



---

# Technische Dokumentation zur GEKID-Spezifikation

Erfassungsjahr 2016

---

Stand: 03. November 2015 (V02)

# Impressum

**Herausgeber:**

AQUA – Institut für angewandte Qualitätsförderung und Forschung im Gesundheitswesen GmbH

**Thema:**

Technische Dokumentation zur GEKID-Spezifikation – Erfassungsjahr 2016

**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](mailto: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](mailto:k.bezold@dkfz-heidelberg.de)

**gültig ab:**

01. Januar 2016

**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](mailto:office@aqua-institut.de)

[www.aqua-institut.de](http://www.aqua-institut.de)

# Inhaltsverzeichnis

Tabellenverzeichnis .....	4
Abbildungsverzeichnis.....	5
1 Einleitung .....	6
1.1 Neueste Nachrichten .....	6
1.2 Projekthintergrund.....	6
1.3. Gültigkeit der GEKID-Spezifikation .....	8
1.4 Lieferumfang .....	8
2 Aufbau der Spezifikationsdatenbank.....	9
2.1 Allgemeine Anmerkungen zur Struktur der Spezifikationsdatenbank.....	9
2.1.1 Abfragen der Datenbank .....	9
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.....	25
3.4.2 Kompositionsmodell .....	25
3.4.3 Schnittstellen.....	26
3.4.4 Darstellung der XML-Struktur.....	28
3.4.5 Grundstruktur der XML-Dateien .....	29
3.4.6 Wurzelement <root>.....	29
3.4.7 Header-Bereich .....	30
3.4.8 Body-Bereich .....	41

# Tabellenverzeichnis

Tabelle 1: Datenexport – Unterschiede zwischen dem QS-Verfahren und dem EKR-Verfahren .....	13
Tabelle 2: Kodes der Krebsregister .....	14
Tabelle 3: Ausfüllen der Elemente eines Validation-Items in Abhängigkeit von den Fehlerarten .....	19
Tabelle 4: Verwendbare Schemata und Ablageort .....	26
Tabelle 5: Weiche Schemavarianten für das Datenprüfprogramm .....	27
Tabelle 6: XML-Schemata für die Rückprotokollierung .....	27
Tabelle 7: Symbole in den XML-Schema-Diagrammen .....	28
Tabelle 8: Root-Element Attribute .....	29
Tabelle 9: Kind-Elemente des Elements document .....	31
Tabelle 10: Kind-Elemente des Elements software .....	32
Tabelle 11: Kind-Elemente des Elements information_system .....	32
Tabelle 12: Angabe des betreffenden Datenflusses .....	33
Tabelle 13: Angabe des betreffenden Datenfluss-Ziels .....	33
Tabelle 14: Attribute des Elements header/provider .....	34
Tabelle 15: Attribute des Elements header/protocol .....	35
Tabelle 16: Attribute des Elements validation_provider .....	36
Tabelle 17: Attribute des Elements validation_item .....	37
Tabelle 18: Attribute des Elements status_document.....	37
Tabelle 19: Attribut des Elements status .....	38
Tabelle 20: Attribut des Elements error .....	39
Tabelle 21: Kind-Elemente des Elements error .....	39
Tabelle 22: feedback_key: Kind-Elemente .....	40
Tabelle 23: Leistungserbringeridentifizierende Daten im ambulanten Bereich .....	42
Tabelle 24: Leistungserbringeridentifizierende Daten im stationären Bereich (Krankenhaus) .....	42
Tabelle 25: Attribute des Elements cases .....	43
Tabelle 26: Verfahrenskennung: „pseud_procedure“ .....	44
Tabelle 27: Kind-Elemente des Elements case_admin .....	45
Tabelle 28: Kind-Elemente des Elements <statistic>.....	47
Tabelle 29: Attribut des Elements sent .....	47
Tabelle 30: Kind-Elemente des Elements statistic/sent.....	48
Tabelle 31: Kind-Elemente des Elements statistic/processed .....	48

# Abbildungsverzeichnis

Abbildung 1: Beziehungen zwischen <validation_item> im header und <validation_item> im body über die id ..	21
Abbildung 2: Aufnahme des XSLT-Pfads in das XML-Protokoll.....	23
Abbildung 3: HTML-Darstellung nach einer XSLT-Transformation am Beispiel einer QS Übertragung.....	24
Abbildung 4: Dateiordner der Schnittstellen-Schemata .....	25
Abbildung 5: Root-Element und Kind-Elemente header und body .....	29
Abbildung 6: Aufbau des Elements header .....	30
Abbildung 7: Aufbau des Elements document .....	30
Abbildung 8: Aufbau des Elements software .....	32
Abbildung 9: Aufbau des Elements information_system .....	32
Abbildung 10: Aufbau des Elements provider .....	34
Abbildung 11: Aufbau des Elements header/protocol .....	35
Abbildung 12: Aufbau und Kind-Elemente des Elements validation_provider .....	35
Abbildung 13: Aufbau und Kind-Elemente des Elements validation_item .....	36
Abbildung 14: Aufbau des Elements status_document.....	37
Abbildung 15: Aufbau und Kind-Elemente des Elements status .....	38
Abbildung 16: Aufbau des Elements error .....	38
Abbildung 17: Aufbau und Attribute des Elements encryption .....	39
Abbildung 18: Aufbau und Attribute des Elements encryption(AP-selektiv/-kollektiv) .....	40
Abbildung 19: Aufbau des Elements feedback_key .....	41
Abbildung 20: Aufbau des Elements body .....	41
Abbildung 21: Aufbau des Elements body/data_container.....	41
Abbildung 22: Aufbau des Elements care_provider – ambulant.....	42
Abbildung 23: Aufbau des Elements care_provider – stationär.....	42
Abbildung 24: Aufbau des Elements cases .....	43
Abbildung 25: Aufbau des Elements case.....	44
Abbildung 26: Aufbau des Elements case_admin .....	45
Abbildung 27: Aufbau des Elements case_admin/protocol .....	46
Abbildung 28: Aufbau des Elements statistic .....	47
Abbildung 29: Aufbau des Elements sent .....	47
Abbildung 30: Aufbau des Elements processed.....	48
Abbildung 31: Integration des MDS in das Primärmodul.....	49
Abbildung 32: Diagramme „Bogen komplex“ und „Bogen einfach“ .....	50
Abbildung 33: Ausprägungen des qs_data-Elements (Erfassungsmodule) .....	50

# 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.

Das Format für die zu exportierenden Datensätze wurde zum Erfassungsjahr 2015 von CSV auf XML umgestellt. Die Umstellung erfolgt analog zum Exportformat der Datensätze der QS-Dokumentation.<sup>1</sup> Informationen zur XML-Struktur sind Kapitel 3 zu entnehmen.

Mit der Spezifikation 2016\_BASIS\_QSDOK\_V02 wird im Datenfeld zum Entlassungsgrund ein Schlüsselwert korrigiert:

2015 V03	2016 V02
16 = interne Verlegung mit Wechsel zwischen den Entgeltbereichen der DRG-Fallpauschalen, nach der BPflV oder für besondere Einrichtungen nach §17b Abs. 1 Satz 15 KHG	16 = externe Verlegung mit Rückverlegung oder Wechsel zwischen den Entgeltbereichen der DRG-Fallpauschalen, nach der BPflV oder für besondere Einrichtungen nach §17b Abs. 1 Satz 15 KHG mit Rückverlegung

Der Basistyp des Feldes FACHABT wird in der Spezifikation 2016 V02 von NUMSCHLUESSEL zu SCHLUESSEL geändert, sodass die Übermittlung der in den vierstelligen Schlüsselwerten bereits vorhandenen führenden Null sichergestellt ist.

