

Bericht

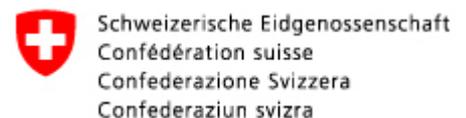
Detailkonzept Meldungsformat nach eCH-0058v4

Schweizerische Steuerkonferenz SSK
p.a. Steuerverwaltung Basel Stadt
Fischmarkt 10
CH-4001 Basel

Bundesamt für Sozialversicherungen
Effingerstrasse 20
CH-3003 Bern

Verein eAHV/IV – eAVS/AI
p.a. mundi consulting AG
Marktgasse 55
CH-3000 Bern 7

26.01.2017





Dokumentinformationen

Titel:	Detailkonzept Meldungsformat nach eCH-0058v4	
Projektnummer:	12.516.00.15	
Veröffentlichungsdatum:	26.01.2017	
Gespeichert:	1. Februar 2017	
Anzahl Seiten:	74 exkl. Beilagen	
Dateiname:	Ber_Detailkonzept_Meldungsformat_v2-33.docx	
Dokumentverantwortlicher:	BSV, SSK	
Geprüft durch:	Projektteam DKMF	Datum:

Versionen

Version des Dokuments	Version des common.xsd	Datum	Wichtigste Änderungen	Verantwortlich
V1.0-1.2	--	25.10.2010	Finalisierung für Phase Pilot Ergänzung businessProcessId im Meldungsrahmen für Meldungsprozesse mit mehr als zwei Meldungen. Ergänzungen Kodierung und Rückwärtskompatibilität	M. Gomez S. Pasquier C. Pich
V2.0	--	20.09.2011	Ergänzungen aus Besprechung vom 19.9.2011 BSV / ZAS / AWK	H. Häfliger C. Pich
V2.1	AHV/IV: eahv-iv-common-4-0.xsd SSK: ssk-common-2-0.xsd	27.06.2013	Aktualisierung aufgrund finalisiertem eCH-0058 Version 4; Dokument soll gemeinsam mit SSK weitergeführt werden → entsprechende Anpassungen	M. Obrecht C. Kuhn A. Meier
V2.2	AHV/IV: eahv-iv-common-4-0.xsd SSK: ssk-common-2-0.xsd	10.07.2013	Anpassungen aufgrund von Rückmeldungen der ZAS	M. Obrecht C. Kuhn
V2.21	AHV/IV: eahv-iv-common-4-1.xsd SSK: ssk-common-2-1.xsd	24.06.2015	Einarbeiten sämtlicher erfasster Change Requests	M. Obrecht C. Kuhn
V2.3	dito	19.10.2016	Ergänzung der Header-Anzeigetexte im Kap. A.2.1	C. Kuhn M. Obrecht
V2.31	dito	28.11.2016	Anpassung action=10 von Transmission zu Transfert	C. Kuhn
V2.32	dito	12.12.2016	Einarbeitung der Rückmeldungen aus dem Review durch die Wartungsgruppe DA	C. Kuhn

© AWK Group AG

Dieser Bericht ist vertraulich und nur für den Auftraggeber bestimmt. Diesem steht das Recht zu, die Arbeitsergebnisse von AWK für den vereinbarten Zweck zu verwenden. Eine über den Auftrag hinausgehende Verwendung ist nicht zulässig.



Version des Documents	Version des common.xsd	Datum	Wichtigste Änderungen	Verantwortlich
V2.33	dito	26.01.2017	Anpassung des eCH-0044 von der Version 4.1 zu 4.0 (Grund: Die Version 4.1 enthält ein zusätzliches Geschlecht „unbestimmt“, dessen Einführung bei allen Teilnehmern einen Umsetzungsaufwand bedeuten würde, fachlich aber nicht benötigt wird.)	C. Kuhn

Abkürzungen und Begriffe

Abkürzung	Beschreibung
Meldung	In diesem Dokument wird der Begriff „Meldung“ für alle Meldungen mit den Action Codes 1, 3, 4, 5, 6, 10 und 12 verwendet. Meldungen mit den Action Codes 8 und 9 werden als fachliche Quittungen bezeichnet.
Antwort (response)	Senden von Daten (Action „6“), welche mittels Action „5“ angefordert wurden (eCH-0090, messageClass 1). Es handelt sich dabei nicht um eine Antwortmeldung im Sinne einer fachlichen Quittung des eCH-0058v4-Standards, sondern um eine neue Lieferung mit den angeforderten Daten ([11], Seite 12).
Fachliche Quittung (Antwortmeldung)	Wird in diesem Dokument von einer fachlichen Quittung gesprochen, so handelt es sich um eine Meldung, welche aus einem Meldungskopf mit Aktionscode 9 oder 8 und einer Quittung zu einer Meldung mit fachlichen Daten besteht ([11], Seite 11). Im eCH-0058v4 wird für die fachliche Quittung der Begriff „Antwortmeldung“ verwendet. Da dieser Begriff zu Verwechslungen mit der Antwort (action „6“) führen kann, wird in diesem Dokument auf den Begriff „Antwortmeldung“ verzichtet.
Sedex Quittung	Technische Empfangsbestätigung, welche dem Sender die erfolgreiche oder nicht erfolgreiche Übermittlung auf Ebene Sedex Client bestätigt.
MT	messageType
SMT	subMessageType
sMC	sM-Client

Referenzierte Dokumente

Titel	Autor / Herausgeber	Datum
[1] Architekturkonzept der Projekte CH-Meldewesen Steuern und DA	SSK & Verein eAHV/IV	05.12.2008
[2] Spezifikation sedex Meldungs-Client der Projekte CH-Meldewesen Steuern und DA	SSK & Verein eAHV/IV	05.12.2008
[3] Technisches Prozesshandbuch der Projekte CH-Meldewesen Steuern und DA	SSK & Verein eAHV/IV	05.12.2008
[4] Das Handbuch für den Software-Lieferanten im Datenaustausch für die Registerharmonisierung, Volkszählung und sedex.	BFS/OFS	21.07.2008
[5] eCH-0058 Datenstandard Meldungsrahmen Vorschlag vom 28.02.09	eCH / M. Stingelin	28.02.2009
[6] eCH-0090 Datenstandard sedex Umschlag	eCH / I. Metz, A. Scheidegger	17.12.2007



Titel	Autor / Herausgeber	Datum
[7] eCH Basisstandards - eCH-0007 - eCH-0008 - eCH-0010 - eCH-0011 - eCH-0033 - eCH-0044 - eCH-0075 - eCH-0097 - eCH-0158	eCH / Verschiedene	Verschiedene
[8] Meldungsaufbau zu CH-Meldewesen Steuern (Konzept)	SSK	30.06.2008
[9] Teilnehmer- und Meldungsverzeichnisse der Projekte CH-Meldewesen Steuern und DA	SSK BSV	Wird laufend aktualisiert
[10] Technisches Design sM-Client	CTP	V3.11, 21.06.2012
[11] eCH-0058 Datenstandard Meldungsrahmen http://www.ech.ch/vechweb/page?p=dossier&documentNumber=eCH-0058&documentVersion=4.00	eCH / Willy Müller, M. Stingelin	V4.00 28.06.2012
[12] Detailkonzept Meldungsformat	SSK & Verein eAHV/IV	V1.3, 02.08.2011
[13] Dokumenttypen des Projekts DA-Dossier <i>Sharepoint eAHV/IV: SA01 - DA → Dokumente → Dokumente → Allgemeinguelteige_Dokumente → Ber_121101_Dokumenttypen_vxx.xls</i>	Verein eAHV/IV	Wird laufend aktualisiert (Wartungsgruppe DA)
[14] DA Dossier: Datenschnittstelle (Konzeptionelles Design)	Verein eAHV/IV	V2.0 14.12.2012
[15] Dokumentation auf http://www.chm-steuern.ch → Spezifikation Basislayout	SSK & Verein eAHV/IV	V1.1
[16] sedex-Handbuch		V4.0.3, 15.11.2012



Inhaltsverzeichnis

1. Zweck des Dokuments	8
2. Einleitung.....	8
2.1. Unterschiede von eCH-0058 Version 4 zu früheren Versionen	8
2.2. Abgrenzung und Gültigkeitsbereich	9
2.3. Definition einer Meldung	9
2.4. Anforderungen.....	9
2.5. Prozess Meldungsübermittlung.....	10
2.6. Aufbau des Dokuments.....	13
Teil A – Asynchroner Meldungs austausch	13
3. Aufbau einer Meldung.....	13
3.1. Grundlagen.....	13
3.2. Aufbau des Nutzdatenpakets.....	14
3.2.1. Grundsätzlicher Aufbau des Nutzdatenpakets (Einzel- und Sammelmeldungen)	14
3.2.2. Aufbau des message_A.xml.....	15
3.2.3. Organisation der Anhänge.....	16
3.2.4. Spezifitäten von Sammelmeldungen nach eCH-0058v4	16
3.2.5. Datei- und Ordnernamen innerhalb des Nutzdatenpakets	17
3.3. Meldungsrahmen	17
3.3.1. Von eCH-0058 vererbte Elemente.....	18
3.3.2. Fakultative domänenspezifische Elemente.....	24
3.3.2.1. Definition des Elements ‚attachment‘	24
3.3.2.2. Erweiterung des Headers (Element „extension“)	26
3.4. Meldungsinhalt	27
3.4.1. Strukturierter Inhalt (<content></content>)	28
3.4.2. Unstrukturierter Inhalt.....	28
3.5. Weitere Aspekte des Meldungs aufbaus	28
3.5.1. Mehrsprachigkeit.....	28
3.5.1.1. Elemente	28
3.5.1.2. PDF Generierung.....	28
3.5.2. Namespaces	28
3.5.3. Versionierung der XSD.....	30
3.5.4. Kodierung.....	31
3.5.5. XML Escape Sequenzen.....	31
3.5.6. Annotationen in Meldungs-XSD	31
3.5.7. Umgang mit NULL-Werten	31
4. Einbettung in die Meldungsprozesse	32
4.1. Aufbau des Headers bei verschiedenen action-Codes.....	32
4.1.1. Meldungs aufbau bei action=“1“ (neu).....	32



4.1.2.	Meldungsaufbau bei action="3" (Widerruf).....	33
4.1.3.	Meldungsaufbau bei action="4" (Korrektur)	34
4.1.4.	Meldungsaufbau bei action="5" (Anfrage)	34
4.1.5.	Meldungsaufbau bei action="6" (Antwort).....	35
4.1.6.	Meldungsaufbau bei action="10" (Weiterleitung)	35
4.1.7.	Meldungsaufbau bei action="12" (Mahnung)	36
4.2.	Fachliche Quittungen (action-Code 8 und 9).....	36
4.2.1.	Aufbau der fachlichen Quittungen	37
4.2.2.	Header von fachlichen Quittungen	38
4.2.3.	Inhalt von fachlichen Quittungen (infoType)	39
4.3.	Beispiele von Meldungsabläufen mit fachlichen Quittungen.....	41
4.3.1.	Neue Meldung mit positiver Quittierung.....	42
4.3.2.	Neue Meldung mit negativer Quittierung	42
4.3.3.	Anfrage mit negativer Quittierung	43
4.3.4.	Anfrage ohne Quittierung, Antwort mit negativer Quittierung	43
4.3.5.	Widerruf/ Korrektur mit negativer Quittierung.....	44
4.4.	Fehlerhandling	44
4.4.1.	Grundsatz des Fehlerhandlings.....	44
4.4.2.	Mögliche Fehler und deren Behandlung.....	44
5.	Vergabe messageType / subMessageType	48
5.1.	Nummernbereiche messageType	48
5.2.	Zuordnung messageType zu Meldungsprozessen.....	48
5.3.	Verwendung subMessageType.....	48
5.4.	Beispiel.....	48
Teil B – Synchroner Meldungen		49
6.	Aufbau einer Meldung.....	49
7.	Einbettung in die Meldungsprozesse	49
Teil C – Fachbereichsspezifische Datentypen		50
8.	Fachbereichsspezifische Datentypen	50
8.1.	Basistypen	51
8.1.1.	Built-in datatypes.....	51
8.1.2.	Durch eCH standardisierte Basistypen.....	51
8.1.2.1.	eCH-0007 - Datenstandard Gemeinden	52
8.1.2.2.	eCH-0008: Datenstandard Staaten und Gebiete	53
8.1.2.3.	eCH-0010: Datenstandard Postadresse für natürliche Personen, Firmen, Organisationen und Behörden.....	53
8.1.2.4.	eCH-0011: Datenstandard Personendaten.....	55
8.1.2.5.	eCH-0044: Datenstandard Austausch von Personenidentifikationen	56
8.1.2.6.	eCH-0097 Schnittstellenstandard Unternehmensidentifikation	57
8.1.3.	Weitere Basistypen	57
8.2.	Komplexe Typen.....	58



8.2.1.	In ssk-common und eahv-iv-common definierte Typen	60
8.2.1.1.	Natürliche Person Steuermeldungen AHV (naturalPersonsTaxReturnsOASIDType) ..	60
8.2.2.	In ssk-common definierte Typen.....	61
8.2.2.1.	Natürliche Person Steuern (naturalPersonsTaxType)	61
8.2.2.2.	Natürliche Person Steuern erweitert (extendedNaturalPersonsTaxType)	61
8.2.2.3.	Natürliche Person Steuern Melap (melapNaturalPersonsTaxType)	61
8.2.2.4.	Juristische Person Steuern (legalEntitiesTaxType)	62
8.2.2.5.	Versicherung Melap (melapInsuranceType)	62
8.2.2.6.	Kollektivversicherungsnehmer Melap (melapCollectiveInsuredPartyType)	63
8.2.2.7.	Grundstück/Liegenschaft (realEstatePropertyType)	63
8.2.3.	In eahv-iv-common definierte Typen	64
8.2.3.1.	Natürliche Person AHV-IV (naturalPersonsOASIDType).....	64
A.	Anhang	65
A.1.	Mehrsprachigkeit	65
A.1.1.	Übersicht.....	65
A.1.2.	Konzept.....	65
A.1.3.	XSL-Dateien.....	66
A.1.4.	Generierung der „Property“ Dateien	67
A.2.	Basislayout	67
A.2.1.	Aufbereitung des Meldungskopfs	68
A.2.2.	Aufbereitung des fachlichen Inhaltes.....	70
A.2.3.	Fusszeile.....	70
A.3.	Meldungsspezifikation und Umsetzung	71
A.3.1.	Vorgehen für Erstellung von Meldungsspezifikationen	71
A.3.2.	Struktur der Meldungsspezifikation.....	72
A.3.3.	eCH-Standards	72
A.4.	sedex-Umschlag.....	73
A.4.1.	Zusammenhang von Nutzdatenpaket und Sedex-Umschlag	73
A.4.2.	Aufbau des sedex-Umschlags.....	73



1. Zweck des Dokuments

Die vorliegende Version des Detailkonzepts Meldungsformat beschreibt den Aufbau und die zugehörige Struktur der Meldungen die auf eCH-0058 Version 4 (kurz eCH58v4) basieren, welche im Rahmen des Meldewesens in der Domäne Sozialversicherungen BSV (Bereiche AHV, IV, EO, FamZ und EESSI) und der Domäne SSK ausgetauscht werden. Auch domänenverwandte Projekte wie DA-PV (Prämienverbilligung) können bei Bedarf darauf aufsetzen.

2. Einleitung

eCH58v4 wurde im Frühling 2012 als neuer Standard für den Meldungsrahmen für Datenaustausch in den Domänen BSV und SSK definiert.

Das vorliegende Dokument beschreibt einen einheitlichen Aufbau von Meldungen, welche auf den Standard eCH-0058v4 aufbauen und gibt Richtlinien betreffend Elementen und Typen, die verwendet werden. Es dient auch als Leitfaden für die Durchführungsstellen, Pools und deren Lieferanten für die Umsetzung der Meldungen.

Ziel dieses Dokuments ist, eine einheitliche Grundlage für sämtliche Datenaustauschprojekte innerhalb der Domänen BSV und SSK zu definieren. Die eigentlichen Meldungsinhalte und Ausprägungen werden von den Projekten in den jeweiligen Meldungsspezifikationen definiert. Diese sollen durch die allgemeingültigen Vorgaben in diesem Dokument möglichst schlank gehalten werden können.

Das vorliegende Dokument ist eine Weiterentwicklung des Detailkonzepts, welches auf eCH0058 Version 2 und 3 aufbaut [12]. Es geht auf die speziellen Anforderungen im Bereich der Sozialversicherungen und Steuerämter ein und beschreibt die Änderungen, die sich aus der Definition eCH0058 Version 4 ergeben.

In den Domänen der SSK und des BSV wird für den Meldungsversand und -empfang grösstenteils der sedex Meldungs-Client (sM-Client) eingesetzt. Aus diesem Grund wird im vorliegenden Dokument wo sinnvoll abgegrenzt, welche Aufgaben vom sM-Client wahrgenommen werden und was in den Fachapplikationen berücksichtigt werden sollte.

2.1. Unterschiede von eCH-0058 Version 4 zu früheren Versionen

Mit eCH-0058v4 wurden die folgenden Neuerungen eingeführt:

- header.xml entfällt: Neu wird das Header-Element nur noch im message.xml geführt und nicht mehr doppelt sowohl im message.xml als auch im header.xml
- Möglichkeit für Sammelmeldungen: Mit eCH-0058v4 ist es erlaubt, 1 bis n Einzelmeldungen (message.xml) in einem zip zu übermitteln. Damit unterscheidet sich neu die Spezifikation von Einzel- und Sammelmeldungen nicht mehr – es kann dasselbe xsd verwendet werden. Die Dateinamen der message.xml müssen mit einer Nummer ergänzt werden (message_X.xml), um sie in ein zip packen zu können. Früher wurden Sammelmeldungen auf unterschiedliche Arten implementiert. Dies entfällt somit.
- Anpassungen an den Header-Elementen: Es wurden die folgenden Elemente neu in den Header aufgenommen: businessCaseClosed, responseExpected und businessProcessId. Die folgenden Elemente wurden entfernt: object.
- Der actionCode „11“ - „Fachliche Rückmeldung“ entfällt.



2.2. Abgrenzung und Gültigkeitsbereich

Das vorliegende Dokument ist gültig für die sedex-Domänen der SSK und des BSV (Meldungen der SSK, des BSV und eAHV/IV). Zweck des gemeinsamen Dokuments ist es, auch domänenübergreifend eine einheitliche Struktur der Meldungen sicherzustellen, so dass auch domänenübergreifende Meldungen einheitlich ausgetauscht werden. Dabei wird bewusst Wert darauf gelegt, dass die Domänen unabhängig bleiben.

Neben der asynchronen Sicht, deckt das Dokument auch die wichtigsten Aspekte des synchronen Datenaustausches ab. Dies ist wichtig, da es Meldungen gibt, welche sowohl synchron als auch asynchron angeboten werden. Dabei sollen die synchronen und asynchronen Meldungen möglichst identisch aufgebaut werden.

Verbindlichkeit seitens BSV

Seitens BSV wird das Dokument in eine Weisung aufgenommen, so dass es innerhalb der Domäne verbindlich angewendet werden muss. Begründeten Ausnahmen müssen beim BSV beantragt werden.

Verbindlichkeit seitens SSK

Die SSK hat nur Empfehlungscharakter, so dass das Dokument nicht als verbindlich erklärt werden kann. Da Projekte innerhalb der SSK-Domäne jedoch grundsätzlich von zentraler Stelle initiiert werden, soll das Detailkonzept Meldungsformat jeweils bei der Initialisierung von SSK-Projekten kommuniziert, resp. im Projektauftrag berücksichtigt werden.

2.3. Definition einer Meldung

Aus technischer Sicht stellt jede Informationsübermittlung von einem Absender zu einem Empfänger eine Meldung dar. Ein fachlicher Prozess mit einer Anfrage- und einer Antwort besteht daher aus zwei Meldungen. Die sedex-Quittung (Bestätigung der Zustellung auf Stufe sedex Client) repräsentiert keine Meldung im Sinne des zu definierenden Meldungsformats. Details zum technischen Ablauf des Meldungsaustausches können dem Kap. 2.5 entnommen werden.

2.4. Anforderungen

Die Anforderungen an das zu definierende Datenformat sind nachfolgend aufgeführt.

- **Standardisierung**
Das Meldungsformat muss soweit wie möglich standardisiert werden, damit der Meldungsaustausch zu anderen Stellen einfach realisiert werden kann. Hierzu soll auf den bestehenden Standards von eCH aufgebaut werden. Falls es noch keine eCH-Standards gibt, sollen welche angeregt werden.
- **Sedex-Tauglichkeit**
Die Nutzdaten müssen über sedex verschickt werden können. Die Vorgaben von sedex verlangen, dass die Nutzdaten nur aus einer einzigen Datei, die aber auch eine Zip Datei sein kann, bestehen dürfen und die Datei analog der sedex Umschlagsdatei benannt werden muss. Siehe [16] Kapitel 3.2 und 3.4.
- **Unterstützung strukturierter und unstrukturierter Meldungen**
Ein langfristiges Ziel besteht darin, vor allem strukturierte und damit automatisch zu verarbeitende Meldungen auszutauschen. Für eine länger dauernde Übergangszeit muss es jedoch möglich sein, auch unstrukturierte Meldungen zu versenden, wobei



der eigentliche Inhalt der Meldung in einem PDF- oder Bild-Dokument (TIFF) enthalten ist (siehe Kap. 3.2.3). Diese unstrukturierten Inhalte sollen jedoch mit Indexdaten zur automatischen Vorverarbeitung/Triage versehen werden können.

- Die eigentlichen Inhalte einer strukturierten Meldung werden über eine XML-Struktur definiert. Die Generierung einer menschenlesbaren Fassung der Meldung übernimmt bei Bedarf der Empfänger (dafür werden in der sM-Client Ablage standardisierte XSLT zur Verfügung gestellt).
- Anhänge zu Meldungen müssen unterstützt werden. Es muss möglich sein, die Anhänge zu priorisieren (ein allfälliges Hauptdokument (Leading Document) muss von Nebendokumenten [z.B. Belegen oder Beilagen] unterschieden werden können).
- Optimale Unterstützung des nachfolgend aufgeführten Prozesses Meldungsübermittlung (Kap. 2.5). Insbesondere ist dem Schritt 3 des Meldungsempfangs Beachtung zu schenken.

2.5. Prozess Meldungsübermittlung

Der Empfang einer asynchronen Meldung wird in den meisten teilnehmenden Stellen dreistufig erfolgen. In den Schritten „Abholung durch sedex-Client“ und „Vorverarbeitung / Triage“ (durch Fachapplikation) kann eine Quittung erfolgen (hellrot hinterlegt in Abbildung 1).

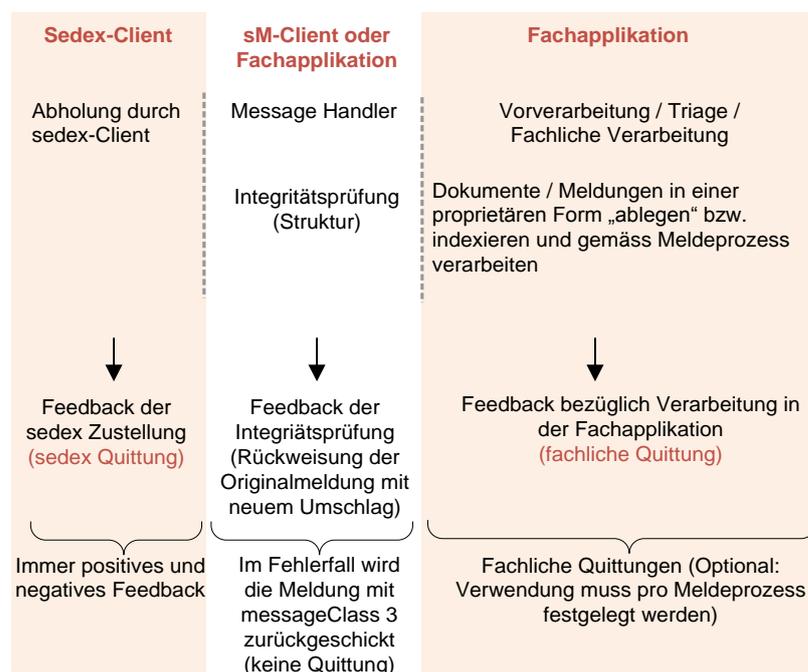


Abbildung 1: Prozess Meldungsempfang (asynchron)

- 1) Die Meldung wird vom sedex-Client des Empfängers vom sedex-Server abgeholt und im Meldungseingangsordner (inbox) abgelegt. Die vom sedex-Client automatisch generierte sedex-Quittung bestätigt dem Sender die erfolgreiche oder nicht erfolgreiche Übermittlung. Dieser Schritt erfolgt bei allen Teilnehmern automatisch und wird von sedex vorgegeben.
- 2) Im zweiten Schritt erfolgt die Integritätsprüfung der Meldung, wobei geprüft wird, ob der Aufbau der Nutzdatei korrekt ist und die Inhalte lesbar sind. Im Feh-



lerfall wird die Meldung verworfen und die Meldung an den Absender zurückgeschickt. *Hinweis:* Dieser Schritt wird bei vielen Teilnehmern vom sM-Client wahrgenommen (siehe 4.4.2 und [10]). Falls der sM-Client nicht eingesetzt wird oder die Meldungsvalidierung deaktiviert ist, sollten die entsprechenden Funktionalitäten/ Rückmeldungen (gemäss [10]) in der Fachapplikation umgesetzt werden.

- 3) Als nächster Schritt werden die Meldungen anhand ihres Typs triagiert (Funktionalität des sM-Client) und durch die Fachapplikation anhand des betroffenen Subjekts verarbeitet. Dabei werden die Meldungsinhalte beispielsweise in ein Dossier abgelegt oder einer bestimmten Fachapplikation zur Weiterverarbeitung übergeben. Dieser Schritt erfolgt bei den Empfängern aufgrund der unterschiedlichen Systemlandschaften und Bedürfnisse individuell.
Es kann pro Meldungsprozess festgelegt werden, ob fachliche Quittungen (positiv und negativ) verschickt werden sollen, um die korrekte end-zu-end-Verarbeitung sicherzustellen. Diese Quittungen sind in Kap. 4.2 beschrieben.
Ebenfalls in Kap. 4.2 sind die möglichen Fehlerfälle aufgeführt, welche mittels der fachlichen negativen Quittung abgedeckt werden können. Darüber hinausgehende Fehlerfälle sollen nicht via sedex, sondern im direkten Kontakt mit dem Absender behoben werden (siehe Kap. 4.4).
Bei der Verwendung der fachlichen Quittung ist zu beachten, dass diese nur verwendet werden kann, wenn sowohl der Absender als auch der Empfänger die Meldung intergiert verarbeitet.

