Technische Spezifikation KLE 2.0.1

Version 20260306

Anforderungen Transmitter

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 9
5. CheckInteroperability 12
6. RegisterOrganizationAuthentication 16
6.1. Übersicht 16
6.2. Initiale synchrone Meldung 16
6.3. Datenstruktur der initialen Meldung 17
6.4. Datenstruktur der initialen Antwort 19
6.5. Datenstruktur der Synchronize-Meldung 20
6.6. Datenstruktur der Synchronize-Antwort 23
7. SubscribeOrganization 26
7.1. Übersicht 26
7.2. Initiale synchrone Meldung 26
7.3. Datenstruktur der initialen Meldung 28
7.4. Datenstruktur der initialen Antwort 30
7.5. Datenstruktur der Synchronize-Meldung 31
7.6. Datenstruktur der Synchronize-Antwort 34
8. DeclareIncident 37
8.1. KLE Ereignismeldung 37
8.2. Initiale synchrone Meldung 37
8.3. Ablauf / Protokoll 38
8.4. Datenstruktur der initialen Meldung 40
8.5. Datenstruktur der initialen Antwort 42
8.6. Datenstruktur der Synchronize-Meldung 43
8.7. Datenstruktur der Synchronize-Antwort 47
9. Use Cases 50
9.1. UC001 Initiale Meldung senden 50
9.2. UC002 Status abholen 52
9.3. UC003 Testmeldung kennzeichnen 54
9.4. UC004 Ersatzmeldung kennzeichnen 55
9.5. UC005 Synchronisieren 56
9.6. UC006 Prozesskontrolle durchführen 58
9.7. UC007 Fall schliessen 59
9.8. UC008 Stories melden und quittieren 59
9.9. UC009 Stories abholen und verarbeiten 60
9.10. UC010 Datenflusskontrolle durchführen 61
9.11. UC011 Completion aufrufen 62
9.12. UC012 Dialog Message abwickeln 63
9.13. UC013 Anmeldungsprozess (SubscribeOrganization) 63
9.14. UC014 Unternehmensauthentifizierung (RegisterOrganization) 64
9.15. UC015 Security anwenden 65
9.16. UC016 Transmitter konfigurieren 65
9.17. UC017 Supportinformationen anzeigen 65
9.18. UC018 Erreichbarkeit prüfen 66
9.19. UC019 Interoperabilität prüfen 67
A. Referenzierte Dokumente 70
B. Glossar 71

© Swissdec 2026

C. Detailspezifikation Swissdec Unternehmens-Authentifizierung SUA 73
C.1. Einleitung 73
C.2. SUA Prozess 73
C.3. Technische Anforderungen an die Zertifkate 75
D. Detailspezifikation DialogMessages 78
D.1. Einleitung 78
D.2. Die Verwendung von DialogMessages 78
D.3. Standard DialogMessages 84
E. Detailspezifikation Completion URL 85
E.1. Aufbau und Verarbeitung der Completion URL 85
F. Bausteine 88
F.1. AB-01 Konfiguration 88
F.2. AB-02 Erreichbarkeit 88
F.3. AB-03 Interoperabilität 89
F.4. AB-04 Anmeldung 89
F.5. AB-05 Explizite Adressierung (direkte Adressierung) 90
F.6. AB-06 Implizite Adressierung (Gruppenadressierung) 91
F.7. AB-07 Absichtserklärung 91
F.8. AB-08 Identifikation eines Geschäftsvorfalls 92
F.9. AB-09 Ersatzmeldung 92
F.10. AB-10 Request und Response IDs 93
F.11. AB-11 Testmeldung 93
F.12. AB-12 Plausibilisierung 94
F.13. AB-13 Filterung 95
F.14. AB-14 Versionsmapping 96
F.15. AB-15 Doublettenerkennung 96
F.16. AB-16 Anonymisierung 97
F.17. AB-17 Meldungsfragmentierung 97
F.18. AB-18 Swissdec Unternehmensauthentifizierung (SUA) 97
F.19. AB-19 Archivdateien 98
F.20. SB-01 Transportsicherheit (HTTPS / TLS) 98
F.21. SB-02 Transport Authentication (TLS Mutual Authentication) 99
F.22. SB-03 Nutzdatenverschlüsselung (WS-Encryption) 99
F.23. SB-04 Nutzdatensignatur (WS-Signature) 99
F.24. SB-05 Unternehmens-Authentifizierung SUA Signatur (WS-Signatur) 100
F.25. SB-06 Nichtabstreitbarkeit 100
F.26. PB-01 Synchrone Verteilung 101
F.27. PB-02 Asynchrone Verteilung 101
F.28. PB-03 Synchronisierung 102
F.29. PB-04 Dialogisierung 102
F.30. PB-05 Completion 103
F.31. PB-06 Prozessnavigation 103
F.32. PB-07 Digitalisierungsbereich 104
F.33. PB-08 Daten anfordern 104
G. Technische Dokumentationen 106
G.1. Schema Dokumentation IncidentDeclarationServiceTypes.xsd 106
G.2. Schema Dokumentation IncidentDeclarationContainer.xsd 106
G.3. Schema Dokumentation IncidentDeclaration.xsd 113
G.4. Schema Dokumentation SwissdecComponents.xsd 136
G.5. Schema Dokumentation Common.xsd 151

© 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 7
4.1. Schemabild Ping 9
4.2. Schemabild PingResponse 10
5.1. Schemabild CheckInteroperability 12
5.2. Schemabild CheckInteroperabilityResponse 14
6.1. Synchroner Meldungsprozess 16
6.2. Schemabild RegisterOrganizationAuthentication 17
6.3. Schemabild Addressee 18
6.4. Schemabild RegisterOrganizationAuthenticationResponse 19
6.5. Schemabild SynchronizeRegisterOrganizationAuthentication 20
6.6. Schemabild CaseContext 22
6.7. Schemabild SynchronizeRegisterOrganizationAuthenticationResponse 23
7.1. Synchroner Meldungsprozess 27
7.2. Schemabild SubscribeOrganization 28
7.3. Schemabild Addressee 29
7.4. Schemabild SubscribeOrganizationResponse 30
7.5. Schemabild SynchronizeSubscribeOrganization 31
7.6. Schemabild CaseContext 33
7.7. Schemabild SynchronizeSubscribeOrganizationResponse 34
8.1. Synchroner Meldungsprozess 38
8.2. DeclareIncident Protokollstatus 39
8.3. Schemabild DeclareIncident 40
8.4. Schemabild Addressee 41
8.5. Schemabild DeclareIncidentResponse 42
8.6. Schemabild SynchronizeDeclareIncident 43
8.7. Schemabild CaseContext 46
8.8. Schemabild SynchronizeDeclareIncidentResponse 47
D.1. DialogMessage XML-Schema Struktur 79
D.2. Vorgehen bei der Darstellung der DialogMessage (Aktivitäts Diagramm) 82
E.1. Aufbau URL gemäss RFC3986 87
F.1. AddresseeType 90

© 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 Ping 9
4.2. Feldbeschreibungen UserAgent 10
4.3. Feldbeschreibungen PingResponse 10
4.4. Feldbeschreibungen UserAgent 11
5.1. Feldbeschreibungen CheckInteroperability 12
5.2. Feldbeschreibungen UserAgent 13
5.3. Feldbeschreibungen CheckInteroperabilityResponse 14
5.4. Feldbeschreibungen UserAgent 15
6.1. Feldbeschreibungen RegisterOrganizationAuthentication 17
6.2. Feldbeschreibungen RequestContext 17
6.3. Feldbeschreibungen Job 18
6.4. Feldbeschreibungen Addressee 18
6.5. Feldbeschreibungen RegisterOrganizationAuthenticationResponse 19
6.6. Feldbeschreibungen ResponseContext 19
6.7. Feldbeschreibungen SynchronizeRegisterOrganizationAuthentication 20
6.8. Feldbeschreibungen RequestContext 20
6.9. Feldbeschreibungen Sender 21
6.10. Feldbeschreibungen Addressee 21
6.11. Feldbeschreibungen Case 21
6.12. Feldbeschreibungen CaseContext 22
6.13. Feldbeschreibungen SynchronizeRegisterOrganizationAuthenticationResponse 23
6.14. Feldbeschreibungen ResponseContext 23
6.15. Feldbeschreibungen Error 24
6.16. Feldbeschreibungen SynchronizeRegisterOrganizationAuthenticationConsumer 24
6.17. Feldbeschreibungen AddresseeContext 24
6.18. Feldbeschreibungen Addressee 25
6.19. Feldbeschreibungen Case 25
7.1. Feldbeschreibungen SubscribeOrganization 28
7.2. Feldbeschreibungen RequestContext 28
7.3. Feldbeschreibungen Job 29
7.4. Feldbeschreibungen Addressee 29
7.5. Feldbeschreibungen SubscribeOrganizationResponse 30
7.6. Feldbeschreibungen ResponseContext 30
7.7. Feldbeschreibungen SynchronizeSubscribeOrganization 31
7.8. Feldbeschreibungen RequestContext 32
7.9. Feldbeschreibungen Sender 32
7.10. Feldbeschreibungen Addressee 32
7.11. Feldbeschreibungen Case 32
7.12. Feldbeschreibungen CaseContext 33
7.13. Feldbeschreibungen SynchronizeSubscribeOrganizationResponse 34
7.14. Feldbeschreibungen ResponseContext 34
7.15. Feldbeschreibungen Error 35
7.16. Feldbeschreibungen SynchronizeSubscribeOrganizationConsumer 35
7.17. Feldbeschreibungen AddresseeContext 35
7.18. Feldbeschreibungen Addressee 36
7.19. Feldbeschreibungen Case 36
8.1. Zustandstabelle 39
8.2. Feldbeschreibungen DeclareIncident 40
8.3. Feldbeschreibungen RequestContext 41
8.4. Feldbeschreibungen Job 41
8.5. Feldbeschreibungen Addressee 41

© Swissdec 2026

8.6. Feldbeschreibungen DeclareIncidentResponse 42
8.7. Feldbeschreibungen ResponseContext 42
8.8. Feldbeschreibungen SynchronizeDeclareIncident 43
8.9. Feldbeschreibungen RequestContext 43
8.10. Feldbeschreibungen Sender 44
8.11. Feldbeschreibungen Addressee 44
8.12. Feldbeschreibungen Case 44
8.13. Feldbeschreibungen CaseContext 46
8.14. Feldbeschreibungen SynchronizeDeclareIncidentResponse 47
8.15. Feldbeschreibungen ResponseContext 47
8.16. Feldbeschreibungen Error 48
8.17. Feldbeschreibungen SynchronizeDeclareIncidentConsumer 48
8.18. Feldbeschreibungen AddresseeContext 48
8.19. Feldbeschreibungen Addressee 49
8.20. Feldbeschreibungen Case 49
9.1. UC001 Initiale Meldung senden 50
9.2. RequestContextType 51
9.3. UserAgentType 51
9.4. UC002 Status abholen 52
9.5. UC003 Testmeldung kennzeichnen 54
9.6. UC003 Testmeldung kennzeichnen 55
9.7. UC005 Synchronisieren 56
9.8. UC006 Prozesskontrolle durchführen 58
9.9. UC007 Fall schliessen 59
9.10. UC008 Stories melden und quittieren 59
9.11. UC009 Stories abholen und verarbeiten 60
9.12. UC011 Completion aufrufen 62
9.13. UC012 Dialog Message abwickeln 63
9.14. UC013 Anmeldungsprozess (SubscribeOrganization) 63
9.15. UC017 Supportinformationen anzeigen 65
9.16. UC018 Erreichbarkeit prüfen 66
9.17. UC019 Interoperabilität prüfen 67
9.18. Vorbedingungen (Transmitter) 68
9.19. Auswertung und Antwort Distributor 69
9.20. Auswertung Transmitter 69
F.1. AB-01 Konfiguration 88
F.2. AB-02 Erreichbarkeit 88
F.3. AB-03 Interoperabilität 89
F.4. AB-04 Anmeldung 89
F.5. AB-05 Explizite Adressierung (direkte Adressierung) 90
F.6. AB-06 Implizite Adressierung (Gruppenadressierung) 91
F.7. AB-07 Absichtserklärung 91
F.8. AB-08 Identifikation eines Geschäftsvorfalls 92
F.9. AB-09 Ersatzmeldung 92
F.10. AB-10 Request und Response IDs 93
F.11. AB-11 Testmeldung 93
F.12. AB-12 Datenqualitätssicherung 94
F.13. AB-13 Filterung 95
F.14. AB-14 Versionsmapping 96
F.15. AB-15 Doublettenerkennung 96
F.16. AB-16 Anonymisierung 97
F.17. AB-18 Swissdec Unternehmensauthentifizierung (SUA) 97
F.18. AB-19 Archivdateien 98
F.19. SB-01 Transportsicherheit (HTTPS / TLS) 98
F.20. SB-02 Transport Authentication (TLS Mutual Authentication) 99
F.21. SB-03 Nutzdatenverschlüsselung (WS-Encryption) 99

© Swissdec 2026

F.22. SB-04 Nutzdatensignatur (WS-Signature) 99
F.23. SB-05 Unternehmens-Authentifizierung SUA Signatur (WS-Signatur) 100
F.24. SB-06 Nichtabstreitbarkeit 100
F.25. PB-01 Synchrone Verteilung 101
F.26. PB-02 Asynchrone Verteilung 101
F.27. PB-03 Synchronisierung 102
F.28. PB-04 Dialogisierung 102
F.29. PB-05 Completion 103
F.30. PB-06 Prozessnavigation 103
F.31. PB-07 Digitalisierungsbereich 104
F.32. PB-08 Daten anfordern 104

© Swissdec 2026

1 Einleitung

Dieses Dokument richtet sich an Software-Hersteller und -Entwickler, die für die technische Umsetzung eines Swissdec-Standards verantwortlich sind. Es enthält funktionale, technische und zusätzliche Anforderungen an Transmitter, die an den Swissdec-Distributor übermitteln wollen. Ein Transmitter wird dazu verwendet, Meldungen aus einem Sendersystem via Distributor an einen oder mehrere Endreceiver elektronisch zu versenden, damit es im entsprechenden Empfängersystem verarbeitet werden kann.

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

Sendersystem	Beim Sendersystem handelt es sich um ein System, welches 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, welches 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 Transmitter und Distributor, nicht aber jene zwischen Distributor und Endreceiver.

Der Transmitter wird im Rahmen der Swissdec-Zertifizierung auf seine Funktionalität geprüft. Die Testfälle zur Zertifizierung finden sich unter [[TFBASIS]]

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 Bschreibungen der Bausteine, aus denen die Use Cases 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.

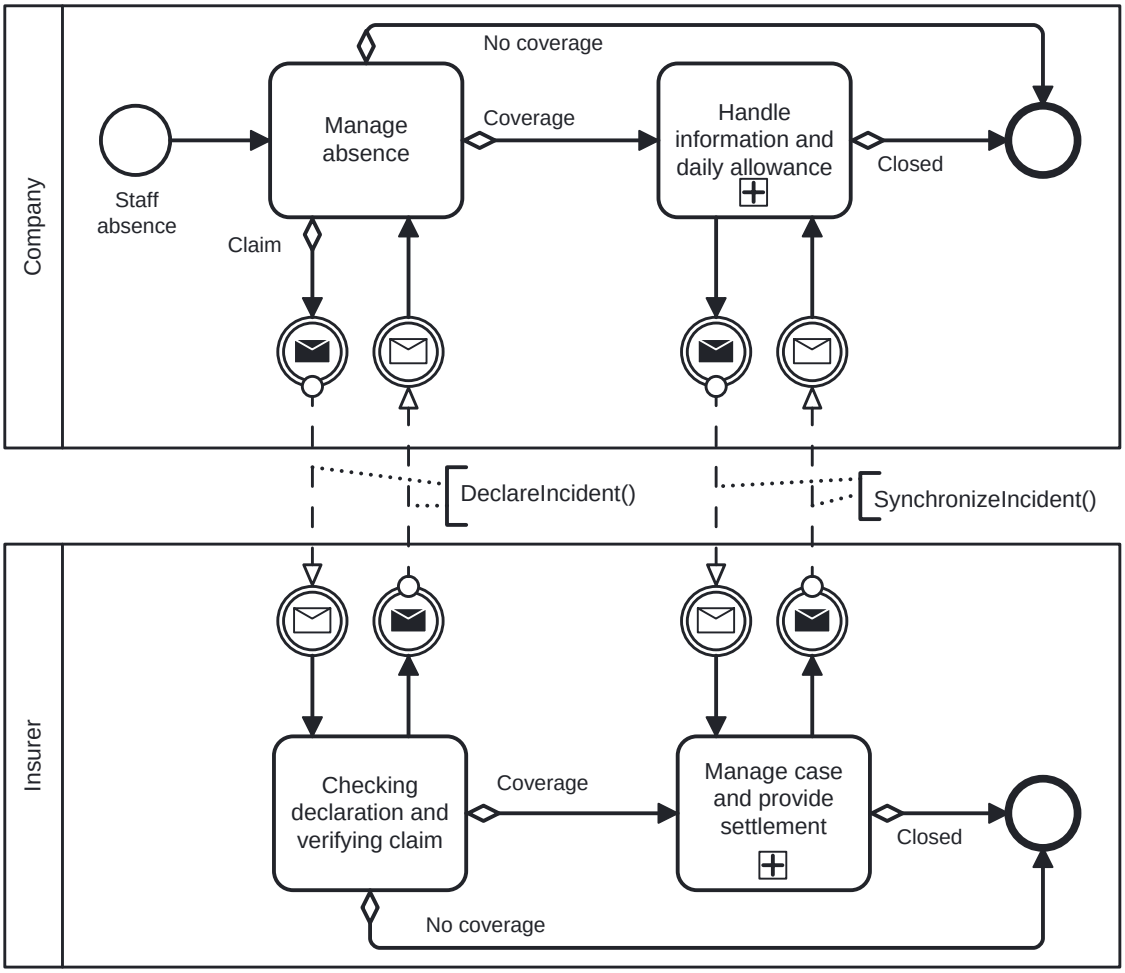


Abbildung 2.1. Prozessübersicht Swissdec KLE-Standard

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 9.1, „UC001 Initiale Meldung senden“). Die Meldung kann dabei als Testfall (Abschnitt 9.3, „UC003 Testmeldung kennzeichnen“) gekennzeichnet werden. Für eine erfolgreiche Übermittlung müssen Request und Response den Security-Richtlinien entsprechen (Abschnitt 9.15, „UC015 Security anwenden“), was im Falle der Response vom Transmitter verifiziert werden **muss**.

Die Use Cases Abschnitt 9.2, „UC002 Status abholen“ und Abschnitt 9.4, „UC004 Ersatzmeldung kennzeichnen“ werden von KLE nicht unterstützt.

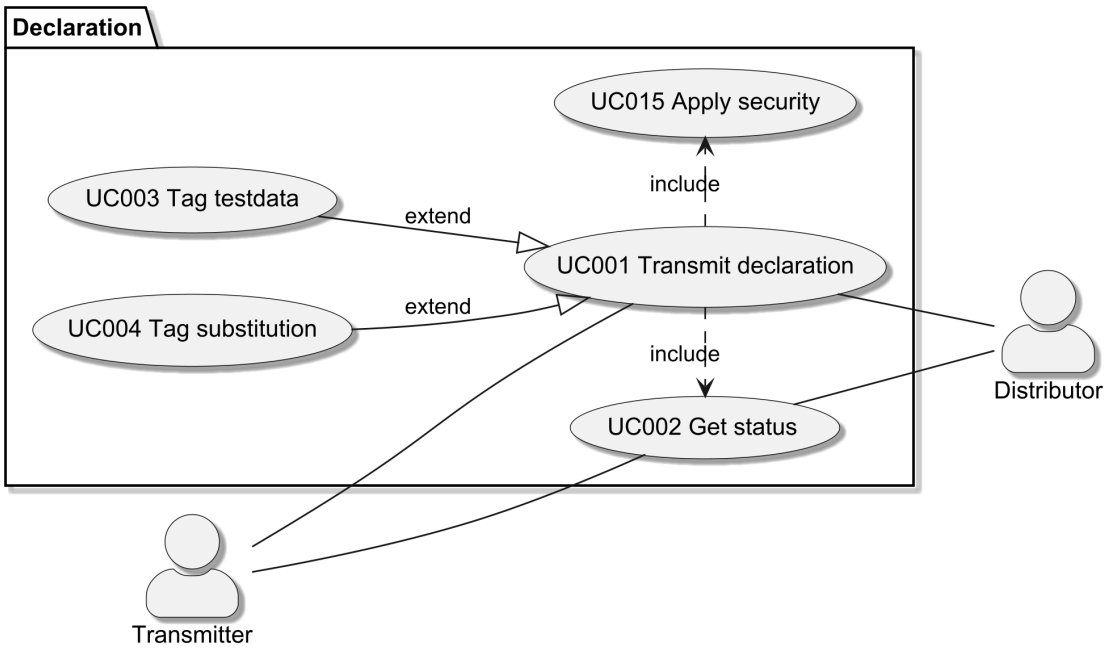


Abbildung 3.1. Use Cases Initiale Meldung

Nach erfolgreicher Deklaration werden Synchronisierungen (Abschnitt 9.5, „UC005 Synchronisieren“) durchgeführt, währenddessen werden Stories gemeldet und quittiert (Abschnitt 9.8, „UC008 Stories melden und quittieren“), aber auch beim Empfänger abgeholt und verarbeitet (Abschnitt 9.9, „UC009 Stories abholen und verarbeiten“). Dies soll anhand der Prozesskontrolle (Abschnitt 9.6, „UC006 Prozesskontrolle durchführen“) funktionieren, wobei Abschnitt 9.12, „UC012 Dialog Message abwickeln“ auch Teil davon ist. Der Geschäftsfall wird solange synchronisert bis der Fall mit dem State *CommunicationStopped* abgschlossen wird (Abschnitt 9.7, „UC007 Fall schliessen“).

© Swissdec 2026

Um eine Überlastung der betroffenen Systeme zu verhindern, können alle beteiligten Systeme den Datenfluss kontrollieren (Abschnitt 9.10, „UC010 Datenflusskontrolle durchführen“). Für eine erfolgreiche Synchronisierung müssen Request und Response den Security-Richtlinien entsprechen (Abschnitt 9.15, „UC015 Security anwenden“), was im Falle der Response vom Transmitter verifiziert werden **muss**.

Der Use Case Abschnitt 9.11, „UC011 Completion aufrufen“ wird von KLE nicht unterstützt.

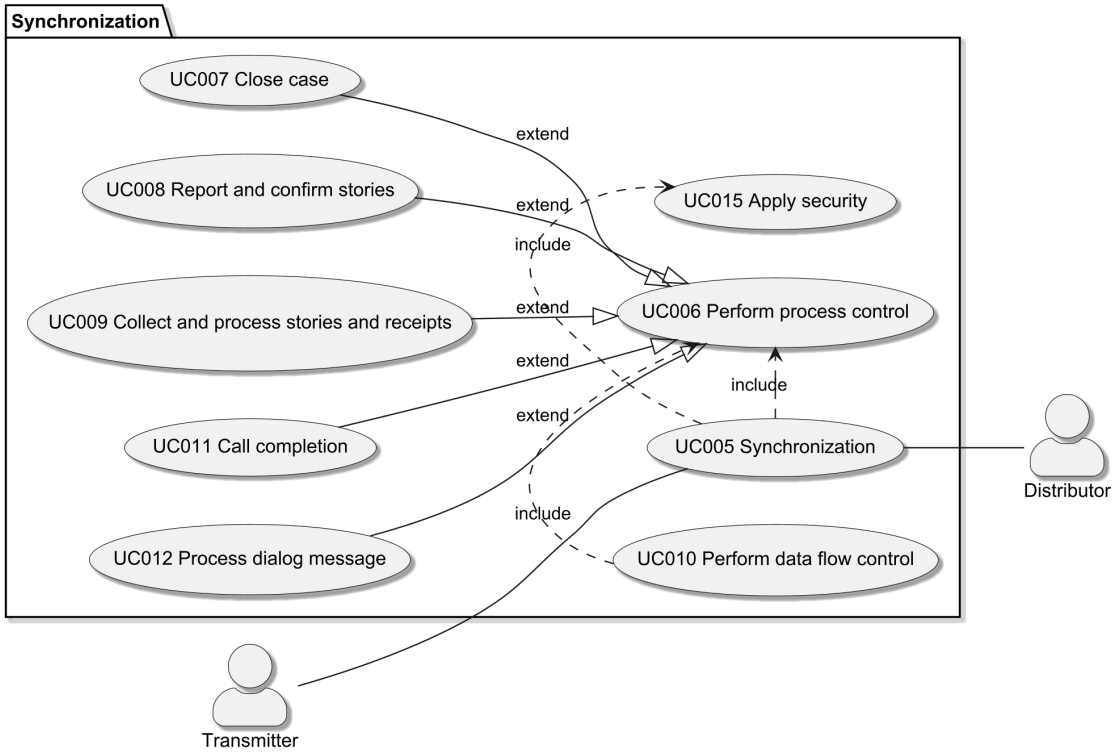


Abbildung 3.2. Use Cases Synchronisierung

Weitere Use Cases umfassen das vorgängige Konfigurieren eines Transmitters für die Übermittlung, d.h. das Einpflegen der aktuell gültigen Versicherungsprofile (Abschnitt 9.16, „UC016 Transmitter konfigurieren“) und die Darstellung von Supportinformationen (Abschnitt 9.17, „UC017 Supportinformationen anzeigen“) im Fehlerfall.

Des Weiteren gibt es einen Erreichbarkeits- (Abschnitt 9.18, „UC018 Erreichbarkeit prüfen“) und einen Interoperabilitätstest (Abschnitt 9.19, „UC019 Interoperabilität prüfen“), die bei der Erstinstallation des ERP sowie im Supportfall aufgerufen werden.