Weiterhin wird das Datenfeld Institutionskennzeichen der Krankenkasse der Versichertenkarte [KASSEIKNR] neu aufgenommen.



### Achtung Datenexport

Die Felder VERSICHERTENIDNEU und KASSEIKNR werden im Schema abgebildet. Diese sind – wie durch die Datenbank zur QS-Dokumentation und durch die vorliegende technische Dokumentation spezifiziert – im Element <qs\_data> im Bogen B zu exportieren.

In der Technischen Dokumentation 2016\_TechDok\_GEKID\_V02 zum Spezifikationspaket V04 wird ein neuer Abschnitt mit Hinweisen zur Gültigkeit der GEKID-Spezifikation eingefügt.

## 1.2 Projekthintergrund

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

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

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

Die Weiterleitung der auf diesem Wege erzeugten Transaktionsdateien muss mit dem jeweiligen Landeskrebsregister 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-Mindestdatensätze verantwortlich und fungiert nicht als Datenannahmestelle. Für die inhaltliche Bearbeitung sind die Krebsregister NRW und Baden-Württemberg zuständig, an die auch Änderungsvorschläge zu adressieren sind. Ansprechpartner sind:

**Epidemiologisches Krebsregister Nordrhein-Westfalen GmbH**

Robert-Koch-Str. 40

48149 Münster

Dr. Volkmar Mattauch

E-Mail: [volkmar.mattauch@krebsregister.nrw.de](mailto: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](mailto:k.bezold@dkfz-heidelberg.de)

## 1.3 Gültigkeit der GEKID-Spezifikation

Die GEKID-Spezifikation findet ihre Gültigkeit zum 1. Januar 2016. Vor dem Hintergrund des Krebsfrüherkennungs- und -registergesetzes (KFRG) vom 3. April 2013 wird ab dem Jahr 2016 eine flächendeckende klinische Krebsregistrierung in den Bundesländern etabliert. Das entsprechende Meldewesen basiert dann auf dem gemeinsamen onkologischen Basisdatensatz der Arbeitsgemeinschaft Deutscher Tumorzentren e.V. (ADT) und der Gesellschaft der epidemiologischen Krebsregister in Deutschland e.V. (GEKID). Das AQUA-Institut stellt hierfür eine ADT/GEKID-Spezifikation zur Verfügung.<sup>2</sup> In diesem Rahmen können auch die epidemiologischen Meldungen über den ADT/GEKID-Basisdatensatz erfolgen.

Im Bundesland Nordrhein-Westfalen ist die Lieferung von Datensätzen nach der GEKID-Spezifikation im gesamten Erfassungsjahr 2016 weiterhin möglich. Eine Annahme der Daten nach der ADT/GEKID-Spezifikation ist ab dem 1. April 2016 vorgesehen.

Im Bundesland Baden-Württemberg erfolgt die standardmäßige Datenannahme nach ADT/GEKID-Spezifikation ab dem 1. Juli 2016 – eine parallele Nutzung der GEKID-Spezifikation ist hier nicht weiter vorgesehen. Nach Absprache mit dem Krebsregister Baden-Württemberg, kann eine Schnittstellenabnahme und Pilotierung der ADT/GEKID-Spezifikation ab dem 1. April 2016 erfolgen.

Für eine Nutzung der GEKID-Spezifikation in anderen Bundesländern ist eine Abstimmung mit den dortigen Krebsregistern zu Möglichkeiten und Zeiträumen der Datenannahme erforderlich.

## 1.4 Lieferumfang

Die vorliegende Spezifikation besteht aus folgenden Teilen<sup>3</sup>:

- Technische Dokumentation: 2016\_TechDok\_GEKID\_V<Versionsnummer>.pdf
- Datenbank: 2016\_Basis\_QSDOK\_V<Versionsnummer>.mdb (Modul: GEKID)<sup>4</sup>
- Ausfüllhinweis: Ausfuellhinweise\_GEKID.htm
- Dokumentationsbogen: GEKID.pdf

---

<sup>2</sup> Veröffentlicht auf <https://www.aqua-institut.de/de/projekte/spezifikation-krebsregister/index.html>

<sup>3</sup> 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>)

<sup>4</sup> 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 2016 ist die Spezifikation `2016_Basis_QSDOK_V045.mdb` gültig.

Weitere Erläuterungen finden Sie in der Technischen Dokumentation „2016\_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.

#### 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.

---

<sup>5</sup> Die Versionsnummer der gültigen Spezifikation (z.B. V01, V02, usw.) ist dem zuletzt veröffentlichten Update zu entnehmen.

- **Ersatzfelder**  
Dies ist eine Auflistung der zu anonymisierenden Bogenfelder für alle spezifizierten Module.
- **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 ZAHN und GANZEZAHN) mit den jeweiligen Wertebereichen.
- **WertebereicheNumerischerFelderFürEinModul**  
Hier werden die numerischen Datenfelder (Typ ZAHN und GANZEZAHN) 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 Namens- teil 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 Attribut<sup>6</sup>, welches durch Setzen eines weiteren, eindeutigen Index (bestehend aus einem oder mehreren Attributen) definiert ist.

---

### Beispiel2:

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

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<FremdTabellenName>
```

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:

```
<fkFremdTabellenName><Rolle>
```

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

---

<sup>6</sup> Oder eine identifizierende Attributkombination, die einen eindeutigen Schlüssel definiert.

N-M-Beziehungen werden wie üblich über Verknüpfungstabellen realisiert. In der Spezifikation haben Verknüpfungstabellen gewöhnlich keinen Primärschlüssel<sup>7</sup>, 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 „2016\_TechDok\_LE\_XML\_V<Versionsnummer>“ entnommen werden, die auf der folgenden Webseite veröffentlicht ist: <https://www.sgg.de/datenservice/spezifikationen-downloads/index.html>.

---

<sup>7</sup> 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 Dokumentationsabschlusses<sup>8</sup> 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.

<sup>8</sup> 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 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>`

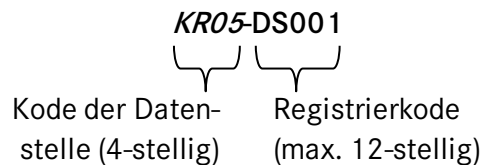
Der Kode der Datenstelle „Krebsregister“ setzt sich zusammen aus dem Kürzel KR und dem amtlichen Länderkode (siehe Tabelle 2). Die maximal 12-stelligen Registrierkodes werden von den Krebsregistern gepflegt.

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
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.).<sup>9</sup>

**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 Leistungsfalles 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.

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 Datenvvalidierung und der Qualitätsverbesserung muss es möglich sein, über die Vorgangsnummer Zugang zur Fall- bzw. Patienten-Akte bekommen zu können.

<sup>9</sup> Fiktives Beispiel.

### **Annahme oder Ablehnung unterschiedlicher Versionen eines Datensatzes**

Datensätze, die bei der entgegennehmenden Stelle eingehen, werden anhand der Registrierungsnummer und Vorgangsnummer als ein Vorgang identifiziert. Der für den Vorgang gespeicherte Datensatz kann durch eine neuere Version (mit höherer Versionsnummer) überschrieben werden.<sup>10</sup>

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.

### **Ausgangvalidierung 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 Ausgangvalidierung:

- 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

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.<sup>11</sup>

---

<sup>10</sup> Ggf. ist der geänderte Datensatz mit einer neuen Versionsnummer zu übermitteln.



## 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 „&amp;“ bzw. „&lt;“ geschehen. Die schließende spitze Klammer (>) kann durch die Zeichenkette „&gt;“ dargestellt werden. Um Attributwerten zu erlauben, sowohl das einfache als auch das doppelte Anführungszeichen zu enthalten, kann der Apostroph (') als „&apos;“ und das doppelte Anführungszeichen (") als „&quot;“ 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 `Bogen-Feld`) eines Moduls enthalten sind, werden von der QS-Dokumentationssoftware ausgefüllt.<sup>12</sup>

Die Zusatzfelder sind in der Tabelle `ZusatzFeld` 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.

---

<sup>11</sup> <http://www.w3.org/XML/Schema>

<sup>12</sup> Hier gilt also nicht der Grundsatz, dass Felder nicht vorbelegt sein dürfen.

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 Teildatensatzes<sup>13</sup>

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 gestellt. 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>;
```

<sup>13</sup> Der Bezug zum Modul kann entfallen, da dieses über die Vorgangsnummer identifiziert werden kann.

