Bundesverband der Deutschen Volksbanken und Raiffeisenbanken e. V.
Bundesverband deutscher Banken e. V.
Bundesverband Öffentlicher Banken Deutschlands e. V.
Deutscher Sparkassen- und Giroverband e. V.
Verband deutscher Pfandbriefbanken e. V.

Die Deutsche Kreditwirtschaft

FinTS

Financial Transaction Services

Schnittstellenspezifikation

XML Syntax

Herausgeber:
Bundesverband deutscher Banken e.V., Berlin
Deutscher Sparkassen- und Giroverband e.V., Bonn/Berlin
Bundesverband der Deutschen Volksbanken und Raiffeisenbanken e.V., Berlin
Bundesverband Öffentlicher Banken Deutschlands e.V., Berlin

Version: 4.1 FV Stand: 20.01.2014 Final Version

Die vorliegende Schnittstellenspezifikation für eine automatisiert nutzbare multibankfähige Banking-Schnittstelle (im Folgenden: Schnittstellenspezifikation) wurde im Auftrag der Deutschen Kreditwirtschaft entwickelt. Sie wird hiermit zur Implementation in Kunden- und Kreditinstitutssysteme freigegeben.

Dieses Dokument kann im Internet abgerufen werden unter http://www.fints.org.

Versionsführung

Das vorliegende Dokument wurde von folgenden Personen erstellt bzw. geändert:

Name	Organi-sation	Datum	Version	Dokumente	Anmerkungen
	SIZ	02.04.2004	4.0 Final Version	FinTS_4.0_XML-Syntax.doc	
Haubner	für SIZ	20.01.2014	4.1 Final Version	FinTS_4.1_XML-Syntax_2014-01-20_FV.docx	

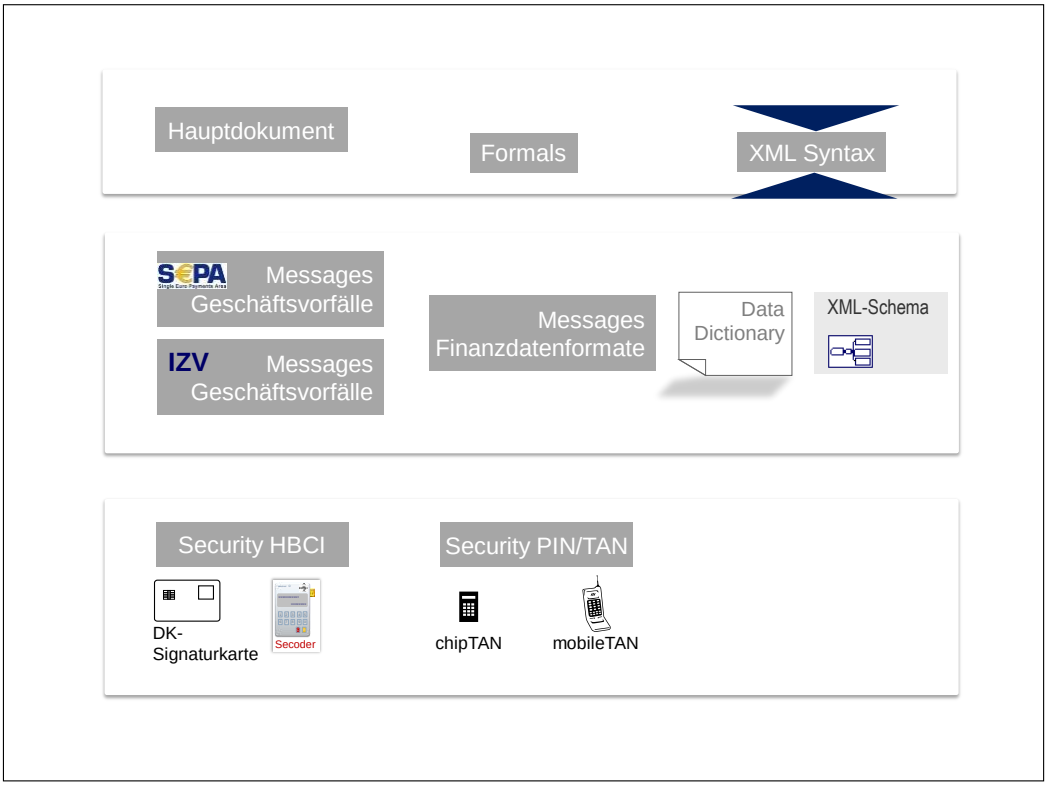
Änderungen gegenüber der Vorversion:

Änderungen sind im Dokument durch einen Randbalken markiert. Falls sich die Kapitelnummerierung geändert hat, bezieht sich die Kapitelangabe auf die neue Nummerierung.

lfd. Nr.	Kapitel	Kapitelnummer	Kennung[1]	Art[2]	Beschreibung
1					

Dokumentenstruktur

Das vorliegende Dokument steht in folgendem Bezug zu den anderen Bänden der FinTS-Spezifikation:



Dokumenteninhalte, Abkürzungen, Definitionen und Literaturhinweise befinden sich im FinTS Hauptdokument [Master].

---

1 nur zur internen Zuordnung

2 F = Fehler; Ä = Änderung; K = Klarstellung; E = Erweiterung

Inhaltsverzeichnis

Abbildungsverzeichnis

I. EINLEITUNG

Dieses Dokument beschreibt im ersten Teil „Nachrichtensyntax“ die allgemeinen Anforderungen an das XML-Format von FinTS-Nachrichten. Die Anforderungen sind Einschränkungen in der Benutzung der XML-Standards, die in FinTS angewendet werden. Hier wird FinTS mit den folgenden XML-Standards in Beziehung gesetzt:

- mit dem XML-Basisstandard ([XML1.0]),
- der Verwendung von Namensräumen ([Namespaces]),
- der Validierung durch XML-Schemas ([XML Schema 1], [XML Schema 2]) und
- der Referenzierung von Nachrichtenelementen mit XPath-Ausdrücken ([XPath]).

Im zweiten Teil „Nachrichtenaufbau“ werden die XML-Datenformate der FinTS-Nachrichten in detaillierter Form dokumentiert. In diesem Abschnitt geht es um anwendungsunabhängige XML-Konventionen für FinTS. Diese Konventionen gelten auch für verbandsspezifische Erweiterungen von FinTS.

Grafiken stellen die Struktur der zugrunde liegenden FinTS-Schemas zum allgemeinen Nachrichtenaufbau und zu den administrativen Geschäftsvorfällen dar. Im Text sind zu den einzelnen Elementen Belegungsrichtlinien angegeben, sofern sie über die im [DataDictionary] vorhandenen Erläuterungen hinausgehen.

Die Abschnitte drei und vier befassen sich mit den Sicherheitsmechanismen von XML. Dazu gehören

- das Signieren ([XML Signature]),
- die Referenzierung mit XPath-Ausdrücken in einer Signatur ([XPath Filter]),
- die Kanonisierung ([Canonical XML], [Exclusive Canonical XML]) und
- das Verschlüsseln ([XML Encryption])

von FinTS-Nachrichten.

Die Komprimierung von FinTS-Nachrichtenteilen wird als eine spezielle Form der Verschlüsselung betrachtet. Für beide Verfahren wird der [XML Encryption]-Standard angewendet.

Der letzte Abschnitt beschreibt den Einsatz von Webservices im Zusammenhang mit FinTS. Die referenzierten Webservice Standards des W3C werden

- als Übertragungsprotokoll ([SOAP 1], [SOAP 2])
- und zur Definition der FinTS-Webservices ([WSDL 1], [WSDL 2])

eingesetzt. Für den Einsatz beider Standards im Zusammenhang mit FinTS werden Beispiele gezeigt.

Alle vorangehend genannten XML-Standards wurden vom W3C (World Wide Web Consortium) normiert. Die üblicherweise von XML-Implementierungen unterstützten Sicherheitsverfahren des W3C werden allerdings durch die [HBCI]-Sicherheitsverfahren erweitert.

II. NACHRICHTENSYNTAX

II.1 XML-Schemas

Die FinTS-Nachrichten werden in XML-Syntax anhand von XML-Schemas spezifiziert.



Die syntaktische Prüfung und die Schema-Validierung einer FinTS-Nachricht erfolgt durch einen XML-Parser. Der validierende XML-Parser muss XML-Schemas gemäß [XML Schema 1] bzw. [XML Schema 2] unterstützen.

Whitespace in FinTS-Nachrichten ist signifikant.



Bei XML-Parsern kann die Behandlung von Whitespace üblicherweise konfiguriert werden. Die XML-Parser und alle Komponenten einer Anwendung müssen FinTS-Nachrichten so verarbeiten, dass Whitespace erhalten bleibt.

II.2 Zeichensatz und Kodierung

Der Zeichensatz zur Verarbeitung eines XML-Dokuments wird durch das *encoding*-Attribut der XML-Verarbeitungsanweisung (processing instruction) festgelegt, die in der ersten Zeile eines XML-Dokuments steht.

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

Die Verarbeitungsanweisung muss sich in allen FinTS-Dokumenten und allen zugehörigen Schema-Dokumenten befinden, um eine ordnungsgemäße Verarbeitung zu gewährleisten. Alle XML-Dokumente, die FinTS-Nachrichten enthalten, müssen im Unicode-Format [UTF-8] kodiert werden. FinTS-Nachrichten in anderen Kodierungen sind nicht zulässig.

Auch die Entwicklungswerkzeuge und XML-basierte Anwendungen für FinTS-Nachrichten (z. B. XSLT-Stylesheets) müssen die Kodierung [UTF-8] verwenden, um Fehler bei der Abbildung zwischen verschiedenen Zeichensätzen zu vermeiden.

Für FinTS-Nachrichten, die in XML serialisiert werden, wird der in XML übliche Kodierungsmechanismus verwendet. In XML müssen die für die Auszeichnung verwendeten Zeichen &, <, >, ’ und “ kodiert werden. Die Zeichen werden in XML mit den sogenannten allgemeinen Entitäten (entities) *&amp*;, *<*, *>*, *'* und *"* dargestellt. Dies sind vordefinierte Zeichen-Entitäten (character entities), die nicht in einem Schema deklariert werden müssen.

Beispiel:

vor Entwertung (kein wohlgeformtes XML):

```
Betrag > EUR 1000
```

nach Entwertung (wohlgeformtes XML):

```
Betrag &gt; EUR 1000
```

II.3 Namensräume und Schema-Module

Namensräume ([Namespaces]) werden zur Vermeidung von Namenskonflikten in XML-Dokumenten eingesetzt. Darüber hinaus werden sie in den XML-Schemas zur Validierung der XML-Dokumente verwendet, um eindeutige Deklaration von Elementen und eindeutige Definitionen von Typen zu gewährleisten.

In FinTS legen Namensräume die Ebene (Nachrichten- oder Transaktionsebene) eines Bezeichners in einer FinTS-Nachricht fest. Außerdem erleichtern die FinTS-Namensräume die Wiederverwendung von Schemakomponenten außerhalb von FinTS.

II.3.1 Aufbau des Namensraum-URIs

Die Namensräume werden entsprechend dem W3C-Standard [Namespaces] durch die Zuordnung eines URI zu einem Namensraum-Präfix bzw. durch die Zuordnung des URI zum voreingestellten Namensraum (default namespace) definiert.

Die URIs für die DK-weit gültigen Namensräume von FinTS besitzen den folgenden Aufbau:

```
<Domain>/spec/<Schematyp>/<Version>/<Modulverzeichnis>
```

- Die Domain ist der feste Wert *http://www.fints.org/*
- Der Schematyp ist *xmlschema* für XML-Schemas (andere Schematypen werden derzeit nicht unterstützt).
- Die Versionsnummer entspricht der FinTS-Version.

Beispiele:

```
http://www.fints.org/spec/xmlschema/4.1/transactions
http://www.fints.org/spec/xmlschema/4.1/admintransactions
```

II.3.2 Zuordnung der XML-Schemas zu den Namensräumen

In FinTS werden vier Namensräume verwendet.

- FinTS-Typen (Modulverzeichnis: *types*)
- FinTS-Nachrichten (Modulverzeichnis: *messages*)
- FinTS-Administrative-Geschäftsvorfälle (Modulverzeichnis: *admintransactions*)
- FinTS-Geschäftsvorfälle (Modulverzeichnis: *transactions*)

Dem Namensraum für allgemeine Strukturen (FinTS-Typen) werden alle Schemas zugeordnet, die wiederverwendbare Komponenten enthalten.

Beispiel:

```
<xs:schema
 targetNamespace="http://www.fints.org/spec/xmlschema/4.1/types"
  ...
```

Die zugehörigen XML-Schemas sind gemäß ihrer Semantik in mehrere Module unterteilt:

Schemadatei	Inhalt
common.xsd	allgemeine abstrakte Basistypen und häufig verwendete Komponenten
formats.xsd	Basisformate und abgeleitete Formate
patterns.xsd	häufig verwendete bankfachliche Typen

Dem Namensraum für den allgemeinen Nachrichtenaufbau (FinTS-Nachrichten) werden alle Schemas zugeordnet, die den Aufbau einer FinTS-Nachricht beschreiben.

Beispiel:

```
<xs:schema
 targetNamespace="http://www.fints.org/spec/xmlschema/4.1/messages"
  ...
```

Die zugehörigen XML-Schemas beschreiben den Aufbau der Kunden- und Kreditinstitutsnachrichten sowie der Parametersegmente:

Schemadatei	Inhalt
message.xsd	Grundgerüst des Nachrichtenaufbaus
paramdata.xsd	Typdefinitionen für Bank- und User-Parameterdaten
structures.xsd	Definitionen und Deklarationen für administrative Nachrichtentypen und -komponenten

Die Bestandteile zur Definition von administrativen Geschäftsvorfällen (Benutzeraufträge, Kreditinstitutsrückmeldungen und Bankparameterdaten) sind dem entsprechenden Namensraum zugeordnet:

Beispiel:

```
<xs:schema
 targetNamespace="http://www.fints.org/spec/xmlschema/4.1/admintransactions"
  ...
```

Die folgende Liste zeigt die Schema-Module der administrativen Geschäftsvorfälle in FinTS:

Schemadatei	Inhalt
ActivateMobilePhoneConnection-2.xsd	Mobilfunkverbindung freigeben
AddReg-1.xsd	Adressenregistrierung
AddRegDelete-1.xsd	Adressenregistrierung löschen
AddRegInfo-1.xsd	Adressenregistrierungsinformationen
BankParamData-1.xsd	Bankparameterdaten anfordern
BlockPIN-1.xsd	PIN sperren
ChangePIN-1.xsd	PIN ändern
ChangeMobilePhoneConnection-2.xsd	Mobilfunkverbindung ändern
ChangeTANGenerator-2.xsd	TAN-Generator an- bzw. ummelden
DeleteTANMedium-1.xsd	Deaktivieren / Löschen von TAN-Medien
DisplayTANGeneratorList-4.xsd	Anzeigen der verfügbaren TAN-Medien
DistSigsDelete-1.xsd	Auftrag mit verteilten Signaturen löschen
DistSigsInfo-3.xsd	Informationen zu Auftrag mit verteilten Signaturen
DistSigsSign-2.xsd	Auftrag mit verteilten Signaturen signieren
DistSigsSubmit-2.xsd	Auftrag mit verteilten Signaturen einreichen
InterList-1.xsd	Liste der Intermediäre
InterSignOff-1.xsd	Beim Intermediär abmelden
InterSignOn-1.xsd	Beim Intermediär anmelden
Receipt-1.xsd	Quittung
RegisterMobilePhoneConnection-2.xsd	Mobilfunkverbindung registrieren
RevokePINBlock-1.xsd	PIN-Sperre aufheben
StatProt-1.xsd	Statusprotokoll
Subscription-1.xsd	Abonnement einreichen
SubscriptionDelete-1.xsd	Abonnement löschen
SubscriptionInfo-1.xsd	Abonnementsinformationen anfordern

SynchronizeTANGenerator-1.xsd	TAN-Generator Synchronisierung
TANListDisplay-1.xsd	TAN-Verbrauchsinformationen anfordern
UPDIEdit-1.xsd	UPDI ändern
UserParamData-1.xsd	User-Parameterdaten anfordern (UPD)

Die allgemeinen ZKA-weit definierten Geschäftsvorfälle finden sich in [Messages].

II.3.3 Verwendung der Namensraum-Präfixe

Bei der Verwendung von Namensräumen in XML-Dokumenten kann entweder der voreingestellte Namensraum belegt werden oder es können beliebige Präfixe definiert werden, die einen Namensraum repräsentieren.

In einer FinTS-Nachricht sollte der voreingestellte Namensraum immer mit dem Namensraum für den allgemeinen Nachrichtenaufbau (FinTS-Typen) belegt werden, weil sich dadurch die Schreibweise der Elemente verkürzt. Zur Belegung des voreingestellten Namensraumes wird dem Attribut *xmlns* der entsprechende Namensraum-URI zugewiesen.

Beispiel:

```
<ReqMsg
 xmlns="http://www.fints.org/spec/xmlschema/4.1/types">
  ...
</ReqMsg>
```

Zum Vergleich das Beispiel ohne die Verwendung eines voreingestellten Namensraumes:

```
<fintstype:ReqMsg
 xmlns:fintstype="http://www.fints.org/spec/xmlschema/4.1/types">
  ...
</fintstype:ReqMsg>
```

Hier wird das Präfix *fintstype* durch das Attribut *xmlns:fintstype* mit dem Namensraum-URI für FinTS-Typen belegt. Das Präfix muss in diesem Beispiel allen FinTS-Elementen vorangestellt werden, die den allgemeinen Nachrichtenaufbau beschreiben.

Um für FinTS eine eindeutige leicht verständliche Vergabe von Präfixen zu gewährleisten, wird empfohlen, dass als Präfix-Bezeichner

- *fintsmsg* für den allgemeinen Nachrichtenaufbau,
- *fintstype* für verwendete Strukturen
- *fintstrans* für Transaktionen

verwendet werden.


Da der voreingestellte Namensraum bereits mit dem URI für FinTS-Typen belegt ist, entfällt in XML-Dokumenten die Notwendigkeit der Verwendung des Präfixes *fintstype*. Dieses Präfix sollte ausschließlich in Schema-Modulen zur Definition von verbandsspezifischen Typen und Geschäftsvorfällen verwendet

werden (siehe *II.4 Verbandseigene Geschäftsvorfälle*).

Im folgenden Beispiel werden der voreingestellte Namensraum und das Präfix *fintstrans* für eine Einzelüberweisung (*SEPASingRemitt*) definiert:

```
<fintstrans:SEPASingRemitt 1 Req
 xmlns="http://www.fints.org/spec/xmlschema/4.1/types"
 xmlns:fintstrans="http://www.fints.org/spec/xmlschema/4.1/transactions">
  ...
</fintstrans:SEPASingRemitt_1_Req>
```

Das Element *SEPASingRemitt_1_Req* ist im Namensraum für FinTS-Transaktionen deklariert. Deshalb wird für dieses Element das Präfix *fintstrans* verwendet. Für allgemeine Datenstrukturen, die in der SEPA-Einzelüberweisung vorkommen, muss kein Präfix definiert werden, da der voreingestellte Namensraum (FinTS-Typen) an die eingeschachtelten Elemente vererbt wird. Analog zum voreingestellten Namensraum vererben sich auch die Präfix-Deklarationen auf die, in der Deklaration geschachtelten Datenstrukturen.

II.4 Verbandseigene Geschäftsvorfälle

Die Unterscheidung zwischen DK-weit definierten und verbandsspezifischen Geschäftsvorfällen erfolgt ebenfalls durch die Verwendung von Namensräumen. Für verbandseigene von FinTS abgeleitete Schemas müssen eigene Namensräume definiert werden.

Die URIs für die verbandseigenen Namensräume werden in FinTS nicht festgelegt. Sie sollten sich aber an dem oben (siehe *II.3.1 Aufbau des Namensraum-URIs*) beschriebenen Konzept orientieren:

```
<Verbands-Domain>/spec/<Schematyp>/<Version>/<Modulverzeichnis>
```

Schemas für verbandseigene Geschäftsvorfälle werden im jeweiligen Namensraum für Transaktionen abgelegt. Eine verbandsweit abweichende Verwendung der Unterteilung in die beiden Namensräume Typen und Transaktionen ist nicht zulässig. Die Schema-Module sollten immer in dem durch den Namensraum-URI spezifizierten Unterverzeichnis abgelegt werden.

Beispiel:

Die *xy-bank* könnte einen selbstdefinierten Geschäftsvorfall *t* unter dem Namensraum-URI

```
http://www.xy-bank.org/spec/xmlschema/4.1/transactions
```

als Schema unter dem URI

```
http://www.xy-bank.org/spec/xmlschema/4.1/transactions/t.xsd
```

ablegen und in den XML-Dokumenten das Präfix

```
xmlns:fintstrans-xy="http://www.xy-bank.org/spec/xmlschema/4.1/transactions"
```

verwenden.

Für verbandseigene von FinTS abgeleitete Schemas sollten der besseren Lesbarkeit wegen eigene Namensraum-Präfixe definiert werden. Die Präfixe sollten mit *fintstype-* bzw. *fintstrans-* beginnen, um zu verdeutlichen, dass von einem FinTS-Nachrichtenformat abgeleitet wurde.

Beispiele:

- BdB: *fintstype-b*, *fintstrans-b*
- BVR: *fintstype-g*, *fintstrans-g*
- DSGV: *fintstype-s*, *fintstrans-s*
- VÖB: *fintstype-v*, *fintstrans-v*

Die XML-Elemente für Benutzerauftrag, Kreditinstitutsantwort und Bankparameterdaten eines Geschäftsvorfalls sind grundsätzlich im selben Namensraum zu definieren. Die Benennung ist einheitlich anhand des folgenden Schemas vorzunehmen:

Benutzerauftrag:

```
XMLTag          ::= ID ' Req'
```

Kreditinstitutsrückmeldung:

```
XMLTag          ::= ID ' Resp'
```

Bankparameterdaten:

```
XMLTag          ::= ID '_Par'
```

mit:

```
ID              ::= Name ' ' Version
Name            ::= NCName
Version         ::= [1-9]([0-9])*
```

Beispiel: Einzelüberweisung (*SEPASingRemitt*) Version 1

```
SEPASingRemitt_1_Req
SEPASingRemitt_1_Resp
SEPASingRemitt_1_Par
```

Die Systematik ermöglicht u. a. die Zuordnung der Bankparameterdaten zu Geschäftsvorfällen, ohne zusätzliche Zuordnungstabellen.

Hinweis:

Die formale Grammatik verwendet eine einfache *Extended Backus-Naur Form* (EBNF), wie sie im [XML1.0] beschrieben ist. Das verwendete Nichtterminal *NCName* ist in [Namespaces] beschrieben.

II.5 Komposition und Validierung einer FinTS-Nachricht

FinTS-Nachrichten müssen gemäß der Schema-Validierung nach [XML Schema 1] bzw. [XML Schema 2] validiert werden.

Zur Validierung von FinTS-Nachrichten müssen diese im Namensraum für FinTS-Typen deklariert sein. Die eingebetteten Geschäftsvorfälle müssen im Namensraum für FinTS-Transaktionen oder in den Namensräumen der verbandsspezifischen Geschäftsvorfälle deklariert sein. Dazu sind auf der Nachrichtenebene und auf der Transaktionsebene die entsprechenden Schemas mit dem Attribut *schemaLocation* zuzuordnen. Das Attribut *schemaLocation* ist im Namensraum für Instanzen von XML-Schemas *http://www.w3.org/2001/XMLSchema-instance* definiert. Deshalb wird dem Attribut *schemaLocation* das entsprechend deklarierte Präfix *xsi* vorangestellt.

Das Attribut *schemaLocation* enthält als Wertepaar den Zielnamensraum (*targetNamespace*), dem die Schemas zugeordnet sind, sowie den URL auf das zugeordnete Schema-Dokument. Beide Werte sind durch ein Leerzeichen getrennt.

Beispiel:

```
<ReqMsg
  xmlns=http://www.fints.org/spec/xmlschema/4.1/messages
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xsi:schemaLocation="http://www.fints.org/spec/xmlschema/4.1/messages
                      http://www.fints.org/spec/xmlschema/4.1/messages/message.xsd">
...
  <Orders>
    <fintstrans:SEPASingRemitt_1_Req
        xmlns:fintstrans=http://www.fints.org/spec/xmlschema/4.1/transactions
        xsi:schemaLocation=http://www.fints.org/spec/xmlschema/4.1/transactions
                           http://www.fints.org/spec/xmlschema/4.1/transactions/
                           SEPASingRemitt-1.xsd">
        ...
    </fintstrans:SEPASingRemitt_1_Req>
  </Orders>
  ...
</ReqMsg>
```

Im Beispiel wird eine SEPA-Einzelüberweisung als Auftrag in eine Benutzernachricht eingebettet. Die Benutzernachricht kann mit Ausnahme der enthaltenen Geschäftsvorfälle durch die Angabe der beiden Schema-Lokationen für *Messages* (*http://www.fints.org/spec/xmlschema/4.1/messages/message.xsd*) und *Types* (*http://www.fints.org/spec/xmlschema/4.1/types, für die Includes common.xsd, formats.xsd und patterns.xsd*) validiert werden.

Für jeden Geschäftsvorfall existiert ein eigenes Schema-Dokument zur Validierung. Im Beispiel wird daher dem eingebetteten Geschäftsvorfall das Schema *http://www.fints.org/spec/xmlschema/4.1/transactions/SEPASingRemitt-1.xsd* zur Validierung zugeordnet.

Normalerweise ist es effizienter, die FinTS-Schemas zur Validierung lokal zu speichern. Bei der lokalen Speicherung der Schemas sollte der Verzeichnispfad nach der FinTS-Domain (siehe *II.3.1 Aufbau des Namensraum-URIs*) erhalten bleiben.

Wenn ein System lokal gespeicherte Schemas verwendet, wird es evtl. auch in der zu sendenden FinTS-Nachricht diese lokale Schema-Adresse angeben (siehe nachfolgendes Beispiel).

Die jeweils empfangende Seite wird die lokalen Schema-Adressen der sendenden Seite im Allgemeinen nicht auflösen können. Es wird insbesondere für die Kreditinstitutsseite grundsätzlich empfohlen, die Schema-Adresse im Dokument zu ignorieren und die lokal gespeicherten Schemas anhand geeigneter Mechanismen selbstständig zu identifizieren.

Beispiel mit lokalen Schema-Adressen:

```
<ReqMsg   xmlns="http://www.fints.org/spec/xmlschema/4.1/messages"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.fints.org/spec/xmlschema/4.1/messages
                  c:/fints/spec/xmlschema/4.1/messages/message.xsd">
  ...
  <Orders>
    <fintstrans:SEPASingRemitt_1_Req
     xmlns:fintstrans="http://www.fints.org/spec/xmlschema/4.1/transactions"
     xsi:schemaLocation="http://www.fints.org/spec/xmlschema/4.1/transactions"
                       c:/fints/spec/xmlschema/4.1/transactions/SEPASingRemitt-1.xsd">
     ...
   </fintstrans:SEPASingRemitt_1_Req>
  </Orders>
  ...
</ReqMsg>
```

II.6 Kombinierter Einsatz von FinTS mit anderen XML-Formaten

FinTS lässt sich durch die Verwendung von Namensräumen in Kombination mit anderen XML-Formaten einsetzen, so dass Namenskonflikte bei der Verarbeitung leicht aufgelöst werden können. Dazu bieten sich vielfältige Möglichkeiten. Der folgende Abschnitt zeigt die Anforderungen an die in Kombination mit FinTS verwendeten Fremdformate.

II.6.1 Anforderungen

Bei den Anforderungen handelt es sich um Empfehlungen, die zur Modularisierung, Robustheit und einfachen Wartbarkeit von XML-Anwendungen beitragen.

XML-Schemas erlauben die Validierung von Teilen eines XML-Dokuments gegen mehrere verschiedene Schemas. Somit können XML-Nachrichten in heterogenen Formaten typsicher in einem einzigen XML-Dokument transportiert werden.

Wenn XML-Fremdformate in Kombination mit FinTS eingesetzt werden, dann sollten deren Typdefinitionen und Deklarationen in einem separaten Namensraum vorgenommen werden. Dadurch können die Anwendungen bei der Verarbeitung der XML-Dokumente eindeutig entscheiden, welchem Nachrichtenformat ein XML-Fragment zuzuordnen ist. Außerdem sollte eindeutig festgelegt werden, wie die XML-Dokumente zu validieren sind.

Falls in einem XML-Dokument FinTS-Nachrichten mit anderen Nachrichtentypen (z. B. S.W.I.F.T.-Nachrichten) gemischt werden (siehe *II.6.3 Integration von FinTS-Nachrichten in Fremdformate*), sollte der voreingestellte Namensraum für die FinTS-fremden Datenstrukturen mit einem zum Fremdformat passenden Namensraum-URI belegt werden.

Falls das Fremdformat nicht in einem Namensraum definiert wurde, ist der voreingestellte Namensraum auf den Standardwert zurückzusetzen:

```
<Fremdformat xmlns="">
  ...
</Fremdformat>
```

II.6.2 Integration fremder Transaktionsformate in FinTS

FinTS-Nachrichten erlauben die Integration von Geschäftsvorfällen in Fremdformaten an bestimmten dafür vorgesehenen Stellen im Nachrichtenaufbau. An diesen Stellen sind in den XML-Schemas von FinTS beliebige Elemente des XML-Schematyps *any* vorhanden.

Beispiel:

```
<xs:any namespace="##any" processContents="lax"/>
```

An den Stellen, wo Fremdformate in die FinTS-Nachricht integriert werden können, werden sie validiert, sofern der Parser in der Lage ist, dies zu tun (Attribut *processContents="lax"*).

Im folgenden Beispiel wird ein [SEPA]-Geschäftsvorfall in eine FinTS-Nachricht integriert:

```
<ReqMsg
 xmlns="http://www.fints.org/spec/xmlschema/4.1/messages"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.fints.org/spec/xmlschema/4.1/
  fints/spec/xmlschema/4.1/messages/message.xsd">
  ...
  <Orders>
    <pain.001.003.03
     xmlns=" urn:iso:std:iso:20022:tech:xsd:pain.001.003.03"
     xsi:schemaLocation="urn:iso:std:iso:20022:tech:xsd:pain.001.003.03.xsd">
      ...
    </pain.001.003.03>
  </Orders>
</ReqMsg>
```

II.6.3 Integration von FinTS-Nachrichten in Fremdformate

FinTS-Nachrichten können unter Erhaltung einer syntaktisch korrekten XML-Struktur des Gesamtdokumentes in ein XML-Dokument eingebettet werden. Damit bleiben sie beim Mischen von Fremdformaten mit FinTS-Nachrichten als eindeutig validierbare Einheit erhalten.

Der Namensraum der FinTS-Nachrichten macht diese im Bezug zum übrigen Dokument unterscheidbar. Die Angabe des Schemas gewährleistet die typsichere Verarbeitung der Nachricht.

Beispiel:

```
...
<ReqMsg
  xmlns="http://www.fints.org/spec/xmlschema/4.1/messages"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.fints.org/spec/xmlschema/4.1/messages
                      http://www.fints.org/spec/xmlschema/4.1/messages/message.xsd">
  ...
  <Orders>
    <pain.001.003.03   xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.003.03"
  xsi:schemaLocation="urn:iso:std:iso:20022:tech:xsd:pain.001.003.03.xsd">
        ...
    </pain.001.003.03>
  </Orders>
  ...
</ReqMsg>

...
  <ReqMsg
    xmlns="http://www.fints.org/spec/xmlschema/4.1/messages"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.fints.org/spec/xmlschema/4.1/messages

http://www.fints.org/spec/xmlschema/4.1/messages/message.xsd">
    ...
  </ReqMsg>
...
```

II.6.4 Integration von FinTS-Transaktionen in Fremdformate

Die Integration von FinTS-Geschäftsvorfällen in Fremdformate geschieht analog zu *II.6.3 Integration von FinTS-Nachrichten in Fremdformate*. Der Unterschied besteht lediglich darin, dass für die FinTS-Transaktionen ein zusätzliches Namensraum-Präfix deklariert werden sollte, um die Lesbarkeit zu erhöhen.

Beispiel:

```
...
  <ReqMsg
    xmlns="http://www.fints.org/spec/xmlschema/4.1/messages"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.fints.org/spec/xmlschema/4.1/messages

http://www.fints.org/spec/xmlschema/4.1/messages/message.xsd">
    ...
  </ReqMsg>    ...
```

II.7 FinTS-Datentypen

Die FinTS-Datentypen sind konform zur XML Schema Spezifikation des W3C modelliert. Der Abschnitt vermittelt die wichtigsten Modellierungsprinzipien für XML-Schemas, die in FinTS angewendet werden. Leser, die mit der Modellierung von XML-Schemas vertraut sind, können diesen Abschnitt überspringen.

II.7.1 Binäre Daten

Binäre Daten werden durch den FinTS-Datentyp *bin* modelliert, der vom XML-Schema-Datentyp *base64Binary* abgeleitet wird. Es gelten dieselben Kodierungsregeln wie für den *base64Binary*-Datentyp gemäß [XML Schema 2].

II.7.2 Transparente Daten

Transparente Daten (siehe [Formals], Abschnitt *II.2 Syntaktische Festlegungen*) werden als binäre Daten eingestellt (siehe *II.7.1 Binäre Daten*).

II.7.3 Status und Anzahl

Status und Anzahl eines Elements werden in XML-Schema durch die Facetten *minOccurs* und *maxOccurs* für minimale und maximale Anzahl modelliert. Eine Muss-Datenstruktur erhält in der Schemabeschreibung den Wert *1* für *minOccurs*, eine optionale Datenstruktur erhält den Wert *0*. Die maximale Anzahl für das Auftreten einer Datenstruktur wird durch den Wert der Facette *maxOccurs* bestimmt. Der Wert *unbounded* definiert eine unbeschränkte Häufigkeit für das Auftreten einer Datenstruktur.

