Swissdec-Adapter

Betriebshandbuch

Versionsgeschichte			
Version 1.0	2013-08-07	Marco Stettler	Version 1.0
Version 1.2	2014-01-30	Christoph Brunner	Recovery Tool & Patch Release 2.0_2
Version 1.3	2015-03-02	Marco Stettler	Patch Release 2.0_4
Version 1.4	2016-06-29	Marco Stettler	Release 2.1_0
Version 1.5	2017-06-19	Marco Stettler	Patch Release 2.1_1
Version 1.6	2020-09-15	Christoph Brunner	Release 3.0_0
Version 1.7	2021-07-28	Christoph Brunner	Patch Release 3.0_5

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Liste der Beispiele

1. Einführung

Dieses Dokument beschreibt die *Swissdec* Adapter Installation, Migration, Konfiguration und den Betrieb. Der SwissdecAdapter ist eine Java Applikation, bestehend aus zwei Webapplikationen, um Swissdec Meldungen für die Domänen Tax (Lohnausweise), TaxAtSource (Quellensteuerabrechnungen) sowie TaxCrossborder (Grenzgängermeldungen) auszutauschen. Er wurde zur Integration mit dem *sM-Client* entwickelt, kann jedoch auch standalone betrieben werden.

1.1. Übersicht Swissdec

Swissdec ist ein nicht gewinnorientiertes Gemeinschaftsprojekt mehrerer unabhängiger Partner und das Qualitätslabel für den elektronischen Datenaustausch zwischen Unternehmen und Versicherern sowie Behörden.

Als zentrale Informationsplattform zur Standardisierung des elektronischen Datenaustausches bietet Swissdec folgende Dienstleistungen:

- Swissdec stellt Know-how für die Standardisierung bereit
- dient dem Informationsaustausch zwischen allen Interessierten wie Software-Anwendern, ERP-Herstellern, Unternehmen, Verbänden, Ämtern und Organisationen
- überwacht die sichere Datenübertragung
- zertifiziert die erfolgreich geprüften Lohnprogramme.

DIE MISSION VON SWISSDEC

Swissdec vereinfacht den Datenaustausch zwischen Unternehmen und den bei Swissdec beteiligten Partnern unter folgenden Aspekten:

- Erarbeiten von Standards
- Sicherheit und Datenschutz für alle
- Kosteneinsparungen für alle
- Qualitätssicherung

Die Swissdec-Plattform ist eine Service orientierte Client - Server Plattform mit dem Distributor als Intermediär. Folgendes Diagram gibt eine kurze Übersicht über die Swissdec Architektur:

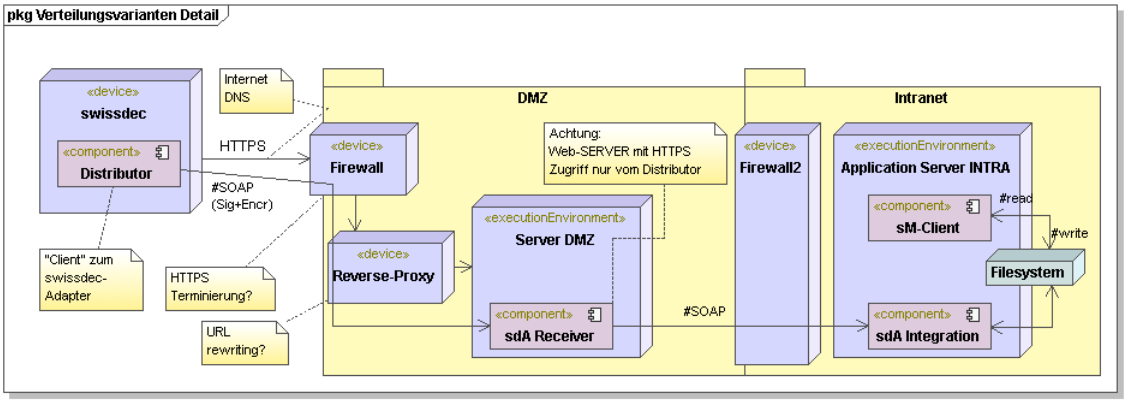

Abbildung 1.1. Installationsskizze Swissdec-Adapter

1.2. Swissdec-Adapter

Der Swissdec-Adapter bietet folgende Funktionalität:

- Empfang und Versand von protokollkonformen *Swissdec*-Meldungen
- sicherer Transport der Lohndaten mittels SSL/TLS und WS-Security
- Integritätsprüfung mittels WS-Security Signatur

Die Swissdec Plattform ist synchron. Das bedeutet, dass jede Meldung von einem Unternehmen vom Distributor sofort verarbeitet und an die Endempfänger weitergeleitet wird. Das Unternehmen (Lohndatenquelle) erlebt das ganze System somit als Einheit. Sollte ein Endempfänger nicht in geforderter Qualität betrieben werden, vermindert dieser Empfänger die Zuverlässigkeit des ganzen Systems. Alle Teilnehmer müssen sich deshalb auf eine minimale Zuverlässigkeit einigen.

Wir möchten eine kundenorientierte Sicht einnehmen, das heisst eine Internetlösung wird sich nicht nach üblichen Bürozeiten richten. Andererseits müssen beim Betrieb eines Endempfängers die ökonomischen Aspekte (Finanzierbarkeit) ebenfalls berücksichtigt werden.

Die Verfügbarkeiten der Systeme sind daher als zukünftige Zielwerte zu verstehen, das heisst die Bedeutung der Lösung nimmt zu und damit auch seine Verfügbarkeit. Ziel ist eine pragmatische Lösung ("lightweight construction" und "best effort"). Definierte Zeitbereiche:

- 7 Tage pro Woche mal 24 Stunden
- Spitzenzeiten: 6 Uhr bis 20 Uhr

Definierte Wertebereiche:

- Spitzenzeiten: Verfügbarkeit der Endempfänger (m2m) mindestens 99,52%
- Randzeiten: Verfügbarkeit der Endempfänger (m2m) mindestens 93,00%

1.3. Schnittstellen

1.3.1. Swissdec

Die Swissdec-Schnittstelle wurde im itServe Produkt STEP bereits implementiert und bei mehreren Kunden installiert sowie von der Swissdec abgenommen. Die Swissdec Dokumentation ist auf der Webseite [http://www.swissdec.ch] des Vereins verfügbar.

1.3.2. sM-Client

Zwischen dem Swissdec-Adapter und dem *sM-Client* kommt eine filebasierte Schnittstelle zum Einsatz. Der Adapter schreibt die eingehenden Meldungen in ein Inbox-Verzeichnis, das vom *sM-Client* regelmässig gelesen wird.

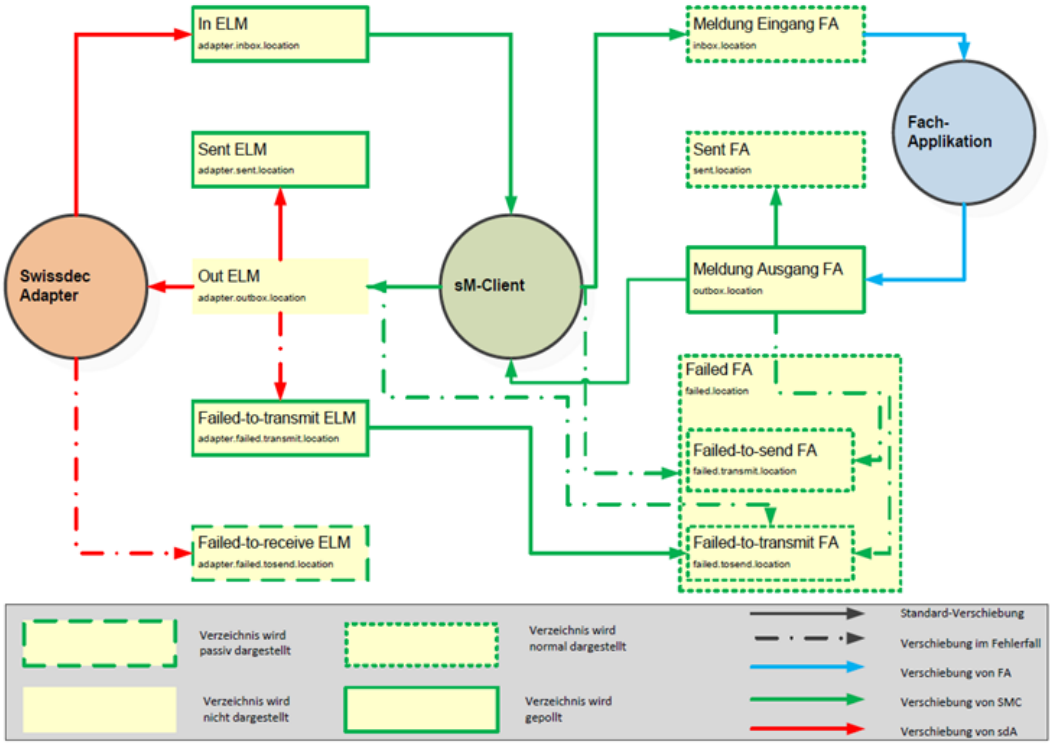

Abbildung 1.2. Dateisystemschnittstelle Swissdec-Adapter - sM-Client

sM-Client (Message-handler-elm.properties)	**PathID**	**Swissdec-Adapter (application.properties)**
Im sM-Client sind diese Verzeichnisse relativ zu base.dir.adapter		sdA braucht absoluten Pfad in dieser Konfigurationsdatei
adapter.inbox.location	PATH_ELM_IN	released.filesystem.parent.xml
adapter.outbox.location	PATH_ELM_OUT	result.filesystem.parent.xml
adapter.sent.location	PATH_ELM_SENT	sent.filesystem.parent.xml
adapter.failed.receive	PATH_ELM_FAILED_TO_RX	failed.filesystem.parent.xml
adapter.failed.transmit.location	PATH_ELM_FAILED_TO_TX	undeliverable.filesystem. parent.xml

Tabelle 1.1. Pfade der Schnittstelle Swissdec-Adapter - sM-Client

Die komplette Schnittstellenspezifikation entnehmen Sie bitte der [DETAILSPEZ]

1.4. Referenzen

- [ENDRECREQ] Swissdec Anforderungen Endempfänger
- [DETAILSPEZ] Detailspezifikation Swissdec-Adapter

2. Betriebsanforderungen

Der Swissdec-Adapter besteht aus zwei Java Applikationen. Beide Applikationen (Receiver und Integration) und benötigen eine Java Runtime. Das Installationspaket beinhaltet die vorkonfigurierten Applikationen und kann daher einfach entpackt werden.

