SwissdecAdapter

Betriebshandbuch

Versionsgeschichte			
Version 1.0.0	2013-08-07	Marco Stettler	Release 1.0
Version 2.0.0	2014-01-30	Christoph Brunner	Release 2.0
Version 2.1.0	2016-06-29	Marco Stettler	Release 2.1_0
Version 3.0.0	2020-09-15	Christoph Brunner	Release 3.0_0
Version 3.0.4	2021-03-11	Christoph Brunner	Release 3.0_4
Version 3.1.0	2023-08-22	Christoph Brunner	Release 3.1_0

Inhaltsverzeichnis

1. Zum aktuellen Release 1
1.1. Zusammenfassung Release 1
1.2. Migrationspfad 2
2. Einführung SwissdecAdapter 3
2.1. Übersicht Swissdec 3
2.2. Swissdec-Adapter 3
2.3. Schnittstellen 4
2.4. Referenzen 5
3. Betriebsanforderungen 6
3.1. Systemanforderungen 6
4. Installation 8
4.1. Delivery Paket 8
4.2. Installation unter Linux 8
4.3. Installation unter Windows 11
5. Konfiguration 13
5.1. Receiver 13
5.2. Integration 16
6. Security 20
6.1. Transport Layer TLS 21
6.2. Webservice Security 22
7. Betrieb 24
7.1. Logging-Konfiguration 24
7.2. Wartungsfenster 24
7.3. Monitoring 25
7.4. Admin-Konsole 25
A. Anhang 27
A.1. Apache Reverse Proxy 27
A.2. Testtool 28
A.3. Unicode Tabelle 29
A.4. Referenzierte Dokumente 29
A.5. Glossar 29

1. Zum aktuellen Release

Im diesem Kapitel werden die Neuerungen und der Migrationspfad des aktuellen Release beschrieben.

1.1. Zusammenfassung Release

Zusammengefasst die wichtigsten Änderungen im SwissdecAdapter 3.1.0

- Update auf ELM 5.1 mit der Implementierung der neuen Swissdec Domäne "Unemployment Certificate" (Arbeitslosenbescheinigung).
- Optimierung des HousekeepingJob, so dass OutOfMemory Exception nicht mehr vorkommt
- Meldungen von Mitarbeiterbeteiligungen werden getrennt von den Lohnausweisen abgelegt
- Technische Duplikate werden mit der DeclarationID der originalen Meldung im Syslog ausgewiesen, sodass der Kundensupport verbessert werden kann
- Validierung der Antworten, wenn await.result=true gesetzt ist. Die Validierungsfehler sind im Log ersichtlich. Das Unternehmen erhält ein "Processing" zurück (Verhalten, wie noch kein Resultat verfügbar).
- Upgrade auf JDK 17 (Aktuelle LTS Version von JDK)
- Upgrade der verwendeten Bibliotheken und Frameworks.

1.2. Migrationspfad

Mit folgenden Schritten kann die Migration in wenigen Minuten erledigt werden.

Warnung

Das Wartungsfenster muss vorgängig geplant sein, da der Swissdec Distributor nur alle 30 Minuten prüft, ob ein Wartungsfenster gesetzt ist.

Wenn nicht anders beschrieben müssen die folgenden Schritte auf beiden SwissdecAdapter Applikationen durchgeführt werden.

1. **Services stoppen**
2. **Konfiguration sichern**

   Es wird dringend empfohlen vor dem Update die folgenden Daten in einen separaten Ordner zu sichern.

   a. Konfiguration

      Die Files application.properties in beiden Applikationen

   b. Integration: Datenbank sichern (Ordner "derbydb")

   c. Receiver: Zertifikate sichern (Im Ordner "security")
3. **Java 17 auf System installieren und Systemkonfiguration prüfen**

   Linux: sudo apt install openjdk-17-jdk

   Windows: Installation JDK 17 (Zum Beispiel: Adoptium [https://adoptium.net/])

   Im Terminal des Systems, muss der Befehl "java -version" die installierte Java Version ausgeben. Ansonsten muss die Servicekonfiguration so angepasst werden, dass der Service mit Java 17 gestartet wird.
4. **Konfiguration anpassen**

   Receiver: application.properties

   - Der Key "logging.file" umbenennen nach "logging.file.name"

   Integration: application.properties

   - Der Key "logging.file" umbenennen nach "logging.file.name"
5. ### Warnung

   Bitte den Wert "logging.file.max-history" prüfen. Ist der Wert auf 1, wird das LOG nur für einen Tag gesichert. Es wird ein Wert von 60 empfohlen.
6. **Services starten**

2. Einführung SwissdecAdapter

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), TaxCrossborder (Grenzgängermeldungen) und UnemploymentCertificate (Arbeitslosenbescheinigung) auszutauschen. Er wurde zur Integration mit dem *sM-Client* entwickelt, kann jedoch auch standalone betrieben werden.