Für die Facetten in XML Schema existieren Standard-Belegungen, so dass in einem Schema nicht alle Facetten definiert sein müssen. Die Facette *minOccurs* ist mit dem Vorgabewert *1* belegt. Deshalb handelt es sich um eine Muss-Datenstruktur, falls sie nicht explizit mit einem Wert belegt ist. Auch die Facette *maxOccurs* ist mit dem Standardwert *1* belegt (s. Beispiel).

Im Beispiel wird festgelegt, dass das Element *PublicKey* bis zu zweimal auftreten oder entfallen kann.

```
<xs:element name="PublicKey" minOccurs="0" maxOccurs="2">
  ...
</xs:element>
```

II.7.4 Längenangaben

Die Facetten *length* bzw. *minLength* und *maxLength* legen im XML-Schema die Länge eines einfachen (unstrukturierten) Typs (simple type) fest. Für die in FinTS verwendeten einfachen Typen *string* und *binary* existieren Facetten, welche die Länge eines Typs beschreiben. Für Datums-Typen und numerische Typen hingegen nicht.

Die minimale und maximale Länge eines Typs werden in XML Schema durch die Facetten *minLength* und *maxLength* beschrieben. Die Facette *maxLength* wird zur Definition einer variablen Länge mit einer Längenbegrenzung im Schema mit dem Wert der Länge des Feldes belegt. Bei einer fest vorgeschriebenen Feldlänge wird die Facette *length* belegt.

Beispiel für eine fest vorgeschriebene Feldlänge:

```
<xs:element name="CountryCode" length="3"/>
```

II.7.5 Aufzählungstypen

Aufzählungstypen werden in FinTS durch die Ableitung von einem einfachen FinTS-Datentyp beschrieben. Dabei wird der Wertebereich des abgeleiteten Typs durch eine Liste von Aufzählungswerten eingeschränkt.

Im Beispiel wird die *Rolle des Signierenden* durch drei Aufzählungswerte beschrieben. Der Typ der Aufzählungswerte ist durch Einschränkung vom Typ *an* (alphanumerisch) abgeleitet.

```
<xs:element name="SignerRole">
  <xs:simpleType>
    <xs:restriction base="an">
      <xs:enumeration value="ISS"/>
      <xs:enumeration value="WIT"/>
      <xs:enumeration value="MSG"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

II.8 Referenzierung mit XPath-Ausdrücken

In FinTS-Nachrichten können Elemente durch [XPath]-Ausdrücke referenziert werden. Die Referenzierung von Elementen innerhalb einer Nachricht wird in verschiedenen Kontexten angewendet: Bei den Rückmeldungen zu Benutzernachrichten, bei der Signierung und als Kennzeichnung des Überbringers eines Nachrichtenteils. In FinTS sind nur bestimmte Formen der Referenzierung zulässig, die weiter unten in diesem Abschnitt beschrieben werden.

Die Vorschriften zur Bildung eines XPath-Ausdrucks werden bei der Beschreibung der jeweiligen Verwendung (z. B. im Abschnitt *IV.3 Botensignatur*) anhand der folgenden Attribute festgelegt:

Referenz-Ziel(e)

Es wird festgelegt, auf welche Elemente referenziert werden darf bzw. welche zwingend referenziert werden müssen.

Gültigkeitsbereich

Der Gültigkeitsbereich gibt das Wurzelelement *R* desjenigen Teilbaums an, innerhalb dessen der Ausdruck gültig sein muss. Dieses bedeutet, dass das Ergebnis eines jeden Lokalisierungsschrittes innerhalb eines Lokalisierungspfades ein Element der Knotenmenge M, gebildet aus *R* sowie allen seinen Nachfolgern, sein muss.

Kontext

Optional wird der Kontext zur Auswertung des XPath-Ausdrucks angegeben. Die Angabe kann entfallen, wenn der Kontext durch den äußeren Rahmen vorgegeben ist (z. B. Verwendung in [XPath Filter]).

Zulässiger Ausdruck

Es wird festgelegt, welche der unten beschriebenen Lokalisierungspfade für die Angabe in einem XPath-Element erlaubt sind.

Zur Referenzierung wird eine eingeschränkte Form der abgekürzten Syntax gemäß [XPath] verwendet. Die Referenzierungsausdrücke müssen - soweit nicht anders angegeben - einem der folgenden Ausdrücke für Lokalisierungspfade genügen.

Absoluter Lokalisierungspfad:

```
AbsoluteLocationPath ::= '/' RelativeLocationPath?
```

Relativer Lokalisierungspfad:

```
RelativeLocationPath ::= Step ( '/' Step )*
```

Relativer Lokalisierungspfad für Verwendung in [XPath Filter]-Ausdrücken:

```
HereLocationPath     ::= ('here()/' RelativeLocationPath)
```

mit:

```
Step                 ::= ( QName Predicate? )
                         | '..'
QName                ::= Prefix ':' LocalName
LocalName            ::= NCName
Predicate            ::= '[' Index ']'
Index                ::= [1-9]([0-9])*
```

Es gelten die Hinweise zur formalen Grammatik aus *II.4 Verbandseigene Geschäftsvorfälle*.

- **Erläuterungen:**

  Ein Lokalisierungsschritt (*Step*) besteht aus einem Knotentest mit qualifiziertem Elementnamen sowie einem Prädikat für den Index des Knotens. Der Lokalisierungsschritt *prefix:localName[i]* verweist somit auf das i-te Kindelement des Kontextknotens mit dem Namen *localName* und dem durch *prefix* definierten Namensraum.

  Der Lokalisierungsschritt „*..*“ verweist auf das Elternelement des Kontextknotens.

  Ein relativer Lokalisierungspfad (*RelativeLocationPath*) wird aus einer Folge von einem oder mehreren Lokalisierungsschritten gebildet, welche durch das Zeichen / getrennt und von links nach rechts interpretiert werden. Jeder Schritt verweist auf genau einen Elementknoten, welcher den Kontextknoten für den nächsten Schritt darstellt. Der Kontextknoten für den ersten Schritt ist das Elternelement des Textknotens, welcher den XPath-Ausdruck enthält.

  Ein absoluter Lokalisierungspfad (*AbsoluteLocationPath*) wird von dem Zeichen / angeführt, welches den Kontextknoten für den ersten Lokalisierungsschritt auf den Wurzelknoten des Dokuments setzt, das den Knoten mit dem XPath-Ausdruck enthält.

  Hinweis:

  Der Wurzelknoten des Dokuments ist nicht zu verwechseln mit dem Elementknoten des Dokumentenelements, welches seinerseits ein Kind des Wurzelknotens darstellt (siehe auch [XPath], Abschnitt Datenmodell).

  Die Funktion *here()* ist eine Erweiterung zum [XPath]-Standard gemäß [XPath Filter]. Sie setzt den Kontextknoten für den ersten Lokalisierungsschritt eines Lokalisierungspfades innerhalb einer [XPath Filter]-Transformation auf den Elternknoten des Textknotens, welcher den XPath-Ausdruck enthält und ermöglicht somit die Bildung von Lokalisierungspfaden relativ zu diesem Kontextknoten (*HereLocationPath*) Ohne Verwendung der *here()*-Funktion ist der Kontextknoten einer [XPath Filter]-Transformation der Wurzelknoten des Dokuments, welches die Eingabeknotenmenge der Transformation enthält.

  Bei der Verwendung in [XPath Filter] ist insbesondere die Vereinigung von Lokalisierungspfaden mittels des Vereinigungsoperators | erlaubt. Die resultierende Knotenmenge nach der [XPath-Filter]-Transformation ist die Vereinigung der resultierenden Knotenmengen der einzelnen Lokalisierungspfade in Dokumentordnung.

  Das Namensraumpräfix muss gemäß [Namespaces] in dem Element deklariert werden, welches den XPath-Ausdruck enthält, oder in einem dazu übergeordneten Element.

Beispiel:

Das Element *BankCode* aus einer Benutzernachricht, enthielt eine falsche Kennung. Deshalb enthält die Kreditinstitutsnachricht im Element *ElementRef* eine Referenz auf den ungültigen Eintrag.

```
...
  <MsgRespState xmlns="http://www.fints.org/spec/xmlschema/4.1/mesages">
    <RespCode>9210</RespCode>
    <ElementRef
     xmlns:fintstype="http://www.fints.org/spec/xmlschema/4.1/messages">
/fintstype:ReqMsg[1]/fintstype:ReqMsgHeader[1]/fintstype:BankID[1]/fintstype:BankCode
[1]</ElementRef>
    <RespText>Kennung nicht erlaubt.</RespText>
  </MsgRespState>
...
```

II.9 Symbole in den Schemadiagrammen

Die nächsten Kapitel beschreiben die Schemas zum Aufbau von FinTS-Nachrichten in der Form von Diagrammen. In den beiden folgenden Beispielen werden die in den Diagrammen verwendeten Symbole beschrieben.

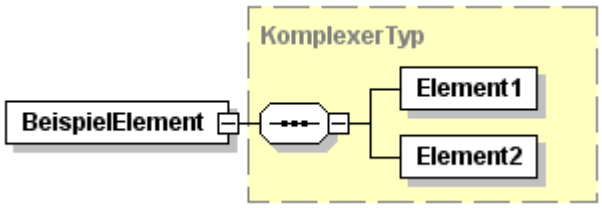


Abbildung 1: Beispiel für eine Elementdeklaration

In den Diagrammen zu den FinTS-Schemas werden Elemente durch eine Textbox abgebildet, die den Elementnamen enthält. Das deklarierte Element befindet sich auf der linken Seite des Diagramms. Zu jedem Element gehört eine hierarchische Typbeschreibung (Inhaltsmodell), die sich rechts vom Element befindet. Wenn zur Beschreibung des Inhaltsmodells auf einen vordefinierten Typ referenziert wird, ist dieser Typ von einem gestrichelten Rahmen umgeben. Wenn kein vordefinierter Typ verwendet wird, handelt es sich um eine namenlose (anonyme) Typdefinition.

Im Beispiel wird das Inhaltsmodell des Elements *BeispielElement* durch eine Sequenz von zwei Elementen (*Element1* und *Element2*) definiert. Die Definition des Inhaltsmodells erfolgt durch die Verwendung des Typs *complexType*.

Bei einer Typdefinition wird das Inhaltsmodell auf dieselbe Weise abgebildet wie bei einer Elementdeklaration. Zur Unterscheidung wird der Typ durch eine Textbox mit abgerundeten Ecken dargestellt.

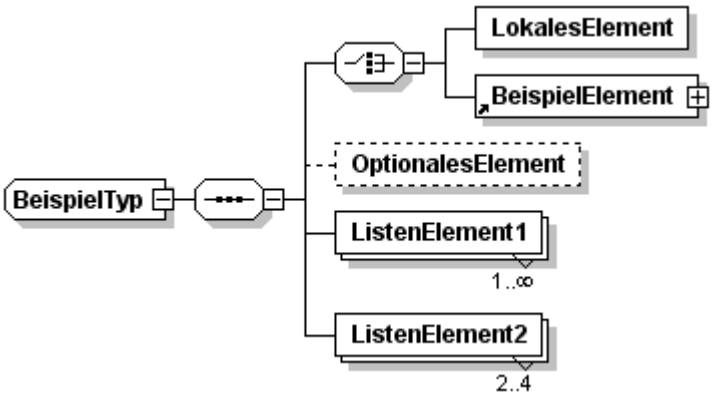


Abbildung 2: Beispiel für eine Typdefinition

Die Inhaltsmodelle der definierten Typen enthalten Sequenzen und Wahlmöglichkeiten. Bei einer Sequenz enthält das übergeordnete Element (auf der linken Seite des Sequenzsymbols) die untergeordneten Elemente (auf der rechten Seite des Sequenzsymbols) in der im Diagramm von oben nach unten festgelegten Reihenfolge. Bei einer Wahlmöglichkeit enthält das übergeordnete Element genau eines der sich unter dem Schaltersymbol befindlichen Elemente.

Optionale Teile des Inhaltsmodells werden durch gestrichelte Linien und Textboxen repräsentiert. Wenn Komponenten des Inhaltsmodells häufiger als einmal auftreten können, wird an den zugehörigen Symbolen der Häufigkeitsbereich annotiert. Der

Häufigkeitsbereit 1..∞ für ein Element gibt an, dass dieses Element einmal oder beliebig oft nacheinander auftreten kann.

Im vorangehenden Beispiel besteht das Inhaltsmodell des Typs *BeispielTyp* aus der Sequenz „Wahlmöglichkeit zwischen *LokalesElement* und *BeispielElement*“, *OptionalesElement*, *ListenElement1* und *ListenElement2*. Der Typ von *LokalesElement* wird in der Typdefinition von *BeispielTyp* festgelegt. Das Element *BeispielElement* wird durch die Referenz auf eine globale Elementdeklaration beschrieben. Eine solche Referenz wird durch einen schrägen Pfeil symbolisiert. Das in der Sequenz nach der Wahlmöglichkeit positionierte Element *OptionalesElement* kann einmal auftreten oder entfallen. Das Element *ListenElement1* tritt mindestens einmal und beliebig oft auf. Das Element *ListenElement2* wird aufeinander folgend 2 bis 4-mal verwendet.

III. NACHRICHTENAUFBAU

III.1 Überblick

Die folgenden Abbildungen geben einen Überblick über den Nachrichtenaufbau in FinTS. Eine FinTS-Nachricht ist entweder eine Benutzer- oder eine Kreditinstitutsnachricht. Beide Nachrichtenformen folgen demselben prinzipiellen Aufbau.

Sie sind in einen Nachrichtenkopf (Header) und einen Nachrichtenkörper (Body) unterteilt. Der Nachrichtenkörper enthält genau einen von acht möglichen Nachrichtentypen. Der Aufbau dieser Nachrichtentypen wird in den folgenden Abschnitten genauer beschrieben.

Die generelle Verwendung der Nachrichten im FinTS-Protokoll ist in den anderen Teilen der Spezifikation ([Formals], [HBCI], [PINTAN]) beschrieben. Die Belegung einzelner Datenelemente kann außerdem im [DataDictionary] nachgesehen werden, das Dictionary ist nach den deutschsprachigen Begriffen sortiert, die in den Abbildungen jeweils als Anmerkung an den Elementen angebracht sind. Spezielle Belegungsvorschriften im besonderen Kontext eines Nachrichtentyps sind bei der jeweiligen Nachrichtenbeschreibung angegeben.

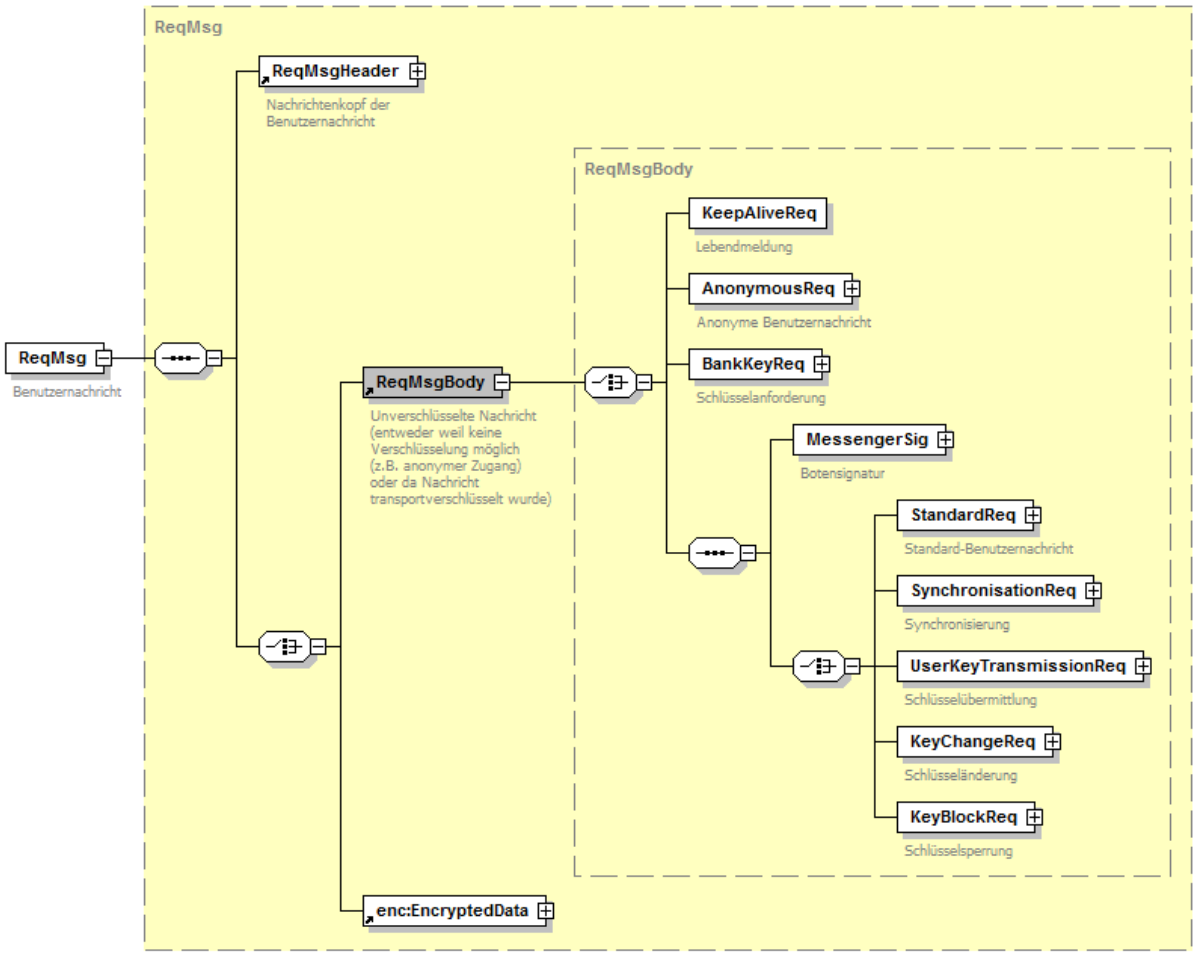


Abbildung 3: Benutzernachricht

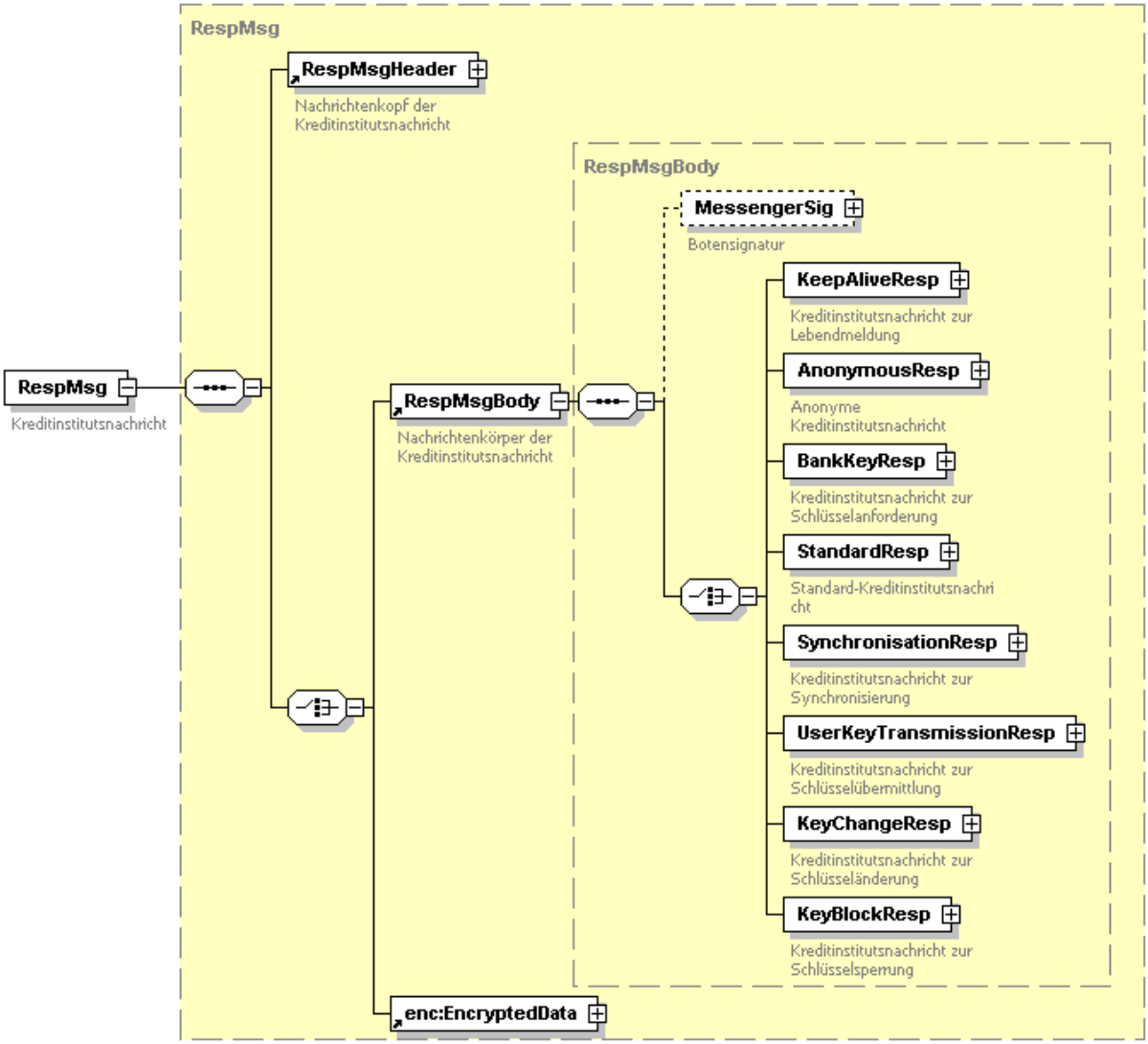


Abbildung 4: Kreditinstitutsnachricht

III.2 Allgemeiner Aufbau von Benutzernachricht und Kreditinstitutsnachricht

Dieser Abschnitt erläutert den grundsätzlichen Aufbau der Nachrichten. Die Besonderheiten der acht unterschiedlichen Nachrichtentypen werden in den Abschnitten *III.3 Verschiedene Benutzer- und Antwortnachrichten* und *III.4 Keymanagement-Nachrichten* näher beleuchtet. Zum Nachrichtenaufbau siehe auch [Formals], Abschnitt *II.9 Benutzernachrichten allgemein* und [Formals], Abschnitt *II.10 Kreditinstitutsnachrichten allgemein*.

III.2.1 Benutzernachricht

Eine Benutzernachricht besteht aus dem Nachrichtenkopf (*ReqMsgHeader*) und dem Nachrichtenkörper (*ReqMsgBody*) (siehe auch Abbildung 3).

Der Nachrichtenkörper kann verschlüsselt (als *enc:EncryptedData*) oder unverschlüsselt (als *ReqMsgBody*) übertragen werden.

Bei der verschlüsselten Übertragung ersetzt das *EncryptedData*-Element das Element *ReqMsgBody*. Dabei wird der Nachrichtenkörper durch den [XML Encryption]-Standard verschlüsselt. Bei der Verschlüsselung des Nachrichtenkörpers befindet sich der verschlüsselte Inhalt von *ReqMsgBody* innerhalb des *EncryptedData*-Elements (siehe *V.2 Verschlüsselung des Nachrichtenkörpers*). In welchen Fällen der Nachrichtenkörper verschlüsselt werden muss, ist in [Formals], Abschnitt *II.13 Verschlüsselung der Kommunikation* festgelegt. Zusätzlich oder statt der Verschlüsselung kann auf formal gleiche Art eine Komprimierung des Nachrichtenkörpers stattfinden, siehe [Formals], Abschnitt *II.14 Komprimierung* und *V.4 Komprimierung*.

Der Inhalt des Nachrichtenkörpers sowie des Nachrichtenkopfes wird bei den meisten Nachrichtentypen mit einer Botensignatur im Element *MessengerSig* versehen. Die Botensignatur wird im Abschnitt *IV.3 Botensignatur* beschrieben. Festlegungen zur Verwendung der Signaturen in FinTS finden sich in [Formals], Abschnitt *II.12.4 Vorgehensweise beim Signieren und Verschlüsseln* und [Formals], Abschnitt *II.4 Signatur-Rollenverteilung bei Kommunikation mit und ohne Intermediär*.

Signierte (personalisierte) Nachrichtentypen sind

- die Standard-Benutzernachricht (*StandardReq*),
- die Synchronisierung (*SynchronisationReq*),
- die erstmalige Übermittlung eines Kundenschlüssels (*UserKeyTransmissionReq*),
- die Schlüsseländerung (*KeyChangeReq*),
- sowie die Schlüsselsperrung (*KeyBlockReq*).

Unsignierte (anonyme) Nachrichtentypen sind

- die anonyme Benutzernachricht (*AnonymousReq*),
- die Lebendmeldung (*KeepAliveReq*),
- sowie die Anforderung eines Bank-Schlüssels (*BankKeyReq*).

III.2.1.1 Nachrichtenkopf einer Benutzernachricht

Der Nachrichtenkopf der Benutzernachricht enthält die Identifizierung des Kreditinstituts (*BankID*), eine Nachrichtennummer (*MsgNo*) und die Benutzer- und Kreditinstitutsseitigen Referenzen (*UserRef*, *UserTextRef* bzw. *BankRef*) zur Steuerung des Dialogablaufs (siehe dazu auch [Formals], Abschnitt *II.6 Synchrone Kommunikationsverfahren*). Er liegt außerhalb des verschlüsselbaren Teilbaums, wird aber von der Botensignatur signiert.

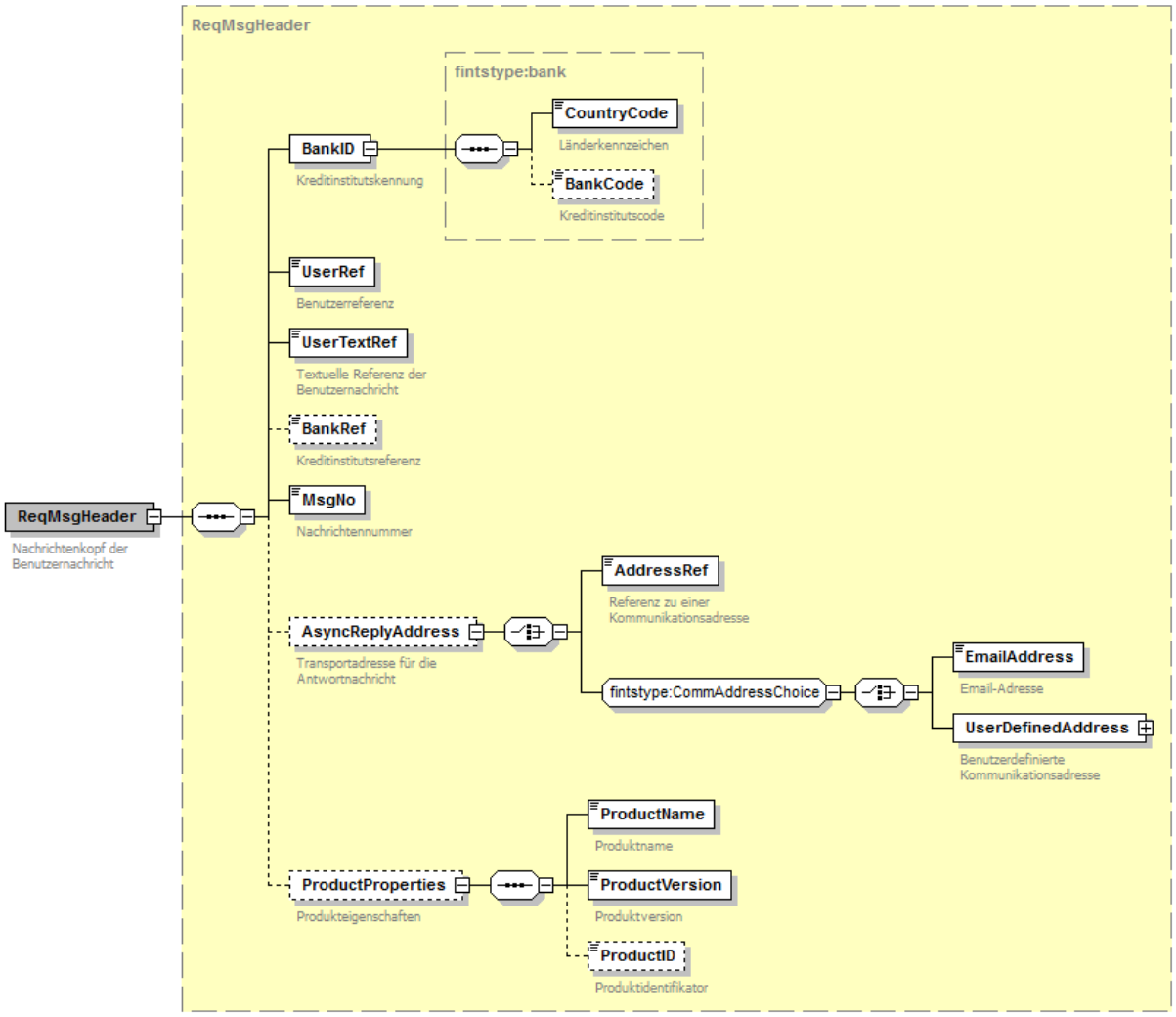


Abbildung 5: Nachrichtenkopf einer Benutzernachricht

Für asynchrone Kommunikationsverfahren (siehe [Formals], Abschnitt *II.7 Asynchrone Kommunikationsverfahren (Datagramme)*) ist die Angabe einer Antwort-Email-Adresse (*AsyncReplyAddress*) möglich.

III.2.1.2 Nachrichtenkörper einer Benutzernachricht

Alle Nachrichtentypen außer dem der Lebendmeldung enthalten einen oder mehrere Auftragsteile (*RequestList*), in die unterhalb von *Orders* jeweils mehrere Aufträge eingestellt werden können (siehe [Formals], Abschnitt *II.9 Benutzernachrichten allgemein*). Auftragsteile können in gleicher Weise wie der Nachrichtenkörper verschlüsselt und/oder komprimiert werden (siehe *V.3 Verschlüsselung von Aufträgen und Auftragsantworten* und *V.4 Komprimierung*).

Zum Einsatz der Auftragsverschlüsselung in FinTS siehe auch [Formals], Abschnitt *II.12.2 Teilverschlüsselte Nachrichten*.

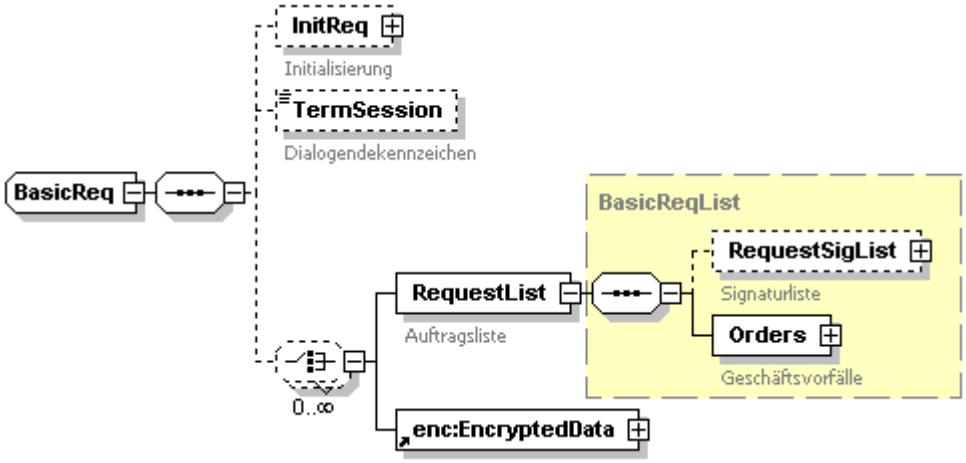


Abbildung 6: Basistyp für die Nachrichtenkörper der Benutzernachricht

In den personalisierten Nachrichtentypen kann ein Auftragsteil mit einer oder mehreren Auftragssignaturen in der *RequestSigList* versehen werden. Die Auftragssignatur wird im Abschnitt *IV.4 Auftragssignatur* beschrieben.

Alle Nachrichtentypen außer der Lebendmeldung enthalten eine Initialisierung (*InitReq*), die zur Einleitung eines Dialogs dient (siehe [Formals], Abschnitt *II.6 Synchrone Kommunikationsverfahren*). In einem Dialog mit Standard-Benutzernachrichten ist sie ausschließlich in der ersten Nachricht zu verwenden. Bei allen anderen Nachrichtentypen ist sie ein Pflichtfeld, dadurch kann eine solche Nachricht nicht als Folgenachricht einer Dialogfolge verwendet werden.

Mit dem Element *TermSession* wird angezeigt, dass ein Dialog mit dieser Nachricht endet (vgl. [Formals], Abschnitt *II.6.2 Dialogbeendigung und Endenachrichten*). Es muss in einem Dialog mit Standard-Benutzernachrichten genau in der letzten Benutzernachricht eingefügt werden, bei allen anderen Nachrichtentypen ist es zwingend vorhanden - damit kann einer solchen Nachricht keine weitere Nachricht eines Dialogs folgen.

Zusammen mit der zwingenden Initialisierung folgt daraus, dass alle Nachrichtentypen außer der Standard-Benutzernachricht und der Lebendmeldung ausschließlich als FinTS-Datagramme verwendet werden (vgl. [Formals], Abschnitt *II.7 Asynchrone Kommunikationsverfahren (Datagramme)*).

III.2.1.3 Initialisierung

Die Initialisierung zur Einleitung eines Dialogs besteht aus den Segmenten zur Identifikation (*Identification*) und zur Verarbeitungsvorbereitung (*ProcPreparation*), vgl. auch [Formals], Abschnitt *II.15 Initialisierung*.

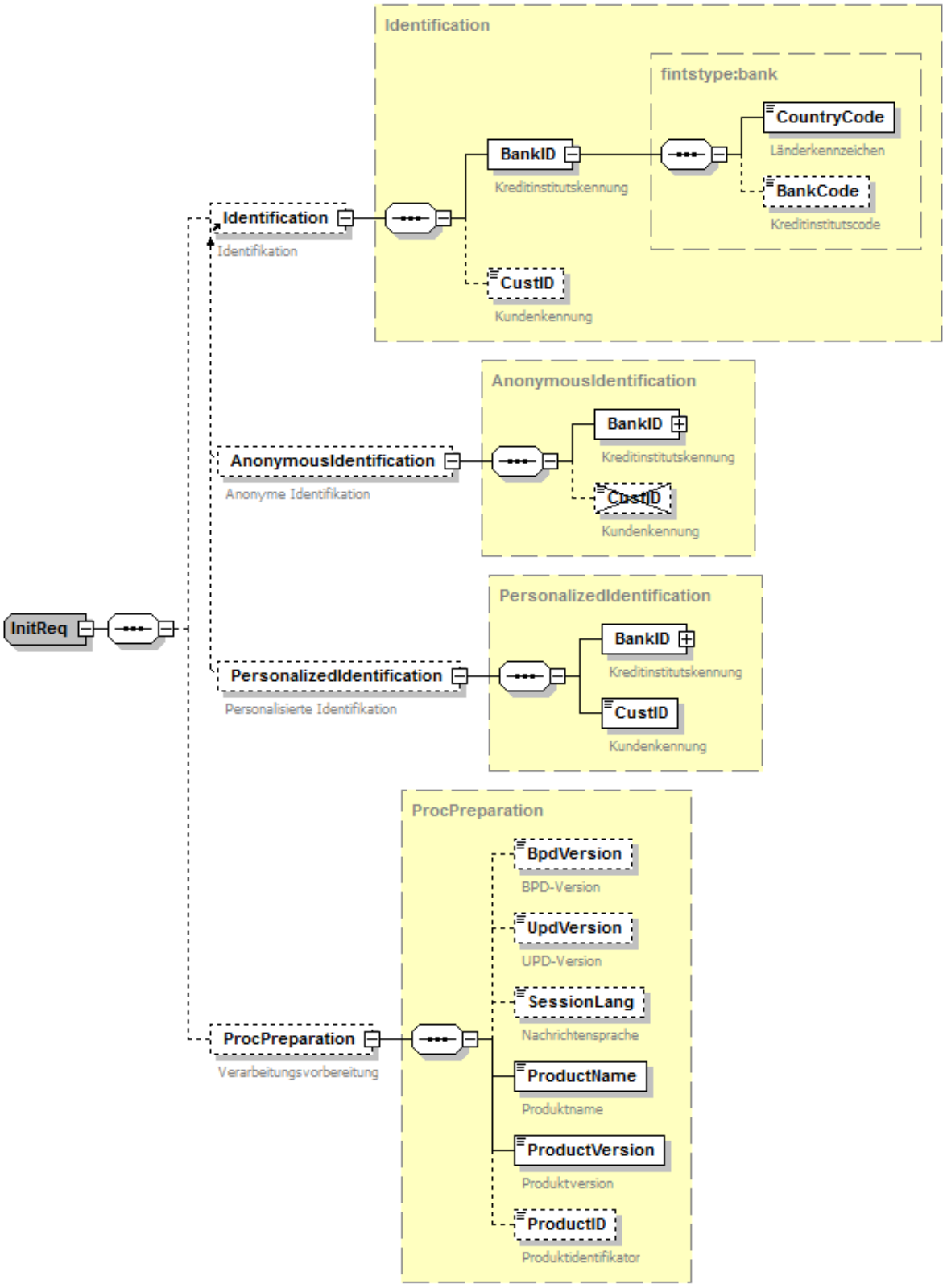


Abbildung 7: Initialisierung

Das Segment zur Identifikation existiert in einer anonymen Ausprägung (*AnonymousIdentification*) für die anonyme Benutzernachricht und einer personalisierten Ausprägung (*PersonalizedIdentification*) für die anderen (personalisierten) Nachrichtentypen. Sie unterscheiden sich nur darin, dass die personalisierte Variante zwingend eine Kundenkennung (*CustID*) enthält, die anonyme Variante hingegen keine Kundenkennung enthalten kann.

III.2.1.4 Auftrag

Ein Auftrag ist in der Auftragsliste unterhalb des Elements *Orders* als freies Inhaltsmodell definiert. Hier wird in der Standard-Benutzernachricht ein beliebiger Geschäftsvorfall aus dem Namensraum für FinTS-Geschäftsvorfälle, aus dem Namensraum für administrative FinTS-Geschäftsvorfälle oder einem individuellen Namensraum (vgl. *II.4 Verbandseigene Geschäftsvorfälle*) eingestellt.

- Die in der FinTS-Spezifikation definierten Transaktionsaufträge und Abholaufträge finden sich in [Messages].
- Abschnitt *III.7 Administrative Aufträge* enthält administrative Aufträge zur Verwendung in der Standard-Benutzernachricht.
- In anonymen Benutzernachrichten können nur bestimmte Geschäftsvorfälle verwendet werden (vgl. [Formals], Abschnitt *II.17 Anonymer Zugang*).
- Für die anderen Nachrichtentypen sind jeweils spezielle administrative Geschäftsvorfälle definiert, die nur zusammen mit diesem Nachrichtentyp verwendet werden können - siehe dazu *III.3 Verschiedene Benutzer- und Antwortnachrichten*, *III.4 Keymanagement-Nachrichten*.

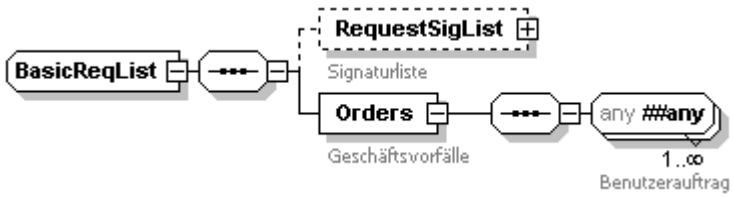


Abbildung 8: Benutzerauftrag im abstrakten Basistyp der Auftragsliste

Eine Auftragsliste stellt die kleinste syntaktische Einheit in einer Benutzernachricht dar, die FinTS-konform, unabhängig vom äußeren Kontext verarbeitbar ist.

III.2.2 Kreditinstitutsnachricht

Eine Kreditinstitutsnachricht hat denselben prinzipiellen Aufbau wie eine Benutzernachricht. Sie besteht aus einem Nachrichtenkopf (*RespMsgHeader*) und einem unverschlüsselten oder verschlüsselten Nachrichtenkörper (siehe Abbildung 4). Botensignatur, Verschlüsselung und Komprimierung sind analog zur Benutzernachricht modelliert, vgl. *III.2.1 Benutzernachricht*.

Den acht unterschiedlichen Typen der Benutzernachricht sind korrespondierende Typen der Kreditinstitutsnachricht zugeordnet.

III.2.2.1 Nachrichtenkopf einer Kreditinstitutsnachricht

Die folgende Abbildung zeigt den Aufbau eines Nachrichtenkopfs einer Kreditinstitutsnachricht.

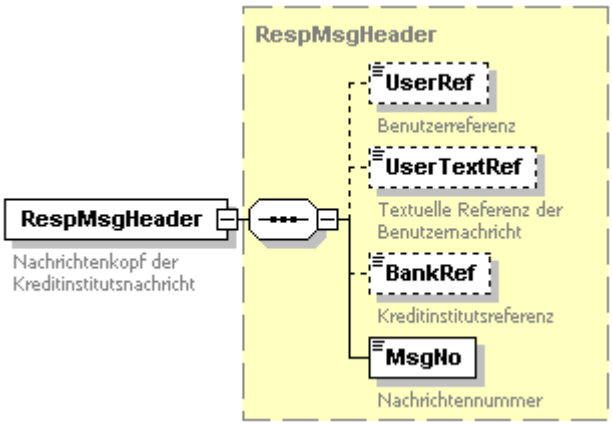


Abbildung 9: Nachrichtenkopf einer Kreditinstitutsnachricht

Im Nachrichtenkopf der Kreditinstitutsnachricht werden die Referenzen des Benutzers (*UserRef*, *UserTextRef*) und die Referenz des Kreditinstituts (*BankRef*) sowie die Nachrichtennummer der Benutzernachricht (*MsgNo*) zurück gemeldet (siehe dazu auch [Formals], Abschnitt *II.15.2 Kreditinstitutsnachricht*).

III.2.2.2 Nachrichtenkörper einer Kreditinstitutsnachricht

Der Nachrichtenkörper einer Kreditinstitutsnachricht ähnelt inhaltlich dem einer Benutzernachricht. Bei der Kreditinstitutsnachricht ist die Botensignatur allerdings für alle acht Nachrichtentypen optional. Die Inhaltsmodelle für die Nachrichtenkörper der verschiedenen Nachrichtentypen mit Ausnahme der Lebendmeldung sind von einem gemeinsamen Basistyp abgeleitet, Abbildung 10 zeigt diesen allgemeinen Aufbau.

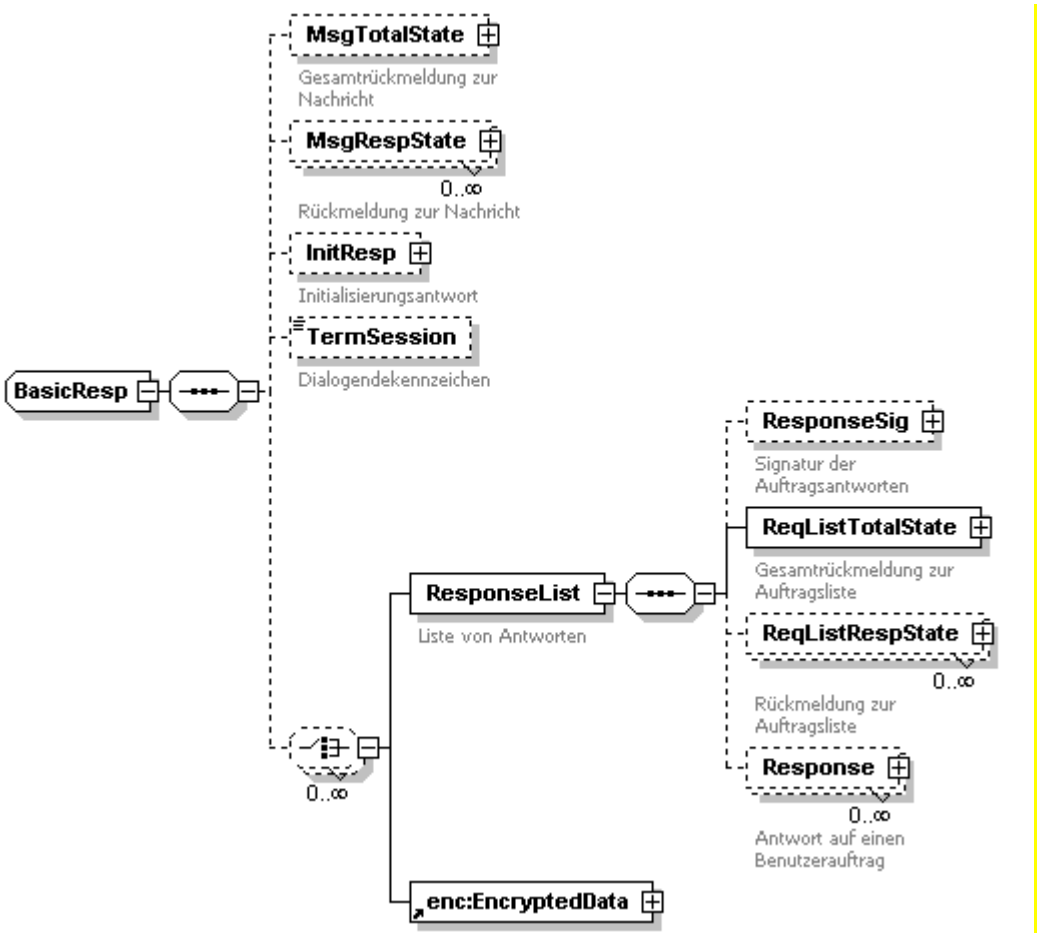


Abbildung 10: Basistyp für die Nachrichtenkörper der Kreditinstitutsnachricht

Die Nachrichten enthalten im Nachrichtenkörper beliebig viele Antwortlisten (*ResponseList*), die mit den Auftragslisten (*RequestList*) der Benutzernachricht korrespondieren und die Antworten zu den einzelnen Aufträgen transportieren. Verschlüsselung, Komprimierung und Signatur der Auftragsantworten sind analog zu *III.2.1 Benutzernachricht* modelliert.

Eine Antwortliste stellt die kleinste syntaktische Einheit in einer Kreditinstitutsnachricht dar, die [HBCI]-konform, unabhängig vom äußeren Kontext verarbeitbar ist.

Es ist wichtig, dass die Reihenfolge der Elemente *ResponseList* der Reihenfolge der korrespondierenden Elemente *RequestList* in der Benutzernachricht entspricht, damit dem Empfänger eine Zuordnung der Auftragsantworten möglich ist.

Die Initialisierung der Benutzernachricht wird in der Kreditinstitutsnachricht mit einer Initialisierungsantwort (*InitResp*) beantwortet, sofern zurück zu meldende Inhalte vorliegen.

Rückmeldungen zur Gesamtnachricht werden in den Elementen *MsgRespState* und *MsgTotalState* geliefert. Die Antwortlisten enthalten Rückmeldungen zu den Auftragsteilen sowie zu jedem Auftrag eine Antwort (*Response*).

Mit dem Element *TermSession* zeigt das Kreditinstitut analog zur Benutzernachricht das Ende eines synchronen Dialogs an. Das Kreditinstitut spiegelt dabei das Element *TermSession* aus der Benutzernachricht wieder. Davon abweichend wird das Element immer im Falle eines Dialogabbruches (siehe [Formals], Abschnitt *II.16 Dialogabbruchnachricht*) durch das Kreditinstitut gesetzt.

III.2.2.3 Initialisierungsantwort

Die Rückmeldung des Kreditinstituts auf die Initialisierung des Benutzers enthält

- aktuelle Bankparameterdaten, wenn der Benutzer keine oder eine veraltete Version gemeldet hat (*BankParamData*).
- aktuelle User-Parameterdaten, wenn der Benutzer keine oder eine veraltete Version gemeldet hat (*UserParamData*).
- im Sicherheitsverfahren HBCI den - für das verwendete Sicherheitsprofil gültigen - aktuellen öffentlichen Verschlüsselungsschlüssel des Kreditinstituts (*PublicKey*), wenn der Benutzer in seiner Nachricht einen veralteten Schlüssel verwendet hat.
- Kreditinstitutsmeldungen (*BankMessage*) (freie Textinformationen des Instituts)

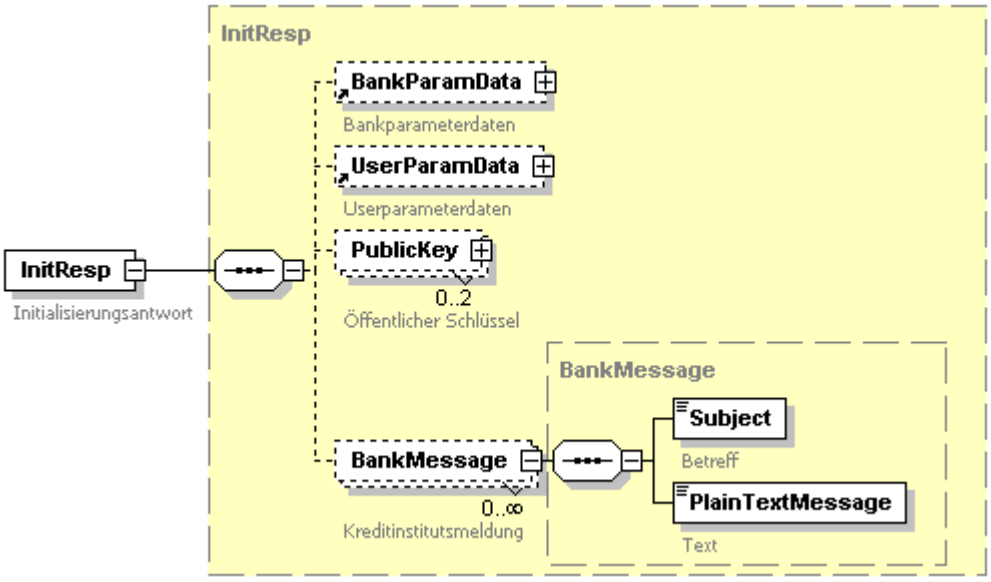


Abbildung 11: Initialisierungsantwort

III.2.2.4 Gesamtrückmeldung zur Nachricht

Das Element *MsgTotalState* enthält die Gesamtrückmeldung zur Nachricht (siehe auch [Formals], Abschnitt *II.10.1 Rückmeldungen zur Nachricht*). Es besitzt das Inhaltsmodell des Typs *TotalState*:

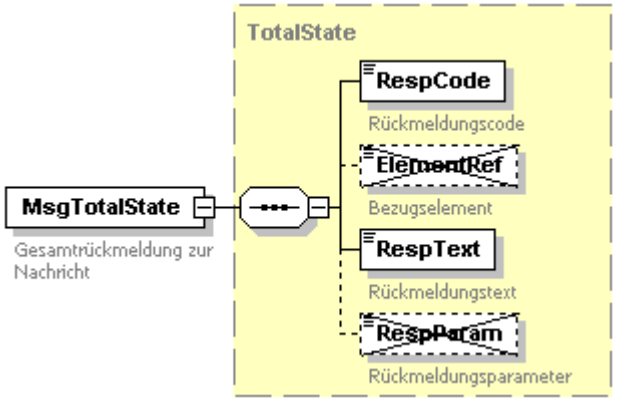


Abbildung 12: Gesamtrückmeldung zur Nachricht

RespCode und *RespText* enthalten den Status der Nachricht.

III.2.2.5 Rückmeldungen zur Nachricht

Das Element *MsgRespState* enthält eine Rückmeldung zur Nachricht (siehe auch [Formals], Abschnitt *II.10.1 Rückmeldungen zur Nachricht*). Es besitzt das Inhaltsmodell des Typs *ResponseState*:

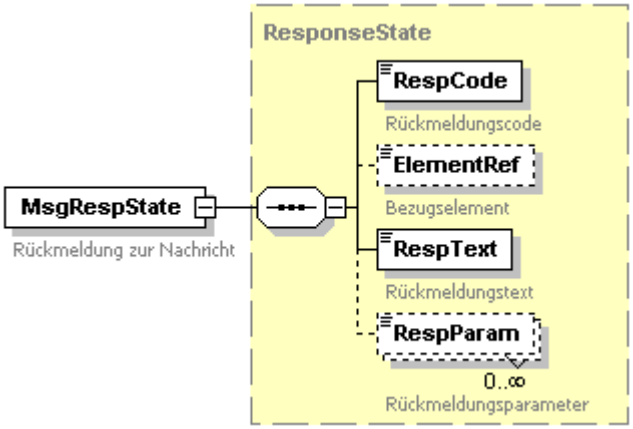


Abbildung 13: Rückmeldung zur Nachricht

RespCode und *RespText* enthalten eine Fehlerbeschreibung. In den *RespParam*-Elementen können zusätzliche Erläuterungen zu dieser Beschreibung aufgeführt werden (vgl. [Formals], Abschnitt *II.11 Rückmeldungscodes*).

Durch die Angabe eines XPath-Ausdrucks im Element *ElementRef* kann in der Kreditinstitutsnachricht ein exakter Bezugspunkt in einer Benutzernachricht referenziert werden. Diese Referenz ist bei Elementrückmeldungen zu befüllen,

nicht aber bei Meldungen zur Gesamtnachricht (siehe dazu [Formals], Abschnitt *II.10.1 Rückmeldungen zur Nachricht*).

Neben den Festlegungen in Abschnitt *II.8 Referenzierung mit XPath-Ausdrücken* gilt für die Belegung des Elements *ElementRef*:

Referenz-Ziel	Ein Element der Benutzernachricht, auf die sich die Kreditinstitutsnachricht bezieht.
Gültigkeitsbereich	Dokumentenelement der Benutzernachricht, auf die sich die Kreditinstitutsnachricht bezieht.
Kontext	Der Kontext zur Auswertung des Lokalisierungspfades ist der Wurzelknoten der Benutzernachricht, auf die sich die Kreditinstitutsnachricht bezieht.

Zulässiger Ausdruck	Der Ausdruck im Element *ElementRef* muss der Bildungsregel für *LocationPath* genügen:
	
	`LocationPath ::= AbsoluteLocationPath`
	`::= | RelativeLocationPath`

Beispiel:

```
<MsgRespState
 xmlns:fintstype=http://www.fints.org/spec/xmlschema/4.1/messages>
  <RespCode>9110</RespCode>

<ElementRef>/fintstype:ReqMsg[1]/fintstype:ReqMsgHeader[1]/fintstype:BankID[1]</Eleme
ntRef>
  <RespText>Unbekannter Aufbau</RespText>
</MsgRespState>
```

III.2.2.6 Gesamtrückmeldung zum Auftragsteil

Das Element *ReqListTotalState* enthält die Gesamtrückmeldung zur Auftragsliste (siehe auch [Formals], Abschnitt *II.10.2 Rückmeldungen zum Auftragsteil*). Es ist wie *MsgTotalState* nach dem Inhaltsmodell des Typs *TotalState* aufgebaut.

III.2.2.7 Rückmeldungen zum Auftragsteil

Das Element *ReqListRespState* enthält eine Rückmeldung zum Auftragsteil. Es ist nach dem in *III.2.2.5 Rückmeldungen zur Nachricht* beschriebenen Inhaltsmodell *ResponseState* aufgebaut.

Für die Verwendung dieser Rückmeldung und insbesondere die Element-Referenzierung mittels *ElementRef* siehe [Formals], Abschnitt *II.10.2 Rückmeldungen zum Auftragsteil*. Für *ElementRef* gelten darüber hinaus die Festlegungen in Abschnitt *II.8 Referenzierung mit XPath-Ausdrücken* sowie:

Referenz-Ziel	Ein Element der Benutzernachricht, auf die sich die Kreditinstitutsnachricht bezieht.
Gültigkeitsbereich	Wurzelelement der RequestList, auf die sich die Rückmeldung bezieht.
Kontext	Wurzelelement der RequestList, auf die sich die Rückmeldung bezieht.
Zulässiger Ausdruck	Der Ausdruck in *XPath* muss der Bildungsregel für *RelativeLocationPath* genügen.

Beispiel:

```
<ResponseList xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ReqListRespState>
    <RespCode>9050</RespCode>
    <RespText>Teilweise fehlerhaft</RespText>
  </ ReqListRespState >
  <ReqListRespState>
    <RespCode>9330</RespCode>
    <ElementRef>ds:Signature[1]</ElementRef>
    <RespText>Signatur gesperrt</RespText>
  </ ReqListRespState >
  <Response>
...
  </Response>
</ResponseList>
```

III.2.2.8 Auftragsantwort

Die Antwort zu einem Auftrag (*Response*) enthält in der Standard-Kreditinstitutsnachricht und in der anonymen Kreditinstitutsnachricht ein Element mit freiem Inhaltsmodell zur Aufnahme der Antwortdaten von Abholaufträgen. Für Geschäftsvorfälle, die solche Antwortdaten liefern, muss das Antwortelement im gleichen Schema modelliert sein und zum selben Namensraum gehören wie der Auftrag (vgl. [Messages] und *III.7 Administrative Aufträge*).

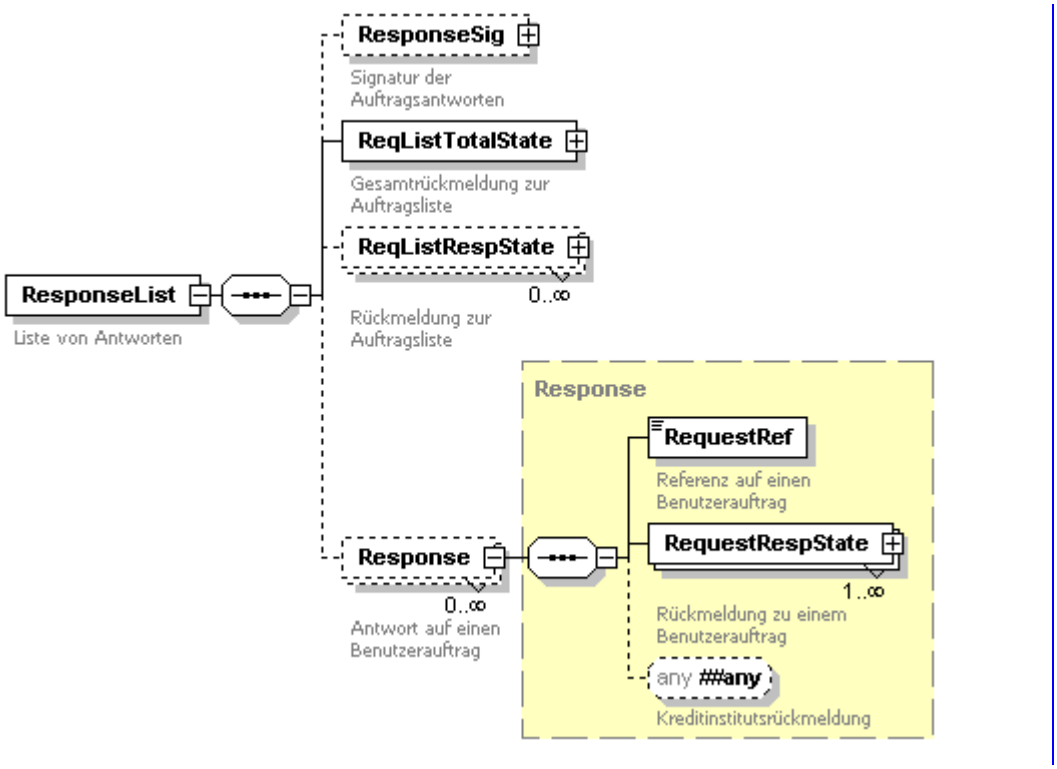


Abbildung 14: Auftragsantwort

Für die administrativen Nachrichtentypen ist das Format der Antwort fest vorgegeben.

Das Element *RequestRef* ist eine Referenz auf den Benutzerauftrag. Hier ist ein XPath-Ausdruck angegeben, der die Position des zugehörigen Auftrags im Auftragsdokument relativ zu seiner *RequestList* angibt. Es gelten die folgenden Regelungen:

Referenz-Ziel	Wurzelelement des Benutzerauftrags, auf den sich die *Response* bezieht.
Gültigkeitsbereich	Wurzelelement der Liste von Aufträgen (*RequestList*), auf die sich die Liste der Antworten (*ResponseList*) bezieht.
Kontext	Wurzelelement der Liste von Aufträgen (*RequestList*), auf die sich die Liste der Antworten (*ResponseList*) bezieht.
Zulässiger Ausdruck	Der Ausdruck in *XPath* muss der Bildungsregel für *RelativeLocationPath* genügen.

Die Antwort enthält außerdem eine Rückmeldung *RequestRespState*, die nach dem in *III.2.2.5 Rückmeldungen zur Nachricht* beschriebenen Inhaltsmodell *ResponseState* aufgebaut ist. Belegungsrichtlinien hierfür finden sich in [Formals], Abschnitt *II.10.3 Rückmeldungen zu Aufträgen*.

Für die Belegung von *ElementRef* in *RequestRespState* gelten folgende Festlegungen:

Referenz-Ziel	Ein Element des Benutzerauftrags, auf den sich die Auftragsantwort bezieht.
Gültigkeitsbereich	Wurzelelement des Benutzerauftrags, auf den sich die Auftragsantwort bezieht.
Kontext	Wurzelelement des Benutzerauftrags, auf den sich die Auftragsantwort bezieht.
Zulässiger Ausdruck	Der Ausdruck in *XPath* muss der Bildungsregel für *RelativeLocationPath* genügen.

Beispiel:

```
<ResponseList
 xmlns:fintstype="http://www.fints.org/spec/xmlschema/4.1/types"
 xmlns:fintstrans="http://www.fints.org/spec/xmlschema/4.1/transactions">
  <Response>
    <RequestRef>fintstype:Orders[1]/fintstrans:SEPASingRemitt_1_Req[1]</RequestRef>
    <RequestRespState>
      <RespCode>9110</RespCode>
      <ElementRef>fintstrans:OrderingCustAccount[1]/fintstype:BankID[1]</ElementRef>
      <RespText>Unbekannter Aufbau</RespText>
    </RequestRespState>
  </Response>
</ResponseList>
```

III.3 Verschiedene Benutzer- und Antwortnachrichten

Dieser Abschnitt beschreibt vier der acht unterschiedlichen Nachrichtentypen: die Standard-Nachricht, die Anonyme Nachricht, die Lebendmeldung und die Synchronisierung. Die restlichen vier Nachrichtentypen sind in *III.4 Keymanagement-Nachrichten* zusammengefasst.

III.3.1 Standard-Nachricht

Die Standard-Nachricht ist derjenige Nachrichtentyp, der für den überwiegenden Teil der Kommunikation zwischen Benutzer und Kreditinstitut eingesetzt wird. In ihr können alle operativen Transaktions- und Abholaufträge aus [Messages] sowie alle administrativen Aufträge aus *III.7 Administrative Aufträge* transportiert werden. Die Verwendung der Auftrags- und Antwortteile, Verschlüsselung, Signatur und Initialisierung ist unter [Formals], Abschnitt *II.9 Benutzernachrichten allgemein* beschrieben.

Eine Standard-Kreditinstitutsnachricht kann auch als Antwort auf andere Nachrichtentypen verwendet werden, wenn in einem Fehlerfall die Benutzernachricht nicht interpretierbar ist. In solchen Fällen darf die Kreditinstitutsnachricht keine Auftragsantworten, sondern nur eine Fehlerrückmeldung zur Gesamtnachricht enthalten.

a) Benutzernachricht

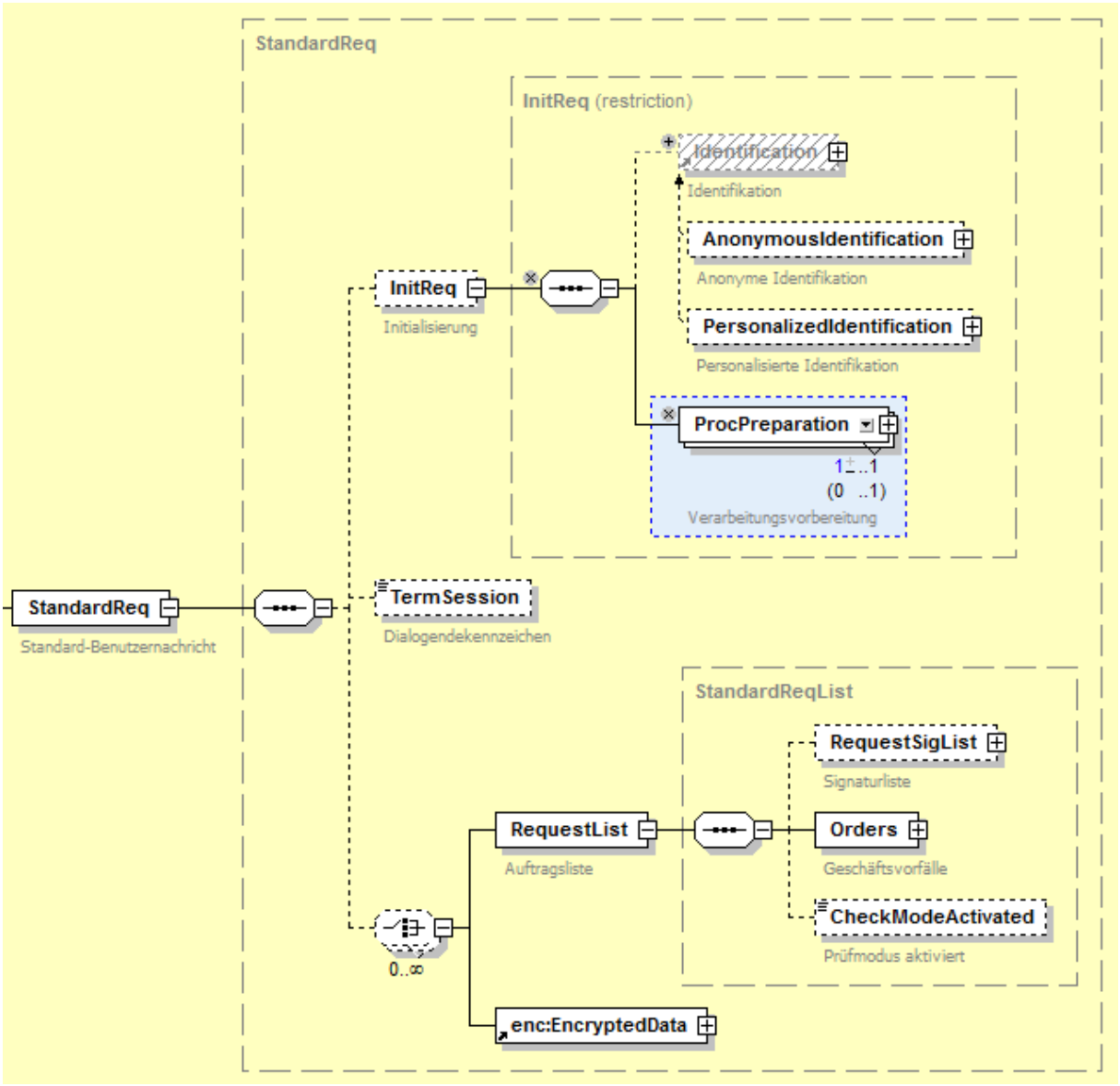


Abbildung 15: Standard-Benutzernachricht

b) Kreditinstitutsnachricht

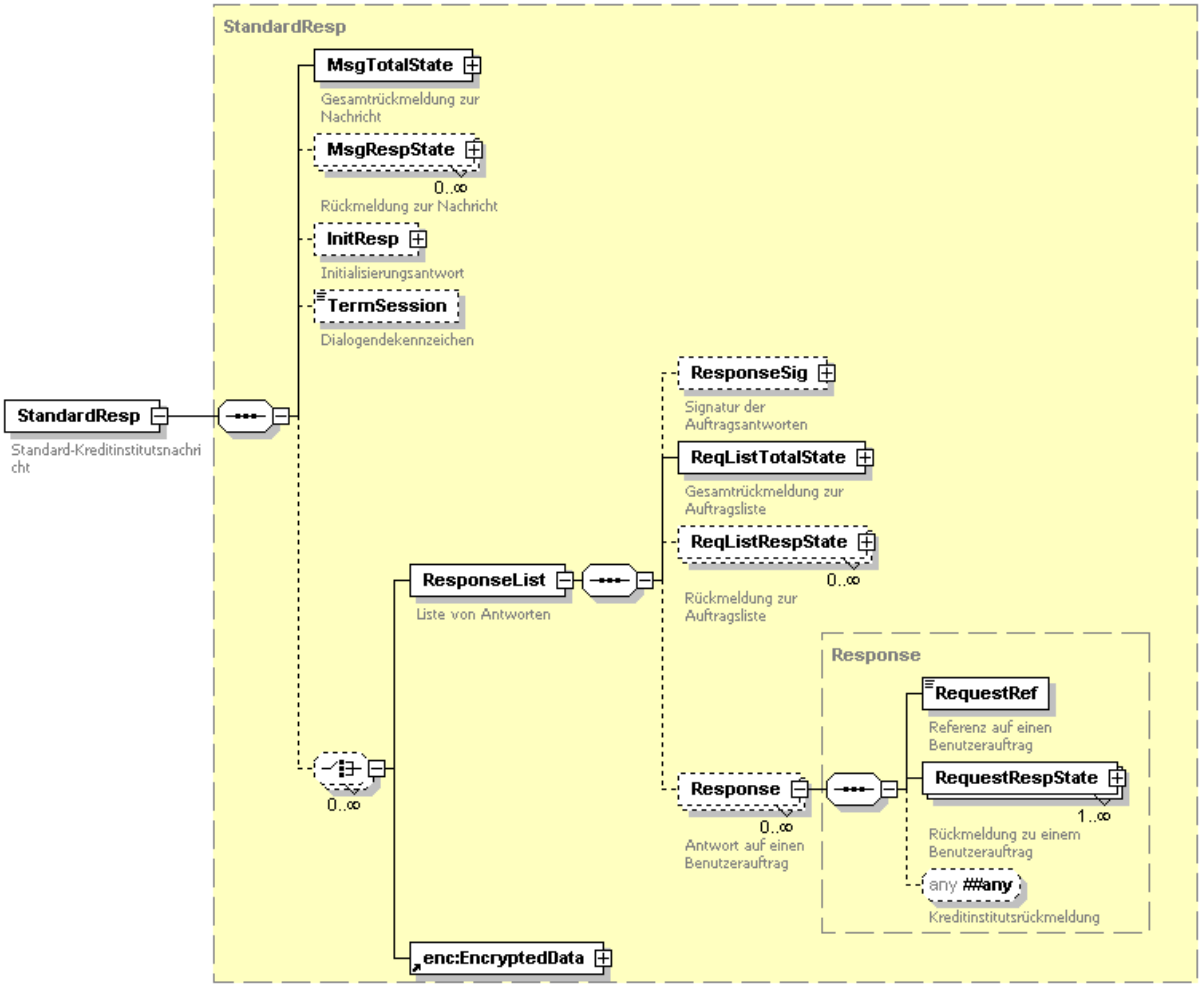


Abbildung 16: Standard-Kreditinstitutsnachricht

III.3.2 Anonyme Nachricht

Bei anonymer Kommunikation nach [Formals], Abschnitt *II.17 Anonymer Zugang* werden Nachrichten nicht verschlüsselt und nicht signiert, Komprimierung ist allerdings möglich. Der sonstige Aufbau entspricht der Standard-Nachricht.

a) Benutzernachricht

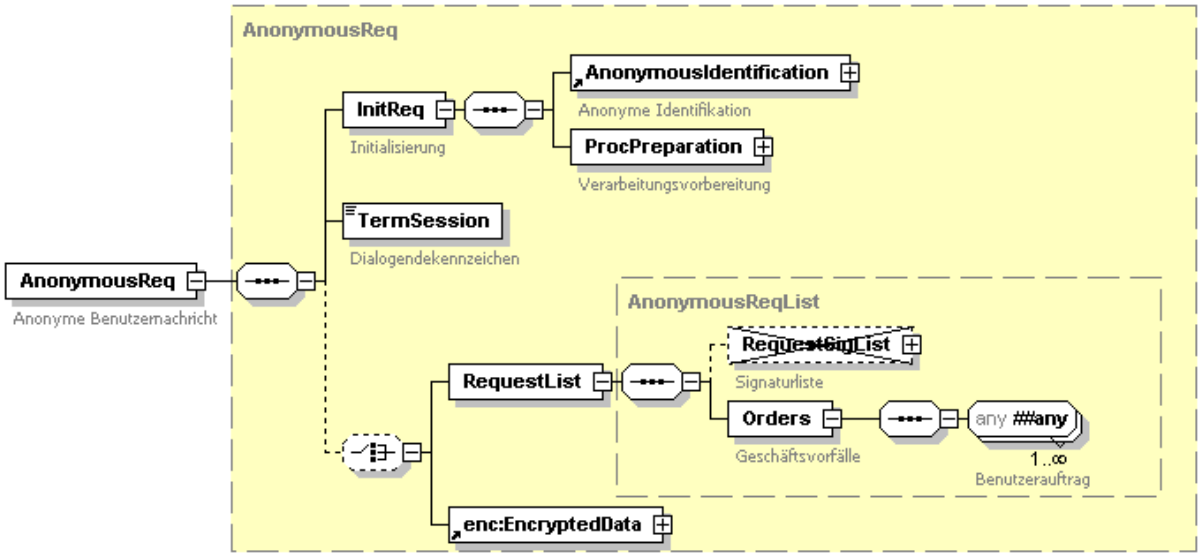


Abbildung 17: Anonyme Benutzernachricht

b) Kreditinstitutsnachricht

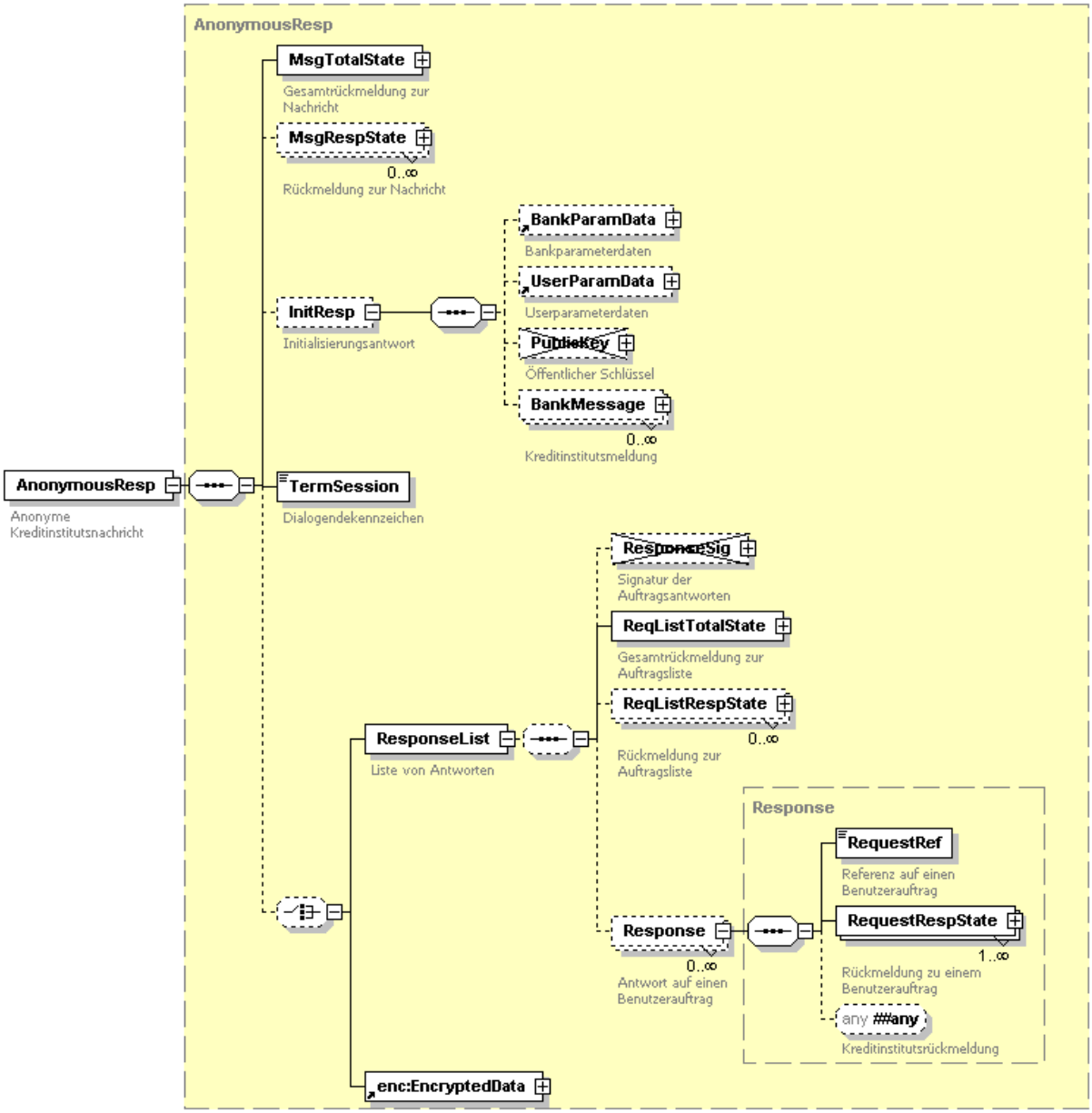


Abbildung 18: Anonyme Kreditinstitutsnachricht

Die Initialisierungsantwort der anonymen Kreditinstitutsnachricht enthält keine öffentlichen Kreditinstitutsschlüssel.

III.3.3 Lebendmeldung

Die Lebendmeldungsnachricht dient zur Vermeidung eines kreditinstitutsseitigen Verbindungsabbruchs bei Zeitüberschreitung in einem synchronen Dialog (siehe [Formals], Abschnitt *III.6 Lebendmeldung in Dialogen*. Der Nachrichtenkörper ist sowohl in der Benutzernachricht als auch in der Kreditinstitutsnachricht leer, sowohl Nachricht als auch Antwort sind anonym.

a) Benutzernachricht

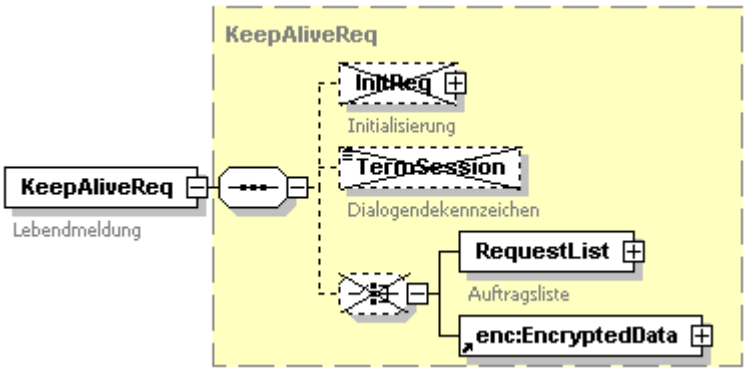


Abbildung 19: Benutzernachricht Lebendmeldung

Es gelten besondere Belegungsrichtlinie für den Nachrichtenkopf einer Lebendmeldung (nicht in der Abbildung gezeigt):

Nachrichtennummer

Ist mit der Nummer 0 zu belegen.

Textuelle Referenz des Benutzers

Bei dieser Nachricht muss die textuelle Referenz nicht eindeutig sein, da die Nachricht nicht im Statusprotokoll vermerkt wird.

b) Kreditinstitutsnachricht

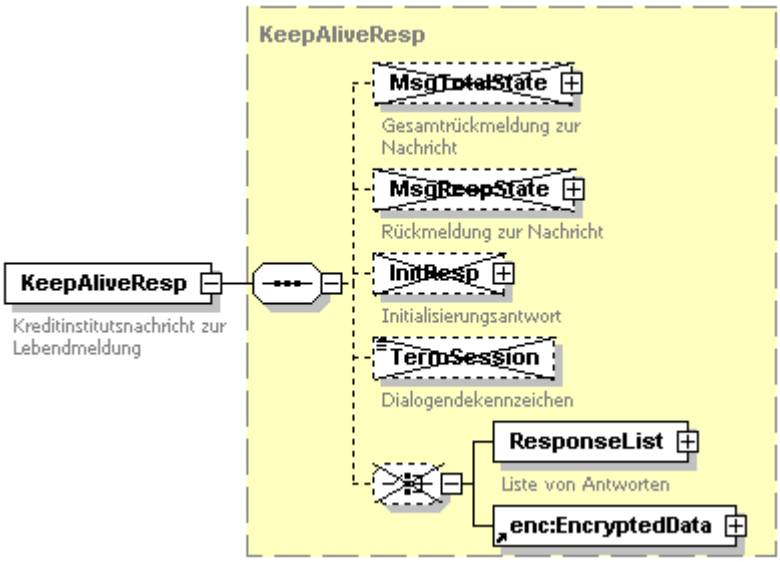


Abbildung 20: Kreditinstitutsnachricht zur Lebendmeldung

Diese Nachricht kann keine Rückmeldungen transportieren. Falls der Dialog bei Eintreffen der Lebendmeldung bereits geschlossen ist, antwortet das Institut mit einer Standard-Kreditinstitutsnachricht mit entsprechender Fehlermeldung.

III.3.4 Synchronisierung

Die Synchronisierung ([Formals], Abschnitt *III.3 Synchronisierung*) dient zum Abgleich verschiedener beim Benutzer und/oder beim Institut geführter Informationen und Zählerwerte.

a) Benutzernachricht

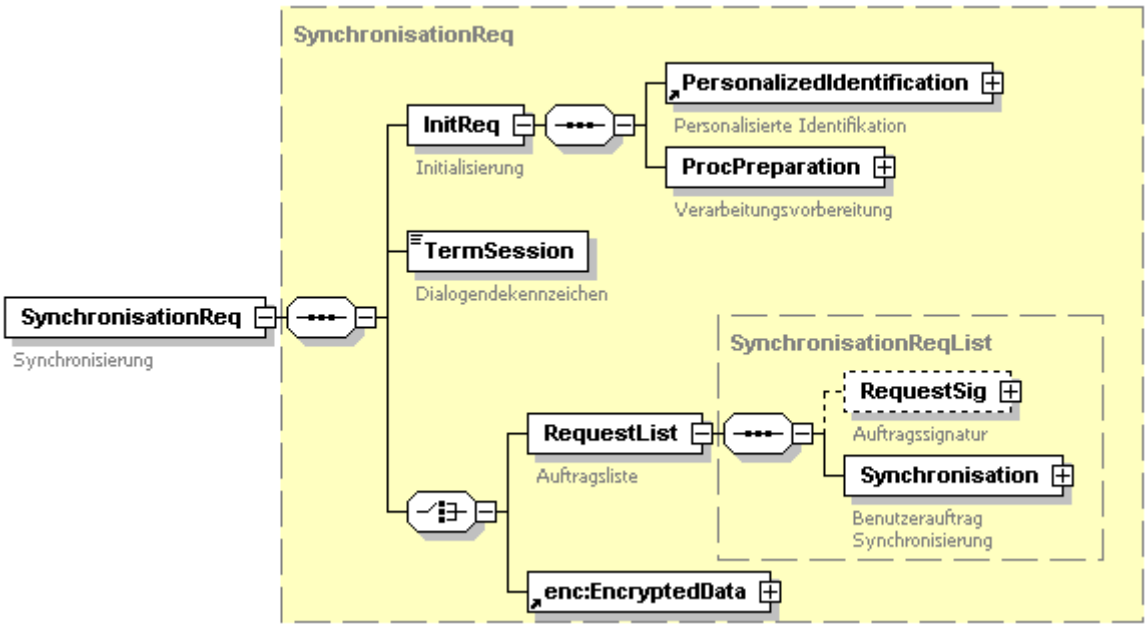


Abbildung 21: Benutzernachricht Synchronisierung

In die Auftragsliste kann ausschließlich der Auftragstyp „Benutzerauftrag Synchronisierung“ eingestellt werden:

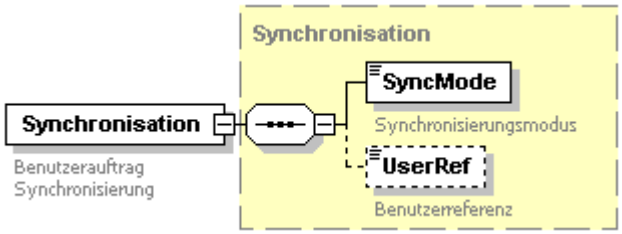


Abbildung 22: Benutzerauftrag Synchronisierung

Zu diesem Auftrag gibt es keine geschäftsvorfallspezifischen Parameter. Der Auftrag kann durch das Kreditinstitut im Rahmen der BPD nicht ausgeschlossen werden, in den UPD wird er nicht aufgeführt.

Benutzerreferenz

Das Element *UserRef* darf nur im Synchronisierungsmodus *SyncMode = 3* eingestellt werden.

b) Kreditinstitutsnachricht

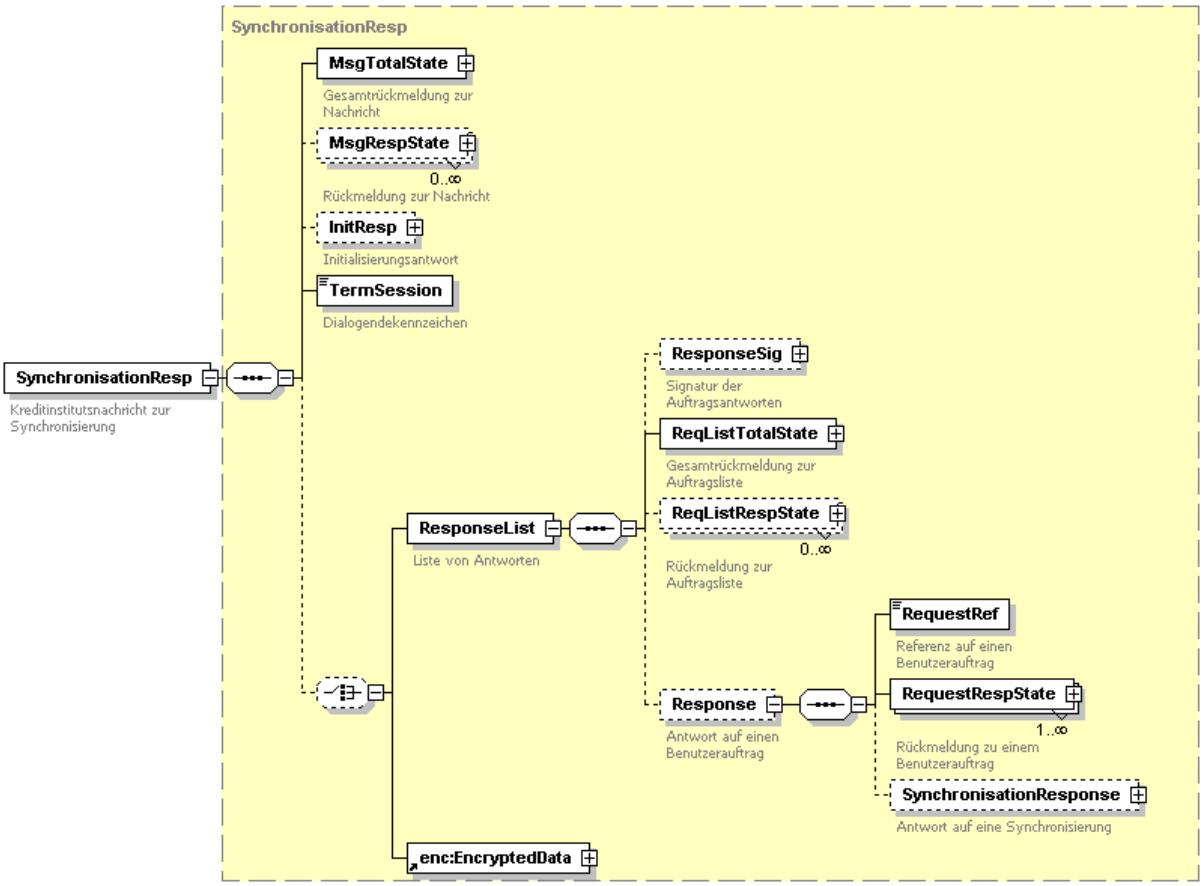


Abbildung 23: Kreditinstitutsnachricht zur Synchronisierung

In die Liste von Antworten kann ausschließlich der Auftragstyp „Antwort auf eine Synchronisierung“ eingestellt werden:

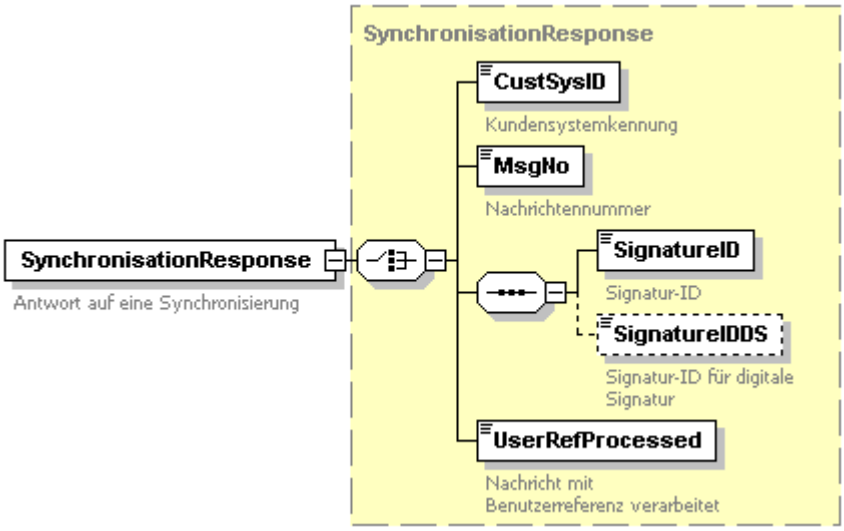