Die Receiver-Applikation implementiert das Swissdec Protokoll und muss aus dem Internet via HTTPS erreichbar sein. Typischerweise wird diese in der *DMZ* installiert.

Die Integrationsapplikation stellt die Kommunikation mit dem *sM-Client* sicher. Diese Schnittstelle ist dateisystembasiert. Das heisst, die Integrationsapplikation und der *sM-Client* benötigen ein gemeinsames Dateisystem. Die Integrationsapplikation kann sowohl auf dem gleichen System wie der Receiver als auch auf einem dedizierten System oder dem System des *sM-Client*s installiert werden.

Die Kommunikation zwischen den beiden Applikationen geschieht mittels HTTP/SOAP. Es muss sichergestellt werden, dass eine solche Verbindung von der Receiver-Applikation (Client, initiiert immer die Verbindung) zur Integrationsapplikation geöffnet werden kann.

2.1. Systemanforderungen

2.1.1. Unterstützte Platformen

Der Swissdec-Adapter ist eine reine Java Applikation. Es wird eine installiertes Java JDK in der Version 11 vorausgesetzt. Empfohlen wird OpenJDK. Getestete und unterstützte Plattformen:

- Ubuntu 20.04 LTS
- Windows Server 64bit
- Windows 10

2.1.2. Als Dienst starten

Aufgrund der synchronen Swissdec Architektur muss der Swissdec-Adapter permanent laufen. Um dies sicherzustellen sollte der Swissdec-Adapter als Service (auf Windows) oder Start-Skript (auf Unix) konfiguriert werden.

2.1.3. Erreichbarkeit

Der SwissdecAdapter muss vom Distributor gemäss den Swissdec Richtlinien erreichbar sein. Details dazu gibt es im Kapitel Konnektivität Abschnitt 3.5, „Konfiguration SwissdecAdapter Receiver“ beim Receiver.

2.1.4. CPU

Die Anforderungen an die CPU sind gering. Es wird der Einsatz von aktuellen 64-Bit Multi-Core Prozessoren empfohlen.

Anmerkung

Durch die Swissdec-Architektur ergibt sich die Problematik, dass der Distributor auf alle Endempfänger warten muss, bevor die Quittung an den Kunden gesendet wird. Die Swissdec hat deshalb folgende Perfomance-Anforderungen spezifiziert:

- [ENDRECREQ], Kapitel 3.10: Eine Lohnmeldung mit 100 Personen sollte in weniger als 20 Sekunden verarbeitet sein.
- [ENDRECREQ], Kapitel 3.10: Eine Lohnmeldung mit 100 Personen muss in einer Minute verarbeitet werden können.
- [ENDRECREQ], Kapitel 3.10: Eine Lohnmeldung mit 2000 Personen sollte in einer Minute verarbeitet sein.

2.1.5. Arbeitsspeicher

Es sollten mindestens 1 Gigabyte für den Swissdec-Adapter reserviert werden. Damit lassen sich Meldungen bis etwa 6'000 Mitarbeiter empfangen. Falls grössere Meldungen empfangen werden sollen, empfehlen wir 4 Gigabytes zu reservieren. Damit lassen sich Meldungen bis etwa 20'000 Mitarbeiter empfangen.

Anmerkung

Seit der Version 2.1_0 verarbeitet der Swissdec-Adapter die Webservice Sicherheit streambasiert. Der Memory Verbrauch ist so deutlich effizienter.

2.1.6. Speicherplatz

Die Installation benötigt in etwa 150 Megabytes. Der benötigte Festplattenplatz zur Laufzeit variert deutlich und ist abhängig von der Anzahl und der Grösse der empfangenen Meldungen. Als Startwert wird 1 Gigabyte freier Festplattenplatz empfohlen.

2.1.7. Systemzeit

Es ist wichtig, dass die Systemzeit des Systems, auf dem der Swissdec-Adapter (insbesondere Receiver) läuft, korrekt ist. Dies weil auf Web Service Security Ebene ein Timestamp generiert wird. Wenn die Zeit des Distributors und des Endempfängers auseinander driftet, werden Meldungen abgelehnt. Es wird empfohlen, die Zeit automatisch mittels *NTP (Network Time Protocol)* zu aktualisieren.

2.2. Migrationspfad

Das Update von Version 2.x auf 3.0 kann ohne Datenverlust durchgeführt werden. Dafür müssen die folgenden Schritte durchgeführt werden. Der Ablauf wird hier grob beschrieben. Die Installation wird im Abschnitt 3.3, „Installation unter Windows“ und im Abschnitt 3.2, „Installation unter Linux“ genau beschrieben.

1. Wartungsfenster in der alten Installation erfassen, so dass eine korrekte Fehlermeldung erscheint, wenn die Migration durchgeführt wird. Dazu wird ein XML-File im konfigurierten Ordner abgelegt.

   Der Distributor sendet alle 30 Minuten eine Ping Operation, bei welcher die Informationen zum Wartungsfenster aus dem XML-File auf dem Distributor gesichert werden.

2. Stoppen der alten SwissdecAdapter Services (Integration und Receiver)

3. Sichern der Datenbank

   Die Datenbank befindet sich im Ordner derbydb. Der Ordner derbydb muss gesichert werden. Wenn die neue Installation korrekt startet, kann die mitgelieferte Datenbank mit dem Ordner aus der alten Installation ersetzt werden.

4. Wir empfehlen die Konfiguration der alten Installation zu sichern. Die Konfiguration besteht aus folgenden Files:

   <SDA_INTEGRATION>/conf/server.xml
   <SDA_INTEGRATION>/conf/swissdecAdapter.properties
   <SDA_RECEIVER>/conf/server.xml
   <SDA_RECEIVER>/conf/swissdecAdapter.properties

   Die Konfigruations-Differenzen zwischen 2.x und 3.x werden im Abschnitt A.3, „Konfigurationsvergleich 2.x zu 3.x“ beschrieben

5. Sichern der Zertifikate:

   Der Pfad zu den Zertifikaten ist in den beiden Konfigurationsfiles des Receivers ersichtlich. Diese Files müssen gesichert werden, da diese in der neuen Installation wieder benötigt werden.

6. Installation von Java 11 (Empfohlen wird OpenJDK)

7. Entpacken, Installieren und Konfigurieren der neuen SwissdecAdapter Applikationen, gemäss Abschnitt 3.4, „Konfiguration SwissdecAdapter Integration“ und Abschnitt 3.5, „Konfiguration SwissdecAdapter Receiver“ .

8. Neue Installation testen mit mitgeliefertem Testtool

9. Stoppen der SwissdecAdapter Integration, Ersetzen der mitgelieferten leeren Datenbank mit der gesicherten (Ordner derbydb von der alten Installation) und Neustarten der Integration.

10. Damit ist nun der SwissdecAdapter in der Version 3.0.0 installiert.

3. Installation und Konfiguration

3.1. Delivery Paket

Im Delivery Paket, das als Zip File ausgeliefert, wird befindet sich die Dokumentation, die Applikationen und das Test-Tool um die Funktionalität des SwissdecAdapters lokal zu testen.

Verzeichnis	**Beschreibung**
doc	Enthält das Betriebshandbuch und die Detailspezifikation
integration	Enthält die SwissdecAdapter-Integration Applikation (swissdecAdapter-integration.jar), die Standardkonfiguration (application.properties), die Datenbank (derbydb) und die Windows Service-Installations Wrapper Dateien.
receiver	Enthält die SwissdecAdapter-Receiver Applikation (swissdecAdapter-receiver.jar), die Standard-Konfiguration (application.properties) und die Windows Service-Installations Wrapper Dateien.
swissdecAdapter-testtool.zip	Testtool mit Testapplikation, Testdaten und Testdokumentation

Tabelle 3.1. swissdecAdapter-delivery-3.0_5.zip

3.2. Installation unter Linux

Die Installation wurde unter Ubuntu 20.04 LTS getestet. Sie kann aber unter jedem Linux-Derivat installiert werden.

Anmerkung

Die Installationsanleitung geht davon aus, dass der Swissdec Adapter im Verzeichnis /opt/swissdecAdapter installiert wird. Selbstverständlich kann auch in einem anderen Verzeichnis installiert werden.

Die Applikation wird in der Anleitung vom System User "springboot" gestartet.

Diese Anleitung hat nicht den Anspruch, dass sie auf jedem System funktionieren wird, sondern soll einem Systemadministrator die Leitplanken geben, so dass die Installation selbstständig durchgeführt werden kann.

3.2.1. Systemvorbereitung

Der SwissdecAdapter benötigt ein Java Development Kit (JDK) in der Version 11.