2.1. 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 Knowhow 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:

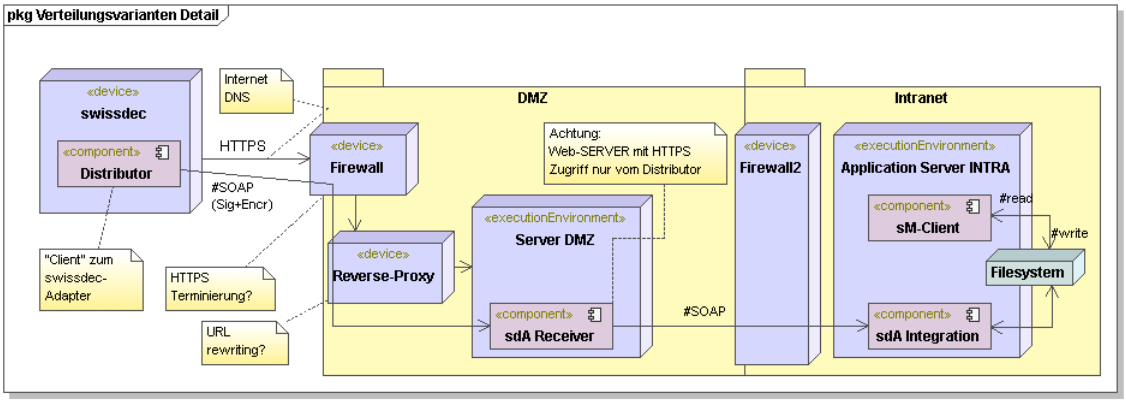


Abbildung 2.1. Installationsskizze Swissdec-Adapter

2.2. 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%

2.3. Schnittstellen

2.3.1. 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.

2.3.2. 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.

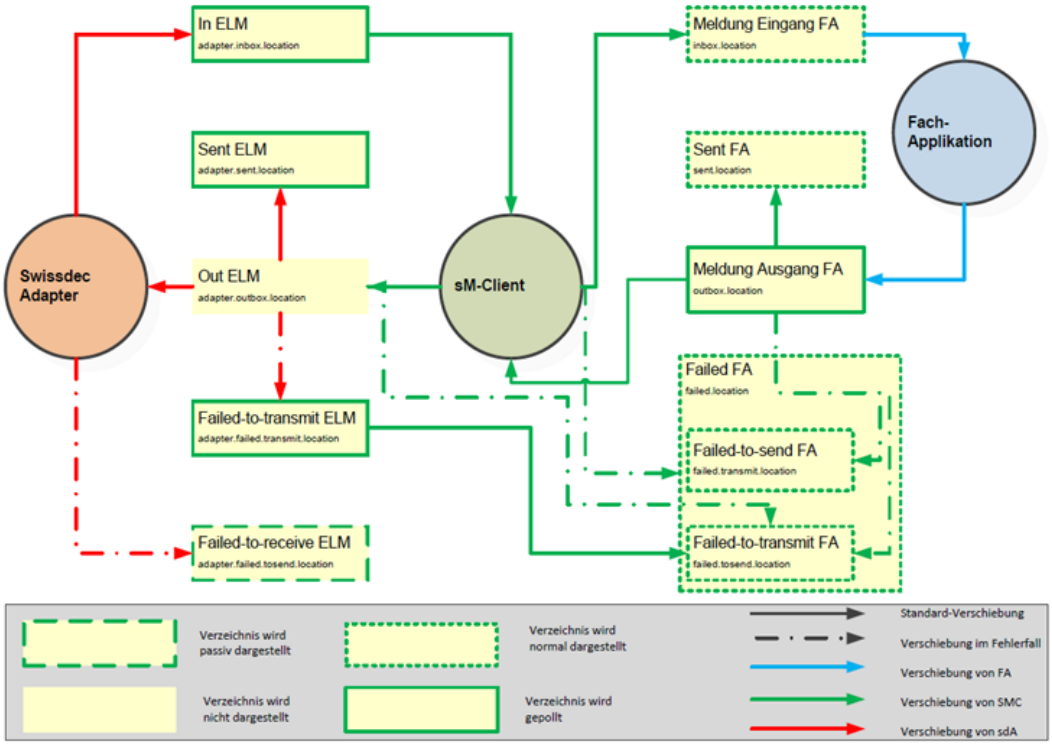


Abbildung 2.2. Dateisystemschnittstelle Swissdec-Adapter - sM-Client

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 2.1. Pfade der Schnittstelle Swissdec-Adapter - sM-Client

Die komplette Schnittstellenspezifikation entnehmen Sie bitte der [DETAILSPEZ]

2.4. Referenzen

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

3. 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.

