Technische Spezifikation KLE 2.0.1

Version 20260306

Anforderungen Receiver

Swissdec
Postfach, 6004 Luzern
www.swissdec.ch

Diese Richtlinien wurden in Zusammenarbeit mit den Mitgliedern des Vereins Swissdec erarbeitet:

- Suva
- Verein eAHV / IV mit den Mitgliedern:
  - Konferenz der kantonalen Ausgleichskassen
  - Vereinigung der Verbandsausgleichskassen
- Schweizerische Steuerkonferenz mit den Mitgliedern:
  - Eidgenössische Steuerverwaltung
  - Kantonale Steuerverwaltungen
- Bundesamt für Statistik
- Schweizerischer Versicherungsverband

Herausgeber
Swissdec
Fluhmattstrasse 1
6004 Luzern

https://www.swissdec.ch

© Swissdec 2026

Versionsgeschichte

Version 2.0.1	25.11.2026	ile, mas
Republikation, KLE Version 2.0.1		
Version 2.0	06.03.2026	ile, mas
Erste Ausgabe Spezifikation für KLE 2.0.		

© Swissdec 2026

Konventionen in diesem Dokument

Folgende Schriftarten werden in diesem Dokument verwendet:

Text	Dokumentation
`Text`	Code
`<Text>`	XML-Element
[Text]	Referenz auf ein anderes Dokument

Die Verbindlichkeit von Anforderungen ist wie folgt definiert:

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

Tabelle 1. Verbindlichkeit von Anforderungen

Achtung

Für das konzeptionelle Verständnis werden teilweise schematische Darstellungen verwendet, die nicht zwingend den aktuellen Stand widerspiegeln.

Verbindlich sind ausschliesslich die jeweils veröffentlichten offiziellen XML-Schemata.

Achtung

Bei Abweichungen zwischen der deutschen, französischen und italienischen Ausgabe der Richtlinien ist die deutsche Version massgebend.

© Swissdec 2026

Inhaltsverzeichnis

1. Einleitung 1
1.1. Aufbau des Dokuments 1
1.2. Leitgedanke Bausteine 2
2. Geschäftsprozess zur Übermittlung von Leistungsfällen mittels KLE 3
3. Use Cases und zugehörige Operationen 5
3.1. Übersicht 5
3.2. Grafische Darstellung der Geschäftsprozesse und ihren Use Cases 5
3.3. Uses Cases und ihre Operationen 7
4. Ping 8
5. RegisterOrganizationAuthentication 12
5.1. Übersicht 12
5.2. Initiale synchrone Meldung 12
5.3. Datenstruktur der initialen Meldung 14
5.4. Datenstruktur der initialen Antwort 15
5.5. Datenstruktur der Synchronize-Meldung 16
5.6. Datenstruktur der Synchronize-Antwort 16
6. SubscribeOrganization 17
6.1. Übersicht 17
6.2. Initiale synchrone Meldung 17
6.3. Datenstruktur der initialen Meldung 19
6.4. Datenstruktur der initialen Antwort 20
6.5. Datenstruktur der Synchronize-Meldung 21
6.6. Datenstruktur der Synchronize-Antwort 21
7. DeclareIncident 23
7.1. KLE Ereignismeldung 23
7.2. Initiale synchrone Meldung 23
7.3. Ablauf / Protokoll 24
7.4. Datenstruktur der initialen Meldung 26
7.5. Datenstruktur der initialen Antwort 27
7.6. Datenstruktur der Synchronize-Meldung 28
7.7. Datenstruktur der Synchronize-Antwort 28
8. Use Cases 29
8.1. UC001 Initiale Meldung empfangen 29
8.2. UC002 Duplikate behandeln 31
8.3. UC003 Testmeldung empfangen 32
8.4. UC004 Ersatzmeldung verarbeiten 33
8.5. UC005 Synchronisieren 33
8.6. UC006 Prozesskontrolle durchführen 35
8.7. UC007 Fall schliessen 36
8.8. UC008 Stories verarbeiten und quittieren 36
8.9. UC009 Stories melden 37
8.10. UC010 Datenflusskontrolle 38
8.11. UC011 Completion 39
8.12. UC012 Dialog Message abwickeln 42
8.13. UC013 Anmeldungsprozess (SubscribeOrganizationConsumer) 43
8.14. UC014 Unternehmensauthentifizierung (RegisterOrganizationAuthentication) 43
8.15. UC015 Security anwenden 44
8.16. UC016 Wartungsfenster 45
8.17. UC017 Supportanfrage bearbeiten 47
8.18. UC018 Erreichbarkeit prüfen 49
A. Referenzierte Dokumente 52
B. Glossar 53
C. Detailspezifikation Swissdec Unternehmens-Authentifizierung SUA 55
C.1. Einleitung 55

© Swissdec 2026

C.2. SUA Prozess 55
C.3. Technische Anforderungen an die Zertifkate 57
D. Detailspezifikation DialogMessages 60
D.1. Einleitung 60
D.2. Die Verwendung von DialogMessages 60
D.3. Standard DialogMessages 66
E. Detailspezifikation Completion URL 67
E.1. Aufbau und Verarbeitung der Completion URL 67
F. Bausteine 70
F.1. AB-01 Konfiguration 70
F.2. AB-02 Erreichbarkeit 70
F.3. AB-03 Interoperabilität 71
F.4. AB-04 Anmeldung 71
F.5. AB-05 Explizite Adressierung (direkte Adressierung) 72
F.6. AB-06 Implizite Adressierung (Gruppenadressierung) 72
F.7. AB-07 Absichtserklärung 73
F.8. AB-08 Identifikation eines Geschäftsvorfalls 74
F.9. AB-09 Ersatzmeldung 74
F.10. AB-10 Request und Response IDs 75
F.11. AB-11 Testmeldung 75
F.12. AB-12 Plausibilisierung 76
F.13. AB-13 Filterung 77
F.14. AB-14 Versionsmapping 78
F.15. AB-15 Doublettenerkennung 78
F.16. AB-16 Anonymisierung 79
F.17. AB-17 Meldungsfragmentierung 79
F.18. AB-18 Swissdec Unternehmensauthentifizierung (SUA) 79
F.19. SB-01 Transportsicherheit (HTTPS / TLS) 80
F.20. SB-02 Transport Authentication (TLS Mutual Authentication) 80
F.21. SB-03 Nutzdatenverschlüsselung (WS-Encryption) 80
F.22. SB-04 Nutzdatensignatur (WS-Signature) 81
F.23. SB-05 Unternehmens-Authentifizierung SUA Signatur (WS-Signatur) 81
F.24. SB-06 Nichtabstreitbarkeit 81
F.25. PB-01 Synchrone Verteilung 82
F.26. PB-02 Asynchrone Verteilung 82
F.27. PB-03 Synchronisierung 83
F.28. PB-04 Dialogisierung 83
F.29. PB-05 Completion 84
F.30. PB-06 Prozessnavigation 84
F.31. PB-07 Digitalisierungsbereich 85
F.32. PB-08 Daten anfordern 85
G. Technische Dokumentationen 87
G.1. Schema Dokumentation IncidentDeclarationConsumerServiceTypes.xsd 87
G.2. Schema Dokumentation IncidentDeclarationConsumerContainer.xsd 87
G.3. Schema Dokumentation IncidentDeclarationContainer.xsd 90
G.4. Schema Dokumentation IncidentDeclaration.xsd 98
G.5. Schema Dokumentation SwissdecComponents.xsd 120
G.6. Schema Dokumentation Common.xsd 135

© Swissdec 2026

Abbildungsverzeichnis

2.1. Prozessübersicht Swissdec KLE-Standard 3
3.1. Use Cases Initiale Meldung 5
3.2. Use Cases Synchronisierung 6
3.3. Andere Use Cases 6
4.1. Schemabild PingConsumer 8
4.2. Schemabild PingConsumerResponse 10
5.1. Synchroner Meldungsprozess 13
5.2. Schemabild RegisterOrganizationAuthenticationConsumer 14
5.3. Schemabild Addressee 15
5.4. Schemabild RegisterOrganizationAuthenticationConsumerResponse 15
5.5. Schemabild SynchronizeRegisterOrganizationAuthenticationConsumer 16
5.6. Schemabild SynchronizeRegisterOrganizationAuthenticationConsumerResponse 16
6.1. Synchroner Meldungsprozess 18
6.2. Schemabild SubscribeOrganizationConsumer 19
6.3. Schemabild Addressee 20
6.4. Schemabild SubscribeOrganizationConsumerResponse 20
6.5. Schemabild SynchronizeSubscribeOrganizationConsumer 21
6.6. Schemabild SynchronizeSubscribeOrganizationConsumerResponse 21
7.1. Synchroner Meldungsprozess 24
7.2. DeclareIncident Protokollstatus 25
7.3. Schemabild DeclareIncidentConsumer 26
7.4. Schemabild Addressees 27
7.5. Schemabild DeclareIncidentConsumerResponse 27
7.6. Schemabild SynchronizeDeclareIncidentConsumer 28
7.7. Schemabild SynchronizeDeclareIncidentConsumerResponse 28
8.1. Aufbau URL gemäss RFC3986 42
8.2. Schemabild PingConsumer 46
8.3. Schemabild PingConsumerResponse 47
D.1. DialogMessage XML-Schema Struktur 61
D.2. Vorgehen bei der Darstellung der DialogMessage (Aktivitäts Diagramm) 64
E.1. Aufbau URL gemäss RFC3986 69
F.1. AddresseeType 72

© Swissdec 2026

Tabellenverzeichnis

1. Verbindlichkeit von Anforderungen iv
1.1. Beteiligte am Swissdec-Übermittlungsprozess 1
3.1. Use Cases und Operationen 7
4.1. Feldbeschreibungen PingConsumer 8
4.2. Feldbeschreibungen UserAgent 9
4.3. Feldbeschreibungen PingConsumerResponse 10
4.4. Feldbeschreibungen UserAgent 10
5.1. Feldbeschreibungen RegisterOrganizationAuthenticationConsumer 14
5.2. Feldbeschreibungen RequestContext 14
5.3. Feldbeschreibungen Job 15
5.4. Feldbeschreibungen Addressee 15
5.5. Feldbeschreibungen RegisterOrganizationAuthenticationConsumerResponse 15
5.6. Feldbeschreibungen SynchronizeRegisterOrganizationAuthenticationConsumer 16
5.7. Feldbeschreibungen SynchronizeRegisterOrganizationAuthenticationConsumerResponse 16
6.1. Feldbeschreibungen SubscribeOrganizationConsumer 19
6.2. Feldbeschreibungen RequestContext 19
6.3. Feldbeschreibungen Job 20
6.4. Feldbeschreibungen Addressee 20
6.5. Feldbeschreibungen SubscribeOrganizationConsumerResponse 20
6.6. Feldbeschreibungen SynchronizeSubscribeOrganizationConsumer 21
6.7. Feldbeschreibungen SynchronizeSubscribeOrganizationConsumerResponse 22
7.1. Zustandstabelle 25
7.2. Feldbeschreibungen DeclareIncidentConsumer 26
7.3. Feldbeschreibungen RequestContext 27
7.4. Feldbeschreibungen Job 27
7.5. Feldbeschreibungen Addressees 27
7.6. Feldbeschreibungen DeclareIncidentConsumerResponse 27
7.7. Feldbeschreibungen SynchronizeDeclareIncidentConsumer 28
7.8. Feldbeschreibungen SynchronizeDeclareIncidentConsumerResponse 28
8.1. UC001 Initiale Meldung empfangen 29
8.2. UC002 Duplikate behandeln 31
8.3. UC003 Testmeldung empfangen 32
8.4. UC004 Ersatzmeldung verarbeiten 33
8.5. UC005 Synchronisieren 33
8.6. UC006 Prozesskontrolle durchführen 35
8.7. UC007 Fall schliessen 36
8.8. UC008 Stories verarbeiten und quittieren 36
8.9. UC009 Stories melden 37
8.10. UC010 Datenflusskontrolle 38
8.11. UC011 Completion 39
8.12. UC012 Dialog Message abwickeln 42
8.13. UC013 Anmeldungsprozess (SubscribeOrganizationConsumer) 43
8.14. UC014 Unternehmensauthentifizierung (RegisterOrganizationAuthentication) 43
8.15. UC015 Security anwenden 44
8.16. UC016 Wartungsfenster 45
8.17. UC017 Supportanfrage bearbeiten 47
8.18. UC018 Erreichbarkeit prüfen 49
F.1. AB-01 Konfiguration 70
F.2. AB-02 Erreichbarkeit 70
F.3. AB-03 Interoperabilität 71
F.4. AB-04 Anmeldung 71
F.5. AB-05 Explizite Adressierung (direkte Adressierung) 72
F.6. AB-06 Implizite Adressierung (Gruppenadressierung) 72
F.7. AB-07 Absichtserklärung 73

© Swissdec 2026

F.8. AB-08 Identifikation eines Geschäftsvorfalls 74
F.9. AB-09 Ersatzmeldung 74
F.10. AB-10 Request und Response IDs 75
F.11. AB-11 Testmeldung 75
F.12. AB-12 Datenqualitätssicherung 76
F.13. AB-13 Filterung 77
F.14. AB-14 Versionsmapping 78
F.15. AB-15 Doublettenerkennung 78
F.16. AB-16 Anonymisierung 79
F.17. AB-18 Swissdec Unternehmensauthentifizierung (SUA) 79
F.18. SB-01 Transportsicherheit (HTTPS / TLS) 80
F.19. SB-02 Transport Authentication (TLS Mutual Authentication) 80
F.20. SB-03 Nutzdatenverschlüsselung (WS-Encryption) 80
F.21. SB-04 Nutzdatensignatur (WS-Signature) 81
F.22. SB-05 Unternehmens-Authentifizierung SUA Signatur (WS-Signatur) 81
F.23. SB-06 Nichtabstreitbarkeit 81
F.24. PB-01 Synchrone Verteilung 82
F.25. PB-02 Asynchrone Verteilung 82
F.26. PB-03 Synchronisierung 83
F.27. PB-04 Dialogisierung 83
F.28. PB-05 Completion 84
F.29. PB-06 Prozessnavigation 84
F.30. PB-07 Digitalisierungsbereich 85
F.31. PB-08 Daten anfordern 85

© Swissdec 2026

1 Einleitung

Dieses Dokument enthält funktionale, technische und zusätzliche Anforderungen an Endreceiver, die Daten vom Swissdec Distributor erhalten wollen. Es adressiert die technischen Aspekte, nicht die fachliche Logik. Ein Endreceiver wird dazu verwendet, Meldungen, die vom Sendersystem via Distributor elektronisch versendet wurden, zu empfangen und auf diese zu reagieren.

Es sind die mitgeltenden Dokumente im Anhang zu beachten. Besonders in den fachlichen Richtlinien sind für den Endreceiver wesentliche Aspekte mit spezifiziert.

Hierbei ist es wichtig, zwischen den verschiedenen Akteuren im Swissdec-Prozess zu unterscheiden:

Sendersystem	Beim Sendersystem handelt es sich um ein System, das Daten aufbereitet und für den Versand an Endempfänger bereitstellt. Hier werden vor allem die fachlichen Anforderungen technisch korrekt umgesetzt (Beispiel: ERP).
Transmitter	Der Transmitter übermittelt die vom Sendersystem bereitgestellten Daten an den Swissdec Distributor und empfängt die erhaltenen Antworten (Responses), um diese zu verifizieren und an das Sendersystem weiterzuleiten.
Distributor	Der Distributor ist das zentrale System, das Daten vom Transmitter erhält, validiert, plausibilisiert und an die adressierten Endreceiver weiterleitet, sowie die erhaltenen Antworten zurück an den Transmitter sendet. Er übernimmt die Filterung und Verteilung der Daten.
Endreceiver	Der Endreceiver ist das technische Gegenstück zum Transmitter. Er empfängt und validiert die vom Distributor erhaltenen Daten und leitet diese an ein Empfängersystem weiter. Die vom Empfängersystem bereitgestellten Antworten werden danach vom Endreceiver in die Response an den Distributor eingefügt.
Empfängersystem	Das Empfängersystem erhält seine Daten vom Endreceiver um diese dann auf fachlicher Ebene verarbeiten zu können. Es stellt ausserdem die Antworten bereit, die danach über den Endreceiver zurück an den Distributor und das Sendersystem übermittelt werden sollen. (Beispiel: Versicherer, Behörde)

Tabelle 1.1. Beteiligte am Swissdec-Übermittlungsprozess

Die elektronische Übermittlung erfolgt via Webservice (SOAP) und erfolgt jeweils via Request-Response-Kommunikation. Die Kommunikation findet jeweils zwischen Transmitter und Distributor sowie zwischen Distributor und Endreceiver statt.

Dieses Dokument behandelt die Schnittstelle zwischen Distributor und Endreceiver, nicht aber jene zwischen Transmitter und Distributor.

1.1 Aufbau des Dokuments

Dieses Dokument beschreibt im Nachfolgenden:

- Die Geschäftsprozesse, die den zu zertifizierenden Standard definieren. Dieses Kapitel beschreibt, wie die fachlich definierten Prozesse aus technischer Sicht umgesetzt werden.
- Die Operationen, die es ermöglichen, die Geschäftsprozesse abzubilden.
- Die Use Cases, die die Grundlage für die Operationen bilden.
- Im Anhang befinden sich ausserdem detaillierte Beschreibungen der Bausteine, aus denen die Bausteine definiert werden.

© Swissdec 2026

Der Grundgedanke bei der Beschreibung der Operationen besteht darin, einen Bezugspunkt zwischen fachlicher Dokumentation und technischer Spezifikation wie WSDL und XSD zu liefern. Dank grafischer Darstellung der XSDs und Beschreibungen der einzelnen Elemente, sollten die Operationen auf verständliche Weise den Geschäftsprozessen zugeordnet werden können.

Die Use Cases gehen von einem grundlegend technischen Ansatz aus und beschreiben die diversen Anforderungen an die Übermittlungen im Detail. Hier werden zusätzlich zum erfolgreichen Durchlaufen eines Prozesses auch mögliche alternative Abläufe und Fehlersituationen beschrieben. Ausserdem findet man hier zusätzliche Anforderungen und Informationen zur Umsetzung von Spezialfällen.

Die Use Cases setzen sich zwar aus den Bausteinen zusammen, aber es ist nicht zwingend notwendig, alle einzelnen Bausteine zu kennen und zu verstehen. Ihre Auflistung im Anhang dient nur zum besseren Verständnis, wie der vorliegende Swissdec Standard aufgebaut ist.

1.2 Leitgedanke Bausteine

Sämtliche Swissdec Standards basieren auf Bausteinen, die für alle Standards gleich sein müssen. Diese Bausteine sind Grundkonstrukte, auf denen die Use Cases aufbauen.

Es gibt drei Gruppen von Bausteinen:

- **Architekturbausteine:**

  Die Architekturbausteine bilden das "Fundament" und beschreiben die grundlegenden Funktionen, die für die Umsetzung eines Standards nötig sind.

- **Securitybausteine:**

  Die Securitybausteine ermöglichen eine sichere Übermittlung zwischen allen beteiligten Systemen. Sie beschreiben sowohl das Anwenden von Security-Elementen auf die zu übermittelnden Daten, als auch die Analyse der Security auf erhaltenen Daten.

- **Prozessbausteine:**

  Die Prozessbausteine bieten Grundbausteine für das Zusammenfügen eines Übermittlungsprozesses. Sie beschreiben die Grundelemente, aus denen die verschiedenen Übermittlungschoreografien eines Standards zusammengestellt werden können.

© Swissdec 2026

2 Geschäftsprozess zur Übermittlung von Leistungsfällen mittels KLE

Der KLE Standard ist Teil des Datenraums Leistung und unterstützt den Meldungs-Prozess bei Unfall oder Krankheit. Der SOLL-Prozess berücksichtigt dabei vier Rollen:

- der/die MitarbeiterIn (indirekt)
- das Unternehmen (Arbeitgeber)
- der Versicherer
- Dritte wie der Leistungserbringer

Der Schwerpunkt der Kommunikation liegt zwischen Unternehmen und Versicherer.

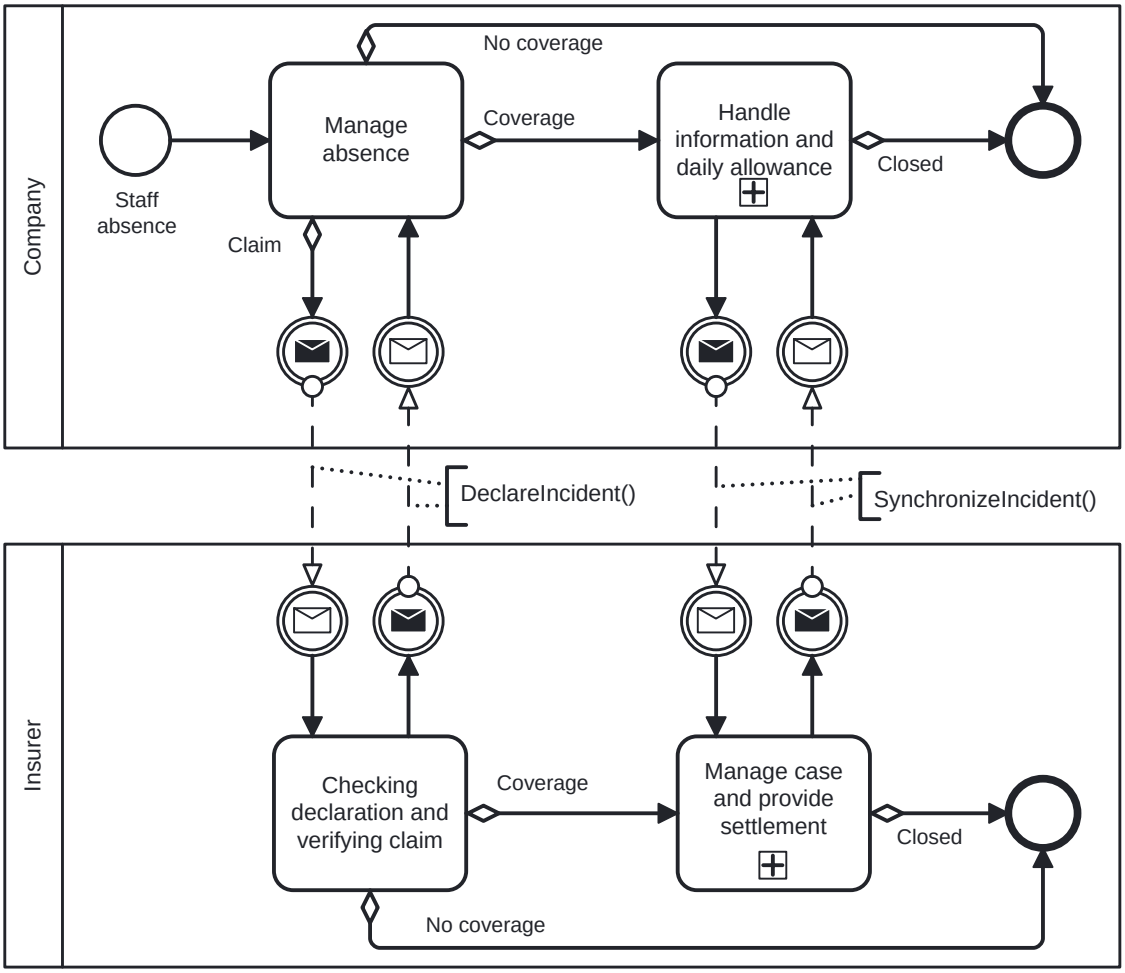


Abbildung 2.1. Prozessübersicht Swissdec KLE-Standard

Der SOLL-Prozess beschreibt den standardisierten Ablauf eines Leistungsfalls vom Ereignis bis zur Leistungserbringung. Auslöser ist, dass ein Mitarbeitender aufgrund von Krankheit, Unfall, Mutterschaft oder Rückfall arbeitsunfähig wird und das Unternehmen informiert. Das Unternehmen prüft im Absenzenmanagement, ob eine Meldung nötig ist, und erfasst die Ereignismeldung direkt im Swissdec-zertifizierten ERP-System. Wird dem Versicherer von Dritten eine Meldung auf separatem Weg zugestellt und kein entsprechendes Ereignis erfasst ist, dann informiert der Versicherer das Unternehmen, das wiederum im Rahmen des Absenzenmanagements die Anmeldung beim Versicherer prüft.

Sobald der Versicherer die Ereignismeldung empfängt, wird ein Dossier eröffnet und die entsprechende *InsuranceCaseID* dem Unternehmen zurückgemeldet. Im Falle eines Rückfalls kann auch ein bestehendes Dossier reaktiviert werden. Falls Informationen fehlen, kann das Unternehmen, der/die MitarbeiterIn oder Drit-

© Swissdec 2026

te (Ärzte) zusätzliche Angaben liefern. Liegen dem Versicherer alle Unterlagen vor, entscheidet dieser über den Leistungsanspruch.

Wird der Anspruch anerkannt, beginnt die Fallführung und die Leistungserbringung durch den Versicherer. Währenddessen tauschen Unternehmen und Versicherer fortlaufend Ereignisinformationen aus, wie Änderungen der Arbeitsunfähigkeit oder Rückfragen. Auch der Mitarbeitende ist verpflichtet über Genesung oder Arbeitsfähigkeit zu informieren.

Der Datenaustausch zwischen Unternehmen und Versicherer erfolgt in strukturierter Form während des gesamten Fallverlaufs, bei dem beide Seiten den gleichen Informationsstand zum Ereignis aufrechterhalten. Änderungen im Verlauf des Falls werden nicht als spezielle Korrekturmeldungen behandelt, sondern durch wiederholte Übermittlung aktualisierter Daten synchronisiert. Der Absender übermittelt dabei jeweils vollständige Informationen, während der Empfänger diese mit seinen vorhandenen Daten vergleicht und Abweichungen als Anpassungen erkennt. Ergänzend zeigt ein Statusmechanismus dem Unternehmen jederzeit den Bearbeitungsstand des Ereignisses und die noch offenen Schritte im Prozess.

Zur eindeutigen Zuordnung eines Ereignisses verwendet KLE zudem mehrere Fallkennungen auf Ebene Unternehmen (CompanyCaseID), Versicherer (InsuranceCaseID) und Übermittlung (IncidentCaseID). Diese IDs werden während der gesamten Fallführung mitgeführt und stellen sicher, dass alle Beteiligten denselben Leistungsfall korrekt referenzieren können.

Der KLE-Prozess kann ausschliesslich durch den Versicherer abgeschlossen werden, sobald der Mitarbeitende wieder arbeitsfähig ist, austritt, verstirbt, sämtliche Taggelder vollständig ausgerichtet wurden, eine erste Rente zugesprochen wird oder der Versicherer das Ereignis ablehnt.

© Swissdec 2026

3 Use Cases und zugehörige Operationen

3.1 Übersicht

Ein Anwendungsfall (Use Case) bündelt alle möglichen Szenarien, die eintreten können, wenn ein Akteur versucht, mithilfe des betrachteten Systems ein bestimmtes fachliches Ziel zu erreichen. Er beschreibt, was inhaltlich bei dem Versuch der Zielerreichung passieren kann und abstrahiert von konkreten technischen Lösungen.

In diesem Kapitel werden die Geschäftsprozesse des Leistungsstandards KLE in Use Cases aufgeteilt, die es ermöglichen sollen, die Prozesse mit möglichen Alternativszenarien abzubilden.

Während die Operationen standardspezifisch sind, werden die Use Cases generisch gehalten und können in verschiedenen Swissdec-Standards äquivalent wiederverwendet oder eingeschränkt werden.

3.2 Grafische Darstellung der Geschäftsprozesse und ihren Use Cases

Die initiale Meldung besteht darin, das Ereignis vom Transmittern an den Distributor zu übermitteln (Abschnitt 8.1, „UC001 Initiale Meldung empfangen“). Die Meldung kann dabei als Testfall (Abschnitt 8.3, „UC003 Testmeldung empfangen“) gekennzeichnet werden. Für eine erfolgreiche Übermittlung müssen Request und Response den Security-Richtlinien entsprechen (Abschnitt 8.15, „UC015 Security anwenden“), was im Falle der Response vom Transmitter verifiziert werden **muss**.

Die Use Cases Abschnitt 8.2, „UC002 Duplikate behandeln“ und Abschnitt 8.4, „UC004 Ersatzmeldung verarbeiten“ werden von KLE nicht unterstützt.

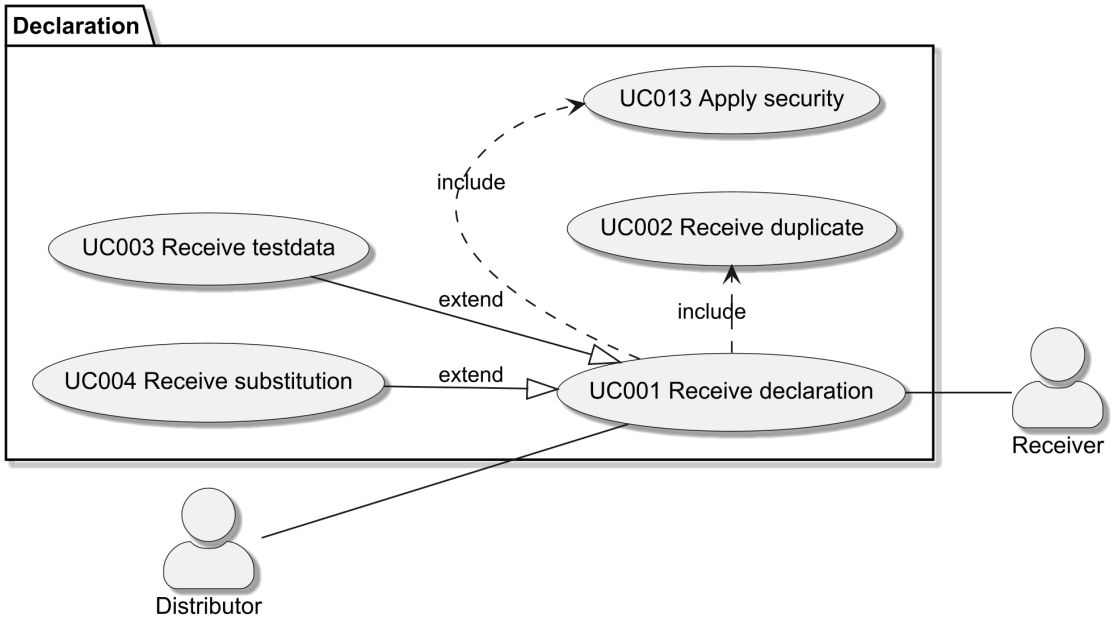


Abbildung 3.1. Use Cases Initiale Meldung

Nach erfolgreicher Deklaration werden Synchronisierungen (Abschnitt 8.5, „UC005 Synchronisieren“) durchgeführt, währenddessen werden Stories gemeldet und quittiert (Abschnitt 8.8, „UC008 Stories verarbeiten und quittieren“), aber auch beim Empfänger abgeholt und verarbeitet (Abschnitt 8.9, „UC009 Stories melden“). Dies soll anhand der Prozesskontrolle (Abschnitt 8.6, „UC006 Prozesskontrolle durchführen“) funktionieren, wobei Abschnitt 8.12, „UC012 Dialog Message abwickeln“ auch Teil davon ist. Der Geschäftsfall wird solange synchronisert bis der Fall mit dem State *CommunicationStopped* abgeschlossen wird (Abschnitt 8.7, „UC007 Fall schliessen“).

Um eine Überlastung der betroffenen Systeme zu verhindern, können alle beteiligten Systeme den Datenfluss kontrollieren (Abschnitt 8.10, „UC010 Datenflusskontrolle“). Für eine erfolgreiche Synchronisierung müssen

© Swissdec 2026

Request und Response den Security-Richtlinien entsprechen (Abschnitt 8.15, „UC015 Security anwenden“), was im Falle der Response vom Transmitter verifiziert werden **muss**.

Der Use Case Abschnitt 8.11, „UC011 Completion“ wird von KLE nicht unterstützt.

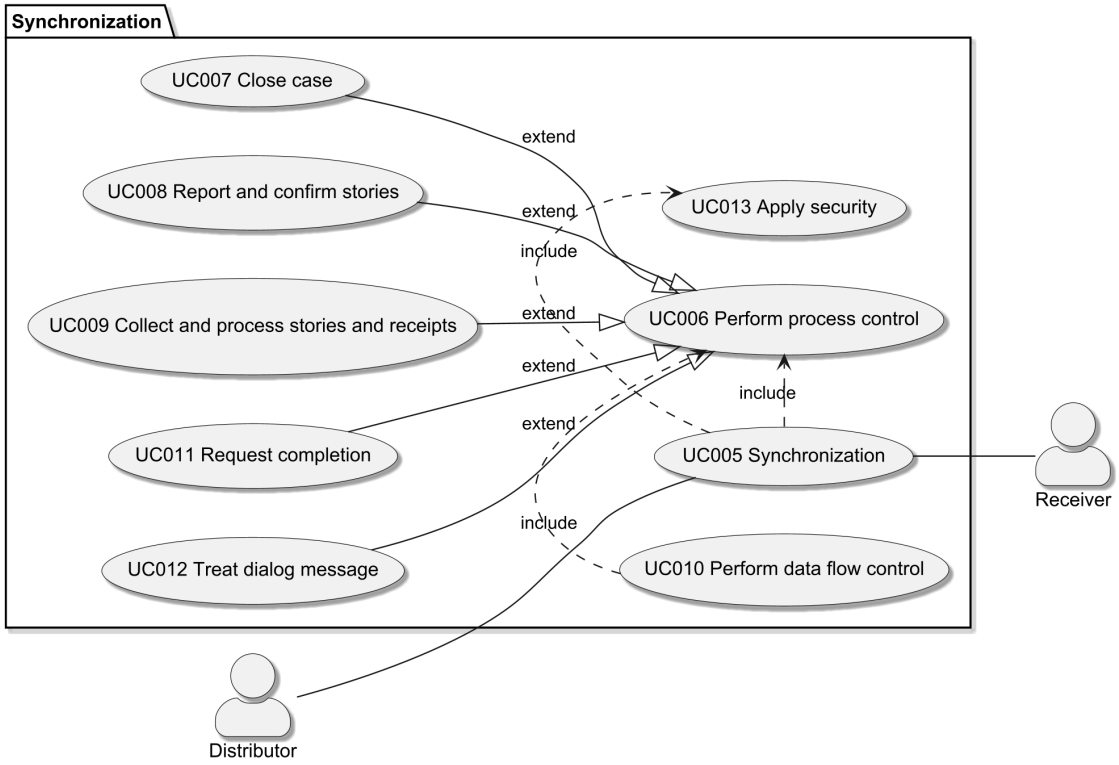


Abbildung 3.2. Use Cases Synchronisierung

Weitere Use Cases umfassen das vorgängige Konfigurieren eines Transmitters für die Übermittlung, d.h. das Einpflegen der aktuell gültigen Versicherungsprofile (Abschnitt 8.16, „UC016 Wartungsfenster“) und die Darstellung von Supportinformationen (Abschnitt 8.17, „UC017 Supportanfrage bearbeiten“) im Fehlerfall.

Des Weiteren gibt es einen Erreichbarkeitstest (Abschnitt 8.18, „UC018 Erreichbarkeit prüfen“), der vom Distributor regelmässig durchgeführt wird.

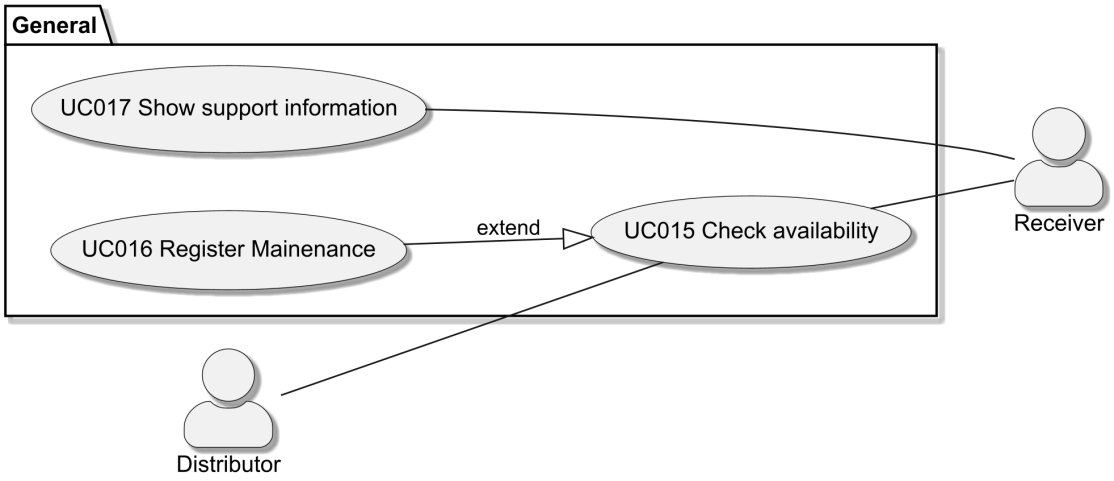


Abbildung 3.3. Andere Use Cases

© Swissdec 2026

3.3 Uses Cases und ihre Operationen

In der nachfolgenden Tabelle werden die Operationen des Leistungsstandards auf die generisch gehaltenen Use Cases abgebildet. Dies soll dem besseren Verständnis der Zusammenhänge zwischen den fachlichen Geschäftsprozessen und den technischen Strukturen und Abläufen dienen.

Use Case	Operation / Element
UC001 Initiale Meldung senden	• DeclareIncidentConsumer: Operation zur Registrierung eines Ereignis (Unfall oder Krankheit) • DeclareIncidentConsumerResponse: Rückmeldung zur Registrierung eines Ereignisn • IncidentDeclarationConsumerFault: Rückmeldung im Fehlerfall
UC005 Synchronisieren	• SynchronizeDeclareIncidentConsumer: Aktualisierung des Ereignises • SynchronizeDeclareIncidentConsumerResponse: Rückmeldung der Aktualisierungsanfrage • IncidentDeclarationConsumerFault: Rückmeldung im Fehlerfall
UC013 Anmeldungsprozess	• SubscribeOrganizationConsumer: Anmeldungsprozess • SubscribeOrganizationConsumerResponse: Rückmeldung zum Anmeldungsprozess • SynchronizeSubscribeOrganizationConsumer: Aktualisierung des Anmeldungsprozesses • SynchronizeSubscribeOrganizationConsumerResponse: Rückmeldung der Aktualisierungsanfrage • IncidentDeclarationConsumerFault: Rückmeldung im Fehlerfall
UC014 Unternehmensauthentifizierung (SUA)	• RegisterOrganizationAuthenticationConsumer: SUA-Unternehmensauthentifizierung • RegisterOrganizationAuthenticationConsumerResponse: Rückmeldung zur SUA-Unternehmensauthentifizierung • SynchronizeRegisterOrganizationAuthenticationConsumer: Aktualisierung des SUA-Prozesses • SynchronizeRegisterOrganizationAuthenticationConsumerResponse: Rückmeldung der Aktualisierungsanfrage • IncidentDeclarationConsumerFault: Rückmeldung im Fehlerfall
UC015 Erreichbarkeit prüfen	• PingConsumer • PingConsumerResponse

Tabelle 3.1. Use Cases und Operationen

© Swissdec 2026

4 Ping

Die Erreichbarkeit des Endempfängers muss geprüft werden. Dazu wird eine einfache Anfrage vom Distributor an den Endempfänger geschickt. Die Antwort des Endempfängers bestätigt die Erreichbarkeit.

Mit dem Ping-Aufruf wird die Systemzeit übermittelt, sodass es möglich ist, die Zeiten von Distributor und Endempfängers zu vergleichen. Damit lassen sich Timestamp-Probleme aufdecken. Dieser Use Case dient der Qualitätssicherung bei der Installation und der Entwicklung.

Die Anforderungen sind im Abschnitt 8.15, „UC015 Security anwenden“ beschrieben.

Der Endempfänger kann in der Antwort auch auf allfällige Wartungsfenster in der Zukunft hinweisen.

Die Anforderungen sind im Abschnitt 8.16, „UC016 Wartungsfenster“ beschrieben.

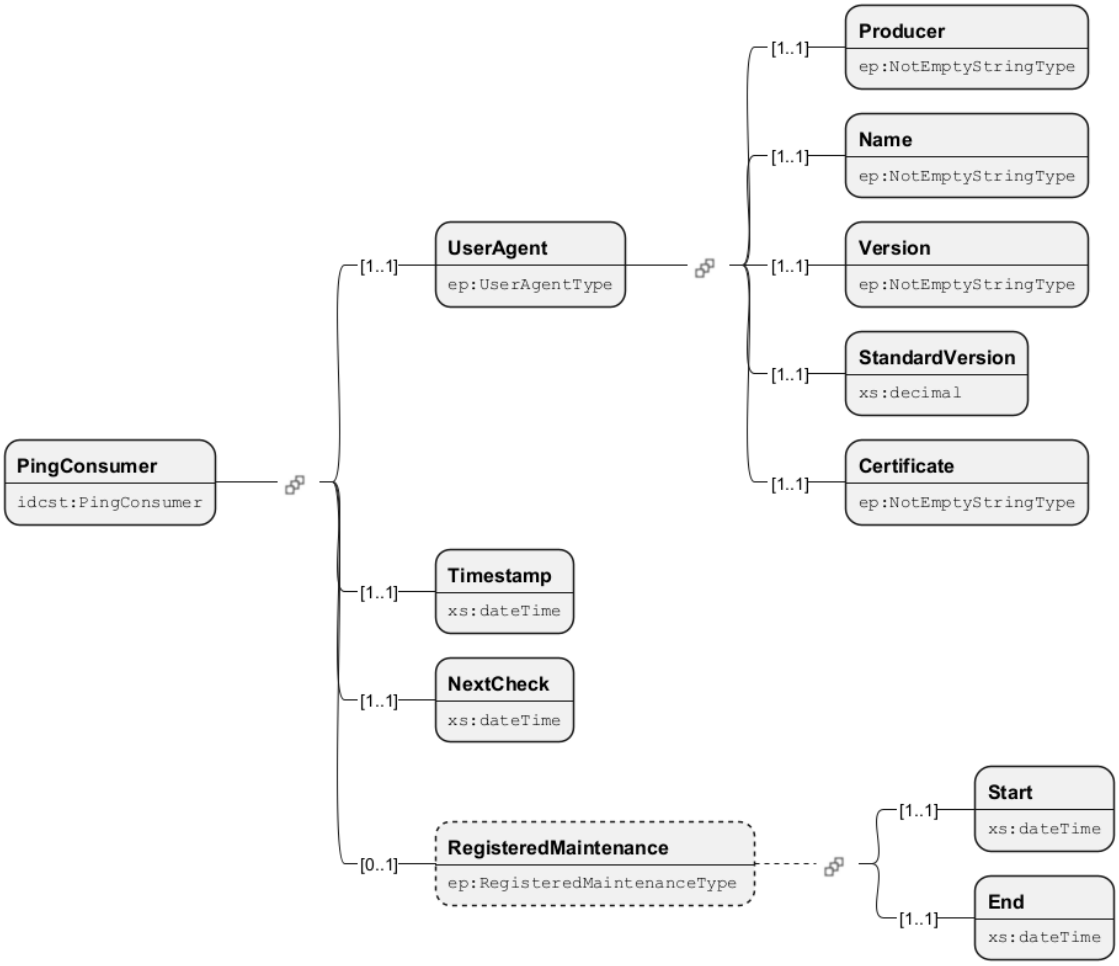


Abbildung 4.1. Schemabild PingConsumer

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType
Timestamp	Ein Zeitstempel (englisch timestamp) wird benutzt, um einem Ereignis einen eindeutigen Zeitpunkt zuzuordnen.	xs dateTime

© Swissdec 2026

Feldname	Beschreibung	Typ
NextCheck		xs dateTime
RegisteredMaintenance		ep RegisteredMaintenanceType

Tabelle 4.1. Feldbeschreibungen PingConsumer

Feldname	Beschreibung	Typ
Producer	Hersteller der Applikation	ep NotEmptyStringType
Name	Name des Produktes + eventuelle Zusatzinformationen (Kauftransmitter, etc.)	ep NotEmptyStringType
Version	Produkt-Version	ep NotEmptyStringType
StandardVersion	Die Version des Standard-CH (z.b. ELM, KLE, SUA, usw.) nach der die Zertifizierung durchgeführt wurde.(z.B. 1.0) Achtung: Der Distributor kann zum Teil Versionen transformieren und wird dann automatisch eine tiefere Version setzen (z.B. 2.2).	xs decimal
Certificate	Zertifikatsnummer, xxxx.yy wie auf dem physischen Zertifikat abgebildet	ep NotEmptyStringType

Tabelle 4.2. Feldbeschreibungen UserAgent

© Swissdec 2026

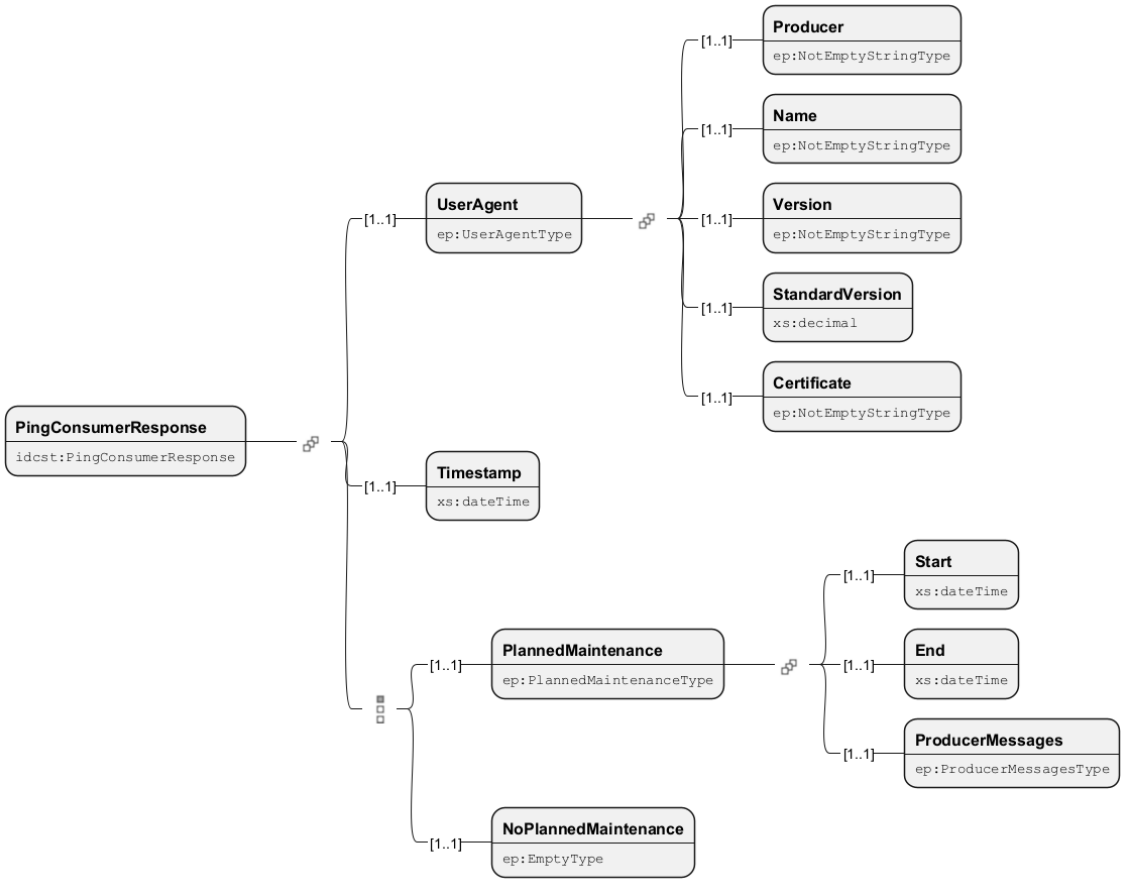


Abbildung 4.2. Schemabild PingConsumerResponse

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType
Timestamp		xs dateTime
PlannedMaintenance		ep PlannedMaintenanceType
NoPlannedMaintenance		ep EmptyType

Tabelle 4.3. Feldbeschreibungen PingConsumerResponse

Feldname	Beschreibung	Typ
Producer	Hersteller der Applikation	ep NotEmptyStringType
Name	Name des Produktes + eventuelle Zusatzinformationen (Kauftransmitter, etc.)	ep NotEmptyStringType
Version	Produkt-Version	ep NotEmptyStringType
StandardVersion	Die Version des Standard-CH (z.b. ELM, KLE, SUA, usw.) nach der die Zertifizierung durchgeführt wurde.(z.B. 1.0) Achtung: Der	xs decimal

© Swissdec 2026

Feldname	Beschreibung	Typ
	Distributor kann zum Teil Versionen transformieren und wird dann automatisch eine tiefere Version setzen (z.B. 2.2).	
Certificate	Zertifikatsnummer, xxxx.yy wie auf dem physischen Zertifikat abgebildet	ep_NotEmptyString-Type

Tabelle 4.4. Feldbeschreibungen UserAgent

© Swissdec 2026

5 RegisterOrganizationAuthentication

5.1 Übersicht

RegisterOrganizationAuthentication ist der unterstützende Prozess für die Swissdec Unternehmens-Authentifizierung SUA. Die Verwendung der vorliegenden Operationen wird im Anhang C, *Detailspezifikation Swissdec Unternehmens-Authentifizierung SUA* beschrieben.

