgZ3FCRmx5dk5NDQptbmJlQXp0TmI0L2gxQjZCR1NEMkhnbUfH3BBa2tY
S0tkWnVkvGQRSHVnbWlR2tDTSFUXQWIyOEtPQmpsSmZCDQo4VDQzcxRRtG9rMVhHVFPoV1c4cm9ydgNny
UmJHdVVMc0xOeEdUQ3RwaU1ZQWJ0OXNZNk1TYVVLRLN3YzB4bTVzDQp3dHJza3V1Z1NFtNvsQ2sxSzN1
QVRCEwDyL0g1M1NZeHVQNThwazFaMERAUVFZU3NXUGg3cG5IenQ3bH1UdzJIDQpud3NNAgM5dDIrWGY2
RXFrSk5CdmtN1000b1RFSHNYbVhLQTZBUjGdz1wSG16MDU5WTR1S1lqMWxSVWF6YnhkDQpkaVQzQW9E
dkZH2F2FyEw9CUkFws3FLdHd4cEk0WgtxZGI2WgtxNTB2eldWakZtdWtTMXFFb1V1Wmd3M0VtV0cvDQpC
WVN0NENkdFEVWUdWOCtvd0p6cGVua0ZkVWMyQ01Va0NicWszbFltZVRZQUNTYkFLWmlyZ1cwaXRwck1V
Y29wDQpVe1NsNytEV1ZBc3pXZFgvalptdXpVVFQ0U0dxS292Q3VEYWNSK3k3MEQ2QTRHb3ZpdVpnS0xz
cnJ0Y3VKWDJoDQpIOWtmR29aVFDhR0UzWUtHaU1U2d1lBOXL1EQ2tkZ0ZzbEtXS24yVmcxdE5EVy9XampL
QnZDbUNGdTYwY3YyQ3ZGQDpuV0N5cGZucU9nS1JfQz1zdTRiaG5leXNDelFtQU5SakkyWDR2ZDkvWmZZ
R1ROTxpSdz09DQotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0t
        </ns:participantX509Certificate>
      </ns:sender>
      <ns:recipient>
        <ns:participantSedexId>3-CH-9985</ns:participantSedexId>
      </ns:recipient>
    </ns:sedexExternalAuthorisationRequest>
  </ns:Body>
</ns:Header>
```

```

        <ns:requestSenderSedexId>true</ns:requestSenderSedexId>
    </ns:sedexExternalAuthorisationRequest>
</soapenv:Body>
</soapenv:Envelope>

```

#### 7.6.4.2 Beispiel SOAP-Response

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <sedexExternalAuthorisationResponse xmlns="http://sedex.admin.ch/sedexExternalAuthorisationSchema/1">
      <isAuthorised>true</isAuthorised>
      <resultErrorCode>0</resultErrorCode>
      <resultErrorMessage>OK - is authorised (Sender SedexId=3-CH-9984, Recipient SedexId=3-CH-9985, internal errorMessage: 100, null)</resultErrorMessage>
      <senderSedexId>3-CH-9984</senderSedexId>
    </sedexExternalAuthorisationResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

Erläuterungen:

- Ein Service-Request gilt nur dann als autorisiert, wenn das Element `<isAuthorised>` den Wert true enthält. Andernfalls ist der Request nicht autorisiert und muss idealerweise mit dem HTTP-Status-Code 403 Forbidden beantwortet werden.
- Die sedexId zum Sender-Zertifikat ist im Element `<senderSedexId>` enthalten.

#### 7.6.4.3 Wichtiger Hinweis auf einen fundamentalen Autorisierungs-Grundsatz von sedex

Das System sedex kennt zwei voneinander komplett isolierte Teilnehmergruppen:

- Produktive Teilnehmer (z.B. mit der sedexId 1-203-1)
- Test-Teilnehmer (z.B. mit der sedexId T1-203-1)

Ein Testteilnehmer und ein produktiver Teilnehmer können somit niemals untereinander (asynchrone) sedex-Meldungen versenden.

Ein Testteilnehmer und ein produktiver Teilnehmer werden somit in Kombination niemals ein positives Autorisierungsergebnis für synchrone SOAP-Kommunikation erhalten. D.h. die Verwendung von z.B. 2-ZH-2 (als Sender) mit T1-203-1 (als Empfänger) wird niemals autorisiert werden!

#### 7.6.5 Erstellen von Keystore-/Truststore-Dateien

In dieser Integrationsanleitung werden an verschiedenen Stellen *Keystores* und *Truststores* verwendet. Das frei erhältliche Tool „Portecle“ (<http://portecle.sourceforge.net/>) bietet eine grafische Oberfläche, womit sich diese Dateien auf eine einfache Art und Weise erzeugen und konvertieren lassen.

# 8 Membrane Service Proxy

## 8.1 Einführung

Membrane Service Proxy ist ein Reverse-HTTP-Proxy, geschrieben in Java und lizenziert unter der Open-Source-Lizenz ASF 2.0. Membrane kann verwendet werden, um:

- Service Proxies von SOAP- und REST-Webservices zu realisieren
- Interne Services im Internet verfügbar zu machen
- Services abzusichern
- SOAP-Webservices zu REST-Ressourcen umzuwandeln
- Verschiedene Services über HTTP zu integrieren

Membrane kann durch Plugins (sog. *Interceptoren*) um spezifische neue Funktionalität erweitert werden. Die Plattform sedex stellt solche Plugins für Membrane zur Verfügung, so dass Membrane die Aufgaben Authentisierung und Autorisierung mithilfe des sedex-Webservice externalAuthorization durchführen kann.

Eine umfassende Dokumentation zum Membrane Service Proxy kann unter der Adresse <http://www.membrane-soa.org> gefunden werden.

## 8.2 Download

Der Membrane Service Proxy kann von der zugehörigen Website heruntergeladen und frei eingesetzt werden:

<http://www.membrane-soa.org/downloads/index.htm>

### Hinweis:

Die vorliegende Anleitung bezieht sich auf den Membrane Service Proxy in der Version „4.0.18 (stable)“. Falls neuere Versionen zu Abweichungen gegenüber dieser Dokumentation führen, sind wir dankbar um einen Hinweis an [harm@bfs.admin.ch](mailto:harm@bfs.admin.ch).

## 8.3 Installation

Membrane wird in einer ZIP-Datei geliefert und kann an einen beliebigen Ort hin entpackt werden, z.B. nach c:\membrane. Nachfolgend wird der Pfad zu diesem Verzeichnis mit `<membrane>` bezeichnet.

## 8.4 Konfiguration

Die Konfiguration von Membrane erfolgt durch Editieren einer zentralen Konfigurationsdatei. Der Pfad dieser Datei lautet: `<membrane>/conf/proxies.xml`.

Die Dokumentation auf der Website von Membrane erläutert detailliert die verschiedenen Konfigurationsparameter.

Membrane überwacht die Konfigurationsdatei regelmässig und liest diese nach einer Änderung automatisch neu ein, so dass ein Neustart nur in den seltensten Fällen nötig ist.

Konfigurationserläuterungen für den Einsatz im Kontext von eUmzugCH können jeweils direkt in den Abschnitten zu den verschiedenen Szenarien entnommen werden.

Die nachfolgenden Abschnitte erläutern zusätzlich einige Konfigurationsoptionen zusätzlich.



**Hinweis:**

Wir empfehlen mit der von sedex zur Verfügung gestellten und ausführlich kommentierten **Musterkonfigurations-Datei** zu starten und diese so anzupassen, dass sie auf den jeweiligen Anwendungsfall passt. Siehe Abschnitt 8.6.



**Vorsicht vor Security Issue!**

Membrane besitzt nach der Installation bereits drei vorkonfigurierte Service- bzw. SOAP-Proxys als Beispiele. Diese werden nicht benötigt und müssen zwingend entfernt werden.

## 8.4.1 Logging

Grundsätzlich können beim Membrane Service Proxy zwei verschiedene Arten von Logs unterschieden werden:

- A. Technisches Log
- B. Fachliches Log der vermittelten Service-Requests und Service-Responses

### A. Technisches Log

Das technische Log (auf Konsole und in Datei `<membrane>/memrouter.log`) kann in der Datei `<membrane>/conf/log4j.properties` konfiguriert werden. Details zur Konfiguration können z.B. hier entnommen werden: [http://www.tutorialspoint.com/log4j/log4j\\_configuration.htm](http://www.tutorialspoint.com/log4j/log4j_configuration.htm).