© Swissdec 2026

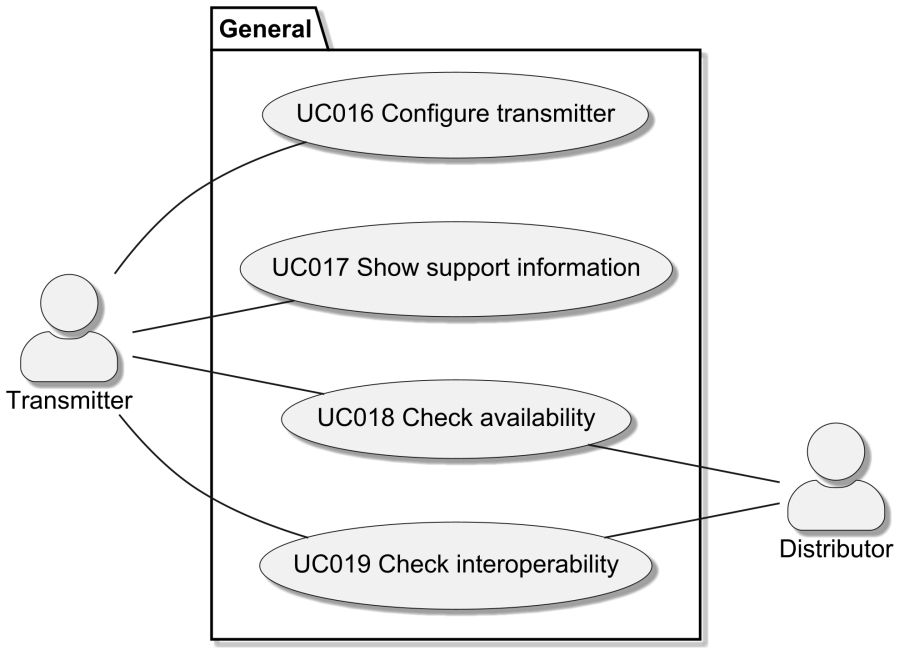


Abbildung 3.3. Andere Use Cases

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	• DeclareIncident: Operation zur Registrierung eines Ereignis (Unfall oder Krankheit) • DeclareIncidentResponse: Rückmeldung zur Registrierung eines Ereignisn • IncidentDeclarationFault: Rückmeldung im Fehlerfall
UC005 Synchronisieren	• SynchronizeDeclareIncident: Aktualisierung des Ereignises • SynchronizeDeclareIncidentResponse: Rückmeldung der Aktualisierungsanfrage • IncidentDeclarationFault: Rückmeldung im Fehlerfall
UC013 Anmeldungsprozess	• SubscribeOrganization: Anmeldungsprozess • SubscribeOrganizationResponse: Rückmeldung zum Anmeldungsprozess • SynchronizeSubscribeOrganization: Aktualisierung des Anmeldungsprozesses • SynchronizeSubscribeOrganizationResponse: Rückmeldung der Aktualisierungsanfrage • IncidentDeclarationFault: Rückmeldung im Fehlerfall
UC014 Unternehmensauthentifizierung (SUA)	• RegisterOrganizationAuthentication: SUA-Unternehmensauthentifizierung

© Swissdec 2026

Use Case	Operation / Element
	• RegisterOrganizationAuthenticationResponse: Rückmeldung zur SUA-Unternehmensauthentifizierung • SynchronizeRegisterOrganizationAuthentication: Aktualisierung des SUA-Prozesses • SynchronizeRegisterOrganizationAuthenticationResponse: Rückmeldung der Aktualisierungsanfrage • IncidentDeclarationFault: Rückmeldung im Fehlerfall
UC015 Erreichbarkeit prüfen	• Ping • PingResponse
UC016 Interoperabilität prüfen	• CheckInteroperability • CheckInteroperabilityResponse

Tabelle 3.1. Use Cases und Operationen

© Swissdec 2026

4 Ping

Die Erreichbarkeit des Distributors **muss** geprüft werden. Dazu wird eine einfache Anfrage an den Distributor geschickt. Die Antwort des Distributors bestätigt die Erreichbarkeit.

Mit dem Ping-Aufruf wird vom Transmitter wie auch vom Distributor die Systemzeit übermittelt, sodass es möglich ist, die Zeiten der an der Übermittlung beteiligten Systeme zu vergleichen. Da der Ping weder signiert noch verschlüsselt ist, ist es mit diesem Zeitvergleich möglich, Timestamp-Probleme aufzudecken.

Dieser Use Case dient der Qualitätssicherung bei der Installation und im Supportfall. Er **muss** stets manuell ausgeführt werden und **darf nicht** automatisiert und nicht in regelmässigen Abständen aufgerufen werden. Der Ping *kann* vom Endbenutzer ausgelöst werden. Er **muss** im Supportfall von der supportverantwortlichen Person aufgerufen werden können. Dies *kann* auch über eine, für den Endbenutzer nicht sichtbare, Administrationsoberfläche geschehen.

Die Anforderungen sind im Abschnitt 9.18, „UC018 Erreichbarkeit prüfen“ beschrieben.

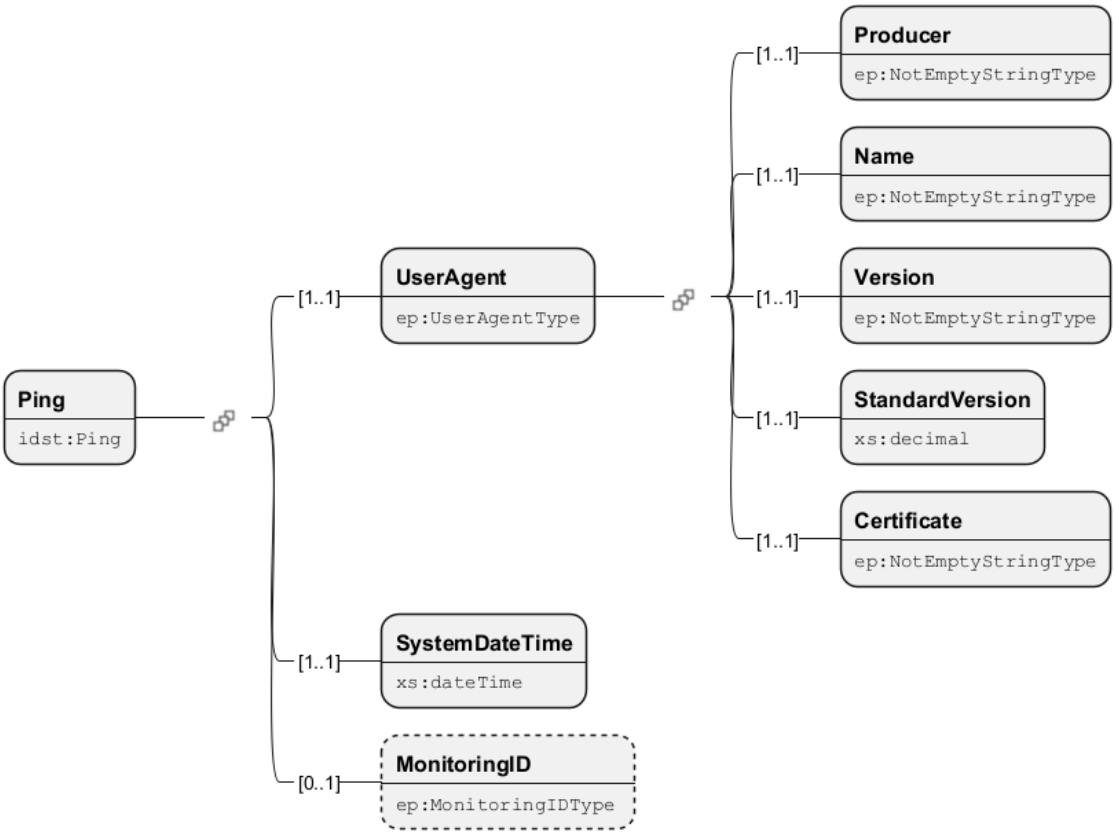


Abbildung 4.1. Schemabild Ping

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType
SystemDateTime	Aktuelle Systemzeit	xs dateTime
MonitoringID	Die KontrollID wird vor allem in der Referenzapplikation zur Aufteilung der Daten verwendet	ep MonitoringIDType

© Swissdec 2026

Feldname	Beschreibung	Typ

Tabelle 4.1. Feldbeschreibungen Ping

Feldname	Beschreibung	Typ
Producer	Hersteller der Applikation	ep NotEmptyString-Type
Name	Name des Produktes + eventuelle Zusatzinformationen (Kauf-transmitter, etc.)	ep NotEmptyString-Type
Version	Produkt-Version	ep NotEmptyString-Type
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 NotEmptyString-Type

Tabelle 4.2. Feldbeschreibungen UserAgent

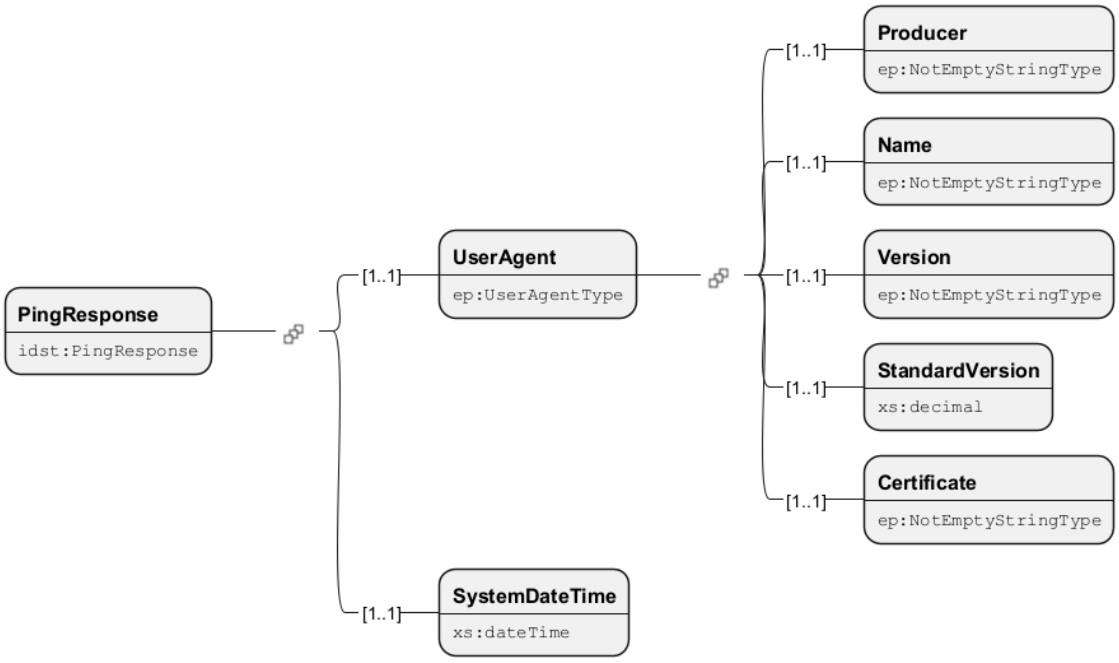


Abbildung 4.2. Schemabild PingResponse

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType

© Swissdec 2026

Feldname	Beschreibung	Typ
SystemDateTime	Aktuelle Systemzeit	xs dateTime

Tabelle 4.3. Feldbeschreibungen PingResponse

Feldname	Beschreibung	Typ
Producer	Hersteller der Applikation	ep NotEmptyString-Type
Name	Name des Produktes + eventuelle Zusatzinformationen (Kauf-transmitter, etc.)	ep NotEmptyString-Type
Version	Produkt-Version	ep NotEmptyString-Type
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 NotEmptyString-Type

Tabelle 4.4. Feldbeschreibungen UserAgent

© Swissdec 2026

5 CheckInteroperability

Eine spezielle Meldung wird an den Distributor gesendet, um die Interoperabilität (z.B. Encoding, Marshalling, Zeitangaben etc.) zwischen Transmitter und Distributor zu prüfen.

Damit die Interoperabilität zwischen einem Transmitter und dem Distributor überprüft werden kann, **muss** der Transmitter einen „CheckInteroperabilityRequest“ absetzen können. Dieser **muss** manuell ausgeführt werden.

Der Interoperabilitätstest wird zu Entwicklungszwecken, bei der Installation und im Supportfall verwendet, um die Interoperabilität zwischen einem Transmitter und dem Distributor zu gewährleisten. Die grössten zu erwartenden Schwierigkeiten liegen dabei in den Bereichen der Codierung von Zeichenketten (Encoding) und der Interpretation von Fliesskommazahlen. Beide Systeme (Transmitter und Distributor) **müssen** dabei bestimmte Auswertungen vornehmen, um bei einem eventuellen Fehler auf den Verursacher schliessen zu können.

Der Interoperabilitätstest **darf nicht** im Rahmen eines automatisierten Prozesses routinemässig aufgerufen werden.

Die Anforderungen sind im Abschnitt 9.19, „UC019 Interoperabilität prüfen“ beschrieben.

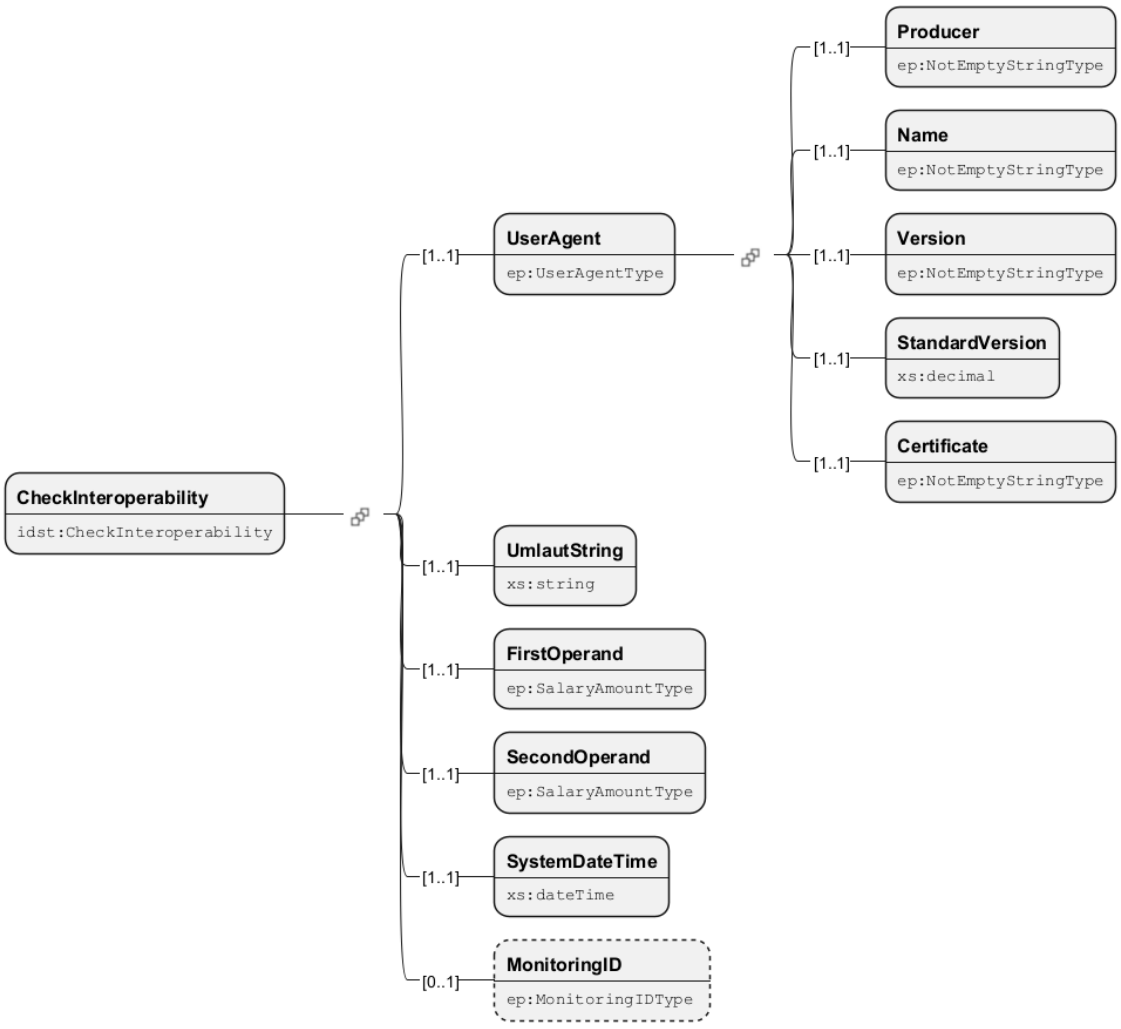


Abbildung 5.1. Schemabild CheckInteroperability

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType

© Swissdec 2026

Feldname	Beschreibung	Typ
UmlautString	Anordnung von Umlauten zum Testen des Encodings bei Sonderzeichen.	xs string
FirstOperand	1. Operand wird mit 2. Operand zusammengerechnet zum Testen, dass keine Datentyp-Fehler passieren.	ep SalaryAmountType
SecondOperand	2. Operand wird mit 1. Operand zusammengerechnet zum Testen, dass keine Datentyp-Fehler passieren.	ep SalaryAmountType
SystemDateTime	Aktuelle Systemzeit	xs dateTime
MonitoringID	Die KontrollID wird vor allem in der Referenzapplikation zur Aufteilung der Daten verwendet	ep MonitoringIDType

Tabelle 5.1. Feldbeschreibungen CheckInteroperability

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 5.2. Feldbeschreibungen UserAgent

© Swissdec 2026

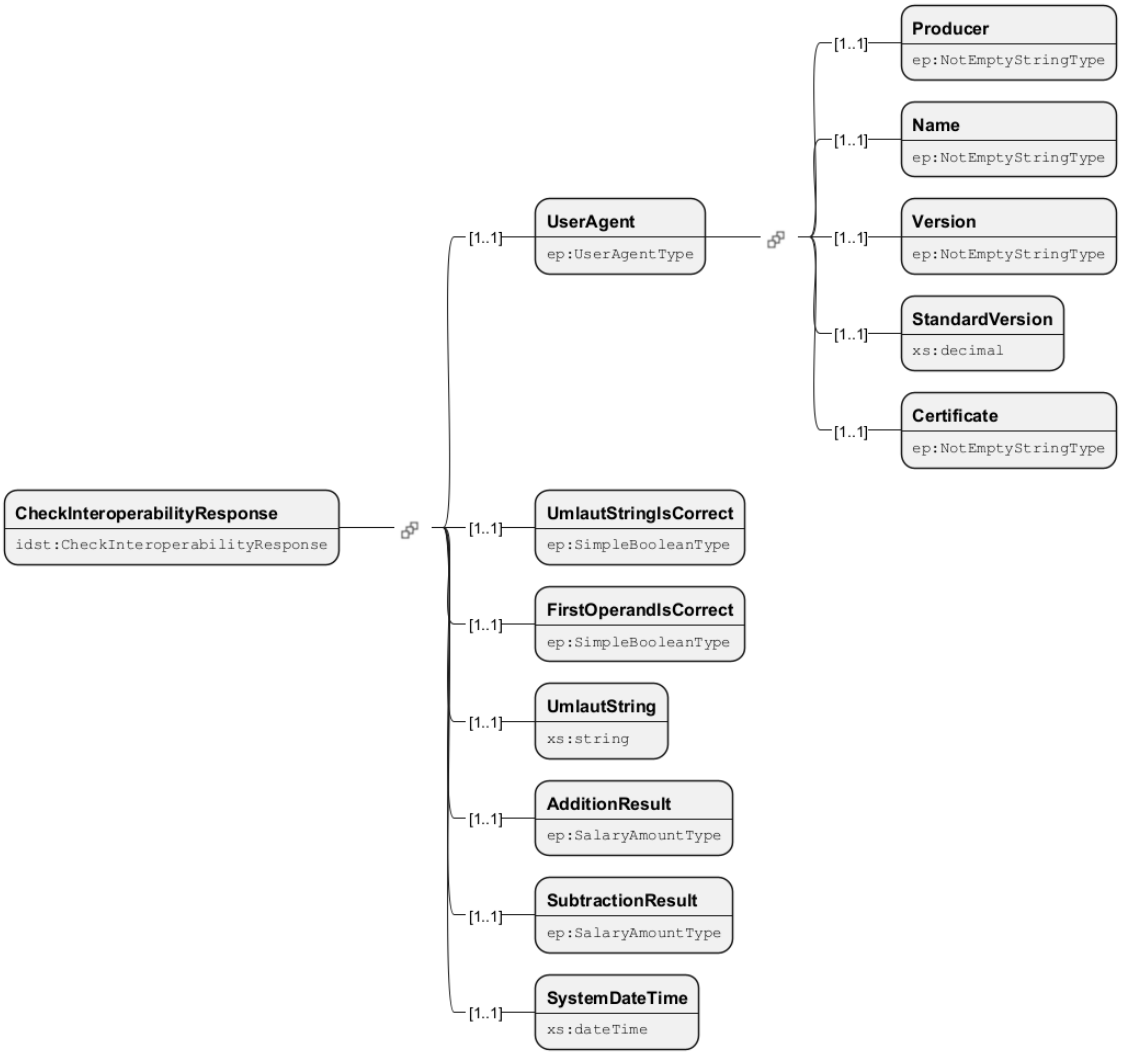


Abbildung 5.2. Schemabild CheckInteroperabilityResponse

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType
UmlautStringIsCorrect	Ergebnis der Prüfung des UmlautStrings	ep SimpleBooleanType
FirstOperandIsCorrect	Ergebnis des Vergleichs der Operanden	ep SimpleBooleanType
UmlautString	Anordnung von Umlauten zum Testen des Encodings bei Sonderzeichen.	xs string
AdditionResult	Summe der beiden Operanden des Interoperabilitätstests	ep SalaryAmountType
SubtractionResult	Ergebnis des Substraktionstests in der Interoperabilitätsprüfung	ep SalaryAmountType
SystemDateTime	Aktuelle Systemzeit	xs dateTime

© Swissdec 2026

Feldname	Beschreibung	Typ

Tabelle 5.3. Feldbeschreibungen CheckInteroperabilityResponse

Feldname	Beschreibung	Typ
Producer	Hersteller der Applikation	ep NotEmptyString-Type
Name	Name des Produktes + eventuelle Zusatzinformationen (Kauf-transmitter, etc.)	ep NotEmptyString-Type
Version	Produkt-Version	ep NotEmptyString-Type
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 NotEmptyString-Type

Tabelle 5.4. Feldbeschreibungen UserAgent

© Swissdec 2026

6 RegisterOrganizationAuthentication

6.1 Übersicht

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

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

Der Registrierung bleibt längerfristig als offener Fall bestehen, da diese für die Erneuerung der SUA Zertifikate verwendet wird.

6.2 Initiale synchrone Meldung

Die Aktivitätsdiagramme sind ohne den Endreceiver dargestellt. Die wesentliche Transmitterfunktionalität *soll* dargestellt werden. Die Anzeigen erfolgen wahlweise in Transmitter oder im ERP.