Der Endreceiver unterstützt den SUA Prozess im Rahmen der Registrierung zur Überprüfung der Identität des Unternehmens basierend auf der bestehenden Vertragsbeziehung. Der Endreceiver prüft die übermittelte Vetragsbeziehung und meldet dem Distributor, ob die Vertragsbeziehung gültig ist. In der Antwortet werden die Stammdaten des Unternehmens an den Distributor gemeldet. Zu diesen Informationen gehören der Name des Unternehmens, die UID, die hinterlegte Unternehmensadresse, die Kontaktperson sowie allfällige Stellvertreter. Diese Informationen sind essenziell für die Sicherheit des SUA Prozesses und werden vom Distributor benötigt, um die Identität des Unternehmens zu überprüfen.

Nach der erfolgreichen Registrierung wird der SUA Prozess hauptsächlich vom Distributor gehandhabt. Die SUA Zertifikate werden auf dem Distributor erstellt und signiert. Der Distributor behält sich aber vor, bei einer Erneuerung eines SUA Zertifikats die Vertragsverbindung wieder zu prüfen. Deshalb bleibt die Registrierung längerfristig als offener Fall bestehen.

5.2 Initiale synchrone Meldung

Die Aktivitätsdiagramme sind ohne den Transmitter dargestellt. Die wesentliche Receiverfunktionalität *soll* dargestellt werden. Die Anzeigen erfolgen wahlweise in Receiver oder im Empfängersystem.

© Swissdec 2026

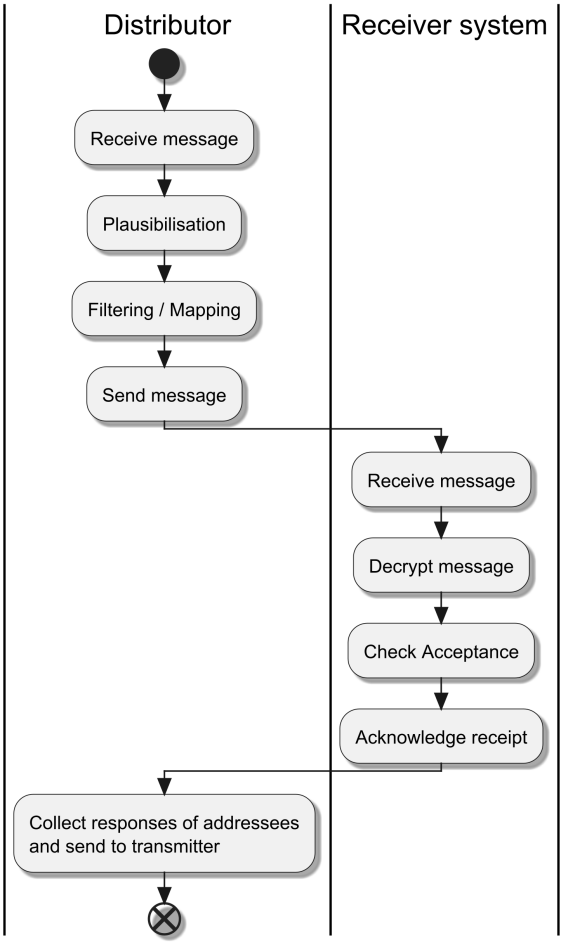


Abbildung 5.1. Synchroner Meldungsprozess

© Swissdec 2026

5.3 Datenstruktur der initialen Meldung

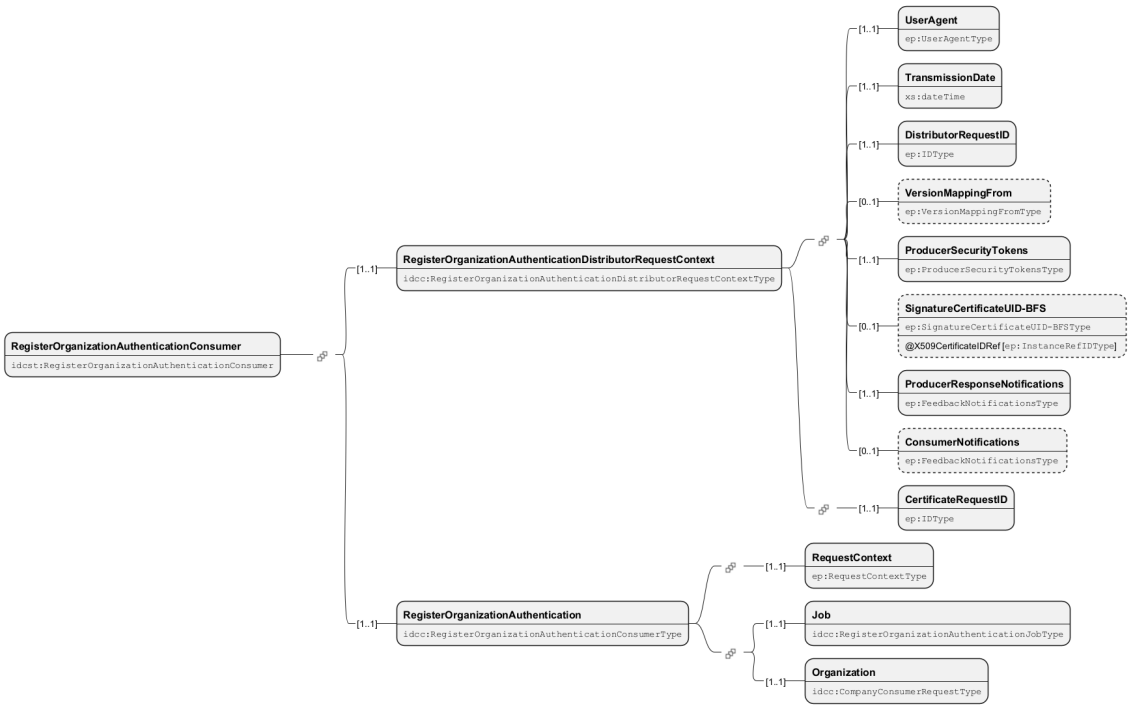


Abbildung 5.2. Schemabild RegisterOrganizationAuthenticationConsumer

Feldname	Beschreibung	Typ
RegisterOrganizationAuthentication-DistributorRequest-Context		idcc RegisterOrganizationAuthenticationDistributorRequestContextType
RegisterOrganizationAuthentication		idcc RegisterOrganizationAuthenticationConsumerType

Tabelle 5.1. Feldbeschreibungen RegisterOrganizationAuthenticationConsumer

Feldname	Beschreibung	Typ
UserAgent	Beschreibung der wesentlichen System-Identifikation-Daten	ep UserAgentType
CompanyName	Beschreibung der wesentlichen Unternehmensdaten	ep NotEmptyStringType
TransmissionDate	Übertragungszeitpunkt	xs dateTime
RequestID	AuftragsID	ep IDType
LanguageCode	Sprachcode	ep LanguageCodeType
MonitoringID	Die MonitoringID erlaubt es, auf den Testsystemen Übermittlungen den Benutzern zuzuweisen. Sie ist in der Produktion überflüssig.	ep MonitoringIDType

© Swissdec 2026

Feldname	Beschreibung	Typ

Tabelle 5.2. Feldbeschreibungen RequestContext

Feldname	Beschreibung	Typ
Addressee	Adressierung der Endempfänger	ep AddresseeJobType
TestCase		ep EmptyType

Tabelle 5.3. Feldbeschreibungen Job

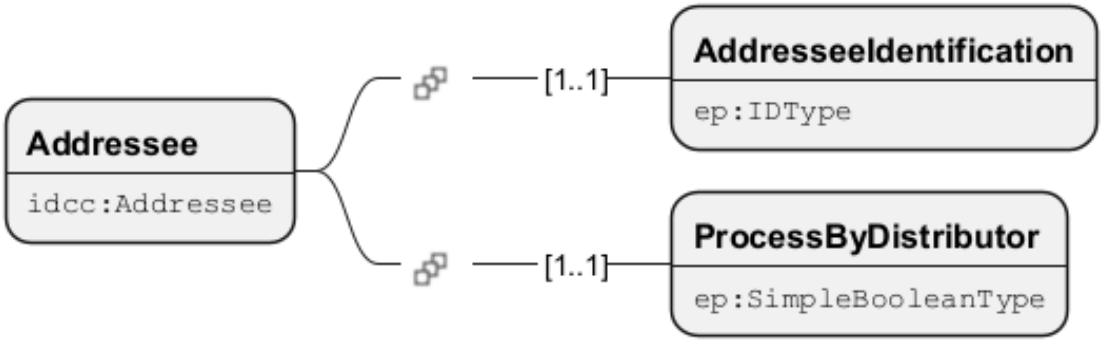


Abbildung 5.3. Schemabild Addressee

Feldname	Beschreibung	Typ
AddresseeIdentification	Identifikation des Adressaten	ep IDType
ProcessByDistributor	Verteilung durch Distributor	ep SimpleBooleanType
@addresseeID	Referenz auf Adressat	ep InstanceRefIDType

Tabelle 5.4. Feldbeschreibungen Addressee

5.4 Datenstruktur der initialen Antwort



Abbildung 5.4. Schemabild RegisterOrganizationAuthenticationConsumerResponse

Feldname	Beschreibung	Typ
RegisterOrganizationAuthenticationResponse		c RegisterOrganizationAuthenticationAddresseeSuccessJobStateType

Tabelle 5.5. Feldbeschreibungen RegisterOrganizationAuthenticationConsumerResponse

© Swissdec 2026

5.5 Datenstruktur der Synchronize-Meldung

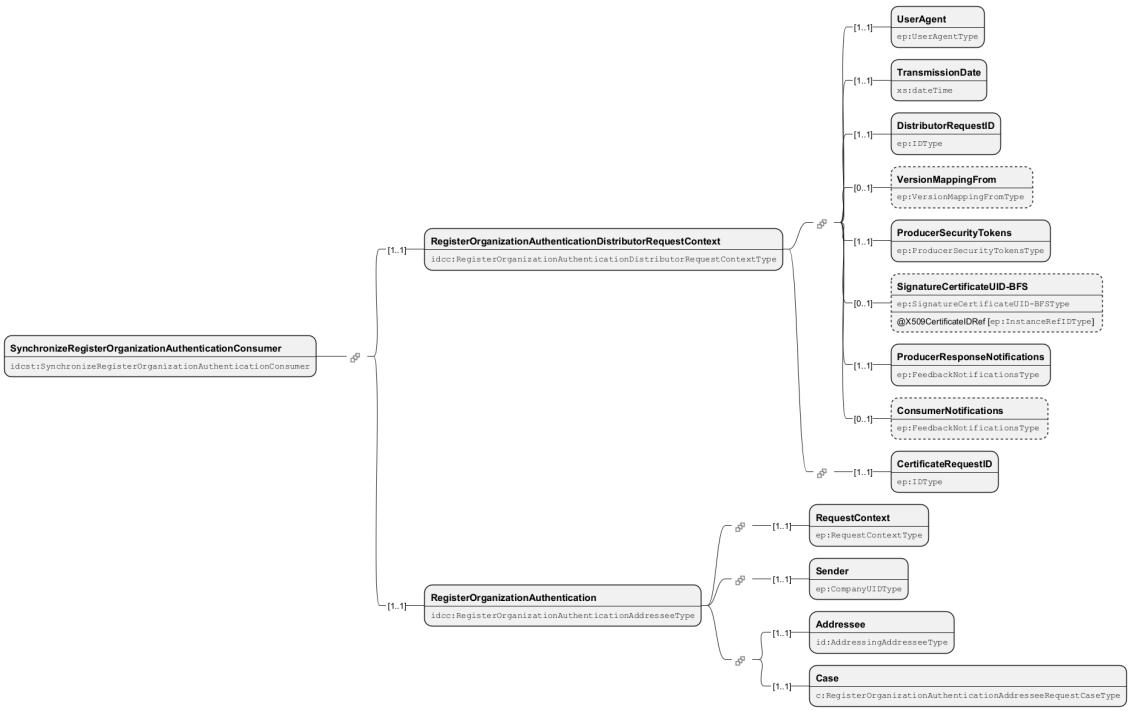


Abbildung 5.5. Schemabild SynchronizeRegisterOrganizationAuthenticationConsumer

Feldname	Beschreibung	Typ
RegisterOrganizationAuthentication-DistributorRequest-Context		idcc RegisterOrganizationAuthenticationDistributorRequestContextType
RegisterOrganizationAuthentication		idcc RegisterOrganizationAuthenticationAddresseeType

Tabelle 5.6. Feldbeschreibungen SynchronizeRegisterOrganizationAuthenticationConsumer

5.6 Datenstruktur der Synchronize-Antwort



Abbildung 5.6. Schemabild SynchronizeRegisterOrganizationAuthenticationConsumerResponse

Feldname	Beschreibung	Typ
SynchronizeRegisterOrganizationAuthentication		c SynchronizeRegisterOrganizationAuthenticationAddresseeResponseType

Tabelle 5.7. Feldbeschreibungen SynchronizeRegisterOrganizationAuthenticationConsumerResponse

© Swissdec 2026

6 SubscribeOrganization

6.1 Übersicht

Ein Sendersystem muss sich bei einem Empfängersystem für einen späteren Übermittlungsprozess (z.B. EO) anmelden können. Mit diesem Anmeldeprozess wird der zukünftige Informationsaustausch über den Distributor zwischen den Beteiligten in die Wege geleitet. Der Anmeldeprozess dient dazu, Informationen auf Vertragsebene auszutauschen und keine fachlichen Inhalte. Diese werden je nach Swissdec Standard dann erst nach erfolgreich abgeschlossenem Anmeldeprozess mittels anderer Operationen übermittelt.

Der Anmeldeprozess besteht aus drei Schritten:

- **SubscribeOrganization**: In einem ersten Schritt übermittelt das Unternehmen seine Informationen an den Endempfänger, um sich für den späteren Informationsaustausch anzumelden. Als Antwort darauf erhält der Absender eine SubscriptionID, die in den weiteren Schritten zur Identifikation beim Endempfänger genutzt wird.

- **SynchronizeSubscribeOrganization**: In allen weiteren Schritten können nun mittels SubscriptionID Informationen beim Endempfänger abgeholt werden.

- **SynchronizeSubscribeOrganization**: Es *sollten* etwa 1x wöchentlich weitere Synchronisierungen durchgeführt werden. So wird das Sendersystem zeitnah über Änderungen seitens Endempfänger informiert.

6.2 Initiale synchrone Meldung

Die Aktivitätsdiagramme sind ohne den Transmitter dargestellt. Die wesentliche Receiverfunktionalität *soll* dargestellt werden. Die Anzeigen erfolgen wahlweise in Receiver oder im Empfängersystem.

© Swissdec 2026

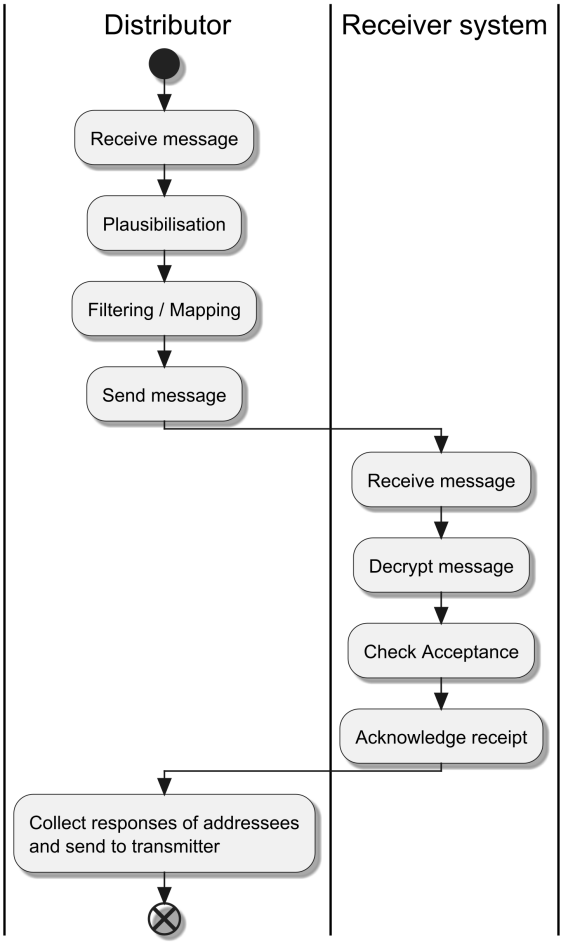


Abbildung 6.1. Synchroner Meldungsprozess

© Swissdec 2026

6.3 Datenstruktur der initialen Meldung

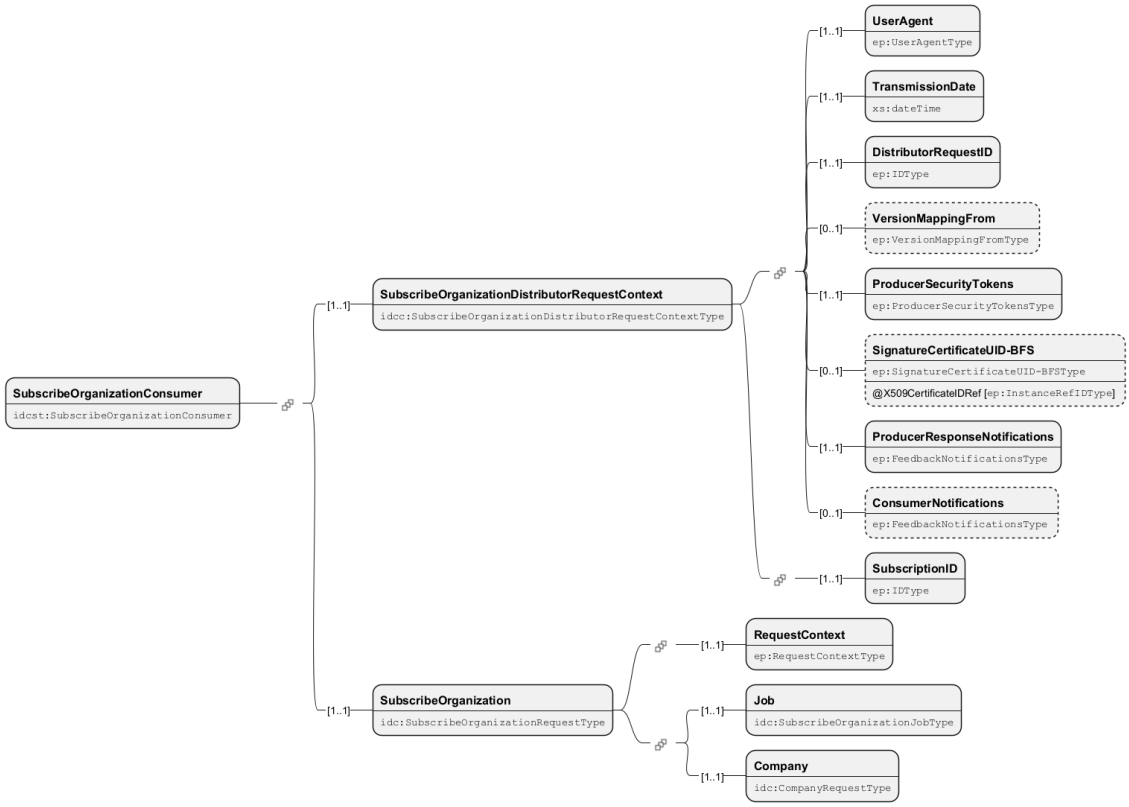


Abbildung 6.2. Schemabild SubscribeOrganizationConsumer

Feldname	Beschreibung	Typ
SubscribeOrganizationDistributorRequestContext		idcc SubscribeOrganizationDistributorRequestContextType
SubscribeOrganization		idc SubscribeOrganizationRequestType

Tabelle 6.1. Feldbeschreibungen SubscribeOrganizationConsumer

Feldname	Beschreibung	Typ
UserAgent	Beschreibung der wesentlichen System-Identifikation-Daten	ep UserAgentType
CompanyName	Beschreibung der wesentlichen Unternehmensdaten	ep NotEmptyStringType
TransmissionDate	Übertragungszeitpunkt	xs dateTime
RequestID	AuftragsID	ep IDType
LanguageCode	Sprachcode	ep LanguageCodeType

© Swissdec 2026

Feldname	Beschreibung	Typ
MonitoringID	Die MonitoringID erlaubt es, auf den Testsystemen Übermittlungen den Benutzern zuzuweisen. Sie ist in der Produktion überflüssig.	ep MonitoringIDType

Tabelle 6.2. Feldbeschreibungen RequestContext

Feldname	Beschreibung	Typ
Addressee	Adressierung der Endempfänger	ep AddresseeJobType

Tabelle 6.3. Feldbeschreibungen Job

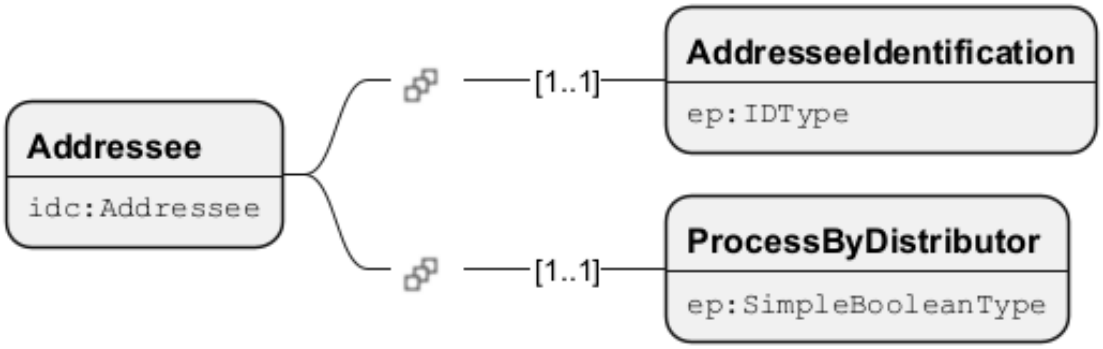


Abbildung 6.3. Schemabild Addressee

Feldname	Beschreibung	Typ
AddresseeIdentification	Identifikation des Adressaten	ep IDType
ProcessByDistributor	Verteilung durch Distributor	ep SimpleBooleanType
@addresseeID	Referenz auf Adressat	ep InstanceRefIDType

Tabelle 6.4. Feldbeschreibungen Addressee

6.4 Datenstruktur der initialen Antwort



Abbildung 6.4. Schemabild SubscribeOrganizationConsumerResponse

Feldname	Beschreibung	Typ
SubscribeOrganizationResponse	Initialer Status des Falls.	idc SubscribeOrganizationAddresseeSuccessJobStateType

Tabelle 6.5. Feldbeschreibungen SubscribeOrganizationConsumerResponse

© Swissdec 2026

6.5 Datenstruktur der Synchronize-Meldung

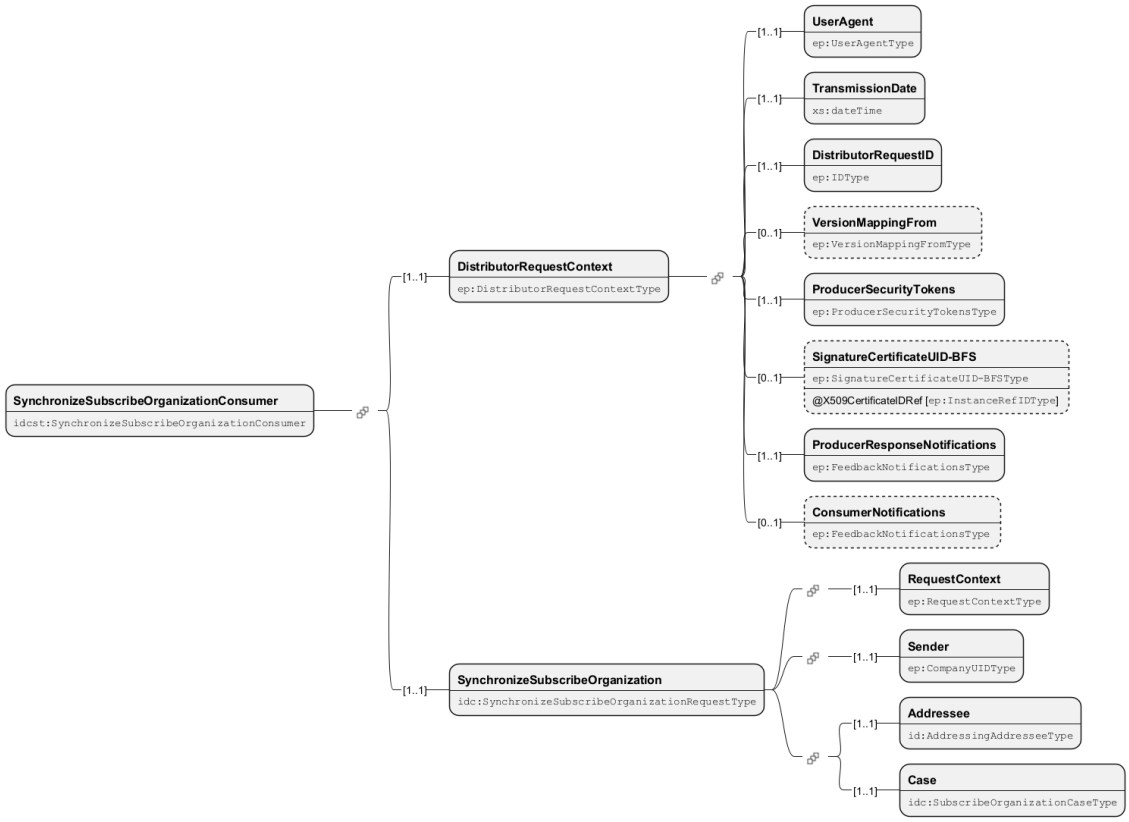


Abbildung 6.5. Schemabild SynchronizeSubscribeOrganizationConsumer

Feldname	Beschreibung	Typ
DistributorRequest-Context		ep DistributorRe-questContextType
SynchronizeSub-scribeOrganization		idc SynchronizeSub-scribeOrganization-RequestType

Tabelle 6.6. Feldbeschreibungen SynchronizeSubscribeOrganizationConsumer

6.6 Datenstruktur der Synchronize-Antwort



Abbildung 6.6. Schemabild SynchronizeSubscribeOrganizationConsumerResponse

© Swissdec 2026

Feldname	Beschreibung	Typ
SynchronizeSubscribeOrganizationResponse		idc SynchronizeSubscribeOrganizationConsumerType

Tabelle 6.7. Feldbeschreibungen SynchronizeSubscribeOrganizationConsumerResponse

© Swissdec 2026

7 DeclareIncident

7.1 KLE Ereignismeldung

Die Richtlinien für Lohndatenverarbeitung beschreiben, welche Daten zu welchem Zeitpunkt übermittelt werden. Dieses Dokument beschränkt sich darauf, die technischen Informationen zur Übermittlung zu liefern.

Die KLE Ereignismeldung findet in mehreren Etappen statt, welche nachfolgend beschrieben werden.

7.1.1 Ereignis melden: DeclareIncidentConsumer

In diesem ersten Schritt wird dem Endempfänger in einer synchronen Übermittlung ein Ereignis gemeldet. Die Ereignismeldung geht vom Transmitter an den Distributor und wird von diesem sofort weitergeleitet zum Endreceiver. Der Endreceiver überprüft die Meldung auf Akzeptanzstufe (z.B. Vertragsnummer) und sendet via Distributor zurück, ob er das Ereignis annimmt. Im Erfolgsfall kann nun vom Sendersystem mit dem nächsten Schritt, der Synchronisierung, fortgefahren werden.

7.1.2 Synchronisierung des Ereignisses: SynchronizeDeclareIncident

Sender- und Empfängersystem kommunizieren nun mittels Stories, in denen zusätzliche Informationen zum Ereignis, sowie Zahlungsinformationen vom Endempfänger übermittelt werden können. Der Endempfänger kann dem Sendersystem jeweils mitteilen, wenn weitere Stories zur Bearbeitung des Falls benötigt werden und auch darüber informieren, wenn Informationen seitens Empfänger zum Abholen bereitstehen. So kann der Fall in einer flexiblen Anzahl Synchronisierungen abgewickelt werden.

7.1.3 Regelmässiges Synchronisieren der Ereignisse

Auch wenn kein konkreter Fall in Arbeit ist, führt das Sendersystem regelmässige Synchronisierungen über ein ganzes Unternehmen durch. So kann der Endempfänger informieren, wenn für ein aktuell inaktives Ereignis neue Informationen zur Verfügung stehen.

7.2 Initiale synchrone Meldung

Die Aktivitätsdiagramme sind ohne den Transmitter dargestellt. Die wesentliche Receiverfunktionalität *soll* dargestellt werden. Die Anzeigen erfolgen wahlweise in Receiver oder im Empfängersystem.

© Swissdec 2026

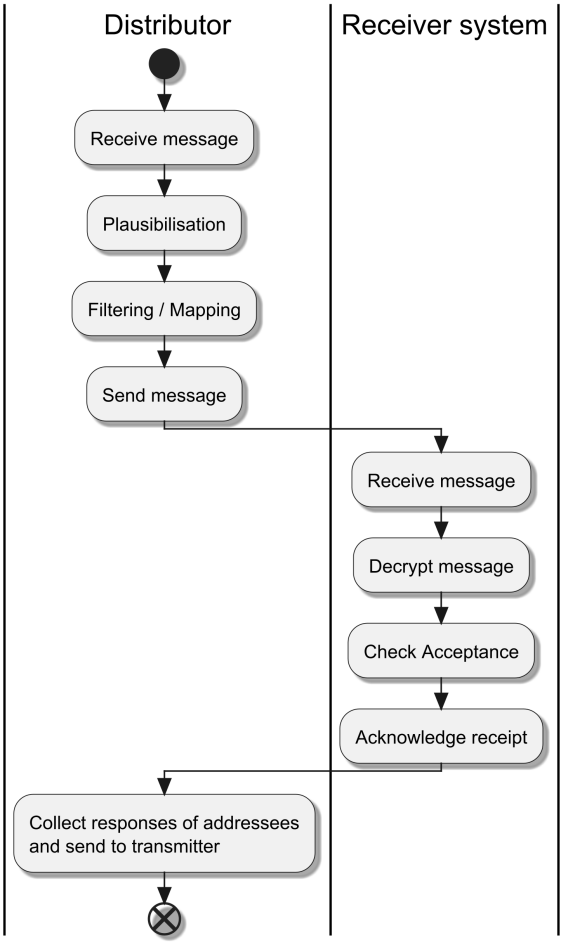


Abbildung 7.1. Synchroner Meldungsprozess

7.3 Ablauf / Protokoll

Das nachfolgende Zustandsdiagramm und die zugehörige Zustandstabelle beschreiben den vorgesehenen Ablauf sowie die möglichen Statusmeldungen innerhalb des Protokolls.

Die Darstellung ist nicht als Zustandsmaschine (Finite State Machine, FSM) zu verstehen. Insbesondere wird der Folgezustand nicht durch den Transmitter bestimmt, sondern durch den Emfpänger im Rahmen der Verarbeitung ermittelt und als `<State>` zurückgemeldet. Die Kontrolle über den Ablauf obliegt somit dem Emfpängersystem und damit verbunden die Sicherstellung des korrekten Ablaufes mit Stories und Status.

Der Status repräsentiert den aktuellen Prozesszustand und hat Priorität vor eventuell gemeldeten Stories. Das heisst, Stories welche mit dem Status "finished" gemeldet werden, müssen verarbeitet werden. Der Prozess gilt danach für den Endempfänger als abgeschlossen.

Abweichungen vom vorgesehenen Ablauf dürfen nicht unmittelbar zu einer Blockierung der beteiligten Systeme führen. Die Implementierungen müssen fehlertolerant ausgelegt sein und insbesondere temporäre Inkonsistenzen oder unerwartete Statusmeldungen tolerieren können.

Bei Abweichungen muss eine erneute Synchronisation zwischen den beteiligten Systemen möglich sein, sodass die Verarbeitung nach Möglichkeit ohne manuellen Eingriff fortgesetzt werden kann. Erst wenn eine Abweichung dauerhaft besteht und auch durch erneute Synchronisationsversuche nicht behoben werden kann, darf der betroffene Geschäftsvorfall als fehlerhaft behandelt werden.

Ziel ist eine robuste und entkoppelte Protokollverarbeitung, bei der vorübergehende Störungen nicht zu einem dauerhaften Verarbeitungsabbruch führen.

© Swissdec 2026

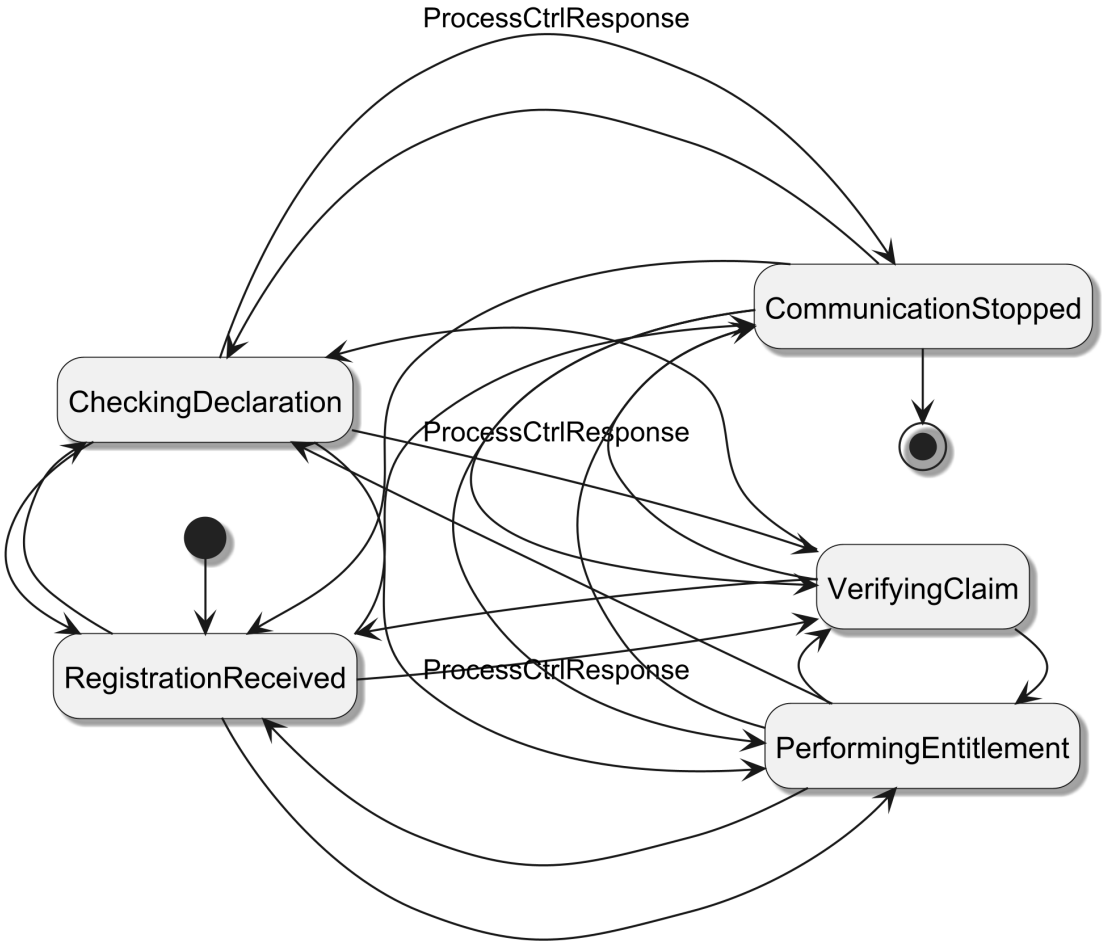


Abbildung 7.2. DeclareIncident Protokollstatus

Aktueller Zustand	Story	Gemeldeter Zustand
[START]	*Initiale Meldung*	RegistrationReceived
RegistrationReceived	*ohne Story*	CheckingDeclaration
RegistrationReceived	*ohne Story*	VerifyingClaim
RegistrationReceived	*ohne Story*	PerformingEntitlement
RegistrationReceived	*ohne Story*	CommunicationStopped
CheckingDeclaration	*ohne Story*	RegistrationReceived
CheckingDeclaration	ProcessCtrlResponse	VerifyingClaim
CheckingDeclaration	ProcessCtrlResponse	PerformingEntitlement
CheckingDeclaration	ProcessCtrlResponse	CommunicationStopped
VerifyingClaim	*ohne Story*	RegistrationReceived
VerifyingClaim	*ohne Story*	CheckingDeclaration
VerifyingClaim	*ohne Story*	PerformingEntitlement
VerifyingClaim	*ohne Story*	CommunicationStopped
PerformingEntitlement	*ohne Story*	RegistrationReceived
PerformingEntitlement	*ohne Story*	CheckingDeclaration

© Swissdec 2026

Aktueller Zustand	Story	Gemeldeter Zustand
PerformingEntitlement	*ohne Story*	VerifyingClaim
PerformingEntitlement	*ohne Story*	CommunicationStopped
CommunicationStopped	*ohne Story*	RegistrationReceived
CommunicationStopped	*ohne Story*	CheckingDeclaration
CommunicationStopped	*ohne Story*	VerifyingClaim
CommunicationStopped	*ohne Story*	PerformingEntitlement
CommunicationStopped		*[END]*

Tabelle 7.1. Zustandstabelle

7.4 Datenstruktur der initialen Meldung

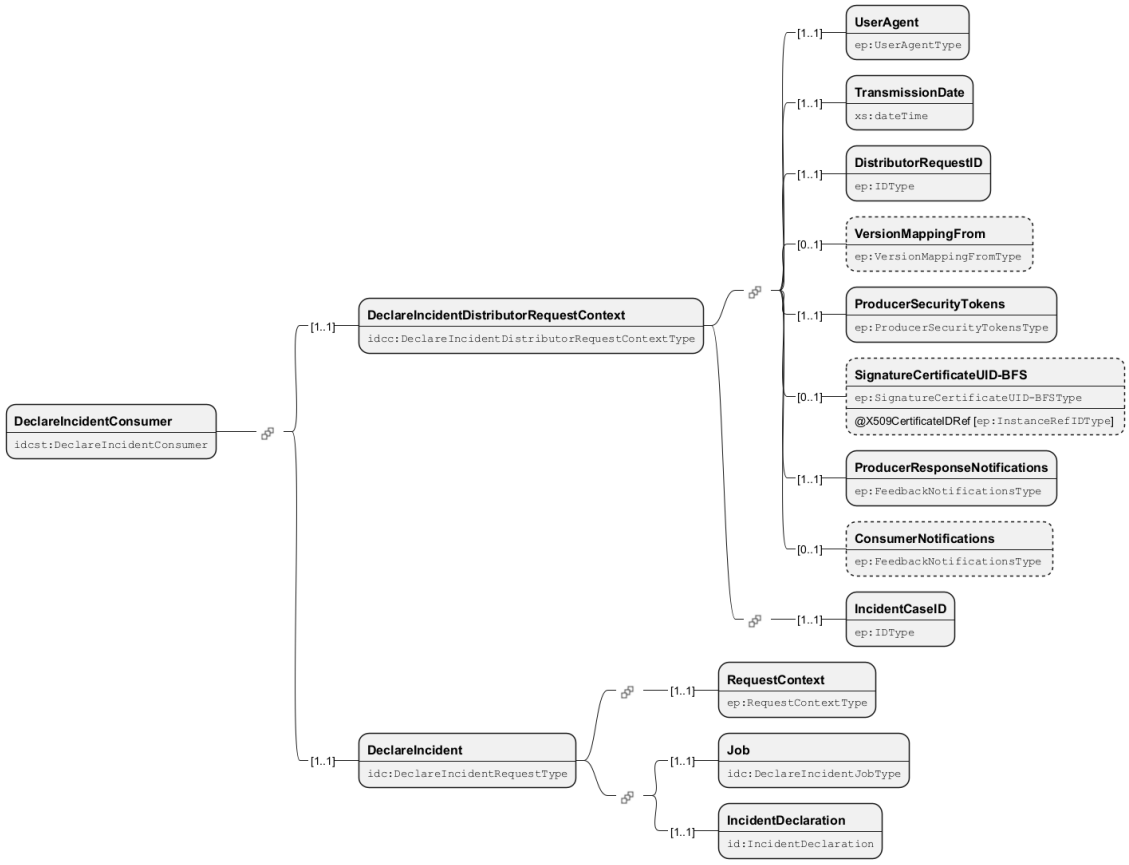


Abbildung 7.3. Schemabild DeclareIncidentConsumer

Feldname	Beschreibung	Typ
DeclareIncidentDistributorRequestContext		idcc DeclareIncidentDistributorRequestContextType
DeclareIncident		idc DeclareIncidentRequestType

Tabelle 7.2. Feldbeschreibungen DeclareIncidentConsumer

© Swissdec 2026

Feldname	Beschreibung	Typ
UserAgent	Beschreibung der wesentlichen System-Identifikation-Daten	ep UserAgentType
CompanyName	Beschreibung der wesentlichen Unternehmensdaten	ep NotEmptyStringType
TransmissionDate	Übertragungszeitpunkt	xs dateTime
RequestID	AuftragsID	ep IDType
LanguageCode	Sprachcode	ep LanguageCodeType
MonitoringID	Die MonitoringID erlaubt es, auf den Testsystemen Übermittlungen den Benutzern zuzuweisen. Sie ist in der Produktion überflüssig.	ep MonitoringIDType

Tabelle 7.3. Feldbeschreibungen RequestContext

Feldname	Beschreibung	Typ
Addressees		idc DeclareIncidentAddresseesType
TestCase		ep EmptyType

Tabelle 7.4. Feldbeschreibungen Job



Abbildung 7.4. Schemabild Addressees

Feldname	Beschreibung	Typ
Addressee	Adressierung der Endempfänger	ep AddresseeJobType

Tabelle 7.5. Feldbeschreibungen Addressees

7.5 Datenstruktur der initialen Antwort



Abbildung 7.5. Schemabild DeclareIncidentConsumerResponse

Feldname	Beschreibung	Typ
DeclareIncidentResponse	Initialer Status des Falls.	idc DeclareIncidentAddresseeSuccessJobStateType

Tabelle 7.6. Feldbeschreibungen DeclareIncidentConsumerResponse

© Swissdec 2026

7.6 Datenstruktur der Synchronize-Meldung

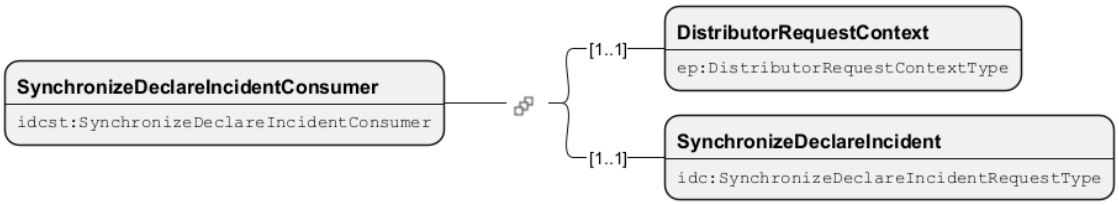


Abbildung 7.6. Schemabild SynchronizeDeclareIncidentConsumer

Feldname	Beschreibung	Typ
DistributorRequest-Context		ep DistributorRe-questContextType
SynchronizeDecla-reIncident		idc SynchronizeDe-clareIncidentRe-questType

Tabelle 7.7. Feldbeschreibungen SynchronizeDeclareIncidentConsumer

7.7 Datenstruktur der Synchronize-Antwort



Abbildung 7.7. Schemabild SynchronizeDeclareIncidentConsumerResponse

Feldname	Beschreibung	Typ
SynchronizeDecla-reIncidentResponse		idc SynchronizeDe-clareIncidentConsu-merType

Tabelle 7.8. Feldbeschreibungen SynchronizeDeclareIncidentConsumerResponse

© Swissdec 2026

8 Use Cases

8.1 UC001 Initiale Meldung empfangen

Use Case Diagramm: siehe Abbildung 3.1, „Use Cases Initiale Meldung“ .

Kurzbeschreibung	Der Distributor schickt gemäss Job die gefilterte Meldung des Unternehmens an den Endreceiver. Dieser prüft, quittiert und übergibt die Daten zur Freigabe durch den Absender (Buchhalter des Unternehmens)
Akteure	Sendersystem, Distributor, Endreceiver
Auslöser	Der Distributor hat eine Meldung vom Sendersystem erhalten und gemäss Adressierung und Job die gefilterten Daten an den Endreceiver transferiert.
Vorbedingungen	Der Distributor hat die Meldung validiert und plausibilisiert (gem. Swissdec-Standard /[[ACKNOTIF]]). Der Endreceiver ist bereit zum Empfangen einer Meldung.
Nachbedingungen	Der Endreceiver hat die Daten für den Abschnitt 8.5, „UC005 Synchronisieren“ bereitgestellt und die Antwort mit den Credentials an den Distributor zurückgeliefert.
Included Use Cases	Abschnitt 8.2, „UC002 Duplikate behandeln“, Abschnitt 8.15, „UC015 Security anwenden“
Standardablauf	1. Die vom Distributor gefilterten Meldungsdaten werden empfangen. Eine Validierung der Daten *kann* durchgeführt werden. 2. Die zusätzlichen Distributor-Transportdaten müssen gemäss Distributorkopplung geprüft werden. (Achtung: Dublettendetektierung [a], ...). Anomalien in der Meldung können geprüft werden. 3. Die Daten werden gemäss Swissdec-Standard geprüft (Security und Akzeptanz). 4. Sichern der Meldung. Alle personenbezogenen Daten werden gemäss Datenschutz gesichert. 5. Credentials mit Key und Password bilden. 6. Die Rückantwort zum Distributor bzw. Unternehmen wird gemäss Swissdecstandard (inklusive der vom Distributor bereits erstellten Nachrichten `<ProducerResponseNotifications>` [[ACKNOTIF]]) aufgebaut.
Alternative Abläufe	{Schritt 2 und 3 Reihenfolge} Schritt 3 darf auch vor Schritt 2 durchgeführt werden. {Schritt 3: Die Stammdaten des Kunden sind nicht verfügbar} Die Meldung kann ungeprüft mit „Akzeptiert“ quittiert werden und eine Prüfung gegen die Stammdaten erfolgt später (manuell oder automatisch). Damit lässt sich eine hohe Verfügbarkeit erzielen. {Schritt 3: Die Meldung wurde vom Sendersystem mittels SUA Zertifikat signiert} Der Distributor liefert die SUA Zertifikatsinformationen im `*Distributor-RequestContext/SignatureCertificateUID-BFS` mit. {Schritt 5: Dublette wurde erkannt} Vorgehen gemäss Abschnitt 8.2, „UC002 Duplikate behandeln“ {Schritt 1: geplanter Unterbruch / Dienst ist nicht verfügbar}

© Swissdec 2026

	Die Information zum Unterbruch (von-bis) wurde bereits mittels Abschnitt 8.18, „UC018 Erreichbarkeit prüfen“ an den Distributor übermittelt. In dieser Zeit werden dann Response-Meldungen (mit dieser Unterbruchinformation) vom Distributor direkt an die anfragende Lohnbuchhaltung zurückgegeben. {Schritt 1: ungeplanter Unterbruch / Dienst ist nicht verfügbar} In dieser Zeit werden dann Response-Meldungen (unbestimmte Dauer) vom Distributor direkt an die anfragende Lohnbuchhaltung zurückgegeben.
Fehlerliste	Fachliche Fehler: • Die Meldung verstösst gegen die Akzeptierungsregeln gemäss Swissdecstandard (z.B. Kunde ist bei der Institution unbekannt) Technische Fehler: • Meldung ist nicht schemavalid • Meldung kann nicht entschlüsselt werden • Meldung ist zu gross • usw.

