Richtlinien für die Lohndatenübermittlung

Übersicht der Spezifikation Version 4

Lohnstandard-CH (ELM) / SalaryDeclaration

(Einheitliches Lohnmeldeverfahren)

swissdec

Inhaltsverzeichnis

Konventionen

- Text — Dokumentation
- `Text` — Code
- `<Text>` — XML-Element
- [TEXT] — Referenz auf anderes Dokument (jeweils rechts oben auf der Folie)

- Verbindlichkeit von Anforderungen:

Verbindlichkeit	*Wort*
Plicht	*muss*
Wunsch	*soll (sollte)*
Absicht	*wird*
Vorschlag	*kann*

- Für das konzeptionelle Verständnis genügen oft ältere XML *Schema*-, *Instanzdokument*- oder *Path Language (Xpath)*- Bilder d.h.
  verbindlich sind immer nur die **offiziellen XML-Files**!

Grundlagen

- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

Kontext

§

Umsetzung der Gesetze, Verordnungen und Usancen

Richtlinien Lohndaten-
- Verarbeitung
- Transmitter
- Endreceiver
- Barcode

Zertifizierung

Referenz-applikation Lohn-rechner

Test

- Rund um den Lohn

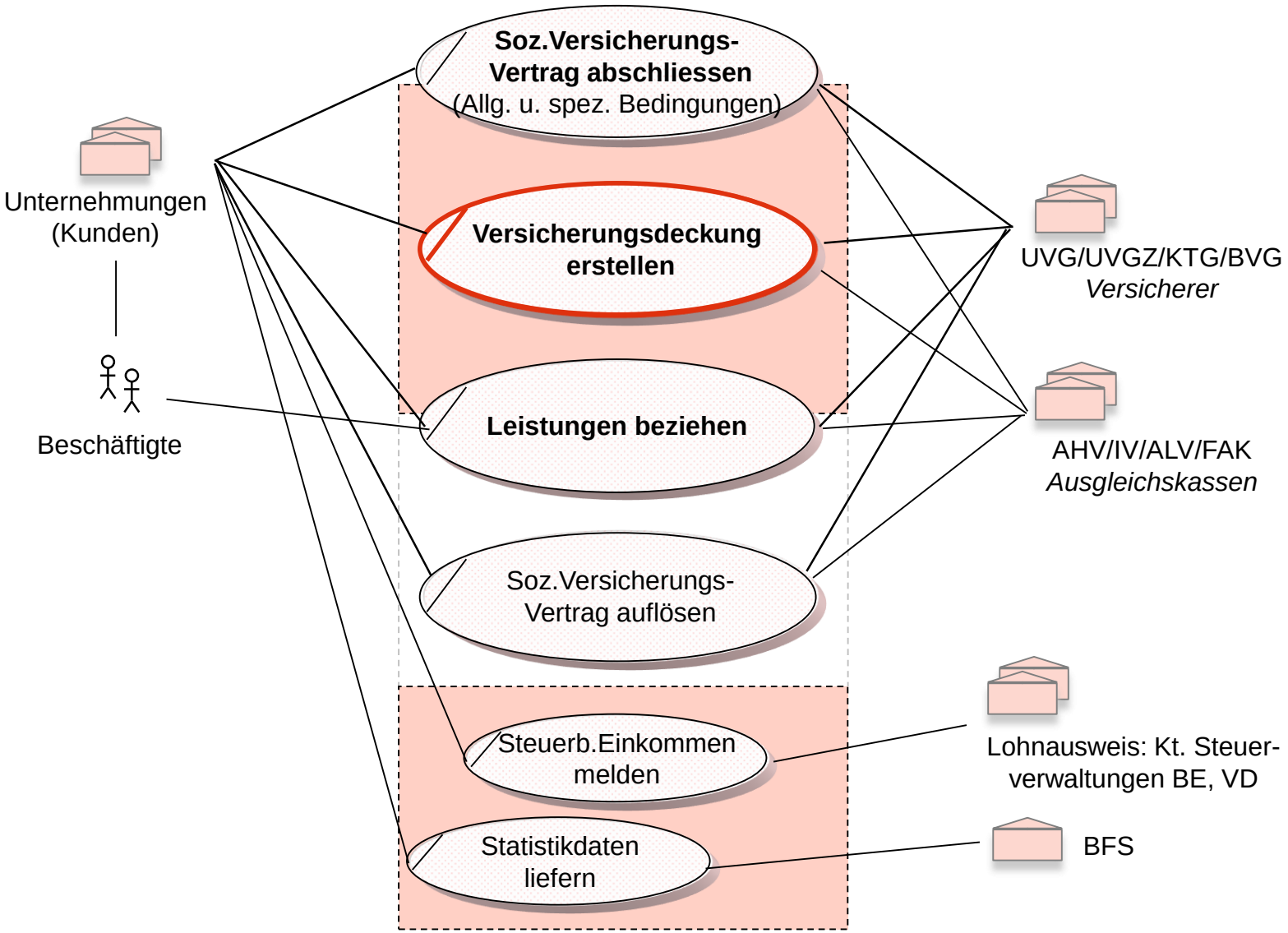
Distributor

Daten

«Mehrwert»

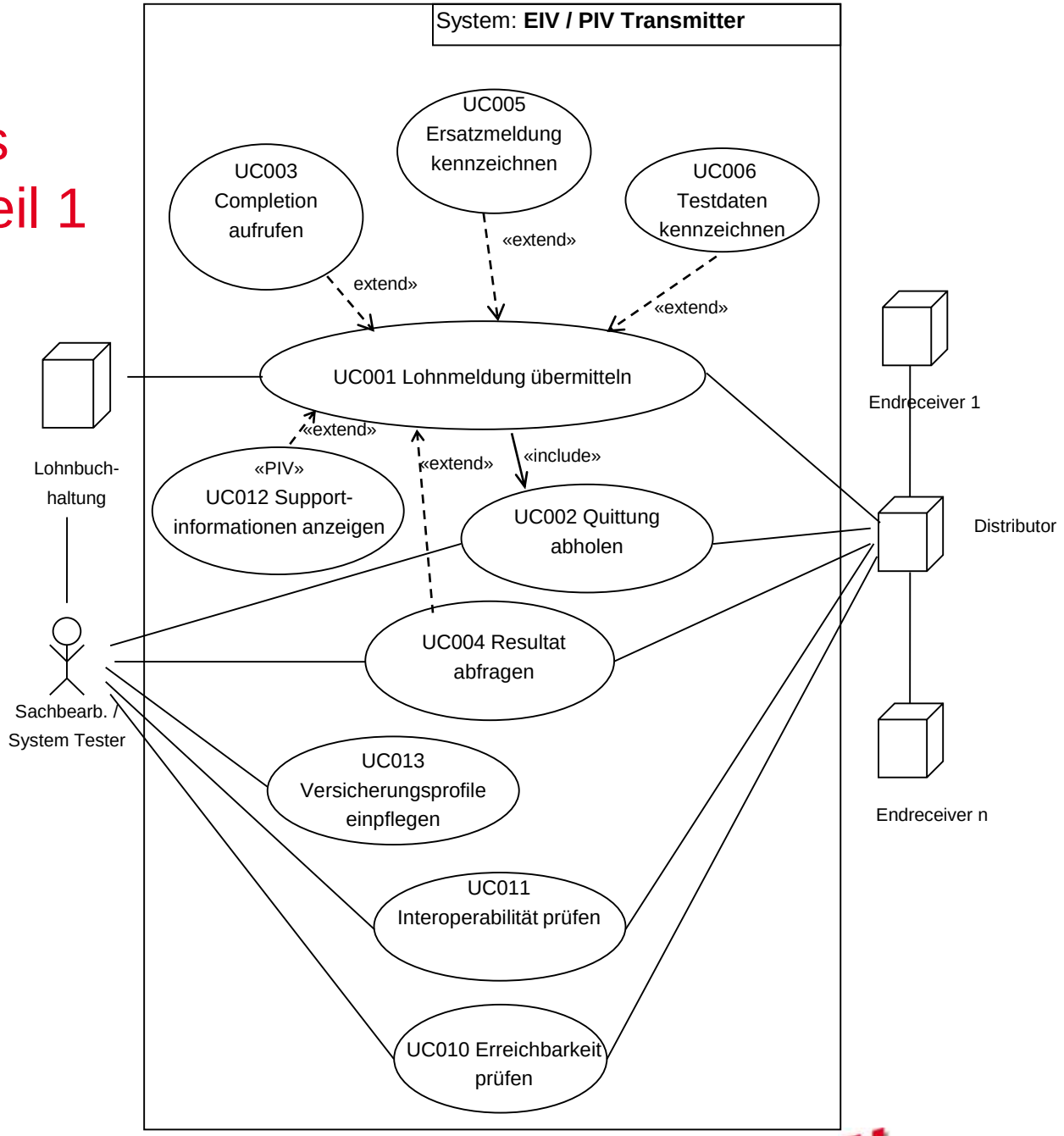
Revision

Gesamte Geschäftsprozesse



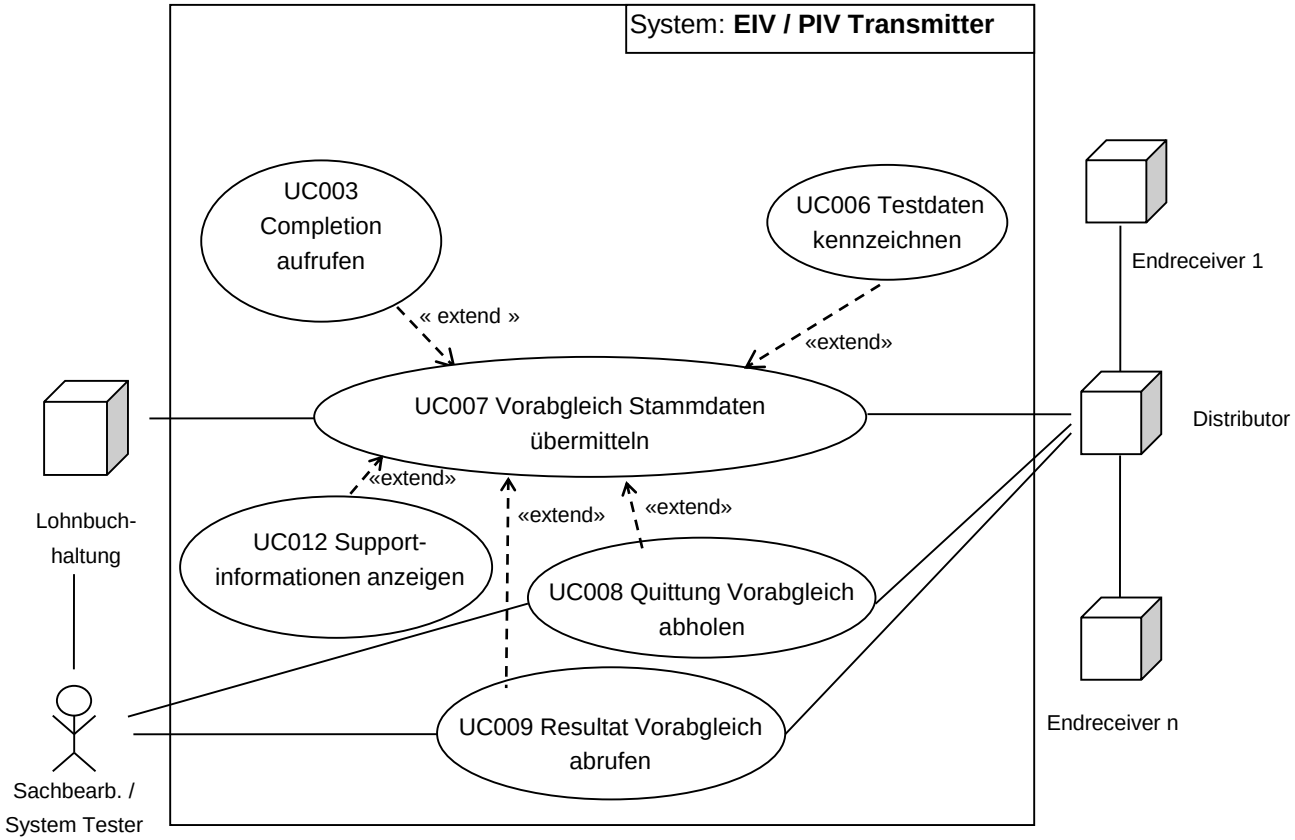
swissdec

Transmitter Requirements Use Cases Teil 1

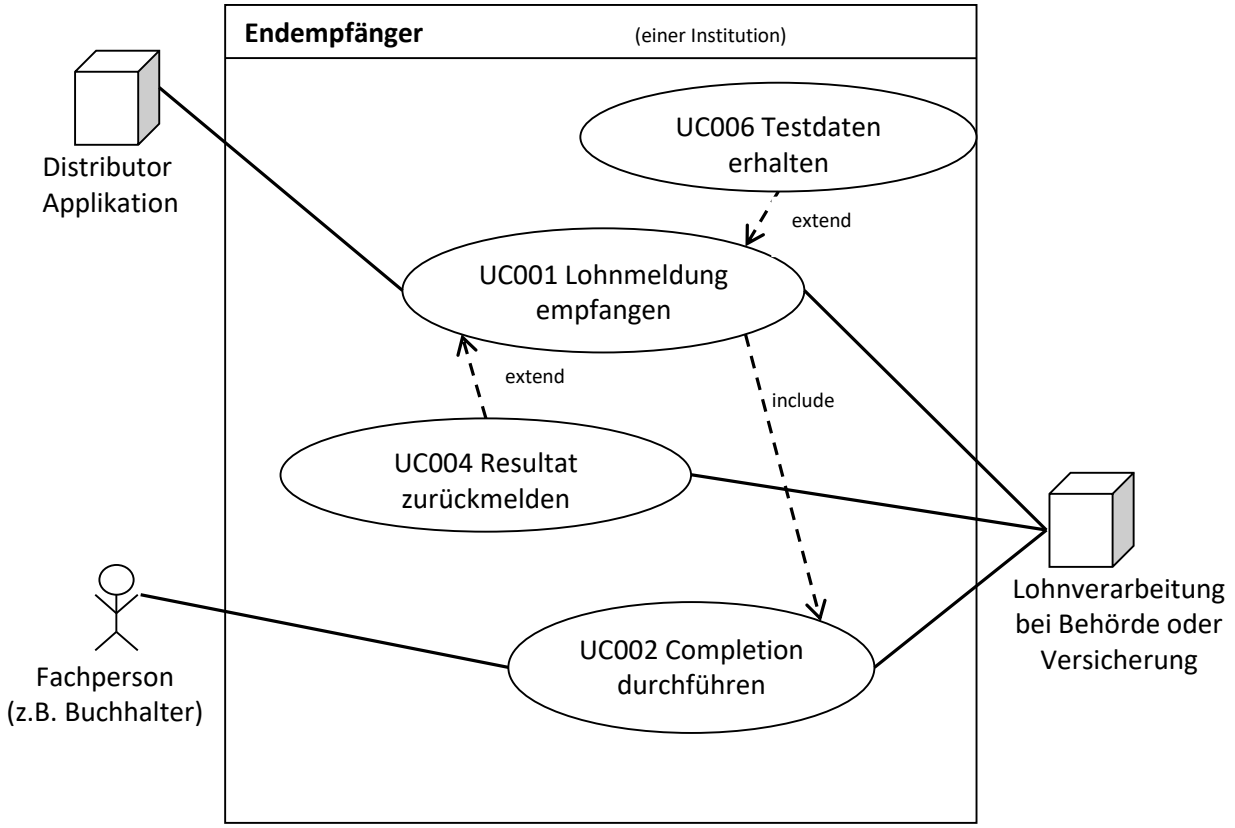


swissdec

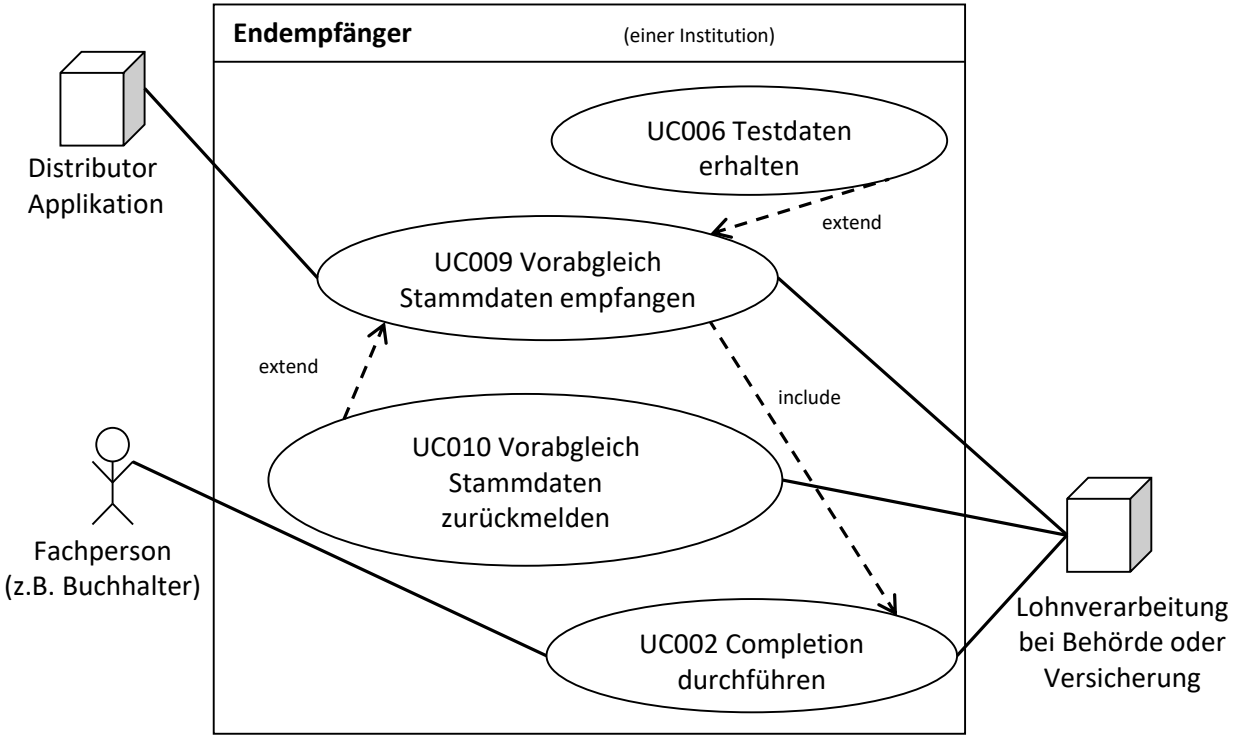
TransmitterRequirements
Use Cases Teil 2



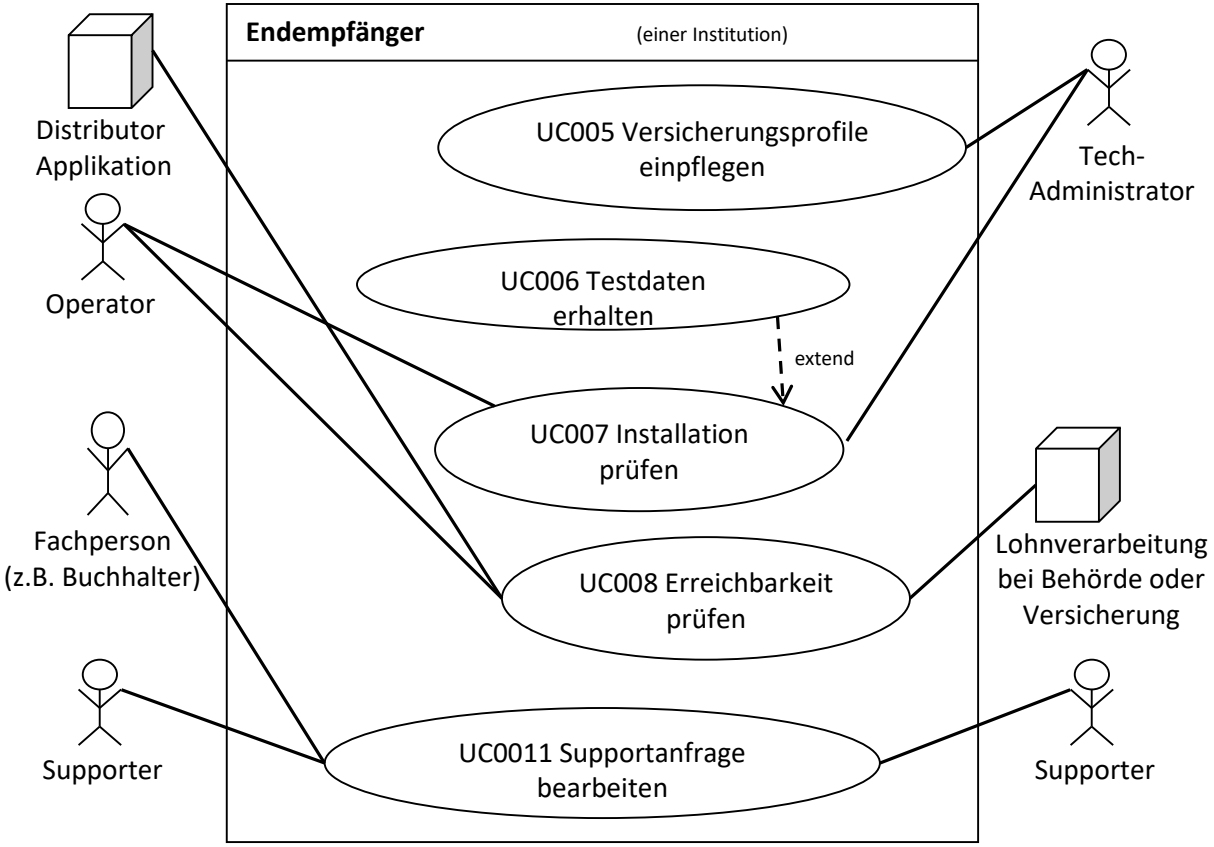
EndReceiverRequirements
Use Cases Teil 1