Abbildung 24: Antwort auf eine Synchronisierung

III.4 Keymanagement-Nachrichten

Das Sicherheitsverfahren [HBCI] kennt vier administrative Geschäftsvorfälle zur Schlüsselverwaltung. Da diese nicht mit anderen Geschäftsvorfällen in einem Dialog gemischt werden dürfen und da sie teilweise besondere Anforderungen an Signatur und Verschlüsselung stellen, sind in der FinTS-Syntax dafür vier spezielle Nachrichtentypen definiert. Jeder Auftrag des Keymanagements darf nur in seinem zugehörigen Nachrichtentyp transportiert werden, umgekehrt können diese Nachrichtentypen keine anderen Geschäftsvorfälle aufnehmen. Zu diesen administrativen Aufträgen gibt es keine geschäftsvorfallspezifischen Parameter. Sie können durch das Kreditinstitut im Rahmen der BPD nicht ausgeschlossen werden, in den UPD werden sie nicht aufgeführt.

III.4.1 Anforderung der Kreditinstitutsschlüssel

Diese Nachricht dient bei RAH-Verfahren dazu, die öffentlichen Schlüssel des Kreditinstituts zum Benutzer zu übertragen. Sowohl Auftrag als auch Antwort sind unverschlüsselt. Die Benutzernachricht ist unsigniert, die Kreditinstitutsnachricht ist mit dem im Auftrag übertragenen Schlüssel signiert, falls das Kreditinstitut seine Nachrichten signiert. Komprimierung ist in beiden Richtungen möglich. Siehe [HBCI], Abschnitt *II.6.1.2 Erstmalige Anforderung der Schlüssel des Kreditinstituts*.

a) Auftrag

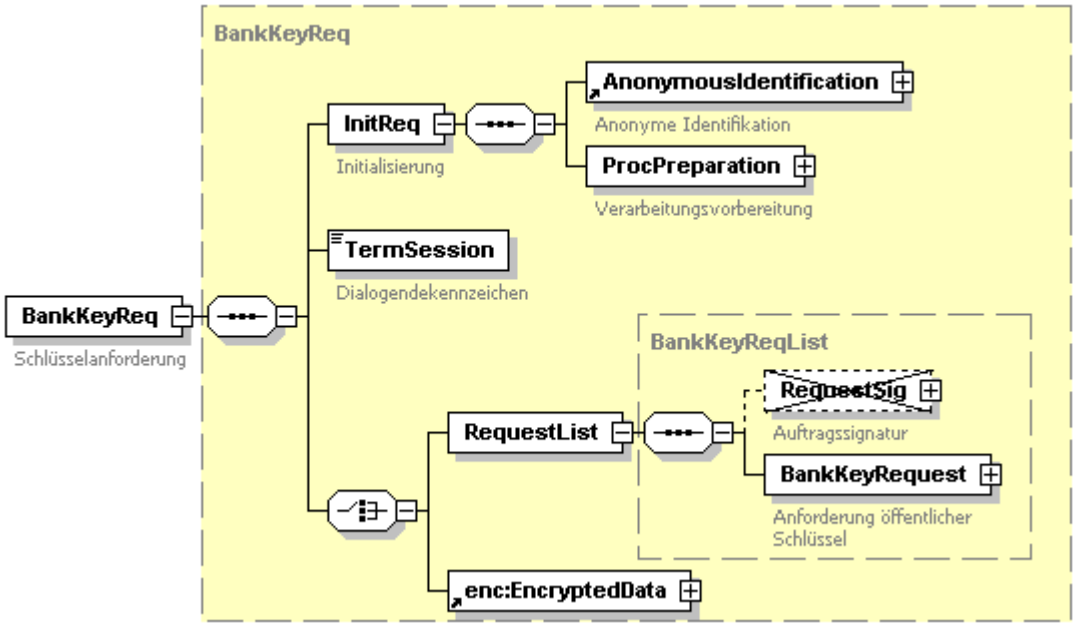


Abbildung 25: Benutzernachricht Anforderung der Kreditinstitutsschlüssel

In die Auftragsliste kann ausschließlich der Auftragstyp „Anforderung öffentlicher Schlüssel“ eingestellt werden:

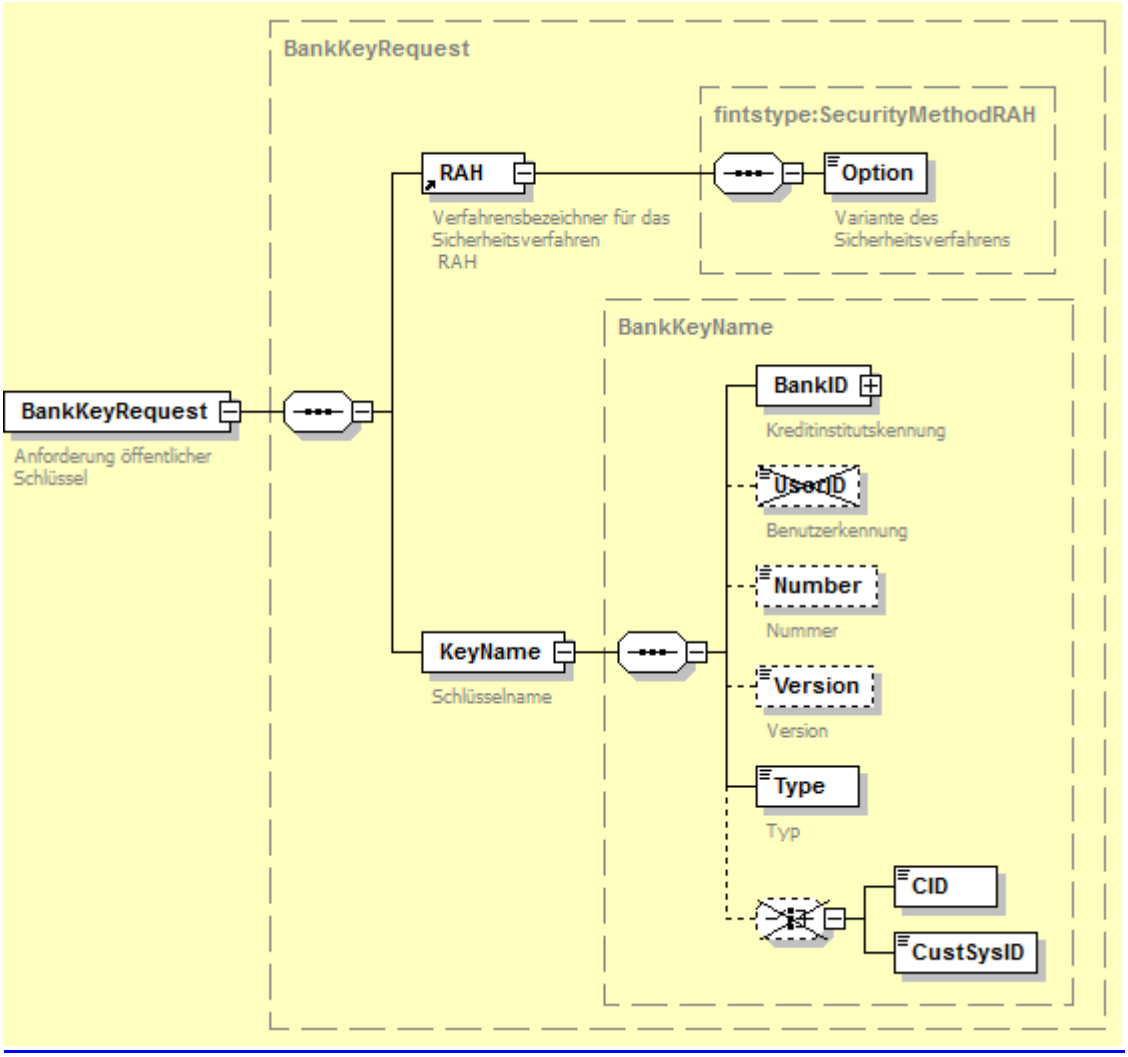


Abbildung 26: Anforderung öffentlicher Schlüssel

Schlüsselname

Wenn bekannt, werden hier Nummer und Version der aktuellen Kreditinstitutsschlüssel eingestellt. Werden mit Nummer und Version bereits abgelaufene Schlüssel bezeichnet, obliegt es dem Kreditinstitut, ob es diese oder die aktuellen Schlüssel sendet, siehe [HBCI], Abschnitt *II.6.1.2 Erstmalige Anforderung der Schlüssel des Kreditinstituts*.

Typ

Es wird immer ‚S’ für den Signierschlüssel angegeben. Das Kreditinstitut sendet in der Antwort einen oder beide Schlüssel, siehe auch [HBCI], Abschnitt *II.6.1.2 Erstmalige Anforderung der Schlüssel des Kreditinstituts*. Über den bei RAH-7 vorhandenen dritten Schlüssel für die digitale Signatur verfügt das Kreditinstitut nicht.

b) Antwort

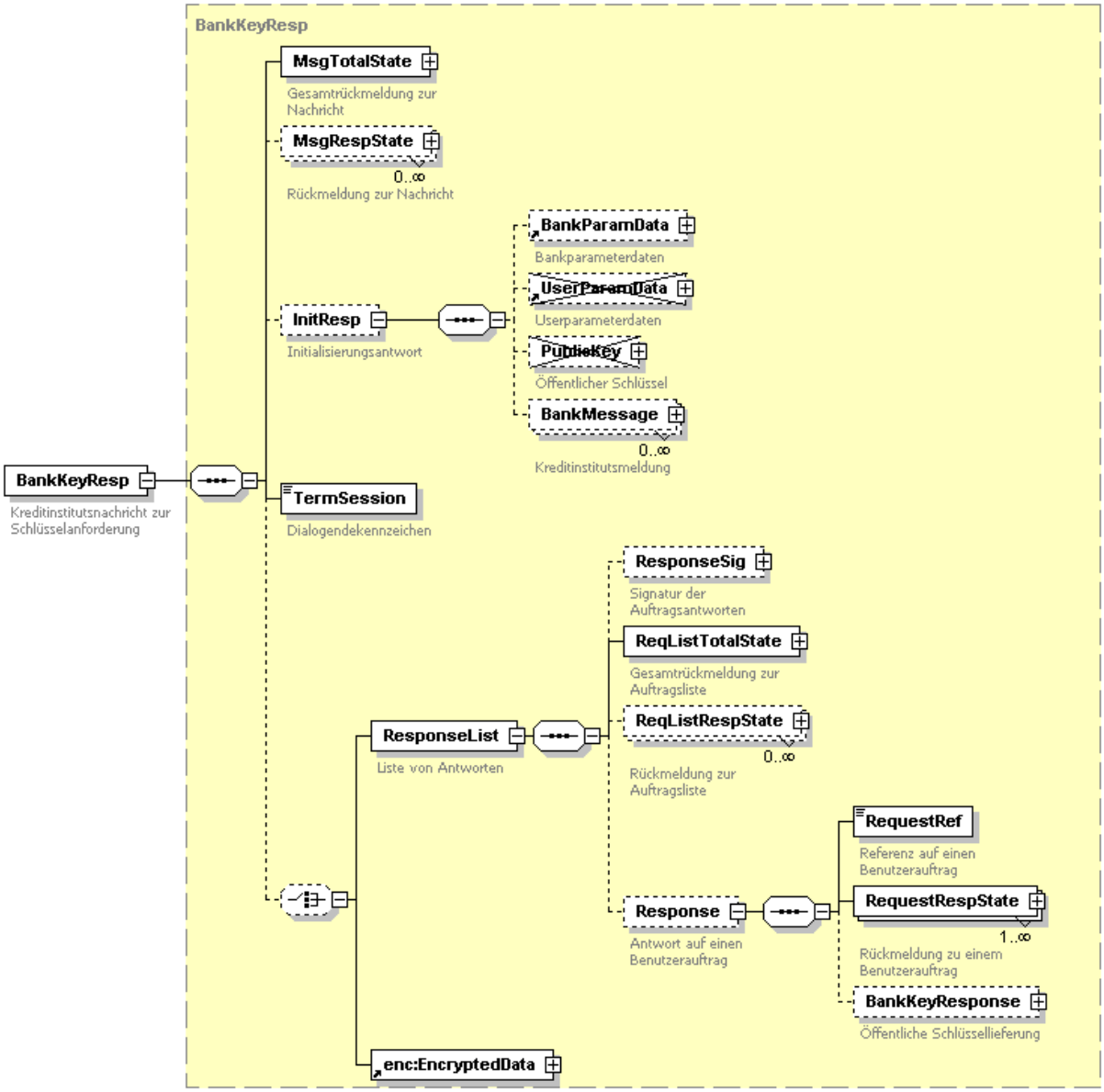


Abbildung 27: Kreditinstitutsnachricht Anforderung der Kreditinstitutsschlüssel

Die Initialisierungsantwort der Kreditinstitutsnachricht enthält keine User-Parameterdaten und keine öffentlichen Kreditinstitutsschlüssel.

In die Liste von Antworten kann ausschließlich der Auftragstyp „Öffentliche Schlüssellieferung“ eingestellt werden:

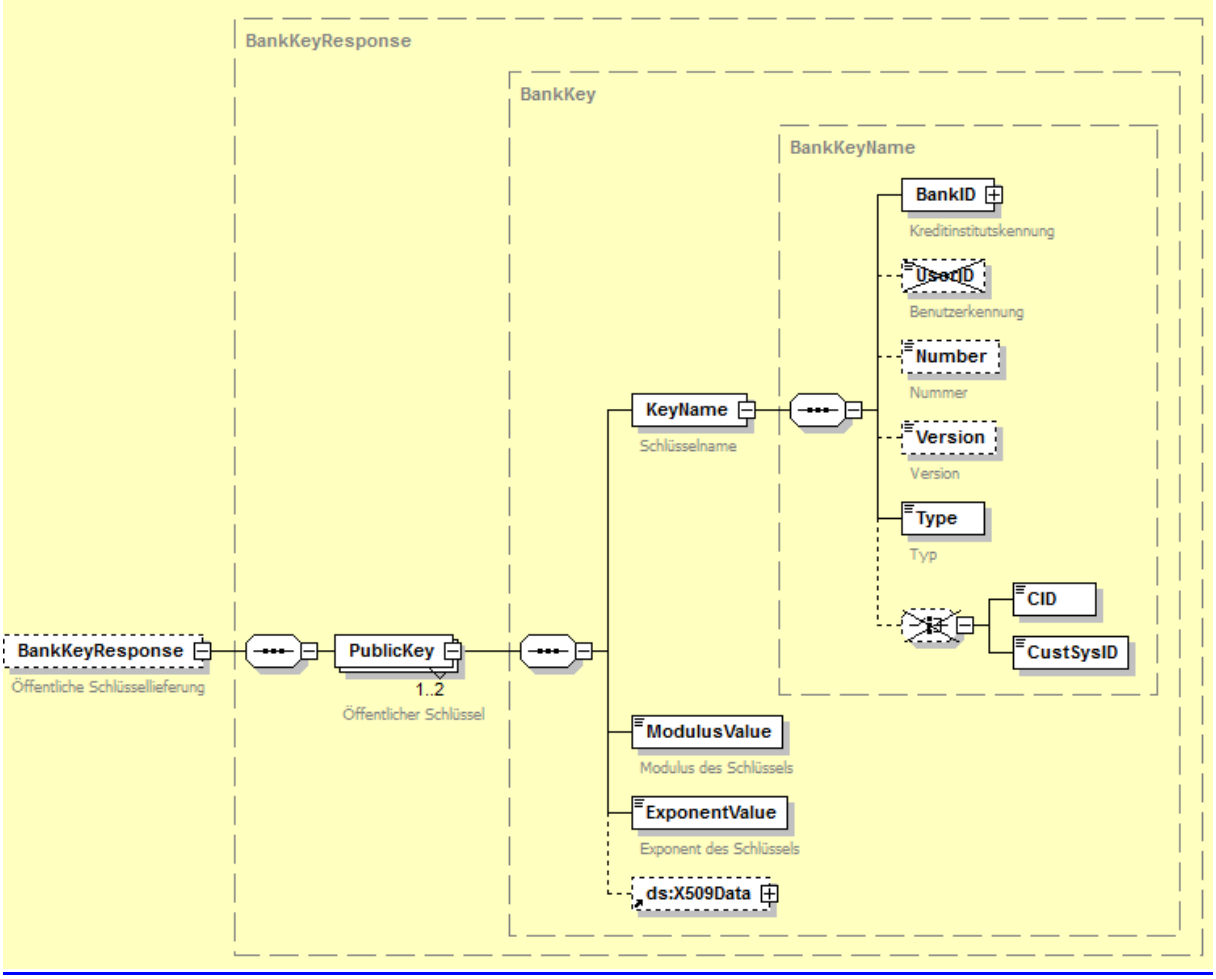


Abbildung 28: Öffentliche Schlüssellieferung

III.4.2 Erstmalige Übermittlung eines Kundenschlüssels

Mit einer Nachricht dieses Typs übermittelt der Benutzer im Rahmen der RAH-Erstinitialisierung seine öffentlichen Schlüssel an das Kreditinstitut. Die Benutzernachricht ist verschlüsselt und bereits mit dem übertragenen Signierschlüssel signiert, die Kreditinstitutsantwort ist im Erfolgsfall signiert, aber nicht verschlüsselt. Komprimierung ist in beiden Richtungen möglich. Siehe [HBCI], Abschnitt *II.6.1.3 Erstmalige Übermittlung der Schlüssel des Benutzers*.

a) Auftrag

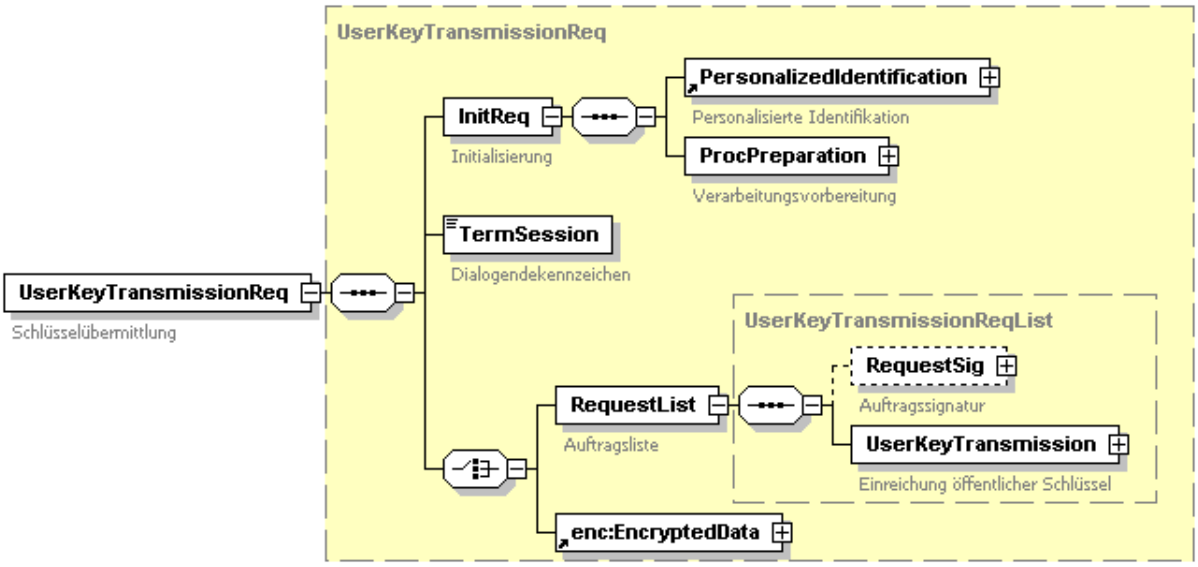


Abbildung 29: Benutzernachricht Übermittlung eines Kundenschlüssels

In die Auftragsliste kann ausschließlich der Auftragstyp „Einreichung öffentlicher Schlüssel“ eingestellt werden:

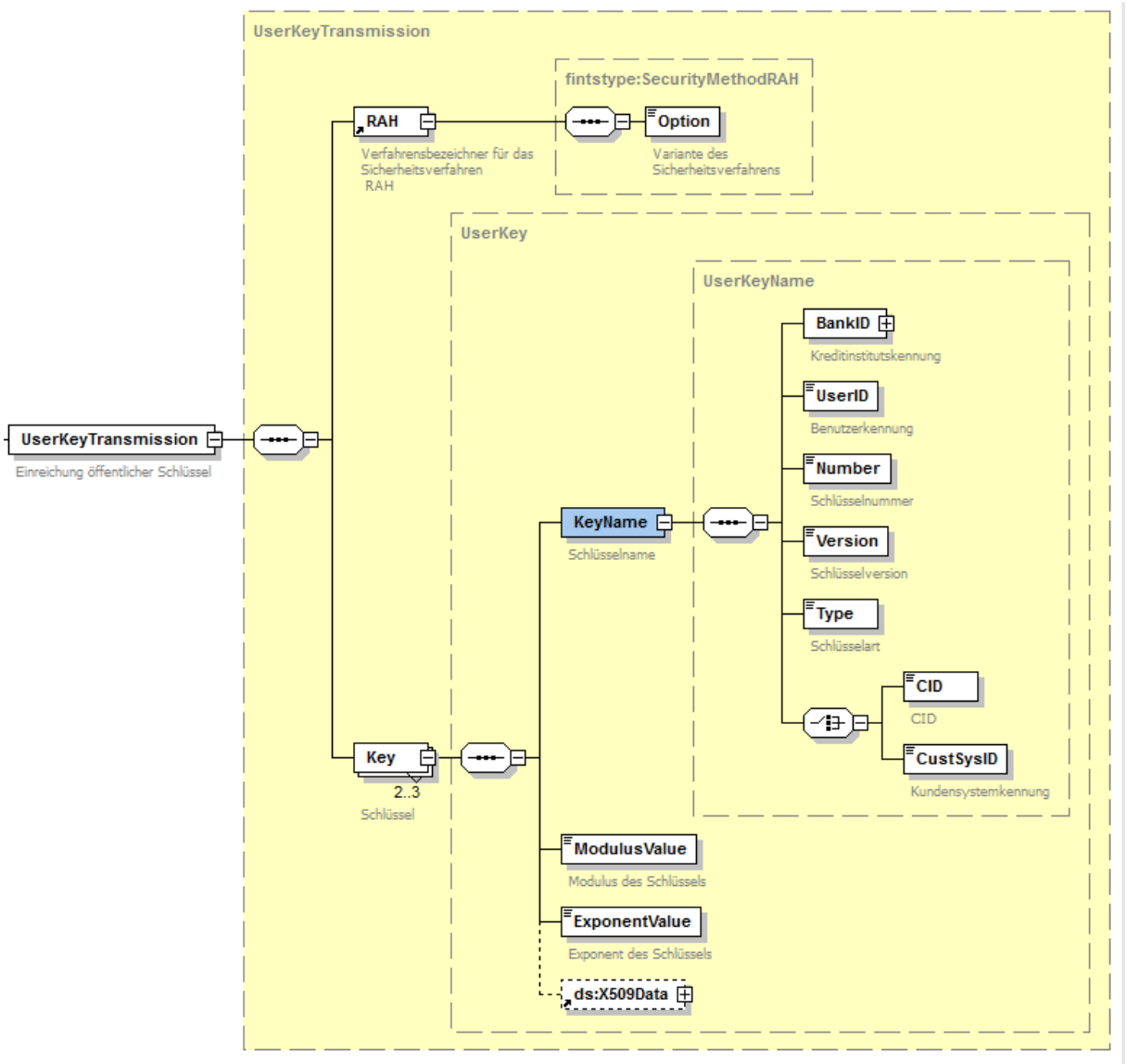


Abbildung 30: Einreichung öffentlicher Schlüssel

b) Antwort

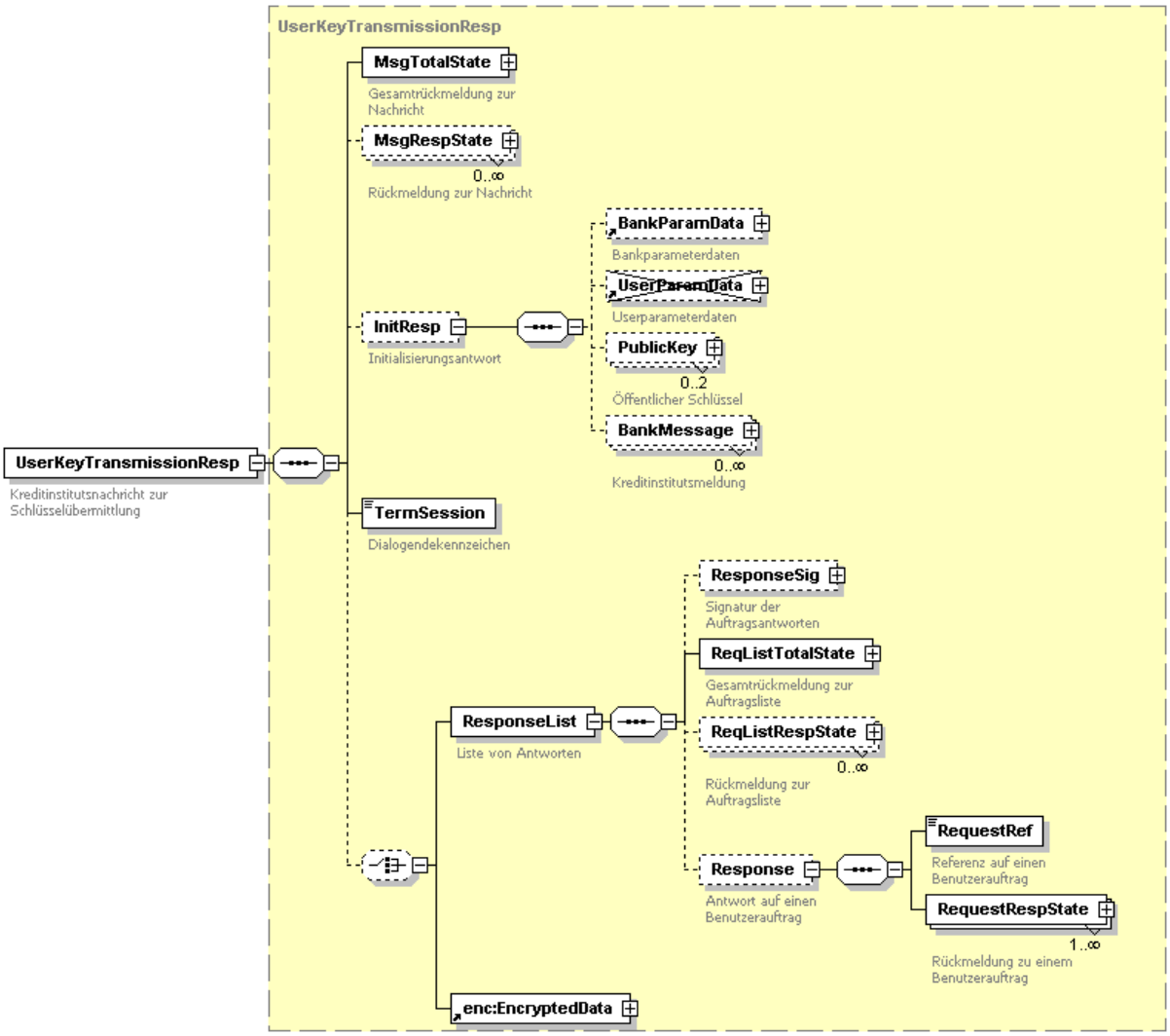


Abbildung 31: Kreditinstitutsnachricht zur Übermittlung eines Kundenschlüssels

Die Initialisierungsantwort der Kreditinstitutsnachricht enthält keine User-Parameterdaten.

Die Antwortnachricht meldet außer dem *RequestRespState* keine weiteren Daten zurück.

III.4.3 Schlüsseländerung

Bei RAH-Verfahren kann der Benutzer mit diesem Nachrichtentyp seine Schlüssel wechseln. Beide Nachrichten sind personalisiert, insbesondere ist die Antwort im Erfolgsfall bereits mit dem neuen Benutzerschlüssel verschlüsselt. Komprimierung ist in beiden Richtungen möglich. Siehe [HBCI], Abschnitt *II.6.1.1 Änderung eines öffentlichen Schlüssels des Benutzers*.

a) Auftrag

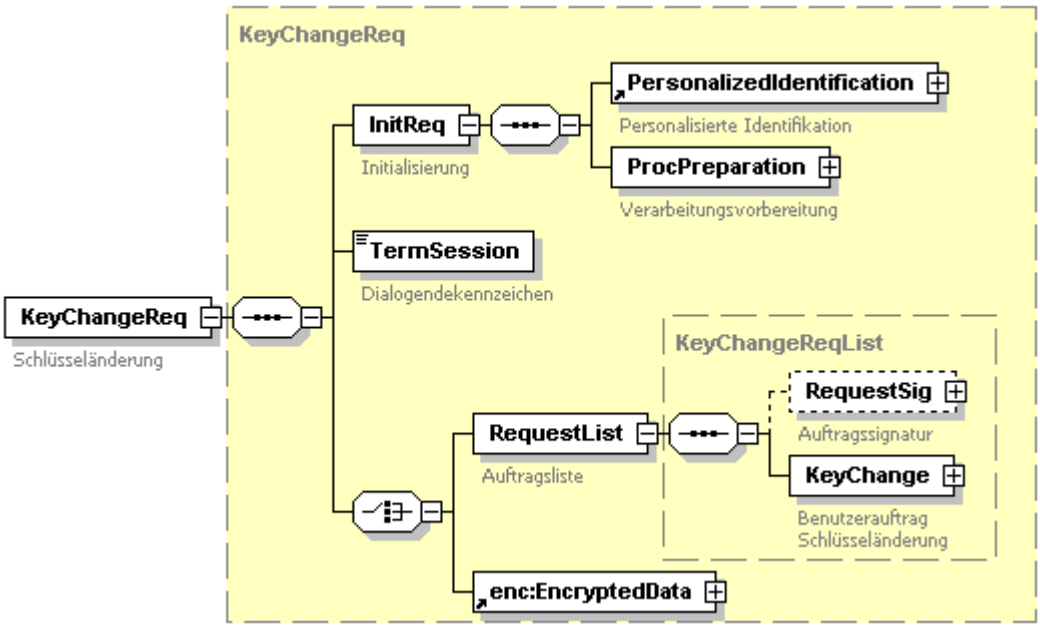


Abbildung 32: Benutzernachricht Schlüsseländerung

In die Auftragsliste kann ausschließlich der Auftragstyp „Benutzerauftrag Schlüsseländerung“ eingestellt werden:

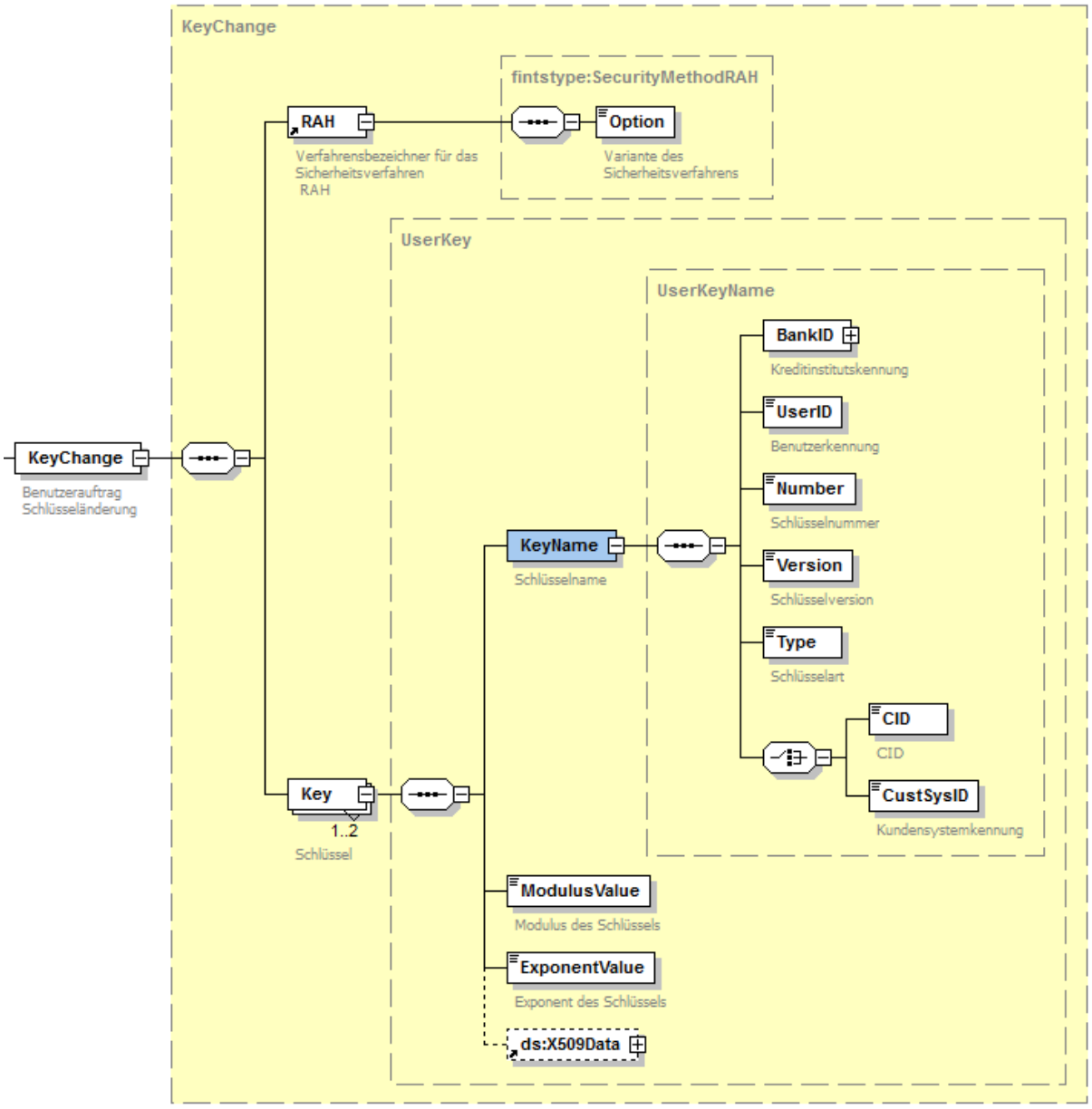


Abbildung 33: Benutzerauftrag Schlüsseländerung

RAH

Verfahren, für das der neue Schlüssel (die neuen Schlüssel) gilt (gelten).

Schlüssel