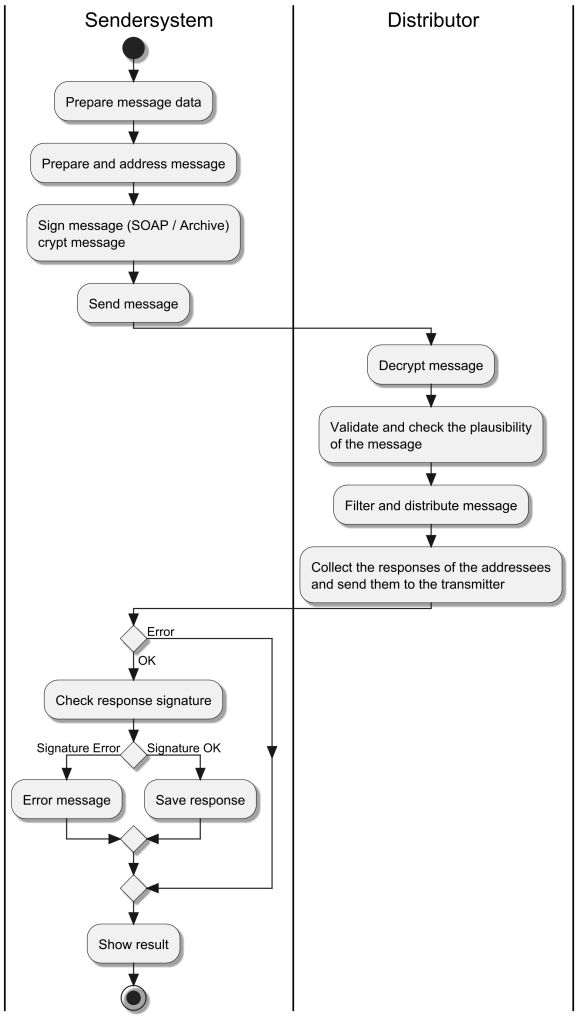


Abbildung 6.1. Synchroner Meldungsprozess

© Swissdec 2026

6.3 Datenstruktur der initialen Meldung

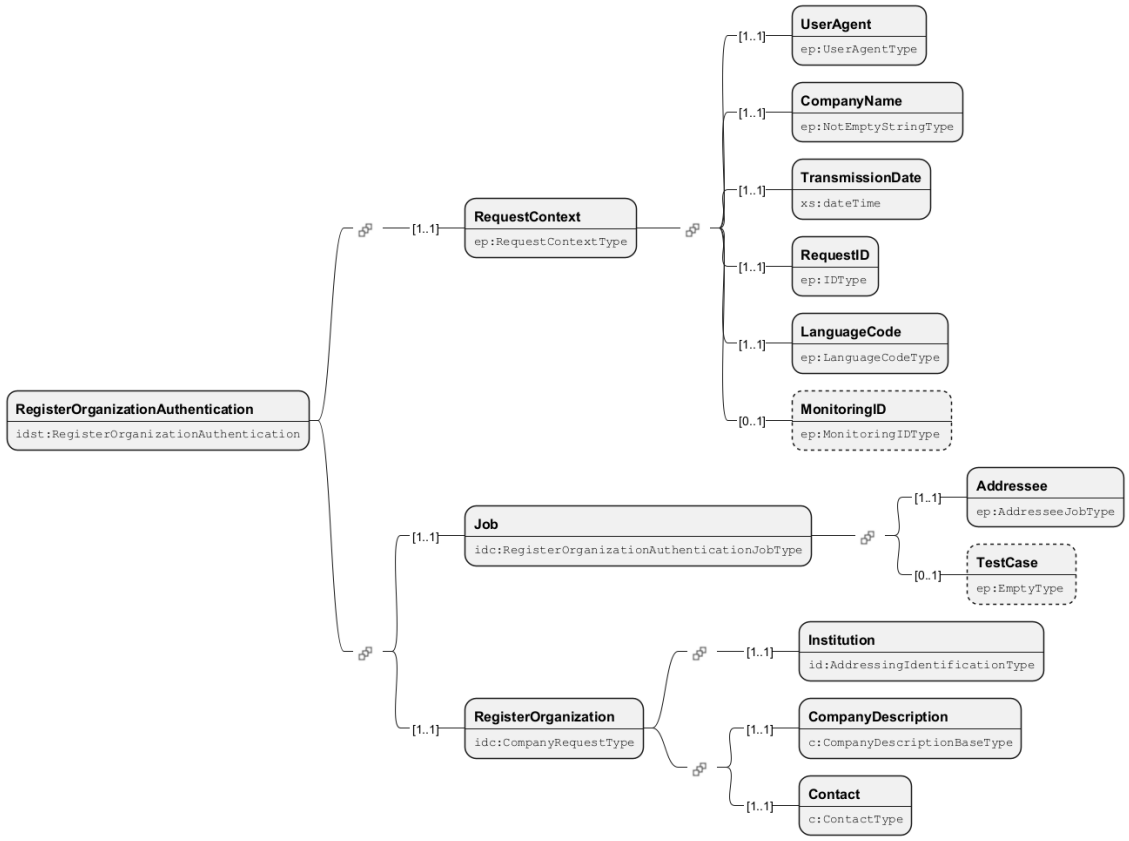


Abbildung 6.2. Schemabild RegisterOrganizationAuthentication

Feldname	Beschreibung	Typ
RequestContext	Der RequestContext enthält Informationen zum Zeitpunkt der Übermittlung, dem Absender und der Sprache.	ep RequestContext-Type
Job	Steuerungselement für die Verarbeitung der Meldung.	idc RegisterOrganizationAuthenticationJobType
RegisterOrganization	Registrierungsanfrage zum Lösen eines SUA-Zertifikats.	idc CompanyRequestType
@schemaVersion	Angabe der verwendeten Minor-Version des Schemas.	id SupportedMinorSchemaVersionAttributeType

Tabelle 6.1. Feldbeschreibungen RegisterOrganizationAuthentication

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType
CompanyName	Beschreibung der wesentlichen Unternehmensdaten	ep NotEmptyStringType

© Swissdec 2026

Feldname	Beschreibung	Typ
TransmissionDate	Übertragungszeitpunkt	xs dateTime
RequestID	AuftragsID	ep IDType
LanguageCode	Sprachcode	ep LanguageCodeType
MonitoringID	Monitoring Identifikation	ep MonitoringIDType

Tabelle 6.2. Feldbeschreibungen RequestContext

Feldname	Beschreibung	Typ
Addressee	Auflistung der Endempfänger, die die übermittelten Daten erhalten sollen.	ep AddresseeJobType
TestCase	Testmeldung kennzeichnen. Mit diesem Flag darf die Meldung nicht produktiv verarbeitet werden.	ep EmptyType

Tabelle 6.3. Feldbeschreibungen Job

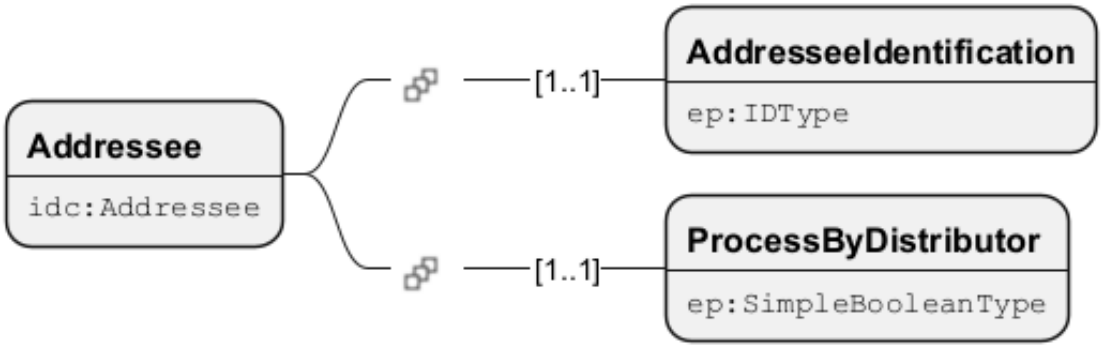


Abbildung 6.3. Schemabild Addressee

Feldname	Beschreibung	Typ
AddresseeIdentification	Identifikation des Adressaten	ep IDType
ProcessByDistributor	Dieses Flag entscheidet, ob an den Distributor übermittelte Daten von ihm verarbeitet oder ignoriert werden sollen. Steht 'false', wird der Distributor keine Verteilung vornehmen und die gesendeten Inhalte für diese Institution ignorieren.	ep SimpleBooleanType
@addresseeID	Referenz auf Adressat	ep InstanceRefIDType

Tabelle 6.4. Feldbeschreibungen Addressee

© Swissdec 2026

6.4 Datenstruktur der initialen Antwort

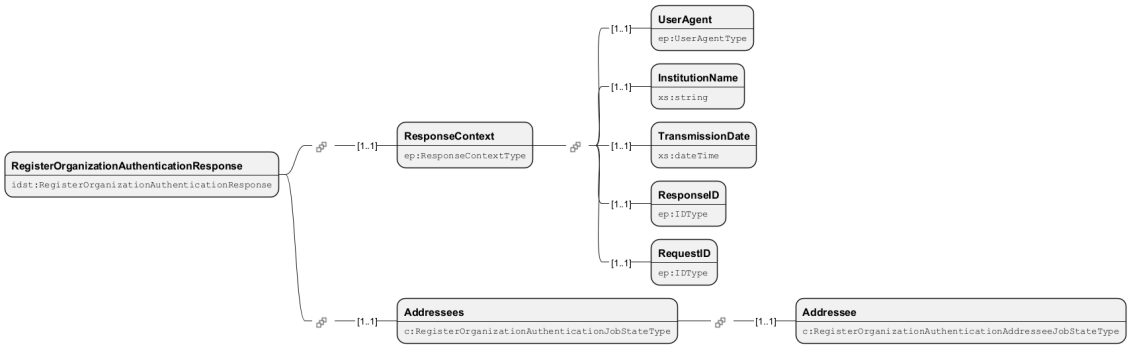


Abbildung 6.4. Schemabild RegisterOrganizationAuthenticationResponse

Feldname	Beschreibung	Typ
ResponseContext	Der ResponseContext enthält Informationen zum Zeitpunkt der Übermittlung, dem Absender und der Sprache.	ep ResponseContextType
Addressees	Auswahl der Endempfänger und Markierung als Testfall	c RegisterOrganizationAuthenticationJobStateType

Tabelle 6.5. Feldbeschreibungen RegisterOrganizationAuthenticationResponse

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType
InstitutionName	Institutionsname	xs string
TransmissionDate	Übertragungszeitpunkt	xs dateTime
ResponseID	AntwortID	ep IDType
RequestID	AuftragsID	ep IDType

Tabelle 6.6. Feldbeschreibungen ResponseContext

© Swissdec 2026

6.5 Datenstruktur der Synchronize-Meldung

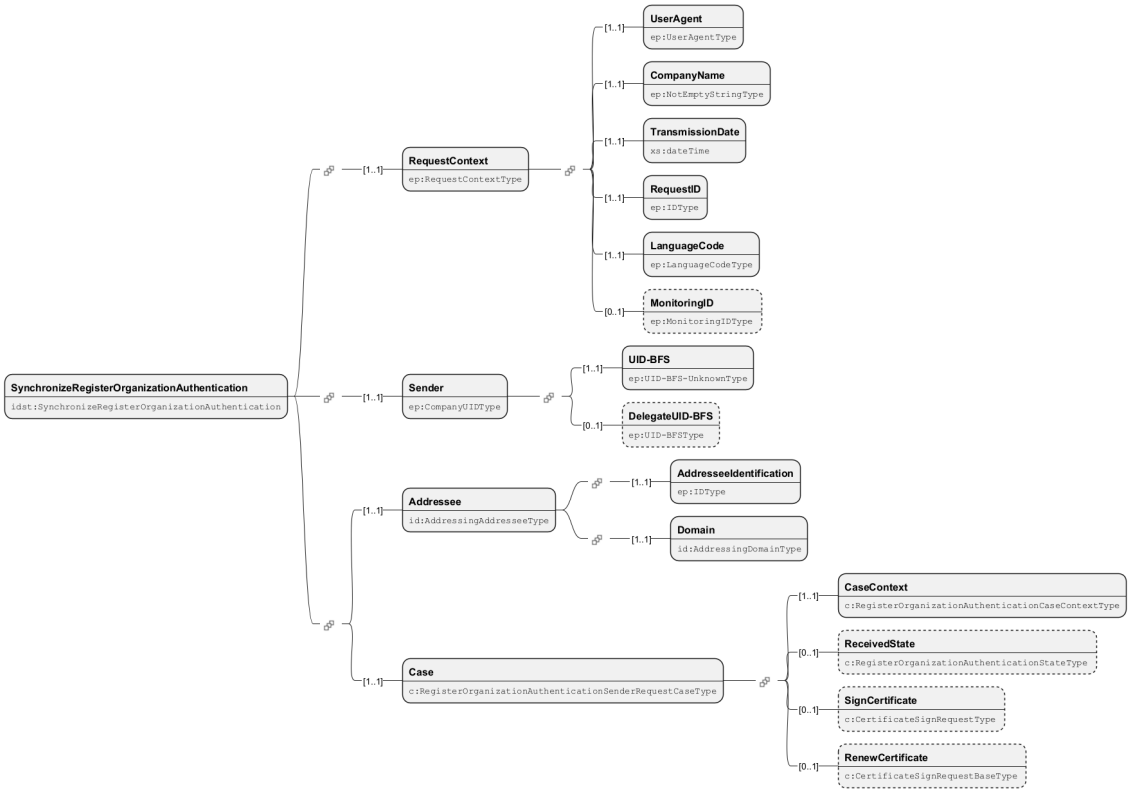


Abbildung 6.5. Schemabild SynchronizeRegisterOrganizationAuthentication

Feldname	Beschreibung	Typ
RequestContext	Der RequestContext enthält Informationen zum Zeitpunkt der Übermittlung, dem Absender und der Sprache.	ep RequestContext-Type
Sender	Absender der übermittelten Informationen	ep CompanyUIDTy-pe
Addressee	Informationen zum Adressaten	id AddressingAd-dresseeType
Case	Fallcontext und Stories	c RegisterOrganizat-ionAuthenticati-onSenderRequest-CaseType
@schemaVersion	Angabe der verwendeten Minor-Version des Schemas.	id SupportedMi-norSchemaVer-sionAttributeType

Tabelle 6.7. Feldbeschreibungen SynchronizeRegisterOrganizationAuthentication

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType

© Swissdec 2026

Feldname	Beschreibung	Typ
CompanyName	Beschreibung der wesentlichen Unternehmensdaten	ep NotEmptyStringType
TransmissionDate	Übertragungszeitpunkt	xs dateTime
RequestID	AuftragsID	ep IDType
LanguageCode	Sprachcode	ep LanguageCodeType
MonitoringID	Monitoring Identifikation	ep MonitoringIDType

Tabelle 6.8. Feldbeschreibungen RequestContext

Feldname	Beschreibung	Typ
UID-BFS	Unternehmens-Identifikationsnummer / UID-BFS	ep UID-BFS-UnknownType
DelegateUID-BFS	Unternehmens UID-BFS Typ	ep UID-BFSType

Tabelle 6.9. Feldbeschreibungen Sender

Feldname	Beschreibung	Typ
AddresseeIdentification	Identifikation des Adressaten	ep IDType
Domain	Domäne	id AddressingDomainType

Tabelle 6.10. Feldbeschreibungen Addressee

Feldname	Beschreibung	Typ
CaseContext	Dieses Element enthält Informationen und Identifikatoren zum Fall.	c RegisterOrganizationAuthenticationCaseContextType
ReceivedState	Der Status lautet: Übermittlung empfangen	c RegisterOrganizationAuthenticationStateType
SignCertificate	Anfrage zur Signierung eines Zertifikats	c CertificateSignRequestType
RenewCertificate	Anfrage zur Erneuerung eines SUA-Zertifikats	c CertificateSignRequestBaseType

Tabelle 6.11. Feldbeschreibungen Case

© Swissdec 2026

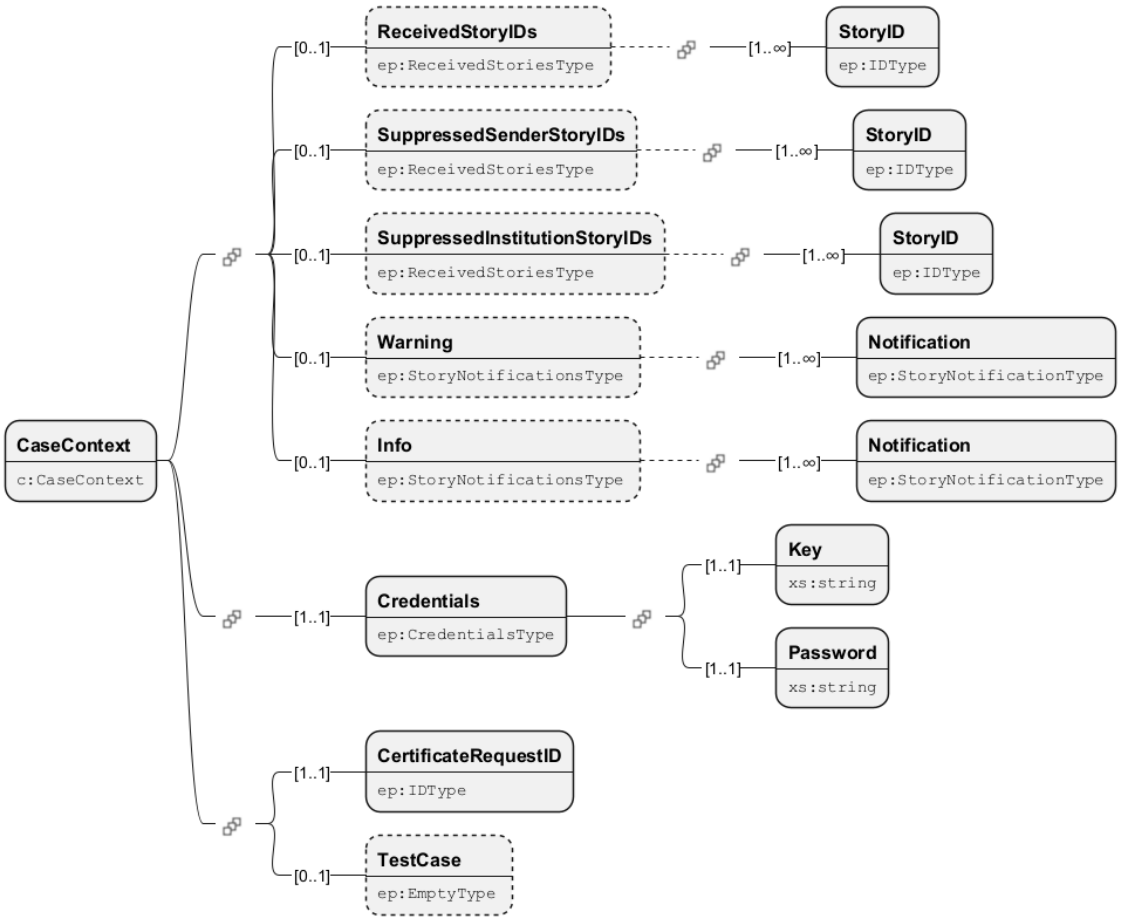


Abbildung 6.6. Schemabild CaseContext

Feldname	Beschreibung	Typ
ReceivedStoryIDs	Die Empfangenen StoryIDs als Quittungsliste.	ep ReceivedStories-Type
SuppressedSen-derStoryIDs	Die StoryIDs des Senders, die vom Distributor unterdrückt wur-den. Das kann bei Mapping-Problemen geschehen.	ep ReceivedStories-Type
SuppressedInstituti-onStoryIDs	Die StoryIDs des Empfängers, die vom Distributor unterdrückt wurden. Das kann bei Mapping-Problemen geschehen.	ep ReceivedStories-Type
Warning	Warnungen, die an den Producer / Transmitter zurückgeschickt werden.	ep StoryNotificati-onsType
Info	Informationen, die an den Producer / Transmitter zurückgeschickt werden.	ep StoryNotificati-onsType
Credentials	Berechtigungsnachweis	ep CredentialsType
CertificateReques-tID	ID zur Anforderung des Zertifikats	ep IDType
TestCase	Testfall	ep EmptyType

© Swissdec 2026

Feldname	Beschreibung	Typ

Tabelle 6.12. Feldbeschreibungen CaseContext

6.6 Datenstruktur der Synchronize-Antwort

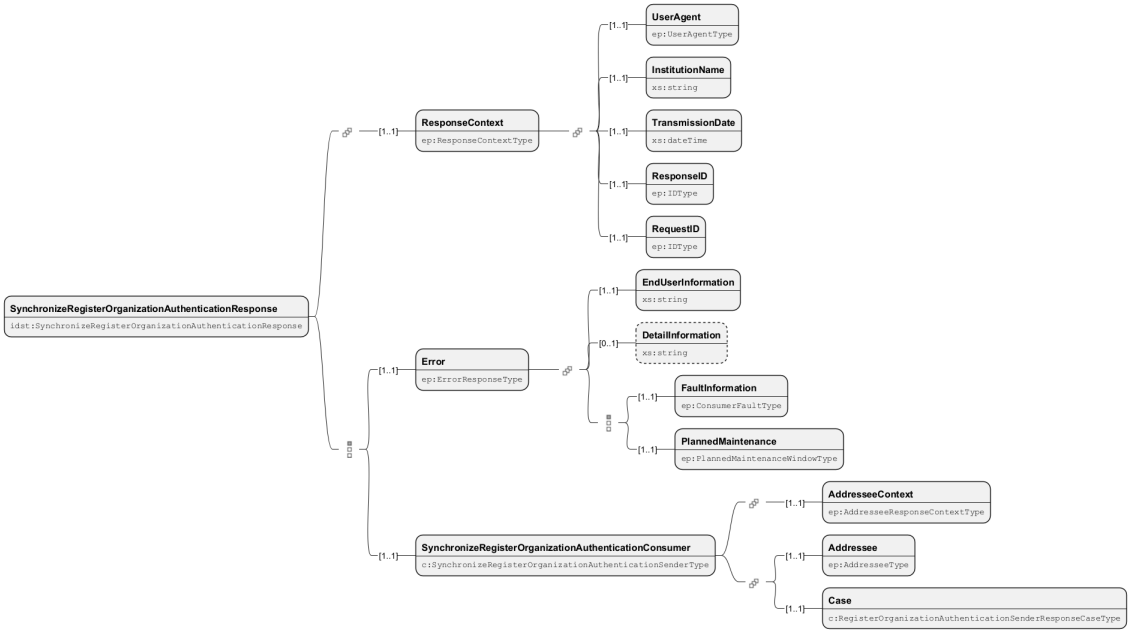


Abbildung 6.7. Schemabild SynchronizeRegisterOrganizationAuthenticationResponse

Feldname	Beschreibung	Typ
ResponseContext	Der ResponseContext enthält Informationen zum Zeitpunkt der Übermittlung, dem Absender und der Sprache.	ep ResponseContextType
Error	Die Verarbeitung war nicht erfolgreich und es sind Fehler aufgetreten.	ep ErrorResponseType
SynchronizeRegisterOrganizationAuthenticationConsumer	Organisationsregristration und -authentifizierung	c SynchronizeRegisterOrganizationAuthenticationSenderType

Tabelle 6.13. Feldbeschreibungen SynchronizeRegisterOrganizationAuthenticationResponse

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType
InstitutionName	Institutionsname	xs string
TransmissionDate	Übertragungszeitpunkt	xs dateTime

© Swissdec 2026

Feldname	Beschreibung	Typ
ResponseID	AntwortID	ep IDType
RequestID	AuftragsID	ep IDType

Tabelle 6.14. Feldbeschreibungen ResponseContext

Feldname	Beschreibung	Typ
EndUserInformation	Informationen zum Endnutzer	xs string
DetailInformation	Informationen im Detail	xs string
FaultInformation	Informationen zum Fehler	ep ConsumerFault-Type
PlannedMaintenan-ce	Geplante Wartung	ep PlannedMainten-anceWindowType

Tabelle 6.15. Feldbeschreibungen Error

Feldname	Beschreibung	Typ
AddresseeContext	Kontextangaben zum Adressaten	ep AddresseeRes-ponseContextType
Addressee	Angaben zum Adressaten	ep AddresseeType
Case	Informationen zum Fall	c RegisterOrganizat-ionAuthenticati-onSenderRespon-seCaseType

Tabelle 6.16. Feldbeschreibungen SynchronizeRegisterOrganizationAuthenticationConsumer

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType
InstitutionName	Institutionsname	xs string
TransmissionDate	Übertragungszeitpunkt	xs dateTime
ResponseID	AntwortID	ep IDType
RequestID	AuftragsID	ep IDType
ProducerResponse-Notifications	Antwort vom Producer zurückgegeben	ep FeedbackNotifi-cationsType
Warning	Warnung wurde zurückgegeben	ep NotificationsType
Info	Informationen wurden zurückgegeben	ep NotificationsType

© Swissdec 2026

Feldname	Beschreibung	Typ

Tabelle 6.17. Feldbeschreibungen AddresseeContext

Feldname	Beschreibung	Typ
AddresseeIdentification	Identifikation des Adressaten	ep IDType

Tabelle 6.18. Feldbeschreibungen Addressee

Feldname	Beschreibung	Typ
CaseContext	Fallkontext	c RegisterOrganizationAuthenticationCaseContextType
State	Status des Falls	c RegisterOrganizationAuthenticationStateType
Quittance	Quittierung der Regristrationsmeldung	c RegisterOrganizationAuthenticationQuittanceType
Certificate	Zertifikat	c CertificateSignResponseType

Tabelle 6.19. Feldbeschreibungen Case

© Swissdec 2026

7 SubscribeOrganization

7.1 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.

7.2 Initiale synchrone Meldung

Die Aktivitätsdiagramme sind ohne den Endreceiver dargestellt. Die wesentliche Transmitterfunktionalität *soll* dargestellt werden. Die Anzeigen erfolgen wahlweise in Transmitter oder im ERP.