[a]Auch bei Dubletten werden die gesamten Daten übermittelt

Tabelle 8.1. UC001 Initiale Meldung empfangen

Bemerkungen

Wie ist das Verhalten beim Senden von mehreren **unterschiedlichen** Meldungen eines Unternehmens, **bevor** eine dieser Meldungen abgeschlossen wird (Abschnitt 8.5, „UC005 Synchronisieren“)?

- **Jede Übermittlung wird eigenständig behandelt.** Der Benutzer (Unternehmen) entscheidet, welche Meldung er abschliesst oder löscht.

Dabei können Anomalien nicht verhindert werden.

Nach einer ersten Meldung könnte zum Beispiel eine Substitutionsmeldung vorliegen, die vom Benutzer abgeschlossen wird. Zusätzlich wird noch die erste Meldung gelöscht, die substituiert werden sollte, d.h. es existiert eine Substitution ohne einen Vorgänger.

Funktion: „Meldungsleichen“ [1] werden nach Ablauf einer bestimmten Zeit gelöscht.

Probleme bzw. Verwirrung entstehen bei einer Freigabe-Mahnung oder einer späteren automatischen Konsumation von diesen „Meldungsleichen“. Vorsicht: Gemäss Swissdecschema kann der Prozess vom Endreceiver gesteuert werden.

Die Completionapplikation (s. Abschnitt 8.11, „UC011 Completion“) *sollte* dem Benutzer einen Hinweis über die anstehenden und freigegebenen Meldungen liefern. Damit werden überflüssige Supportanfragen vermieden.

Nach einer Meldungsfreigabe mittels Completion **muss** eine weitere Meldungsübermittlung immer zulässig sein (z.B. Substitutionsmeldung). Optional könnte in der Quittung dazu eine Bemerkung stehen (Info oder Warning).

Im Moment wird auf der Transportebene mit möglichst wenigen Einschränkungen gearbeitet. Evtl. könnte dies später fachliche Probleme erzeugen (z.B. Kunde schickt Daten aus zwei Systemen (Angestellte und Geschäftsleitung) [2]).

[1]Eine übermittelte Meldung (m2m; Receiver), die nie freigegeben wurde (h2m; Completion)
[2]Achtung: Teillieferungen sind im Standard nicht explizit definiert, d.h. eine Lösung muss mittels separater Vertragsidentifikation erfolgen.

© Swissdec 2026

8.2 UC002 Duplikate behandeln

Kurzbeschreibung	Die Dubletten werden mittels einer Hashwert-Berechnung bzw. eines Vergleiches bereits auf dem Distributor erkannt. Sinnvollerweise werden nichtfachliche, zeitbezogene Daten bei der Hashwertberechnung nicht berücksichtigt.
Akteure	Distributor, Endreceiver
Auslöser	Eine Meldung wird empfangen, die bereits zuvor übermittelt wurde.
Vorbedingungen	Eine identische Meldung wurde bereits zuvor an den Endreceiver übermittelt.
Nachbedingungen	Das Auftreten von Dubletten wird vom Distributor mit der `*DistributorRequestContext/Duplicate` an den Endreceiver signalisiert.
Included Use Cases	-
Standardablauf	1. Der Distributor erkennt eine Dublette und markiert dies im DistributorRequestContext/Duplicate, um es dem Endempfänger zu signalisieren. 2. Der Endempfänger erkennt die Übermittlung einer Dublette anhand der Distributor-Information. 3. Es **muss** immer ein neuer Fall eröffnet werden. Entsprechend werden neue Credentialsinformation mit Key und Password zurückgegeben. Der alte, bestehende Fall *kann* gelöscht werden. Das Vorgehen im Umgang mit Dubletten bleibt grösstenteils dem Endempfänger überlassen. Der Umgang mit Dubletten ist unabhängig von der Freigabe im Abschnitt 8.11, „UC011 Completion“ . Eine erneute Übermittlung und Freigabe muss immer möglich sein. [a]
Alternative Abläufe	1. Das Duplikat ist mit einer unbekannten LastDistributorRequestID gekennzeichnet. In diesem Fall wird die Duplikatsmeldung ignoriert und die Meldung wie eine neue Meldung behandelt und für die Synchronisierung bereitgestellt. 2. Der Endempfänger erhält eine Meldung mit bereits bekannter DeclarationID aber ohne Duplikat-Kennzeichnung. Diese Meldung *kann* wie ein Duplikat behandelt oder abgelehnt werden.
Fehlerliste	Fachliche Fehler: • Bei dieser Art der Detektierung können nur technische (Byte) Dubletten erkannt werden.

[a] Evtl. wurde in der ersten Übermittlung und Freigabe etwas vergessen zu ergänzen.

Tabelle 8.2. UC002 Duplikate behandeln

Elemente, die bei der Hashwertberechnung nicht einbezogen werden:

- `{urn:ch:swissdec:basis:v1:*:components}TransmissionDate`
- `{urn:ch:swissdec:basis:v1:*:components}RequestID`
- `{urn:ch:swissdec:basis:v1:*:components}CreationDate`
- `Attribut {urn:ch:swissdec:basis:v1:*:components}addresseeID`
- `Attribut {urn:ch:swissdec:basis:v1:*:components}addresseeIDRef`

Der Distributor meldet dem Endreceiver eine Dublette mithilfe von `Duplicate`.

Einschränkungen der Dublettenerkennung:

- Bei dieser Art der Detektierung können nur technische (Byte) Dubletten erkannt werden.

© Swissdec 2026

- Unterschiedliche Reihenfolge der Personen, Löhne, etc.. Fachlich wäre dies ein Duplikat, technisch sind die Daten aber unterschiedlich.
- Bei paralleler Übermittlung ist unter Umständen beim Eintreffen der einen Meldung die Prüfung der anderen noch nicht abgeschlossen. Dadurch fehlt der Hashwert und es wird kein Duplikat erkannt.
- Je nach Standard kann eine solche Dublettenerkennung nicht oder nicht vollständig implementiert werden.

8.3 UC003 Testmeldung empfangen

Kurzbeschreibung	Testmeldungen dienen zur **Lokalisierung** von Problemen in der **produktiven Übermittlungskette**. Dabei werden Meldungen durch die gesamte Automatisierungskette geschleust, ohne einen echten Geschäftsvorfall anzustossen. Es werden keine Folgeprozesse ausgelöst, und keine Aktionen vom Endemfpägnersystem vorgenommen.
Akteure	Sendersystem, Distributor, Endreceiver, Endemfpängersystem
Auslöser	Das Sendersystem sendet eine als Test markierte Meldung mit dem Element `<TestCase/>`.
Vorbedingungen	Die Meldung ist als Testfall gekennzeichnet.
Nachbedingungen	Die Testmeldung wurde empfangen und verarbeitet, ohne einen produktiven Geschäftsvorfall auszulösen. Der Endreceiver **muss** das TestCase-Element ebenfalls in der Response setzen. Alle weiteren Operationen sowie die Responses dazu **müssen** als Testcase markiert sein.
Included Use Cases	-
Standardablauf	1. Die Meldung wird mit dem Element `<TestCase/>` empfangen. 2. Die Meldung wird wie eine normale Meldung verarbeitet, jedoch ohne produktive Auswirkungen. 3. Die Response **muss** ebenfalls als Testfall markiert zurückgesendet.
Alternative Abläufe	-
Fehlerliste	-

Tabelle 8.3. UC003 Testmeldung empfangen

Verwendungszweck und Einschränkungen:

Dieser Use Case soll nur in Ausnahmefällen eine Verwendung finden. Als Demo- oder Entwicklungssystem darf er nicht genutzt werden. Für diese Zwecke stehen die Swissdec Testsysteme zur Verfügung.

Das Ziel ist es, möglichst viele potenzielle Endbenutzer zu einer elektronischen Meldung zu motivieren.

Aus diesem Grund soll dieser Use Case nun von den Endbenutzern vermehrt genutzt werden. Damit können diese erste Erfahrungen mit ihrer produktiven Meldung machen, ohne damit einen echten Geschäftsfall zu generieren.

Grundsätzlich sollen dabei nur **korrekte** und **vollständige** Lohnmeldungen geschickt werden. Einzig das optionale Containerelement `<TestCase/>` muss vom Sendersystem bzw. Distributor zusätzlich übermittelt werden. Der Endreceiver **muss** dieses ebenfalls in der Response setzen.

Wichtig: Sobald eine Meldung als Testcase gekennzeichnet ist, **müssen** auch alle folgenden Operationen sowie die Responses dazu als Testcase markiert sein.

© Swissdec 2026

8.4 UC004 Ersatzmeldung verarbeiten

Kurzbeschreibung	Meldungen, die das Element `<Substitution>` enthalten, werden vom Unternehmen als Ersatz für bereits übermittelte und freigegebene Meldungen verschickt. Hierfür werden bestimmte IDs der vorangegangenen (alten) Meldung eingebunden [[CONTAINERXSD]]. Anhand der IDs **müssen** die Daten der alten Meldung identifiziert und durch die Daten der neuen Meldung **ersetzt** werden können. Ansonsten verhält sich der Ablauf wie bei einer gewöhlichen Meldung.
Akteure	Unternehmen, Distributor, Endreceiver
Auslöser	Eine bereits übermittelte und freigegebene Meldung muss korrigiert werden.
Vorbedingungen	Eine vorangegangene Meldung wurde bereits übermittelt und freigegeben.
Nachbedingungen	Die alte Meldung wurde durch die neue Ersatzmeldung ersetzt.
Included Use Cases	-
Standardablauf	1. Die Meldung mit dem Element `<Substitution>` wird empfangen. 2. Anhand der IDs wird die vorangegangene (alte) Meldung identifiziert. 3. Die Daten der alten Meldung werden durch die Daten der neuen Meldung ersetzt. 4. Die Ersatzmeldung wird wie eine gewöhnliche Meldung weiter verarbeitet.
Alternative Abläufe	-
Fehlerliste	Fachliche Fehler: • Die vorangegangene Meldung konnte anhand der IDs nicht identifiziert werden. ([[ACKNOTIF]])

Tabelle 8.4. UC004 Ersatzmeldung verarbeiten

8.5 UC005 Synchronisieren

Nach jeder Meldung wird mit dem Aufruf `SynchronizeConsumer` der Fall synchronisiert. Dies kann, muss aber nicht, folgende Elemente beinhalten:

- Abschnitt 8.6, „UC006 Prozesskontrolle durchführen“
- Abschnitt 8.7, „UC007 Fall schliessen“
- Abschnitt 8.8, „UC008 Stories verarbeiten und quittieren“
- Abschnitt 8.9, „UC009 Stories melden“

Bei der Synchronisierung werden Datenpakete (Stories) ausgetauscht. Das Ziel ist immer die Schliessung des Falls, die Stories dienen dazu, alle nötigen Informationen dazu auszutauschen.

Kurzbeschreibung	Der Fall wird synchronisiert.
Akteure	Sendersystem, Distributor, Endreceiver
Auslöser	Der Akteur will mit einem Empfänger Stories austauschen. Abhängig vom Standard findet der Austausch nur vom Endempfänger zum Sendersystem statt oder bidirektional.
Vorbedingungen	• Abschnitt 8.1, „UC001 Initiale Meldung empfangen“ wurde erfolgreich durchgeführt • Alle Identifikatoren für den Fall sind korrekt
Nachbedingungen	Der Fall wurde zwischen Sender und Empfänger synchronisiert. Falls die Synchronisierung nicht abgeschlossen ist, ist der Sender darüber in Kenntnis gesetzt worden.

© Swissdec 2026

Included Use Cases	Abschnitt 8.6, „UC006 Prozesskontrolle durchführen“ Abschnitt 8.7, „UC007 Fall schliessen“ Abschnitt 8.8, „UC008 Stories verarbeiten und quittieren“ Abschnitt 8.9, „UC009 Stories melden“ Abschnitt 8.15, „UC015 Security anwenden“
Standardablauf	1. Das Sendersystem will mit dem Endreceiver den aktuellen Prozess synchronisieren. 2. Der Endreceiver verarbeitet die neu erhaltenen Stories. Abschnitt 8.8, „UC008 Stories verarbeiten und quittieren“ 3. Der Endreceiver stellt Stories in der Response zur Verfügung. Abschnitt 8.9, „UC009 Stories melden“ 4. Der Endreceiver gibt den aktuellen State des Prozesses zurück. 5. Der Endreceiver quittiert alle für diesen Fall erhaltenen Stories.
Alternative Abläufe	1. Das Element `*DistributorRequestContext/SignatureCertificateUID-BFS` ist vorhanden. Der Endreceiver prüft neben den Credentials auch die Zertifikatsinformationen aus der SUA Signatur. 2. Der Akteur wählt keinen Fall und keine Stories aus, die er übermitteln will und sendet ein leeres Synchronize. 3. Der Endreceiver analysiert alle offenen Fälle dieses Vertrages und sammelt die DomainIDs im Available Element. 4. Der Akteur analysiert die erhaltene Antwort, und synchronisiert die Fälle aus dem Available mittels Standardablauf. **Für ELM:** • Der Endempfänger stellt zur Beantwortung seiner Fragen eine Webseite zur Verfügung (Completion Abschnitt 8.11, „UC011 Completion“). Dazu werden die URL, die Login-Credentials sowie ein allfälliges Ablaufdatum derselben zurückgeliefert. Hat der Benutzer die Completion erfolgreich durchgeführt, antwortet der Endempfänger auf das nächste Synchronize mit einer Quittung.
Fehlerliste	Fachliche Fehler: • Die Meldung verstösst gegen die Plausibilisierungsregeln • Es wird kein Fall mit dem zu synchronisierenden Fall gefunden Technische Fehler: • Fehler beim Signieren oder Verschlüsseln • Der Endreceiver ist nicht erreichbar • Die vom Sendersystem aufbereitete Meldung entspricht nicht dem Schema (Validität nicht gegeben) • Die vom Sendersystem aufbereitete Meldung ist nicht identifizierbar (ungültige Identifikatoren)

Tabelle 8.5. UC005 Synchronisieren

© Swissdec 2026

8.5.1 Spezielle Anforderungen

8.5.1.1 Unique StoryID

Die StoryIDs müssen Fallübergreifend eindeutig sein.

8.5.1.2 Available (Neue Informationen verfügbar)

Mit dem Available Element kann der Endempfänger signalisieren, dass unter dem gemeldeten Vertrag neue Informationen zu bestehenden Cases vorhanden sind.

8.5.1.3 OpenCase

Mit dem OpenCase Element in der Response kann der Endempfänger dem Transmitter signalisieren, dass ein neuer Case eröffnet wurde. Dieser wird jeweils mit einer Prozessspezifischen Identifikation angegeben.

Anschliessend kann der Transmitter den effektiven Case mittels dieser Identifikation synchronisieren.

8.5.1.4 Leere Synchronisierung

Mit einem leeren Synchronize (bestehenden aus den Elementen RequestContext, Sender und Addressee) wird beim Endempfänger überprüft, ob Aktualisierungen zu bestehenden Cases vorhanden sind.

8.6 UC006 Prozesskontrolle durchführen

Kurzbeschreibung	Bei jeder Synchronisierung **muss** das Empfängersystem den laufenden Prozess kontrollieren, d.h. überprüfen, was auf beiden Seiten noch offen ist: • Welche Stories werden vom Endreceiver noch erwartet (`AwaitStory`) • Hat der Endreceiver noch Stories aus anderen Prozessen aus dem selben Vertrag, die er dem Sendersystem liefern muss, müssen diese mittels `Available` zurückgegeben werden. • Hat der Endreceiver noch Prozesse, bei welchem der aktuelle State vom Sendersystem nicht bestätigt wurde, müssen diese mittels `Available` zurückgegeben werden. • Sind weitere Änderungen notwendig?
Akteure	Sendersystem, Distributor, Endreceiver
Auslöser	Das Sendersystem führt eine Synchronisierung durch.
Vorbedingungen	• Abschnitt 8.1, „UC001 Initiale Meldung empfangen“ wurde erfolgreich durchgeführt • Alle Identifikatoren für den Fall sind korrekt
Nachbedingungen	Der Endreceiver verarbeitet neue Stories und aktualisiert den Status des Falles. Bei einem Fehlschlag: • Fehlermeldung
Included Use Cases	keine
Standardablauf	1. Das Sendersystem führt eine Synchronisierung durch (siehe Abschnitt 8.5, „UC005 Synchronisieren“). 2. Der Endempfänger überprüft, ob in der Response AwaitStory-Elemente mitzusenden sind. **Achtung!** Diese Funktion existiert nicht in allen Swissdec-Standards.

© Swissdec 2026

	3. Der Endempfänger überprüft, ob in der Response Available-Elemente zu senden sind. **Achtung!** Diese Funktion existiert nicht in allen Swissdec-Standards.
Alternative Abläufe	keine
Fehlerliste	Analog zu Abschnitt 8.1, „UC001 Initiale Meldung empfangen“, Abschnitt 8.5, „UC005 Synchronisieren“

Tabelle 8.6. UC006 Prozesskontrolle durchführen

8.7 UC007 Fall schliessen

Kurzbeschreibung	Ein Fall wird vom Endreceiver abgeschlossen. Dieses Vorgehen kann je nach Swissdec-Standard unterschiedlich sein und wird in den Richtlinien für Lohndatenverarbeitung fachlichen Richtlinien genauer beschrieben.
Akteure	Sendersystem, Distributor, Endreceiver
Auslöser	Seitens Endreceiver ist ein Fall abgeschlossen.
Vorbedingungen	• Es besteht ein Fall, der das Sendersystem und den Endreceiver betrifft. • Der Endreceiver hat den Status des Falles auf den finalen Wert gesetzt. • Der Endreceiver hat keine Stories mehr zu liefern (`<Available>`)
Nachbedingungen	• Der Fall wurde von beiden Seiten (Sendersystem und Endempfänger) geschlossen. • Der Fall wurde vom Sendersystem archiviert.
Included Use Cases	Abschnitt 8.5, „UC005 Synchronisieren“ , Abschnitt 8.6, „UC006 Prozesskontrolle durchführen“
Standardablauf	1. Der Endreceiver erhält ein SynchronizeConsumer auf einen abgeschlossenen Fall. 2. Der Endreceiver gibt den finalen Status zurück.
Alternative Abläufe	**Der Endreceiver verlangt weitere Stories** {nach Schritt 2} 1. Das Sendersystem lässt den Fall offen und bearbeitet es mit weiteren Synchronisierungen, bis es erneut geschlossen werden kann. (Abschnitt 8.5, „UC005 Synchronisieren“)
Fehlerliste	Technische Fehler gemäss Abschnitt 8.5, „UC005 Synchronisieren“ Fachliche Fehler gemäss den Richtlinien

Tabelle 8.7. UC007 Fall schliessen

8.8 UC008 Stories verarbeiten und quittieren

Kurzbeschreibung	Das Sendersystem meldet eine oder mehrere Stories an den Endreceiver
Akteure	Sendersystem, Distributor, Endreceiver
Auslöser	Das Sendersystem hat Informationen, die an den Endempfänger gelangen sollten.

© Swissdec 2026

Vorbedingungen	• Der Fall wurde dem Endreceiver erfolgreich gemeldet. • Identifikatoren sind bekannt und korrekt gesetzt.
Nachbedingungen	• Der Endreceiver hat die Stories empfangen und quittiert.
Included UCs	Abschnitt 8.5, „UC005 Synchronisieren“ , Abschnitt 8.6, „UC006 Prozesskontrolle durchführen“
Standardablauf	1. Der Endreceiver erhält die Stories vom Distributor in einer SynchronizeConsumer Operation und verarbeitet die erhaltenen Stories. 2. Die StoryIDs aller vom Endreceiver empfangenen Stories (vom aktuellen Fall), werden im Element <CaseContext><ReceivedStoryIDs> in der SynchronizeConsumerResponse eingetragen. 3. Die Response wird korrekt aufbereitet (Abschnitt 8.5, „UC005 Synchronisieren“)
Alternative Abläufe	*Keine Stories erhalten* 1. Der Endreceiver quittiert nur erhaltene Stories des aktuellen Fall, hat dabei aber keine neuen Stories zu verarbeiten. Der Request enthält nur einen aktualisierten CaseContext aber keine weiteren Stories. 2. Den Empfang quittieren
Fehlerliste	1. Es gibt Probleme bei der Übermittlung. Die Übermittlung *kann* wiederholt werden. 2. Im Element <CaseContext>/<SuppressedSenderStoryIDs> sind StoryIDs vorhanden: Die Story mit diesen IDs können vom Endreceiver nicht empfangen werden. (z.B: Mapping) Die IDs müssen für jeden folgenden SynchronizeConsumer des aktuellen Prozesses versandt werden. 3. Im Element <CaseContext>/<SuppressedInstitutionStoryIDs> sind StoryIDs vorhanden: Die Story mit diesen IDs können vom Transmitter nicht empfangen werden. (z.B: Mapping) Die IDs müssen für jeden folgenden SynchronizeConsumer des aktuellen Prozesses versandt werden, so wird dem Endreceiver signalisiert, dass der Transmitter diese Story nicht unterstützt. 4. Es gibt fachliche Probleme bei der Auswahl der Stories. Im CaseContext können Info und Warning Notification gesetzt werden. Diese können mit der StoryID versehen werden, so dass der Sender informiert werden kann, welche Story Probleme bereitet.

Tabelle 8.8. UC008 Stories verarbeiten und quittieren

8.9 UC009 Stories melden

Kurzbeschreibung	Anfrage des Sendersystems, um Stories zu erhalten.
Akteure	Sendersystem, Distributor, Endreceiver
Auslöser	Aufgrund einer Synchronisationsanfrage hat der Endreceiver neue Stories für das Sendersystem.
Vorbedingungen	• Der Fall wurde dem Endreceiver erfolgreich gemeldet.

© Swissdec 2026

	• Identifikatoren sind bekannt und korrekt gesetzt.
Nachbedingungen	• Der Endreceiver stellt die Stories dem Sender zur Verfügung.
Included UCs	Abschnitt 8.5, „UC005 Synchronisieren“ , Abschnitt 8.6, „UC006 Prozesskontrolle durchführen“
Standardablauf	*Das Sendersystem hat noch keine Stories* 1. Das Sendersystem schickt via Distributor eine Anfrage, um die ersten Stories zu bekommen. (Abschnitt 8.5, „UC005 Synchronisieren“) 2. Der Endreceiver antwortet mit einem ersten Satz von Stories. *Das Sendersystem hat bereits Stories und sendet diese mit Antworten* 1. Das Sendersystem schickt via Distributor die bekannten Stories mit Antworten. 2. Der Endreceiver verarbeitet die Stories mit den Antworten. 3. Aufgrund der gelieferten Stories (und den Antworten) werden neue Stories erstellt und an das Sendersystem zurückgeschickt.
Alternative Abläufe	1. Der Endreceiver muss jeweils alle verfügbaren Stories für das Sendersystem zur Verfügung stellen, welche im aktuellen ReceivedStoryIDs vom Sendersystem fehlen.
Fehlerliste	1. Es gibt Probleme bei der Übermittlung. Die Übermittlung *kann* wiederholt werden. 2. Es gibt fachliche Probleme bei der Auswahl der Stories.

Tabelle 8.9. UC009 Stories melden

8.10 UC010 Datenflusskontrolle

Kurzbeschreibung	Die Datenflusskontrolle soll verhindern, dass es zu einer Überlastung der betroffenen Systeme der Übermittlung kommt. Der Endreceiver kann beeinflussen, wie viele Informationen er innerhalb einer Response senden will. Das Sendersystem kann den Umfang des Datenflusses kontrollieren, indem nur ein oder wenige Fälle zur Synchronisation ausgewählt werden.
Akteure	Sendersystem, Distributor, Endreceiver
Auslöser	Das Sendersystem fragt mittels Synchronize Operation beim Endreceiver nach aktualisierten Fällen an.
Vorbedingungen	Das Empfängersystem hat zu mehreren Fällen Informationen bereitgestellt.
Nachbedingungen	Der Datenfluss wurde kontrolliert und die Überlastung der Systeme verhindert.
Included Use Cases	-
Standardablauf	1. Der Endreceiver teilt dem Sendersystem eingschränkt mit, zu welchen Fällen er noch Informationen zu liefern hat (<Available>). Der Endreceiver muss nicht zwingend alle Availables liefern, sondern kann eine beschräntke Anzahl von Availables zurückgeben. 2. Das Sendersystem fragt bei den erhaltenen Fällen nach.. 3. Der Endreceiver liefert die Antworten zu den abgefragten Fällen und reagiert für nicht abgefragte Fälle weiterhin mit <Available>.

© Swissdec 2026

Alternative Abläufe	-
Fehlerliste	Technische Fehler: • Überlastung der Systeme durch unkontrollierten Datenfluss.

Tabelle 8.10. UC010 Datenflusskontrolle

8.11 UC011 Completion

Kurzbeschreibung	Die Webseite der Completion eines Endreceivers wird aufgerufen.
Akteure	Sachbearbeiter
Auslöser	Der Akteur möchte die übermittelte Meldung ergänzen und freigeben.
Vorbedingungen	• Der Fall wurde dem Endreceiver erfolgreich gemeldet • Basis-URL, Key und Passwort aus der Story des Endreceivers sind verfügbar • Das <ExpiryDate> des Logins für die Completion ist noch gültig.
Nachbedingungen	• Die Daten sind gesichert oder gelöscht. • Personenbezogene Daten sind gemäss Datenschutz gesichert. • Wurde eine Meldung erfolgreich freigegeben, so ist es im weiteren Verlauf **nicht mehr möglich**, ein `CompletionReleaseIsMissing` zu erhalten. • Bei einem Fehlschlag: Webseite nicht erreichbar: Fehlermeldung
Included UCs	Keine
Standardablauf	1. Der Sachbearbeiter meldet sich mit Key und Password am System an (Login kann sichtbar sein und ***muss*** **vollständig** vorbefüllt werden, d.h. **Key und Password**). Bei korrekter Anmeldung ist es zulässig, den Logindialog zu überspringen und dem Endbenutzer direkt die erste Maske zu zeigen. Ziel ist ein möglichst einfacher Prozess, d.h. die Lohnbuchhaltung startet den Browser mit „Key“ und „Password“ als Parameter in der Completion-URL, und der Endbenutzer muss nur noch den Login-Knopf drücken. Diese Parameternamen **müssen** case-insensitiv sein, d.h. sowohl „key“ als auch „Key“ sind gültig. 2. Der Sachbearbeiter kontrolliert die gesendeten Lohndaten. Dabei sollen nur relevante Summen bzw. Zahlen angezeigt werden. Auf einzelne Personendaten sollte, wenn immer möglich, aus Sicherheitsgründen verzichtet werden. Zusätzlich kann die Meldung ergänzt und gedruckt werden. 3. Die Meldung wird freigegeben. Es ist zulässig, die Completion für einen TestCase aus Abschnitt 8.3, „UC003 Testmeldung empfangen“ durchzuführen, sofern der TestCase für den Sachbearbeiter ersichtlich wird (in den Masken und auf der PDF- oder HTML-Quittung). 4. PDF- oder HTML-Quittung mit Kontrollsummen und den Ergänzungen an den Sachbearbeiter zurückgeben. 5. Löschen von kritischen Daten (gemäss Datenschutz)
Alternative Abläufe	{Schritt 3: löschen statt freigeben} Die Meldung **muss** komplett gelöscht werden, d.h. direkt zum Schritt 5.

© Swissdec 2026

	{Schritt 1: Meldung ist bereits gelöscht} Über den Zustand *soll* eine Information zurückgeben werden, z.B. Meldung ist „freigegeben zur Verarbeitung“ oder ist „gelöscht ohne Freigabe zur Verarbeitung“. Damit werden überflüssige Supportanfragen vermieden. {Schritt 1: ExpiryDate ist abgelaufen} Aus Sicherheitsgründen **muss** die Gültigkeit der Completionlogins überprüft werden. Abgelaufene Lohnmeldungen werden automatisch gelöscht, d.h. direkt zum Schritt 5 Optional kann der Absender (Unternehmen) vor dem Ablauf der Gültigkeit mittels E-Mail erinnert werden. {Schritt 1: Logindialog } Sollte das direkte Login mit Key und Passwort in der URL fehlschlagen, dann **muss** die Möglichkeit bestehen, dass sich der User in einem Logindialog einloggen kann.
Fehlerliste	Technische Fehler: Anmeldung ist nicht möglich

Tabelle 8.11. UC011 Completion

8.11.1 Standardablauf Completion

Zur Verbesserung der Usability wird der vereinfachte Ablauf standardisiert. Dabei könnten auch domänenspezifische Abläufe modelliert werden. Jede Implementierung sollte diesem Ablauf folgen. Ein Buchhalter kann dadurch die verteilten Lohnmeldungen effizient freigeben. Diese Empfehlung ist für eine effiziente und klare Kommunikation zwischen allen Partnern [3] essentiell.

Der Ablauf umfasst minimal folgende Zustände:

- **ReadyForCompletion**

  Die Daten wurden vom Endreceiver erfolgreich empfangen, gesichert und mit akzeptiert quittiert. Die Lohndaten stehen bereit zur Kontrolle und Freigabe. Die übermittelten Lohndaten können jetzt mittels Completion-Applikation ergänzt werden. Die Lohnmeldung ist nun mit dem Login erreichbar.

- **released**

  Die evtl. ergänzten Lohndaten stehen bereit zur eigentlichen Verarbeitung. Diese Freigabe wird noch mit einem PDF-Dokument (Zusammenfassungsdaten und Ergänzungen) für das Unternehmen (Absender) quittiert.

  Die Lohnmeldung ist nun mit dem Login nicht mehr erreichbar.

- **removed**

  Die übermittelte Lohnmeldung wird gelöscht und es findet keine Verarbeitung statt. Die Lohnmeldung ist nun mit dem Login nicht mehr erreichbar.

Folgende Standardteile *sollen* in der Completion implementiert werden:

[3] Fach- und Technikebene für Unternehmen (Buchhalter), Lohnbuchhaltungshersteller und Empfänger

© Swissdec 2026

1. **Zusammenfassung** (nach dem Login)

   Hier möchte der Unternehmer (Lohnbuchhalter, Absender) die wesentlichen Daten seiner Lohnmeldung überprüfen. Weiter mit Ergänzungen, Freigabe oder löschen.

2. **Ergänzungen**

   Hier kann oder muss der Unternehmer (Lohnbuchhalter, Absender) zusätzliche Angaben machen. Weiter mit Freigabe oder Löschen.

3. **Freigabe**

   Die Freigabe wird noch mit einem PDF oder HTML quittiert.

Es werden keine weiteren Anforderungen an den Ablauf definiert, d.h. GUI-Masken und Abläufe sind nur Empfehlungen. Die Elementlabels sind zum Teil im Swissdecstandard [[CONTAINERXSD]] beschrieben. Eine Anpassung muss über die entsprechende Swissdec Fachgruppe erfolgen.

8.11.2 Spezielle Anforderungen

8.11.2.1 Allgemeines

Die Completionapplikation *sollte* dem Benutzer einen Hinweis über die anstehenden und freigegebenen Lohnmeldungen liefern. Damit werden überflüssige Supportanfragen vermieden.

Nach einer Meldungsfreigabe mittels Completion **muss** eine weitere Meldungsübermittlung immer zulässig sein (z.B. Substitutionsmeldung). Optional könnte in der Quittung dazu eine Bemerkung stehen (Info oder Warning).

8.11.2.2 Completion-URL und Credentials

Das Element `<Credentials>` mit `<Key>` und `<Password>` wird in der XML-Reponse **nicht** in eine URL-kodierte Form gebracht. Dies ist notwendig, da die Web-Eingabeformulare für die Completion keine URL-kodierte Form verlangen und der Benutzer auf diese Weise mit direktem Copy-Paste arbeiten kann, falls es Probleme mit der automatischen Vorbefüllung der Eingabemaske gibt.

Hingegen muss die Completion-URL `<Completion>/<Url>` in eine korrekte, URL-kodierte Form gebracht werden, bevor diese an den Transmitter zurückgeschickt wird.

Verbotene Sonderzeichen

Die aufgeführten Sonderzeichen dürfen **nicht** verwendet werden:

- In URL, Key und Password: "#", "<" und ">".
- In Key und Password: "&" und "?".

Daraus folgt auch, dass URLs **keine** Fragmente enthalten dürfen.

Ausserdem **muss** darauf geachtet werden, dass bei allen in URL, Key oder Password enthaltenen Sonderzeichen die Character Entity Reference (z.B. & anstelle von „&“) verwendet wird, damit es in der XML-Response nicht zu einem Validierungsfehler kommt. Achtung: Es **muss** dringend darauf geachtet werden, dass der Charakter nicht versehentlich doppelt codiert wird!

Ausgangs-URL; im XML so nicht valid:

```
http://www.completion-url.ch/?language=de&info=test
```


© Swissdec 2026

Richtig und im XML valid:

```
http://www.completion-url.ch/?language=de&amp;info=test
```

Falsch, da doppelt codiert:

```
http://www.completion-url.ch/?language=de&amp;amp;info=test
```

Die meisten Entwicklungsplattformen haben in ihren Bibliotheken Tools zur URL-Kodierung. Die URL Kodierung wird in [[RFC3986]] beschrieben.

Aufbau einer URL gemäss[[RFC3986]]:

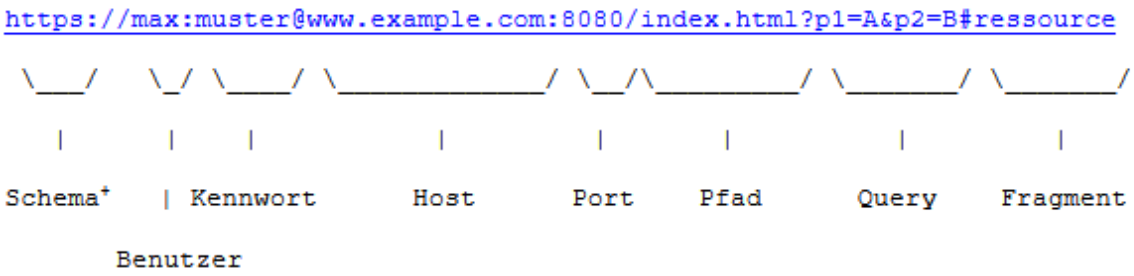


Abbildung 8.1. Aufbau URL gemäss RFC3986

8.11.2.3 Testfall

Wurde eine Meldung mit <TestCase> markiert, dann sind in der Completionapplikation die folgenden Anforderungen zu **erfüllen**:

- Auf **jeder** Seite der Completion ist **klar** ersichtlich, dass es sich um einen Testfall handelt.
- Falls eine Quittung (PDF, etc.) erstellt wird, so muss diese ebenfalls einen **klar** ersichtlichen Hinweis auf den Testfall enthalten.
- Am Ende wird der Benutzer darauf hingewiesen, dass die Daten nicht weiter verarbeitet werden.

8.12 UC012 Dialog Message abwickeln

Kurzbeschreibung	Die Ergänzung/Freigabe wird mittels DialogMessage durchgeführt.
Akteure	Sachbearbeiter
Auslöser	Der Akteur möchte die übermittelte Meldung ergänzen und freigeben.
Vorbedingungen	• Der Fall wurde dem Endreceiver erfolgreich gemeldet • Der Endreceiver hat eine <DialogMessage> Story gesendet
Nachbedingungen	DialogMessage wurde komplett abgewickelt, es ist kein weiterer Aufruf nötig.
Included UCs	-
Standardablauf	1. Der Endreceiver sendet mit der <DialogMessage> Story die zu ergänzenden Daten. 2. Der Sachbearbeiter ergänzt die geforderten Angaben und sendet diese mit der <DialogMessage> Story mittels <Synchronize> an den Endreceiver.

© Swissdec 2026

	3. Der Endreceiver bestätigt mit <ReceivedStoryIDs> den Erhalt der DialogMessage. Nach der Quittierung in den <ReceivedStoryIDs> wird die DialogMessage nicht erneut geliefert.
Alternative Abläufe	DialogMessage nicht abgeschlossen {nach Schritt 3} 1. Der Endreceiver bestätigt mit <ReceivedStoryIDs> den Erhalt der DialogMessage und verlangt weitere Ergänzungsangaben. {Weiter mit Schritt 3}
Detailspezifikation	Anhang D, *Detailspezifikation DialogMessages*
Fehlerliste	Fehler: • Die empfangenen Daten können nicht verarbeitet werden

Tabelle 8.12. UC012 Dialog Message abwickeln

8.13 UC013 Anmeldungsprozess (SubscribeOrganizationConsumer)

Kurzbeschreibung	Ein Sendersystem muss sich bei einem Empfängersystem für einen späteren Übermittlungsprozess anmelden können. Mit diesem Anmeldeprozess wird der zukünftige Informationsaustausch über den Distributor zwischen den Beteiligten in die Wege geleitet.
Akteure	Sendersystem, Distributor, Endreceiver
Auslöser	Ein Sendersystem möchte sich für einen Übermittlungsprozess beim Endreceiver anmelden.
Vorbedingungen	Das Sendersystem verfügt über einen gültigen Vertrag und ein SUA Zertifikat für die Anmeldung.
Nachbedingungen	Die Anmeldung wurde erfolgreich durchgeführt und der zukünftige Informationsaustausch ist eingerichtet.
Included Use Cases	-
Standardablauf	1. Das Sendersystem sendet eine Anmeldungsanfrage an den Distributor. 2. Der Distributor leitet die Anfrage an den Endreceiver weiter. 3. Der Endreceiver verarbeitet die Anmeldung und bestätigt diese. 4. Die Bestätigung wird über den Distributor an das Sendersystem zurückgesendet.
Alternative Abläufe	-
Fehlerliste	-

Tabelle 8.13. UC013 Anmeldungsprozess (SubscribeOrganizationConsumer)

8.14 UC014 Unternehmensauthentifizierung (RegisterOrganizationAuthentication)

Kurzbeschreibung	Verschiedene Prozesse verlangen eine eindeutige Authentifizierung eines Unternehmens beim Endreceiver. Dies bedingt eine Signierung mit SUA-Zertifikat. Ein solches SUA-Zertifikat muss mittels des SUA-Prozesses angefordert werden können.
Akteure	Unternehmen, Sendersystem, Distributor, Endreceiver

© Swissdec 2026

Auslöser	Ein Prozess verlangt eine eindeutige Authentifizierung des Unternehmens mit SUA-Zertifikat.
Vorbedingungen	Das Unternehmen benötigt ein SUA-Zertifikat für die Authentifizierung.
Nachbedingungen	Das SUA-Zertifikat wurde angefordert und die Authentifizierung kann durchgeführt werden.
Included Use Cases	-
Standardablauf	1. Das Unternehmen fordert ein SUA-Zertifikat an. 2. Der Distributor prüft und verifiziert die Angaben (auch gegen BFS) und liefert nur die Vertragsdaten an den Endreceiver. 3. Der Endempfänger prüft die erhaltenen Vertragsdaten und generiert Credentials gemäss Abschnitt 8.1, „UC001 Initiale Meldung empfangen“ . 4. Beim ersten Synchronize gibt der Endreceiver die Adressdaten des Unternehmens gemäss Schema zurück. 5. Der Distributor vergleicht die Daten von Unternehmen, BFS und Endempfänger und versendet bei Erfolg einen Brief mit Passwort an das Unternehmen. 6. Das Unternehmen kann sein SUA Zertifikat lösen.
Alternative Abläufe	-
Detailspezifikation	Anhang C, *Detailspezifikation Swissdec Unternehmens-Authentifizierung SUA*
Fehlerliste	-

Tabelle 8.14. UC014 Unternehmensauthentifizierung (RegisterOrganizationAuthentication)

8.15 UC015 Security anwenden

Kurzbeschreibung	Jede Übermittlung zum Endreceiver muss signiert und verschlüsselt werden. Einzelheiten dazu finden sich in den Dokumenten zur Sicherheit auf Receiverseite[[SECRXPDF]]. Ebenfalls zu beachten sind die zusätzlichen Informationen zur doppelten Signierung (SUA) auf Transmitterseite und deren Auswirkungen auf Seite Endreceiver. Der Transport muss zwingend über HTTPS erfolgen. Das TLS kann mit einem akkreditierten oder einem selbstsignierten Zertifikat verschlüsselt sein. Es wird empfohlen den Transport mittels Two-Way-TLS (mTLS) abzusichern.
Akteure	Distributor, Endreceiver
Auslöser	Eine Meldung wird übermittelt.
Vorbedingungen	Die notwendigen Zertifikate für Transport, Signierung und Verschlüsselung sind vorhanden.
Nachbedingungen	Die Meldung wurde signiert und verschlüsselt übermittelt.
Included Use Cases	-
Standardablauf	1. Die Meldung wird signiert. 2. Die signierte Meldung wird verschlüsselt. 3. Die signierte und verschlüsselte Meldung wird übermittelt.
Alternative Abläufe	-
Fehlerliste	Technische Fehler: • Zertifikat ist ungültig oder abgelaufen.

© Swissdec 2026

	• Signierung oder Verschlüsselung schlägt fehl.

Tabelle 8.15. UC015 Security anwenden

8.16 UC016 Wartungsfenster

Kurzbeschreibung	Das Endreceiversystem **muss** die Möglichkeit haben, in einem `PingConsumerResponse` dem Distributor ein Wartungsfenster mitzuteilen. Wartungsfenster zeigen beabsichtigte Betriebsunterbrüche an, während denen der Distributor keine Meldungen an den Endreceiver weiterleitet und den Benutzer über das Wartungsfenster informiert.
Akteure	Endreceiver, Distributor
Auslöser	Ein geplanter Betriebsunterbruch steht bevor.
Vorbedingungen	Der Endreceiver hat einen geplanten Betriebsunterbruch.
Nachbedingungen	Der Distributor wurde über das Wartungsfenster informiert und leitet während diesem Zeitraum keine Meldungen an den Endreceiver weiter.
Included Use Cases	-
Standardablauf	1. Der Endreceiver registriert ein Wartungsfenster mit dem Element <`PlannedMaintenance`> in der `PingConsumerResponse`. 2. Der Distributor speichert die Information über das Wartungsfenster. 3. Während des Wartungsfensters leitet der Distributor keine Meldungen an den Endreceiver weiter und informiert den Benutzer.
Alternative Abläufe	{Wartungsfenster löschen} Der Endreceiver kann ein bereits registriertes Wartungsfenster mit dem Element <`NoPlannedMaintenance`> löschen.
Fehlerliste	Technische Fehler: • Das Wartungsfenster konnte nicht registriert werden.

Tabelle 8.16. UC016 Wartungsfenster

Da der Distributor keine Meldungen speichert, muss der Benutzer seine Meldung zu einem späteren Zeitpunkt wiederholen.

8.16.1 Notizen zur Operation PingConsumer

Die Operation `PingConsumer` ist eine einfache, synchrone Request Response Kommunikation.

Damit werden folgende Funktionen aus Use Case 018 Abschnitt 8.18, „UC018 Erreichbarkeit prüfen“ abgedeckt:

Die Verfügbarkeit der Systeme (Distributor und die gekoppelten Receiver der Endreceiver) wird periodisch mittels „Polling" durch den Distributor getestet. Der Distributor liefert die aktuelle Systemzeit. Sofern ein Wartungsfenster registriert wurde, wird zusätzlich **eine** geplante Start- und Endzeit angegeben.

Der Endreceiver antwortet mit seinem aktuellen Timestamp. Er kann in der Response optional mit dem Element <`PlannedMaintenance`> ein geplantes Wartungsfenster registrieren oder mit dem Element <`NoPlannedMaintenance`> ein bereits registriertes Wartungsfenster löschen.

Durch die Angabe des Zeitpunktes im Request Element <`NextCheck`> kann der Endreceiver seinerseits den „minimalen" Vorlauf zum Aufschalten des Wartungsfensters ersehen.

© Swissdec 2026

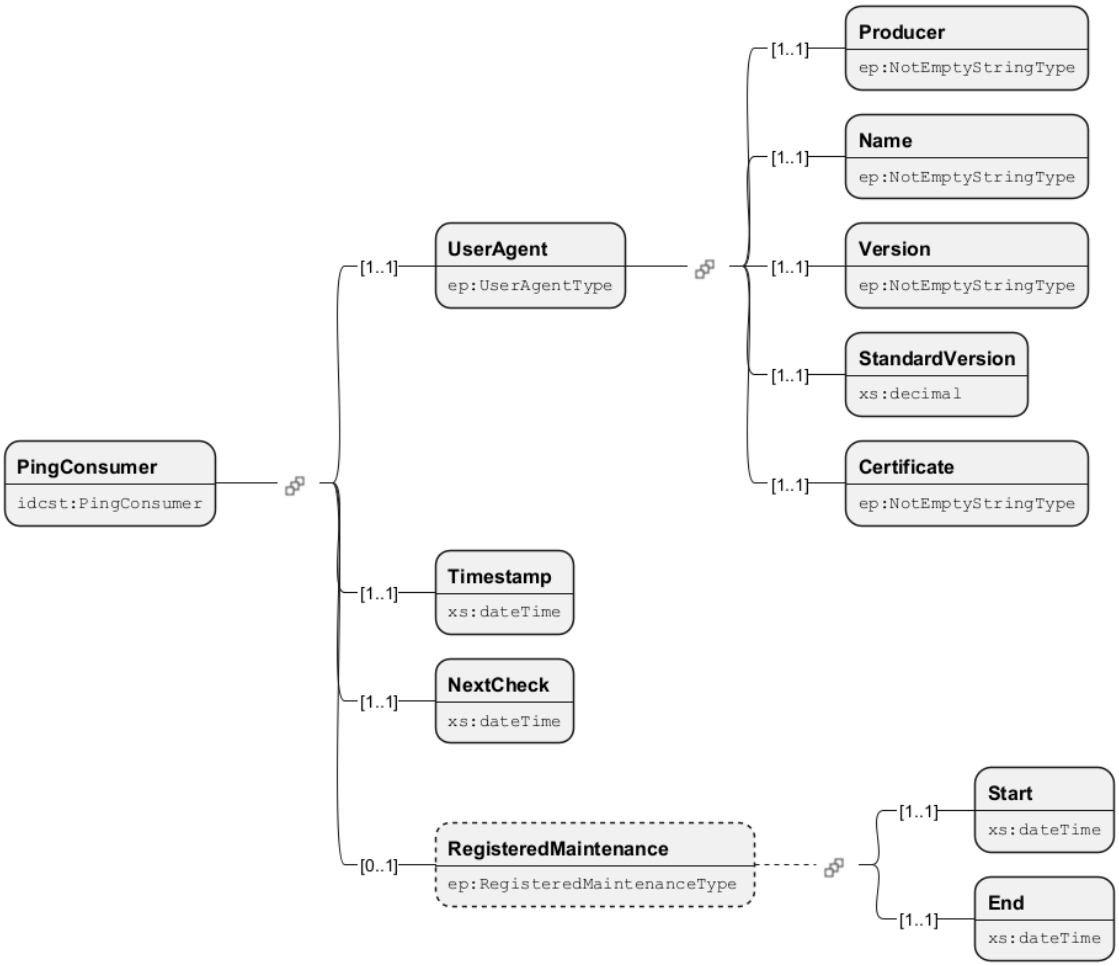


Abbildung 8.2. Schemabild PingConsumer

© Swissdec 2026

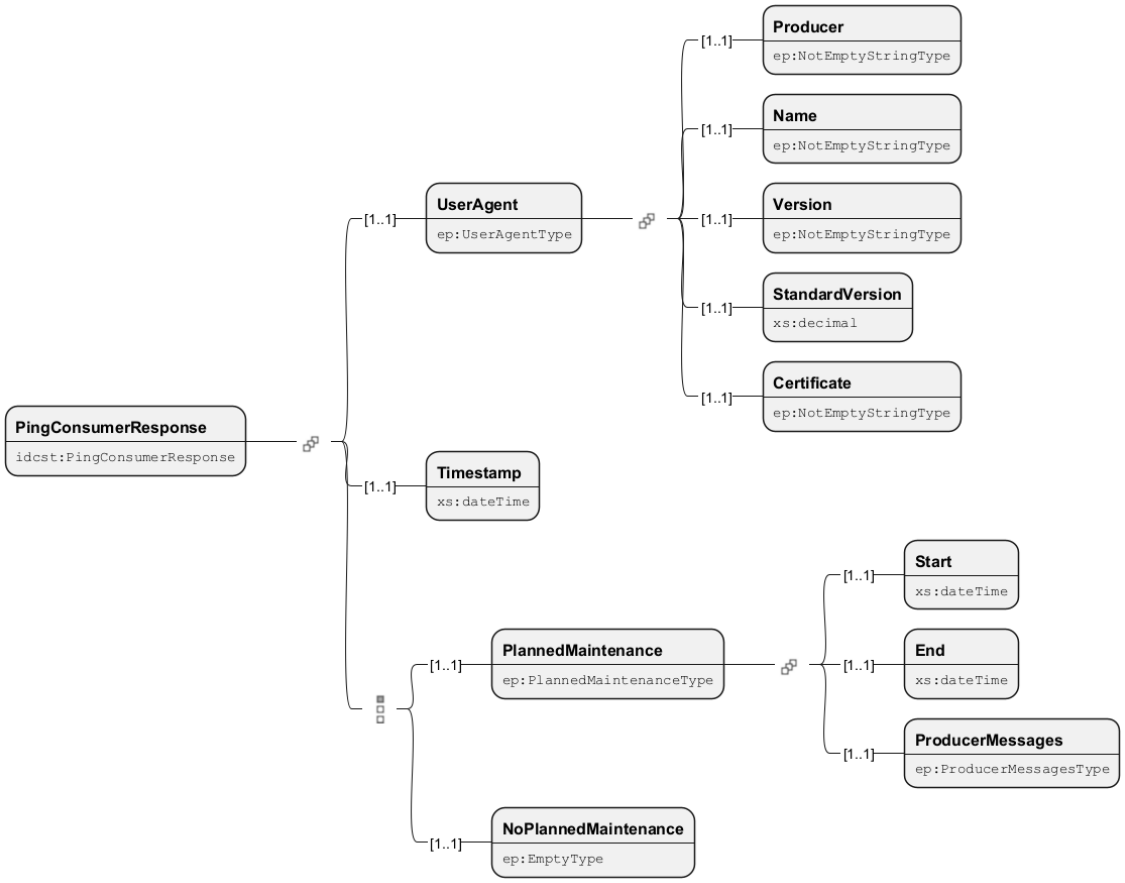


Abbildung 8.3. Schemabild PingConsumerResponse

8.17 UC017 Supportanfrage bearbeiten

Kurzbeschreibung	Ausnahmen, Störfall und andere Probleme behandeln
Akteure	Lohnbuchhalter, IT-Spezialist
Auslöser	Der Lohnbuchhalter oder IT-Spezialist stellt per Mail oder Telefon eine Supportanfrage
Vorbedingungen	Keine
Nachbedingungen	Die Supportanfrage konnte erfolgreich bearbeitet werden
Included Use Cases	Keine
Standardablauf	1. Es wird per Mail oder Telefon eine neue Supportanfrage von einem Lohnbuchhalter oder IT-Spezialist gestellt. 2. Das Problem wird analysiert und beantwortet.
Alternative Abläufe	{nach Schritt 1} 1. Das Problem wird eskaliert und gelangt zum Second oder Third Level Support. {weiter mit Schritt 2}
Fehlerliste	Keine

Tabelle 8.17. UC017 Supportanfrage bearbeiten

© Swissdec 2026

Wichtig ist im Umgang mit Supportfällen, dass die Supportinformationen einheitlich kommuniziert werden. Fehler, Warnungen und Informationen müssen gemäss [[ACKNOTIF]] erstellt und in die Quittung eingefügt werden. Die in diesem Dokument beschriebenen Codes sind verbindlich.

