



Technische Dokumentation zur GEKID-Spezifikation

Stand: 10. Dezember 2014

Impressum

Herausgeber:

AQUA – Institut für angewandte Qualitätsförderung und Forschung im Gesundheitswesen GmbH

Thema:

Technische Dokumentation zur GEKID-Spezifikation

Auftraggeber:

Epidemiologisches Krebsregister NRW gGmbH

Ansprechpartner:

Epidemiologisches Krebsregister Nordrhein-Westfalen gGmbH

Robert-Koch-Str. 40

48149 Münster

Dr. Volkmar Mattauch

E-Mail: volkmar.mattauch@krebsregister.nrw.de

Epidemiologisches Krebsregister Baden-Württemberg

Deutsches Krebsforschungszentrum

Im Neuenheimer Feld 581

69120 Heidelberg

Kathrin Bezold

E-Mail: k.bezold@dkfz-heidelberg.de

gültig ab:

1. Januar 2015

Hinweis:

Aus Gründen der leichteren Lesbarkeit wird im Folgenden auf eine geschlechtsspezifische Differenzierung verzichtet. Entsprechende Begriffe gelten im Sinne der Gleichbehandlung für beide Geschlechter.

Anschrift des Herausgebers:

AQUA – Institut für angewandte Qualitätsförderung und

Forschung im Gesundheitswesen GmbH

Maschmühlenweg 8-10 · 37073 Göttingen

Telefon: (+49) 0551 - 789 52 -0

Telefax: (+49) 0551 - 789 52-10

office@aqua-institut.de

www.aqua-institut.de

Inhaltsverzeichnis

Tabellenverzeichnis	4
Abbildungsverzeichnis.....	5
Erratum.....	7
1. Einleitung	8
1.1. Neueste Nachrichten	8
1.2. Projekthintergrund.....	8
1.3. Lieferumfang	9
2. Aufbau der Spezifikationsdatenbank.....	10
2.1. Allgemeine Anmerkungen zur Struktur der Spezifikationsdatenbank.....	10
2.1.1. Abfragen der Datenbank	10
2.1.2. Tabellenstruktur der Datenbank.....	11
2.2. Weiterführende Erläuterungen	12
3. Datenexport im GEKID-Verfahren	13
3.1. Gesicherte Datenübertragung.....	14
3.1.1. Übertragungswege.....	14
3.1.2. Registrierungen	14
3.1.3. Eindeutige Kennzeichnung der XML-Datenpakete.....	15
3.1.4. Identifizierung von Datensätzen	15
3.1.5. Übermittlung der Daten im Datenfluss	16
3.2. Erzeugen der Exportdatei.....	17
3.2.1. Export von Teildatensätzen	17
3.2.2. Aufbau der Exportdatei	17
3.3. Die Rückprotokollierung.....	18
3.3.1. Das Datenflussprotokoll.....	18
3.3.2. Die Fehlermeldungen.....	18
3.3.3. Prüfungsprozess und Ergebnisprotokollierung.....	20
3.4. Aufbau der XML-Exportdatei	24
3.4.1. XML-Schemata.....	24
3.4.2. Grundstruktur der XML-Dateien	27
3.4.3. Wurzelement <root>.....	28
3.4.4. Header-Bereich	29
3.4.5. Body-Bereich	37

Tabellenverzeichnis

Tabelle 1: Datenexport – Unterschiede zwischen dem QS-Verfahren und dem EKR-Verfahren	13
Tabelle 2: Codes der Krebsregister	14
Tabelle 3: Ausfüllen der Elemente eines Validation-Items in Abhängigkeit von den Fehlerarten	19
Tabelle 4: Verwendbare Schemata, Ablageort und Verwendungszweck	25
Tabelle 5: Weiche Schemavarianten für das Datenprüfprogramm	27
Tabelle 6: Symbole in den XML-Schema-Diagrammen	27
Tabelle 7: Root-Element Attribute	28
Tabelle 8: Kind-Elemente des Elements document	29
Tabelle 9: Kind-Elemente des Elements software	31
Tabelle 10: Kind-Elemente des Elements information_system	31
Tabelle 11: Angabe des betreffenden Datenflusses	31
Tabelle 12: Angabe des betreffenden Datenfluss-Ziels	32
Tabelle 13: Attribute des Elements header/provider	32
Tabelle 14: Attribute des Elements header/protocol	33
Tabelle 15: Attribute des Elements validation_provider	34
Tabelle 16: Attribute des Elements validation_item	35
Tabelle 17: Attribute des Elements status_document.....	35
Tabelle 18: Attribut des Elements status	36
Tabelle 19: Attribut des Elements error	36
Tabelle 20: Kind-Elemente des Elements error	36
Tabelle 21: Leistungserbringeridentifizierende Daten im ambulanten Bereich	39
Tabelle 22: Leistungserbringeridentifizierende Daten im stationären Bereich (Krankenhaus)	39
Tabelle 23: Attribute des Elements cases	40
Tabelle 24 Verfahrenskennung: „pseud_procedure“	40
Tabelle 25: Kind-Elemente des Elements case_admin	41
Tabelle 26: Kind-Elemente des Elements <statistic>	43
Tabelle 27: Attribut des Elements sent	44
Tabelle 28: Kind-Elemente des Elements statistic/sent.....	44
Tabelle 29: Attribute des Elements expected.....	45
Tabelle 30: Kind-Elemente des Elements statistic/processed	45

Abbildungsverzeichnis

Abbildung 1: Beispiel einer Registriernummer	15
Abbildung 2: Beispiel eines Protokolls	19
Abbildung 3 :Beziehungen zwischen <validation_item> im header und <validation_item> im body über die id .	21
Abbildung 4: Aufnahme des XSLT-Pfads in das XML-Protokoll.....	23
Abbildung 5: HTML-Darstellung nach einer XSLT-Transformation am Beispiel einer QS Übertragung.....	24
Abbildung 6: Dateiordner der Schnittstellen-Schemata	25
Abbildung 7: Weiche Schnittstellen-Schemata	26
Abbildung 8: Root-Element und Kind-Elemente header und body	28
Abbildung 9: Aufbau des Elements header	29
Abbildung 10: Aufbau des Elements document	29
Abbildung 11: Aufbau des Elements software	30
Abbildung 12:Aufbau des Elements information_system	31
Abbildung 13: Aufbau des Elements provider	32
Abbildung 14: Aufbau des Elements header/protocol	33
Abbildung 15: Aufbau und Kind-Elemente des Elements validation_provider	34
Abbildung 16: Aufbau und Kind-Elemente des Elements validation_item	34
Abbildung 17: Aufbau des Elements status_document.....	35
Abbildung 18: Aufbau und Kind-Elemente des Elements status	36
Abbildung 19: Aufbau des Elements error	36
Abbildung 20: Aufbau und Attribute des Elements encryption	37
Abbildung 21: Aufbau des Elements body	37
Abbildung 22: Aufbau des Elements body/data_container.....	38
Abbildung 23: Aufbau des Elements care_provider – ambulant.....	38
Abbildung 24: Aufbau des Elements care_provider – stationär.....	38
Abbildung 25: Aufbau des Elements cases	39
Abbildung 26: Aufbau des Elements case.....	40
Abbildung 27: Aufbau des Elements case_admin	41
Abbildung 28: Aufbau des Elements patient.....	42
Abbildung 29: Aufbau des Elements pid	42
Abbildung 30: Aufbau des Elements case_admin/protocol	43
Abbildung 31: Aufbau des Elements statistic	43
Abbildung 32: Aufbau des Elements sent	44
Abbildung 33: Aufbau des Elements expected.....	44
Abbildung 34: Aufbau des Elements processed.....	45

Abbildung 35: Integration des MDS in das Primärmodul.....	46
Abbildung 36: Diagramme „Bogen komplex“ und „Bogen einfach“	47
Abbildung 37 Ausprägungen des qs_data-Elements (Erfassungsmodule)	47

Erratum

In der Spezifikation 2015 wurde das Feld eGK-Versichertennummer (VERSICHERTENIDNEU) als Bogen- und Exportfeld in das GEKID-Modul aufgenommen.

In der Datenbank zur QS-Dokumentation wird eine Umsetzung von QS-Daten im XML über die Tabellen `ExportZielXml` und `ExportZiele` gesteuert. Das Feld `VERSICHERTENIDNEU` ist für das Modul GEKID in diesen Tabellen nicht aufgeführt, sodass es im Element `<qs_data>` enthalten sein müsste. Analog dazu wird das Element `<patient>` im Abschnitt 3.4 der vorliegenden technischen Dokumentation für das GEKID-Modul als irrelevant definiert. Dieses steht im Widerspruch zum XML-Schema, in dem dieses Exportfeld im Element `<patient>` vorgesehen ist.

Um bereits erfolgte Umsetzungen sowohl nach dem XML-Schema der Version 2015 V05 als auch nach der Datenbank zur QS-Dokumentation der Version 2015 V03 für den Export zulässig zu machen, wird für das Erfassungsjahr 2015 eine Schemaerweiterung vorgesehen. Das XML-Schema für das GEKID-Modul wird dahingehend erweitert, dass der Export über das `<patient>`-Element als auch über das Element `<qs_data>` zulässig ist.

Diese Erweiterung ist im Spezifikationspaket ab der Version 2015 V06 Bestandteil der Schemadefinition für das Modul GEKID. Der Hinweis in Abschnitt 3.4, dass das `<patient>`-Element für das Modul GEKID irrelevant ist, ist nur dann zutreffend, wenn das Exportfeld `VERSICHERTENIDNEU` über das Element `<qs_data>` exportiert wird.

1. Einleitung

1.1. Neueste Nachrichten

Zum Erfassungsjahr 2014 wurde die Spezifikation der GEKID (Gesellschaft der Epidemiologischen Krebsregister in Deutschland e.V.) als freiwilliges landesbezogenes Modul in die QS-Dokumentations-Spezifikation (ohne Implementierung in die QS-Filter-Spezifikation) integriert.

Mit der Spezifikation 2015_Basis_QSDOK_V01 wird das neue Bogenfeld „eGK-Versichertennummer“ integriert. Das Format für die zu exportierenden Datensätze wird von CSV auf XML umgestellt. Die Umstellung erfolgt analog zum Exportformat der Datensätze der QS-Dokumentation.¹ Informationen zur XML-Struktur sind Kapitel 3 zu entnehmen.

Mit der Spezifikation 2015_Basis_QSDOK_V02 entfallen die beiden Regeln 14571 und 14597.

Mit der Spezifikation 2015_Basis_RB_XML_V06 wird in der vorliegenden Technischen Dokumentation ein Erratum eingefügt, um den Export des neuen Feldes VERSICHERTENIDNEU sicherzustellen. Das XML-Schema wurde entsprechend angepasst.

1.2. Projekthintergrund

In allen Bundesländern ist zwischenzeitlich durch Landesgesetze geregelt worden, ob eine Meldepflicht bösar-tiger Neubildungen an die Epidemiologischen Krebsregister besteht. In den meisten Bundesländern wurde dies-bezüglich eine gesetzliche Meldepflicht eingeführt. Die Gesellschaft der Epidemiologischen Krebsregister in Deutschland e.V. (GEKID) hat Mitte 2008 erstmals einen Mindestdatensatz definiert, der allen Meldungen zu-grunde zu legen ist (siehe <http://www.sgg.de>).

In der vorliegenden GEKID-Spezifikation werden sämtliche Merkmale des GEKID-Mindestdatensatzes so defi-niert, dass sie Eingang in die KIS bzw. in die QS-Systeme finden können.

Die Weiterleitung der auf diesem Wege erzeugten Transaktionsdateien muss mit dem jeweiligen Landeskrebs-register vereinbart werden. Meldungen auf der Grundlage des Mindestdatensatzes der GEKID werden aber prinzipiell von allen Landeskrebsregistern entgegengenommen.

Das AQUA-Institut ist ausschließlich für die technische Umsetzung der Meldungen der GEKID-Mindestdaten-sätze verantwortlich und fungiert nicht als Datenannahmestelle. Für die inhaltliche Bearbeitung sind die Krebs-register NRW und Baden-Württemberg zuständig, an die auch Änderungsvorschläge zu adressieren sind. An-sprechpartner sind:

Epidemiologisches Krebsregister Nordrhein-Westfalen GmbH
Robert-Koch-Str. 40
48149 Münster
Dr. Volkmar Mattauch
E-Mail: volkmar.mattauch@krebsregister.nrw.de

Epidemiologisches Krebsregister Baden-Württemberg
Deutsches Krebsforschungszentrum
Im Neuenheimer Feld 581
69120 Heidelberg
Kathrin Bezold
E-Mail: k.bezold@dkfz-heidelberg.de

¹ <https://www.sgg.de/datenservice/spezifikationen-downloads/index.html>

1.3. Lieferumfang

Die vorliegende Spezifikation besteht aus folgenden Teilen²:

- Technische Dokumentation: 2015_TechDok_GEKID_V<Versionsnummer>.pdf
- Datenbank: 2015_Basis_QSDOK_V<Versionsnummer>.mdb (Modul: GEKID)³
- Ausfüllhinweis: Ausfuellhinweise_GEKID.htm
- Dokumentationsbogen: GEKID.pdf

² Die Versionsnummer der gültigen Spezifikation (z.B. V01, V02, usw.) ist dem zuletzt veröffentlichten Update zu entnehmen. (siehe Komponentenliste auf: <https://www.sqg.de/datenservice/spezifikationen-downloads/index.html>)

³ Die GEKID-Spezifikation ist als Modul in die Datenbank zur QS-Dokumentation (2015_Basis_QSDOK_V<Versionsnummer>) integriert (siehe Abschnitt 2.1).

2. Aufbau der Spezifikationsdatenbank

2.1. Allgemeine Anmerkungen zur Struktur der Spezifikationsdatenbank

Die Spezifikation ist in einer relationalen Datenbank abgelegt, die auch die Spezifikation zur QS-Dokumentation enthält. Zurzeit wird sie ausschließlich als Access-Datenbank (MS Access 2000) zur Verfügung gestellt.

Der Name der Spezifikation richtet sich nach folgendem Schema:

`<Erfassungsjahr>_Basis_QSDOK_V<Versionsnummer>.mdb`

`<Erfassungsjahr>` bezeichnet das Jahr, in dem die Dokumentation stattfindet. `<Versionsnummer>` bezeichnet die 2-stellige Versionsnummer (z.B. 02).

Beispiel:

Im Erfassungsjahr 2015 ist die Spezifikation `2015_Basis_QSDOK_V014.mdb` gültig.

Weitere Erläuterungen finden Sie in der Technischen Dokumentation „2015_TechDok_LE_XML_V<Versionsnummer>“.

Die GEKID-Spezifikation ist in die Spezifikation für QS-Dokumentation als freiwilliges landesbezogenes Modul integriert. Eine Übersicht über die in der jeweiligen Version enthaltenen Module liefert die Abfrage „Datensätze“ der Access-Datenbank (siehe Abschnitt 2.1.1).

2.1.1. Abfragen der Datenbank

Die Abfragen der Access-Datenbank geben einen vereinfachenden Überblick über die Inhalte der Spezifikation. Da die Spezifikation auch die (inhaltlichen und technischen) Vorgaben zur QS-Dokumentation beinhaltet, sind nicht alle Abfragen für die Dokumentation im Rahmen der Epidemiologischen Krebsregister (GEKID-Spezifikation) relevant.