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 eines Protokolls:

```
<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="AUFNADATUM"/>
        <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) GEKNI 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>
```

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.

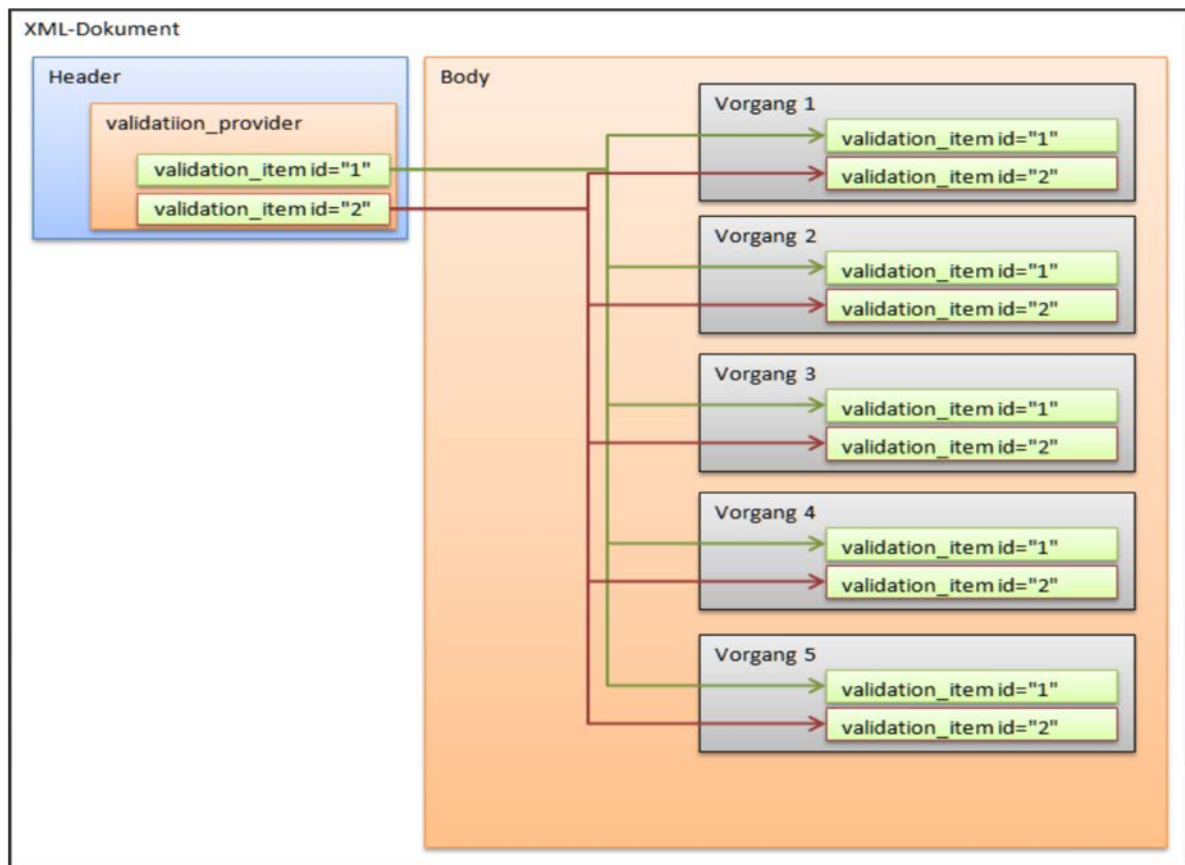


Abbildung 1: Beziehungen zwischen `<validation_item>` im header und `<validation_item>` im body über die `id`

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 1).

### 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:

- (10) → <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.
- (11) → <status\_document> auslesen.
- (12) → 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.sqg.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 2: 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			
<a href="#">» Details ein/aus</a>			
<b>Bundesauswertestelle</b> Aqua-Institut GmbH (BU10000) Maschmühlenweg 8–10, 37073 Göttingen		tel: (+49)0551/789 52-296 fax: (+49)0551/789 52-295 xml-ba@aqua-institut.de	
Validierung 776	2012-10-11T13:17:57	Spezifikation	ERROR
<b>Unbekannte Funktion</b> <b>Unbekannter Provider</b>		tel: - fax: -	
Validierung 774	2012-10-11T13:17:57	Spezifikation	OK
<b>Bundesauswertestelle</b> <b>name (1234567)</b> 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 3: 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.



#### **Achtung** **Datenexport**

Die Felder VERSICHERTENIDNEU und KASSEIKNR sind im Element <qs\_data> im Bogen B zu exportieren.

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

Die XML-Schema-Datei (XSD) ist eine Empfehlung des W3C<sup>14</sup> zum Definieren von Strukturen für XML-Dokumente.

In der Basisspezifikation werden die XML-Schemata aus der der Spezifikationsdatenbank abgeleitet und haben die Aufgabe, die aktuellen Datenflussmodelle der G-BA-Richtlinien abzubilden:

- Abbildung der zu exportierenden QS-Daten
- Abbildung der Schnittstellen an den Institutionsübergängen (z.B. die Schnittstelle zwischen DAS und VST 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.

### 3.4.2 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.

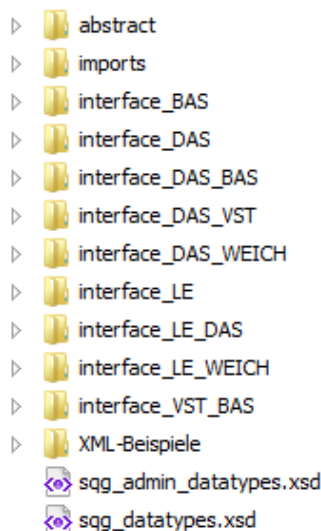


Abbildung 4: Dateiordner der Schnittstellen-Schemata

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. Dieses Kompositionsmodell wird hier näher erläutert.

<sup>14</sup> <http://www.w3.org/XML/Schema>

**Beispiel:**

Die leistungserbringeridentifizierenden Daten existieren gemäß Datenflussmodell der G-BA-Richtlinien in drei Ausprägungen:

- im Klartext (Schnittstelle LE)
- pseudonymisiert (Schnittstelle DAS)
- verschlüsselt (Schnittstelle DAS-VST)

Bei dieser Konzeption werden alle drei Ausprägungen vom selben Basisdatentyp geerbt, in einem zweiten Schritt die drei Ausprägungen konkretisiert und angepasst, und über „includes“ in die jeweilige Schnittstelle integriert.