Es **muss** eine Möglichkeit bestehen, im Falle einer Supportanfrage auf die nötigen Informationen für die Behandlung des Problems zuzugreifen, wie zB. mit Hilfe der DeclarationID, der RequestID oder der DistributorRequestID.

8.17.1 Support und Reaktionszeit

Es werden nur technische Aspekte zum Support festgelegt, d.h. hier werden nur Informationsstrukturen für alle Systeme in der Prozesskette definiert. Die organisatorische Verwendung dieser Daten erfolgt in einem separaten Dokument.

Der Support **muss** in den Sprachen Deutsch, Französisch und Italienisch für folgende Bereiche bzw. Akteure erbracht werden:

- Unternehmen und ihre Lohnbuchhaltungshersteller
- Endreceiver Institutionen

D.h. auch Fehlermeldungen sind in den entsprechenden Sprachen auszugeben. Siehe in der Meldung:

`…/RequestContext/LanguageCode`

Zur Bestimmung einer Reaktionszeit werden folgende **Fehlerklassen** definiert

- Critical = 15 Min
- Medium = 4 h
- Uncritical = 1 Tag

Diese Fehlerklassen werden in verschiedenen Systemen (Applikationen, Logfiles, Überwachungstool, …) später entsprechend verwendet.

Zusätzlich **muss** der 2nd Level Support zu den Applikationsentwicklern koordiniert werden.

8.17.2 Spezielle Anforderungen

8.17.2.1 Identifikation eines Geschäftsvorfalls

Der Distributor fügt bei den Meldungen eine Identifikation des Geschäftsvorfalls ein. Für Rückfragen an den Support des Endreceivers **müssen** alle Meldungen eines Geschäftsvorfalls anhand dieser ID zugeordnet werden können. Der Empfänger muss die ID für einen Geschäftsvorfall anzeigen können.

Diese ID dient im Supportfall der Identifikation von Meldungen, die zum selben Geschäftsvorfall gehören. Ein Geschäftsvorfall ist die abgeschlossene Übermittlung einer Meldung. Die Übermittlung einer Ersatzmeldung unterbricht und beendet diesen Geschäftsvorfall. Es beginnt ein eigener, neuer Geschäftsvorfall, für den eine neue ID generiert wird.

Anmerkung

Der Name der Identifikation des Geschäftsvorfalls ist prozess- beziehungsweise operationsabhängig. In den operationsspezifischen Kapiteln wird auf den Namen hingewiesen.

Die Identifikation des Geschäftsvorfalls ist identisch in allen Requests (nicht vorhanden in: Transmitter → Distributor), Responses, Masken und Pdfs, die zu ein und demselben Geschäftsvorfall gehören. Damit kann sie als Fallnummer für den Support verwendet werden.

© Swissdec 2026

Zusätzlich dienen die RequestID und die Distributor-ResponseID - Paare zur Identifikation der einzelnen Requests und Responses innerhalb des Geschäftsvorfalls. Das System muss die RequestID und die ResponseID ebenfalls verwenden und sie der Identifikation des Geschäftsvorfalls zuordnen können.

8.18 UC018 Erreichbarkeit prüfen

Kurzbeschreibung	Die Erreichbarkeit des Endreceivers muss geprüft werden. Dazu wird eine einfache Anfrage an den Endreceiver geschickt. Die Rückantwort des Endreceivers bestätigt die Erreichbarkeit.
Akteure	Distributor, Swissdec Operator, Endreceiver
Auslöser	Zyklische Überprüfung vom Distributor, Swissdec Operator im Störfall
Vorbedingungen	Keine
Nachbedingungen	Keine
Included UCs	Keine
Standardablauf	1. Der Distributor stösst den Erreichbarkeitstest an. 2. Die Anfrage wird an den Endreceiver gesendet. Zusätzlich wird das Intervall des Pollings mitgeteilt. Intervall: zurzeit 30 Minuten (auch während eines Wartungsfensters; Intervall ist damit dynamisch) 3. Der Endreceiver antwortet mit seinem aktuellen Timestamp. Optional kann dem Distributor ein geplantes **Wartungsfenster** (Nichtverfügbarkeit von x bis y) mitgeteilt werden. Diese Funktion **muss** möglich sein. 4. Sollte ein Wartungsfenster gemeldet werden, kann diese Information **automatisch** einem allfälligen Absender (Unternehmen) direkt vom Distributor weitervermittelt werden.
Alternative Abläufe	Keine
Fehlerliste	Fehler: • der Endreceiver ist nicht erreichbar. In diesem Fall sendet der Distributor via Mail eine Warnung an die eigens dafür eingerichte Mailadresse des Endreceivers.

Tabelle 8.18. UC018 Erreichbarkeit prüfen

Der Endreceiver ist **verpflichtet**, den Posteingang der entsprechenden Mailadresse **zu überwachen** und bei Nichterreichbarkeit **die notwendigen Massnahmen zu treffen**.

8.18.1 Verfügbarkeit

Die Betrachtungseinheit umfasst den Distributor und alle gekoppelten Endreceiver, d.h. das Unternehmen erlebt das ganze System als Einheit. Sollte ein Endreceiver nicht in geforderter Qualität betrieben werden, vermindert dieser Empfänger die Zuverlässigkeit des ganzen Systems. Alle Teilnehmer **müssen** sich deshalb auf eine **minimale** Zuverlässigkeit einigen.

Anforderung aus dem Swissdecstandard

- Die gesamte Übermittlung in der 1. Phase: m2m (Machine to Machine) erfolgt in „**Echtzeit**“. (**7x24h Internet-Verfügbarkeit**)

Diese Anforderung hat für den Empfänger folgende Konsequenzen

- auch die Institutionen bzw. ihre Endreceiver **müssen** mindestens zum **Empfangen der Daten einen 7x24h Dienst anbieten**.

© Swissdec 2026

- **Geplante Unterbrüche** [4] müssen an Randzeiten durchgeführt werden und **müssen** vorher angekündigt werden.

- Nach **ungeplantem Unterbruch** *sollten* betroffene Unternehmen, die eine missglückte Übermittlung hatten, automatisch über die erneute Verfügbarkeit des Empfängers benachrichtigt werden (siehe dazu Abschnitt 8.1, „UC001 Initiale Meldung empfangen“). Allerdings gibt es in dieser Version keine automatische Benachrichtigung seitens Distributors, d.h. der Endreceiver kann nur die ihm direkt bekannten Unternehmen benachrichtigen.

- Sollten interne Dienste zur Überprüfung der Akzeptierung **nicht zur Verfügung** stehen, *kann* trotzdem mit einer Akzeptierung (inklusive Completion) quittiert werden. Dies *sollte* mit einer Warning/Notification in der Quittung dem Absender mitgeteilt werden. Führt eine spätere Datenprüfung zur Ablehnung der Meldung, muss diese dem Kunden ausserhalb dieser Systemspezifikation mitgeteilt werden (siehe auch Abschnitt 8.11, „UC011 Completion“: Lohnmeldung freigeben (Completion)).

- Sollte die Completion ebenfalls nicht verfügbar sein, *sollte* dies mit einer Warning/Notification in der Quittung dem Absender mitgeteilt werden.

Zielorientiertes Vorgehen bezüglich des Themas Verfügbarkeit:

Wir möchten eine **kundenorientierte Sicht** einnehmen. Die Verfügbarkeiten der Systeme sind als **zukünftige Zielwerte** zu verstehen. Damit werden die Unternehmen motiviert, ihre Meldungen elektronisch zu übermitteln. Bezüglich Verfügbarkeit ist keine Kontrolle vorgesehen. Deshalb werden hier nur die wesentlichen Richtwerte definiert und entsprechende Grundlagen in den Anhang verschoben.

8.18.2 Definierte Zeitbereiche

- Betriebszeit des gesamten Systems (Distributor, Kommunikation und Endreceiver; m2m Strecke bis zur Quittungs-Response an das Unternehmen)
  - **7 Tage pro Woche mal 24 Stunden**
  - **Spitzenmonate: Januar und Februar**
  - **Quellensteuer, Statistik und «EMA» werden monatlich gemeldet**
  - **Spitzenzeiten: 6 Uhr bis 20 Uhr und von Dezember bis April**
- Wartungsfenster für Korrekturen und Updates
  - **10 Stunden pro Woche**
  - **Ausserhalb der Spitzenzeiten und Spitzenmonate, wenn möglich zwischen 2 Uhr und 5 Uhr morgens**
- Service- und Supportzeit für die Systemteilnehmer (Distributor und seine Endreceiver wie Versicherungen und Behörden)
  - **Montag bis Freitag von 8 Uhr bis 18 Uhr in den Spitzenmonaten**
  - **In den restlichen Monaten gelten die üblichen Bürozeiten**
  - **Support für Wartungsfenster auf Anmeldung**

8.18.3 Definierte Wertebereiche

Ziel ist eine pragmatische Lösung = „lightweight construction“ und „Best Effort“

[4] Gilt für normale Wartungsarbeiten; ausgenommen ist ein Hotfix oder Patch

© Swissdec 2026

Einerseits haben wir pro Unternehmen mit nur einer Übermittlung pro Jahr zu rechnen, d.h. der Endbenutzer hat wenig Erfahrung mit dem gesamten System. Deshalb muss jede Störung [5] vermieden werden. Andererseits möchten Endreceiver mit einem kleineren Kundenstamm keinen erhöhten Aufwand betreiben.

- In den **Spitzenzeiten** *soll* die Verfügbarkeit der Endreceiver (m2m) mindestens **99,52%** sein.
- In den **Randzeiten** *soll* die Verfügbarkeit der Endreceiver (m2m) mindestens **93,00%** sein.

[5] Störung meint hier: Der Lohnbuchhalter des Unternehmens kann *nicht alles* mit *einem* „einfachen" Knopfdruck melden. Im Störfall, d.h. ein Endreceiver ist nicht verfügbar, benötigt er Prozesswissen, um dieses Problem zu beheben (Alles noch einmal übermitteln? Nur diesen noch einmal übermitteln? Support anfragen? Endreceiver anfragen?). Diese Unsicherheit führt zu höheren Prozesskosten und einer schlechteren Akzeptanz.

© Swissdec 2026

A Referenzierte Dokumente

[CONTAINERXSD] *Container Schema*. Swissdec.

[ACKNOTIF] *Richtlinien für Lohndatenübermittlung*. Quittierung und Notifications. Swissdec. https://infopoint.swissdec.ch.

[UpgrDistri_Anforderungen] *Upgrade Distributor*. Anforderungen. Swissdec.

[SECPDF] *Richtlinien für Swissdec-Übermittlungen*. Sicherheit (Transmitter). Swissdec. https://infopoint.swissdec.ch.

[SECRXPDF] *Richtlinien für Swissdec-Übermittlungen*. Sicherheit (Endreceiver). Swissdec. https://infopoint.swissdec.ch.

[Richtlinien für Lohndatenverarbeitung] *Fachliche Swissdec-Richtlinien*. Fachliche Spezifikation. Swissdec. https://www.swissdec.ch.

[TFBASIS] *Testfälle Übermittlungstest Basisdienste*. https://www.swissdec.ch.

[RFC3986] *Uniform Resource Identifier (URI): Generic Syntax*. ietf. https://datatracker.ietf.org/doc/html/rfc3986.

© Swissdec 2026

B Glossar

D

Distributor	Der Distributor ist das zentrale System, das Daten vom Transmitter erhält, validiert, plausibilisiert und an die adressierten Endreceiver weiterleitet, sowie die erhaltenen Antworten zurück an den Transmitter sendet. Er übernimmt die Filterung und Verteilung der Daten.
Domäne	Organisation, das Daten übermittelt werden. Im Swissdec-Ökosystem bekannte Domänen sind; AHV, FAK, UVG, UVGZ, KTG, BVG, Lohnausweis, Quellensteuer, Grenzgänger und Statistik.

E

Empfängersystem	Das Empfängersystem erhält seine Daten vom Endreceiver, um diese dann auf fachlicher Ebene verarbeiten zu können. Es stellt ausserdem die Antworten bereit, die danach über den Endreceiver zurück an den Distributor und das Sendersystem übermittelt werden sollen. (Beispiel: Versicherer, Behörde)
Endreceiver	Der Endreceiver ist das technische Gegenstück zum Transmitter. Er empfängt und validiert die vom Distributor erhaltenen Daten und leitet diese an ein Empfängersystem weiter. Die vom Empfängersystem bereitgestellten Antworten werden danach vom Endreceiver in die Response an den Distributor eingefügt.

I

Initialmeldung	Die erste Meldung, mit der sich ein Sendersystem an den/die Endempfänger wendet.
Institution	Empfänger, der Daten erhält. Hier handelt es sich um Versicherungen, die den jeweiligen Domänen angehören. Eine Firma kann innerhalb einer Domäne mehrere Institutionen kontaktieren. Eine Institution kann mehrere Domänen unterstützen.

S

Sendersystem	Beim Sendersystem handelt es sich um ein System, das Daten aufbereitet und für den Versand an Endempfänger bereitstellt. Hier werden vor allem die fachlichen Anforderungen technisch korrekt umgesetzt (Beispiel: ERP)
Story	Einheit, die einen Aspekt der gesamten Übermittlung beschreibt. Abhängig vom Standard kann es sich hier um Quittungen, zusätzliche Fallinformationen oder anderes handeln. Stories können vom Sendersystem oder dem Endempfänger aus übermittelt werden und beinhalten Informationen, die die andere Seite zur Weiterverarbeitung des Falls benötigt.

T

Transmitter	Der Transmitter übermittelt die vom Sendersystem bereitgestellten Daten an den Swissdec Distributor und empfängt die erhaltenen Antworten (Responses), um diese zu verifizieren und an das Sendersystem weiterzuleiten.

© Swissdec 2026

U

UID	Jedes in der Schweiz aktive Unternehmen erhält eine einheitliche Unternehmens-Identifikationsnummer (UID). Diese UID wird im Auftrag des Bundes vom Bundesamts für Statistik BFS geführt. Beispiel UID: CHE-111.111.111.
UID-Register	Das Bundesamt für Statistik BFS führt das Register (https://www.uid.admin.ch), wo alle Unternehmen mit ihrer UID aufgelistet sind.

© Swissdec 2026

C Detailspezifikation Swissdec Unternehmens-Authentifizierung SUA

C.1 Einleitung

Verschiedene Swissdec Prozesse verlangen die eindeutige Authentifizierung des übermittelnden Unternehmens. Dies wird im Rahmen der "Swissdec Unternehmens-Authentifizierung" SUA gemacht. Aus technischer Sicht **muss** jedes Unternehmen ein von Swissdec ausgestelltes und signiertes SUA Zertifikat beantragen, das anschliessend für die zusätzliche elektronische Unterschrift (Signatur) der einzelnen Übermittlungen verwendet werden kann.

Vereinfachter Ablauf für jedes Unternehmen:

- Registrierung und Erstellung des SUA Zertifikats
- Verwendung des SUA Zertifikats zur Signierung von Übermittlungen
- Erneuerung oder Sperrung des SUA Zertifikats

Ein SUA Zertifikat kann während seiner Gültigkeitsdauer für alle Standards und Endreceiver verwendet werden. Somit muss ein Unternehmen nur ein SUA Zertifikat beantragen und kann dies für alle anderen Standards und Prozesse wiederverwenden.

Grundlage einer SUA Registrierung ist eine bestehende Geschäftsbeziehung des Unternehmens zu einer Versicherung, welche Identität des Unternehmens bereits geprüft hat. Auf diese überprüfte Identität stützt sich Swissdec für die Identifikation des Unternehmens. Der Distributor prüft während der Registrierung die Angaben zum Unternehmen sowie die bestehende Vertragsbeziehung beim Versicherer.

Folgende Angaben pro Unternehmen müssen bekannt sein:

Benötigte Angaben	Validierung
Name des Unternehmens	Identisch mit Angaben aus dem UID-Register
UID Nummer des Unternehmens	Identisch mit Angaben aus dem UID-Register Identisch mit Angaben beim Versicherer
Bestehende Vertragsverbindung (Vertragsnummer und Kundenummer)	Identisch mit Angaben beim Versicherer

C.2 SUA Prozess

C.2.1 Registrierung und Erstellung des SUA Zertifikats

Prozess-Schritt: Ein Unternehmen beweisst seine Identität und bekommt das Unternehmen spezifische SUA Zertifikat vom Distributor.

Ein Mitarbeiter des Unternehmens meldet die benötigten Stammdaten des Unternehmens über den Transmitter beim Distributor an. Dafür werden die UID-Nummer, der Firmenname sowie die bestehende Vertragsbeziehung mit dem adressierten Versicherer an den Distributor gemeldet.

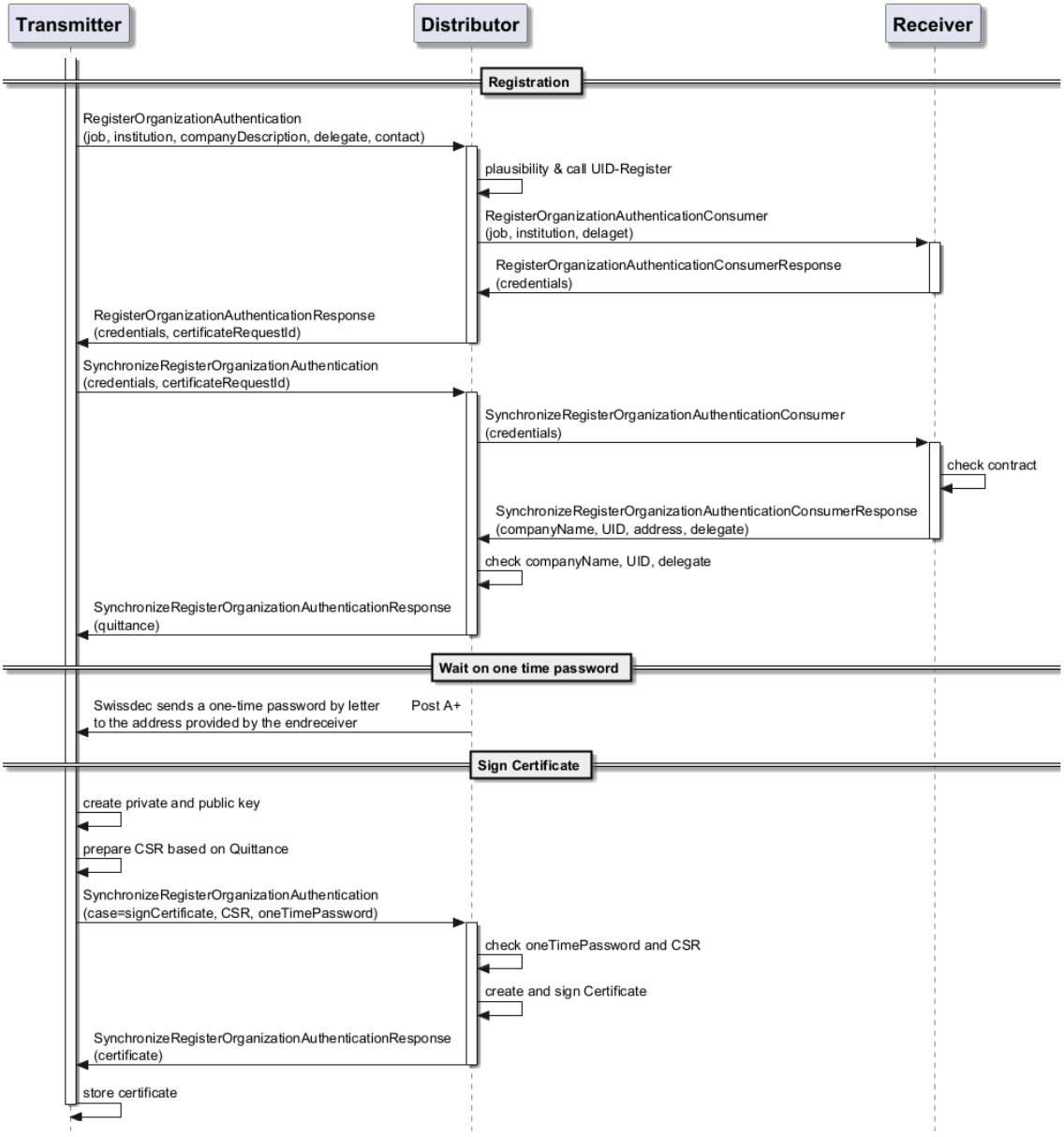
Der Distributor überprüft die Stammdaten des Unternehmens und vergleicht diese mit den Angaben aus dem UID-Register. Das Unternehmen muss gemäss UID-Register zu diesem Zeitpunkt aktiv ist. Zusätzlich werden die Angaben zur Vertragsverbindung dem adressierten Versicherer übermittelt. Der Versicherer sendet die Unternehmensdaten und dessen Adresse zurück. Beim Versicherer muss die identische UID des Unternehmens hinterlegt sein. Im Erfolgsfall wird eine Quittung an den Transmitter zurückgesendet.

Nach erfolgreicher Prüfung auf dem Distributor wird dem Transmitter eine Quittung zurückgesendet, die wichtige Informationen zum Erstellen des SUA Zertifikats enthält. Zu diesem Zeitpunkt wird ein Brief and die beim Versicherer hinterlegten Adresse des Unternehmens gesendet. Dieser Brief enthält ein einmaliges Registrie-

© Swissdec 2026

rungspasswort sowie das Sperrpasswort. Der physische Versand wird aus Sicherheitsgründen gemacht und stellt sicher, dass das korrekte Unternehmen Zugriff erhält und ein kompromittiertes System oder Angreifer kein SUA Zertifikat beantragen kann.

Das Registrierungspasswort wird zusammen mit der Zertifikatsanfrage, einem Abschnitt C.3.3, „Certificate Sign Request (CSR)“ mit den Angaben aus der Quittung, an den Distributor gesendet. Der CSR wird sowie das Passwort werden vom Distributor überprüft. Erst wenn alles korrekt ist, erstellt der Distributor über eine von Swissdec autorisierte Certificate Authority (CA) das SUA Zertifikat und sendet es an den Transmitter zurück. Der Transmitter installiert das SUA Zertifikat und stellt sicher, dass nur das registreirte Unternehmen sein eigenes SUA Zertifikat verwenden kann.



Ist eine Registrierung nicht gültig oder unvollständig, so wird ein entsprechender Fehler zurückgegeben. Diese Fehler sind im Dokument [[ACKNOTIF]] spezifiziert.

C.2.1.1 Registrierung als Stellvertreter eins Unternehmens (Delegate)

Als Stellvertreter eines Unternehmens kann ein anderer Benutzer im Auftrag des Unternehmens die Registrierung durchführen. In der Registrierung muss der Stellvertreter entsprechend deklariert werden. Als Sicher-

© Swissdec 2026

heitsmassnahme muss der identische Stellvertreter beim Versicherer auch hinterlegt sein, was der Distributor im Rahmen der Registrierung überprüft.

C.2.1.2 Registrierung als Testcase

Aus Sicherheitsgründen kann der SUA-Prozess als Testfall gestartet, jedoch nicht abgeschlossen werden. Die Registrierung mit Testcase kann informell zur Überprüfung durchgeführt werden, ob eine Registrierung möglich ist. In Falle des Testcase werden alle fachlichen Prüfungen normal durchgeführt jedoch anstelle der positiven Quittung ein spezifischer Fehlercode zurückgegeben, der auf den Testcase hinweist.

C.2.2 Verwendung des SUA Zertifikats

Ziel: Das Unternehmen verwendet das SUA Zertifikat zum zusätzlichen Signieren von Meldungen, um zu beweisen, dass die Meldung vom Unternehmen gesendet.

Das ausgestellte und gültige SUA Zertifikat wird als digitaler Ausweis des Unternehmens verwendet. Der Transmitter kann Meldung zusätzlich mit dem SUA Zertifikat signieren, dadurch kann der Distributor sowie die Endreceiver eindeutig prüfen, von welchem Unternehmen die Daten stammen und ob sie unterwegs verändert wurden.

Die technische Verwendung des SUA Zertifikats zur Erstellung der zusätzlichen Signatur wird im Dokument [[SECPDF]] spezifiziert.

C.2.3 Erneuerung des SUA Zertifikats

Ziel: Das Unternehmen erneuert sein SUA Zertifikat bevor es abläuft.

Das SUA Zertifikat ist zeitlich begrenzt. Kurz vor Ablauf initiiert der Transmitter automatisch einen Erneuerungsprozess, damit die Kommunikation nicht unterbrochen wird. Dabei sendet die Software eine neue Zertifikatsanfrage an den Distributor. Die Erneuerung kann nur während einer bestimmten Frist vor Ablauf des Zertifikats durchgeführt werden, um stetige Erneuerungen zu verhindern.

Eine Neuregistrierung des Unternehmens ist nicht immer erforderlich, da die Identität bereits bekannt ist. Aktuell kann ein SUA Zertifikat nicht unbegrenzt erneuert werden. Ist eine Zertifikat abgelaufen, kann die Erneuerung nicht mehr durchgeführt werden. In diesem Fall muss eine neue Registrierung durchgeführt werden.

Die genauen Anforderungen an die Gültigkeitsdauer, Erneuerungen und Fristen für die Erneuerung sind im Kapitel Abschnitt C.3.1, „Erneuerung der SUA Zertifikate“ spezifiziert.

C.2.4 Sperrung des SUA Zertifikats

Ziel: Das Unternehmen kann ein SUA Zertifikat von Swissdec sperren lassen

Die Sperrung soll durchgeführt werden, wenn ein ausgestelltes SUA Zertifikat nicht mehr vertrauenswürdig ist. Dies ist bei Schlüsselverlust, bei Sicherheitsvorfällen beim Transmitter oder Unternehmen sowie bei Wechsel des Transmitters notwendig. Es kann auch auf Wunsch des Unternehmens gemacht werden.

Zur Authentisierung verwendet die Firma das bei der Registrierung erhaltene Sperrpasswort und meldet damit die Sperrung an den Swissdec Support. Anschließend wird das Zertifikat bei der Certification Authority widerrufen. In der Fachsprache wird dies als "revoked" bezeichnet. Ab diesem Zeitpunkt soll der Transmitter das gesperrte SUA Zertifikat nicht mehr verwenden. Der Distributor würde jede mit dem gesperrten SUA Zertifkat signierte Meldung ablehnen.

C.3 Technische Anforderungen an die Zertifkate

Diese nachfolgenden technischen Anforderungen haben dynamischen Character und können im Rahmen eines Distributor Releases den steigenden Sicherheitsanforderungen angepasst werden. Die Anpassungen werden mit dem Release von Swissdec kommuniziert und sind im Dokument [[SECPDF]] enthalten.

© Swissdec 2026

C.3.1 Erneuerung der SUA Zertifikate

Kontext	Wert	Beschreibung
Sicherheitsmerkmal Registrierung	Brief/ A+	Passwörter versendet an die beim Versicherer hinterlegten Adresse. Zweiter, nicht elektronsicher Kanal.
Registrierungspasswort	Min. 12 Zeichen	Zahlen, Gross- und Kleinbuchstaben
	1 Jahr	Gültigkeitszeitraum
Sperrpasswort	Min. 12 Zeichen	Zahlen, Gross- und Kleinbuchstaben
	5 Jahr	Gültigkeitszeitraum
Erneuerungsprozess	60 Tage	Zeitraum vor Ablauf des Zertifikats, ab welchem der Transmitter die Erneuerung starten darf.
	3x (3 Jahre)	Anzahl möglicher automatischer Erneuerungen

C.3.2 Aufbau Zertifikat nach X509

Die SUA Zertifikate werden als X509 Zertifikate nach RFC 5280 von einer von Swissdec autorisierten Certificate Authority (CA) ausgestellt. Das dazugehörige Schlüsselpaar (Private + Public Key) muss vom Transmitter in einer sicheren Infrastruktur erzeugt und persistiert werden. Der Transmitter ist für die Sicherheit des Private Keys verantwortlich. Zu Erstellung und Signierung des SUA Zertifikats wird nur der CSR verwendet. Damit wird sichergestellt, dass der Private Key das Transmitter System nie verlassen muss oder darf.

Attribut	Wert	Beschreibung
Zertifikat		
Version	3	Zertifikatsversion gemäss RFC 5280
Serial Number	{Wert}	Eindeutige Identifikation des Zertifikats.
Validity	1 Jahr	Zeitraum der Gültigkeit des Zertifikats
Public Key	RSA 2048 RSA 3072 RSA 4096	Akzeptierte Schlüssellänge
Signature Algorithm	Sha256WithRSA	Verwendeter Signature Algorithmus
Signature Value	{Wert}	Signatur des Zertifikats
Issuer		
CN	Verein Swissdec Issuing CA by DigiCert	Common Name (OID 2.5.4.3)
O	Verein Swissdec	OrganizationName (OID 2.5.4.10)
C	CH	CountryName (OID 2.5.4.6)
Subject		
C	CH	CountryName (OID 2.5.4.6)
ST *Optional*	{Kanton des Unternehmens}	StateOrProvinceName (OID 2.5.4.8)
L *Optional*	{Ort des Unternehmens}	LocalityName (OID 2.5.4.7)
CN	NTRCH-{UID}@swissdec.ch	Common Name (OID 2.5.4.3)
O	{Name aus dem UID Register}	OrganizationName (OID 2.5.4.10)

© Swissdec 2026

Attribut	Wert	Beschreibung
ORG_ID	NTRCH-{UID}	OrganizationIdentifier (OID 2.5.4.97)

C.3.3 Certificate Sign Request (CSR)

Ein Certificate Signing Request (CSR) ist die Anfrage, um ein digitales Zertifikat zu erhalten. Der CSR enthält den gewünschten Subject und den PublicKey, für den das Zertifikat ausgestellt werden soll. Diese Angaben werden von Distributor überprüft und müssen der Quittung entsprechen, die bei der Registrierung erstellt wurde.

Attribut	Beschreibung
Format	PEM, PKCS #10, gemäss RFC 2986
Public Key	Der öffentliche Schlüssel, der vom Transmitter generiert wurde
Subject	Informationen zum Zertifikatsinhaber gemäss Quittung nach Abschnitt C.3.2, „Aufbau Zertifikat nach X509“
Signature Algorithm	Sha256WithRSA
Signature Value	Signatur des Zertifikats, erstellt mit dazugehörigem Private Key

© Swissdec 2026

D Detailspezifikation DialogMessages

D.1 Einleitung

Eine DialogMessage wird dazu verwendet, einen Dialog zwischen verschiedenen Teilnehmern eines Standards zu ermöglichen. Dabei werden Nachrichten schemavalid und sicher zwischen z.B. Unternehmen und Versicherer versendet. Abhängig vom verwendeten Standard bietet DialogMessage Möglichkeiten vom Austausch einfacher Informationen bis hin zur Umsetzung einer gesicherten Messenger-Applikation.

Hierbei wird zwischen Standard-DialogMessages und freien DialogMessages unterschieden. Standard DialogMessages besitzen eine spezifizierte Struktur mit genau definierten Ids. Freie DialogMessages setzen sich beliebig aus den Schema-Elementen der DialogMessage zusammen.

Freie DialogMessages erhöhen die Flexibilität des swissdec Standards, indem sie neue Strukturen von Fragen und Antworten zwischen den beteiligten Partnern ermöglichen, ohne neue Versionen des swissdec Standards abwarten zu müssen. Falls sich ein Typ einer freien DialogMessage als allgemeiner Bedarf herausstellt, kann dieser Typ bei einer neuen Version dieses Dokuments als neue Standard DialogMessage spezifiziert werden. In einigen Standards sind freie DialogMessages aus Datenschutzgründen nicht gestattet.

Dieser Anhang enthält die Anforderungen an das generelle Vorgehen zur Darstellung von DialogMessages sowie Anforderungen an die Darstellung von vorgegebenen Standard DialogMessages.

D.2 Die Verwendung von DialogMessages

D.2.1 Test und Abnahme

Transmitterseitig wird die DialogMessage im Rahmen der Synchronisierungen abgehandelt. Die richtige Darstellung, sowie die richtige Durchführung eines Dialogs werden von Swissdec geprüft.

Auf Endreceiver-Seite steht es dem Endempfänger frei, welche Form von DialogMessage er verwendet. Er kann auf Standard-Messages zurückgreifen oder eigene, freie Messages definieren. Einschränkungen für bestimmte Domänen und Standards werden separat publiziert.

Standard-Messages werden im Dialogmessage-Katalog publiziert und können vom Swissdec Infopoint (https://infopoint.swissdec.ch) heruntergeladen werden.

Auf Transmitter-Seite *sollen* momentan keine Standard oder freien DialogMessages verfasst werden. Der Transmitter *soll* nur jeweils auf die DialogMessages vom Endreceiver reagieren.

D.2.2 Darstellung der DialogMessage

Auf Transmitter-Seite **muss** ein generischer Renderer implementiert werden, welcher alle freien und Standard DialogMessages darstellen und beantworten können **muss**. Auch der Endreceiver *sollte* einen generischen Renderer implementieren, da in zukünftigen Versionen auch vom Transmitter her Mitteilungen per DialogMessage möglich werden könnten.

Anstelle des generischen Renderes *können* für Standard DialogMessages spezielle Renderer implementiert werden.

	Transmitter als Reactor		**Endreceiver als Actor**
	Darstellen	**Antworten wenn <Answer>**	**Verfassen und evtl. Antwort verarbeiten**
StandardDialogMessage **"SimpleMessage"**	**muss**	Nicht vorgesehen	*kann*
Weitere DialogMessages	**muss**	**muss**	*kann*
Freie DialogMessages	**muss**	**muss**	*kann*

© Swissdec 2026

D.2.3 Elemente der Story DialogMessage

Die Elemente (Datenstruktur) aus welchen sich eine DialogMessage zusammenstellen lässt, sind im zugehörigen Schema des entsprechenden Standards und in den entsprechenden Richtlinien für Lohndatenverarbeitung beschrieben.

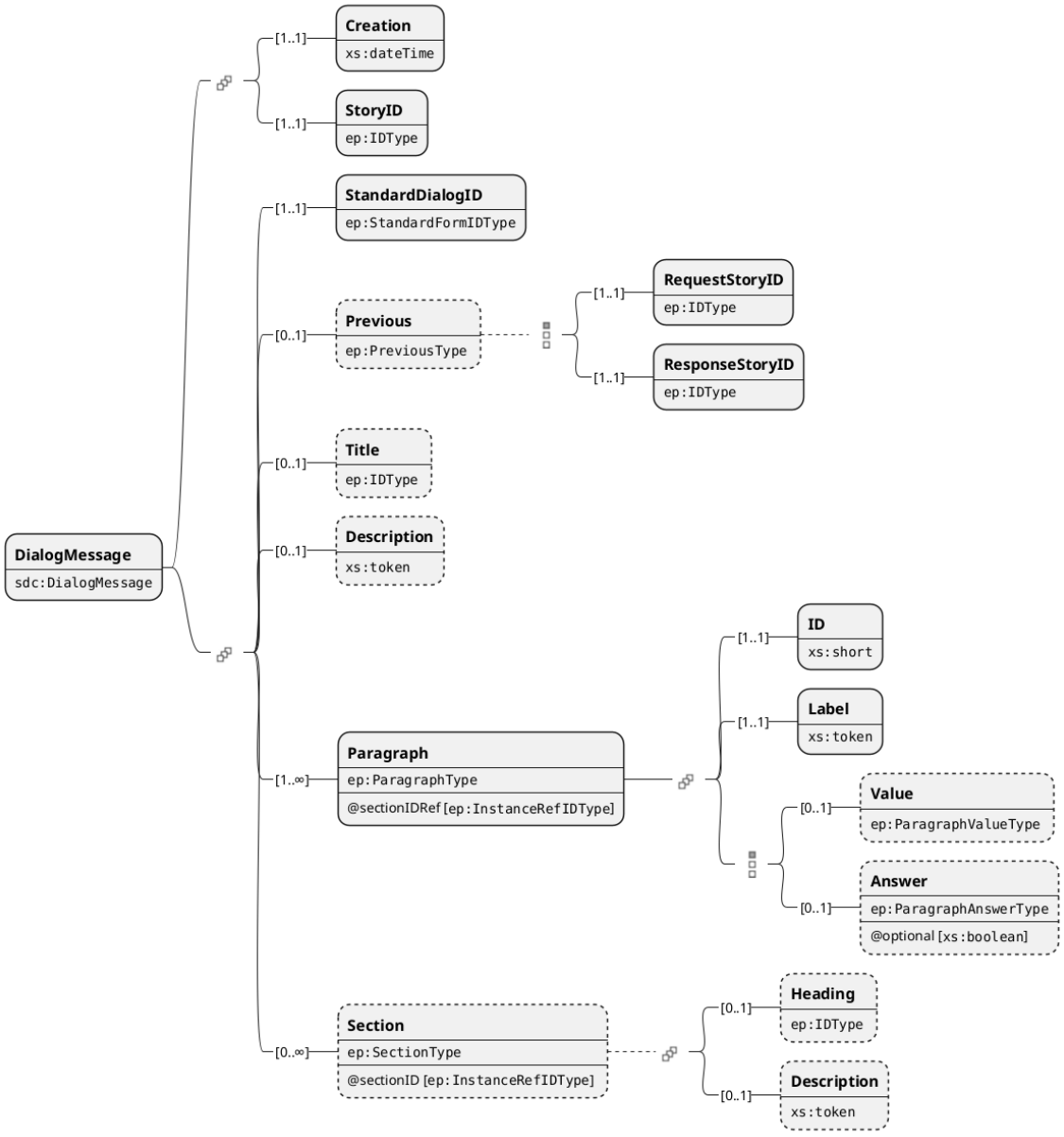


Abbildung D.1. DialogMessage XML-Schema Struktur

Die GUI-Darstellung der einzelnen Elemente entspricht in etwa denjenigen eines Formulars z.B. in HTML oder einer Programmiersprache. Die Darstellung kann als einfacher Text oder aber unter Verwendung von GUI-Elementen wie Checkboxes, Dropdowns oder Kalenderkomponenten erfolgen.

Das Element «Paragraph/Label» wird sowohl als Label, als auch als alleinstehender Text verwendet. Zu beachten: Werte unter Paragraph/Answer können einen vorgegebenen Default-Wert haben, welcher dargestellt werden muss.

© Swissdec 2026

Element / Attribut	Editierbar	Darstellung	Bemerkung
Creation	Nein	Kann	Datum und Zeit der Erstellung der DialogMessage
StoryID	Nein	Kann	StoryID der DialogMessage: ID muss für den Geschäftsfall eindeutig sein. Ausserhalb eines Geschäftsfalls muss sie ebenfalls eindeutig sein.
StandardDialogID	Nein	Kann	Kennzeichnung als standardisierte DialogMessage oder «notStandard»
Previous	Nein	Kann	RequestStoryID oder ResponseStoryID der vorangegangenen DialogMessage, je nach dem ob die DialogMessage in einem Request oder einer Response enthalten war.
Title	Nein	Muss	Optionaler Titel der DialogMessage. Muss dargestellt werden, wenn vorhanden.
Description	Nein	Muss	Optionale Beschreibung der DialogMessage. Muss dargestellt werden, wenn vorhanden, z.B. als Tooltip
Section/sectionID	Nein	Kann	
Section/Heading	Nein	Muss	
Section/Description	Nein	Muss	
Paragraph	Nein	Muss	Absatz mit Inhalt
Paragraph/SectionIDRef	Nein	Kann	Attribut welches die Zugehörigkeit eines Absatzes zu einer Section anzeigt
Paragraph/ID	Nein	Kann	
Paragraph/Label	Nein	Muss	Text in einem Absatz, Label für Value oder Answer
Paragraph/Value/String	Nein	Muss	
Paragraph/Value/Integer	Nein	Muss	
Paragraph/Value/Double	Nein	Muss	
Paragraph/Value/Boolean	Nein	Muss	Bei einer Gruppe von Booleans ist keine ausschliessende Auswahl möglich.
Paragraph/Value/Date	Nein	Muss	

© Swissdec 2026

Element / Attribut	Editierbar	Darstellung	Bemerkung
Paragraph/Value/ DateTime	Nein	Muss	
Paragraph/Value/ YesNoUnknown	Nein	Muss	
Answer/<Type>/Default: darf vom Actor gesetzt, vom Reactor in der Antwort aber nicht verändert werden			
Paragraph/Answer/String/ Default	Nein	Muss	
Paragraph/Answer/String/ Value	Muss	Muss	
Paragraph/Answer/Integer/Default	Nein	Muss	
Paragraph/Answer/Integer/Value	Muss	Muss	
Paragraph/Answer/Double/Default	Nein	Muss	
Paragraph/Answer/Double/Value	Muss	Muss	
Paragraph/Answer/Boolean/Default	Nein	Muss	Bei einer Gruppe von Booleans ist keine ausschliessende Auswahl möglich.
Paragraph/Answer/Boolean/Value	Muss	Muss	Bei einer Gruppe von Booleans ist keine ausschliessende Auswahl möglich.
Paragraph/Answer/Date/ Default	Nein	Muss	z.B. Kalender mit Textfeld
Paragraph/Answer/Date/ Value	Muss	Muss	z.B. Kalender mit Textfeld
Paragraph/Answer/DateTime/Default	Nein	Muss	z.B. Kalender mit Textfeld
Paragraph/Answer/DateTime/Value	Muss	Muss	z.B. Kalender mit Textfeld
Paragraph/Answer/YesNoUnknown/Default	Nein	Muss	z.B. DropDown
Paragraph/Answer/YesNoUnknown/Value	Muss	Muss	z.B. DropDown

D.2.4 Abfolge bei der Darstellung von DialogMessages

Das Aktivitätsdiagramm beschreibt das generische Vorgehen beim Empfang einer Response oder eines Requests mit enthaltenen DialogMessages. Die Antwort auf eine DialogMessage erfolgt, sofern notwendig, zu einem späteren Zeitpunkt in einem neuen Request-Response-Zyklus.

© Swissdec 2026

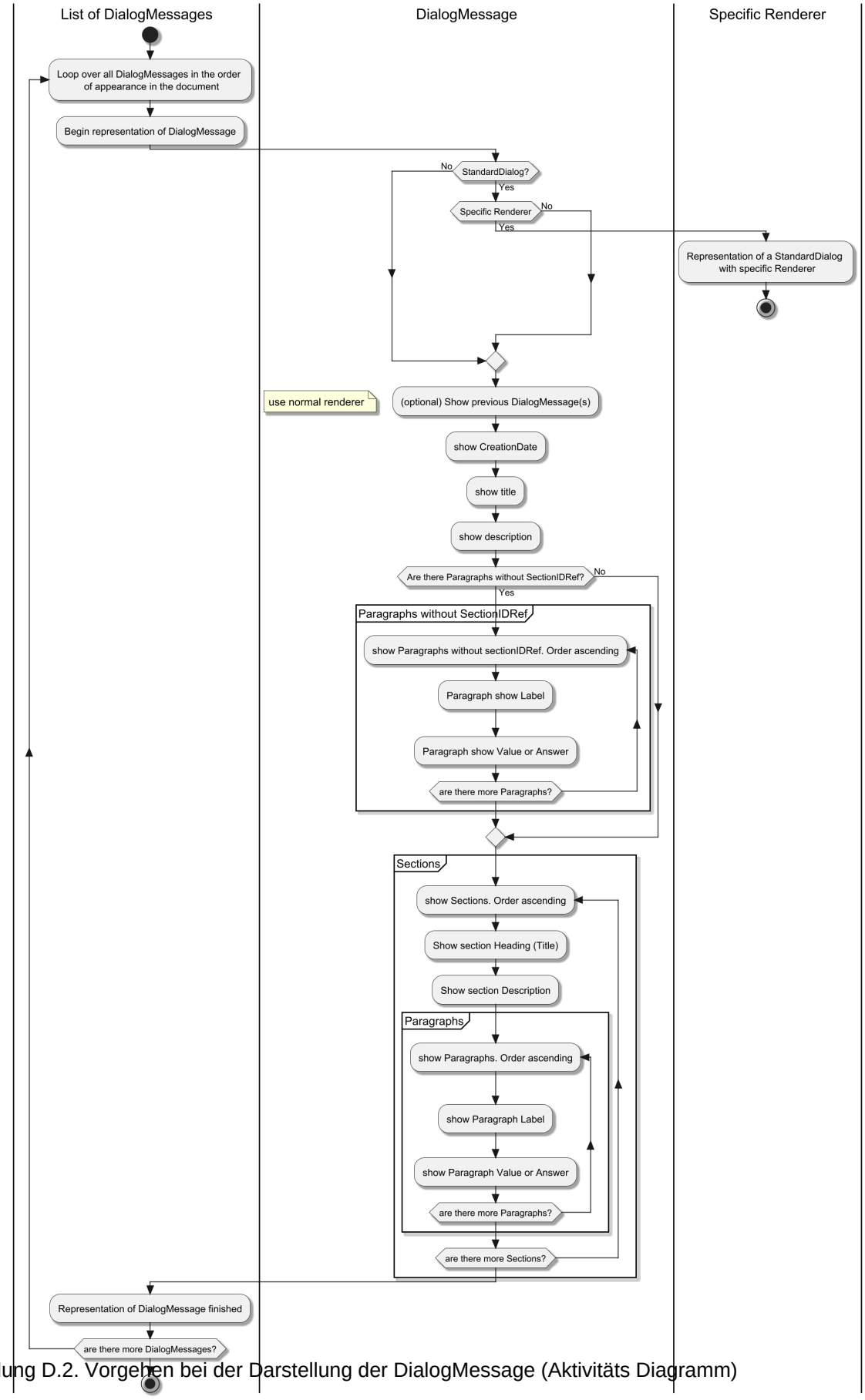


Abbildung D.2. Vorgehen bei der Darstellung der DialogMessage (Aktivitäts Diagramm)

© Swissdec 2026

D.2.5 Antwort als Reactor mittels einer DialogMessage

Bei der Antwort eines Reactors auf eine DialogMessage, wird zwischen folgenden Situationen unterschieden.