- **Datensätze**
Diese Abfrage liefert einen Überblick über die in der Spezifikation enthaltenen Module (verpflichtende und freiwillige Module).
- **Datenfeldbeschreibung**
Hier sind alle Bogenfelder der spezifizierten Module, sortiert nach Modulname, Bogenname und Zeilennummer der Bogenfelder dargestellt.
- **DatenfeldbeschreibungFürEinModul**
Wird diese Abfrage aufgerufen, so muss der Modulname (z.B. „GEKID“, „15/1“) angegeben werden. Anschließend erhält man eine entsprechende modulbezogene Auswahl der Datenfeldbeschreibung.
- **Plausibilitätsregeln**
Diese Abfrage enthält alle Plausibilitätsregeln der spezifizierten Module, sortiert nach Modulname und Nummer der Regel.
- **PlausibilitätsregelnFürEinModul**
Wenn man diese Abfrage aufruft, so muss der Modulname (z.B. „GEKID“, „17/1“) angegeben werden und man erhält eine entsprechende modulbezogene Auswahl der Plausibilitätsregeln.
- **Teildatensätze**
Diese Abfrage liefert einen Überblick über die Teildatensätze und die Regeln für das Anlegen von Teildatensätzen.
- **Ersatzfelder**
Dies ist eine Auflistung der zu anonymisierenden Bogenfelder für alle spezifizierten Module.

⁴ Die Versionsnummer der gültigen Spezifikation (z.B. V01, V02, usw.) ist dem zuletzt veröffentlichten Update zu entnehmen.

- `OPSListen`
Diese Abfrage liefert einen Überblick über die Codes der OPS-Listen.
- `ICDListen`
Hier sind die Codes der ICD-Listen dargestellt.
- `Exportfelder`
Wenn man diese Abfrage aufruft, erhält man eine Übersicht über alle Exportfelder. Exportfelder für Listenfelder werden seit der Spezifikation 2015 nicht mehr pro Listenelement, sondern pro Listenfeld dargestellt. Die Anzahl der Elemente ist der Abfrage zu entnehmen (`Exportfelder.elemente`)
- `ExportfelderFürEinModul`
Diese Abfrage zeigt eine Auswahl der Exportfelder eines Moduls (Modulname ist explizit anzugeben). Man erhält eine Übersicht über die zu exportierenden Felder inkl. Zuordnung zum Teildatensatz. Exportfelder für Listenfelder werden seit der Spezifikation 2015 nicht mehr pro Listenelement, sondern pro Listenfeld dargestellt. Die Anzahl der Elemente ist der Abfrage zu entnehmen (`ExportfelderFürEinModul.elemente`)
- `Feldgruppen`
Diese Abfrage liefert eine Übersicht über alle Feldgruppen.
- `FeldgruppenFürEinModul`
Wenn man diese Abfrage aufruft, so muss der Modulname (z.B. „GEKID“, „HCH“) angegeben werden und man erhält eine entsprechende modulbezogene Auswahl der Feldgruppen eines Moduls.
- `WertebereicheNumerischerFelder`
Diese Abfrage liefert eine modulübergreifende Anzeige der numerischen Datenfelder (Typ `ZAHL` und `GANZEZAHL`) mit den jeweiligen Wertebereichen.
- `WertebereicheNumerischerFelderFürEinModul`
Hier werden die numerischen Datenfelder (Typ `ZAHL` und `GANZEZAHL`) und deren Wertebereiche für ein Modul angezeigt. Das Modul muss direkt angegeben werden.
- `ÜberschriftenFürEinModul`
Diese Abfrage liefert eine Anzeige der Überschriften für das angegebene Modul. Angegeben werden Start- und Ende-Felder der Überschriften, sowie die Ebene der Überschriften.
- `Schlüsselcodes`
Diese Abfrage zeigt alle Schlüssel und die zugehörigen Schlüsselwerte an.
- `Ausfüllhinweise`
Hier wird die Zuordnung von Ausfüllhinweisen (*htm.Dateien*) zu den Feldern in den einzelnen Modulen angezeigt.
- `AusfüllhinweiseFürEinModul`
Hier wird die Zuordnung von Ausfüllhinweisen (*htm.Dateien*) zu den Feldern eines Moduls angezeigt. Das Modul muss direkt angegeben werden.

2.1.2. Tabellenstruktur der Datenbank

Die Tabellen und Spalten (Attribut) unterliegen einem einheitlichen Namensschema. Erlaubte Zeichen sind die Buchstaben a-z, A-Z und die Ziffern 0-9. Umlaute und Sonderzeichen werden nicht verwendet. Das erste Zeichen eines Namens darf keine Ziffer sein.

Ein Tabellenname beginnt immer mit einem Großbuchstaben und ein Attributname immer mit einem Kleinbuchstaben. Wenn ein Name aus mehreren Teilen (z.B. Substantiven) besteht, so beginnt jeder nachfolgende Namensteil mit einem Großbuchstaben.

Beispiel:

```
BasisTyp (Tabelle)
idBasisTyp (Spalte)
```

Für jede Tabelle ist in der Spezifikation höchstens ein Primärschlüssel definiert, der nach folgendem Schema aufgebaut ist:

```
id<TabellenName>
```

Der Ausdruck in den eckigen Klammern ist ein Platzhalter für den Namen der Tabelle.

Die meisten Tabellen haben einen einfachen Primärschlüssel vom Typ `AUTOINCREMENT`. Zusätzlich enthalten derartige Tabellen mindestens ein identifizierendes Attribut⁵, welches durch Setzen eines weiteren, eindeutigen Index (bestehend aus einem oder mehreren Attributen) definiert ist.

Beispiele:

- Identifizierendes Attribut: Attribut `name` in Tabelle `BasisTyp`
- Identifizierende Attributkombination: Attribute `code` und `fkSchluessel` in Tabelle `SchluesselWert`

Es gibt auch Tabellen, deren einziger eindeutiger Schlüssel der Primärschlüssel ist. Ein Beispiel ist die Tabelle `MussKann` mit dem Primärschlüssel `idMussKann` vom Typ `TEXT(1)` (entspricht `VARCHAR(1)`). Diese Tabellen sind als einfache „Nachschlagetabellen“ zu interpretieren. Im Fall der Tabelle `MussKann` soll im entsprechenden Fremdschlüsselfeld der verknüpften Detailtabelle durch das Datenbankschema gewährleistet werden, dass nur ein ‚M‘ oder ‚K‘ eingegeben werden darf.

Die Namen von Fremdschlüsseln sind analog zum Namen der Primärschlüssel aufgebaut:

```
fk<FremdTabelleName>
```

Die Namensgebung von Primär- und Fremdschlüsseln vereinfacht den Aufbau von komplexeren Abfragen, welche sich über mehrere Tabellen erstrecken (Inklusionsverknüpfungen, Joins).

Die Fremdschlüsselattribute (Namen beginnen mit `fk`) wurden als Datenbankattribute zum Nachschlagen eingerichtet. Zum Beispiel wird beim Fremdschlüsselattribut `fkModul` in der Tabelle `Bogen` nicht mehr der Primärschlüssel des jeweiligen Moduls, sondern der Name des Moduls angezeigt. Diese Änderung betrifft nur die Anzeige, nicht jedoch die Struktur der Datenbank. Sind zwei Tabellen mehrfach durch Schlüssel-Fremdschlüssel-Beziehungen miteinander verknüpft, so kann der Name eines Fremdschlüssels auch folgendermaßen aufgebaut sein:

```
<fkFremdTabelleName><Rolle>
```

<Rolle> ist der Platzhalter für eine zusätzliche Qualifizierung der Relation.

N-M-Beziehungen werden wie üblich über Verknüpfungstabellen realisiert. In der Spezifikation haben Verknüpfungstabellen gewöhnlich keinen Primärschlüssel⁶, jedoch einen eindeutigen Schlüssel, der über die Fremdschlüsselfelder definiert ist. Ein Beispiel hierfür ist die Tabelle `RegelFelder`, welche die Tabellen `BogenFeld` und `Regel` verknüpft.

Folgende Attribute treten in vielen Tabellen auf:

- `name` ist in der Regel als technischer Name zu verstehen. Z.B. wird `Feld.name` als Variablenname in den Plausibilitätsregeln verwendet.
- `bezeichnung` ist eine kurze Beschreibung. Z.B. ist `BogenFeld.bezeichnung` der Text, welcher ein Feld auf einem Eingabeformular beschreibt.
- `bedingung` enthält einen logischen Ausdruck. Prominentester Vertreter dieses Attributtyps ist das Attribut `bedingung` in der Tabelle `Regeln`.

2.2. Weiterführende Erläuterungen

Nähere Informationen zur Spezifikationsdatenbank (wie eine Datenfeldbeschreibung) können der Technischen Dokumentation „2015_TechDok_LE_XML_V<Versionsnummer>“ entnommen werden, die auf der folgenden Webseite veröffentlicht ist: <https://www.sgg.de/datenservice/spezifikationen-downloads/index.html>.

⁵ Oder eine identifizierende Attributkombination, die einen eindeutigen Schlüssel definiert.

⁶ Hier: Primärschlüssel im Sinne der Access-Definition eines Primärschlüssels. Streng genommen wird über die beiden Fremdschlüssel ein neuer Primärschlüssel definiert.

3. Datenexport im GEKID-Verfahren

Die Erfassung und der Export der GEKID-Daten lehnen sich an die externe QS der QESÜ und QSKH-Richtlinie an. Aus Gründen der Vereinfachung werden die Krankenhäuser im Folgenden mit dem allgemeineren Begriff „Leistungserbringer“ bezeichnet.

Der Datenexport des Verfahrens der Epidemiologischen Krebsregister (EKR) unterscheidet sich von dem des QS-Verfahrens, obwohl die Spezifikationsdatenbank gleich aufgebaut ist (siehe Tabelle 1). Die QS-Software (QSD) stellt die modulübergreifende Funktionalität Datenexport bereit. Der Datenexport wird üblicherweise nicht durch die für die Dokumentation verantwortlichen ärztlichen oder pflegerischen Mitarbeiter, sondern durch einen Mitarbeiter der Krankenhaus-EDV durchgeführt. Exportiert werden ausschließlich diejenigen Datensätze, welche von den ärztlich-pflegerischen Mitarbeitern im Rahmen eines Dokumentationsabschlusses⁷ freigegeben sind. Die Vorschriften zur Anonymisierung und Pseudonymisierung der QS-Verfahren entfallen in der GEKID-Spezifikation.

Die von der QS-Software generierten Transaktionsdateien werden durch eine von der GEKID entwickelte Software weiterverarbeitet. Diese Software überträgt die Inhalte der Transaktionsdateien an die sogenannte GEKID-Schnittstelle, die von den Epidemiologischen Krebsregistern entgegengenommen wird. Die genauen Modalitäten der Übermittlung müssen mit dem jeweils zuständigen Landeskrebsregister vereinbart werden. Für die GEKID Spezifikation spielt die Übertragung von Dateien mit E-Mail zurzeit keine Rolle.

Tabelle 1: Datenexport – Unterschiede zwischen dem QS-Verfahren und dem EKR-Verfahren

Schritt	QS-Verfahren	EKR-Verfahren
Die zu exportierenden Datensätze auswählen	QSD	QSD
Exportdateien erzeugen	QSD	QSD
Transaktionsdatei erzeugen	QSD	QSD
Transaktionsdatei an Datenstelle übermitteln	QSD	GEKID-Software
Antwortdatei der Datenstelle entpacken und einlesen	QSD	QSD

Der Export und die Übermittlung der EKR-Daten finden seit dem Erfassungsjahr 2015 im XML-Format statt. In den folgenden Abschnitten wird das XML-Schema beschrieben, welches den Aufbau der XML-Datei definiert.

⁷ Eine Dokumentation kann nicht abgeschlossen werden, wenn harte Plausibilitätsregeln verletzt werden.

3.1. Gesicherte Datenübertragung

3.1.1. Übertragungswege

Eine an der Datenübertragung beteiligte Einrichtung kann auf zwei verschiedenen Übertragungswegen Daten entgegennehmen:

- **Verschlüsselung und Authentifizierung des Übertragungswegs:**

Hier werden Daten auf Übertragungswegen versendet, die eine Authentifizierung des Absenders und eine Transportverschlüsselung automatisch implizieren, wie beispielsweise D2D, KV-Connect und KV-SafeNet. Eine zusätzliche Registrierung des Absenders ist nicht nötig, da dieser so bereits eindeutig identifiziert werden kann.

- **Verschlüsselung und Authentifizierung des Datenpakets:**

Hier ist die Auswahl für den Übertragungsweg frei. In der Regel kommen Datenträger und E-Mail Übertragungen zum Einsatz. Die Transportverschlüsselung und der E-Mail-Versand sind in dieser Spezifikation geregelt. Über den vorgelagerten Prozess einer Registrierung werden Authentifizierung und Verschlüsselung ermöglicht.

3.1.2. Registrierungen

Die Registrierung eines Dokumentationssystems bei der Datenannahmestelle ist Voraussetzung für die Datenübermittlung. Die Registrierung unterstützt die sichere Übertragung von Daten per E-Mail und stellt sicher, dass Vorgangsnummern in Verbindung mit der Registriernummer eindeutig sind.

Werden Daten zusätzlich an das epidemiologische Krebsregister übermittelt, so müssen sich die Krankenhäuser beim Krebsregister registrieren lassen.

Jedes registrierte Dokumentationssystem bekommt vom zuständigen Krebsregister eine Registriernummer zugewiesen, welche folgendermaßen aufgebaut ist:

<Registriernummer> = <KodeDatenstelle>-<Registrierkode>

Der Kode der Datenstelle „Krebsregister“ setzt sich zusammen aus dem Kürzel KR und dem amtlichen Länderkode (siehe Tabelle 2). Die maximal 12-stelligen Registrierkodes werden von den Krebsregistern gepflegt.

Tabelle 2: Kodes der Krebsregister

Kode der Datenstelle	Bundesland
KR01	Schleswig-Holstein
KR02	Hamburg
KR03	Niedersachsen
KR04	Bremen
KR05	Nordrhein-Westfalen
KR06	Hessen
KR07	Rheinland-Pfalz
KR08	Baden-Württemberg
KR09	Bayern
KR10	Saarland
KR11	Berlin
KR12	Brandenburg

Kode der Datenstelle	Bundesland
KR13	Mecklenburg-Vorpommern
KR14	Sachsen
KR15	Sachsen-Anhalt
KR16	Thüringen

Beispiel:

Das epidemiologische Krebsregister NRW hat den Code *KR05*. Die in den Krankenhäusern installierten Dokumentationssysteme seien in diesem Beispiel fortlaufend durchnummeriert (DS001, DS002 etc.).⁸

Registriernummer:

Abbildung 1: Beispiel einer Registriernummer

3.1.3. Eindeutige Kennzeichnung der XML-Datenpakete

Jedes Datenpaket wird durch eine universell eindeutige ID (GUID) von der QS-Software gekennzeichnet.

Ein Globally Unique Identifier (GUID) ist eine global eindeutige Zahl mit 128 Bit, die eine Implementierung des Universally Unique Identifier Standards (UUID) darstellt.

GUIDs haben das Format XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX, wobei jedes X für ein Zeichen aus dem Hexadezimalsystem steht und damit eine Ziffer 0–9 oder ein Buchstabe A–F sein kann.

Erläuterung zur GUID:

- Die GUID wird im Exportprozess von der QS-Software einem bestimmten Datenpaket zugewiesen.
- Jeder Übermittlung an eine Datenannahmestelle ordnet das registrierte Dokumentationssystem eine eindeutige GUID (Globally Unique Identifier) zu.
- Diese GUID wird im XML-Code des Dokuments als ID gesetzt. Diese GUID muss bei dateibasierten Übertragungsverfahren in der Dateibenennung verwendet werden.