EndReceiverRequirements
Use Cases Teil 2

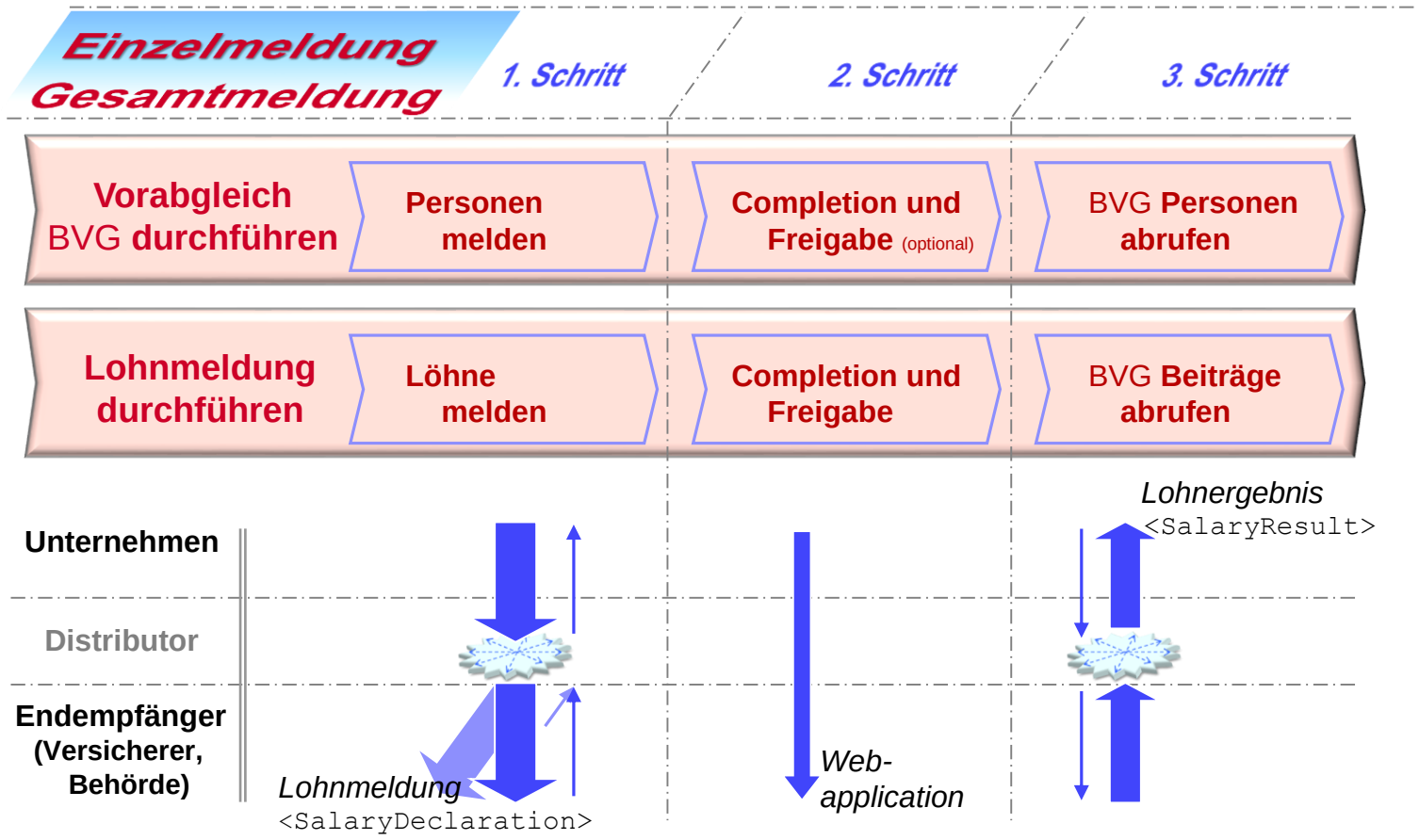


EndReceiverRequirements
Use Cases Teil 3



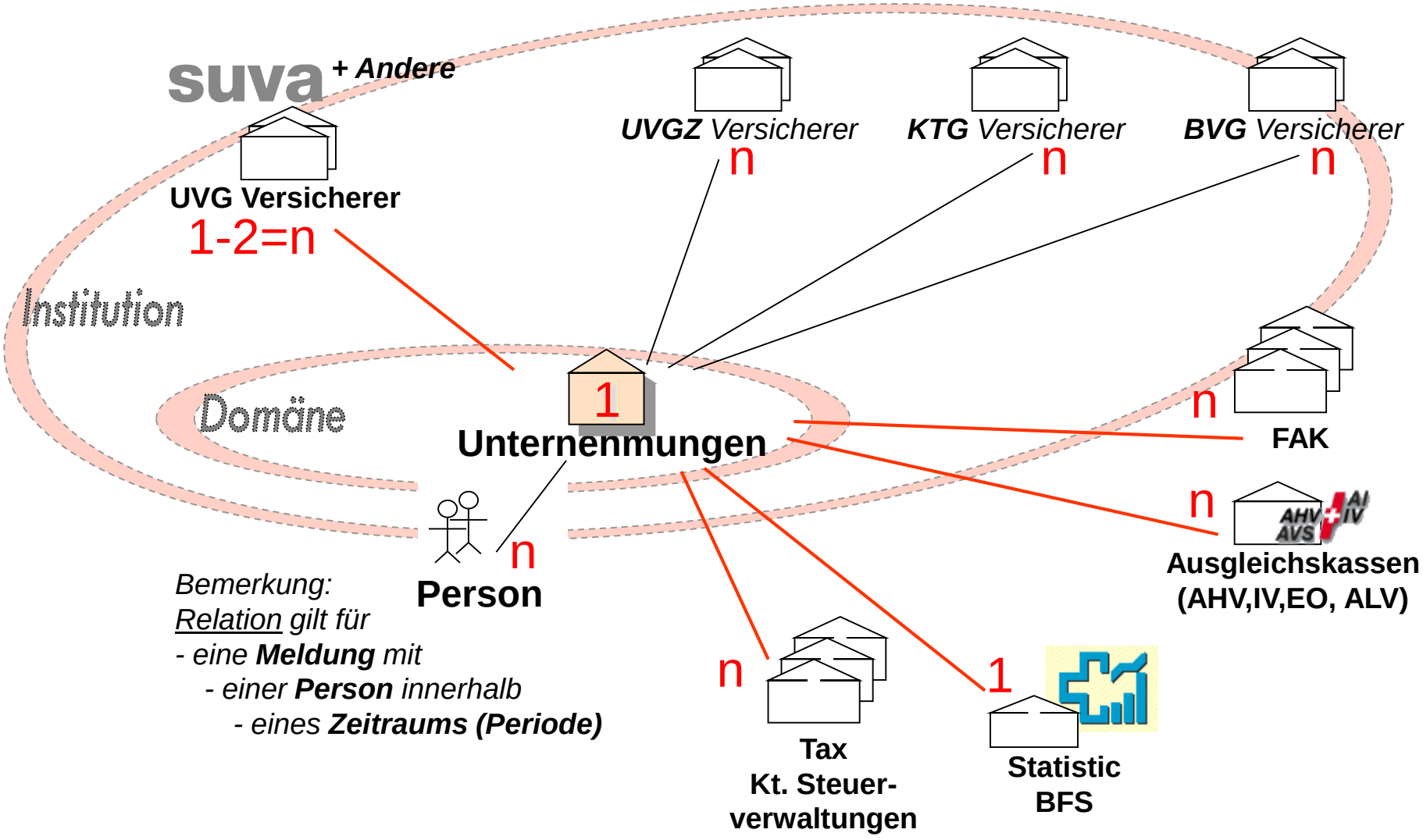
swissdec

Lohnstandard-CH (ELM)



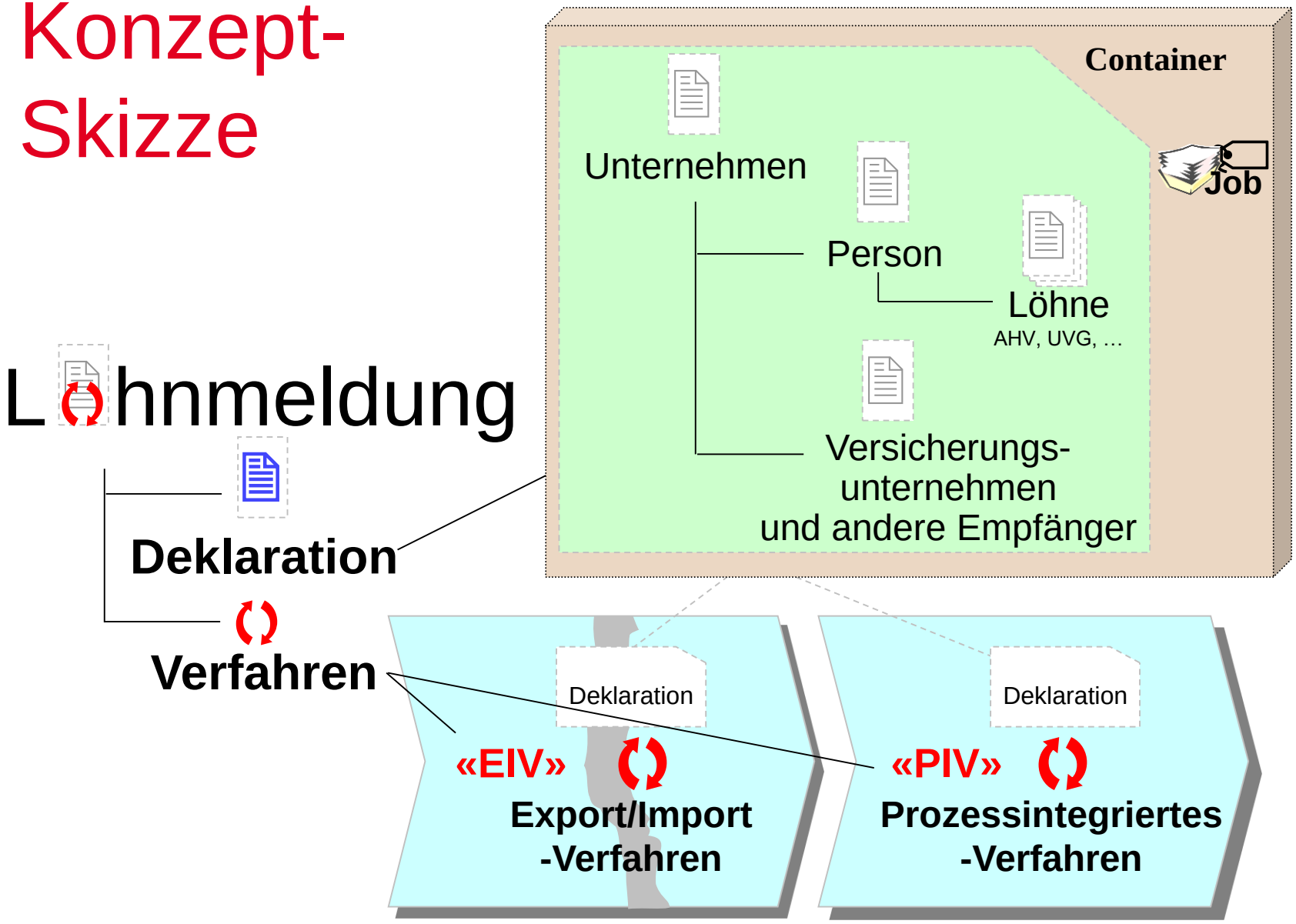
swissdec

Relationen aus Sicht Unternehmung inklusive Partner (Domäne u. Institution)



swissdec

Konzept-Skizze



swissdec

Konzept-Skizze (II)

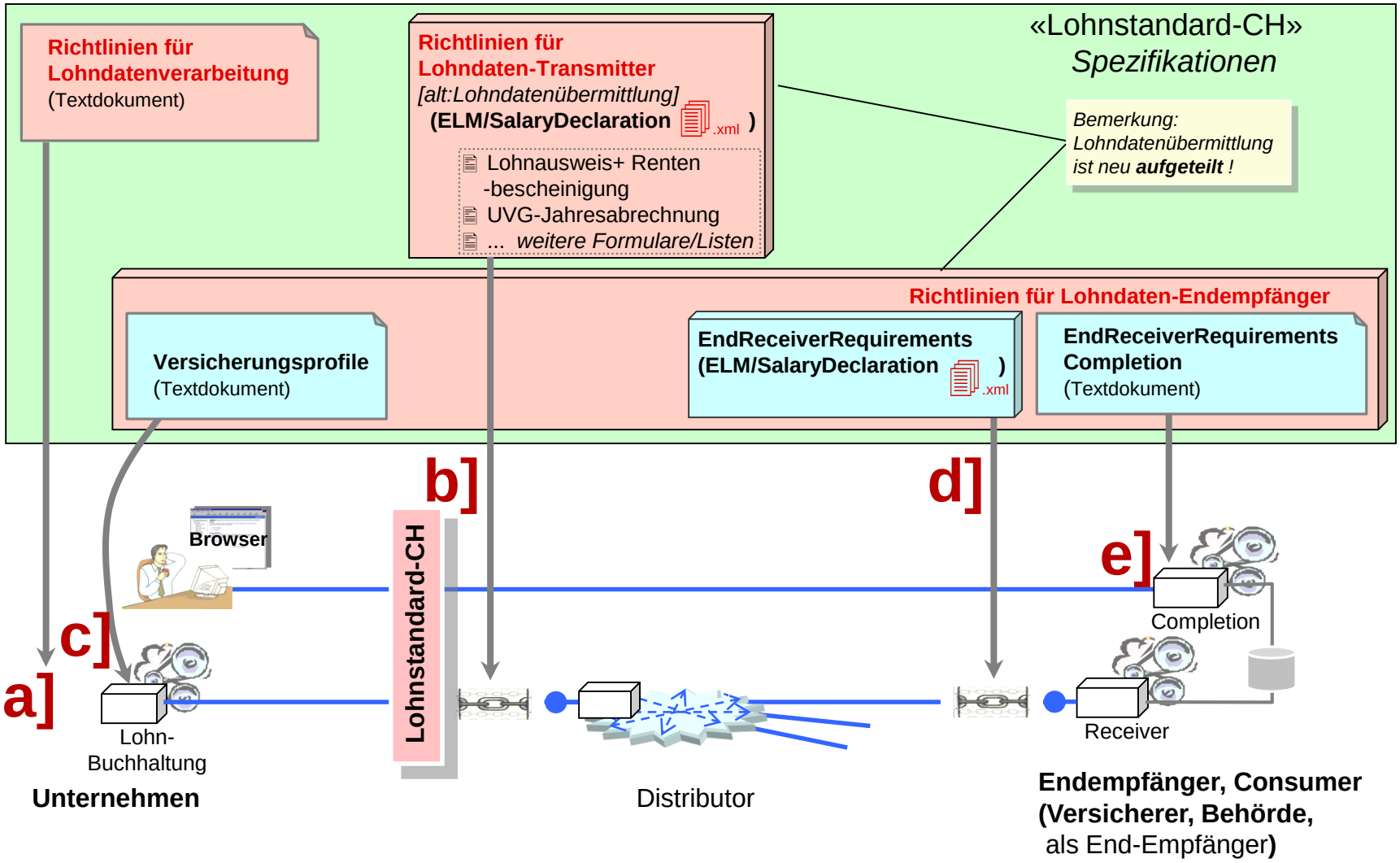
Die Lohnmeldung besteht aus zwei wesentlichen Elementen:

- DEKLARATION (**WAS** wird übermittelt?)
  - In der Deklaration werden Informationen verwaltet, die einer eigentlichen Lohnmeldung entsprechen.

- VERFAHREN (**WIE** wird übermittelt?)
  - Im Verfahren wird der Datenaustausch zwischen den Parteien definiert. Dabei werden zwei Verfahrensvarianten unterstützt:
    - Prozessintegriertes Verfahren (PIV) *[m2m und h2m Ansatz]*
    - Export/Import Verfahren (EIV) *[h2m Ansatz]*

h2m (Human to Machine)
m2m (Machine to Machine)

Was wird wo spezifiziert



swissdec

a] bis e] Beschreibung

RL-LDV

a] **Richtlinen für Lohndatenverarbeitung**
Fachliche Anforderungen für eine zertifizierte Lohnbuchhaltung

RL-LDT

b] **Richtlinen für Lohndaten Transmitter**
(Richtlinien für Lohndatenübermittlung alt)
Technische Anforderungen für eine zertifizierte Lohnbuchhaltung
Lohnstandard-CH Version M.m wsdl: **SalaryDeclarationService.wsdl**
Namespace: http://www.swissdec.ch/schema/sd/yyyymmdd/SalaryDeclarationService

RL-LDE
Richtlinen für Lohndaten End-Empfänger
Technische Anforderungen für End-Empfänger Institutionen, Consumer

c] **Versicherungsprofile**
Konfigurationsdaten zur Adressierung, Produkte (Code), ...

d] **EndReceiverRequirements**
Anforderungen an die Standard-Kopplung für eine End-Empfänger-Institution
Lohnstandard-CH Version M.m wsdl: **SalaryDeclarationConsumerService.wsdl**
Namespace: http://www.swissdec.ch/schema/sd/yyyymmdd/SalaryDeclarationConsumerService

e] **EndReceiverReqCompletion**
Maskenbeschreibung zur WebApplikation zur Freigabe der Lohnmeldung

Deklaration (Datenstruktur zur Lohnmeldung)

Übersicht

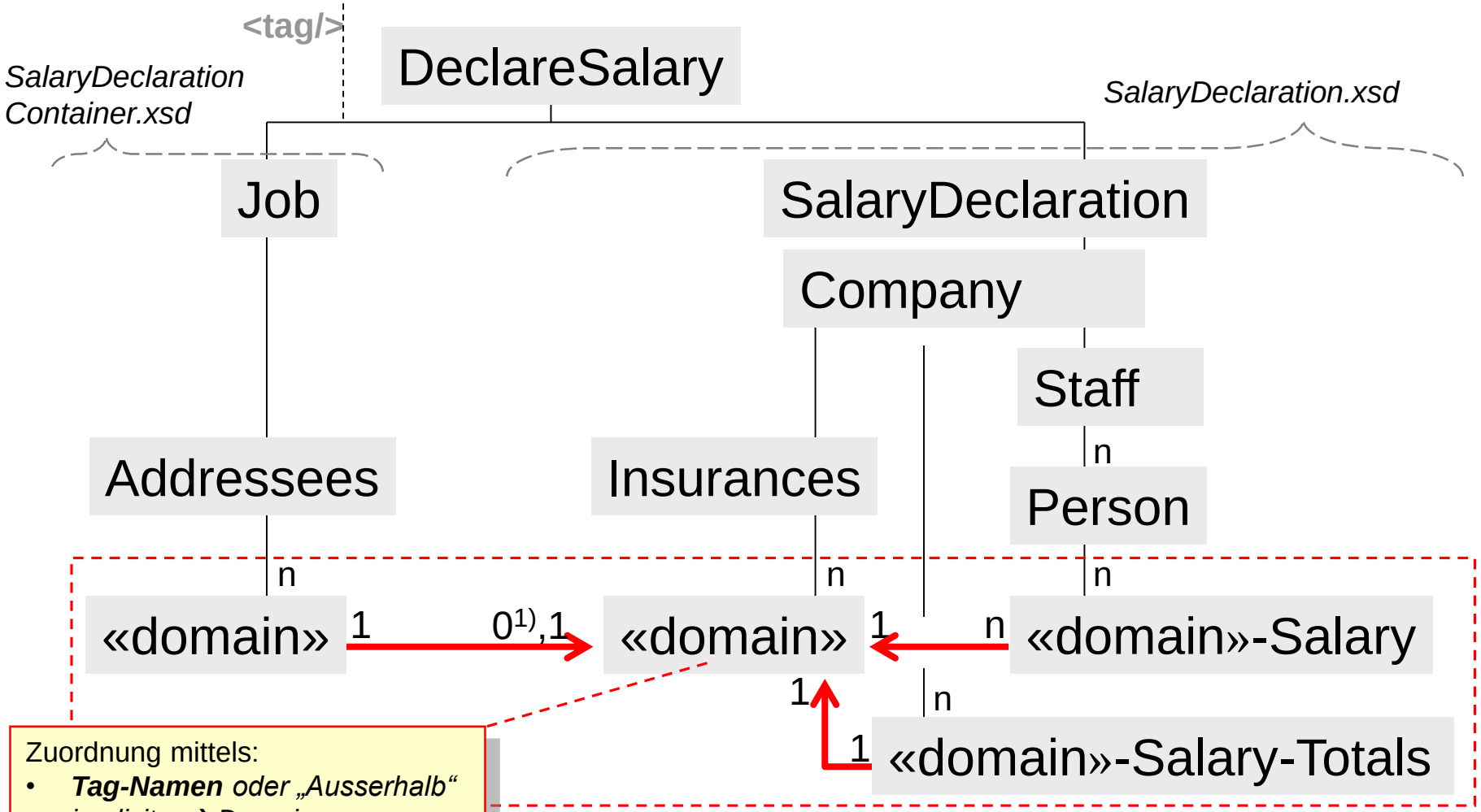
- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

Deklaration

- Die Deklaration beschreibt die Datenstruktur und antwortet entsprechend auf die Frage:

Was wird übermittelt?

Essentielle Entitäten: Hierarchie und Kardinalität



swissdec

Bemerkungen zu verschiedenen Identifikationen (I)

In den xml-Strukturen (wsdl, xsd) finden sich verschiedene **Identifikationen**, welche in den folgenden Folien kurz beschrieben werden.