Im synchronen Fall wird der Request in jedem Fall beantwortet – entweder mit einer Antwort (action=6) oder mit einer fachlichen Quittung.

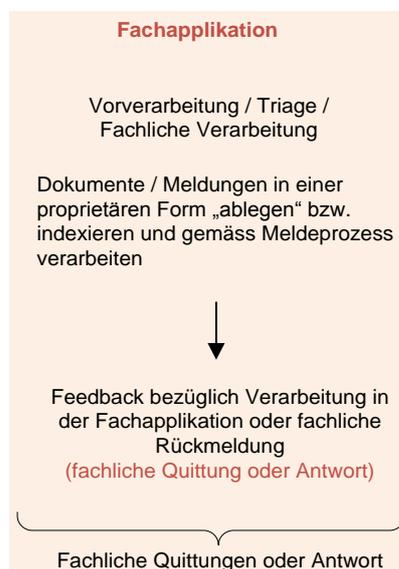


Abbildung 2: Prozess Meldungsempfang (synchron)

In der folgenden Abbildung 3 (asynchrone Sicht, resp. Abbildung 4 für die synchrone Sicht) ist der gesamte Prozess der Meldungsübermittlung end-zu-end vom Sender zum Empfänger dargestellt:

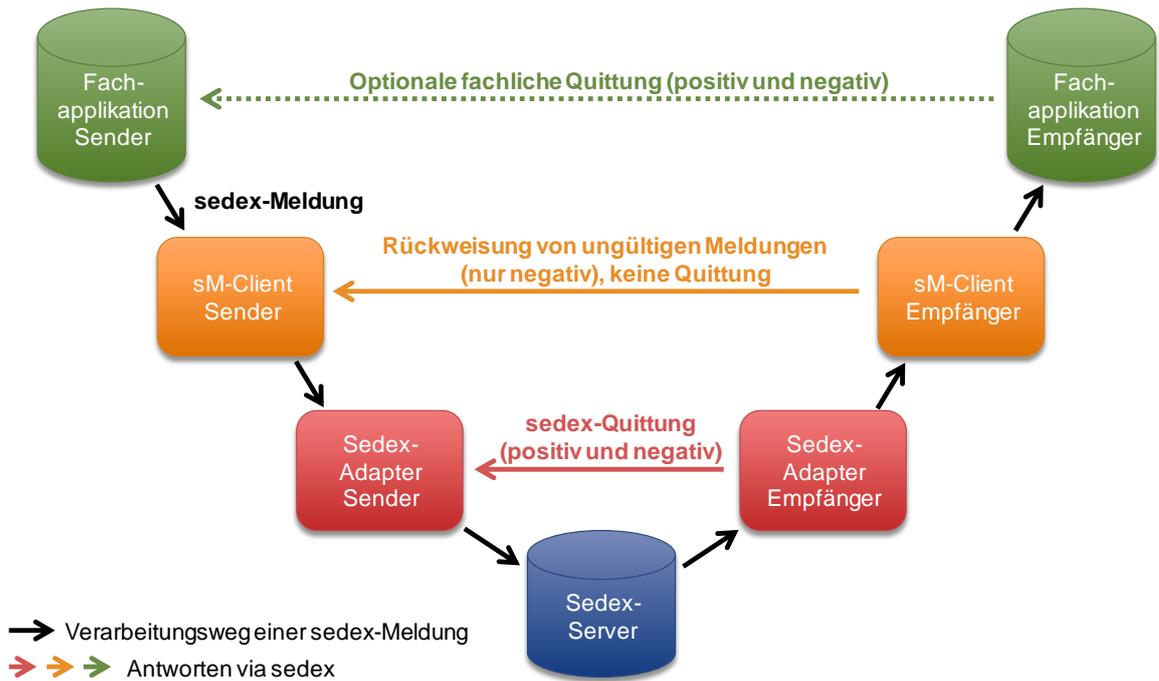


Abbildung 3: Gesamtsicht einer Meldungsübermittlung (inkl. Quittungen) – asynchrone Sicht.

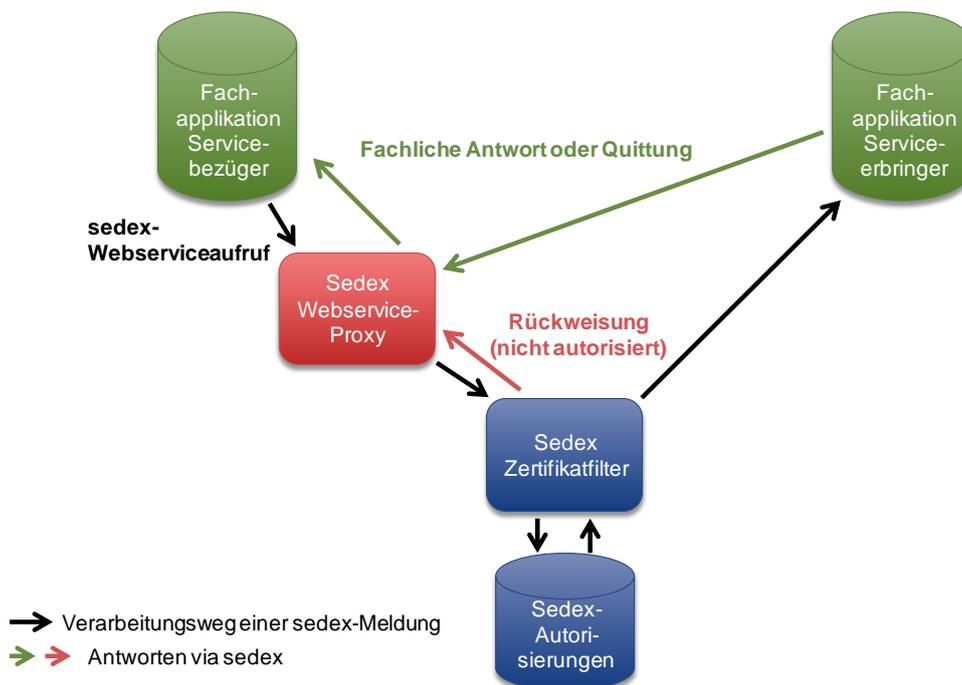


Abbildung 4: Gesamtsicht einer Meldungsübermittlung (inkl. Quittungen) – synchrone Sicht.



2.6. Aufbau des Dokuments

Das Dokument ist in drei Teile gegliedert:

- Im Teil A wird der asynchrone Meldungs austausch beschrieben.
- Teil B ergänzt Aspekte des synchronen Meldungs austausches. Dabei wird der Fokus primär auf die Unterschiede zum asynchronen Meldungs austausch gelegt.
- Teil C schlussendlich listet generische Datentypen auf, welche pro Domäne definiert werden und von den Meldungen verwendet werden können.

Teil A – Asynchroner Meldungs austausch

3. Aufbau einer Meldung

3.1. Grundlagen

Die Meldungen sind zweistufig aufgebaut (siehe Abbildung 5). Die untere Stufe, wird als sedex-Nutzdatenpaket bezeichnet und die obere als sedex-Umschlag. Der sedex-Umschlag (definiert im eCH-0090 Version 1.0) dient der Zustellung der Meldung durch sedex und enthält nur die zu diesem Zweck relevanten Informationen.

Das Nutzdatenpaket ist eine gezippte Datei mit einem Meldungsrahmen nach eCH-0058 und dem eigentlichen Meldungsinhalt. Dabei enthält der Meldungsinhalt primär fachliche Informationen der Meldung und repräsentiert damit die Meldung an sich. Der Meldungsrahmen enthält die technischen Informationen der Meldung. Der Aufbau des sedex-Nutzdatenpakets ist im Kap. 3.2 detailliert beschrieben.

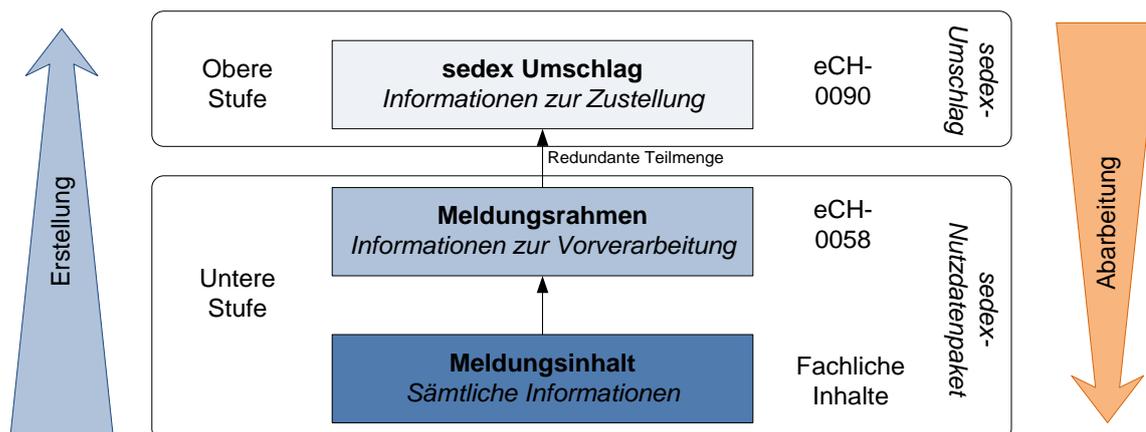


Abbildung 5: Aufbau der Meldung aus Sicht Information

Mit den Informationen der unteren Stufe (Nutzdatenpaket, insbesondere Meldungsrahmen) kann die Meldung beim Empfänger vorverarbeitet/ triagiert werden und damit den korrekten Systemen und insbesondere den korrekten Objekten (z.B. Person, Gebäude, Fall, Dossier, ...) zugewiesen werden.

Der sedex-Umschlag enthält eine redundante Teilmenge der Informationen der unteren Stufe. Dadurch wird erreicht, dass die Transport-Stufe (der sedex-Umschlag) nach ihrer



zweckgebundenen Verwendung „weggeworfen“ werden kann. Bei Verwendung des sM-Client wird das Entfernen des sedex-Umschlags vom sM-Client übernommen. Die Fachapplikation empfängt also nur noch das sedex-Nutzdatenpaket.

Wichtig: Wird der sM-Client (sMC) für den Meldungsverkehr eingesetzt, so wird der sedex-Umschlag komplett vom sMC gehandelt, d.h. der sMC ergänzt den sedex-Umschlag beim Versand einer Meldung und entfernt den Umschlag beim Empfang. Damit muss die Fachapplikation beim Versand dem sMC ausschliesslich das sedex-Nutzdatenpaket übergeben (ohne sedex-Umschlag). Dasselbe gilt für den Empfang: Die Fachapplikation erhält vom sMC nur das sedex-Nutzdatenpaket ohne Umschlag.

Da die meisten Teilnehmer innerhalb der sedex-Domänen SSK und BSV den sM-Client für die Meldungsverarbeitung verwenden, wird in diesem Dokument nicht weiter auf den sedex-Umschlag eingegangen, resp. ist im Anhang dokumentiert (siehe Kap. A.4).

3.2. Aufbau des Nutzdatenpakets

3.2.1. Grundsätzlicher Aufbau des Nutzdatenpakets (Einzel- und Sammelmeldungen)

Wie bereits in Kap. 3.1 erwähnt, besteht das sedex-Nutzdatenpaket aus einem zip-File (ZIP-Standard 2.0). In diesem zip-File können 1 bis n message_A.xml abgelegt werden (A steht hierbei für eine ganze Zahl, siehe Kap. 3.2.5). Das in früheren Standards separat mitzuschickende header.xml (durch den sM-Client generiert) entfällt im eCH-0058v4. Mit dieser Neuerung können Meldungen an denselben Empfänger auf einfache Weise zusammengefasst und als Sammelmeldung verschickt werden. Die Meldungsdefinitionen der Einzel- und Sammelmeldung unterscheiden sich dadurch nicht mehr. Eine Einzelmeldung kann als Sonderfall einer Sammelmeldung mit nur einer Meldung aufgefasst werden.

Zusätzlich zum message_A.xml enthält das zip 1 bis n optionale Ordner (attachments_A/) für Beilagen/ Attachments. In den folgenden Abbildungen ist das sedex-Nutzdatenpaket für eine Einzel- (siehe Abbildung 6) resp. Sammelmeldung (siehe Abbildung 7) dargestellt.

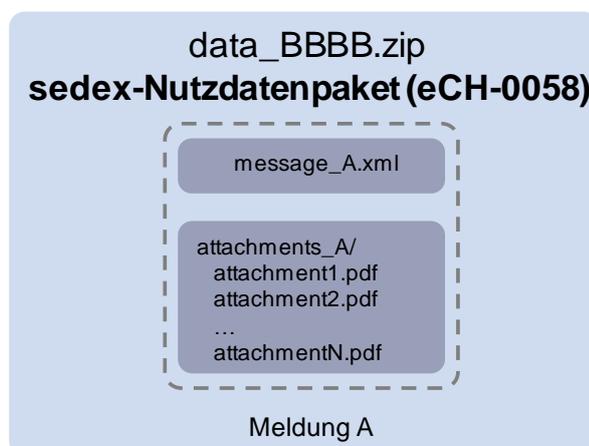


Abbildung 6: sedex-Nutzdatenpaket einer Einzelmeldung.

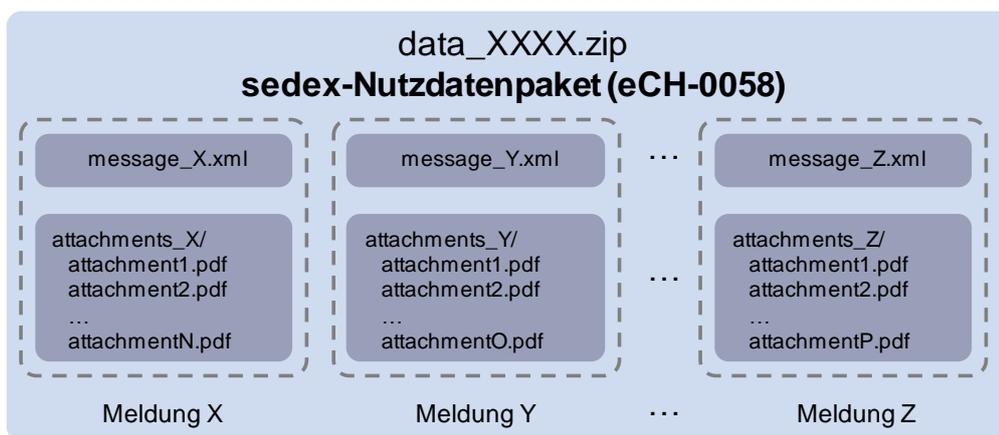


Abbildung 7: sedex-Nutzdatenpaket einer Sammelmeldung.

3.2.2. Aufbau des message_A.xml

Die einzelnen Meldungsdateien (message_A.xml) innerhalb des sedex-Nutzdatenpakets (zip) bestehen jeweils aus einem Meldungsrahmen (Header) und einem optionalen, strukturierten Meldungsinhalt (Content). Im Header werden technische Informationen wie Empfänger, Meldungstyp, etc. übertragen und er ist gemäss eCH-0058 definiert (siehe Kap. 3.3). Der Content wird in der jeweiligen Meldungsspezifikation definiert und enthält diejenigen fachlichen Inhalte, welche strukturiert abgebildet werden können (siehe Kap. 3.2.5).

Damit sind alle strukturierten Inhalte einer Meldung in einem einzigen XML-Dokument enthalten (message_A.xml) enthalten.

Der Aufbau einer Meldung (message_A.xml) ist in der folgenden Abbildung 8 gezeigt.

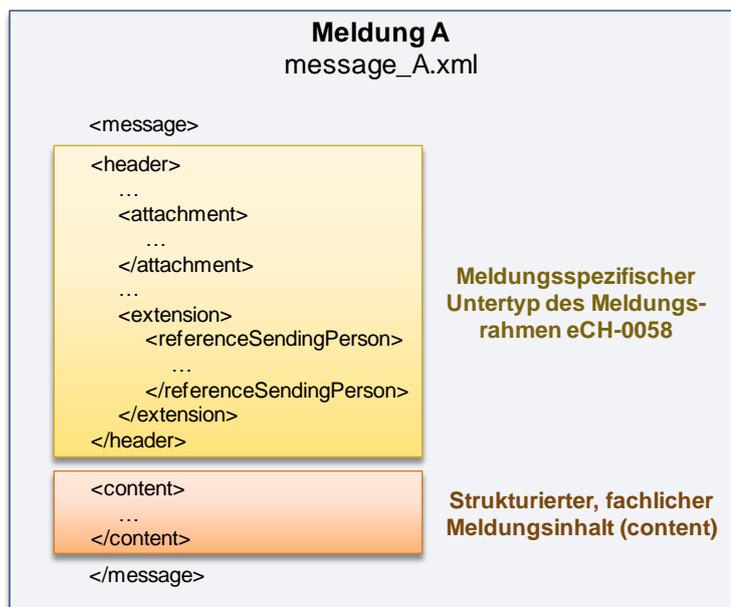


Abbildung 8: Aufbau einer Meldung message_A.xml innerhalb eines sedex-Nutzdatenpakets.

Hinweis: Dieser Meldungs Aufbau ist nur gültig für Meldungen vom Sender zum Empfänger gemäss Abbildung 3, resp. Abbildung 4. Allfällige fachliche Quittungen sind gemäss Kap. 4.2 aufgebaut.



3.2.3. Organisation der Anhänge

Anhänge zur Meldung „message_A.xml“ werden im attachments_A-Ordner (siehe Abbildung 6) innerhalb des sedex-Nutzdatenpakets (zip) abgelegt. Als Anhänge sind TIF- (TIFF Faxgruppe 4 singlepage) oder PDF/A-Dateien (inkl. aller PDF/A Untertypen) erlaubt. Der Ordner attachments_A/ darf selbst auch weitere Unterordner enthalten. Aus Performancegründen wird jedoch empfohlen, nicht mehr als drei Ordnerhierarchien zu verwenden. Enthält die Meldung keine unstrukturierten Daten, kann auf den Unterordner „attachments_A“ verzichtet werden.

Die Dateien der Attachments können grundsätzlich beliebig benannt sein und werden innerhalb des Meldungsrahmens (Header, siehe Kap. 3.2.2) über einen Pfad der Form „attachments_A/dateiname.pdf“ referenziert (vgl. Kapitel 3.3.2.1 – Element attachments). Im Element attachments werden pro Anhang noch weitere Informationen wie Dokumentenart, Datum, etc. übermittelt.

Eine Einzelmeldung kann mehrere Attachments enthalten wobei jedes dieser Attachments aus mehreren Dateien bestehen darf, sofern alle Dateien vom selben Format sind (z.B. mehrseitige TIFF-Dokumente).

Die Struktur einer Meldung mit Anhängen ist in Abbildung 9 schematisch dargestellt; die einzelnen Kästchen (header, attachment, file) werden entsprechend als XML-Elemente repräsentiert (siehe Kap. 3.3.2.1).

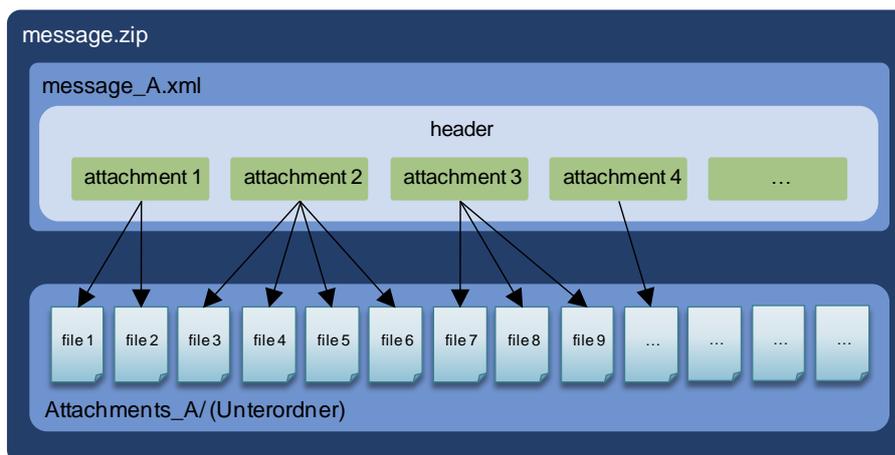


Abbildung 9: Meldung mit Anhängen bestehend aus mehreren Files.

3.2.4. Spezifitäten von Sammelmeldungen nach eCH-0058v4

In vorangehenden Versionen des eCH-0058 Standards wurden Sammelmeldungen innerhalb eines einzelnen XML-Files zusammengefasst und verschickt. Die konkrete Umsetzung der Meldungen war allerdings nicht einheitlich geregelt.

Mit der Version 4 des eCH-Standards 0058 besteht die Möglichkeit, mehrere Einzelmeldungen an denselben Empfänger in einer Sammelmeldung zu übermitteln. Die Einzelmeldungen werden in einer einzigen ZIP-Datei zusammengefasst. Analoges gilt für mehrere Antwortmeldungen an denselben Empfänger.

Dabei sind folgende Vorgaben einzuhalten:

- Es dürfen nur Meldungen an denselben Empfänger und von demselben Absender in einem ZIP zusammengefasst werden. (Die Eindeutigkeit der Empfänger wird vom sM-Client R5 sichergestellt.)



- Es dürfen nur Meldungen des gleichen Meldungstyps in einem ZIP zusammengefasst werden. (Dies ist Aufgabe der Fachapplikation.)
- Meldungen und Antwortmeldungen (= fachliche Quittungen) dürfen gemäss dem eCH-0058v4 Standard nicht in der gleichen Lieferung gemischt werden (als Meldungen gelten Meldungen mit den folgenden action-Codes (siehe Kap. 4): 1=neu, 3=Widerruf, 4=Korrektur, 5=Anfrage, 6= Antwort, 10=Weiterleitung, 12=Mahnung; als Antwortmeldungen gelten: 8=negative fachliche Quittung, 9=positive fachliche Quittung. Diese Trennung ist Aufgabe der Fachapplikation.

Sammelmeldungen machen nicht für alle Meldungstypen Sinn. Ob Sammelmeldungen ausgetauscht werden, soll deshalb in der jeweiligen Meldungsspezifikation festgelegt werden. Die Entscheidung, Sammelmeldungen zuzulassen oder zu deaktivieren, wird dabei auf rein konzeptioneller Ebene festgehalten und muss in den Fachapplikationen implementiert werden. Die Schemadefinitionen der Meldungen ändern sich dabei nicht.

Hinweis: Meldungen, mit inhaltlich zwar separierbarem Inhalt, deren Inhalte aber voneinander abhängig sind und nur als Ganzes verarbeitet werden können, fallen nicht unter den Begriff Sammelmeldung und müssen als eine Einzelmeldung spezifiziert werden. Die dabei entstehenden Beschränkungen von Performanz und XML-Grösse werden in Kauf genommen. Der Grund für diese Abgrenzung liegt in der gewählten Fehlerbehandlung von Sammelmeldung, welche im Kap. 4.4 beschrieben wird. Voneinander abhängige Einzelmeldungen können damit nicht gehandhabt werden.

3.2.5. Datei- und Ordnernamen innerhalb des Nutzdatenpakets

Das Nutzdatenpaket (ZIP Datei) hat den folgenden Inhalt:

- (1 bis n) XML-Dateien (z.B. **message_A.xml**)
- (0 bis n) Verzeichnisse (z.B. **attachments_A**) für allfällige Beilagen (z.B. im PDF/A- oder TIFF-Format (TIFF Faxgruppe 4 singlepage))

wobei A aus Buchstaben, Ziffern und Bindestrichen (jedoch keine Underscores „_“) bestehen kann (bis 20 Stellen sind erlaubt) und pro Meldung eindeutig sein muss.

Wenn in der Meldungsspezifikation nicht anderweitig definiert, sind die einzelnen Meldungen mit 00001 beginnend und fünf-stellig zu inkrementieren. Bei 12 Meldungen wären dies „message_00001.xml“, „message_00002.xml“, ..., „message_00012.xml“¹. Bei einer einzelnen Meldung heisst diese entsprechend „message_00001.xml“.

Der Name kann auch global eindeutig sein und/oder projektspezifisch weiter definiert werden.

3.3. Meldungsrahmen

Der Meldungsrahmen wird im Standard eCH-0058v4 definiert. Er ist so aufgebaut, dass bestimmte Elemente fest vorgegeben werden, während andere pro Meldung definiert werden müssen. In den Meldungen kann somit nicht direkt der Meldungsrahmen aus eCH-0058v4 verwendet werden, sondern es wird ein meldungsspezifischer Subtyp definiert. Neben der notwendigen meldungsspezifischen Definition gewisser Elemente, sollen Elemente, welche in den Fachbereichen SSK und AHV/IV nicht benötigt werden, bewusst

¹ Hinweis: Der Empfänger soll bei der Umsetzung nicht davon ausgehen, dass es keine Lücken in der Inkrementierung geben kann. Diese können z.B. auftreten bei ungültigen Einzelmeldungen, welche durch den sM-Client gemäss Kap. 4.4.2 entfernt und retourniert werden.



ausgeschlossen werden. Damit kann der Implementierungsaufwand in den Fachsystemen reduziert werden (keine Berücksichtigung von Sonderfällen).

Ein solcher spezifischer Header wird anschliessend als ein Untertyp von eCH-0058 verstanden. Sowohl hier, als auch später im Kapitel 3.3.1 ist häufig von Vererbung die Rede. Mit einigen Ausnahmen ist damit aber nur ein Zusammenhang auf konzeptioneller Ebene gemeint. Sofern im vorliegenden Dokument nicht explizit erwähnt, werden Vererbungsrelationen *nicht* durch den XSD-Inheritance Mechanismus (über Dateigrenzen hinweg) erzwungen, sondern die „vererbten“ Typen nach dem Vorbild des angegebenen Basistypen in den jeweiligen XSD-Files neu definiert. Die so aufgebaute Typhierarchie gemäss Abbildung 10 hat deswegen aber nicht weniger bindenden Charakter.