© Swissdec 2026

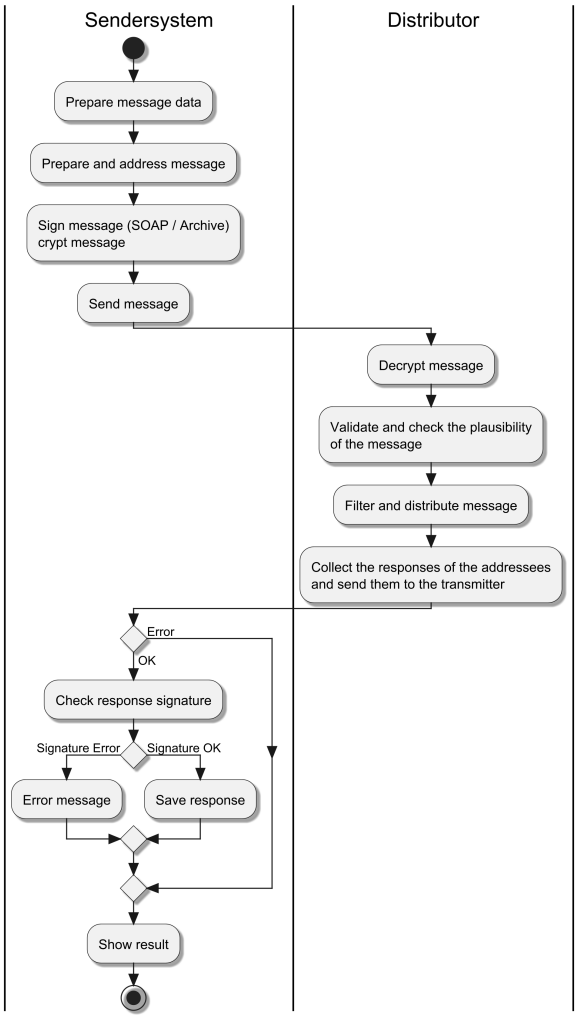


Abbildung 7.1. Synchroner Meldungsprozess

© Swissdec 2026

7.3 Datenstruktur der initialen Meldung

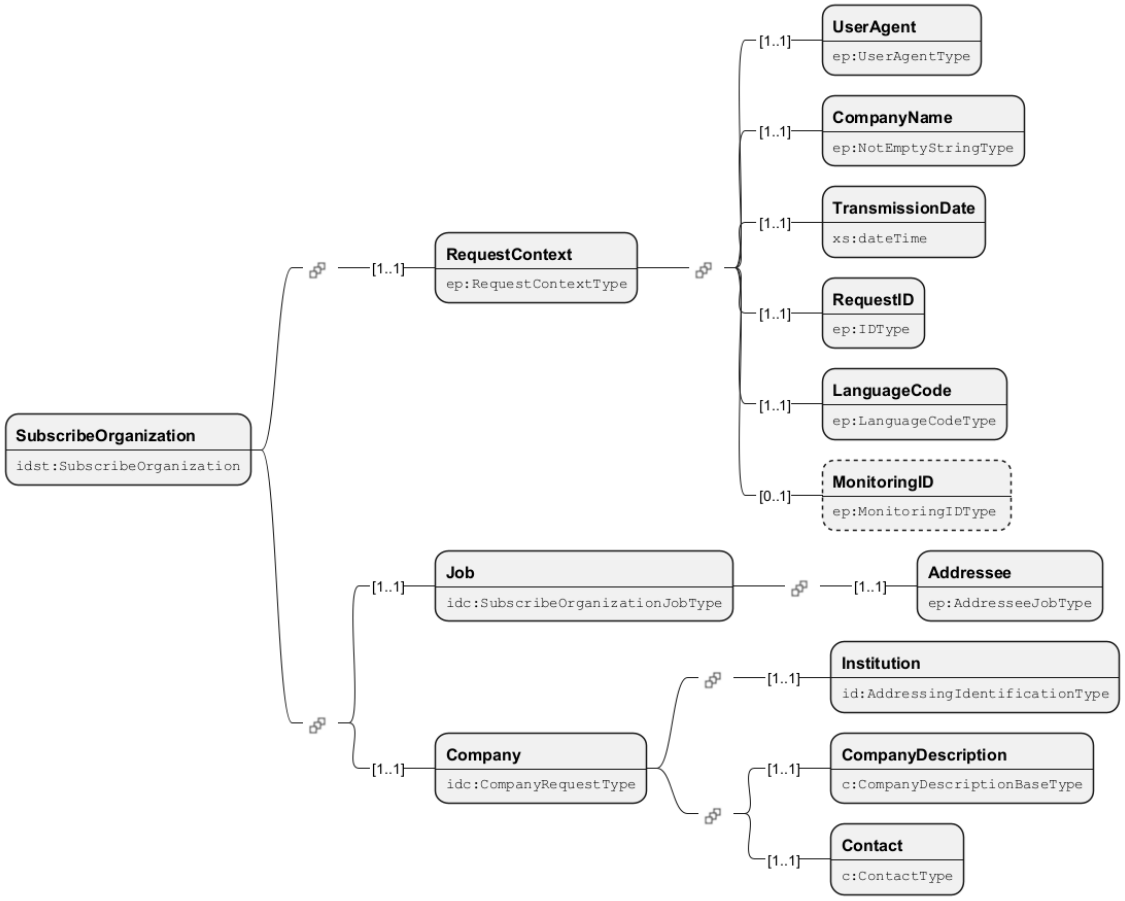


Abbildung 7.2. Schemabild SubscribeOrganization

Feldname	Beschreibung	Typ
RequestContext	Der RequestContext enthält Informationen zum Zeitpunkt der Übermittlung, dem Absender und der Sprache.	ep RequestContext-Type
Job	Steuerungselement für die Verarbeitung der Meldung.	idc SubscribeOrganizationJobType
Company	Anmeldungsanfrage zur Kommunikation mit einem Empfänger.	idc CompanyRequestType
@schemaVersion	Angabe der verwendeten Minor-Version des Schemas.	id SupportedMinorSchemaVersionAttributeType

Tabelle 7.1. Feldbeschreibungen SubscribeOrganization

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType

© Swissdec 2026

Feldname	Beschreibung	Typ
CompanyName	Beschreibung der wesentlichen Unternehmensdaten	ep NotEmptyStringType
TransmissionDate	Übertragungszeitpunkt	xs dateTime
RequestID	AuftragsID	ep IDType
LanguageCode	Sprachcode	ep LanguageCodeType
MonitoringID	Monitoring Identifikation	ep MonitoringIDType

Tabelle 7.2. Feldbeschreibungen RequestContext

Feldname	Beschreibung	Typ
Addressee	Auflistung der Endempfänger, die die übermittelten Daten erhalten sollen.	ep AddresseeJobType

Tabelle 7.3. Feldbeschreibungen Job

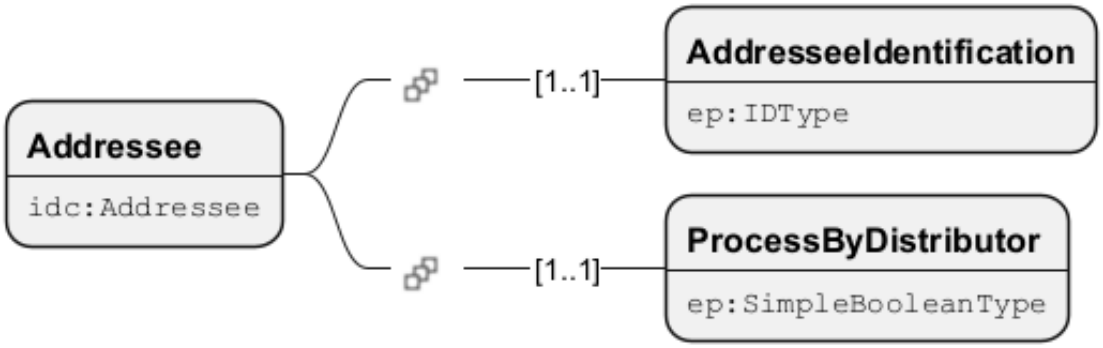


Abbildung 7.3. Schemabild Addressee

Feldname	Beschreibung	Typ
AddresseeIdentification	Identifikation des Adressaten	ep IDType
ProcessByDistributor	Dieses Flag entscheidet, ob an den Distributor übermittelte Daten von ihm verarbeitet oder ignoriert werden sollen. Steht 'false', wird der Distributor keine Verteilung vornehmen und die gesendeten Inhalte für diese Institution ignorieren.	ep SimpleBooleanType
@addresseeID	Referenz auf Adressat	ep InstanceRefIDType

Tabelle 7.4. Feldbeschreibungen Addressee

© Swissdec 2026

7.4 Datenstruktur der initialen Antwort

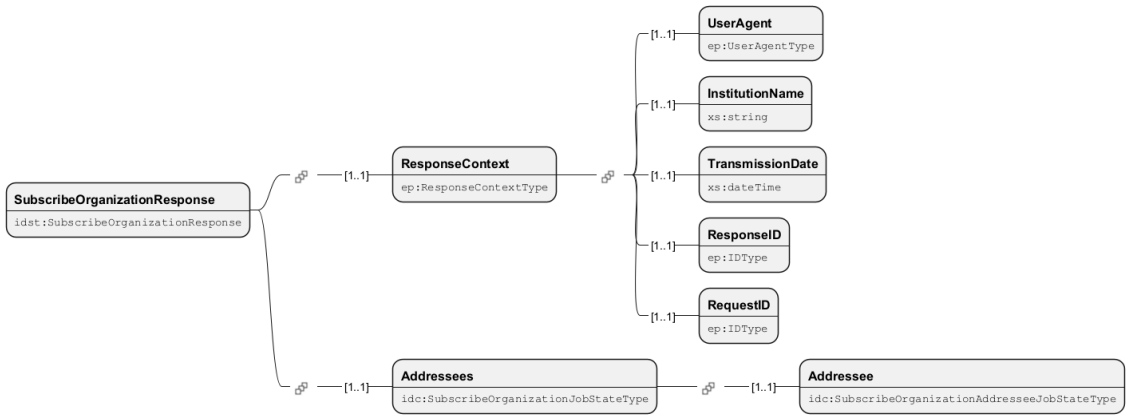


Abbildung 7.4. Schemabild SubscribeOrganizationResponse

Feldname	Beschreibung	Typ
ResponseContext	Der ResponseContext enthält Informationen zum Zeitpunkt der Übermittlung, dem Absender und der Sprache.	ep ResponseContextType
Addressees	Initialer Status des Falls.	idc SubscribeOrganizationJobStateType

Tabelle 7.5. Feldbeschreibungen SubscribeOrganizationResponse

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType
InstitutionName	Institutionsname	xs string
TransmissionDate	Übertragungszeitpunkt	xs dateTime
ResponseID	AntwortID	ep IDType
RequestID	AuftragsID	ep IDType

Tabelle 7.6. Feldbeschreibungen ResponseContext

© Swissdec 2026

7.5 Datenstruktur der Synchronize-Meldung

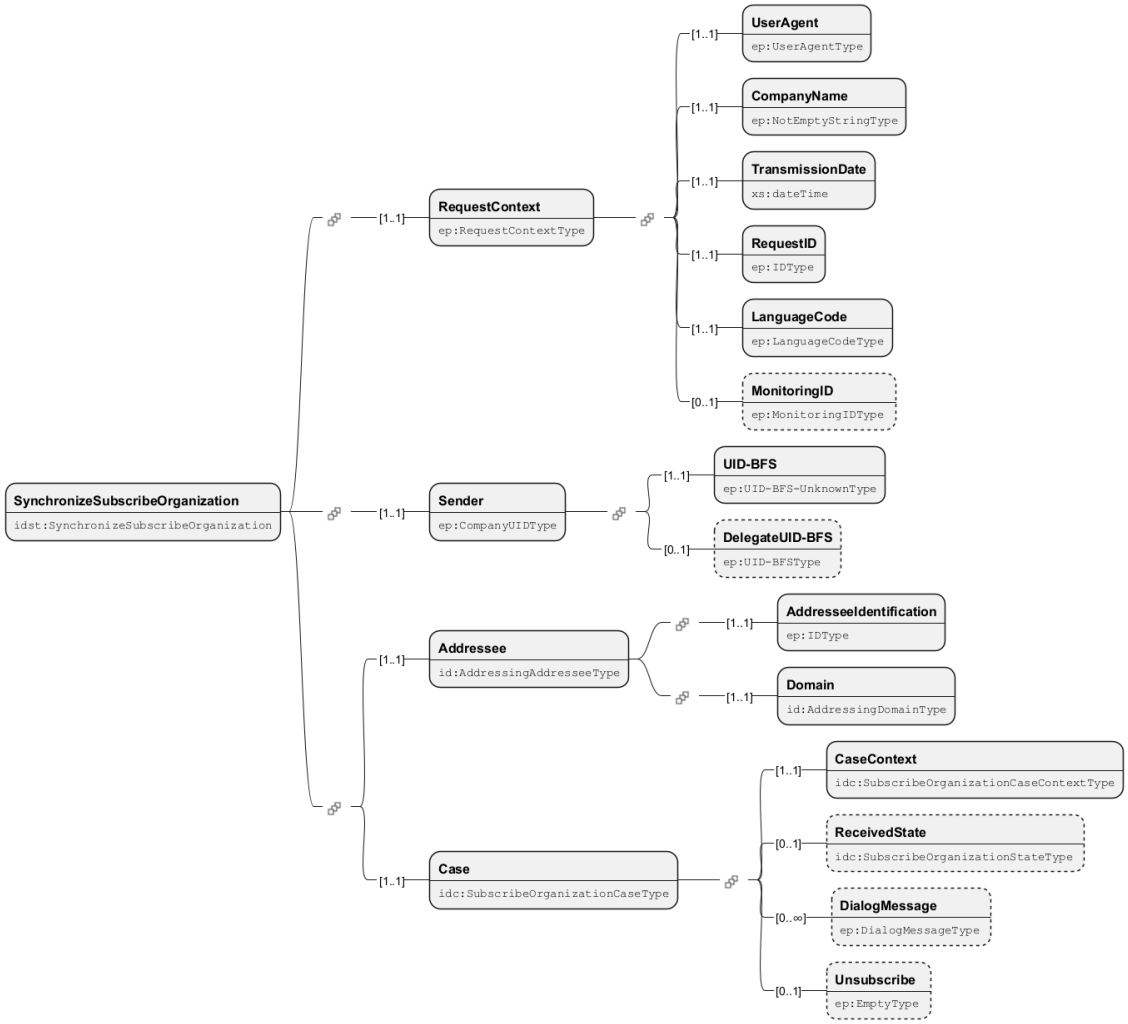


Abbildung 7.5. Schemabild SynchronizeSubscribeOrganization

Feldname	Beschreibung	Typ
RequestContext	Der RequestContext enthält Informationen zum Zeitpunkt der Übermittlung, dem Absender und der Sprache.	ep RequestContext-Type
Sender	Absender der übermittelten Informationen	ep CompanyUIDTy-pe
Addressee	Informationen zum Adressaten	id AddressingAd-dresseeType
Case	Informationen zum Fall.	idc SubscribeOrga-nizationCaseType
@schemaVersion	Angabe der verwendeten Minor-Version des Schemas.	id SupportedMi-norSchemaVer-sionAttributeType

Tabelle 7.7. Feldbeschreibungen SynchronizeSubscribeOrganization

© Swissdec 2026

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType
CompanyName	Beschreibung der wesentlichen Unternehmensdaten	ep NotEmptyStringType
TransmissionDate	Übertragungszeitpunkt	xs dateTime
RequestID	AuftragsID	ep IDType
LanguageCode	Sprachcode	ep LanguageCodeType
MonitoringID	Monitoring Identifikation	ep MonitoringIDType

Tabelle 7.8. Feldbeschreibungen RequestContext

Feldname	Beschreibung	Typ
UID-BFS	Unternehmens-Identifikationsnummer / UID-BFS	ep UID-BFS-UnknownType
DelegateUID-BFS	Unternehmens UID-BFS Typ	ep UID-BFSType

Tabelle 7.9. Feldbeschreibungen Sender

Feldname	Beschreibung	Typ
AddresseeIdentification	Identifikation des Adressaten	ep IDType
Domain	Domäne	id AddressingDomainType

Tabelle 7.10. Feldbeschreibungen Addressee

Feldname	Beschreibung	Typ
CaseContext	Dieses Element enthält Informationen und Identifikatoren zum Fall.	idc SubscribeOrganizationCaseContextType
ReceivedState		idc SubscribeOrganizationStateType
DialogMessage	Semistrukturierte Daten für Dialog zwischen den Fall-Teilnehmern.	ep DialogMessageType
Unsubscribe	Abmeldung durch das Sendersystem.	ep EmptyType

Tabelle 7.11. Feldbeschreibungen Case

© Swissdec 2026

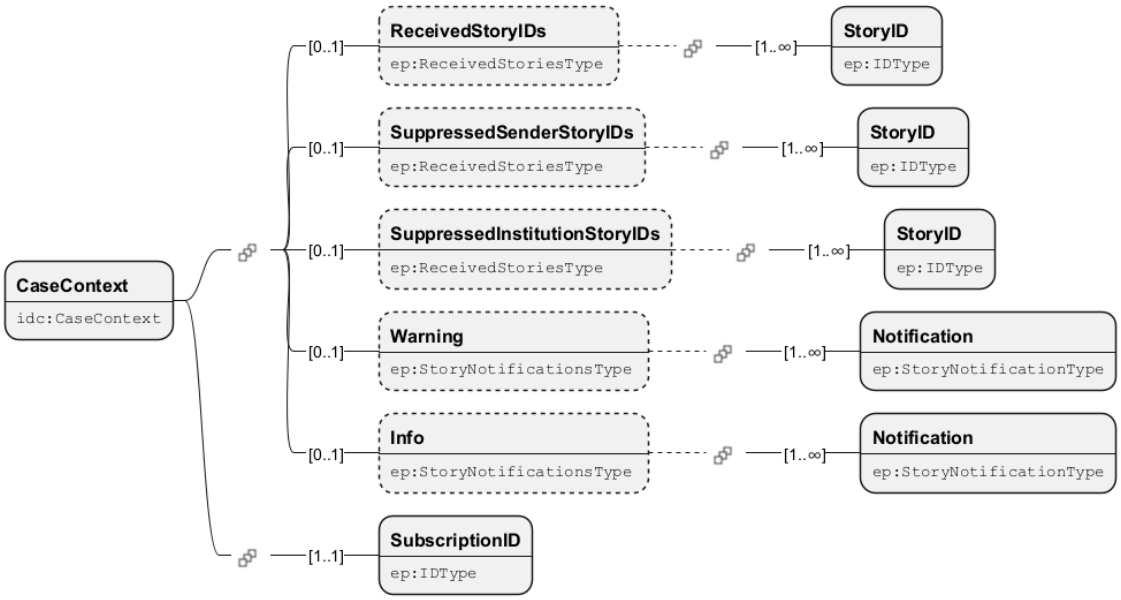


Abbildung 7.6. Schemabild CaseContext

Feldname	Beschreibung	Typ
ReceivedStoryIDs	Die Empfangenen StoryIDs als Quittungsliste.	ep ReceivedStories-Type
SuppressedSen-derStoryIDs	Die StoryIDs des Senders, die vom Distributor unterdrückt wur-den. Das kann bei Mapping-Problemen geschehen.	ep ReceivedStories-Type
SuppressedInstituti-onStoryIDs	Die StoryIDs des Empfängers, die vom Distributor unterdrückt wurden. Das kann bei Mapping-Problemen geschehen.	ep ReceivedStories-Type
Warning	Warnungen, die an den Producer / Transmitter zurückgeschickt werden.	ep StoryNotificati-onsType
Info	Informationen, die an den Producer / Transmitter zurückgeschickt werden.	ep StoryNotificati-onsType
SubscriptionID	Fachverfahrensspezifische Fallnummer des Distributors.	ep IDType

Tabelle 7.12. Feldbeschreibungen CaseContext

© Swissdec 2026

7.6 Datenstruktur der Synchronize-Antwort



Abbildung 7.7. Schemabild SynchronizeSubscribeOrganizationResponse

Feldname	Beschreibung	Typ
ResponseContext	Der ResponseContext enthält Informationen zum Zeitpunkt der Übermittlung, dem Absender und der Sprache.	ep ResponseContextType
Error	Die Verarbeitung war nicht erfolgreich und es sind Fehler aufgetreten.	ep ErrorResponseType
SynchronizeSubscribeOrganizationConsumer	Dieses Element synchronisiert die Anmeldung der Organisation.	idc SynchronizeSubscribeOrganizationConsumerType

Tabelle 7.13. Feldbeschreibungen SynchronizeSubscribeOrganizationResponse

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType
InstitutionName	Institutionsname	xs string
TransmissionDate	Übertragungszeitpunkt	xs dateTime
ResponseID	AntwortID	ep IDType

© Swissdec 2026

Feldname	Beschreibung	Typ
RequestID	AuftragsID	ep IDType

Tabelle 7.14. Feldbeschreibungen ResponseContext

Feldname	Beschreibung	Typ
EndUserInformation	Informationen zum Endnutzer	xs string
DetailInformation	Informationen im Detail	xs string
FaultInformation	Informationen zum Fehler	ep ConsumerFaultType
PlannedMaintenance	Geplante Wartung	ep PlannedMaintenanceWindowType

Tabelle 7.15. Feldbeschreibungen Error

Feldname	Beschreibung	Typ
AddresseeContext	Kontextangaben zum Adressaten	ep AddresseeResponseContextType
Addressee	Informationen zum Adressaten	ep AddresseeType
Case	Informationen zum Fall.	idc SubscribeOrganizationConsumerCaseType
Available	Informationen zur Verfügbarkeit.	idc AvailableType

Tabelle 7.16. Feldbeschreibungen SynchronizeSubscribeOrganizationConsumer

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType
InstitutionName	Institutionsname	xs string
TransmissionDate	Übertragungszeitpunkt	xs dateTime
ResponseID	AntwortID	ep IDType
RequestID	AuftragsID	ep IDType
ProducerResponseNotifications	Antwort vom Producer zurückgegeben	ep FeedbackNotificationsType
Warning	Warnung wurde zurückgegeben	ep NotificationsType
Info	Informationen wurden zurückgegeben	ep NotificationsType

© Swissdec 2026

Feldname	Beschreibung	Typ

Tabelle 7.17. Feldbeschreibungen AddresseeContext

Feldname	Beschreibung	Typ
AddresseeIdentification	Identifikation des Adressaten	ep IDType

Tabelle 7.18. Feldbeschreibungen Addressee

Feldname	Beschreibung	Typ
CaseContext	Dieses Element enthält Informationen und Identifikatoren zum Fall.	idc SubscribeOrganizationCaseContextType
State		idc SubscribeOrganizationStateType
DialogMessage	Semistrukturierte Daten für Dialog zwischen den Fall-Teilnehmern.	ep DialogMessageType

Tabelle 7.19. Feldbeschreibungen Case

© Swissdec 2026

8 DeclareIncident

8.1 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.

8.1.1 Ereignis melden: DeclareIncident

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. Dieser sendet via Distributor zurück, ob er das Ereignis annimmt. Im Erfolgsfall kann nun mit dem nächsten Schritt, der Synchronisierung, fortgefahren werden.

8.1.2 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.

8.1.3 Regelmässiges Synchronisieren der Ereignisse

Auch wenn kein konkreter Fall in Arbeit ist, führt das Sendersystem regelmässige Synchronisierungen über ein ganzes Unternehmen durch. So wird es auch informiert, wenn für ein aktuell inaktives Ereignis neue Informationen zur Verfügung stehen.

8.2 Initiale synchrone Meldung

Die Aktivitätsdiagramme sind ohne den Endreceiver dargestellt. Die wesentliche Transmitterfunktionalität *soll* dargestellt werden. Die Anzeigen erfolgen wahlweise in Transmitter oder im ERP.

© Swissdec 2026

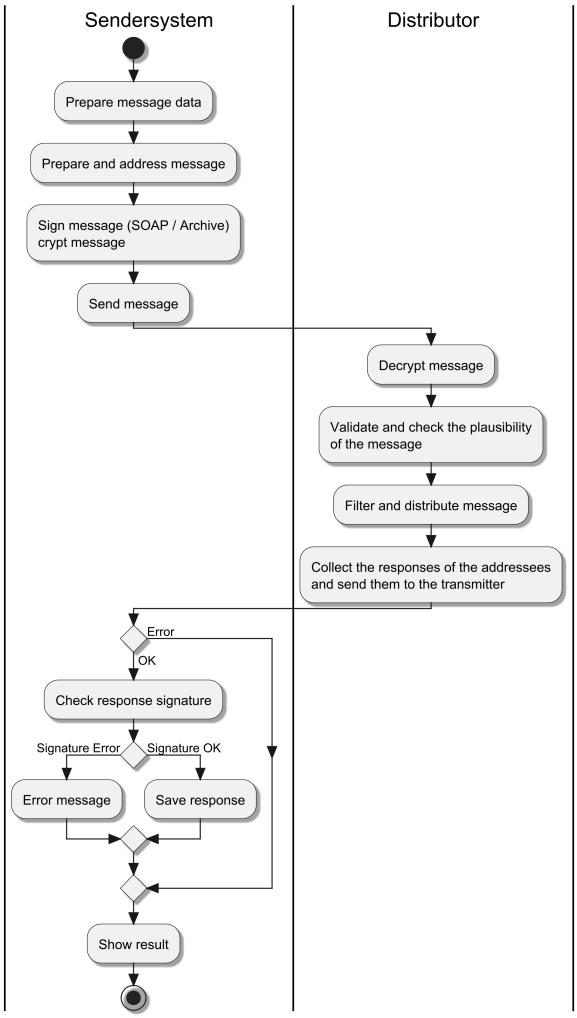


Abbildung 8.1. Synchroner Meldungsprozess

8.3 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.