Die Installation von Java 11 kann über apt, yum oder manuell erfolgen. Wichtig ist, dass die Umgebungsvariablen korrekt gesetzt sind, so dass der Befehl "java -version" die entsprechende JDK Version ausgibt.

Es wird empfohlen den SwissdecAdapter mit einem System User zu starten.

3.2.2. Installation der Applikationen

Falls noch kein System User besteht wird empfohlen, einen zu erstellen:

```
sudo useradd -r springboot
```

Die Ordner integration und receiver aus dem Delivery-Paket (Abschnitt 3.1, „Delivery Paket“) werden im Ordner /opt entpackt.

Die "wrapper" Dateien können entfernt werden. Diese werden nur für die Windows Installation benötigt.

```
unzip swissdecAdapter-delivery-3.0_5.zip
sudo mkdir /opt/swissdecAdapter
sudo cp -r swissdecAdapter-delivery-3.0_5/integration /opt/swissdecAdapter/.
sudo cp -r swissdecAdapter-delivery-3.0_5/receiver /opt/swissdecAdapter/.
sudo rm /opt/swissdecAdapter/*/*wrapper*
```

Bevor die Applikationen korrekt gestartet werden können, müssen noch die entsprechenden Berechtigungen gesetzt werden:

```
sudo chown -R springboot:springboot /opt/swissdecAdapter/integration
sudo chown -R springboot:springboot /opt/swissdecAdapter/receiver
sudo chmod +x /opt/swissdecAdapter/integration/swissdecAdapter-integration.jar
sudo chmod +x /opt/swissdecAdapter/integration/swissdecAdapter-receiver.jar
```

Die Applikationen sind nun bereit, um gestartet zu werden. Es wird empfohlen aus dem Installationsverzeichnis zu starten, da die Konfigurationsangaben zur Datenbank relativ gesetzt sind.

```
cd /opt/swissdecAdapter/integration
java -jar swissdecAdapter-integration.jar
cd /opt/swissdecAdapter/receiver
java -jar swissdecAdapter-receiver.jar
```

3.2.3. Service Installation

3.2.3.1. /etc/init.d Service

Die Installation als Service funktioniert mit einem Softlink auf das Jar File. Da die Version nicht im Dateinamen enthalten ist, wird der Service auch nach künftigen Updates funktionieren.

```
sudo ln -s /opt/swissdecAdapter/integration/swissdecAdapter-integration.jar
/etc/init.d/swissdecAdapter-integration
sudo update-rc.d swissdecAdapter-integration defaults
sudo ln -s /opt/swissdecAdapter/integration/swissdecAdapter-receiver.jar
/etc/init.d/swissdecAdapter-receiver
sudo update-rc.d swissdecAdapter-receiver defaults
```

3.2.3.2. systemd Installation

Bei moderneren Linux Installationen wird häufig über systemd installiert. Dazu werden folgende Files benötigt:

/etc/systemd/system/swissdecAdapter-integration.service

```
[Unit]
Description=SwissdecAdapter Integration
After=syslog.target

[Service]
User=springboot
ExecStart=/opt/swissdecAdapter/integration/swissdecAdapter-integration.jar
SuccessExitStatus=143

[Install]
WantedBy=multi-user.target
```

/etc/systemd/system/swissdecAdapter-receiver.service

```
[Unit]
Description=SwissdecAdapter Integration
After=syslog.target

[Service]
User=springboot
ExecStart=/opt/swissdecAdapter/integration/swissdecAdapter-receiver.jar
SuccessExitStatus=143

[Install]
WantedBy=multi-user.target
```

3.3. Installation unter Windows

Die Installation wurde unter Windows 10 getestet, kann aber unter jedem modernen Windows System installiert werden.

Anmerkung

Die Installationsanleitung geht davon aus, dass der Swissdec Adapter im Verzeichnis C:\swissdecAdapter installiert wird. Selbstverständlich kann auch in einem anderen Verzeichnis installiert werden.

Diese Anleitung hat nicht den Anspruch, dass sie auf jedem System funktionieren wird, sondern soll einem Systemadministrator die Leitplanken geben, so dass die Installation selbstständig durchgeführt werden kann.

3.3.1. Systemvorbereitung

Der SwissdecAdapter benötigt ein Java Development Kit (JDK) in der Version 11.

Die Installation kann über einen Windows Installer oder per Installation in ein beliebiges Verzeichnis erfolgen. Es ist von Vorteil, dass die Umgebungsvariablen korrekt gesetzt sind, so dass der Befehl in der Kommandozeile "java -version" die entsprechende JDK Version ausgibt.

3.3.2. Installation der Applikation

Die Ordner integration und receiver aus dem Delivery-Paket (Abschnitt 3.1, „Delivery Paket“) werden im Ordner C:\swissdecAdapter entpackt.

Die Applikationen sind nun bereit um gestartet zu werden. Es wird empfohlen aus dem Installationsverzeichnis zu starten, da die Konfigurationsangaben zur Datenbank relativ gesetzt sind.

```
cd C:\swissdecAdapter\integration
java -jar swissdecAdapter-integration.jar
cd /opt/swissdecAdapter/receiver
java -jar swissdecAdapter-receiver.jar
```

3.3.3. Service Installation

Um die SwissdecAdapter Applikationen als Service zu installieren wird der Windows Service Wrapper (github.com/winsw/winsw [https://github.com/winsw/winsw]) mitgeliefert. Das Executable mit der entsprechenden Konfiguration liegt im Verzeichnis integration bzw. receiver.

Um die Dienste zu installieren müssen im Command Prompt (als Administrator ausführen) folgende Befehle abgesetzt werden:

```
C:\swissdecAdapter\integration\swissdecAdapter-integration-wrapper.exe install
C:\swissdecAdapter\receiver\swissdecAdapter-receiver-wrapper.exe install
```

Die Konfiguration für den Service Wrapper kann in der Datei swissdecAdapter-*-wrapper.xml angepasst werden:

Property	**Beschreibung**
id	Eindeutige ID des künftigen Windows Dienstes
name	Anzeigename des Dienstes
description	Beschreibung des Dienstes
executable	Ausfzuführendes Executable. Falls nicht die Java Version aus der Umgebungsvariable gestartet werden soll, kann hier auch direkt auf ein java.exe aus einer alternativ installierten Java Version umgestellt werden.
arguments	Argumente hinter dem Executable.
logpath	Hier werden die Logs geschrieben.

Property	**Beschreibung**
log mode	Der Default Wert wird auf "none" geschaltet. Das Applikationslog wird von Spring Boot gesteuert (application.properties)

Tabelle 3.2. Beschreibung WinSW-Konfiguration

3.4. Konfiguration SwissdecAdapter Integration

Die Konfiguration der SwissdecAdapter Integration befindet sich in der Datei application.properties im Verzeichnis integration/conf.

3.4.1. System-Konfiguration

Option	Standardwert	Beschreibung
server.port	9090	TCP Port auf dem die Applikation gestartet wird. Muss im Receiver-Teil entsprechend konfiguriert werden
derby.system.home	derbydb	Pfad zur Datenbank
housekeeping.days	180	Wieviele Tage werden die Deklarationen gesichert
monitoring.enabled	false	Wird die Monitoring-Seite gestartet. (Erreichbar unter http://[HOST]:[PORT]/api/monitoring)
monitoring.user	admin	Benutzername für die Monitoring-Seite
monitoring.pass	admin	Passwort für den Zugang zur Monitoring-Seite

Tabelle 3.3. SwissdecAdapter Integration technische Systemkonfiguration (integration/conf/application.properties)

3.4.2. Applikation-Konfiguration

Option	Standardwert	Beschreibung
institution.canton	BE	Technische Angabe, für welche Institution QST- Meldungen empfangen werden (zum Beispiel „BE“, „ZH“, „GE“, ...). MUSS richtig konfiguriert sein, da sonst Meldungen abgelehnt werden
await.result	false	Sind QST-Abrechnungsresultate zu erwarten? Wird nur benutzt, wenn eine Backend-Integration verfügbar ist.
processing.default.hours	48	Nach wievielen Stunden kann ein allfälliges QST-Abrechnungsresultat erwartet werden.
testcase.auto.quittance	false	Verhindert die Backend-Integration beim Empfang einer Testdeklaration. (nur aktiv wenn await.result=true)
write.original.xml	false	Schreibt bei aktivem Splitting die vollständige Originaldatei auch in die ZIP-Datei.
commune.splitting	false	Aktiviert Gemeindesplitting für QST-Abrechnungen. Bei aktivem Gemeindesplitting wird await.result implizit „false“ gesetzt, da Gemeinden keine Antworten geben können.
la.splitting	false	Aktiviert Lohnausweissplitting pro empfangene Person.