**3.4.3 Schnittstellen**

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

Tabelle 4: Verwendbare Schemata und Ablageort

Schnittstelle	Schema	Verfahren	Absender	Empfänger
LE interface_LE	2016_bas_dv_1.0_Export	Direkte Verfahren (dv)	KH	BAS
	2016_kv_pid_1.0_Export	PID-Verfahren (pid)	KH/PR-Kollektiv	KV
	2016_lqs_iv_1.0_Export	Indirekte Verfahren (iv)	KH	LQS
	2016_lqs_pid_1.0_Export	PID-Verfahren (pid)	KH	LQS/LKG
	2016_sv_pid_1.0_Export	PID-Verfahren (pid)	PR-Selektiv	DAS-SV
LE/DAS interface_LE_DAS	interface_LE_BAS	Direkte Verfahren (dv)	KH	BAS
	interface_LE_KV	PID-Verfahren (pid)	PR-Kollektiv	KV
	interface_LE_LQS_iv	IV-Verfahren	KH	LQS
	interface_LE_LQS_pid	PID-Verfahren	KH	LQS/LKG
	interface_LE_SV	PID-Verfahren	PR-Selektiv	DAS-SV
DAS interface_DAS	interface_KV_psn	PID-Verfahren	Nach der LE-Pseudonymisierung	
	interface_LQS_iv	IV-Verfahren	Nach der QS-Entschlüsselung	
	interface_LQS_pid	PID-Verfahren	Nach der QS-Entschlüsselung	
	interface_LQS_psn	Alle Verfahren	Nach der LE-Pseudonymisierung	
	interface_SV_psn	PID-Verfahren	Nach der LE-Pseudonymisierung	
	Interface_KV_soll.xsd	Sollstatistik	Erstellung der Sollstatistik	
	Interface_SV_soll.xsd	Sollstatistik	Erstellung der Sollstatistik	
DAS/VST interface_DAS_VST	interface_DAS_VST	PID-Verfahren	DAS	VST
DAS/BAS interface_DAS_BAS	interface_DAS_BAS.xsd	IV-Verfahren	LQS	BAS
	Interface_DAS_BAS_soll.xsd	Sollstatistik	DAS	BAS
VST/BAS interface_VST_BAS	interface_VST_BAS.xsd	PID-Verfahren	VST	BAS

Um nach einer Schemavalidierung der XML-Dateien die Weiterverarbeitung und dementsprechend die spezifikationskonforme Protokollierung auf Datensatzebene weiterhin zu ermöglichen, wurden neben der oben in Tabelle 4 beschriebenen Schemavariante ein „weiches“ Schema für die Schnittstellen LE und DAS eingeführt. Diese weiche Variante wird ausschließlich mit dem Datenprüfprogramm verwendet. Es ist dabei zu beachten, dass nur die DAS, die die QS-Daten entschlüsseln, diese weiche Variante benötigen. Diese sind LQS/LKG und BAS.

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

Tabelle 5: Weiche Schemavarianten für das Datenprüfprogramm<sup>15</sup>

Schnittstelle	Schema	Verwendungszweck in Zusammenhang mit dem DPP
LE interface_LE_WEICH	2016_bas_dv_1.0_Export	Validierung der direkten Module mit dem Datenprüfprogramm (DPP)
	2016_kv_pid_1.0_Export	Validierung der Module für die KV mit dem Datenprüfprogramm (DPP)
	2016_lqs_iv_1.0_Export	Validierung der indirekten Module für die LQS mit dem DPP
DAS interface_DAS_WEICH	2016_lqs_pid_1.0_Export	Validierung der PID-Verfahren <sup>16</sup> für die LQS mit dem DPP
	2016_sv_pid_1.0_Export	Validierung der Module für die DAS-SV mit dem Datenprüfprogramm (DPP)
	interface_LQS_iv	Validierung der indirekten Module mit dem DPP (nach der Entschlüsselung der QS-Daten)
	interface_LQS_pid	Validierung der PID-Verfahren mit dem DPP (nach der Entschlüsselung der QS-Daten)

In der folgenden Tabelle werden Schema-Dateien aufgeführt, die im Rahmen der Rückprotokollierung Verwendung finden.

Tabelle 6: XML-Schemata für die Rückprotokollierung

Schnittstelle	Schema	Verwendungszweck in Zusammenhang mit dem DPP
LE/DAS interface_LE_DAS	response_DAS_LE.xsd	Datenflussprotokoll der DAS
	response_receipt.xsd	Empfangsbestätigung der DAS
DAS/VST interface_DAS_VST	response_VST_DAS.xsd	Datenflussprotokoll der VST
	response_receipt.xsd	Empfangsbestätigung der VST
DAS/BAS interface_DAS_BAS	response_BAS_DAS.xsd	Datenflussprotokoll der BAS
VST/BAS interface_VST_BAS	response_BAS_VST.xsd	Datenflussprotokoll von der BAS an die VST
	response_receipt.xsd	Empfangsbestätigung


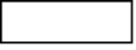
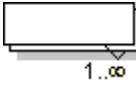

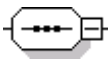


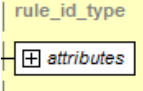
<sup>15</sup> Die Zielgruppe ist aus der Schnittstelle zu entnehmen.

<sup>16</sup> Follow-up-Verfahren mit PID.

### 3.4.4 Darstellung der XML-Struktur

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

Tabelle 7: Symbole in den XML-Schema-Diagrammen

Symbol	Beschreibung
	<b>Optionales Element</b> Kardinalität 0..1 („0 oder 1“)
	<b>Obligatorisches Element</b> Kardinalität 1: das Element muss genau einmal vorkommen
	<b>Mehrfach wiederholbares Element</b> Kardinalität: die erlaubte Anzahl der Elemente wird unter dem Symbol dargestellt (Beispiel: 1..n, n..m).
	<b>Referenzelement</b> Das referenzierte globale Element ist an anderer Stelle im Schema definiert.
	<b>Eine Folge von Elementen</b> Die Elemente müssen genau in der Reihenfolge vorkommen, in der sie im Schemadiagramm angezeigt sind.
	<b>Eine Auswahl von Elementen</b> Nur ein einziges Element aus der Liste kann ausgewählt werden.
	Ein Element mit Kind-Elementen
	<b>Komplexer Datentyp</b> Der komplexe Datentyp wird mit einem Rahmen mit einem gelben Hintergrund angezeigt.

Die wesentlichen Bestandteile der XML-Schemata werden in den nachfolgenden Abschnitten dargestellt. Die Darstellung umfasst folgende Eigenschaften des betrachteten Elements:

- Grafische Abbildung der Kind-Elemente und -Attribute
- Auflistung der Kind-Elemente
- Auflistung der Kind-Attribute sowie ihre Eigenschaften wie:
  - Name
  - XML-Datentyp (technische Bezeichnung: „Type“)
  - Muss-Kann-Feld (technische Bezeichnung: „Use“)
  - Konstante (technische Bezeichnung: „Fixed“)
  - Kurze Beschreibung (technische Bezeichnung: „Annotation“)

### 3.4.5 Grundstruktur der XML-Dateien

Grundsätzlich beginnt jede XML-Exportdatei mit einer Headerzeile gefolgt vom Wurzelement `<root>`, das den gesamten Inhalt einschließt. Als Zeichensatz wird „UTF-8“ (Unicode-Codierung) verwendet.