3.1. Systemanforderungen

3.1.1. Unterstützte Platformen

Der Swissdec-Adapter ist eine reine Java Applikation. Es wird eine installiertes Java JDK in der Version 17 vorausgesetzt. Empfohlen wird OpenJDK. Getestete und unterstützte Plattformen:

- Ubuntu 22.04 LTS
- Windows Server 64bit
- Windows 10 / Windows 11

3.1.2. 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.

3.1.3. Erreichbarkeit

Der SwissdecAdapter muss vom Distributor gemäss den Swissdec Richtlinien erreichbar sein. Details dazu gibt es im Kapitel Konnektivität ???beim Receiver.

3.1.4. 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.

3.1.5. 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 stream-basiert. Der Memory Verbrauch ist so deutlich effizienter.

3.1.6. 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.

3.1.7. 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.

4. Installation

4.1. 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 (integration.properties), die Datenbank (derbydb) und die Windows Service-Installations Wrapper Dateien.
receiver	Enthält die SwissdecAdapter-Receiver Applikation (swissdecAdapter-receiver.jar), die Standard-Konfiguration (receiver.properties) und die Windows Service-Installations Wrapper Dateien.
swissdecAdapter-testtool.zip	Testtool mit Testapplikation, Testdaten und Testdokumentation

Tabelle 4.1. swissdecAdapter-delivery-3.1_x.zip

4.2. Installation unter Linux

Die Installation wurde unter Ubuntu 22.04 (LTS) getestet. Sie kann aber unter jedem Linux-Derivat installiert werden.

Warnung

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.

4.2.1. Systemvorbereitung

Der SwissdecAdapter benötigt ein Java Development Kit (JDK) in der Version 17.

Die Installation von Java 17 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 Benutzer zu starten.

4.2.2. Installation der Applikationen

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

```
sudo useradd -r springboot
```

Die Ordner integration und receiver aus dem Delivery-Paket (Abschnitt 4.1, „Delivery Paket“) werden im Ordner /opt entpackt.

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

```
unzip swissdecAdapter-delivery-3.1_0.zip
sudo mkdir /opt/swissdecAdapter
sudo cp -r swissdecAdapter-delivery-3.1_0/integration /opt/swissdecAdapter/.
sudo cp -r swissdecAdapter-delivery-3.1_0/receiver /opt/swissdecAdapter/.
sudo rm /opt/swissdecAdapter/*/*wrapper*
```

Bevor die Applikationen korrekt gestartet werden können, müssen 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
```

4.2.3. Service Installation

4.2.3.1. /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
```

4.2.3.2. systemd Installation

Bei neueren Linux Installationen wird häufig ein Service 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
```

4.3. Installation unter Windows

Die Installation wurde unter Windows 10 und 11 getestet, kann aber unter jedem modernen Windows System installiert werden.

> **Warnung**
>
> 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.

4.3.1. Systemvorbereitung

Der SwissdecAdapter benötigt ein Java Development Kit (JDK) in der Version 17.

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.

4.3.2. Installation der Applikation

Die Ordner integration und receiver aus dem Delivery-Paket (Abschnitt 4.1, „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
```

4.3.3. 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.

Windows Service Wrapper benötigt .NET Framework

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	Auszuführendes 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.
log mode	Der Default Wert wird auf "none" geschaltet, es handelt sich um das Log des Service Wrapper. Das Log des SwissdecAdapter wird von Spring Boot gesteuert (application.properties)

Tabelle 4.2. Beschreibung WinSW-Konfiguration

5. Konfiguration

5.1. Receiver

5.1.1. Konfigurationsoptionen Receiver

In diesem Abschnitt werden die Properties mit ihren Default Values aufgelistet und kurz beschrieben, welchen Einfluss auf das Verhalten diese Einstellungen haben.