Option	Standardwert	Beschreibung
map.tas.toV5	false	QST-Meldungen werden von ELM v4.0 nach ELM v5.0 gemappt, falls die Bedingung von map.tas.fromPeriod erfüllt sind.
map.tas.fromPeriod	2021-01	Wenn map.tas.toV5 aktiviert ist werden alle QST-Meldungen, die neuer oder gleich sind wie die gesetzte Periode, gemappt. Es wird mit dem CurrentMonth aus der Lohnmeldung verglichen. (SalaryDeclaration/ Staff/ Person/ TaxAtSourceSalaries/ TaxAtSourceSalary/ CurrentMonth)

Tabelle 3.4. SwissdecAdapter Integration fachliche Applikationskonfiguration (integration/conf/application.properties)

3.4.3. Datenkonfiguration

Option	Standardwert	Beschreibung
released.filesystem.parent.xml	data/received	Pfad zum Meldungseingang (PATH_ELM_IN). Bitte absolute Pfadangabe. Muss mit dem sM-Client korrespondieren. Beispiel für Windows: C:/sdA/data/received
failed.filesystem.parent.xml	data/failed	Pfad zum "failed to receive" (PATH_ELM_FAILED_TO_RX). Bitte absolute Pfadangabe. Muss mit dem sM-Client korrespondieren. Beispiel für Windows: C:/sdA/data/failed
result.filesystem.parent.xml	data/result	Pfad zum Meldungsausgang (PATH_ELM_OUT). Bitte absolute Pfadangabe. Muss mit dem sM-Client korrespondieren. Beispiel für Windows: C:/sdA/data/result
sent.filesystem.parent.xml	data/sent	Pfad zum "sent" (PATH_ELM_SENT). Bitte absolute Pfadangabe. Muss mit dem sM-Client korrespondieren. Beispiel für Windows: C:/sdA/data/sent
undeliverable.filesystem.parent.xml	data/undeliverable	Pfad zum "failed to send" (PATH_ELM_FAILED_TO_TX). Bitte absolute Pfadangabe. Muss mit dem sM-Client korrespondieren. Beispiel für Windows: C:/sdA/data/undeliverable

Tabelle 3.5. Swissdec-Adapter Integration Grundkonfiguration (integration/conf/application.properties)

3.4.4. Verzeichniskonfiguration

Folgende Tabelle gibt eine Übersicht über die Verzeichnisse im Swissdec-Adapter, welche für die Übermittlung der Meldungen eine Rolle spielen.

PathID	**Beschreibung**
PATH_ELM_IN	• sind im Swissdec XML-Standard (schemavalid) • haben den Dateinamen: ["Tax"|"TaxAtSource"|"Tax5"|"TaxAtSource5"|"TaxCrossborder5"]_ DeclarationId_ResultIdentifier_["MIXD"|"SPLT"].zip • konsumierte Meldungen werden vom sM-Client verschoben/gelöscht
PATH_ELM_OUT	• Antworten werden nur für Quellensteuerabrechnungen erstellt (Quellensteuerabrechnungsresultat) • sind im Swissdec XML-Standard (Schemavalid), gezippt • haben den Dateinamen: DeclarationId.zip • konsumierte Meldungen werden vom Swissdec-Adapter verschoben • nicht konsumierte Meldungen werden vom Swissdec-Adapter verschoben
PATH_ELM_SENT	• Versendete Quellensteuerabrechnungsresultate • sind im Swissdec XML-Standard (Schemavalid), gezippt • haben den Dateinamen: DeclarationId_ResultIdentifier.zip • wird vom sM-Client gelesen, um den Prozess abschliessen zu können
PATH_ELM_FAILED_TO_RX	• Lohnausweis und QST-Abrechnung, Fehler beim Empfang • die Antworten des Swissdec-Adapters werden auch geschrieben • nicht in jedem Fall vorhanden
PATH_ELM_FAILED_TO_TX	• QST-Abrechnungsresultate, Fehler beim Versand

Tabelle 3.6. Swissdec-Adapter Verzeichnisse

3.5. Konfiguration SwissdecAdapter Receiver

Die Konfiguration des SwissdecAdapter Receivers befindet sich in der Datei application.properties im Verzeichnis receiver/conf.

3.5.1. Systemkonfiguration

Option	Standardwert	Beschreibung
server.port	8080	TCP Port auf dem die Applikation gestartet wird.
integration.service.host	localhost	Auf diesem Host läuft die Swissdec-Adapter Integration. (Kann auch IP Adresse sein, wenn kein DNS vorhanden ist.)
integration.service.port	9090	Port auf dem die SwissdecAdapter Integration läuft.
integration.service.protocol	http://	Über welches Protokoll ist die Swiss-decAdapter Integration erreichbar.

Tabelle 3.7. SwissdecAdapter Receiver technische Systemkonfiguration (receiver/conf/ application.properties)

3.5.2. Applikationskonfiguration

Option	Standardwert	Beschreibung
institution.canton	BE	Technische Angabe, für welche Institution QST- Meldungen empfangen werden (zum Beispiel „BE“, „ZH“, „GE“, ...). MUSS richtig konfiguriert sein, da sonst Meldungen abgelehnt werden
tax.accept.ex	false	Definiert, ob Lohnausweise für im Ausland wohnhafte Personen akzeptiert werden. Bitte sprechen Sie mit Ihrer Fachabteilung, ob diese Option gewünscht ist. Falls ja, nehmen Sie bitte Kontakt mit Swissdec auf (JIRA oder E-Mail), da dies auf dem Distributor konfiguriert werden muss.
institution.name	KSTV Bern	Beschreibender Name der Steuerverwaltung. Achtung! Sonderzeichen müssen mit Unicode Zeichen ersetzt werden. Beispiel gibt es im Abschnitt A.4, „Unicode Tabelle für gängige Sonderzeichen“
elm.tac.enabled	true	Die Installation ist bereit für den Empfang von Lohnausweisen von Grenzgängern
elm.tax.enabled	true	Die Installation ist bereit für den Empfang von Lohnausweisen mit ELM v5.0.
elm.tas.enabled	true	Die Installation ist bereit für den Empfang von Quellensteuermeldungen mit ELM v5.0.
elm4.tax.enabled	true	Die Installation ist bereit für den Empfang von Lohnausweisen mit ELM v4.0.

Option	Standardwert	Beschreibung
elm4.tas.enabled	true	Die Installation ist bereit für den Empfang von Quellensteuermeldungen mit ELM v4.0.

Tabelle 3.8. SwissdecAdapter Receiver fachliche Applikationskonfiguration (receiver/conf/application.properties)

3.5.3. Konfiguration der Security

Die Funktionsweise der Security wird im Kapitel 4, *Security* detailliert beschrieben.

3.5.3.1. WS-Security

Das Zertifikat für die WS-Security ist in einem Java Keystore (*.jks) gesichert. Die Konfiguration wird folgendermassen gemacht:

Option	Standardwert	Beschreibung
wss4j.org.apache.ws.security.crypto.provider	org.apache.wss4j.common.crypto.Merlin	Security Provider
wss4j.org.apache.ws.security.crypto.merlin.file	security/TestConsumerAll.jks	Pfad zum Java Keystore
wss4j.org.apache.ws.security.crypto.merlin.keystore.type	jks	Keystore Typ
wss4j.org.apache.ws.security.crypto.merlin.keystore.password	default	Passwort um den Java Keystore zu entschlüsseln
wss4j.org.apache.ws.security.crypto.merlin.keystore.alias	1	Alias Name
wss4j.privateKeyPassword	default	Passwort zum Private Key

Tabelle 3.9. SwissdecAdapter Receiver WS-Security (receiver/conf/application.properties)

3.5.3.2. SSL/TLS Security

Die Konfiguration für die Transport Layer Security (TLS) kann direkt vom Swissdec-Adapter übernommen werden. Es wird allerding empfohlen den HTTPS Traffic auf einem anderen System zu machen (Z.B.: Apache, WAF, etc.).

Option	Standardwert	Beschreibung
server.ssl.key-store	security/TestConsumerAll.jks	Pfad zum Keystore
server.ssl.key-store-type	JKS	Typ des Keystore (JKS oder PKCS)
server.ssl.key-store-password	default	Passwort zum Java Keystore
server.ssl.key-alias	1	Alias Name
server.ssl.key-password	default	Passwort zum Private Key
server.ssl.trust-store	security/truststore.jks	Pfad zum Truststore (für Mutual Authentication)
server.ssl.trust-store-password	default	Passwort zum Truststore (für Mutual Authentication)

Tabelle 3.10. SwissdecAdapter Receiver SSL/TLS Security (receiver/conf/application.properties)

Anmerkung

Die Erfahrung im Rollout hat gezeigt, dass gerade auf virtuellen Windows Systemen die oben beschriebene Konfiguration zu Fehlern führt.

Steht kein anderes System zur Terminierung von SSL/TLS zur Verfügung kann auf dem Host, auf dem auch der SwissdecAdapter-Receiver installiert ist, ein Apache Reverse Proxy installiert werden.

Abschnitt B.2, „Installationsanleitung für Apache Reverse Proxy“

3.5.4. Konnektivität

Der Swissdec-Adapter startet einen Webservice-Endpunkt, der aus dem Internet erreichbar sein muss. Es muss also ein Pfad aus dem Internet zum Swissdec-Adapter geschaffen werden. Dazu müssen folgende Dinge konfiguriert werden:

- URL: Um aus dem Internet erreichbar zu sein, benötigt der Swissdec-Adapter fünf URLs. Die Swissdec schreibt vor, dass pro Domäne (Quellensteuer/TaxAtSource, Steuern/Tax/Lohnausweis) eine eindeutige Adresse vergeben werden muss. Mehr dazu im Abschnitt 3.5.4.1, „Reverse Proxy“.
- SSL/TLS: Um die Verbindung abzusichern, schreibt die Swissdec SSL/TLS mit Klienten-Authentifizierung vor. Das heisst nebst „normalem“ HTTPS muss vom Klienten ein Zertifikat verlangt und geprüft werden. Mehr dazu im Abschnitt 3.5.4.2, „SSL/TLS“.
- Firewall: Der Swissdec-Adapter öffnet keine Verbindungen ins Internet. Daher muss nur der eingehende Pfad konfiguriert werden. Die Verbindung kann auf den Distributor eingeschränkt werden. Mehr dazu im Abschnitt 3.5.4.3, „Firewall“.