© Swissdec 2026

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

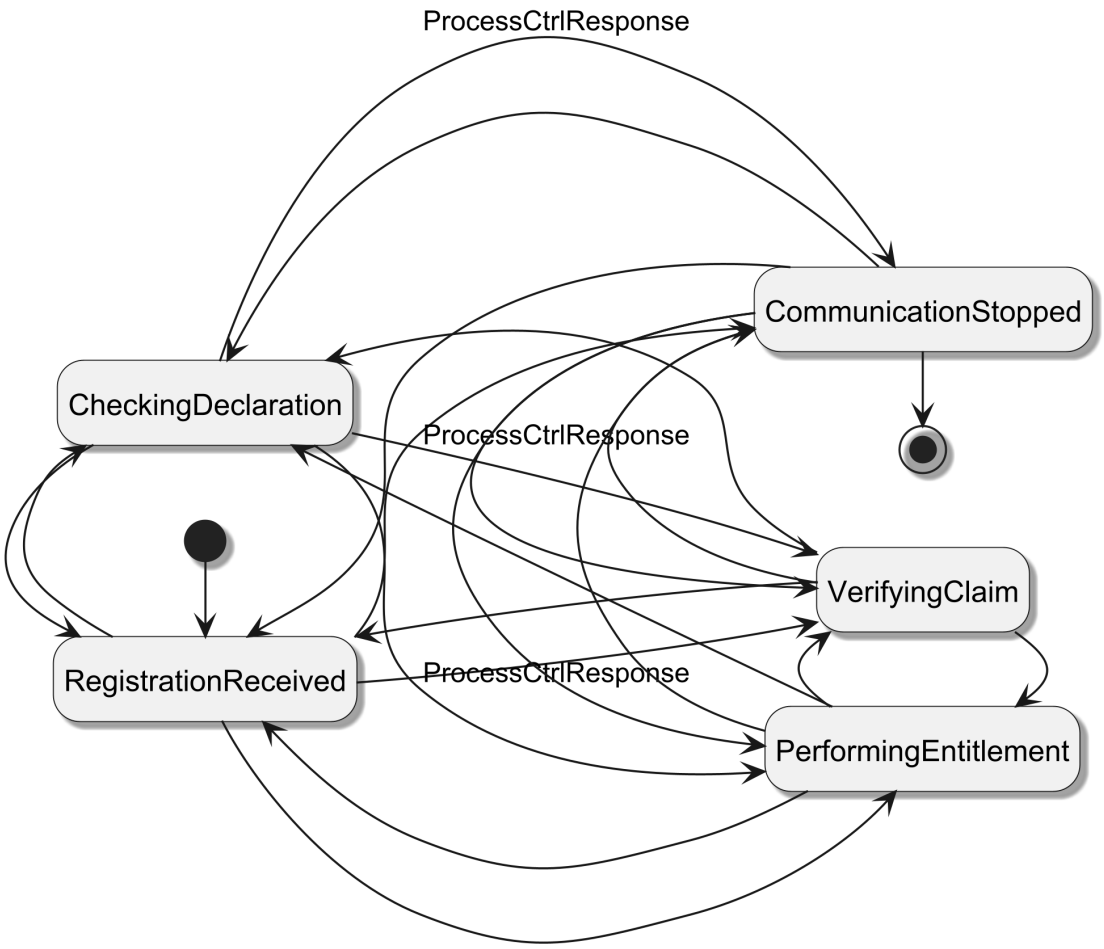


Abbildung 8.2. DeclareIncident Protokollstatus

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

© Swissdec 2026

Aktueller Zustand	Story	Gemeldeter Zustand
PerformingEntitlement	*ohne Story*	RegistrationReceived
PerformingEntitlement	*ohne Story*	CheckingDeclaration
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 8.1. Zustandstabelle

8.4 Datenstruktur der initialen Meldung

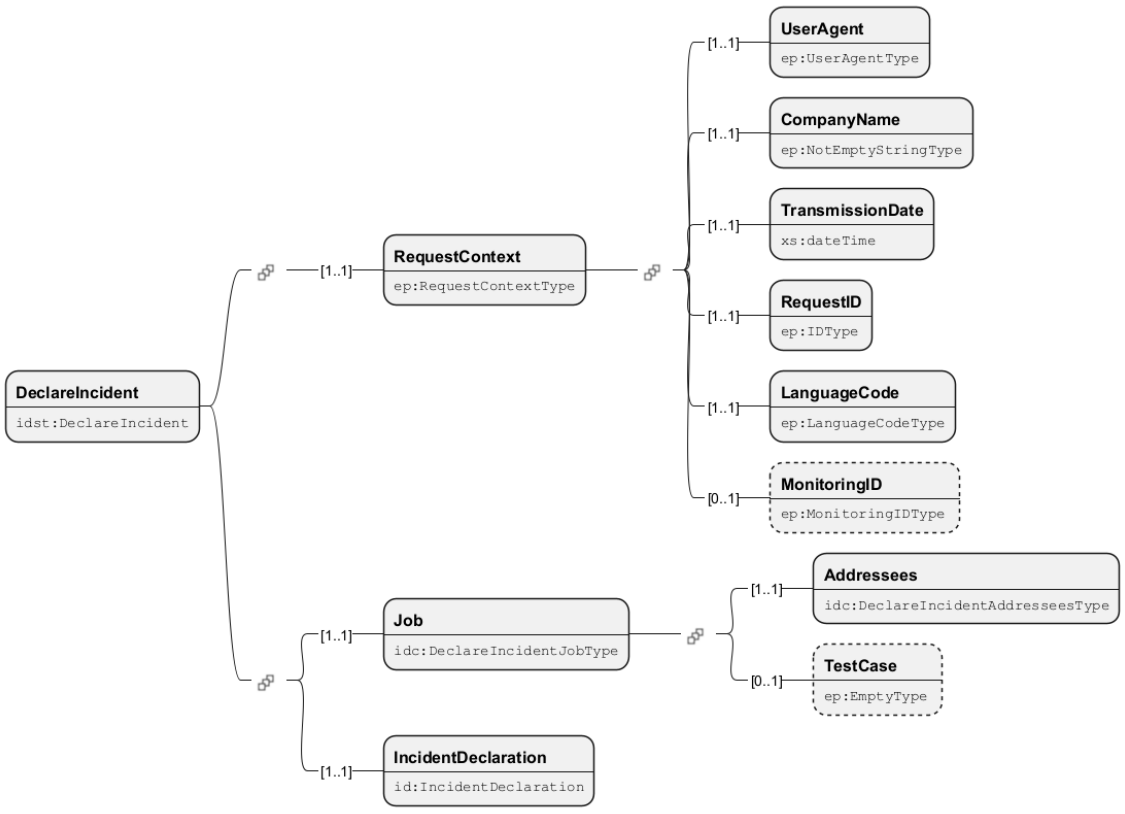


Abbildung 8.3. Schemabild DeclareIncident

Feldname	Beschreibung	Typ
RequestContext	Der RequestContext enthält Informationen zum Zeitpunkt der Übermittlung, dem Absender und der Sprache.	ep_RequestContext-Type
Job	Steuerungselement für die Verarbeitung der Meldung.	idc_DeclareIncident-JobType

© Swissdec 2026

Feldname	Beschreibung	Typ
IncidentDeclaration	Fachliche Datenstruktur für die Ereignismeldung.	id IncidentDeclaration

Tabelle 8.2. Feldbeschreibungen DeclareIncident

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType
CompanyName	Beschreibung der wesentlichen Unternehmensdaten	ep NotEmptyStringType
TransmissionDate	Übertragungszeitpunkt	xs dateTime
RequestID	AuftragsID	ep IDType
LanguageCode	Sprachcode	ep LanguageCodeType
MonitoringID	Monitoring Identifikation	ep MonitoringIDType

Tabelle 8.3. Feldbeschreibungen RequestContext

Feldname	Beschreibung	Typ
Addressees	Angaben zum Adressaten.	idc DeclareIncidentAddresseesType
TestCase	Testmeldung kennzeichnen. Mit diesem Flag darf die Meldung nicht produktiv verarbeitet werden.	ep EmptyType

Tabelle 8.4. Feldbeschreibungen Job

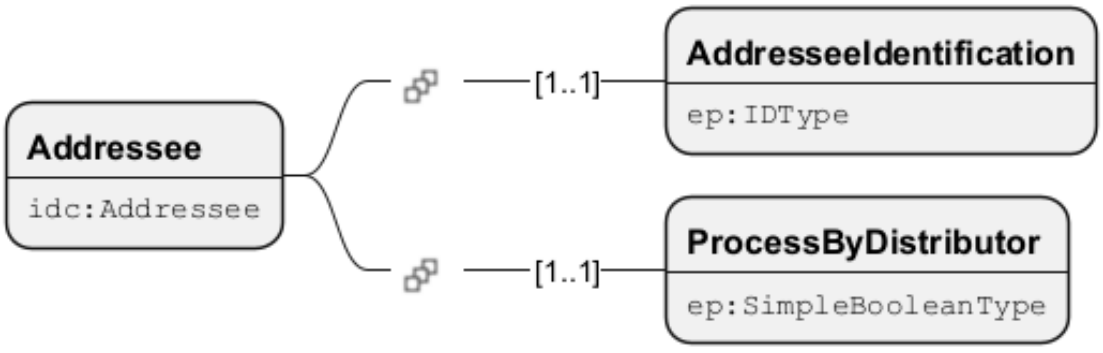


Abbildung 8.4. Schemabild Addressee

Feldname	Beschreibung	Typ
AddresseeIdentification	Identifikation des Adressaten	ep IDType

© Swissdec 2026

Feldname	Beschreibung	Typ
ProcessByDistributor	Dieses Flag entscheidet, ob an den Distributor übermittelte Daten von ihm verarbeitet oder ignoriert werden sollen. Steht 'false', wird der Distributor keine Verteilung vornehmen und die gesendeten Inhalte für diese Institution ignorieren.	ep SimpleBooleanType
@addresseeID	Referenz auf Adressat	ep InstanceRefIDType

Tabelle 8.5. Feldbeschreibungen Addressee

8.5 Datenstruktur der initialen Antwort

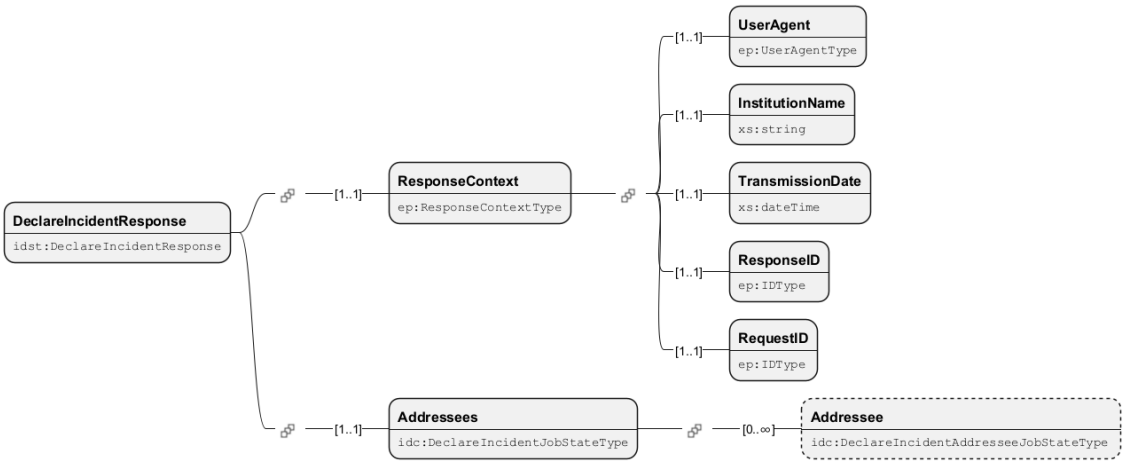


Abbildung 8.5. Schemabild DeclareIncidentResponse

Feldname	Beschreibung	Typ
ResponseContext	Der ResponseContext enthält Informationen zum Zeitpunkt der Übermittlung, dem Absender und der Sprache.	ep ResponseContextType
Addressees	Initialer Status des Falls.	idc DeclareIncidentJobStateType

Tabelle 8.6. Feldbeschreibungen DeclareIncidentResponse

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType
InstitutionName	Institutionsname	xs string
TransmissionDate	Übertragungszeitpunkt	xs dateTime
ResponseID	AntwortID	ep IDType

© Swissdec 2026

Feldname	Beschreibung	Typ
RequestID	AuftragsID	ep IDType

Tabelle 8.7. Feldbeschreibungen ResponseContext

8.6 Datenstruktur der Synchronize-Meldung

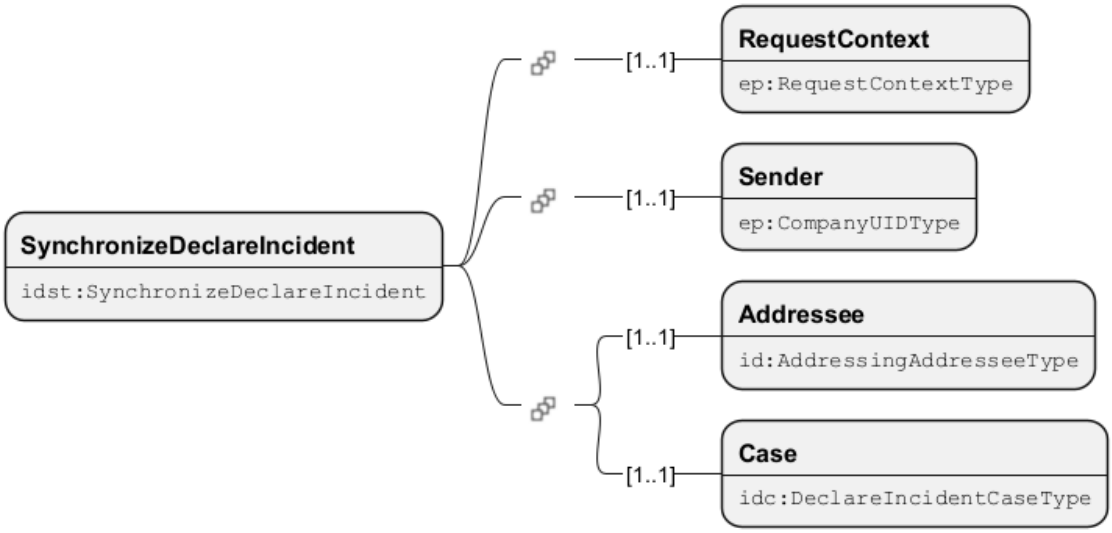


Abbildung 8.6. Schemabild SynchronizeDeclareIncident

Feldname	Beschreibung	Typ
RequestContext	Der RequestContext enthält Informationen zum Zeitpunkt der Übermittlung, dem Absender und der Sprache.	ep RequestContextType
Sender	Absender der übermittelten Informationen	ep CompanyUIDType
Addressee	Informationen zum Adressaten	id AddressingAddresseeType
Case	Informationen zum Fall.	idc DeclareIncidentCaseType
@schemaVersion	Angabe der verwendeten Minor-Version des Schemas.	id SupportedMinorSchemaVersionAttributeType

Tabelle 8.8. Feldbeschreibungen SynchronizeDeclareIncident

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType
CompanyName	Beschreibung der wesentlichen Unternehmensdaten	ep NotEmptyStringType

© Swissdec 2026

Feldname	Beschreibung	Typ
TransmissionDate	Übertragungszeitpunkt	xs dateTime
RequestID	AuftragsID	ep IDType
LanguageCode	Sprachcode	ep LanguageCodeType
MonitoringID	Monitoring Identifikation	ep MonitoringIDType

Tabelle 8.9. Feldbeschreibungen RequestContext

Feldname	Beschreibung	Typ
UID-BFS	Unternehmens-Identifikationsnummer / UID-BFS	ep UID-BFS-UnknownType
DelegateUID-BFS	Unternehmens UID-BFS Typ	ep UID-BFSType

Tabelle 8.10. Feldbeschreibungen Sender

Feldname	Beschreibung	Typ
AddresseeIdentification	Identifikation des Adressaten	ep IDType
Domain	Domäne	id AddressingDomainType

Tabelle 8.11. Feldbeschreibungen Addressee

Feldname	Beschreibung	Typ
CaseContext	Dieses Element enthält Informationen und Identifikatoren zum Fall.	idc DeclareIncidentCaseContextType
ReceivedState	Initialer Status des Falls.	idc DeclareIncidentStateType
DialogMessage	Semistrukturierte Daten für Dialog zwischen den Fall-Teilnehmern.	ep DialogMessageType
IncapacitiesToWork	Arbeitsunfähigkeits-Story	id IncapacitiesToWorkType
TakeWorkUpFully	Arbeitswiederaufnahme-Story	id TakeWorkUpFullyType
Treatments	Medizinische Behandlung	id TreatmentsType
Agreement	Andere Versicherer	id AgreementType
AnnualSalary	Jahreslohn-Story	id AnnualSalaryType

© Swissdec 2026

Feldname	Beschreibung	Typ
AnnualWorkingTime	Arbeitszeit-Story	id AnnualWorkingTimeType
Payment	Zahlungsverbindung	id PaymentType
AdditionsPerson	Arbeitsvertrag-Story	id AdditionsPersonType
Absences	Ausfallstunden	id AbsencesType
Vacations	Ferien-Story	id VacationsType
SpecialCode	Spezialcode-Story	id SpecialCodeType
AccidentDescription	Unfallbeschreibung	id AccidentDescriptionType
SalaryRaise	Lohnerhöhung	id SalaryRaiseType
PersonMutated	Mutieren von Personendaten	id PersonMutatedType
SpecialCoverage	Spezialfälle	id SpecialCoverageType
OtherEmployers	Andere Arbeitgeber	ep BooleanStoryType
ShortTimeWork	Kurzarbeit	ep BooleanStoryType
Attachment	Dokumente	id AttachmentType
ProcessCtrl	Prozess-Kontrolle	id ProcessCtrlRequestType

Tabelle 8.12. Feldbeschreibungen Case

© Swissdec 2026

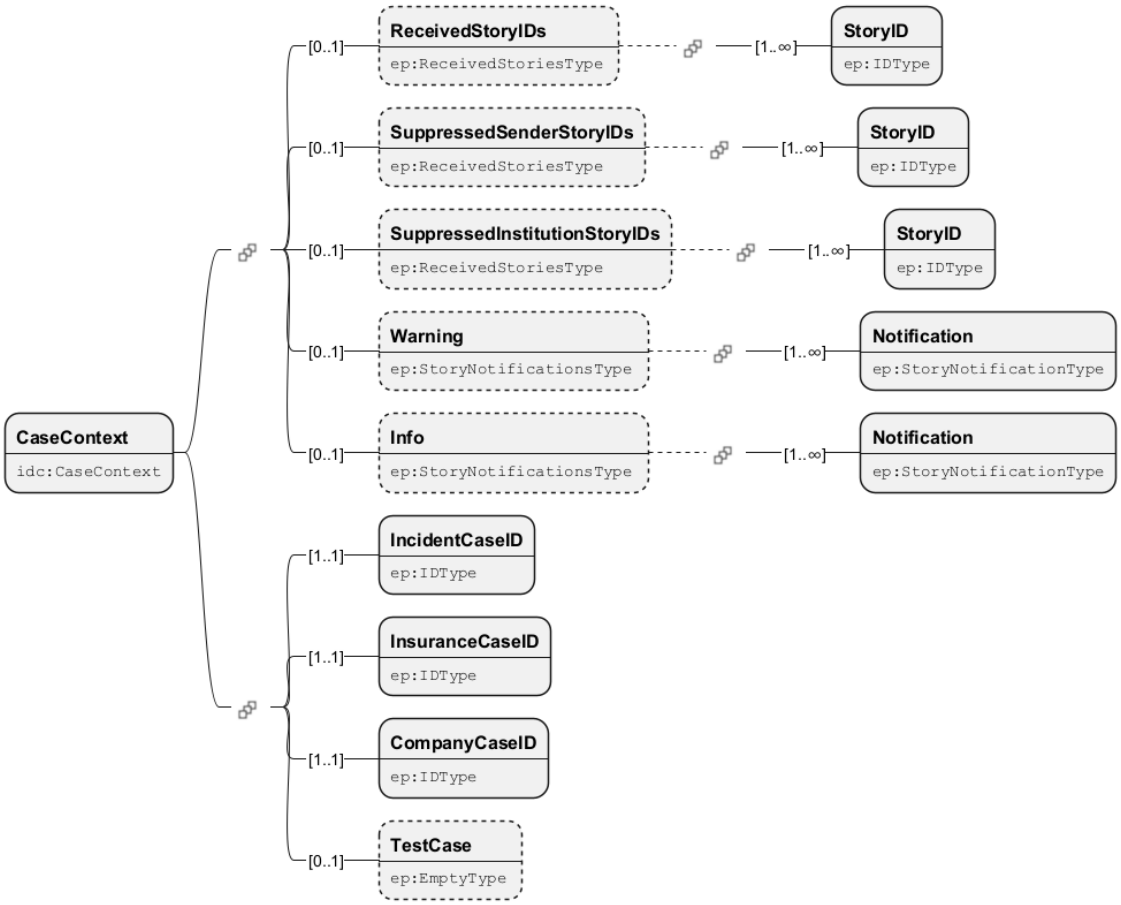


Abbildung 8.7. Schemabild CaseContext

Feldname	Beschreibung	Typ
ReceivedStoryIDs	Die Empfangenen StoryIDs als Quittungsliste.	ep ReceivedStories-Type
SuppressedSen-derStoryIDs	Die StoryIDs des Senders, die vom Distributor unterdrückt wur-den. Das kann bei Mapping-Problemen geschehen.	ep ReceivedStories-Type
SuppressedInstituti-onStoryIDs	Die StoryIDs des Empfängers, die vom Distributor unterdrückt wurden. Das kann bei Mapping-Problemen geschehen.	ep ReceivedStories-Type
Warning	Warnungen, die an den Producer / Transmitter zurückgeschickt werden.	ep StoryNotificati-onsType
Info	Informationen, die an den Producer / Transmitter zurückgeschickt werden.	ep StoryNotificati-onsType
IncidentCaseID	Fachverfahrensspezifische Fallnummer des Distributors.	ep IDType
InsuranceCaseID	Fallnummer des Empfängersystems.	ep IDType
CompanyCaseID	Fallnummer des Sendersystems.	ep IDType

© Swissdec 2026

Feldname	Beschreibung	Typ
TestCase	Testmeldung kennzeichnen. Mit diesem Flag darf die Meldung nicht produktiv verarbeitet werden.	ep EmptyType

Tabelle 8.13. Feldbeschreibungen CaseContext

8.7 Datenstruktur der Synchronize-Antwort

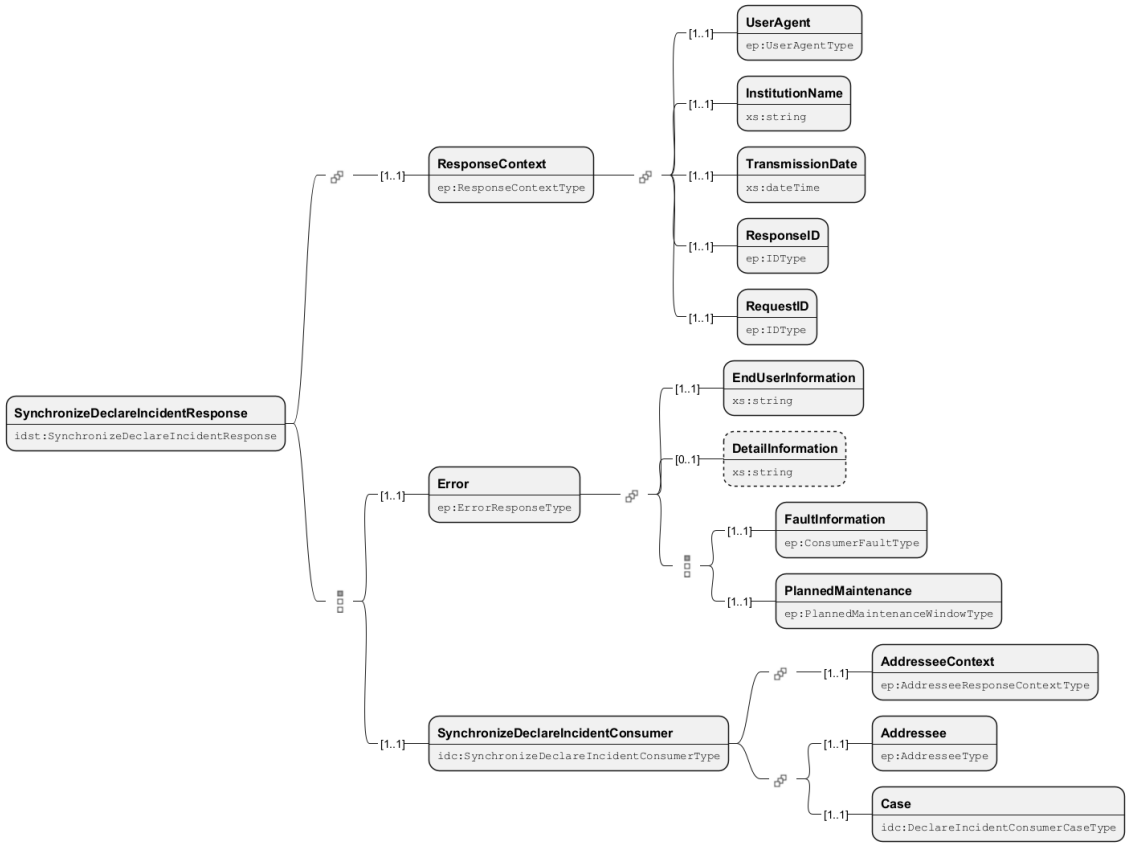


Abbildung 8.8. Schemabild SynchronizeDeclareIncidentResponse

Feldname	Beschreibung	Typ
ResponseContext	Der ResponseContext enthält Informationen zum Zeitpunkt der Übermittlung, dem Absender und der Sprache.	ep ResponseContextType
Error	Die Verarbeitung war nicht erfolgreich und es sind Fehler aufgetreten.	ep ErrorResponseType
SynchronizeDeclareIncidentConsumer		idc SynchronizeDeclareIncidentConsumerType

Tabelle 8.14. Feldbeschreibungen SynchronizeDeclareIncidentResponse

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType

© Swissdec 2026

Feldname	Beschreibung	Typ
InstitutionName	Institutionsname	xs string
TransmissionDate	Übertragungszeitpunkt	xs dateTime
ResponseID	AntwortID	ep IDType
RequestID	AuftragsID	ep IDType

Tabelle 8.15. Feldbeschreibungen ResponseContext

Feldname	Beschreibung	Typ
EndUserInformation	Informationen zum Endnutzer	xs string
DetailInformation	Informationen im Detail	xs string
FaultInformation	Informationen zum Fehler	ep ConsumerFaultType
PlannedMaintenance	Geplante Wartung	ep PlannedMaintenanceWindowType

Tabelle 8.16. Feldbeschreibungen Error

Feldname	Beschreibung	Typ
AddresseeContext	Kontextangaben zum Adressaten	ep AddresseeResponseContextType
Addressee	Informationen zum Adressaten	ep AddresseeType
Case	Informationen zum Fall.	idc DeclareIncidentConsumerCaseType

Tabelle 8.17. Feldbeschreibungen SynchronizeDeclareIncidentConsumer

Feldname	Beschreibung	Typ
UserAgent	Zur Qualitätsicherung des Übermittlungsprozesses werden die wesentlichen Informationen der beteiligten gesichert	ep UserAgentType
InstitutionName	Institutionsname	xs string
TransmissionDate	Übertragungszeitpunkt	xs dateTime
ResponseID	AntwortID	ep IDType
RequestID	AuftragsID	ep IDType
ProducerResponseNotifications	Antwort vom Producer zurückgegeben	ep FeedbackNotificationsType

© Swissdec 2026

Feldname	Beschreibung	Typ
Warning	Warnung wurde zurückgegeben	ep NotificationsType
Info	Informationen wurden zurückgegeben	ep NotificationsType

Tabelle 8.18. Feldbeschreibungen AddresseeContext

Feldname	Beschreibung	Typ
AddresseeIdentification	Identifikation des Adressaten	ep IDType

Tabelle 8.19. Feldbeschreibungen Addressee

Feldname	Beschreibung	Typ
CaseContext	Dieses Element enthält Informationen und Identifikatoren zum Fall.	idc DeclareIncidentCaseContextType
State	Initialer Status des Falls.	idc DeclareIncidentStateType
DialogMessage	Semistrukturierte Daten für Dialog zwischen den Fall-Teilnehmern.	ep DialogMessageType
DigitalizationScope	Digitalisierungsbereiche	id DigitalizationScopeType
Settlement	Taggeldleistungsabrechnung	id SettlementType
Repayment	Rückforderung	id RepaymentType
CoveredDailyAllowance	Taggeldansatz	id CoveredDailyAllowanceType
ProcessCtrl	Prozess-Kontrolle	id ProcessCtrlResponseType

Tabelle 8.20. Feldbeschreibungen Case

© Swissdec 2026

9 Use Cases

9.1 UC001 Initiale Meldung senden

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

Kurzbeschreibung	Eine elektronische Meldung **muss** an einen oder mehrere Endreceiver versendet werden. Die Rückantwort der Endreceiver wird ausgewertet und abgelegt. Ein Archiv-File der gesendeten Meldung wird ebenfalls gesichert.
Akteure	Sendersystem, Distributor, Endreceiver
Auslöser	Das Sendersystem detektiert eine Situation, die per Swissdec gemeldet werden muss/kann.
Vorbedingungen	Das Sendersystem ist in der Lage, elektronische Swissdec-Meldungen zu versenden und die Antwort zu verarbeiten.
Nachbedingungen	Die Meldung wurde vom Endreceiver empfangen und durch eine Rückantwort quittiert. Bei einem Fehlschlag: • Fehlermeldung
Included Use Cases	Abschnitt 9.2, „UC002 Status abholen“ , Abschnitt 9.15, „UC015 Security anwenden“
Standardablauf	1. Das Sendersystem übergibt dem Transmitter die Daten mit den Empfängeradressen. 2. Der Transmitter bereitet die Meldung als SOAP-Request mit zugehöriger Adressierung (Job) auf. 3. Die Meldung wird nach Spezifikation signiert und verschlüsselt. 4. Der Transmitter sendet die aufbereitete und signierte Meldung über HTTPS an den Distributor. Dieser Schritt hat 2 mögliche Ausprägungen: a. Asynchroner Aufruf, bei dem der Distributor mit einem JobKey antwortet und der Transmitter mittels <GetStatus> Polling die Antworten der Endreceiver erhält. b. Synchroner Aufruf, ein blockierender Aufruf, auf dem die Antworten der Endreceiver direkt als Antwort auf die Meldung zurückgegeben werden. Beim synchronen Aufruf sind die Informationen, welche beim asynchronen Aufruf mittels GetStatus abgeholt werden bereits in der Response vorhanden. Welche Methode bei welcher Operation benötigt wird, ist in den operationsspezifischen Kapiteln definiert. 5. Der Distributor prüft die Meldung auf Validität und Plausibilität. 6. Der Distributor bereitet eine oder mehrere Meldungen für die gewählten Endreceiver auf und sendet diese an den/die Endreceiver. 7. Der Endreceiver prüft die Meldung und startet die Verarbeitung des Jobs. 8. Der Endreceiver meldet seine Antwort dem Distributor, der sie dem Sendersystem zukommen lässt (synchron oder asynchron). 9. Der Transmitter wartet die Rückantwort des Distributors ab. 10. Das Ergebnis der Übermittlung wird auf Transmitterseite aufbereitet und angezeigt.
Alternative Abläufe	**Abschnitt 9.3, „UC003 Testmeldung kennzeichnen“** {nach Schritt 1}

© Swissdec 2026

	1. b) Die Meldung wird als Testmeldung gekennzeichnet. (Ein Element Test-Case wird im Job Element eingefügt). {weiter mit Schritt 2} ACHTUNG: Wird ein Fall als Testfall gemeldet, **müssen** auch alle weiteren Synchronisierungen (im CaseContext) als Testfall markiert sein (Abschnitt 9.5, „UC005 Synchronisieren“). {nach Schritt 1} 2. c) Die Meldung wird als Ersatzmeldung gekennzeichnet. (Ein Element Substitution mit dem Identifikator der vorangegangenen Meldung wird in die neue Meldung eingefügt [[CONTAINERXSD]]). {weiter mit Schritt 2}
Fehlerliste	Fachliche Fehler: • Die Meldung verstösst gegen die Plausibilisierungsregeln Technische Fehler: • Fehler beim Signieren oder Verschlüsseln. In diesem Fall *sollte* die Übermittlung zu einem späteren Zeitpunkt wiederholt werden, oder, falls das Problem langfristig besteht, Rückfrage beim Hersteller genommen werden. • Der Endreceiver befindet sich in einem geplanten Wartungsfenster (nur bei synchroner Übermittlung möglich). Der Benutzer *sollte* darauf hingewiesen werden, dass nach Ende des Wartungsfensters erneut übermittelt kann. • Der Endreceiver ist nicht erreichbar (nur bei synchroner Übermittlung möglich). Die Übermittlung kann zu einem späteren Zeitpunkt erneut versucht werden. • Der Endreceiver ist nicht adressierbar (nur bei synchroner Übermittlung möglich). Überprüfung der Adressierung seitens Sendersystem/Transmitter. • Die vom Sendersystem aufbereitete Meldung entspricht nicht dem Schema (Validität nicht gegeben) • JobKey ungültig. Dies kann sowohl ein fehlerhafter JobKey sein, als auch ein Ablauf der Gültigkeit.