3.1.4. Identifizierung von Datensätzen

Die Vorgangsnummer (auch Datensatz-ID oder ID genannt) kennzeichnet in eindeutiger Weise jeden dokumentierten Vorgang eines registrierten Dokumentationssystems.

Im einfachsten Fall könnten die Vorgangsnummern jeweils um 1 inkrementiert werden, wenn ein neuer Datensatz angelegt wird. Wenn während eines stationären Aufenthalts z.B. zwei QS-Dokumentationen eines Leistungsfalls angelegt werden, so müssen auch unterschiedliche Vorgangsnummern vergeben werden. Die Vorgangsnummer ist daher allein ein Merkmal des QS-Dokumentationssystems, um einen Datensatz innerhalb des registrierten Dokumentationssystems eindeutig identifizieren zu können. Aus diesem Grund wäre es falsch, die Patientenidentifikationsnummer oder die Fallnummer aus dem KIS/AIS zu verwenden bzw. zu pseudonymisieren. Eine Vorgangsnummer darf keine Rückschlüsse auf Personen ermöglichen. In der Vorgangsnummer sollte z.B. nicht das Geburtsdatum enthalten sein.

Die genaue Umsetzung kann der Softwareanbieter weitgehend frei gestalten; Voraussetzung ist jedoch, dass die Eindeutigkeit der Vorgangsnummer gewährleistet ist.

⁸ Fiktives Beispiel.

Die QS-Dokumentationssoftware verwaltet jahrgangsübergreifend die Vorgangsnummern der QS-Dokumentationen. Sie soll dem Leistungserbringer eine Zuordnung der Vorgangsnummern zu krankenhausinternen Fall- oder Patientennummern (vgl. nicht übermitteltes Datenfeld `IDNRPAT`) ermöglichen. Zum Zweck der Datenvalidierung und der Qualitätsverbesserung muss es möglich sein, über die Vorgangsnummer Zugang zur Fall- bzw. Patienten-Akte bekommen zu können.

Annahme oder Ablehnung unterschiedlicher Versionen eines Datensatzes

Datensätze, die bei der entgegennehmenden Stelle eingehen, werden anhand der Kombination aus Registrierungsnummer und Vorgangsnummer als ein Vorgang identifiziert. Der für den Vorgang gespeicherte Datensatz kann durch eine neuere Version (mit höherer Versionsnummer) überschrieben werden.⁹

Unterschiedliche Versionen eines Datensatzes müssen sich auf denselben Behandlungsabschnitt beziehen. Wenn z.B. ein Patient zwei stationäre Aufenthalte in demselben Krankenhaus hatte, so sind zwei Dokumentationen mit unterschiedlichen Vorgangsnummern anzulegen.

Ebenso wenig darf eine Vorgangsnummer, welche im Dokumentationssystem bereits für eine onkologische Dokumentation verwendet wurde, für ein Modul des QS-Verfahrens wiederverwendet werden.

3.1.5. Übermittlung der Daten im Datenfluss

Die folgenden Aspekte der Datenübermittlung werden spezifiziert:

- Datenpaket, innere Struktur
- Datenpaket, äußere Struktur

Während die innere Struktur immer eingehalten werden muss, ist die äußere Struktur nur dann einzuhalten, wenn als Übertragungsweg der Versand per E-Mail oder auf einem anderen Datenträger gewählt wird.

Die äußere Struktur dient allein einem sicheren Übertragungsprozess. Dieser Übertragungsprozess ist für den Austausch von Dateien (z.B. per E-Mail) spezifiziert. Es können andere sichere Übertragungsprozesse verwendet werden, die z.B. eine geschützte Übertragung in einem Stream erlauben. In diesem Fall muss die innere Struktur des Datenpakets eingehalten werden **und** es müssen datenschutzrechtlich unbedenkliche Übertragungsverfahren gewählt werden. Eine Abweichung von der Übertragung mittels E-Mail soll im Konsens zwischen den Übertragungspartnern getroffen werden, wenn die Unbedenklichkeit der Übertragung sichergestellt ist.

Die innere Struktur jedes Datenpakets stellt alle notwendigen Metainformationen bereit, um dieses eindeutig zuzuordnen. Die Unbedenklichkeit der Übertragung muss nachgewiesen werden. Die innere Struktur wird durch ein XML-Schema (Übertragungsschema) definiert.

Zur äußeren Struktur gehören Festlegungen zu Dateibenennung, Transportverschlüsselung, Archivierung und Archivbenennung.

Ausgangsvalidierung gegen das Übertragungsschema

Als letzte Maßnahme vor der Übertragung des Dokuments muss die innere Struktur des Dokuments gegen das Übertragungsschema auf Gültigkeit geprüft werden.

Die Vorteile der Ausgangsvalidierung:

- Sicherstellung der Datenintegrität nach Verarbeitung der Daten
- frühe Feststellung von Fehlerquellen in der eigenen Datenverarbeitung
- Entlastung des nachfolgenden Datenservices von nicht validen Daten
- Vermeidung des Versands von Daten, die gegen den Datenschutz verstoßen

⁹ Ggf. ist der geänderte Datensatz mit einer neuen Versionsnummer zu übermitteln.

Durch diese Prüfung wird sichergestellt, dass die richtigen Bereiche des XML-Codes verschlüsselt sind und ausgeschlossen ist, dass kritische Daten versehentlich unverschlüsselt die nächste Stelle im Datenfluss erreichen. Sie schließt ebenfalls von vornherein aus, dass Daten an den nachfolgenden Datenservice übermittelt werden, die dieser nicht verarbeiten kann.

Das an einer Übertragungsstelle gültige Schema kann der Dokumentation über die Schemafamilie entnommen werden. Die Validierung kann über zahlreiche frei verfügbare Tools vorgenommen werden.¹⁰

3.2. Erzeugen der Exportdatei

Die Daten der zu exportierenden Dokumentationen werden vom Dokumentationssystem in Exportdateien geschrieben und die entsprechenden Vorgänge (identifiziert durch Vorgangsnummern) im Dokumentationssystem als „exportiert“ markiert.

3.2.1. Export von Teildatensätzen

Beim Export einer Dokumentation durch ein Dokumentationssystem werden die Inhalte der für den betreffenden Vorgang angelegten Teildatensätze hierarchisch in das XML des passenden Mutterbogens geschrieben und können nur gemeinsam mit dem Inhalt des Mutterbogens exportiert werden. Die Struktur der Einbettung ist durch den Datentyp des Exportmoduls im Schema definiert.

3.2.2. Aufbau der Exportdatei

Die innere Struktur der Exportpakete ergibt sich aus der Datenfeldbeschreibung des GEKID-Moduls.

Aufbauend auf dieser Beschreibung wird ein XML-Schema abgeleitet. Die Struktur der Exportdatei wird durch dieses XML-Schema festgelegt (siehe Abschnitt 3.4.1).

Das XML-Schema beschreibt und definiert die Struktur des XML-Dokuments (Exportdatei) sowie den Inhaltstyp (Datentypen der einzelnen Bögen und Felder).

Die Exportdateien sind wie folgt aufgebaut:

- XML-Format als UTF-8-kodiert
- Header-Bereich enthält die Metadaten
- Body-Bereich enthält die tatsächlichen Daten der Datenlieferung

Sonderzeichen in XML

Das Und-Zeichen (&) und die spitzen Klammern (<, >) müssen geschützt werden, falls sie an anderer Stelle benötigt werden. Dies kann durch „&“ bzw. „<“ geschehen. Die schließende spitze Klammer (>) kann durch die Zeichenkette „>“ dargestellt werden. Um Attributwerten zu erlauben, sowohl das einfache als auch das doppelte Anführungszeichen zu enthalten, kann der Apostroph (') als „'“ und das doppelte Anführungszeichen (") als „"“ dargestellt werden.

Felder der Exportdatei

Einen Überblick über die zu exportierenden Felder des GEKID-Moduls liefert die Abfrage *ExportFelderFürEinModul*.

Zusatzfelder des Datenexports

Zusatzfelder und administrative Felder im Header, die nicht in der Datenfeldbeschreibung (Tabelle *Bogenfelder*) eines Moduls enthalten sind, werden von der QS-Dokumentationssoftware ausgefüllt.¹¹

¹⁰ <http://www.w3.org/XML/Schema>

¹¹ Hier gilt also nicht der Grundsatz, dass Felder nicht vorbelegt sein dürfen.

Die Zusatzfelder sind in der Tabelle `ZusatzField` definiert. Das übertragene Speicherdatum `DokAbschlDat` (*Datum des Dokumentationsabschlusses bzw. der Freigabe des Datensatzes für den Export*) ist nicht Teil der Datenbank für Auswertungen und wird nur für organisatorische Zwecke verwendet.

Ausgangsvalidierung gegen das Übertragungsschema (siehe Abschnitt 3.4.1)

Als letzte Maßnahme vor der Weiterleitung des Dokuments muss das Dokument gegen das Übertragungsschema auf Gültigkeit geprüft werden. Durch diese Prüfung wird sichergestellt, dass die richtigen Bereiche des XML verschlüsselt sind, und sie schließt aus, dass kritische Daten versehentlich unverschlüsselt den Leistungserbringer verlassen.

Das für einen bestimmten Leistungserbringer geltende Schema kann der gesonderten Beschreibung der Schema-Familie entnommen werden.

3.3. Die Rückprotokollierung

3.3.1. Das Datenflussprotokoll

Die vorgenommenen Prüfungen werden in den dafür vorgesehenen Bereichen im XML des übermittelten Dokuments protokolliert. Das Protokoll des Dokuments wächst damit mit jeder Prüfung an.

Nachdem alle Prüfungen der datenentgegennehmenden Stelle abgeschlossen sind, wird für die Rückprotokollierung der Prüfungsergebnisse eine Kopie des Dokuments von allen EKR-Daten (Element `<qs_data>`) befreit. Das übriggebliebene XML enthält innerhalb der ursprünglichen Struktur des Dokuments die bis dahin protokollierten Prüfungen und die sich daraus ergebenden Statusmeldungen der Datensätze und des Dokuments.

Das XML-Protokoll kann von der datenentgegennehmenden Stelle oder von der QS-Software mithilfe einer Template-Definition z.B. nach HTML transformiert werden.

Um dem Leistungserbringer das konkrete Ergebnis seiner Datenlieferung mitteilen zu können, wird auch das bis zum Schluss weitergeführte Dokument von EKR-Daten befreit und als **Datenflussprotokoll** an den Leistungserbringer versandt.

Miniprotokoll

Die Erstellung eines vollständigen Datenflussprotokolls ist nur möglich, wenn die ursprüngliche XML-Datei lesbar ist und nach Entfernung der EKR-Daten schemakonform bleibt. Andernfalls ist ein reduziertes Protokoll („Miniprotokoll“) zu erstellen, das die ID (GUID) des Dokuments (sofern diese zur Verfügung stand und lesbar war, sonst wird diese nicht angegeben) und die konkrete Fehlermeldung enthält.

3.3.2. Die Fehlermeldungen

Für jeden Fehler wird wenigstens das Element `<error_message>` ausgefüllt. Andere Elemente bleiben bei einzelnen Fehlerarten leer. Tabelle 3 gibt einen Überblick darüber, unter welchen Bedingungen in den Feldern der Fehlerdatei Angaben erforderlich sind.

Die *Bogenliste* `<list>` umfasst einen oder mehrere Namen von Teildatensätzen, welche einen Bezug zu einer Regel haben. Entscheidend für den Bogenbezug sind die in der Tabelle `Regeln` formulierten Regeln, nicht die für den Exportdatensatz umformulierten Regeln.

Die *Bogenfeldliste* umfasst einen oder mehrere Namen von Bogenfeldern, welche einen Bezug zum Fehler haben. Bei der Fehlerart WERT enthält die Liste nur ein Element. Der Bogenfeldname umfasst auch den Namen des zugehörigen Teildatensatzes¹²

Für jede Regel gibt es eine Liste von Bogenfeldern, identifiziert über die Feldnamen der Regeln. Damit die Liste nicht durch Parsen ermittelt werden muss, wird sie auch über die Tabelle `RegelFelder` zur Verfügung ge-

¹² Der Bezug zum Modul kann entfallen, da dieses über die Vorgangsnummer identifiziert werden kann.

stellt. Über die Regelnummer können die Teildatensätze, welche Bezug zu einer Regel haben, durch folgende Abfrage identifiziert werden:

```
SELECT DISTINCT Bogen.name FROM (Modul INNER JOIN (Feld INNER JOIN (Bogen
INNER JOIN BogenFeld ON Bogen.idBogen = BogenFeld.fkBogen) ON Feld.idFeld
= BogenFeld.fkFeld) ON Modul.idModul = Bogen.fkModul) INNER JOIN Regel-
Felder ON BogenFeld.idBogenFeld = RegelFelder.fkBogenFeld WHERE RegelFel-
der.fkRegeln=<Regelnummer>;
```

Für die Regelnummer `<rule_id>` ist die entsprechende Nummer (Attribut `idRegeln`) der Tabelle `Regeln` anzugeben.

Bei Teildatensätzen, welche mehrfach angelegt werden können, muss die Nummer des betreffenden Teildatensatzes in eckigen Klammern angehängt werden (z.B. `P[1]`, `P[2]` usw.). Mit „Nummer des betreffenden Teildatensatzes“ ist der Inhalt des in der Tabelle `BogenFeld` unter „`fkEindeutigBogenFeld`“ definierten Feldes gemeint.

Tabelle 3: Ausfüllen der Elemente eines Validation-Items in Abhängigkeit von den Fehlerarten

Feld (csv) -> Element (xml) ->	Fehlerart <error_type>	Regelnr <rule_id>	Regeltyp <rule_type>	Liste <list>	Meldung <error_message>
	STEUER	-	-	-	ja
	EXPORT	Ja	-	<Bogen>	ja
	DOPPELT	Ja	-	-	ja
	TDS	Ja	-	<Bogenliste>	ja
	WERT	Ja	-	<Bogenfeldliste>	ja
	REGEL	Ja	ja	<Bogenliste>	ja

Beispiel:

```
<protocol>
  <status_case V="ERROR"/>
  <validation_item id="1" V="Spezifikation">
    <status V="ERROR">
      <error>
        <error_type V="Regel"/>
        <rule_id V="14641"/>
        <list V="AUFNEDATUM"/>
        <error_message V="Das Aufnahmedatum liegt nach dem aktuellen Datum"/>
      </error>
      <error>
        <error_type V="Regel"/>
        <rule_id V="14422"/>
        <list V="BUNDESLAND"/>
        <error_message V="'Bundesland' ist nicht 'Niedersachsen' [3], obwohl 'Patient(in) hat gemäß §3 (2) GEKI in die Meldung eingewilligt' ausgefüllt ist"/>
      </error>
      <error>
        <error_type V="Wert"/>
        <rule_id V="1001019"/>
        <list V="PLZ"/>
        <error_message V="Der Wert '370811' des Datenfeldes PLZ überschreitet die zulässige Feldlänge 5."/>
      </error>
    </status>
  </validation_item>
</protocol>
```

Abbildung 2: Beispiel eines Protokolls

Um die Fehleranalyse zu vereinfachen, wurden die potenziellen Fehler in Kategorien (XML-Element `<error_type>`) unterteilt, die bestimmte Prüfprozesse (XML-Element `<validation_item>`) durchlaufen.

Standardisierung der Meldungen bei Bestätigungsstatus mit Fehlerart DOPPELT.

Es wurde bereits ein Datensatz mit derselben Registrier- und Vorgangsnummer und derselben oder einer höheren Versionsnummer übermittelt.

Standardisierung der Meldungen bei der Fehlerart WERT.