```
## direct log messages to stdout ###
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n

log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=memrouter.log
log4j.appender.file.MaxFileSize=100MB
log4j.appender.file.MaxBackupIndex=10
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n

log4j.rootLogger=info, stdout

log4j.logger.com.predic8=info, file
```

Man kann hier die Log-Level (TRACE, DEBUG, INFO, WARN, ERROR, FATAL ) steuern aber auch spezifische Logeinträge in eine andere Datei oder die Konsole umleiten lassen. Sollen mehr Log-Einträge während der Ausführung von Membrane protokolliert werden, so können in den letzten beiden Zeilen die Log-Levels von `info` auf `DEBUG` oder `TRACE` geändert werden.

### B. Fachliches Log der vermittelten Service-Requests und Service-Responses

In der zentralen Konfigurationsdatei `<membrane>/conf/proxies.xml` kann für Service-Proxies zusätzlich das fachliche Logging eingeschaltet werden, so dass Requests und/oder Responses ebenfalls auf Konsole bzw. ins Logfile protokolliert werden. Details: <http://www.membrane-soa.org/service-proxy-doc/4.0/configuration/reference/log.htm>

### 8.4.2 Option: SSL-Verbindung von Membrane zur Service-Implementation

Wir gehen davon aus, dass innerhalb eines Rechenzentrums das Protokoll HTTP verwendet wird. Sollte aber die Verbindung zwischen Membrane und dem Personenidentifikationsservice ebenfalls mit SSL/TLS gesichert werden und somit über das Protokoll HTTPS laufen, müssen in der Konfiguration des Serviceproxy dem `<target>`-Element folgendes `<ssl>`-Element hinzugefügt und die enthaltenen Parameter entsprechend gesetzt werden:

```
<spring:beans xmlns="http://membrane-soa.org/proxies/1/"
  xmlns:spring="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://membrane-soa.org/proxies/1/ http://membrane-soa.org/schemas/proxies-
1.xsd">

  <router>
    [...]
    <serviceProxy port="80">
      <path>/PersonIdentificationService/PersonIdentification</path>
      <wsdlRewriter protocol="https" host="personenident.bern.ch" />
      <target host="vmpersident01" port="8080">
        <ssl protocol="TLS" clientAuth="false">
          <truststore location="c:/credentials/trust.jks" password="secret"
            type="JKS" />
        </ssl>
      </target>
    </serviceProxy>
    [...]
  </router>
</spring:beans>
```

In diesem Beispiel wird ein Truststore vom Typ JKS (Java-Keystore) verwendet. Analog dazu kann auch ein Keystore vom Typ PKCS12 (P12) angegeben werden.

### 8.4.3 Option: Membrane-Administrationskonsole

Membrane verfügt über eine Administrationskonsole, wie sie in Abbildung 10 ersichtlich ist:



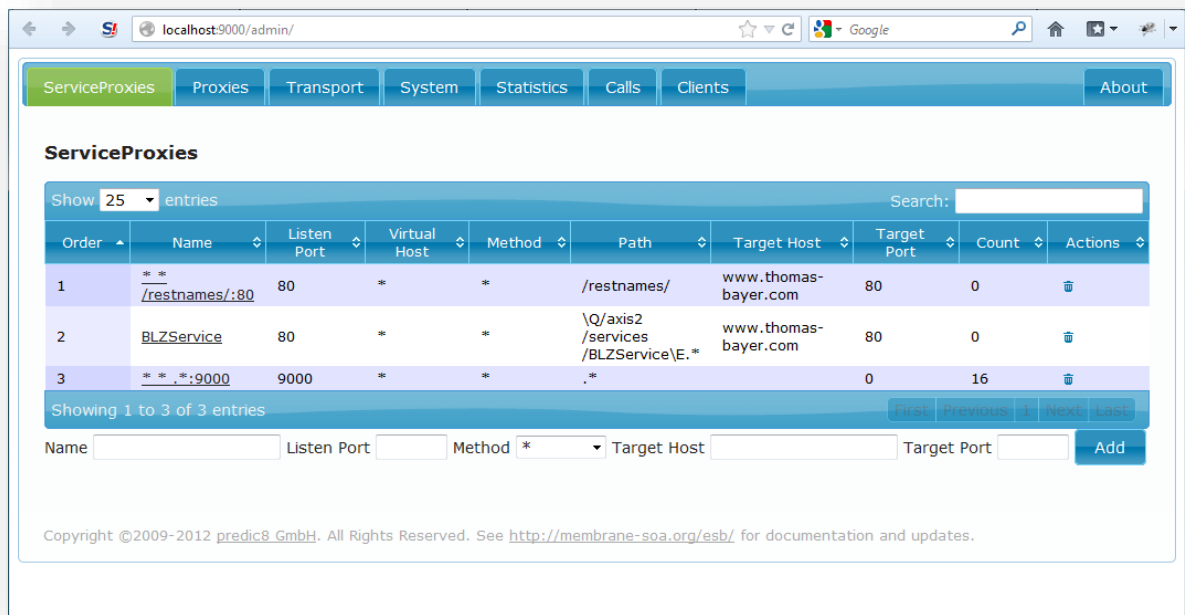


Abbildung 10: Administrationskonsole des Membrane Service Proxy

Die Administrationskonsole kann mit folgendem Eintrag in der Konfiguration aktiviert werden:

```
<spring:beans xmlns="http://membrane-soa.org/proxies/1/"
  xmlns:spring="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
  http://membrane-soa.org/proxies/1/ http://membrane-soa.org/schemas/proxies-
1.xsd">

  <router>
    [...]
    <serviceProxy name="Console" port="9000">
      <basicAuthentication>
        <user name="admin" password="membrane" />
      </basicAuthentication>
      <adminConsole />
    </serviceProxy>
    [...]
  </router>
</spring:beans>
```



#### Wichtig!

Zum Absichern der Administrationskonsole sollten ein entsprechender Benutzername und ein Passwort gewählt werden. Und falls die Administrationskonsole aktiviert wird, sollte dringend sichergestellt werden, dass diese nicht aus dem Internet zugänglich ist!

#### 8.4.4 Option: Schutz gegen XML- und DoS-Attacken

Membrane bietet gewisse Schutzmechanismen gegen XML- und Denial-of-Service- (DoS) Attacken. Um diesen Schutz für einen bestimmten Proxy hinzuzufügen, muss wie im unten aufgezeigten Beispiel das XML-Element `<xmlProtection />` innerhalb der Proxy-Definition direkt nach dem `<path>`-Element hinzugefügt werden:

```
<spring:beans xmlns="http://membrane-soa.org/proxies/1/"
  xmlns:spring="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://membrane-soa.org/proxies/1/ http://membrane-soa.org/schemas/proxies-
1.xsd">

  <router>
    <serviceProxy port="80">
      <path>/PersonIdentificationService/PersonIdentification</path>
      <xmlProtection />
      <wsdlRewriter protocol="https" host="personenident.bern.ch" />
      <target host="vmpersident01" port="8080"/>
    </serviceProxy>
  </router>
</spring:beans>
```

## 8.5 Plugins/Interceptoren

Die Funktionalität von Membrane lässt sich mittels Plugins erweitern. Um für Dienstanbieter die Integration mit sedex zu vereinfachen, stellt sedex mehrere spezifische Erweiterungs-Plugins für Membrane zur Verfügung. Diese Plugins realisieren sogenannte *Interceptoren*, welche in die Verarbeitungskette des Membrane-Proxy integriert werden können.

### 8.5.1 Installation eines Plugins/Interceptors

Ein Membrane-Plugin bzw. -Interceptor wird in Form einer JAR-Datei zur Verfügung gestellt. Installieren lässt sich ein Plugin, indem dessen JAR-Datei in nachfolgendes Verzeichnis von Membrane kopiert wird:

```
<membrane>/lib
```

### 8.5.2 sedex Client Certificate Interceptor

#### Interceptor-Funktion

Der Client Certificate Interceptor extrahiert das Client-Zertifikat des Dienstverwenders aus der bestehenden SSL-Verbindung und schreibt es für nachfolgende Verarbeitungsschritte in ein definiertes Feld des HTTP-Request-Headers. Erfolgt der Aufruf beim Dienstverwender über einen sedex-Webservice-Proxy, so handelt es sich beim extrahierten Zertifikat um das sedex-Teilnehmerzertifikat des Clients.

#### Technische Details

Als *Outputdatum* wird das Client-Zertifikat im binären DER-Format als BASE64-kodierte Zeichenkette in folgenden HTTP-Request-Header geschrieben:

```
X-Client-Certificate
```

#### Hinweise

Der Interceptor kann nur dann funktionieren, falls die SSL-Verbindung von Membrane selber terminiert wird (Szenario 3, „nur Membrane“). Erfolgt die Terminierung der SSL-Verbindung in einem vorgelagerten Netzwerkbaustein, so muss dieser Baustein das Zertifikat in den HTTP-Request-Header schreiben

und der Client Certificate Interceptor darf in Membrane nicht konfiguriert werden.

### Benutzung/Konfiguration

In der Membrane-Konfiguration kann der Interceptor wie folgt definiert werden:

```
<spring:bean id="clientCertificateInterceptor" class="ch.sedex.membrane.interceptor.clientcert.ClientCertificateInterceptor" />
```

In einem bestehenden Service-Proxy aktiviert man den Interceptor dann wie folgt:

```
<interceptor refid="clientCertificateInterceptor" />
```

## 8.5.3 sedex External Authorization Interceptor

### Interceptor-Funktion

Der External Authorization Interceptor autorisiert einen eintreffenden Client-Request mit Hilfe des Webservice sedex External Authorization. Hierzu entnimmt der Interceptor das Client-Zertifikat aus dem entsprechenden HTTP-Request-Header-Feld und verwendet dieses als Input-Parameter beim Aufruf des Webservices sedex External Authorization. Nach erfolgter Autorisierung wird die ermittelte sedexId des Client als zusätzliches HTTP-Request-Header-Feld für nachfolgende Verarbeitungsschritte mitgegeben.

### Technische Details

Als *Inputdatum* wird das Client-Zertifikat in folgendem HTTP-Request-Header-Feld erwartet:

```
X-Client-Certificate
```

Das Zertifikat kann in den X509-Formaten DER oder PEM als BASE64-encoded String vorliegen.

Als *Outputdatum* wird nach erfolgreicher Autorisierung die ermittelte sedexId des Client in folgendem zusätzlichen HTTP-Request-Header-Feld hinzugefügt:

```
X-Client-SedexId
```

### Hinweise

Der Interceptor kann nur dann funktionieren, wenn das Client-Zertifikat im Header des HTTP-Requests vorhanden ist. Dieses Header-Feld `X-Client-Certificate` muss entweder vom Membrane Client Certificate Interceptor oder einem anderen vorgelagerten Netzwerkbaustein dem HTTP-Request hinzugefügt werden.

### Benutzung/Konfiguration

In der Membrane-Konfiguration kann der Interceptor einer Service-Definition wie folgt hinzugefügt werden:

```
<spring:bean id="extAuthInterceptor" class="ch.sedex.membrane.interceptor.externalauthorization.ExternalAuthInterceptor">
    <spring:property name="sedexId" value="1-500-1" />
</spring:bean>
```

```

    <spring:property name="messageType" value="100" />

    <spring:property name="url" value="http://localhost:8080/wsproxy/se
rvices/sedexExternalAuthorisationService/" />

    <spring:property name="cacheTimeInMinutes" value="30" />

</spring:bean>

```

Folgende Parameter des Interceptors müssen bzw. können konfiguriert werden:

Interceptor Parameter	Beschreibung
sedexId	sedexId des Dienstbringers
messageType	Meldungstyp des zu autorisierenden Services (vgl. 7.6.4)
url	Endpunkt des Webservice sedex External Authorization. Zeigt in der Regel auf den lokal installierten sedex-WSPProxy.
cacheTimeInMinutes	<p><i>Optional Parameter:</i></p> <p>Definiert, wie viele Minuten eine Autorisierungs-Antwort des External Authorization Webservices im Interceptor zwischengespeichert wird, bevor wieder eine Autorisierungsanfrage gestellt wird.</p> <p>Zum Deaktivieren des Caches kann der Wert (temporär) auf 0 gesetzt werden. In produktiven Umgebungen sollte allerdings immer eine cacheDuration von 30 Minuten oder mehr verwendet werden.</p> <p>Default-Wert: 30 Minuten.</p>

In einem bestehenden Service-Proxy aktiviert man den Interceptor dann wie folgt:

```

<interceptor refid="extAuthInterceptor" />

```

## 8.5.4 sedex Whitelist Interceptor

### Interceptor-Funktion

Der Whitelist Interceptor autorisiert nur Aufrufe definierter Clients. Alle anderen Clients werden mit einer entsprechenden Meldung abgewiesen.

### Technische Details

Als *Inputdatum* wird die Client-sedexId in folgendem HTTP-Request-Header-Feld erwartet:

```

X-Client-SedexId

```

### Hinweise

Der Interceptor kann nur dann funktionieren, wenn die Client-sedexId im Header des HTTP-Requests vorhanden ist. Dieses Header-Feld *X-Client-SedexId* muss entweder vom Membrane External Authorization Interceptor oder einem anderen vorgelagerten Netzwerkbaustein dem HTTP-Request hinzugefügt werden.

## Benutzung/Konfiguration

Der Interceptor autorisiert nur diejenigen sedexIds, welche explizit in einer Whitelist-Datei aufgeführt sind („whitelisting“).

Die Whitelist-Datei ist eine einfache Textdatei, in welcher die für die Verwendung des Service zugelassenen sedexIds durch eines der folgenden Zeichen separiert sind:

- Komma (,)
- Semikolon (;)
- Leerzeichen („blank“)
- Zeilenvorschub

Um eine Fehlkonfiguration zu verhindern, sind leere Whitelist-Dateien grundsätzlich nicht zugelassen.

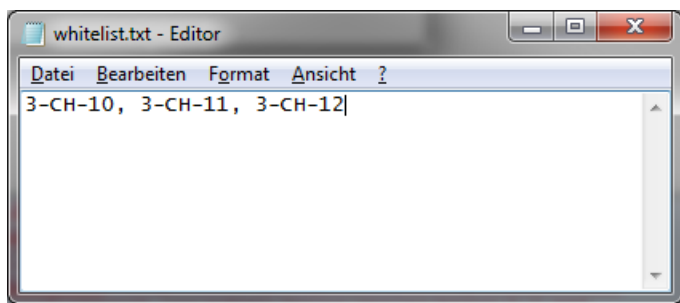


Abbildung 11: Beispiel einer gültigen Whitelist-Datei

In der Membrane-Konfiguration kann der Interceptor einer Service-Definition wie folgt hinzugefügt werden:

```
<spring:bean id="whitelListInterceptor" class="ch.sedex.membrane.interceptor
.whitelListInterceptor">
    <spring:property name="whitelist" value="whitelist.txt" />
</spring:bean>
```

Parameter	Beschreibung
whitelist	Pfad zur Whitelist-Datei.  Hinweis: Relative Pfadangaben beziehen sich auf das Hauptverzeichnis <membrane>.

In einem bestehenden Service-Proxy aktiviert man den Interceptor dann wie folgt:

```
<interceptor refid="whitelListInterceptor" />
```

Die vom Interceptor verwendete Whitelist wird beim Starten des Interceptors im Logfile eingetragen und ist jederzeit in der interaktiven webbasierten Administrationskonsole von Membrane einsehbar (siehe Kap. 8.5.5).

## 8.5.5 Konfiguration der Interceptoren in der Membrane-Administrationskonsole

Falls die Membrane-Administrationskonsole im Konfigurationsfile aktiviert ist (vgl. Kap. 8.4.3), kann die effektiv resultierende Konfiguration damit einfach überprüft werden. Hierzu ist in der Administrationskonsole unter dem Menüpunkt „Service Proxies“ der „PersonIdent“-ServiceProxy auszuwählen.

Wie in Abbildung 12 dargestellt wird der Fluss eines Requests durch alle aktiven Interceptoren grafisch dargestellt. Details zu den einzelnen Interceptoren (Name, Beschreibung, aktive Konfiguration) werden nach einem Klick auf die entsprechende Schaltfläche des Interceptors aufgeklappt und dargestellt.

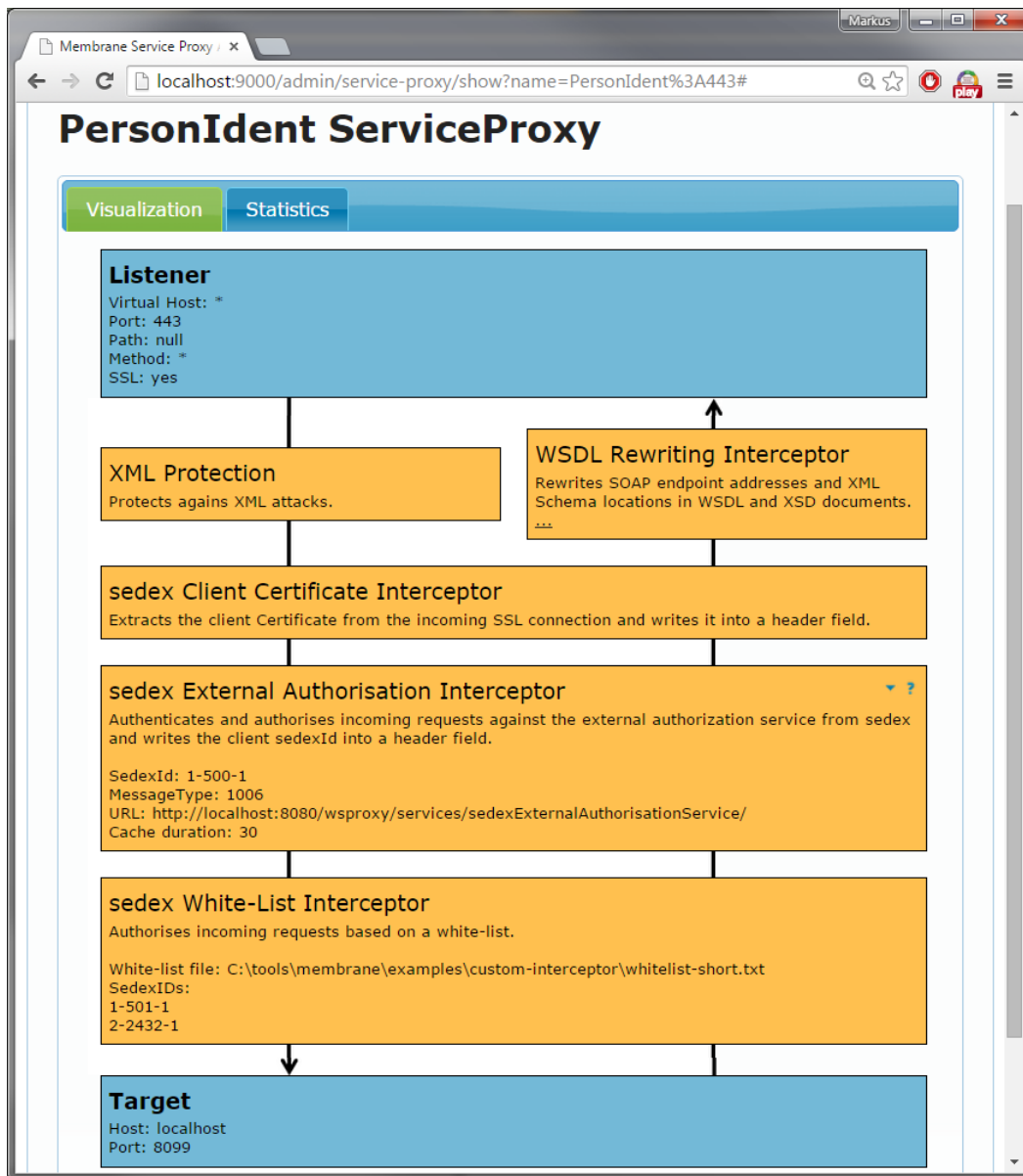


Abbildung 12: Die Interceptoren in der Membrane-Administrationskonsole

## 8.6 Kommentierte Muster-Konfigurationsdatei

Die nachfolgende kommentierte Muster-Konfigurationsfile eignet sich als Ausgangspunkt bei der Erstellung einer eigenen Membrane-Konfiguration.

```
<!--
****
* Membrane Example Configuration File for the usage with sedex Plugins.
```

```

*****

This file serves as a starting point for the integration of sedex external authorization using Membrane Service Proxy.

IMPORTANT:
The shown configuration is an example only and must be adapted to your actual setup and requirements.

*****
* The scenarios
*****

The comments in this file refer to one of the following operation scenarios:

Scenario 1: No Membrane - your existing reverse proxy does it all
Actually in this scenario you probably have no need for a Membrane Service Proxy.
You have to integrate your existing reverse proxy on your own.

Scenario 2: Your existing reverse proxy only terminates SSL, Membrane does the rest
In this scenario Membrane service Proxy does
- read client certificate from HTTP header field X-Client-Certificate
- call sedex externalAuthorisation webservice for authentication and authorization
- add HTTP header field X-Sedex-ClientId
- optionally: check clients sedexId against a local whitelist

Scenario 3: Membrane does it all
In this scenario Membrane service Proxy does
- SSL endpoint and terminate SSL connection
- add HTTP header field X-Client-Certificate
- read client certificate from HTTP header field X-Client-Certificate
- call sedex externalAuthorisation webservice for authentication and authorization
- add HTTP header field X-Sedex-ClientId
- optionally: check clients sedexId against a local white-list

*****
* Membrane Routing Configuration
*****

Have a look at
HTTP://membrane-soa.org/service-proxy-doc/current/configuration/proxy-configuration.htm for
documentation and a reference explaining what XML elements can be used how and where.

Changes to this file will be picked up almost instantly if Membrane is
running once this file has been saved. Any dynamic configuration changes
made (for example, via the adminConsole) will be forgotten.
-->
<spring:beans xmlns="HTTP://membrane-soa.org/proxies/1/"
  xmlns:spring="HTTP://www.springframework.org/schema/beans"
  xmlns:xsi="HTTP://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="HTTP://www.springframework.org/schema/beans HTTP://www.springframework.org/schema/beans/spring-beans-3.0.xsd
    HTTP://membrane-soa.org/proxies/1/ HTTP://membrane-soa.org/schemas/proxies-1.xsd">

  <!-- Note: Comment out every part of the configuration file that you want to disable. -->

  <!-- ===== -->
  <!-- Definition of sedex Client Certificate Interceptor -->
  <!-- Adds HTTP header field X-ClientCertificate to the request. -->
  <!-- Needed in scenario 3 only. -->
  <!-- ===== -->
  <spring:bean id="clientCertificateInterceptor" class="ch.sedex.membrane.interceptor.clientcert.ClientCertificateIntercep-
tor" />

  <!-- ===== -->
  <!-- Definition of sedex External Auth Interceptor -->
  <!-- Reads the client certificate from HTTP header field X-ClientCertificate. -->
  <!-- Calls sedex externalAuthAuthorization webservice for authentication and authorization. -->
  <!-- Adds HTTP header field X-Sedex-ClientId. -->
  <!-- Needed in scenarios 2 and 3. -->
  <!-- ===== -->
  <spring:bean id="extAuthInterceptor" class="ch.sedex.membrane.interceptor.externalauthorization.ExternalAuthInterceptor">