3.5.4.1. Reverse Proxy

Falls ein Reverse Proxy im Einsatz ist, müssen folgende URLs von aussen erreichbar sein:

```
http://<HOST>:<PORT>/webservice/tax/SalaryDeclarationConsumerService
```

```
http://<HOST>:<PORT>/webservice/tas/SalaryDeclarationConsumerService
```

```
http://<HOST>:<PORT>/webservice/tax/SalaryDeclarationConsumerService20200220
```

```
http://<HOST>:<PORT>/webservice/tas/SalaryDeclarationConsumerService20200220
```

```
http://<HOST>:<PORT>/webservice/tac/SalaryDeclarationConsumerService20200220
```

3.5.4.2. SSL/TLS

Die Swissdec schreibt für Endempfänger nebst „normalem“ HTTPS vor, dass der Klient (Distributor) sich authentifizieren muss. Dies als Schutz der Endempfänger. Grundsätzlich liegt die Entscheidung, wie und wo das SSL/TLS mit Mutual Authentication terminiert wird, beim Endempfänger (Betreiber). Folgende Lösungen sind dabei denkbar:

- Die Terminierung findet auf dem Entry-Server oder der Firewall statt.
- Die Terminierung findet auf dem Receiver des Swissdec-Adapters statt. Die Konfiguration hierzu finden Sie im Anhang.

Die Zertifikate für beide Lösungen erhalten Sie via *sM-Client* Support (https://jira.ctp-consulting.com/browse/SMCSUPPORT).

3.5.4.3. Firewall

Es müssen eingehende Verbindungen auf den Swissdec-Adapter erlaubt werden. Per default wird der Swissdec-Adapter Receiver auf Port 8080 gestartet.

Anmerkung

Da der Swissdec-Adapter von nur einem Klienten (Distributor) bedient wird, kann die eingehende Verbindung auf dessen IP restriktiert werden. Für den produktiven Distributor ist dies die IP:

- 194.11.148.11

Der Testdistributor der Swissdec Referenz Applikationen hat folgende IPs:

- 193.247.121.163
- 193.247.102.165

4. Security

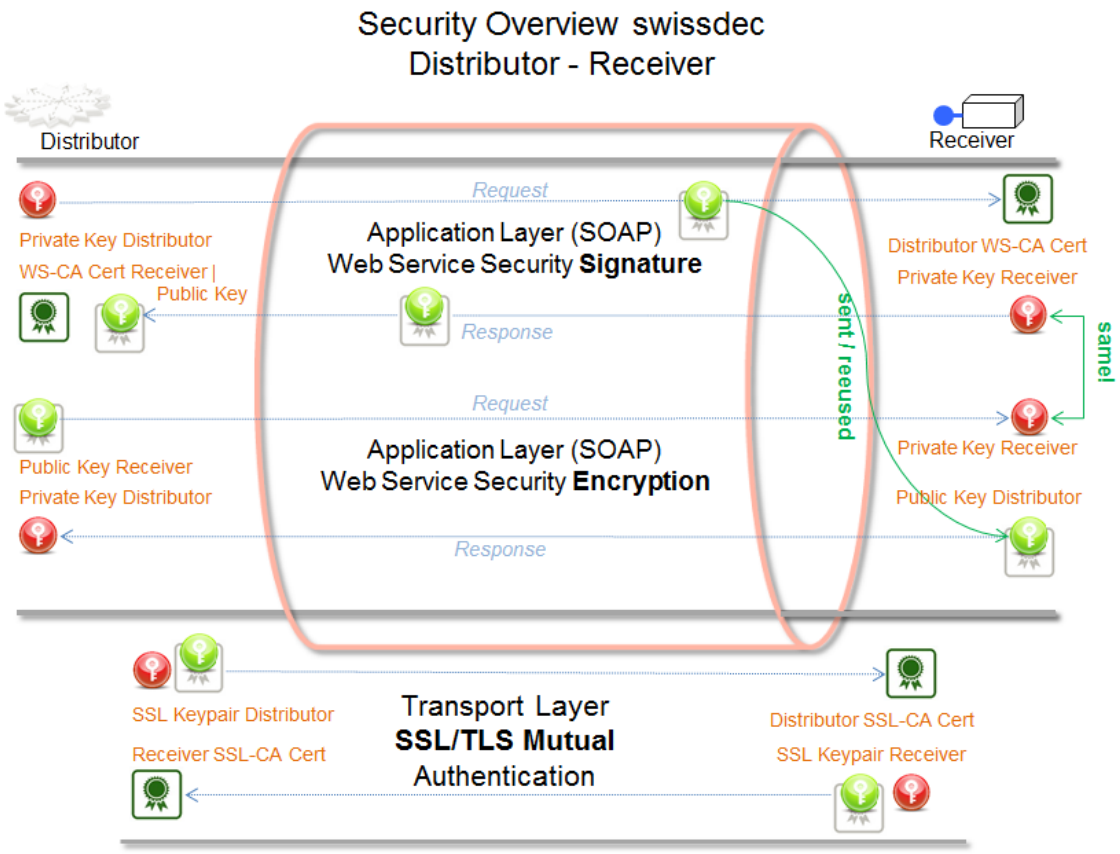

Abbildung 4.1. Security Overview

	Privater Schlüssel (Private Key)
	Zertifikat mit öffentlichem Schlüssel (Public Key). Im Folgenden werden "öffentlicher Schlüssel" (public Key) gleichgesetzt mit dem Zertifikat, welches den öffentlichen Schlüssel beinhaltet.
	CA Zertifikat

Tabelle 4.1. Legende zu Abbildung Security Overview

4.1. Transport Layer (SSL/TLS)

Der Client baut eine Verbindung zum Server auf. Für gewöhnlich authentifiziert sich zuerst der Server gegenüber dem Client mit einem Zertifikat. Dann schickt entweder der Client dem Server eine mit dem öffentlichen Schlüssel des Servers verschlüsselte geheime Zufallszahl, oder die beiden Parteien berechnen mit dem Diffie-Hellman-Schlüsselaustausch ein gemeinsames Geheimnis. Aus dem Geheimnis wird dann ein kryptographischer Schlüssel abgeleitet. Dieser Schlüssel wird in der Folge benutzt, um alle Nachrichten der Verbindung mit einem symmetrischen Verschlüsselungsverfahren zu verschlüsseln und zum Schutz von Nachrichtenintegrität und -authentizität durch einen Message Authentication Code abzusichern.

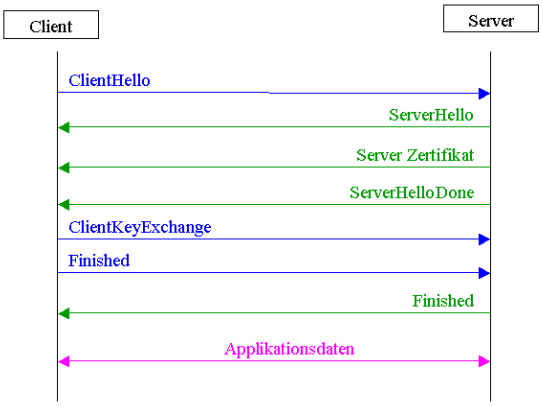

Abbildung 4.2. SSL Handshake

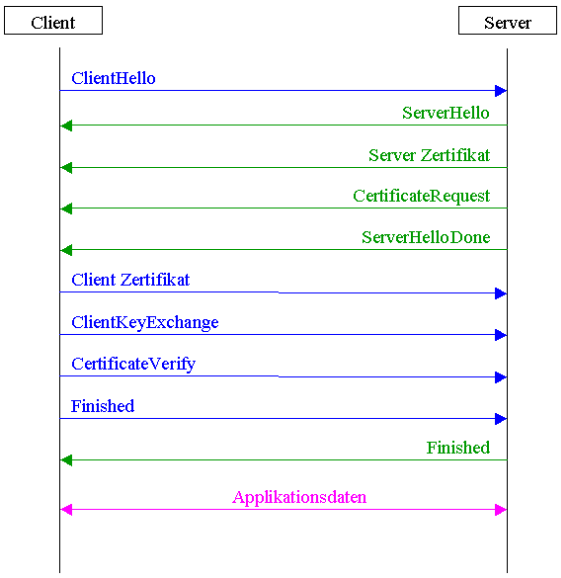

Abbildung 4.3. SSL Handshake mit Mutual Authentication

Bei der Mutual Authentication (gegenseitige Authentifizierung) verlangt der Server vom Klienten ebenfalls, dass sich dieser mit einem Zertifikat authentifiziert. Dies ist im Swissdec-Standard vorgeschrieben um sicherzustellen, dass alle Parteien einander gegenseitig vertrauen.

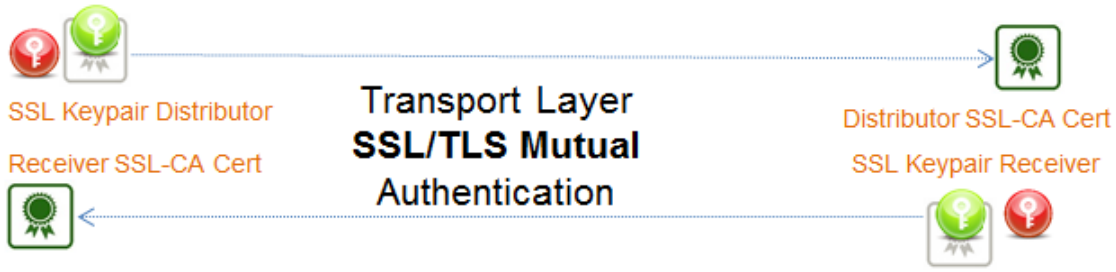

Abbildung 4.4. Transport Layer Security Overview

In der vereinfachten Ansicht wird ersichtlich, dass auf beiden Seiten (Klient/Swissdec Distributor, Server/Swissdec-Adapter) sowohl ein Schlüssel-Paar (privater Schlüssel mit dazugehörigem öffentlichen Schlüssel) um sich auszuweisen sowie das CA-Zertifikat des Gegenübers zum verifizieren benötigt wird.