5.1.1.1. SwissdecAdapter Verhalten

Diese Properties haben Einfluss auf das Applikationsverhalten des SwissdecAdapter

- **institution.name=KSTV Bern**

  Ausgeschriebener Name der Steuerverwaltung (siehe auch:Abschnitt A.3, „Unicode Tabelle“)

- **institution.canton=BE**

  Kürzel des Kanton

- **elm.tac.enabled=true**

  Der SwissdecAdapter kann TaxCrossborder (Grenzgänger) Meldungen empfangen

- **elm.tas.enabled=true**

  Der SwissdecAdapter kann TaxAtSource (Quellensteuer) Meldungen empfangen

- **elm.tax.enabled=true**

  Der SwissdecAdapter kann Tax (Lohnausweise, Mitarbeiterbeteiligungen und Rentenbescheinigungen) Meldungen empfangen

- **elm.tuc.enabled=true**

  Der SwissdecAdapter kann Unemployment Certificate (Arbeitslosenbescheinigungen) Meldungen empfangen

- **elm4.tas.enabled=true**

  Der SwissdecAdapter kann TaxAtSource (Quellensteuer) Meldungen empfangen (ELM Version 4.0)

- **elm4.tax.enabled=true**

  Der SwissdecAdapter kann Tax (Lohnausweise, Mitarbeiterbeteiligungen und Rentenbescheinigungen) Meldungen empfangen (ELM Version 4.0)

- **tax.accept.ex=false**

  Der SwissdecAdapter kann Lohnausweise von Personen mit Wohnsitz im Ausland empfangen.

5.1.1.2. Systemkonfiguration

Diese Properties dienen der generellen Systemkonfiguration

- **server.port=8080**

  Auf diesem Port startet die Applikation SwissdecAdapter-Receiver

- **integration.service.protocol=http://**

  Über welches Protokoll ist die SwissdecAdapter-Integration erreichbar

- **integration.service.host=localhost**

Auf diesem Host ist die SwissdecAdapter-Integration Integration

- **integration.service.port=9090**

  Auf diesem Port läuft die SwissdecAdapter-Integration Integration

- **logging.level.ch.itserve.step=INFO**

  Einstellung des LOG Level (Möglichkeiten: DEBUG, INFO, WARN, ERROR)

- **logging.file=log/swissdecAdapter-receiver.log**

  Standardmässiges LOG File (wird täglich rotiert)

- **logging.file.max.history=60**

  Nach wie vielen Tagen werden die alten Log-Files gelöscht

5.1.1.3. WS Security Konfiguration (siehe auch:Kapitel 6, *Security*)

Mit diesen properties wird die Webservice Security Konfiguration gesteuert

- **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 Keystore, in welchem das WS Zertifikat abgelegt ist

- **wss4j.org.apache.ws.security.crypto.merlin.keystore.type=jks**

  Art des Keystores

- **wss4j.org.apache.ws.security.crypto.merlin.keystore.password=default**

  Passwort zum öffnen des Keystore

- **wss4j.org.apache.ws.security.crypto.merlin.keystore.alias=1**

  Alias unter welchem das Zertifikat im Keystore gesichert ist

- **wss4j.privateKeyPassword=default**

  Passwort für den Private Key

5.1.1.4. TLS Security Konfiguration (siehe auch:Kapitel 6, *Security*)

Mit diesen properties wird die TLS Security Konfiguration konfiguriert. Diese Einstellungen sind optional, da die TLS Terminierung häufig auf einem vorgelagerten System (z.B. Proxy, Gateway, WAF oder ähnliches) passiert.

Die Erfahung hat gezeigt, dass die TLS Terminierung direkt im SwissdecAdapter Receiver nicht immer funktioniert. Erfolgt die TLS Terminierung auf dem System, auf welchem der SwissdecAdapter Receiver installiert ist kann das alternativ auch mit einem Apache Server gemacht werden.

- **server.ssl.key-store=security/TestConsumerAll.jks**

  Pfad zum TLS Zertifikate, welche in einem *JKS* zur Verfügung gestellt ist

- **server.ssl.key-store-type=JKS**

  Art des Keystore

- **server.ssl.key-store-password=default**

  Passwort zum Keystore

- **server.ssl.key-alias=1**

  Alias Name des Zertifikats innerhalb des Keystore

- **server.ssl.key-password=default**

  Passwort zum Private Key

5.1.2. Referenzkonfiguration

```
                #
# Copyright 1996-2024 itServe AG. All rights reserved.
#
# This software is the proprietary information of itServe AG
# Bern Switzerland. Use is subject to license terms.
#


#
# Copyright 1996-2023 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
elm.tuc.enabled=true

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

tax.accept.ex=false

# TLS-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

# TLS-Security with PKCS12
#server.ssl.key-store=classpath:keystore/TestConsumerAll.p12
#server.ssl.key-store-type=PKCS12
#server.ssl.key-store-password=default
#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.name=log/swissdecAdapter-receiver.log
logging.file.max-history=60
```