    <!--The sedexId of the service provider. This is probably the sedexId configured in your sedex client. -->
    <spring:property name="sedexId" value="a-bbbbbbb-c" />
    <!--The messageType for your webservice. This value is provided by sedex. E.g. for personIdentification in eUmzugCH
it is 1006. -->
    <spring:property name="messageType" value="1006" />
    <!--URL of the sedex External Authorisation webservice. Normally this is the endpoint of the service at your local
sedex WS-Proxy. -->
    <spring:property name="url" value="HTTP://localhost:8080/wsproxy/services/sedexExternalAuthorisationService/" />
    <!--Cache duration in minutes. Defines for how long sedex External Authorisation results will be cached. (0=disable
caching) -->
    <spring:property name="cacheTimeInMinutes" value="30" />
  </spring:bean>

  <!-- ===== -->
  <!-- Definition of sedex White List Interceptor -->
  <!-- Reads the client sedexId from the HTTP header field X-Client-SedexId. -->
  <!-- Checks the clients sedexId against a local white-list. -->
  <!-- Optional in scenarios 2 and 3. -->

```

```

<!-- ===== -->
<spring:bean id="whiteListInterceptor" class="ch.sedex.membrane.interceptor.WhiteListInterceptor">

    <!--The path to a white-list-file containing the allowed sedexIds. Relative paths start at the base directory of Mem-
brane. -->
    <!--The sedexIds in the white-list-file can be separated by the following chars: coma, semicolon, newline, space -->
    <spring:property name="whitelist" value="conf/yourWhitelistFile.txt" />
</spring:bean>

<!-- The main configuration is contained in this element. One or more serviceProxy elements have to be present. -->
<router>

    <!-- ===== -->
    <!-- Service Proxy definition for PersonIdentification -->
    <!-- ===== -->
    <!-- The port number defines where to listen for incoming requests. -->
    <!-- Typically port number is 443 for SSL/https endpoints (scenario 3) and 80 or 8080 for http endpoints (scenario
2). -->
    <serviceProxy port="443" name="PersonIdentification">

        <!-- Optional (but recommended) Path Filter.-->
        <!-- If a path filter is set, Membrane will only consider this service proxy, if the path of incoming HTTP re-
quests matches. -->
        <!-- See membrane reference documentation for details. -->
        <!-- Note: If set, it is recommended that the path on Membrane is identical to the path on the destination host.
-->
        <!--<path>/PersonIdentificationService/PersonIdentification</path>-->

        <!-- SSL endpoint configuration. Needed in scenario 3. -->
        <!-- See membrane reference documentation for details. -->
        <!-- You can use Portecle (http://portecle.sourceforge.net/) to convert certificates into Java keystore files
(JKS). -->
        <!-- Comment out this element if the ssl connection is terminated by an external reverse proxy-->
        <ssl protocol="TLS" clientAuth="need">
            <!-- Configure the path to your own server certificate. -->
            <keystore location="yourKeystore.jks" password="yourKeystorePassword" keyPassword="yourKeyPassword"
type="JKS" />
            <!-- Use the provided truststore to accept sedex client certificates. -->
            <truststore location="sedexTrust.jks" password="trustme" />
        </ssl>

        <!-- Comment out this element to disable the optional XML- & DoS protection. -->
        <xmlProtection />

        <!-- Configure the host to match your public endpoint address for external users. -->
        <!-- Comment out this element to disable wsdl rewriting. (Not recommended) -->
        <wsdlRewriter protocol="HTTPS" host="personident.bern.ch" />

        <!-- Client Certificate Interceptor as defined above. Needed in scenario 3. -->
        <interceptor refid="clientCertificateInterceptor" />

        <!-- External Authorisation Interceptor as defined above. Needed in scenario 2 and 3. -->
        <interceptor refid="extAuthInterceptor" />

        <!-- White List Interceptor as defined above. Optionally in scenario 2 and 3. -->
        <interceptor refid="whiteListInterceptor" />

        <!-- Configure the host and port to match your endpoint of the real service implementation. -->
        <target host="localhost" port="8099" />
    </serviceProxy>

    <!-- ===== -->
    <!-- Membrane Administration Console -->
    <!-- For security reasons, this proxy should NOT be exposed to the internet. -->
    <!-- The console can be accessed in a web browser at http://localhost:9000 -->
    <!-- ===== -->
    <serviceProxy name="Console" port="9000">

        <adminConsole />

        <!-- Enable access control element to restrict access to certain IP address ranges. -->
        <!-- See membrane reference documentation for more information about the possibilities of access control. -->
        <!-- <accessControl file="config/acl.xml" /> -->