Bei feldbezogenen Fehlern sind die standardisierten Fehlermeldungen der Plausibilitätsregeln zu verwenden.

Standardisierung der Meldungen bei Bestätigungsstatus mit Fehlerart TDS.

- Wenn ein obligatorischer Teildatensatz (Attribut `Bogen.fkBogenZahl` + oder 1 ist, oder ein zu einem Kindteildatensatz zugehöriger Mutterteildatensatz) eines Vorgangs in den Exportdateien einer Transaktion nicht vorkommt, ist folgende Fehlermeldung auszugeben:

Erforderlicher Teildatensatz <Bogen.name> ("<Bogen.bezeichnung>*") existiert nicht.*

- Wenn die Existenzbedingung eines Kindteildatensatzes im zugehörigen Mutterteildatensatz erfüllt ist, aber kein Kindteildatensatz vorhanden ist, so ist folgende Fehlermeldung auszugeben:

Die Angaben im Datensatz erfordern einen Teildatensatz <Bogen.name> ("<Bogen.bezeichnung>*"). Dieser fehlt.*

- Wenn die Existenzbedingung eines Kindteildatensatzes im zugehörigen Mutterteildatensatz nicht erfüllt ist, aber trotzdem ein Kindteildatensatz existiert, ist folgende Fehlermeldung zu erzeugen:

Die Angaben im Datensatz lassen keinen weiteren Teildatensatz <Bogen.name> ("<Bogen.bezeichnung>*") zu, obwohl ein solcher übermittelt wurde.*

Dieser Fehler kann wie folgt hervorgerufen werden: es wird ein TDS `GEKID:TUBASIS[1]` erwartet, aber zwei TDS `GEKID:TUBASIS[1]` werden geliefert (gleicher Eintrag in `fkeindeutigBogenFeld`).

Die Fehlermeldungen, die nicht von der Tabelle Regel in der Access-Datenbank abgedeckt sind, sind in der Tabelle Fehlermeldungen hinterlegt. Diese sollen eine Standardisierung von Fehlermeldungen und klaren Bedeutungen unterstützen.

3.3.3. Prüfungsprozess und Ergebnisprotokollierung

Ausgangspunkt ist eine prinzipiell offene Anzahl von Prüfungen. Welche Prüfungen konkret durchgeführt werden, ist abhängig vom Datenfluss. Für die Protokollierung der Prüfungen und deren Ergebnisse gibt es auf Dokumentenebene im Header und auf Fallebene im `<case_admin>` das Element `<protocol>`.

Auf Dokumentenebene sind **alle** Prüfungen zu dokumentieren, einschließlich der Prüfungen, die ausschließlich die Datensätze betreffen. Eine prüfende Einrichtung trägt sich als `<validation_provider>` in die entsprechende Auflistung ein und dokumentiert dann ihre durchgeführten Prüfungen in der Auflistung `<validation_item>` (Ausnahme: Prüft der Leistungserbringer, sind in keinem Fall die Daten des Leistungserbringers einzutragen. In diesem Fall wird der Softwareanbieter als `<validation_provider>` eingetragen).

Prüfungen, die – wie z.B. die Schemakonformität – das Dokument insgesamt betreffen, sind ausschließlich im Headerbereich einzutragen.

Prüfungen, die – wie zum Beispiel die Prüfung auf Plausibilitätsregeln – auf Fallebene erfolgen, müssen folgendermaßen protokolliert werden:

- Das Ergebnis in Bezug auf das gesamte Dokument muss im `<header>` eingetragen werden.
- Das Ergebnis der Fallprüfung muss in `<case_admin>` eingetragen werden, sofern der Status dieser Prüfung nicht OK ist (siehe auch unten).

- Alle Ergebnisse einer Prüfung, die auf Fallebene erfolgt, müssen mit einer gemeinsamen, dokumentweit eindeutigen ID im Attribut ID des Elements `<validation_item>` eingetragen werden. Dadurch ist es möglich, über die ID eines Prüfungsergebnisses, die man auf Fallebene findet, auf Dokumentenebene den `<validation_provider>` eindeutig zu identifizieren.

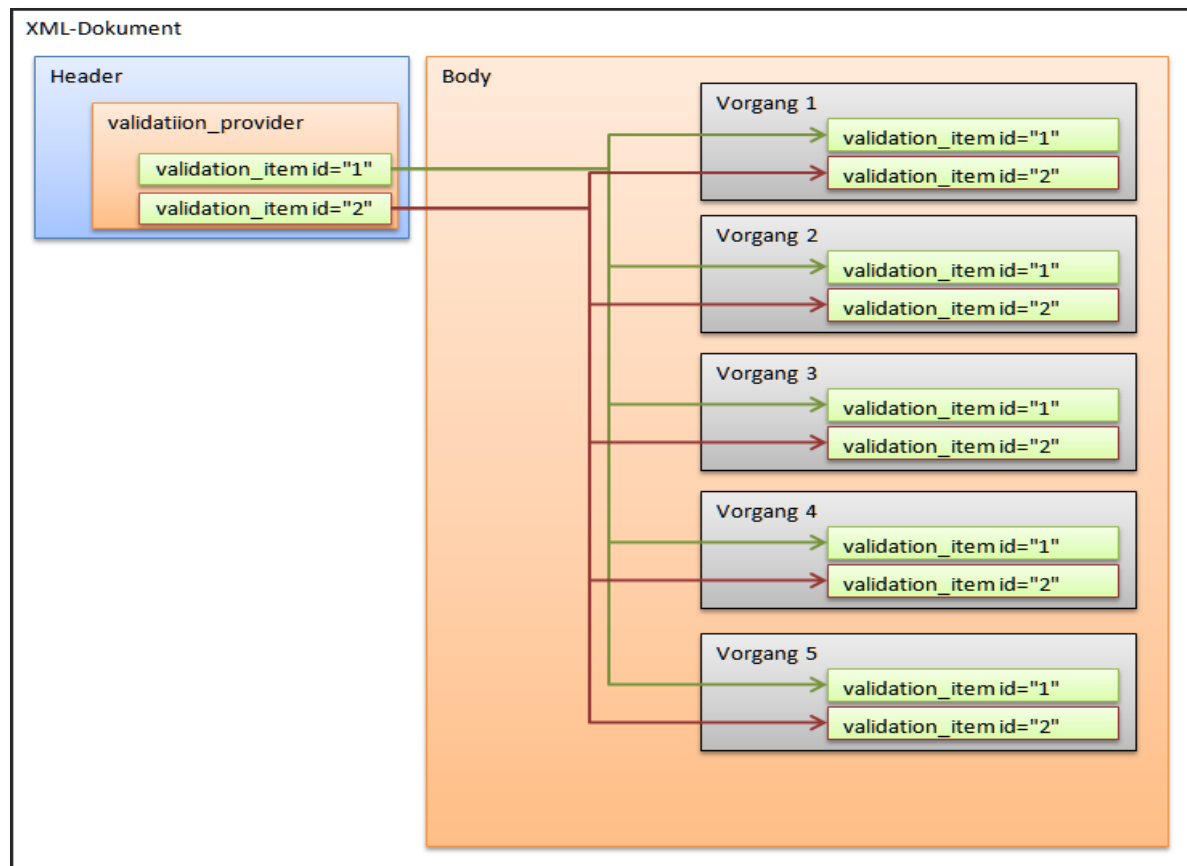


Abbildung 3 :Beziehungen zwischen `<validation_item>` im header und `<validation_item>` im body über die id

Zur Veranschaulichung dieser Konstruktion soll im Folgenden eine Analogie aus dem relationalen DB-Modell bemüht werden. So kann die Dokumentenebene als Master-Tabelle und die Fallebene als Detail-Tabelle bezeichnet werden. Letztere enthält die zum Master gehörenden Detail-Datensätze, auf die über das Attribut „ID“ referenziert werden kann (siehe Abbildung 3).

Prüfungsergebnisse

Prinzipiell wird als Ergebnis jeder Prüfung eine der folgenden Aussagen über das geprüfte Objekt getroffen:

- Keine Auffälligkeiten
- Auffälligkeiten, die einer Weiterverarbeitung nicht im Weg stehen
- Auffälligkeiten bzw. Fehler, die eine Weiterverarbeitung des Objekts ausschließen.

In der datentechnischen Übersetzung wird dieses durch

- OK
- WARNING
- ERROR

ausgedrückt, die das Ergebnis der Prüfung im Attribut „V“ des Elements `<status>` im Element `<validation_item>` zusammenfassen.

Darüber hinaus gibt es die Möglichkeit, eine beliebige Anzahl von `<error>` Elementen mit einer `<error_message>` im `<status>` Element unterzubringen.

Im Fall einer Auffälligkeit muss wenigstens eine standardisierte Fehlermeldung im <status> Element der von der Prüfung betroffenen Ebene (Vorgang oder Dokument) untergebracht werden.

Beziehungen Vorgangsebene/Dokumentenebene

Es gibt zwei Kategorien geprüfter Objekte, die zueinander in einer hierarchischen Beziehung stehen:

- Erste Hierarchieebene: das gesamte Dokument
- Zweite Hierarchieebene: der Fall

Jedes dieser Objekte hat einen Status in Bezug auf die Weiterverarbeitung, der sich auf die Ergebnisse der durchgeführten Prüfungen bezieht.

Auf Dokumentenebene ist dieser Status im Unterelement <status_document> von <protocol> im Attribut „V“ abgelegt.

Auf Fallebene ist dieser Status ebenfalls in einem Attribut „V“ eines Unterelements von <protocol> abgelegt, welches hier aber <status_case> benannt wird.

In Bezug auf die Weiterverarbeitung gibt es folgende Regeln:

Ein „ERROR“ in einer der Prüfungen verhindert die Weiterverarbeitung des geprüften Objekts. Eine oder mehrere Auffälligkeiten („WARNING“, „ERROR“) auf Fallebene bedeuten ein „WARNING“ in dem korrespondierenden Eintrag auf Dokumentenebene. Wenn bei einer fallbezogenen Validierung in **allen** Fällen auf Status ERROR erkannt wird, muss auch für das Dokument abweichend von der Regel unter 2. der korrespondierende Eintrag auf Dokumentenebene auf ERROR gesetzt werden.

Der Status (<status_case>/<status_document>) eines Objekts kann nicht „besser“ sein als sein schlechtestes Prüfergebnis.

Szenarien

Aus diesen Regeln abgeleitet, soll der Status jedes geprüften Objekts nach jeder Prüfung entsprechend dem Prüfergebnis aktualisiert werden. Daraus ergibt sich folgender Aktualisierungs- und Protokollierungsplan:

Vor der Prüfung und Protokollierung:

- (1) → Feststellen der höchsten ID in Bezug auf vorhandene <validation_item>-Elemente.
- (2) → Festgestellte ID um 1 inkrementieren und als ID der anstehenden Prüfung festlegen.

Protokollierung der fallbezogenen Prüfung:

Nachdem die fallbezogene Prüfung erfolgt ist, ist dies auf der Fallebene und der Dokumentenebene folgendermaßen zu protokollieren:

Protokollierung auf Fallebene:

Positive Prüfungen werden auf Fallebene nicht protokolliert.

Wenn eine Prüfung auf Fallebene keine Auffälligkeit feststellt, wird dieses Ergebnis nicht dokumentiert. Das Ergebnis OK ist implizit anzunehmen, wenn kein Fehler protokolliert wurde.

Auf Fallebene wird nur dann protokolliert, wenn bei der Prüfung eine Auffälligkeit festgestellt wurde. Falls eine Auffälligkeit festgestellt wird, sind die Schritte 3-6 abzuarbeiten.

- (3) → <validation_item> der Liste hinzufügen, dabei die unter 2. ermittelte ID verwenden.
- (4) → <status_case> des Falls auslesen
- (5) → Ergebnis der Prüfung mit dem Status des Falls vergleichen. In den Fällen, bei denen das Ergebnis der Prüfung schlechter ist als der aktuelle Status des Falls, wird der Status mit dem Ergebnis der Prüfung aktualisiert.
- (6) → Falls ein Ergebnis der Prüfung schlechter ist als „OK“, muss dieses als dokumentbezogenes Ergebnis „WARNING“ vermerkt werden.

Protokollierung auf Dokumentenebene:

- (7) →<validation_item> mit dem unter 4. ermittelten Prüfungsergebnis der fallbasierten Prüfung unterhalb des Elements <validation_provider> eintragen. Falls <validation_provider> für die eigene Einrichtung noch nicht besteht, muss er angelegt werden.
- (8) →<status_document> auslesen.
- (9) →Das unter 6. ermittelte Gesamtergebnis der Prüfung muss mit dem Status des Dokuments verglichen werden. Falls das Ergebnis der Prüfung schlechter ist als der aktuelle Status des Dokuments, muss dessen Status mit dem Ergebnis der Prüfung aktualisiert werden.

Protokollierung der dokumentenbezogenen Prüfung:

Nachdem die dokumentenbezogene Prüfung erfolgt ist, ist dies auf der Dokumentenebene folgendermaßen zu protokollieren:

- (1) →<validation_item> mit dem Prüfungsergebnis unterhalb des Elements <validation_provider> eintragen. Falls <validation_provider> für die eigene Einrichtung noch nicht besteht, muss er angelegt werden.
- (2) →<status_document> auslesen.
- (3) →Das Ergebnis der Prüfung mit dem Status des Dokuments vergleichen und in dem Fall, in dem das Ergebnis der Prüfung schlechter ist als der aktuelle Status des Dokuments, dessen Status mit dem Ergebnis der Prüfung aktualisieren.

Rückprotokoll – Bereitstellung eines XSLT für die Transformation

Für alle Leistungserbringer/Krankenkassen, die keine Möglichkeit haben, das Datenflussprotokoll in die QS-Dokumentationssoftware zu importieren und in geeigneter Form darzustellen, stellt das AQUA-Institut ein XSLT-Skript zur Verfügung, das die Darstellung der XML-Protokolle in Browsern in vereinfachter HTML-Darstellung ermöglicht.

Das Skript kann unter <http://www.sgg.de/downloads/protocol.xsl> heruntergeladen werden.

Lokale Transformation (Empfehlung)

Die einfachste und sicherste Variante ist das Transformieren vom Browser selbst. Dafür soll das XML-Protokoll im Browser (z.B. Internet-Explorer, Firefox) geöffnet werden.

Die Voraussetzung für die fehlerfreie Umwandlung ist,

- das lokale Ablegen des dazugehörigen XSLT-Skripts
- der entsprechende Link zum XSLT-Skript muss in das XML-Protokoll unmittelbar nach der ersten Zeile eingetragen werden:

```

1 | <?xml version="1.0" encoding="UTF-8"?>
2 | <?xml-stylesheet type="text/xsl" href="protocol.xsl"?>
3 | <root xmlns="urn:gba:sgg" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4 |     container_version="1.0" content_version="1.0">
5 |   <header>
6 |     <document>
7 |       <id V="{12345678-1234-1234-1234-123456789012}"/>
8 |       <set_id V="{12345678-1234-1234-1234-123456789012}"/>
9 |       <origination_dttm V="2012-05-17T09:30:47Z"/>
10 |      <modification_dttm V="2012-09-11T15:11:31"/>

```

Abbildung 4: Aufnahme des XSLT-Pfads in das XML-Protokoll

Der Eintrag der Referenz (siehe Pfeil) im Rückprotokoll erfolgt durch die DAS.

In der HTML-Darstellung wird der Inhalt des Protokolls besser lesbar und durch die Kategorisierung der Prüfergebnisse unter Verwendung einer Ampelanzeige

- Rot für fehlerhafte Datensätze (ERRORS)
- Gelb für Datensätze mit Hinweisen (WARNINGS)
- Grün für fehlerfreie Datensätze (OK)

übersichtlicher gestaltet.