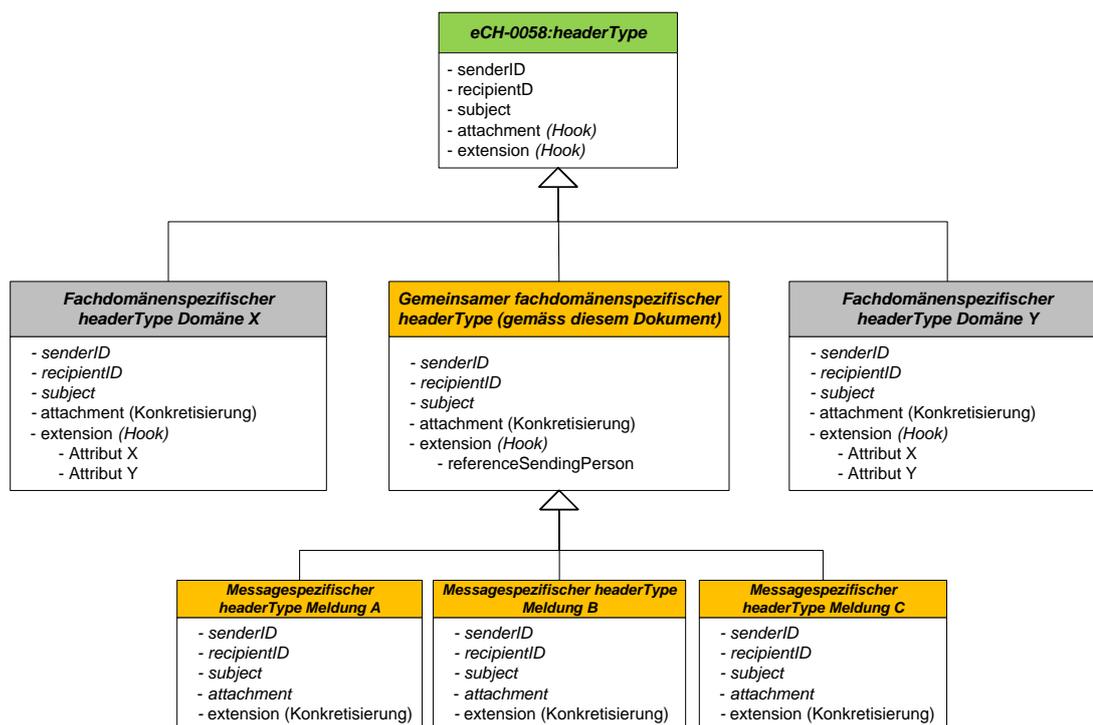


Abbildung 10: Typhierarchie der Meldungsrahmenformate.

Logisch erfolgt die Definition eines fachbereichsspezifischen Subtypen von eCH-0058 somit durch Vererbung der zwingenden Eigenschaften und Konkretisierung der abstrakten Elemente „attachment“ und „extension“. Letzteres ist zur Ergänzung mit fachspezifischen Eigenschaften gedacht.

Die in den nachfolgenden Unterkapiteln beschriebenen Elemente werden für den Meldungsrahmen (headerType) definiert und sollen entsprechend der Beschreibung eingesetzt werden.

3.3.1. Von eCH-0058 vererbte Elemente

Folgende Elemente sind im Standard eCH-0058v4 definiert und werden ohne fachbereichsspezifische Anpassungen übernommen. Das Vorkommen (pflicht, optional, nicht verwendet) wird in der Spalte „Vorkommen“ festgelegt. Stellenweise werden mehrere Möglichkeiten aufgeführt oder das Vorkommen wird mit einem Asterisken „*“ bezeichnet. In solchen Fällen muss in der Meldungsspezifikation entschieden werden, welches Vorkommen definiert wird, wobei die Verwendung des Asterisken ein Action-Code-abhängiges Vorkommen anzeigt (vgl. Kapitel 3.5.7). Bei Meldungen, welche mehrere Action-Codes zulassen, kann es dabei passieren, dass sich einzelne Vorkommen wider-



sprechen. In diesen Fällen muss in der Meldungsspezifikation ein gemeinsamer Nenner, welcher für alle verlangten Action-Codes funktioniert, gefunden werden.

Es kommt vor, dass sich das Vorkommen eines Elements aus anderen Gründen (nicht abhängig von Überlegungen zum Action-Code) nicht festlegen lässt, in solchen Fällen werden die verschiedenen Möglichkeiten mit Hilfe des „|“ Zeichens unterschieden. Die Festlegung des Vorkommens muss in diesen Fällen in der Meldungsspezifikation vorgenommen werden.

Die Elemente, welche dem headerType als Basis dienen sind:

Element	Typ	Verwendung	Vorkommen
senderId	eCH-0058: participantIdType	Absender entsprechend dem sedex-Adressierungskonzept. Entgegen der Empfehlung von [11] soll ausschliesslich die sedexID verwendet werden ohne vergebende Stelle. Bsp: 3-CH-1	1
originalSenderId	eCH-0058: participantIdType	Wird bei einer Weiterleitung (action="10") gesetzt, ursprünglicher Absender der Meldung. Falls keine Weiterleitung vorgesehen ist, soll das Element weggelassen werden.	*
declarationLocalReference	eCH-0058: declarationLocalReferenceType	Fachliche Identifikation des Absenders für Nachfragen. Da dieses Element im Standard den Typ xs:token der Länge 1 bis 100 hat, SSK und BSV jedoch eine Strukturierung mittels eines komplexen Typen verwenden möchten, wird die declarationLocalReference nicht verwendet und ein eigener Typ in der extension definiert. Es wurde ein Change Request an die eCH-Fachgruppe gestellt, das Element im Standard als xs:anyType zu definieren.	0
recipientId	eCH-0058: participantIdType	Empfänger entsprechend dem sedex-Adressierungskonzept. Falls etwas an mehrere Empfänger gemeldet wird, muss aus Gründen der Nachvollziehbarkeit und Fehlerbehandlung für jeden Empfänger eine eigene Meldung zu senden. Siehe auch Kapitel A.4. Bei Sammelmeldungen muss beim Erstellen der Meldung sichergestellt werden, dass alle enthaltenen message.xml denselben Empfänger enthalten.	1 ²
messageId	eCH-0058: messageIdType	Die messageId muss zusammen mit der senderId eindeutig sein. Versenden mehrere Fachsysteme Meldungen, müssen entsprechende Zahlenbereiche vergeben werden, damit nicht zwei Meldungen die gleiche messageId zugewiesen bekommen. eCH58v4 empfiehlt eine universelle eindeutige Identifikation (UUID) zu verwenden. Damit im Umschlag (siehe Kap. A.4) dieselbe messageId verwendet werden kann, sollte die messageId gemäss dem folgenden Format aufgebaut sein: ([a-zA-Z][0-9]){1,36}, also alphanumerisch (plus '-') sein, aus maximal 36 Zeichen bestehen und insbesondere keinen Underscore „_“ enthalten (Inkonsistenz zwischen eCH-0058 und eCH-0090).	1

² Dies stellt eine Einschränkung zu dem im Sedex-Umschlag definierten Vorkommens (1..n) dar. Es wird bewusst auf die Möglichkeit verzichtet, mit sedex mehrere Empfänger gleichzeitig zu adressieren.



Element	Typ	Verwendung	Vor- kommen
referenceMessageld	eCH-0058: messageIdType	Dieses Element wird von einer Anwendung gesetzt, wenn sie einer anderen Anwendung eine Antwort oder eine Fehlermeldung auf eine Meldung sendet. Das Element enthält die ID der ursprünglich gesendeten Meldung. Dieses Element ist nicht für jede Meldung Pflicht. Meldungen welche nur action=„1“ (Neu) zulassen, dürfen und sollten das Vorkommen dieses Elements als 0 definieren.	*
business-ProcessId	eCH-0058: businessProcessId- Type	Aus Sicht der für den Geschäftsfall zuständigen Stelle eindeutige Identifikation des Geschäftsfalles. Die eindeutige Geschäftsfall-ID wird vom Absender bzw. vom ursprünglichen Absender initiiert. Es wird empfohlen eine UUID (RFC 4122) zu verwenden. Innerhalb eines Meldungsprozesses muss das Feld zwingend zurückgemeldet werden. Die businessProcessId identifiziert die <i>Instanz</i> eines Geschäftsfalles, und nicht den <i>Typ</i> des Geschäftsfalles. Für ein Anwendungsbeispiel vgl. Abbildung 11.	1
ourBusinessReferenceld	eCH-0058: businessReference IdType	Weitere Identifikationsangabe, die der Sender mitgeben wird. Vereinfacht v.a. die Vorverarbeitung von Antworten. Muss nicht eindeutig im Kontext des Senders sein (entgegen der messageId). Dieses Feld wird benutzt um Sequenzen von Meldungen innerhalb eines Geschäftsfalles zu identifizieren. Die Verwendung wird dem Sender überlassen, resp. in der Meldungsspezifikation festgelegt. Für ein Anwendungsbeispiel vgl. Abbildung 11.	0 0..1 1
yourBusinessReferenceld	eCH-0058: businessReference IdType	Referenz auf die ourBusinessReferenceld bei einer Antwort oder einer anderen Referenzmeldung innerhalb der gleichen Sequenz eines Geschäftsfalles. Für ein Anwendungsbeispiel vgl. Abbildung 11.	*
uniqueIdBusinessTransaction	eCH-0058: uniqueIdBusiness TransactionType	Ermöglicht es dem Absender mehrere Meldungslieferungen als Teil einer Transaktion zu kennzeichnen. Wird aktuell nicht verwendet.	0
messageType (MT)	eCH-0058: messageTypeType	Analog sedex Umschlag gemäss Kapitel A.4. Bei Sammelmeldungen müssen alle Message.xml denselben messageType haben. (vgl. [11] Kapitel 2.4)	1
subMessageType (SMT)	eCH-0058: subMessageType- Type	Möglichkeit, den messageType zu verfeinern, z.B. durch Gruppierung von Meldungen unter einem messageType. <i>Hinweis:</i> Da gemäss Standard eCH58v4 Meldungen desselben messageTypes in einer Sammelmeldung übermittelt werden können, kann eine Triage nach subMessageType neu erst nach Aufsplitten der Sammelmeldung in Einzelmeldungen vorgenommen werden. Da der sM-Client die Meldung als Sammelmeldung an die Fachapplikation weitergibt, ist im sM-Client eine Triage nach subMessageType nicht mehr möglich.	1
sendingApplication	eCH-0058: sendingApplication- Type	Identifikation der Anwendung, welche die Meldung aufbereitet hat. Setzt sich zusammen aus den zwingend zu liefernden Elementen: <ul style="list-style-type: none"> • manufacturer – xs:token: Firmenbezeichnung des Herstellers der Anwendung • product – xs:token: Bezeichnung / Produktname der Anwendung • productVersion –xs: token: Produktversion der Anwendung 	1



Element	Typ	Verwendung	Vorkommen
partialDelivery	eCH-0058: partialDeliveryType	Möglichkeit, grosse Meldungen in Teilpakete aufzuteilen. Muss in der Meldungsspezifikation explizit definiert werden falls das Verschieben von gesplitteten Meldungen zugelassen werden soll.	0 0..1
subject	eCH-0058: subjectType	Betreff, zwingend pro Meldung zu definieren. Dieser wird wie folgt aufgebaut: "<Name Meldung> - <Identification object>", wobei sich der Name der Meldung auf die kleinstmögliche Granularität bezieht (messageType / subMessageType). Bei natürlichen Personen entspricht das Identification object „Name, Vorname“ der Person ³ , bei juristischen Personen dem „Namen“. Beispiele: <ul style="list-style-type: none"> „Kapitalleistungen Säule 2 und 3a – Muster, Hans“ „Beschluss an AK – Muster, Bettina“ „Steuerausscheidung JP – Musterfirma“ Der Betreff sollte beim Versand prinzipiell automatisch durch die Fachapplikation erstellt werden. Aktuell wird bewusst auf Mehrsprachigkeit verzichtet. Sind zu einem späteren Zeitpunkt sämtliche Objekte definiert und die zugehörigen Identifikationen bekannt, kann allenfalls eine dynamische Generierung des Betreffs bei der Anzeige erfolgen und damit Mehrsprachigkeit erreicht werden.	1
comment	eCH-0058: commentType	Der comment im header soll nur für technische Bemerkungen verwendet werden. Für fachliche Kommentare, wird ein separates Element im content definiert (vgl. Kap. 3.4). Vorkommen und Verwendung muss in der Meldungsspezifikation definiert werden.	0 0..1
messageDate	eCH-0058: messageDateType	Datum, an welchem die Meldung generiert wurde.	1
initialMessageDate	eCH-0058: messageDateType	Meldungsdatum der initialen Meldung bei Weiterleitungen (action = 10). Dieses Element bleibt bei Weiterleitungen über mehrere Instanzen unverändert.	*
eventDate	eCH-0058: eventDateType	Datum des gemeldeten Ereignisses. Dieses Element soll nicht verwendet werden und die Information im Content abgebildet werden. Grund für diesen Entscheid: Die Information ist fachlicher Natur.	0
modificationDate	eCH-0058: eventDateType	Bearbeitungsdatum. Keine Verwendung	0

³ Im Fachbereich eAHV/IV wird für die versicherte Person das Element „insuredPerson“ im content eingefügt (siehe Kap.3.4)



Element	Typ	Verwendung	Vor- kommen
action	eCH-0058: actionType	<p>Aktionscode, definiert Art der Meldung. Siehe insbesondere die Verwendung in den Prozessen (Kapitel 3.5.7). Verwendung pro Meldungsprozess mit mehreren Meldungen (z.B. Anfrage -> Antwort) zu definieren. Dabei ist zu beachten, dass es Meldungen gibt, welche mehrere Action-Codes unterstützen (siehe auch Kap. A.3).</p> <p>„1“= „neu“/„new“ Erstmaliges Liefern von Daten. Diese Aktion darf für eine individuelle Meldung nur einmal verwendet werden.</p> <p>„3“= „Widerruf“ / „recall“ Eine zu Unrecht gelieferte Meldung rückgängig machen.</p> <p>„4“= „Korrektur“ / „correction“ Bereits gesendete, aber falsche Daten korrigieren, Rektifikat.</p> <p>„5“= „Anfrage“ / „request“ Daten beim Absender explizit verlangen.</p> <p>„6“ = „Antwort“ / „response“ Senden von Daten, welche mittels „5“ angefordert wurden.</p> <p>„8“ = „Fehlerreport“ „negativeReport“ Meldung von Fehlern zu einer Ereignislieferung</p> <p>„9“ = „Meldungsreport“ / „positiveReport“ Rückmeldung des korrekten Empfangs einer Meldung</p> <p>„10“= „Weiterleitung“ / „forward“ Daten weiterleiten</p> <p>„12“ = „Mahnung“ Wird nur selektiv zugelassen, um eine „Anfrage“ als Mahnung zu kennzeichnen.</p> <p>Die Action-Codes „8“ und „9“ sind für den headerType der Meldungen (in den jeweiligen Meldungsspezifikationen zu definieren) nicht relevant <u>und sollen nicht zugelassen werden</u>. Quittungsmeldungen werden als <eventReport> verschickt und folgen dem speziellen Aufbau welcher im Kapitel 4.2 behandelt wird. Der headerType für Quittungen wird gemeinsam für alle Meldungen im common-File definiert und im Kap. 4.2.2 behandelt.</p>	1
testDeliveryFlag	eCH-0058: testDeliveryFlag Type	<p>Ermöglicht die Markierung von Testlieferungen. Durch eine Triage muss verhindert werden, dass diese Meldungen in die Fachapplikation gelangen.</p> <p>true = „ja“ (Testlieferung) false = „nein“ (default)</p> <p>Es muss zwischen der SSK und der BSV Domäne unterschieden werden:</p> <ul style="list-style-type: none"> In der BSV Domäne werden Tests normalerweise mittels Testanschlüssen durchgeführt. Dabei wird empfohlen, dieses Flag auf „True“ zu setzen. Sind Testmeldungen in der produktiven Domäne nicht zu vermeiden, ist das Setzen des Flags zwingend. In der SSK Domäne gibt es keine fest verteilten Testanschlüsse. Entsprechend muss im produktiven Betrieb unterschieden werden, ob es sich um eine Testlieferung handelt oder nicht. 	1



Element	Typ	Verwendung	Vor- kommen
response Expected	eCH-0058:response ExpectedType (alias für xs:boolean)	Angabe ob der Absender eine fachliche Quittung (vgl. Kap. 4.2) erwartet. Es handelt sich dabei weder um eine „Response“ (definiert über den Action-Code „6“) auf einen „Request“ (definiert über den Action-Code „5“) noch um eine Rückmeldung zur Integritätsprüfung der Nutzdaten (siehe Schritt 2 in Abbildung 1), sondern um eine fachliche Quittung, welche von der Fachapplikation erstellt werden kann. Wenn eine fachliche Quittung durch dieses Element verlangt wird, so muss der Empfänger sowohl eine positive als auch eine negative Rückmeldung zurückliefern. Wird das Element nicht mitgeschickt, sollen nur negative Rückmeldungen gegeben werden (Hinweis: Aktuell im Standard nicht so dokumentiert, jedoch so vorgesehen). <ul style="list-style-type: none"> • „true“ = ja – es wird eine fachliche Quittung (positiv und negativ) erwartet • „false“ = nein – es wird keine fachliche Quittung erwartet • Element wird nicht geschickt = nur negative fachliche Quittungen werden erwartet <p>Es muss in der Meldungsspezifikation festgelegt werden, ob fachliche Quittungen verschickt werden sollen. Sollen in einem Meldungsprozess keine Fachquittungen ausgetauscht werden, wird empfohlen, im xsd den entsprechenden Wert (false) einzuschränken.</p> <p>Zu beachten ist ausserdem, dass eine Fachquittung nur verwendet werden kann, wenn alle Teilnehmer den Meldungsprozess integriert verarbeiten.</p>	1 0..1
businessCase- Closed	eCH-0058: businessCase- ClosedType	Angabe ob der Geschäftsfall mit der Meldung abgeschlossen wurde, oder nicht. (vgl. Abbildung 11) <ul style="list-style-type: none"> • „true“ = ja – der Geschäftsfall ist abgeschlossen • „false“ = nein – der Geschäftsfall ist nicht abgeschlossen 	1
attachment	common: attachmentType	Siehe Kap. 3.3.2.1	0 0..n 1..n
extension	common: extensionType oder meldungsspezi- fische Definition	Die Erweiterung ermöglicht spezifische Ergänzungen des Meldungskopfs für eine Fachbereich oder einen Meldungsprozess. Die Ergänzungen der Fachbereiche AHV/IV und SSK sind in Kap. 3.3.2.2 aufgeführt.	1

Tabelle 1: Von eCH-0058 vererbte Elemente des Headers

In der folgenden Abbildung 11 wird die Verwendung der Elemente businessProcessId, ourBusinessReferenceld und yourBusinessReferenceld detailliert. Der Geschäftsfall betrifft die Person „Hans Meier“. Innerhalb des Geschäftsfalles gibt es aufgrund einer Einsprache mehrere Sequenzen.

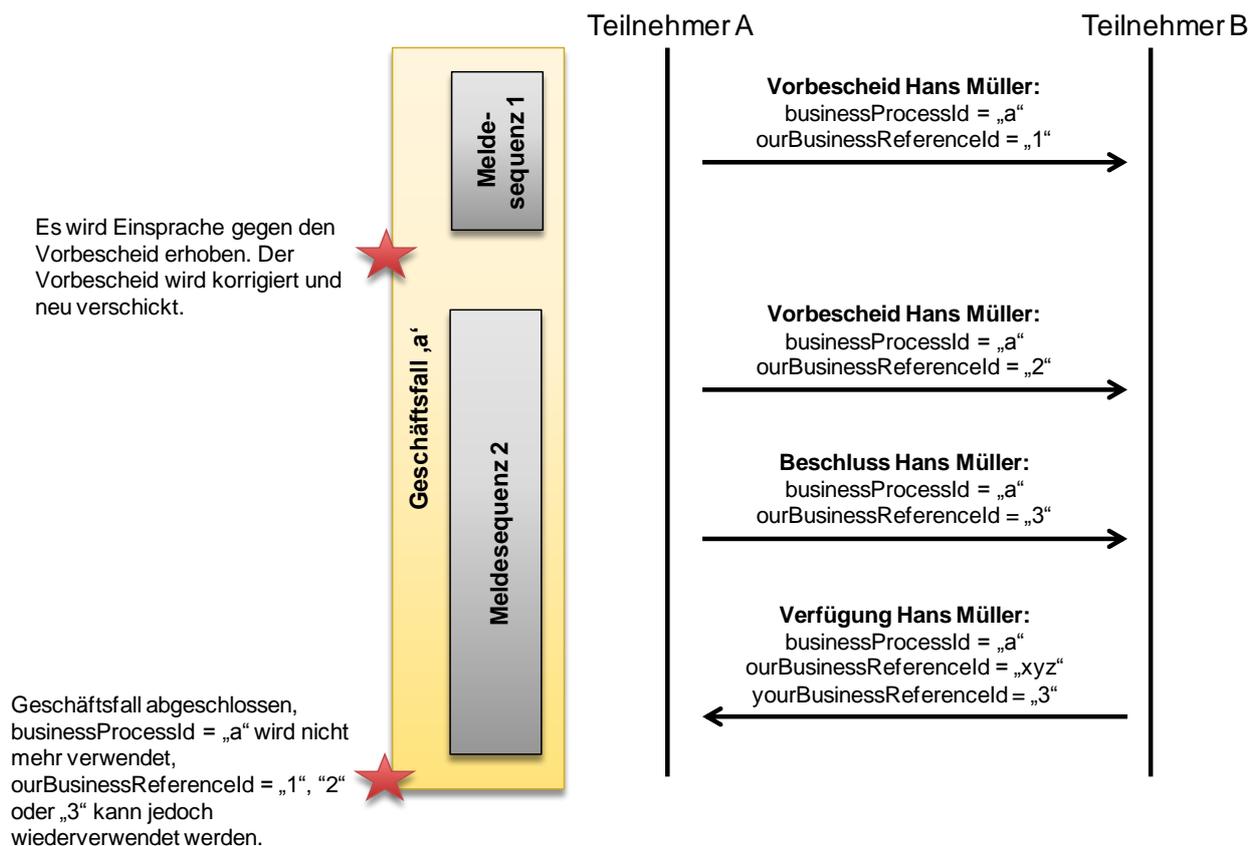


Abbildung 11: Illustration der Verwendung von businessProcessId, ourBusinessProcessId und yourBusinessProcessId.

3.3.2. Fakultative domänenspezifische Elemente

Folgende fakultative Elemente werden zusätzlich definiert, beziehungsweise sind in eCH-0058 nur abstrakt definiert (Hooks). Sie werden in der folgenden Tabelle wo möglich domänenspezifisch festgelegt. Wo keine fachbereichsspezifische Definition möglich ist, können die Elemente bei Bedarf pro Meldung konkretisiert werden.

3.3.2.1. Definition des Elements ‚attachment‘

Anhänge werden im Header durch das Element „attachment“ organisiert. Das Element wurde in eCH-0058 abstrakt (xs:anyType) definiert und wird im Subtypen konkretisiert. Es ist fachlich pro Meldung zu definieren, ob Attachments erlaubt sind und wenn ja, welche und wie viele. Entsprechend muss das Vorkommen des Elementes nicht hier, sondern in der jeweiligen Meldungsspezifikation festgelegt werden.

Der „attachmentType“ wird im fachbereichsspezifischen common-File als komplexer Typ mit den Elementen gemäss folgender Tabelle 2 definiert. In der Meldungsspezifikation kann, falls gewünscht, direkt dieses ‚attachment‘-Element aus dem common-File verwendet werden. Es dient aber auch als Vorbild, um meldungsspezifische attachmentType’s in den jeweiligen Meldungsspezifikationen zu definieren. Für die meldungsspezifischen Definitionen der attachmentType’s sollen nach Möglichkeit die Typen des im common-File definierten komplexen attachmentType benutzt werden.



Element	Typ	Verwendung	Vor- kommen
title	common: attachmentTitleType (base = xs:token, max. 100 Zeichen)	Titel des Dokuments; nach Möglich- keit sollte der Dokumententyp als Titel verwendet werden.	1
documentDate	common: documenDateType (base = xs:date)	Als Dokumenten-Datum ist das Datum der Aufnahme des Dokuments ins Dossier/ in die Fachapplikation zu setzen, was in den meisten Fällen dem Eingangsdatum ent- spricht. Dieses kann für die Einhaltung der Chronologie im Dossier dienen.	1
leadingDocument	common: leadingDocumentType (base = xs:boolean)	Das führende Dokument (z.B. Begleitbrief) wird mit „True“ markiert, alle allfälligen an- deren mitverschickten Dokumente mit „Fal- se“. Wird nur ein Dokument geschickt, ist dies somit automatisch das führende Do- kument.	1
sortOrder	common: sortOrderType (base = xs: nonNegativeInteger)	Sortierung der Dokumente (aufsteigende Nummern). Allfällige Vorgaben an die Sor- tierung (z.B. nach Datum) der Dokumente können pro Meldung vorgegeben werden. Das als leading-Dokument markierte Do- kument erhält immer die Nr. 1.	1
document Format	common: documentFormatType (base = xs:token)	Dokumenttyp als MIME-Type; mögliche Formate: <ul style="list-style-type: none"> • PDF: application/pdf • TIFF: image/tiff (Für eine allgemeinere Beschreibung vgl. Kap. 3.2.2)	1
documentType	common:documentKindType	Typ des Dokuments, nach Möglichkeit sollte in allen Fachbereichen die folgende Defini- tion verwendet werden: „(xx)(\.[0-9][0-9])**“, dabei wird xx folgendermassen reserviert: <ul style="list-style-type: none"> • 00-09: AK/IV • 10-19: SSK Hinweis: Seitens AHV/IV ist 01 bereits für AK-Dokumente, 02 für IV-Dokumente ver- geben. Bei fachbereichsübergreifendem Dokumen- tenaustausch muss in der Meldungsspezifi- kation definiert werden, welche Dokumen- tentypen-Definition (aus welchem Fachbe- reich) verwendet werden soll. Seitens AHV/IV wird im common-File nur noch die Struktur des Dokumenttyps und nicht mehr die Bedeutung der einzelnen Dokumenttypen definiert.	1



Element	Typ	Verwendung	Vorkommen
file	common: attachmentFileType	<p>Pfade zu den Dateien im Nutzdatenpaket, aus welchen das Dokument besteht und Angaben zur Reihenfolge.</p> <p>Komplexer Typ mit zwei Elementen wovon jedes genau einmal vorkommt:</p> <ul style="list-style-type: none"> pathFileName: base = xs:token; maximal 250 Zeichen internalSortOrder: sortOrderType <p>Beispiele (<pathFileName, internalSortOrder>):</p> <ul style="list-style-type: none"> <attachment/xy.pdf, 1> bei einem PDF [<attachment/aa.tiff,1>; <attachment/bb.tiff,2>; <attachment/cc.tiff,3>] bei einem mehrseitigen, im TIFF-Format gescannten Dokument <p>Die Dateinamen pathFileName müssen in UTF-8 codiert sein und werden zusätzlich mittels einer regexp auf für alle gängigen Betriebssystem verarbeitbare Zeichen eingeschränkt (alle alphanumerischen Zeichen plus: „_“, „/“, „.“, „.“ → [A-Za-z0-9/_\-.]*).</p>	1..n

Tabelle 2: Angaben zu Attachments im Header

3.3.2.2. Erweiterung des Headers (Element „extension“)

Für die Umsetzung domänen- oder meldungsspezifischer Elemente innerhalb des Meldungsrahmens ist das Element „extension“ vorgesehen. Es ist definiert als xs:anyType und kann folglich eine Anzahl weiterer Elemente beliebigen Inhalts aufnehmen.