- Die **Identifikation von Personen** wird mittels folgender Elemente gesichert (z.B. im Abgleich) :
  - Sozialversicherungsnummer
  - Geschlecht
  - Geburtsdatum

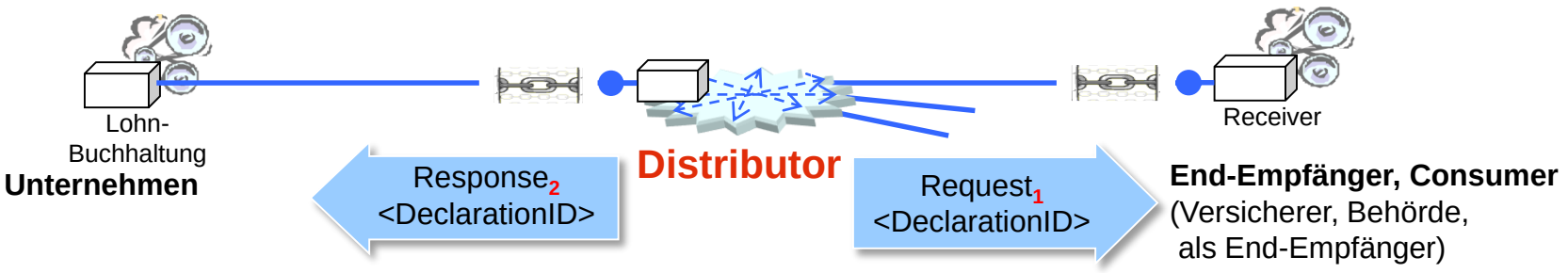
Bemerkungen zu verschiedenen Identifikationen (II)

In den xml-Strukturen finden sich verschiedene IDs, welche hier kurz beschrieben werden.

- RequestID
  - Aus der **Sicht des Transmitters** hat jeder Request eine eindeutige ID, auch wenn mehrmals dieselben Daten übermittelt werden.
- ResponseID
  - Aus der **Sicht des End-Empfängers** hat jede Response eine eindeutige ID
  - Zusätzlich werden die RequestID‘s gespiegelt. Zusammen ergeben sie damit für **Transmitter und End-Empfänger** eine eindeutige ID.
  - In Lohnstandard-CH / RL-LDÜ Version 2.2 dienen Request- und ResponseID als Fallnummer für den Support
- MonitoringID
  - Die MonitoringID wird benötigt, um der RefApps mitzuteilen, woher eine Übermittlung stammt und welche Konfiguration auf sie angewendet werden soll
  - In der Produktion ist die MonitoringID überflüssig
- institutionID und institutionIDRef (xml Attribute)
  - Verknüpfung von Adressierung, Löhnen, Lohn-Total und Versicherung innerhalb des SalaryDeclaration Instanzdokumentes
- DeclarationID
  - Gleiche ID in allen Requests (Distributor→End-Empfänger), Responses (Distributor→Transmitter), Masken und Pdf‘s, welche zum **identischen Geschäftsvorfall** gehören. Damit kann sie neu als **Fallnummer** für den **Support** verwendet werden.
- InsuranceID
  - Eindeutige ID, die jedem Versicherer dauerhaft zugeteilt wird.
  - Die InsuranceID kann aus der Liste der Endempfänger auf http://www.swissdec.ch ausgelesen werden

DeclarationID

- Gleiche ID in allen Requests$_1$, Responses$_2$, Masken und Pdf's, welche zum **identischen Geschäftsvorfall** gehören. Damit kann sie als Fallnummer für den Support verwendet werden.
- Bemerkung zu Requests$_1$
  Hier sind die Requests vom **Distributor zum End-Empfänger** gemeint
- Bemerkung zu Responses$_2$
  Hier sind die Responses vom **Distributor zum Transmitter** gemeint
- Die DeclarationID wird nur vom **Distributor vergeben**, kontrolliert und jedem Partner (Lohnbuchhaltung/Transmitter und End-Empfänger) in die **zusammengehörigen Responses** oder **ConsumerRequests** automatisch eingesetzt.

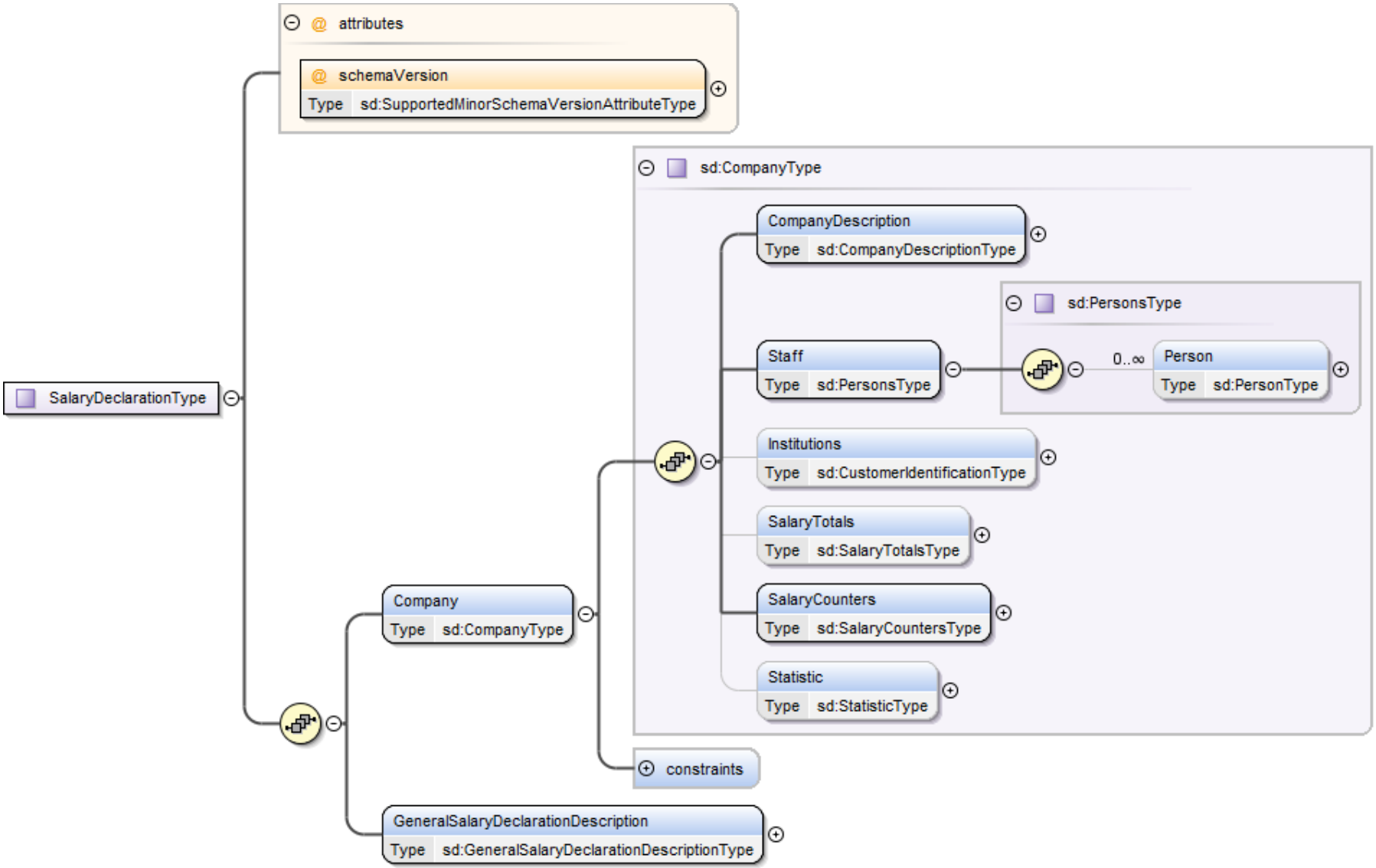


Deklaration (Datenstruktur zur Lohnmeldung)

SalaryDeclaration

- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

Struktur von SalaryDeclaration.xsd



swissdec

SalaryDeclaration.xsd

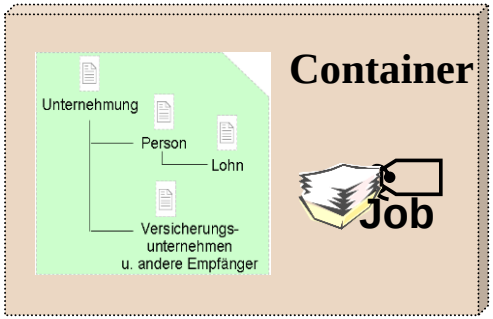
- Das Schema SalaryDeclaration.xsd beschreibt sämtliche relevanten Daten zu
  - Unternehmen
  - Angestellte
  - Institutionen
  - Löhne
- In einem SalaryDeclaration-Instanzdokument befinden sich also alle Informationen, die für den Endempfänger relevant sind.

Deklaration (Datenstruktur zur Lohnmeldung)

SalaryDeclarationContainer

- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

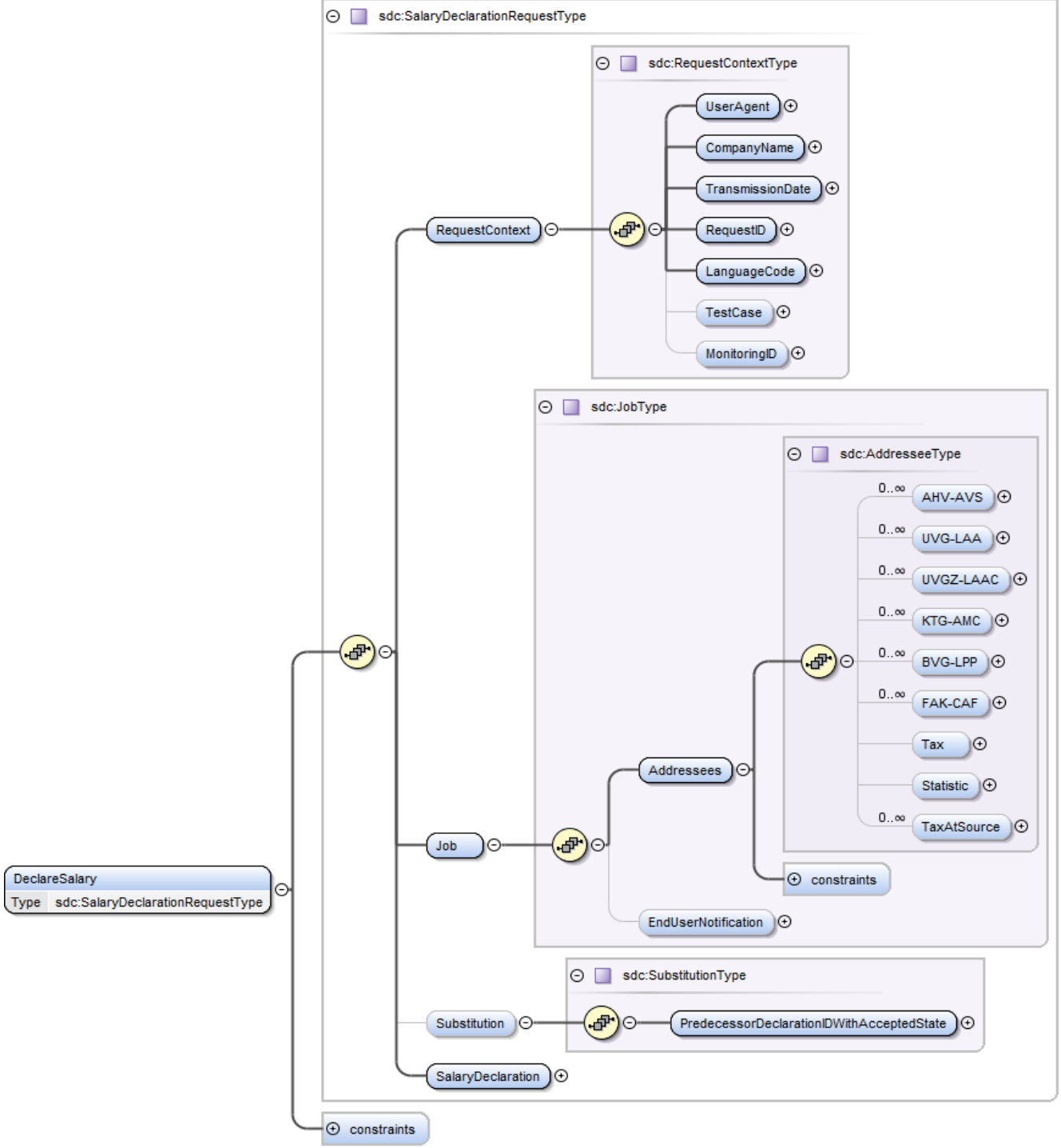
Container



Hier werden technische Informationen zu Kontrolle und Steuerung des Übermittlungsverfahrens definiert.

- Auftragskontext
- Arbeitsauftrag mit Adressierung der Empfänger
- Substitutionsdaten
  - Kennzeichnung als eine Ersatzmeldung
- Der Container enthält die relevanten Informationen für die Kommunikation zwischen Transmitter und Distributor
- *und „eingeschlossen“ die eigentlichen Nutzdaten der Lohnmeldung*

Container

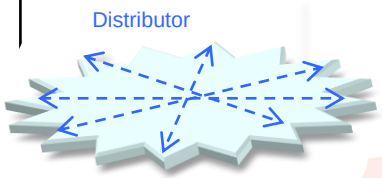

DeclareSalary
Type sdc:SalaryDeclarationRequestType
sdc:SalaryDeclarationRequestType
RequestContext
sdc:RequestContextType
UserAgent
CompanyName
TransmissionDate
RequestID
LanguageCode
TestCase
MonitoringID
Job
sdc:JobType
Addressees
sdc:AddresseeType
0..∞ AHV-AVS
0..∞ UVG-LAA
0..∞ UVGZ-LAAC
0..∞ KTG-AMC
0..∞ BVG-LPP
0..∞ FAK-CAF
Tax
Statistic
0..∞ TaxAtSource
constraints
EndUserNotification
Substitution
sdc:SubstitutionType
PredecessorDeclarationIDWithAcceptedState
SalaryDeclaration
constraints

Verfahren

Übersicht

- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

Übermittlungsverfahren und Verteilung mittels Distributor

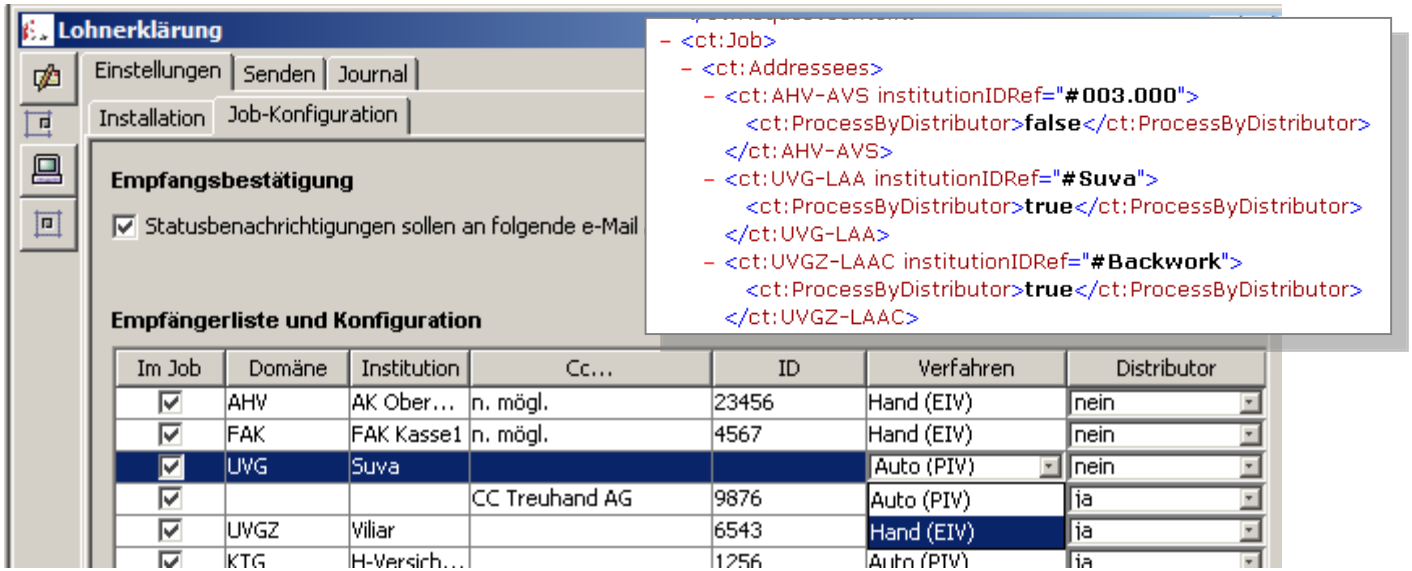
	Prozessintegriertes Verfahren (PIV)	*Export/Import Verfahren (EIV)*
 **Verteilung mittels Distributors**	• UVG • UVGZ • KTG • BVG • Tax • Statistic • AHV • FAK • TaxAtSource	• UVG • UVGZ • KTG • ~~BVG~~ • Tax • Statistic • AHV ohne EMA • FAK • ~~TaxAtSource~~
Direkter File-Upload mittels eigener Portal- oder Web-Lösung		• AHV • FAK

swissdec

Übermittlungsverfahren und Verteilung mittels Distributor

- Übermittlungsverfahren
  - Prozessintegriertes Verfahren (PIV) und Export/Import Verfahren (EIV)
  - Der Distributor unterstützt beide Verfahren (PIV und EIV)
  - AHV und FAK werden im Moment nur direkt und mittels EIV erreicht
- Verteilung mittels Distributor

  `<Addressees>...`**`<ProcessByDistributor>`**
  - Damit kann durch einfaches Umstellen von true auf false, eine neue Verteilung ohne Entfernen der Daten erfolgen.
  - In einer gemischten Übermittlung (z. B. AHV mittels EIV und der Rest mittels PIV) kann das gleiche Instanzdokument mit den entsprechenden Werten verwendet werden.