Dokument-Status: ERROR			
			» Details ein/aus
Bundesauswertestelle Aqua-Institut GmbH (BU10000) Maschmühlenweg 8–10, 37073 Göttingen		tel: (+49)0551/789 52-296 fax: (+49)0551/789 52-295 xml-ba@aquainstitut.de	
Validierung 776	2012-10-11T13:17:57	Spezifikation	ERROR
Unbekannte Funktion Unbekannter Provider		tel: - fax: -	
Validierung 774	2012-10-11T13:17:57	Spezifikation	OK
Bundesauswertestelle name (1234567) address		tel: phone fax: fax email@domain.tld	
Validierung 1	2012-10-11T11:43:09	Spezifikation	WARNING
Validierung 3	2012-10-11T13:18:30	Spezifikation	ERROR

Abbildung 5: HTML-Darstellung nach einer XSLT-Transformation am Beispiel einer QS Übertragung

Ausschließlich für die Erfassungssoftware im Krankenhaus gilt, dass diese in der Lage sein muss, die Fehlermeldungen und Warnungen der datenentgegennehmenden Stelle einzulesen und den Anwender durch eine möglichst komfortable Nachbearbeitungsfunktion für die betreffenden Datensätze zu unterstützen.

3.4. Aufbau der XML-Exportdatei

Die folgende Beschreibung der XML-Exportdatei lehnt sich an die QS-Verfahren der QSKH und QESÜ-RL des G-BA an. Für GEKID gilt abweichend:

- Das Element Patient ist nicht relevant
- Das Elementes header/document/data_flow darf nur QS_Laenderbezogen beinhalten.
- Die Registriernummer wird für im Rahmen der GEKID-Spezifikation nicht vom AQUA-Institut sondern von den Epidemiologischen Krebsregistern vergeben.

DAS/VST und BAS sind für den GEKID Datenfluss nicht relevant. Das Übertragungsschema orientiert sich an der Schnittstelle interface_LE. Der beschriebene Aufbau enthält die Beschreibung der XML-Datei entsprechend dieser Schnittstelle.

3.4.1. XML-Schemata

Das XML-Schema hat die Aufgabe, das aktuelle Datenflussmodell der G-BA-Richtlinie abzubilden:

- Abbildung der zu exportierenden QS-Daten
- Abbildung der Schnittstellen an den Institutionen-Übergängen (z.B. DAS-VST)
- Abbildung der vorzunehmenden Datentransformation in den jeweiligen Einrichtungen des Datenflusses (z.B. LE-Pseudonymisierung bei der DAS)
- Abbildung der Rückprotokollierung

Aus diesem Grund gibt es bei der Erstellung von Schemata, welche die Konformität von Richtlinie und Datenschutz sicherstellen sollen, kein „Allround-Schema“, welches alle Anforderungen an alle Beteiligten abdeckt, sondern eine „Schema-Familie“, aus der heraus gezielt für jede Schnittstelle („Interface“) eine passende Datenstruktur definiert wird.

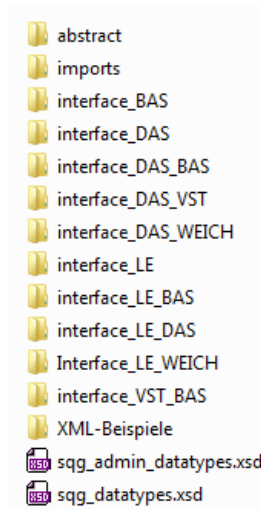


Abbildung 6: Dateiodner der Schnittstellen-Schemata

Kompositionsmodell

Um diese Schema-Familie besser warten zu können und gleiche Teilstrukturen nur einmal definieren zu müssen, wurde bei der Schema-Erstellung auf ein Kompositionsmodell zurückgegriffen, in dem sich alle Teilschemata am Ende einen Namensraum teilen. Als Bezeichnung des Namensraums wurde „urn:gba:sqg“ gewählt. Zu diesem Namensraum werden die Bausteine je nach Bedarf über „includes“ zusammengestellt.

Das Kompositionsmodell macht es möglich, Konzepte aus der objektorientierten Programmierung – darunter fallen die Konzepte abstrakter Typ, Ersetzbarkeit von Typen, Wiederverwendung und Polymorphismus – zu nutzen. Dadurch können Schemata erstellt oder genutzt werden, die generische Grundtypen definieren und diese Typen so erweitern, dass sie schnittstellenspezifisch sind, ohne das ursprüngliche Schema zu beeinflussen.

Verwendbare Schemata, Ablageort und Verwendungszweck

In der folgenden Tabelle werden Schema-Dateien aufgeführt, die im Rahmen der Übermittlung der QS-Daten Verwendung finden. Andere Dateien haben zwar ebenfalls die Dateierdung .xsd, sind aber keine vollständigen Schemata, sondern Bausteine für Schnittstellen.

Tabelle 4: Verwendbare Schemata, Ablageort und Verwendungszweck¹³

Schnittstelle/Ablageort	Schema	Verwendungszweck
LE interface_LE	2015_dv_1.0_Export.xsd	Schemavalidierung der direkten Module
	2015_iv_1.0_Export.xsd	Schemavalidierung der indirekten Module
	2015_pid_1.0_Export.xsd	Schemavalidierung der PID-Module
LE/DAS interface_LE_DAS	inter- face_LE_DAS_indirekt.xsd	Schemavalidierung der indirekten Module vor der Entschlüsselung der QS-Daten
	interface_LE_DAS_pid.xsd	Schemavalidierung der PID-Module vor der Entschlüsselung der QS-Daten
	inter- face_LE_DAS_precheck.xsd	Einlesen der Metadaten für die Steuerung der Datenannahme
	response_DAS_LE.xsd	Datenflussprotokoll der DAS
	response_receipt.xsd	Empfangsbestätigung

¹³ Die Zielgruppe ist aus der Schnittstelle zu entnehmen.

Schnittstelle/Ablageort	Schema	Verwendungszweck
DAS interface_DAS	interface_DAS_indirekt.xsd	Schemavalidierung der indirekten Module nach der Entschlüsselung der QS-Daten
	interface_DAS_pid.xsd	Schemavalidierung der indirekten Module nach der Entschlüsselung der QS-Daten
	interface_DAS.xsd	Schemavalidierung aller Module (indirekte und PID-Module) nach der Entschlüsselung der QS-Daten
	interface_DAS_psn.xsd	Schemavalidierung aller Module nach der Pseudonymisierung der LE-Daten
DAS/VST interface_DAS_VST	interface_DAS_VST.xsd	Datenübermittlung von der DAS an die VST
	interfa- ce_DAS_VST_precheck.xsd	Einlesen der Metadaten für die Steuerung der Datenannahme
	response_VST_DAS.xsd	Datenflussprotokoll von der VST an die DAS
	response_receipt.xsd	Empfangsbestätigung
DAS/BAS interface_DAS_BAS	interface_DAS_BAS.xsd	Datenübermittlung von der DAS an die BAS (Indirekte Module)
	response_BAS_DAS.xsd	Datenflussprotokoll der BAS
VST/BAS interface_VST_BAS	interface_VST_BAS.xsd	Datenübermittlung von der VST an die BAS
	response_BAS_VST.xsd	Datenflussprotokoll von der BAS an die VST
	response_receipt.xsd	Empfangsbestätigung
BAS	interface_BAS.xsd	Datenstruktur der Datenübermittlung nach der Entschlüsselung der QS- und LE-Daten

Um nach einer Schemavalidierung der XML-Dateien die Weiterverarbeitung und dementsprechend die spezifikationskonforme Protokollierung auf Datensatzebene weiterhin zu ermöglichen, wurden neben der oben in der beschriebenen Schemavariante ein „weiches“ Schema für die Schnittstellen LE und DAS eingeführt. Diese weiche Variante wird ausschließlich mit dem Datenprüfprogramm verwendet.

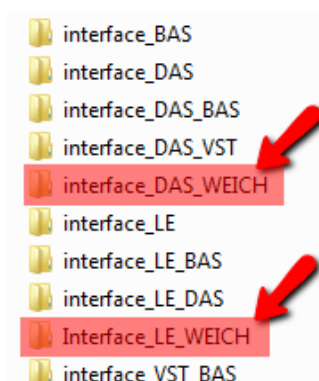


Abbildung 7: Weiche Schnittstellen-Schemata

In der folgenden Tabelle werden die Schema-Dateien aufgeführt, die im Rahmen der Prüfung mit dem Datenprüfprogramm Verwendung finden.

Tabelle 5: Weiche Schemavarianten für das Datenprüfprogramm¹⁴

Schnittstelle/Ablageort	Schema	Verwendungszweck
LE interface_LE_WEICH	2015_dv_1.0_Export.xsd	Validierung der direkten Module mit dem Datenprüfprogramm (DPP)
	2015_iv_1.0_Export.xsd	Validierung der indirekten Module mit dem DPP
	2015_pid_1.0_Export.xsd	Validierung der PID- Module mit dem DPP
DAS interface_DAS_WEICH	interface_DAS_indirekt.xsd	Schemavalidierung der indirekten Module nach der Entschlüsselung der QS-Daten
	interface_DAS_pid.xsd	Validierung der indirekten Module mit dem DPP (nach der Entschlüsselung der QS-Daten)
	interface_DAS.xsd	Validierung aller Module mit dem DPP (indirekte und PID-Module) nach der Entschlüsselung der QS-Daten
	interface_DAS_psn.xsd	Validierung aller Module mit dem DPP (nach der Pseudonymisierung der LE-Daten)

3.4.2. Grundstruktur der XML-Dateien


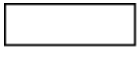
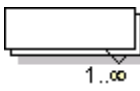


Grundsätzlich beginnt jede XML-Exportdatei mit einer Headerzeile

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



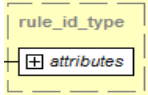
gefolgt vom Wurzelement **<root>**, das den gesamten Inhalt einschließt. Als Zeichensatz wird „UTF-8“ (Unicode-Codierung) verwendet.

Zur Veranschaulichung der verwendeten XML-Schemata werden Diagramme verwendet, deren Symbole in der folgenden Tabelle kurz dargestellt und erläutert werden.

Tabelle 6: Symbole in den XML-Schema-Diagrammen

Symbol	Beschreibung
	Optionales Element Kardinalität 0..1 („0 oder 1“)
	Obligatorisches Element Kardinalität 1: das Element muss genau einmal vorkommen
	Mehrfach wiederholbares Element Kardinalität: die erlaubte Anzahl der Elemente wird unter dem Symbol dargestellt (Beispiel: 1..n, n..m).
	Referenzelement Das referenzierte globale Element ist an anderer Stelle im Schema definiert.
	Eine Folge von Elementen Die Elemente müssen genau in der Reihenfolge vorkommen, in der sie im Schemadiagramm angezeigt sind.

¹⁴ Die Zielgruppe ist aus der Schnittstelle zu entnehmen.

Symbol	Beschreibung
	Eine Auswahl von Elementen Nur ein einziges Element aus der Liste kann ausgewählt werden.
	Ein Element mit Kind-Elementen
	Komplexer Datentyp Der komplexe Datentyp wird mit einem Rahmen mit einem gelben Hintergrund angezeigt.

3.4.3. Wurzelement <root>

Das Root-Element ist eine Art Umschlag oder Wurzelement für alle XML-Typen in den QS-Verfahren. Das Wurzelement besteht immer aus zwei Kind-Elementen (Zweige) **<header>** und **<body>**.

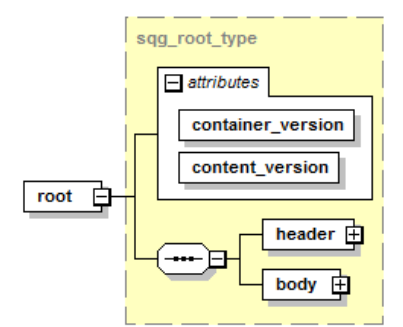


Abbildung 8: Root-Element und Kind-Elemente header und body

Das Root-Element hat zusätzlich zwei Attribute:

Tabelle 7: Root-Element Attribute

Name	Type	Use	Fixed	Beschreibung
container_version	xs:string	required	2.0	Ist ein fixer Wert und definiert die aktuell gültige Versionsnummer des Containers. Die Versionsnummer wird erhöht, wenn Änderungen am Schema des Containers (Umschlags) gemacht werden. Bei kleinen optionalen Änderungen wird die Versionsnummer beibehalten, um die Aufwärtskompatibilität zu gewährleisten. Ein XML-Dokument, das einen alten Wert dieses Attributs enthält, muss von der Datenannahmestelle zurückgewiesen werden.
content_version	content_version_Datentyp	required	1.0	Ist ein fixer Wert und definiert die aktuell gültige Versionsnummer des Inhalts der QS-Daten. Die Versionsnummer wird erhöht, wenn unterjährig das Schema geändert wird.

3.4.4. Header-Bereich

Element header

Das Element Header besteht aus Metadaten (administrative und meldebezogene Daten) zu den QS-Daten, die im **<body>** enthalten sind.

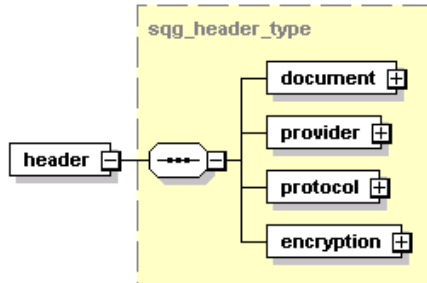


Abbildung 9: Aufbau des Elements header

Element header/document

Das Element enthält allgemeine Informationen zum erstellten Dokument. Dieses Element ist weitestgehend über den gesamten Datenfluss hinweg beständig.

Nur das Element **<modification_dttm>** (Modifikationsdatum) wird bei jeder Bearbeitung neu gesetzt.

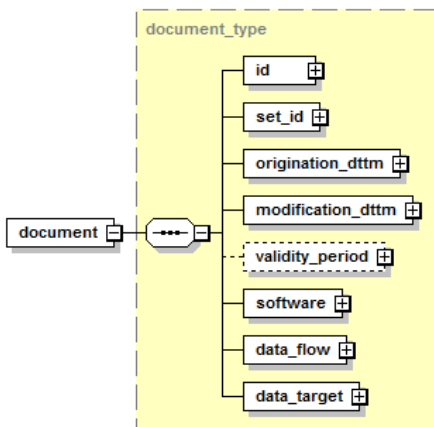


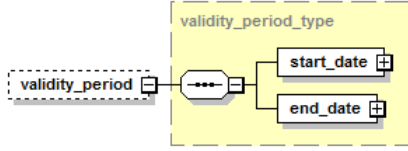
Abbildung 10: Aufbau des Elements document

Dieses Element hat folgende Kind-Elemente:

Tabelle 8: Kind-Elemente des Elements document

<id>	Nach Erstellung nicht mehr modifizierbar. Eindeutige ID des Dokuments, wird vom Dokumentenersteller als GUID erzeugt. ¹⁵
<set_id>	Nach Erstellung nicht mehr modifizierbar. Eindeutige ID für mehrere Dokumente, die im selben Zusammenhang stehen; wird vom Leistungserbringer erzeugt. Dafür könnte z.B. die GUID vom ersten Dokument des Zusammenhangs verwendet werden.

¹⁵ Ein Globally Unique Identifier oder kurz GUID ist eine global eindeutige Zahl mit 128 Bit (16 Bytes), die ein Dokument weltweit eindeutig identifiziert.