Beispiel 5.1. receiver.properties

5.2. Integration

5.2.1. Konfigurationsoptionen Integration

5.2.1.1. SwissdecAdapter Verhalten

Diese Properties haben Einfluss auf das Applikationsverhalten des SwissdecAdapter

- **institution.canton=BE**

  Kürzel des Kanton

- **await.result=false**

  Quellensteuer: Wenn dieser Wert auf true gesetzt ist, erwartet der SwissdecAdapter ein Response File, welches dem Unternehmen bei einem GetResult Call zurückgegeben wird.

- **commune.splitting=false**

  Quellensteuer: Die Quellensteuermeldung wird aufgesplittet und pro Municipality-ID (Gemeinde) ins Zip File geschrieben.

- **testcase.auto.quittance=false**

  Wenn await.result=true. Ist await.true=true, werden Meldungen, welche als Testfall markiert sind, direkt vom SwissdecAdapter verarbeitet.

- **map.tas.toV5=false**

  Quellensteuer: Die ELMv4 Meldungen werden in ELMv5 Meldungen "gemappt"

- **map.tas.fromPeriod=2021-01**

  Wenn map.tas.toV5=true: Ab dieser Meldeperiode werden die Lohnmeldungen in ELMv5 Meldungen "gemappt"

- **processing.default.hours=48**

  Wenn await.result=true. Rechnet den Wert im Element ExpectedAvailability in der Processing Antwort Struktur.

- **la.splitting=false**

  Lohnausweis: Ist diese Option auf true, wird pro Mitarbeiter ein XML File ins Zip geschrieben.

- **or.enabled=false**

  Mitarbeiterbeteiligung: Ist diese Option auf true, werden die Meldungen im Filesystem gesichert.

- **or.splitting=false**

  Mitarbeiterbeteiligung: Ist diese Option auf true, wird pro Mitarbeiter ein XML File ins Zip geschrieben.

- **uc.splitting=false**

  Arbeitslosenbescheinigung: Ist diese Option auf true, wird pro Mitarbeiter ein XML File ins Zip geschrieben.

- **write.original.xml=false**

  Wird die Meldung gesplittet und diese Option ist auf true wird auch die originale Lohnmeldung ins Zip geschrieben.

5.2.1.2. Erweiterte Einstellungen

Weitere Einstellungen können in folgenden Properties geändert werden.

- **housekeeping.schedule=0 */15 * * * ?**

  Cron Wert wie häufig der Housekeeping Job gestartet wird. (wikipedia.org/wiki/Cron [https://de.wikipedia.org/wiki/Cron])

Der Default Wert entspricht alle 15 Minuten

- **housekeeping.elm.enabled=true**

  Aktivierung des Housekeeping Job

- **housekeeping.elm.days=365**

  Wie lange sollen die Lohnmeldungen in der Datenbank gesichert werden. (Mindestwert ist 90 Tage)

- **backup.derby.onstartup=true**

  Soll die Datenbank beim Start der Applikation gesichert werden

- **cleanup.derby.onstartup=true**

  Indexierung der Datenbank zur Optimierung des Speicherplatzes

- **monitoring.enabled=false**

  Aktivierung der Monitoring Console / REST Schnittstelle

- **monitoring.user=admin**

  Benutzername für Monitoring Console

- **monitoring.pass=admin**

  Passwort für Monitoring Console

5.2.1.3. Systemkonfiguration

Diese Properties dienen der generellen Systemkonfiguration

- **server.port=9090**

  Auf diesem Port startet die Applikation SwissdecAdapter-Integration

- **derby.system.home=C:/swissdecAdapter/derbydb**

  In diesem Directory befindet sich die Derby Datenbank

- **spring.datasource.url=jdbc:derby:C:/swissdecAdapter/derbydb/step**

  Link zur Derby Datenbank

- **released.filesystem.parent.xml=C:/swissdecAdapter/data/received**

  In diesem Ordner werden verarbeitete Lohnmeldungen gesichert

  (sieheTabelle 2.1, „Pfade der Schnittstelle Swissdec-Adapter - sM-Client“)