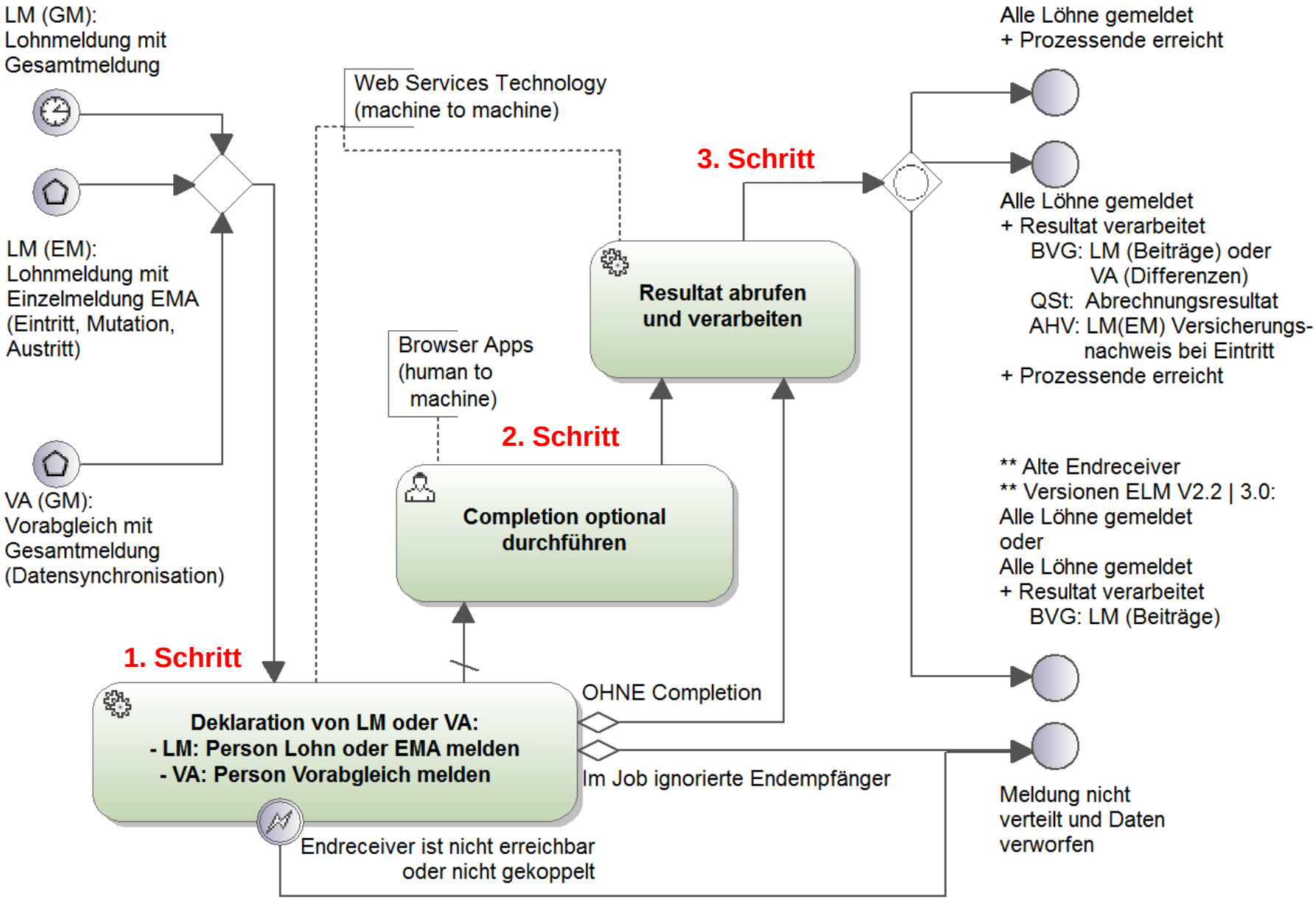
swissdec

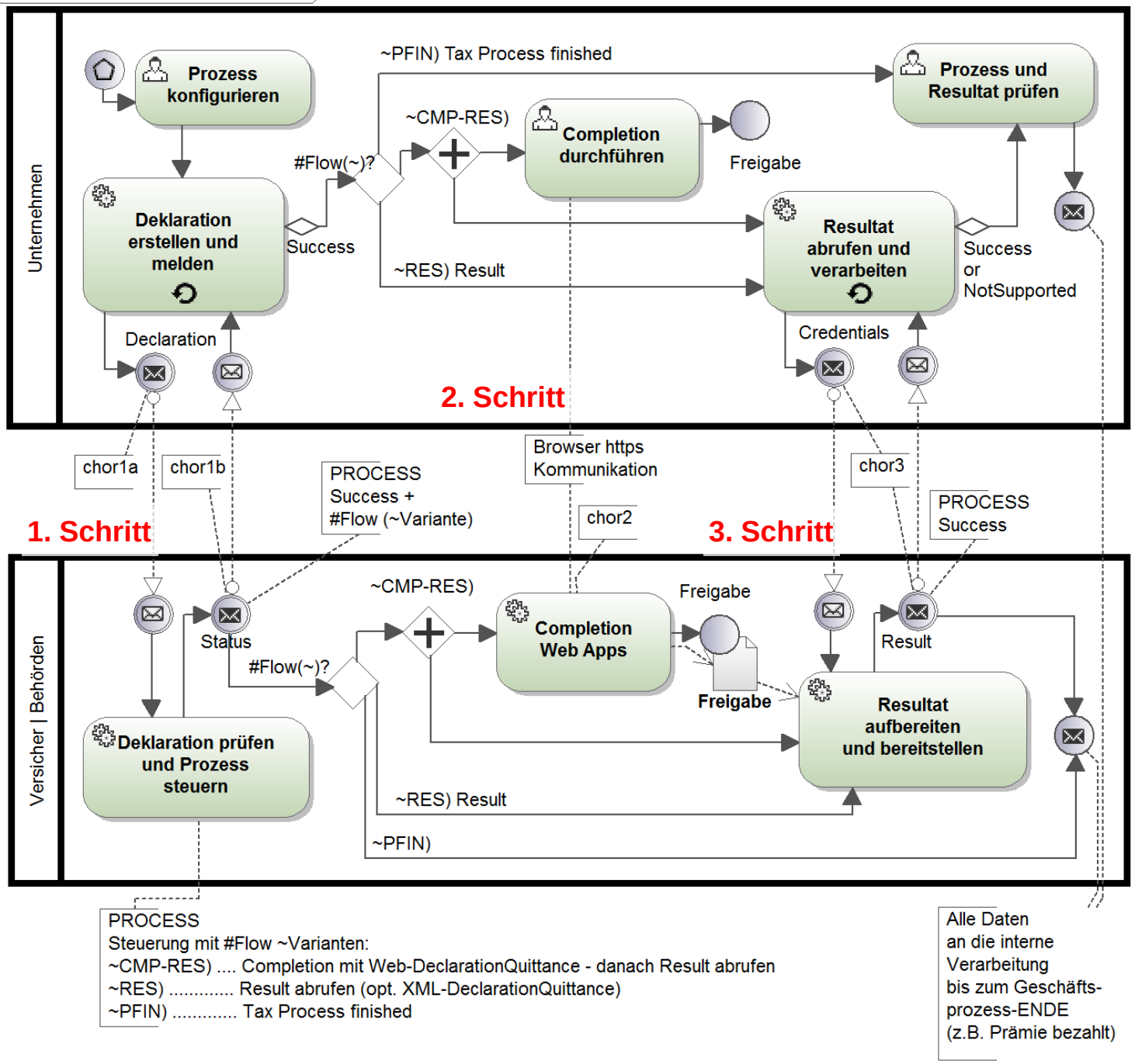
Verfahren

Prozess

- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

2a. (BPMN2) SOLL essentieller Prozess für ELM V4





PROZESS-MUSTER

neu für V4

swissdec

Verfahren

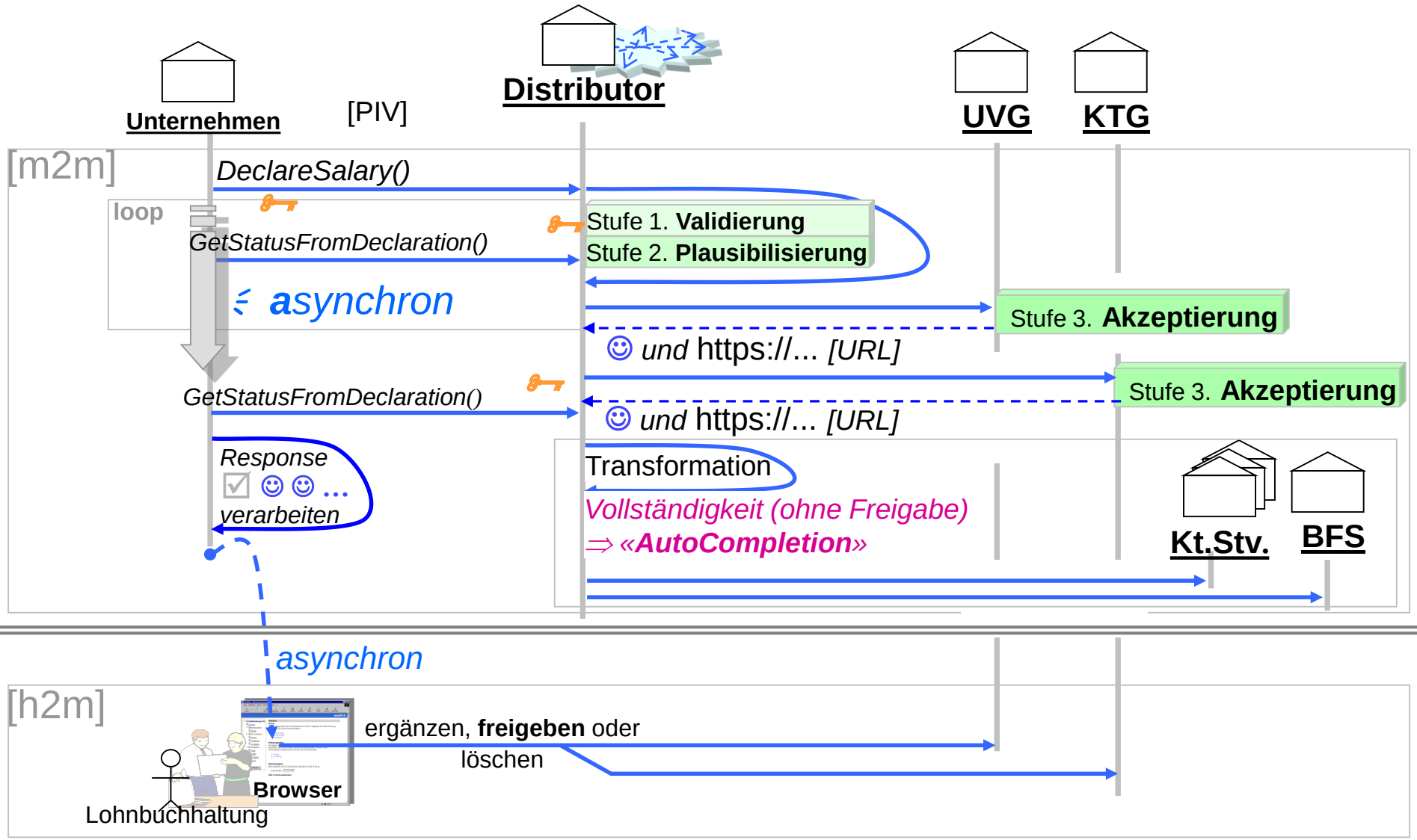
PIV

- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

Prozessintegriertes Verfahren (PIV)

1. Der Transmitter erzeugt aus den Lohndaten ein SalaryDeclaration-Instanzdokument, welches mittels WebService-Technologie via Distributor an die Versicherung oder andere Empfänger übertragen wird. Danach wird die Quittung automatisch ausgewertet (z.B. Fehler, Signatur, …) und die für die Completion relevanten Informationen werden dargestellt.
2. Ergänzen und Freigabe durch den Benutzer/Kunden zwecks Bearbeitung durch die Versicherung
3. Abfrage eines Resultats/einer Quittung beim Distributor, um den Prozess abzuschliessen

Distributor Workflow «ASYN»



swissdec

Asynchrones Protokoll

Ziel: ***Möglichst skalierbar und flexibel***

- **Erste Operation** `DeclareSalary()` mit Job (Adressierung und Lohndaten) zum Distributor und seinem Agenten liefert Key für die zweite Operation.
- **Zweite Operation** `GetStatusFromDeclareSalary()` mit Key vermittelt mehrere URL's / States *(Stufe:* ***Akzeptanz*** *)* der verschiedenen End-Empfänger
  Der Agent hält die Details zum Job
  - Eine Liste mit sämtliche States und URL's der adressierten Empfänger
- Muster «m2m\h2m» integriert den neuen Distributor-Knoten:
  - Response-Struktur wird um Job-Details (alle States, URLs) erweitert
  - Die Links werden von der Lohnbuchhaltung direkt aus dem Response aktiviert.

- Timeout: Lösung: erneutes Senden
- Unvollständiger Job: Lösung: erneutes Holen der Job-Daten (polling)
- SalaryDeclarationFault = «Meldung wurde **nicht** akzeptiert !!»

Synchrones Protokoll

- Da die asynchrone Übermittlung mehr Vorteile aufzuweisen hat und der Übermittlungsprozess als Ganzes so oder so asynchron ist (DeclareSalary, getStatus, getResult), wird das synchrone Übermittlungsprotokoll in V4 nicht mehr unterstützt.

Verfahren

EIV

- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

Export/Import-Verfahren (EIV)

«File-Upload»

1. Der Transmitter erzeugt aus den Lohndaten ein SalaryDeclaration-XML-Instanzdokument[1)], welches mittels «speichern unter» in einer Datei auf dem lokalen Filesystem gesichert wird.
2. Manuelles Suchen der entsprechenden Uploadpage der Versicherung (URL) oder der EIV-Uploadpage auf dem Distributor
3. Upload der Datei, prüfen durch Versicherung und Freigabe seitens Benutzer/Kunde zwecks Bearbeitung durch die Versicherung.

Zusätzlich wären statt eines Online-Uploads auch andere Transportvarianten wie CD-ROM oder DVD (postale Übermittlung) möglich. ***Achtung: Datenschutz***

1) Inkl. SOAP und Container

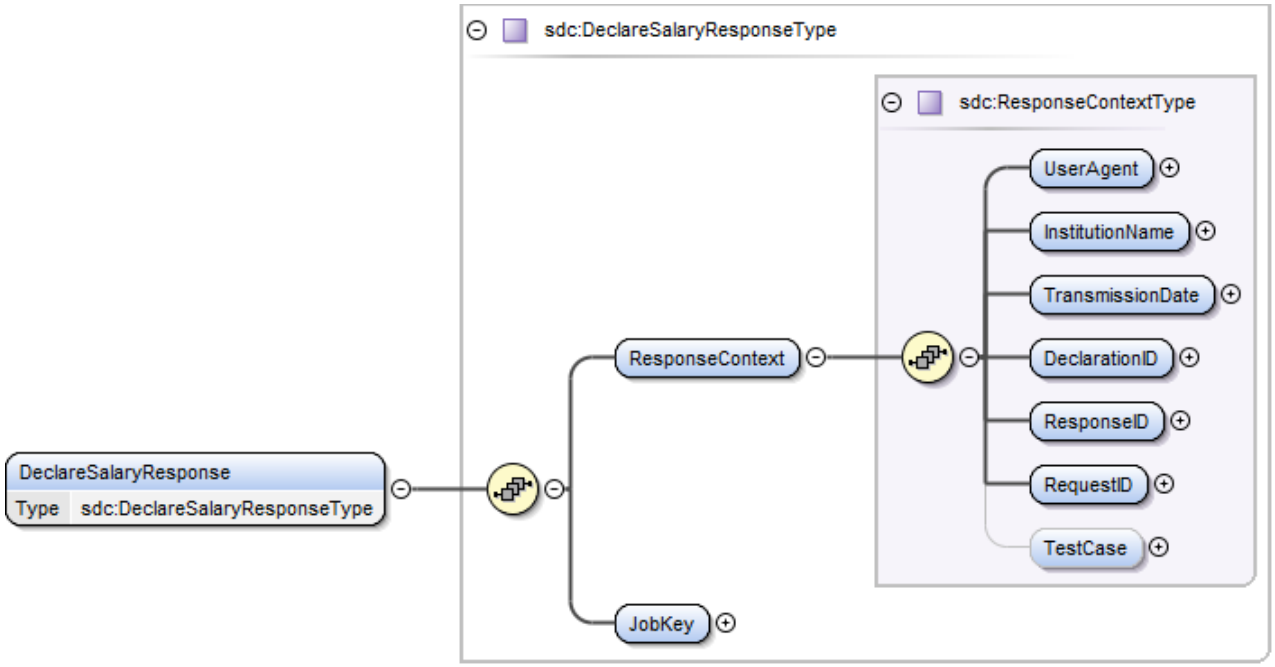
Verfahren

Status

- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

DeclareSalaryResponse

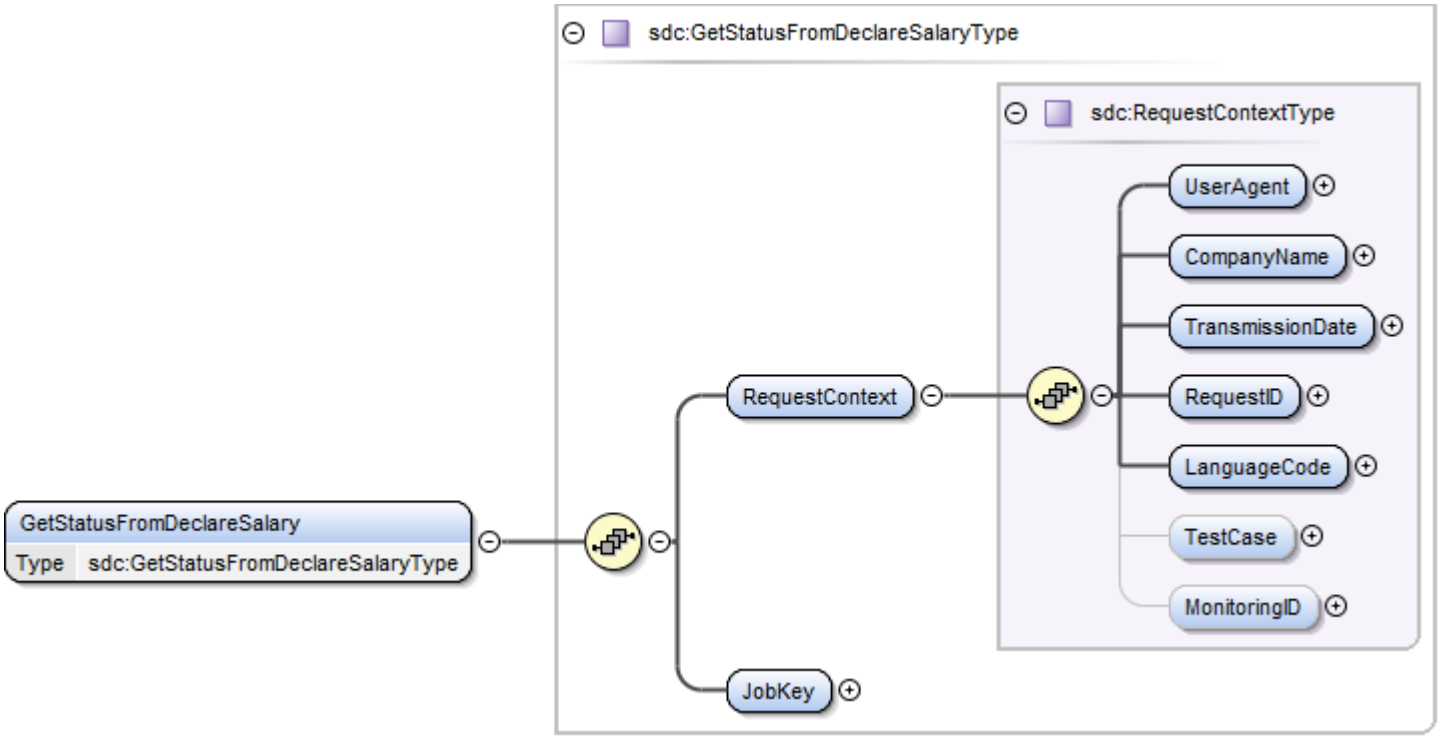
- Auf eine Lohndatenübermittlung mit DeclareSalary wird vom Distributor eine DeclareSalaryResponse zurückgesendet. Diese enthält
  - JobKey fürs Polling
  - ReponseContext mit zusätzlichen Informationen



swissdec

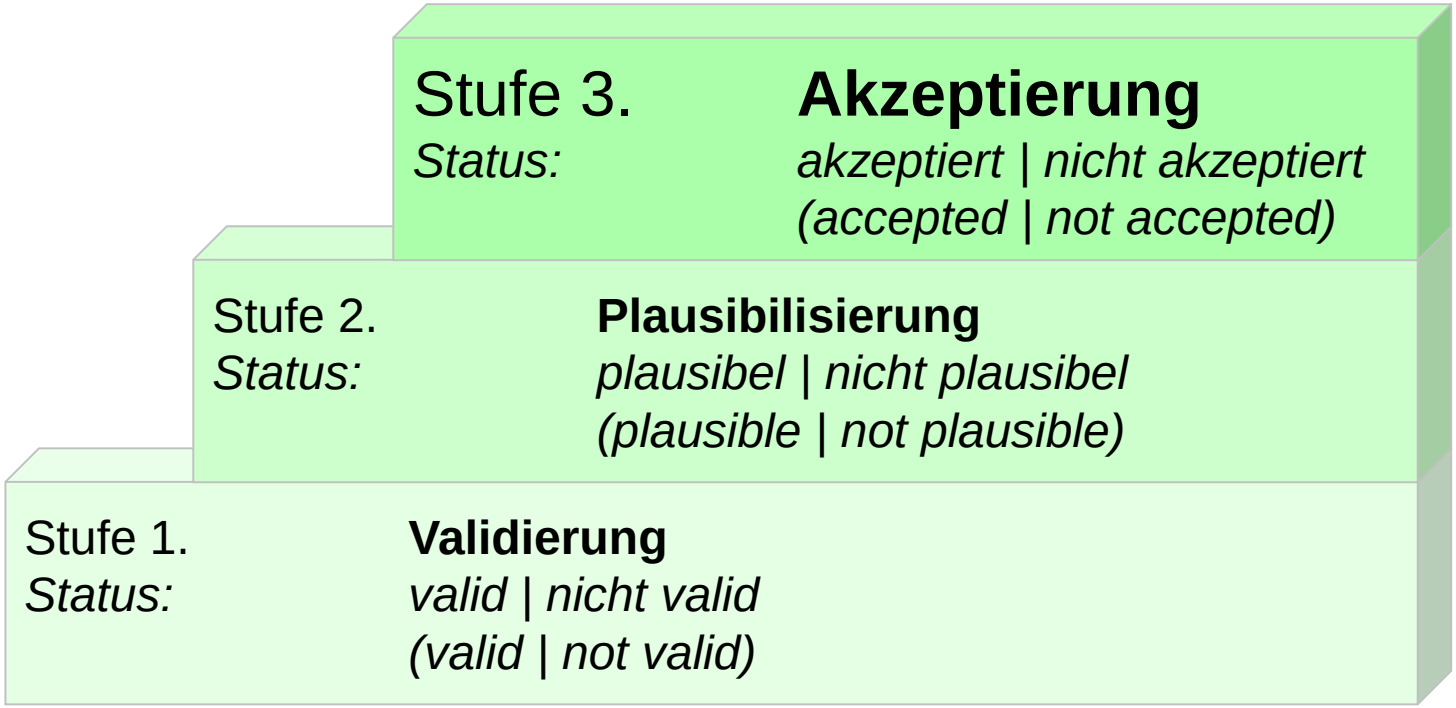
GetStatusFromDeclareSalary

- Mit dem JobKey aus DeclareSalaryResponse wird nun eine Statusabfrage durchgeführt, um einen Statusbericht vom Distributor zu erhalten.