Immer vorgesehen ist das Element contactInformation. Aus diesem Grund wird im common-File eine extension mit diesem Element definiert. Sollen weitere meldungsspezifische Elemente ergänzt werden, muss die extension im Meldungs-xsd definiert werden.

Somit ist die extension folgendermassen strukturiert:

```
<extension>
  <contactInformation></contactInformation>
</extension>
```

Element	Typ	Verwendung	Vorkommen
contactInformation	contactInformationType	<p>Angaben über die fachliche Stelle (Fachabteilung, Sachbearbeiter) welche auf Seite des Absenders für das Geschäft zuständig ist und für fachliche Fragen zur Meldung kontaktiert werden kann. (Details im nächsten Abschnitt.)</p> <p>Im contactInformation Element muss mindestens eine Pool-Emailadresse oder eine Telefonnummer angegeben werden.</p>	1

Tabelle 3: Extensions im Header



Da der eCH-0058v4 Standard einen `declarationLocalReferenceType` im Header vorsieht, diesen aber als unstrukturierte Zeichenfolge definiert, wird an seiner Stelle der „`contactInformationType`“ aus [14] (DA Dossier) übernommen. In diesem Typ wird eine fachliche Ansprechperson übermittelt, welche auf Seite des Absenders für das Geschäft zuständig ist und für fachliche Fragen zur Meldung kontaktiert werden kann. Falls die Meldung automatisch aus der Fachapplikation generiert wird und kein Kontakt zugewiesen werden kann, ist z.B. eine Hotline anzugeben. Der Typ ist wie folgt aufgebaut:

Element	Typ	Beschreibung	Vorkommen
name	xs:token (min. 1, max. 100 Zeichen)	Name	0..1
department	xs:token (min. 1, max. 100 Zeichen)	Abteilung	0..1
phone	eCH-0046:phoneNumberType ⁴	Telefon	1..1
email	eCH-0046:emailAddressType ⁴	eMail-Adresse	1..1
other	xs:token (min. 1, max. 100 Zeichen)	Freitext	0..1

Tabelle 4: `contactInformationType`

Falls nicht alle Informationen vorhanden sind oder die Meldungen automatisiert verschickt werden, sollen die Informationen einer generellen Hotline oder Supportstelle eingetragen werden. Unter „name“ wird in diesem Fall der Name der Hotline eingetragen.

3.4. Meldungsinhalt

Meldungen können strukturierten Inhalt, unstrukturierten Inhalt oder beides enthalten. Attachments zählen zu den unstrukturierten Inhalten.

Die header der Meldungen sollen unabhängig davon, ob die Meldungen ausschliesslich unstrukturierte Inhalte, nur strukturierte Inhalte oder beides enthalten, identisch aufgebaut werden. Damit wird sichergestellt, dass die Verarbeitung immer gleich umgesetzt werden kann.

In früheren Versionen dieses Standards wurden innerhalb des Meldungsrahmens (vgl. Kap. 3.3) auch Informationen zur betroffenen Person (`<object>`), sowie ein Datum, auf welches sich die Meldung bezog (`<eventDate>`), übermittelt. Diese Elemente wurden aus dem Meldungskopf entfernt, da es sich dabei um fachliche Informationen handelt. Innerhalb gewisser Meldeprozesse sind diese Elemente allerdings von grosser Bedeutung für die fachliche Verarbeitung. Aus diesem Grund können sie, falls es der betroffene Prozess verlangt, in der entsprechenden Meldungsspezifikation im Meldungsinhalt definiert werden. Dabei müssen die Elementnamen „object“ und „eventDate“ nicht zwingend übernommen werden.

Dies gilt unabhängig davon, ob eine Meldung strukturiert oder unstrukturiert übermittelt wird. Bei unstrukturierten Meldungen wird, falls nötig, ein strukturierter Teil hinzugefügt, welcher die Informationen zu „object“ und „eventDate“ enthält.

Innerhalb der Domäne des BSV und auch für den domänenübergreifenden Meldungs-austausch mit der SSK wird als „object“ – falls die Meldung im Zusammenhang mit einer versicherten Person steht – ein Element „insuredPerson“ vom Typ `naturalPersonsOASIDI`-Type verwendet (vgl. Kap. 8.2.3.1).

⁴ Der eCH-0046 basiert auf der Version 2.1.



Es kommt vor, dass fachliche Bemerkungen übermittelt werden müssen. In solchen Fällen wird nicht das Element „comment“ im header verwendet, sondern ein separates Element im content definiert. Um die beiden Elemente besser unterscheiden zu können, wird davon abgeraten, jenes im fachlichen Teil ebenfalls „comment“ zu nennen, die genaue Benennung ist in der jeweiligen Meldungsspezifikation festzulegen.

3.4.1. *Strukturierter Inhalt (<content></content>)*

Der strukturierte Meldungsinhalt wird mittels xml-Element innerhalb des content abgebildet. Hierfür muss pro Meldung definiert werden, welche Elemente (Typ, Vorkommen) zu melden sind.

3.4.2. *Unstrukturierter Inhalt*

Der unstrukturierte Meldungsinhalt wird als Attachments übermittelt. Die Meldung hat deswegen aber trotzdem einen strukturierten Header, welcher unter Anderem auch die Struktur der mitgeschickten Attachments ausweist. Dies kann beispielsweise für die Übermittlung eines Dossiers oder einer nicht standardisierbaren Meldung notwendig sein.

3.5. **Weitere Aspekte des Meldungsaufbaus**

3.5.1. *Mehrsprachigkeit*

3.5.1.1. Elemente

Die Meldungen werden in sprachunabhängiger Form ausgetauscht. Die Elemente und Typen in den XML resp. den XSD Dateien sind gemäss eCH-0018 (XML best practices) im Allgemeinen mit englischsprachigen Namen zu benennen (Ausnahmen sind z.B. landestypische juristische Begriffe wie AHV, IV, VN, usw.). Sind Namen aus zusammengesetzten Begriffen nötig, wird die CamelCase Notation mit kleinem Anfangsbuchstaben benutzt (z.B. recordNumber).

3.5.1.2. PDF Generierung

Da eCH die Sprachen nicht in jedem Standard in den XSD vollständig führt, können sie auch im angestrebten Meldungsformat nicht im XSD geführt werden. Da zudem die Anforderung besteht, dass die Sprachen der Elemente bei der PDF-Generierung zur Verfügung stehen, wird pro Meldung ein XSLT erstellt, welches in Form eines Wörterbuchs die Übersetzung für sämtliche Anwendungszwecke bereitstellt.

Das Konzept hinsichtlich Mehrsprachigkeit und die detaillierte Spezifikation der Wörterbuchdatei ist im Anhang A.1 aufgeführt.

3.5.2. *Namespaces*

Es ist zu unterscheiden zwischen den Begriffen „Domäne“ und „Fachbereich“. Aktuell relevant für dieses Dokument, sind die Domänen SSK und BSV sowie die Fachbereiche „SSK“ und „eAHV/IV“. Pro Fachbereich wird mindestens ein eigenständiger Namespace definiert, in welchem allgemeingültige Elemente definiert werden. Typen von Objekten, welche in fachbereichsübergreifenden Meldungen verwendet werden, sollen wenn möglich dem Namespace des initiierenden Fachbereichs zugeordnet sein.

Der Namespace der Meldungsspezifikationen folgt folgendem Schema (gemäss eCH-0033 [7]):



- [Fachbereichspezifischer Pfad]/xmlns/[Fachbereich]-[MT⁵]-[SMT⁶]/[Version]
- Beispiel: <http://www.steuerkonferenz.ch/xmlns/ssk-3002-000101/1>

Der Namespace der common-Files eines Fachbereichs lautet folgendermassen:

- [Fachbereichspezifischer Pfad]/xmlns/[Fachbereich]-common-[NNN⁷]/[Version]
- Beispiel: <http://www.steuerkonferenz.ch/xmlns/ssk-common-000/1>

Die 3-stellige Nummer [NNN] wird verwendet, um die gemeinsamen Definitionen auf mehrere Files aufzuteilen und dadurch bei Anpassungen mehr Flexibilität zu erreichen (resp. eine Reduktion der abhängigen und allenfalls anzupassenden xsd).

Der Namespace der common-Files eines Meldungstypen lautet folgendermassen (*Hinweis*: Ein common-File bei Meldungstypen kann nützlich sein für Definitionen, welche innerhalb des Meldeprozesses mehrmals benötigt werden, jedoch nicht allgemeingültig und deshalb nicht im common-File des Fachbereichs abgedeckt sind.):

- [Fachbereichspezifischer Pfad]/xmlns/[Fachbereich]-[MT³]-common-[NNN]/[Version]
- Beispiel: <http://www.eahv-iv.ch/xmlns/eahv-iv-2015-common-000/1>

Domäne	Fachbereich	Namespace der Basis- und komplexen Typen	Namespaces für Meldungen (Beispiel)
SSK	SSK	http://www.steuerkonferenz.ch/xmlns/ssk-common/1	http://www.steuerkonferenz.ch/xmlns/ssk-3002-000101/1
BSV	eAHV/IV	http://www.eahv-iv.ch/xmlns/eahv-iv-common/1	http://www.eahv-iv.ch/xmlns/eahv-iv-0001-000101/1
	BSV	http://www.bsv.admin.ch/xmlns/ebsv-cNNN/1	http://www.bsv.admin.ch/xmlns/ebsv-2028-000001/1/ebsv-2028-000001-1-0.xsd
eCH	eCH	http://www.ech.ch/xmlns/eCH-0097/1 (Beispiel)	-
BFS	BFS	-	http://www.bfs.admin.ch/xmlns/bfs-5051-000101/1
PV	PV	http://www.gdk-cds.ch/xmlns/pv-common/1	http://www.gdk-cds.ch/xmlns/pv-5201-000101/1
ZAS	ZAS	http://www.zas.admin.ch/RC	

Table 5: Namespaces der Domänen

Es ist dabei nicht nötig, dass die Namespace URL auch tatsächlich existiert.

Gemäss der in eCH-0018 ausgesprochenen Empfehlung soll bei der Einbindung von Namespaces nicht mittels des xsi:schemaLocation Attributs auf die Schemata verwiesen werden. Als relevante Identifikation eines eingebundenen Schemas soll der entsprechende Namespace verwendet werden. Um die jeweiligen Schemata zu finden, generiert der sM-Client z.B. ein catalog.xml (siehe <http://xerces.apache.org/xerces2-j/faq-xcatalogs.html#faq-2>). Das catalog.xml enthält eine Zuordnung von Namespaces zu lokalen Speicherorten. Das setzt voraus, dass sowohl die eCH-Schemata als auch die fachspezifischen Schemata Bestandteil der installierten Applikation sind und als lokale Kopie vorliegen.

⁵ messageType

⁶ subMessageType

⁷ 3-stellige Nummer



3.5.3. Versionierung der XSD

Jede neue Version, also Änderung oder Erweiterung der XML-Schemadefinition, erfordert eine neue Versionsnummer, die sich aus einem Major- und einem Minor-Teil zusammensetzt.

- Erhöhung des Minor-Teils: Genügt, falls jede Meldung, die bezüglich der alten Schemadefinition gültig war, auch bezüglich der neuen gültig bleibt (Rückwärtskompatibilität). Bei einem Minor Update kann im sM-Client für die verschiedenen Minor-Versionen kein unabhängiges Gültigkeitsdatum gesetzt werden. Wird ein solches benötigt, ist ein Major Update notwendig.
- Erhöhung des Major-Teils: Gemäss Empfehlung in eCH-0033 erforderlich, falls alte Meldungen bezüglich der neuen Schemadefinition nicht mehr zwingend gültig sind (keine Rückwärtskompatibilität). Der Minor-Teil wird dabei auf 0 gesetzt.

Dabei ist zu beachten, dass der sM-Client zwar das parallele Führen von zwei Major-Versionen erlaubt, jedoch nicht von zwei Minor-Versionen.

Gemäss eCH-0018 wird der Minor-Teil der Versionsnummer im Document durch das Attribut `minorVersion` angegeben. Beispiel für den Minor-Teil der Version 1.3:

```
minorVersion="3"
```

Der Major-Teils der Versionsnummer muss im Namespace nach dem zuletzt vorkommenden „/“ angegeben werden. Beispiel für den Major-Teil der Version 1.3:

```
xmlns:eahv-iv-2001-000101="http://www.eahv-iv.ch/xmlns/eahv-iv-2001-000101/1"
```

Zusätzlich wird die vollständige Version (major und minor) im Attribut `version` des Elementes `schema` angegeben:

```
<xs:schema xmlns:... version="1.3">
```

Zum Beispiel wird eine Schemadefinition der Versionsnummer 1.0 zu einer Versionsnummer mit 1.1 erweitert, wenn sich für ein bestimmtes Element die Menge der zulässigen Werte von „0 bis 10“ auf „0 bis 20“ erweitert oder ein neues optionales Element hinzugefügt wird und sonst keine Änderungen gemacht werden (neues Schema ist rückwärtskompatibel).

Hingegen lautet die neue Versionsnummer 2.0, wenn z.B. die Menge der zulässigen Werte für dieses Element von „0 bis 10“ auf „0 bis 5“ eingeschränkt wird oder ein optionales Element neu als Pflichtelement definiert wird (neues Schema ist nicht rückwärtskompatibel).

Die Versionsnummer der Meldungsspezifikation und jene des zugehörigen XSD-Files sollen soweit möglich synchron gehalten werden. Zusätzlich soll in der Meldungsspezifikation (z.B. in der change history) angegeben werden, welches die aktuelle Version des xsd ist. Das Gleiche gilt auch für das vorliegende Dokument und die common-Files.

Die xsd sollen folgendermassen bezeichnet werden (wobei x für die Major-, y für die Minor-Version steht):

- eahv-iv-common-58v4-x-y.xsd
- eahv-iv-2011-000103-x-y.xsd
- ssk-common-58v4-x-y.xsd
- ssk-3002-000101-x-y.xsd



Die Namenskonventionen der Sprachfiles (xsl) sind im Anhang A.1 festgehalten.

3.5.4. *Kodierung*

Alle Meldungen müssen die Kodierung UTF-8 (Unicode Transformation Format 8) verwenden, d.h., die entsprechenden XML-Dokumente müssen mit der Zeile

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

beginnen.

Gemäss eCH-0018 (XML best practices) sollte das XML so formatiert werden, dass es gut menschenlesbar ist (Whitespaces wie Zeilenumbrüche und Einrückungen), soweit dies aus Platzgründen vertretbar ist. Die Applikationen dürfen aber nicht davon ausgehen, dass der Whitespace in XML-Dokumenten in einer bestimmten Art formatiert ist.

3.5.5. *XML Escape Sequenzen*

Es werden ausschliesslich folgende fünf XML Escape Sequenzen verwendet:

- " = "
- ' = '
- < = <
- > = >
- & = &

3.5.6. *Annotationen in Meldungs-XSD*

Die Meldungsbezeichnung soll im xsd dreisprachig geführt werden. Damit ist beim Öffnen des XSDs bekannt, um welche Meldung es sich handelt. Weitergehende Annotationen (Element) sind nicht notwendig.

Beispiel:

```
<xs:annotation>  
  <xs:documentation xml:lang="de">xy</xs:documentation>  
  <xs:documentation xml:lang="fr">xy</xs:documentation>  
  <xs:documentation xml:lang="it">xy</xs:documentation>  
</xs:annotation>
```

3.5.7. *Umgang mit NULL-Werten*

Schickt ein Teilnehmer ein leeres xml-Element, kann der Empfänger nicht unterscheiden, ob diese Information dem Absender nicht zur Verfügung steht oder ob das Element mit Absicht leer gelassen wurde (z.B. der Teilnehmer kennt die Adresse einer Person nicht). Damit solche Missverständnisse vermieden werden können, sollen grundsätzlich keine leeren xml-Elemente verschickt werden (ausser es soll explizit ein Leerstring übermittelt werden). Steht eine Information nicht zur Verfügung, ist das Element komplett wegzulassen. Diese Handhabung bedingt, dass bei solchen Elementen das Vorkommen als 0..1 definiert wird.



4. Einbettung in die Meldungsprozesse

Die im eCH-0058v4 Standard aufgeführten Prozesse (action-Codes) müssen durch das Meldungsformat unterstützt werden. Nachfolgend werden diese „Spezialmeldungen“ definiert.

Die in früheren Versionen des Meldungsformats vorgeschriebene, positive Protokollquittung entfällt (diese hat jeweils der sM-Client verschickt). Die Fehlerbehandlung technischer Fehler wird im Kap. 4.4 beschrieben.

Falls es aus fachlichen Gründen nötig ist, den Empfang einer Meldung zu quittieren, so muss dies als Teil des fachlichen Meldungsprozesses festgelegt werden.

4.1. Aufbau des Headers bei verschiedenen action-Codes

Wird eine neue Meldung spezifiziert, muss festgelegt werden, welche der in den folgenden Kapiteln detaillierten Aktionen grundsätzlich zugelassen werden sollen und welche nicht benötigt werden. Dabei sollte der Grundsatz gelten, so wenige Action-Codes wie möglich zuzulassen. Damit reduziert sich der Integrationsaufwand in den Fachapplikationen.

Grundsätzlich gibt es die folgenden beiden Typen von Meldungsprozessen:

- 1) Unidirektionaler Meldungsprozess (ein Teilnehmer schickt eine gewisse Information einem anderen Teilnehmer). Diese Meldungen werden mit dem action-Code 1 (=neu) gekennzeichnet.
- 2) Bidirektionaler Meldungsprozess (ein Teilnehmer schickt eine Anfrage und erhält darauf eine Antwort). Diese Meldungen werden mit den action-Codes 5 (=Anfrage) und 6 (=Antwort) gekennzeichnet.

Natürlich sind beliebige Erweiterungen dieser Grundtypen denkbar und können in den Meldungsspezifikationen entsprechend definiert werden.

Bidirektionale Meldungsprozesse werden grundsätzlich mit separaten Meldungen pro Prozessschritt umgesetzt werden, wobei sich die Meldungen untereinander referenzieren müssen.

Die Meldungen mit unterschiedlichen action-Codes unterscheiden sich jeweils dadurch, dass gewisse Elemente geliefert oder nicht geliefert werden und auch durch Unterschiede im Inhalt (content). Die verschiedenen Arten der Meldungen werden dabei mit Hilfe des Elements „action“ identifiziert. Im Folgenden wird der Aufbau der Meldungen sortiert nach Action-Code vorgestellt. Die eigentlichen Meldungsinhalte werden in den jeweiligen Spezifikationen definiert.

4.1.1. Meldungsaufbau bei action="1" (neu)

Erstmaliges Liefern von Daten. Diese Aktion darf für eine individuelle Meldung nur einmal verwendet werden.

Element	Verwendung	Vorkommen
originalSenderId	Nicht gesetzt	0
referenceMessageld	Nicht gesetzt	0
businessProcessId	Gemäss Tabelle 1	1



Element	Verwendung	Vorkommen
yourBusinessReferenceld	Nicht gesetzt	0
initialMessageDate	Nicht gesetzt	0
action	„1“ = „Neue Meldung“	1
responseExpected	Gemäss Tabelle 1 Default-Wert: „False“	1
businessCaseClosed	Gemäss Tabelle 1 Default-Wert: „True“	1

Die restlichen Elemente werden analog des „headerType“ aus Kapitel 3.3 definiert.

Weitere Meldungen von A nach B im selben Geschäftsprozess, die technisch gesehen ebenfalls initiale Meldungen sind (z.B. wenn A an B zuerst den Vorbescheid und dann den Beschluss sendet, ohne dass B irgendwann in der Zwischenzeit auf den Vorbescheid geantwortet hätte), werden auf gleiche Weise wie initiale Meldungen definiert. Dabei soll die businessProcessId identisch mit derjenigen aus der initialen Meldung sein.

4.1.2. *Meldungsaufbau bei action=“3“ (Widerruf)*

Die Möglichkeit, eine Meldung zu widerrufen ist mit dem actionCode „3“ möglich. Ein Widerruf ist im Normalfall nicht vorgesehen, und ist bei explizitem fachlichen Bedarf in der Meldungsspezifikation zu definieren.

Die zu widerrufende Meldung wird mit folgenden Spezifitäten nochmals gesendet; Attachments müssen nicht nochmals angefügt werden.

Element	Verwendung	Vorkommen
originalSenderId	Nicht gesetzt	0
referenceMessageld	Enthält die ID der ursprünglichen, zu widerrufenden Meldung	1
businessProcessId	businessProcessId der ursprünglichen Meldung.	1
ourBusinessReferenceld	ourBusinessReferenceld der ursprünglichen Meldung.	Analog dem Vorkommen der zu Widerrufenden Meldung
yourBusinessReferenceld	yourBusinessReferenceld der ursprünglichen Meldung	Analog dem Vorkommen der zu Widerrufenden Meldung
initialMessageDate	initialMessageDate der ursprünglichen Meldung.	Analog dem Vorkommen der zu Widerrufenden Meldung
action	3“= „Widerruf“ / „recall“	1
responseExpected	Angabe, ob der Absender der Meldung eine fachliche Quittung erwartet oder nicht. Default-Wert „False“ (falls die Meldungsverarbeitung sowohl beim Sender als auch Empfänger integriert erfolgt, kann die Verwendung der fachlichen Quittung in der Meldungsspezifikation definiert werden.)	1
businessCaseClosed	Angabe, ob aus Sicht des Absenders der Meldung der Geschäftsfall mit dieser Meldung abgeschlossen ist oder nicht. Default-Wert: „True“	1



4.1.3. Meldungsaufbau bei action="4" (Korrektur)

Es ist möglich, auf eine Fehlermeldung des Empfängers hin oder auf eine fachlich falsche Meldung ein Rektifikat (Korrektur der ursprünglichen Nachricht) zu senden. Eine Korrekturmeldung ist im Normalfall nicht vorgesehen, und ist bei explizitem fachlichen Bedarf in der Meldungsspezifikation zu definieren.

Eine Korrektur wird realisiert, indem die gleiche Meldung mit folgenden Spezifitäten nochmals gesendet wird:

Element	Verwendung	Vorkommen
originalSenderId	Nicht gesetzt	0
referenceMessageId	Enthält die ID der ursprünglichen, zu korrigierenden Meldung	1
businessProcessId	businessProcessId der ursprünglichen Meldung.	1
ourBusinessReferenceId	ourBusinessReferenceId der ursprünglichen Meldung.	Analog dem Vorkommen der zu Widerrufenden Meldung
yourBusinessReferenceId	yourBusinessReferenceId der ursprünglichen Meldung	Analog dem Vorkommen der zu korrigierenden Meldung
initialMessageDate	initialMessageDate der ursprünglichen Meldung.	Analog dem Vorkommen der zu korrigierenden Meldung
action	4"= „Korrektur“ / „correction“	1
responseExpected	Angabe, ob der Absender der Meldung eine fachliche Quittung erwartet oder nicht. Default-Wert „False“ (falls die Meldungsverarbeitung sowohl beim Sender als auch Empfänger integriert erfolgt, kann die Verwendung der fachlichen Quittung in der Meldungsspezifikation definiert werden)	1
businessCaseClosed	Angabe, ob aus Sicht des Absenders der Meldung der Geschäftsfall mit dieser Meldung abgeschlossen ist oder nicht. Default-Wert: „True“	1

4.1.4. Meldungsaufbau bei action="5" (Anfrage)

Daten werden beim Absender explizit verlangt (z.B. Bestellung). Eine Anfrage wird mit einer „Response“ (action="6") beantwortet. Die „Response“ ist nicht mit einer fachlichen Quittung zu verwechseln.

Element	Verwendung	Vorkommen
originalSenderId	Nicht gesetzt	0
referenceMessageId	Nicht gesetzt	0
businessProcessId	Gemäss Tabelle 1	1
yourBusinessReferenceId	Nicht gesetzt	0
initialMessageDate	Nicht gesetzt	0
action	„5“ = „Anfrage“ / „Request“	1
responseExpected	Gemäss Tabelle 1 Default-Wert: „false“	1
businessCaseClosed	Gemäss Tabelle 1 Fixer-Wert: „false“	1



4.1.5. Meldungsaufbau bei action="6" (Antwort)

Senden von Daten, welche mittels „5“ angefordert wurden. Es handelt sich dabei um eine fachliche Antwort mit den angeforderten Daten. Einer solchen Antwort geht immer ein „Request“ (action="5") voraus und sie ist nicht mit einer fachlichen Quittung (action="8" resp. 9) zu verwechseln.

Element	Verwendung	Vorkommen
originalSenderId	Nicht gesetzt	0
referenceMessageld	Technische Referenz auf die Meldung, auf welche sich die Antwort bezieht. Bei „halb-integrierten“ Prozessen, welche von Sender auf Papier gestartet wurden und dann elektronisch beantwortet werden, wird die referenceMessageld nicht gesetzt.	0..1 1
businessProcessId	Gemäss Tabelle 1 Die eindeutige Geschäftsfall-Id wird vom Absender bzw. vom ursprünglichen Absender initiiert.	1
yourBusinessReferenceld	Referenz auf die Meldung, auf welche sich die Antwort bezieht. Bei „halb-integrierten“ Prozessen, welche von Sender auf Papier gestartet wurden und dann elektronisch beantwortet werden, wird die yourBusinessReferenceID nicht gesetzt oder das entsprechende Referenzelement der Papiermeldung abgeschrieben.	Analog ourBusinessReferenceld der entsprechenden Anfrage 0..1
initialMessageDate	Nicht gesetzt	0
messageType/subMessageType	Gemäss Vorgabe (nicht gleich wie erste Meldung)	1
action	„6“ = „Antwort“	1
responseExpected	Gemäss Tabelle 1 Default-Wert „False“	1
businessCaseClosed	Gemäss Tabelle 1 Default-Wert: „True“	1

4.1.6. Meldungsaufbau bei action="10" (Weiterleitung)

Weiterleitung sollten nicht verwendet werden. Stattdessen wird empfohlen, mit dem Absender Kontakt aufzunehmen und ihn über den Falschversand zu informieren. Würden Meldungen weitergeleitet, wird der ursprüngliche Absender den Fehler in seiner Applikation nicht bemerken und auch zukünftig Meldungen an falsche Empfänger schicken.

Soll die Weiterleitung trotzdem zugelassen werden, ist sie in der Meldungsspezifikation zu definieren.