Die SSL/TLS Konfiguration (Tabelle 3.10, „SwissdecAdapter Receiver SSL/TLS Security (receiver/conf/application.properties)“) für den Swissdec-Adapter befindet sich in der Datei "application.properties". Falls die Terminierung auf dem Adapter gemacht werden soll, befindet sich eine beispielhafte Konfiguration in der selben Datei.

Anmerkung

Keystore (mit dem Schlüsselpaar) und Truststore (mit dem CA-Zertifikat des Swissdec Distributors) werden der Einfachheit halber in einer Datei geliefert. Damit sind beide Angaben identisch!

Die produktiven Zertifikate können via sM-Client Support (https://smcsupport.atos-solutions.ch/) bestellt werden. Dazu muss der Hostname (DNS-Eintrag) bekannt gegeben werden, mit dem die Swissdec-Adapter Installation aus dem Internet erreichbar ist.

4.2. Webservice Security

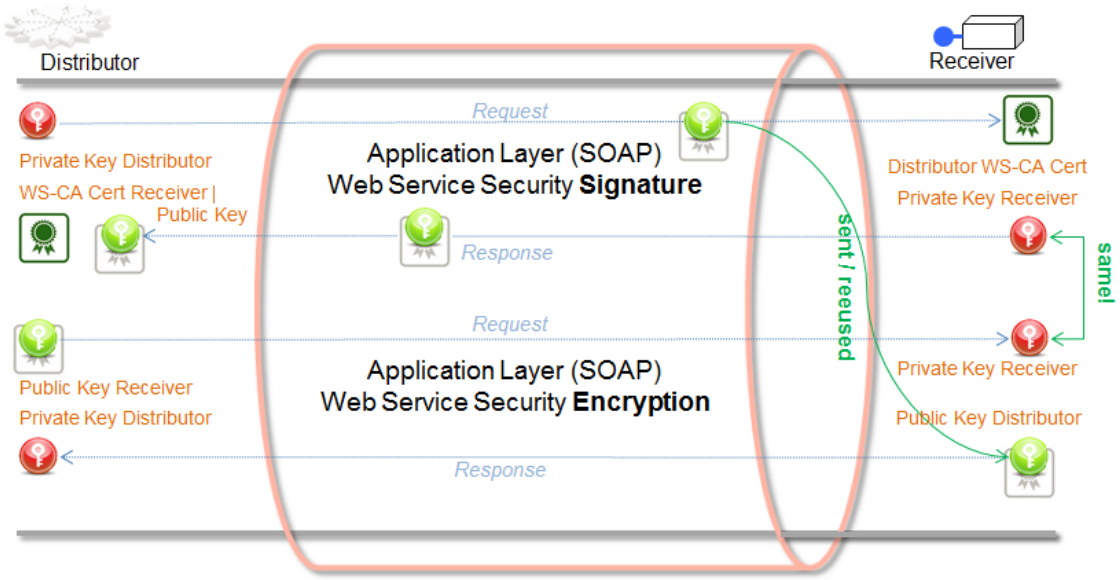

Abbildung 4.5. Webservice Security Overview

4.2.1. Signatur

Request	Der Klient (Swissdec Distributor) signiert den Request mit seinem privaten Schlüssel (Private Key) und schickt seinen öffentlichen Schlüssel (Public Key) mit dem Request mit. Auf Serverseite (Swissdec-Adapter) wird der mitgeschickte öffentliche Schlüssel gegen Swissdec CA-Zertifikat geprüft und die Signatur verifiziert. Serverseitig wird also nur das Swissdec CA-Zertifikat benötigt, um die Signatur zu prüfen.
Response	Der Swissdec-Adapter signiert die Antwort mit seinem privaten Schlüssel. Dem Swissdec Distributor muss dessen öffentlicher Schlüssel, beziehungsweise das CA-Zertifikat bekannt sein.

Tabelle 4.2. Webservice Security Signature

Für den Webservice Security Signatur Prozess wird auf Serverseite also das CA-Zertifikat des Klienten sowie das eigene Schlüsselpaar (privater und öffentlicher Schlüssel) benötigt.

4.2.2. Verschlüsselung

Request	Der Klient (Swissdec Distributor) verschlüsselt den Request mit dem öffentlichen Schlüssel (Public Key) des Empfängers. Auf Serverseite (Swissdec-Adapter) wird der Request mit dem eigenen privaten Schüssel dechiffriert.
Response	Der Swissdec-Adapter verschlüsselt die Antwort mit dem öffentlichen Schlüssel des Swissdec Distributors, der im Request für die Webservice Security Signatur mitgeschickt wurde. Damit wird sichergestellt, dass nur der Inhaber des dazugehörigen privaten Schlüssels die Antwort dechiffrieren kann.

Tabelle 4.3. Webservice Security Encryption

Für den Webservice Security Encryption Prozess wird auf Serverseite nur der private Schlüssel des Endempfängers benötigt, der der Gleiche ist wie für die Webservice Security Signatur.

4.2.3. Konfiguration

Aus der Abbildung 3.2 geht hervor, dass serverseitig (Swissdec-Adapter) für die Webservice Security zwei Dinge benötig werden:

- Swissdec CA-Zertifikat
- Schlüsselpaar (privater und öffentlicher Schlüssel) des Endempfängers

Da das CA-Zertifikat nur bekannt sein muss, wird es nicht explizit konfiguriert. Es reicht, wenn es im konfigurierten *Keystore* vorhanden ist. Der private Schlüssel des Endempfängers muss jedoch explizit im *Keystore* adressiert werden. Die folgenden Konfigurationsoptionen (Tabelle 3.9, „SwissdecAdapter Receiver WS-Security (receiver/conf/application.properties)“) (application.properties) steuern das Verhalten des Swissdec-Adapters für die Webservice Security.

Anmerkung

Die produktiven Zertifikate können via *sM-Client* Support (http://jira.ctp-consulting.com) bestellt werden.

Warnung

Sobald die Webservice Security Zertifikate für die Produktion installiert sind, kann sowohl von den Refapps wie auch vom TestTool keine Übermittlung mehr akzeptiert werden. Dies weil die Signaturen der Meldungen nicht mehr verifiziert werden können, da der produktive Distributor Zertifikate einer anderen CA (Certificate Authority) verwendet.

5. Hinweise für den Betrieb

5.1. Wartungsfenster

Da die Meldungen auf der Swissdec Platform synchron (Echtzeit) übertragen werden sehen die Kunden bei der Übermittlung, welche Endempfänger nicht erreichbar waren. Es besteht die Möglichkeit, geplante Betriebsunterbrüche mittels Wartungsfenster anzukündigen. Diese sollten nach Möglichkeit zwischen 20:00 und 6:00 Wochentags oder an Wochenenden geplant werden. Damit wird den Kunden eine Mitteilung anstelle eines Fehlers geliefert. Um ein Wartungsfenster zu registrieren, gehen Sie wie folgt vor:

1. Erstellen Sie ein "PlannedMaintenance" XML. Im Verzeichnis integration/templates finden Sie ein Template (PlannedMaintenance.xml).

2. Speichern Sie es bei der Swissdec-Adapter Integrationsapplikation ins Verzeichnis "maintenance.filesystem.parent.xml". Der Name spielt dabei keine Rolle, die Dateiendung muss ".xml" sein.

3. Das Wartungsfenster wird beim nächsten Ping des Distributors mitgeteilt. Die Erreichbarkeit des Endempfängers wird vom Distributor aus zyklisch (alle 30 Minuten) geprüft. Dazu wird die Webservice-Operation "PingConsumer" des Endempfängers aufgerufen, der seinerseits die Erreichbarkeit mit der Antwort bestätigt. In diesem Prozess können in der Antwort des Endempfängers Wartungsfenster registriert werden.

4. Sobald das Wartungsfenster vorbei ist, wird das "PlannedMaintenance" XML automatisch gelöscht.

Anmerkung

- Der Dateiname spielt keine Rolle, die Dateiendung muss zwingend ".xml" sein
- Die Zeiten "Start" und "End" im "PlannedMaintenance" XML sind vom Typ xs:dateTime. Das heisst, dass auch die Zeitzone sowie Sommer-/Winterzeit darin enthalten sein müssen.
  - 2013-08-31T13:40:47.0Z: das Z steht für UTC
  - 2014-01-31T13:40:47.0+01:00: UTC +1 Stunde, entspricht mitteleuropäischer Zeit (Winterzeit)
  - 2013-08-31T13:40:47.0+02:00: UTC +2 Stunden, entspricht mitteleuropäischer Sommerzeit
- Die "Messages" werden bis zur Lohnbuchhaltung kommuniziert. Bitte wählen Sie vernünftige Texte in den 3 Sprachen (Deutsch, Französisch, Italiensich).
- Es können mehrere "PlannedMaintenance" XML ins Verzeichnis "maintenance.filesystem.parent.xml" gelegt werden. Der Swissdec-Adapter wird immer das nächste dem Distributor melden und bei Ablauf automatisch löschen bzw. das nächste melden.