Tabelle 9.1. UC001 Initiale Meldung senden

9.1.1 Spezielle Anforderungen

9.1.1.1 RequestContext

Jede Swissdec Operation enthält das Element RequestContext, welches korrekt abgefüllt werden muss. Der RequestContext dient zur Identifikation und Nachvollziehbarkeit von Übermittlungen.

UserAgent	**Wichtig!** Der UserAgent muss im Anschluss an die abgeschlossene Swissdec-Zertifizierung aktualisiert werden und muss sämtliche Informationen zum ERP-System (und nicht zum Endbenutzer) enthalten. Producer: Name des Softwareherstellers Name: Name des Produktes Version: Version der Transmittersoftware

© Swissdec 2026

	StandardVersion: Version des Swissdec Standards (inkl. Minor-Version), z.b. 6.0 Certificate: Zertifikat ID aus der Swissdec Zertifizierung. Tabelle 9.3. UserAgentType
CompanyName	Name der übermittelten Firma
TransmissionDate	Zeitstempel der Übermittlung
RequestID	Eindeutige RequestID der Übermittlung (technisch).
LanguageCode	Sprache der übermittelnden Firma oder verantwortlichen Person (wird für Fehlerausgaben verwendet)
MonitoringID	Diese ID wird verwendet, um auf den Swissdec Testsystemen den Softwarehersteller zu identifizieren. Die MonitoringID ist für die Verwendung der Testsysteme zwingend, wird aber im produktiven Umfeld nicht gebraucht. *(Sollte auf produktiven Systemen leer sein)*

Tabelle 9.2. RequestContextType

9.1.1.2 Auswahl der Empfänger und Filterung

Im Element <Job> werden die Endempfänger adressiert, die bei dieser Übermittlung Daten erhalten sollen (Addressees). Für jeden Addressee gibt es das Element <ProcessByDistributor> welches als Boolean auf true oder false gesetzt werden kann. Ist der Wert 'false', wird diese Institution unabhängig von den übermittelten Daten vom Distributor ignoriert und nicht verarbeitet.

Das Element <ProcessByDistributor> *kann* somit genutzt werden, um die Verarbeitung einer Institution momentan auszusetzen. Dies kann zum Beispiel beim Erstellen einer Ersatzmeldung von Nutzen sein. Es wird empfohlen, dass bei der Übermittlung grösserer Datenmengen (>2000 Personen) die Daten bereits vorher gefiltert werden und die Filterung nicht dem Distributor überlassen wird.

9.1.1.3 Archiv-Files erstellen

Mit dieser Anforderung *wird* sichergestellt, dass eine Kopie jeder gesendeten und empfangenen Meldung gesichert wird. Die Daten **müssen** zu einem SOAP-Request aufbereitet und als XML-Instanzdokument abgelegt werden. Archivdateien **müssen** signiert sein, dürfen aber nicht verschlüsselt sein. Welche Dateien langfristig archiviert werden und wie lange sie zur Verfügung stehen müssen, ist abhängig von der Domäne/Institution und wird von der Fachseite vorgegeben.

9.2 UC002 Status abholen

Kurzbeschreibung	Siehe Abschnitt 9.1, „UC001 Initiale Meldung senden“ . Für jede asynchrone Operation existiert eine dazugehörige <GetStatusFrom*> Operation, um eine Statusmeldung für die übermittelten Daten abzuholen. Diese zeigt an, ob die adressierten Endreceiver die übermittelten Daten aus technischer Sicht akzeptiert haben. Der JobKey in der Antwort der Abschnitt 9.1, „UC001 Initiale Meldung senden“ wird dazu verwendet, die zur Meldung gehörige Statusmeldung abzuholen.
Akteure	Sendersystem, Distributor

© Swissdec 2026

Auslöser	Es wurde eine Lohnmeldung übermittelt und vom Distributor eine Bestätigung mit JobKey empfangen.
Vorbedingungen	Das Sendersystem verfügt über einen JobKey, der es ermöglicht, den Status einer laufenden Übermittlung abzufragen.
Nachbedingungen	Der aktuelle Status der Übermittlung wurde beim Distributor abgeholt. Bei einem Fehlschlag: • Fehlermeldung
Included Use Cases	Abschnitt 9.15, „UC015 Security anwenden“
Standardablauf	1. Der Transmitter bereitet eine Statusabfrage, die den JobKey des aktuellen Falls enthält, auf. 2. Die Meldung wird nach Spezifikation signiert und verschlüsselt. 3. Der Transmitter sendet die aufbereitete und signierte Meldung über HTTPS an den Distributor. 4. Der Distributor prüft die Meldung auf Validität und Plausibilität. 5. Der Distributor überprüft, ob der JobKey bekannt ist und antwortet dem Transmitter mit der dem JobKey entsprechenden Statusmeldung. 6. Der Transmitter wartet die Rückantwort des Distributors ab. 7. Das Ergebnis der Job-Verarbeitung wird auf Transmitterseite aufbereitet und angezeigt.
Alternative Abläufe	keine
Fehlerliste	Technische Fehler: • Fehler beim Signieren oder Verschlüsseln. In diesem Fall *sollte* die Übermittlung zu einem späteren Zeitpunkt wiederholt werden, oder, falls das Problem langfristig besteht, Rückfrage beim Hersteller genommen werden. • JobKey ungültig. Dies kann sowohl ein fehlerhafter JobKey sein, als auch ein Ablauf der Gültigkeit. • Der Endreceiver befindet sich in einem geplanten Wartungsfenster. Der Benutzer *sollte* darauf hingewiesen werden, dass nach Ende des Wartungsfensters erneut übermittelt kann. • Der Endreceiver ist nicht erreichbar. Die Übermittlung kann zu einem späteren Zeitpunkt erneut versucht werden. • Der Endreceiver ist nicht adressierbar. Überprüfung der Adressierung seitens Sendersystem/Transmitter.

Tabelle 9.4. UC002 Status abholen

9.2.1 Spezielle Anforderungen

9.2.1.1 Polling

Die Statusabfrage **muss** mehrmals durchgeführt werden können. Sobald das Element <JobFinished> auf ‚true‘ steht, ist die Übermittlung abgeschlossen. Es darf keine weitere Statusabfrage durchgeführt werden können. Die Abfrage des Status kann automatisiert werden. Der Abstand zwischen den einzelnen automatisierten Abfragen **muss** jedoch mindestens zehn Sekunden betragen. Eine erste Statusabfrage *kann* direkt im Anschluss an die Initialmeldung durchgeführt werden. Weitere Statusabfragen **müssen** in Abständen erfolgen, die abhängig von der Grösse der gesendeten Initialmeldung sind. Nach 2-3 Pollingversuchen *kann* der automatische Prozess gestoppt werden. Er **muss** in diesem Fall aber manuell wieder angestossen werden können.

9.2.1.2 Identifikation des Prozesses (nachfolgend DomainID)

Jeder Swissdec Prozess enthält eine eindeutige Identifikation. Diese DomainID oder Fallbezeichnung wird in der Response der GetStatus Operation zurückgegeben.

© Swissdec 2026

Die DomainID dient während der ganzen Phase des Prozesses als Identifizierung und muss deutlich erkennbar angezeigt werden. Wichtig: DomainID ist hier nur eine beschreibende Bezeichnung. Je nach Swissdec Standard trägt diese ID unterschiedliche Bezeichnungen (z.B. DeclarationID, NotificationID).

9.2.1.3 Credentials

In der Antwort sind für Prozesse ohne SUA Signatur die Credentials für das Weiterverarbeiten des Prozessablaufs zu finden.

Ist für eine Operation eine SUA Signatur zwingend, kann auf die Credentials verzichtet werden.

9.2.1.4 ProducerResponseNotifications (Plausibilisierung)

Der Distributor prüft den fachlichen Inhalt auf Plausibilität. Falls das Regelwerk Informationen oder Warnungen findet, werden diese im Element ProducerResponseNotifications zurückgegeben. Diese Notifications müssen für den Endkunden klar ersichtlich sein, so das allfällige fachliche Unklarheiten bereinigt werden können.

9.2.1.5 Plausibilisierungsfehler

Wenn die Daten fehlerhafte Daten enthalten (fachlich) detektiert werden, so werden diese im Error Element zurückgegeben.

9.2.1.6 Wartungsfenster

Befindet sich der Empfänger in einem Wartungsfenster, so wird das Zeitfenster und der Grund der Wartungsarbeiten zurückgegeben. Erst wenn die Zeit des Wartungsfenster vorbei ist, kann eine erneute Übermittlung versucht werden. Möglicherweise gleichzeitig adressierte Endempfänger, die kein Wartungsfenster melden, können normal weiterverarbeitet werden.

9.3 UC003 Testmeldung kennzeichnen

Kurzbeschreibung	Es **muss** möglich sein, die Meldung als Testcase zu kennzeichnen, um Tests bei der Installation oder bei Problemen in der Produktion zu ermöglichen. Der Ablauf einer Testmeldung, sowie die Beschreibung, wie Testdaten gekennzeichnet werden **müssen**, sind im Abschnitt 9.1, „UC001 Initiale Meldung senden“ unter alternative Abläufe zu finden. Das TestCase Flag ist jeweils auf dem Job bei der initialen Meldung oder im CaseContext bei der Synchronisation gesetzt. Die technischen Operationen, wie GetStatus oder ein leeres Synchronize können kein TestCase Flag enthalten.
Akteure	Sendersystem, Distributor, Endreceiver
Auslöser	Das Sendersystem will eine Testmeldung übermitteln (Supportfall, Test, etc.).
Vorbedingungen	Das Sendersystem ist in der Lage, elektronische Swissdec-Meldungen zu versenden und die Antwort zu verarbeiten.
Nachbedingungen	Die Meldung wurde vom Endreceiver empfangen. Sie **darf nicht** produktiv weiterverarbeitet werden. Bei einem Fehlschlag: • Fehlermeldung
Included Use Cases	keine
Standardablauf	1. Das Sendersystem übergibt dem Transmitter Daten, welche als Testfall markiert sind. 2. Der Transmitter bereitet die Meldung als SOAP-Request mit zugehöriger Adressierung (Job) auf.

© Swissdec 2026

	3. Die Meldung wird nach Spezifikation signiert und verschlüsselt. 4. Der Transmitter sendet die aufbereitete und signierte Meldung über HTTPS an den Distributor. 5. Der Distributor prüft die Meldung auf Validität und Plausibilität. 6. Der Distributor bereitet eine oder mehrere Meldungen für die gewählten Endreceiver auf und sendet diese an den/die Endreceiver. 7. Der Endreceiver prüft die Meldung, erkennt sie als Testfall und stoppt die Verarbeitung.
Alternative Abläufe	keine
Fehlerliste	Analog zu Abschnitt 9.1, „UC001 Initiale Meldung senden“ , Abschnitt 9.5, „UC005 Synchronisieren“

Tabelle 9.5. UC003 Testmeldung kennzeichnen

9.4 UC004 Ersatzmeldung kennzeichnen

Kurzbeschreibung	Falls die Meldung unvollständig oder (fachlich) fehlerhaft übermittelt und bereits freigegeben wurde, *muss* es möglich sein, eine neue Übermittlung zu machen und diese als Ersatzmeldung zu kennzeichnen. Der Ablauf wie die Meldung als Ersatzmeldung gekennzeichnet werden muss, ist im Abschnitt 9.1, „UC001 Initiale Meldung senden“ unter alternative Abläufe zu finden. Die Details zur Ersatzmeldung ist in den operationsspezifischen Kapiteln definiert.
Akteure	Sendersystem, Distributor, Endreceiver
Auslöser	Das Sendersystem will eine Ersatzmeldung übermitteln.
Vorbedingungen	Das Sendersystem ist in der Lage, elektronische Swissdec-Meldungen zu versenden und die Antwort zu verarbeiten.
Nachbedingungen	Die Meldung wurde vom Endreceiver empfangen und wird vom Empfängersystem verarbeitet. Bei einem Fehlschlag: • Fehlermeldung
Included Use Cases	keine
Standardablauf	1. Das Sendersystem übergibt dem Transmitter Daten, welche als Ersatzmeldung markiert sind. Die zu ersetzende Meldung wird anhand ihrer DeclarationID ausgewählt. 2. Der Transmitter bereitet die Meldung als SOAP-Request mit zugehöriger Adressierung (Job) auf. 3. Die Meldung wird nach Spezifikation signiert und verschlüsselt. 4. Der Transmitter sendet die aufbereitete und signierte Meldung über HTTPS an den Distributor. 5. Der Distributor prüft die Meldung auf Validität und Plausibilität. 6. Der Distributor bereitet eine oder mehrere Meldungen für die gewählten Endreceiver auf und sendet diese an den/die Endreceiver. Sollte ein Endempfänger diese Meldung als Duplikat erhalten, wird dies vom Distributor entsprechend gekenntzeichnet. 7. Der Endreceiver prüft die Meldung, erkennt sie als Ersatzmeldung und handelt entsprechend.
Alternative Abläufe	keine

© Swissdec 2026

Fehlerliste	Analog zu Abschnitt 9.1, „UC001 Initiale Meldung senden“ , Abschnitt 9.5, „UC005 Synchronisieren“

Tabelle 9.6. UC003 Testmeldung kennzeichnen

9.5 UC005 Synchronisieren

Nach jeder Meldung ist mit dem Aufruf <Synchronize> der Fall zu synchronisieren. Dies kann, muss aber nicht, folgende Elemente beinhalten:

- Abschnitt 9.6, „UC006 Prozesskontrolle durchführen“
- Abschnitt 9.7, „UC007 Fall schliessen“
- Abschnitt 9.8, „UC008 Stories melden und quittieren“
- Abschnitt 9.9, „UC009 Stories abholen und verarbeiten“

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 9.1, „UC001 Initiale Meldung senden“ und eventuell Abschnitt 9.2, „UC002 Status abholen“ wurde erfolgreich durchgeführt • Alle Identifikatoren für den Fall sind bekannt
Nachbedingungen	Der Fall wurde zwischen Sender und Empfänger synchronisiert. Falls die Synchronisierung nicht abgeschlossen ist, ist der Sender darüber in Kenntnis gesetzt worden.
Included Use Cases	Abschnitt 9.6, „UC006 Prozesskontrolle durchführen“ Abschnitt 9.8, „UC008 Stories melden und quittieren“ Abschnitt 9.9, „UC009 Stories abholen und verarbeiten“ Abschnitt 9.15, „UC015 Security anwenden“
Standardablauf	1. Der Akteur wählt die Stories aus, die er übermitteln will und sendet diese dem Endreceiver. Zusätzlich quittiert er den aktuellen State (siehe Abschnitt 9.5.1.2, „Quittierung“ 2. Dieser wertet die Stories aus und meldet zurück, was bei ihm noch an offenen Stories vorhanden ist und/oder was der Akteur von ihm verlangt hat. 3. Der Akteur analysiert die erhaltene Antwort, überprüft die empfangenen Stories und pflegt alle relevanten Daten ins Sendersystem ein.
Alternative Abläufe	1. Der Akteur wählt keinen Fall und keine Stories aus, die er übermitteln will und sendet ein leeres Synchronize. 2. Der Endreceiver analysiert alle offenen Fälle dieses Vertrages und sammelt die DomainIDs im Available Element. 3. Der Akteur analysiert die erhaltene Antwort, und synchronisiert die Fälle aus dem Available mittels Standardablauf.
Fehlerliste	Fachliche Fehler: • Die Meldung verstösst gegen die Plausibilisierungsregeln • Es wird kein Fall mit dem zu synchronisierenden Fall gefunden • Eine Story wurde nicht quittiert. Die Story muss erneut gesendet werden.

© Swissdec 2026

	Technische Fehler: • Fehler beim Signieren oder Verschlüsseln • Der Endreceiver befindet sich in einem geplanten Wartungsfenster • Der Endreceiver ist nicht erreichbar • Der Endreceiver ist nicht adressierbar • Die vom Sendersystem aufbereitete Meldung entspricht nicht dem Schema (Validität nicht gegeben) • Die vom Sendersystem aufbereitete Meldung ist nicht identifizierbar (ungültige Identifikatoren)

Tabelle 9.7. UC005 Synchronisieren

9.5.1 Spezielle Anforderungen

9.5.1.1 Unique StoryID

Die StoryIDs müssen Fallübergreifend eindeutig sein.

9.5.1.2 Quittierung

Bei jeder Synchronisierung **muss** der bei der letzten Synchronisierung erhaltene State quittiert werden. Dies geschieht dadurch, dass der in der vorhergehenden Response enthaltene State im Feld <ReceivedState> zurückgegeben wird. Ausnahme: Beim Abschluss des Prozesses (State=Finished) *kann* diese Quittierung des States weggelassen werden.

9.5.1.3 Available (Neue Informationen verfügbar)

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

Sind Available Elemente vorhanden muss der Sender diese synchronisieren und seine offenen Prozesse aktualisieren.

9.5.1.4 OpenCase

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

Anschliessend wird der effektive Case mittels dieser Identifikation synchronisiert.

9.5.1.5 Leere Synchronisierung

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

9.5.1.6 Maximale Synchronisierungsfrequenz

Distributor und Endempfänger verarbeiten eine grosse Anzahl paralleler Fälle verschiedener Sendersysteme; zu häufige Anfragen belasten das Gesamtsystem und verzögern die Verarbeitung für alle Beteiligten. Zwischen zwei Synchronisierungen desselben Falls beim selben Endempfänger **muss** deshalb ein Abstand von mindestens zehn Sekunden liegen. Während eines geplanten Wartungsfensters des Endempfängers (`<PlannedMaintenance>`) **dürfen keine** automatisierten Synchronisierungen ausgeführt werden.

Best Practice: Solange ein Fall nicht abgeschlossen ist, *sollen* die Abstände zwischen automatisierten Synchronisierungen schrittweise erhöht werden, z. B.:

© Swissdec 2026

- 3 × im Abstand von 10 Sekunden
- 3 × im Abstand von 2 Minuten
- 3 × im Abstand von 10 Minuten
- 3 × im Abstand von 30 Minuten
- 3 × im Abstand von 1 Stunde
- danach alle 8 Stunden

9.6 UC006 Prozesskontrolle durchführen

Kurzbeschreibung	Bei jeder Synchronisierung **muss** das Sendersystem 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? Das Sendersystem muss dem Benutzer aufzeigen, welche Stories er dem Endreceiver liefern muss und bis wann, falls unter der entsprechenden Story eine Frist gesetzt wurde. Ebenfalls **muss** dem Benutzer klar aufgezeigt werden, für welchen Fall noch Stories abgeholt werden müssen.
Akteure	Sendersystem, Distributor, Endreceiver
Auslöser	Das Sendersystem führt eine Synchronisierung durch.
Vorbedingungen	Das Sendersystem hat mindestens eine Meldung gemacht, die nun synchronisiert werden kann.
Nachbedingungen	Das Sendersystem informiert den User klar und verständlich darüber, welche Schritte als Nächstes auszuführen sind. Bei einem Fehlschlag: • Fehlermeldung
Included Use Cases	keine
Standardablauf	1. Das Sendersystem führt eine Synchronisierung durch (siehe Abschnitt 9.5, „UC005 Synchronisieren“). 2. Der Transmitter überprüft, ob in der Response AwaitStory-Elemente enthalten sind. **Achtung!** Diese Funktion existiert nicht in allen Swissdec-Standards. 3. Der Transmitter überprüft, ob in der Response Available-Elemente enthalten sind. **Achtung!** Diese Funktion existiert nicht in allen Swissdec-Standards. 4. Dem Benutzer wird dargestellt, ob bestimmte Stories abgeholt oder geliefert werden müssen.
Alternative Abläufe	keine

© Swissdec 2026

Fehlerliste	Analog zu Abschnitt 9.1, „UC001 Initiale Meldung senden“ , Abschnitt 9.5, „UC005 Synchronisieren“

Tabelle 9.8. UC006 Prozesskontrolle durchführen

9.7 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 Sendersystem ist ein Fall abgeschlossen.
Vorbedingungen	• Es besteht ein Fall, der Sendersystem und Endreceiver betrifft. • Der Endreceiver hat keine Stories mehr zu liefern (`<Available>`) • Der Endreceiver erwartet keine Stories mehr (`<AwaitStory>`)
Nachbedingungen	• Der Fall wurde von beiden Seiten (Sendersystem und Endempfänger) geschlossen. • Der Fall wurde vom Sendersystem archiviert.
Included Use Cases	Abschnitt 9.5, „UC005 Synchronisieren“ , Abschnitt 9.6, „UC006 Prozesskontrolle durchführen“
Standardablauf	1. Das Sendersystem synchronisiert den zu schliessenden Fall. 2. Das Sendersystem bearbeitet die Antwort vom Endreceiver gemäss dem normalen Ablauf (Abschnitt 9.9, „UC009 Stories abholen und verarbeiten“). 3. Das Sendersystem entscheidet anhand standardspezifischer Informationen, ob der Fall geschlossen werden kann. Dies wird in den Kapiteln zu den standardspezifischen Operationen genauer beschrieben. 4. Das Sendersystem archiviert den Fall.
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 9.5, „UC005 Synchronisieren“)
Fehlerliste	Technische Fehler gemäss Abschnitt 9.5, „UC005 Synchronisieren“ Fachliche Fehler gemäss den Richtlinien

Tabelle 9.9. UC007 Fall schliessen

9.8 UC008 Stories melden 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.
Vorbedingungen	• Der initiale Fall wurde dem Endreceiver erfolgreich gemeldet. • Identifikatoren sind bekannt und korrekt gesetzt.

© Swissdec 2026

Nachbedingungen	• Der Endreceiver hat die Stories empfangen und quittiert.
Included UCs	Abschnitt 9.5, „UC005 Synchronisieren“ , Abschnitt 9.6, „UC006 Prozesskontrolle durchführen“
Standardablauf	1. Im Sendersystem wird ausgewählt, welche Stories gemeldet werden sollen. 2. Im Element <ReceivedStoryIDs> im <CaseContext> der vorangehenden Synchronize Operation quittiert der Endreceiver aller erhaltenen Stories. Wurden vom Transmitter Stories versandt und diese werden nicht quittiert, **müssen** diese erneut gesendet werden. 3. Die StoryIDs aller vom Endreceiver empfangenen Stories (vom aktuellen Fall), werden im Element <ReceivedStoryIDs> im <CaseContext> eingetragen. 4. Es wird ein SynchronizeRequest aufbereitet, der die ausgewählten Informationen beinhaltet. 5. Der Fall wird synchronisiert (Abschnitt 9.5, „UC005 Synchronisieren“)
Alternative Abläufe	1. **Keine Stories zu quittieren** Es werden nur neue Stories gemeldet, da keine erhaltenen Stories zu quittieren sind. 2. **Keine Stories zu melden** Das Sendersystem quittiert nur erhaltene Stories, hat dabei aber keine neuen Stories zu melden. Der Request enthält nur einen aktualisierten CaseContext aber keine weiteren Stories.
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 Synchronize 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 Synchronize 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 der Response können Info und Warning Notification vorhanden sein. Diese können mit der StoryID versehen werden, so dass der Akteur informiert werden kann, welche Story Probleme bereitet.