**Beispiel:**

Headerzeile

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

### 3.4.6 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>`.

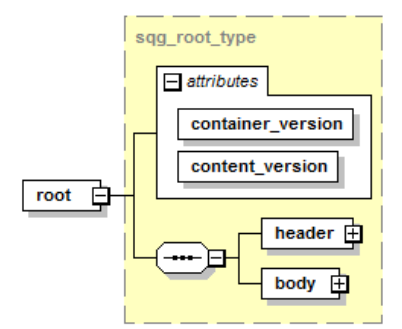


Abbildung 5: Root-Element und Kind-Elemente header und body

Das Root-Element hat zusätzlich zwei Attribute:

Tabelle 8: Root-Element Attribute

Name	Type	Use	Fixed	Beschreibung
container_version	xs:string	required	2.0	<p>Ist ein fixer Wert und definiert die aktuell gültige Versionsnummer des Containers.</p> <p>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.</p> <p>Ein XML-Dokument, das einen alten Wert dieses Attributs enthält, muss von der Datenannahmestelle zurückgewiesen werden.</p>

Name	Type	Use	Fixed	Beschreibung
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 unabhängig von der zugrundeliegenden Spezifikationsdatenbank geändert wird.

### 3.4.7 Header-Bereich

#### Element header

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

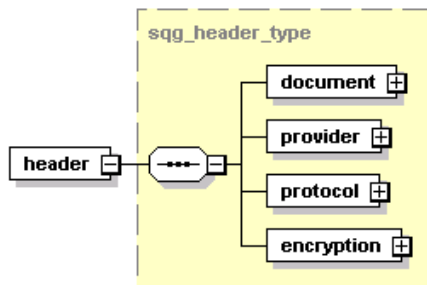


Abbildung 6: Aufbau des Elements header

#### 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.

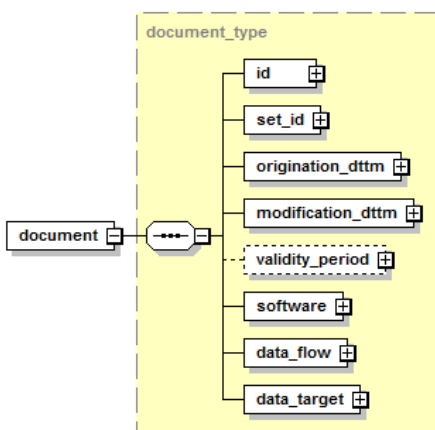
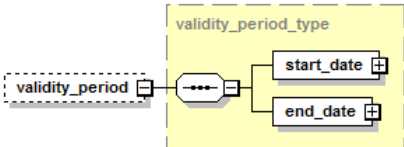


Abbildung 7: Aufbau des Elements document

Dieses Element hat folgende Kind-Elemente:

Tabelle 9: Kind-Elemente des Elements document

Kind-Elemente	Beschreibung
<id>	Nach Erstellung nicht mehr modifizierbar. Eindeutige ID des Dokuments, wird vom Dokumentenersteller als GUID erzeugt. <sup>17</sup>
<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.
<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 &lt;validity_period&gt;-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 &lt;start_date&gt; und &lt;end_date&gt; 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 Universal Time Coordinated (UTC) handelt. Folgt der Zeitangabe statt eines „Z“ ein Plus- oder Minuszeichen, gibt die darauf folgende Angabe im Format „hh:mm“ die Differenz zur UTC an (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

<sup>17</sup> Ein Globally Unique Identifier oder kurz GUID ist eine global eindeutige Zahl mit 128 Bit (16 Bytes), die ein Dokument weltweit eindeutig identifiziert.

### Element header/document/software

Sammelement für Angaben zur eingesetzten QS-Software.

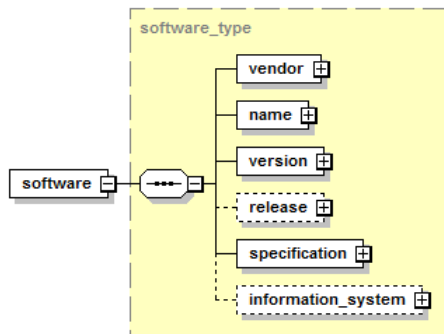


Abbildung 8: Aufbau des Elements software

Dieses Element enthält Kind-Elemente, die in der folgenden Tabelle beschrieben werden.

Tabelle 10: Kind-Elemente des Elements software

Kind-Elemente	Beschreibung
<vendor>	Enthält Informationen über den Softwarehersteller.
<name>	Enthält den Namen der eingesetzten Software.
<version>	Enthält die Version der eingesetzten Software.
<release>	Enthält die Releasenummer der eingesetzten Software.
<specification>	Enthält einen Wert aus der enumeration enum_spez_type: 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)

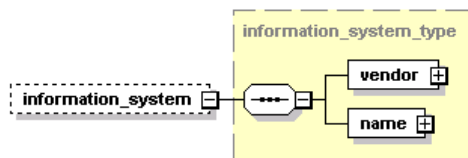


Abbildung 9: Aufbau des Elements information\_system

Dieses Element enthält Kind-Elemente, die in der folgenden Tabelle beschrieben werden.

Tabelle 11: Kind-Elemente des Elements information\_system

Kind-Elemente	Beschreibung
<vendor>	Enthält Informationen über den Softwarehersteller.
<name>	Enthält den Namen der eingesetzten Software.



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

### Element header/document/data\_flow

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

Tabelle 12: Angabe des betreffenden Datenflusses

Datenfluss	Beschreibung
QS-Bundesbezogen	für direkte Verfahren, die direkt vom LE an die BAS übermittelt werden müssen
QS-Laenderbezogen	für indirekte Verfahren(mit und ohne PID), die an die Landesgeschäftsstellen (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 (DAS-SV) 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.

Tabelle 13: Angabe des betreffenden Datenfluss-Ziels

Datenfluss	Ziel
Echtdatenpool	Echtdaten für den Echtbetrieb
Probedatenpool	Echtdaten für vorläufige Auswertungen wie der Sonderexport
Testdatenpool	Testdaten für Testzwecke



#### **Achtung Datenverlust**

Nur Daten mit der Kennzeichnung „Echtdatenpool“ werden für die Erstellung der Bundes-, Landes- und Rückmeldeberichte berücksichtigt.

## 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.

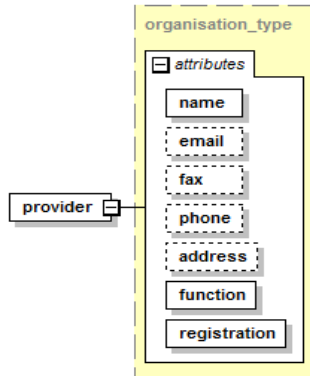


Abbildung 10: Aufbau des Elements provider

Tabelle 14: Attribute des Elements header/provider

Name	Type	Use	Beschreibung
name	xs:string	required	Name der Institution
email	emailAddress_type	optional	E-Mail-Adresse
fax	xs:string	optional	Faxnummer
phone	xs:string	optional	Telefonnummer
address	xs:string	optional	Adresse
function	enum_organisation_type	required	Bundesauswertestelle/ Datenannahmestelle Vertrauensstelle/ Softwarehersteller/undefined
registration	registration_type	required	Registrierungsnummer. Sollte für die Datenübertragung an die DAS (KV/ DAS-SV) keine Registrie- rungsnummer erforderlich, ist das Ele- ment mit einem Dummy-Wert „000“ zu füllen.