Beispiel eines "PlannedMaintenance" XML:

```
<PlannedMaintenance xmlns="http://www.itserve.ch/step/core"
                xmlns:ct="http://www.itserve.ch/step/coreTypes">
   <ct:Start>2050-01-01T00:00:00.0+01:00</ct:Start>
   <ct:End>2050-01-01T01:00:00.0+01:00</ct:End>
   <ct:Messages>
      <ct:Language>de</ct:Language>
      <ct:Value>Test Wartungsfenster sdA</ct:Value>
   </ct:Messages>
   <ct:Messages>
      <ct:Language>fr</ct:Language>
      <ct:Value>Fenêtre de maintenance sdA</ct:Value>
   </ct:Messages>
   <ct:Messages>
      <ct:Language>it</ct:Language>
      <ct:Value>Finestra di manutenzione sdA</ct:Value>
   </ct:Messages>
</PlannedMaintenance>
```

5.2. Logging-Konfiguration

Die Konfiguration für das Logging befindet sich pro Applikation in der Datei application.properties.

Option	Default	Beschreibung
logging.level.ch.itserve.step	INFO	Log Level aller Meldungen aus dem Package ch.itserve.step
logging.file	log/swissdecAdapter-receiver.log	In dieses File werden die Log's geschrieben.
logging.file.max-history	60	Gibt an wieviele Tage die Log-Files gesichert werden. Es wird empfohlen, die LOG's mindestens zwei Monate zu sichern.

Tabelle 5.1. Log-Einstellungen

Mehr Informationen entnehmen Sie bitte folgender Webseite: https://docs.spring.io/spring-boot/docs/2.1.13.RELEASE/reference/html/boot-features-logging.html

5.3. Monitoring

Der Swissdec-Adapter kann wie jeder andere Standardprozess oder Service überwacht werden. Falls der Swissdec-Adapter unter Windows als Service registriert wurde, kann die „Windows Computer Management Console“ benutzt werden, um den Status des Dienstes zu prüfen/ändern.

5.3.1. Eingebautes Monitoring

Das eingebaute Monitoring mittels REST-Schnittstelle wird im Kaptiel Admin-Konsole erläutert.

5.4. Admin-Konsole

Zur Authentifizierung an der REST-Schnittstelle stehen in der Konfiguration folgende Optionen zur Verfügung:

```
monitoring.user = admin
monitoring.pass = admin
```

Die RESTful Schnittstelle ist unter folgender URL erreichbar:

```
http://<INTEGRATION_HOST>:<INTEGRATION_PORT>/api/
```

Bitte stellen Sie beim Abrufen sicher, dass der HTTP-Request Header "Accept: application/json" gesetzt ist. Die Authentifizierung ist mittels HTTP-BASIC gewährleistet. Beispiel Request:

```
GET http://localhost:8280/api/
Accept: application/json
Username: admin
```

Mehr Informationen zur Authentifizierung finden Sie im Wikipedia [https://en.wikipedia.org/wiki/Basic_access_authentication]

5.4.1. Resourcen

5.4.1.1. Statistiken

Übersicht:

```
http://<INTEGRATION_HOST>:<INTEGRATION_PORT>/api/statistics
```

5.4.1.2. Monitoring

Erreichbar unter der URL:

```
http://<INTEGRATION_HOST>:<INTEGRATION_PORT>/api/monitoring
```

Beispielausgabe des eingebauten Monitorings:

```
{

  "connectivity": {
    "lastpingDateTime": "20.12.2016 09:37:44",
    "lastpingTimestamp": "1482223064619"
  },
  "database": {
    "path": "derbydb",
    "version": "3.0_5 (Build 24685)"
  },
  "system": {
    "java": "Oracle Corporation 1.8.0_131 (/usr/lib/jvm/jdk1.8.0_131/jre)",
    "arch": "Linux 4.4.0-79-generic (amd64)",
    "appVersion": "2.1_1 (Build 22586)",
  },
  "settings": {
     ...
  },
  "systemproperties": {
     ...
  }

}
```

5.5. Installation testen

Um die Installation prüfen zu können, stellt die Swissdec ein TestTool zur Verfügung. Es handelt sich dabei um einen einfachen Webservice- Klienten, mit dem Quellensteuerabrechnungen und Lohnausweise im Swissdec ELM v5.0 Format versendet werden können.

5.5.1. Installation

Das TestTool wird als ZIP-Datei geliefert und muss nur entpackt werden. Zum Ausführen wird eine Java-Installation benötigt.

5.5.2. Konfiguration

Bevor Sie testen können, müssen die Dateien

- config/tas20130514.xml
- config/tas20130514.xml
- config/tas20200220.xml
- config/tax20200220.xml
- config/tac20200220.xml

angepasst werden. Es handelt sich dabei um Properties-Files im XML-Format.

Option	**Beschreibung**
endpointURL	Die zu testende Adresse des Swissdec-Adapters.
wssEncryptionCert	Das Zertifikat wird zum Verschlüsseln der Meldungen benutzt (Public Key des Swissdec-Adapters).
sigrootcert	Das Zertifikat wird zum Verifizieren der Signatur der Antwort benutzt (Public Key des Swissdec-Adapters).

Option	Beschreibung
sslcert	Zertifikat mit dem Public Key zum SSL-Private Key
sslkey	SSL-Private Key für die Klientenauthentifizierung
sslpartnercacert	CA- Zertifikat zum Prüfen des SSL/TLS Serverzertifikats.

Tabelle 5.2. TestTool Konfiguration

Mehr Informationen zur Security entnehmen Sie bitte dem Kapitel 4, *Security*.

Anmerkung

Die mitgelieferte Konfiguration passt zur Default-Konfiguration des Swissdec-Adapters.

Die Option "sslpartnercacert" wird nur bei aktivem HTTPS benötigt.

Die Optionen "sslcert" und "sslkey" werden nur bei aktivem Mutual Authentication benötigt.

5.5.3. Ausführen der Tests

Um einen Test auszuführen, wird das dazugehörige Batch- beziehungsweise Shellskript ausgeführt. Eine ausführliche Dokumentation zu den Testfällen wird mit dem TestTool mitgeliefert.

6. Häufige Problem und deren Lösungen

Meldung

```
Caused by: org.apache.ws.security.components.crypto.CredentialException: Failed to load credentials.
   Inner Exception: [Keystore was tampered with, or password was incorrect]
```

Falsches *Keystore* Passwort in Swissdec-Adapter Receiver application.properties (Property 'wss4j.org.apache.ws.security.crypto.merlin.keystore.password').

Meldung

```
Caused by: java.io.FileNotFoundException: abTst_server.jks (No such file or directory)
```

Der *Keystore* referenziert in Swissdec-Adapter Receiver application.properties existiert nicht (Property 'wss4j.org.apache.ws.security.crypto.merlin.file')

Meldung

```
Caused by: java.security.UnrecoverableKeyException: Cannot recover key
```

Falsches Private Key Passwort in Swissdec-Adapter Receiver application.properties (Property 'wss4j.privateKeyPassword')

Meldung

```
Caused by: javax.xml.ws.soap.SOAPFaultException: Marshalling Error: Connection refused
 at org.apache.cxf.jaxws.JaxWsClientProxy.invoke(JaxWsClientProxy.java:156)
 at $Proxy205.saveSalaryDeclaration(Unknown Source)
```

Die Receiver Applikation hat keine Verbindung zur Integration. Bitte kontrollieren Sie folgende Konfiguration im Receiver application.properties:

```
integration.service.protocol=http://
integration.service.host=localhost
integration.service.port=9090
```

Stellen Sie sicher, dass die Konfiguration an Ihre Infrastruktur angepasst wurde und dass eine HTTP Verbindug von der Receiver Applikation auf die konfigurierte URL geöffnet werden kann.

A. Anhang

A.1. Referenzierte Dokumente

[ENDRECREQ] *Richtlinien für Lohndatenübermittlung*. Endreceiver Requirements. Swissdec. Version 5 (20200220).

[DETAILSPEZ] *Swissdec-Adapter*. Detailspezifikation. itServe AG. V02.01.

A.2. Glossar

DMZ	ein Computernetz mit sicherheitstechnisch kontrollierten Zugriffsmöglichkeiten auf die daran angeschlossenen Server
Keystore	Ein Keystore ist eine Datenbank von Schlüsseln und Zertifikaten in einer Datei.
NTP	Standard zur Synchronisierung von Uhren in Computersystemen.
sM-Client	Der sM-Client bietet Funktionalität zum Versenden und Empfangen von Meldungen in diversen Formaten
STEP	Produkt der Firma itServe AG, *Swissdec* Empfänger
Swissdec	Qualitätslabel für Lohnbuchhaltungssysteme. XML-Standards für die Einheitliche Lohn-Meldung (ELM).

A.3. Konfigurationsvergleich 2.x zu 3.x

Um die Migration zu vereinfachen werden die Differenzen der Konfiguration der Versionen 2.x und 3.x hier aufgelistet.