Es können bis zu zwei zu ändernde Schlüssel gleichzeitig angegeben werden. Der im Verfahren RAH-7 definierte zusätzliche Schlüssel für digitale Signatur kann nicht geändert werden.

b) Antwort

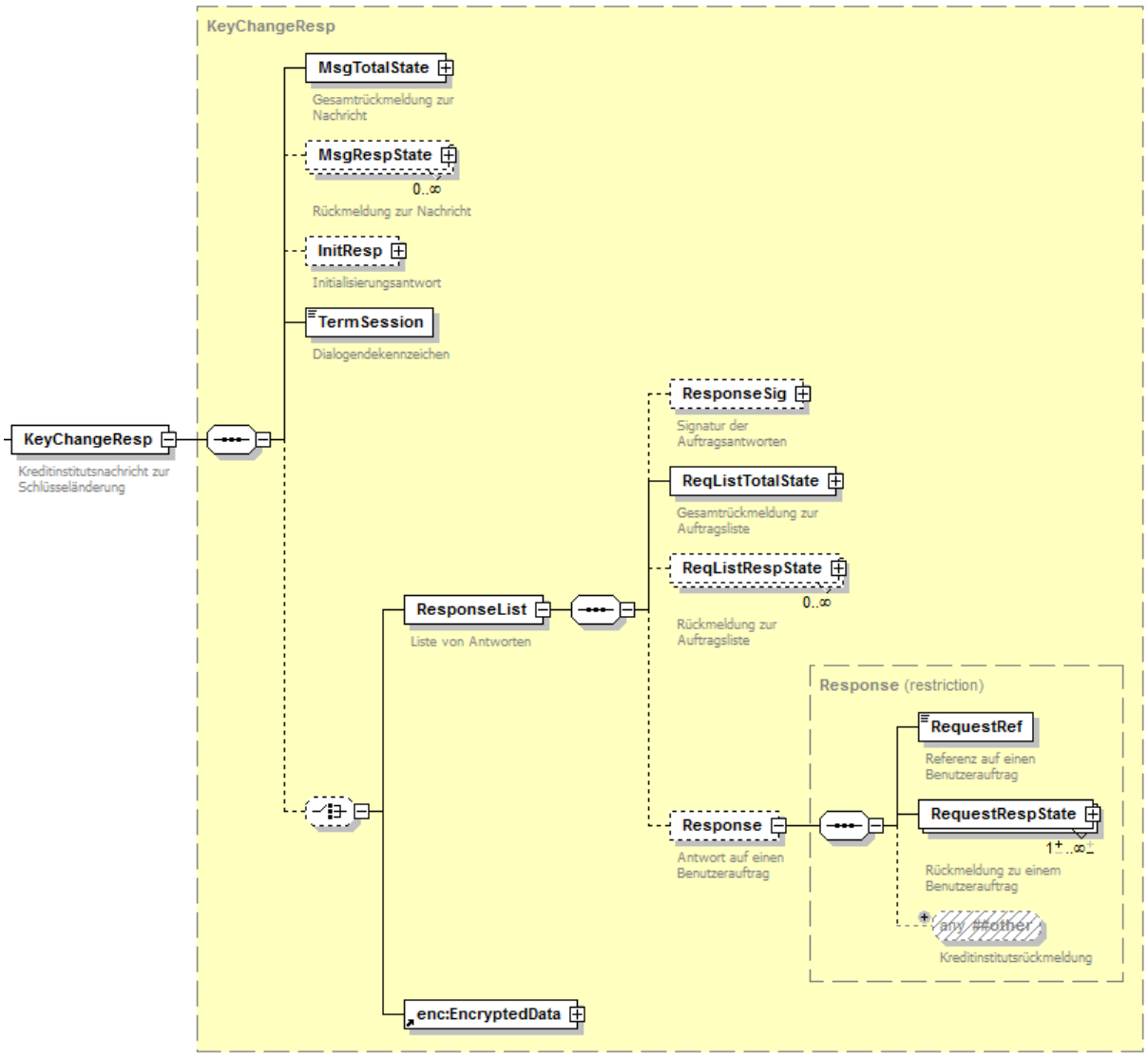


Abbildung 34: Kreditinstitutsnachricht zur Schlüsseländerung

Die Antwortnachricht meldet außer dem *RequestRespState* keine weiteren Daten zurück.

III.4.4 Schlüsselsperrung

Mit dieser Nachricht kann ein Benutzer im RAH-Verfahren seine Schlüssel sperren. Die Benutzernachricht ist personalisiert, die Kreditinstitutsnachricht ist bei erfolgreicher Sperrung je nach Sicherheitsverfahren teilweise oder vollständig anonym. Komprimierung ist in beiden Richtungen möglich. Siehe [HBCI], Abschnitt *II.6.1.4 Schlüsselsperrung durch den Benutzer*.

a) Auftrag

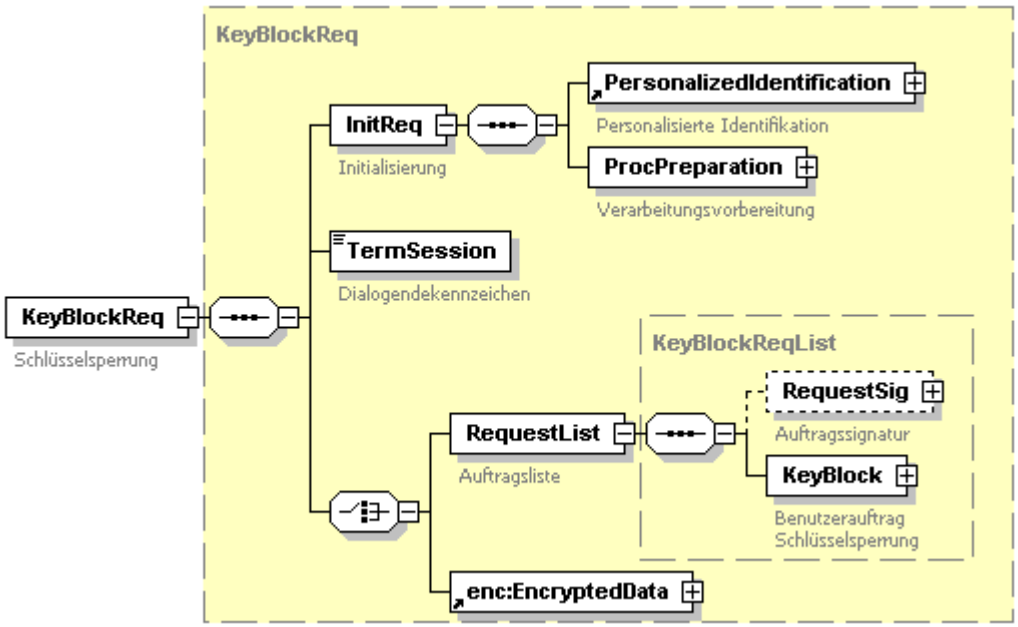


Abbildung 35: Benutzernachricht Schlüsselsperrung

In die Auftragsliste kann ausschließlich der Auftragstyp „Benutzerauftrag Schlüsselsperrung“ eingestellt werden:

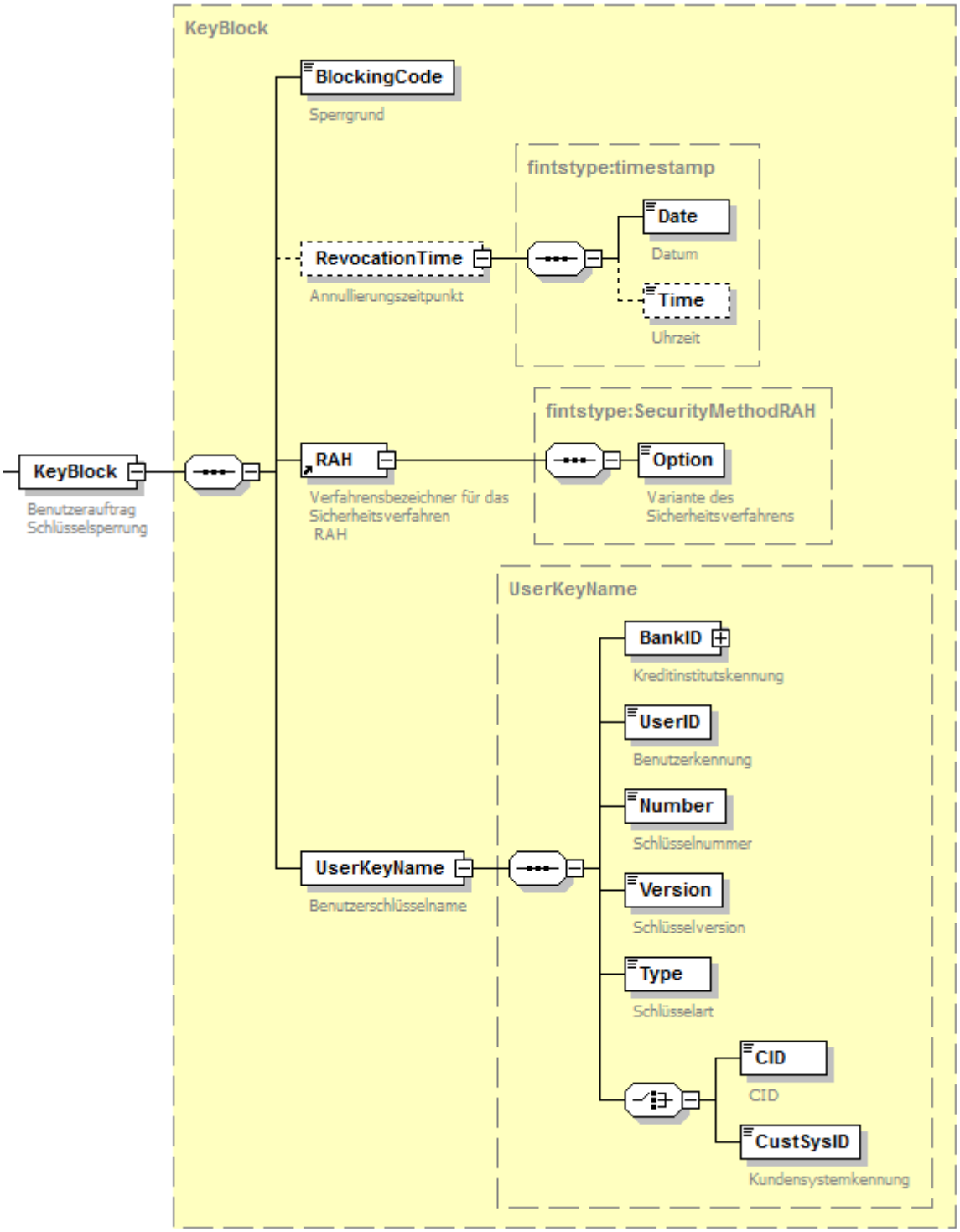


Abbildung 36: Benutzerauftrag Schlüsselsperrung

b) Antwort

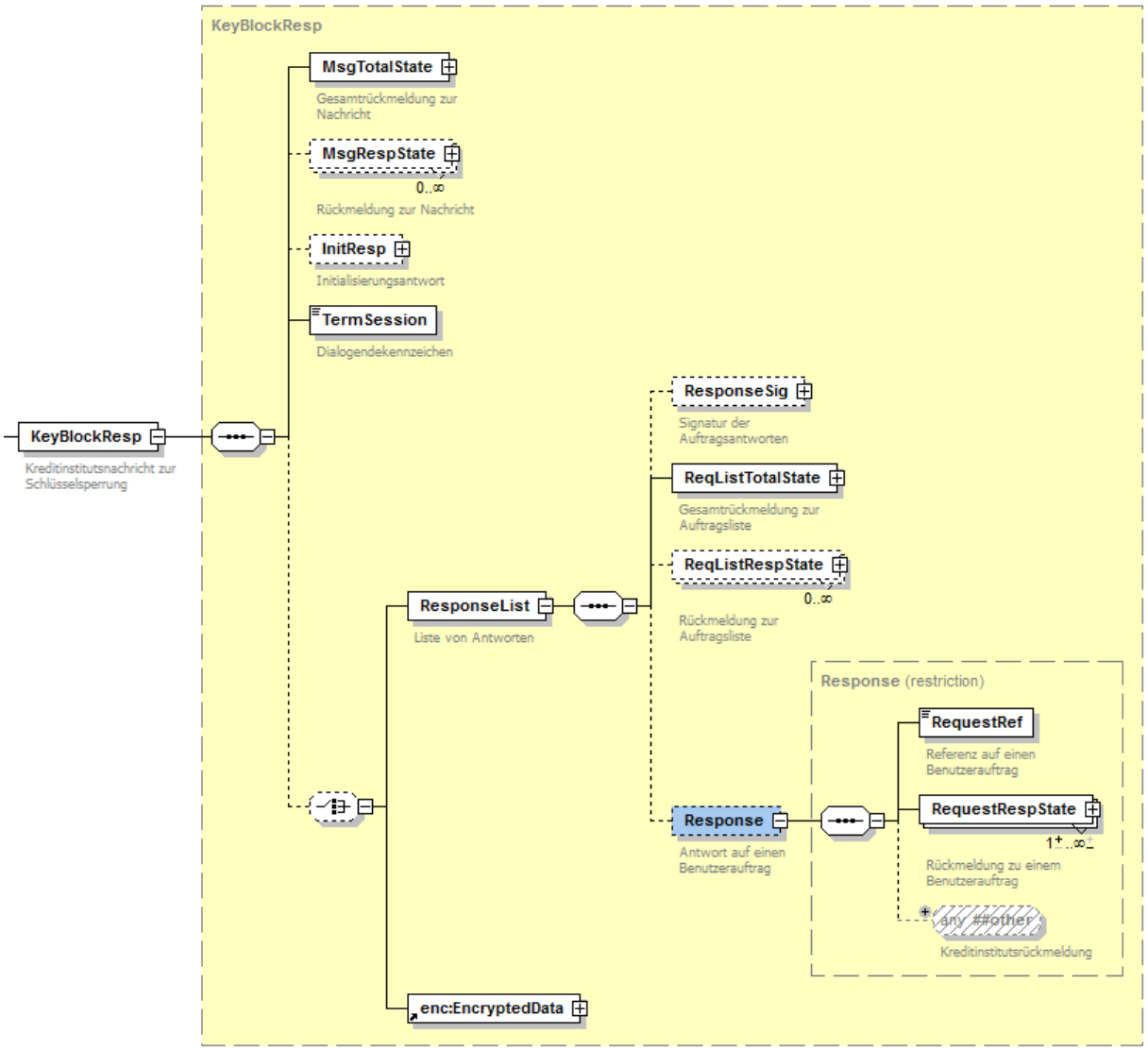


Abbildung 37: Kreditinstitutsnachricht zur Schlüsselsperrung

Die Antwortnachricht meldet außer dem *RequestRespState* keine weiteren Daten zurück.

III.5 Bankparameterdaten

In den Bankparameterdaten (BPD) legt das Kreditinstitut den unterstützten Leistungsumfang fest und beschreibt diesen durch Parameter (siehe [FORMALS], Abschnitt *IV. BANKPARAMETERDATEN (BPD)*). Die BPD werden entweder in der Initialisierungsantwort zum Benutzer übertragen oder als Antwort auf einen administrativen Abruf-Auftrag (siehe *III.2.2.3 Initialisierungsantwort*, *III.7 Administrative Aufträge*). Die Bankparameterdaten sind in mehrere Segmente unterteilt, die im Folgenden gezeigt werden.

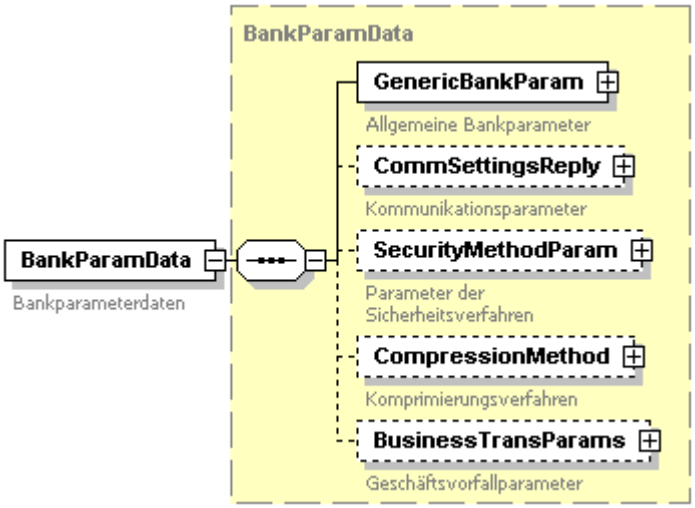


Abbildung 38: Bankparameterdaten

a) Allgemeine Bankparameter

Dieses Segment enthält allgemeine Informationen zum Kreditinstitut und zur unterstützten FinTS-Version.

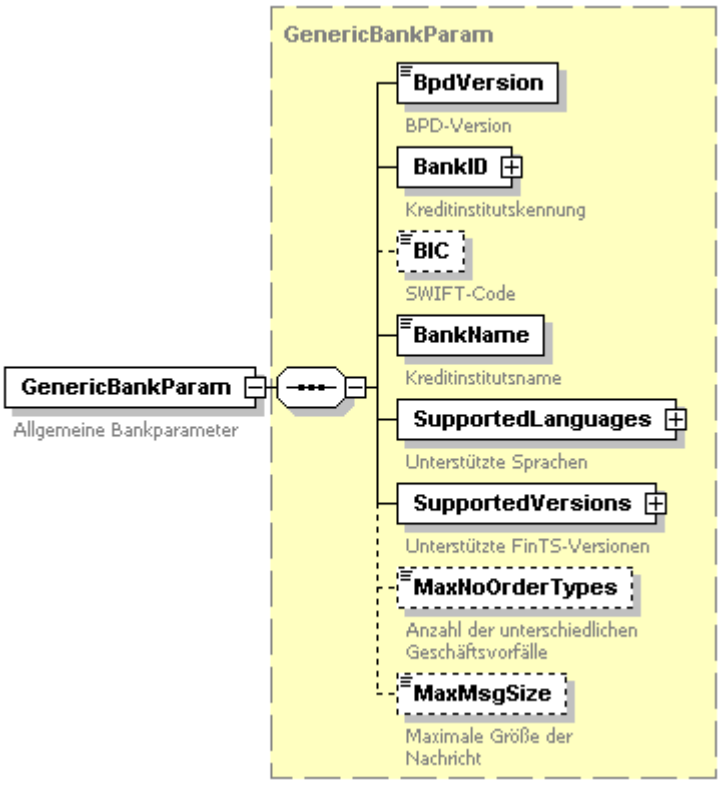


Abbildung 39: Allgemeine Bankparameterdaten

b) Kommunikationsparameter

Dieses Segment enthält Informationen zu den möglichen Kommunikationsverbindungen mit dem Kreditinstitut.

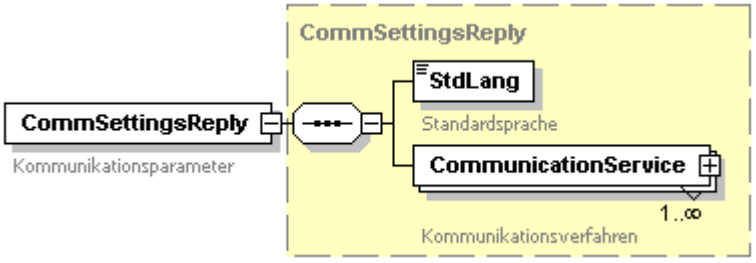


Abbildung 40: Kommunikationsparameter

Das Kommunikationsverfahren definiert spezifische Parameter eines Transportprotokolls:

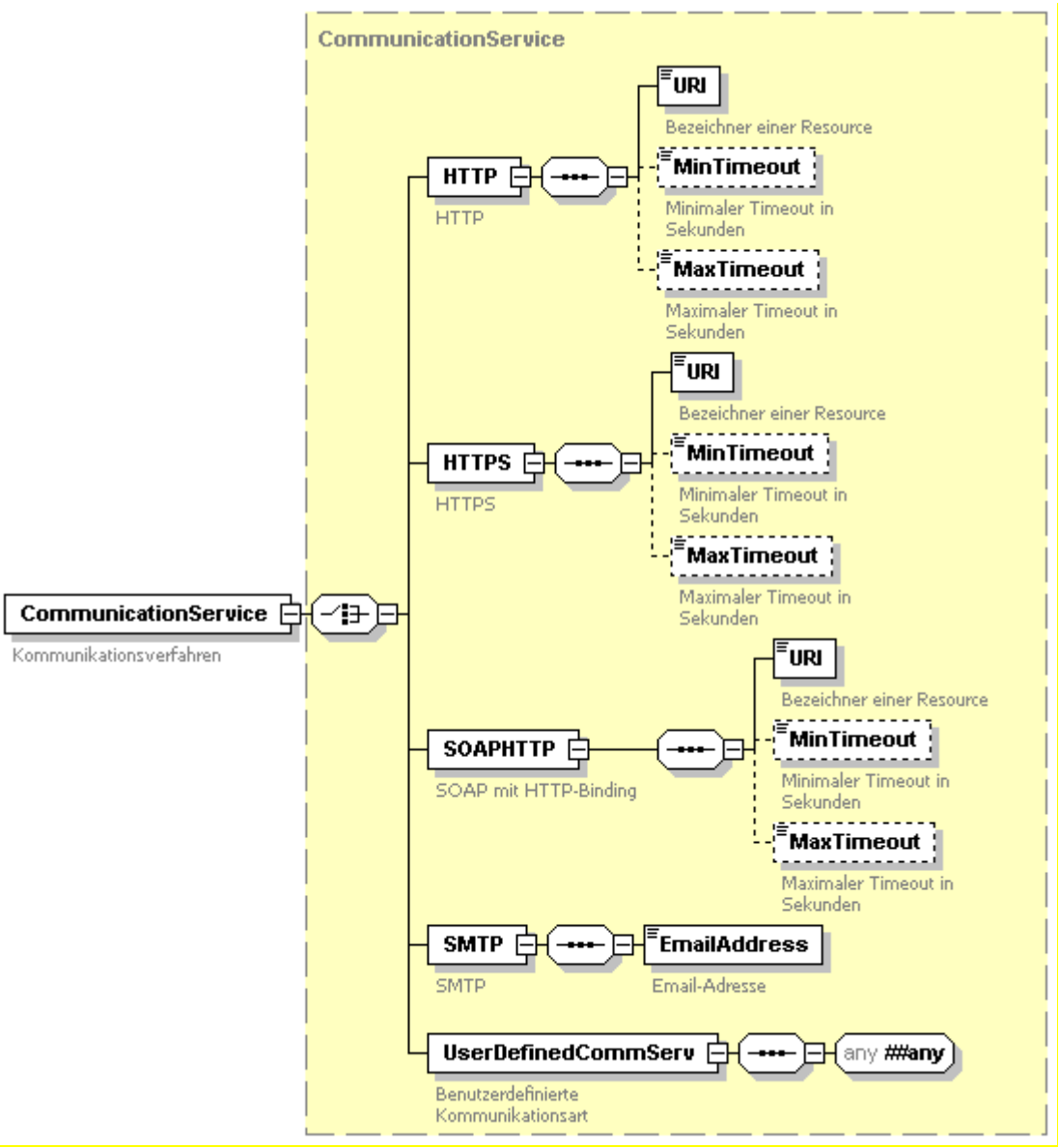


Abbildung 41: Kommunikationsverfahren

c) Parameter der Sicherheitsverfahren

In diesem Segment der BPD gibt das Kreditinstitut die unterstützten Sicherheitsverfahren und ihre Parameter an. Wenn ein Verfahren nicht unterstützt wird, ist es in der Aufzählung nicht vorhanden, ansonsten zählt zu seinen Parametern auch die Festlegung der mit diesem Verfahren signierbaren Geschäftsvorfälle. Die Geschäftsvorfälle können dabei entweder in Listen nach Namensraum sortiert aufgezählt oder aber generell zugelassen werden.

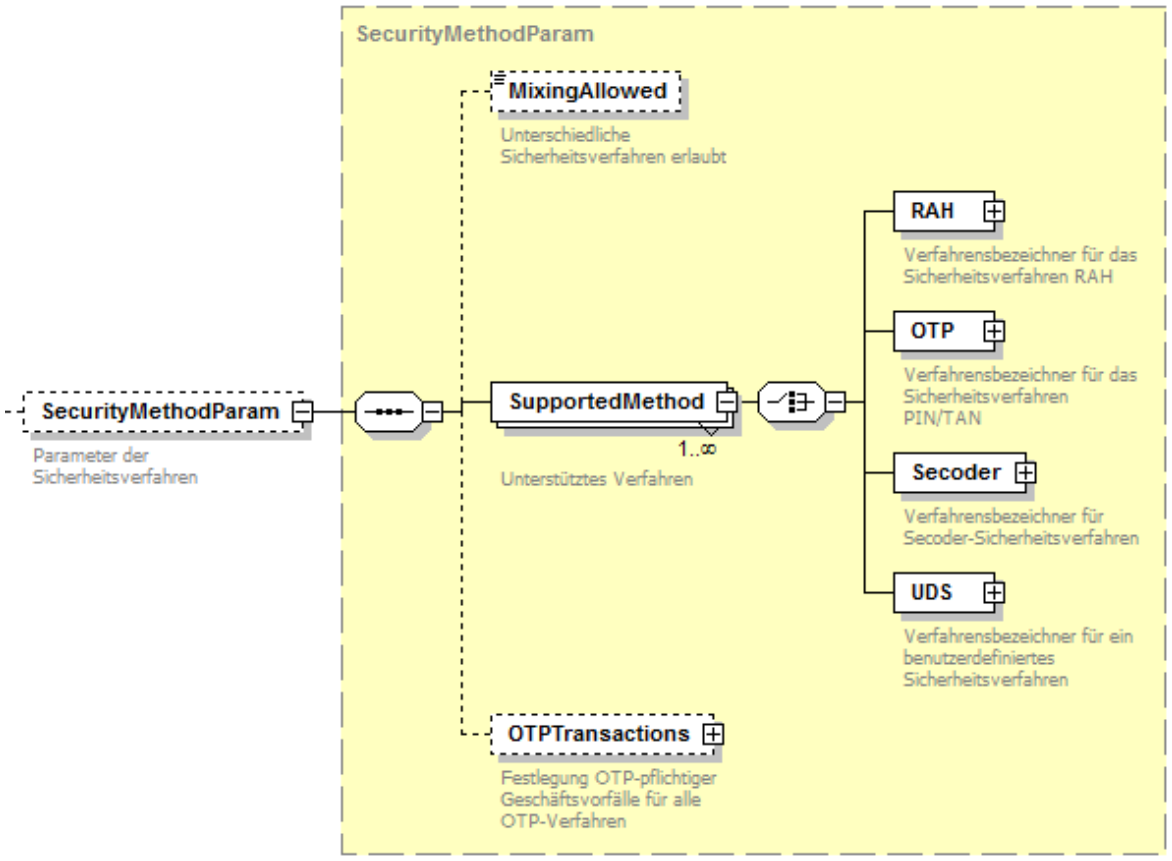


Abbildung 42: Parameter der Sicherheitsverfahren

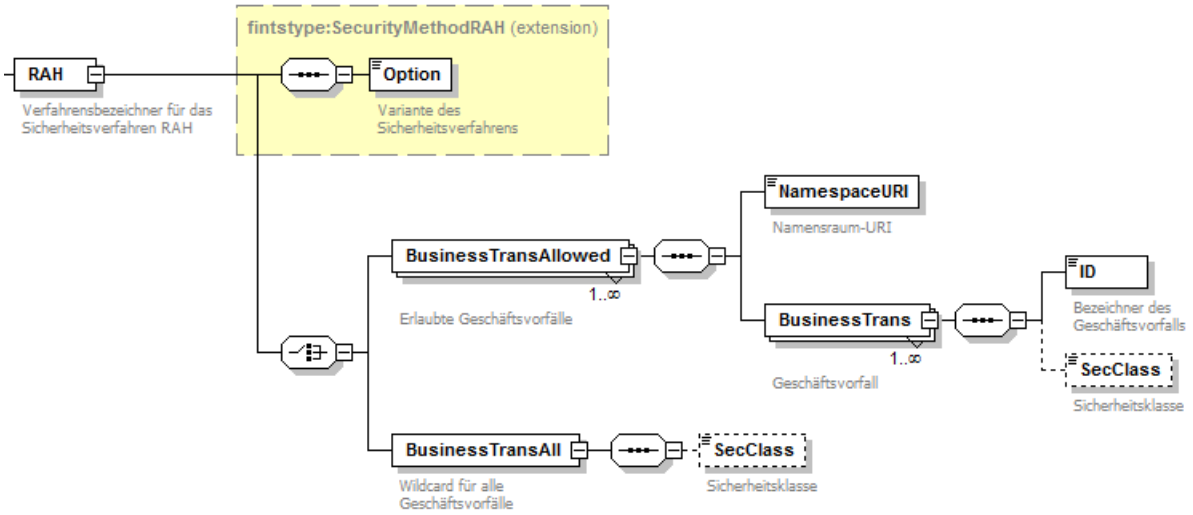


Abbildung 43: Sicherheitsverfahren RAH

Sicherheitsklasse

Bei den Sicherheitsverfahren RAH-7 und RAH-9 ist die Sicherheitsklasse verpflichtend anzugeben. Bei RAH-10 darf keine Sicherheitsklasse angegeben werden. Unterstützt ein Kreditinstitut ausschließlich das PIN/TAN-Verfahren, so ist in das DE ‚Sicherheitsklasse' des jeweiligen Geschäftsvorfallparametersegmentes als Füllwert ‚0' einzustellen. Die Sicherheitsklasse hat bei PIN/TAN für die Verarbeitung keine Bedeutung und darf vom Kundenprodukt für PIN/TAN nicht ausgewertet werden.

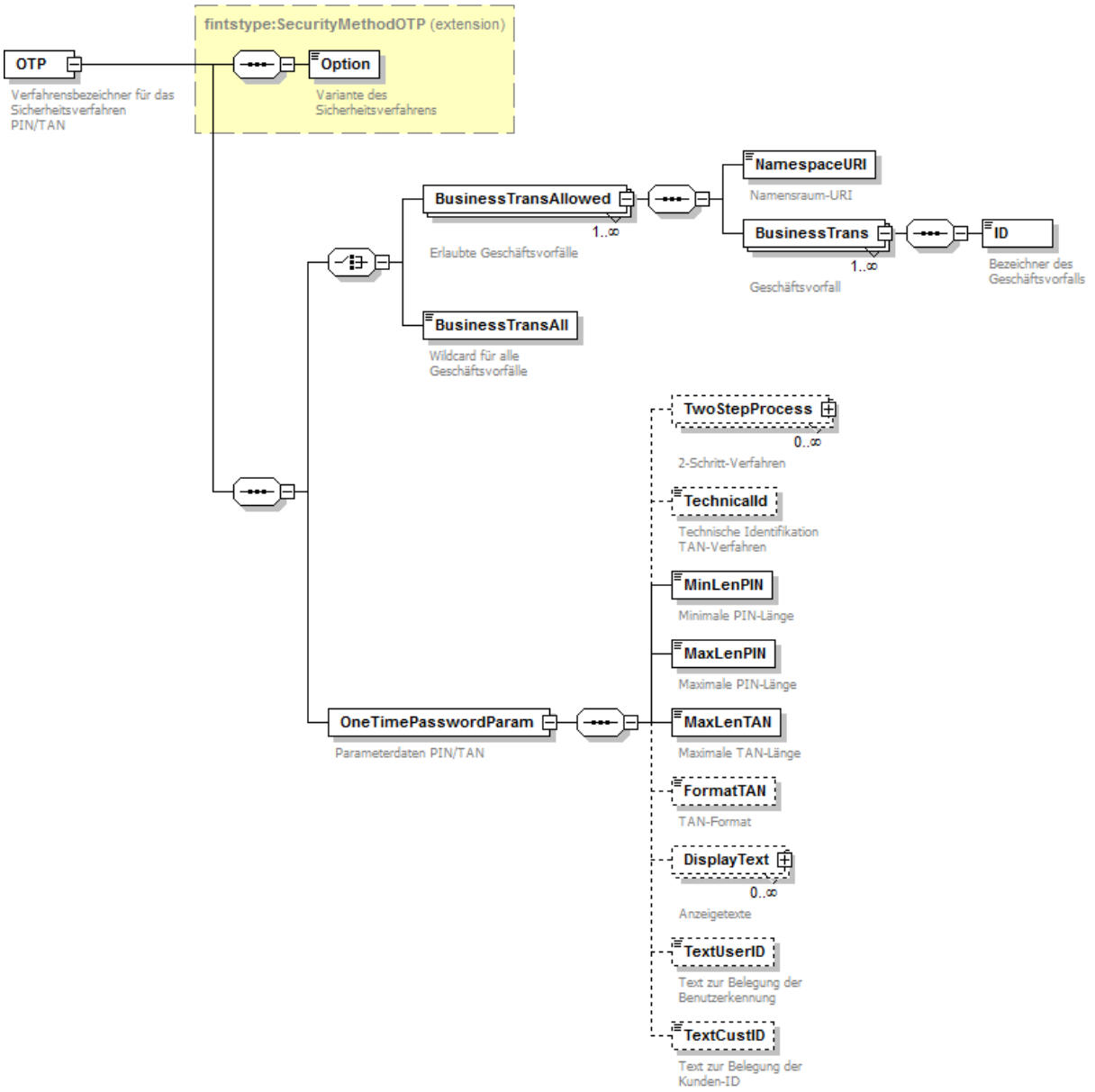


Abbildung 44: Sicherheitsverfahren OneTimePassword

Im Segment TwoStepProcess sind die Parameter für Zwei-Schritt-TAN-Verfahren enthalten (vgl. [PIN/TAN], *Abschnitt II.2ff*).

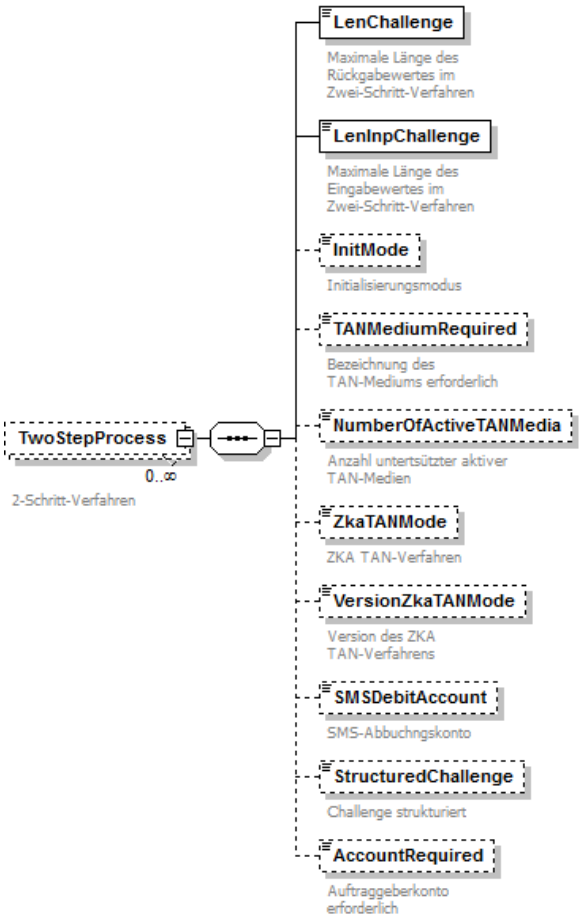


Abbildung 45: Sicherheitsverfahren OneTimePassword

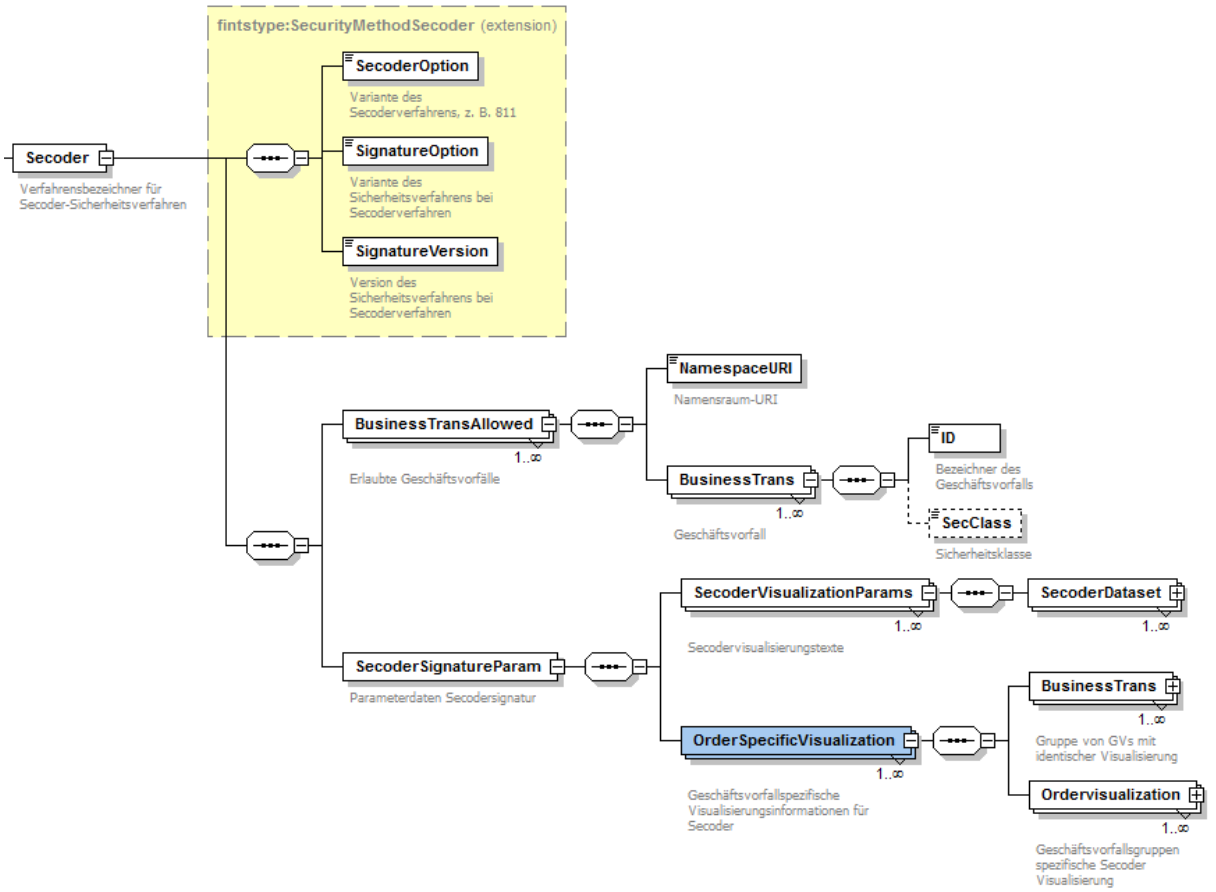


Abbildung 46: Sicherheitsverfahren Secoder

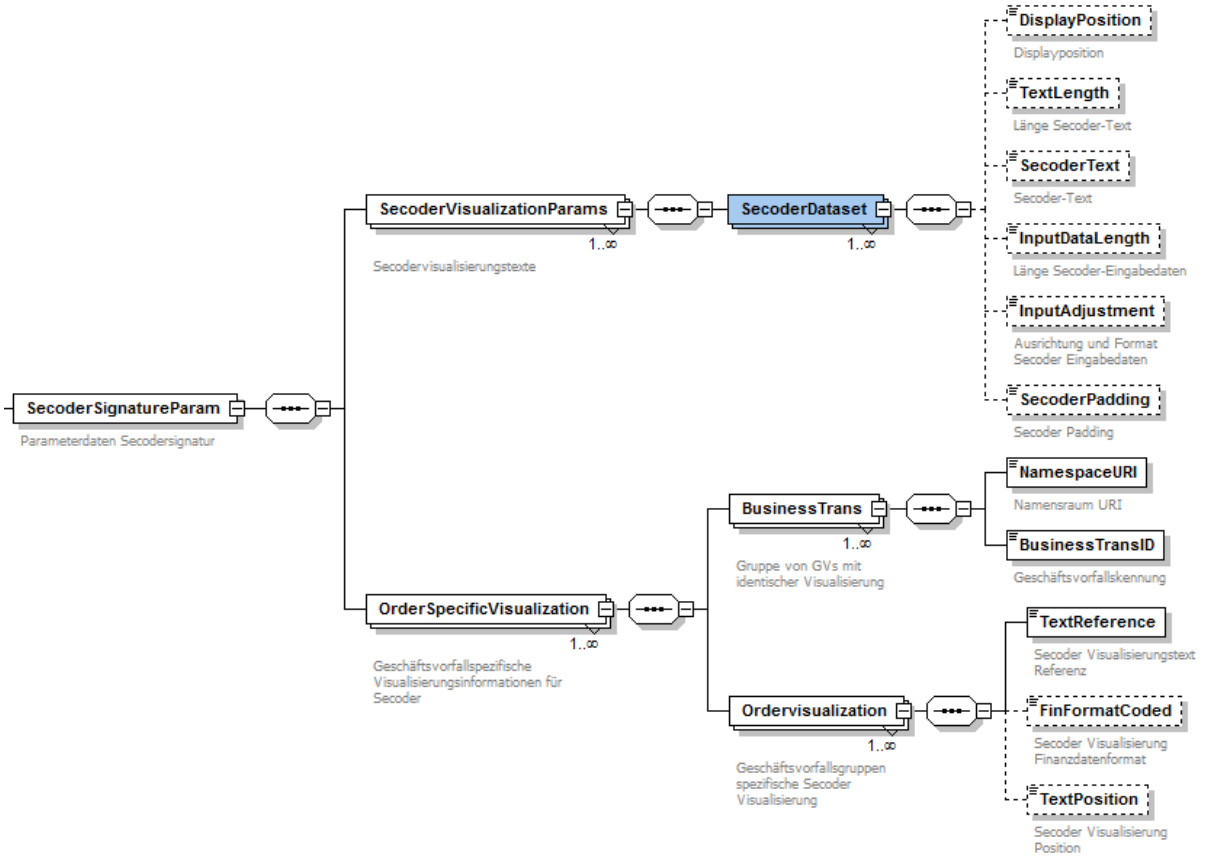


Abbildung 47: Sicherheitsverfahren Secoder – Parameterdaten Secodersignatur

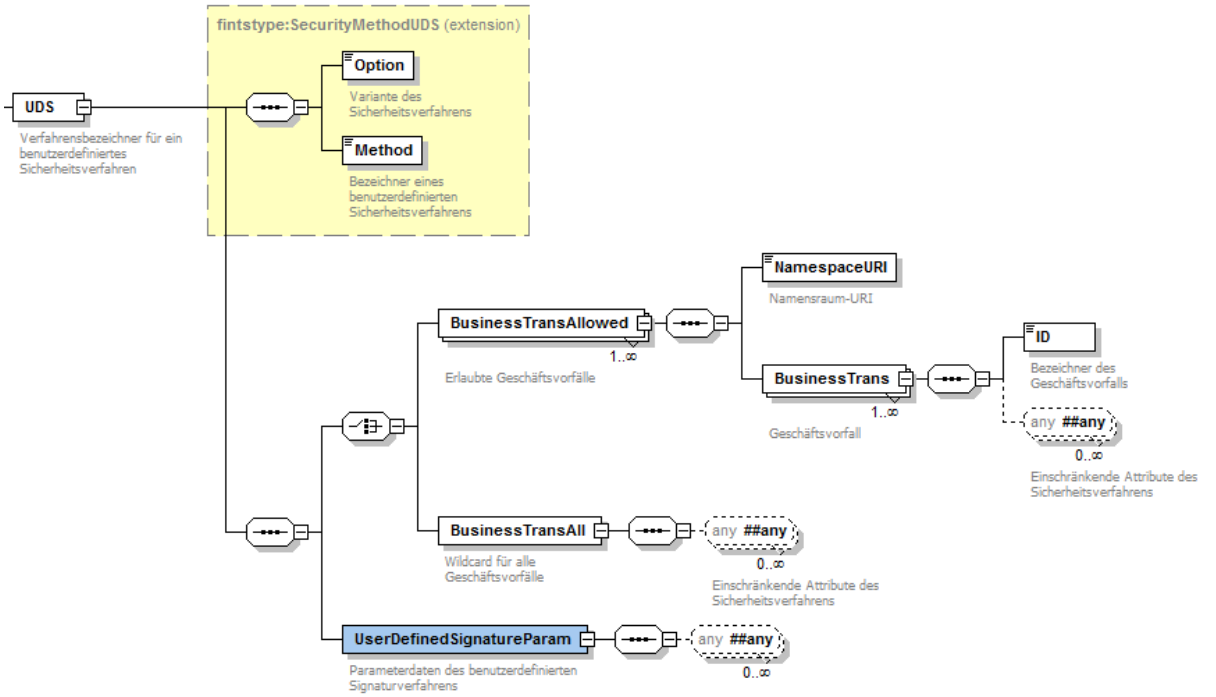


Abbildung 48: Sicherheitsverfahren UserDefinedSignature

d) Komprimierungsverfahren

In diesem Segment gibt das Kreditinstitut die unterstützten Komprimierungsverfahren an.

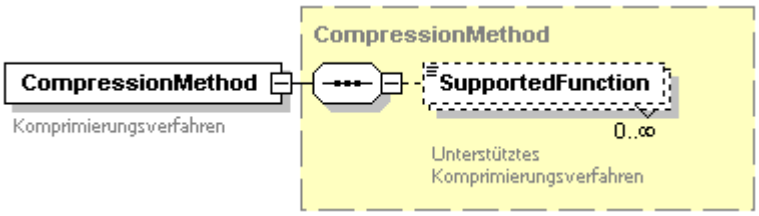


Abbildung 49: Komprimierungsverfahren

e) Geschäftsvorfallparameter

Zu jedem unterstützten Geschäftsvorfall enthalten die BPD ein Parametersegment. Die Anwesenheit oder Abwesenheit dieses Segments legt damit verbindlich fest, ob der Geschäftsvorfall unterstützt wird oder nicht. Wie in *II.4 Verbandseigene Geschäftsvorfälle* dargestellt, besitzt das Parametersegment eines Geschäftsvorfalls das XML-Tag