1. Es wird eine DialogMessage-Story mit «Paragraph/Answer»-Elementen empfangen. Die DialogMessage muss dargestellt werden und es muss eine Antwort mit der gleichen Struktur gesendet werden. Die Antwort wird wahrscheinlich in einem späteren Request-Response Zyklus geschickt.

2. Es wird eine DialogMessage-Story ohne «Paragraph/Answer»-Elemente empfangen. Die DialogMessage muss dargestellt werden. Auf diese DialogMessage muss nicht geantwortet werden. Es muss keine Datenstruktur kopiert werden.

3. Es wird eine DialogMessage empfangen, welche sich auf eine beliebige, vorangegangene Story bezieht. Der Bezug wird durch das Element «Previous» hergestellt, s. Verkettung.

D.2.5.1 Antwort auf eine DialogMessage mit enthaltenen Paragraph/Answer-Elementen

Die Antwort auf eine DialogMessage mit enthaltenen «Paragraph/Answer»-Elementen ist wiederum eine DialogMessage mit «Paragraph/Answer»-Elementen. Die Antwort wird wahrscheinlich in einem späteren Request-Response Zyklus erfolgen.

In der Antwort-DialogMessage muss das Element «Previous» verwendet werden, so dass sowohl für den Actor, wie auch für den Reactor nachvollziehbar ist, auf welche DialogMessage-Story sich die Antwort bezieht. Im Transmitter wird immer das Element «Previous/ResponseStoryID» verwendet, da die anfragende DialogMessage vom Endreceiver immer in einer SynchroniceIncidentResponse enthalten war. Im Endreceiver wird immer «Previous/RequestStoryID» verwendet, da die anfragende DialogMessage vom Transmitter immer in einem SynchroniceIncidentRequest enthalten war.

Bis auf «Creation», «StoryID» und «Previous» wird die gesamte DialogMessage mit enthaltenen «Para-graph/Answer»-Elementen gespiegelt.

Die durch den Dialogteilnehmer geänderten Werte, ansonsten die Default-Werte, müssen bei der Antwort immer in die Paragraph/Answer/<Type>/Value Elemente eingetragen werden, nicht in die «Paragraph/Answer/<Type>/Default»-Elemente. Dabei wird <Type> durch den jeweils übergebenen Werte-Typ ersetzt, z.B. «Paragraph/Answer/Boolean/Value».

«Answer»-Elemente, welche nur optional ausgefüllt werden müssen, sind durch das Attribut «optional» gekennzeichnet. Jedes andere «Paragraph/Answer/<Type>/Value»-Element muss bei der Antwort gesetzt sein. Das bedeutet, dass alle mit «Paragraph/Answer»-Elementen angefragten Werte ohne das Attribut «optional» Pflichtfelder sind, auch wenn diese nach XML-Schema leer zurückgesandt werden könnten.

D.2.5.2 Verkettung von DialogMessages

Die verschiedenen DialogMessages können miteinander verkettet werden. Sind in einer anfragenden DialogMessage «Answer»-Elemente enthalten, muss die Antwort-DialogMessage mit der anfragenden DialogMessage durch das Element «Previous» verkettet werden. Damit lässt sich in der Visualisierung eine ganze Kette von DialogMessages darstellen. In dieser Sicht werden einzelne DialogMessages für den Benutzer verständlicher, da diese in einem Ablaufkontext stehen und dann betrachtet werden können.

Die Verkettung wird mit dem Element «DialolgMessage/Previous» gesteuert.

1. Ein Actor initiert im ersten Schritt einen Dialog und kann optional im Element «Previous» den Link zu einer bestehenden StoryID setzen. Dabei wird unterschieden zwischen einer beliebigen bestehenden Response- oder RequestStoryID.

2. Ein Reactor muss bei einer Antwort auf eine Actor DialogMessage immer ein «Previous»-Element setzen und zwar die StoryID der zuvor erhaltenen Actor DialogMessage.

© Swissdec 2026

D.3 Standard DialogMessages

Die im DialogMessage-Katalog beschriebenen DialogMessages sind standardisiert und müssen im Rahmen einer Swissdec Zertifizierung nach Tabelle 3 Anforderungen zur Umsetzung von DialogMessages erzeugt und dargestellt werden können. Die definierten Strukturen müssen eingehalten werden.

Eine DialogMessage muss die StandardDialogID ihres Typs im Instanzdokument setzen, s. nachfolgende Tabelle.

DialogMessage	StandardDialogID
SimpleMessage	0000.0001.0001-001
TaskWithDeadline	0000.0001.0001-002
Freie DialogMessage	notStandard

Die letzten drei Ziffern sind die Minor-Version der in diesem Dokument spezifizierten StandardDialogMessages. Sollte eine spezifizierte DialogMessage Änderungen erfahren, wird ihre Minor-Version erhöht.

Zu allen spezifizierten Standard DialogMessages gibt es mitgeltende Beispiele in Form von Instanzdokumenten im DialogMessage-Katalog. (https://infopoint.swissdec.ch).

© Swissdec 2026

E Detailspezifikation Completion URL

E.1 Aufbau und Verarbeitung der Completion URL

E.1.1 Parsen der Basis-URL

Die Completion-URL führt auf eine Webseite mit Login-Maske. Key und Password aus dem XML-Dokument **müssen** als Parameter an die URL angehängt werden. Dies ermöglicht eine Vorbefüllung der Login-Felder. Der Benutzer muss somit Key und Passwort nicht von Hand eingeben.

Die in der Story mitgelieferte Completion-URL darf bereits Parameter wie z.B. Sprachangaben enthalten. Wenn dies der Fall ist, werden Key und Password jeweils mit einem „&“ angefügt. Handelt es sich beim Key jedoch um den ersten Parameter der URL, wird dieser mit einem „?“ angefügt.

Beispiel 1: https://www.institutionA.ch

Beispiel 2: https://www.institutionA.ch?language=fr

Das System muss dies erkennen können und die URL richtig ergänzen.

Beispiel 1: https://www.institutionA.ch?key=u1&password=cxsy23450dl

Beispiel 2: https://www.institutionA.ch?language=fr&key=u1&password=cxsy23450dl

Für den Fall, dass es empfängerseitig Probleme mit der Vorbefüllung der Felder gibt, müssen Key und Passwort dem Benutzer auf der Benutzeroberfläche eindeutig dargestellt werden, sodass er sie notfalls durch Copy-Paste in die Felder der Completion-Maske einfügen kann.

E.1.2 Kodierung von URL und Parametern

Bei der Übermittlung von Completion-Informationen **muss** beachtet werden, dass Key, Passwort und URL Sonderzeichen enthalten können, welche im XML oder in der URL nicht dargestellt werden können/dürfen. Die Codierung dieser Sonderzeichen muss korrekt umgesetzt werden, damit die Completion reibungslos funktioniert.

Beispiel Completion-Informationen beim Empfänger:

- Completion-URL: https://institutionA.ch?parameter1=test&parameter2=info
- Key: u1#
- Passwort: cxsy2%@=30#dlü

Diese Completion-URL, sowie Key und Passwort werden in der XML-Response zurückgegeben. Dies kann bedeuten, dass einige Sonderzeichen, welche im XML nicht verwendet werden dürfen, mit Character Entity References codiert werden (z.B. & anstelle von „&“ oder < anstelle von „<“). Es ist darauf zu achten, dass diese Character Entity References im Sendersystem dem Benutzer wieder in entschlüsselter Form dargestellt werden, und vor allem, dass die Sonderzeichen dem Browser auf korrekte Weise übergeben werden, damit sie richtig interpretiert werden können:

Beispiel

Completion-URL in der XML-Response:

```
https://www.completion-url.ch/?parameter1=test&amp;parameter2=info
```

Muss dargestellt/verarbeitet werden als:

```
https://www.completion-url.ch/?parameter1=test&parameter2=info
```


© Swissdec 2026

In einer zweiten Etappe geht es darum, dass Key und Passwort der URL als Parameter angefügt werden, um auf diese Weise die Vorbefüllung der Felder im Completion-Login zu erhalten. Hier ist darauf zu achten, dass <Key> und <Password> im Element <Credentials> vor dem Anhängen an die Completion-URL vom Transmitter in eine URL-kodierte Form gebracht werden müssen, damit eventuell darin enthaltene Sonderzeichen vom Browser richtig interpretiert werden können.

Dem Endbenutzer dürfen Key und Passwort allerdings nicht in URL-kodierter Form dargestellt werden, sondern exakt so, wie sie in der XML-Quittung enthalten sind. Dies ist notwendig, da die Web-Eingabeformulare keine URL-kodierte Form verlangen und der Benutzer auf diese Weise mit direktem Copy-Paste arbeiten kann.

Beispiel

Darstellung in Sendersystem:

- Completion-URL:: https://www.completion-url.ch/?parameter1=test&parameter2=info
- Key: u1#
- Passwort: cxsy2%@=30#dlü

Zusammengefügte Completion-URL für den Aufruf der Completion mit vorbefüllten Feldern für Key und Passwort mit URL-kodierten Parametern:

```
https://www.institutiona.ch/?key=u1%23&password=cxsy2%25%40%3d30%23dl%c3%bc
```

Gesamtsituation

Empfänger sendet in der XML-Response:

Completion-URL (mit Character Entity Reference, mit URL-Encoding)

```
https://www.completion-url.ch/?parameter1=test&amp;parameter2=info%23
```

Key und Passwort (keine Character Entity Reference, kein URL-Encoding)

```
u1#; cxsy2%@=30#dlü
```

Darstellung der Response im Sendersystem:

Completion-URL (keine Character Entity Reference, mit URL-Encoding)

```
https://www.completion-url.ch/?parameter1=test&parameter2=info%23
```

Key und Passwort (keine Character Entity Reference, kein URL-Encoding)

```
u1#; cxsy2%@=30#dlü
```

Key und Passwort werden hier im Original dargestellt, damit sie notfalls per Copy-Paste ins Login-WebFormular eingefügt werden können.

Aufruf der Completion im Browser:

Completion-URL mit angehängten Parametern (keine Character Entity Reference, mit URL-Encoding)

```
http://www.institutionA.ch?key=u1%23&password=cxsy2%25%40%3d30%23dl%c3%bc
```

Key und Passwort werden hier ebenfalls mit URL-Encoding angehängt!

Die meisten Entwicklungsplattformen haben in ihren Bibliotheken Tools zur URL-Kodierung. Die URL Kodierung wird in [[RFC3986]] beschrieben.

Aufbau einer URL gemäss[[RFC3986]]:

© Swissdec 2026

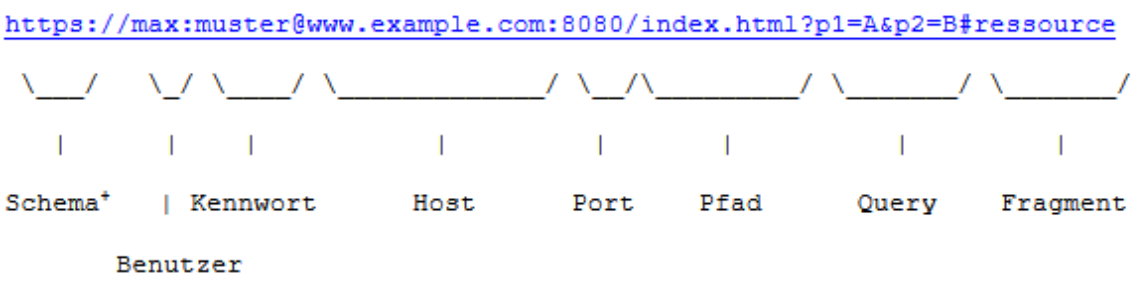


Abbildung E.1. Aufbau URL gemäss RFC3986

© Swissdec 2026

F Bausteine

F.1 AB-01 Konfiguration

AB-01: Konfiguration	
Kurzbeschreibung	Korrekte und sichere Adressierung des Distributors
Akteure	Sendersystem
Fachliche Anforderung	Das Sendersystem ist für die korrekte Adressierung des Distributors verantwortlich. Dazu **muss** das Empfängersystem via korrekter URL angesprochen werden. Aus Sicherheitsgründen *sollte* die Adressierung vom Endbenutzer nicht beliebig verändert werden können. Der Distributor adressiert korrekt die Empfängersysteme und verfügt über die nötigen Zertifikate für die sichere Kommunikation mit diesen.
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Konfiguration des Empfängersystems via URL
Ergebnis	Die korrekte Adressierung zwischen Sender und Empfänger ist gewährleistet.
Fehlerbehandlung	Abhängig von den beteiligten Systemen.
Technische Implikationen	keine

Tabelle F.1. AB-01 Konfiguration

F.2 AB-02 Erreichbarkeit

AB-02: Erreichbarkeit	
Kurzbeschreibung	Erreichbarkeitstest zwischen Distributor und Empfängersystem
Akteure	Distributor, Empfängersystem
Fachliche Anforderung	Die Erreichbarkeit des Endempfängers muss geprüft werden. Dazu wird eine einfache Anfrage vom Distributor an den Receiver geschickt. Die Rückantwort des Receivers an den Distributor bestätigt die Erreichbarkeit.
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Der Distributor löst die Überprüfung aus. 2. Der Distributor sendet eine einfache Serveranfrage (Ping) an die Zieladresse des Receivers. 3. Der Distributor wertet die Rückantwort des Receivers aus.
Ergebnis	Die Rückantwort des Receivers enthält einen Timestamp mit der Systemzeit des Receivers oder eine Fehlermeldung.
Fehlerbehandlung	1. Der Receiver ist nicht erreichbar. Fehlermeldung.
Technische Implikationen	keine

Tabelle F.2. AB-02 Erreichbarkeit

© Swissdec 2026

F.3 AB-03 Interoperabilität

AB-03: Interoperabilität	
Kurzbeschreibung	Kommunikationstest zwischen Sendersystem und Distributor
Akteure	Sendersystem
Fachliche Anforderung	Die Kommunikation zwischen Sendersystem und Distributor muss geprüft werden. Zu diesem Zweck werden zwei Operanden und ein Umlautstring an den Distributor gesendet. Die Rückantwort des Distributors ermöglicht es, Probleme mit Encoding oder Byte-Order zu erkennen.
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Der Akteur löst die Überprüfung aus. 2. Das Sendersystem sendet eine Interoperabilitätsanfrage (2 Operanden, Umlautstring) an die Zieladresse des Distributors. 3. Das Sendersystem wertet die Rückantwort des Distributors aus.
Ergebnis	Die Rückantwort des Distributors enthält Informationen zur distributorseitigen Interoperabilität sowie Ergebnisse, welche transmitterseitig verifiziert werden müssen.
Fehlerbehandlung	1. Distributor nicht erreichbar. Fehlermeldung. 2. Fehler in der Interoperabilität. Der Fehler muss dem Benutzer klar verständlich angezeigt werden.
Technische Implikationen	keine

Tabelle F.3. AB-03 Interoperabilität

F.4 AB-04 Anmeldung

AB-04: Anmeldung	
Kurzbeschreibung	Anmeldung beim Empfängersystem für eine spätere Übermittlung
Akteure	Sendersystem
Fachliche Anforderung	Ein Sendersystem muss sich bei einem Empfängersystem für einen späteren Übermittlungsprozess anmelden können. Mit diesem Anmeldeprozess wird der zukünftige Informationsaustausch über den Distributor zwischen den beiden Beteiligten in die Wege geleitet.
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Der Akteur stellt die benötigten Informationen für eine Anmeldung bereit. 2. Das Sendersystem adressiert den gewünschten Empfänger korrekt. 3. Das Sendersystem übermittelt die Anmeldung an den Distributor. 4. Die Antwort vom Distributor wird ausgewertet und dem Benutzer angezeigt.
Ergebnis	Das Sendersystem konnte sich bei einem Empfänger für einen späteren Übermittlungsprozess anmelden.
Fehlerbehandlung	1. Der Empfänger ist nicht erreichbar. Fehlermeldung. 2. Die Anmeldung verläuft nicht erfolgreich. Die Fehlermeldung vom Empfänger/Distributor wird ausgegeben.

© Swissdec 2026

AB-04: Anmeldung	
Technische Implikationen	keine

Tabelle F.4. AB-04 Anmeldung

F.5 AB-05 Explizite Adressierung (direkte Adressierung)

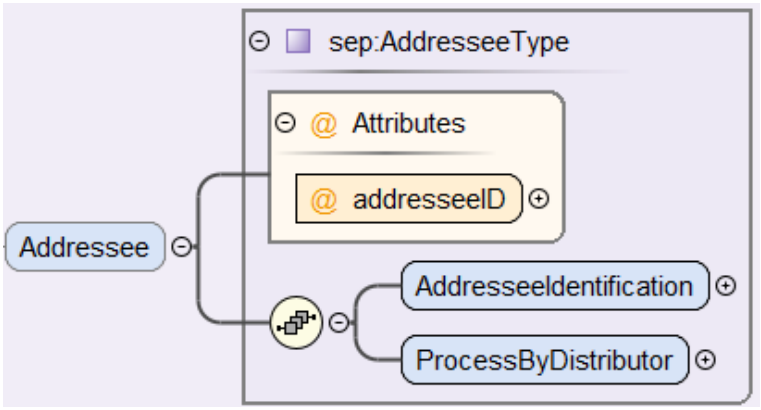
AB-05: Explizite Adressierung eines Empfängers (direkte Adressierung)	
Kurzbeschreibung	Explizite Adressierung zur Datenübermittlung
Akteure	Sendersystem, Distributor
Fachliche Anforderung	Rahmenbedingung aus [[UpgrDistri_Anforderungen]]: Die Lohndaten werden sicher und direkt an die vom Kunden (Unternehmen) adressierten Endempfänger übermittelt. Zusätzliche Intermediäre dürfen aus Datenschutzgründen nicht beliefert werden.
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Der Benutzer definiert die Adressaten 2. Der Distributor leitet die Meldung allen Adressaten weiter 3. Die Adressaten prüfen, ob Sie die Daten erhalten dürfen (richtige Adressierung) 4. Der Benutzer erhält eine Quittung / Fehler pro Adressat
Ergebnis	Alle Adressaten wurden bedient, der Benutzer hat pro Adressat eine Quittung/Fehler
Fehlerbehandlung	1. Der Receiver ist nicht erreichbar. Fehlermeldung.
Technische Implikationen	 Abbildung F.1. AddresseeType • Definiertes Feld mit Identifier des Adressaten in den Daten (zum Beispiel InstitutionID, Canton, ...) • Antwortstruktur mit Quittung/Fehler pro Adressat

Tabelle F.5. AB-05 Explizite Adressierung (direkte Adressierung)

F.6 AB-06 Implizite Adressierung (Gruppenadressierung)

AB-06 Implizite Adressierung (Gruppenadressierung)	
Kurzbeschreibung	Domänenabhängige implizite Adressierung

© Swissdec 2026

AB-06 Implizite Adressierung (Gruppenadressierung)	
Akteure	Sendersystem, Distributor
Fachliche Anforderung	Die Lohndaten werden sicher und direkt an alle Endempfänger, die die Meldung erhalten wollen und dürfen, übermittelt. Zusätzliche Intermediäre dürfen aus Datenschutzgründen nicht beliefert werden.
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Der Benutzer definiert die Adressaten als Gruppe. 2. Der Distributor filtert und verteilt die Inhalte an alle Adressaten, wobei er die Adressierung aus spezifischen Inhalten herausliest (z.B. Kanton). 3. Die Adressaten prüfen, ob sie die Daten erhalten dürfen (richtige Adressierung) 4. Der Benutzer erhält eine Quittung / Fehler pro Adressat
Ergebnis	Alle Adressaten wurden bedient, der Benutzer hat pro Adressat eine Quittung/Fehler
Fehlerbehandlung	1. Der Distributor ist nicht erreichbar. Fehlermeldung. 2. Ein oder mehrere Adressaten sind nicht erreichbar/unbekannt. Verifizieren der Adressierung und erneute Übermittlung.
Technische Implikationen	

Tabelle F.6. AB-06 Implizite Adressierung (Gruppenadressierung)

F.7 AB-07 Absichtserklärung

AB-07 Absichtserklärung	
Kurzbeschreibung	Der Absender muss eindeutig erklären können, an wen Daten zu senden sind.
Akteure	Sendersystem, Distributor
Fachliche Anforderung	Der Absender soll bei grösseren und komplexeren Daten verständlich erklären, was für Daten an wen zu senden sind. Dabei sorgt der Distributor für eine korrekte Filterung und Verteilung der Meldung, d.h. die „identische Meldung“ kann unterschiedlich verteilt werden. Diese Verteilung mit entsprechender Filterung muss nicht durch den Absender vorgenommen bzw. entwickelt werden, sondern wird zentral gelöst.
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Der Absender kann eindeutig auswählen, welche Empfänger Daten erhalten sollen. 2. Die Daten für die adressierten Empfänger werden vom System für den Versand aufbereitet und sind valid und plausibel. 3. Die ausgewählten Daten werden mit entsprechender Absichtserklärung an den Distributor versendet. 4. Für jeden adressierten Empfänger wird eine Antwort erwartet, ausgewertet und dargestellt.
Ergebnis	Unabhängig von den in der XML-Datei enthaltenen Daten kann ausgewählt werden, welche Empfänger vom Distributor beliefert werden sollen.
Fehlerbehandlung	1. Endempfänger unbekannt. Adressierung überprüfen und erneut übermitteln.

© Swissdec 2026

AB-07 Absichtserklärung	
Technische Implikationen	keine

Tabelle F.7. AB-07 Absichtserklärung

F.8 AB-08 Identifikation eines Geschäftsvorfalls

AB-08 Identifikation eines Geschäftsvorfalls	
Kurzbeschreibung	Ein Geschäftsvorfall muss eindeutig identifizierbar sein
Akteure	Distributor
Fachliche Anforderung	Ein realer Geschäftsvorfall soll für alle Teilnehmer eindeutig identifizierbar sein. Der Geschäftsvorfall umfasst alle Aktivitäten und den Austausch von Informationen im Rahmen des durchgeführten öffentlichen Geschäftsprozesses. Diese Identifikation wird für verschiedene Aufgaben benötigt (Support, Ersatzmeldung, allgemeine Verbindungen, Dokumentation usw.).
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Der Distributor vergibt eine übergreifende ID, die allen am Prozess beteiligten Teilnehmern bekannt sein muss. 2. Dem Empfängersystem muss die Identifikation des Geschäftsvorfalls bekannt sein und es muss damit auf den entsprechenden Fall zugreifen können.
Ergebnis	Der Geschäftsvorfall wird von allen an der Übermittlung beteiligten Parteien erkannt und kann eindeutig zugeordnet werden.
Fehlerbehandlung	1. Unbekannte Fallnummer. Fehlermeldung. 2. Fallnummer nicht mehr gültig. Fehlermeldung mit Statusinformation zum Geschäftsvorfall.
Technische Implikationen	keine

Tabelle F.8. AB-08 Identifikation eines Geschäftsvorfalls

F.9 AB-09 Ersatzmeldung

AB-09 Ersatzmeldung	
Kurzbeschreibung	Eine Meldung kann als Ersatzmeldung gekennzeichnet werden, um eine bereits erfolgreich übermittelte Meldung zu ersetzen.
Akteure	Sendersystem, Endreceiver
Fachliche Anforderung	Eine gesendete und erfolgreich abgeschlossene Meldung enthält systematisch falsche Daten (zum Beispiel aufgrund eines technischen Fehlers). In diesem Fall kann dafür eine Ersatzmeldung übermittelt werden.
Abhängige Bausteine	AB-05, AB-06, AB-07, AB-08
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Die ursprünglich gesendete Meldung wurde erfolgreich abgeschlossen. 2. Es wird eindeutig definiert, für wen die Ersatzmeldung bestimmt ist. 3. Es ist eindeutig ersichtlich, welche Meldung ersetzt werden muss. 4. Der Akteur löst die Ersatzmeldung aus.

© Swissdec 2026

AB-09 Ersatzmeldung	
	5. Das Endsystem erhält die Ersatzmeldung via Distributor. 6. Anhand der IDs wird die ursprüngliche Meldung identifiziert und durch die Ersatzmeldung ersetzt.
Ergebnis	Die ursprünglich übermittelte Meldung wurde durch die Ersatzmeldung ersetzt.
Fehlerbehandlung	1. Die zu ersetzende Meldung wurde nicht gefunden. Fehlermeldung. 2. Die zu ersetzende Meldung war noch nicht abgeschlossen. Fehlermeldung mit Information zum Status der offenen Meldung. 3. Die Meldung kann nicht mehr ersetzt werden, weil sie bereits vom Empfänger bearbeitet wurde. Fehlermeldung.
Technische Implikationen	DRAFT: Genauere Informationen zur Spezifikation der Ersatzmeldung folgen.

Tabelle F.9. AB-09 Ersatzmeldung

F.10 AB-10 Request und Response IDs

AB-10 Request und Response IDs	
Kurzbeschreibung	Alle Requests und Responses verfügen über eindeutige IDs.
Akteure	Sendersystem, Distributor, Endreceiver
Fachliche Anforderung	Es muss zu jedem Zeitpunkt gewährleistet sein, dass die Nachvollziehbarkeit der gesamten Kommunikation sichergestellt ist. Dies wird mit Hilfe von eindeutigen IDs auf Requests und Responses sichergestellt.
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Jeder Request und jede Response muss eindeutig identifizierbar sein (auch bei leeren Synchronisierungen, Polling, etc.) 2. Doppelt empfangene IDs müssen erkannt werden.
Ergebnis	Jeder übermittelte Request enthält eine RequestID. Jede Response enhält eine ResponseID sowie die RequestID, auf welche sie sich bezieht.
Fehlerbehandlung	Keine Vorgaben
Technische Implikationen	keine

Tabelle F.10. AB-10 Request und Response IDs

F.11 AB-11 Testmeldung

AB-11 Testmeldung	
Kurzbeschreibung	Ein Fall kann als Testfall markiert werden und wird dann entsprechend verarbeitet.
Akteure	Sendersystem, Endreceiver
Fachliche Anforderung	Es müssen Testdaten übermittelt werden können, die durch ein Element „TestCase“ gekennzeichnet sind. Dabei werden folgende Ziele verfolgt: • Ermöglichung von Tests bei der Installation und Konfiguration von produktiven Anwendungen

© Swissdec 2026

AB-11 Testmeldung	
	• Ermöglichung von Tests bei Problemen in der Produktion Dabei sollen Meldungen vom Unternehmen durch die gesamte Automatisierungskette der beteiligten Systeme (Anwendung, Transmitter, Distributor, Endempfänger und Anwendung) und ihrer Komponenten geschleust werden, **ohne einen echten Geschäftsvorfall anzustossen**(zum Beispiel ohne Rechnungen auszulösen). Es muss immer und überall (Anwendungsmasken, Output, usw.) klar ersichtlich sein, dass es sich bei dem vorliegenden Geschäftsprozess lediglich um einen Testfall handelt. Die Integrationstiefe bzw. der Prozessumfang kann meist nicht vollumfänglich abgedeckt werden. Im Vordergrund stehen dabei aber die kommunikativen Aspekte des öffentlichen Geschäftsprozesses. Die Verwendung dieses Falls ist auf Ausnahmefälle beschränkt. Eine Nutzung als Demonstrations- oder Entwicklungssystem ist nicht gestattet. Für diese Zwecke stehen die Referenzapplikationen (RefApps, ShowCase, usw.) zur Verfügung.
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Eine Geschäftsvorfall kann als Testfall markiert werden. 2. Ein Testfall muss im Laufe seines Lebenszyklus ein Testfall bleiben. 3. Ein Geschäftsvorfall kann im Laufes seines Lebenszyklus nicht zum Testfall werden. 4. Jede Meldung, die in Zusammenhang mit dem Geschäftsvorfall versendet wird, muss die Testfall-Markierung tragen.
Ergebnis	Ein Geschäftsvorfall kann von Anfang bis Ende als Testfall durchgeführt werden.
Fehlerbehandlung	1. Testfall-Markierung fehlt auf einem Testfall. Warnung. Fall muss weiterhin als Testfall verarbeitet werden. 2. Testfall-Markierung erscheint auf einem produktiven Geschäftsvorfall. Warnung. Die Meldung wird ignoriert.
Technische Implikationen	keine

Tabelle F.11. AB-11 Testmeldung

F.12 AB-12 Plausibilisierung

AB-12 Datenqualitätssicherung	
Kurzbeschreibung	Qualitätssicherung der Daten bei der Übermittlung
Akteure	Sendersystem, Distributor, Endempfänger
Fachliche Anforderung	Eine Meldung durchläuft drei Prüfungsstufen (Validierung, Plausibilisierung, Akzeptierung). Werden auf einer Prüfungsstufe keine Fehler ausgegeben, wird die Meldung an die nächste Prüfungsstufe weitergegeben (inklusive allfälliger Warnungs- und Infomeldungen). Damit soll die Qualität des gesamten Geschäftsprozesses erhöht werden, d.h. • mehr Automatisation und weniger manuelle Arbeiten • geringere Durchlaufzeiten Die Prüfungsstufen befinden sich auf folgenden Systemteilen:

© Swissdec 2026

AB-12 Datenqualitätssicherung	
	• System Distributor Die eingehenden und ausgehenden Meldungen werden gegen das XML-Schema validiert. Beim Distributor lassen sich neue Checks für den Geschäftsprozess „einfach“ einführen. Die **Plausibilisierungsregeln** müssen daher nur hier umgesetzt und geprüft werden. • System Transmitter (Client, Requester) Die eingehenden und ausgehenden Meldungen werden gegen das XML-Schema validiert. Die ausgehenden Meldungen *können* ebenfalls gemäss den definierten Plausibilisierungsregeln geprüft werden. • Endreceiver (Responder, Server) Die eingehenden und ausgehenden Meldungen werden gegen das XML-Schema validiert. Nur der Endreceiver kann die Akzeptanzstufe durchführen. Diese ist immer fachlich und wird daher hier nicht näher beschrieben.
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Jede Meldung muss eine Plausibilisierung durchlaufen. Diese findet auf dem Distributor statt. 2. Das Sendersystem kann die Meldung vor dem Versenden auf ihre Plausibilität prüfen. Dies verhindert etwaige Fehlermeldungen der Plausibilisierungsstufe nach dem Versand der Daten. Der Endreceiver muss die Plausibilisierung auf Akzeptanzstufe durchführen.
Ergebnis	Das Sendersystem erhält eine Auflistung von Fehlermeldungen und Warnungen der Plausibilisierungsstufe.
Fehlerbehandlung	Entsprechend der aufgetretenen Fehler in der Plausibilisierung werden Fehler und Warnungen ausgegeben. Das Sendersystem stellt diese dem Benutzer dar.
Technische Implikationen	keine

Tabelle F.12. AB-12 Datenqualitätssicherung

F.13 AB-13 Filterung

AB-13 Filterung	
Kurzbeschreibung	Filterung der Daten für unterschiedliche Endempfänger
Akteure	Distributor
Fachliche Anforderung	Nicht nur Daten, sondern auch Geschäftsprozesse können „redundant“ sein. In einem Geschäftsprozess sollen Daten an verschiedene Endempfänger verteilt werden (Relation: Ein Transmitter übermittelt zu mehreren Endreceivern). Eine gleichzeitige Verteilung an mehrere Endreceiver kann dann unter Umständen eine **empfängergerechte Filterung der Daten erfordern**.
Abhängige Bausteine	AB-05, AB-06, AB-07
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Keine. Die Filterung wird vom Distributor vorgenommen.
Ergebnis	Jeder Empfänger erhält nur die für ihn bestimmten Daten.
Fehlerbehandlung	Etwaige Probleme mit der Filterung werden vom Distributor geregelt. Siehe auch AB-05 und AB-06.

© Swissdec 2026

AB-13 Filterung	
Technische Implikationen	keine

Tabelle F.13. AB-13 Filterung

F.14 AB-14 Versionsmapping

AB-14 Versionsmapping	
Kurzbeschreibung	Unterstützung mehrerer Versionen eines Swissdec-Standards durch Versionsmapping
Akteure	Distributor, Endreceiver
Fachliche Anforderung	Bei Swissdec müssen verschiedene Versionen eines Standards über längere Zeit „koexistieren“. Aus diesem Grund müssen unterschiedliche Versionen „gemappt“ werden.
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Der Distributor ist verantwortlich für ein korrektes Mapping zwischen den derzeit unterstützten Versionen.
Ergebnis	Erfolgreiches Mapping zwischen derzeit unterstützten Versionen.
Fehlerbehandlung	Keine seitens Endreceiver
Technische Implikationen	keine

Tabelle F.14. AB-14 Versionsmapping

F.15 AB-15 Doublettenerkennung

AB-15 Dublettenerkennung	
Kurzbeschreibung	Erkennen und korrektes Verarbeiten von doppelten (identischen) Meldungen
Akteure	Distributor, Endreceiver
Fachliche Anforderung	In der Übermittlung von Daten können unbeabsichtigt „Wiederholungen“ identischer Meldungen entstehen. Diese müssen erkannt und im Geschäftsprozess korrekt behandelt werden.
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Die Doublettenerkennung wird vom Distributor durchgeführt. 2. Entsprechende Warnungen müssen sowohl Sender- als auch Empfängerseitig dargestellt werden.
Ergebnis	Doubletten werden vom Distributor erkannt. Entsprechende Warnungen werden Sender und Empfänger ausgegeben.
Fehlerbehandlung	Darstellung der entsprechenden Fehlermeldungen und Notifications.
Technische Implikationen	keine

Tabelle F.15. AB-15 Doublettenerkennung

© Swissdec 2026

F.16 AB-16 Anonymisierung

AB-16 Anonymisierung	
Kurzbeschreibung	Anonymisierung von Datenstrukturen
Akteure	Sendersystem, Endreceiver
Fachliche Anforderung	Ein Teilnehmer benötigt Datenstrukturen, die nur teilweise verwendet werden. Allerdings basieren diese Daten auf einer bestehenden Struktur, die nicht geändert werden kann/darf. Aus diesen Gründen müssen bestimmte Teile der Struktur anonymisiert werden.
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Eine Anonymisierung der Datenstrukturen muss gewährleistet sein, wenn ein Teilnehmer sie nicht vollständig verwenden darf. 2. Die Anonymisierung der Daten wird vom Distributor durchgeführt.
Ergebnis	Die Daten, auf die der Teilnehmer nicht zugreifen darf, sind anonymisiert.
Fehlerbehandlung	Fehlermeldungen müssen vom Teilnehmer dargestellt werden.
Technische Implikationen	keine

Tabelle F.16. AB-16 Anonymisierung

F.17 AB-17 Meldungsfragmentierung

Meldungsfragmentierung wird zum aktuellen Zeitpunkt noch nicht umgesetzt.

F.18 AB-18 Swissdec Unternehmensauthentifizierung (SUA)

AB-18 Swissdec Unternehmensauthentifizierung (SUA)	
Kurzbeschreibung	Unterstützung des SUA-Prozesses bei der Anforderung eines SUA-Zertifikats.
Akteure	Endreceiver, (Sendersystem, Distributor)
Fachliche Anforderung	Verschiedene Prozesse verlangen eine eindeutige Authentifizierung eines Unternehmens beim Endempfänger. Dies bedingt eine Signierung mit SUA-Zertifikat. Ein solches SUA-Zertifikat muss mittels des SUA-Prozesses angefordert werden können.
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	Der Endreceiver muss die Funktionalitäten zu SUA unterstützen, wenn er Standards empfängt, die SUA verlangen.
Ergebnis	SUA wird vom Endsystem gemäss Anhang C, *Detailspezifikation Swissdec Unternehmens-Authentifizierung SUA* umgesetzt.
Fehlerbehandlung	Fehlermeldungen im Rahmen des SUA-Prozesses müssen zurückgemeldet werden.
Technische Implikationen	Umsetzung gemäss Anhang C, *Detailspezifikation Swissdec Unternehmens-Authentifizierung SUA*.

Tabelle F.17. AB-18 Swissdec Unternehmensauthentifizierung (SUA)

© Swissdec 2026

F.19 SB-01 Transportsicherheit (HTTPS / TLS)

SB-01 Transportsicherheit (HTTPS / TLS)	
Kurzbeschreibung	Umsetzung der Transportsicherheit
Akteure	Distributor, Endempfänger
Fachliche Anforderung	keine
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	Der Übermittlungskanal muss verschlüsselt sein. Alle Verbindungen sind mittels TLS gesichert. Der Datenstrom zwischen allen Akteuren ist sicher.
Ergebnis	Der Übermittlungskanal zwischen Distributor und Endsystem ist verschlüsselt.
Fehlerbehandlung	Fehler in der Transportsicherheit: Abbruch der Übermittlung
Technische Implikationen	Siehe [[SECPDF]]

Tabelle F.18. SB-01 Transportsicherheit (HTTPS / TLS)

F.20 SB-02 Transport Authentication (TLS Mutual Authentication)

SB-02 Transport Authentication (TLS Mutual Authentication)	
Kurzbeschreibung	Sicherheit auf dem Transportkanal (TLS)
Akteure	Sendersystem, Distributor, Empfängersystem
Fachliche Anforderung	keine
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	Alle Teilnehmer (auch Klienten) authentifizieren sich über den Transportkanal. Wird zum Zeitpunkt des Erstellens dieses Dokumentes nur in der Verbindung vom Distributor zu den Endempfängern praktiziert.
Ergebnis	TLS Mutual Authentication ist gewährleistet.
Fehlerbehandlung	
Technische Implikationen	Siehe [[SECPDF]]

Tabelle F.19. SB-02 Transport Authentication (TLS Mutual Authentication)

F.21 SB-03 Nutzdatenverschlüsselung (WS-Encryption)

SB-03 Nutzdatenverschlüsselung (WS-Encryption)	
Kurzbeschreibung	WS-Encryption der übermittelten Daten
Akteure	Sendersystem, Distributor, Empfängersystem
Fachliche Anforderung	Keine
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	Die Vertraulichkeit der Daten ist während der Übermittlung sichergestellt.

© Swissdec 2026

SB-03 Nutzdatenverschlüsselung (WS-Encryption)	
Ergebnis	Sämtliche übermittelten Nutzdaten sind verschlüsselt.
Fehlerbehandlung	Fehler in der Verschlüsselung: Die Übermittlung wird abgebrochen.
Technische Implikationen	Siehe [[SECPDF]]

Tabelle F.20. SB-03 Nutzdatenverschlüsselung (WS-Encryption)

F.22 SB-04 Nutzdatensignatur (WS-Signature)

SB-04 Nutzdatensignatur (WS-Signature)	
Kurzbeschreibung	WS-Signierung der übermittelten Daten
Akteure	Sendersystem, Distributor, Empfängersystem
Fachliche Anforderung	keine
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	Die Datenintegrität sowie die Authentisierung der Quelle der Meldung sind sichergestellt.
Ergebnis	Sämtliche übermittelten Nutzdaten sind signiert.
Fehlerbehandlung	Fehler bei der Nachrichtensicherheitsüberprüfung. Abbruch der Übermittlung.
Technische Implikationen	Siehe [[SECPDF]]

Tabelle F.21. SB-04 Nutzdatensignatur (WS-Signature)

F.23 SB-05 Unternehmens-Authentifizierung SUA Signatur (WS-Signatur)

SB-05 Unternehmens-Authentifizierung SUA Signatur (WS-Signatur)	
Kurzbeschreibung	Unterstützung des SUA-Prozesses seitens Endempfänger
Akteure	Empfängersystem
Fachliche Anforderung	keine
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	SUA wird gemäss Spezifikation umgesetzt.
Ergebnis	Die Informationen zum SUA-Zertifikat werden zurückgeliefert.
Fehlerbehandlung	Der Antragssteller ist unbekannt. Fehlermeldung. Es werden keine Informationen geliefert.
Technische Implikationen	Siehe [[SECPDF]]

Tabelle F.22. SB-05 Unternehmens-Authentifizierung SUA Signatur (WS-Signatur)

F.24 SB-06 Nichtabstreitbarkeit

SB-06 Nichtabstreitbarkeit	
Kurzbeschreibung	Eindeutige Zuordnung an spezifische Teilnehmer

© Swissdec 2026

SB-06 Nichtabstreitbarkeit	
Akteure	Sendersystem, Distributor, Empfängersystem
Fachliche Anforderung	keine
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	Alle Meldungen sind eindeutig einem spezifischen Teilnehmer zugeordnet und können im Nachhinein nicht geleugnet werden.
Ergebnis	Die Nichtabstreitbarkeit ist gewährleistet.
Fehlerbehandlung	
Technische Implikationen	Siehe [[SECPDF]] und Anhang C, *Detailspezifikation Swissdec Unternehmens-Authentifizierung SUA*

Tabelle F.23. SB-06 Nichtabstreitbarkeit

F.25 PB-01 Synchrone Verteilung

PB-01: Synchrone Verteilung	
Kurzbeschreibung	Synchrone Übermittlung
Akteure	Sendersystem
Fachliche Anforderung	Die Meldungen werden direkt über den Distributor an alle adressierten Empfänger geschickt. Die Rückantworten werden gesammelt und an den Sender als Quittung zurückgeliefert.
Abhängige Bausteine	Alle Architekturbausteine und Securitybausteine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	keine
Ergebnis	Die Meldungen wurden verteilt, die Antworten zurückgeliefert und alle fachlichen Daten auf dem Distributor gelöscht.
Fehlerbehandlung	
Technische Implikationen	keine

Tabelle F.24. PB-01 Synchrone Verteilung

F.26 PB-02 Asynchrone Verteilung

PB-02: Asynchrone Verteilung	
Kurzbeschreibung	Asynchrone Übermittlung mittels Polling
Akteure	Sendersystem, Distributor
Fachliche Anforderung	Die Meldungen werden direkt vom Distributor an alle adressierten Empfänger geschickt. Die Rückantworten werden gesammelt und der Sender kann diese mit dem JobKey im GetStatus abholen.
Abhängige Bausteine	Alle Konstruktionsbausteine und Securitybausteine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	keine

© Swissdec 2026

PB-02: Asynchrone Verteilung	
Ergebnis	Die Meldungen wurden verteilt, die Antworten zurückgeliefert und alle fachlichen Daten gelöscht.
Fehlerbehandlung	
Technische Implikationen	keine

Tabelle F.25. PB-02 Asynchrone Verteilung

F.27 PB-03 Synchronisierung

PB-03: Synchronisierung	
Kurzbeschreibung	Synchronisierung zur Aktualisierung der Fälle
Akteure	Sendersystem
Fachliche Anforderung	Die am Geschäftsprozess beteiligten Systeme müssen über gemeinsame Informationen verfügen, damit sie korrekt zusammenarbeiten können. Dies wird durch eine Synchronisierung von Daten (Story) aufgebaut und regelmässig aktualisiert.
Abhängige Bausteine	
Ausschliessende Bausteine	
Anforderungen an Akteure	Die Akteure müssen die Fähigkeit besitzen, ihre Ereignisse (Stories) bereitzustellen (gemäss Spezifikation in Umfang und Aktualität).
Ergebnis	Alle Akteure besitzen genügend Informationen, um damit ihren korrekten Teil am Geschäftsprozess zu erfüllen.
Fehlerbehandlung	Im Fall von Verlusten oder Unstimmigkeiten sollte eine Wiederherstellung möglich sein.
Technische Implikationen	Es sollte eine einfache Art von „Story-Confirmation“ ermöglicht werden. Zusätzlich muss zwischen den technischen und fachlichen Quittungen unterschieden werden. Der Datenaustausch zwischen Unternehmen und Versicherer erfolgt in strukturierter Form während des gesamten Fallverlaufs, bei dem beide Seiten den gleichen Informationsstand zum Ereignis aufrechterhalten.

Tabelle F.26. PB-03 Synchronisierung

F.28 PB-04 Dialogisierung

PB-04: Dialogisierung	
Kurzbeschreibung	Umsetzung des DialogMessage-Prozesses
Akteure	Sendersystem, Empfängersystem
Fachliche Anforderung	Ein einfacher und dynamischer Austausch von Informationen sollte jederzeit möglich sein, d.h. grössere IT-Unterstützung ist dann nicht erforderlich (z.B. eine Art „Chat“ mit optionalen Strukturen). Die Benutzung ist mindestens manuell durch den Endbnutzer sicherzustellen. Es kann aber durch eine offizielle Publikation der Definition später auch automatisiert werden. Weitere Details finden sich bei der Swissdec DialogMessage.
Abhängige Bausteine	
Ausschliessende Bausteine	

© Swissdec 2026

PB-04: Dialogisierung	
Anforderungen an Akteure	Innerhalb der Automatisierung muss bei neuen DialogMessages auch eine manuelle Verarbeitung möglich sein.
Ergebnis	Alle Akteure besitzen auch kurzfristig und dynamisch über genügend Informationen, um damit ihren korrekten Teil am Geschäftsprozess zu leisten.
Fehlerbehandlung	
Technische Implikationen	Eine sehr dynamische Lösung muss mit entsprechender Interoperabilität, Sicherheit und Datenschutz betrieben werden.

Tabelle F.27. PB-04 Dialogisierung

F.29 PB-05 Completion

PB-05: Completion	
Kurzbeschreibung	Umsetzung des Freigabeprozesses
Akteure	Sendersystem, Endempfänger
Fachliche Anforderung	Es wird ein dynamischer Datenaustausch verlangt, d.h. es sollte ohne Anpassungen der Anwendungen und des Transports in eine separate Webapplication verzweigt werden. Hier kann der Benutzer manuell alle Eingaben und Kontrollen durchführen. Es stehen alle gängigen Browserfunktionen zur Verfügung.
Abhängige Bausteine	
Ausschliessende Bausteine	
Anforderungen an Akteure	Einerseits muss dazu eine Webapplication zur Verfügung stehen. Diese Webapplication muss mit dem aktuellen Protokoll verbunden sein, sofern eine Prozessabhängigkeit besteht. Die aktuelle Anwendung verliert sonst den Ablauf. Andererseits muss ein entsprechender Browser mit URL automatisch gestartet werden. Die Sicherheit (Pfad, login usw.) muss dabei immer gewährleistet sein. Weitere Details finden sich bei der Swissdec ELM Completion.
Ergebnis	
Fehlerbehandlung	
Technische Implikationen	Die Erfahrungen aus der ELM Completion übernehmen. Es ist aber auch möglich einen Link zu einem weiteren System zu vermitteln (ohne Browser; siehe auch KLE Cross Channel Link)

Tabelle F.28. PB-05 Completion

F.30 PB-06 Prozessnavigation

PB-06: Prozessnavigation	
Kurzbeschreibung	Umsetzung der Prozessnavigation
Akteure	Distributor, Endempfänger
Fachliche Anforderung	In komplexen Abläufen ist auf der Ebene Geschäftsprozess oft eine Navigation wichtig. Es gibt vom Receiver eine fallbezogene, dynamische Steuerung. Der Transmitter bzw. der Endbenutzer muss darüber informiert werden.
Abhängige Bausteine	Es besteht evtl. eine semantische Abhängigkeit oder Überlappung mit den möglichen **Protokoll-Zuständen**.