2.x	**3.x**	**Differenz**
	server.port	Konfiguration in 2.x in conf/ server.xml
	server.ssl.key-store-type	Konfiguration in 2.x in conf/ server.xml
	server.ssl.key-store	Konfiguration in 2.x in conf/ server.xml
	server.ssl.key-store-password	Konfiguration in 2.x in conf/ server.xml
	server.ssl.key-alias	Konfiguration in 2.x in conf/ server.xml
	server.ssl.key-password	Konfiguration in 2.x in conf/ server.xml
userAgent.institutionName	institution.name	Konfiguration ersetzt
institution.tax.id	institution.canton	Konfiguration zusammengefasst
institution.tas.id	institution.canton	Konfiguration zusammengefasst
	elm.tac.enabled	Neues Feature
	elm.tas.enabled	Neues Feature
	elm.tax.enabled	Neues Feature
	elm4.tas.enabled	Neues Feature
	elm4.tax.enabled	Neues Feature
tax.accept.ex	tax.accept.ex	Keine Änderung
salarydeclaration.service.protocol	integration.service.protocol	Konfiguration ersetzt
salarydeclaration.service.host	integration.service.host	Konfiguration ersetzt

2.x	3.x	Differenz
salarydeclaration.service.port	integration.service.port	Konfiguration ersetzt
salarydeclaration.service.context		Nicht mehr benötigt
salarydeclaration.service.name		Nicht mehr benötigt
	wss4j.org.apache.ws.security. crypto.provider	Neue Konfiguration
org.apache.ws.security. crypto.merlin.file	wss4j.org.apache.ws.security. crypto.merlin.file	Konfiguration ersetzt
org.apache.ws.security. crypto.merlin.keystore.type	wss4j.org.apache.ws.security. crypto.merlin.keystore.type	Konfiguration ersetzt
org.apache.ws.security. crypto.merlin.keystore.password	wss4j.org.apache.ws.security. crypto.merlin.keystore.password	Konfiguration ersetzt
org.apache.ws.security. crypto.merlin.alias.password		Nicht mehr benötigt
org.apache.ws.security. crypto.merlin.keystore.alias	wss4j.org.apache.ws.security. crypto.merlin.keystore.alias	Konfiguration ersetzt
wss4j.privateKeyPassword	wss4j.privateKeyPassword	Keine Änderung
wss4j.signatureUser=1		Nicht mehr benötigt
monitoring.enabled		Nicht mehr benötigt
monitoring.user		Nicht mehr benötigt
monitoring.pass		Nicht mehr benötigt
	logging.level.ch.itserve.step	Neues Feature
	logging.file	Neues Feature
	logging.file.max-history	Neues Feature

Tabelle A.1. Swissdec Adapter Receiver - Version 2.x vs. 3.x

2.x	3.x	Differenz
	server.port	Konfiguration in 2.x in conf/ server.xml
institution.tax.id	institution.canton	Konfiguration zusammengefasst
institution.tas.id	institution.canton	Konfiguration zusammengefasst
released.filesystem.parent.xml	released.filesystem.parent.xml	keine Änderung
failed.filesystem.parent.xml	failed.filesystem.parent.xml	keine Änderung
result.filesystem.parent.xml	result.filesystem.parent.xml	keine Änderung
sent.filesystem.parent.xml	sent.filesystem.parent.xml	keine Änderung
undeliverable.filesystem.parent.xml	undeliverable.filesystem.parent.xml	keine Änderung
maintenance.filesystem.parent.xml	maintenance.filesystem.parent.xml	keine Änderung
await.result	await.result	keine Änderung
commune.splitting	commune.splitting	keine Änderung
la.splitting	la.splitting	keine Änderung
testcase.auto.quittance	testcase.auto.quittance	keine Änderung
write.original.xml	write.original.xml	keine Änderung
	map.tas.toV5	Neues Feature ab 3.x
	map.tas.fromPeriod	Neues Feature ab 3.x
	derby.system.home	Konfiguration in 2.x in System Properties
processing.default.hours	processing.default.hours	keine Änderung

2.x	3.x	Differenz
housekeeping.days	housekeeping.days	keine Änderung
monitoring.enabled	monitoring.enabled	keine Änderung
monitoring.user	monitoring.user	keine Änderung
monitoring.pass	monitoring.pass	keine Änderung
	logging.level.ch.itserve.step	Neues Feature
	logging.file	Neues Feature
	logging.file.max-history	Neues Feature

Tabelle A.2. Swissdec Adapter Integration - Version 2.x vs. 3.x

A.4. Unicode Tabelle für gängige Sonderzeichen

Sonderzeichen im insitution.name müssen ersetzt werden

Beispiel für "Steuerverwaltung Kt. Graubünden" wird gesetzt mit:

```
institution.name=Steuerverwaltung Kt. Graub\u00FCnden
```

Sonderzeichen	Unicode
Ä	\u00C4
ä	\u00E4
Ö	\u00D6
ö	\u00F6
Ü	\u00DC
ü	\u00FC
é	\u00E9
è	\u00E8
ç	\u00E7

Tabelle A.3. Unicode Tabelle für Sonderzeichen

B. Beispiele

B.1. Konfiguration

```
                #
# Copyright 1996-2021 itServe AG. All rights reserved.
#
# This software is the proprietary information of itServe AG
# Bern Switzerland. Use is subject to license terms.
#

# System Settings
server.port=8080

# SwissdecAdapter Receiver Settings
institution.name=KSTV Bern
institution.canton=BE

elm.tac.enabled=true
elm.tas.enabled=true
elm.tax.enabled=true

elm4.tas.enabled=true
elm4.tax.enabled=true

tax.accept.ex=false

# SSL-Security with JKS
#server.ssl.key-store=classpath:security/TestConsumerAll.jks
#server.ssl.key-store-type=JKS
#server.ssl.key-store-password=default
#server.ssl.key-alias=1
#server.ssl.key-password=default

# SSL-Security with PKCS12
#server.ssl.key-store=classpath:keystore/baeldung.p12
#server.ssl.key-store-type=PKCS12
#server.ssl.key-store-password=password
#server.ssl.key-alias=1

# WS-Security
wss4j.org.apache.ws.security.crypto.provider=org.apache.wss4j.common.crypto.Merlin
wss4j.org.apache.ws.security.crypto.merlin.file=security/TestConsumerAll.jks
wss4j.org.apache.ws.security.crypto.merlin.keystore.type=jks
wss4j.org.apache.ws.security.crypto.merlin.keystore.password=default
wss4j.org.apache.ws.security.crypto.merlin.keystore.alias=1
wss4j.privateKeyPassword=default

# Remote Services
integration.service.protocol=http://
integration.service.host=localhost
integration.service.port=9090

# LOG Settings
logging.level.ch.itserve.step=INFO
logging.file=log/swissdecAdapter-receiver.log
logging.file.max-history=60
```

Beispiel B.1. Standardkonfiguration Receiver

```
                #
# Copyright 1996-2021 itServe AG. All rights reserved.
#
# This software is the proprietary information of itServe AG
# Bern Switzerland. Use is subject to license terms.
#

# System Settings
server.port=9090

# SwissdecAdapter Integration Settings
institution.canton=BE

released.filesystem.parent.xml=C:/swissdecAdapter/data/received
failed.filesystem.parent.xml=C:/swissdecAdapter/data/failed
result.filesystem.parent.xml=C:/swissdecAdapter/data/result
```

```
sent.filesystem.parent.xml=C:/swissdecAdapter/data/sent
undeliverable.filesystem.parent.xml=C:/swissdecAdapter/data/undeliverable
maintenance.filesystem.parent.xml=C:/swissdecAdapter/data/maintenance

await.result=false
commune.splitting=false
la.splitting=false
testcase.auto.quittance=false
write.original.xml=false
map.tas.toV5=false
map.tas.fromPeriod=2021-01

derby.system.home=C:/swissdecAdapter/derbydb
spring.datasource.url=jdbc:derby:C:/swissdecAdapter/derbydb/step

processing.default.hours=48
housekeeping.days=180

monitoring.enabled=false
monitoring.user=admin
monitoring.pass=admin

# LOG Settings
logging.level.ch.itserve.step=INFO
logging.file=log/swissdecAdapter-integration.log
logging.file.max-history=60
```

Beispiel B.2. Standardkonfiguration Integration

B.2. Installationsanleitung für Apache Reverse Proxy

Diese Installationsaleitung dient als Anlehnung und kann nach Systemvorgaben angepasst werden

In diesem Beispiel wird der Endpunkt von swissdec.be.ch konfiguriert mit dem Zertifikat C:/Zertifikate/server.crt, dem Key C:/Zertifikate/server.key. Der SwissdecAdapter Receiver läuft auf dem selben System auf Port 8080.

Das Mutual Zertifikat liegt im C:/Zertifikate/mutual.pem

1. Download Apache von ApacheHaus (apachehaus.com) (Apache 2.4.x OpenSSL 1.1.1 VC15)
2. Extrahieren in Ordner (Beispiel: C:/Apache24)
3. Apache als Service installieren und starten:

```
C:/Apache24/bin/httpd -k install
C:/Apache24/bin/httpd -k start
```

4. Konfiguration von C:/Apache24/conf/httpd.conf

   mod_proxy and mod_proxy_http laden (Kommentar "#" entfernen)

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```

5. Konfiguration von SSL in C:/Apache24/conf/extra/httpd-ahssl.conf

```
<VirtualHost swissdec.be.ch:443>
   SSLEngine on
   ServerName swissdec.be.ch:443
   SSLCertificateFile "C:/Zertifikate/server.crt"
   SSLCertificateKeyFile "C:/Zertifikate/server.key"
   # Die nächste zwei Zeilen werden für die Mutual Authentication benutzt
   SSLCACertificateFile "C:/Zertifikate/mutual.pem"
   SSLVerifyClient require
   DocumentRoot "${SRVROOT}/htdocs"
   # DocumentRoot access handled globally in httpd.conf
   CustomLog "${SRVROOT}/logs/ssl_request.log" "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
      <Directory "${SRVROOT}/htdocs">
         Options Indexes Includes FollowSymLinks
         AllowOverride AuthConfig Limit FileInfo
         Require all granted
      </Directory>
   ProxyPass         /    http://localhost:8080/
```

```
    ProxyPassReverse    /    http://localhost:8080/
</VirtualHost>
```

6. Apache Service neu starten

Sind die Zertifikate nur im JKS Format, kann man diese mit dem Keystore Explorer (https://keystore-explorer.org/) relativ einfach in die gewünschten Formate exportieren.