- **failed.filesystem.parent.xml=C:/swissdecAdapter/data/failed**

  In diesem Ordner werden fehlgeschlagene Lohnmeldungen gesichert

  (sieheTabelle 2.1, „Pfade der Schnittstelle Swissdec-Adapter - sM-Client“)

- **result.filesystem.parent.xml=C:/swissdecAdapter/data/result**

  In diesem Verzeichnis werden die Antwort Files erwartet (wenn await.result=true)

  (sieheTabelle 2.1, „Pfade der Schnittstelle Swissdec-Adapter - sM-Client“)

- **undeliverable.filesystem.parent.xml=C:/swissdecAdapter/data/undeliverable**

  Diese Antwortdaten konnten nicht abgeholt werden

  (sieheTabelle 2.1, „Pfade der Schnittstelle Swissdec-Adapter - sM-Client“)

- **maintenance.filesystem.parent.xml=C:/swissdecAdapter/data/maintenance**

  In diesen Ordner gehört das Wartungsfenster-XML

- **logging.level.ch.itserve.step=INFO**

  Einstellung des LOG Level (Möglichkeiten: DEBUG, INFO, WARN, ERROR)

- **logging.file=log/swissdecAdapter-integration.log**

  Standardmässiges LOG File (wird täglich rotiert)