<origination_dttm>	Das Element ist der Zeitstempel der ursprünglichen Dokumenterzeugung. Darf nach seiner Erstellung nicht mehr modifiziert werden. Format: CCYY-MM-DDThh:mm:ss
<modification_dttm>	Dieses Element ist ein Zeitstempel und muss bei jeder Modifikation des Dokuments aktualisiert werden. Das Modifikationsdatum darf nicht vor dem Erstelldatum liegen. Format: CCYY-MM-DDThh:mm:ss.
<validity_period>	 <p>Mithilfe des optionalen <validity_period>-Elements wird der Bezugszeitraum der Datei angegeben. Dies kann ein Jahr oder ein Quartal oder ein frei wählbarer Bereich sein. Zur Qualifizierung sind die Felder <start_date> und <end_date> entsprechend zu füllen. Im QSKH-Bereich hat das Element noch keine Anwendung.</p>

<origination_dttm> und **<modification_dttm>** sind vom Datentyp „dateTime“, der einen Zeitpunkt darstellt (ISO 8601). Es handelt sich um das Format CCYY-MM-DDThh:mm:ss:

- „CC“ steht für das Jahrhundert,
- „YY“ steht für das Jahr,
- „MM“ steht für den Monat und
- „DD“ für den Tag.
- Der Buchstabe „T“ dient als Trennzeichen zwischen Datum und Zeit.
- „hh“, „mm“ und „ss“ repräsentieren jeweils Stunden, Minuten und Sekunden.

Dieser Darstellung kann direkt ein „Z“ nachgestellt werden, um anzuzeigen, dass es sich um die Coordinated Universal Time (UTC) handelt. Folgt der Zeitangabe statt eines „Z“ ein Plus- oder Minuszeichen bedeutet das, dass die darauf folgende Angabe im Format „hh:mm“ die Differenz zur UTC angibt (der Minutenanteil ist erforderlich).

Beispiele:

- 2011-11-01T21:32:52
- 2011-11-01T21:32:52+02:00 (Zeitzonendifferenz von plus 2 Stunden)
- 2011-11-01T19:32:52Z

Element header/document/software

Sammelement für Angaben zur eingesetzten QS-Software.

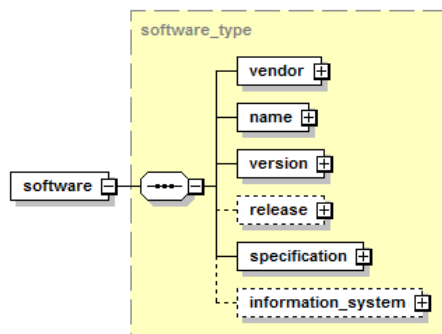


Abbildung 11: Aufbau des Elements software

Dieses Element enthält folgende Kind-Elemente:

Tabelle 9: Kind-Elemente des Elements *software*

<vendor>	Enthält Informationen über den Softwarehersteller.
<name>	Enthält den Softwarenamen der eingesetzten Software.
<version>	Enthält die Version der eingesetzten Software.
<release>	Enthält das Release der eingesetzten Software.
<specification>	Enthält einen Wert aus der enumeration <code>enum_spez_type</code> : Version der Spezifikation, auf deren Basis die QS-Dokumentationssoftware entwickelt wurde.
<information_system>	Enthält Angaben zum eingesetzten Informationssystem (KIS/AIS).

Element header/document/software/information_system

Sammelement für Angaben zum eingesetzten Informationssystem (KIS/AIS)

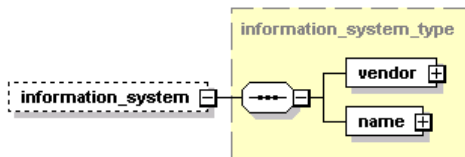


Abbildung 12: Aufbau des Elements *information_system*

Dieses Element enthält folgende Kind-Elemente:

Tabelle 10: Kind-Elemente des Elements *information_system*

<vendor>	Enthält Informationen über den Softwarehersteller.
<name>	Enthält den Softwarenamen der eingesetzten Software.

Beim Element **<software>/<vendor>/<registration>** müssen die ambulanten Leistungserbringer die KBV-Prüfnummer eintragen. Im stationären Bereich müssen die Softwareanbieter die von AQUA-Institut vergebene Registriernummer verwenden.

Element header/document/data_flow

Dieses Element gibt an, für welchen Datenfluss (Datenannahmestelle) dieses Dokument erzeugt wurde.

Tabelle 11: Angabe des betreffenden Datenflusses

QS-Bundesbezogen	für direkte Verfahren, die direkt vom LE an das AQUA-Institut übermittelt werden müssen
QS-Laenderbezogen	für indirekte Verfahren, die an die Landesstellen (LQS/LKGen) übermittelt werden müssen
QS-Kollektivvertraglich	für die Verfahren, die an die kassenärztlichen Vereinigungen (KVen) übermittelt werden müssen
QS-Selektivvertraglich	für die Verfahren, die an die Vertrauensstelle als Datenannahmestelle übermittelt werden müssen
Strukturierter Dialog	<noch nicht benötigt>

Datenvalidierung	<noch nicht benötigt>
Protokoll	<noch nicht benötigt>
Sollstatistik	für die Übermittlung der Sollstatistik

Element header/document/data_target

Dieses Element gibt an, welches Ziel der Datenfluss hat. Mögliche Werte sind:

Tabelle 12: Angabe des betreffenden Datenfluss-Ziels

Echtdatenpool	Echtdaten für den Echtbetrieb
Probdatenpool	Echtdaten für vorläufige Auswertungen wie der Sonderexport
Testdatenpool	Testdaten für Testzwecke

Hinweis:

Im Probetrieb der Qesü-Verfahren sowie im Rahmen der QSKH-Follow-up-Sonderexporte werden nur Datensätze für den Test- und den Probdatenpool von der Datenannahmestelle angenommen (das entspricht den Einträgen „Testdatenpool“ und „Probdatenpool“ im XML-Element **<data_target>**).

Die Kennzeichnung „Echtdatenpool“ ist nur für einen Echtbetrieb vorgesehen.

Element header/Provider

Das Element **<provider>** gibt an, welche Institution dieses Dokument zuletzt bearbeitet hat. Es wird in jeder am Datenfluss beteiligten Instanz durch diese ersetzt und so zur nächsten Instanz geschickt. Einzige Ausnahme ist die Vertrauensstelle, die dort nur im Fall selektivvertraglicher ambulanter Leistungserbringer als „provider“ auftritt. In der Regel lässt die Vertrauensstelle das Provider-Element unberührt.

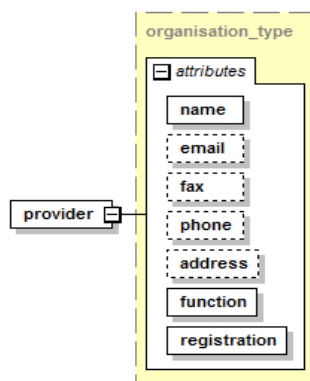


Abbildung 13: Aufbau des Elements provider

Tabelle 13: Attribute des Elements header/provider

Name	Type	Use	Beschreibung
name	xs:string	required	Name der Institution
email	<u>emailAddress_type</u>	optional	E-Mail-Adresse
fax	xs:string	optional	Faxnummer
phone	xs:string	optional	Telefonnummer

Name	Type	Use	Beschreibung
address	xs:string	optional	Adresse
function	enum_organisation_type	required	Bundesauswertestelle/Datenannahmestelle Vertrauensstelle/Softwarehersteller/undefined
registration	registration_type	required	Registriernummer

Element header/protocol

Das Element [<protocol>](#) nimmt Informationen zu Prüfungen auf, die im Datenfluss durchgeführt wurden. Es ist Teil der Rückprotokollierung. Dieses Element ist nicht optional und soll gemeinsam mit dem Unterelement [<status_document>](#) von Anfang an im Datenfluss vorhanden sein, um nachfolgende im Datenfluss vorgenommene Prüfergebnisse aufzunehmen.

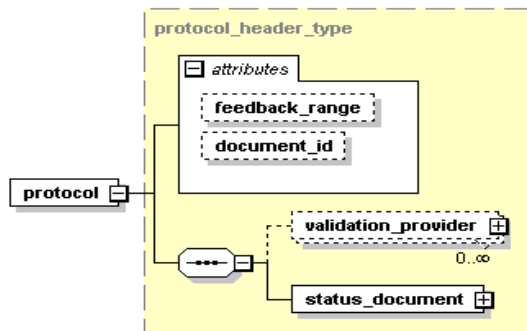


Abbildung 14: Aufbau des Elements header/protocol

Dieses Element hat zusätzlich zu den optionalen Attributen `feedback_range` und `document_id` zwei Kind-Elemente:

[<validation_provider>](#) und [<status_document>](#)

Tabelle 14: Attribute des Elements header/protocol

Name	Type	Use	Beschreibung
feedback_range		optional	Da die Transaktionsprotokolle durch die Empfangsbestätigungen ersetzt worden, ist nur der Wert (dataflow) zu verwenden.
document_id		optional	Soweit die GUID des Exportdokuments lesbar ist, muss sie in das Attribut <code>document_id</code> eingetragen werden.

Element header/protocol/validation_provider

Hier gibt sich die Stelle zu erkennen, die einen oder mehrere Prüfungsschritte durchgeführt hat. Die Ergebnisse der Prüfung werden in diesem Container abgelegt und werden Teil der Rückprotokollierung.

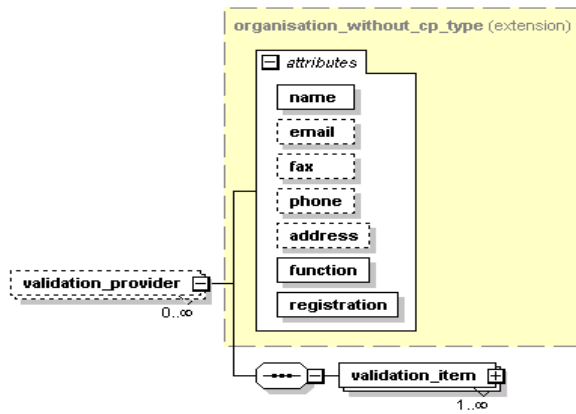


Abbildung 15: Aufbau und Kind-Elemente des Elements validation_provider

Tabelle 15: Attribute des Elements validation_provider

Name	Type	Use
name	xs:string	required
email	emailAddress_type	optional
fax	xs:string	optional
phone	xs:string	optional
address	xs:string	optional
function	enum_validation_provider_type	required
registration	registration_type	required

Element header/protocol/validation_provider/validation_item

Auf Dokumentenebene sind alle Prüfungen zu dokumentieren. Eine prüfende Einrichtung trägt sich als **<validation_provider>** in die entsprechende Auflistung ein und dokumentiert dann ihre durchgeführten Prüfungen in der Auflistung **<validation_item>**.

Es wird als Ergebnis jeder Prüfung eine der folgenden Aussagen über das geprüfte Objekt getroffen:

- OK (Keine Auffälligkeiten)
- WARNING (Auffälligkeiten, die einer Weiterverarbeitung nicht im Weg stehen)
- ERROR (Auffälligkeiten bzw. Fehler, die eine Weiterverarbeitung des Objekts ausschließen)

Das Ergebnis der Prüfung wird in das Attribut „V“ des Elements **<status>** im Element **<validation_item>** eingetragen.

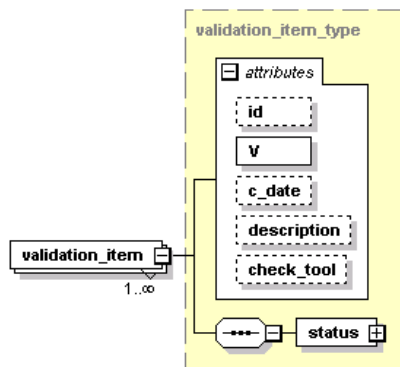


Abbildung 16: Aufbau und Kind-Elemente des Elements validation_item

Tabelle 16: Attribute des Elements *validation_item*

Name	Type	Beschreibung
id	xs:int	Diese ID ist dokumentweit gültig und darf im Header nur einmal vorkommen. Prüfungen auf Datensatzebene (Element „case“), die zu dieser Prüfung gehören, werden über diese ID zugeordnet. Die ID muss nur dann vergeben werden, wenn eine Prüfung auf Fallebene stattfindet.
V	enum_validation_type	Dieser Wert bezeichnet die durchgeführte Prüfung anhand einer „enumeration“, die in sqg_protocol.xsd definiert wird. Gültige Werte sind: Dechiffrierung, LE_Pseudonym, PID_Pseudonym, Protokoll, Schema, Spezifikation, Transaktion, sonstige Prüfung, sonstige Prüfung der XML-Transaktionsdatei.
c_date	xs:dateTime	Hier kann ein Zeitstempel für die Verarbeitung angegeben werden.
description	xs:string	Prüfungsbeschreibung laut Spalte „Prüfung“ in Sicht vPruefung in QSDok Datenbank
check_tool	xs:string	Versionsnummer des Prüftools

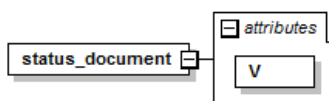
Attribut `/protocol/validation_provider/validation_item/@check_tool`

Hier können beim Einsatz eines Tools für die Prüfung der XML-Dateien der Name und die Versionsnummer des Tools hinterlassen werden (beim Einsatz des Datenprüfprogramms wird die Versionsnummer des XSLT-Skripts eingetragen).

Das Datenprüfprogramm trägt automatisch die Versionsnummer in das `<validation_item>`-Element ein. Damit das Protokoll nicht unnötig groß wird, wird die Information über das Tool nur auf Dokumentenebene aufgenommen (`header/protocol/validation_provider/validation_item/`).

Element `header/protocol/status_document`

Hier wird der Gesamtstatus des Dokuments angegeben, das Attribut `V` kann also auf „OK“, „WARNING“ oder „ERROR“ stehen. Dieser Status kann nur geändert werden, wenn sich der Status des Dokuments verschlechtert oder gleich bleibt. „ERROR“ bedeutet, dass das Dokument komplett zurückgewiesen werden muss.

Abbildung 17: Aufbau des Elements *status_document*Tabelle 17: Attribute des Elements *status_document*

Name	Type	Use	Beschreibung
V	status_type	required	Mögliche Werte: OK/WARNING/ERROR

Element `header/protocol/validation_provider/validation_item/status`

Es gibt an, ob die betroffene Testeinheit ohne Fehler („OK“), mit Fehlern („WARNING“) oder mit fatalem Fehler („ERROR“) abgeschlossen wurde. Der Gesamtstatus des Dokuments entspricht jeweils dem schlechtesten Prüfergebnis. Bei der ersten Prüfung mit dem Ergebnis „ERROR“ muss die Weiterverarbeitung abgebrochen werden.

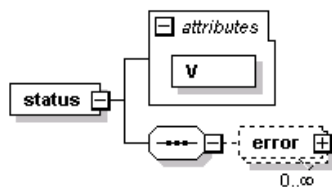


Abbildung 18: Aufbau und Kind-Elemente des Elements status

Tabelle 18: Attribut des Elements status

Name	Type	Use	Beschreibung
v	status_type	required	

Darüber hinaus gibt es die Möglichkeit, eine beliebige Anzahl von **<error>**-Elementen mit einer **<error_message>** im **<status>**-Element unterzubringen.