### 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.

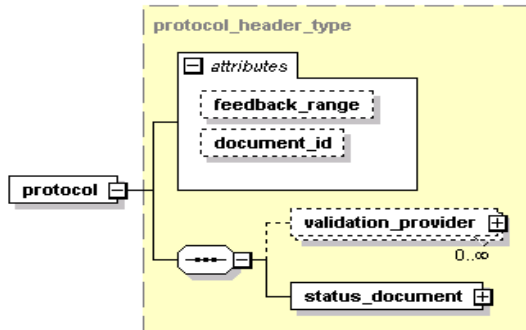


Abbildung 11: Aufbau des Elements header/protocol

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

`<validation_provider>` und `<status_document>`

Tabelle 15: Attribute des Elements header/protocol

Name	Type	Use	Beschreibung
feedback_range		optional	Da die Transaktionsprotokolle durch die Empfangsbestätigungen ersetzt wurden, ist nur der Wert „dataflow“ zu verwenden.
document_id		optional	Soweit die GUID des Exportdokuments lesbar ist, muss sie in das Attribut document_id 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.

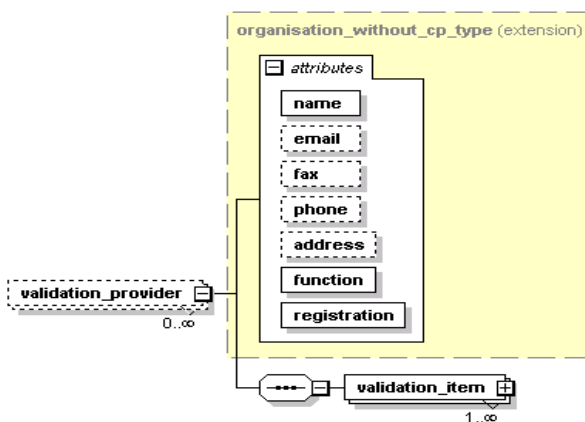


Abbildung 12: Aufbau und Kind-Elemente des Elements validation\_provider

Tabelle 16: 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 Datensatzes oder des Dokumentes ausschließen)

Das Ergebnis der Prüfung wird in das Attribut `@V` des Elements `<status>` im Element `<validation_item>` eingetragen.

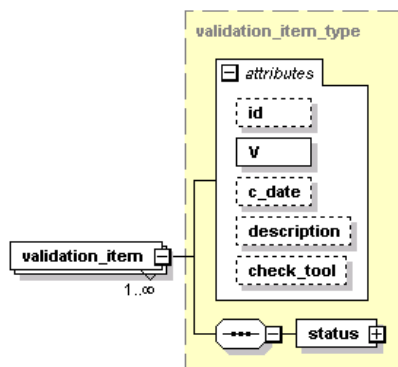
Abbildung 13: Aufbau und Kind-Elemente des Elements *validation\_item*

Tabelle 17: 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.
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 (z.B. das Datenprüfprogramm)

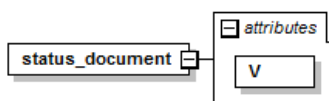
#### Attribut `/protocol/validation_provider/validation_item/@check_tool`

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 14: Aufbau des Elements *status\_document*Tabelle 18: 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.

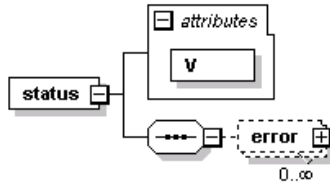


Abbildung 15: Aufbau und Kind-Elemente des Elements status

Tabelle 19: Attribut des Elements status

Name	Type	Use	Beschreibung
v	status_type	required	Status einer Prüfung mit folgenden, möglichen Werten: OK, WARNING oder ERROR .

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:

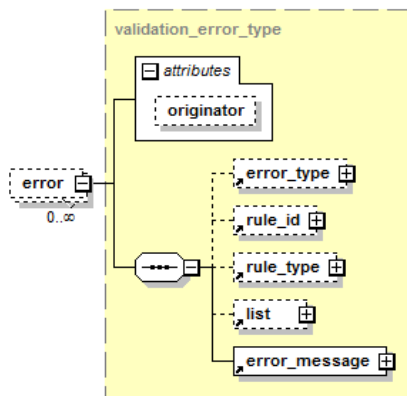


Abbildung 16: Aufbau des Elements error

Tabelle 20: Attribut des Elements error

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

Tabelle 21: Kind-Elemente des Elements error

Kind-Elemente	Beschreibung
<rule_id>	Nummer der Regel (idRegeln in Tabelle Regeln), oder Nummer der Fehlermeldung aus der Tabelle Fehlermeldung (idFehlermeldung)
<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
<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 Alters-/Geschlechtsangaben 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 (XPacker).

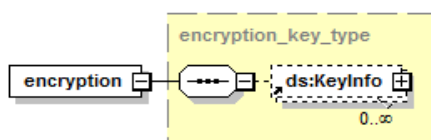


Abbildung 17: Aufbau und Attribute des Elements encryption

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_data`) chiffriert.
- Der Schlüssel wird nun mit dem „public Key“ des Empfängers (z.B. LQS/LKG) 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.

### Element header/encryption/feedback\_key

Gemäß der Qesü-RL erstellt die BAS Rückmeldeberichte (RB) für die LE. Diese RB werden von der BAS über die DAS an die jeweiligen LE geleitet. Dabei ist sicherzustellen, dass die DAS keine Möglichkeit zur Einsicht in die Berichte hat. Ausdrücklich ausgenommen von diesem Ausschluss der Möglichkeit der Einsichtnahme sind allein die Landesgeschäftsstellen für Qualitätssicherung (LQS) bzw. die Landeskrankenhausgesellschaften (LKG).

Selektiv- und kollektivvertraglich tätige LE müssen daher, damit sie den RB von der BAS erhalten können, eine Zusatzinformation an die BAS liefern. In den Header der QS-Datenlieferung ist ein mit dem öffentlichen Schlüssel der BAS verschlüsseltes Passwort zu übermitteln. Dieses Passwort wird von der BAS entschlüsselt und später wiederum zur Verschlüsselung des RB verwendet. Auf diese Weise ist sichergestellt, dass nur die BAS und der jeweilige LE Einblick in den RB erhalten.

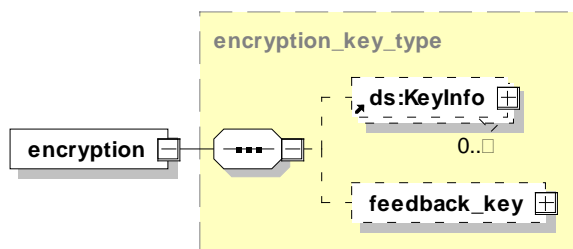


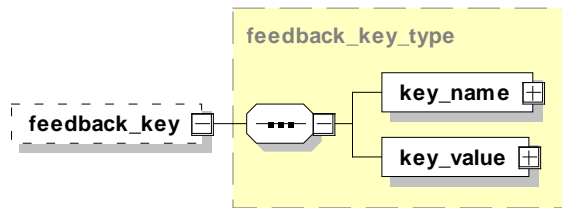
Abbildung 18: Aufbau und Attribute des Elements `encryption` (AP-selektiv/-kollektiv)

Die Kind-Elemente werden in folgender Tabelle beschrieben.

Tabelle 22: `feedback_key`: Kind-Elemente

Kind-Elemente	Beschreibung
<code>&lt;key_name&gt;</code>	Enthält den Namen des Berichts (hier Rückmeldebericht)
<code>&lt;key_value&gt;</code>	Hier wird das erzeugte verschlüsselte Passwort aufgenommen



Abbildung 19: Aufbau des Elements `feedback_key`

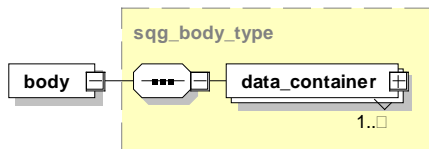
### Mehrfache Datenlieferung

Übermittelt der LE eine Datenlieferung mehr als einmal, hat er dafür zu sorgen, dass in jeder Datenlieferung das verschlüsselte Passwort enthalten ist.