Der fachliche Teil einer weitergeleiteten Meldung entspricht der ursprünglichen Meldung. Der headerType wird gemäss folgenden Vorgaben abgeändert:



Element	Verwendung	Vorkommen
senderId	Absender der Weiterleitung entsprechend dem sedex-Adressierungskonzept.	1
originalSenderId	Originaler Absender der Meldung	1
referenceMessageId	Wird nicht benötigt.	0
businessProcessId	Gleicher Inhalt wie originale Meldung	1
ourBusinessReferenceId	Gleicher Inhalt wie originale Meldung	0..1
yourBusinessReferenceId	Wird nicht benötigt. (Das Element ist nur nötig für Antwortmeldungen und Antwortmeldungen sind ausgeschlossen)	0
sendingApplication	Identifikation der meldungserstellenden Applikation (Weiterleitungs-Meldung)	1
initialMessageDate	Datum der originalen Meldung. Bei mehrfacher Weiterleitung wird immer das Datum der ersten Meldung weitergeleitet.	1
action	„10“= „Weiterleitung“ / „forward“	1
responseExpected	Gemäss Tabelle 1 Default-Wert „False“	1
businessCaseClosed	Gemäss Tabelle 1 Default-Wert: „True“	1
contactInformation	Angaben aus der der originalen Meldung	1

4.1.7. Meldungsaufbau bei action=“12“ (Mahnung)

Eine Mahnung ist im Normalfall nicht vorgesehen, und ist bei explizitem fachlichen Bedarf in der Meldungsspezifikation zu definieren.

Element	Verwendung	Vorkommen
originalSenderId	Nicht gesetzt	0
referenceMessageId	Message ID der initialen Meldung, auf welche sich die Mahnung bezieht	1
businessProcessId	Gemäss Tabelle 1. Die eindeutige Geschäftsfall-Id wird vom Absender bzw. vom ursprünglichen Absender initiiert.	1
yourBusinessReferenceId	Nicht gesetzt	0
initialMessageDate	Datum der initialen Meldung, auf welche sich die Mahnung bezieht.	1
action	„12“ = „Mahnung“ / „Reminder“	1
responseExpected	Gemäss Tabelle 1 Default-Wert „False“, d.h. es wird keine fachliche Quittung verlangt.	1
businessCaseClosed	Gemäss Tabelle 1 Default-Wert: „False“	1

4.2. Fachliche Quittungen (action-Code 8 und 9)

Gemäss Abbildung 3 werden auf Stufe sedex-Client in jedem Fall Quittungen ausgetauscht. Dies wird durch den sedex-Client gehandhabt.



Bei Meldungen nach eCH-0058v2/3 wurden Protokollquittungen (technische Quittungen) ausgetauscht um dem Absender zu bestätigen, dass eine empfangene Meldung geöffnet und validiert werden konnte. Protokollquittungen haben bisher jedoch mehr Probleme verursacht als gelöst. Deshalb werden sie bei Meldungen nach eCH-0058v4 nicht mehr verwendet. Einzig nicht-valide Meldungen sollen an den Absender zurückgeschickt werden (siehe Kap.4.4), dies jedoch nicht im Sinne einer Quittung.

Zusätzlich können optional fachliche Quittungen zwischen den Fachapplikationen ausgetauscht werden (siehe Schritt 3 in Abbildung 1). Damit wird eine end-to-end-Protokollierung des Meldungs-austausches ermöglicht. Es ist jedoch zu beachten, dass die Verwendung der fachlichen Quittungen nur möglich ist, wenn beide Partner die entsprechende Meldung integriert verarbeiten (= Verarbeitung in der Fachapplikation und nicht Ausdruck via sM-Client). Damit ist die Verwendung der fachlichen Quittungen nur möglich, wenn eine Meldung von allen Partnern integriert verarbeitet wird.

Bei Sammelmeldungen soll pro Einzelmeldung eine fachliche Quittung erstellt werden. Diese Quittungen können wiederum als Sammelmeldung übertragen werden. Dabei ist es nicht zwingend, alle Quittungen einer Sammelmeldung in einer gemeinsamen Sammelmeldung zurückzumelden.

4.2.1. Aufbau der fachlichen Quittungen

Fachliche Quittungen werden mit Hilfe der Action-Codes 8 und 9 gekennzeichnet, so bedeutet action="8" eine negative und action="9" eine positive fachliche Quittung. Die Meldung ist grundsätzlich mit einem headerType und einem infoType nach eCH-0058v4 aufgebaut. Diese beiden Elemente werden als eventReport verschickt (siehe auch Abbildung 12):

```
<eventReport>
  <header></header>
  <info></info>
</eventReport>
```

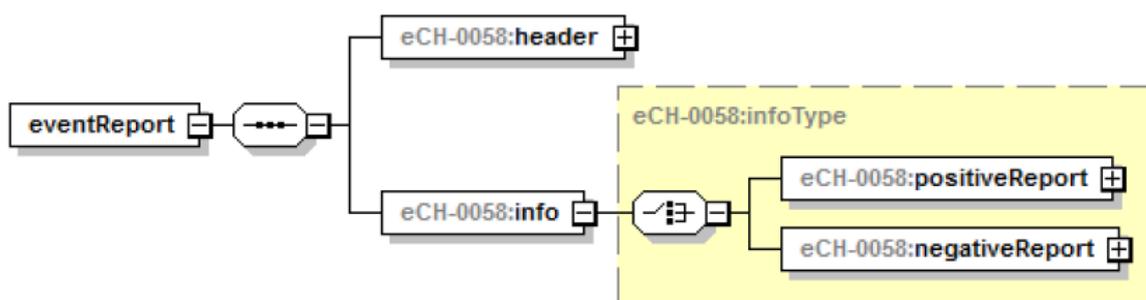


Abbildung 12: Aufbau von asynchronen fachlichen Quittungen.

Damit die Quittungen nicht in jeder Meldungsspezifikation komplett definiert werden müssen, werden der headerType und der infoType im common-File definiert. Die entsprechenden Typen (headerType und infoType) werden in den folgenden Kapiteln 4.2.2 und 4.2.3 festgelegt. Der eventReport muss in der Meldung selber definiert werden, jedoch nur, wenn die fachliche Quittung in der entsprechenden Meldung verwendet wird. Dabei können direkt die Definitionen von headerType und infoType aus dem common-File referenziert werden, sofern die entsprechenden Definitionen ausreichend sind.



4.2.2. Header von fachlichen Quittungen

Ein Grossteil der aufgeführten Elemente sind identisch mit jenen, die im Kap. 3.3.1 „von eCH-58 vererbte Elemente“ aufgelistet sind. Gewisse Elemente machen für den Header der Quittung aber keinen Sinn und werden weggelassen. Um eine eindeutige Definition angeben zu können, werden im Folgenden alle Elemente des Headers der Quittung nochmal aufgelistet, unabhängig davon, ob sie mit jenen aus dem Kapitel 3.3.1 übereinstimmen, oder nicht. Der Quittungsheader wird als eigenständiger headerType im common-file definiert.

Element	Typ	Verwendung	Vorkommen
senderId	eCH-0058: participantIdType	Analog zu Kap. 3.3.1“	1
originalSenderId	eCH-0058: participantIdType	Wird nicht verwendet	0
declarationLocalReference	eCH-0058: declarationLocalReferenceType	Wird nicht verwendet	0
recipientId	eCH-0058: participantIdType	senderId aus der originalen und zu quittierenden Meldung	1
messageId	eCH-0058: messageIdType	Analog zu Kap. 3.3.1	1
referenceMessageId	eCH-0058: messageIdType	Technische Referenz auf die Meldung, auf welche sich die Quittung bezieht..	1
businessProcessId	eahv-iv-common: businessProcessIdType	businessProcessId aus der originalen und zu quittierenden Meldung	1
ourBusinessReferenceId	eCH-0058: businessReferenceIdType	Wird nicht verwendet	0
yourBusinessReferenceId	eCH-0058: businessReferenceIdType	ourBusinessReferenceId aus der originalen und zu quittierenden Meldung	0..1
uniqueIdBusinessTransaction	eCH-0058: uniqueIdBusinessTransactionType	Wird nicht verwendet	0
messageType	eCH-0058: messageTypeType	messageType aus der der originalen und zu quittierenden Meldung	1
subMessageType	eCH-0058: subMessageTypeType	subMessageType der originalen und zu quittierenden Meldung	1
sendingApplication	eCH-0058: sendingApplicationType	Analog zu Kap. 3.3.1	1
partialDelivery	eCH-0058: partialDeliveryType	Wird nicht verwendet	0
subject	eCH-0058: subjectType	Falls das subject verwendet wird, soll es folgendermassen gefüllt werden: - Positive Quittung: "Positive Quittung – 'Subject der quittierten Meldung'" - Negative Quittung: "Negative Quittung – 'Subject der quittierten Meldung'"	0..1
comment	eCH-0058: commentType	Wird nicht verwendet	0



Element	Typ	Verwendung	Vorkommen
initialMessageDate	eCH-0058: messageDateType	Wird nicht verwendet	0
eventDate	eCH-0058: eventDateType	Wird nicht verwendet	0
modificationDate	eCH-0058: eventDateType	Wird nicht verwendet	0
action	eCH-0058: actionType	Aktionscode, definiert Art der Meldung. Im Falle der Quittung nur positive oder negative Bestätigung. <ul style="list-style-type: none"> „8“ = „Fehlerreport / negativeReport“ Meldung von Fehlern zu einer Meldung. „9“ = „Meldungsreport / positiveReport“ Rückmeldung der korrekten Verarbeitung einer Meldung. 	1
testDeliveryFlag	eCH-0058: testDeliveryFlag- Type	testDeliveryFlag der originalen und zu quittierenden Meldung	1
responseExpected	eCH-0058: response ExpectedType (alias für xs:boolean)	Angabe, ob der Absender der Meldung eine Antwort erwartet oder nicht. <ul style="list-style-type: none"> Bei negativen Quittungen (Action-Code „8“) wird keine Antwort erwartet⁸. Fixer Wert: „False“⁹ Bei positiven Quittungen (Action-Code „9“) wird keine Antwort erwartet. Fixer Wert: „False“ 	1
businessCaseClosed	eCH-0058: businessCase- ClosedType	Angabe, ob aus Sicht des Absenders der Meldung der Geschäftsfall mit dieser Meldung abgeschlossen ist oder nicht. <ul style="list-style-type: none"> Bei negativen Quittungen (Action-Code „8“) gilt der Geschäftsfall nicht als abgeschlossen. Fixer Wert: „False“ Bei positiven Quittungen (Action-Code „9“) gilt der Geschäftsfall als abgeschlossen. Fixer Wert: „True“ 	1
attachment	xs:anyType	Wird nicht verwendet	0
extension	xs:anyType	Wird nicht verwendet	0

4.2.3. Inhalt von fachlichen Quittungen (infoType)

Der Inhalt der fachlichen Quittungen ist im infoType festgelegt, welcher im common-file des Fachbereichs definiert wird. Es wird unterschieden zwischen negativer und positiver fachlicher Quittung. Die folgenden zwei Elemente sind als Choice implementiert, so dass eine Quittung nur jeweils genau ein Element enthalten kann:

⁸ Damit muss eine Fachapplikation fehlerhafte, eingegangene Meldungen nicht tracken. Es liegt in der Verantwortung der sendenden Applikation, die negative Quittung des Empfängers auszuwerten und die Meldung nochmals zu verschicken.

⁹ Diese Definition ist eine Abweichung vom Standard ([11], Kapitel 3.4.33). Diese Entscheidung wurde zugunsten einer einheitlichen Bedeutung des Elements „responseExpected“ gefällt: Das Element bezieht sich ausschliesslich auf fachliche Quittungen.



Element (Choice)	Typ	Verwendung	Vor- kommen
negativeReport	complexType	<p>Sequence von 2 Elementen:</p> <ul style="list-style-type: none"> • notice <ul style="list-style-type: none"> ○ <i>code</i>, 1..1, <i>xs:token</i> (Länge fixiert: 5, <i>regexp=([EW][0-9]{4}){0{5}}</i>): Kodierung des Ereignisses, 4 stelliger Fehlercode mit vorangehendem Bezeichner für den Typ der Quittung. Es werden zwei Typen unterschieden <ul style="list-style-type: none"> ▪ Fehler (E) ▪ Warnungen (W) Die folgenden Codes werden für global definierte Fehler und Warnungen reserviert: <ul style="list-style-type: none"> ▪ Code 00000 für eine korrekte Meldung (nur innerhalb positive-Report) ▪ Code Wxxxx [xxxx=0001-9999] für Warnungen (positive und negative Quittungen) ▪ Code Exxxx [xxx=0001-9999] für Fehler (nur innerhalb negativeReport) Der komplette Code, inklusive Bezeichner ist immer 5-stellig. Codes grösser als X1000 sind für fachliche Fehler, welche in den jeweiligen Meldungsspezifikationen definiert werden, reserviert. ○ <i>codeDescription</i>, 0..1, <i>xs:token</i> (Länge 1 bis 300): Maschinelle Beschreibung des Ereignisses (automatisch generierter Text). Bei einem Inter-Event-Fehler (W0002) wird hier die Meldung (oder Leistung) angezeigt, mit der ein Konflikt besteht. ○ <i>faultyField</i>, 0..n, <i>xs:token</i> (Länge 1 bis 100): Name des fehlerhaften Feldes (nur sinnvoll für negative Quittung) ○ <i>timestamp</i>, 0..1, <i>xs:dateTime</i>: Zeitstempel an dem der Fehler passiert ist ○ <i>comment</i>, 0..1, <i>xs:string</i> (Länge 1 bis 5000): Beschreibung des Fehlers bei manueller Bearbeitung. • data <ul style="list-style-type: none"> ○ <i>echoRequest</i>, 0..1, <i>xs:anyType</i>: Kopie der ursprünglichen Meldung (komplettes XML), von Fall zu Fall zu entscheiden ob sinnvoll (je nach Grösse der Meldung) ○ <i>xsdVersion</i>, 0..1, <i>xs:token</i> (Länge 1 bis 10): Version und Name des Schemas, gegen welches die Meldung validiert wurde ○ <i>additionalData</i>, 0..1, <i>xs:anyType</i>: um zusätzliche Informationen mitzuteilen 	<p>1..n</p> <p>0..1</p>



Element (Choice)	Typ	Verwendung	Vorkommen
positiveReport	Analog dem negativeReport	Positive Quittungen erhalten den Aktionscode 9 (action= 9). Diese wird aufbereitet, falls responseExpected=true und die Verarbeitung ohne Fehler abgeschlossen worden ist. Im Element „Code“ wird bei fehlerfreiem Empfang der Code 00000 übermittelt. Ist ein Warning aufgetreten, wird der entsprechende Warning-Code verschickt.	

Die folgenden Fehler- und Warning-Codes sind übergreifend festgelegt. Diese Liste kann bei Bedarf (auch mit Warnungen) erweitert werden.

00000	Korrekte Meldung
W0001	Nicht referenzierte Dokumente (im Attachment_X Verzeichnis vorhanden, aber im Header nicht referenziert)
W0002	Inter-Event-Fehler: Für Fälle, bei denen sich zwei an sich korrekte Meldeprozesse nicht überlappen dürfen und deswegen die Kombination einen Fehler darstellt.
E0001	nicht XSD-valides xml
E0002	Fehlende Attachments (im Header referenziert, jedoch nicht vorhanden)
E0003	nicht zuständig (für den Fall dass ein Empfänger Nachrichten erhält für die er nicht zuständig ist)
E0004	Attachement nicht lesbar
E0005	Nicht autorisiert (keine Berechtigung, vor allem für synchrone Meldungen)
E0006	Service nicht verfügbar (nur im synchronen Fall)
E0007	senderId des eCH-0090-Umschlags nicht identisch mit demjenigen im Meldungsheader (eCH-0058)
E0008	recipientId des eCH-0090-Umschlags nicht identisch mit demjenigen im Meldungsheader (eCH-0058)
E0009	messageType des eCH-0090-Umschlags nicht identisch mit demjenigen im Meldungsheader (eCH-0058)
E0010	Meldungen und Antwortmeldungen (= fachliche Quittungen) in der gleichen Sammelmeldung
E0999	Weitere Fehler

Tabelle 6: Fehler- und Warning-Codes.

4.3. Beispiele von Meldungsabläufen mit fachlichen Quittungen

In den folgenden Kapiteln werden Beispiele von Meldungsabläufen mit fachlichen Quittungen visualisiert.



4.3.1. Neue Meldung mit positiver Quittierung

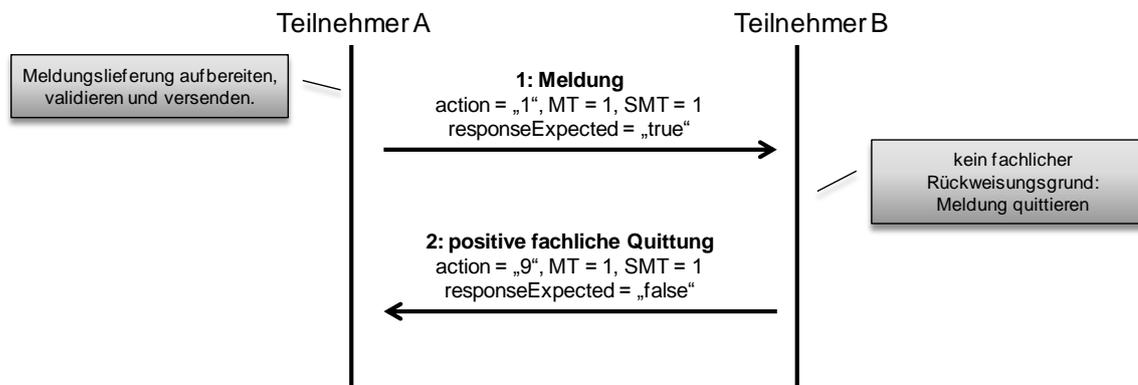


Abbildung 13: Neue Meldung mit positiver Quittierung.

4.3.2. Neue Meldung mit negativer Quittierung

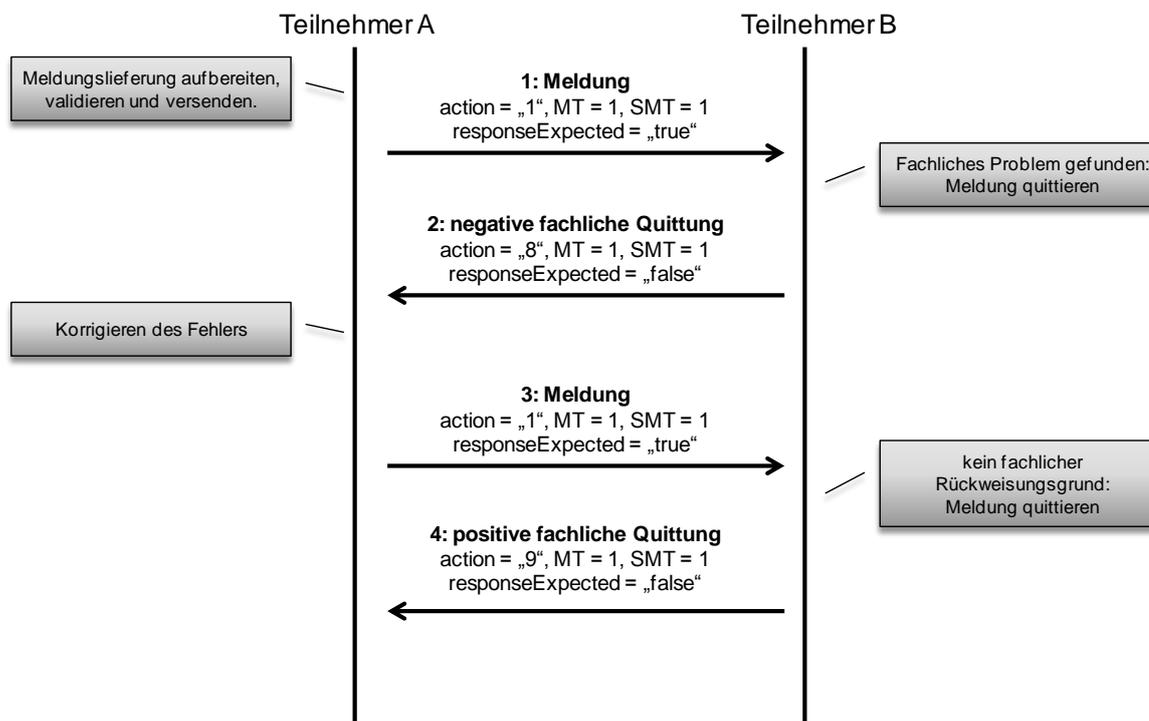


Abbildung 14: Neue Meldung mit negativer Quittierung.

Hinweis: Anstelle der Meldung 3 mit action Code 1 schreibt der eCH-0058v4-Standard die Verwendung einer Korrektur (action = 4) vor. Im Rahmen der Meldungen der Fachbereiche SSK und AHV/IV werden Korrekturen jedoch im Sinne von inhaltlichen Korrekturen (z.B. fehlerhafte Werte bei der initialen Lieferung) verschickt, welche vom Empfänger nicht erwartet werden.



4.3.3. Anfrage mit negativer Quittierung

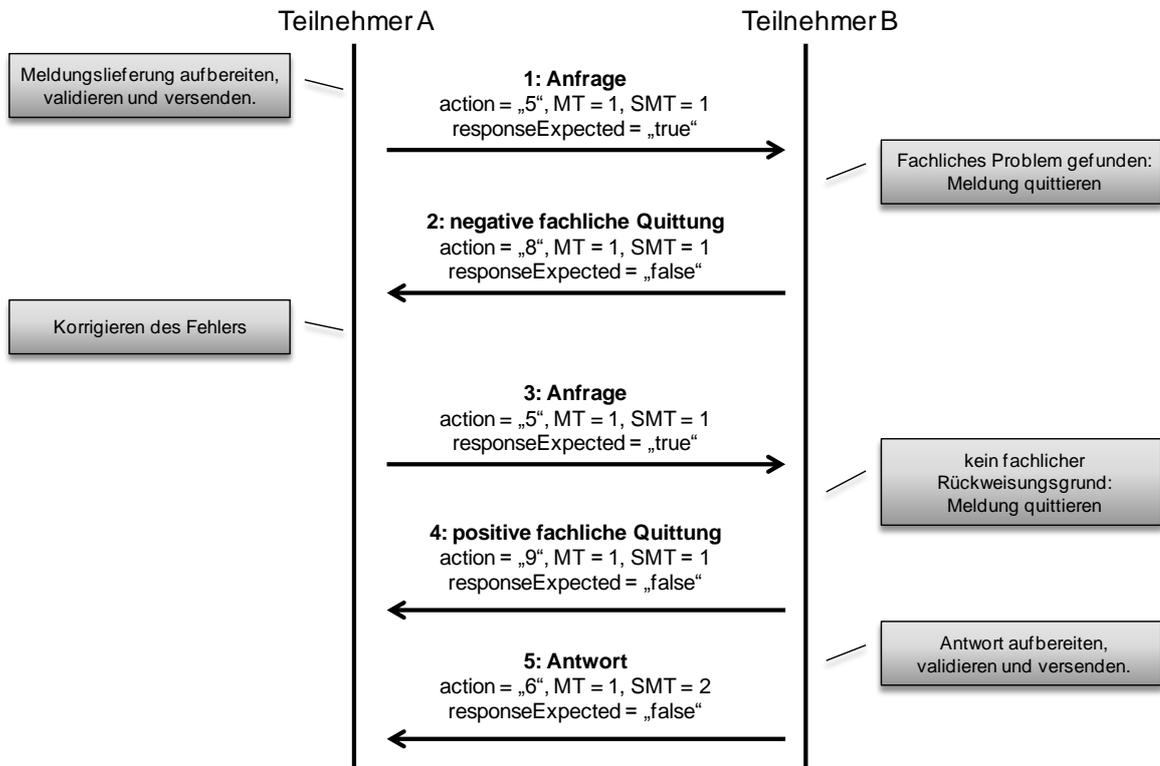


Abbildung 15: Anfrage mit negativer Quittierung.

4.3.4. Anfrage ohne Quittierung, Antwort mit negativer Quittierung

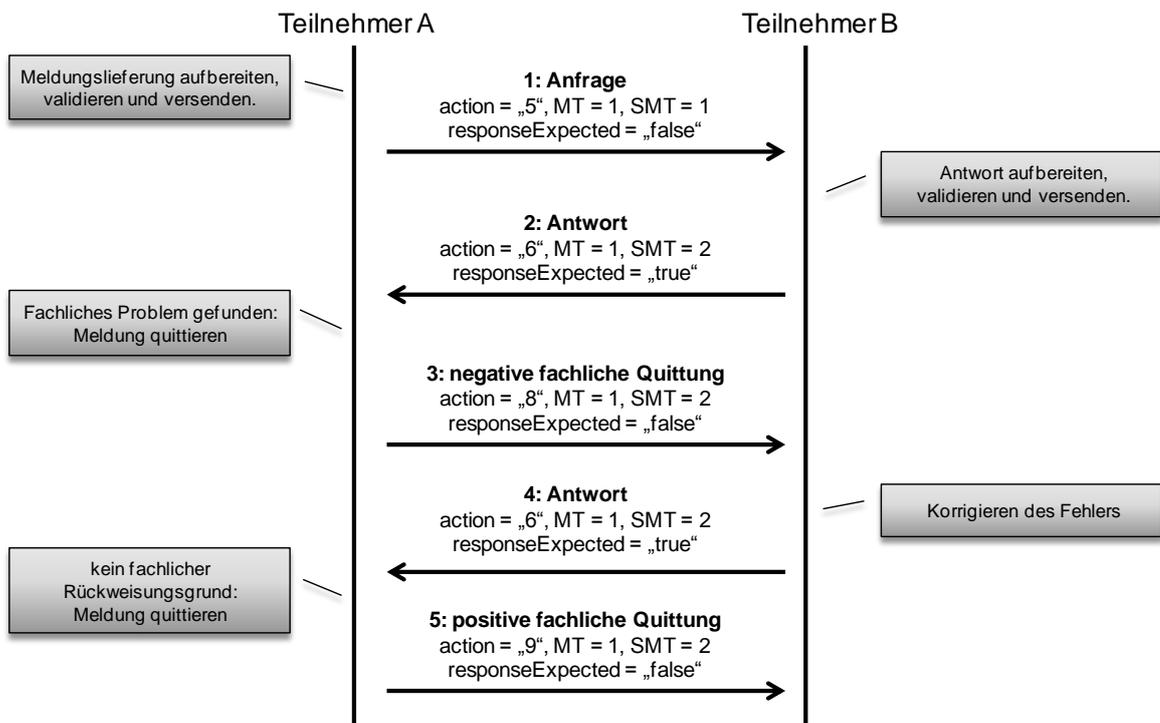


Abbildung 16: Anfrage ohne Quittierung, Antwort mit negativer Quittierung.