Datengüte der Deklaration

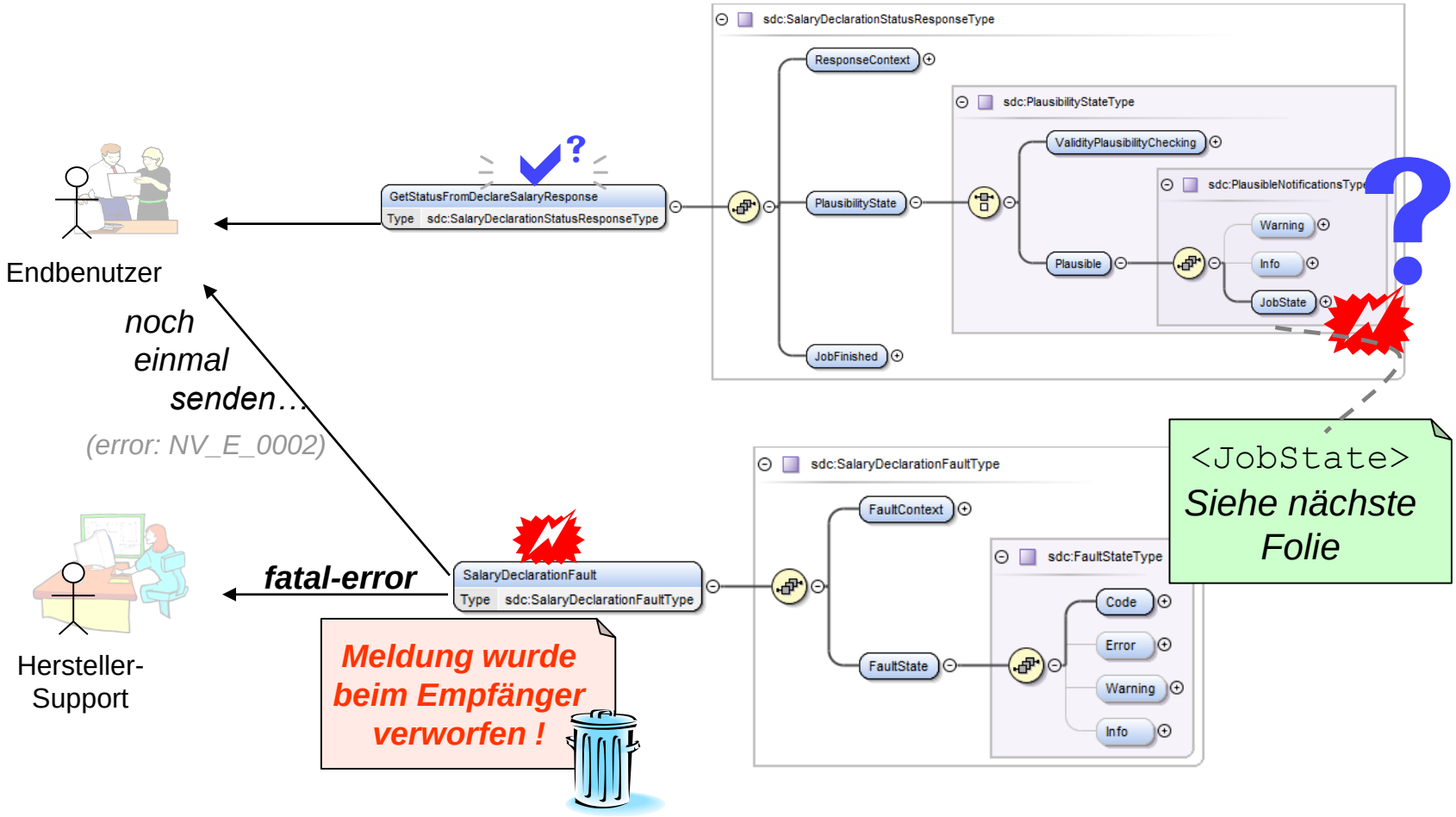
Die übermittelten Daten durchlaufen jeweils folgende Prüfstufen:



Datengüte der Deklaration (II)

Stufe	Fokus	Status
1) Validierung	Überprüft, ob die übermittelten Daten wohlgeformt (well-formed) und in Bezug auf das Schema valide sind.	valid | nicht valid *(valid | not valid)*
2) Plausibilisierung	Überprüft Anforderungen der Geschäftslogik (business logic), welche nicht durch die Validierung abgedeckt werden können. Beispiele sind berechnete oder aggregierte Werte oder auch chronologische Anforderungen bei Datumsangaben. Die Plausibilisierung bezieht sich auf Anforderungen der Geschäftslogik, welche generell sind und für alle Versicherungstypen (-branchen) zutreffen.	plausibel | nicht plausibel *(plausible | not plausible)*
3) Akzeptierung	Überprüft Anforderungen der Geschäftslogik (business logic), welche nicht generell sind. Beispiele sind die Gültigkeit einer Kunden- oder Identifikationsnummer.	akzeptiert | nicht akzeptiert *(accepted | not accepted)*

Quittierung einer Übermittlung



swissdec

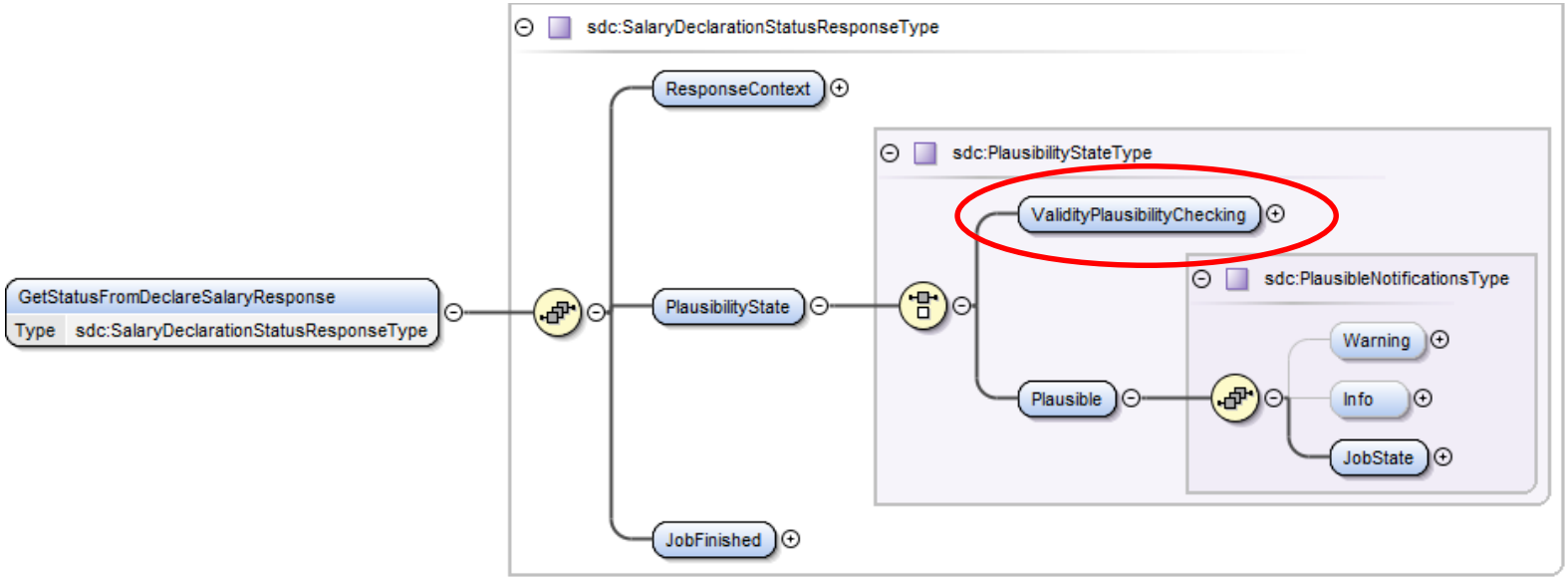
Ziele der Statusmeldung

Klarheit über den Erfolg bzw. Misserfolg der Lohnmeldung

- Erfolg:
  - Status der Bearbeitung im Prozess angeben
  - Weiterführende Informationen (z.B. Completion)
- Misserfolg:
  - Status der Bearbeitung im Prozess bzw. es findet **keine Verarbeitung** statt
  - Weiterführende Informationen zur Behebung des Problems
    - Auffällige, präzise und verständliche Fehlermeldungen
    - Evtl. konstruktive Hilfestellungen anbieten

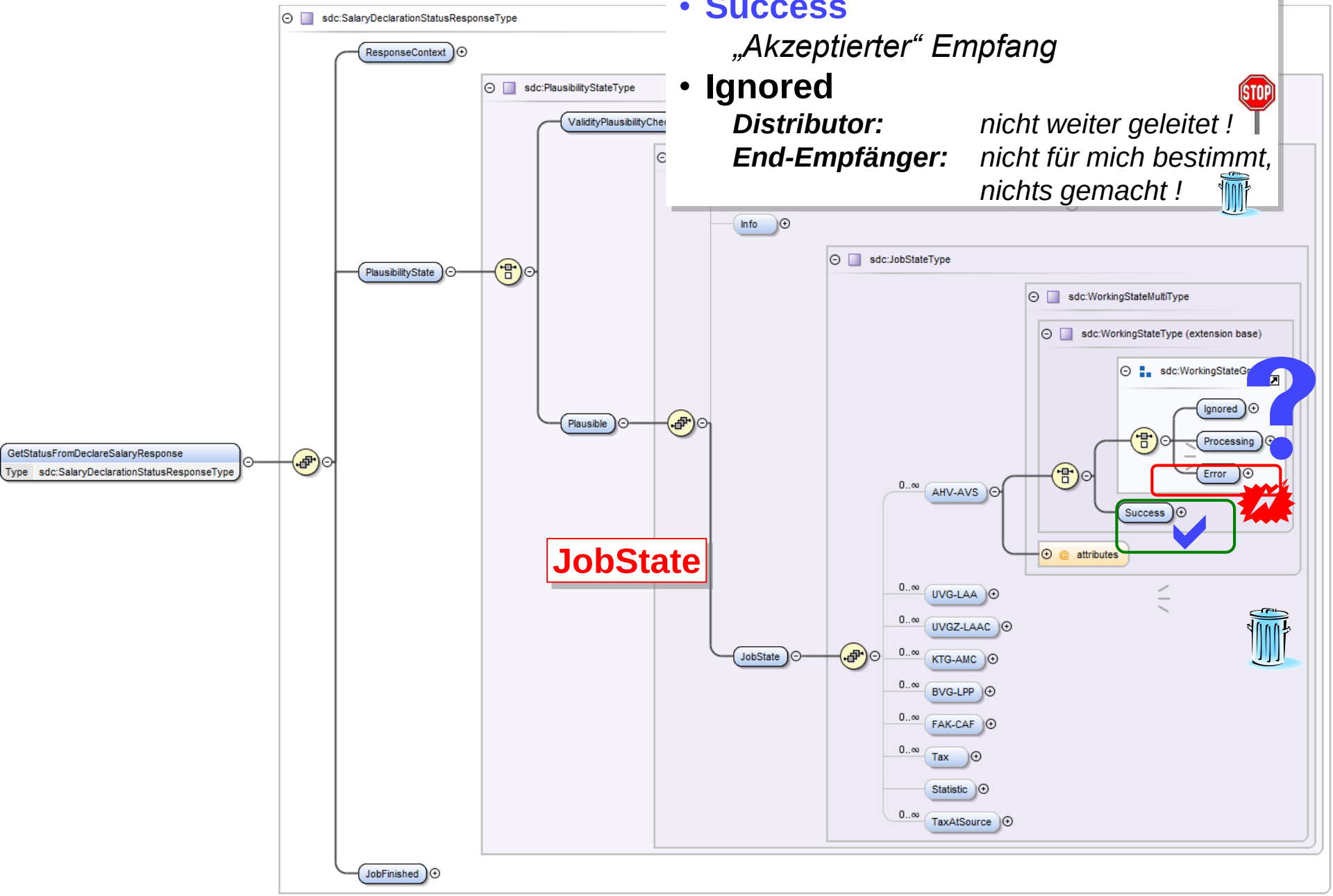
ValidityPlausibilityChecking

- Hat der Distributor die empfangene Datei noch nicht fertig geprüft, wird **kein JobState** ausgegeben. Der Status lautet in diesem Zustand <ValidityPlausibilityChecking>



swissdec

- **Success**
  „Akzeptierter“ Empfang
- **Ignored**
  Distributor: *nicht weiter geleitet !*
  End-Empfänger: *nicht für mich bestimmt, nichts gemacht !*



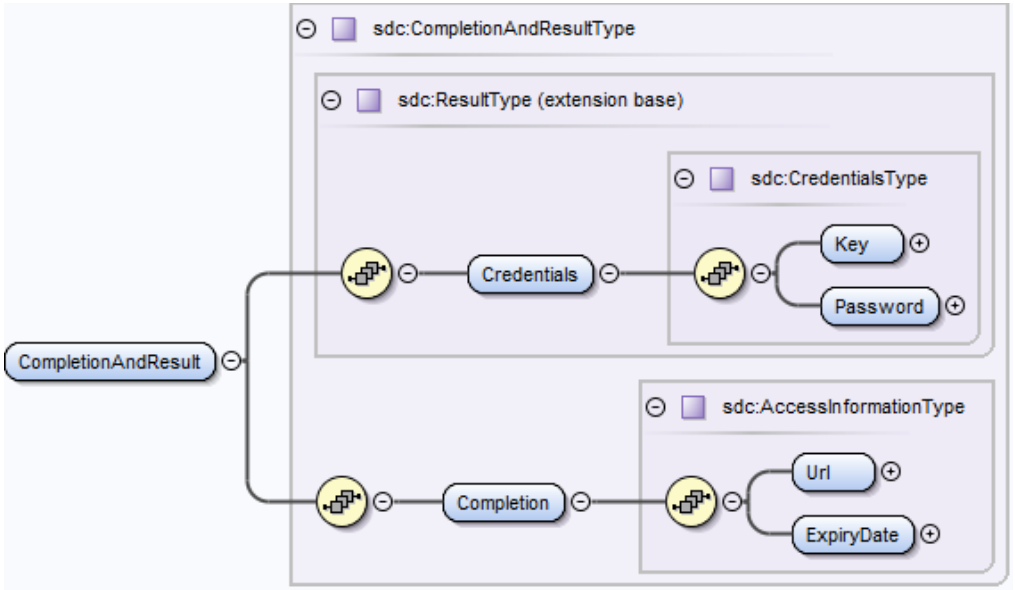
Verfahren

Completion und Freigabe

- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

Completion / Freigabe

Eine GetStatusFromDeclareSalaryResponse einer akzeptierten Lohnmeldungsübertragung beinhaltet ein Completion-Element <CompletionAndResult> mit den Subelementen <Credentials> und <Completion>. Die Angaben in diesen Elementen sind für die Ergänzung und Freigabe der übertragenen Lohndaten relevant. Die Ergänzung und Freigabe der übertragenen Lohndaten erfolgt in einem Webbrowser.



Url:	Adresse des Webservers, auf dem die Ergänzung/Freigabe der übertragenen Lohndaten durchgeführt werden kann. Wichtig: Die URL soll keine Query-Strings beinhalten, die den Werten der Elemente Key und Password entsprechen.
Key:	Ein Schlüssel für die Ergänzung/Freigabe der übertragenen Lohndaten
Password:	Ein Passwort für die Ergänzung/Freigabe der übertragenen Lohndaten
ExpiryDate:	Datum und Uhrzeit, bis wann eine Ergänzung/Freigabe auf dem Webserver möglich ist. In der Regel ist dieser Zeitpunkt **24 Stunden** nach der Lohnmeldungsübertragung erreicht.

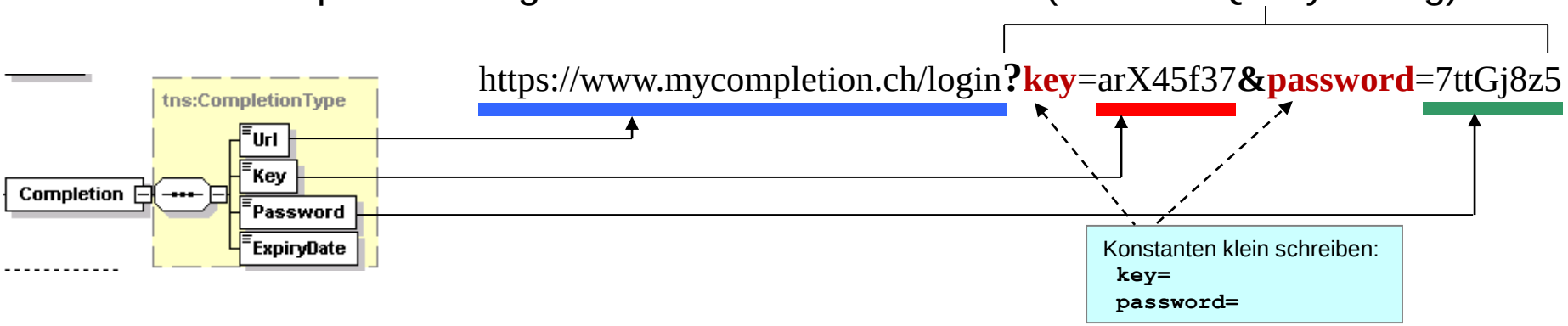
Completion/Freigabe: server-seitige Login-Anforderungen

Um den Kunden ein komfortables, automatisches Login für die Completion/Freigabe zu ermöglichen (d.h. keine manuelle Eingabe von Key und Password erforderlich), sind bei Implementierungen folgende Anforderungen zu realisieren:

Transmitter (client-seitige Anforderung)

Achtung:
Url parsen, damit '**?**' und '**&**' richtig gesetzt werden

Aufruf der Completion/Freigabe mittels erweiterter URL (URL mit Query-String)



Receiver (server-seitige Anforderung)

Verarbeitung des Requests inklusive Query-String und direktes Weiterleiten an Completion/Freigabe (evtl. ohne Login-Seite), wenn Query-String richtige Werte hat.

Verfahren

Resultat und Quittung

- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

Übersicht

- Seit V3 ist es möglich, nach erfolgreicher Übermittlung der BVG-Abrechnung die BVG-Beiträge abzufragen.
- Bei V4 kommt neu die Quellensteuer dazu. Bei integrierten Kantonen können hier Korrekturen und Tarifmitteilungen abgefragt werden.
- Weitere Domänen liefern Quittungen zur Bestätigung der erfolgreichen Übermittlung zurück.
- Erst mit dem Abholen des Resultates ist der Übermittlungsprozess komplett abgeschlossen (Ausnahme: Tax)

Thema Quittung

- Der neuen Prozess basiert auf folgender Analogie:
  Schritt: Ich melde „3+4“ und erhalte dafür die **Quittung:** „empfangen: 3+4“
  Schritt: Ich rufe das Ergebnis ab und erhalte dafür das **Resultat** „=7“

- Im 1. Schritt wird durch den Job mittels des Distributors eine 1:n Verteilung ausgeführt. Der Absender erhält einen Jobkey. Mit diesem Jobkey werden danach mehrere Abfragen (getStatus) auf alle Endempfänger **gleichzeitig** ausgeführt, bis alle ihren Empfang mit Success oder Error quittiert haben (Polling).
- Die Quittungen „spiegeln“ die **SalaryTotals** der gesendeten Meldung

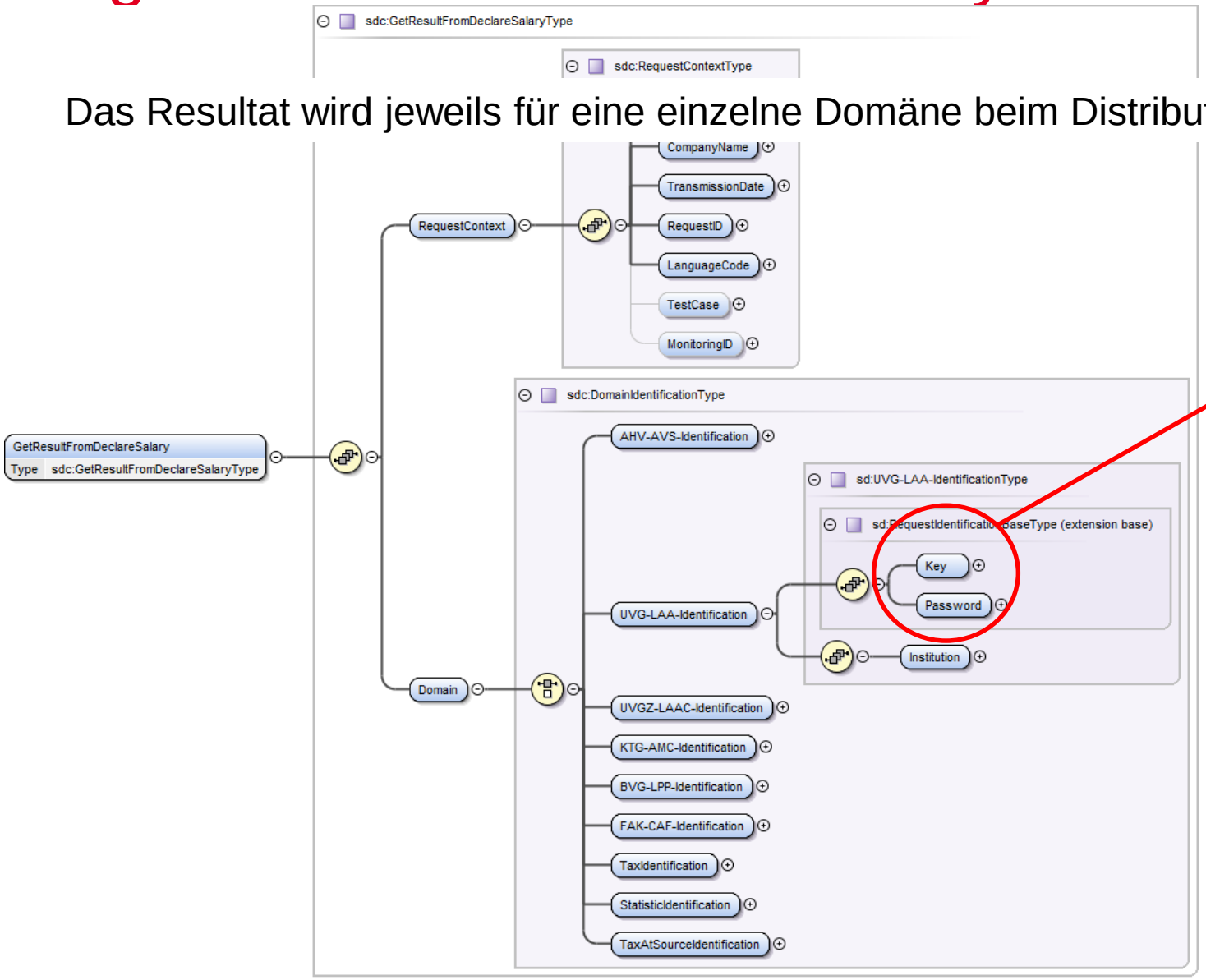
*Achtung: Diese **Quittung** wird auf dem Distributor **gespeichert**!*

- Im 3. Schritt wird über alle Endempfänger ein „Loop“ durchgeführt, bis alle Endempfänger ihre Resultate erbracht haben. Innerhalb dieses „Loops“ werden für jeden **einzelnen Endempfänger** mehrere Abfragen (getResult) durchgeführt, bis ein Success (mit Daten) oder Error abgerufen werden konnte (Polling).