Die BAS verschlüsselt den Bericht mit dem zuletzt an sie gesendeten Passwort. Es ist dabei zu beachten, dass das Passwort verfahrensbezogen generiert werden muss.

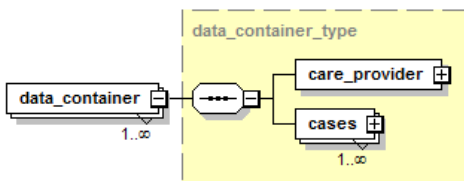
### 3.4.8 Body-Bereich

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 20: 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. einer Datenannahmestelle) erstellt wird, die Daten mehrerer Leistungserbringer sammelt. Dies ist der Fall bei der Erstellung der Sollstatistik durch die KV und die DAS-SV.

Abbildung 21: 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 (Krankenhaus oder selektiv-/kollektivvertraglich) durch einen unterschiedlichen Aufbau.

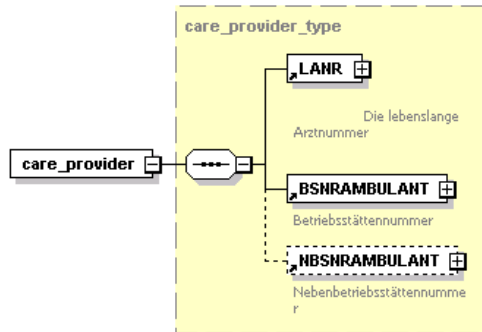


Abbildung 22: Aufbau des Elements `care_provider` – ambulante

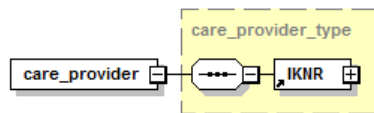


Abbildung 23: Aufbau des Elements `care_provider` – stationäre

Die Kind-Elemente für die kollektiven, selektiven oder stationären (Krankenhaus) Bereiche werden in der folgenden Tabelle beschrieben.

Tabelle 23: Leistungserbringeridentifizierende Daten im ambulanten Bereich

Kind-Elemente	Beschreibung
<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 ambulante – 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 24: Leistungserbringeridentifizierende Daten im stationären Bereich (Krankenhaus)

Kind-Element	Beschreibung
<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 Primärmoduls. Das Element enthält einen oder mehrere Vorgänge<sup>18</sup>.

Für unterschiedliche Module müssen jeweils mehrere `<cases>` angelegt werden. Die Ausweisung eines `<cases>`-Elements für Daten eines bestimmten Primärmoduls erfolgt über dessen Attribut `module`.

Die Minimaldatensätze sind unbedingt im korrespondierenden `<cases>`-Element des ausgelösten Primärmoduls unterzubringen. MDS gelten nicht als unabhängige Module, sondern sind Ersatzmodule eines jeweiligen Primärmoduls.

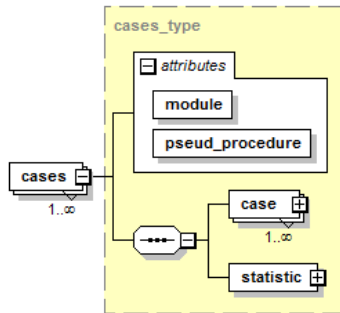


Abbildung 24: Aufbau des Elements `cases`

Tabelle 25: Attribute des Elements `cases`

Name	Type	Use	Beschreibung
<code>module</code>	<code>enum_module_type</code>	required	Das Erfassungsmodul
<code>pseud_procedure</code>	<code>enum_procedure_type</code>	required	Zuordnung des Moduls zu einem Pseudonymisierungsverfahren. Ist dem Modul kein Pseudonymisierungsverfahren zugeordnet, ist das Attribut auf „undefined“ zu setzen.

Das Element `<cases>` enthält das 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.

<sup>18</sup> Fälle und Vorgänge werden hier als Synonyme verwendet.

Tabelle 26: Verfahrenskennung: „pseud\_procedure“

Betrieb	Exportmodul	Verfahrenskennung	XML (Umsetzung)
QSKH 2015	HEP	HEP	<cases module="HEP" pseud_procedure="HEP">
	KEP	KEP	<cases module="KEP" pseud_procedure="KEP">
	09/1 09/2 09/3	09/1_09/2_09/3	<cases module="09/1" pseud_procedure="09/1_09/2_09/3"> <cases module="09/2" pseud_procedure="09/1_09/2_09/3"> <cases module="09/3" pseud_procedure="09/1_09/2_09/3">
Qesü 2016	PCI_KV PCI_LKG PCI_SV	PCI	<cases module="PCI_KV" pseud_procedure="PCI"> <cases module="PCI_LKG" pseud_procedure="PCI"> <cases module="PCI_SV" pseud_procedure="PCI">
Regelbetrieb Testbetrieb	Nicht-PID	undefined	<cases module="PNEU" pseud_procedure="undefined">

### 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 <patient>.



#### Achtung Datenexport GEKID

Das Element <patient> ist nicht relevant. Die Felder VERSICHERTENIDNEU und KASSEIKNR sind im Element <qs\_data> im Bogen B zu exportieren.

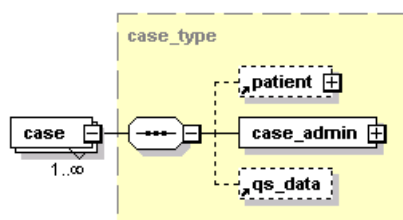


Abbildung 25: Aufbau des Elements case

## 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 (Datensatzebene) werden von jeder Prüfstelle der Status der Prüfung und ggf. die Fehler in das Element `<protocol>` eingetragen.

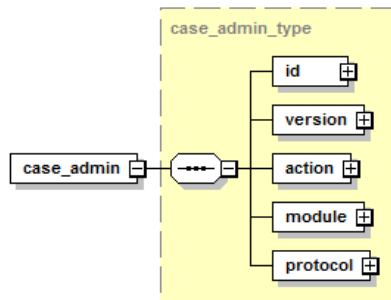


Abbildung 26: Aufbau des Elements case\_admin

Im Folgenden werden die einzelnen Kind-Elemente beschrieben:

Tabelle 27: Kind-Elemente des Elements case\_admin

Kind-Elemente	Beschreibung
<code>&lt;id&gt;</code>	<p>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 Krankenhausaufenthalts zwei QS-Dokumentationen eines Falles 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 Datenannahmestelle nicht auf Personen zu beziehen sein.</p> <p>Da im ambulanten Bereich keine Registrierung der Softwareinstanzen vorgesehen ist, muss bezogen auf einen Leistungserbringer (BSNRAMBULANT) die Vorgangsnummer immer eindeutig sein. Hierfür ist die Vorgangsnummer als GUID von der QS-Software zu generieren.</p>
<code>&lt;version&gt;</code>	<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 1 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 Datenannahmestelle einen Datensatz mit derselben Versionsnummer ein zweites Mal erhält, so wird dieser zurückgewiesen.</p>
<code>&lt;module&gt;</code>	<p>Verfahrensabkürzung. Hier ist zu beachten, dass dieser Wert identisch zu den Attributwerten im Element <code>&lt;cases&gt;</code> und <code>&lt;qs_data&gt;</code> sein muss.</p> <p>Wenn die Datenannahmestelle unterschiedliche Modulbezeichnungen innerhalb eines <code>&lt;cases&gt;</code>-Elements erhält, wird die ganze Datenlieferung zurückgewiesen.</p>

Kind-Elemente	Beschreibung
<protocol>	Protokoll auf Vorgangsebene
<action>	<p>Definiert die gewünschte Aktion, kann „create“, „update“ und „delete“ sein. „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 &lt;action&gt; auf „delete“ gesetzt werden. Die Datenannahmestelle 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 case\_Datentyp/case\_admin/protocol

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

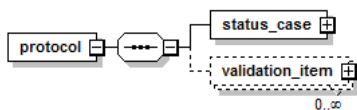


Abbildung 27: Aufbau des Elements case\_admin/protocol

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.