© Swissdec 2026

PB-06: Prozessnavigation	
Ausschliessende Bausteine	
Anforderungen an Akteure	Diese Navigation muss geführt und übermittelt werden.
Ergebnis	Alle Teilnehmer können sich dadurch orientieren und gewinnen damit mehr Verständnis, Gewissheit und Vertrauen.
Fehlerbehandlung	
Technische Implikationen	Sofern keine Historisierung des Pfades [a] notwendig ist, sollte keine Story verwendetet werden.

[a] alle zeitlichen Zustände oder Orte aufzeichnen

Tabelle F.29. PB-06 Prozessnavigation

F.31 PB-07 Digitalisierungsbereich

PB-07: Digitalisierungsbereich	
Kurzbeschreibung	Unterstützung der verschiedenen Digitalisierungsbereiche
Akteure	Sendersystem, Empfängersystem
Fachliche Anforderung	Grundsätzlich sollte der gesamte Prozess digital erledigt werden. Der Digitalisierungsbereich soll aber die Kommunikation bzw. den Prozessablauf von ungleichen Teilnehmern akzeptabel steuern (Usability). Ein Digitalisierungsbereich erlaubt die internen Prozesse beim Teilnehmer schrittweise anzupassen. Damit wird die notwendige Tiefe einer Integration in die Quellsysteme flexibler.
Abhängige Bausteine	
Ausschliessende Bausteine	
Anforderungen an Akteure	Jeder muss die unterschiedlichen Digitalisierungsbereiche kontrollieren und steuern. Dies erhöht die Komplexität im Prozess und Protokoll erheblich.
Ergebnis	Unterschiedliche Automatisierungen der Teilnehmer sind erlaubt und ein Standard kann sich unter Umständen schneller durchsetzen (z.B. nur bei grossen Mengen ist der volle Bereich sinnvoll und bei kleinen Mengen wird teilweise manuell gearbeitet).
Fehlerbehandlung	
Technische Implikationen	Sollten beide Seiten Digitalisierungsbereiche verlangen, dann kann es sehr schnell überaus komplex werden und am Ende versteht niemand mehr diesen Prozess.

Tabelle F.30. PB-07 Digitalisierungsbereich

F.32 PB-08 Daten anfordern

PB-08: Daten anfordern	
Kurzbeschreibung	Anfordern von Stories beim Sendersystem
Akteure	Sendersystem, Endempfänger
Fachliche Anforderung	Aus Gründen des Datenschutzes sollten in einem Prozess nur die dafür notwendigen Informationen ausgetauscht werden. Oft sind diese bei dynamischem Verlauf am Anfang des Übermittlungsprozesses nicht bekannt. Aus diesem Grund

© Swissdec 2026

PB-08: Daten anfordern	
	können weitere definierte Daten zu einem späteren Zeitpunkt verlangt werden. Für jede verlangte Story lässt sich optional noch ein Fristdatum setzen.
Abhängige Bausteine	
Ausschliessende Bausteine	
Anforderungen an Akteure	Einerseits müssen diese Stories dynamisch angefordert werden. Andererseits müssen diese angeforderten Daten aufbereitet und als Stories übermittelt werden.
Ergebnis	
Fehlerbehandlung	
Technische Implikationen	Die Synchronisation muss einerseits die angeforderten X-Stories definieren. Andererseits muss mit einer speziellen AwaitStory/X-Story diese definiert werden, damit diese dann angefordert werden kann.

Tabelle F.31. PB-08 Daten anfordern

© Swissdec 2026

G Technische Dokumentationen

G.1 Schema Dokumentation IncidentDeclarationConsumerServiceTypes.xsd

Target Namespace	urn:ch:swissdec:kle:v2:20260306:incidentdeclaration:consumer:service:types
Declared Namespaces	• c : urn:ch:swissdec:common:v3:20260306 • idcc : urn:ch:swissdec:kle:v2:20260306:incidentdeclaration:consumer:container • ep : urn:ch:swissdec:basis:v1:20260306:components • xs : http://www.w3.org/2001/XMLSchema • idcst : urn:ch:swissdec:kle:v2:20260306:incidentdeclaration:consumer:service:types
Version	0.0

G.2 Schema Dokumentation IncidentDeclarationConsumerContainer.xsd

Target Namespace	urn:ch:swissdec:kle:v2:20260306:incidentdeclaration:consumer:container
Declared Namespaces	• c : urn:ch:swissdec:common:v3:20260306 • idc : urn:ch:swissdec:kle:v2:20260306:incidentdeclaration:container • idcc : urn:ch:swissdec:kle:v2:20260306:incidentdeclaration:consumer:container • ep : urn:ch:swissdec:basis:v1:20260306:components • id : urn:ch:swissdec:kle:v2:20260306:incidentdeclaration • xs : http://www.w3.org/2001/XMLSchema
Version	0.0

ComplexType: CompanyConsumerRequestType

Abstract	no
Parent type	„ComplexType: AddressingType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: AddressingType“] <Institution> „ComplexType: AddressingIdentificationType“ </Institution> [1..1] [END BASE TYPE] <WithDelegate> „ComplexType: EmptyType“ </WithDelegate> [0..1] </...>

ComplexType: DeclareIncidentConsumerRequestType

Abstract	no
XML Instance Representation	<...> <DeclareIncidentDistributorRequestContext> „ComplexType: DeclareIncidentDistributorRequestContextType DeclareIncidentDistributorRequestContext> [1..1] <DeclareIncident> „ComplexType: DeclareIncidentRequestType“ </DeclareInci\ dent> [1..1] </...>

ComplexType: DeclareIncidentConsumerResponseType

Abstract	no
XML Instance Representation	<...> <DeclareIncidentResponse> „ComplexType: DeclareIncidentAddresseeSuccessJobStateType“ </ DeclareIncidentResponse> [1..1] </...>

ComplexType: DeclareIncidentDistributorRequestContextType

Abstract	no

© Swissdec 2026

Parent type	„ComplexType: DistributorRequestContextType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: DistributorRequestContextType“] <UserAgent> „ComplexType: UserAgentType“ </UserAgent> [1..1] <TransmissionDate>xs:dateTime</TransmissionDate> [1..1] <DistributorRequestID> „SimpleType: IDType“ </DistributorRequestID> [1..1] <VersionMappingFrom> „ComplexType: VersionMappingFromType“ </VersionMapping\ From> [0..1] <ProducerSecurityTokens> „ComplexType: ProducerSecurityTokensType“ </ProducerSecu\ rityTokens> [1..1] <SignatureCertificateUID-BFS> „ComplexType: SignatureCertificateUID-BFSType“ </ SignatureCertificateUID-BFS> [0..1] <ProducerResponseNotifications> „ComplexType: FeedbackNotificationsType“ </Produ\ cerResponseNotifications> [1..1] <ConsumerNotifications> „ComplexType: FeedbackNotificationsType“ </ConsumerNotifi\ cations> [0..1] [END BASE TYPE] <IncidentCaseID> „SimpleType: IDType“ </IncidentCaseID> [1..1] </...>

ComplexType: RegisterOrganizationAuthenticationAddresseeType

Abstract	no
Parent type	„ComplexType: SynchronizeRequestType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: SynchronizeRequestType“] [BASE TYPE „ComplexType: RequestType“] <RequestContext> „ComplexType: RequestContextType“ </RequestContext> [1..1] [END BASE TYPE] <Sender> „ComplexType: CompanyUIDType“ </Sender> [1..1] [END BASE TYPE] <Addressee> „ComplexType: AddressingAddresseeType“ </Addressee> [1..1] <Case> „ComplexType: RegisterOrganizationAuthenticationAddresseeRequestCaseType“ </ Case> [1..1] </...>

ComplexType: RegisterOrganizationAuthenticationConsumerRequestType

Abstract	no
XML Instance Representation	<...> <RegisterOrganizationAuthenticationDistributorRequestContext> „ComplexType: RegisterOrganizationAuthen RegisterOrganizationAuthenticationDistributorRequestContext> [1..1] <RegisterOrganizationAuthentication> „ComplexType: RegisterOrganizationAuthenticationConsumerType“ </ RegisterOrganizationAuthentication> [1..1] </...>

ComplexType: RegisterOrganizationAuthenticationConsumerType

Abstract	no
Parent type	„ComplexType: RequestType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: RequestType“] <RequestContext> „ComplexType: RequestContextType“ </RequestContext> [1..1] [END BASE TYPE] <Job> „ComplexType: RegisterOrganizationAuthenticationJobType“ </Job> [1..1] <Organization> „ComplexType: CompanyConsumerRequestType“ </Organization> [1..1] </...>

ComplexType: RegisterOrganizationAuthenticationDistributorRequestContextType

Abstract	no
Parent type	„ComplexType: DistributorRequestContextType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: DistributorRequestContextType“] <UserAgent> „ComplexType: UserAgentType“ </UserAgent> [1..1] <TransmissionDate>xs:dateTime</TransmissionDate> [1..1] <DistributorRequestID> „SimpleType: IDType“ </DistributorRequestID> [1..1] <VersionMappingFrom> „ComplexType: VersionMappingFromType“ </VersionMapping\ From> [0..1] <ProducerSecurityTokens> „ComplexType: ProducerSecurityTokensType“ </ProducerSecu\ rityTokens> [1..1] <SignatureCertificateUID-BFS> „ComplexType: SignatureCertificateUID-BFSType“ </ SignatureCertificateUID-BFS> [0..1] <ProducerResponseNotifications> „ComplexType: FeedbackNotificationsType“ </Produ\ cerResponseNotifications> [1..1]

© Swissdec 2026

	<ConsumerNotifications> „ComplexType: FeedbackNotificationsType“ </ConsumerNotifi\ cations> [0..1] [END BASE TYPE] <CertificateRequestID> „SimpleType: IDType“ </CertificateRequestID> [1..1] </...>

ComplexType: RegisterOrganizationAuthenticationJobType

Abstract	no
Translation	Job Steuerung
Technical description	Steuerungselement für die Verarbeitung der Meldung.
XML Instance Representation	<...> <Addressee> „ComplexType: AddresseeJobType“ </Addressee> [1..1] <TestCase> „ComplexType: EmptyType“ </TestCase> [0..1] </...>

ComplexType: SubscribeOrganizationConsumerRequestType

Abstract	no
XML Instance Representation	<...> <SubscribeOrganizationDistributorRequestContext> „ComplexType: SubscribeOrganizationDistributorReques\ SubscribeOrganizationDistributorRequestContext> [1..1] <SubscribeOrganization> „ComplexType: SubscribeOrganizationRequestType“ </Subscribe\ Organization> [1..1] </...>

ComplexType: SubscribeOrganizationConsumerResponseType

Abstract	no
XML Instance Representation	<...> <SubscribeOrganizationResponse> „ComplexType: SubscribeOrganizationAddresseeSuccessJobStateType“ </\ SubscribeOrganizationResponse> [1..1] </...>

ComplexType: SubscribeOrganizationDistributorRequestContextType

Abstract	no
Parent type	„ComplexType: DistributorRequestContextType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: DistributorRequestContextType“] <UserAgent> „ComplexType: UserAgentType“ </UserAgent> [1..1] <TransmissionDate>xs:dateTime</TransmissionDate> [1..1] <DistributorRequestID> „SimpleType: IDType“ </DistributorRequestID> [1..1] <VersionMappingFrom> „ComplexType: VersionMappingFromType“ </VersionMapping\ From> [0..1] <ProducerSecurityTokens> „ComplexType: ProducerSecurityTokensType“ </ProducerSecu\ rityTokens> [1..1] <SignatureCertificateUID-BFS> „ComplexType: SignatureCertificateUID-BFSType“ </ SignatureCertificateUID-BFS> [0..1] <ProducerResponseNotifications> „ComplexType: FeedbackNotificationsType“ </Produ\ cerResponseNotifications> [1..1] <ConsumerNotifications> „ComplexType: FeedbackNotificationsType“ </ConsumerNotifi\ cations> [0..1] [END BASE TYPE] <SubscriptionID> „SimpleType: IDType“ </SubscriptionID> [1..1] </...>

ComplexType: SynchronizeDeclareIncidentConsumerRequestType

Abstract	no
XML Instance Representation	<...> <DistributorRequestContext> „ComplexType: DistributorRequestContextType“ </Distribu\ torRequestContext> [1..1] <SynchronizeDeclareIncident> „ComplexType: SynchronizeDeclareIncidentRequestType“ </ SynchronizeDeclareIncident> [1..1] </...>

ComplexType: SynchronizeDeclareIncidentConsumerResponseType

Abstract	no

© Swissdec 2026

XML Instance Representation	<...> <SynchronizeDeclareIncidentResponse> „ComplexType: SynchronizeDeclareIncidentConsumerType“ </ SynchronizeDeclareIncidentResponse> [1..1] </...>

ComplexType: SynchronizeRegisterOrganizationAuthenticationConsumer

Abstract	no
Parent type	„ComplexType: SynchronizeRequestType“

XML Instance Representation	<...>
	  [BASE TYPE „ComplexType: SynchronizeRequestType“]
	  [BASE TYPE „ComplexType: RequestType“]
	  <RequestContext> „ComplexType: RequestContextType“ </RequestContext> [1..1]
	  [END BASE TYPE]
	  <Sender> „ComplexType: CompanyUIDType“ </Sender> [1..1]
	  [END BASE TYPE]
	</...>

ComplexType: SynchronizeRegisterOrganizationAuthenticationConsumerRequest-Type

Abstract	no

XML Instance Representation	<...>
	  <RegisterOrganizationAuthenticationDistributorRequestContext> „ComplexType: RegisterOrganizationAuthe
	  RegisterOrganizationAuthenticationDistributorRequestContext> [1..1]
	  <RegisterOrganizationAuthentication> „ComplexType: RegisterOrganizationAuthenticationAddresseeType“ <
	  RegisterOrganizationAuthentication> [1..1]
	</...>

ComplexType: SynchronizeSubscribeOrganizationConsumerRequestType

Abstract	no

XML Instance Representation	<...>
	  <DistributorRequestContext> „ComplexType: DistributorRequestContextType“ </Distribu\
	  torRequestContext> [1..1]
	  <SynchronizeSubscribeOrganization> „ComplexType: SynchronizeSubscribeOrganizationRequestType“ </
	  SynchronizeSubscribeOrganization> [1..1]
	</...>

ComplexType: SynchronizeSubscribeOrganizationConsumerResponseType

Abstract	no
XML Instance Representation	<...> <SynchronizeSubscribeOrganizationResponse> „ComplexType: SynchronizeSubscribeOrganizationConsumerType SynchronizeSubscribeOrganizationResponse> [1..1] </...>

G.3 Schema Dokumentation IncidentDeclarationContainer.xsd

Target Namespace	urn:ch:swissdec:kle:v2:20260306:incidentdeclaration:container
Declared Namespaces	• c : urn:ch:swissdec:common:v3:20260306 • idc : urn:ch:swissdec:kle:v2:20260306:incidentdeclaration:container • ep : urn:ch:swissdec:basis:v1:20260306:components • id : urn:ch:swissdec:kle:v2:20260306:incidentdeclaration • xs : http://www.w3.org/2001/XMLSchema
Version	0.0

ComplexType: AvailableType

Abstract	no
Technical description	Informationen zur Verfügbarkeit.
XML Instance Representation	<...> <DeclareIncident> „ComplexType: DeclareIncidentAvailableType“ </DeclareInci\ dent> [0..1]

© Swissdec 2026

	</...>

ComplexType: CompanyRequestType

Abstract	no
Parent type	„ComplexType: AddressingType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: AddressingType“] <Institution> „ComplexType: AddressingIdentificationType“ </Institution> [1..1] [END BASE TYPE] <CompanyDescription> „ComplexType: CompanyDescriptionBaseType“ </CompanyDescripti\ on> [1..1] <Contact> „ComplexType: ContactType“ </Contact> [1..1] </...>

ComplexType: DeclareIncidentAddresseeContextType

Abstract	no
Parent type	„ComplexType: AddresseeResponseContextType“
Technical description	Kontextangaben zum Adressaten, wie Informationen zum Zeitpunkt der Übermittlung, dem Absender und der Sprache.
XML Instance Representation	<...> [BASE TYPE „ComplexType: AddresseeResponseContextType“] [BASE TYPE „ComplexType: ResponseContextType“] <UserAgent> „ComplexType: UserAgentType“ </UserAgent> [1..1] <InstitutionName>xs:string</InstitutionName> [1..1] <TransmissionDate>xs:dateTime</TransmissionDate> [1..1] <ResponseID> „SimpleType: IDType“ </ResponseID> [1..1] <RequestID> „SimpleType: IDType“ </RequestID> [1..1] [END BASE TYPE] <ProducerResponseNotifications> „ComplexType: FeedbackNotificationsType“ </Produ\ cerResponseNotifications> [1..1] <Warning> „ComplexType: NotificationsType“ </Warning> [0..1] <Info> „ComplexType: NotificationsType“ </Info> [0..1] [END BASE TYPE] <IncidentCaseID> „SimpleType: IDType“ </IncidentCaseID> [1..1] <InsuranceCaseID> „SimpleType: IDType“ </InsuranceCaseID> [1..1] <CompanyCaseID> „SimpleType: IDType“ </CompanyCaseID> [1..1] <TestCase> „ComplexType: EmptyType“ </TestCase> [0..1] </...>

ComplexType: DeclareIncidentAddresseeJobStateType

Abstract	no
Parent type	„ComplexType: AddresseeJobType“
Translation	OpenCase Status
Short description	Initialer Status des Falls.
XML Instance Representation	<...> [BASE TYPE „ComplexType: AddresseeJobType“] [BASE TYPE „ComplexType: AddresseeType“] <AddresseeIdentification> „SimpleType: IDType“ </AddresseeIdentificati\ on> [1..1] [END BASE TYPE] <ProcessByDistributor> „SimpleType: SimpleBooleanType“ </ProcessByDistribu\ tor> [1..1] [END BASE TYPE] [START CHOICE] <Success> „ComplexType: DeclareIncidentAddresseeSuccessJobStateType“ </Suc\ cess> [1..1] [END CHOICE] </...>

ComplexType: DeclareIncidentAddresseeSuccessJobStateType

Abstract	no
Translation	OpenCase Status
Short description	Initialer Status des Falls.
XML Instance Representation	<...>

© Swissdec 2026

	<AddresseeContext> „ComplexType: DeclareIncidentAddresseeContextType“ </Addressee\ Context> [1..1] </.../>

ComplexType: DeclareIncidentAddresseesType

Abstract	no
Technical description	Angaben zum Adressaten.
XML Instance Representation	<.../> <Addressee> „ComplexType: AddresseeJobType“ </Addressee> [0..unbounded] </.../>

ComplexType: DeclareIncidentAvailableType

Abstract	no
Technical description	Informationen zur Verfügbarkeit.
XML Instance Representation	<.../> <IncidentCaseID> „SimpleType: IDType“ </IncidentCaseID> [1..unbounded] </.../>

ComplexType: DeclareIncidentCaseContextType

Abstract	no
Parent type	„ComplexType: CaseContextBaseType“
Translation	Kontext des Falls
Technical description	Dieses Element enthält Informationen und Identifikatoren zum Fall.
XML Instance Representation	<.../> [BASE TYPE „ComplexType: CaseContextBaseType“] <ReceivedStoryIDs> „ComplexType: ReceivedStoriesType“ </ReceivedStoryIDs> [0..1] <SuppressedSenderStoryIDs> „ComplexType: ReceivedStoriesType“ </SuppressedSenderS\ toryIDs> [0..1] <SuppressedInstitutionStoryIDs> „ComplexType: ReceivedStoriesType“ </SuppressedIn\ stitutionStoryIDs> [0..1] <Warning> „ComplexType: StoryNotificationsType“ </Warning> [0..1] <Info> „ComplexType: StoryNotificationsType“ </Info> [0..1] [END BASE TYPE] <IncidentCaseID> „SimpleType: IDType“ </IncidentCaseID> [1..1] <InsuranceCaseID> „SimpleType: IDType“ </InsuranceCaseID> [1..1] <CompanyCaseID> „SimpleType: IDType“ </CompanyCaseID> [1..1] <TestCase> „ComplexType: EmptyType“ </TestCase> [0..1] </.../>

ComplexType: DeclareIncidentCaseType

Abstract	no
Translation	Fall
Technical description	Informationen zum Fall.
XML Instance Representation	<.../> <CaseContext> „ComplexType: DeclareIncidentCaseContextType“ </CaseContext> [1..1] <ReceivedState> „ComplexType: DeclareIncidentStateType“ </ReceivedState> [0..1] <DialogMessage> „ComplexType: DialogMessageType“ </DialogMessage> [0..unbounded] </.../>

ComplexType: DeclareIncidentConsumerCaseType

Abstract	no
Translation	Fall
Technical description	Informationen zum Fall.
XML Instance Representation	<.../> <CaseContext> „ComplexType: DeclareIncidentCaseContextType“ </CaseContext> [1..1] <State> „ComplexType: DeclareIncidentStateType“ </State> [1..1] <DialogMessage> „ComplexType: DialogMessageType“ </DialogMessage> [0..unbounded] </.../>

© Swissdec 2026

ComplexType: DeclareIncidentJobStateType

Abstract	no
Translation	OpenCase Status
Short description	Initialer Status des Falls.
XML Instance Representation	<...> <Addressee> „ComplexType: DeclareIncidentAddresseeJobStateType“ </ Addressee> [0..unbounded] </...>

ComplexType: DeclareIncidentJobType

Abstract	no
Translation	Job Steuerung
Technical description	Steuerungselement für die Verarbeitung der Meldung.
XML Instance Representation	<...> <Addressees> „ComplexType: DeclareIncidentAddresseesType“ </Addressees> [1..1] <TestCase> „ComplexType: EmptyType“ </TestCase> [0..1] </...>

ComplexType: DeclareIncidentRequestType

Abstract	no
Parent type	„ComplexType: RequestType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: RequestType“] <RequestContext> „ComplexType: RequestContextType“ </RequestContext> [1..1] [END BASE TYPE] <Job> „ComplexType: DeclareIncidentJobType“ </Job> [1..1] <null>idc:anonymous</null> [1..1] </...>

ComplexType: DeclareIncidentResponseType

Abstract	no
Parent type	„ComplexType: ResponseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: ResponseType“] <ResponseContext> „ComplexType: ResponseContextType“ </ResponseContext> [1..1] [END BASE TYPE] <Addressees> „ComplexType: DeclareIncidentJobStateType“ </Addressees> [1..1] </...>

ComplexType: DeclareIncidentStateType

Abstract	no
Translation	OpenCase Status
Short description	Initialer Status des Falls.
XML Instance Representation	<...> <Process> „SimpleType: DeclareIncidentStateEnumType“ </Process> [1..1] <Coverage> „SimpleType: CoverageType“ </Coverage> [1..1] <SmallRemark> „SimpleType: NotEmptyStringType“ </SmallRemark> [0..1] </...>

ComplexType: RegisterOrganizationAuthenticationJobType

Abstract	no
Translation	Job Steuerung
Technical description	Steuerungselement für die Verarbeitung der Meldung.
XML Instance Representation	<...>

© Swissdec 2026

	<Addressee> „ComplexType: AddresseeJobType“ </Addressee> [1..1] <TestCase> „ComplexType: EmptyType“ </TestCase> [0..1] </...>

ComplexType: RegisterOrganizationAuthenticationRequestType

Abstract	no
Parent type	„ComplexType: RequestType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: RequestType“] <RequestContext> „ComplexType: RequestContextType“ </RequestContext> [1..1] [END BASE TYPE] <Job> „ComplexType: RegisterOrganizationAuthenticationJobType“ </Job> [1..1] <RegisterOrganization> „ComplexType: CompanyRequestType“ </RegisterOrganizati\ on> [1..1] </...>

ComplexType: SubscribeOrganizationAddresseeContextType

Abstract	no
Parent type	„ComplexType: AddresseeResponseContextType“
Technical description	Kontextangaben zum Adressaten, wie Informationen zum Zeitpunkt der Übermittlung, dem Absender und der Sprache.
XML Instance Representation	<...> [BASE TYPE „ComplexType: AddresseeResponseContextType“] [BASE TYPE „ComplexType: ResponseContextType“] <UserAgent> „ComplexType: UserAgentType“ </UserAgent> [1..1] <InstitutionName>xs:string</InstitutionName> [1..1] <TransmissionDate>xs:dateTime</TransmissionDate> [1..1] <ResponseID> „SimpleType: IDType“ </ResponseID> [1..1] <RequestID> „SimpleType: IDType“ </RequestID> [1..1] [END BASE TYPE] <ProducerResponseNotifications> „ComplexType: FeedbackNotificationsType“ </Produ\ cerResponseNotifications> [1..1] <Warning> „ComplexType: NotificationsType“ </Warning> [0..1] <Info> „ComplexType: NotificationsType“ </Info> [0..1] [END BASE TYPE] <SubscriptionID> „SimpleType: IDType“ </SubscriptionID> [1..1] </...>

ComplexType: SubscribeOrganizationAddresseeJobStateType

Abstract	no
Parent type	„ComplexType: AddresseeJobType“
Translation	OpenCase Status
Short description	Initialer Status des Falls.
XML Instance Representation	<...> [BASE TYPE „ComplexType: AddresseeJobType“] [BASE TYPE „ComplexType: AddresseeType“] <AddresseeIdentification> „SimpleType: IDType“ </AddresseeIdentificati\ on> [1..1] [END BASE TYPE] <ProcessByDistributor> „SimpleType: SimpleBooleanType“ </ProcessByDistribu\ tor> [1..1] [END BASE TYPE] [START CHOICE] <Success> „ComplexType: SubscribeOrganizationAddresseeSuccessJobStateType“ </Suc\ cess> [1..1] [END CHOICE] </...>

ComplexType: SubscribeOrganizationAddresseeSuccessJobStateType

Abstract	no
Translation	OpenCase Status
Short description	Initialer Status des Falls.
XML Instance Representation	<...> <AddresseeContext> „ComplexType: SubscribeOrganizationAddresseeContextType“ </ AddresseeContext> [1..1]

© Swissdec 2026

	</...>

ComplexType: SubscribeOrganizationCaseContextType

Abstract	no
Parent type	„ComplexType: CaseContextBaseType“
Translation	Kontext des Falls
Technical description	Dieses Element enthält Informationen und Identifikatoren zum Fall.
XML Instance Representation	<...> [BASE TYPE „ComplexType: CaseContextBaseType“] <ReceivedStoryIDs> „ComplexType: ReceivedStoriesType“ </ReceivedStoryIDs> [0..1] <SuppressedSenderStoryIDs> „ComplexType: ReceivedStoriesType“ </SuppressedSenderS\ toryIDs> [0..1] <SuppressedInstitutionStoryIDs> „ComplexType: ReceivedStoriesType“ </SuppressedIn\ stitutionStoryIDs> [0..1] <Warning> „ComplexType: StoryNotificationsType“ </Warning> [0..1] <Info> „ComplexType: StoryNotificationsType“ </Info> [0..1] [END BASE TYPE] <SubscriptionID> „SimpleType: IDType“ </SubscriptionID> [1..1] </...>

ComplexType: SubscribeOrganizationCaseType

Abstract	no
Translation	Fall
Technical description	Informationen zum Fall.
XML Instance Representation	<...> <CaseContext> „ComplexType: SubscribeOrganizationCaseContextType“ </CaseCon\ text> [1..1] <ReceivedState> „SimpleType: SubscribeOrganizationStateType“ </ReceivedSta\ te> [0..1] <DialogMessage> „ComplexType: DialogMessageType“ </DialogMessage> [0..unbounded] <Unsubscribe> „ComplexType: EmptyType“ </Unsubscribe> [0..1] </...>

ComplexType: SubscribeOrganizationConsumerCaseType

Abstract	no
Translation	Fall
Technical description	Informationen zum Fall.
XML Instance Representation	<...> <CaseContext> „ComplexType: SubscribeOrganizationCaseContextType“ </CaseCon\ text> [1..1] <State> „SimpleType: SubscribeOrganizationStateType“ </State> [1..1] <DialogMessage> „ComplexType: DialogMessageType“ </DialogMessage> [0..unbounded] </...>

ComplexType: SubscribeOrganizationJobStateType

Abstract	no
Translation	OpenCase Status
Short description	Initialer Status des Falls.
XML Instance Representation	<...> <Addressee> „ComplexType: SubscribeOrganizationAddresseeJobStateType“ </Addres\ see> [1..1] </...>

ComplexType: SubscribeOrganizationJobType

Abstract	no
Translation	Job Steuerung
Technical description	Steuerungselement für die Verarbeitung der Meldung.

© Swissdec 2026

XML Instance Representation	<...> <Addressee> „ComplexType: AddresseeJobType“ </Addressee> [1..1] </...>

ComplexType: SubscribeOrganizationRequestType

Abstract	no
Parent type	„ComplexType: RequestType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: RequestType“] <RequestContext> „ComplexType: RequestContextType“ </RequestContext> [1..1] [END BASE TYPE] <Job> „ComplexType: SubscribeOrganizationJobType“ </Job> [1..1] <Company> „ComplexType: CompanyRequestType“ </Company> [1..1] </...>

ComplexType: SubscribeOrganizationResponseType

Abstract	no
Parent type	„ComplexType: ResponseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: ResponseType“] <ResponseContext> „ComplexType: ResponseContextType“ </ResponseContext> [1..1] [END BASE TYPE] <Addressees> „ComplexType: SubscribeOrganizationJobStateType“ </Addressees> [1..1] </...>

ComplexType: SynchronizeDeclareIncidentConsumerType

Abstract	no
Parent type	„ComplexType: AddresseeResponseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: AddresseeResponseType“] <AddresseeContext> „ComplexType: AddresseeResponseContextType“ </AddresseeCon\ text> [1..1] [END BASE TYPE] <Addressee> „ComplexType: AddresseeType“ </Addressee> [1..1] <Case> „ComplexType: DeclareIncidentConsumerCaseType“ </Case> [1..1] </...>

ComplexType: SynchronizeDeclareIncidentRequestType

Abstract	no
Parent type	„ComplexType: SynchronizeRequestType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: SynchronizeRequestType“] [BASE TYPE „ComplexType: RequestType“] <RequestContext> „ComplexType: RequestContextType“ </RequestContext> [1..1] [END BASE TYPE] <Sender> „ComplexType: CompanyUIDType“ </Sender> [1..1] [END BASE TYPE] <Addressee> „ComplexType: AddressingAddresseeType“ </Addressee> [1..1] <Case> „ComplexType: DeclareIncidentCaseType“ </Case> [1..1] </...>

ComplexType: SynchronizeDeclareIncidentResponseType

Abstract	no
Parent type	„ComplexType: ResponseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: ResponseType“] <ResponseContext> „ComplexType: ResponseContextType“ </ResponseContext> [1..1] [END BASE TYPE] [START CHOICE] <Error> „ComplexType: ErrorResponseType“ </Error> [1..1] <SynchronizeDeclareIncidentConsumer> „ComplexType: SynchronizeDeclareIncidentConsumerType“ </ SynchronizeDeclareIncidentConsumer> [1..1] [END CHOICE] </...>

© Swissdec 2026

ComplexType: SynchronizeRegisterOrganizationAuthenticationRequestType

Abstract	no
Parent type	„ComplexType: SynchronizeRequestType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: SynchronizeRequestType“] [BASE TYPE „ComplexType: RequestType“] <RequestContext> „ComplexType: RequestContextType“ </RequestContext> [1..1] [END BASE TYPE] <Sender> „ComplexType: CompanyUIDType“ </Sender> [1..1] [END BASE TYPE] <Addressee> „ComplexType: AddressingAddresseeType“ </Addressee> [1..1] <Case> „ComplexType: RegisterOrganizationAuthenticationSenderRequestCaseType“ </Case> [1..1] </...>

ComplexType: SynchronizeSubscribeOrganizationConsumerType

Abstract	no
Parent type	„ComplexType: AddresseeResponseType“
Technical description	Dieses Element synchronisiert die Anmeldung der Organisation.
XML Instance Representation	<...> [BASE TYPE „ComplexType: AddresseeResponseType“] <AddresseeContext> „ComplexType: AddresseeResponseContextType“ </AddresseeContext> [1..1] [END BASE TYPE] <Addressee> „ComplexType: AddresseeType“ </Addressee> [1..1] <Case> „ComplexType: SubscribeOrganizationConsumerCaseType“ </Case> [1..1] <Available> „ComplexType: AvailableType“ </Available> [0..1] </...>

ComplexType: SynchronizeSubscribeOrganizationRequestType

Abstract	no
Parent type	„ComplexType: SynchronizeRequestType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: SynchronizeRequestType“] [BASE TYPE „ComplexType: RequestType“] <RequestContext> „ComplexType: RequestContextType“ </RequestContext> [1..1] [END BASE TYPE] <Sender> „ComplexType: CompanyUIDType“ </Sender> [1..1] [END BASE TYPE] <Addressee> „ComplexType: AddressingAddresseeType“ </Addressee> [1..1] <Case> „ComplexType: SubscribeOrganizationCaseType“ </Case> [1..1] </...>

ComplexType: SynchronizeSubscribeOrganizationResponseType

Abstract	no
Parent type	„ComplexType: ResponseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: ResponseType“] <ResponseContext> „ComplexType: ResponseContextType“ </ResponseContext> [1..1] [END BASE TYPE] [START CHOICE] <Error> „ComplexType: ErrorResponseType“ </Error> [1..1] <SynchronizeSubscribeOrganizationConsumer> „ComplexType: SynchronizeSubscribeOrganizationConsumerType“ SynchronizeSubscribeOrganizationConsumer> [1..1] [END CHOICE] </...>

SimpleType: DeclareIncidentStateEnumType

Content type	Restriction
Base type	xs:string
Facets	• Enum: RegistrationReceived • Enum: CheckingDeclaration

© Swissdec 2026

	• Enum: VerifyingClaim • Enum: PerformingEntitlement • Enum: CommunicationStopped

SimpleType: SubscribeOrganizationStateType

Content type	Restriction
Base type	xs:string
Facets	• Enum: subscribed • Enum: closed

G.4 Schema Dokumentation IncidentDeclaration.xsd

Target Namespace	urn:ch:swissdec:kle:v2:20260306:incidentdeclaration
Declared Namespaces	• c : urn:ch:swissdec:common:v3:20260306 • ep : urn:ch:swissdec:basis:v1:20260306:components • id : urn:ch:swissdec:kle:v2:20260306:incidentdeclaration • xs : http://www.w3.org/2001/XMLSchema
Version	0.0

ComplexType: AbsenceType

Abstract	no
XML Instance Representation	<...> <Date>xs:date</Date> [1..1] <Hours> „SimpleType: HoursOrLessonsType“ </Hours> [1..1] </...>

ComplexType: AbsencesType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <Absence> „ComplexType: AbsenceType“ </Absence> [1..unbounded] </...>

ComplexType: AccidentAtLeisureType

Abstract	no
XML Instance Representation	<...> <LastWorkingDay>xs:date</LastWorkingDay> [1..1] <ActivityAtTimeOfAccident> „SimpleType: ActivityAtTimeOfAccidentType“ </ActivityAt\ TimeOfAccident> [1..1] </...>

ComplexType: AccidentAtWorkType

Abstract	no
XML Instance Representation	<...> <InvolvedObjects> „SimpleType: NotEmptyStringType“ </InvolvedObjects> [0..1] </...>

ComplexType: AccidentDescriptionType

Abstract	no

© Swissdec 2026

Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID>„SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] [START CHOICE] <Accident>„ComplexType: ClassOfInsuranceType“ </Accident> [1..1] <OccupationalDisease>„ComplexType: EmptyType“ </OccupationalDisease> [1..1] <CaseOfDeath>„ComplexType: CaseOfDeathType“ </CaseOfDeath> [1..1] <Relapse>„ComplexType: StoryRelapseType“ </Relapse> [1..1] [END CHOICE] <IncidentDate>„ComplexType: StoryIncidentDateTimeType“ </IncidentDate> [1..1] <IncidentScene>xs:string</IncidentScene> [1..1] <IncidentDescription>xs:string</IncidentDescription> [1..1] <PoliceReportExists>„SimpleType: YesNoUnknownType“ </PoliceReportExists> [1..1] <Injury>„ComplexType: InjuryType“ </Injury> [1..1] <ExistInvolvedPersons>„SimpleType: YesNoUnknownType“ </ExistInvolvedPer\ sons> [1..1] </...>

ComplexType: AccidentRegistrationType

Abstract	no
XML Instance Representation	<...> <CompanyCaseID>„SimpleType: IDType“ </CompanyCaseID> [1..1] [START CHOICE] <Incident>„ComplexType: IncidentAccidentRegistrationType“ </Incident> [1..1] <Relapse>„ComplexType: RelapseAccidentRegistrationType“ </Relapse> [1..1] [END CHOICE] <CaseOfIncapacitiesToWork>„ComplexType: EmptyType“ </CaseOfIncapacitiesTo\ Work> [0..1] [START CHOICE] <InsuranceCode>„SimpleType: AssuranceCategoryCodeType“ </ InsuranceCode> [1..unbounded] <WithoutContract>„ComplexType: EmptyType“ </WithoutContract> [1..1] [END CHOICE] <HRLanguageCode>„SimpleType: LanguageCodeType“ </HRLanguageCode> [0..1] </...>

ComplexType: AdditionsPersonType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID>„SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <Position>„SimpleType: PositionType“ </Position> [1..1] <Contract>„ComplexType: ContractType“ </Contract> [1..1] <ContractFixedUntil>xs:date</ContractFixedUntil> [0..1] <ContractDissolveFrom>xs:date</ContractDissolveFrom> [0..1] <JobTitle>„SimpleType: IDType“ </JobTitle> [1..1] <TaxAtSource>„ComplexType: TaxAtSourceType“ </TaxAtSource> [0..1] </...>

ComplexType: AddressingAddresseeType

Abstract	no
Parent type	„ComplexType: AddresseeType“
Short description	Informationen zum Adressaten
XML Instance Representation	<...> [BASE TYPE „ComplexType: AddresseeType“] <AddresseeIdentification>„SimpleType: IDType“ </AddresseeIdentification> [1..1] [END BASE TYPE] <Domain>„SimpleType: AddressingDomainType“ </Domain> [1..1] </...>

ComplexType: AddressingIdentificationType

Abstract	no
XML Instance Representation	<...> [START CHOICE]

© Swissdec 2026

	<UVG-LAA> „ComplexType: UVG-LAA-IdentificationType“ </UVG-LAA> [1..1] <UVGZ-LAAC> „ComplexType: IdentificationBaseType“ </UVGZ-LAAC> [1..1] <KTG-AMC> „ComplexType: IdentificationBaseType“ </KTG-AMC> [1..1] [END CHOICE] </...>

ComplexType: AddressingType

Abstract	no
XML Instance Representation	<...> <Institution> „ComplexType: AddressingIdentificationType“ </Institution> [1..1] </...>

ComplexType: AgreementType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] [START CHOICE] <OtherInsurancePredefined> „SimpleType: OtherInsurancePredefinedType“ </OtherIn\surancePredefined> [1..1] <OtherInsurance> „SimpleType: NotEmptyStringType“ </OtherInsurance> [1..1] <NoOtherInsurance> „ComplexType: EmptyType“ </NoOtherInsurance> [1..1] [END CHOICE] </...>

ComplexType: AllowancesPastType

Abstract	no
XML Instance Representation	<...> <Salary> „ComplexType: InsuredSalaryPastType“ </Salary> [1..12] <AllowancesAnnualTotal> „SimpleType: SalaryAmountType“ </AllowancesAnnualTo\tal> [1..1] </...>

ComplexType: AnnualSalaryType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] [START CHOICE] <IncidentMonth>xs:gYearMonth</IncidentMonth> [1..1] <Present> „ComplexType: KindOfSalariesTimedType“ </Present> [0..1] <Past> „ComplexType: MonthSalaryType“ </Past> [0..1] <ManualAdditions> „ComplexType: KindOfSalaryType“ </ManualAdditions> [0..unbounded] <AnnualSalaryTotal> „SimpleType: SalaryAmountType“ </AnnualSalaryTotal> [1..1] <Statistic> „ComplexType: SalaryBasesType“ </Statistic> [1..1] <NoIncapacityToWork> „ComplexType: EmptyType“ </NoIncapacityToWork> [1..1] [END CHOICE] </...>

ComplexType: AnnualWorkingTimeType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] [START CHOICE] <MonthlyWorkingTimes> „ComplexType: MonthlyWorkingTimesType“ </MonthlyWorkingTi\mes> [1..1]

© Swissdec 2026

	<NoData> „ComplexType: EmptyType“ </NoData> [1..1] [END CHOICE] </...>

ComplexType: AttachmentType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <Filename> „SimpleType: NotEmptyStringType“ </Filename> [1..1] <ContentType> „SimpleType: ContentTypeType“ </ContentType> [1..1] <Data>xs:base64Binary</Data> [1..1] </...>

ComplexType: AwaitPartControlType

Abstract	no
XML Instance Representation	<...> <ExpirationDate>xs:dateTime</ExpirationDate> [0..1] </...>

ComplexType: AwaitPartType

Abstract	no
XML Instance Representation	<...> <IncapacitiesToWork> „ComplexType: AwaitPartControlType“ </IncapacitiesTo\ Work> [0..1] <TakeWorkUpFully> „ComplexType: AwaitPartControlType“ </TakeWorkUpFully> [0..1] <Treatments> „ComplexType: AwaitPartControlType“ </Treatments> [0..1] <Agreement> „ComplexType: AwaitPartControlType“ </Agreement> [0..1] <AnnualSalary> „ComplexType: AwaitPartControlType“ </AnnualSalary> [0..1] <AnnualWorkingTime> „ComplexType: AwaitPartControlType“ </AnnualWorking\ Time> [0..1] <Payment> „ComplexType: AwaitPartControlType“ </Payment> [0..1] <AdditionsPerson> „ComplexType: AwaitPartControlType“ </AdditionsPerson> [0..1] <Absence> „ComplexType: AwaitPartControlType“ </Absence> [0..1] <SpecialCode> „ComplexType: AwaitPartControlType“ </SpecialCode> [0..1] <AccidentDescription> „ComplexType: AwaitPartControlType“ </AccidentDescripti\ on> [0..1] </...>

ComplexType: BenefitPeriodType

Abstract	no
Parent type	„ComplexType: TimePeriodType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: TimePeriodType“] <from>xs:date</from> [1..1] <until>xs:date</until> [1..1] [END BASE TYPE] <IncapacityToWork> „SimpleType: PercentType“ </IncapacityToWork> [0..1] </...>

ComplexType: CaseOfDeathType

Abstract	no
XML Instance Representation	<...> <DateOfDeath>xs:date</DateOfDeath> [1..1] <ClassOfInsurance> „ComplexType: ClassOfInsuranceType“ </ClassOfInsurance> [1..1] </...>

ComplexType: ChangeIdentitiesType

Abstract	no
XML Instance Representation	<...>

© Swissdec 2026

	<ValidAsOf>xs:dateTime</ValidAsOf> [1..1] <Company> „ComplexType: CompanyIdentitiesType“ </Company> [0..1] <Institution> „ComplexType: InstitutionIdentitiesType“ </Institution> [0..1] </...>

ComplexType: ClassOfInsuranceType

Abstract	no
XML Instance Representation	<...> [START CHOICE] <AccidentAtWork> „ComplexType: AccidentAtWorkType“ </AccidentAtWork> [1..1] <AccidentAtLeisure> „ComplexType: AccidentAtLeisureType“ </AccidentAtLeisu\ re> [1..1] [END CHOICE] </...>

ComplexType: CompanyIdentitiesType

Abstract	no
XML Instance Representation	<...> <Old> „ComplexType: CompanyUIDType“ </Old> [1..1] <New> „ComplexType: CompanyUIDType“ </New> [1..1] </...>

ComplexType: CompanyUIDType

Abstract	no
XML Instance Representation	<...> <UID-BFS> „ComplexType: UID-BFS-UnknownType“ </UID-BFS> [1..1] <DelegateUID-BFS> „SimpleType: UID-BFSType“ </DelegateUID-BFS> [0..1] </...>

ComplexType: ContractIndemnificationType

Abstract	no
XML Instance Representation	<...> <ContractIdentity> „SimpleType: NotEmptyStringType“ </ContractIdentity> [1..1] <InsuranceCode> „SimpleType: AssuranceCategoryCodeType“ </ InsuranceCode> [0..unbounded] </...>

ComplexType: ContractType

Abstract	no
Translation	Vertrag
Short description	Art des Arbeitsvertrags
XML Instance Representation	<...> <KindOf> „SimpleType: EmploymentContractType“ </KindOf> [1..1] <ValidAsOf>xs:date</ValidAsOf> [1..1] </...>

ComplexType: CoveredDailyAllowanceType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <Allowance> „SimpleType: SalaryAmountType“ </Allowance> [1..1] <ValidAsOf>xs:date</ValidAsOf> [0..1] <ValidUntil>xs:date</ValidUntil> [0..1] <Provisional> „ComplexType: EmptyType“ </Provisional> [0..1] <Description> „SimpleType: NotEmptyStringType“ </Description> [0..1] <Reduction>id:anonymous</Reduction> [0..1] <HospitalCostDeduction> „SimpleType: HospitalCostDeductionEnumType“ </HospitalCost\ Deduction> [0..1]

© Swissdec 2026

	<NetAllowance> „SimpleType: SalaryAmountType“ </NetAllowance> [0..1] </...>

ComplexType: CustomerIdentificationType

Abstract	no
XML Instance Representation	<...> <UVG-LAA> „ComplexType: InstitutionType“ </UVG-LAA> [0..1] <UVGZ-LAAC> „ComplexType: IdentificationBaseType“ </UVGZ-LAAC> [0..unbounded] <KTG-AMC> „ComplexType: IdentificationBaseType“ </KTG-AMC> [0..1] </...>

ComplexType: FavourCompanyType

Abstract	no
XML Instance Representation	<...> <Payment> „SimpleType: SalaryAmountAbsoluteType“ </Payment> [1..1] <PaymentThird> „SimpleType: SalaryAmountAbsoluteType“ </PaymentThird> [0..1] </...>

ComplexType: FavourInstitutionType

Abstract	no
XML Instance Representation	<...> <Repayment> „SimpleType: SalaryAmountAbsoluteType“ </Repayment> [1..1] <RepaymentThird> „SimpleType: SalaryAmountAbsoluteType“ </RepaymentThird> [0..1] </...>

ComplexType: HourlyPayType

Abstract	no
XML Instance Representation	<...> <HourlySalary> „SimpleType: SalaryAmountType“ </HourlySalary> [1..1] <Earnings13th> „SimpleType: PercentType“ </Earnings13th> [0..1] <VacationAndPublicHolidayCompensation> „SimpleType: PercentType“ </VacationAndPubli\ cHolidayCompensation> [0..1] <StatisticCode> „SimpleType: SalaryBaseCodeType“ </StatisticCode> [1..1] <AnnualHourSalary> „SimpleType: SalaryAmountType“ </AnnualHourSalary> [1..1] </...>

ComplexType: HourlyWagePastType

Abstract	no
XML Instance Representation	<...> <Salary> „ComplexType: InsuredSalaryHourlyPastType“ </Salary> [1..12] <HourlyWageAnnualTotal> „SimpleType: SalaryAmountType“ </HourlyWageAnnualTo\ tal> [1..1] </...>

ComplexType: IllnessRegistrationType

Abstract	no
XML Instance Representation	<...> <CompanyCaseID> „SimpleType: IDType“ </CompanyCaseID> [1..1] <IncapacitiesToWork>id:anonymous</IncapacitiesToWork> [0..1] <Incident> „ComplexType: IncidentAccidentRegistrationBaseType“ </Incident> [0..1] <DateOfBirth>xs:date</DateOfBirth> [0..1] <InsuranceCaseID> „SimpleType: IDType“ </InsuranceCaseID> [0..1] <InsuranceCode> „SimpleType: AssuranceCategoryCodeType“ </ InsuranceCode> [1..unbounded] <HRLanguageCode> „SimpleType: LanguageCodeType“ </HRLanguageCode> [0..1] </...>

ComplexType: IncapacitiesToWorkType

Abstract	no
Parent type	„ComplexType: StoryBaseType“

© Swissdec 2026

XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] [START CHOICE] <IncapacityToWork> „ComplexType: IncapacityToWorkType“ </IncapacityToWork> [1..unbounded] <NoIncapacityToWork> „ComplexType: EmptyType“ </NoIncapacityToWork> [1..1] [END CHOICE] </...>