*Aus diesem Grund wird das Resultat auf dem Distributor **nicht gespeichert.***

getResultFromDeclareSalary

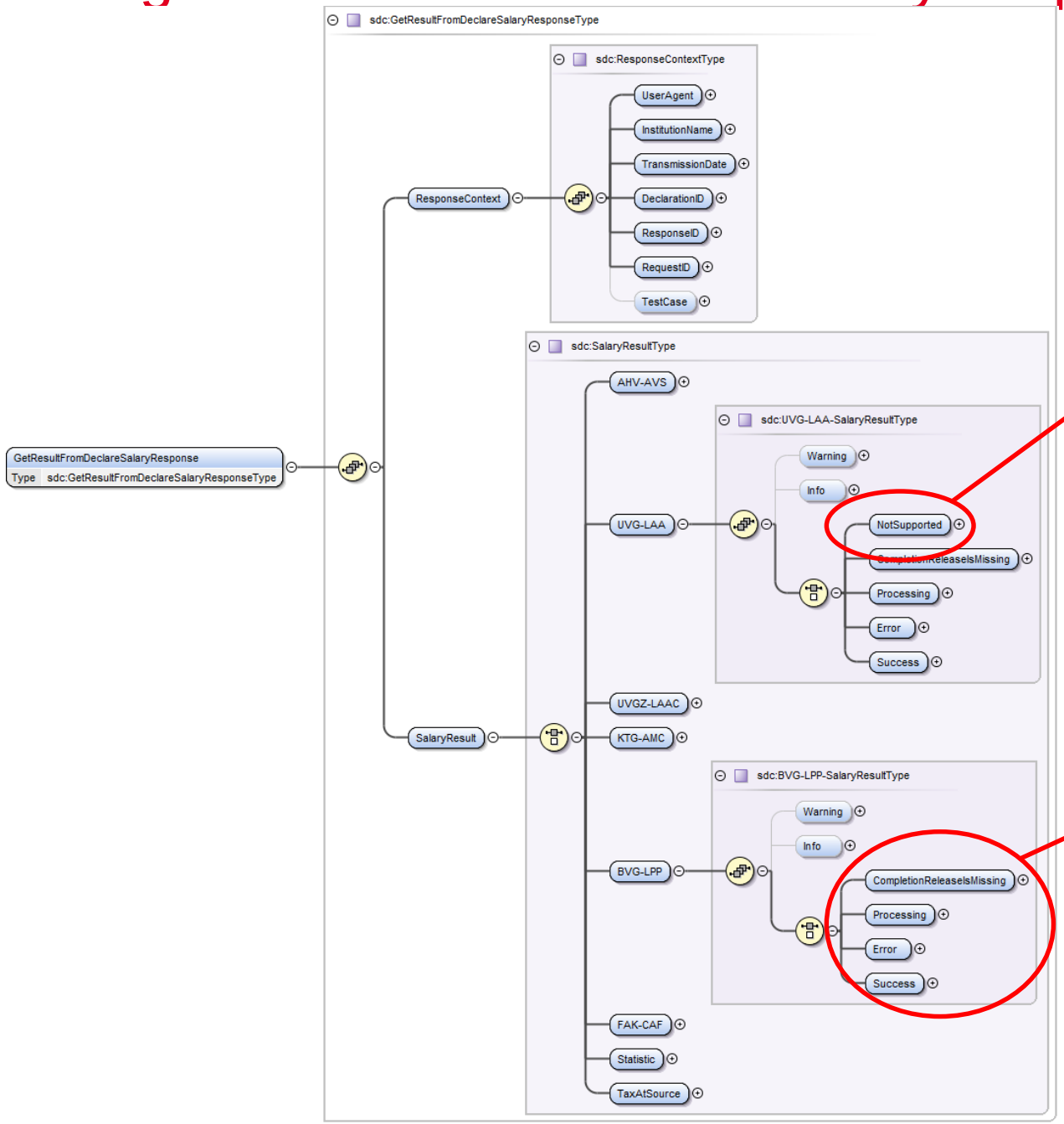
Das Resultat wird jeweils für eine einzelne Domäne beim Distributor abgefragt.



Entsprechend den Credentials aus getStatusFromDeclareSalaryResponse

swissdec

getResultFromDeclareSalaryResponse



NotSupported: Bei AHV, UVG, UVGZ, KTG und FAK

Diese Endempfänger unterstützen teilweise V4 noch nicht.

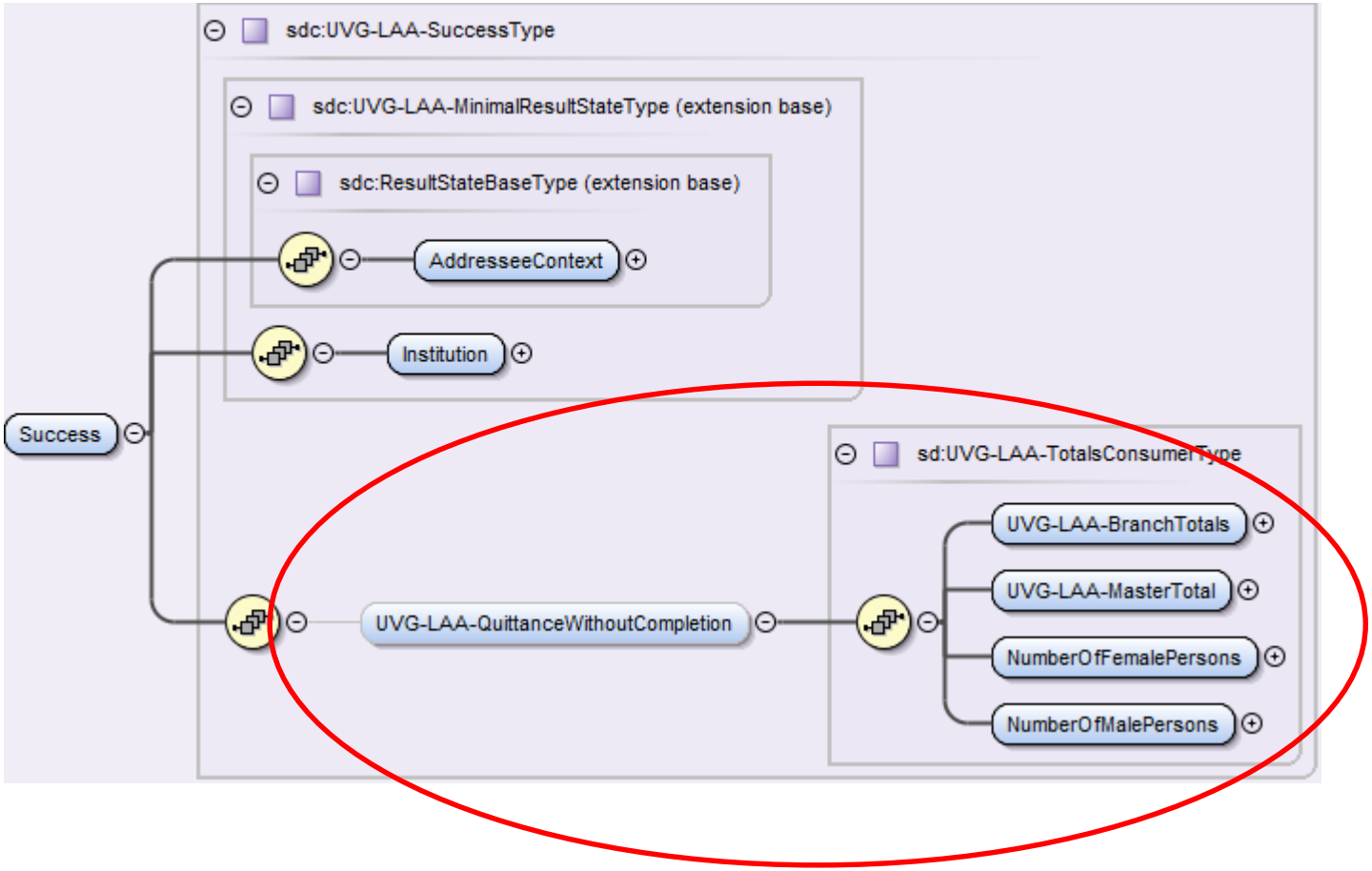
Von sämtlichen Endempfängern unterstützt

swissdec

Resultat: Mögliche Ergebnisse

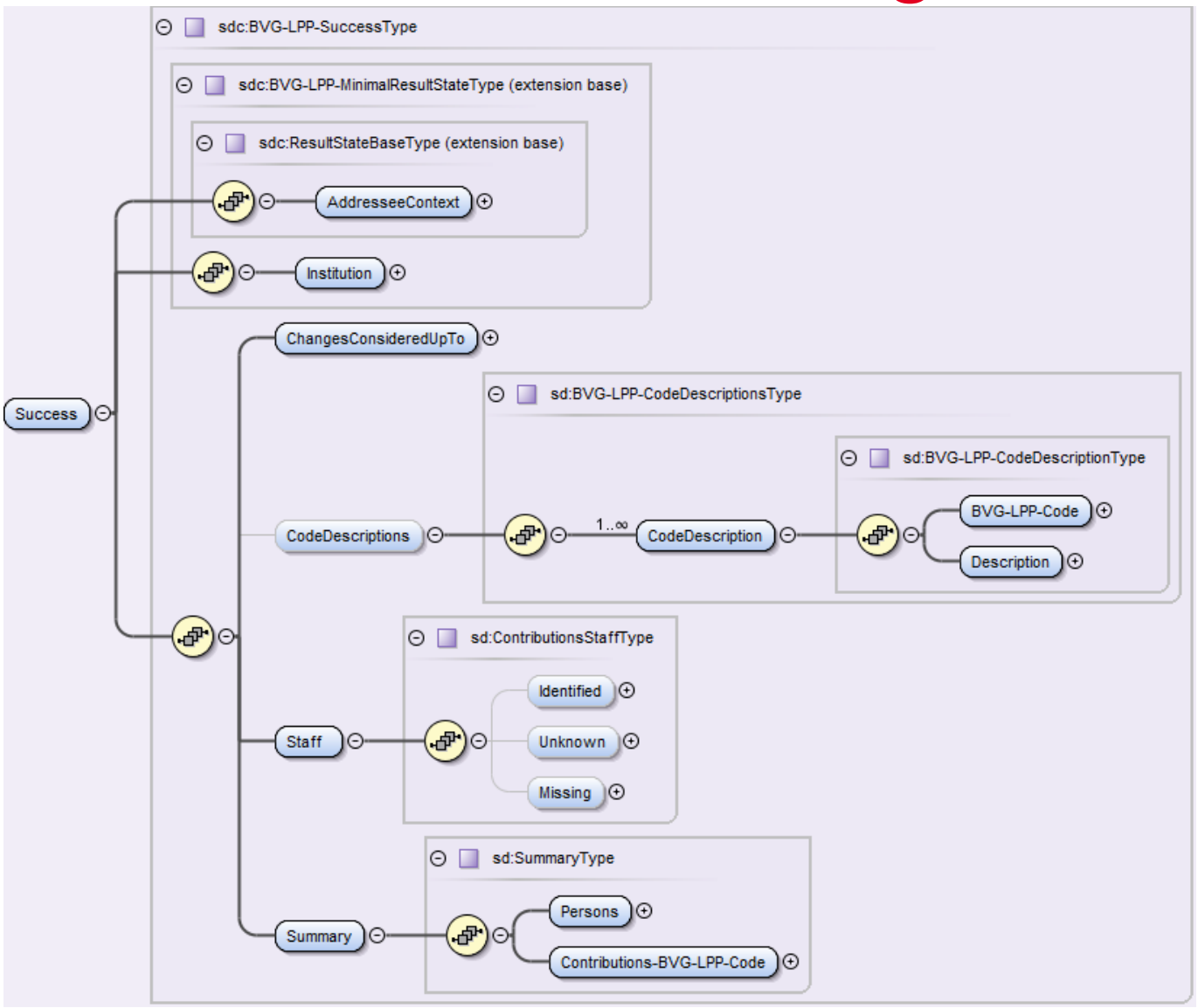
- **CompletionReleaseIsMissing**: Die Meldung wurde nicht freigegeben und konnte daher vom Endempfänger noch nicht verarbeitet werden
- **Processing**: Die Antwort vom Endempfänger ist noch ausstehend
- **Error**: Die Meldung wurde vom Endempfänger zurückgewiesen
- **Success**: Erfolgreiche Übermittlung. Im Success-Element sind Quittung oder Resultate enthalten.

Success: Quittung



swissdec

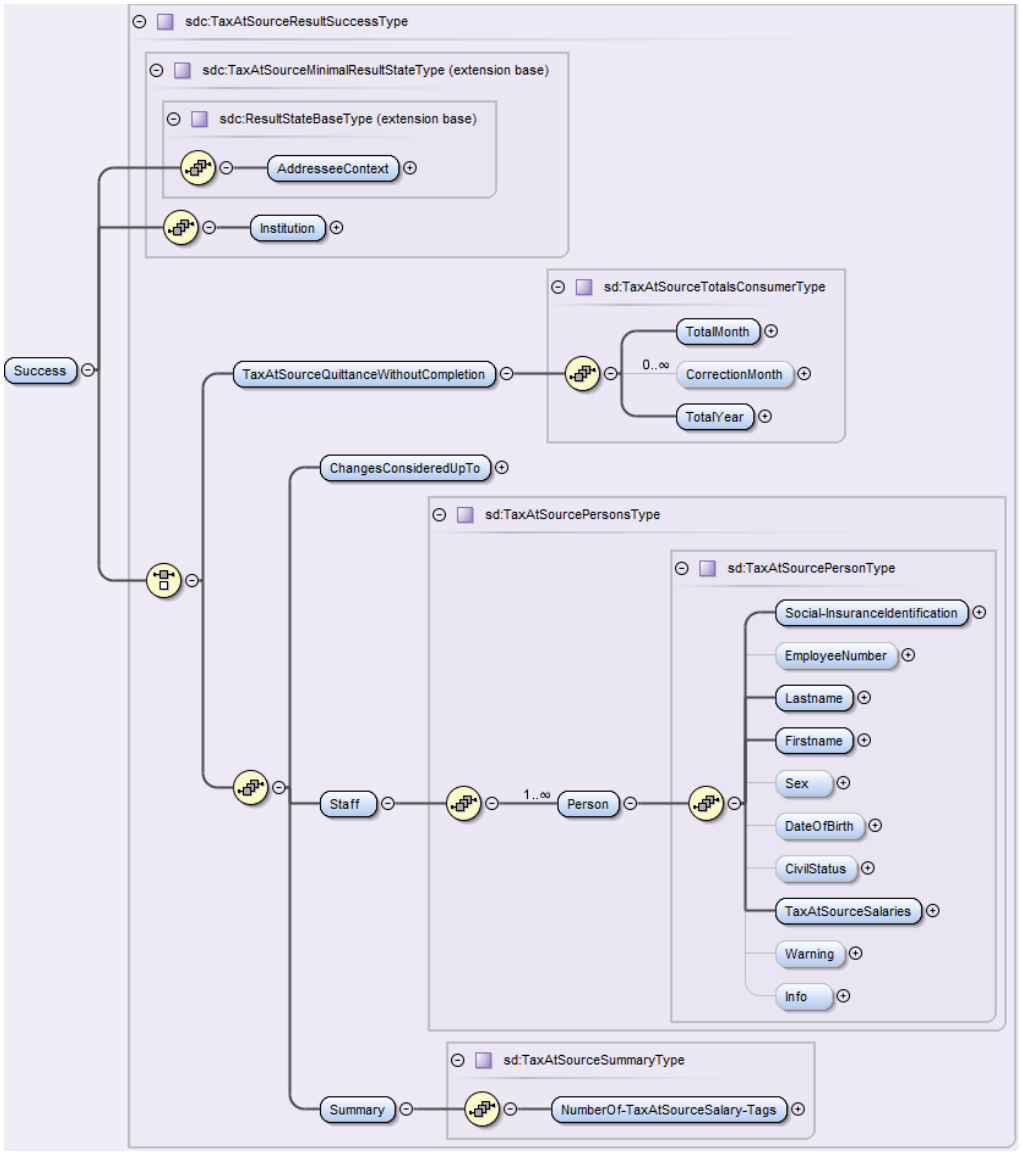
Success: BVG-Beiträge



Detaillierte Informationen zum fachlichen Inhalt der Response finden sich in den RL-LDV.

swissdec

Success: Quellensteuer



Detaillierte Informationen zum fachlichen Inhalt der Response finden sich in den RL-LDV.

swissdec

Ausnahmen:

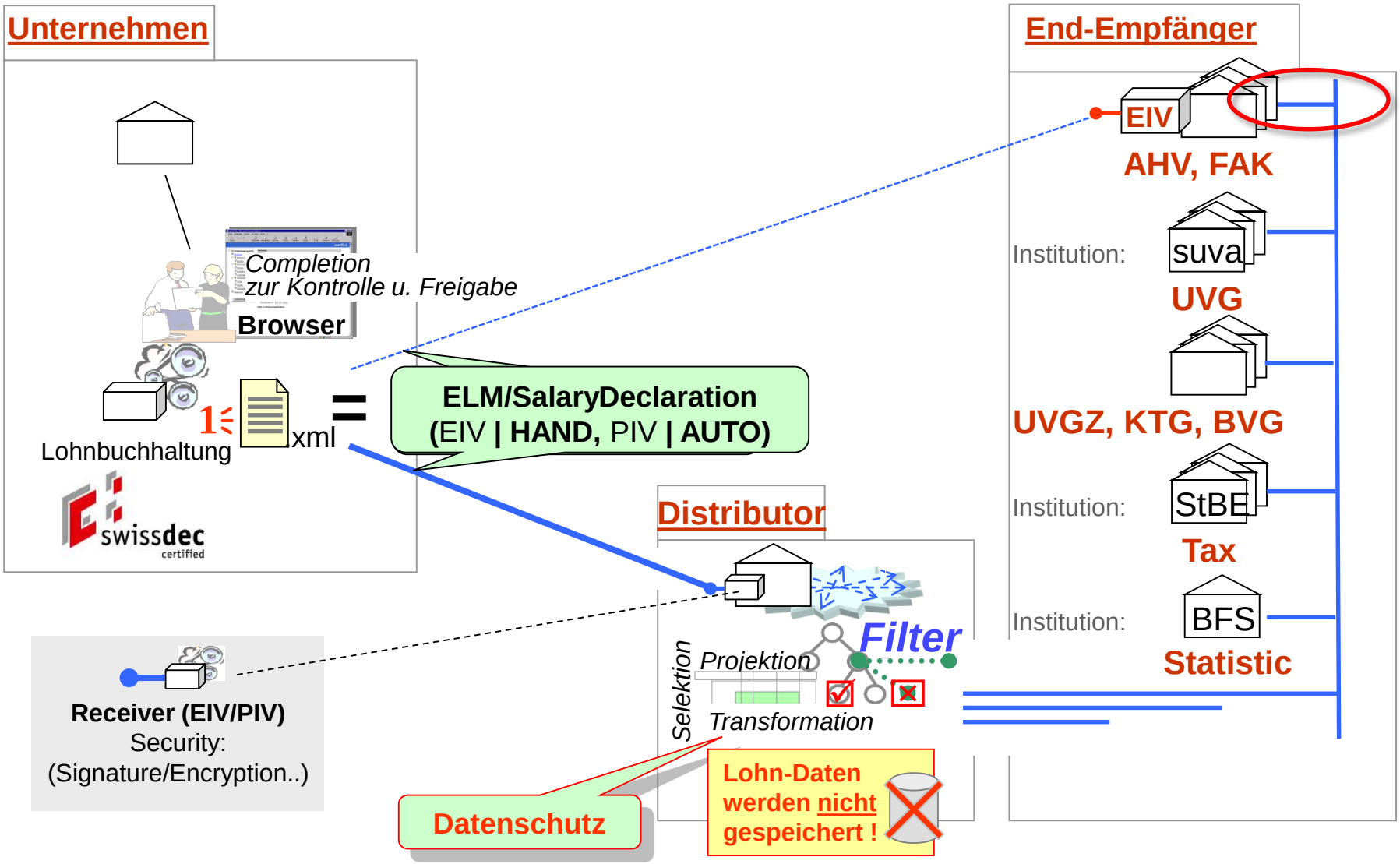
- Die Domäne Tax unterstützt die Abfrage des Resultates nicht und stellt entsprechend auch keine Quittung zur Verfügung
- Endempfänger, welche V3 unterstützen, können noch keine Rückmeldung liefern. Bei diesen wird als Antwort <NotSupported> zurückgegeben.