Element header/protocol/validation_provider/validation_item/status/error

Ein **<error>**-Element nimmt Fehlerdaten auf. Als einzig verpflichtendes Untererelement gilt das **<error_message>**-Element. Die Elemente **<rule_id>** und **<rule_type>** sind spezifisch für die Anwendung von Plausibilitätsregeln für die Spezifikation:

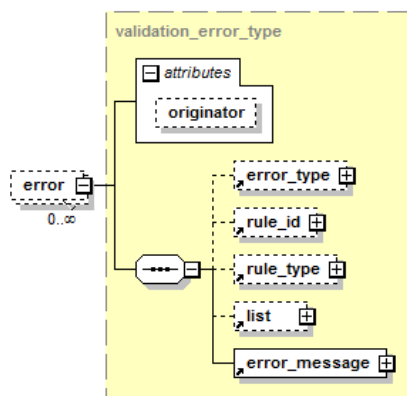


Abbildung 19: Aufbau des Elements error

Tabelle 19: Attribut des Elements error

Name	Type	Use	Beschreibung
originator	enum_organisation_type	optional	Mögliche Werte: Bundesauswertestelle, Datenannahmestelle, Vertrauensstelle, Leistungserbringer, Softwarehersteller, undefined

Tabelle 20: Kind-Elemente des Elements error

<rule_id>	Nummer der Regel (idRegeln in Tabelle Regeln), Oder Nummer der Fehlermeldung aus der Tabelle Fehlermeldungen (idFehlermeldungen)
<rule_type>	Werte H (=hart) oder W (=Warnung bzw. weich)
<liste>	In Abhängigkeit von der Fehlerart entweder Liste von Teildatensätzen oder von Bogenfeldern

<error_message>	Fehlermeldung als Freitext, zzt. nicht fest vorgegeben
<error_type>	Hat folgende Ausprägungen: EXPORT = Formatfehler der Exportdatei DOPPELT = bereits vorhandener Datensatz wird erneut übermittelt TDS = Vollständigkeit und Version der Teildatensätze WERT = Wertebereichsverletzung REGEL = Plausibilitätsverletzung Kollision = Patientenpseudonym mit unterschiedlichen Alter-/Geschlechtsangaben Fehlt = fehlende Daten (LE-Pseudonym, PID-Pseudonym) PID = PID nicht entschlüsselbar (in Kombination mit Dechiffrierung von „validation_item“) QS = QS-Daten nicht entschlüsselbar (in Kombination mit Dechiffrierung von „validation_item“) IST_Statistik = Fehlende/falsche Angaben zur IST-Statistik SOLL_Statistik = Fehlende/falsche Angaben zur SOLL-Statistik

Element header/encryption

Das Element nimmt Informationen über den Schlüssel auf, mit dem die Daten verschlüsselt worden sind. Das Attribut „id“ enthält den Namen des symmetrisch verschlüsselten XML-Knotens. Eine Beispielimplementierung dieser Spezifikation ist ein öffentliches Verschlüsselungsprogramm des AQUA-Instituts (XPack).

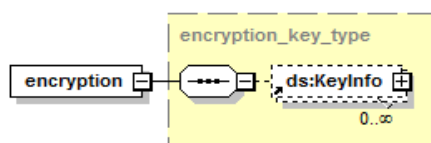


Abbildung 20: Aufbau und Attribute des Elements encryption

Das Programm dient zur Ver- und Entschlüsselung einzelner XML-Elemente (Tags) innerhalb einer XML-Datei, basierend auf einem hybriden Verfahren, das aus folgenden Einzelschritten besteht:

- Ein zufälliger symmetrischer Schlüssel wird erzeugt.
- Mit diesem Schlüssel wird ein XML-Element (z.B. QS-Daten) chiffriert.
- Der Schlüssel wird nun mit dem „public Key“ des Empfängers (z.B. AQUA-Institut) verschlüsselt.
- Der mit dem „public Key“ chiffrierte symmetrische Schlüssel wird dem Empfänger zusammen mit den verschlüsselten Daten übergeben.
- Der Empfänger dechiffriert den Schlüssel mit seinem „private Key“ und erhält so den symmetrischen Schlüssel.
- Mit diesem symmetrischen Schlüssel dechiffriert der Empfänger die verschlüsselten Daten.

Weitere Informationen sind der Dokumentation des Verschlüsselungsprogramms zu entnehmen.

3.4.5. Body-Bereich

Im **<body>**-Element liegen die eigentlichen PID, QS- und LE-Daten. Der Body-Bereich kann einen oder mehrere **<data_container>** enthalten, die einem bestimmten Leistungserbringer zugeordnet sind.

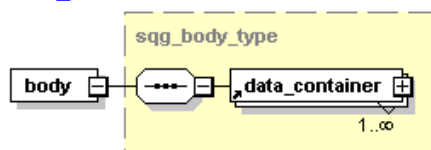


Abbildung 21: Aufbau des Elements body

Kind-Element body/data_container

Ein `<data_container>` ist einem bestimmten Leistungserbringer zuzuordnen. In der Regel sollte in einem Dokument nur ein `<data_container>` vorhanden sein. Da aber mehr als ein `<data_container>` erlaubt ist, können ggf. auch mehrere `<data_container>` für mehrere Leistungserbringer verwendet werden, wenn das Dokument z.B. von einer Stelle (z.B. eine Datenannahmestelle) erstellt wird, die Daten mehrerer Leistungserbringer sammelt.

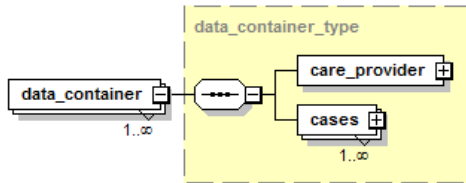


Abbildung 22: Aufbau des Elements body/data_container

Element body/data_container/care_provider

Die Zuordnung zu einem Leistungserbringer erfolgt durch das Element `care_provider`. Dies erfolgt im jeweiligen Sektor (ambulanter oder stationärer Bereich) durch einen unterschiedlichen Aufbau:

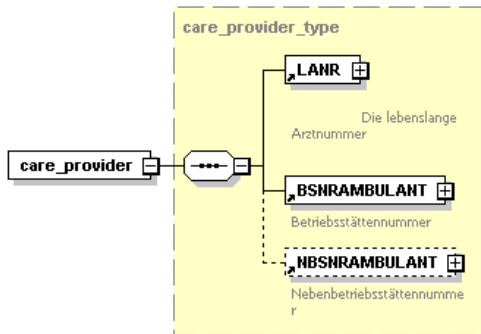


Abbildung 23: Aufbau des Elements care_provider - ambulant

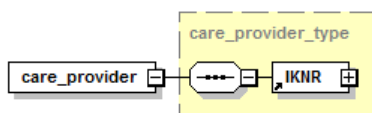


Abbildung 24: Aufbau des Elements care_provider - stationär

Die Kind-Elemente werden in den beiden folgenden Tabellen beschrieben.

Tabelle 21: Leistungserbringeridentifizierende Daten im ambulanten Bereich

<LANR>	Lebenslange Arztnummer – LANR. Für die persönliche Kennzeichnung seiner Leistungen hat jeder Vertragsarzt und -psychotherapeut zum 1. Juli 2008 eine „Lebenslange Arztnummer“ (LANR) erhalten. Diese muss er bei jeder von ihm abgerechneten Leistung und Verordnung angeben.
<BSNRAMBULANT>	Betriebsstättennummer ambulant – BSNR Die BSNR identifiziert die Arztpraxis als abrechnende Einheit und ermöglicht die Zuordnung ärztlicher Leistungen zum Ort der Leistungserbringung. Dabei umfasst der Begriff Arztpraxis auch Medizinische Versorgungszentren (MVZ), Institute, Notfallambulanzen sowie Ermächtigungen von am Krankenhaus beschäftigten Ärzten.
<NBSNRAMBULANT>	Nebenbetriebsstättennummer – NBSNR

Tabelle 22: Leistungserbringeridentifizierende Daten im stationären Bereich (Krankenhaus)

<IKNR>	Institutionskennzeichen – IKNRKH. Gemäß §293 SGB V wird bei der Datenübermittlung zwischen den gesetzlichen Krankenkassen und den Leistungserbringern ein Institutionskennzeichen (IK) als eindeutige Identifizierung verwendet. Mit diesem IK sind auch die für die Vergütung der Leistungen maßgeblichen Kontoverbindungen verknüpft. Die IK werden durch die "Sammel- und Vergabestelle Institutionskennzeichen (SVI)" der Arbeitsgemeinschaft Institutionskennzeichen in Sankt Augustin (SVI, Alte Heerstraße 111, 53757 Sankt Augustin) vergeben und gepflegt. Hier ist das bei der Registrierung für die Qualitätssicherung angegebene IK zu verwenden.
--------	---

Element body/data_container/cases

Container-Element für eine Liste von gleichartigen Fällen (Vorgängen). „Gleichartig“ meint hier Fälle des gleichen Moduls. Das Element enthält einen oder mehrere Vorgänge.

Für unterschiedliche Module müssen jeweils mehrere <cases> angelegt werden.

Die Minimaldatensätze sind unbedingt im korrespondierenden <cases>-Element unterzubringen. MDS gelten nicht als eigenständige Module, sondern sind eine spezielle Datentypausprägung des jeweiligen Erhebungsmoduls.

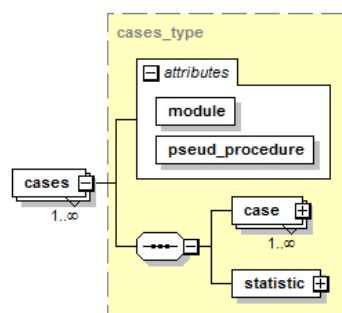


Abbildung 25: Aufbau des Elements cases

Tabelle 23: Attribute des Elements cases

Name	Type	Use	Beschreibung
module	enum_module_type	required	Das Erfassungsmodul
pseud_procedure	enum_procedure_type	required	Zuordnung des Moduls einem Pseudonymisierungsverfahren. Gehört einem Modul keinem Pseuseudonymisierungsverfahren ist das Attribut auf „undefined“ zu setzen.

Das Element **<cases>** enthält ein Attribut „pseud_procedure“: „pseud_procedure“ ist eine Verfahrenskennung, die eindeutig zusammengehörende Exportmodule (z.B. 09/1, 09/2 und 09/3) vermerkt und die es der Vertrauensstelle ermöglicht, die PID verfahrensbezogen zu pseudonymisieren:

Tabelle 24 Verfahrenskennung: „pseud_procedure“

Betrieb	Export-modul	Verfahrenskennung	XML (Umsetzung)
Regelbetrieb 2015	HEP	HEP	<code><cases module="HEP" pseud_procedure="HEP"></code>
	KEP	KEP	<code><cases module="KEP" pseud_procedure="KEP"></code>
	09/1 09/2 09/3	09/1_09/2_09/3	<code><cases module="09/1" pseud_procedure="09/1_09/2_09/3"></code> <code><cases module="09/2" pseud_procedure="09/1_09/2_09/3"></code> <code><cases module="09/3" pseud_procedure="09/1_09/2_09/3"></code>
Sonderexport 2014 (April bis Mai 2015)	17/2 17/3	17/2_17/3	<code><cases module="17/2" pseud_procedure="17/2_17/3"></code> <code><cases module="17/3" pseud_procedure="17/2_17/3"></code>
	17/5 17/7	17/5_17/7	<code><cases module="17/5" pseud_procedure="17/5_17/7"></code> <code><cases module="17/7" pseud_procedure="17/5_17/7"></code>
Regelbetrieb Testbetrieb	Nicht-PID	undefined	<code><cases module="PNEU" pseud_procedure="undefined"></code>

Element body/data_container/cases/case

Das Element **<case>** entspricht einem Vorgang und enthält genau einen QS-Datensatz eines Moduls und abhängig vom Verfahren (direkt, indirekt, PID) die patientenidentifizierenden Daten.

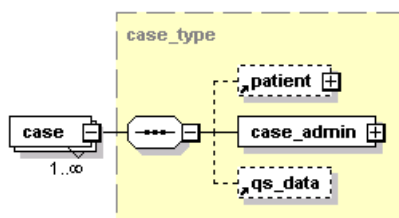


Abbildung 26: Aufbau des Elements case

Element body/data_container/case/case_admin

Das **<case_admin>**-Element enthält weitere Elemente, die einen Vorgang identifizieren. Zusätzlich legt das Element fest, was mit dem Vorgang geschehen soll. Auf Vorgangsebene werden von jeder Prüfstelle der Status der Prüfung und ggf. die Fehler in das Element **<protocol>** eingetragen.

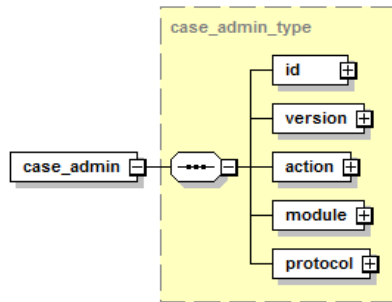


Abbildung 27: Aufbau des Elements case_admin

Im Folgenden werden die einzelnen Kind-Elemente beschrieben:

Tabelle 25: Kind-Elemente des Elements case_admin

<id>	<p>Ist die Vorgangsnummer oder Datensatznummer. Diese Nummer kennzeichnet jeden dokumentierten Datensatz eines Dokumentationssystems eindeutig und zwar unabhängig vom angewandten Modul.</p> <p>Im einfachsten Fall könnte also die Vorgangsnummer um 1 erhöht werden, wenn ein neuer Datensatz angelegt wird. Im stationären Bereich, z.B. wenn während eines Krankenhausaufenthaltes zwei QS-Dokumentationen eines Falls angelegt werden, müssen auch unterschiedliche Vorgangsnummern vergeben werden.</p> <p>Insbesondere ist es falsch, einfach eine Patientenidentifikationsnummer oder die offizielle Fallnummer zu verwenden bzw. zu pseudonymisieren. Bei der Umsetzung hat der Softwareanbieter weitgehende Freiheit, vorausgesetzt die modulübergreifende Eindeutigkeit der Vorgangsnummer ist gewährleistet.</p> <p>Die Vorgangsnummer darf für die datenentgegennehmende Stelle nicht auf Personen zu beziehen sein.</p>
<version>	<p>Enthält eine Versionsnummer des Datensatzes. Sie gibt an, die wievielte Version des Datensatzes übertragen wird.</p> <p>In der Regel wird die Versionsnummer 1 lauten. D.h., dass der nach dem ersten Dokumentationsabschluss freigegebene Datensatz übertragen wird. Muss ein korrigierter Datensatz erneut eingesandt werden, so muss die Versionsnummer vom dokumentierenden System um eins erhöht werden. Die neue Version des Datensatzes wird bei der Entgegennahme geprüft und überschreibt bei Korrektheit die alte Version des Datensatzes.</p> <p>Wenn die datenentgegennehmende Stelle einen Datensatz mit derselben Versionsnummer ein zweites Mal erhält, so wird dieser zurückgewiesen.</p>
<module>	<p>Die Verfahrensabkürzung. Hier ist zu beachten, dass dieser Wert identisch zu den Attributwerten im Element <cases> und <qs_data> sein muss.</p> <p>Wenn die datenentgegennehmende Stelle unterschiedliche Modulbezeichnungen innerhalb eines <cases>-Elements erhält, wird die ganze Datenlieferung zurückgewiesen.</p>
<protocol>	<p>Protokoll auf Vorgangsebene</p>

<action>	<p>Definiert die gewünschte Aktion, kann „create“, „update“ und „delete“ sein.</p> <p>„create“ ist beim ersten Export des Datensatzes zu verwenden, weitere Exporte des Datensatzes werden mit „update“ geliefert.</p> <p>Da nicht alle Datenexporte auch an die Datenannahmestelle verschickt werden (z.B. Testexporte usw.), muss die Datenannahmestelle „update“ und „create“ gleichbehandeln, wenn der erhaltene Datensatz nicht bereits im Datenpool vorhanden ist.</p> <p>Um den Datensatz zu stornieren, muss <action> auf „delete“ gesetzt werden.</p> <p>Die datenentgegennehmende Stelle wird dadurch veranlasst, den betreffenden Datensatz einschließlich aller Vorversionen und Teildatensätze als „storniert“ zu kennzeichnen. Der Stornovorgang wird in der Datenbestätigung protokolliert.</p> <p>Der zu stornierende Datensatz muss ebenfalls eine hochgezählte/fortgeschriebene Versionsnummer enthalten, um die Stornierung unabhängig von der Reihenfolge der Verarbeitung von Datensätzen sicherzustellen. Ein Storno mit einer bereits verwendeten Versionsnummer wird zurückgewiesen (Bestätigungsstatus ERROR, Fehlerart DOPPELT). Ein Stornoversuch eines noch nicht übermittelten Datensatzes wird ebenfalls zurückgewiesen.</p>
-----------------------	---

Element patient (PID-Module)

Das Element enthält die patientenidentifizierenden Daten des übergeordneten Vorgangs. Das Kind-Element von **<patient>** ist das Element **<pid>**. Das Attribut **twodigitik** ist verpflichtend und muss die ersten 2 Stellen des Institutionskennzeichens der Krankenkasse enthalten. Das Attribut ist nicht von der Verschlüsselung betroffen.