```
<Name des GV> ' ' <Version des GV> ' ' Par
```

und ist im Namensraum des Kundenauftrags- bzw. des Kreditinstitutsantwortsegments definiert.

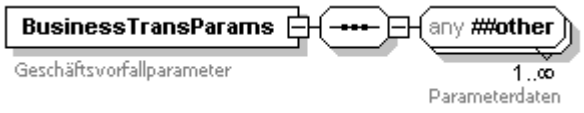


Abbildung 50: Geschäftsvorfallparameter

Die Parameterdefinition muss eine Erweiterung des Typs *ParameterData* aus dem Namensraum der FinTS-Messages darstellen und insbesondere dessen allgemeine Elemente *MaxNoOrders*, *MinNoSig* und *CheckModeSupported* enthalten. Darüber hinaus kann es spezifische Informationen über einen Geschäftsvorfalltyp enthalten.

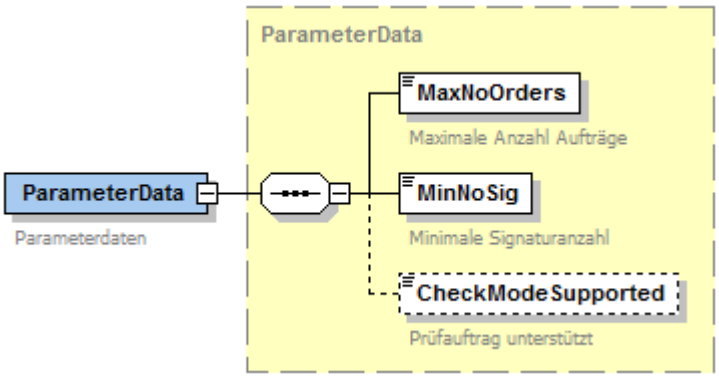


Abbildung 51: Parameterdaten

III.6 User-Parameterdaten

In den User-Parameterdaten legt das Kreditinstitut den unterstützten Leistungsumfang für einen bestimmten Benutzer fest und beschreibt diesen durch Parameter (siehe [FORMALS], Abschnitt *V. USER-PARAMETERDATEN (UPD)*). Je nach Kontext kann es sich um die User-Parameterdaten eines Benutzers für den direkten Zugang (UPD), um die User-Parameterdaten eines Benutzers für den Intermediärzugang (UPDI) oder um die User-Parameterdaten eines Intermediärs (IPD) handeln.

Die UPD werden entweder in der Initialisierungsantwort zum Benutzer übertragen oder als Antwort auf einen administrativen Abruf-Auftrag (siehe *III.2.2.3 Initialisierungsantwort*, *III.7 Administrative Aufträge*). Die UPD sind in mehrere Segmente aufgeteilt, die im Folgenden gezeigt werden.

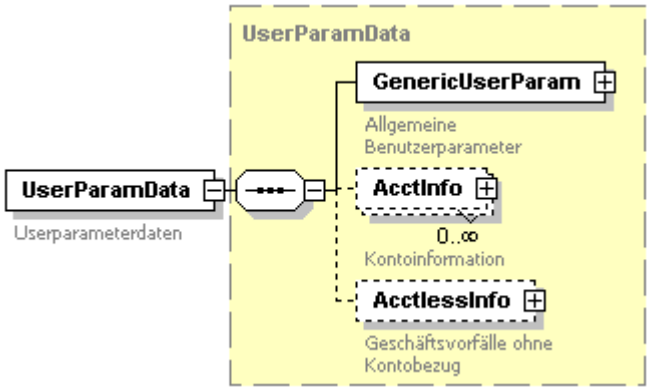


Abbildung 52: User-Parameterdaten

a) Allgemeine Benutzerparameter

Der allgemeine Abschnitt der UPD enthält beschreibende Informationen über die UPD.

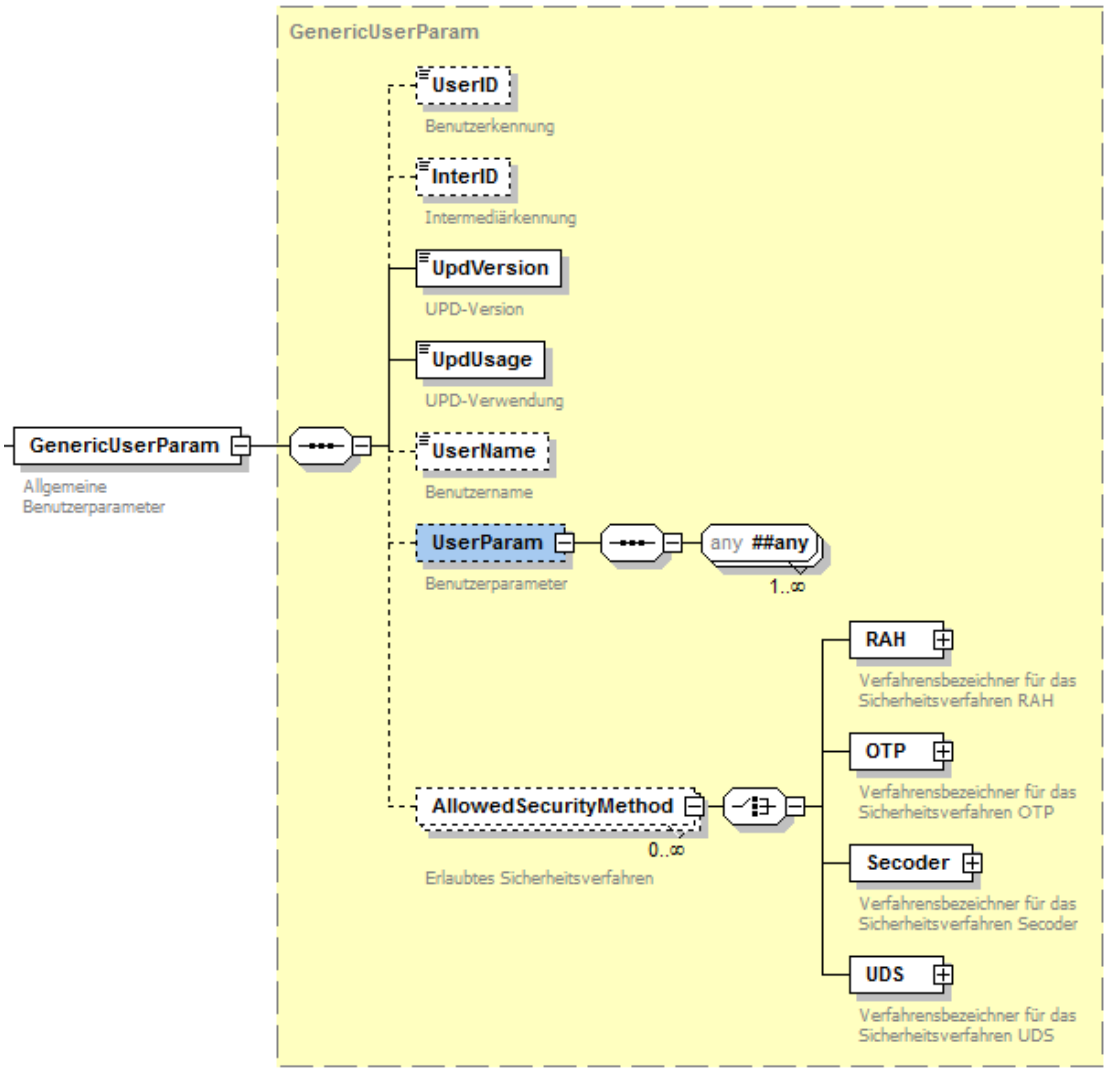


Abbildung 53: Allgemeine UPD

Benutzerkennung/Intermediärkennung

Handelt es sich um die UPD eines Benutzers, ist die Benutzerkennung belegt. Handelt es sich um die IPD eines Intermediärs, ist die Intermediärkennung belegt. In den UPD eines Benutzers für einen Intermediärzugang (UPDI) sind beide Kennungen belegt. Bei UPD für den anonymen Zugang (Gast-UPD) fehlen beide Kennungen.

b) Kontoinformation

Zu jedem Konto des Benutzers ist in den UPD ein Segment Kontoinformation hinterlegt. Es enthält Details zum Konto sowie eine Liste der für dieses Konto erlaubten Geschäftsvorfalltypen und optional ein kontospezifisches Limit. Bei UPD für den anonymen Zugang sowie bei IPD sind keine Segmente mit Kontobezug vorhanden.

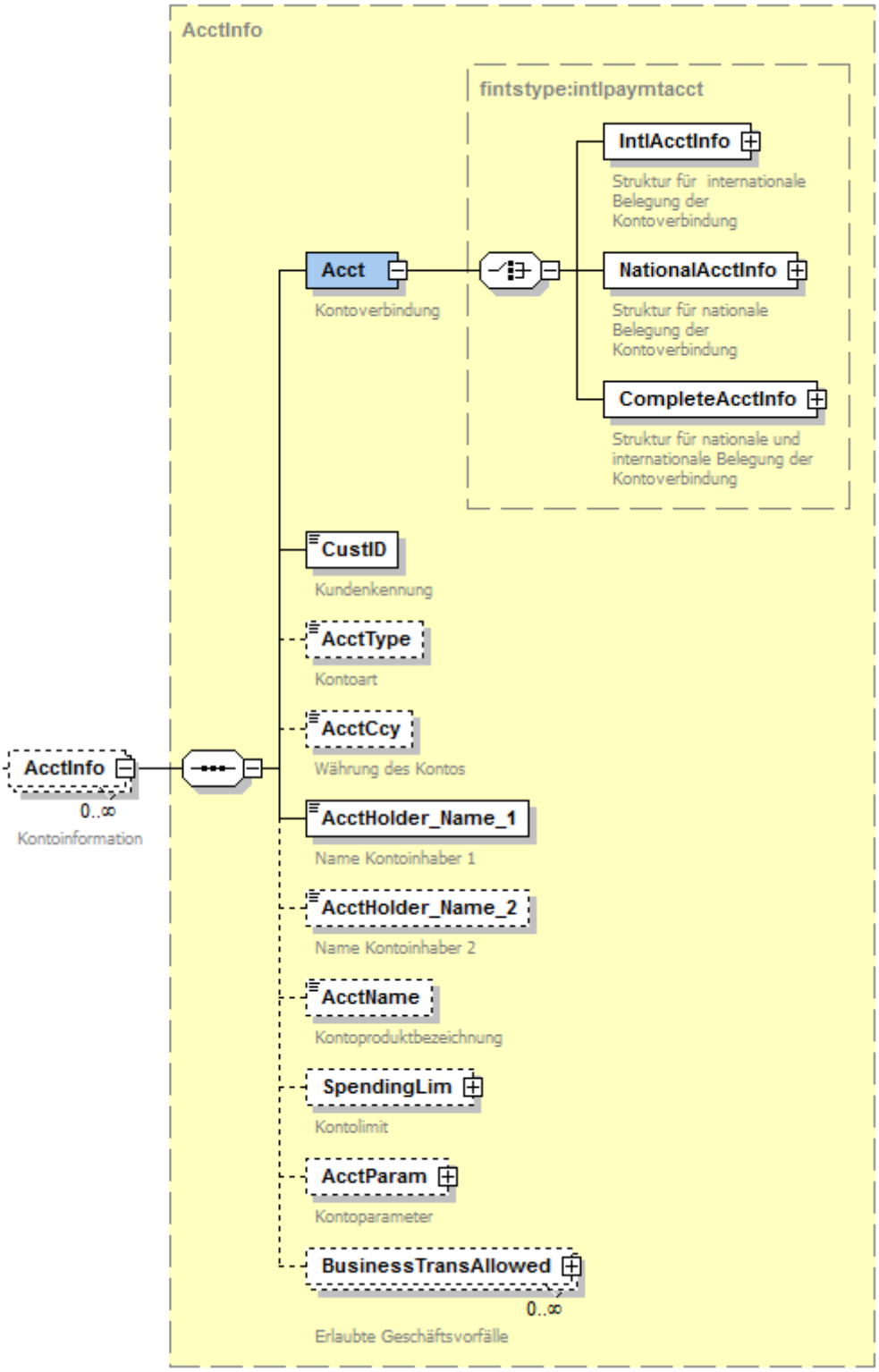


Abbildung 54: Kontoinformationen

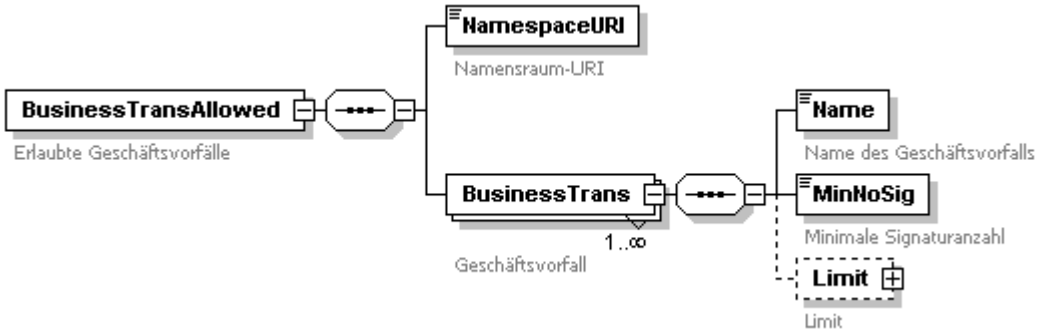


Abbildung 55: Erlaubte Geschäftsvorfälle

c) Geschäftsvorfälle ohne Kontobezug

Alle Geschäftsvorfälle, die im Auftrag keinen Bezug zu einem Konto aufweisen, werden in den UPD in einem eigenen Segment gruppiert. Sie sind auf jedem Konto zulässig. Die Geschäftsvorfall-Einträge in den *AcctInfo* und *AcctlessInfo*-Elementen sollten sich nicht überschneiden. Bei UPD für den anonymen Zugang sowie bei IPD ist nur dieses Segment belegt, Geschäftsvorfälle mit Kontobezug gibt es in diesen Fällen nicht.

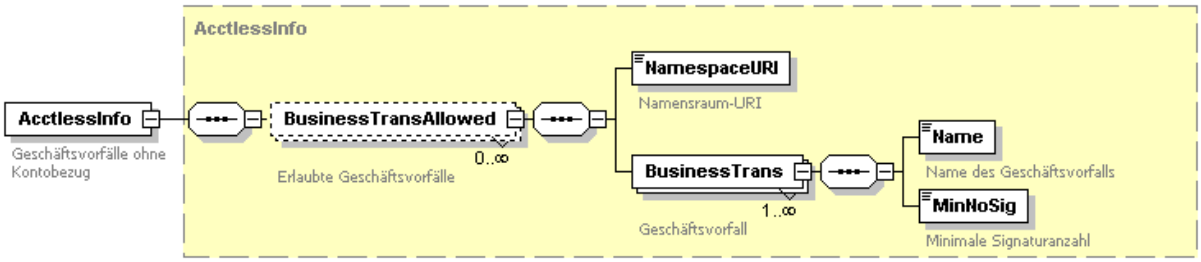


Abbildung 56: Geschäftsvorfälle ohne Kontobezug

III.7 Administrative Aufträge

Dieser Abschnitt beschreibt die administrativen Aufträge in FinTS. Dieses sind die Anforderungen von BPD, UPD, IPD und UPDI, Aufträge für Intermediärszenarien, die PIN/TAN-Aufträge, Aufträge in Zusammenhang mit dem Abonnement, der Adressenregistrierung, verteilten Signaturen, der Quittung und dem Statusprotokoll. Alle diese Aufträge werden in Standard-Benutzernachrichten eingestellt.

Die Beschreibung eines Auftrags gliedert sich jeweils in die Abschnitte Benutzerauftrag, Kreditinstitutsrückmeldung und Bankparameterdaten (die Geschäftsvorfallparameter zu diesem Auftrag). Zu den enthaltenen Elementen sind Texte zur Belegung unter den Abbildungen enthalten, wenn sie nicht der Standardbedeutung aus dem [DataDictionary] entsprechen.

Die Bankparameterdaten enthalten jeweils mindestens die Felder zur minimal erforderlichen Anzahl der Signaturen und maximalen Anzahl möglicher Aufträge.

In einem ersten Abschnitt werden zunächst allgemeine Typen beschrieben. Darauf folgend sind die administrativen Aufträge beschrieben.

III.7.1 BPD

Neben der Möglichkeit, BPD als Teil der Initialisierungsantwort zu erhalten, können BPD mit einem administrativen Auftrag angefordert werden. Dies ist insbesondere in Intermediärszenarien von Bedeutung.

III.7.1.1 BPD anfordern

Mit diesem Auftrag fordert ein Benutzer die Bankparameterdaten eines Kreditinstituts an (siehe [Formals], Abschnitt *IV.3 Anforderung der BPD in einem Szenario mit Intermediär*).

a) Benutzerauftrag

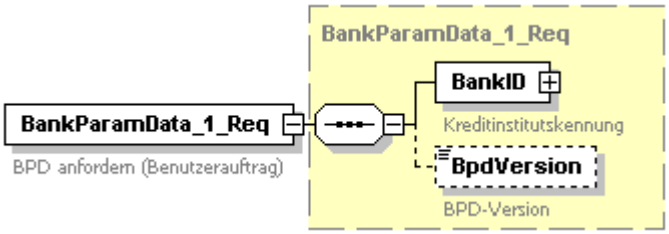


Abbildung 57: Benutzerauftrag BPD anfordern

BPD-Version

Falls die BPD-Version angegeben wird, wird die BPD nur geliefert, wenn sie neuer ist. Andernfalls wird immer die aktuelle BPD geliefert.

b) Kreditinstitutsrückmeldung

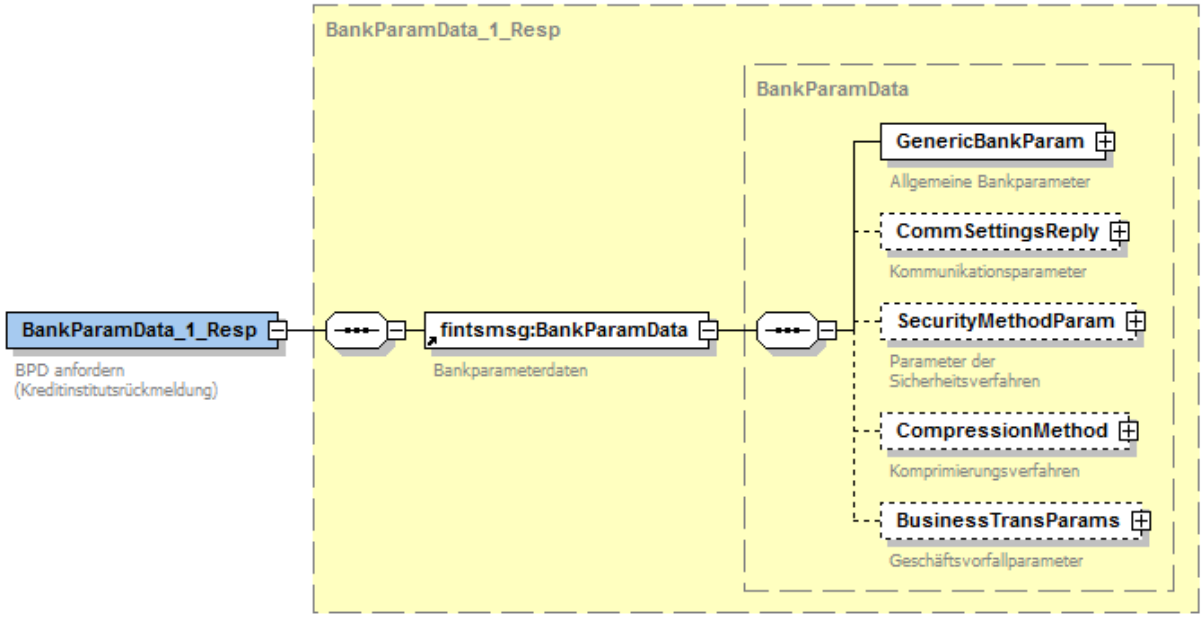


Abbildung 58: Kreditinstitutsrückmeldung BPD anfordern

c) Bankparameterdaten

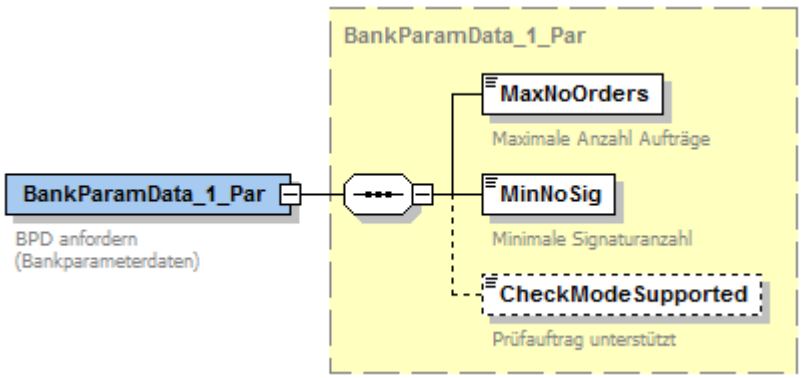


Abbildung 59: Bankparameterdaten BPD anfordern

III.7.2 UPD

Neben der Möglichkeit, UPD als Teil der Initialisierungsantwort zu erhalten, können UPD mit einem administrativen Auftrag angefordert werden. Dies ist insbesondere in Intermediärszenarien von Bedeutung. In diesen Szenarien sind außer den normalen UPD des Benutzers auch UPD des Intermediärs (IPD) und intermediärbezogene UPD des Benutzers (UPDI) definiert, welche ebenfalls mit diesem Auftrag abgeholt werden können.

III.7.2.1 UPD anfordern

Mit diesem Auftrag können die UPD, IPD oder UPDI angefordert werden (siehe [Formals], Abschnitt *V.6 Explizite Anforderung von UPD*).

a) Benutzerauftrag

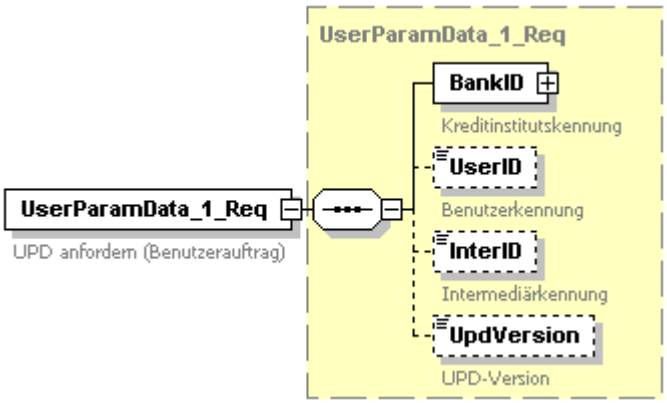


Abbildung 60: Benutzerauftrag UPD anfordern

Benutzerkennung/Intermediärkennung

Bei Anforderung der User-Parameterdaten eines Benutzers für den direkten Zugang (UPD) ist die Benutzerkennung anzugeben. Bei Anforderung der User-Parameterdaten eines Intermediärs (IPD) ist die Intermediärkennung anzugeben. Bei Anforderung der User-Parameterdaten eines Benutzers für einen Intermediärzugang (UPDI) sind beide Kennungen anzugeben. Bei Anforderung der UPD für den anonymen Zugang (Gast-UPD) ist keine der Kennungen anzugeben.

UPD-Version

Falls die UPD-Version angegeben wird, wird die UPD nur geliefert wenn sie neuer ist. Andernfalls wird immer die aktuelle UPD geliefert.

b) Kreditinstitutsrückmeldung

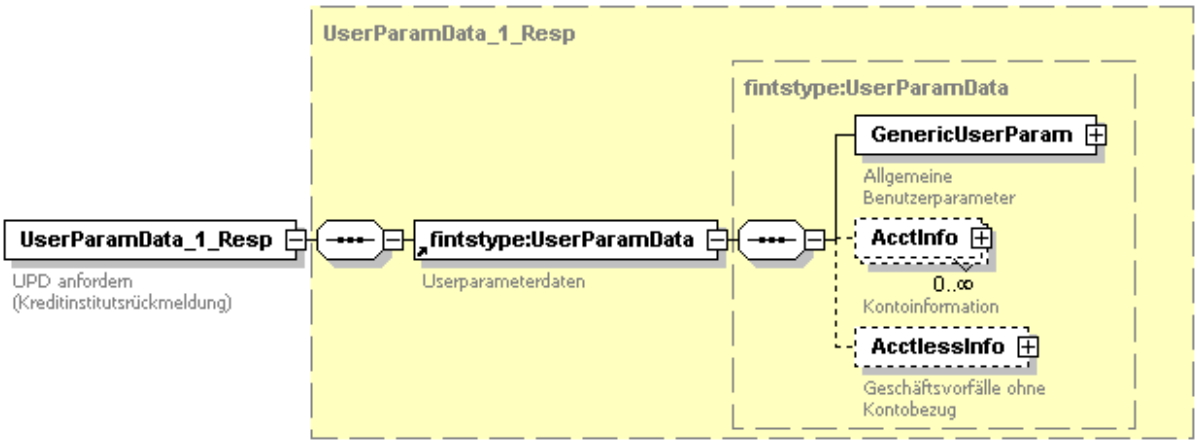


Abbildung 61: Kreditinstitutsrückmeldung UPD anfordern

c) Bankparameterdaten

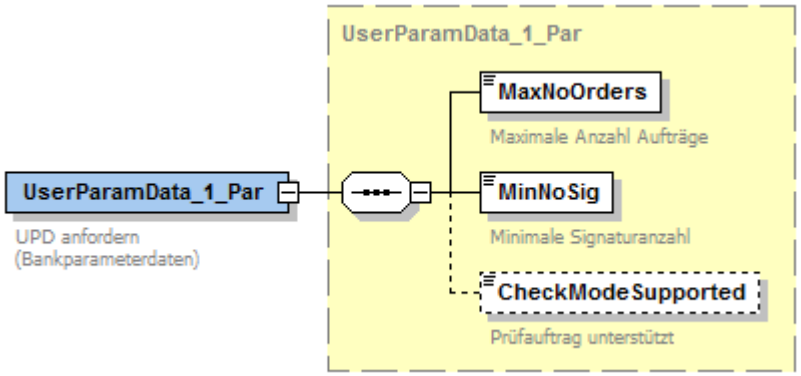


Abbildung 62: Bankparameterdaten UPD anfordern

III.7.3 Intermediärszenarien

Für die Kommunikation über einen Intermediär sind administrative Geschäftsvorfälle definiert, die zur Verwaltung der Intermediärbenutzung durch das Kreditinstitut dienen.