4.3.5. Widerruf/ Korrektur mit negativer Quittierung

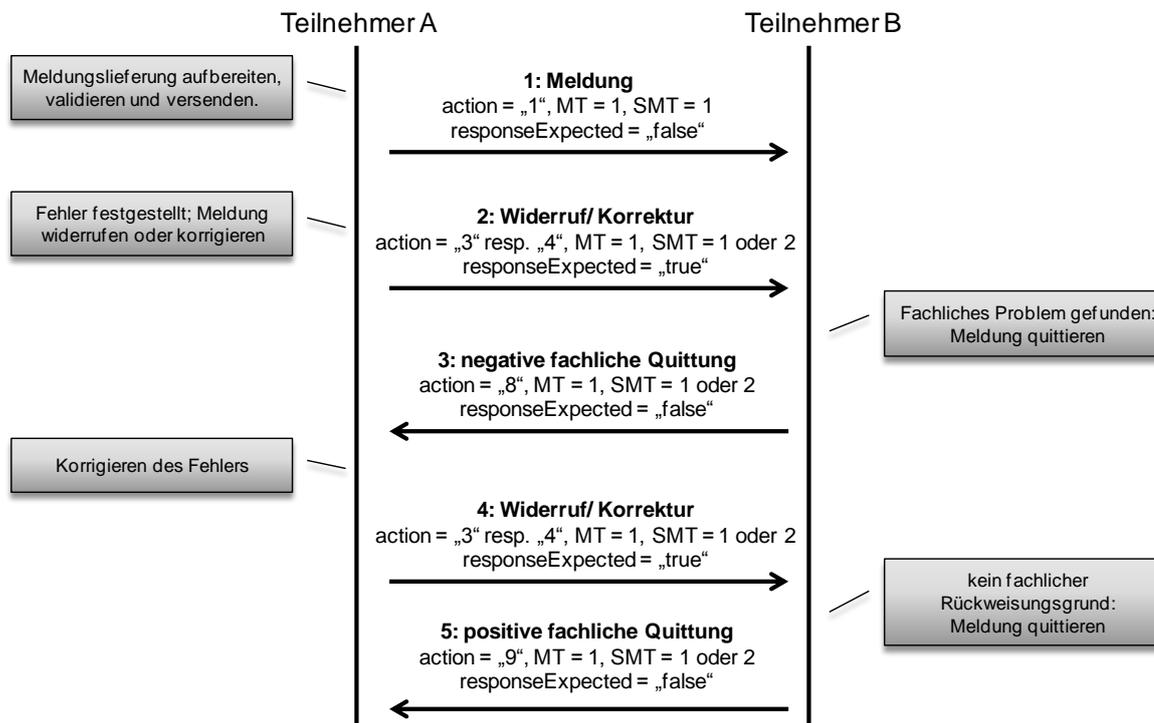


Abbildung 17: Widerruf/ Korrektur mit negativer Quittierung.

4.4. Fehlerhandling

4.4.1. Grundsatz des Fehlerhandlings

Eine integrierte Fehlerbehandlung ist äusserst aufwendig. Wann immer möglich, sollen auftretende Probleme von den Meldungsprozessen losgelöst, zwischenmenschlich per E-Mail oder Telefon abgehandelt werden.

Grundsätzlich gilt, dass der Absender dafür Sorge zu tragen hat, dass diese Fehler bereits vor dem Versand gefunden und behoben werden, so dass nur noch technisch fehlerfreie Meldungen versandt werden. Insbesondere hat der Versender sicherzustellen, dass im Versandprozess (entweder in der Fachapplikation oder im sM-Client) eine Schemaüberprüfung stattfindet.

Obwohl sedex die Integrität der übermittelten Nachrichten garantiert, ist es in Einzelfällen trotzdem denkbar, dass eine Meldung erst beim Empfänger nicht validiert oder geöffnet werden kann. Beispielsweise wenn Sender und Empfänger unterschiedliche Versionen der sM-Client Ablage verwenden, oder der Sender die Validierung schlichtweg deaktiviert hat.

4.4.2. Mögliche Fehler und deren Behandlung

In der folgenden Tabelle 7 werden mögliche Fehler und deren Behandlung aufgelistet. Dabei wird zwischen der Verwendung des sM-Clients und ohne sM-Client unterschieden. Die blauen Zeilen sollen unabhängig von der Verwendung von fachlichen Quittungen immer behandelt werden, die rot markierten Zeilen werden durch fachliche Quittungen abgedeckt.



Fehler	Automatische Behandlung beim Sender	Manuelle Behandlung durch Sender	Automatische Behandlung beim Empfänger	Manuelle Behandlung durch Empfänger
Nicht wohlgeformte Meldung oder korruptes zip (Sender stellt den Fehler fest)	sM-Client: Die Meldung wird in den Ordner „failed-to-send“ verschoben. Ohne sM-Client: Die Meldung muss validiert und darf nicht verschickt werden.	sM-Client: Der Ordner „failed-to-send“ soll vom Betreiber überwacht und entsprechende Meldungen anhand der Fehlermeldung korrigiert und nochmals verschickt werden. Ohne sM-Client: Erneuter Versand der Meldung.	--	--
Nicht erfolgreiche Übermittlung einer Meldung – Stufe sedex (z.B. Empfänger nicht erreichbar).	sM-Client: Die Meldung wird in den Ordner „failed-to-transmit“ verschoben. Ohne sM-Client: Nicht erfolgreicher Versand muss erkannt werden.	sM-Client: Der Betreiber sollte den Ordner „failed-to-transmit“ überwachen und entsprechende Meldungen gegebenenfalls korrigieren und nochmals verschicken ¹⁰ . Falls der Empfänger nicht erreichbar ist, soll dieser informiert werden. Ohne sM-Client: Neuversand und allenfalls Information des Empfängers.	--	--
korruptes zip (Empfänger stellt den Fehler fest)	sM-Client: Die Meldung wird in den sent-Ordner (weil der Empfänger den Fehler erst nach Erhalt der positiven sedex-Quittung bemerkt) und die vom Empfänger erhaltene Rückweisung (vgl. Spalte 3) in den Ordner „failed-to-send“ verschoben (bei Sammelmeldungen werden fehlerhafte Einzelmeldungen separat retourniert und liegen somit separat im „failed-to-send“). Ohne sM-Client: Retournierte Einzelmeldungen (mit messageClass=3) müssen erkannt und behandelt werden.	sM-Client: Der Ordner „failed-to-send“ soll vom Betreiber überwacht und entsprechende Meldungen anhand der Fehlermeldung korrigiert und nochmals verschickt werden. Wenn eine Meldung als invalid erkannt wird, sollte auch die Version der Ablage geprüft werden. Ohne sM-Client: Korrektur der Meldung und Prüfung der verwendeten xsd-Version.	sM-Client: Die gesamte Meldung wird zurückgewiesen. Ohne sM-Client: Retournieren des Zip-Files mit einem neuen Umschlag, in welchem die messageClass auf 3 und die referenceMessage auf die messageId gesetzt wird.	Keine manuelle Behandlung notwendig.

¹⁰ Zur Behandlung von nicht erfolgreichen Übermittlungen wird empfohlen, eine Funktionalität in der Fachapplikation zu implementieren, welche ein einfaches Neuversenden erlaubt.



Fehler	Automatische Behandlung beim Sender	Manuelle Behandlung durch Sender	Automatische Behandlung beim Empfänger	Manuelle Behandlung durch Empfänger
XML-Datei validiert nicht oder Anhänge sind nicht im PDF(/A) oder TIFF Format (Empfänger stellt den Fehler fest)	sM-Client: Die Meldung wird in den sent-Ordner (weil der Empfänger den Fehler erst nach Erhalt der positiven sedex-Quittung bemerkt) und die vom Empfänger erhaltene Rückweisung (vgl. Spalte 3) in den Ordner „failed-to-send“ verschoben (bei Sammelmeldungen werden fehlerhafte Einzelmeldungen separat retourniert und liegen somit separat im „failed-to-send“). Ohne sM-Client: Retournierte Einzelmeldungen (mit messageClass=3) müssen erkannt und behandelt werden.	sM-Client: Der Ordner „failed-to-send“ soll vom Betreiber überwacht und entsprechende Meldungen anhand der Fehlermeldung korrigiert und nochmals verschickt werden. Wenn eine Meldung als invalid erkannt wird, sollte auch die Version der Ablage geprüft werden. Ohne sM-Client: Korrektur der Meldung und Prüfung der verwendeten xsd-Version.	sM-Client: Die ganze Sammelmeldung wird in den failed-Ordner gelegt. Die gültigen Einzelmeldungen werden somit nicht weiterverarbeitet. Ohne sM-Client: Retournieren der invaliden Meldung mit einem neuen Umschlag, in welchem die messageClass auf 3 und die reference-Messageld auf die messageld gesetzt wird.	sM-Client: Überwachen des Failed-Ordners und bei vorhandenen Meldungen 1) die eigene Ablage überprüfen und 2) Kontakt mit dem Sender aufnehmen. Die gültigen Einzelmeldungen können verarbeitet werden Ohne sM-Client: Bei Detektion des Fehlers analoge Behandlung wie mit sM-Client
Sammelmeldung enthält Meldungen an verschiedene Empfänger ¹¹	sM-Client: Der Fall von verschiedenen Empfängern wird vom sM-Client verhindert, die fehlerhafte Meldung wird in den failed_to_send Ordner verschoben. (die restlichen Fehler sind als CR aufgenommen) Ohne sM-Client: Die Fehler müssen von der Fachapplikation geprüft werden.	Korrektur und Neuversand der Meldung	sM-Client: Der Fall von verschiedenen Empfängern wird vom sM-Client verhindert. Dabei wird die gesamte Sammelmeldung mit einem neuen Umschlag, in welchem die messageClass auf 3 und die reference-Messageld auf die messageld gesetzt wird, an den Absender retourniert. Für die anderen 3 Meldungen wurde ein entsprechender CR für den sM-Client aufgenommen. Ohne sM-Client: Ohne sM-Client kann obige Behandlung umgesetzt werden oder es können alternativ einzelne Meldungen zurückgewiesen werden, welche nicht den Angaben im Umschlag entsprechen. Die zurückgewiesenen Meldungen müssen mit der messageClass 3 (eCH-0090 Umschlag) zurückgeschickt werden, damit die Meldungen bei den anderen Teilnehmern im sM-Client verarbeitet werden.	Je nach Umsetzung keine Behandlung notwendig.
Sammelmeldung enthält Meldungen verschiedener Meldungs-typen ⁹				
Sammelmeldung enthält Anfragen und Antwortmeldungen (= fachliche Quittungen) ¹²				
Sammelmeldung enthält Meldungen von verschiedenen Absendern ⁹				

¹¹ Oder die Angaben im eCH-0090 Umschlag stimmen nicht mit denjenigen im eCH-0058 header Überein.

¹² Wenn die „erste“ Meldung einer Sammelmeldung keine Quittung ist, dann darf die Sammelmeldung keine Quittungen beinhalten. Wenn die „erste“ Meldung einer Sammelmeldung eine Quittung ist, dann darf die Sammelmeldungen nur Quittungen beinhalten.



Fehler	Automatische Behandlung beim Sender	Manuelle Behandlung durch Sender	Automatische Behandlung beim Empfänger	Manuelle Behandlung durch Empfänger
Fachlicher Fehler (z.B. Einzelmeldung wurde an falschen Empfänger verschickt)	Keine Behandlung im sM-Client.	a) Ohne fachliche Quittungen: -- b) Mit Verwendung von fachlichen Quittungen: Im Fall von negativen fachlichen Quittungen oder ausbleibenden positiven fachlichen Quittungen ist der Absender für den erneuten Versand verantwortlich.	Keine Behandlung	a) Ohne fachliche Quittungen: Bei fachlichen Fehlern muss der unter contactInformation angegebene Kontakt kontaktiert werden und das Problem bilateral gelöst werden. b) Mit Verwendung von fachlichen Quittungen: negative fachliche Quittung senden.

Tabelle 7: Übersicht über mögliche Fehler und deren Behandlung.



5. Vergabe messageType / subMessageType

5.1. Nummernbereiche messageType

Der messageType wird auf Stufe sedex reglementiert und die Nummernbereiche vom Domänenverantwortlichen vergeben. Mit dieser Information werden die Berechtigungen für den Versand und Empfang von Meldungen geregelt. Folgende Nummernbereiche wurden den Domänen zugeteilt:

BSV	2001 – 2999
SSK:	3001 – 3999
PV:	5200 – 5249

5.2. Zuordnung messageType zu Meldungsprozessen

Gemäss eCH-0058v4 ist festgelegt, dass nur Meldungen des gleichen Meldungstyps in einem ZIP zusammengefasst werden dürfen. (Da im sedex-Umschlag nicht mehrere Meldungstypen angegeben werden können, ist sonst keine eindeutige Beziehung zwischen Umschlag und Nutzdatenpaket gegeben.)

Da es fachlich keinen Sinn macht, Meldungen verschiedener Meldungsprozesse in einer gemeinsamen Sammelmeldung zu verschicken, ist es zwingend, dass pro Meldungsprozess ein separater Meldungstyp vergeben wird. Die einzelnen Prozessschritte eines Meldungsprozesses können mit verschiedenen Submeldungstypen identifiziert werden.

5.3. Verwendung subMessageType

Der subMessageType wird als 6-stellige Zahl aufgebaut: **XXXXXX**

Es wird empfohlen, die einzelnen Meldungen innerhalb des Meldungsprozesses durchnummerieren, beginnend mit 000001.

Die genauen Vergaben werden in den Teilnehmer- und Meldungsverzeichnissen der Projekte geregelt (siehe [9]).

5.4. Beispiel

Die Meldungsprozesse „Beschluss an AK“ und „Zwischenverfügung“ (unstrukturiert) erhalten somit folgende messageType / subMessageType-Kombinationen:

Meldung	messageType	subMessageType
Meldung 1: Vorbescheid (IV -> AK)	20xx	000001
Meldung 2: IV-Beschluss (IV -> AK)	20xx	000002
Meldung 3: Verfügung (AK -> IV)	20xx	000003
Meldung 4: Zwischenverfügung	20xy	000001



Der messageType 20xx resp. 20xy muss vom BSV im Rahmen der Meldespezifikation vergeben werden.

Der subMessageType der Meldungen 20xx wird fortlaufend beginnend von 000001 bis 000003 gewählt, da die drei Meldungen zum selben Meldungsprozess gehören. Sie bezeichnen die Meldungen im Meldungsprozess. Im Meldungsprozess 20xy existiert nur eine Meldung (000001).

Teil B – Synchroner Meldungen

Synchrone Meldungen weisen einige grundsätzliche Unterschiede zu asynchronen Meldungen auf. So werden im synchronen Fall die Nutzdaten nicht gezippt, es werden keine sedex-Umschläge nach eCH-0090 verwendet und es gibt keine Sammelmeldungen. Trotzdem ist es ein Bedürfnis, die Schemata von synchronen und asynchronen Meldungen möglichst identisch aufzubauen. Dies aus dem Grund, dass es Services gibt, welche sowohl synchron als auch asynchron angeboten werden¹³. Bei identischem Aufbau der Nutzdaten, muss dieser Teil nur einmal implementiert werden. Damit können Entwicklungskosten eingespart werden.

Hinweis: Meldungen, welche sowohl synchron als auch asynchron ausgetauscht werden, folgen dem synchronen Meldungsformat.

6. Aufbau einer Meldung

Synchrone Meldungen enthalten keinen sedex-Umschlag gemäss Abbildung 5, sondern bestehen nur aus dem sedex-Nutzdatenpaket.

Das Nutzdatenpaket jedoch entspricht vom Aufbau her der asynchronen Meldung gemäss Abbildung 6. Sammelmeldungen werden im synchronen Fall nicht ausgetauscht. Damit gelten die Inhalte aus Kap. 3.2 bis 3.5 auch für synchrone Meldungen. Im Meldungsrahmen gibt es die folgenden Unterschiede:

- Das Element <subject> kann auch entfallen. Damit ist zusätzlich zum Vorkommen „1“ auch „0“ möglich.
- Im synchronen Fall wird das Element <responseExpected> optional definiert. Wird es verschickt, wird es von der Fachapplikation (Server) ignoriert, da Aufrufe in jedem Fall beantwortet werden. Das Element wird trotzdem beibehalten, um bei einer asynchronen Umsetzung des synchronen Services das Element verwenden zu können.

7. Einbettung in die Meldungsprozesse

Der Aufbau des Headers für die verschiedenen Meldungsarten (action-Code 1, 3, 4, 5, 6, 10, 12) ist im synchronen Fall identisch zum asynchronen Fall (Kap. 4.1).

¹³ Hinweis: Wird eine Meldung sowohl synchron als auch asynchron angeboten, ist eine separate Meldungsdefinition erforderlich.



Die fachlichen Quittungen werden im synchronen Fall jedoch unterschiedlich zu den asynchronen Meldungen aufgebaut: Anstelle des eventReport werden „header“ und „info“ innerhalb eines „message“-Tags verschickt. Dies aus dem Grund, dass der Root-Tag im synchronen Fall pro Anfrage/ Antwort gleich sein muss. Im synchronen Fall wird auf eine neue Meldung (action=1) als Antwort in jedem Fall eine fachliche Quittung erfolgen (action=8 oder 9), bei Anfragen (action=5) wird entweder mit einer Antwort (action=6) oder einen negativen fachlichen Quittung (action=8) geantwortet. Damit müssen action 6, 8 und 9 denselben Root-Tag haben.

Der Aufbau von synchronen Quittungen ist deshalb wie folgt:

```
<message>
  <header></header>
  <info></info>
</message>
```

Die Inhalte von header und info entsprechen denjenigen asynchroner Meldungen.

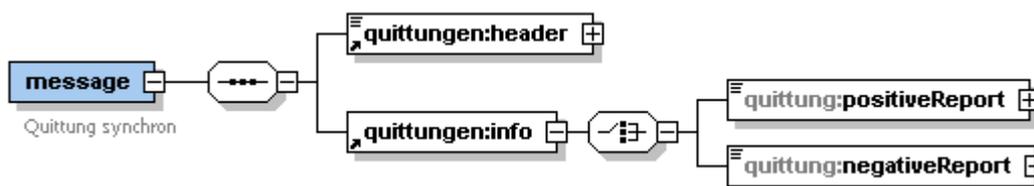


Abbildung 18: Aufbau von synchronen fachlichen Quittungen.

Die Meldungsabläufe synchroner Meldungen entsprechen grundsätzlich denjenigen asynchroner Meldungen (Kap. 4.3), jedoch gibt es weniger mögliche Fälle: So werden Antworten bei synchronen Meldungen nie quittiert, so dass das Beispiel in Kap. 4.3.4 entfällt. Beim Beispiel „Anfrage mit negativer Quittung“ (Kap. 4.3.3) entfällt im synchronen Fall Schritt 4.

Ebenfalls vereinfacht wird das Fehlerhandling: Im synchronen Fall wird bei allen Fehlerfällen eine negative Quittung zurückgeschickt, welche vom Sender entgegen genommen werden muss.

Teil C – Fachbereichsspezifische Datentypen

8. Fachbereichsspezifische Datentypen

Neben dem technischen Aufbau der Meldungen sollen auch die in verschiedenen Meldungen verwendeten Typen und Identifikatoren standardisiert werden. Hierzu gehört beispielsweise, dass der Zivilstand einer natürlichen Person in sämtlichen Meldungen (welche den Zivilstand melden), gleich definiert wird oder auch eine natürliche Person mit den gleichen Identifikatoren gemeldet wird. Wie immer soll, falls möglich, auf Standards von eCH zurückgegriffen werden.



8.1. Basistypen

8.1.1. Built-in datatypes

Für einfache Datentypen wie Zahlen, Daten, etc. werden die XML *built-in datatypes* (siehe <http://www.w3.org/TR/xmlschema-2/#built-in-datatypes>) verwendet. Entsprechende Einschränkungen z.B. von Zahlenbereichen können über sog. *Constraining facets* abgebildet werden.

8.1.2. Durch eCH standardisierte Basistypen

In diesem Kapitel wird eine Auswahl der von eCH standardisierten Typdefinitionen vorgestellt.

Es werden hier nur jene Typen dokumentiert, welche für die Umsetzung der vorliegenden Spezifikation von Bedeutung sind. Für eine komplette Dokumentation sämtlicher standardisierter Typen wird auf die entsprechenden Spezifikationen der eCH-Standards verwiesen.

Im Sinne einer behutsamen Einführung, wird zu jedem durch eCH veröffentlichten Standard (eCH-XXXX) gleichzeitig eine forgiving Version (eCH-XXXXf) ausgegeben, welche sämtliche Elemente optional enthält. Häufig wird in den domänenspezifischen Typen auf diese forgiving-Versionen referenziert.

Wann immer im Folgenden auf das Vorkommen von einzelnen Elementen eingegangen wird, wird unabhängig davon, welche Variante des Standards effektiv eingesetzt wird, auf die non-forgiving Variante Bezug genommen.

Bei allen nachfolgenden Datentypen gilt, dass optionale Elemente immer geliefert werden müssen, wenn sie vorhanden sind; ausgenommen sind projektspezifische Abweichungen von dieser Regel.



8.1.2.1. eCH-0007 - Datenstandard Gemeinden

Die vorliegende Fassung des eCH-0007 basiert auf der Version 5.

Bezeichnung	Typ
Gemeinde-identifikation	<p>eCH-0007:swissMunicipalityType</p> <p>Austausch von Identifikationen und Namen von politischen Gemeinden und gemeindefreien Gebieten. Setzt sich zusammen aus folgenden Elementen:</p> <ul style="list-style-type: none"> municipalityId – eCH-0007:municipalityIdType (base = xs:int, max. 6 Ziffern) Die BFS-Gemeindenummer wurde erstmals im Jahr 1960 vergeben und identifiziert jeden Eintrag in den rechtskräftigen Ständen des amtlichen Gemeindeverzeichnisses der Schweiz in eindeutiger Weise. municipalityName – eCH-0007:municipalityNameType (base = xs:token, max. 40 Zeichen) Der amtliche Gemeindename ist die behördenverbindliche Bezeichnung für jede politische Gemeinde der Schweiz. Der amtliche Gemeindename ist einsprachig und darf zu keiner Verwechslung mit einem anderen, zum gleichen Zeitpunkt im amtlichen Gemeindeverzeichnis verwendeten Namen einer politischen Gemeinde Anlass geben. cantonAbbreviation – eCH-0007:cantonAbbreviationType (base = xs:token, eingeschränkt auf die Kantonskürzel) Jede Gemeinde hat eine eindeutige Kantonszugehörigkeit, welche durch das in der Schweiz allgemein gebräuchliche Kantonskürzel definiert werden kann. historyMunicipalityId – eCH-0007:historyMunicipalityId (base = int) Die Historisierungsnummer dient als Ordnungsnummer im sogenannten historisierten Gemeindeverzeichnis und identifiziert alle seit dem Jahr 1960 im amtlichen Gemeindeverzeichnis enthaltenen Einträge (Aufnahme, Änderungen und Aufhebungen von Gemeinden) in eindeutiger Weise.



8.1.2.2. eCH-0008: Datenstandard Staaten und Gebiete

Die vorliegende Fassung des eCH-0008 basiert auf der Version 3.

Bezeichnung	Typ
Land	<p>eCH-0008:countryType Angaben zur Landeszugehörigkeit, komplexer Typ, der die folgenden Elemente enthält:</p> <ul style="list-style-type: none"> countryId – eCH-0008:countryIdType (base = integer) Vierstellige Nummer zwischen 1000 und 9999. Die Ländernummer wird von Bundesamt für Statistik vergeben und identifiziert den Eintrag im Verzeichnis der Staaten und Gebiete in eindeutiger Weise. (Hinweis: Dieses Element sollte immer verwendet werden, obwohl es im Standard optional definiert ist)¹⁴ countryIdISO2 – eCH-0008:countryIdISO2Type (base = token, maximal zwei Zeichen) Zweistelliger, alphanumerischer Code gemäss ISO 3166. countryNameShort – eCH-0008:countryNameShortType (base = token, maximal 50 Zeichen) Mit der Kurzform des Ländernamens wird die Bedeutung der numerischen BFS-Ländernummer in Textform dokumentiert.

8.1.2.3. eCH-0010: Datenstandard Postadresse für natürliche Personen, Firmen, Organisationen und Behörden

Die vorliegende Fassung des eCH-00010 basiert auf der Version 5.1.

Abbildung 19 zeigt die komplette Struktur des „addressInformationType“ gemäss eCH-0010.

Bezeichnung	Elemente des Typs addressInformationType
Adresszeile 1	eCH-0010:addressLine1 (base=token) Freie Zusatzzeilen für zusätzliche Adressangaben, welche in den übrigen Adressfeldern keinen Platz finden: addressLine1 sollte für personalisierte Adressangaben verwendet werden (z.B. c/o-Adresse).
Adresszeile 2	eCH-0010:addressLine2 (base=token) Freie Zusatzzeilen für zusätzliche Adressangaben, welche in den übrigen Adressfeldern keinen Platz finden: addressLine2 solle für unpersonalisierte Adressangaben verwendet werden (z.B. Zusatzangaben zur Lokalisation, z.B. "Chalet Edelweiss").
Strassenbezeichnung	eCH-0010:street (base = token) Strassenbezeichnungen in Postadressen. Es kann sich dabei auch um den Namen einer Lokalität, eines Weilers etc. handeln.
Hausnummer	eCH-0010:houseNumber (base = token) Hausnummer in Postadressen (inkl. Hausnummerzusatz).
Wohnungsnummer	eCH-0010:dwellingNumber (base=token) Nummer der adressierten Wohnung.

¹⁴ Es wird Rücksprache mit eCH genommen, die countryId im Standard als Pflichtfeld zu definieren. Das Element countryNameShort ist nicht als alleiniges Element für den Austausch von Landesinformationen geeignet, da das Land in diesem Element in beliebiger Sprache übermittelt werden kann (Schweiz, Suisse, Svizera, Svizra, Confédération Helvétique, Ελβετία, ...). → Dies wird von der eCH-Fachgruppe als CR aufgenommen.