- **logging.file.max.history=60**

  Nach wie vielen Tagen werden die alten Log-Files gelöscht

5.2.2. Referenzkonfiguration

```
                #
# Copyright 1996-2024 itServe AG. All rights reserved.
#
# This software is the proprietary information of itServe AG
# Bern Switzerland. Use is subject to license terms.
#


#
# Copyright 1996-2023 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

# TaxAtSource Settings
await.result=false
commune.splitting=false
testcase.auto.quittance=false
map.tas.toV5=false
map.tas.fromPeriod=2021-01
processing.default.hours=48

# Tax Settings
la.splitting=false

# Ownership Right Details
or.enabled=true
or.splitting=true

# Unemployment Certificate Settings
uc.splitting=true

# Global Settings
write.original.xml=false

# Directory Settings
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

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

# Housekeeping Settings
housekeeping.schedule=0 */15 * * * ?
housekeeping.elm.enabled=true
```

```
housekeeping.elm.days=365

# Startup Settings
backup.derby.onstartup=true
cleanup.derby.onstartup=true

# Monitoring Settings
monitoring.enabled=false
monitoring.user=admin
monitoring.pass=admin

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

Beispiel 5.2. integration.properties

6. Security

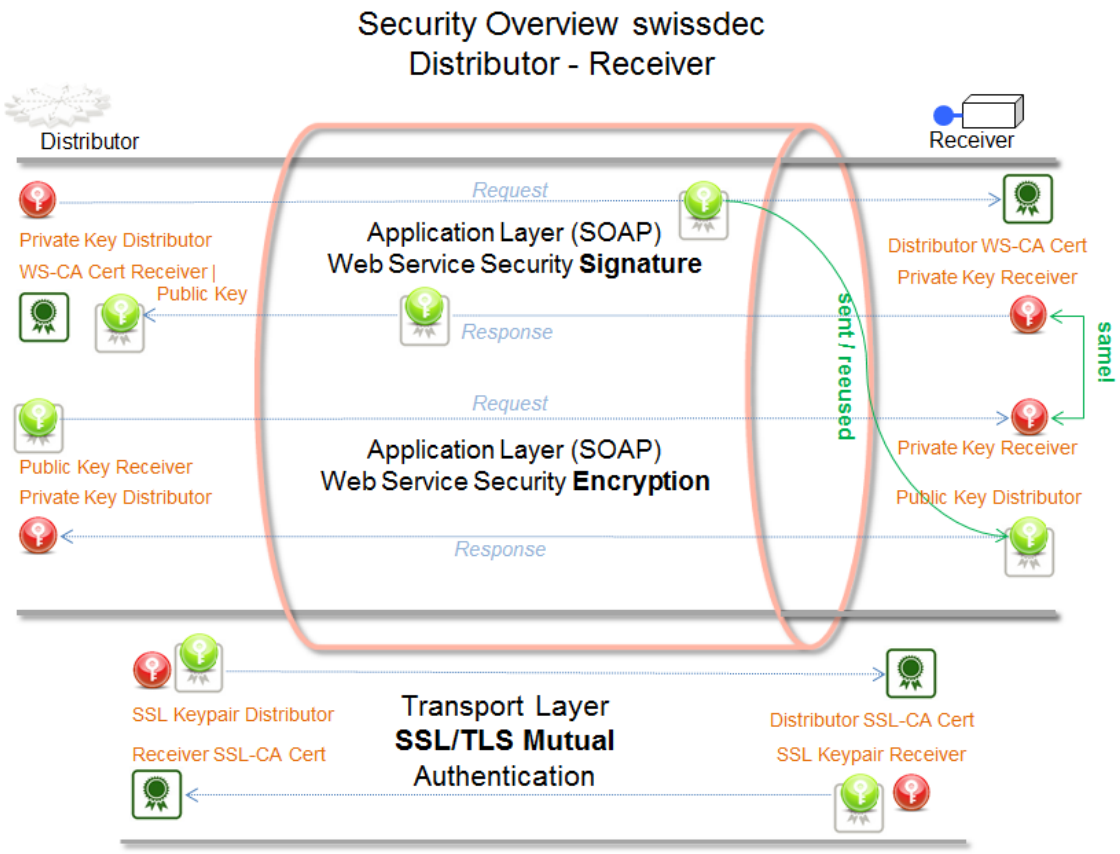


Abbildung 6.1. Security Overview

	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 6.1. Legende zu Abbildung Security Overview

6.1. Transport Layer 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.

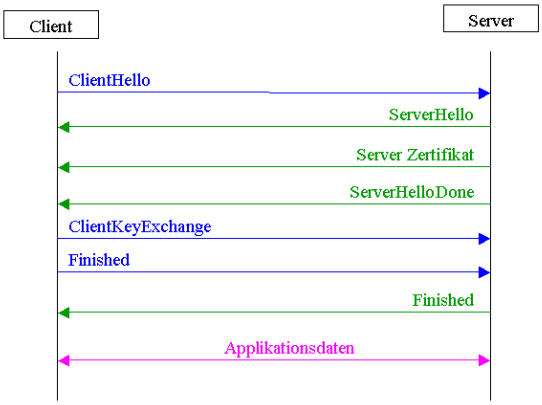


Abbildung 6.2. TLS Handshake

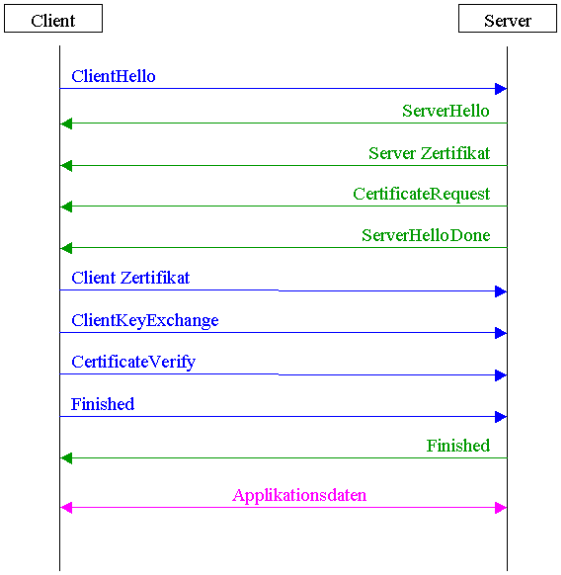


Abbildung 6.3. SSL Handshake mit Mutual Authentication

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.

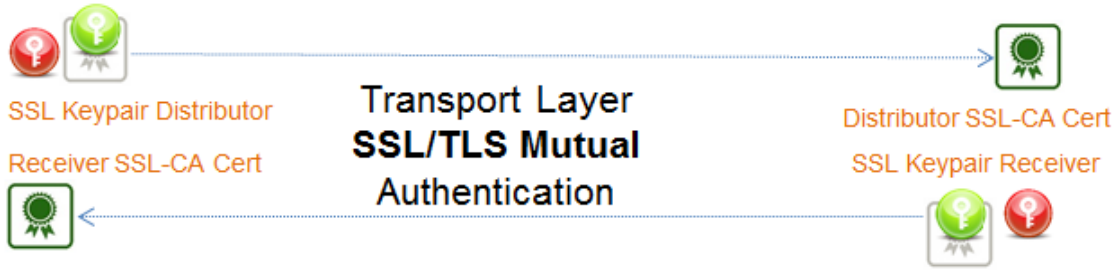


Abbildung 6.4. Transport Layer Security Overview

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 TLS Konfiguration (???) für den Swissdec-Adapter befindet sich in der Datei "receiver.properties". Falls die Terminierung auf dem Adapter gemacht werden soll, befindet sich in dieser Datei ein Beispiel einer Konfiguration.

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.

6.2. Webservice Security

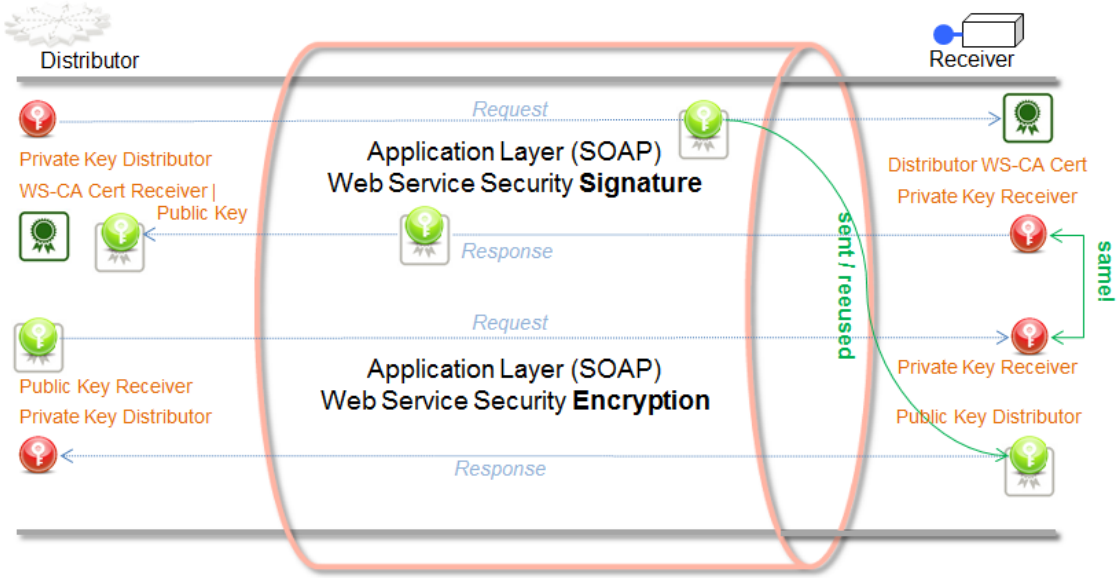


Abbildung 6.5. Webservice Security Overview

6.2.1. 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 6.2. 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.

6.2.2. 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 6.3. 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.

6.2.3. 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 (???) (application.properties) steuern das Verhalten des Swissdec-Adapters für die Webservice Security.

Anmerkung

Die produktiven Zertifikate können via sM-Client Support [https://smcsupport.atos-solutions.ch]bestellt werden.

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.

7. Betrieb

7.1. 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	1	Gibt an wieviele Tage die Log-Files gesichert werden

Tabelle 7.1. 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

7.2. 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:

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

7.3. 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.

7.3.1. Eingebautes Monitoring

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

7.4. 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]

7.4.1. Resourcen

7.4.1.1. Statistiken

Übersicht:

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

7.4.1.2. 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_4 (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": {
      ...
   }

}
```