        <!-- Enable basic user authentication for the admin console. -->
        <basicAuthentication>
            <!--Security issue: Use a more secure password! -->
            <user name="admin" password="yourPassword" />
        </basicAuthentication>
    </serviceProxy>
</router>
</spring:beans>

```



## 9 Fehlereingrenzung und -suche

Meldungen in der Membrane-Logdatei (<membrane>/memrouter.log)

Symptom	Ursachen und Behebung
org.springframework.context.ApplicationContextException: Failed to start bean 'router'; nested exception is java.lang.RuntimeException: com.predic8.membrane.core.config.ConfigurationException: <b>File from Attribute "whitelist" does not contain any sedex ID.</b> Remove this interceptor from your configuration if you do not need a whitelist.	<ul style="list-style-type: none"> <li>Die in der Konfiguration des sedex-Whitelist-Interceptors angegebene Whitelist-Datei (&lt;spring:bean id="whiteListInterceptor" ...&gt;&lt;spring:property name="whitelist" value="..."&gt;) ist leer. Dies gilt als Sicherheitslücke und wird vom sedex-Whitelist-Interceptor nicht akzeptiert. – Wenn die Whitelist-Funktion erforderlich ist, mindestens eine sedex-ID eintragen. Wenn die Whitelist-Funktion nicht erforderlich ist, den Interceptor aus der Membrane-Konfiguration entfernen.</li> </ul>
ERROR HotDeploymentThread:95 - Could not redeploy. org.springframework.context.ApplicationContextException: Failed to start bean 'router'; nested exception is java.lang.RuntimeException: com.predic8.membrane.core.config.ConfigurationException: <b>File from Attribute "whitelist" is not an existing file</b>	<ul style="list-style-type: none"> <li>Die in der Konfiguration des Whitelist-Interceptors angegebene Whitelist-Datei (&lt;spring:bean id="whiteListInterceptor" ...&gt;&lt;spring:property name="whitelist" value="..."&gt;) existiert nicht. Dies gilt als Sicherheitslücke und wird vom sedex-Whitelist-Interceptor nicht akzeptiert. – Wenn die Whitelist-Funktion erforderlich ist, korrekten Pfad angeben. Dabei Unterschied zwischen absoluten und relativen Pfaden beachten. Wenn die Whitelist-Funktion nicht erforderlich ist, den Interceptor aus der Membrane-Konfiguration entfernen.</li> </ul>
ERROR HotDeploymentThread:95 - Could not redeploy. org.springframework.context.ApplicationContextException: Failed to start bean 'router'; nested exception is java.lang.RuntimeException: java.lang.IllegalArgumentException: <b>Could not download the WSDL 'http://...'</b> .	<ul style="list-style-type: none"> <li>Die in der Konfiguration des WSDL-Rewriters angegebene URI (&lt;wsdlRewriter ... host="..."&gt;) ist nicht korrekt, nicht erreichbar oder der Zugriff darauf ist nicht erlaubt.</li> </ul>
ERROR HotDeploymentThread:90 - line 40: <b>Zeichenfolge "--" ist in Kommentaren nicht zulässig.</b>	<ul style="list-style-type: none"> <li>Verschachtelter Kommentar (&lt;!-- ... &lt;!-- ... --&gt; ... --&gt;) in der Membrane-Konfigurationsdatei. – In der angegebenen Zeile oder einer davon abhängigen korrigieren.</li> </ul>
ERROR HotDeploymentThread:95 - Could not redeploy. org.springframework.context.ApplicationContextException: Failed to start bean 'router'; nested exception is java.lang.RuntimeException: com.predic8.membrane.core.transport.PortOccupiedException: <b>Opening a serversocket at port ... failed.</b> Please make sure that the port is not occupied by a different program or change your rule configuration to select another one.	<ul style="list-style-type: none"> <li>Die in der Fehlermeldung und in der Konfiguration des Whitelist-Interceptors angegebene Portnummer (&lt;serviceproxy ... port="..."&gt;) ist bereits belegt. – Prüfen, ob bereits eine andere Instanz von Membrane läuft. Anderenfalls andere Portnummer wählen.</li> </ul>
WARN AbstractHttpHandler:87 - An exception occurred while handling a request: java.lang.RuntimeException: <b>Request caused X-Forwarded-For flood:</b> POST /mockPersonIdentificationBindingByMembrane HTTP/1.1 ... X-Forwarded-For: ... X-Forwarded-For: ..., ... X-Forwarded-For: ..., ...	<ul style="list-style-type: none"> <li>Quelle und Ziel des Membrane Service Proxy (&lt;serviceproxy port="..."&gt; ... &lt;target ... port="..." /&gt;) bilden einen Zyklus, z.B. liegen sie auf dem gleichen Rechner und es wurden die gleichen Portnummern verwendet.</li> </ul>

WARN HttpClient:182 - <b>Unknown host:</b> ...	<ul style="list-style-type: none"> <li>Das Ziel des Membrane Service Proxy (<code>&lt;serviceProxy&gt; ... &lt;target host="..." ... /&gt;</code>) ist nicht auffindbar. – Rechnernamen überprüfen.</li> </ul>
WARN HTTPClientInterceptor:65 - <b>Target http://... is not reachable.</b> java.net.ConnectException: Connection refused: connect	<ul style="list-style-type: none"> <li>Das Ziel des Membrane Service Proxy (<code>&lt;serviceProxy&gt; ... &lt;target host="..." port="..." /&gt;</code>) ist nicht erreichbar. – Port, Zugriffsrechte überprüfen</li> </ul>
ERROR HotDeploymentThread:95 - Could not redeploy. org.springframework.beans.factory.CannotLoadBeanClassException: <b>Cannot find class</b> [...] for bean with name '...' defined in file [...]; nested exception is java.lang.ClassNotFoundException: ...	<ul style="list-style-type: none"> <li>Jar des Interceptors nicht/nicht im korrekten Verzeichnis installiert</li> <li>Schreibfehler im Klassennamen eines Interceptors (<code>&lt;spring:bean ... class="..."</code>)</li> </ul>
ERROR HotDeploymentThread:95 - Could not redeploy. org.springframework.context.ApplicationContextException: Failed to start bean 'router'; nested exception is java.lang.RuntimeException: com.predic8.membrane.core.config.ConfigurationException: <b>The property "messageType" from interceptor ExternalAuth has to be a number.</b> Please configure it in your proxies.xml	<ul style="list-style-type: none"> <li>Schreibfehler im Meldungstyp (<code>&lt;spring:propertyName="messageType" value="..."</code>). – Korrekten Meldungstyp beim Domänenverantwortlichen erfragen und eintragen.</li> </ul>
ERROR HotDeploymentThread:95 - Could not redeploy. org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'router': ... : Initialization of bean failed; nested exception is org.springframework.beans.factory.NoSuchBeanDefinitionException: <b>No bean named '...' is defined</b>	<ul style="list-style-type: none"> <li>In der Routerdefinition wird ein Interceptor verwendet (<code>&lt;interceptor refid="..." /&gt;</code>), der nicht definiert ist. – Schreibweise der ID überprüfen. Definition des Interceptors gemäss Anleitung ergänzen.</li> </ul>