Verfahren

Distributor

- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

Gesamtsicht



swissdec

Verfahren

Vorabgleich

- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

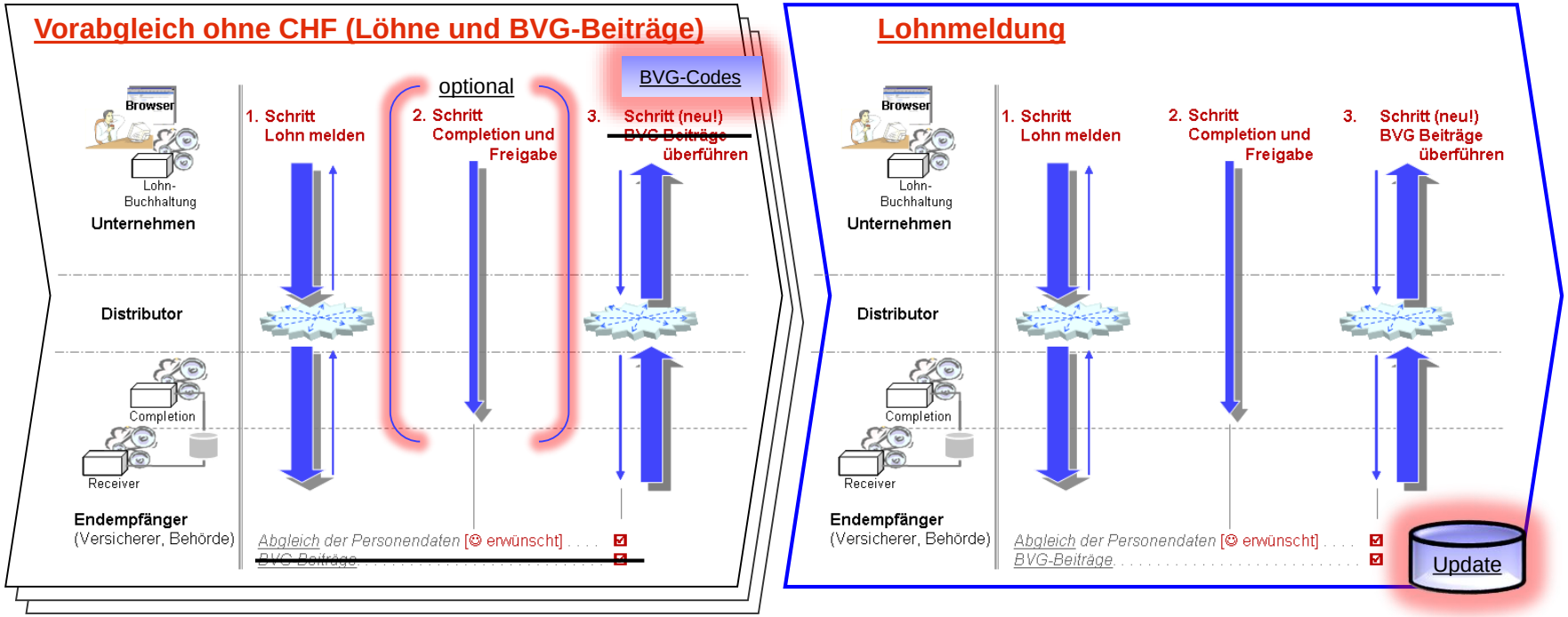
Thema: Vorabgleich

Ziele:

- Datenabgleich zum Ausweisen von *Differenzen* (z.B. Eintritt, Austritt, Code-Wechsel, usw.), welche zu Mutationen beim Versicherer oder Korrekturen in der Lohnbuchhaltung führen.
- *Initialisierung* BVG-Code

Ziele:

- „*Gesamtmutation*“ der **Löhne** (Lohn-Basis)
- Mutation **Beschäftigungsgrad**
- Mutation **BVG-Code**



swissdec

Probleme und Ausgangslage

- BVG-LPP-Code Mutation
  - Code-Mutationen nicht nur zum Jahresbeginn
  - Es gibt **einzelne, unterjährige Mutationen** (z.B. infolge Beförderung oder Zivilstandsänderung, Zusammenlegung einzelner Betriebsteile bzw. Versichertengruppen etc.)
- Welche **Aktionen** sollen aus den Differenzen im Vorabgleich (3. Schritt BVG-Personen abrufen) beim Kunden / Versicherten erfolgen?

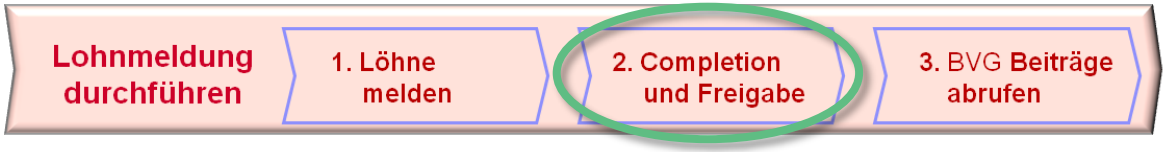
Lohnbuchhaltungs-Hersteller Vorschlag:
Wir erstellen aus diesen Differenzen „flugs“ Eintritts-, Mutations- und Austritts-Formulare für unser Kunden.

➔ grosser Kundennutzen: ☺ , aber Medienbruch in der Automatisierung : 💥 ☹

- ELM verlangt eine **synchrone Datenführung** zwischen Unternehmen und Versicherer, d.h. beide benötigen einfache und mächtige Werkzeuge und nicht alte Papierformulare!
- **Initialisierung** von ***E**intritt **M**utation* und ***A**ustritt* „EMA“
  - Thema: „swissdec / Mutations-Standard-CH“

Aktuelle Lösungen beim Endempfänger

- Im 2. Schritt bzw. der Completion wird ein Abgleich zwischen Meldungsdaten und Stammdaten des Versicherers erstellt und entsprechende **Differenzen** dem Benutzer angezeigt. Der Benutzer kann mittels **„Portal-Links“** direkt *Eintritt*, *Mutation* oder *Austritt* von Personen in einer Webapplication durchführen.



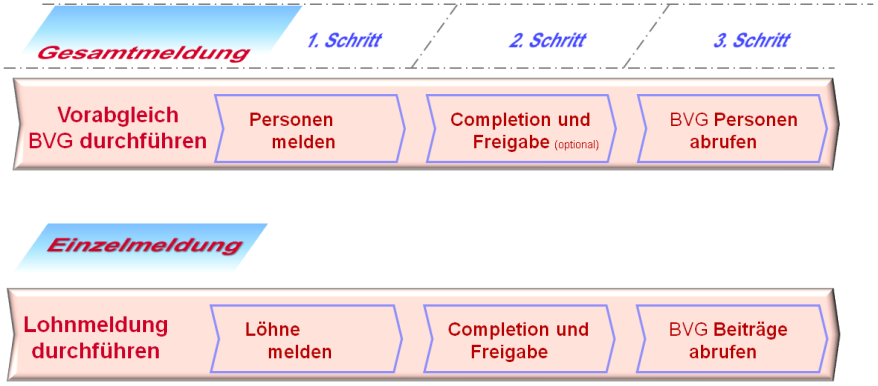
Probleme

- Die gemeldeten Lohndaten müssen an die bestehenden Portal-Masken weitergeben werden, damit keine zusätzliche manuelle Erfassung (Name, Vorname,…) nötig wird.
- Im **Vorabgleich** wird ein **Basis-Lohn mit „0.00“ CHF** gemeldet. In den Erfassungsmasken der Webapplications muss der Betrag deshalb **manuell** erfasst werden.
- Die neuen **Beiträge** bei *Eintritt* und *Mutation* lassen sich im 3. Schritt bzw. bei dem *Vorabgleichssprozess* „BVG Personen abrufen“ **nicht automatisch** in die Lohnbuchhaltung überführen, da keine Beiträge zurückgemeldet werden dürfen. Nur im *Lohnmeldungsprozess* „BVG Beiträge abrufen“ können die Beiträge **automatisch** in die Lohnbuchhaltung überführt werden.
- Im eigentlichen Lohnmeldungsprozess wird eine Abarbeitung von Eintritten, Mutationen und Austritten als **störend** empfunden!
  - Ziel ist der Abschluss und nicht einzelne Mutationen ☹
  - In diesem Moment fehlen mir die Unterlagen, Nachfragen beim Mitarbeiter, … ☹

Vom Jahresende zu den Prozessen innerhalb des Jahres

- **Regelmässig** (12x) wird ein **Vorabgleich** durchgeführt
  :

  Danach führen Differenzen zur **Initialisierung*** von Korrektur, Eintritt, Mutation oder Austritt
  (Direkte Initialisierung von „EMA“, ist auch **ohne vorgängigen Vorabgleich** möglich, d.h. „proaktiv“ melden!)*



- **Jahresende** oder **Spezialfall** (1x) wird mit einer **Lohnmeldung** durchgeführt
  :

  Danach führen Differenzen zu zur Initialisierung von Korrektur, Eintritt, Mutation oder Austritt

Verfahren

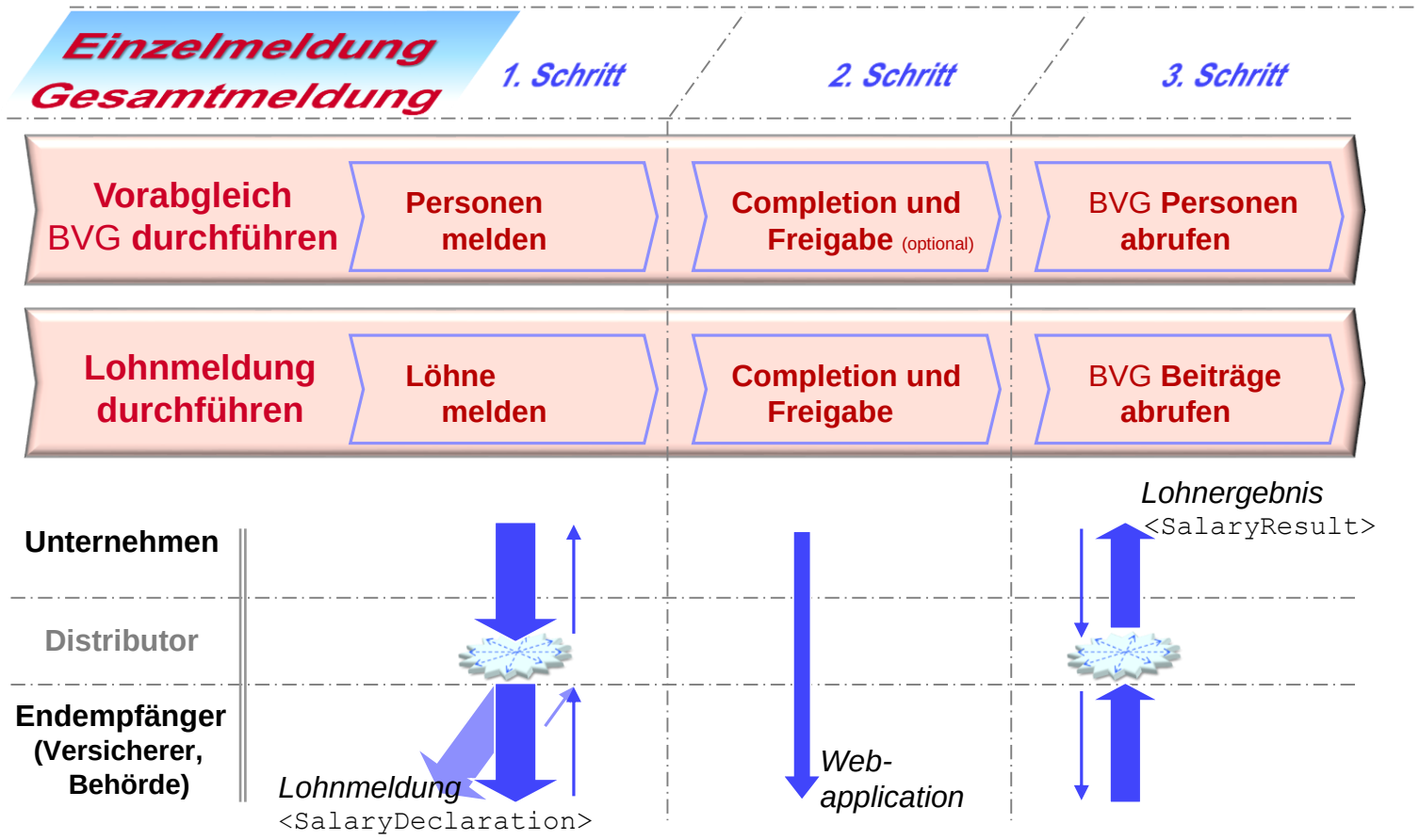
«EMA»

- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

Rahmenbedingungen

- Die **Implementierung** von „EMA“ (Eintritt, Mutation, Austritt) mittels *Einzelmeldung* ist bei den Lohnbuchhaltungsherstellern und Endempfängern **optional**.
- **Papierformulare** werden für „EMA“ **nicht spezifiziert**. (Kundennutzen versus swissdec Internet-Lösungen)

Lohnstandard-CH (ELM)
neu in der 3. Dimension



ELM (Einheitliches Lohnmeldeverfahren)

swissdec

Begriffe und ihre Einordnungen (Taxonomie)

Operationen	Parameter	
Vorabgleich (VA)	*„Gesamt“- Vorabgleich*	*„Einzel“ - Vorabgleich*
Lohnmeldung (LM)	*„Gesamt“ - Lohnmeldung*	*„Einzel“ - Lohnmeldung*
	Gesamtmeldung (GM)	**Einzelmeldung (EM)**

Essentielle Begriffe
(bestehende und neue Beschreibungen)

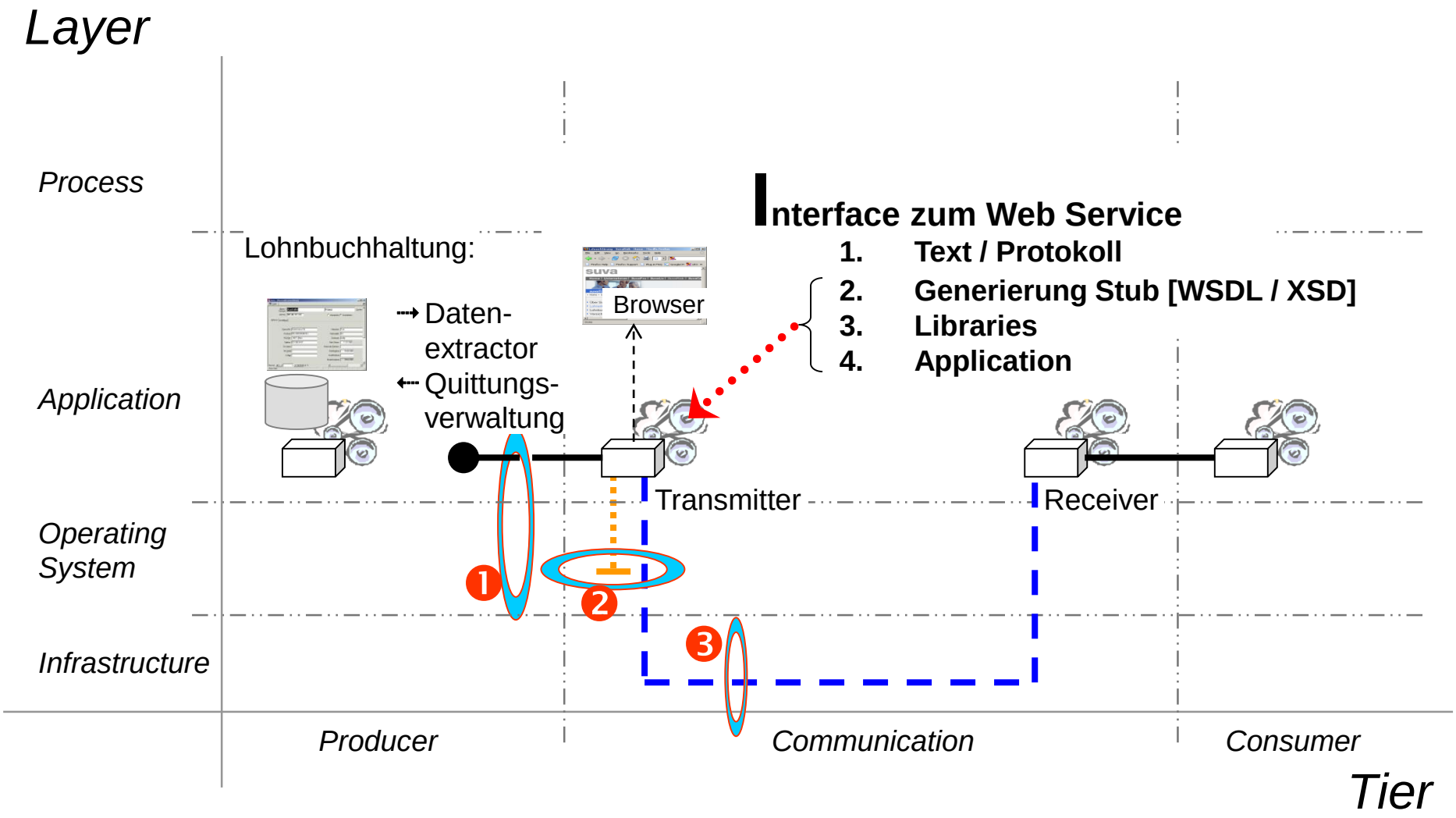
- **Lohnmeldung (LM)**
  - **Prozess** zur Meldung von Lohn- und Personen-Daten an den Versicherer.
- **Vorabgleich (VA)**
  - **Prozess** zur Synchronisierung der Daten zwischen Unternehmen und Versicherer.

- **Gesamtmeldung (GM)** (bisher in V3 nur implizit vorhanden)
  - **Alle neuen Löhne** einer Unternehmung (Übermittlungseinheit pro Buchungskreis) auf einen gemeinsamen Zeitpunkt melden.
  - Im Completion werden durch **eine** Freigabe der Meldung **alle Personen** verarbeitet.
  - **Inhalt** einer *Lohnmeldung* oder eines *Vorabgleichs*
- **Einzelmeldung (EM)**
  - **Einzelne Lohn- und Personen-Daten** an Versicherer zu beliebigem Zeitpunkt melden (z.B. Eintritt, Mutation und Austritt / „EMA“)
  - **Jede Person** muss in der Completion **einzeln** abgearbeitet und freigegeben werden.
  - **Inhalt** einer *Lohnmeldung* oder eines *Vorabgleichs*

Architektur und Installation

- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

Architektur-Skizze Transmitter



swissdec

Architektur-Skizze (II)

- «Datenstrukturen UND Algorithmen» sind ein festes Paar in der Informatik
- XML nicht nur zur Datenstrukturierung, sondern auch als Basis zur **Generierung** von **Stub|Proxy-Operationscode** (wsdl2Java, wsdl2C#, …) in der individuellen Umgebung der SW-Hersteller.
- Die **optimalste Integration** in bestehende Lohnbuchhaltungen (Prozess, Applikation und Infrastruktur)
- Nutzung der **Tools** im Bereich der Sicherheit (Signatur, Verschlüsselung) der einzelnen IT-Lieferanten (W3C, OASIS, …;Sun, Microsoft, Open Source, …)
- Die **Interoperabilität** wird durch die Generierung von Codes gewährleistet;
  d.h. keine Portabilitätsprobleme bei Libraries oder Applikationen in den Unternehmen
  Siehe Architektur-Skizze mit den drei Schnittstellen:
  1. Applikation – Transmitter (File, Pipes, Interprocess, …)
  2. Transmitter läuft auf unterschiedlichen Betriebssystemen und Versionen
  3. Kommunikation ins Internet über Proxies, Firewalls, … sicherstellen
- Eine gute **Skalierbarkeit** in der Unterstützung vom wsdl bis zu eigenen Applikationen (Tools, Libraries, swissdec Transmitter-Applikation, …) wird erst dadurch ermöglicht.
- Basiert auf **internationalen Standards** (W3C, OASIS, …)
  d.h. verbreitetes Know-how mit vielen Tools ist vorhanden und es entsteht dadurch ein Lösungsmarkt (Kosten, Innovation)
- Eine Wiederverwendung in anderem Kontext ist möglich (Know-how bis Code)

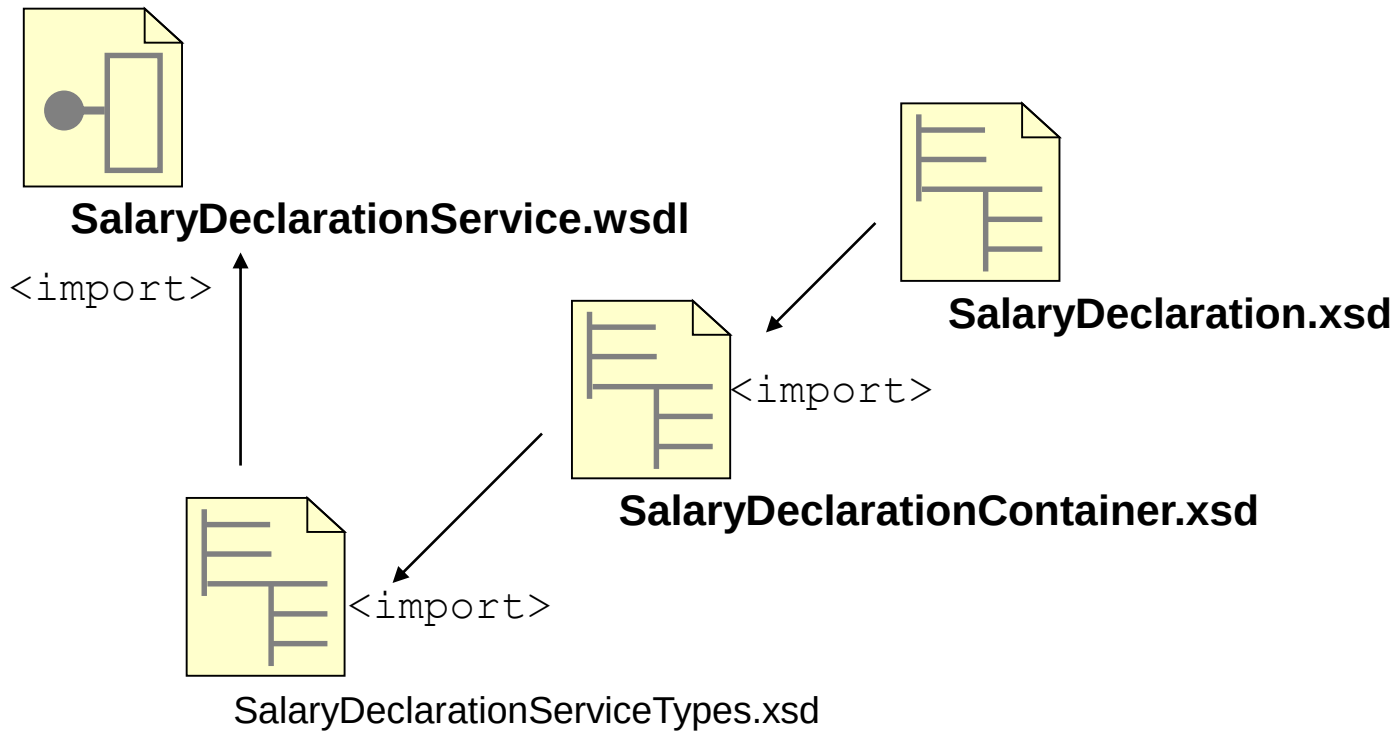
«XML» Basis Versionen

- XML: Version 1.0
- XML-Schema: 2001
  http://www.w3.org/2001/XMLSchema
- WSDL: Version 1.1
  http://schemas.xmlsoap.org/wsdl/
- SOAP: Version 1.1
  http://schemas.xmlsoap.org/soap/envelope/
- WSSecurity: Version 1.0
  http://http://www.oasis-open.org