A. Anhang

A.1. Apache Reverse Proxy

Diese Installationsanleitung dient als Anlehnung und kann nach Systemvorgaben angepasst werden

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

Das Mutual Zertifikat liegt unter 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.

A.2. Testtool

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 v4.0 und ELM v5.x Format versendet werden können.

A.2.1. Installation

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

A.2.2. 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 SwissdecAdapters).
sigrootcert	Das Zertifikat wird zum Verifizieren der Signatur der Antwort benutzt (Public Key des SwissdecAdapters).
sslcert	Zertifikat mit dem Public Key zum TLS-Private Key
sslkey	TLS-Private Key für die Klientenauthentifizierung
sslpartnercacert	CA- Zertifikat zum Prüfen des SSL/TLS Serverzertifikats.

Tabelle A.1. TestTool Konfiguration

Mehr Informationen zur Security entnehmen Sie bitte dem Dokument Kapitel 6, *Security*.

Warnung

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.

A.2.3. 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.

A.3. Unicode Tabelle

Sonderzeichen im institution.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.2. Unicode Tabelle für Sonderzeichen

A.4. Referenzierte Dokumente

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

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

A.5. 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
STEPrx	Produkt der Firma itServe AG,*Swissdec* Empfänger zum Empfang von Swissdec Meldungen.
Swissdec	Qualitätslabel für Lohnbuchhaltungssysteme. XML-Standards für die Einheitliche Lohn-Meldung (ELM).