Tabelle 9.10. UC008 Stories melden und quittieren

9.9 UC009 Stories abholen und verarbeiten

Kurzbeschreibung	Das Sendersystem holt Stories ab, die beim Endreceiver bereitliegen und verarbeitet sie im System.

© Swissdec 2026

Akteure	Sendersystem, Distributor, Endreceiver
Auslöser	Dem Sendersystem stehen Informationen zur Verfügung, die beim Endempfänger abgeholt werden **müssen**.
Vorbedingungen	• Der Fall wurde dem Endreceiver erfolgreich gemeldet. • Identifikatoren sind bekannt und korrekt gesetzt. • Der Endreceiver hat dem Akteur mitgeteilt, dass Stories zur Abholung bereitliegen (`Available`).
Nachbedingungen	• Die vom Endreceiver bereitgestellten Stories wurden abgeholt • Die erhaltenen Informationen wurden ins System eingepflegt
Included UCs	Abschnitt 9.5, „UC005 Synchronisieren“ , Abschnitt 9.6, „UC006 Prozesskontrolle durchführen“
Standardablauf	1. Der Akteur synchronisiert den Fall, von dem der Endreceiver mitgeteilt hat, dass noch Stories abzuholen sind. (Abschnitt 9.5, „UC005 Synchronisieren“) 2. Der Akteur liest aus der `SynchronizeResponse` die erhaltenen Stories aus und pflegt sie ins Sendersystem ein. 3. Im Element <CaseContext>/<SuppressedSenderStoryIDs> sind StoryIDs vorhanden: Die Storys mit diesen IDs können vom Endreceiver nicht empfangen werden. (z.B: Mapping) Die IDs **müssen** für jeden folgenden Synchronize des aktuellen Prozesses versandt werden. 4. Im Element <CaseContext>/<SuppressedInstitutionStoryIDs> sind StoryIDs vorhanden: Die Storys mit diesen IDs können vom Transmitter nicht empfangen werden (z.B: Mapping). Die IDs **müssen** für jeden folgenden Synchronize des aktuellen Prozesses versandt werden, so wird dem Endreceiver signalisiert, dass der Transmitter diese Story nicht unterstützt.
Fehlerliste	Fachliche Fehler: • Es werden keine Stories zurückgegeben. • Die Stories können nicht verarbeitet werden. • Die vom Sendersystem angeforderte Story-ID ist ungültig Technische Fehler: • Es gibt Probleme bei der Übermittlung. Die Übermittlung *kann* wiederholt werden.

Tabelle 9.11. UC009 Stories abholen und verarbeiten

9.10 UC010 Datenflusskontrolle durchführen

Die Datenflusskontrolle soll verhindern, dass es zu einer Überlastung der betroffenen Systeme der Übermittlung kommt.

Der Endreceiver kann von sich aus beeinflussen, wie viele Informationen er innerhalb einer Response senden will, um die Auslastung seines Systems zu optimieren. Um die Infrastruktur eines schwächeren Sendersystems allerdings nicht zu überlasten, sind diesem die Möglichkeiten geboten, den Umfang einer Response einzuschränken.

© Swissdec 2026

Der Endreceiver teilt dem Sendersystem jeweils mit, zu welchen Fällen er noch Informationen zu liefern hat (`<Available>`). Das Sendersystem ist jedoch nicht verpflichtet, diese alle innerhalb des nächsten Requests abzurufen. Vielmehr besteht die Möglichkeit, den Umfang des Datenflusses zu kontrollieren, indem nur ein oder wenige Fälle zur Synchronisation ausgewählt werden. Der Endreceiver reagiert darauf, indem er die Antworten zu diesen Fällen liefert und für jene, die nicht abgefragt wurden, weiterhin mit `<Available>` reagiert.

Der Transmitter muss dem Benutzer die Möglichkeit geben, aus der Liste der verfügbaren Fälle jene auszuwählen, die bei der nächsten Synchronisation abgefragt werden sollen.

9.11 UC011 Completion aufrufen

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
Nachbedingungen	• Ein Browserfenster mit eingetragenen Feldern Key und Passwort wurde geöffnet. • Bei einem Fehlschlag: Webseite nicht erreichbar: Fehlermeldung
Included UCs	-
Standardablauf	1. Der Akteur wählt den Endreceiver aus, bei welchem eine Completion durchgeführt werden soll. 2. Das Sendersystem stellt eine URL mit den Zusatzparametern Key und Passwort zusammen. 3. Das Sendersystem öffnet ein Browserfenster mit der zusammengestellten URL.
Alternative Abläufe	Daten aus Story nicht vorhanden {nach Schritt 2} 3. Fehlermeldung {Ende} 1. Der Akteur wählt den Endreceiver aus, bei welchem eine Completion durchgeführt werden soll. 2. Das Sendersystem stellt eine URL mit den Zusatzparametern Key und Passwort sowie weiteren Parametern zusammen. 3. Das Sendersystem öffnet ein Browserfenster mit der zusammengestellten URL. Alternativer Ablauf: DialogMessage Anstelle einer Completion-URL erhält das Sendersystem die Information, dass die Completion per DialogMessage durchgeführt werden soll (siehe Abschnitt 9.12, „UC012 Dialog Message abwickeln“).
Fehlerliste	Fehler: • Daten aus Story nicht vorhanden

Tabelle 9.12. UC011 Completion aufrufen

9.11.1 Spezielle Anforderungen

9.11.1.1 Aufbau und Verarbeitung der Completion URL

Anhang E, *Detailspezifikation Completion URL*

© Swissdec 2026

9.12 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 Akteur wählt den Endreceiver aus, bei welchem eine DialogMessage abgewickelt werden soll. 2. Der Endreceiver sendet mit der <DialogMessage> Story die zu ergänzenden Daten. 3. Der Sachbearbeiter ergänzt die geforderten Angaben und sendet diese mit der <DialogMessage> Story mittels <Synchronize> an den Endreceiver. 4. Der Endreceiver bestätigt mit <ReceivedStoryIDs> den Erhalt der DialogMessage.
Alternative Abläufe	**Daten aus Statusmeldung nicht vorhanden** {nach Schritt 1} 2. Fehlermeldung {Ende} **DialogMessage nicht abgeschlossen** {nach Schritt 4} 1. Der Endreceiver bestätigt mit <ReceivedStoryIDs> den Erhalt der DialogMessage und verlangt weitere Ergänzungsangaben. {Weiter mit Schritt 4} **DialogMessage nicht quittiert** {nach Schritt 3} 1. Der Endreceiver bestätigt den Erhalt der DialogMessage nicht. Die Story wird noch einmal vom Transmitter an den Empfänger geschickt. {Weiter mit Schritt 3}
Fehlerliste	Fehler: • Daten aus Statusmeldung nicht vorhanden

Tabelle 9.13. UC012 Dialog Message abwickeln

9.13 UC013 Anmeldungsprozess (SubscribeOrganization)

Kurzbeschreibung	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.
Akteure	Sachbearbeiter, ERP-System
Auslöser	Der Akteur möchte die Kommunikation via Swissdec bei einem Empfänger anmelden.

© Swissdec 2026

Vorbedingungen	• Das Sendersystem ist in der Lage, elektronische Swissdec-Meldungen zu versenden und die Antwort zu verarbeiten. • Der Empfänger ist korrekt konfiguriert.
Nachbedingungen	Die Anmeldung wurde vom Endreceiver empfangen und durch eine Rückantwort quittiert.
Included UCs	Abschnitt 9.1, „UC001 Initiale Meldung senden“ , Abschnitt 9.2, „UC002 Status abholen“ , Abschnitt 9.15, „UC015 Security anwenden“
Standardablauf	1. Das Sendersystem übergibt dem Transmitter die Daten mit den Empfängeradressen. 2. Der Transmitter bereitet die Meldung als SOAP-Request mit zugehöriger Adressierung (Job) auf. 3. Die Meldung wird nach Spezifikation signiert und verschlüsselt. 4. Der Transmitter sendet die aufbereitete und signierte Meldung als synchronen Aufruf über HTTPS an den Distributor (siehe Abschnitt 9.1, „UC001 Initiale Meldung senden“ . 5. Der Distributor prüft die Meldung auf Validität und Plausibilität. 6. Der Distributor bereitet eine oder mehrere Meldungen für die gewählten Endreceiver auf und sendet diese an den/die Endreceiver. 7. Der Endreceiver prüft die Meldung und startet die Verarbeitung des Jobs. 8. Der Endreceiver meldet seine Antwort dem Distributor, der sie dem Sendersystem zukommen lässt. 9. Der Transmitter wartet die Rückantwort des Distributors ab. 10 Das Ergebnis der Übermittlung wird auf Transmitterseite aufbereitet und angezeigt.
Alternative Abläufe	-
Fehlerliste	Fachliche Fehler: • Die Meldung verstösst gegen die Plausibilisierungsregeln • Vertrag unbekannt. Technische Fehler: • Fehler beim Signieren oder Verschlüsseln. In diesem Fall *sollte* die Übermittlung zu einem späteren Zeitpunkt wiederholt werden, oder, falls das Problem langfristig besteht, Rückfrage beim Hersteller genommen werden. • Der Endreceiver befindet sich in einem geplanten Wartungsfenster (nur bei synchroner Übermittlung möglich). Der Benutzer *sollte* darauf hingewiesen werden, dass nach Ende des Wartungsfensters erneut übermittelt kann. • Der Endreceiver ist nicht erreichbar (nur bei synchroner Übermittlung möglich). Die Übermittlung kann zu einem späteren Zeitpunkt erneut versucht werden. • Der Endreceiver ist nicht adressierbar (nur bei synchroner Übermittlung möglich). Überprüfung der Adressierung seitens Sendersystem/Transmitter. • Die vom Sendersystem aufbereitete Meldung entspricht nicht dem Schema (Validität nicht gegeben)

Tabelle 9.14. UC013 Anmeldungsprozess (SubscribeOrganization)

9.14 UC014 Unternehmensauthentifizierung (RegisterOrganization)

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. Weitere Informationen zu SUA finden sich unter Anhang C, *Detailspezifikation Swissdec Unternehmens-Authentifizierung SUA* .

© Swissdec 2026

9.15 UC015 Security anwenden

Ausser dem Erreichbarkeitstest *muss* jede Übermittlung signiert und verschlüsselt werden. Einzelheiten dazu finden sich in den Dokumenten zur Sicherheit auf Transmitterseite [[SECPDF]] , sowie in den zusätzlichen Informationen zur doppelten Signierung (SUA) Anhang C, *Detailspezifikation Swissdec Unternehmens-Authentifizierung SUA* .

9.16 UC016 Transmitter konfigurieren

Das Sendersystem ist für die korrekte Adressierung der Endreceiver im Transmitter verantwortlich. Hierbei handelt es sich um die Pflege und regelmässige Aktualisierung der Profile mit gültiger Identifikation der einzelnen Institutionen.

Die jeweils gültigen Profile können direkt beim Endempfänger bezogen werden. Jedoch sind die an den Distributor gekoppelten Institutionen ebenfalls auf der Webseite der Swissdec (https://www.swissdec.ch) zum Download bereit.

Auf welche Weise und wie regelmässig die Profile aktualisiert werden, liegt in der Verantwortung des Sendersystem-Herstellers oder des Endbenutzers.

9.17 UC017 Supportinformationen anzeigen

Kurzbeschreibung	Fehler, Warnungen und Informationen gemäss [[ACKNOTIF]] **müssen** ausgewertet und dem Benutzer angezeigt werden. IDs müssen verwendet werden.
Akteure	ERP, Transmitter, Distributor
Auslöser	Eine Meldung oder eine Anfrage wurde via Distributor an einen Endreceiver gesendet. Die Antwort wird via Distributor empfangen.
Vorbedingungen	Distributor sendet eine Antwort
Nachbedingungen	• Fehler, Warnungen und Informationen aus der Antwort werden aufbereitet und dem Benutzer angezeigt. • Bei einem Fehlschlag: Distributor nicht erreichbar: Fehlermeldung
Included UCs	Abschnitt 9.13, „UC013 Anmeldungsprozess (SubscribeOrganization)“
Standardablauf	1. Das ERP übergibt dem Transmitter die Daten für eine Meldung. 2. Der Transmitter bereitet die Meldung als SOAP-Request mit zugehöriger Adressierung (Job) und ReqestID auf. 3. Die Meldung wird mit dem privaten Schlüssel/Zertifikat des Herstellers nach Spezifikation [[SECPDF]] signiert. 4. Der Transmitter sendet die aufbereitete und signierte Meldung über HTTPS an den Distributor. 5. Der Distributor prüft die Meldung auf Validität und Plausibilität. 6. Der Distributor bereitet die Meldung für die gewählten Endempfänger inklusive einer Identifikation des Geschäftsvorfalls auf und sendet sie an den Endreceiver. 7. Der Endreceiver prüft die Meldung. Eine Antwort wird generiert und an den Distributor gesendet. 8. Der Distributor sendet eine Antwort inklusive einer Identifikation des Geschäftsvorfalls an den Transmitter. 9. Der Transmitter wertet die Antwort aus und stellt sie für das ERP bereit. 10. Fehler, Warnungen und Informationen werden aufbereitet und angezeigt. Die IDs werden gesichert.
Alternative Abläufe	-
Fehlerliste	Technische Fehler:

© Swissdec 2026

	• Fehler beim Signieren • Fehler beim ver-/ entschlüsseln • der Distributor ist nicht erreichbar • die vom ERP aufbereitete Meldung entspricht nicht dem Schema (Validität nicht gegeben)

Tabelle 9.15. UC017 Supportinformationen anzeigen

9.17.1 Spezielle Anforderungen

9.17.1.1 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 Transmitter 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.

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.

9.17.1.2 Gleiche Fehlermeldungen und Informationen von verschiedenen Endempfängern darstellen

Bei gleichen Fehlermeldungen und Warnungen (Notifications) von verschiedenen Endempfängern muss für den Benutzer nachvollziehbar dargestellt sein, von welchen Empfängern welche Meldungen stammen. Ziel dieser Sicht ist es, den Support durch die Endempfänger zu erleichtern. Die Quittung eines Endempfängers bietet dem Endanwender und dem Support hierbei die gleichen Informationen.

Die elektronische Übermittlung ist ein dynamisches System, bei welchem die Antworten automatisch generiert werden, vergleiche [ACKNOTIF]. Einige Fehlermeldungen werden dabei von unterschiedlichen Endempfängern gleich lauten. Ist beispielsweise die Plausibilisierungsregel „Alter muss kleiner als 100 Jahre sein“ bei einer Person verletzt, so wird diese Fehlermeldung von allen Endempfängern kommen, bei welchen die Daten dieser Person verarbeitet wurden. Daher können neben der Forderung in Abschnitt 9.17.1.2, „Gleiche Fehlermeldungen und Informationen von verschiedenen Endempfängern darstellen“ [66] zusätzliche Views, welche das Handling für den Endbenutzer erleichtern (Sortierungen, Filterung der redundanten Meldungen), erstellt werden.

9.18 UC018 Erreichbarkeit prüfen

Kurzbeschreibung	Die Erreichbarkeit des Distributors **muss** geprüft werden. Dazu wird eine einfache Anfrage an den Distributor geschickt. Die Rückantwort des Distributors bestätigt die Erreichbarkeit.

© Swissdec 2026

Akteure	Sachbearbeiter, Distributor
Auslöser	Die Erreichbarkeit des Distributors soll geprüft werden.
Vorbedingungen	-
Nachbedingungen	• Die Rückantwort des Distributors enthält einen Timestamp mit der Systemzeit des Distributors. Bei einem Fehlschlag: • Distributor nicht erreichbar: Fehlermeldung • Inhalt verschieden [[ACKNOTIF]] : Fehlermeldung
Included UCs	-
Standardablauf	1. Der Akteur löst die Überprüfung aus. 2. Der Transmitter sendet eine einfache Serveranfrage (Ping) an die Zieladresse des Distributors. 3. Der Transmitter wertet die Rückantwort des Distributors aus [[ACKNOTIF]] .
Alternative Abläufe	**Distributor nicht erreichbar** {nach Schritt 1} 2. Fehlermeldung {Ende}
Fehlerliste	Fehler: • der Distributor ist nicht erreichbar • der Distributor sendet eine falsche Antwort

Tabelle 9.16. UC018 Erreichbarkeit prüfen

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

9.19 UC019 Interoperabilität prüfen

Kurzbeschreibung	Damit die Interoperabilität zwischen einem Transmitter und dem Distributor überprüft werden kann, **muss** der Transmitter einen „CheckInteroperability-Request“ absetzen können.
Akteure	System-Tester, Distributor
Auslöser	Installation *soll* getestet werden
Vorbedingungen	Abschnitt 9.19.1.1, „Vorbedingungen“
Nachbedingungen	Abschnitt 9.19.1.1, „Vorbedingungen“ Bei einem Fehlschlag: • Distributor nicht erreichbar: Fehlermeldung • Inhalt verschieden [[ACKNOTIF]] : Fehlermeldung
Included UCs	Abschnitt 9.13, „UC013 Anmeldungsprozess (SubscribeOrganization)“
Standardablauf	1. Der Akteur startet die Interoperabilitätsprüfung und gibt Werte für Operand 2 ein. 2. Der Akteur löst das Senden der Daten aus. 3. Der Transmitter bereitet die Serveranfrage auf. 4. Die Meldung wird mit dem privaten Schlüssel/Zertifikat des Herstellers nach Spezifikation [[SECPDF]] signiert. 5. Die Meldung wird mit einem elektronischen Verschlüsselungsverfahren wie in [[SECPDF]] beschrieben verschlüsselt, die Antwort entschlüsselt.

© Swissdec 2026

	6. Der Transmitter sendet die Serveranfrage über HTTPS an den Distributor. 7. Der Distributor bearbeitet die gesendeten Daten (Transformation Umlautstring, Berechnung „FirstOperand +- SecondOperand“) und schickt die Antwort an den Transmitter. 8. Der Transmitter wertet die Antwort des Distributors aus [[ACKNOTIF]] . 9. Der Transmitter zeigt die Antwort des Distributors an.
Alternative Abläufe	**Meldung signieren, Sender** {nach Schritt 4} b) Die Meldung wird mit dem privaten Schlüssel/Zertifikat des Sender nach Spezifikation [[SECPDF]] ein zweites Mal signiert. {weiter mit Schritt 5} **Distributor nicht erreichbar** {nach Schritt 5} 6. Fehlermeldung {Ende}
Fehlerliste	Fachliche Fehler: • Interoperabilität ist nicht gegeben Technische Fehler: • Fehler beim Signieren • Fehler beim ver-/ entschlüsseln • der Distributor ist nicht erreichbar siehe [[ACKNOTIF]]

Tabelle 9.17. UC019 Interoperabilität prüfen

9.19.1 Spezielle Anforderungen

Der Interoperabilitätstest wird zu Entwicklungszwecken und bei der Installation verwendet, um die Interoperabilität zwischen einem Transmitter und dem Distributor zu gewährleisten. Die grössten zu erwartenden Schwierigkeiten liegen dabei in den Bereichen Codierung von Zeichenketten (Encoding) und Interpretation von Fliesskommazahlen. Beide Systeme (Transmitter und Distributor) müssen dabei bestimmte Auswertungen vornehmen, um bei einem eventuellen Fehler auf den Verursacher schliessen zu können.

9.19.1.1 Vorbedingungen

Der Transmitter sendet folgende Daten:

Parametername	Wert	Bemerkung
UmlautString	ÄËÖÜÁÉÓÚÀÈÒÙÂÊÔÛ	fester Wert
FirstOperand	999000000000.00	fester Wert, 999 Milliarden
SecondOperand	keine Vorgabe	beliebige Fliesskommazahl
SystemDateTime	Datum und Zeit des Transmitters	Systemdatum und –zeit

Tabelle 9.18. Vorbedingungen (Transmitter)

© Swissdec 2026

9.19.1.2 Nachbedingungen

Auswertung und Antwort des Distributors:

Parametername	Auswertung / Berechnung	Bemerkung
UmlautStringIsCorrect	UmlautStringTRANS = ÄËÖÜÁÉÓÚÀÈÒÙÂÊÔÛ	Rückgabe: true / false
FirstOperandIsCorrect	999000000000.00	Rückgabe: true / false
UmlautString	äëöüáéóúàèòùâêôû	Rückgabe: UmlautStringDISTRI Gross-zu Kleinbuchstaben.
AdditionResult	AdditionResultDISTRI = FirstOperandTRANS + SecondOperandTRANS	Rückgabe: berechneter Wert AdditionResultDISTRI
SubstractionResult	AddiSubstractionResultDISTRI = FirstOperandTRANS - SecondOperandTRANS	Rückgabe: berechneter Wert SubstractionResultDISTRI
SystemDateTime	Datum und Zeit des Distributors	Rückgabe: Systemdatum und –zeit

Tabelle 9.19. Auswertung und Antwort Distributor

Auswertung des Transmitters:

Parametername	Auswertung / Berechnung	Bemerkung
UmlautStringIsCorrect	UmlautStringIsCorrect = true	*muss* true sein
FirstOperandIsCorrect	FirstOperandIsCorrect = true	*muss* true sein
UmlautString	äëöüáéóúàèòùâêôû	*muss* äëöüáéóúàèòùâêôû sein
AdditionResult	FirstOperandTRANS + SecondOperandTRANS = AdditionResultDISTRI	Berechnung und Vergleich, Genauigkeitsgrad 2 Nachkommastellen
SubstractionResult	FirstOperandTRANS – SecondOperandTRANS = AdditionResultDISTRI	Berechnung und Vergleich, Genauigkeitsgrad 2 Nachkommastellen
SystemDateTime	| SystemDateTimeDISTRI – SystemDateTimeTRANS| < 1 min	Betrag Zeitdifferenz sollte < 1 Minute sein

Tabelle 9.20. Auswertung Transmitter

© 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.

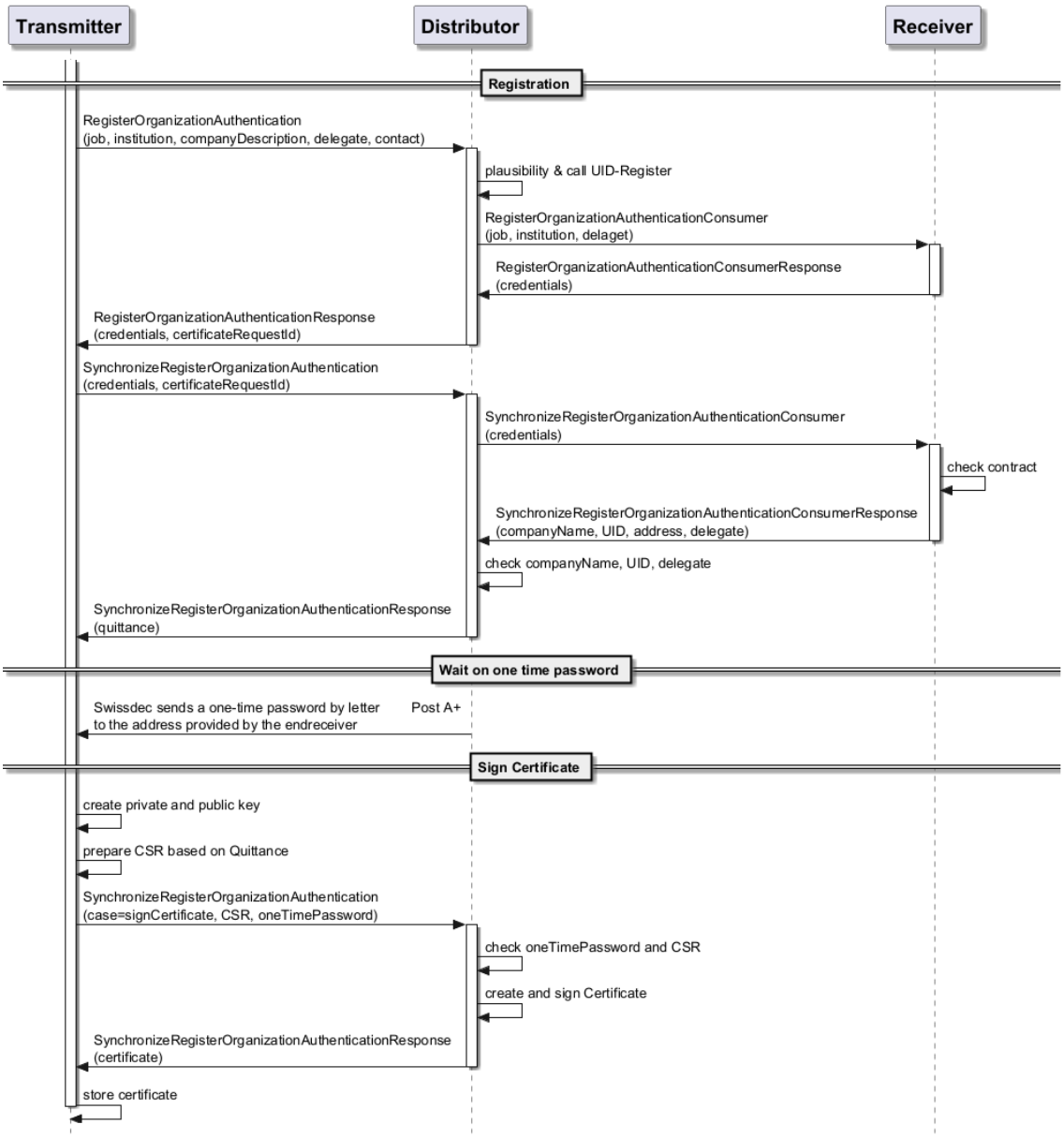
Der Distributor überprüft die Stammdaten des Unternehmens und vergleicht diese mit den Angaben aus dem UID-Register. Das Unternehmen muss gemäss UID-Register zu diesem Zeitpunkt aktiv ist. Zusätzlich werden die Angaben zur Vertragsverbindung dem adressierten Versicherer übermittelt. Der Versicherer sendet 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.