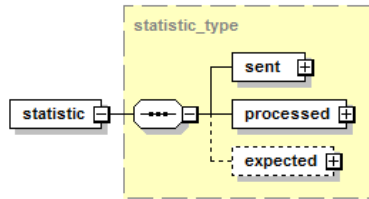


Abbildung 28: Aufbau des Elements statistic

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

Tabelle 28: Kind-Elemente des Elements `<statistic>`

Kind-Elemente	Beschreibung
<code>&lt;sent&gt;</code>	Statistik über die von dem Datenlieferant exportierten Datensätze. Es muss daher vom Datenlieferant selbst ausgefüllt werden.
<code>&lt;processed&gt;</code>	Hat dieselbe Struktur wie <code>&lt;sent&gt;</code> und enthält das Ergebnis der Verarbeitung durch die datenentgegennehmende Stelle.
<code>&lt;expected&gt;</code>	Das optionale Element hat noch keine Anwendung.

### Element statistic/sent

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

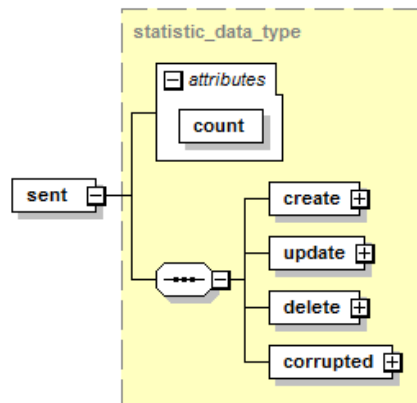


Abbildung 29: Aufbau des Elements sent

Es besteht aus vier Elementen und einem Attribut:

Tabelle 29: Attribut des Elements sent

Name	Type	Use	Beschreibung
count	non_negative_integer_type	required	Gesamtzahl von Vorgängen (Ganze Zahl $\geq 0$ ): Summe von ( <code>&lt;create&gt;</code> + <code>&lt;update&gt;</code> + <code>&lt;delete&gt;</code> + <code>&lt;corrupted&gt;</code> )

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

Kind-Elemente	Beschreibung
<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/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.

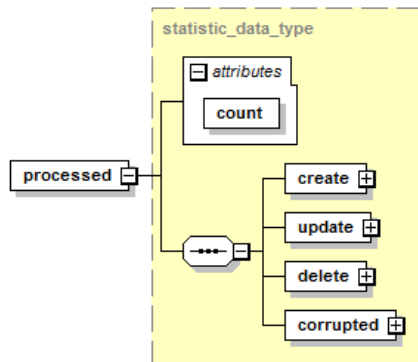


Abbildung 30: Aufbau des Elements *processed*

<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 31: Kind-Elemente des Elements *statistic/processed*

Kind-Elemente	Beschreibung
<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 Zielpool 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.

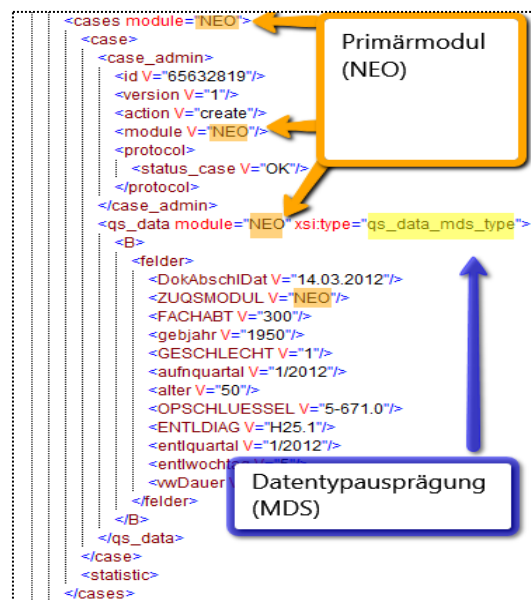


Abbildung 31: Integration des MDS in das Primärmodul

**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.
- Die Reihenfolge der Kinderbögen orientiert sich an der Spezifikationsdatenbank.
- 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:

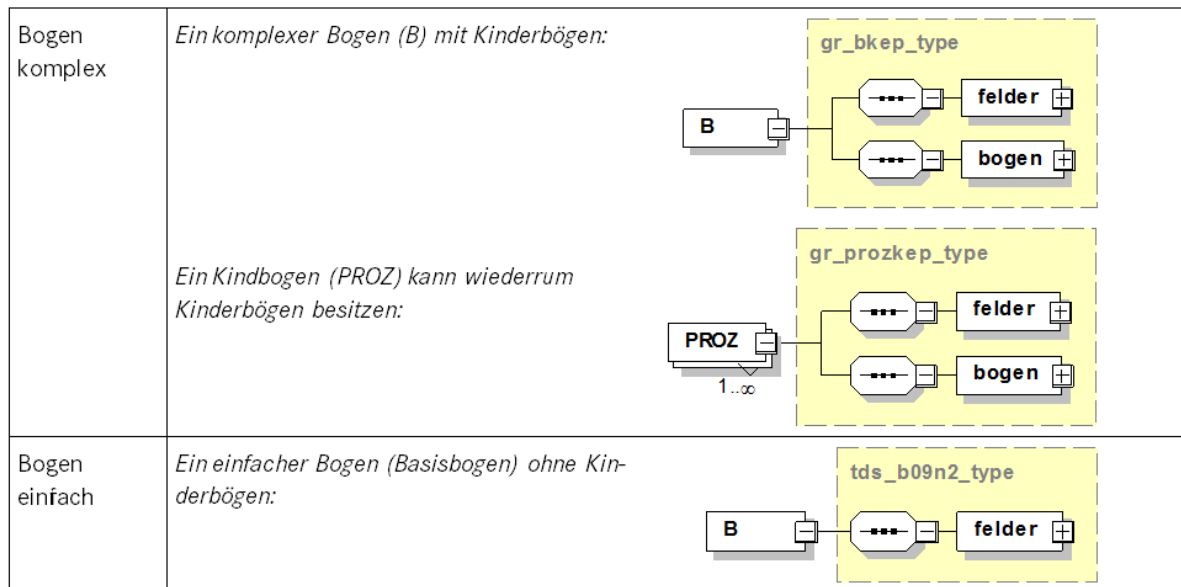


Abbildung 32: Diagramme „Bogen komplex“ und „Bogen einfach“

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

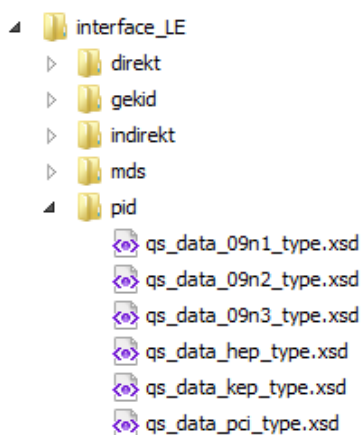


Abbildung 33: Ausprägungen des qs\_data-Elements (Erfassungsmodule)