ComplexType: IncapacityToWorkType

Abstract	no
XML Instance Representation	<...> <Percentage> „SimpleType: Max100PercentType“ </Percentage> [1..1] <EffectiveFrom>xs:date</EffectiveFrom> [1..1] <Valid> „ComplexType: ValidToType“ </Valid> [1..1] <Provisional> „ComplexType: EmptyType“ </Provisional> [0..1] </...>

ComplexType: IncidentAccidentRegistrationBaseType

Abstract	no
XML Instance Representation	<...> <IncidentDate> „ComplexType: RegistrationIncidentDateType“ </IncidentDate> [1..1] </...>

ComplexType: IncidentAccidentRegistrationType

Abstract	no
Parent type	„ComplexType: IncidentAccidentRegistrationBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: IncidentAccidentRegistrationBaseType“] <IncidentDate> „ComplexType: RegistrationIncidentDateType“ </IncidentDate> [1..1] [END BASE TYPE] <InsuranceCaseID> „SimpleType: IDType“ </InsuranceCaseID> [0..1] </...>

ComplexType: IncidentCaseIDType

Abstract	no
Translation	IncidentCaseID
XML Instance Representation	<...> </...>

ComplexType: IncidentContextRequestType

Abstract	no
XML Instance Representation	<...> <ReceivedState> „ComplexType: StateType“ </ReceivedState> [0..1] <Info> „ComplexType: StoryNotificationRequestType“ </Info> [0..unbounded] <Warning> „ComplexType: StoryNotificationRequestType“ </Warning> [0..unbounded] <Error> „ComplexType: StoryNotificationRequestType“ </Error> [0..unbounded] </...>

ComplexType: IncidentDeclarationType

Abstract	no
Technical description	Fachliche Datenstruktur für die Ereignismeldung.
XML Instance Representation	<...> <CompanyDescription> „ComplexType: CompanyDescriptionType“ </CompanyDescription> [1..1] <Person> „ComplexType: PersonIncidentType“ </Person> [1..1] <Institutions> „ComplexType: CustomerIdentificationType“ </Institutions> [1..1] <Contact> „ComplexType: ContactType“ </Contact> [0..1] </...>

© Swissdec 2026

ComplexType: IncidentForCompanyClosedType

Abstract	no
XML Instance Representation	<...> [START CHOICE] <Normal> „ComplexType: EmptyType“ </Normal> [1..1] <Cancel>xs:string</Cancel> [1..1] [END CHOICE] </...>

ComplexType: IncidentStoriesRequestType

Abstract	no
XML Instance Representation	<...> <IncidentContext> „ComplexType: IncidentContextRequestType“ </IncidentCon\ text> [1..1] <IncapacitiesToWork> „ComplexType: IncapacitiesToWorkType“ </IncapacitiesTo\ Work> [0..1] <TakeWorkUpFully> „ComplexType: TakeWorkUpFullyType“ </TakeWorkUpFully> [0..1] <Treatments> „ComplexType: TreatmentsType“ </Treatments> [0..1] <Agreement> „ComplexType: AgreementType“ </Agreement> [0..1] <AnnualSalary> „ComplexType: AnnualSalaryType“ </AnnualSalary> [0..1] <Payment> „ComplexType: PaymentType“ </Payment> [0..1] <AdditionsPerson> „ComplexType: AdditionsPersonType“ </AdditionsPerson> [0..1] <Absences> „ComplexType: AbsencesType“ </Absences> [0..1] <SpecialCode> „ComplexType: SpecialCodeType“ </SpecialCode> [0..1] <AccidentDescription> „ComplexType: AccidentDescriptionType“ </AccidentDescripti\ on> [0..1] <SalaryRaise> „ComplexType: SalaryRaiseType“ </SalaryRaise> [0..1] <PersonMutated> „ComplexType: PersonMutatedType“ </PersonMutated> [0..1] <SpecialCoverage> „ComplexType: SpecialCoverageType“ </SpecialCoverage> [0..1] <OtherEmployers> „ComplexType: BooleanStoryType“ </OtherEmployers> [0..1] <ShortTimeWork> „ComplexType: BooleanStoryType“ </ShortTimeWork> [0..1] <DialogMessage> „ComplexType: DialogMessageType“ </DialogMessage> [0..unbounded] <Attachment> „ComplexType: AttachmentType“ </Attachment> [0..unbounded] <ProcessCtrl> „ComplexType: ProcessCtrlRequestType“ </ProcessCtrl> [0..1] </...>

ComplexType: IndemnificationCorrectionType

Abstract	no
Parent type	„ComplexType: IndemnificationType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: IndemnificationType“] <IndemnificationID> „SimpleType: IDType“ </IndemnificationID> [1..1] <ProductCategory> „SimpleType: NotEmptyStringType“ </ProductCategory> [0..1] <WaitingDays> „SimpleType: DaysType“ </WaitingDays> [0..1] <CoveringDays> „SimpleType: DaysType“ </CoveringDays> [0..1] <IncapacityToWork> „SimpleType: PercentType“ </IncapacityToWork> [0..1] <QuantityOfDays>xs:short</QuantityOfDays> [0..1] <Period> „ComplexType: TimePeriodType“ </Period> [0..1] <EffectiveDailyAllowance> „SimpleType: SalaryAmountType“ </EffectiveDailyAllowan\ ce> [0..1] <Reduction> „ComplexType: ReductionType“ </Reduction> [0..unbounded] <Balance> „ComplexType: BalanceType“ </Balance> [1..1] <Contract> „ComplexType: ContractIndemnificationType“ </Contract> [0..1] <Comment> „ComplexType: NotificationsType“ </Comment> [0..1] [END BASE TYPE] <LinkToOldIndemnification> „ComplexType: LinkToOldIndemnificationType“ </LinkToOl\ dIndemnification> [0..1] </...>

ComplexType: IndemnificationType

Abstract	no
XML Instance Representation	<...> <IndemnificationID> „SimpleType: IDType“ </IndemnificationID> [1..1] <ProductCategory> „SimpleType: NotEmptyStringType“ </ProductCategory> [0..1] <WaitingDays> „SimpleType: DaysType“ </WaitingDays> [0..1] <CoveringDays> „SimpleType: DaysType“ </CoveringDays> [0..1] <IncapacityToWork> „SimpleType: PercentType“ </IncapacityToWork> [0..1] <QuantityOfDays>xs:short</QuantityOfDays> [0..1] <Period> „ComplexType: TimePeriodType“ </Period> [0..1] <EffectiveDailyAllowance> „SimpleType: SalaryAmountType“ </EffectiveDailyAllowan\ ce> [0..1] <Reduction> „ComplexType: ReductionType“ </Reduction> [0..unbounded] <Balance> „ComplexType: BalanceType“ </Balance> [1..1] <Contract> „ComplexType: ContractIndemnificationType“ </Contract> [0..1]

© Swissdec 2026

	<Comment> „ComplexType: NotificationsType“ </Comment> [0..1] </...>

ComplexType: IndemnificationsType

Abstract	no
XML Instance Representation	<...> <Normal> „ComplexType: IndemnificationType“ </Normal> [0..unbounded] <Correction> „ComplexType: IndemnificationCorrectionType“ </ Correction> [0..unbounded] </...>

ComplexType: InjuredPartOfBodyType

Abstract	no
XML Instance Representation	<...> <PartOfBody> „ComplexType: PartOfBodyType“ </PartOfBody> [1..1] [START CHOICE] <TypeOfInjury> „SimpleType: TypeOfInjuryType“ </TypeOfInjury> [1..1] <OtherTypeOfInjury>xs:string</OtherTypeOfInjury> [1..1] [END CHOICE] </...>

ComplexType: InjuryType

Abstract	no
XML Instance Representation	<...> <InjuredPartOfBody> „ComplexType: InjuredPartOfBodyType“ </InjuredPartOfBo\ dy> [1..3] </...>

ComplexType: InstitutionChangeType

Abstract	no
XML Instance Representation	<...> <CustomerIdentity> „SimpleType: NotEmptyStringType“ </CustomerIdentity> [1..1] <ContractIdentity> „SimpleType: NotEmptyStringType“ </ContractIdentity> [0..1] </...>

ComplexType: InstitutionIdentitiesType

Abstract	no
XML Instance Representation	<...> <Old> „ComplexType: InstitutionChangeType“ </Old> [1..1] <New> „ComplexType: InstitutionChangeType“ </New> [1..1] </...>

ComplexType: InstitutionIllnessType

Abstract	no
XML Instance Representation	<...> <InsuranceCompanyName>xs:string</InsuranceCompanyName> [1..1] <CustomerIdentity> „SimpleType: NotEmptyStringType“ </CustomerIdentity> [1..1] <ContractIdentity> „SimpleType: NotEmptyStringType“ </ContractIdentity> [1..1] </...>

ComplexType: InstitutionType

Abstract	no
XML Instance Representation	<...> <InsuranceCompanyName>xs:string</InsuranceCompanyName> [1..1] <CustomerIdentity> „SimpleType: NotEmptyStringType“ </CustomerIdentity> [1..1] [START CHOICE] <ContractIdentity> „SimpleType: NotEmptyStringType“ </ContractIdentity> [1..1] <NoContract> „ComplexType: EmptyType“ </NoContract> [1..1] [END CHOICE] </...>

© Swissdec 2026

ComplexType: InsuredSalaryHourlyPastType

Abstract	no
XML Instance Representation	<...> <Month>xs:gYearMonth</Month> [1..1] <KindOfSalaries> „ComplexType: KindOfSalariesPastType“ </KindOfSalaries> [1..1] <Total> „SimpleType: SalaryAmountType“ </Total> [1..1] <HourlySalary> „SimpleType: SalaryAmountType“ </HourlySalary> [0..1] <HoursPerMonth> „SimpleType: HoursAndIndustryMinutesType“ </HoursPerMonth> [0..1] </...>

ComplexType: InsuredSalaryLessonPastType

Abstract	no
XML Instance Representation	<...> <Month>xs:gYearMonth</Month> [1..1] <KindOfSalaries> „ComplexType: KindOfSalariesPastType“ </KindOfSalaries> [1..1] <Total> „SimpleType: SalaryAmountType“ </Total> [1..1] <LessonSalary> „SimpleType: SalaryAmountType“ </LessonSalary> [0..1] <LessonsPerMonth> „SimpleType: HoursAndIndustryMinutesType“ </LessonsPer\ Month> [0..1] </...>

ComplexType: InsuredSalaryPastType

Abstract	no
XML Instance Representation	<...> <Month>xs:gYearMonth</Month> [1..1] <KindOfSalaries> „ComplexType: KindOfSalariesPastType“ </KindOfSalaries> [1..1] <Total> „SimpleType: SalaryAmountType“ </Total> [1..1] </...>

ComplexType: KindOfSalariesPastType

Abstract	no
XML Instance Representation	<...> <Detail> „ComplexType: KindOfSalaryType“ </Detail> [1..unbounded] </...>

ComplexType: KindOfSalariesTimedPastType

Abstract	no
XML Instance Representation	<...> <Detail> „ComplexType: KindOfSalaryTimedType“ </Detail> [1..unbounded] </...>

ComplexType: KindOfSalariesTimedType

Abstract	no
XML Instance Representation	<...> <HourlyPay> „ComplexType: HourlyPayType“ </HourlyPay> [0..1] <LessonPay> „ComplexType: LessonPayType“ </LessonPay> [0..1] <Detail> „ComplexType: KindOfSalaryType“ </Detail> [0..unbounded] </...>

ComplexType: KindOfSalaryTimedType

Abstract	no
Parent type	„ComplexType: KindOfSalaryType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: KindOfSalaryType“] <KindOfSalary>xs:string</KindOfSalary> [1..1] <Amount> „SimpleType: SalaryAmountType“ </Amount> [1..1] <StatisticCode> „SimpleType: SalaryBaseCodeType“ </StatisticCode> [1..1] [END BASE TYPE] <BenefitPeriod> „ComplexType: BenefitPeriodType“ </BenefitPeriod> [0..unbounded] </...>

© Swissdec 2026

ComplexType: KindOfSalaryType

Abstract	no
XML Instance Representation	<...> <KindOfSalary>xs:string</KindOfSalary> [1..1] <Amount> „SimpleType: SalaryAmountType“ </Amount> [1..1] <StatisticCode> „SimpleType: SalaryBaseCodeType“ </StatisticCode> [1..1] </...>

ComplexType: LessonPayType

Abstract	no
XML Instance Representation	<...> <LessonSalary> „SimpleType: SalaryAmountType“ </LessonSalary> [1..1] <Earnings13th> „SimpleType: PercentType“ </Earnings13th> [0..1] <VacationAndPublicHolidayCompensation> „SimpleType: PercentType“ </VacationAndPubli\ cHolidayCompensation> [0..1] <StatisticCode> „SimpleType: SalaryBaseCodeType“ </StatisticCode> [1..1] <AnnualLessonSalary> „SimpleType: SalaryAmountType“ </AnnualLessonSalary> [1..1] </...>

ComplexType: LessonWagePastType

Abstract	no
XML Instance Representation	<...> <Salary> „ComplexType: InsuredSalaryLessonPastType“ </Salary> [1..12] <LessonWageAnnualTotal> „SimpleType: SalaryAmountType“ </LessonWageAnnualTo\ tal> [1..1] </...>

ComplexType: LinkToOldIndemnificationType

Abstract	no
XML Instance Representation	<...> <SettlementID> „SimpleType: IDType“ </SettlementID> [1..1] <IndemnificationID> „SimpleType: IDType“ </IndemnificationID> [1..1] </...>

ComplexType: MonthSalaryType

Abstract	no
XML Instance Representation	<...> <HourlyWage> „ComplexType: HourlyWagePastType“ </HourlyWage> [0..1] <LessonWage> „ComplexType: LessonWagePastType“ </LessonWage> [0..1] <Allowances> „ComplexType: AllowancesPastType“ </Allowances> [0..1] <WageReplacementBenefits> „ComplexType: WageReplacementBenefitsPastType“ </WageRe\ placementBenefits> [0..1] <PastAnnualTotal> „SimpleType: SalaryAmountType“ </PastAnnualTotal> [1..1] </...>

ComplexType: MonthlyWorkingTimeType

Abstract	no
XML Instance Representation	<...> <Month>xs:gYearMonth</Month> [1..1] <HoursPerMonth> „SimpleType: HoursAndIndustryMinutesType“ </HoursPerMonth> [0..1] <LessonsPerMonth> „SimpleType: HoursAndIndustryMinutesType“ </LessonsPer\ Month> [0..1] </...>

ComplexType: MonthlyWorkingTimesType

Abstract	no
XML Instance Representation	<...> <MonthlyWorkingTime> „ComplexType: MonthlyWorkingTimeType“ </MonthlyWorking\ Time> [1..12] </...>

© Swissdec 2026

ComplexType: OldNewIDType

Abstract	no
XML Instance Representation	<...> <Old> „SimpleType: IDType“ </Old> [1..1] <New> „SimpleType: IDType“ </New> [1..1] </...>

ComplexType: PartOfBodyType

Abstract	no
XML Instance Representation	<...> [START CHOICE] <UnassignedPart> „SimpleType: BodyPartType“ </UnassignedPart> [1..1] <Part> „SimpleType: BodyOrientedPartType“ </Part> [1..1] <Position> „SimpleType: PositionPartType“ </Position> [1..1] [END CHOICE] </...>

ComplexType: PaymentAddressBaseType

Abstract	no
XML Instance Representation	<...> <AccountOwner>xs:string</AccountOwner> [1..1] <AccountOwnerSpecialAddress> „ComplexType: NameAndAddressType“ </AccountOwnerSpe\ cialAddress> [0..1] <IBAN> „SimpleType: IBANType“ </IBAN> [1..1] <FinancialInstitution> „ComplexType: NameAndAddressType“ </FinancialInstituti\ on> [0..1] </...>

ComplexType: PaymentAddressType

Abstract	no
Parent type	„ComplexType: PaymentAddressBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: PaymentAddressBaseType“] <AccountOwner>xs:string</AccountOwner> [1..1] <AccountOwnerSpecialAddress> „ComplexType: NameAndAddressType“ </AccountOwnerSpe\ cialAddress> [0..1] <IBAN> „SimpleType: IBANType“ </IBAN> [1..1] <FinancialInstitution> „ComplexType: NameAndAddressType“ </FinancialInstituti\ on> [0..1] [END BASE TYPE] <ValidAsOf>xs:date</ValidAsOf> [1..1] </...>

ComplexType: PaymentType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] [START CHOICE] <Company> „ComplexType: PaymentAddressBaseType“ </Company> [1..1] <Person> „ComplexType: PaymentAddressType“ </Person> [1..1] [END CHOICE] </...>

ComplexType: PersonIncidentType

Abstract	no
Parent type	„ComplexType: PersonBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: PersonBaseType“]

© Swissdec 2026

	<Particulars> „ComplexType: ParticularsType“ </Particulars> [1..1] <Work>„ComplexType: WorkType“ </Work> [1..unbounded] [END BASE TYPE] <UVG-LAA-Registration>„ComplexType: AccidentRegistrationType“ </UVG-LAA-Registrati\ on> [0..1] <UVGZ-LAAC-Registration>„ComplexType: AccidentRegistrationType“ </UVGZ-LAAC- Registration> [0..unbounded] <KTG-AMC-Registration>„ComplexType: IllnessRegistrationType“ </KTG-AMC-Registrati\ on> [0..1] </...>

ComplexType: PersonMutatedType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID>„SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <ValidAsOf>xs:date</ValidAsOf> [1..1] <Lastname>xs:string</Lastname> [0..1] <Firstname>xs:string</Firstname> [0..1] <WithdrawalDate>xs:date</WithdrawalDate> [0..1] <Address>„ComplexType: AddressType“ </Address> [0..1] </...>

ComplexType: ProcessCtrlRequestType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID>„SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <IncidentForCompanyClosed>„ComplexType: IncidentForCompanyClosedType“ </Incident\ ForCompanyClosed> [0..1] <ChangeIdentities>„ComplexType: ChangeIdentitiesType“ </ChangeIdentities> [0..1] <Comment>„ComplexType: NotificationsType“ </Comment> [0..1] </...>

ComplexType: ProcessCtrlResponseType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID>„SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <ChangeInsuranceCaseID>„ComplexType: OldNewIDType“ </ChangeInsuranceCa\ seID> [0..1] <InstitutionAddress>„ComplexType: NameAndAddressType“ </InstitutionAd\ dress> [0..1] <Contact>„ComplexType: ContactType“ </Contact> [0..1] <AwaitStory>„ComplexType: AwaitPartType“ </AwaitStory> [0..1] <Comment>„ComplexType: NotificationsType“ </Comment> [0..1] </...>

ComplexType: ReceivedStoriesType

Abstract	no
XML Instance Representation	<...> <StoryID>„SimpleType: IDType“ </StoryID> [0..unbounded] </...>

ComplexType: ReductionType

Abstract	no
XML Instance Representation	<...> <ReasonPartialRelease>xs:string</ReasonPartialRelease> [0..1]

© Swissdec 2026

	<ReasonReduction>xs:string</ReasonReduction> [0..1] <ReasonOnCompensation>xs:string</ReasonOnCompensation> [0..1] <ReasonHospitalAliment>xs:string</ReasonHospitalAliment> [0..1] <Percentage> „SimpleType: PercentType“ </Percentage> [0..1] <CoveringDays> „SimpleType: DaysType“ </CoveringDays> [0..1] <QuantityOfDays>xs:short</QuantityOfDays> [0..1] <Period> „ComplexType: TimePeriodType“ </Period> [0..1] <ReductionAllowance> „SimpleType: SalaryAmountAbsoluteType“ </ReductionAllowan\ ce> [0..1] <ReductionAmountFavourInstitution> „SimpleType: SalaryAmountAbsoluteType“ </Reduc\ tionAmountFavourInstitution> [0..1] <AmountBeforeReductionFavourCompany> „SimpleType: SalaryAmountAbsoluteType“ </ AmountBeforeReductionFavourCompany> [0..1] </...>

ComplexType: RegistrationIncidentDateType

Abstract	no
XML Instance Representation	<...> [START CHOICE] <Exact>xs:date</Exact> [1..1] <Approximate>xs:date</Approximate> [1..1] <Clarification> „ComplexType: EmptyType“ </Clarification> [1..1] [END CHOICE] </...>

ComplexType: RelapseAccidentRegistrationType

Abstract	no
XML Instance Representation	<...> <RelapseDate>xs:date</RelapseDate> [1..1] <IncidentDate> „ComplexType: RegistrationIncidentDateType“ </IncidentDate> [1..1] <InsuranceCaseID> „SimpleType: IDType“ </InsuranceCaseID> [1..1] </...>

ComplexType: RepaymentType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <RepaymentID> „SimpleType: IDType“ </RepaymentID> [1..1] <LinkToSettlementID> „SimpleType: IDType“ </LinkToSettlementID> [1..1] <ExpiryDate>xs:dateTime</ExpiryDate> [0..1] <Transfer> „ComplexType: TransferType“ </Transfer> [1..1] <Contact> „ComplexType: ContactType“ </Contact> [0..1] </...>

ComplexType: SalaryBasesType

Abstract	no
XML Instance Representation	<...> <BasicWages> „SimpleType: SalaryAmountType“ </BasicWages> [0..1] <FamilyIncomeSupplement> „SimpleType: SalaryAmountType“ </FamilyIncomeSupple\ ment> [0..1] <VacationAndPublicHolidayCompensation> „SimpleType: SalaryAmountType“ </VacationAnd\ PublicHolidayCompensation> [0..1] <OtherSupplements> „SimpleType: SalaryAmountType“ </OtherSupplements> [0..1] <Gratuity> „SimpleType: SalaryAmountType“ </Gratuity> [0..1] </...>

ComplexType: SalaryRaiseType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE]

© Swissdec 2026

	<ValidAsOf>xs:date</ValidAsOf> [1..1] <Present> „ComplexType: KindOfSalariesTimedType“ </Present> [1..1] <ManualAdditions> „ComplexType: KindOfSalaryType“ </ManualAdditions> [0..unbounded] </...>

ComplexType: SettlementType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <SettlementID> „SimpleType: IDType“ </SettlementID> [1..1] <SettlementDate>xs:dateTime</SettlementDate> [1..1] <TotalBalance> „ComplexType: TotalBalanceType“ </TotalBalance> [1..1] <Coverage> „SimpleType: NotEmptyStringType“ </Coverage> [0..1] <AnnualEarnings> „SimpleType: SalaryAmountType“ </AnnualEarnings> [0..1] <CoveredDailyAllowance> „SimpleType: SalaryAmountType“ </CoveredDailyAllowance> [0..1] <Contact> „ComplexType: ContactType“ </Contact> [0..1] <Indemnifications> „ComplexType: IndemnificationsType“ </Indemnifications> [0..1] <Comment> „ComplexType: NotificationsType“ </Comment> [0..1] </...>

ComplexType: SpecialCodeType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <Code> „SimpleType: NotEmptyStringType“ </Code> [1..1] </...>

ComplexType: SpecialCoverageType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
Translation	Spez.Übernahme
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] [START CHOICE] <FamilyMember> „ComplexType: EmptyType“ </FamilyMember> [1..1] <OptionalEnterpriserInsurance> „ComplexType: EmptyType“ </OptionalEnterpriserInsurance> [1..1] <NegotiatedInsurance> „ComplexType: EmptyType“ </NegotiatedInsurance> [1..1] [END CHOICE] </...>

ComplexType: StateType

Abstract	no
XML Instance Representation	<...> <Process> „SimpleType: MainStateType“ </Process> [1..1] <Coverage> „SimpleType: CoverageType“ </Coverage> [1..1] <SmallRemark> „SimpleType: NotEmptyStringType“ </SmallRemark> [0..1] <CrossChannelLinkIDWaitForUse> „SimpleType: IDType“ </CrossChannelLinkIDWaitForUse> [0..unbounded] </...>

ComplexType: StoriesWithoutDeclarationType

Abstract	no

© Swissdec 2026

XML Instance Representation	<...> <DialogMessage> „ComplexType: DialogMessageType“ </DialogMessage> [0..unbounded] </...>

ComplexType: StoryIncidentDateTimeType

Abstract	no

XML Instance Representation	<...>
	  [START CHOICE]
	  <ExactDateTime>xs:dateTime</ExactDateTime> [1..1]
	  <Approximate>xs:date</Approximate> [1..1]
	  [END CHOICE]
	</...>

ComplexType: StoryNotificationRequestType

Abstract	no
XML Instance Representation	<...> <ResponseStoryID> „SimpleType: IDType“ </ResponseStoryID> [0..1] </...>

ComplexType: StoryRelapseType

Abstract	no
XML Instance Representation	<...> <InsuranceCaseID> „SimpleType: IDType“ </InsuranceCaseID> [1..1] <RelapseDate>xs:date</RelapseDate> [1..1] </...>

ComplexType: TakeWorkUpFullyType

Abstract	no
Parent type	„ComplexType: StoryBaseType“

XML Instance Representation	<...>
	  [BASE TYPE „ComplexType: StoryBaseType“]
	  <Creation>xs:dateTime</Creation> [1..1]
	  <StoryID> „SimpleType: IDType“ </StoryID> [1..1]
	  [END BASE TYPE]
	  [START CHOICE]
	  <TakeWorkUpFullyDate>xs:date</TakeWorkUpFullyDate> [1..1]
	  <ExpectedLastDayOfIncapacityToWork>xs:date</ExpectedLastDayOfIncapacityTo\
	  Work> [1..1]
	  <ExpectedIncapacityToWork> „SimpleType: ExpectedIncapacityToWorkType“ </ExpectedIn\
	  capacityToWork> [1..1]
	  <NoIncapacityToWork> „ComplexType: EmptyType“ </NoIncapacityToWork> [1..1]
	  [END CHOICE]
	</...>

ComplexType: TaxAtSourceType

Abstract	no

XML Instance Representation	<...>
	  <TaxAtSourceCanton> „SimpleType: CantonAddressType“ </TaxAtSourceCanton> [1..1]
	  <TaxAtSourceMunicipalityID> „SimpleType: MunicipalityIDType“ </TaxAtSourceMunicipa\
	  lityID> [1..1]
	  <TaxAtSourceCategory> „ComplexType: TaxAtSourceCategoryType“ </TaxAtSourceCatego\
	  ry> [1..1]
	</...>

ComplexType: TotalBalanceType

Abstract	no

XML Instance Representation	<...>
	  [START CHOICE]
	  <FavourCompany> „ComplexType: FavourCompanyType“ </FavourCompany> [1..1]
	  <FavourInstitution> „ComplexType: FavourInstitutionType“ </FavourInstituti\
	  on> [1..1]
	  <FavourPerson> „SimpleType: SalaryAmountAbsoluteType“ </FavourPerson> [1..1]
	  [END CHOICE]

© Swissdec 2026

	</...>

ComplexType: TransferType

Abstract	no
Parent type	„ComplexType: PaymentAddressBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: PaymentAddressBaseType“] <AccountOwner>xs:string</AccountOwner> [1..1] <AccountOwnerSpecialAddress> „ComplexType: NameAndAddressType“ </AccountOwnerSpe\ cialAddress> [0..1] <IBAN> „SimpleType: IBANType“ </IBAN> [1..1] <FinancialInstitution> „ComplexType: NameAndAddressType“ </FinancialInstituti\ on> [0..1] [END BASE TYPE] <Balance> „SimpleType: SalaryAmountType“ </Balance> [1..1] <Reference>xs:string</Reference> [1..1] <AccountNumber>xs:string</AccountNumber> [0..1] <Line>xs:string</Line> [0..unbounded] </...>

ComplexType: TreatmentAddressExtensionType

Abstract	no
XML Instance Representation	<...> <ZSRNumber>xs:string</ZSRNumber> [0..1] <UID-BFS> „SimpleType: UID-BFSType“ </UID-BFS> [0..1] <GLNNumber>xs:string</GLNNumber> [0..1] <TreatmentInstitution>xs:string</TreatmentInstitution> [1..1] <Address> „ComplexType: AddressType“ </Address> [1..1] <EmailAddress> „SimpleType: EmailAddressType“ </EmailAddress> [0..1] <PhoneNumber>xs:string</PhoneNumber> [0..1] <MobilePhoneNumber>xs:string</MobilePhoneNumber> [0..1] </...>

ComplexType: TreatmentsType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <InitialTreatment> „ComplexType: TreatmentAddressExtensionType“ </InitialTreat\ ment> [1..1] <FurtherTreatment> „ComplexType: TreatmentAddressExtensionType“ </ FurtherTreatment> [0..unbounded] </...>

ComplexType: UVG-LAA-IdentificationType

Abstract	no
XML Instance Representation	<...> <InsuranceCompanyName>xs:string</InsuranceCompanyName> [1..1] [START CHOICE] <CustomerIdentity> „SimpleType: NotEmptyStringType“ </CustomerIdentity> [1..1] [START CHOICE] <ContractIdentity> „SimpleType: NotEmptyStringType“ </ContractIdentity> [1..1] <NoContract> „ComplexType: EmptyType“ </NoContract> [1..1] [END CHOICE] <OrderCustomerIdentity> „ComplexType: EmptyType“ </OrderCustomerIdentity> [1..1] [END CHOICE] </...>

ComplexType: VacationsType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...>

© Swissdec 2026

	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID>„SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <Vacation>„ComplexType: TimePeriodType“ </Vacation> [1..unbounded] </...>

ComplexType: ValidToType

Abstract	no
XML Instance Representation	<...> [START CHOICE] <To>xs:date</To> [1..1] <TakeWorkUpFullyDate>xs:date</TakeWorkUpFullyDate> [0..1] <Unknown>„ComplexType: EmptyType“ </Unknown> [1..1] [END CHOICE] </...>

ComplexType: WageReplacementBenefitsPastType

Abstract	no
XML Instance Representation	<...> <Salary>„ComplexType: WageReplacementBenefitsSalaryPastType“ </Salary> [1..12] <WageReplacementBenefitsAnnualTotal>„SimpleType: SalaryAmountType“ </WageReplace\ mentBenefitsAnnualTotal> [1..1] </...>

ComplexType: WageReplacementBenefitsSalaryPastType

Abstract	no
XML Instance Representation	<...> <Month>xs:gYearMonth</Month> [1..1] <KindOfSalaries>„ComplexType: KindOfSalariesTimedPastType“ </KindOfSala\ ries> [1..1] <Total>„SimpleType: SalaryAmountType“ </Total> [1..1] </...>

SimpleType: ActivityAtTimeOfAccidentType

Translation	Tätigkeit
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: skiing • Enum: snowboarding • Enum: footballOrsoccer • Enum: otherBallGames • Enum: jogging • Enum: fitnessTraining • Enum: bathingOrSwimming • Enum: hiking • Enum: walking • Enum: eBiking • Enum: bicycleOrMBT • Enum: carDriving • Enum: motorbiking • Enum: housekeepingWork • Enum: doItYourselfOrMaintenanceWork • Enum: gardeningOrForestAndAgricultureWork • Enum: brawlingOrFightingOrViolence • Enum: otherSport • Enum: otherActivitiesAtHome • Enum: otherActivities

© Swissdec 2026

SimpleType: AddressingDomainType

Translation	Domäne
Short description	Domäne
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: UVG-LAA • Enum: UVGZ-LAAC • Enum: KTG-AMC

SimpleType: BodyOrientedPartType

Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: eye_12 • Enum: pelvis_40 • Enum: elbow_32 • Enum: finger_36 • Enum: metatarsusWithoutToes_54 • Enum: ankle_53 • Enum: face_11 • Enum: metacarpusWithoutFingers_35 • Enum: wrist_34 • Enum: hipJoint_41 • Enum: jaw_16 • Enum: knee_51 • Enum: groin_43 • Enum: lungsInclusiveAirways_62 • Enum: multipleAreasUpperExtremities_37 • Enum: multipleAreasLowerExtremities_56 • Enum: kidney_65 • Enum: upperArm_31 • Enum: thigh_50 • Enum: earHearing_14 • Enum: rear_21 • Enum: shoulder_30 • Enum: thorax_23 • Enum: lowerArm_33 • Enum: lowerThigh_52 • Enum: teeth_15 • Enum: toes_55

SimpleType: BodyPartType

Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: abdomen_459 • Enum: bladder_669 • Enum: thoracicSpine_259 • Enum: entireBodySystemicEffect_819 • Enum: genitals_449 • Enum: neck_209 • Enum: cervicalSpine_249

© Swissdec 2026

	• Enum: heart_619 • Enum: otherInternalInjuries_609 • Enum: liver_639 • Enum: lumbarSpine_269 • Enum: spleen_649 • Enum: nose_139 • Enum: multipleInjuries_709 • Enum: skullBrain_109 • Enum: shockPsych_809 • Enum: coccyxButtocks_429

SimpleType: ContentTypeType

Translation	Attachment Typ
Short description	Arbeitsunfähigkeitsbescheinigung, Korrespondenz, Rechnung und Übriges
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: certificateOfIncapacityToWork • Enum: correspondence • Enum: invoice • Enum: other

SimpleType: CoverageType

Translation	Zustand der Deckung
Short description	Zustand der Deckung
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: yes • Enum: no • Enum: annulled • Enum: unknown

SimpleType: DaysType

Content type	Restriction
Base type	xs:decimal
Facets	• MinInclusive: -9999.99 • MaxInclusive: 9999.99

SimpleType: DigitalizationScopeType

Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: Basic • Enum: BasicCoverage • Enum: BasicCoverageSettlement • Enum: BasicDialog • Enum: BasicCoverageDialog • Enum: BasicCoverageSettlementDialog

© Swissdec 2026

SimpleType: ExpectedIncapacityToWorkType

Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: to30Days • Enum: moreThan30Days • Enum: unknown

SimpleType: HospitalCostDeductionEnumType

Translation	Attachment Typ
Short description	Arbeitsunfähigkeitsbescheinigung, Korrespondenz, Rechnung und Übriges
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: 10PercentageOr10CHF • Enum: 20PercentageOr20CHF

SimpleType: HoursAndIndustryMinutesType

Content type	Restriction
Base type	xs:decimal
Facets	• Pattern: [0-9]+\.[0-9]{2}

SimpleType: IBANType

Translation	IBAN Nummer
Short description	IBAN Nummer
Technical description	Regex Pattern Sample https://www.six-interbank-clearing.com/de/home/standardization/iso-payments/ customer-bank/implementation-guidelines.html implementation-guidelines-ct-examples.zip - simpleType name="IBAN2007Identifier" / string - pattern value="[A-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,30}" https://www.w3.org/TR/xmlschema-2/ - F Regular Expressions - S{n,m} = S{n} and n=m - All strings in L(S{n,n}). (All sequences of exactly n strings from L(S))
Content type	Restriction
Base type	xs:string
Facets	• Pattern: [A-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,30}

SimpleType: MainStateType

Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: registrationReceived • Enum: checkingDeclaration

© Swissdec 2026

	• Enum: verifyingClaim • Enum: performingEntitlement • Enum: communicationStopped

SimpleType: OtherInsurancePredefinedType

Translation	Definierte andere Versicherer
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: suva • Enum: otherCompulsoryAccidentInsurance • Enum: KTG-AMC • Enum: IV-AI • Enum: AHV-AVS • Enum: occupationalPreventionEstablishment • Enum: militaryInsurance • Enum: unemploymentInsurance • Enum: maternetyIdemnity

SimpleType: PositionPartType

Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: left_1 • Enum: right_2 • Enum: both_3 • Enum: unassigned_9

SimpleType: ReductionReasonEnumType

Translation	Attachment Typ
Short description	Arbeitsunfähigkeitsbescheinigung, Korrespondenz, Rechnung und Übriges
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: provisional • Enum: partialApproval • Enum: legalReduction • Enum: contractualReduction • Enum: overcompensation

SimpleType: SalaryBaseCodeType

Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: BasicWages • Enum: FamilyIncomeSupplement • Enum: VacationAndPublicHolidayCompensation • Enum: OtherSupplements • Enum: Gratuity • Enum: WageReplacement

© Swissdec 2026

SimpleType: SupportedMinorSchemaVersionAttributeType

Translation	Unterstützte Minor Schema Versionen
Short description	Hier werden die kompatiblen Minor Versionen aufgelistet
Technical description	pattern mehrere Vorkommastellen ermöglichen und führenden Nullen ausschliessen [0-9]\.[0-9] = Major eine Stelle, Punkt, Minor eine Stelle oder [1-9]\d*\.[0-9] = beliebige Major ohne führende Nullen, Punkt, eine Stelle Minor
Content type	Restriction
Base type	xs:decimal
Facets	• Pattern: [0-9]\.[0-9]|[1-9]\d*\.[0-9] • Enum: 0.0 • Enum: 0.1

SimpleType: TypeOfInjuryType

Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: bite_01 • Enum: fracture_02 • Enum: inflammation_03 • Enum: contusion_04 • Enum: bruise_05 • Enum: chap_06 • Enum: cut_07 • Enum: shot_08 • Enum: swelling_09 • Enum: graze_10 • Enum: compressionSpine_11 • Enum: sting_12 • Enum: tearFissure_13 • Enum: burn_14 • Enum: sprainTwist_15 • Enum: poisoning_16 • Enum: causticity_17 • Enum: pulledMuscleLigament_18 • Enum: foreignObject_20 • Enum: dislocation_21

G.5 Schema Dokumentation SwissdecComponents.xsd

Target Namespace	urn:ch:swissdec:basis:v1:20260306:components
Declared Namespaces	• ep : urn:ch:swissdec:basis:v1:20260306:components • xs : http://www.w3.org/2001/XMLSchema
Version	0.0

ComplexType: AccessInformationType

Abstract	no

XML Instance Representation	`<...>`
	`  <Url>xs:string</Url>  [1..1]`
	`  <ExpiryDate>xs:dateTime</ExpiryDate>  [1..1]`
	`</...>`

© Swissdec 2026

ComplexType: AddresseeJobType

Abstract	no
Parent type	„ComplexType: AddresseeType“
Translation	Adressierung der Endempfänger
Short description	Adressierung der Endempfänger
Technical description	Auflistung der Endempfänger, die die übermittelten Daten erhalten sollen.
XML Instance Representation	<...> [BASE TYPE „ComplexType: AddresseeType“] <AddresseeIdentification>„SimpleType: IDType“ </AddresseeIdentification> [1..1] [END BASE TYPE] <ProcessByDistributor>„SimpleType: SimpleBooleanType“ </ProcessByDistributor> [1..1] </...>

ComplexType: AddresseeResponseContextType

Abstract	no
Parent type	„ComplexType: ResponseContextType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: ResponseContextType“] <UserAgent>„ComplexType: UserAgentType“ </UserAgent> [1..1] <InstitutionName>xs:string</InstitutionName> [1..1] <TransmissionDate>xs:dateTime</TransmissionDate> [1..1] <ResponseID>„SimpleType: IDType“ </ResponseID> [1..1] <RequestID>„SimpleType: IDType“ </RequestID> [1..1] [END BASE TYPE] <ProducerResponseNotifications>„ComplexType: FeedbackNotificationsType“ </ProducerResponseNotifications> [1..1] <Warning>„ComplexType: NotificationsType“ </Warning> [0..1] <Info>„ComplexType: NotificationsType“ </Info> [0..1] </...>

ComplexType: AddresseeResponseType

Abstract	no
XML Instance Representation	<...> <AddresseeContext>„ComplexType: AddresseeResponseContextType“ </AddresseeContext> [1..1] </...>

ComplexType: AddresseeType

Abstract	no
Short description	Informationen zum Adressaten
XML Instance Representation	<...> <AddresseeIdentification>„SimpleType: IDType“ </AddresseeIdentification> [1..1] </...>

ComplexType: AnswerAmountType

Abstract	no
XML Instance Representation	<...> <Default>„SimpleType: SalaryAmountType“ </Default> [0..1] <Value>„SimpleType: SalaryAmountType“ </Value> [0..1] </...>

ComplexType: AnswerBooleanType

Abstract	no
XML Instance Representation	<...>

© Swissdec 2026

	<Default>xs:boolean</Default> [0..1] <Value>xs:boolean</Value> [0..1] </...>

ComplexType: AnswerDateTimeType

Abstract	no
XML Instance Representation	<...> <Default>xs:dateTime</Default> [0..1] <Value>xs:dateTime</Value> [0..1] </...>

ComplexType: AnswerDateType

Abstract	no
XML Instance Representation	<...> <Default>xs:date</Default> [0..1] <Value>xs:date</Value> [0..1] </...>

ComplexType: AnswerDoubleType

Abstract	no
XML Instance Representation	<...> <Default>xs:double</Default> [0..1] <Value>xs:double</Value> [0..1] </...>

ComplexType: AnswerIntegerType

Abstract	no
XML Instance Representation	<...> <Default>xs:integer</Default> [0..1] <Value>xs:integer</Value> [0..1] </...>

ComplexType: AnswerStringType

Abstract	no
XML Instance Representation	<...> <Default>xs:string</Default> [0..1] <Value>xs:string</Value> [0..1] </...>

ComplexType: AnswerYesNoUnknownType

Abstract	no
XML Instance Representation	<...> <Default> „SimpleType: YesNoUnknownType“ </Default> [0..1] <Value> „SimpleType: YesNoUnknownType“ </Value> [0..1] </...>

ComplexType: AwaitPartControlType

Abstract	no
XML Instance Representation	<...> <ExpirationDate>xs:dateTime</ExpirationDate> [0..1] </...>

ComplexType: BooleanStoryType

Abstract	no

© Swissdec 2026

Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID>„SimpleType: IDType“</StoryID> [1..1] [END BASE TYPE] <Valid>xs:boolean</Valid> [1..1] </...>

ComplexType: CaseContextBaseType

Abstract	no
XML Instance Representation	<...> <ReceivedStoryIDs>„ComplexType: ReceivedStoriesType“</ReceivedStoryIDs> [0..1] <SuppressedSenderStoryIDs>„ComplexType: ReceivedStoriesType“</SuppressedSenderSto\ryIDs> [0..1] <SuppressedInstitutionStoryIDs>„ComplexType: ReceivedStoriesType“</SuppressedIn\stitutionStoryIDs> [0..1] <Warning>„ComplexType: StoryNotificationsType“</Warning> [0..1] <Info>„ComplexType: StoryNotificationsType“</Info> [0..1] </...>

ComplexType: CaseContextType

Abstract	no
Parent type	„ComplexType: CaseContextBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: CaseContextBaseType“] <ReceivedStoryIDs>„ComplexType: ReceivedStoriesType“</ReceivedStoryIDs> [0..1] <SuppressedSenderStoryIDs>„ComplexType: ReceivedStoriesType“</SuppressedSenderS\toryIDs> [0..1] <SuppressedInstitutionStoryIDs>„ComplexType: ReceivedStoriesType“</SuppressedIn\stitutionStoryIDs> [0..1] <Warning>„ComplexType: StoryNotificationsType“</Warning> [0..1] <Info>„ComplexType: StoryNotificationsType“</Info> [0..1] [END BASE TYPE] <Credentials>„ComplexType: CredentialsType“</Credentials> [1..1] </...>

ComplexType: CheckInteroperabilityRequestType

Abstract	no
XML Instance Representation	<...> <UserAgent>„ComplexType: UserAgentType“</UserAgent> [1..1] <UmlautString>xs:string</UmlautString> [1..1] <FirstOperand>„SimpleType: SalaryAmountType“</FirstOperand> [1..1] <SecondOperand>„SimpleType: SalaryAmountType“</SecondOperand> [1..1] <SystemDateTime>xs:dateTime</SystemDateTime> [1..1] <MonitoringID>„SimpleType: MonitoringIDType“</MonitoringID> [0..1] </...>

ComplexType: CheckInteroperabilityResponseType

Abstract	no
XML Instance Representation	<...> <UserAgent>„ComplexType: UserAgentType“</UserAgent> [1..1] <UmlautStringIsCorrect>„SimpleType: SimpleBooleanType“</UmlautStringIsCor\rect> [1..1] <FirstOperandIsCorrect>„SimpleType: SimpleBooleanType“</FirstOperandIsCor\rect> [1..1] <UmlautString>xs:string</UmlautString> [1..1] <AdditionResult>„SimpleType: SalaryAmountType“</AdditionResult> [1..1] <SubtractionResult>„SimpleType: SalaryAmountType“</SubtractionResult> [1..1] <SystemDateTime>xs:dateTime</SystemDateTime> [1..1] </...>

ComplexType: CompanyUIDType

Abstract	no
XML Instance Representation	<...> <UID-BFS>„ComplexType: UID-BFS-UnknownType“</UID-BFS> [1..1] <DelegateUID-BFS>„SimpleType: UID-BFSType“</DelegateUID-BFS> [0..1]

© Swissdec 2026

	</...>

ComplexType: CompletionType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
Short description	Informationen zur Freigabe der Daten.
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID>„SimpleType: IDType“</StoryID> [1..1] [END BASE TYPE] <Completion>„ComplexType: AccessInformationType“</Completion> [1..1] <Credentials>„ComplexType: ResultType“</Credentials> [0..1] </...>

ComplexType: ConsumerFaultType

Abstract	no
Parent type	„ComplexType: ResponseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: ResponseType“] <ResponseContext>„ComplexType: ResponseContextType“</ResponseContext> [1..1] [END BASE TYPE] <FaultState>„ComplexType: FaultStateType“</FaultState> [1..1] </...>

ComplexType: CredentialsType

Abstract	no
Translation	Berechtigungsnachweis
Short description	Berechtigungsnachweis
XML Instance Representation	<...> <Key>xs:string</Key> [1..1] <Password>xs:string</Password> [1..1] </...>

ComplexType: DialogMessageType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
Technical description	Semistrukturierte Daten für Dialog zwischen den Fall-Teilnehmern.
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID>„SimpleType: IDType“</StoryID> [1..1] [END BASE TYPE] <StandardDialogID>„SimpleType: StandardFormIDType“</StandardDialogID> [1..1] <Previous>„ComplexType: PreviousType“</Previous> [0..1] <Title>„SimpleType: IDType“</Title> [0..1] <Description>xs:token</Description> [0..1] <Paragraph>„ComplexType: ParagraphType“</Paragraph> [1..unbounded] <Section>„ComplexType: SectionType“</Section> [0..unbounded] </...>

ComplexType: DistributorRequestContextType

Abstract	no
XML Instance Representation	<...> <UserAgent>„ComplexType: UserAgentType“</UserAgent> [1..1] <TransmissionDate>xs:dateTime</TransmissionDate> [1..1] <DistributorRequestID>„SimpleType: IDType“</DistributorRequestID> [1..1] <VersionMappingFrom>„ComplexType: VersionMappingFromType“</VersionMapping\ From> [0..1] <ProducerSecurityTokens>„ComplexType: ProducerSecurityTokensType“</ProducerSecuri\ tyTokens> [1..1]

© Swissdec 2026

	<SignatureCertificateUID-BFS> „ComplexType: SignatureCertificateUID-BFSType“ </SignatureCertificateUID-BFS> [0..1] <ProducerResponseNotifications> „ComplexType: FeedbackNotificationsType“ </ProducerResponseNotifications> [1..1] <ConsumerNotifications> „ComplexType: FeedbackNotificationsType“ </ConsumerNotifications> [0..1] </...>

ComplexType: DuplicateType

Abstract	no
XML Instance Representation	<...> <LastDistributorRequestID> „SimpleType: IDType“ </LastDistributorRequestID> [1..1] </...>

ComplexType: EmptyType

Abstract	no
XML Instance Representation	<...> </...>

ComplexType: ErrorResponseType

Abstract	no
Translation	Fehler
Short description	Die Verarbeitung war nicht erfolgreich und es sind Fehler aufgetreten.
XML Instance Representation	<...> <EndUserInformation>xs:string</EndUserInformation> [1..1] <DetailInformation>xs:string</DetailInformation> [0..1] [START CHOICE] <FaultInformation> „ComplexType: ConsumerFaultType“ </FaultInformation> [1..1] <PlannedMaintenance> „ComplexType: PlannedMaintenanceWindowType“ </PlannedMaintenance> [1..1] [END CHOICE] </...>

ComplexType: FaultStateType

Abstract	no
Translation	Fehler Status
XML Instance Representation	<...> <Code> „SimpleType: FaultCodeType“ </Code> [1..1] <Error> „ComplexType: NotificationsType“ </Error> [0..1] <Warning> „ComplexType: NotificationsType“ </Warning> [0..1] <Info> „ComplexType: NotificationsType“ </Info> [0..1] </...>