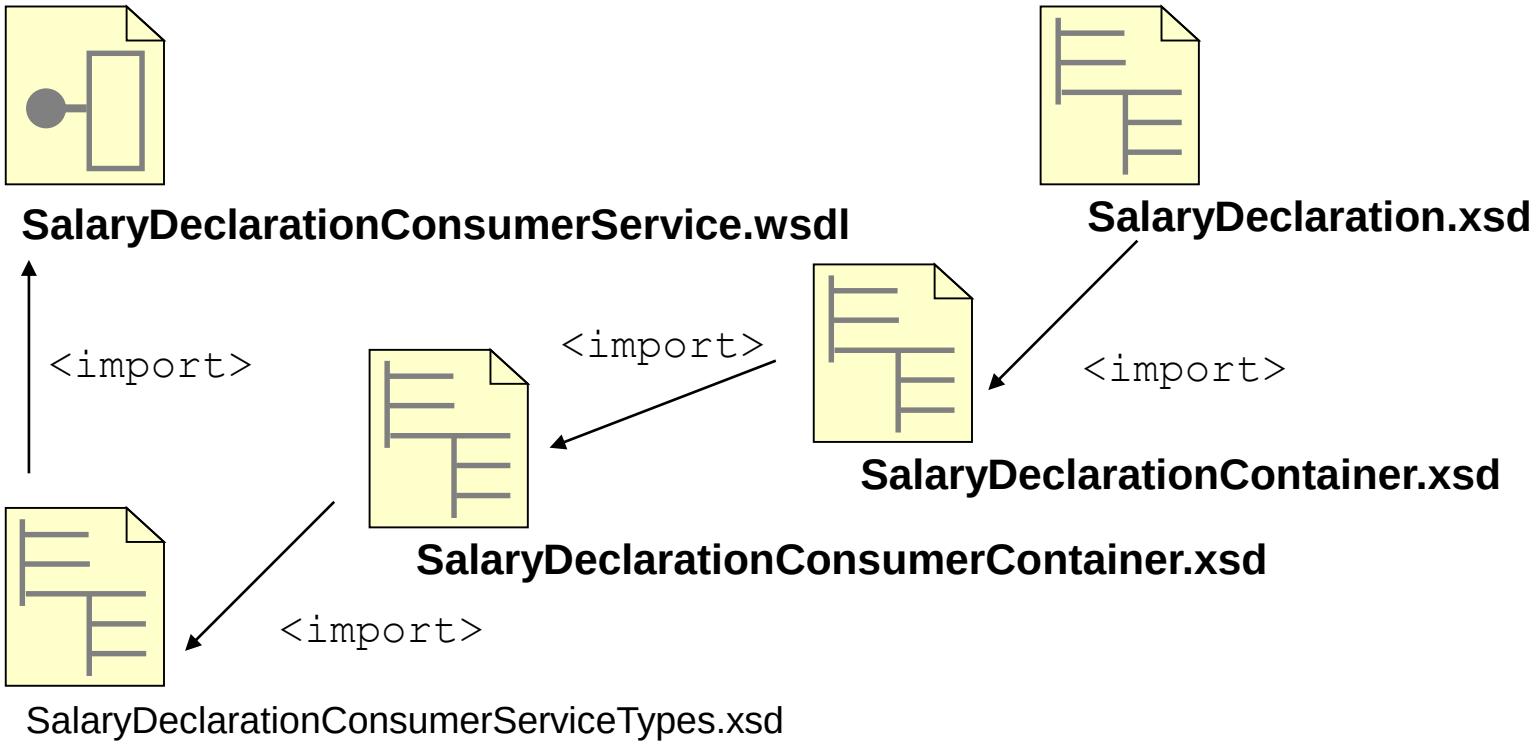
```
        http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd
        http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd
```

Diese Standards gelten für PIV und EIV

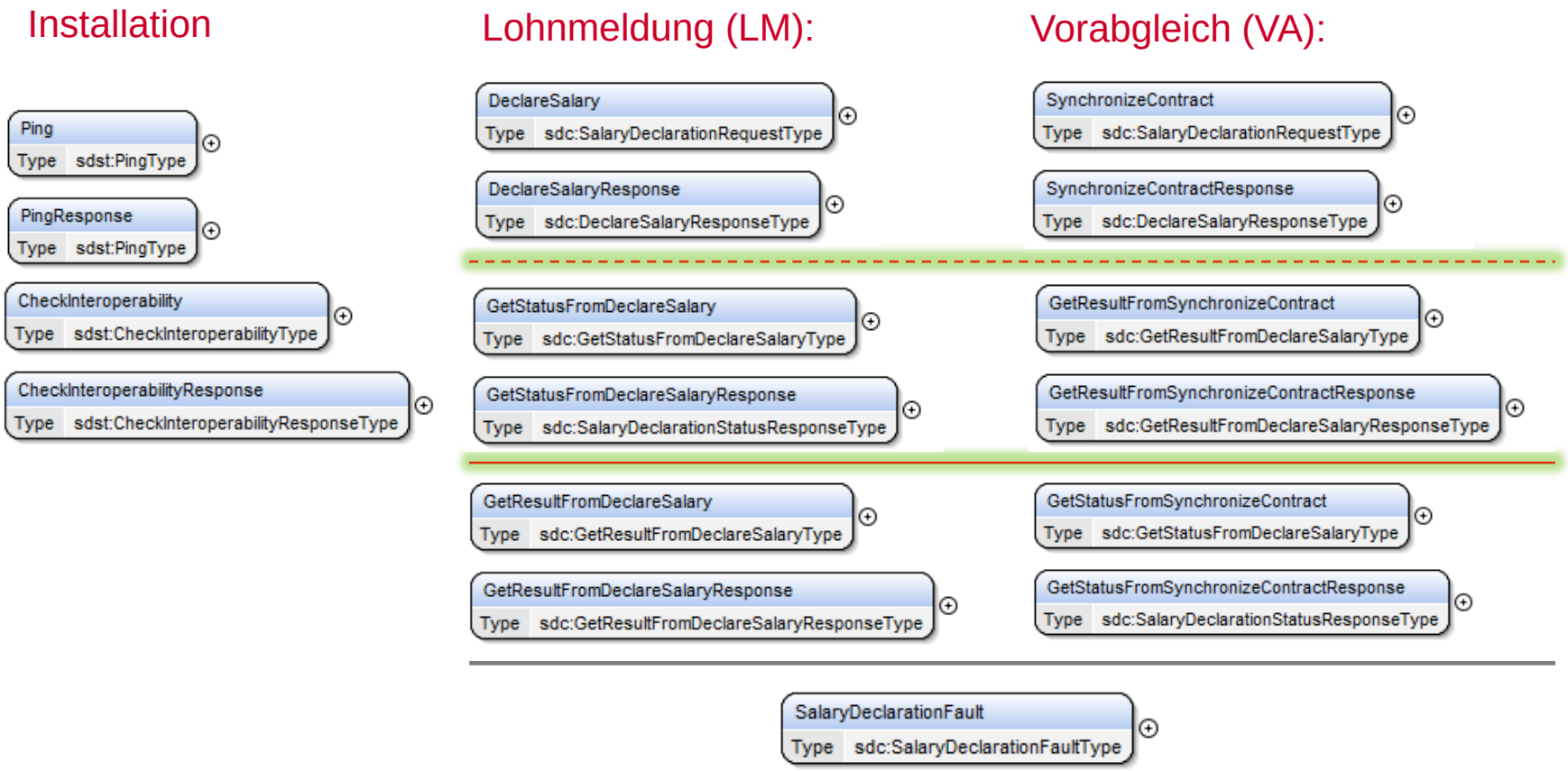
Hierarchie der Schemas



Hierarchie der Schemas



SalaryDeclarationService.wsdl



swissdec

ELM/ SalaryDeclaration V4 Schemas

Prefix	*Namespace (incl. Version . . . /sd/YYYYMMDD/ . . .)*	*Filename*
sdst	http://www.swissdec.ch/schema/sd /20130514 /SalaryDeclarationServiceTypes	SalaryDeclaration ServiceTypes .xsd
sdc	http://www.swissdec.ch/schema/sd /20130514 /SalaryDeclarationContainer	SalaryDeclaration Container .xsd
sd	http://www.swissdec.ch/schema/sd /20130514 /SalaryDeclaration	SalaryDeclaration .xsd
	http://www.swissdec.ch/schema/sd /20130514 /SalaryDeclarationService	SalaryDeclaration Service.wsdl

Namespace prefix

swissdec Namespace (… = http://www.swissdec.ch/schema)

Transmitter SalaryDeclarationService:
xmlns:**sds** ="…/sd/20130514/**S**alary**D**eclaration**S**ervice"(wsdl)
xmlns:**sdst** ="…/sd/20130514/**S**alary**D**eclaration**S**ervice**T**ypes"

xmlns:**sdc** ="…/sd/20130514/**S**alary**D**eclaration**C**ontainer"
xmlns:**sd** ="…/sd/20130514/**S**alary**D**eclaration"

Endreceiver SalaryDeclarationConsumerService:
xmlns:**sdcs** ="…/sd/20130514/**S**alary**D**eclaration**C**onsumer**S**ervice"(wsdl)
xmlns:**sdcst** ="…/sd/20130514/**S**alary**D**eclaration**C**onsumer**S**ervice**T**ypes"
xmlns:**sdcc** ="…/sd/20130514/**S**alary**D**eclaration**C**onsumer**C**ontainer„

TaxAccounting Barcode (TxAB):
xmlns:**sdtab** ="…/sd/20130514/**S**alary**D**eclaration**T**x**AB**"

Major / Minor Version

Major Version Number

- Grundlegende Änderungen der Lohnstandard XML Schemas bedingen eine Inkrementierung der Major Version Number. Die Major Version Number widerspiegelt das Erstellungsdatum (YYYYMMDD) des XML Schemas und ist in dessen Ziel-Namensraum (Targetnamespace) enthalten.
- *Schemas mit unterschiedlicher Major- Version-Number sind inkompatibel!*

Beispiel:

```
<xsd:schema
 targetNamespace="http://www.swissdec.ch/.../20130514/...
```

Minor Version Number

- Alle Lohnstandard XML Schemas enthalten das Attribut "version" im Element "schema" (Root Element). XML Schema Instanz Dokumente müssen zwingend mit dem Attribut "schemaVersion" (ebenfalls auf dem Root Element, Konvention, wird erzwungen) angeben, auf welche Minor Version Number sie sich beziehen.
- Schemas mit gleicher Major- und unterschiedlicher Minor- Version-Number sind rückwärtskompatibel!

Major / Minor Version (II)

- Major Versionen haben folgende Files:
  - **SalaryDeclaration**.xsd
    `http://www.swissdec.ch/schema/sd/20130514/SalaryDeclaration`
  - SalaryDeclaration**Container**.xsd
    `http://www.swissdec.ch/schema/sd/20130514/SalaryDeclarationContainer`
  - SalaryDeclaration**ServiceTypes**.xsd
    `http://www.swissdec.ch/schema/sd/20130514/SalaryDeclarationServiceTypes`
  - SalaryDeclaration**Service**.wsdl
    `http://www.swissdec.ch/schema/sd/20130514/SalaryDeclarationService`

- Minor Version wird nur im **SalaryDeclaration.xsd** geführt !

Sicherheit

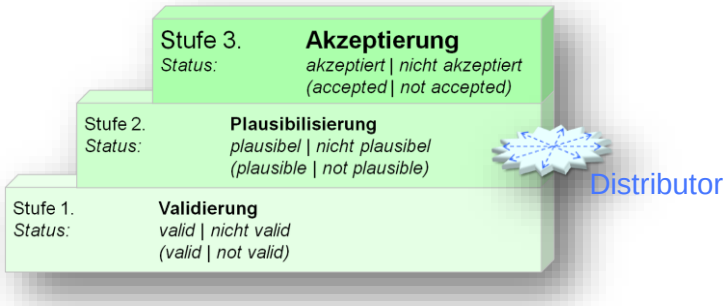
- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

Prozessqualität, Sicherheit und Datenschutz

*Das **Vertrauen** aller Teilnehmer in den gesamten Geschäftsprozess ist zwingend!*

Folgende Massnahmen unterstützen dies (nicht abschliessend):

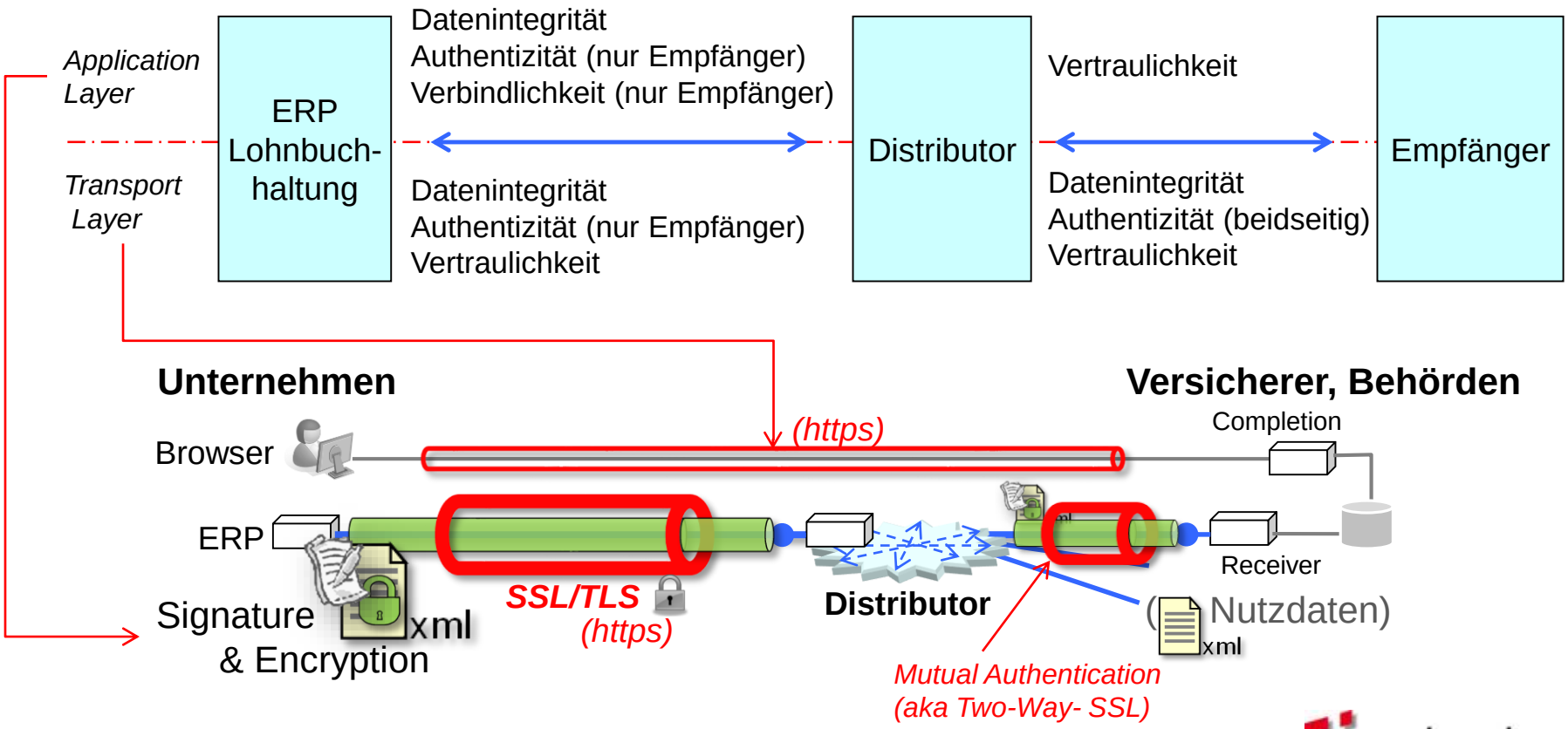
- Neben dem sicheren Transport über https (SSL/TLS) werden die Meldungen zusätzlich **signiert** und ein zweites Mal **verschlüsselt**.
- Kontrolldaten und **3-stufige Qualität** in der Übermittlung (dynamische QS)



- **Datenschutz** mittels Filterung auf dem **Distributor**, damit nur die zur Verarbeitung notwendigen Daten an die Versicherer oder Behörden gelangen (Rechtmässigkeit und Verhältnismässigkeit **ohne** eine Speicherung auf dem Distributor).
- **Zertifizierung** der Software-Lösungen werden wiederkehrend durchgeführt (langfristige QS)
- Prozess-Sicherheit durch weitere separate Schritte wie die Kontrolle in der Rechnungsstellung oder zusätzliche Regeln in der Verarbeitungslogik.

https und WS Signature & Encryption

- Schutzzielabdeckung
  - In Zukunft könnte zwischen dem ERP / Lohnbuchhaltung und dem Distributor die Authentizität und Verbindlichkeit **beidseitig** werden.



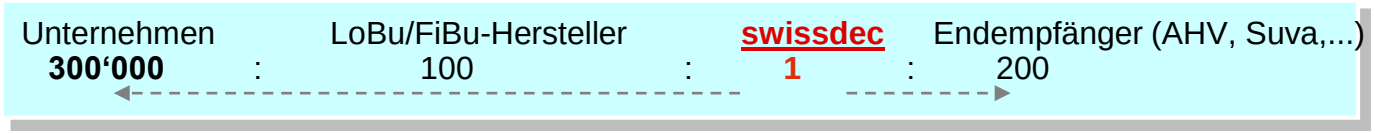
Essenz

- Grundlagen
- Deklaration
- Verfahren
- Architektur und Installation
- Sicherheit
- Essenz

«Essenz»

- Definitionen mittels **«XML-Family»** Plattformunabhängiger W3C-Standard
  → Textuelle Beschreibung, XML zur Generierung von Code & Libs
- Web Service Standard-Bausteine von Java, Microsoft, …verwenden
  → maximale Flexibilität und maximale Integrierbarkeit in Lohnbuchhaltung
- Zusätzlich Prozessintegriertes Verfahren (PIV|auto) zur optimalen Automatisierung
  → vom Formular zum Dialog (z.B. BVG-Beiträge)
- Muster: «m2m\h2m» (m2m= statische + grosse Mengen und h2m= dynamische Daten)
  → «dynamische Multichannel» Lösung

  h2m *(Human to Machine)*
  m2m *(Machine to Machine)*
- Daten, Verfahren und Distributor
  → vereinfachter Workflow (Prozessredundanz), Datenschutz durch Filterung und Verteilung
- Sicherheit mit Signatur (zum SSL-Tunnel zusätzlich optional eine SOAP-Verschlüsselung) und direkter Integration in Lohnbuchhaltung
- Daten mit separaten Sichten (Views: html, pdf, … Formulare, 2DBarcode)
  Sichten sind auch standardisierbar (XSL und XSL-FO / z.B. Lohnausweis)
  → vereinfachte Entwicklung von Formularen und Auswertungen
- EIN Instanzdokument für sämtliche Empfänger und Verfahren (PIV|EIV)
- Methodik: a) Schnittstelle und zusätzlich auch Quellsystem; b) «Fach&Tech» Team in der Suva
- swissdec hat einzigartige Koordinations- und Umsetzungsmöglichkeiten:



Referenzen

ID	Referenzname	Autor	Jahr
[TREQ]	TransmitterRequirements.pdf, enthalten in RL-LDÜ	swissdec	2017
[SALDXSD]	SalaryDeclaration.xsd, enthalten in RL-LDÜ	swissdec	2013
[CONTXSD]	SalaryDeclarationContainer.xsd, enthalten in RL-LDÜ	swissdec	2013
[SALDTYPES]	SalaryDeclarationServiceTypes.xsd, enthalten in RL-LDÜ	swissdec	2013
[SALDWSDL]	SalaryDeclarationService.wsdl, enthalten in RL-LDÜ	swissdec	2013
[ACKNOTIF]	AcknowledgementNotification.pdf https://tst.itserve.ch/swissdec/infopoint/datapool.xhtml	swissdec	Aktuelle Version
[SECPDF]	Security.pdf https://tst.itserve.ch/swissdec/infopoint/datapool.xhtml (Separate Dokumente für Transmitter und Endempfänger)	swissdec	Aktuelle Version