Bezeichnung	Elemente des Typs addressInformationType
Postfach-Nummer	eCH-0010:postOfficeBoxNumber (base = unsignedInt) Nummer des adressierten Postfachs in einer Länge vom maximal 8 Zeichen.
Postfach-Text	eCH-0010:postOfficeBoxText (base = token) Postfachtext in der gewünschten Sprache. Dieses Element wird nur verwendet, wenn anstelle der Postfachnummer nur der Text "Postfach" zu übergeben ist.
Gebiet	eCH-0010:locality (base=token) Gelegentlich muss in ausländischen Adressen zusätzlich zu Ort und Land eine weitere geografische Angabe geführt werden
Ortsname	eCH-0010:town (base = token) Adressierter Ort; Bei Verwendung der Angaben gemäss Post, ist für die Schweizer Ortsbezeichnungen die Langform (27stellig) zu melden.
Schweizer Postleitzahl	eCH-0010:swissZipCode (base = unsignedInt) Von der Schweizer Post vergebene Postleitzahl in der Form, wie sie auf Briefen aufgedruckt wird. .
Zusatzziffer zu Schweizer Postleitzahlen	eCH-0010:swissZipCodeAddOn (base = string) Nur für Schweizer Postleitzahlen, hier jedoch obligatorisch: Schweizer Postleitzahlen sind nicht eindeutig. Dieselbe Postleitzahl kann für unterschiedliche Orte verwendet werden. Zusammen mit der hier abgebildeten zweistelligen Zusatzziffer wird sie jedoch eindeutig. Wenn das Ursprungssystem diese Information führt, hat sie diese weiterzuleiten, damit sie bei Bedarf vom empfangenden System genutzt werden kann.
Ordnungsziffer für Schweizer Postleitzahlen	eCH-0010:swissZipCodeId (base = int) Nur für Schweizer Postleitzahlen: Schweizer Postleitzahlen können im Laufe der Zeit ändern und für andere Zwecke verwendet werden. Die Ordnungsziffer (ORNP) ist stabil und wird in keinem Fall mehr neu vergeben. Wenn das Ursprungssystem diese Information führt, hat sie diese weiterzuleiten, damit sie bei Bedarf vom empfangenden System genutzt werden kann.
Ausländische Postleitzahl	eCH-0010:foreignZipCode (base=token) Von einer Post im Ausland vergebene Postleitzahl.
Land	eCH-0010:countryIdIso2Type (base = token) Zweistelliges, alphanumerisches ISO-Landeskürzel gemäss [ISO 3166-1] des Landes, in dem der adressierte Ort liegt. Das Land definiert die Konventionen zur Adressdarstellung. Die Landesangabe ist auch bei Schweizer Postadressen mitzugeben. Achtung: Politische Änderungen oder Umbenennungen von Ländern haben Anpassungen der ISO-Länderliste zur Folge.

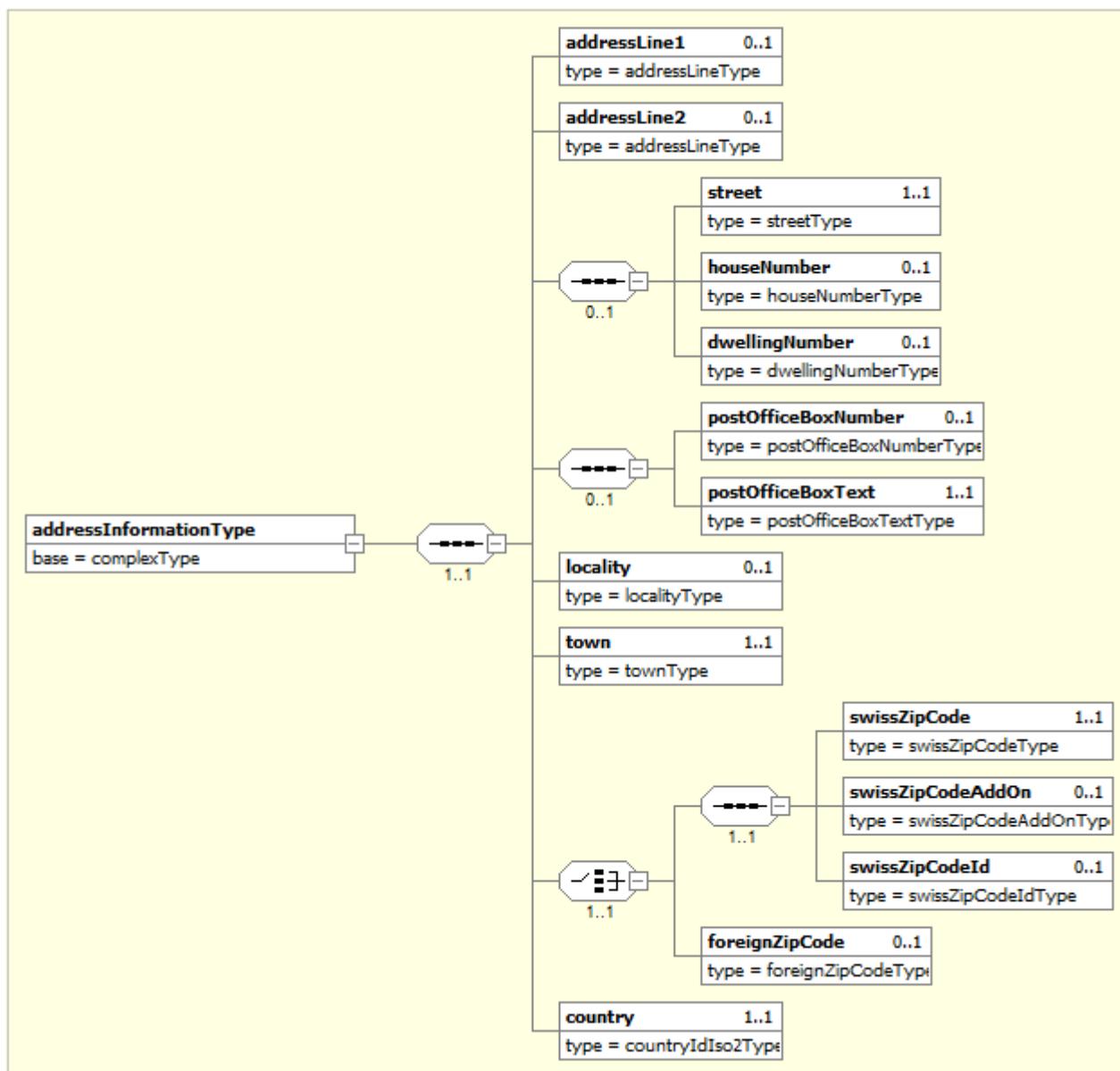


Abbildung 19: komplette Struktur des „addressInformationType“ gemäss eCH-0010.

8.1.2.4. eCH-0011: Datenstandard Personendaten

Die vorliegende Fassung des eCH-0011 basiert auf der Version 7.

Bezeichnung	Typ
Konfessionszugehörigkeit	eCH-0011:religionType (base = string) Dreistellige Zahlencodes um Religionszugehörigkeit festzulegen: 111 = evangelisch-reformierte (protestantische) Kirche 121 = römisch-katholische Kirche 122 = christkatholische (alkatholische) Kirche 211 = israelitische Gemeinschaft / jüdische Glaubensgemeinschaft 711 = Konfessionslos 811 = Zugehörig einer Religionsgemeinschaft, die weder öffentlich-rechtlich noch auf eine andere Weise vom Kanton anerkannt ist. 000 = Unbekannt



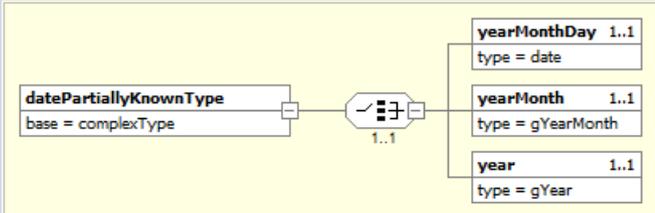
Bezeichnung	Typ
Zivilstand	<p>eCH-0011:maritalStatusType (base = string) Einstellige Zahlencodes:</p> <p>1 = ledig 2 = verheiratet 3 = verwitwet 4 = geschieden 5 = unverheiratet (wegen Ungültigkeitserklärung einer Ehe) 6 = in eingetragener Partnerschaft 7 = aufgelöste Partnerschaft</p>
Nationalität	<p>eCH-0011:nationalityType Angaben zur Nationalität, komplexer Typ, setzt sich zusammen aus folgenden Elementen:</p> <ul style="list-style-type: none"> nationalityStatus – eCH-0011:nationalityStatusType (base = string) Pflichtfeld, enthält grobe Angabe zum Informationsstand über die Nationalität: „0“ = Staatsangehörigkeit unbekannt „1“ = Staatenlos „2“ = Staatsangehörigkeit bekannt country – eCH-0008:countryType Optionales Feld, enthält genauere Angaben zur Nationalität <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> </div> <p>Der countryType ist im eCH-0008 Standard definiert.</p>

8.1.2.5. eCH-0044: Datenstandard Austausch von Personenidentifikationen

Die vorliegende Fassung des eCH-0044 basiert auf der Version 4.0.

Bezeichnung	Typ
Neue AHV-Versicherungsnummer	<p>eCH-0044:vnType (base = unsignedLong) Der Wertebereich für gültige AHV-Versichertennummern liegt zwischen 756000000001 und 756999999999 und ist auch so abgebildet worden.</p>
Benannter Personenidentifikator	<p>eCH-0044:namedPersonIdType Eine benannte PersonId ist ein Personenidentifikator, welcher von einer definierten Kommunikationsgemeinschaft (meist den Benutzerinnen und Benutzern eines bestimmten Systems bzw. von einer definierten Menge von Systemen) zur Identifikation von Personen benutzt wird.</p> <p>Er setzt sich zusammen aus den zwingend zu liefernden Elementen:</p> <ul style="list-style-type: none"> personIdCategory – eCH-0044:personIdCategoryType (base = token) Ein Kennzeichen, welches für die Kommunikationsgemeinschaft bzw. das System steht, welche die PersonIds vergibt. personId – eCH-0044:personIdType (base = token) Dem effektiven Wert, der eine bestimmte Person bezeichnet. Im Fall der alten AHV-Nummer alle Zahlen ohne Punkte. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> </div> <p><i>Hinweis:</i> Für die alte AHV-Nummer wurde ein separater Typ definiert (oldVnType, siehe Kap. 8.1.3).</p>



Einfache Bezeichnung	eCH-0044:baseNameType (base = token) Token der Länge 1-100. Wird intern im eCH-0044 verwendet. Obwohl der Typ im eCH-0044 Standard nicht dokumentiert ist, wird er auch in anderen eCH-Standards verwendet. Aus diesem Grund wird er auch hier benutzt.
Geschlecht	eCH-0044:sexType (base = string) 1 = männlich 2 = weiblich 3 = unbestimmt
Unvollständiges Datum	eCH-0044:datePartiallyKnownType Bietet exklusive Auswahl an built-in Types für unvollständige Datumsangaben an. 

8.1.2.6. eCH-0097 Schnittstellenstandard Unternehmensidentifikation

Die vorliegende Fassung des eCH-0097 basiert auf der Version 1.

Bezeichnung	Typ
UID	eCH-0097:uidStructureType (komplexer Typ mit den Elementen uidOrganisationIdCategory, uidOrganisationId) Typ für die Übermittlung der UID
Unternehmensidentifikator	eCH-0097:namedOrganisationIdType (komplexer Typ mit den Elementen organisationIdCategory, organisationId) Gemeinsamer Typ für alle Identifikatoren zur Identifikation von Unternehmen mit Ausnahme der UID (z.B. BUR-Nr.)
Name der Organisation	eCH-0097:organisationNameType (base = token) Name des Unternehmens
Rechtsform	eCH-0097:legalFormType (base = string) Die Rechtsform des Unternehmens ist der Rechtsformnomenklatur des BUR abgeleitet.

8.1.3. Weitere Basistypen

Die folgenden zwei Basistypen sind nur für das SSK common-file relevant und werden auch nur da implementiert.

Bezeichnung	Typ/Beschreibung
MWST-Nummer	vatNumberType Mehrwertsteuernummer, enthält nur Ziffern
ESTV-Dossier-Nr	dossierNumberFETType ESTV-Dossier-Nr.; max. 7-stellige Zahl



Der folgende Basistyp ist für beide Fachbereiche (AHV/IV und SSK) relevant:

Bezeichnung	Typ/Beschreibung
Alte AHV-Nummer	<pre>oldVnType <xs:restriction base="xs:unsignedLong"> <xs:pattern value="[1-9][0-9]{10}"/> </xs:restriction></pre> <p>Hinweis: Für die alte AHV-Nummer kann alternativ eCH-0044:namedPersonIdType verwendet werden. Für diejenigen Fälle, wo „nur“ die alte AHV-Nummer benötigt wird, ist der eCH-0044-Standard jedoch eher komplex. Aus diesem Grund wurde für diese Fälle ein eigener Typ definiert.</p>

8.2. Komplexe Typen

Bei der Ausarbeitung und Modellierung der komplexen Typen hat sich gezeigt, dass verschiedene Typen für das eigentlich Gleiche benötigt werden. So kann sogar der in den meisten Meldungen verwendete Typ Natürliche Personen schwer standardisiert werden, einerseits weil fachspezifische Unterschiede bestehen oder auch weil in gewissen Applikationen bestimmte Daten nicht vorhanden sind und daher auch nicht gemeldet werden können. Vielerorts wird ein „behutsames“ Vorgehen notwendig sein, indem man die Teilnehmer verpflichtet, ab einem bestimmten Datum bestimmte Elemente zwingend zu liefern. Ab diesem Datum können diese Elemente dann auch über die XML-Definition zwingend verlangt werden (Beispiel: Neue AHV-Nummer, aktuell optional).

Die nachfolgende Abbildung 20 zeigt die für die aktuellen Meldungen notwendigen Typen.

Das Diagramm suggeriert teilweise Vererbungsrelationen, welche so nicht implementiert werden. Die Implementationen der im Folgenden vorgestellten komplexen Typen bauen auf jenen der eCH Standards 0044f und 0097f auf, ohne im Allgemeinen vom xml-Inheritance Mechanismus Gebrauch zu machen. Die Subtypen der erwähnten komplexen Typen werden dann aber wieder mit Hilfe von xml-Vererbung innerhalb des common-Files implementiert.

Beispielsweise ist der „naturalPersonsTaxType“, nach dem Vorbild des im eCH-0044f definierten „personIdentificationType“ konstruiert, mit welchem er weitgehend Deckungsgleich ist. Er wird aber trotzdem nicht mit Hilfe von xml-Inheritance umgesetzt, sondern im common-File komplett neu definiert. Dasselbe gilt für den im eCH-0097f definierten „organisationIdentificationType“.

Der Grund für dieses Vorgehen ist, dass Typen nicht über Dateigrenzen hinweg vererbt werden sollen. Es wird im Folgenden explizit darauf hingewiesen werden, wann tatsächlich Vererbung eingesetzt wird.

Jede Domäne definiert ihre eigenen, spezialisierten Typen. Aus diesem Grund werden diese spezialisierten Typen im Folgenden nach Domäne getrennt aufgelistet. Bei der Definition der spezialisierten Typen, wird von Typen Gebrauch gemacht, welche in den eCH Standards 0044f und 0097f definiert und im Kapitel 8.1.2 vorgestellt worden sind.

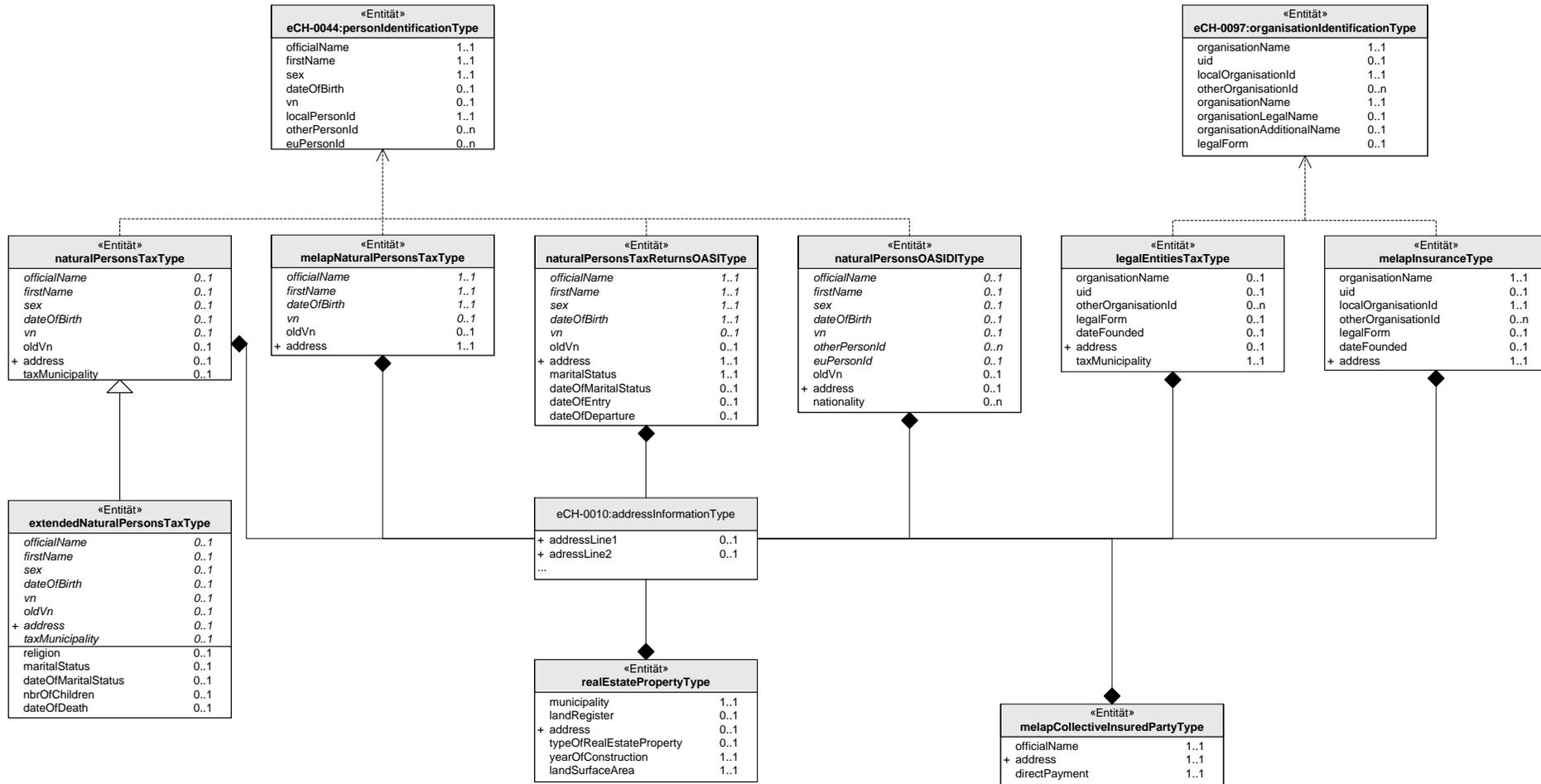


Abbildung 20: In den Fachbereichen SSK und AHV/IV definierte komplexe Typen.



8.2.1. In *ssk-common* und *eahv-iv-common* definierte Typen

Gewisse Typen werden sowohl im *ssk-/* als auch im *eahv-iv-common-file* voneinander unabhängig aber gleichwohl identisch definiert. Die so entstehende Redundanz rechtfertigt sich dadurch, dass die SSK und eAHV/IV Domänen voneinander getrennt bleiben sollen, so dass gemeinsam benutzte Files vermieden werden.

Meist handelt es sich bei den folgenden Typen um kleinere Konstrukte, welche im Rahmen von anderen Definitionen innerhalb des jeweiligen *common-Files* (teils an mehreren Stellen) wiederverwendet werden.

8.2.1.1. Natürliche Person Steuermeldungen AHV (naturalPersonsTaxReturnsOASITyp)

Der Typ Natürliche Person Steuermeldungen AHV umfasst die für die Steuermeldungen AHV notwendigen NP Elemente. Er wird im *common-File* nach dem Vorbild des im eCH-0044f definierten „*personIdentificationType*“ neu definiert.

Element	Typ	Beschreibung	Vorkommen
officialName	eCH-0044f:baseNameType	Amtlicher Name der Person	1..1
firstName	eCH-0044f:baseNameType	Alle Vornamen der Person in der richtigen Reihenfolge	1..1
sex	eCH-0044f:sexType	Geschlecht der Person	1..1
dateOfBirth	eCH-0044f:datePartiallyKnownType	Geburtsdatum der Person	1..1
vn	eCH-0044f:vnType	Versichertennummer (AHVN13)	0..1
oldVn	ssk-common:oldVnType, resp. eahv-iv-common:oldVnType	Alte AHV-Nummer (AHVN11)	0..1
address	eCH-0010f:addressInformationType	Adresse	1..1
maritalStatus	Zivilstand gemäss eCH-0011f	Zivilstand	1..1
dateOfMaritalStatus	xs:date	Zivilstandsdatum	0..1
dateOfEntry	xs:date	Eintrittsdatum	0..1
dateOfDeparture	xs:date	Austrittsdatum	0..1



8.2.2. In *ssk-common* definierte Typen

8.2.2.1. Natürliche Person Steuern (naturalPersonsTaxType)

Der Typ Natürliche Person Steuern wird im common File nach dem Vorbild des im eCH-0044f definierten „personIdentificationType“ neu definiert.

Element	Typ	Beschreibung	Vorkommen
officialName	eCH-0044f:baseNameType	Amtlicher Name der Person	0..1
firstName	eCH-0044f:baseNameType	Alle Vornamen der Person in der richtigen Reihenfolge	0..1
sex	eCH-0044f:sexType	Geschlecht der Person	0..1
dateOfBirth	eCH-0044f:datePartiallyKnownType	Geburtsdatum der Person	0..1
vn	eCH-0044f:vnType	Versichertennummer (AHVN13)	0..1
oldVn	ssk-common:oldVnType	Alte AHV-Nummer (AHVN11)	0..1
address	eCH-0010f:addressInformationType	Adresse der Person	0..1
taxMunicipality	eCH-0007f:swissMunicipalityType	Kennung der Steuergemeinde	0..1

8.2.2.2. Natürliche Person Steuern erweitert (extendedNaturalPersonsTaxType)

Erweitert den Typ „NatürlichePersonSteuern“ um zusätzliche Elemente. Er macht dabei Gebrauch vom xsd Inheritance-Mechanismus.

Element	Typ	Beschreibung	Vorkommen
officialName	Vererbt	Amtlicher Name der Person	Vererbt
firstName	Vererbt	Alle Vornamen der Person in der richtigen Reihenfolge	Vererbt
sex	Vererbt	Geschlecht	Vererbt
dateOfBirth	Vererbt	Geburtsdatum	Vererbt
vn	Vererbt	Versichertennummer (AHVN13)	Vererbt
oldVn	Vererbt	Alte AHV-Nummer (AHVN11)	Vererbt
adress	Vererbt	Adresse	Vererbt
taxMunicipality	Vererbt	Steuergemeinde	Vererbt
religion	eCH-0011f:religionType	Konfession	0..1
maritalStatus	eCH-0011f:maritalStatusType	Zivilstand	0..1
dateOfMarital-Status	xs:date	Zivilstandsdatum	0..1
nbrOfChildren	xs:nonNegativeInteger	Anzahl Kinder	0..1
dateOfDeath	xs:date	Todesdatum	0..1

8.2.2.3. Natürliche Person Steuern Melap (melapNaturalPersonsTaxType)

Der Typ Natürliche Person Steuern Melap wird vom bestehenden Melap System vorgegeben. Er wird im common File nach dem Vorbild des im eCH-0044f definierten „personIdentificationType“ neu definiert.



Element	Typ	Beschreibung	Vorkommen
officialName	eCH-0044f:baseNameType	Amtlicher Name der Person	1..1
firstName	eCH-0044f:baseNameType	Alle Vornamen der Person in der richtigen Reihenfolge	1..1
dateOfBirth	eCH-0044f:datePartiallyKnownType	Geburtsdatum der Person	1..1
vn	eCH-0044f:vnType	Versichertennummer (AHVN13)	0..1
oldVn	ssk-common:oldVnType	Alte AHV-Nummer (AHVN11)	0..1
address	eCH-0010f:addressInformationType	Adresse	1..1

8.2.2.4. Juristische Person Steuern (legalEntitiesTaxType)

Der Typ legalEntitiesTaxType wird im common File nach dem Vorbild des im eCH-0097f definierten organisationIdentificationType neu definiert.

Element	Typ	Beschreibung	Vorkommen
organisation-Name	eCH-0097:organisationNameType	Name der Organisation	1..1
uid	eCH-0097:uidStructureType	UID der Organisation	0..1
otherOrganisationId	eCH-0097:namedOrganisationIdType	Das im namedOrganisationIdType enthaltene Element „organisationIdCategory“ kann einen der 3 folgenden Werte annehmen: <ul style="list-style-type: none"> CH.BUR BUR-Nummer CH.FTANbr ESTV-S-Nummer 	0..n
legalForm	eCH-0097:legalFormType	Rechtsform, codiert	0..1
dateFounded	eCH-0044f:datePartiallyKnownType	Gründungsdatum der Organisation	0..1
address	eCH-0010:addressInformationType	Adresse	0..1
taxMunicipality	eCH-0007f:swissMunicipalityType	Steuergemeinde (aktuell BfS-Nummer, nach 2015 UID)	1..1

8.2.2.5. Versicherung Melap (melapInsuranceType)

Der Typ Versicherung Melap definiert die Versicherung analog dem Melap System. Er wird im common File nach dem Vorbild des im eCH-0097f definierten „organisationIdentificationType“ neu definiert.

Element	Typ	Beschreibung	Vorkommen
organisation-Name	eCH-0097:organisationNameType	Name der Organisation	1..1
uid	eCH-0097:uidStructureType	UID der Organisation	0..1
localOrganisationId	eCH-0097:namedOrganisationIdType	Das im namedOrganisationIdType enthaltene Element „organisationIdCategory“ muss den Wert CH.FTANbr (ESTV-S-Nummer) annehmen.	1..1



Element	Typ	Beschreibung	Vorkommen
otherOrganisationId	eCH-0097:namedOrganisationIdType	Das im namedOrganisationId-Type enthaltene Element „organisationIdCategory“ kann einen der folgenden Werte annehmen: <ul style="list-style-type: none"> CH.BUR BUR-Nummer 	0..n
dateFounded	eCH-0044f:datePartiallyKnownType	Gründungsdatum der Organisation	0..1
address	eCH-0010:addressInformationType	Adresse	1..1

8.2.2.6. Kollektivversicherungsnehmer Melap (melapCollectiveInsuredPartyType)

Der Typ Kollektivversicherungsnehmer Melap definiert die Kollektivversicherungsnehmer analog dem Melap System.