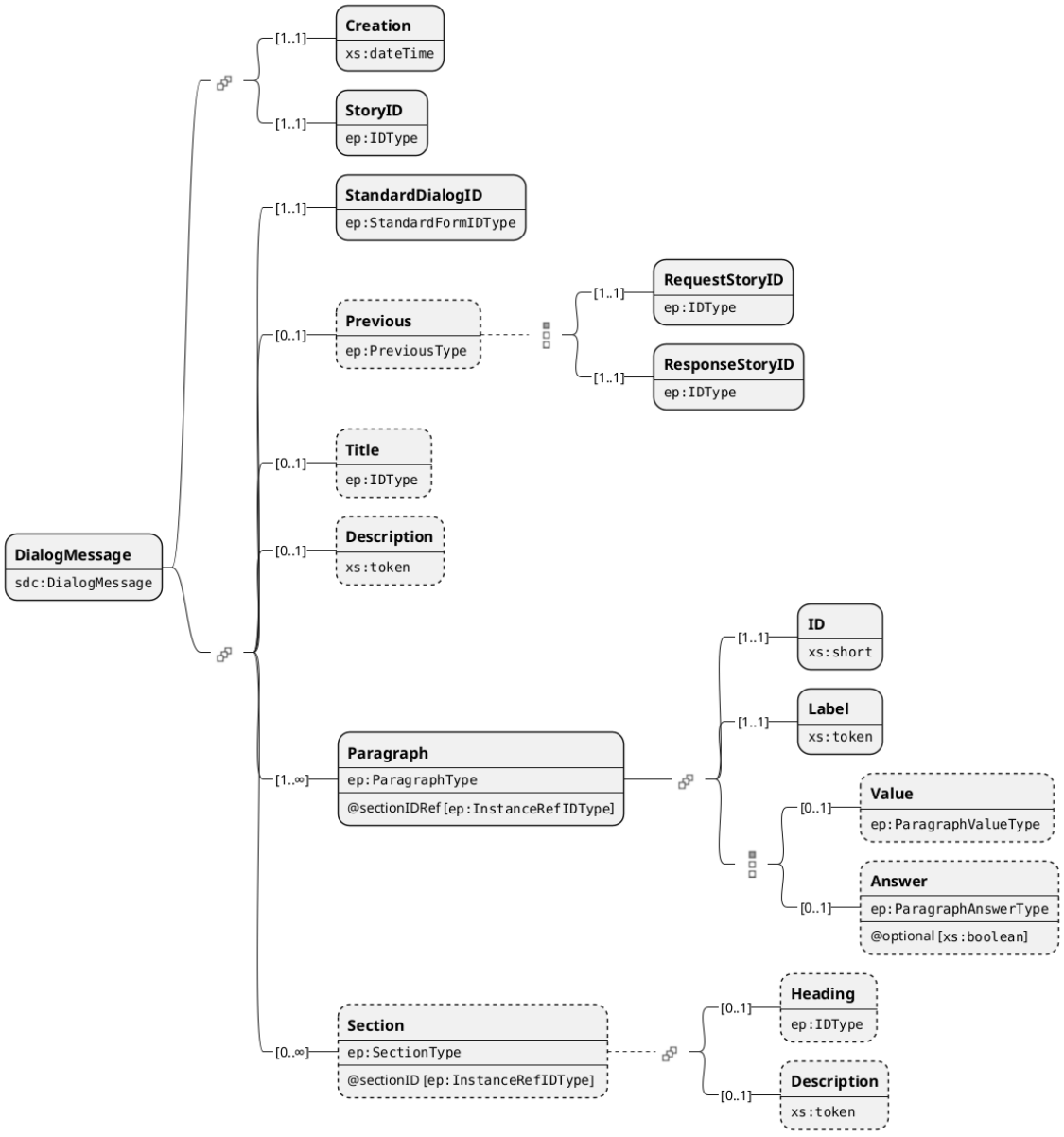


Abbildung D.1. DialogMessage XML-Schema Struktur

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

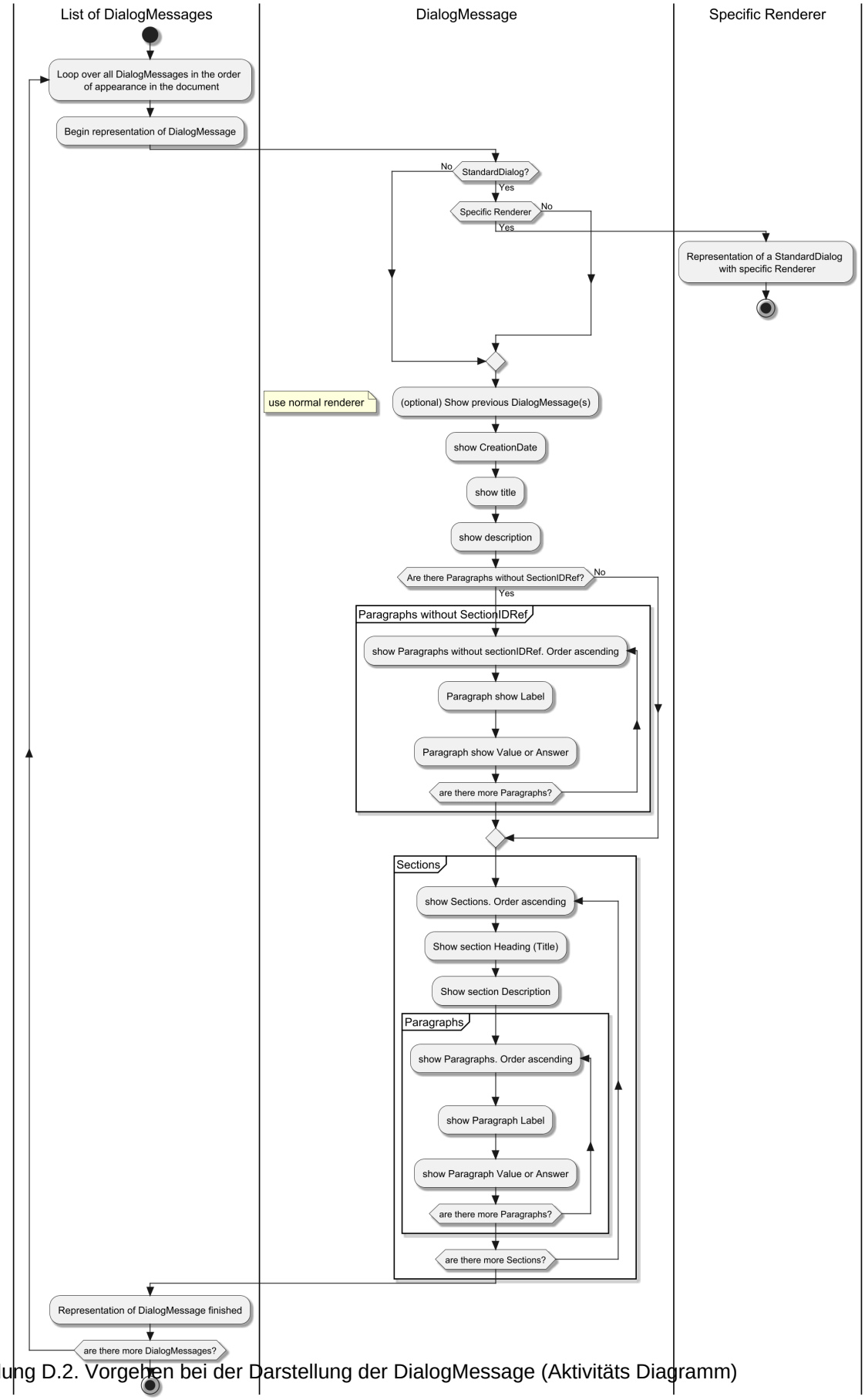


Abbildung D.2. Vorgehen bei der Darstellung der DialogMessage (Aktivitäts Diagramm)

© 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

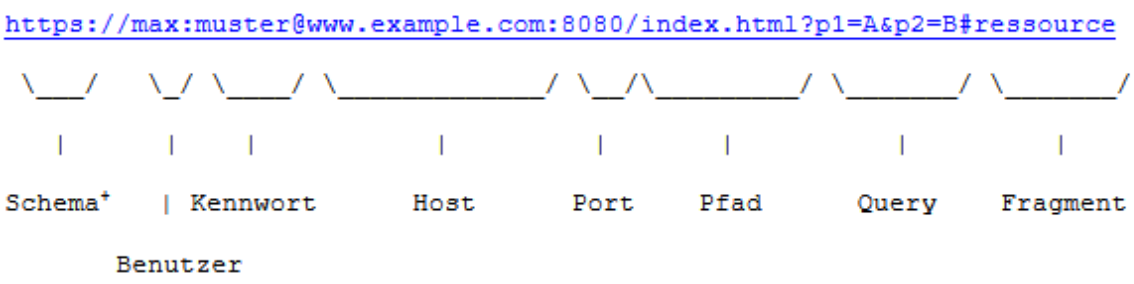


Abbildung E.1. Aufbau URL gemäss RFC3986

© 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 Sendersystem und Distributor
Akteure	Sendersystem
Fachliche Anforderung	Die Erreichbarkeit des Distributors muss geprüft werden. Dazu wird eine einfache Anfrage an den Distributor geschickt. Die Rückantwort des Distributors bestätigt die Erreichbarkeit
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Der Akteur löst die Überprüfung aus 2. Das Sendersystem sendet eine einfache Serveranfrage (Ping) an die Zieladresse des Distributors. 3. Das Sendersystem wertet die Rückantwort des Distributors aus.
Ergebnis	Die Rückantwort des Distributors enthält einen Timestamp mit der Systemzeit des Distributors oder eine Fehlermeldung
Fehlerbehandlung	1. Distributor 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)

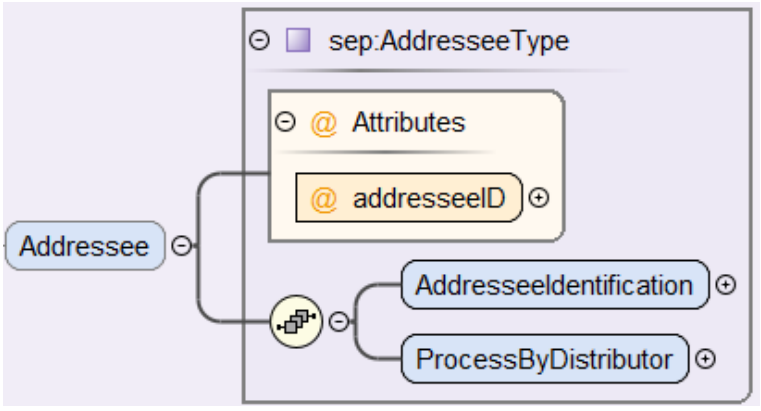
AB-05: Explizite Adressierung eines Empfängers (direkte Adressierung)	
Kurzbeschreibung	Explizite Adressierung zur Datenübermittlung
Akteure	Sendersystem, Distributor
Fachliche Anforderung	Rahmenbedingung aus [[UpgrDistri_Anforderungen]]: Die Lohndaten werden sicher und direkt an die vom Kunden (Unternehmen) 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 Distributor ist nicht erreichbar. Fehlermeldung. 2. Ein oder mehrere Adressaten sind nicht erreichbar/unbekannt. Verifizieren der Adressierung und erneute Übermittlung.
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)

© Swissdec 2026

F.6 AB-06 Implizite Adressierung (Gruppenadressierung)

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

© Swissdec 2026

AB-07 Absichtserklärung	
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.
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. Das Sendersystem muss dem Benutzer die benötigten IDs anzeigen. Dies darf abhängig vom Benutzerstatus sein (Endbenutzer, tech. Support, etc.) 3. 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

© Swissdec 2026

AB-09 Ersatzmeldung	
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. 5. Das Sendersystem sendet die Ersatzmeldung an den Distributor. 6. Das Sendersystem wertet die Rückantwort des Distributors aus.
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	keine

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

© Swissdec 2026

AB-11 Testmeldung	
Fachliche Anforderung	Es müssen Testdaten übermittelt werden können, die durch ein Element „Test-Case“ gekennzeichnet sind. Dabei werden folgende Ziele verfolgt: • Ermöglichung von Tests bei der Installation und Konfiguration von produktiven Anwendungen • 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.

© Swissdec 2026

AB-12 Datenqualitätssicherung	
	• mehr Automatisation und weniger manuelle Arbeiten • geringere Durchlaufzeiten Die Prüfungsstufen befinden sich auf folgenden Systemteilen: • 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.

© Swissdec 2026

AB-13 Filterung	
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.
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 Sendersystem
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.

© Swissdec 2026

AB-15 Dublettenerkennung	
Technische Implikationen	keine

Tabelle F.15. AB-15 Dublettenerkennung

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	Umsetzung des SUA-Prozesses zum Anfordern eines SUA-Zertifikats
Akteure	Sendersystem, (Distributor, Endreceiver)
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	1. Das Sendersystem muss ein SUA-Zertifikat anfordern und verwalten können. 2. Das Sendersystem muss die zu übermittelnde Meldung mit dem SUA-Zertifikat signieren (doppelte Signatur).
Ergebnis	SUA wird vom Sendersystem gemäss Anhang C, *Detailspezifikation Swissdec Unternehmens-Authentifizierung SUA* umgesetzt.

© Swissdec 2026

AB-18 Swissdec Unternehmensauthentifizierung (SUA)	
Fehlerbehandlung	Fehlermeldungen im Rahmen des SUA-Prozesses müssen dem Teilnehmer dargestellt werden.
Technische Implikationen	Umsetzung gemäss Anhang C, *Detailspezifikation Swissdec Unternehmens-Authentifizierung SUA*.

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

F.19 AB-19 Archivdateien

AB-19 Archivdateien	
Kurzbeschreibung	Archivierung übermittelter Daten
Akteure	Sendersystem, Endreceiver
Fachliche Anforderung	Übermittelte Requests und empfangene Responses **müssen** signiert aber unverschlüsselt archiviert werden. Die Dauer der Archivierung hängt von fachlichen Anforderungen ab. Nicht relevante Daten müssen nicht archiviert werden (Polling, etc. Siehe Use Cases)
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	1. Der Akteur sichert fachlich relevante Requests nach dem Signieren aber vor der Verschlüsselung. 2. Der Akteur sichert fachlich relevante Responses nach dem Entschlüsseln, belässt aber die Signatur auf der Response-Datei. 3. Archivdateien müssen gemäss gesetzlichen Anforderungen aufbewahrt werden.
Ergebnis	Sämtliche inhaltlich relevanten Requests und Responses werden mit gültiger Signatur abgelegt. Der Benutzer kann während eines fachlich vorgegebenen Zeitraums darauf zugreifen.
Fehlerbehandlung	Fehler bei der Archivierung von Daten müssen dem Benutzer kommuniziert werden.
Technische Implikationen	keine

Tabelle F.18. AB-19 Archivdateien

F.20 SB-01 Transportsicherheit (HTTPS / TLS)

SB-01 Transportsicherheit (HTTPS / TLS)	
Kurzbeschreibung	Umsetzung der Transportsicherheit
Akteure	Sendersystem, Distributor
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 Sendersystem und Distributor ist verschlüsselt.

© Swissdec 2026

SB-01 Transportsicherheit (HTTPS / TLS)	
Fehlerbehandlung	Fehler in der Transportsicherheit: Abbruch der Übermittlung
Technische Implikationen	Siehe [[SECPDF]]

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

F.21 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.20. SB-02 Transport Authentication (TLS Mutual Authentication)

F.22 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.
Ergebnis	Sämtliche übermittelten Nutzdaten sind verschlüsselt.
Fehlerbehandlung	Fehler in der Verschlüsselung: Die Übermittlung wird abgebrochen.
Technische Implikationen	Siehe [[SECPDF]]

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

F.23 SB-04 Nutzdatensignatur (WS-Signature)

SB-04 Nutzdatensignatur (WS-Signature)	
Kurzbeschreibung	WS-Signierung der übermittelten Daten

© Swissdec 2026

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

F.24 SB-05 Unternehmens-Authentifizierung SUA Signatur (WS-Signatur)

SB-05 Unternehmens-Authentifizierung SUA Signatur (WS-Signatur)	
Kurzbeschreibung	Umsetzung des SUA-Prozesses seitens Sendersystem
Akteure	Sendersystem
Fachliche Anforderung	keine
Abhängige Bausteine	keine
Ausschliessende Bausteine	keine
Anforderungen an Akteure	SUA wird gemäss Spezifikation umgesetzt. Es kann ein SUA-Zertifikat angefordert und installiert werden, und mit diesem Zertifikat kann ein Request doppelt signiert werden.
Ergebnis	Ein gültiges SUA-Zertifikat kann angefordert und installiert werden. Mit diesem Zertifikat kann doppelt signiert werden.
Fehlerbehandlung	1. Kein SUA-Zertifikat vorhanden: Der SUA-Prozess zum Anfordern eines Zertifikats muss durchgeführt werden. 2. Fehler im Anforderungsprozess eines SUA-Zertifikats. Abbruch und erneute Ausführung ab Beginn. 3. Fehler bei der Nachrichtensicherheitsüberprüfung. Abbruch der Übermittlung.
Technische Implikationen	Siehe [[SECPDF]]

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

F.25 SB-06 Nichtabstreitbarkeit

SB-06 Nichtabstreitbarkeit	
Kurzbeschreibung	Eindeutige Zuordnung an spezifische Teilnehmer
Akteure	Sendersystem, Distributor, Empfängersystem
Fachliche Anforderung	keine
Abhängige Bausteine	keine

© Swissdec 2026

SB-06 Nichtabstreitbarkeit	
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.24. SB-06 Nichtabstreitbarkeit

F.26 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.25. PB-01 Synchrone Verteilung

F.27 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
Ergebnis	Die Meldungen wurden verteilt, die Antworten zurückgeliefert und alle fachlichen Daten gelöscht.
Fehlerbehandlung	

© Swissdec 2026

PB-02: Asynchrone Verteilung	
Technische Implikationen	keine

Tabelle F.26. PB-02 Asynchrone Verteilung

F.28 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.27. PB-03 Synchronisierung

F.29 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	
Anforderungen an Akteure	Innerhalb der Automatisierung muss bei neuen DialogMessages auch eine manuelle Verarbeitung möglich sein.

© Swissdec 2026

PB-04: Dialogisierung	
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.28. PB-04 Dialogisierung

F.30 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.29. PB-05 Completion

F.31 PB-06 Prozessnavigation

PB-06: Prozessnavigation	
Kurzbeschreibung	Umsetzung der Prozessnavigation
Akteure	Sendersystem, Distributor
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. Die einfachste Form ist ein Fortschrittsbalken mit %-Angaben.
Abhängige Bausteine	Es besteht evtl. eine semantische Abhängigkeit oder Überlappung mit den möglichen **Protokoll-Zuständen**.
Ausschliessende Bausteine	

© Swissdec 2026

PB-06: Prozessnavigation	
Anforderungen an Akteure	Einerseits muss diese Navigation geführt und übermittelt werden. Andererseits muss der Kontext mit dem aktuellen Ort visualisiert 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.30. PB-06 Prozessnavigation

F.32 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.31. PB-07 Digitalisierungsbereich

F.33 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 können weitere definierte Daten zu einem späteren Zeitpunkt verlangt werden. Für jede verlangte Story lässt sich optional noch ein Fristdatum setzen.

© Swissdec 2026

PB-08: Daten anfordern	
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.32. PB-08 Daten anfordern

© Swissdec 2026

G Technische Dokumentationen

G.1 Schema Dokumentation IncidentDeclarationServiceTypes.xsd

Target Namespace	urn:ch:swissdec:kle:v2:20260306:incidentdeclaration:service:types
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 • idst : urn:ch:swissdec:kle:v2:20260306:incidentdeclaration:service:types
Version	0.0

G.2 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] </...>

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]

© Swissdec 2026

	<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	<...> <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“

© Swissdec 2026

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] </...>

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

© Swissdec 2026

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	<...> <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“]

© Swissdec 2026

	<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] </...>

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

© Swissdec 2026

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.
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] </...>

© Swissdec 2026

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] </...>

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“ </AddresseeCon\ text> [1..1]

© Swissdec 2026

	[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: SynchronizeSubscribeOrganizationConsumerTyp SynchronizeSubscribeOrganizationConsumer> [1..1] [END CHOICE] </...>

SimpleType: DeclareIncidentStateEnumType

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

SimpleType: SubscribeOrganizationStateType

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

G.3 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

© Swissdec 2026

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
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]

© Swissdec 2026

	<...> [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] <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] </...>

© Swissdec 2026

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] <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

© Swissdec 2026

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	<...> <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	<...>

© Swissdec 2026

	<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] <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]

© Swissdec 2026

	</...>

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“
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] </...>

© Swissdec 2026

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“ </CompanyDescripti\ on> [1..1] <Person> „ComplexType: PersonIncidentType“ </Person> [1..1] <Institutions> „ComplexType: CustomerIdentificationType“ </Institutions> [1..1] <Contact> „ComplexType: ContactType“ </Contact> [0..1] </...>

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]

© Swissdec 2026

```
<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] <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

© Swissdec 2026

XML Instance Representation	<...> <InjuredPartOfBody> „ComplexType: InjuredPartOfBodyType“ </InjuredPartOfBody> [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] </...>

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“ </LessonsPerMonth> [0..1] </...>

ComplexType: InsuredSalaryPastType

Abstract	no

© Swissdec 2026

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] </...>

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]

© Swissdec 2026

	<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] </.../>

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]

© Swissdec 2026

	<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“] <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

© 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] <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] <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]

© Swissdec 2026

	<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] <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“ </CoveredDailyAllowan\ ce> [0..1] <Contact> „ComplexType: ContactType“ </Contact> [0..1] <Indemnifications> „ComplexType: IndemnificationsType“ </Indemnifications> [0..1] <Comment> „ComplexType: NotificationsType“ </Comment> [0..1] </...>

ComplexType: SpecialCodeType

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] <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“ </OptionalEnterpriserIn\ surance> [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
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]

© Swissdec 2026

	</...>

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] </...>

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]

© Swissdec 2026

	<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	<...> [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	<...>

© Swissdec 2026

	<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

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

© Swissdec 2026

	• 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 • 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

© Swissdec 2026

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

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

© Swissdec 2026

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 • 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

© Swissdec 2026

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

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

© Swissdec 2026

	• 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.4 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]`
	`</...>`

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]`

© 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“ </Producer\ ResponseNotifications> [1..1] <Warning> „ComplexType: NotificationsType“ </Warning> [0..1] <Info> „ComplexType: NotificationsType“ </Info> [0..1] </...>

ComplexType: AddresseeResponseType

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

© Swissdec 2026

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
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]

© Swissdec 2026

	<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] </...>

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

© Swissdec 2026

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] <SignatureCertificateUID-BFS> „ComplexType: SignatureCertificateUID-BFSType“ </ SignatureCertificateUID-BFS> [0..1] <ProducerResponseNotifications> „ComplexType: FeedbackNotificationsType“ </Producer\ ResponseNotifications> [1..1] <ConsumerNotifications> „ComplexType: FeedbackNotificationsType“ </ConsumerNotifica\ tions> [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]

© Swissdec 2026

	<FaultInformation> „ComplexType: ConsumerFaultType“ </FaultInformation> [1..1] <PlannedMaintenance> „ComplexType: PlannedMaintenanceWindowType“ </PlannedMainten\ ance> [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“ </PlannedMainten\ ance> [1..1] [END CHOICE] </...>

ComplexType: FeedbackNotificationsType

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

© Swissdec 2026

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] </...>

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	<...>

© Swissdec 2026

	<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] </...>

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]

© Swissdec 2026

	<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
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	<...>

© Swissdec 2026

	<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] </.../>

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

© Swissdec 2026

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] </...>

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] </...>

© Swissdec 2026

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] <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

© Swissdec 2026

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
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

© Swissdec 2026

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
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

© Swissdec 2026

	• 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
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

© Swissdec 2026

Base type	xs:NMTOKEN
Facets	• Enum: yes • Enum: no • Enum: unknown

G.5 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] </...>

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] </...>

© Swissdec 2026

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
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	<...>

© Swissdec 2026

	<...> [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] </...>

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]

© Swissdec 2026

	</...>

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] <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

© Swissdec 2026

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]
	  <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“]

© Swissdec 2026

	<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“] <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	<...>

© Swissdec 2026

	<RegisterOrganizationAuthenticationResponse> „ComplexType: RegisterOrganizationAuthenticationAddressee 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“
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]

© Swissdec 2026

	<SignCertificate> „ComplexType: CertificateSignRequestType“ </SignCertificate> [0..1] <RenewCertificate> „ComplexType: CertificateSignRequestBaseType“ </RenewCertificate> [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“ </Quittance> [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“ </AddresseeContext> [1..1] [END BASE TYPE] <Addressee> „ComplexType: AddresseeType“ </Addressee> [1..1] <Case> „ComplexType: RegisterOrganizationAuthenticationAddresseeResponseCaseType“ </Case> [1..1] </...>

ComplexType: SynchronizeRegisterOrganizationAuthenticationConsumerResponseType

Abstract	no
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]

© Swissdec 2026

	[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	<.../> [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}

© Swissdec 2026

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

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

© Swissdec 2026

	• 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 • Enum: NON • Enum: NOY • Enum: SFN

SimpleType: CivilStatusType

Translation	ZivilstandsTyp
Short description	ZivilstandsTyp
Content type	Restriction
Base type	xs:NMTOKEN
Facets	• Enum: unknown

© Swissdec 2026

	• 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 • 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

© Swissdec 2026

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; 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

© Swissdec 2026

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)
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

© Swissdec 2026

	• 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
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