Achtung:

Es dürfen keine PID von nicht gesetzlich versicherten Patienten exportiert werden. Nur PID von GKV-Patienten sind zulässig. QS-Daten (**<qs_data>**) sind davon nicht betroffen und müssen weiterhin unabhängig von dem Versicherungsstatus des Patienten exportiert werden.

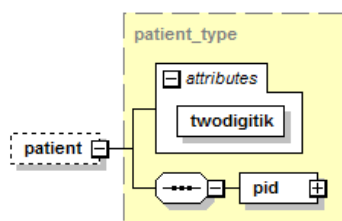


Abbildung 28: Aufbau des Elements patient

Element patient_type/pid

Das Element **<pid>** dient dazu, die tatsächlichen PID aufzunehmen.

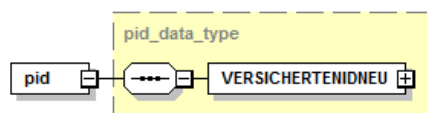


Abbildung 29: Aufbau des Elements pid

Die PID bestehen aus dem Element **<VERSICHERTENIDNEU>**, der eGK-Versichertennummer.

Element case_Datentyp/case_admin/protocol

Dieses Element hat eine auf Dokumentenebene ähnliche Struktur wie das oben beschriebene Element **<protocol>**.

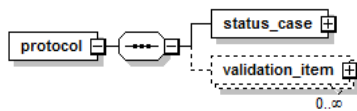


Abbildung 30: Aufbau des Elements case_admin/protocol

Die Unterschiede sind:

- Während das **<protocol>**-Element im Header Ergebnisse der Prüfungen, die das Dokument insgesamt betreffen, aufnimmt, nimmt das Protokoll-Element im Body-Bereich die Ergebnisse der Prüfungen auf, die auf Vorgangsebene (Datensatzebene) erfolgen.
- Für erfolgreiche Prüfergebnisse (status = „OK“) wird nicht explizit das Element **<validation_item>** erstellt. Für die Übermittlung des Status des Datensatzes dient weiterhin der implizite Wert des Elements **<status_case>** (**<status_case V=“OK“>**).
- **<status_case>** beinhaltet das schlechteste Ergebnis aller Prüfungen eines Datensatzes.
- Das Protokoll auf Vorgangsebene hat kein Element **<validation_provider>** (Prüfstelle). Damit auch auf dieser Ebene die Ergebnisse der durchgeführten Prüfungen einer Prüfstelle zugeordnet werden können, müssen alle Ergebnisse einer Prüfung auf Fallebene mit einer gemeinsamen, dokumentweit eindeutigen ID im Attribut ID des Elements **<validation_item>** eingetragen werden.

Element body/data_container/cases/statistic

Das Element **<statistic>** dient dazu, Statistiken über die Datenlieferung des Absenders und über deren Verarbeitung durch die datenentgegennehmende Stelle aufzunehmen.

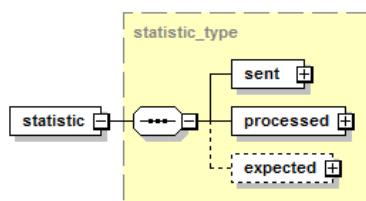


Abbildung 31: Aufbau des Elements statistic

Es ist nach der Prüfung bei der Datenannahmestelle ein Teil des Rückprotokolls und besteht aus folgenden, ähnlichen Kind-Elementen:

Tabelle 26: Kind-Elemente des Elements <statistic>

<sent>	Statistik über die von dem Datenlieferant exportierten Datensätze. Es muss daher vom Datenlieferant selbst ausgefüllt werden.
<processed>	Hat dieselbe Struktur wie <sent> und enthält das Ergebnis der Verarbeitung durch die datenentgegennehmende Stelle.
<expected>	Optionales Element, wird von ambulanten LE verwendet, um für den Exportzeitraum eine Sollstatistik zu übermitteln.

Element statistic/sent

Das Element nimmt Statistiken über die von dem Datenlieferant exportierten Datensätze auf und muss vom Datenlieferant selbst ausgefüllt werden.

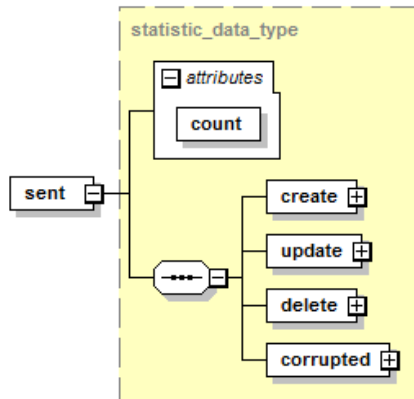


Abbildung 32: Aufbau des Elements sent

Es besteht aus vier Elementen und einem Attribut:

Tabelle 27: Attribut des Elements sent

Name	Type	Use	Beschreibung
count	non_negative_integer_type	required	Gesamtzahl von Vorgängen (Ganze Zahl >= 0): Summe von (<create>+<update>+<delete>+<corrupted>)

Tabelle 28: Kind-Elemente des Elements statistic/sent

<create>	Anzahl der Datensätze, die neu importiert werden sollen.
<update>	Anzahl der Datensätze, die aktualisiert werden sollen (z.B. nach einer Korrektur).
<delete>	Anzahl der Datensätze, die von der Datenannahmestelle/Bundesauswertungsstelle storniert werden müssen.
<corrupted>	Anzahl der Datensätze, die fehlerhaft sind. Der Datenabsender trägt hier „0“ ein.

Element statistic/expected

Das Element nimmt die von dem Datenlieferant anzugebende Sollstatistik auf.

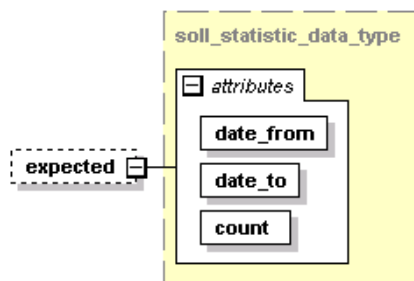


Abbildung 33: Aufbau des Elements expected

Tabelle 29: Attribute des Elements *expected*

Name	Type	Use	Beschreibung
date_from	xs:datetime	required	Dokumentationszeitraum ab
date_to	xs:datetime	required	Dokumentationszeitraum bis
count	non_negative_integer_type	required	Anzahl der erwarteten Fälle

Element *statistic/processed*

Das Element **<processed>** hat dieselbe Struktur wie das Element **<sent>** mit dem Unterschied, dass der Datenempfänger nach der Prüfung der Exportdatei in das Element **<processed>** eintragen soll, wie viele Datensätze er tatsächlich neu importiert, überschrieben und storniert hat und ggf. wie viele Datensätze fehlerhaft sind. Außerdem soll er im Attribut **count** des Elements **<processed>** die Gesamtsumme eintragen.

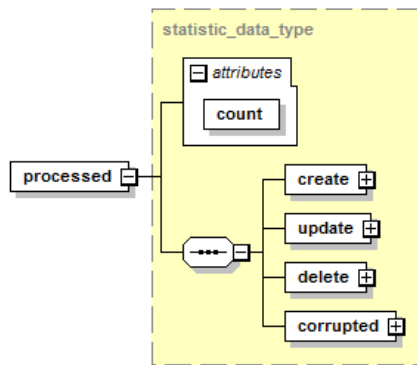


Abbildung 34: Aufbau des Elements *processed*

<processed> ist vom Datenlieferant anzulegen und besteht aus vier Elementen und einem Attribut:

Name	Type	Use	Beschreibung
count	non_negative_integer_type	required	Gesamtzahl von Vorgängen (Datensätzen) : Summe von (<create> + <update> + <delete> + <corrupted>)

Tabelle 30: Kind-Elemente des Elements *statistic/processed*

<create>	Anzahl der Datensätze, die der Datenempfänger nach der Prüfung tatsächlich neu importieren konnte.
<update>	Anzahl der Datensätze, die der Datenempfänger nach der Prüfung tatsächlich aktualisieren konnte.
<delete>	Anzahl der Datensätze, die der Datenempfänger tatsächlich stornieren konnte.
<corrupted>	Anzahl der Datensätze, die wegen Mängeln nicht entsprechend der Absicht des Datenlieferanten in den Datenpool übernommen werden konnten.

Alle Werte müssen vom Leistungserbringer mit der Zahl „0“ vorbelegt werden.

Zusammenfassend kann man die Angaben im Element **<sent>** als Absichtserklärung der Datenlieferung betrachten. Die Angaben im Element **<processed>** sind als Ergebnis der Verarbeitung in Bezug auf den Zieldatenpool zu verstehen und werden daher nur von der Datenannahmestelle gesetzt.

Element `qs_data`

Das `<qs_data>`-Element ist ein Container für die QS-Daten, die verfahrensspezifisch sind. Der Datentyp der konkreten `<qs_data>`-Instanz ist jeweils als Attribut `xsi:type` bei der Dokumenterstellung zu definieren.

Beispiel: `<qs_data xsi:type="qs_data_17n2_type" module="17/2">`

Wie bereits oben erwähnt, würden ggf. vorhandene MDS (Minimaldatensätze) unter dem zugehörigen Modul definiert und eingehängt.

Beispiel: `<qs_data xsi:type="qs_data_mds_type" module="17/2">`

Die Abbildung zeigt als Beispiel, wie ein Minimaldatensatz zum Primärmodul NEO exportiert wird.

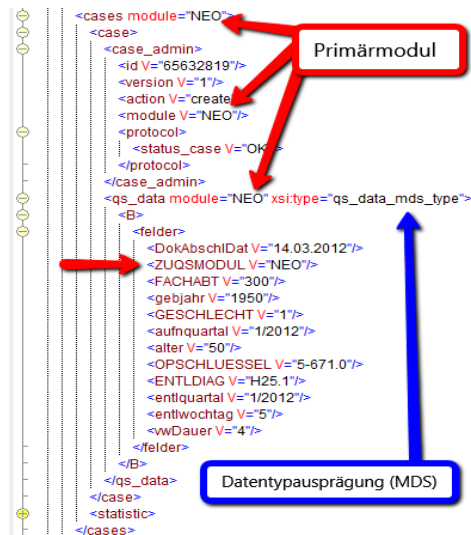


Abbildung 35: Integration des MDS in das Primärmodul

Beispiel: Minimaldatensatz des Hauptmoduls Neonatologie (NEO)

`<qs_data xsi:type="qs_data_mds_type" module="NEO">`

Die genaue Struktur eines Moduls ist aus der Access-Datenbank für die QS-Software und dem entsprechenden XML-Schema zu entnehmen.

Der Aufbau des XML-Elements `<qs_data>` ist variabel und abhängig von der Struktur des jeweiligen Erfassungsmoduls. Die genaue Struktur eines Moduls (nach dem Export) ist von der Spezifikationsdatenbank vorgegeben. Im Allgemeinen gilt Folgendes:

- Jedes Modul hat immer einen einzigen Basisdatensatz.
- Komplexe Module können zusätzlich mehrere Teildatensätze (Bögen) enthalten, die sich hierarchisch anordnen lassen.
- Jeder Teildatensatz besitzt einen Namen, der innerhalb eines Moduls eindeutig ist und unter definierten Bedingungen mehrfach pro Fall erzeugt werden kann.

Folgende Abbildung zeigt einen komplexen und einen einfachen Bogen:

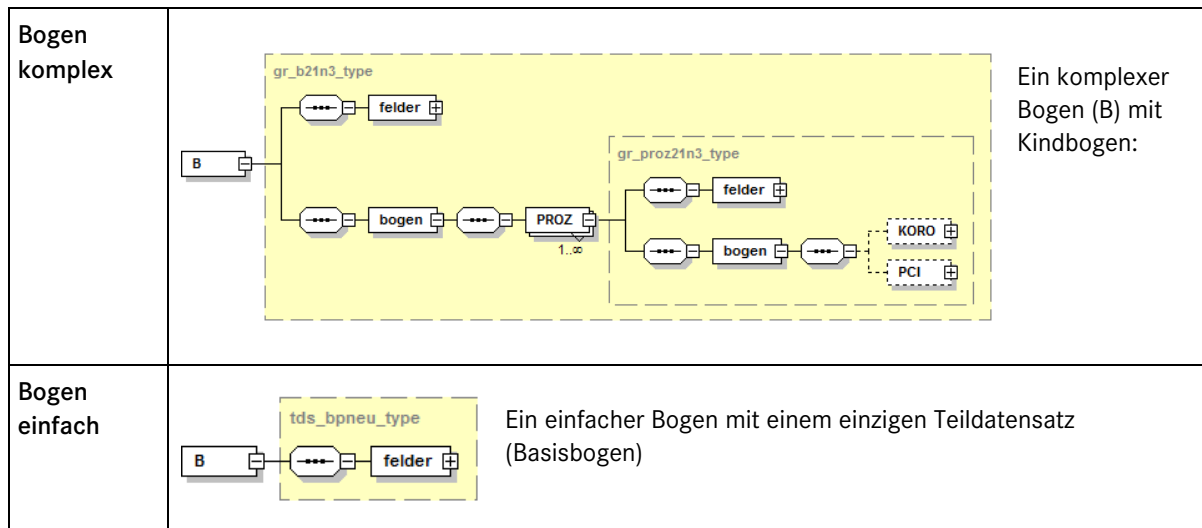


Abbildung 36: Diagramme „Bogen komplex“ und „Bogen einfach“

Wie die folgende Abbildung zeigt, sind Die XML-Schemata der einzelnen Module in der Schnittstelle interface_LE zu finden:

- 4 interface_LE
 - 4 direkt
 - qs_data_hch_type.xsd
 - qs_data_htbfu_type.xsd
 - qs_data_hbxm_type.xsd
 - qs_data_ils_type.xsd
 - qs_data_ilsfu_type.xsd
 - qs_data_itx_type.xsd
 - qs_data_itbfu_type.xsd
 - qs_data_itub_type.xsd
 - qs_data_itubfu_type.xsd
 - qs_data_nls_type.xsd
 - qs_data_nlsfu_type.xsd
 - qs_data_pntx_type.xsd
 - qs_data_pntbfu_type.xsd
 - indirekt
 - mds
 - pid

Abbildung 37 Ausprägungen des qs_data-Elements (Erfassungsmodule)