ComplexType: FaultType

Abstract	no
Parent type	„ComplexType: ResponseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: ResponseType“] <ResponseContext> „ComplexType: ResponseContextType“ </ResponseContext> [1..1] [END BASE TYPE] [START CHOICE] <Fault> „ComplexType: FaultStateType“ </Fault> [1..1] <PlannedMaintenance> „ComplexType: PlannedMaintenanceWindowType“ </PlannedMaintenance> [1..1] [END CHOICE] </...>

ComplexType: FeedbackNotificationsType

Abstract	no
XML Instance Representation	<...> <Warning> „ComplexType: NotificationType“ </Warning> [0..unbounded]

© Swissdec 2026

	<Info> „ComplexType: NotificationType“ </Info> [0..unbounded] </...>

ComplexType: GetStatusRequestType

Abstract	no
Parent type	„ComplexType: RequestType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: RequestType“] <RequestContext> „ComplexType: RequestContextType“ </RequestContext> [1..1] [END BASE TYPE] <JobKey> „SimpleType: JobKeyType“ </JobKey> [1..1] </...>

ComplexType: GetStatusResponseType

Abstract	no
Parent type	„ComplexType: ResponseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: ResponseType“] <ResponseContext> „ComplexType: ResponseContextType“ </ResponseContext> [1..1] [END BASE TYPE] <JobFinished> „SimpleType: SimpleBooleanType“ </JobFinished> [1..1] </...>

ComplexType: InitialResponseType

Abstract	no
Parent type	„ComplexType: ResponseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: ResponseType“] <ResponseContext> „ComplexType: ResponseContextType“ </ResponseContext> [1..1] [END BASE TYPE] <JobKey> „SimpleType: JobKeyType“ </JobKey> [1..1] </...>

ComplexType: MessageType

Abstract	no
XML Instance Representation	<...> </...>

ComplexType: NotificationType

Abstract	no
Translation	Meldung
Short description	Meldung
XML Instance Representation	<...> <QualityLevel> „SimpleType: QualityLevelType“ </QualityLevel> [1..1] <DescriptionCode> „SimpleType: DescriptionCodeType“ </DescriptionCode> [1..1] <Description>xs:string</Description> [1..1] </...>

ComplexType: NotificationsStoryType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <Notifications> „ComplexType: NotificationsType“ </Notifications> [1..1] </...>

© Swissdec 2026

ComplexType: NotificationsType

Abstract	no
XML Instance Representation	<...> <Notification> „ComplexType: NotificationType“ </Notification> [1..unbounded] </...>

ComplexType: ParagraphAnswerType

Abstract	no
XML Instance Representation	<...> [START CHOICE] <String> „ComplexType: AnswerStringType“ </String> [1..1] <Integer> „ComplexType: AnswerIntegerType“ </Integer> [1..1] <Double> „ComplexType: AnswerDoubleType“ </Double> [1..1] <Boolean> „ComplexType: AnswerBooleanType“ </Boolean> [1..1] <Date> „ComplexType: AnswerDateType“ </Date> [1..1] <DateTime> „ComplexType: AnswerDateTimeType“ </DateTime> [1..1] <YesNoUnknown> „ComplexType: AnswerYesNoUnknownType“ </YesNoUnknown> [1..1] <Amount> „ComplexType: AnswerAmountType“ </Amount> [1..1] [END CHOICE] <Problem> „SimpleType: NotEmptyStringType“ </Problem> [0..1] </...>

ComplexType: ParagraphType

Abstract	no
XML Instance Representation	<...> <ID>xs:short</ID> [1..1] <Label>xs:token</Label> [1..1] [START CHOICE] <Value> „ComplexType: ParagraphValueType“ </Value> [0..1] <Answer> „ComplexType: ParagraphAnswerType“ </Answer> [0..1] [END CHOICE] </...>

ComplexType: ParagraphValueType

Abstract	no
XML Instance Representation	<...> [START CHOICE] <String>xs:string</String> [1..1] <Integer>xs:integer</Integer> [1..1] <Double>xs:double</Double> [1..1] <Boolean>xs:boolean</Boolean> [1..1] <Date>xs:date</Date> [1..1] <DateTime>xs:dateTime</DateTime> [1..1] <YesNoUnknown> „SimpleType: YesNoUnknownType“ </YesNoUnknown> [1..1] <Amount> „SimpleType: SalaryAmountType“ </Amount> [1..1] [END CHOICE] </...>

ComplexType: PingConsumerRequestType

Abstract	no
XML Instance Representation	<...> <UserAgent> „ComplexType: UserAgentType“ </UserAgent> [1..1] <Timestamp>xs:dateTime</Timestamp> [1..1] <NextCheck>xs:dateTime</NextCheck> [1..1] <RegisteredMaintenance> „ComplexType: RegisteredMaintenanceType“ </RegisteredMain\ tenance> [0..1] </...>

ComplexType: PingConsumerResponseType

Abstract	no
XML Instance Representation	<...> <UserAgent> „ComplexType: UserAgentType“ </UserAgent> [1..1] <Timestamp>xs:dateTime</Timestamp> [1..1] [START CHOICE] <PlannedMaintenance> „ComplexType: PlannedMaintenanceType“ </PlannedMaintenan\ ce> [1..1] <NoPlannedMaintenance> „ComplexType: EmptyType“ </NoPlannedMaintenance> [1..1] [END CHOICE]

© Swissdec 2026

	</...>

ComplexType: PingRequestType

Abstract	no
XML Instance Representation	<...> <UserAgent> „ComplexType: UserAgentType“ </UserAgent> [1..1] <SystemDateTime>xs:dateTime</SystemDateTime> [1..1] <MonitoringID> „SimpleType: MonitoringIDType“ </MonitoringID> [0..1] </...>

ComplexType: PingResponseType

Abstract	no
XML Instance Representation	<...> <UserAgent> „ComplexType: UserAgentType“ </UserAgent> [1..1] <SystemDateTime>xs:dateTime</SystemDateTime> [1..1] </...>

ComplexType: PlannedMaintenanceType

Abstract	no
XML Instance Representation	<...> <Start>xs:dateTime</Start> [1..1] <End>xs:dateTime</End> [1..1] <ProducerMessages> „ComplexType: ProducerMessagesType“ </ProducerMessages> [1..1] </...>

ComplexType: PlannedMaintenanceWindowType

Abstract	no
XML Instance Representation	<...> <Start>xs:dateTime</Start> [1..1] <End>xs:dateTime</End> [1..1] <Message>xs:string</Message> [1..1] </...>

ComplexType: PreviousType

Abstract	no
XML Instance Representation	<...> [START CHOICE] <RequestStoryID> „SimpleType: IDType“ </RequestStoryID> [1..1] <ResponseStoryID> „SimpleType: IDType“ </ResponseStoryID> [1..1] [END CHOICE] </...>

ComplexType: ProducerMessagesType

Abstract	no
XML Instance Representation	<...> <Message> „ComplexType: MessageType“ </Message> [3..unbounded] </...>

ComplexType: ProducerSecurityTokensType

Abstract	no
XML Instance Representation	<...> <Timestamp>xs:dateTime</Timestamp> [1..unbounded] <X509Certificate> „ComplexType: X509CertificateType“ </ X509Certificate> [1..unbounded] </...>

ComplexType: ReceivedStoriesType

Abstract	no

© Swissdec 2026

XML Instance Representation	<...> <StoryID> „SimpleType: IDType“ </StoryID> [1..unbounded] </...>

ComplexType: RegisteredMaintenanceType

Abstract	no
XML Instance Representation	<...> <Start>xs:dateTime</Start> [1..1] <End>xs:dateTime</End> [1..1] </...>

ComplexType: RequestContextType

Abstract	no
Translation	Kontext der Anfrage
Short description	Der RequestContext enthält Informationen zum Zeitpunkt der Übermittlung, dem Absender und der Sprache.
XML Instance Representation	<...> <UserAgent> „ComplexType: UserAgentType“ </UserAgent> [1..1] <CompanyName> „SimpleType: NotEmptyStringType“ </CompanyName> [1..1] <TransmissionDate>xs:dateTime</TransmissionDate> [1..1] <RequestID> „SimpleType: IDType“ </RequestID> [1..1] <LanguageCode> „SimpleType: LanguageCodeType“ </LanguageCode> [1..1] <MonitoringID> „SimpleType: MonitoringIDType“ </MonitoringID> [0..1] </...>

ComplexType: RequestType

Abstract	no
XML Instance Representation	<...> <RequestContext> „ComplexType: RequestContextType“ </RequestContext> [1..1] </...>

ComplexType: ResponseContextType

Abstract	no
Translation	Kontext der Antwort
Short description	Der ResponseContext enthält Informationen zum Zeitpunkt der Übermittlung, dem Absender und der Sprache.
XML Instance Representation	<...> <UserAgent> „ComplexType: UserAgentType“ </UserAgent> [1..1] <InstitutionName>xs:string</InstitutionName> [1..1] <TransmissionDate>xs:dateTime</TransmissionDate> [1..1] <ResponseID> „SimpleType: IDType“ </ResponseID> [1..1] <RequestID> „SimpleType: IDType“ </RequestID> [1..1] </...>

ComplexType: ResponseType

Abstract	no
XML Instance Representation	<...> <ResponseContext> „ComplexType: ResponseContextType“ </ResponseContext> [1..1] </...>

ComplexType: ResultType

Abstract	no
XML Instance Representation	<...> <Credentials> „ComplexType: CredentialsType“ </Credentials> [1..1] </...>

© Swissdec 2026

ComplexType: SectionType

Abstract	no
XML Instance Representation	<...> <Heading> „SimpleType: IDType“ </Heading> [0..1] <Description>xs:token</Description> [0..1] </...>

ComplexType: SignatureCertificateUID-BFSType

Abstract	no
XML Instance Representation	<...> <CommonName>xs:string</CommonName> [1..1] <UID>xs:string</UID> [1..1] </...>

ComplexType: StoryBaseType

Abstract	no
XML Instance Representation	<...> <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] </...>

ComplexType: StoryNotificationType

Abstract	no
Parent type	„ComplexType: NotificationType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: NotificationType“] <QualityLevel> „SimpleType: QualityLevelType“ </QualityLevel> [1..1] <DescriptionCode> „SimpleType: DescriptionCodeType“ </DescriptionCode> [1..1] <Description>xs:string</Description> [1..1] [END BASE TYPE] <StoryID> „SimpleType: IDType“ </StoryID> [0..1] </...>

ComplexType: StoryNotificationsType

Abstract	no
XML Instance Representation	<...> <Notification> „ComplexType: StoryNotificationType“ </Notification> [1..unbounded] </...>

ComplexType: SubstitutionMappingType

Abstract	no
XML Instance Representation	<...> <MappedPredecessorDistributorRequestID> „SimpleType: IDType“ </MappedPredecessorDis\ tributorRequestID> [1..1] <MappedPredecessorConsumerResponseID> „SimpleType: IDType“ </MappedPredecessorConsu\ merResponseID> [1..1] </...>

ComplexType: SynchronizeRequestType

Abstract	no
Parent type	„ComplexType: RequestType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: RequestType“] <RequestContext> „ComplexType: RequestContextType“ </RequestContext> [1..1] [END BASE TYPE] <Sender> „ComplexType: CompanyUIDType“ </Sender> [1..1] </...>

© Swissdec 2026

ComplexType: TimePeriodType

Abstract	no
XML Instance Representation	<...> <from>xs:date</from> [1..1] <until>xs:date</until> [1..1] </...>

ComplexType: UID-BFS-UnknownType

Abstract	no
Translation	Unternehmens UID-BFS
Short description	Unternehmens-Identifikationsnummer / UID-BFS
XML Instance Representation	<...> [START CHOICE] <UID> „SimpleType: UID-BFSType“ </UID> [1..1] <Unknown> „ComplexType: EmptyType“ </Unknown> [1..1] [END CHOICE] </...>

ComplexType: UserAgentType

Abstract	no
Translation	Applikationsinformation
Short description	Beschreibung der wesentlichen System-Identifikation-Daten
Technical description	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert
XML Instance Representation	<...> <Producer> „SimpleType: NotEmptyStringType“ </Producer> [1..1] <Name> „SimpleType: NotEmptyStringType“ </Name> [1..1] <Version> „SimpleType: NotEmptyStringType“ </Version> [1..1] <StandardVersion>xs:decimal</StandardVersion> [1..1] <Certificate> „SimpleType: NotEmptyStringType“ </Certificate> [1..1] </...>

ComplexType: UserNotificationType

Abstract	no
XML Instance Representation	<...> <Name>xs:string</Name> [1..1] <EmailAddress> „SimpleType: EmailAddressType“ </EmailAddress> [1..1] <PhoneNumber>xs:string</PhoneNumber> [1..1] </...>

ComplexType: VersionMappingFromType

Abstract	no
XML Instance Representation	<...> <WSDLNamespace>xs:anyURI</WSDLNamespace> [1..1] <ServiceTypesNamespace>xs:anyURI</ServiceTypesNamespace> [1..1] <MinorVersion>xs:decimal</MinorVersion> [1..1] </...>

ComplexType: X509CertificateType

Abstract	no
XML Instance Representation	<...> <Usage> „SimpleType: UsageType“ </Usage> [1..1] <DistinguishedNameOwner>xs:string</DistinguishedNameOwner> [1..1] <DistinguishedNameIssuer>xs:string</DistinguishedNameIssuer> [1..1]

© Swissdec 2026

	<SecuredElement>xs:string</SecuredElement> [1..unbounded] <PEM>xs:base64Binary</PEM> [1..1] </...>

SimpleType: DescriptionCodeType

Translation	Beschreibungscode
Technical description	Severity: Information Code 9999 ist immer für Freitext reserviert. Die restlichen Codes sind dynamisch in AcknowledgementNotification spezifiziert.
Content type	Restriction
Base type	xs:nonNegativeInteger
Facets	• TotalDigits: 7

SimpleType: EmailAddressType

Translation	Email Adresse
Short description	Email Adresse
Technical description	Email Adresse
Content type	Restriction
Base type	xs:string
Facets	• Pattern: [^@]+@[^\.]+\..+

SimpleType: FaultCodeType

Translation	Fehler-Code
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: NOT_accepted • Enum: NOT_plausible • Enum: NOT_valid

SimpleType: IDType

Content type	Restriction
Base type	xs:string
Facets	• MinLength: 1 • MaxLength: 255

SimpleType: InstanceRefIDType

Translation	Instance Referenz ID
Content type	Restriction
Base type	xs:string
Facets	• Pattern: #.*

SimpleType: JobKeyType

Translation	Eindeutiger Bezeichner für den Job

© Swissdec 2026

Short description	Eindeutiger Bezeichner für den Job
Content type	Restriction
Base type	xs:string
Facets	• MinLength: 10

SimpleType: LanguageCodeType

Translation	Sprachcode
Short description	Sprachcode
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: de • Enum: fr • Enum: it • Enum: en

SimpleType: Max100PercentType

Translation	Prozent (maximal 100%)
Content type	Restriction
Base type	xs:decimal
Facets	• MaxInclusive: 100.00 • Pattern: [0-9]+\.[0-9]{2}

SimpleType: MonitoringIDType

Translation	KontrollID
Short description	Die MonitoringID erlaubt es, auf den Testsystemen Übermittlungen den Benutzern zuzuweisen. Sie ist in der Produktion überflüssig.
Technical description	Die KontrollID wird vor allem in der Referenzapplikation zur Aufteilung der Daten verwendet
Content type	Restriction
Base type	xs:string
Facets	• MinLength: 1 • MaxLength: 32

SimpleType: NotEmptyStringType

Content type	Restriction
Base type	xs:string
Facets	• MinLength: 1

SimpleType: PercentType

Translation	Prozent
Content type	Restriction

© Swissdec 2026

Base type	xs:decimal
Facets	• Pattern: [0-9]+\.[0-9]{2}

SimpleType: QualityLevelType

Translation	Qualität-Stufen
Short description	Qualität-Stufen der Übermittlung
Technical description	Folgende Stufen gelten Validity: erste Stufe; Plausibility: zweite Stufe; Acceptance: dritte Stufe; Comment: vierte Stufe;
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: Validity • Enum: Plausibility • Enum: Acceptance • Enum: Comment

SimpleType: SalaryAmountType

Translation	LohnbetragsTyp
Short description	LohnbetragsTyp
Technical description	LohnbetragsTyp
Content type	Restriction
Base type	xs:decimal
Facets	• Pattern: [\-]?[0-9]+\.[0-9]{2}

SimpleType: SimpleBooleanType

Translation	SimpleBoolean
Short description	Boolean
Content type	Restriction
Base type	xs:boolean
Facets	• Pattern: true • Pattern: false

SimpleType: StandardFormIDType

Translation	Standard Formular ID
Content type	Restriction
Base type	„SimpleType: IDType“
Facets	• Pattern: notStandard|[0-9]{4}\.[0-9]{4}\.[0-9]{4}-[0-9]{3}

SimpleType: UID-BFSType

Translation	Unternehmens UID-BFS Typ
Short description	Unternehmens UID-BFS Typ

© Swissdec 2026

Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Pattern: CHE-[0-9]{3}\.[0-9]{3}\.[0-9]{3}

SimpleType: UsageType

Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: Signature • Enum: QualifiedSignature • Enum: Encryption

SimpleType: YesNoUnknownType

Translation	ja, nein oder unbekannt
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: yes • Enum: no • Enum: unknown

G.6 Schema Dokumentation Common.xsd

Target Namespace	urn:ch:swissdec:common:v3:20260306
Declared Namespaces	• c : urn:ch:swissdec:common:v3:20260306 • ep : urn:ch:swissdec:basis:v1:20260306:components • xs : http://www.w3.org/2001/XMLSchema
Version	0.0

ComplexType: AddressExtensionType

Abstract	no
Parent type	„ComplexType: AddressWorkplaceType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: AddressWorkplaceType“] <ComplementaryLine>xs:string</ComplementaryLine> [0..1] <Street>xs:string</Street> [0..1] <Locality>xs:string</Locality> [0..1] <ZIP-Code> „SimpleType: ZIP-CodeType“ </ZIP-Code> [1..1] <City>xs:string</City> [1..1] <Country>xs:string</Country> [0..1] [END BASE TYPE] <Canton> „SimpleType: CantonAddressType“ </Canton> [0..1] <MunicipalityID> „SimpleType: MunicipalityIDType“ </MunicipalityID> [0..1] </...>

ComplexType: AddressType

Abstract	no
XML Instance Representation	<...> <ComplementaryLine>xs:string</ComplementaryLine> [0..1] <Street>xs:string</Street> [0..1] <Postbox>xs:string</Postbox> [0..1] <Locality>xs:string</Locality> [0..1] <ZIP-Code> „SimpleType: ZIP-CodeType“ </ZIP-Code> [1..1] <City>xs:string</City> [1..1] <Country>xs:string</Country> [0..1] </...>

© Swissdec 2026

ComplexType: AddressWorkplaceType

Abstract	no
XML Instance Representation	<...> <ComplementaryLine>xs:string</ComplementaryLine> [0..1] <Street>xs:string</Street> [0..1] <Locality>xs:string</Locality> [0..1] <ZIP-Code> „SimpleType: ZIP-CodeType“ </ZIP-Code> [1..1] <City>xs:string</City> [1..1] <Country>xs:string</Country> [0..1] </...>

ComplexType: BalanceType

Abstract	no
XML Instance Representation	<...> [START CHOICE] <FavourCompany> „SimpleType: SalaryAmountAbsoluteType“ </FavourCompany> [1..1] <FavourInstitution> „SimpleType: SalaryAmountAbsoluteType“ </FavourInstituti\ on> [1..1] <FavorPerson> „SimpleType: SalaryAmountAbsoluteType“ </FavorPerson> [1..1] [END CHOICE] </...>

ComplexType: CertificateSignRequestBaseType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <PEM>xs:base64Binary</PEM> [1..1] </...>

ComplexType: CertificateSignRequestType

Abstract	no
Parent type	„ComplexType: CertificateSignRequestBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: CertificateSignRequestBaseType“] [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <PEM>xs:base64Binary</PEM> [1..1] [END BASE TYPE] <OneTimePassword> „SimpleType: NotEmptyStringType“ </OneTimePassword> [1..1] </...>

ComplexType: CertificateSignResponseType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <SubjectDN> „SimpleType: NotEmptyStringType“ </SubjectDN> [1..1] <IssuerDN> „SimpleType: NotEmptyStringType“ </IssuerDN> [1..1] <NotBefore>xs:dateTime</NotBefore> [1..1] <NotAfter>xs:dateTime</NotAfter> [1..1] <PEM>xs:base64Binary</PEM> [1..1] </...>

ComplexType: CivilStatusAndDateType

Abstract	no

© Swissdec 2026

XML Instance Representation	<...> <Status> „SimpleType: CivilStatusType“ </Status> [1..1] <ValidAsOf>xs:date</ValidAsOf> [0..1] </...>

ComplexType: CompanyDescriptionBaseType

Abstract	no
XML Instance Representation	<...> <Name> „ComplexType: CompanyNameType“ </Name> [1..1] <Owner> „ComplexType: CompanyOwnerType“ </Owner> [0..1] <Address> „ComplexType: AddressType“ </Address> [1..1] <UID-BFS> „ComplexType: UID-BFS-UnknownType“ </UID-BFS> [1..1] <Delegate> „ComplexType: DelegateType“ </Delegate> [0..1] </...>

ComplexType: CompanyDescriptionType

Abstract	no
Parent type	„ComplexType: CompanyDescriptionBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: CompanyDescriptionBaseType“] <Name> „ComplexType: CompanyNameType“ </Name> [1..1] <Owner> „ComplexType: CompanyOwnerType“ </Owner> [0..1] <Address> „ComplexType: AddressType“ </Address> [1..1] <UID-BFS> „ComplexType: UID-BFS-UnknownType“ </UID-BFS> [1..1] <Delegate> „ComplexType: DelegateType“ </Delegate> [0..1] [END BASE TYPE] <Workplace> „ComplexType: WorkplaceType“ </Workplace> [1..unbounded] <CompanyWorkingTime> „ComplexType: CompanyWorkingTimeIDType“ </ CompanyWorkingTime> [1..unbounded] </...>

ComplexType: CompanyNameType

Abstract	no
XML Instance Representation	<...> <HR-RC-Name>xs:string</HR-RC-Name> [1..1] <ComplementaryLine>xs:string</ComplementaryLine> [0..2] </...>

ComplexType: CompanyOwnerType

Abstract	no
XML Instance Representation	<...> <Firstname>xs:string</Firstname> [1..1] <Lastname>xs:string</Lastname> [1..1] </...>

ComplexType: CompanyWorkingTimeIDType

Abstract	no
XML Instance Representation	<...> <Description>xs:string</Description> [0..1] [START CHOICE] <WeeklyHours> „SimpleType: HoursOrLessonsType“ </WeeklyHours> [1..1] <WeeklyLessons> „SimpleType: HoursOrLessonsType“ </WeeklyLessons> [1..1] <WeeklyHoursAndLessons> „ComplexType: WeeklyHoursAndLessonsType“ </WeeklyHoursAnd\ Lessons> [1..1] [END CHOICE] </...>

ComplexType: ContactRequestType

Abstract	no
XML Instance Representation	<...> <Contact> „ComplexType: ContactType“ </Contact> [1..1] <Info> „ComplexType: NotificationsType“ </Info> [1..1] </...>

© Swissdec 2026

ComplexType: ContactStoryType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID>„SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <Contact>„ComplexType: ContactType“ </Contact> [1..1] </...>

ComplexType: ContactType

Abstract	no
XML Instance Representation	<...> <Name>„SimpleType: NotEmptyStringType“ </Name> [1..1] <EmailAddress>„SimpleType: EmailAddressType“ </EmailAddress> [0..1] <PhoneNumber>xs:string</PhoneNumber> [0..1] <MobilePhoneNumber>xs:string</MobilePhoneNumber> [0..1] </...>

ComplexType: DelegateType

Abstract	no
XML Instance Representation	<...> <Name>„ComplexType: CompanyNameType“ </Name> [1..1] <Owner>„ComplexType: CompanyOwnerType“ </Owner> [0..1] <Address>„ComplexType: AddressType“ </Address> [1..1] <UID-BFS>„SimpleType: UID-BFSType“ </UID-BFS> [1..1] </...>

ComplexType: IdentificationBaseType

Abstract	no
XML Instance Representation	<...> <InsuranceCompanyName>xs:string</InsuranceCompanyName> [1..1] <CustomerIdentity>„SimpleType: NotEmptyStringType“ </CustomerIdentity> [1..1] <ContractIdentity>„SimpleType: NotEmptyStringType“ </ContractIdentity> [1..1] </...>

ComplexType: InstitutionDescriptionBaseType

Abstract	no
XML Instance Representation	<...> <Name>„ComplexType: CompanyNameType“ </Name> [1..1] <Address>„ComplexType: AddressType“ </Address> [1..1] <UID-BFS>„ComplexType: UID-BFS-UnknownType“ </UID-BFS> [1..1] </...>

ComplexType: NameAndAddressType

Abstract	no
XML Instance Representation	<...> <Name>xs:string</Name> [1..1] <Address>„ComplexType: AddressType“ </Address> [0..1] </...>

ComplexType: ParticularsAddressType

Abstract	no
Parent type	„ComplexType: AddressType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: AddressType“] <ComplementaryLine>xs:string</ComplementaryLine> [0..1] <Street>xs:string</Street> [0..1] <Postbox>xs:string</Postbox> [0..1]

© Swissdec 2026

	<Locality>xs:string</Locality> [0..1] <ZIP-Code> „SimpleType: ZIP-CodeType“ </ZIP-Code> [1..1] <City>xs:string</City> [1..1] <Country>xs:string</Country> [0..1] [END BASE TYPE] <ResidenceCanton> „SimpleType: CantonAndEXType“ </ResidenceCanton> [1..1] <MunicipalityID> „SimpleType: MunicipalityIDType“ </MunicipalityID> [0..1] <DepartureDate>xs:date</DepartureDate> [0..1] </...>

ComplexType: ParticularsAddressesType

Abstract	no
XML Instance Representation	<...> <Address> „ComplexType: ParticularsAddressType“ </Address> [1..unbounded] </...>

ComplexType: ParticularsBaseType

Abstract	no
XML Instance Representation	<...> <Social-InsuranceIdentification> „ComplexType: Social-InsuranceIdentificationTy\ pe“ </Social-InsuranceIdentification> [1..1] <EmployeeNumber>xs:string</EmployeeNumber> [1..1] <Lastname>xs:string</Lastname> [1..1] <Firstname>xs:string</Firstname> [1..1] <Sex> „SimpleType: SexType“ </Sex> [1..1] <DateOfBirth>xs:date</DateOfBirth> [1..1] <Nationality> „SimpleType: NationalityType“ </Nationality> [1..1] <CivilStatus> „ComplexType: CivilStatusAndDateType“ </CivilStatus> [1..1] </...>

ComplexType: ParticularsType

Abstract	no
Parent type	„ComplexType: ParticularsBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: ParticularsBaseType“] <Social-InsuranceIdentification> „ComplexType: Social-InsuranceIdentificationTy\ pe“ </Social-InsuranceIdentification> [1..1] <EmployeeNumber>xs:string</EmployeeNumber> [1..1] <Lastname>xs:string</Lastname> [1..1] <Firstname>xs:string</Firstname> [1..1] <Sex> „SimpleType: SexType“ </Sex> [1..1] <DateOfBirth>xs:date</DateOfBirth> [1..1] <Nationality> „SimpleType: NationalityType“ </Nationality> [1..1] <CivilStatus> „ComplexType: CivilStatusAndDateType“ </CivilStatus> [1..1] [END BASE TYPE] <Addresses> „ComplexType: ParticularsAddressesType“ </Addresses> [1..1] <EmailAddress> „SimpleType: EmailAddressType“ </EmailAddress> [0..1] <PhoneNumber>xs:string</PhoneNumber> [0..1] <MobilePhoneNumber>xs:string</MobilePhoneNumber> [0..1] <ResidenceCategory> „SimpleType: ResidenceCategoryType“ </ResidenceCatego\ ry> [0..1] <LanguageCode> „SimpleType: LanguageCodeType“ </LanguageCode> [1..1] </...>

ComplexType: PersonBaseType

Abstract	no
XML Instance Representation	<...> <Particulars> „ComplexType: ParticularsType“ </Particulars> [1..1] <Work> „ComplexType: WorkType“ </Work> [1..unbounded] </...>

ComplexType: RegisterOrganizationAuthenticationAddresseeContextType

Abstract	no
Parent type	„ComplexType: AddresseeResponseContextType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: AddresseeResponseContextType“] [BASE TYPE „ComplexType: ResponseContextType“] <UserAgent> „ComplexType: UserAgentType“ </UserAgent> [1..1] <InstitutionName>xs:string</InstitutionName> [1..1]

© Swissdec 2026

	<TransmissionDate>xs:dateTime</TransmissionDate> [1..1] <ResponseID> „SimpleType: IDType“ </ResponseID> [1..1] <RequestID> „SimpleType: IDType“ </RequestID> [1..1] [END BASE TYPE] <ProducerResponseNotifications> „ComplexType: FeedbackNotificationsType“ </Produ\ cerResponseNotifications> [1..1] <Warning> „ComplexType: NotificationsType“ </Warning> [0..1] <Info> „ComplexType: NotificationsType“ </Info> [0..1] [END BASE TYPE] <CertificateRequestID> „SimpleType: IDType“ </CertificateRequestID> [1..1] <TestCase> „ComplexType: EmptyType“ </TestCase> [0..1] </...>

ComplexType: RegisterOrganizationAuthenticationAddresseeJobStateType

Abstract	no
Parent type	„ComplexType: AddresseeJobType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: AddresseeJobType“] [BASE TYPE „ComplexType: AddresseeType“] <AddresseeIdentification> „SimpleType: IDType“ </AddresseeIdentificati\ on> [1..1] [END BASE TYPE] <ProcessByDistributor> „SimpleType: SimpleBooleanType“ </ProcessByDistribu\ tor> [1..1] [END BASE TYPE] [START CHOICE] <Success> „ComplexType: RegisterOrganizationAuthenticationAddresseeSuccessJobStateType“ </ Success> [1..1] [END CHOICE] </...>

ComplexType: RegisterOrganizationAuthenticationAddresseeRequestCaseType

Abstract	no
XML Instance Representation	<...> <CaseContext> „ComplexType: RegisterOrganizationAuthenticationCaseContextType“ </ CaseContext> [1..1] <ReceivedState> „SimpleType: RegisterOrganizationAuthenticationStateType“ </Receive\ dState> [0..1] </...>

ComplexType: RegisterOrganizationAuthenticationAddresseeResponseCaseType

Abstract	no
XML Instance Representation	<...> <CaseContext> „ComplexType: RegisterOrganizationAuthenticationCaseContextType“ </ CaseContext> [1..1] <State> „SimpleType: RegisterOrganizationAuthenticationStateType“ </State> [1..1] <Success> „ComplexType: RegisterOrganizationAuthenticationConsumerSuccessType“ </ Success> [0..1] </...>

ComplexType: RegisterOrganizationAuthenticationAddresseeSuccessJobStateType

Abstract	no
XML Instance Representation	<...> <AddresseeContext> „ComplexType: RegisterOrganizationAuthenticationAddresseeContextType“ </ AddresseeContext> [1..1] <Credentials> „ComplexType: CredentialsType“ </Credentials> [1..1] </...>

ComplexType: RegisterOrganizationAuthenticationCaseContextType

Abstract	no
Parent type	„ComplexType: CaseContextType“
Translation	Fallkontext
Short description	Fallkontext
XML Instance Representation	<...> [BASE TYPE „ComplexType: CaseContextType“] [BASE TYPE „ComplexType: CaseContextBaseType“]

© Swissdec 2026

	<ReceivedStoryIDs> „ComplexType: ReceivedStoriesType“ </ReceivedSto\ ryIDs> [0..1] <SuppressedSenderStoryIDs> „ComplexType: ReceivedStoriesType“ </SuppressedSen\ derStoryIDs> [0..1] <SuppressedInstitutionStoryIDs> „ComplexType: ReceivedStoriesType“ </Suppresse\ dInstitutionStoryIDs> [0..1] <Warning> „ComplexType: StoryNotificationsType“ </Warning> [0..1] <Info> „ComplexType: StoryNotificationsType“ </Info> [0..1] [END BASE TYPE] <Credentials> „ComplexType: CredentialsType“ </Credentials> [1..1] [END BASE TYPE] <CertificateRequestID> „SimpleType: IDType“ </CertificateRequestID> [1..1] <TestCase> „ComplexType: EmptyType“ </TestCase> [0..1] </...>

ComplexType: RegisterOrganizationAuthenticationConsumerResponseType

Abstract	no
XML Instance Representation	<...> <RegisterOrganizationAuthenticationResponse> „ComplexType: RegisterOrganizationAuthenticationAddresse RegisterOrganizationAuthenticationResponse> [1..1] </...>

ComplexType: RegisterOrganizationAuthenticationConsumerSuccessType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <CompanyName> „ComplexType: CompanyNameType“ </CompanyName> [1..1] <CompanyAddress> „ComplexType: AddressType“ </CompanyAddress> [1..1] <CompanyUID-BFS> „SimpleType: UID-BFSType“ </CompanyUID-BFS> [1..1] <QualityLevel>xs:int</QualityLevel> [1..1] <InstitutionContactPerson> „ComplexType: ContactType“ </InstitutionContactPer\ son> [1..1] <Delegate> „ComplexType: DelegateType“ </Delegate> [0..unbounded] </...>

ComplexType: RegisterOrganizationAuthenticationJobStateType

Abstract	no
Translation	Job Status
Technical description	Antwort Struktur der Adressaten.
XML Instance Representation	<...> <Addressee> „ComplexType: RegisterOrganizationAuthenticationAddresseeJobStateType“ </ Addressee> [1..1] </...>

ComplexType: RegisterOrganizationAuthenticationQuittanceType

Abstract	no
Parent type	„ComplexType: StoryBaseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: StoryBaseType“] <Creation>xs:dateTime</Creation> [1..1] <StoryID> „SimpleType: IDType“ </StoryID> [1..1] [END BASE TYPE] <X509Subject>c:anonymous</X509Subject> [1..1] <CompanyUID-BFS> „SimpleType: UID-BFSType“ </CompanyUID-BFS> [1..1] <AuthorizedForUID-BFS> „SimpleType: UID-BFSType“ </AuthorizedForUID-BFS> [0..1] <Comment> „ComplexType: NotificationsType“ </Comment> [0..1] </...>

ComplexType: RegisterOrganizationAuthenticationResponseType

Abstract	no
Parent type	„ComplexType: ResponseType“

© Swissdec 2026

XML Instance Representation	<...> [BASE TYPE „ComplexType: ResponseType“] <ResponseContext> „ComplexType: ResponseContextType“ </ResponseContext> [1..1] [END BASE TYPE] <Addressees> „ComplexType: RegisterOrganizationAuthenticationJobStateType“ </Addres\ sees> [1..1] </...>

ComplexType: RegisterOrganizationAuthenticationSenderRequestCaseType

Abstract	no
Technical description	Fallcontext und Stories
XML Instance Representation	<...> <CaseContext> „ComplexType: RegisterOrganizationAuthenticationCaseContextType“ </ CaseContext> [1..1] <ReceivedState> „SimpleType: RegisterOrganizationAuthenticationStateType“ </Receive\ dState> [0..1] <SignCertificate> „ComplexType: CertificateSignRequestType“ </SignCertifica\ te> [0..1] <RenewCertificate> „ComplexType: CertificateSignRequestBaseType“ </RenewCertifica\ te> [0..1] </...>

ComplexType: RegisterOrganizationAuthenticationSenderResponseCaseType

Abstract	no
Translation	Fall
Short description	Informationen zum Fall
Technical description	Informationen zum Fall
XML Instance Representation	<...> <CaseContext> „ComplexType: RegisterOrganizationAuthenticationCaseContextType“ </ CaseContext> [1..1] <State> „SimpleType: RegisterOrganizationAuthenticationStateType“ </State> [1..1] <Quittance> „ComplexType: RegisterOrganizationAuthenticationQuittanceType“ </Quit\ tance> [0..1] <Certificate> „ComplexType: CertificateSignResponseType“ </Certificate> [0..1] </...>

ComplexType: Social-InsuranceIdentificationType

Abstract	no
XML Instance Representation	<...> [START CHOICE] <SV-AS-Number> „SimpleType: SV-AS-NumberType“ </SV-AS-Number> [1..1] <unknown> „ComplexType: EmptyType“ </unknown> [1..1] [END CHOICE] </...>

ComplexType: SynchronizeRegisterOrganizationAuthenticationAddresseeResponseType

Abstract	no
Parent type	„ComplexType: AddresseeResponseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: AddresseeResponseType“] <AddresseeContext> „ComplexType: AddresseeResponseContextType“ </AddresseeCon\ text> [1..1] [END BASE TYPE] <Addressee> „ComplexType: AddresseeType“ </Addressee> [1..1] <Case> „ComplexType: RegisterOrganizationAuthenticationAddresseeResponseCaseType“ </ Case> [1..1] </...>

ComplexType: SynchronizeRegisterOrganizationAuthenticationConsumerResponseType

Abstract	no

© Swissdec 2026

XML Instance Representation	<...> <SynchronizeRegisterOrganizationAuthentication> „ComplexType: SynchronizeRegisterOrganizationAuthentic SynchronizeRegisterOrganizationAuthentication> [1..1] </...>

ComplexType: SynchronizeRegisterOrganizationAuthenticationResponseType

Abstract	no
Parent type	„ComplexType: ResponseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: ResponseType“] <ResponseContext> „ComplexType: ResponseContextType“ </ResponseContext> [1..1] [END BASE TYPE] [START CHOICE] <Error> „ComplexType: ErrorResponseType“ </Error> [1..1] <SynchronizeRegisterOrganizationAuthenticationConsumer> „ComplexType: SynchronizeRegisterOrganizatio SynchronizeRegisterOrganizationAuthenticationConsumer> [1..1] [END CHOICE] </...>

ComplexType: SynchronizeRegisterOrganizationAuthenticationSenderType

Abstract	no
Parent type	„ComplexType: AddresseeResponseType“
XML Instance Representation	<...> [BASE TYPE „ComplexType: AddresseeResponseType“] <AddresseeContext> „ComplexType: AddresseeResponseContextType“ </AddresseeCon\ text> [1..1] [END BASE TYPE] <Addressee> „ComplexType: AddresseeType“ </Addressee> [1..1] <Case> „ComplexType: RegisterOrganizationAuthenticationSenderResponseCaseType“ </ Case> [1..1] </...>

ComplexType: TaxAtSourceCategoryType

Abstract	no
XML Instance Representation	<...> [START CHOICE] <TaxAtSourceCode> „SimpleType: TaxAtSourceCodeType“ </TaxAtSourceCode> [1..1] <CategoryPredefined> „SimpleType: CategoryPredefinedType“ </CategoryPredefi\ ned> [1..1] <CategoryOpen> „SimpleType: IDType“ </CategoryOpen> [1..1] [END CHOICE] </...>

ComplexType: WeeklyHoursAndLessonsType

Abstract	no
XML Instance Representation	<...> <WeeklyHours> „SimpleType: HoursOrLessonsType“ </WeeklyHours> [1..1] <WeeklyLessons> „SimpleType: HoursOrLessonsType“ </WeeklyLessons> [1..1] </...>

ComplexType: WorkType

Abstract	no
XML Instance Representation	<...> <WorkingTime>c:anonymous</WorkingTime> [1..1] <EntryDate>xs:date</EntryDate> [1..1] <WithdrawalDate>xs:date</WithdrawalDate> [0..1] </...>

ComplexType: WorkplaceType

Abstract	no
XML Instance Representation	<...>

© Swissdec 2026

	[START CHOICE] <BUR-REE-Number> „SimpleType: BUR-REE-NumberType“ </BUR-REE-Number> [1..1] <InHouseID> „SimpleType: IDType“ </InHouseID> [1..1] [END CHOICE] <AddressExtended> „ComplexType: AddressExtensionType“ </AddressExtended> [1..1] </...>

SimpleType: AssuranceCategoryCodeType

Translation	Versicherungskategorie-Code Typ
Short description	Versicherungskategorie-Code Typ
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Pattern: [A-Z0-9]{2}

SimpleType: BUR-REE-NumberType

Translation	BUR-Nummer Typ
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Pattern: [A-Z][0-9]{8}

SimpleType: CantonAddressType

Translation	KantonsTyp
Short description	KantonsTyp der Schweiz
Content type	Restriction
Base type	„SimpleType: CantonAndEXType“
Facets	• Enum: AG • Enum: AI • Enum: AR • Enum: BE • Enum: BL • Enum: BS • Enum: FR • Enum: GE • Enum: GL • Enum: GR • Enum: JU • Enum: LU • Enum: NE • Enum: NW • Enum: OW • Enum: SG • Enum: SH • Enum: SO • Enum: SZ • Enum: TG • Enum: TI • Enum: UR • Enum: VD • Enum: VS • Enum: ZG • Enum: ZH

© Swissdec 2026

SimpleType: CantonAndEXType

Translation	KantonsTyp (inkl. EX)
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: AG • Enum: AI • Enum: AR • Enum: BE • Enum: BL • Enum: BS • Enum: FR • Enum: GE • Enum: GL • Enum: GR • Enum: JU • Enum: LU • Enum: NE • Enum: NW • Enum: OW • Enum: SG • Enum: SH • Enum: SO • Enum: SZ • Enum: TG • Enum: TI • Enum: UR • Enum: VD • Enum: VS • Enum: ZG • Enum: ZH • Enum: EX

SimpleType: CategoryPredefinedType

Translation	Vordefinierte Kategorien
Short description	Mögliche Werte der vordefinierte Kategorien: - HEN, HEY: VR-Honorar an qsP mit Wohnsitz im Ausland - MEN, MEY: Leistungen aus exportierten Mitarbeiterbeteiligungen an qsP mit Wohnsitz im Ausland - NON, NOY: Nicht quellensteuerpflichtig - SFN: Sondervereinbarung mit Frankreich
Technical description	4 mögliche Werte der vordefinierten Kategorien.
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: HEN • Enum: HEY • Enum: MEN • Enum: MEY

© Swissdec 2026

	• Enum: NON • Enum: NOY • Enum: SFN

SimpleType: CivilStatusType

Translation	ZivilstandsTyp
Short description	ZivilstandsTyp
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: unknown • Enum: single • Enum: married • Enum: widowed • Enum: divorced • Enum: separated • Enum: registeredPartnership • Enum: partnershipDissolvedByLaw • Enum: partnershipDissolvedByDeath • Enum: partnershipDissolvedByDeclarationOfLost

SimpleType: EmploymentContractType

Translation	Arbeitsvertragsart
Short description	Arbeitsvertragsart ist als folgende mögliche Werte anzugeben: - indefiniteSalaryMth = unbefristeter Vertrag mit Monatslohn - indefiniteSalaryMthAWT = unbefristeter Vertrag mit Monatslohn und Jahresarbeitszeit - indefiniteSalaryHrs = unbefristeter Vertrag mit Stundenlohn - indefiniteSalaryNoTimeConstraint = unbefristeter Vertrag mit Provision, Pauschale, Akkordlohn, usw. - fixedSalaryMth = befristeter Vertrag mit Monatslohn - fixedSalaryHrs = befristeter Vertrag mit Stundenlohn - fixedSalaryNoTimeConstraint = befristeter Vertrag mit Provision, Pauschale, Akkordlohn, usw. - apprentice = Lehrvertrag - internshipContract = Praktikumsvertrag
Technical description	Arbeitsvertragsart ist aus 9 möglichen Werte anzugeben.
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: indefiniteSalaryMth • Enum: indefiniteSalaryMthAWT • Enum: indefiniteSalaryHrs

© Swissdec 2026

	• Enum: indefiniteSalaryNoTimeConstraint • Enum: fixedSalaryMth • Enum: fixedSalaryHrs • Enum: fixedSalaryNoTimeConstraint • Enum: apprentice • Enum: internshipContract

SimpleType: HoursOrLessonsType

Translation	Stunden oder Lektionen
Content type	Restriction
Base type	xs:decimal
Facets	• Pattern: [0-9]+\.[0-9]{2}

SimpleType: MunicipalityIDType

Translation	Gemeindenummer
Content type	Restriction
Base type	xs:int
Facets	• MinInclusive: 1 • MaxInclusive: 9999 • TotalDigits: 4

SimpleType: NationalityType

Translation	Staatsangehörigkeit
Short description	Die Codes für eine korrekte Zuteilung der Staatsangehörigkeit sind bei der UNO hinterlegt (ISO 3166)
Technical description	Erweiterung 11= unbekannt 22= staatenlos: Die ISO-Codes wurden auf dem Prod-Distri noch nie geprüft: In der Qualitätsstufe Plausibilität kann evtl. gegen die ISO 3166 Codes geprüft und eine Warnung angezeigt werden. Achtung: Probleme mit bestehenden Datenbeständen (Änderungen der Ländernamen und Verwechslungen)
Content type	Restriction
Base type	xs:string
Facets	• Pattern: [A-Z][A-Z]|11|22

SimpleType: PositionType

Translation	Berufliche Stellung
Short description	Berufliche Stellung des Arbeitnehmers: Mögliche Werte: highestCadre: Oberes Kader; middleCadre: Mittleres Kader; lowerCadre: Unteres Kader; lowestCadre: Verantwortlich für die Ausführung der Arbeiten;

© Swissdec 2026

	noCadre:ohne Kaderfunktion;
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: highestCadre • Enum: middleCadre • Enum: lowerCadre • Enum: lowestCadre • Enum: noCadre

SimpleType: RegisterOrganizationAuthenticationStateType

Translation	Status des Falls
Short description	Status des Falls
Content type	Restriction
Base type	xs:string
Facets	• Enum: processing • Enum: registered • Enum: rejected • Enum: verified • Enum: expired

SimpleType: ResidenceCategoryType

Translation	Aufenthaltskategorien
Short description	Mögliche Werte der Aufenthaltskategorien für Ausländer: shortTerm-L = Kurzaufenthalter (L) annual-B = Jahresaufenthalter (B) settled-C = Niedergelassene (C) crossBorder-G = Grenzgänger (G) asylumSeeker-N = Asylsuchender (N) needForProtection-S = Schutzbedürftige (S) NotificationProcedureForShorttermWork90Days = Meldeverfahren für kurzfristige Erwerbstätigkeit (bis 90 Tage, Schengenregelung) NotificationProcedureForShorttermWork120Days = Meldeverfahren für kurzfristige Erwerbstätigkeit bis 120 Tage ProvisionallyAdmittedForeigners (F) = Vorläufig aufgenommene Ausländer ResidentForeignNationalWithGainfulEmployment (Ci) = Aufenthaltsbewilligung mit Erwerbstätigkeit othersNotSwiss = Andere (nicht Schweizer)

© Swissdec 2026

Technical description	Es gibt 11 mögliche Werte der Aufenthaltskategorien für Ausländer.
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: shortTerm-L • Enum: annual-B • Enum: settled-C • Enum: crossBorder-G • Enum: asylumSeeker-N • Enum: needForProtection-S • Enum: NotificationProcedureForShorttermWork90Days • Enum: NotificationProcedureForShorttermWork120Days • Enum: ProvisionallyAdmittedForeigners-F • Enum: ResidentForeignNationalWithGainfulEmployment-Ci • Enum: othersNotSwiss

SimpleType: SV-AS-NumberType

Translation	13-stellige SV-Nr Typ
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Pattern: [0-9]{3}\.[0-9]{4}\.[0-9]{4}\.[0-9]{2} • MaxLength: 16

SimpleType: SalaryAmountAbsoluteType

Translation	Absolut-LohnbetragsTyp
Short description	Absolut-LohnbetragsTyp ohne Vorzeichen
Content type	Restriction
Base type	„SimpleType: SalaryAmountType“
Facets	• Pattern: [0-9]+\.[0-9]{2}

SimpleType: SexType

Translation	Geschlecht
Short description	F weiblich; M männlich
Technical description	In diesem Element wird das Geschlecht der Person angegeben.
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: F • Enum: M

SimpleType: TaxAtSourceCodeType

Translation	QST-Code
Short description	QST-Code
Technical description	QST-Code
Content type	Restriction
Base type	xs:NMTOKEN

© Swissdec 2026

Facets	• Pattern: [A-Z][0-9][Y,N]

SimpleType: ZIP-CodeType

Translation	PLZ Typ
Short description	Postleitzahl
Content type	Restriction
Base type	xs:string
Facets	• MinLength: 1 • MaxLength: 15

© Swissdec 2026