INFO [AbstractHttpRequestDesktopPanel] Error getting response for [PersonIdentificationBinding.searchPersonIdentification:Request 1]; javax.net.ssl.SSLHandshakeException: <b>Remote host closed connection during handshake</b>	<ul style="list-style-type: none"> <li>Es konnte keine SSL-/TLS-Verbindung zum Endpunkt beim Dienstanbieter aufgebaut werden. – In Szenario 3, „Nur Membrane“, dazu den Client-Certificate-Interceptor installieren und konfigurieren.</li> </ul>
INFO [AbstractHttpRequestDesktopPanel] Error getting response for [PersonIdentificationBinding.searchPersonIdentification:Request 1]; java.net.SocketException: <b>Software caused connection abort: recv failed</b>	<ul style="list-style-type: none"> <li>Es konnte eine SSL-/TLS-Verbindung zum Endpunkt beim Dienstanbieter aufgebaut werden, aber der es ist beim Aufruf kein oder ein ungültiges Client-SSL-Zertifikat verwendet worden.</li> </ul>
WARN SSLContextCollection:106 – <b>Could not retrieve DNS hostname for certificate, using</b> <b>**</b> : membrane.jks	Name des Rechners, auf dem Membrane läuft, stimmt nicht mit dem Namen im Server-Zertifikat überein. – Überprüfen und richtiges Zertifikat verwenden.

#### Meldungen in der SOAP-Antwort

Symptom	Ursache und Behebung
Unauthorized. - Access denied: No Access without a client certificate.	<ul style="list-style-type: none"> <li>sedex-externalAuth-Webservice wurde mit HTTP aufgerufen. – HTTPS verwenden.</li> </ul>
Unauthorized. - Access denied: Authorization service currently not available.	<ul style="list-style-type: none"> <li>sedex-externalAuth-Webservice ist nicht erreichbar. – Prüfen, ob der sedex-Client gestartet ist, ob der sedex-externalAuth-Webservice installiert ist, ob der sedex-externalAuth-Webservice im sedex-WS-Proxy korrekt konfiguriert ist, ob die Adresse des sedex-externalAuth-Webservice in Membrane richtig konfiguriert</li> </ul>

	ist ( <code>&lt;spring:property name="url" value="..." /&gt;</code> ).
Unauthorized. - Access denied: Access denied: You are not allowed to access this service.	<ul style="list-style-type: none"> <li>Einer der drei folgenden Parameter, sedex-ID des Dienstanbieters (<code>&lt;spring:property name="sedexId" value="..."</code>), sedex-ID des aufrufenden Dienstverwenders (über sedex-externalAuth-Webservice ermittelt), sedex-Meldungstyp für Authentisierung der Personenidentifizierung (<code>&lt;spring:property name="..." value="..."</code>), ist nicht korrekt konfiguriert. – Korrekte Werte beim Domänenverantwortlichen erfragen und eintragen. Über den Domänenverantwortlichen prüfen lassen, das sedex die Abfrage mit den selben Werten erlaubt hat.</li> </ul>
ch.admin.bit.sedex.wsproxy.proxyws.cbr.RoutingException: Could not find a matching route (municipalityId = ...)	<ul style="list-style-type: none"> <li>SOAP-Aufruf enthält eine unbekannte Gemeinde-Nr. – Erlaubte Werte beim Domänenverantwortlichen erfragen und eintragen. Über den Domänenverantwortlichen prüfen lassen, das sedex den erwünschten Wert konfiguriert hat.</li> </ul>
Internal Server Error - Target host ... is unknown. DNS was unable to resolve host name.	<ul style="list-style-type: none"> <li>Das Ziel des Membrane Service Proxy (<code>&lt;serviceProxy&gt; ... &lt;target host="..." ... /&gt;</code>) ist nicht auffindbar. – Rechnernamen überprüfen.</li> </ul>
Bad Gateway - Target ... is not reachable.	<ul style="list-style-type: none"> <li>Das Ziel des Membrane Service Proxy (<code>&lt;serviceProxy&gt; ... &lt;target host="..." port="..." /&gt;</code>) ist nicht erreichbar. – Port, Zugriffsrechte überprüfen.</li> </ul>
Bad Request - This request was not accepted by Membrane Service Proxy. Please correct the request and try again.	<ul style="list-style-type: none"> <li>Pfad (<code>&lt;serviceProxy&gt; ... &lt;path&gt;...&lt;/path&gt;</code>) überprüfen. Der konfigurierte Pfad muss ein Präfix des Pfades in der Anfrage sein.</li> </ul>
Unauthorized. - Access denied: Client certificate authentication failed.	<ul style="list-style-type: none"> <li>Es konnte kein Client-SSL-Zertifikat an den sedex-externalAuth-Webservice übergeben werden. – Der die SSL-/TLS-Verbindung terminierende Proxy muss das HTTP-Headerfeld „X-Client-Certificate“ setzen. In Szenario 3, „Nur Membrane“, dazu den Client-Certificate-Interceptor installieren und konfigurieren.</li> <li>Das an den sedex-externalAuth-Webservice übergebene Client-SSL-Zertifikat ist ungültig.</li> <li>Szenarien 1 und 2: Das an den sedex-externalAuth-Webservice übergebene Client-SSL-Zertifikat ist beim Terminieren der SSL-/TLS-Verbindung nicht korrekt extrahiert und BASE64-kodiert worden.</li> <li>External-Authentication-Interceptor ist vor dem Client-Certificate-Interceptor konfiguriert statt danach.</li> </ul>
Forbidden - Access denied: Your sedexId is not unlocked to access this service.	<ul style="list-style-type: none"> <li>Die sedex-ID des Aufrufers ist nicht in der Whitelist aufgeführt.</li> </ul>
Service Unavailable</h1> <p>Access denied: Failed to authorize your sedexId to access this service.	<ul style="list-style-type: none"> <li>Whitelist-Interceptor ist vor dem External-Authentication-Interceptor konfiguriert statt danach.</li> </ul>

#### Symptome in der Admin-Konsole

Symptom	Ursache und Behebung
	<ul style="list-style-type: none"> <li></li> </ul>
	<ul style="list-style-type: none"> <li></li> </ul>

## 10 Typische Fragen (FAQ)

Frage	Antwort
Wo finde ich weitergehende Informationen?	Die referenzierten Dokumente in Kapitel 2 sollten zuerst konsultiert werden. Darüber hinaus bietet das Webportal zu sedex unter <a href="http://www.sedex.ch">www.sedex.ch</a> weitere Informationen an.
An wen kann ich mich bei Fragen wenden?	Folgende Stellen können Auskunft erteilen:  1. Projektleiter eUmzugZH  2. Service Clientèle BFS Das Bundesamt für Statistik (BFS) betreibt einen Service clientèle für die Registerharmonisierung E-Mail: <a href="mailto:harm@bfs.admin.ch">harm@bfs.admin.ch</a> Hotline: 0800 866 700  3. sedex-Entwicklung Via Service Clientèle BFS
Stellt sedex die SSL-Zertifikate zur Verfügung?	Nein. Die sedex-Zertifikate dienen ausschliesslich zur Identifikation der sedex-Teilnehmer. Ein SSL-Zertifikat bescheinigt den Eigentümer einer bestimmten URL und muss spezifisch für jeden Endpunkt durch den Dienstanbieter selber bei einer Zertifikatsausgabestelle beschafft werden.