Element	Typ	Beschreibung	Vorkommen
name	string	Name	1..1
address	eCH-0010:addressInformationType	Adresse	1..1
directPayment	Short, erlaubte Werte „0“ und „1“ Nicht als Boolean ausgeführt, da sonst auch Werte „True“ und „False“ erlaubt wären.	Direktzahlung: Nein = 0 Ja = 1	1..1

8.2.2.7. Grundstück/Liegenschaft (realEstatePropertyType)

Element	Typ	Beschreibung	Vorkommen
municipality	eCH-0007f:swissMunicipalityType	Gemeinde (aktuell BfS-Nummer, nach 2015 UID)	1..1
landRegister	<name, number>	Grundbuch <Grundbuch / Sektion, Parzelle oder Nummer>	0..1
address	eCH-0010:addressInformationType	Adresse	0..1
typeOfRealEstateProperty	int, eingeschränkt auf bestimmte Werte	Art der Liegenschaft Einfamilienhaus = 0 Mehrfamilienhaus = 1 Stockwerkeigentum = 2 Gewerbe- und Industriegebäude = 3 Landwirtschaft = 4 Bauland = 5 Kulturland = 6 Waldland = 7 Weitere = 90	0..1
yearOfConstruction	gYear	Baujahr Optional, falls bekannt zwingend anzugeben	0..1
landSurfaceArea	positiveInteger	Fläche des Bodens Quadratmeter	0..1



A. Anhang

A.1. Mehrsprachigkeit

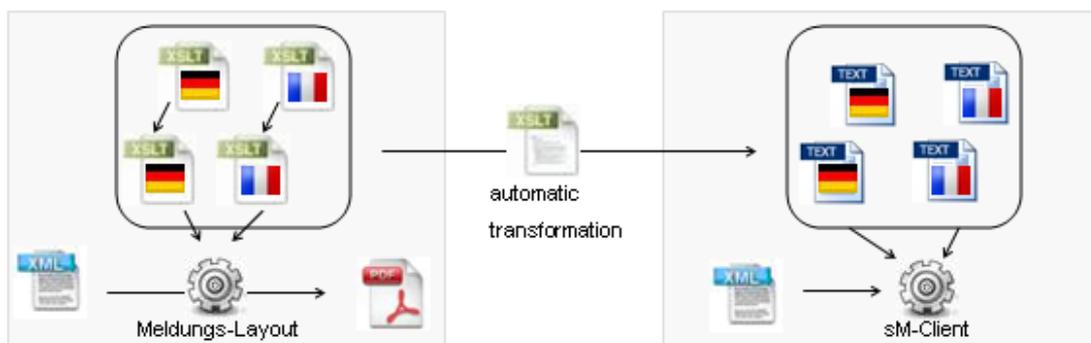
Die nachfolgenden Inhalte wurden von der Firma Information Factory im Auftrag der SSK und des BSV erarbeitet und im Rahmen der Überarbeitung des Dokuments nach Bedarf angepasst.

A.1.1. Übersicht

Die Meldungen werden zwischen den Systemen in sprachunabhängiger Form ausgetauscht; den Benutzern in sprachabhängiger Form angezeigt. Die Transformierung in die gewünschte Sprache wird an zwei Stellen umgesetzt:

- sM-Client
- PDF-Generierung einer Meldung

Dieses Mapping (Meldungselement zu Elementbezeichnung) wird für beide Bereiche einmal festgelegt. Die Generierung der PDF-Dateien wird mittels XSLT realisiert, diese Mappings müssen somit aus XSLT zugreifbar sein. Der sM-Client ist in Java implementiert und ist dadurch flexibler. Der Vorschlag ist an das Konzept der Java Property Dateien angelehnt. Java Property Dateien können in sprachabhängiger Form vorliegen. Je nach Sprache (z.B.: de, fr, it) wird die entsprechende Property Datei geladen. Die sprachabhängigen Bezeichnungen sind pro Sprache in unterschiedlichen XSLT Dateien vorliegend und können in einer Transformation in eine Property-Datei überführt werden.



A.1.2. Konzept

Ziel

Zentrale Verwaltung der sprachabhängigen Elementbezeichnungen von XML-Messages. Bezeichnungen sollen in XSL-Stylesheets und als Java-Property-Datei zur Verfügung stehen.

Ausgangslage

Verschiedene Schemas (eCH und eigene), jedes Schema definiert einen eigenen Name-Space.

Ansatz



Zentrale Quelle für Feldbezeichnung sind XSL-Stylesheets mit Variablendefinitionen (Name = Schlüssel, Inhalt = Wert). Diese können automatisiert in Java Property Dateien mit den gleichen Schlüssel-Wert Paaren konvertiert werden. Schlüssel = <namespace-prefix>'<elementname>. (Der Punkt wird verwendet, weil, Java-Properties keine Doppelpunkte im Schlüssel zulassen)

<namespaceprefix>'<elementname> = Bezeichnung in der entsprechenden Sprache

Voraussetzung

Pro Namespace darf der gleiche Elementname nicht mit verschiedenen Bezeichnungen verwendet werden.

Beispiele

In ssk-common ist der Typ "naturalPersonsTaxType" definiert, enthält u.a. die Elemente "firstName" und "officialName". Meldung 3001-000201 benutzt diesen Typ für den Message-Content "taxpayer". Das Element "taxpayer" ist also im Namespace "ssk-3001-000201" definiert. Sein Inhalt ist im Namespace "ssk-common" definiert. Es braucht also (u.a.) folgende Variablen-Definitionen:

```
<xsl:variable name="ssk-common.firstName">Vorname</xsl-variable>
<xsl:variable name="ssk-common.officalName">Name</xsl-variable>
<xsl:variable name="ssk-3001-000201.taxpayer">Steuerpflichtige Person</xsl-variable>
```

Diese können in eine Property-Datei konvertiert werden:

```
ssk-common.firstName=Vorname
ssk-common.officalName=Name
ssk-3001-000201.taxpayer=Steuerpflichtige Person
```

A.1.3. XSL-Dateien

Die sprachabhängigen Bezeichnungen sind in folgenden Dateien zu finden:

Bezeichnung definiert in einem der eCH-Standards:

- ech-xxxx-language-y-z_fr.xml
- ech-xxxx-language-y-z_de.xml
- ech-xxxx-language-y-z_it.xml

wobei xxxx für die Nummer des Standards, y für die Major-Version und z für die Minor-Version steht.

Bezeichnungen definiert für alle Meldungen des Vereins eAHV/IV:

- eahv-iv-common-58v4-language-y-z_fr.xml
- eahv-iv-common-58v4-language-y-z_de.xml
- eahv-iv-common-58v4-language-y-z_it.xml



Bezeichnungen definiert in der Meldung:

- 2001-000201-language-y-z_fr.xml
- 2001-000201-language-y-z_de.xml
- 2001-000201-language-y-z_it.xml

Dateien, welche sprachabhängige Bezeichnungen beinhalten, haben im Dateinamen das Textfragment „language“.

Für die Generierung der Meldungs-Layouts werden weitere XSL-Dateien benötigt, welche aber keine sprachabhängigen Bezeichnungen beinhalten:

- eahv-iv-common-58v4-y-z_fr.xml
- eahv-iv-common-58v4-y-z_de.xml
- eahv-iv-common-58v4-y-z_it.xml
- 2001-000201-y-z_fr.xml
- 2001-000201-y-z_de.xml
- 2001-000201-y-z_it.xml
- usw.

A.1.4. Generierung der „Property“ Dateien

Die „Property“ Dateien können mittels XSLT generiert werden.

Die Generierung mittels XSLT kann auch für andere Ziel-Formate verwendet werden z.B.: Java XML Properties.

A.2. Basislayout

Die Umwandlung der Meldungen in ein menschenlesbares Format wird mittels Layout-Anweisungen (sog. XSLT Dokument) vorgenommen. Für die Aufbereitung der Meldungen lässt sich ein gemeinsames Basisformat definieren. Abbildung 21 enthält ein Beispiel einer solchen grafischen Aufbereitung welche im Folgenden näher erläutert wird.



Steuermeldung Kapitaleistungen Säule 3b			
www.chm-steuern.ch			
Absender	ESTV	Aktion	Neue Nachricht
Empfänger	KSTV Bern	Geschäftsfall abgeschlossen	
Nachrichtendatum	12.12.2008	Originalabsender	
Kontaktinformationen	Fiskus Peter ESTV T: 0312223344 peter.fiskus@estv.admin.ch	Datum erste Nachricht	
		Unsere Geschäftsreferenz-ID	13121110987654321
		Ihre Geschäftsreferenz-ID	
		Seiten	2
		Anlagen	1

** Markierung Testlieferung **			
Kapitaleistungen 3b - Muster, Heidi			
Anspruchsberechtigte Person	Name Vorname Geburtsdatum Neue AHV-Nummer Alte AHV-Nummer Adresse	Muster Heidi 26.03.1974 756.1111.1111.11 111.11.111.000 Seeweg 6, CH-1324 Musterberg	
Ereignisdatum	01.01.2009		
Anlagen	Kantons-Dossiermeldung NP	Dateiname Scheidungsurteil.pdf	Dokumentenformat PDF
Kommentar	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.		

Abbildung 21: Beispiel eines Meldungslayouts für die Aufbereitung als PDF

A.2.1. Aufbereitung des Meldungskopfs

Im oberen Teil der ersten Seite werden Informationen aus dem Meldungskopf aufbereitet.

Der graue Balken enthält den Inhalt des Elements "subject". Der anzuzeigende Text wird in der Meldungsspezifikation definiert. Falls das Element "subject" nicht oder leer geliefert wird, soll der graue Balken grau bleiben.

Falls das Element „testDeliveryFlag“ „true“ ist, soll oberhalb des grauen Balkens mit dem „subject“ ein weiterer grauer Balken mit dem Text „** Markierung Testlieferung **“ angezeigt werden. Ist das „testDeliveryFlag“ „false“, soll der graue Balken entfallen.

Die Informationen des Meldungskopfs werden tabellarisch dargestellt, wobei die Bezeichnungen der XML-Elemente nicht für die Tabelleneinträge übernommen werden können, sondern sprachabhängig in eine lesbare Form gebracht werden müssen. Die Zuordnung dafür ist wie folgt:

Tabelleneintrag D	Tabelleneintrag F	Tabelleneintrag I	XML-Element
Absender	Expéditeur	Mittente	senderId
Empfänger	Destinataire	Ricevente	recipientId
Nachrichtendatum	Date du message	Data messaggio	messageDate



Tabelleneintrag D	Tabelleneintrag F	Tabelleneintrag I	XML-Element
Kontaktinformationen	Informations de contact	Informazioni di contatto	extension/contactInformation
Aktion	Action	Azione	action
Neue Nachricht	Message nouveau	Nuovo messaggio	action.1
Widerruf	Révocation	Revoca	action.3
Korrektur	Correction	Correzione	action.4
Anfrage	Demande	Domanda	action.5
Antwort	Réponse	Risposta	action.6
Negative Quittung	Quittance négative	Quietanza negativa	action.8
Positive Quittung	Quittance positive	Quietanza positiva	action.9
Weiterleitung	Transfert	Inoltro	action.10
Mahnung	Rappel	Richiamo	action.12
Geschäftsprozess-ID	ID-processus-métier	ID-processo	businessProcessId
Geschäftsfall abgeschlossen	Cas d'affaires terminé	Caso d'affare completato	businessCaseClosed
Seiten	Pages	Pagine	Wird bei der PDF-Generierung erzeugt (Anzahl Seiten des PDF)
Anlagen	Pièces jointes	Allegati	Wird bei der PDF-Generierung erzeugt (Anzahl Anhänge)
Titel	Titre	Titolo	title
Datei	Fichier	File	file
Dateiname	Nom de fichier	Nome del file	pathFileName
Interne Sortierung	Triage interne	Smistamento interno	internalSortOrder
Hauptdokument	Document principal	Documento principale	leadingDocument
Sortierreihenfolge	Ordre de triage	Ordine di smistamento	sortOrder
Dokumentenformat	Format document	Formato documento	documentFormat
Dokumententyp	Type document	Genere documento	documentType
Dokumentendatum	Date document	Data documento	documentDate
Markierung Testlieferung	Fanion de livraison test	Dati di TEST	testDeliveryFlag

Die folgenden vier Elemente sind Action-Code und/oder meldungsabhängig. Es muss deshalb in der Meldungsspezifikation definiert werden, ob sie im PDF dargestellt werden oder nicht.

Tabelleneintrag D	Tabelleneintrag F	Tabelleneintrag I	XML-Element
Originalabsender	Expéditeur d'origine	Mittente originale	originalSenderId
Datum erste Nachricht	Date du message initial	Data messaggio iniziale	initialMessageDate
Unsere Geschäftsreferenz-ID	Notre ID référence métier	Nostro ID riferimento	ourBusinessReferenceId
Ihre Geschäftsreferenz-ID	Votre ID référence métier	Vostro ID riferimento	yourBusinessReferenceId



Die verbleibenden Elemente aus dem Meldungskopf werden nicht aufbereitet. Der Vollständigkeit halber werden im Folgenden trotzdem die Übersetzungen aufgeführt.

Tabelleneintrag D	Tabelleneintrag F	Tabelleneintrag I	XML-Element
Nachrichtentyp	Type de message	Tipo di messaggio	messageType
Subnachrichtentyp	Type de sous-message	Tipo di submessage	subMessageType
Sendeanwendung	application émettrice	Applicazione invio	sendingApplication
Teillieferung	livraison partielle	Invio parziale	partialDelivery
Eindeutige ID Business Case	ID cas d'affaire	ID caso d'affare	uniqueIDBusinessCase
Gesamte Anzahl Pakete	Nombre total des paquets	Numero complessivo dei pacchi	totalNumberOfPackages
Anzahl vorhandene Pakete	Nombre réel des paquets	Numero dei pacchi disponibili	numberOfActualPackage
Subjekt	Sujet	Soggetto	subject
Antwort erwartet	Réponse attendue	In attesa di risposta	responseExpected
Positiver Bericht	Rapport positif	Rapporto positivo	positiveReport
Negativer Bericht	Rapport négatif	Rapporto negativo	negativeReport

A.2.2. Aufbereitung des fachlichen Inhaltes

Der zweite Teil der ersten Seite enthält fachliche Informationen. Er beginnt gegebenenfalls mit einem Hinweis auf Testlieferungen gefolgt vom „subject“ der Meldung.

Es folgen fachliche Angaben, welche variieren können. Dabei handelt es sich z.B. um das ehemalige „object“ und das ehemalige „eventDate“. Da diese Elemente aus dem Meldungskopf entfernt wurden, übernehmen, falls es der spezifizierte Prozess erfordert, Elemente aus dem Meldungsinhalt deren Funktion. Welche Elemente das genau sind, bleibt durch die Meldungsspezifikation festzulegen. (Im Beispiel: „object“ → „Anspruchsberechtigte Person“ und „eventDate“ → Ereignisdatum.)

Auf die beiden variablen Elemente folgen Informationen zu den enthaltenen Anhängen, falls solche vorhanden sind.

Der Kommentar am Schluss des zweiten Teils ist optional und wird nicht bei allen Meldungen übermittelt. Es ist deshalb ebenfalls in der Meldungsspezifikation zu definieren ob der Kommentar aufbereitet wird, oder nicht.

Tabelleneintrag D	Tabelleneintrag F	Tabelleneintrag I	XML-Element
Kommentar	Commentaire	Commento	comment
Anlagen	Pièces jointes	Allegati	attachment
Dateiname	Nom de fichier	Nome del file	pathFileName
Dokumentenformat	Format document	Formato documento	documentFormat

A.2.3. Fusszeile

In der Fusszeile der Meldungen werden hauptsächlich technische Informationen in der nachfolgend definierten Form aufbereitet:



<messageType>/<subMessageType> | Nachrichten-ID <messageId> | Referenznachrichten-ID <referenceMessageId> | Geschäftsprozess-ID <businessProcessId> | Eingangsdatum <messageDate>

Steht eine Information nicht zur Verfügung, so wird die Angabe weggelassen.

Die Zeile wird umgebrochen und die Bezeichnungen zwischen den aus der Meldung übernommenen Informationen sind mehrsprachig gehalten:

Tabelleneintrag D	Tabelleneintrag F	Tabelleneintrag I	xml-Element-Name
Nachrichten-ID	ID message	ID messaggio	messageId
Referenznachrichten-ID	ID message de référence	ID messaggio di riferimento	referenceMessageId
Geschäftsprozess-ID	ID processus métier	ID processo	businessProcessId
Eingangsdatum	Date de réception	Data d'arrivo	Vom sM-Client generiert

Auf der rechten Seite der Fusszeile werden die Seitenzahlen dargestellt:
„Seite Nr.“ / „von # Seiten“

Ein Beispiel für eine Fusszeile ist in Abbildung 22 gegeben.

2021/000102 Nachrichten-ID a1de9781-1c33-4a81-9e86-919a33bc1bcb Referenznachrichten-ID 53e0e8b2-55f8-47a1-b492-f53bfeefac4 Geschäftsprozess-ID 20110701_Dossierabtretung_Stutz_Gottfried Eingangsdatum 17.01.2013	1/1
---	-----

Abbildung 22- Beispiel einer Fusszeile

A.3. Meldungsspezifikation und Umsetzung

A.3.1. Vorgehen für Erstellung von Meldungsspezifikationen

Für die Erstellung von Meldungsspezifikationen sollen die folgenden Schritte befolgt werden:

- 1) Analysieren und Modellieren des Prozesses, welcher mittels Meldungen abgebildet werden soll. Festlegen der notwendigen Meldungen. *Hinweis:* Für die Prozessmodellierung sollen eCH-0075 und eCH-0158 befolgt werden.
- 2) Erlaubte action-Codes der Meldungen festlegen. Dabei soll das Prinzip gelten, dass ein möglichst minimales Set von action-Codes verwendet wird.
- 3) Festlegen, welche action-Codes in einer gemeinsamen Meldung (z.B. 1 und 4) und welche als separate Meldung umgesetzt werden sollen (z.B. 3). Bei Meldungen mit mehreren erlaubten action-Codes muss sichergestellt sein, dass für jeden action-Code die Inhalte der Elemente klar sind.
- 4) Strukturierungsgrad des Meldungsinhalts festlegen: Soll der Inhalt als Attachment, als strukturierter Inhalt oder als Mischform verschickt werden?
- 5) Festlegen, ob Sammelmeldungen zugelassen werden sollen.
- 6) Festlegen des Headers: Die Festlegung ist basierend auf dem vorliegenden Dokument und mit den festgelegten action-codes bis auf die in Kap. 3.3 mit || markierten Elemente eindeutig möglich.

Die Vorkommen der mit einem Asterisken ,*' markierten Elemente ergeben sich aus dem gewählten Set an erlaubten action-codes. Sie müssen so gewählt werden, dass alle der in Kap. 4.1 für die vorgesehenen action-codes definierten Vorkommen durch den spezifizierten Header gleichzeitig erfüllt werden können¹⁶.

¹⁶ **Werden zum Beispiel** die action-codes 1="Neu" und 3="Widerruf" für eine Meldung spezifiziert, so muss das ,*' -Element „referenceMessageId“ folglich das Vorkommen 0..1 erhalten um den Anforderungen beider action-codes gerecht werden zu können.



Mit || markierte Elemente müssen im Projektteam diskutiert und bei fachlich begründetem Bedürfnis zugelassen werden. Es soll das Prinzip „So wenig wie möglich; so viel wie nötig“ gelten.

- 7) Festlegen des Content: Der Content muss gemeinsam mit dem Fach festgelegt werden. Dabei sollen die Elemente so genau wie möglich beschrieben werden. Auch die xsd-Definition sollte so exakt wie möglich sein (Auflistungen wo möglich, xs:token anstatt xs:string, etc.).
- 8) Dokumentieren der Entscheidungen bezüglich Layout:
 - Welche Elemente aus dem Meldungskopf werden aufbereitet?
 - Welche fachlichen Elemente werden im unteren Teil der ersten Seite dargestellt?

A.3.2. Struktur der Meldungsspezifikation

Von Seiten der Projekte wird für jeden Meldungsprozess (z.B. Anfrage → Antwort) eine eigene Spezifikation erstellt. Hierzu wurde in Anlehnung an die eCH-Standards eine Vorlage mit folgender Struktur erstellt:

1. Zusammenfassung
2. Einleitung
3. Meldungsprozess (Prozessübersicht als BPMN und/oder als Sequenzdiagramm für alle vorkommenden Meldungsaustausche inkl. allfälliger fachlicher Quittungen)
4. Elemente im Meldungsrahmen (header)
5. Fachliche Elemente (content)
6. Layout der Meldungen
7. Sicherheitsüberlegungen
- A. Anhang (inkl. Beispielmeldungen)

A.3.3. eCH-Standards

Im Rahmen der Meldungsspezifikation und xsd-Erstellung sollten mindestens die folgenden eCH-Standards berücksichtigt werden:

- eCH-0074: Geschäftsprozesse grafisch darstellen (BPMN)
- eCH-0158: BPMN-Modellierungskonventionen für die öffentliche Verwaltung
- eCH-0018: XML Best Practices
- eCH-0033: Beschreibung von XML Namespaces
- eCH-0035: Design von XML Schemas
- eCH-0036: Dokumentation für den XML-orientierten Datenaustausch
- eCH-0050: Hilfskomponenten zur Konstruktion von XML Schemas
- eCH-0062: Design von XML Schemas – Zusammenfassung



A.4. sedex-Umschlag

A.4.1. Zusammenhang von Nutzdatenpaket und Sedex-Umschlag

sedex schreibt folgenden Meldungsaufbau vor:

Das Nutzdatenpaket („data_“) und der sedex Umschlag („envl_“) werden über den Dateinamen verknüpft; *envl_BBBB.xml* und *data_BBBB.[xxx]*, wobei der Dateityp von *data_BBBB.[xxx]* aus Sicht sedex frei gewählt werden kann. In den Fachbereichen SSK und AHV/IV hingegen wird der Dateityp auf ein zip (d.h. *[xxx] = zip*) begrenzt. Der Name der Dateien wird vom Sender vergeben. Dieser bleibt über den Transport hinweg jedoch nicht erhalten, so dass die Dateien beim Empfänger einen anderen Namen aufweisen.

A.4.2. Aufbau des sedex-Umschlags

Der Aufbau des sedex-Umschlags ist durch eCH-0090 geregelt. Wie bereits in Kap. 3.1 erwähnt, wird der sedex-Umschlag vom sM-Client beim Empfang verworfen und nur die Nutzdaten bereitgestellt. Eine ausführliche Beschreibung der einzelnen Elemente kann dem sedex-Handbuch [4] Kap. 5.6 entnommen werden. Nachfolgend werden nur die wichtigsten Punkte und die Spezifitäten für den Einsatz in den Projekten CH-Meldewesen und DA aufgeführt:

Element	Häufigkeit	Verwendung
messageId	1..1	Transport-ID ohne Bezug zu einer Einzelmeldung (vollständige Entkopplung von Transport -und fachlicher ID). Die Bedeutung dieser messageId beschränkt sich auf die Funktion als Transportinformation in eCH0090 für sedex. Die Transport-ID wird bei Verwendung vom sM-Client durch diesen beim Versand der Meldung generiert. Der sedex-Client garantiert die Eindeutigkeit der Message ID bis zur Meldungsankunft beim Empfänger. Der Adapter speichert alle mit dem sedex-Umschlag verbundenen Informationen bis zur Ankunft der Meldung. Solange diese Nachricht gespeichert ist, wird keine andere Nachricht mit derselben Message ID für die Sendung akzeptiert. Der Versender muss somit sicherstellen, dass eine messageId nicht wiederholt verwendet wird. Bei Sammelmeldungen wird eine von den Meldungen in der Sammelmeldung unabhängige messageId generiert.
messageType	1..1	Der messageType wird vom Domänenverantwortlichen der sedex Domäne für alle Meldungen innerhalb der Domäne vergeben (z.B. BSV, SSK, PV). Dabei können Meldungen gruppiert und über den subMessageType im Meldungsrahmen verfeinert definiert werden.
messageClass	1..1	Definiert innerhalb eines Meldungstyps die Bedeutung der Meldung. Das Element „messageClass“ wird durch den Aktionscode aus dem Meldungsrahmen eindeutig bestimmt. „0“ = „Normale Meldung“ (action = „1“, „3“, „4“, „5“, „6“ ¹⁷ , „8“, „9“, „10“, „12“) „1“ = „Antwort auf Meldung“ (-) „2“ = „Applikatorische Quittung“ (-) „3“ = „Fehler“ (-)

¹⁷ Gemäss [11] muss der action-Code 6 der messageClass 1 zugewiesen werden. Dies führt jedoch zu folgender Problematik: Werden Meldungen mit action-Code 1 und 6 in derselben Sammelmeldung verschickt, ist die messageClass, welche diese Sammelmeldung zugewiesen werden müsste, nicht eindeutig.



Element	Häufigkeit	Verwendung
referenceMessageld	0..1	Dieses Element wird gesetzt, wenn eine Antwort oder eine Fehlermeldung auf eine Meldung gesendet wird. Das Element enthält die messageId der ursprünglichen Meldung. Muss gesetzt werden, wenn messageClass = „1“ (Response), = „2“ (Receipt) oder = „3“ (Error) ist.
senderId	1..1	Absender entsprechend dem sedex-Adressierungskonzept.
recipientId	1..n	Empfänger (1..n) entsprechend dem sedex-Adressierungskonzept. Innerhalb der Fachbereiche SSK und AHV/IV sollen jedoch keine Meldungen gleichzeitig an mehrere Empfänger verschickt werden (primär aus Gründen der Nachvollziehbarkeit). Aus diesem Grund wurde das entsprechende Element im Header auf 1..1 eingeschränkt. Sollte trotzdem ein Meldungsversand an mehrere Teilnehmer notwendig sein, soll für jeden Empfänger eine eigene Meldung aufbereitet werden.
eventDate	1..1	Das Element wird vom sM-Client mit der aktuellen Uhrzeit gefüllt (gemäss [10], Kap. 7.4, C-4-3-9).
messageDate	1..1	Zeitpunkt der Übergabe der Meldung durch den Absender an den sedex-Adapter (Versanddatum). Das Element wird vom sM-Client mit der aktuellen Uhrzeit gefüllt (gemäss [10], Kap. 7.4, C-4-3-9).
loopback	0..1	Wird nicht verwendet.
testData	0..n	Wird nicht verwendet.

Tabelle 8: sedex Umschlag

Beim Einsatz des sM-Clients braucht sich die Fachapplikation nicht um den Umschlag zu kümmern. Die Generierung des Umschlags erledigt der sM-Client beim Versenden einer Meldung resp. eines Nutzdatenpakets. Ebenso entfernt der sM-Client den Umschlag bei einer eingehenden Meldung. Damit braucht sich die Fachapplikation nur um das Nutzdatenpaket zu kümmern.