III.7.3.1 Liste der Intermediäre

Mit diesem Auftrag kann der Benutzer Informationen über Intermediäre abrufen (siehe [Formals], Abschnitt *V.7 Pflege der Intermediärzugänge und der UPDI*).

a) Benutzerauftrag

Abbildung 63: Benutzerauftrag Liste der Intermediäre

b) Kreditinstitutsrückmeldung

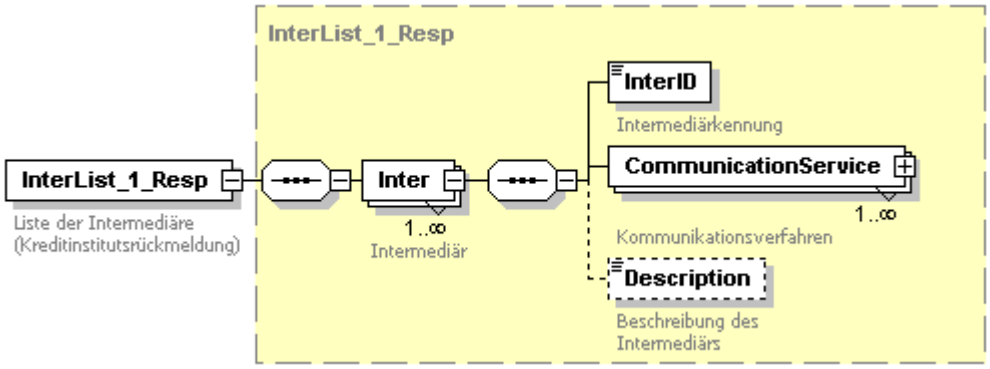


Abbildung 64: Kreditinstitutsrückmeldung Liste der Intermediäre

Beschreibung des Intermediärs

Hier kann eine textuelle Beschreibung des Intermediärs angegeben werden.

c) Bankparameterdaten

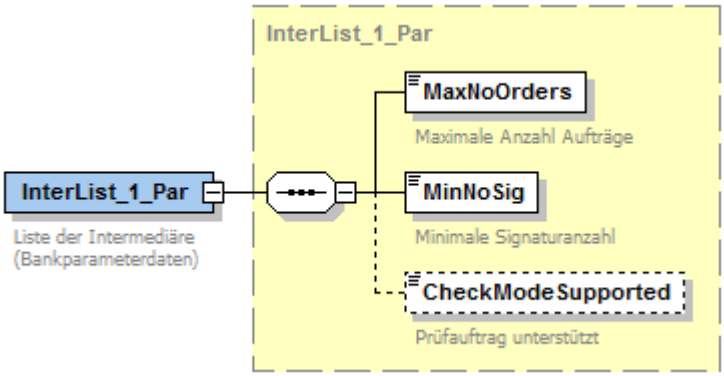


Abbildung 65: Bankparameterdaten Liste der Intermediäre

III.7.3.2 Für einen Intermediär anmelden

Mit diesem Auftrag kann ein Benutzer sich für die Benutzung bei einem Intermediär anmelden (siehe [Formals], Abschnitt *V.7 Pflege der Intermediärzugänge und der UPDI*).

a) Benutzerauftrag

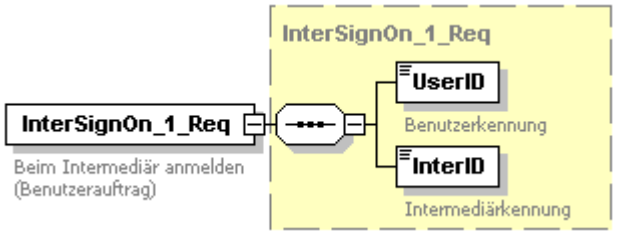


Abbildung 66: Benutzerauftrag Für einen Intermediär anmelden

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

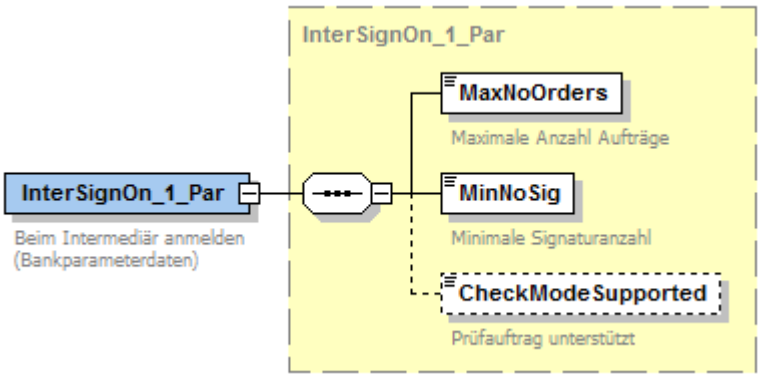


Abbildung 67: Bankparameterdaten Für einen Intermediär anmelden

III.7.3.3 Für einen Intermediär abmelden

Mit diesem Auftrag kann ein Benutzer sich für die Benutzung bei einem Intermediär abmelden (siehe [Formals], Abschnitt *V.7 Pflege der Intermediärzugänge und der UPDI*).

a) Benutzerauftrag

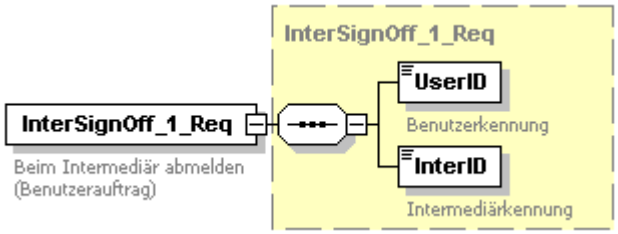


Abbildung 68: Benutzerauftrag Für einen Intermediär abmelden

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

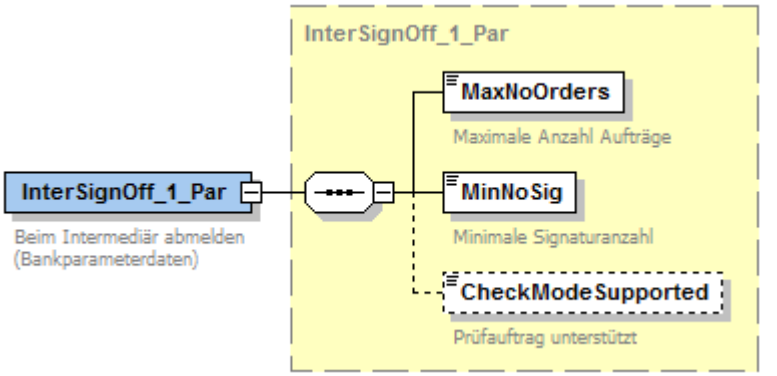


Abbildung 69: Bankparameterdaten Für einen Intermediär abmelden

III.7.3.4 UPDI ändern

Mit diesem Auftrag kann der Benutzer geänderte UPDI beim Kreditinstitut einreichen (siehe [Formals], Abschnitt *V.7 Pflege der Intermediärzugänge und der UPDI*).

a) Benutzerauftrag

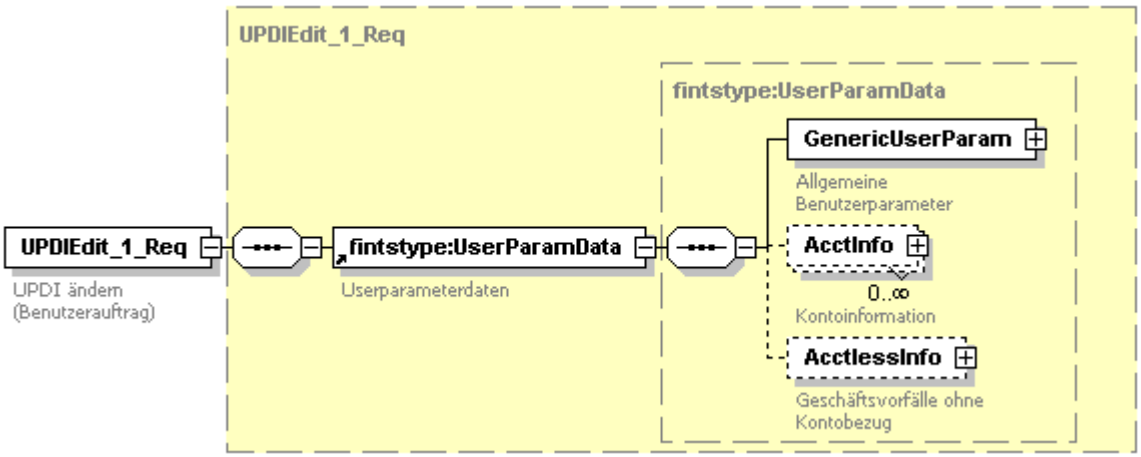


Abbildung 70: Benutzerauftrag UPDI ändern

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

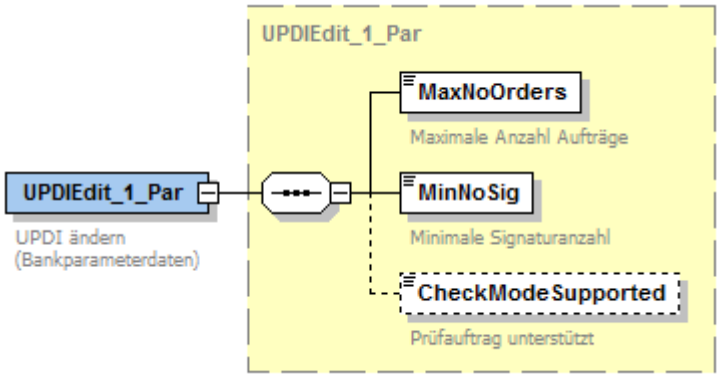


Abbildung 71: Bankparameterdaten UPDI ändern

III.7.4 PIN/TAN

Im Sicherheitsverfahren PIN/TAN werden administrative Geschäftsvorfälle zur Verwaltung der Online-Banking-PIN und der Parameter der unterschiedlichen TAN-Verfahren benötigt.

III.7.4.1 Online-Banking-PIN ändern

Mit diesem Auftrag kann ein Benutzer seine PIN ändern (siehe [PINTAN], Abschnitt *II.6.1.1 Online-Banking-PIN ändern*).

a) Benutzerauftrag

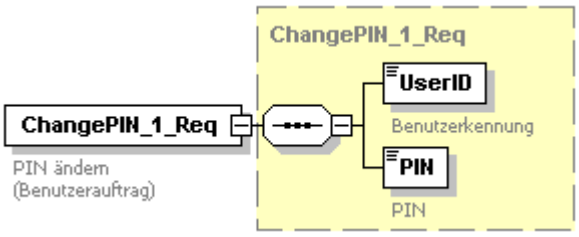


Abbildung 72: Benutzerauftrag PIN ändern

PIN

Hier ist die neu einzurichtende PIN anzugeben.

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

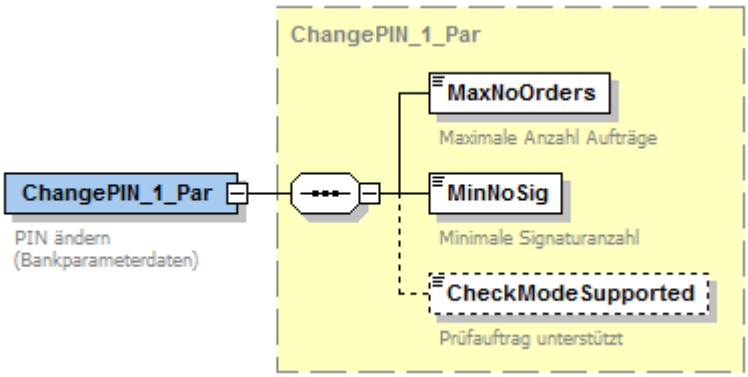


Abbildung 73: Bankparameterdaten PIN ändern

III.7.4.2 Online-Banking-PIN sperren

Mit diesem Auftrag kann ein Benutzer seine PIN sperren (siehe [PINTAN], Abschnitt *II.8.2.3 Online-Banking-PIN sperren*).

a) Benutzerauftrag

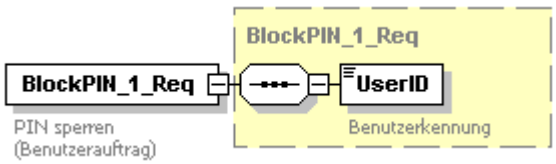


Abbildung 74: Benutzerauftrag PIN-Sperre

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

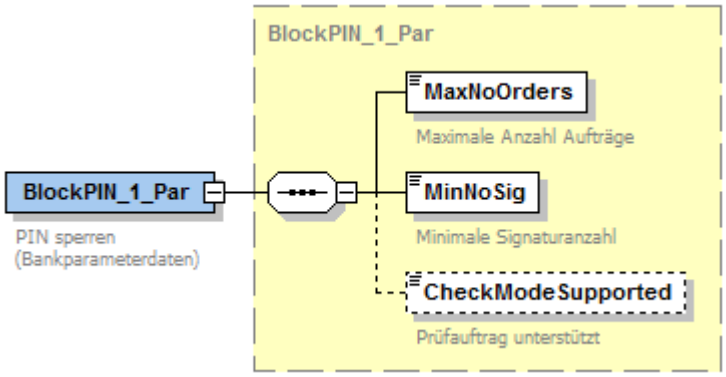


Abbildung 75: Bankparameterdaten PIN-Sperre

III.7.4.3 Online-Banking-PIN-Sperre aufheben

Mit diesem Auftrag kann ein Benutzer selbst eine durch ihn veranlasste PIN-Sperre aufheben (siehe [PINTAN], Abschnitt *II.8.2.4 Online-Banking-PIN-Sperre aufheben*).

a) Benutzerauftrag

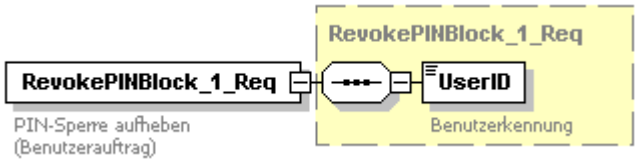


Abbildung 76: Benutzerauftrag PIN-Sperre aufheben

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

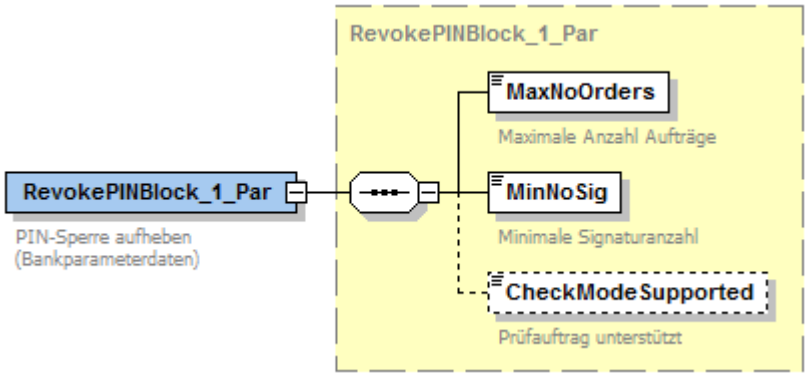


Abbildung 77: Bankparameterdaten PIN-Sperre aufheben

III.7.4.4 TAN-Verbrauchsinformationen anzeigen

Mit diesem Auftrag kann ein Benutzer sich anzeigen lassen, welche TANs er bereits verbraucht hat (siehe [PINTAN], Abschnitt *II.8.2.1 TAN-Verbrauchsinformationen anzeigen*).

a) Benutzerauftrag

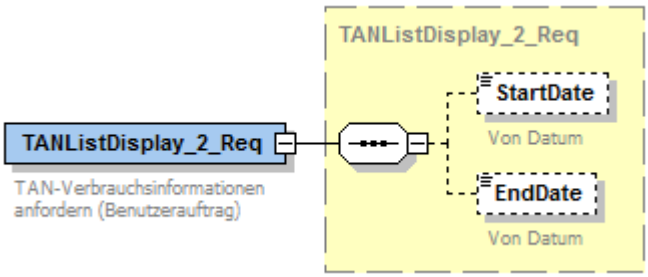


Abbildung 78: Benutzerauftrag TAN-Verbrauchsinformationen anzeigen

b) Kreditinstitutsrückmeldung

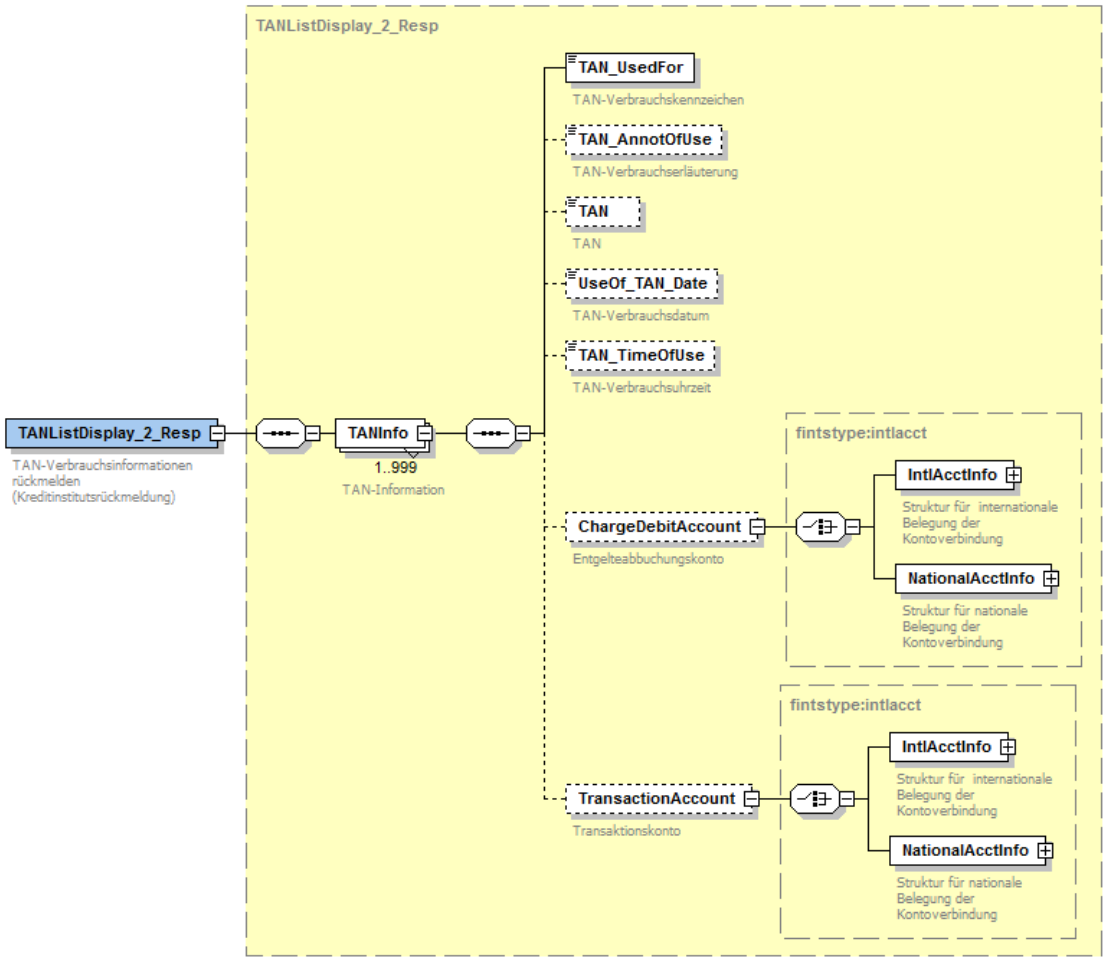


Abbildung 79: Kreditinstitutsrückmeldung TAN-Verbrauchsinformationen anzeigen

c) Bankparameterdaten

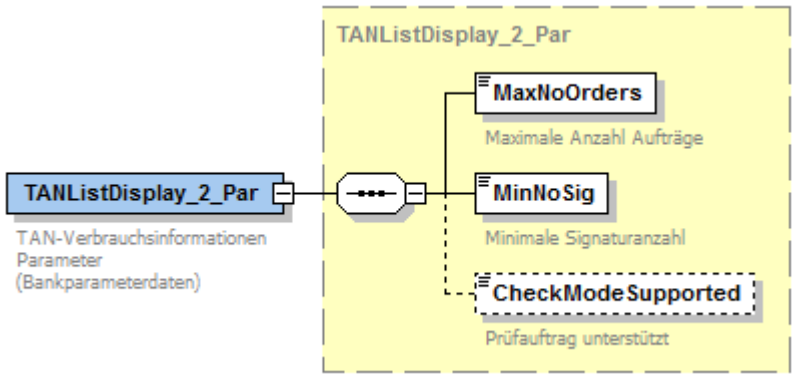


Abbildung 80: Bankparameterdaten TAN-Verbrauchsinformationen anzeigen

III.7.4.5 Anzeige der verfügbaren TAN-Medien

Dem Benutzer wird eine Übersicht über seine verfügbaren TAN-Medien für chipTAN und mobileTAN angezeigt. (siehe [PINTAN], Abschnitt *II.8.2.2 Anzeige der verfügbaren TAN-Medien*).

a) Benutzerauftrag

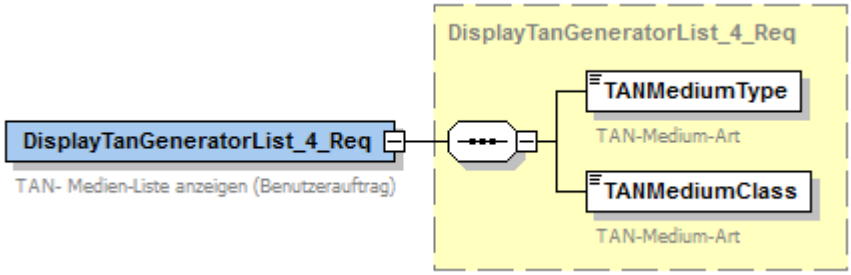


Abbildung 81: Benutzerauftrag Anzeige der verfügbaren TAN-Medien

b) Kreditinstitutsrückmeldung

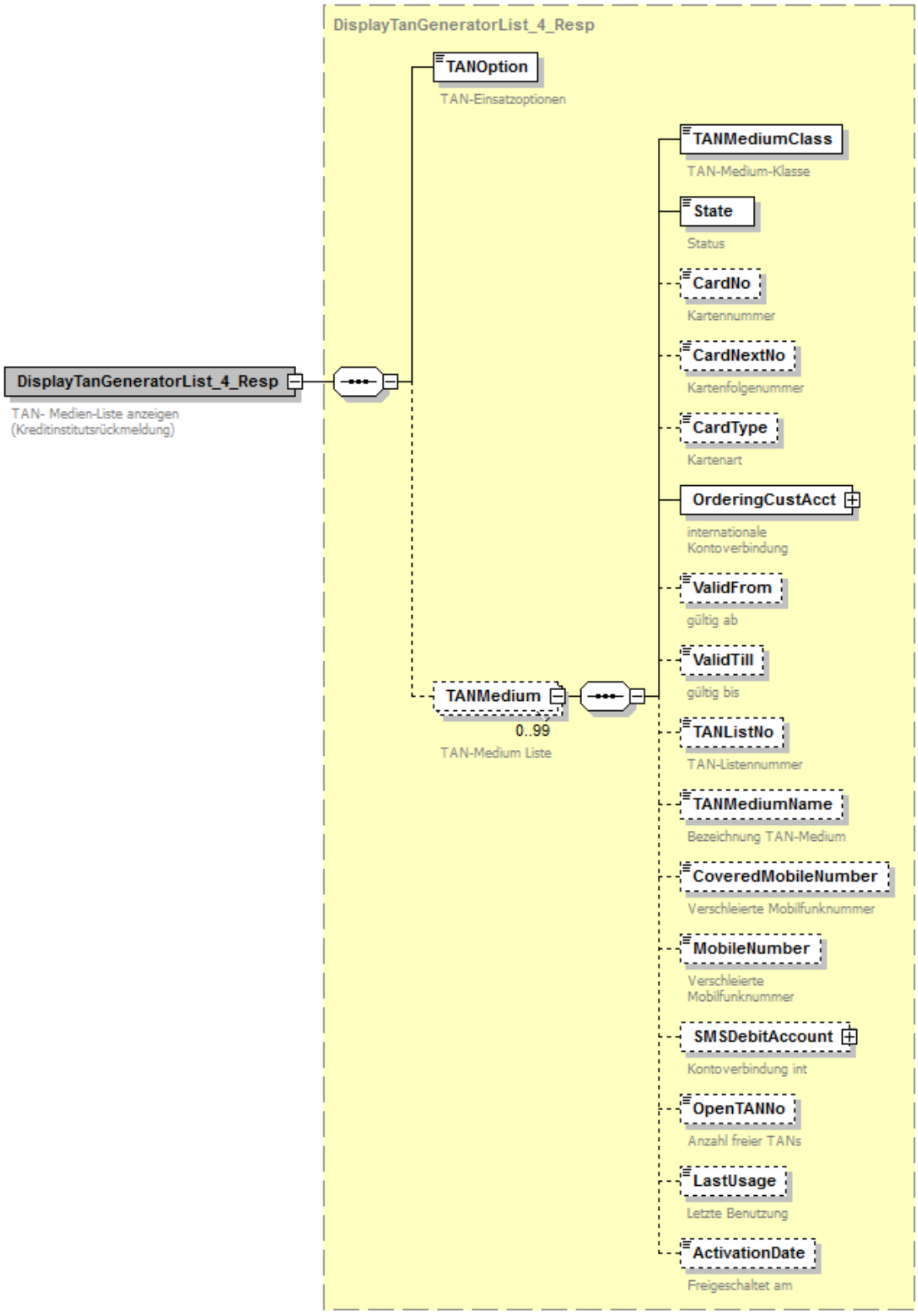


Abbildung 82: Kreditinstitutsrückmeldung Anzeige der verfügbaren TAN-Medien

c) Bankparameterdaten

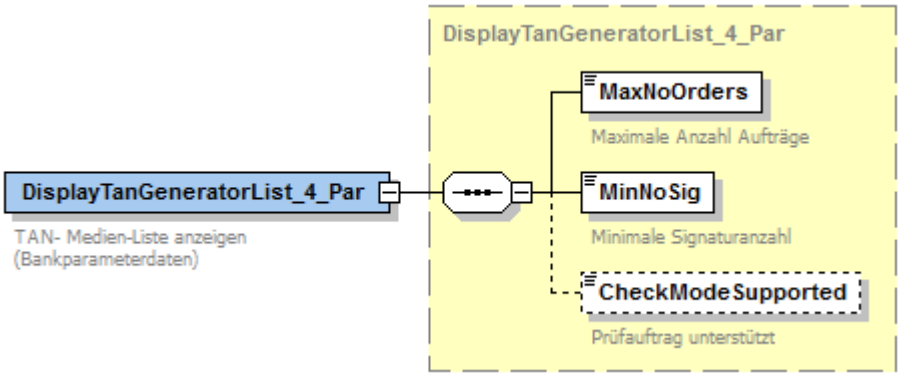


Abbildung 83: Bankparameterdaten Anzeige der verfügbaren TAN-Medien

III.7.4.6 TAN-Generator an- bzw. ummelden

Mit diesem Geschäftsvorfall kann der Benutzer seinem Kreditinstitut mitteilen, welches Medium (Chipkarte, TAN-Generator) er für die Autorisierung der Aufträge per TAN verwenden wird. (siehe [PINTAN], Abschnitt *II.8.2.4 TAN-Generator an- bzw- ummelden*).

a) Benutzerauftrag

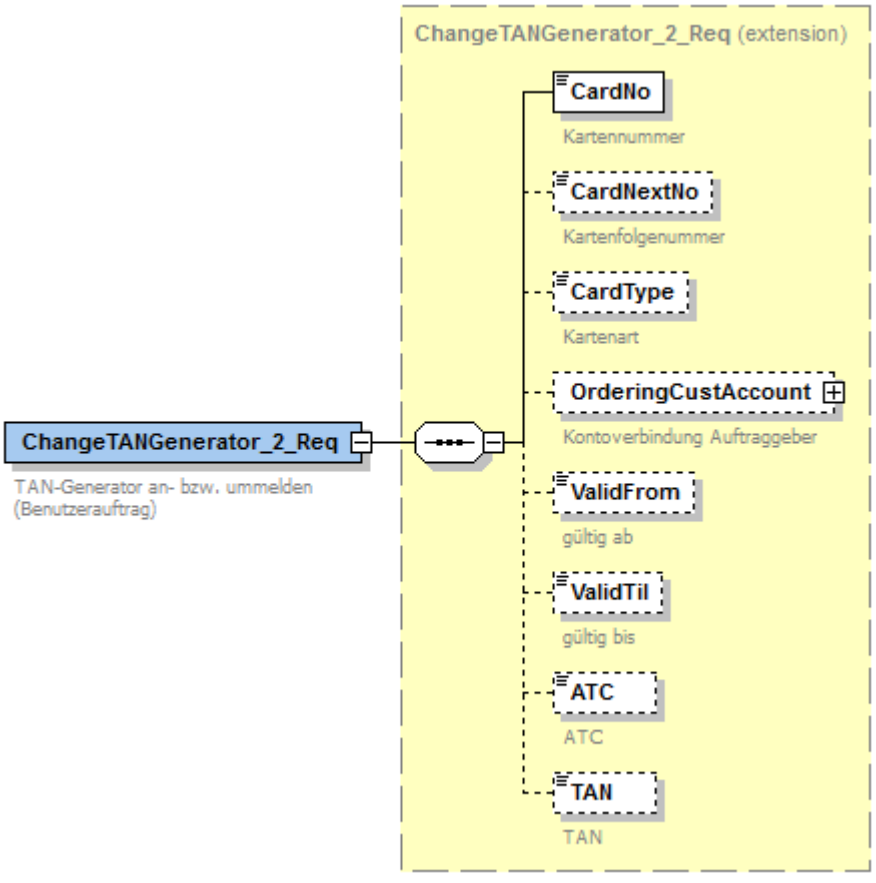


Abbildung 84: Benutzerauftrag TAN-Generator an- bzw. ummelden

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

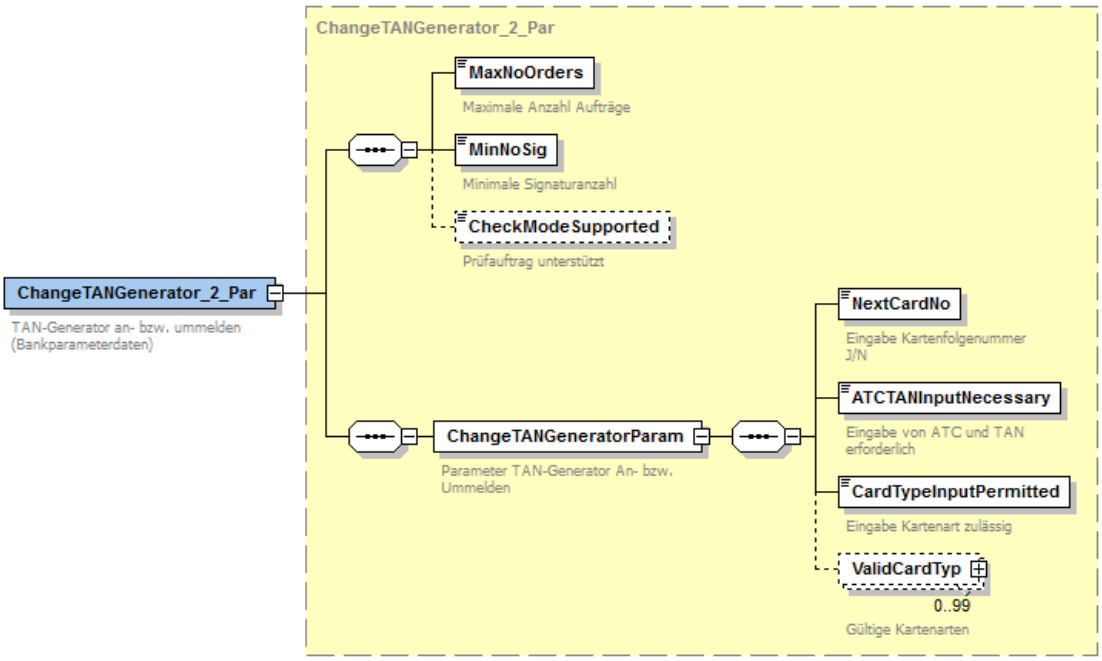


Abbildung 85: Bankparameterdaten TAN-Generator an- bzw. ummelden

III.7.4.7 TAN-Generator Synchronisierung

Mit diesem Geschäftsvorfall ist eine explizite Synchronisierung eines TAN-Generators nach HHD-Standard möglich. (siehe [PINTAN], Abschnitt *II.8.2.4 TAN-Generator Synchronisierung*).

a) Benutzerauftrag

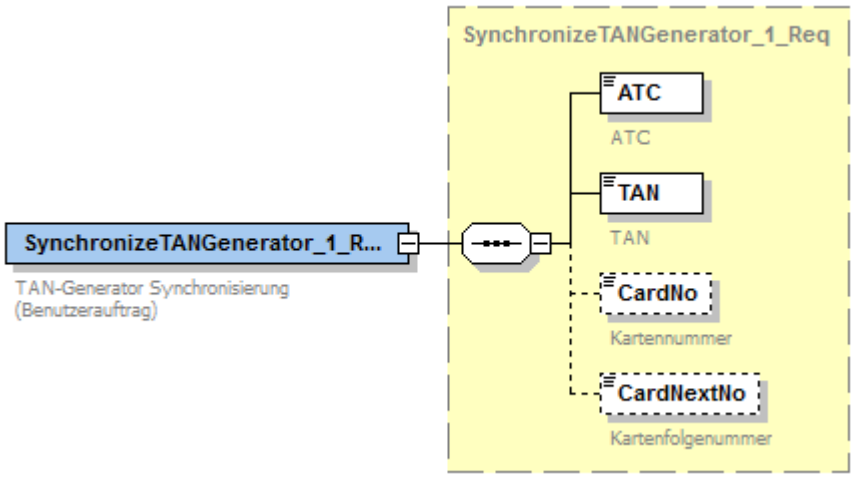


Abbildung 86: Benutzerauftrag *TAN-Generator Synchronisierung*

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

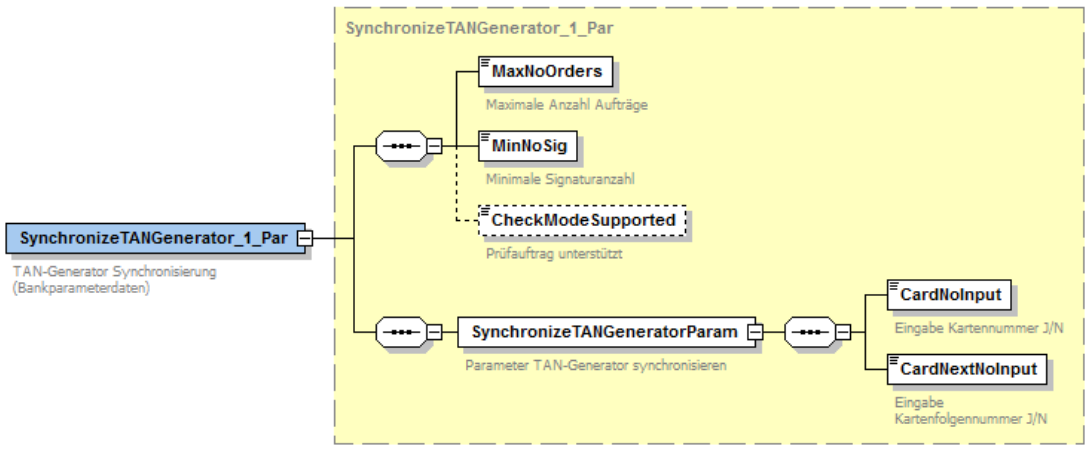


Abbildung 87: Bankparameterdaten TAN-Generator Synchronisierung

III.7.4.8 Mobilfunkverbindung registrieren

Mit diesem Geschäftsvorfall kann ein Benutzer seine Mobilfunkverbindung registrieren. (siehe [PINTAN], Abschnitt *II.8.2.5 Mobilfunkverbindung registrieren*).

a) Benutzerauftrag

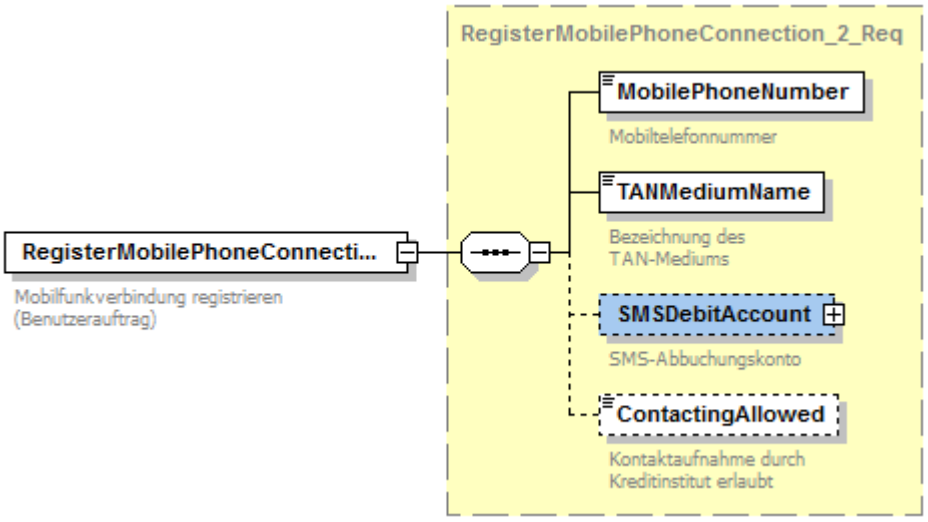


Abbildung 88: Benutzerauftrag Mobilfunkverbindung registrieren

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

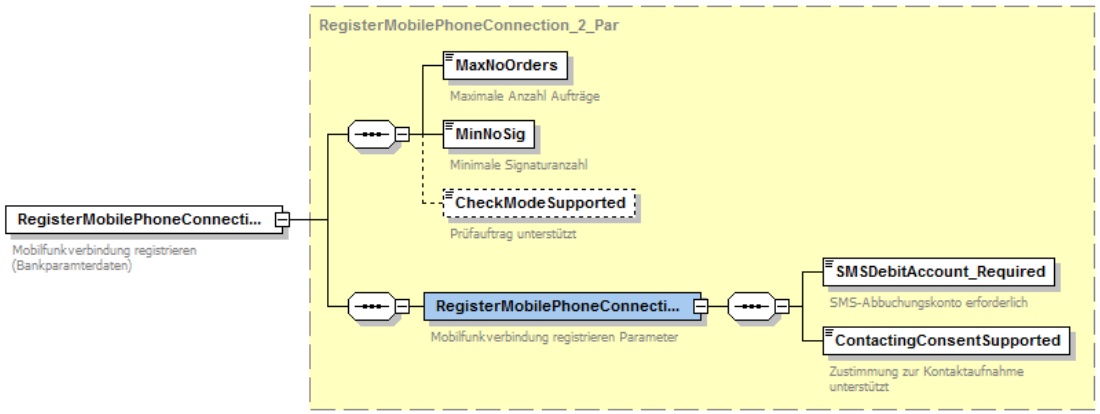


Abbildung 89: Bankparameterdaten Mobilfunkverbindung registrieren

III.7.4.9 Mobilfunkverbindung freischalten

Mit diesem Geschäftsvorfall kann ein Benutzer seine Mobilfunkverbindung freischalten. (siehe [PINTAN], Abschnitt *II.8.2.6 Mobilfunkverbindung freischalten*).

a) Benutzerauftrag

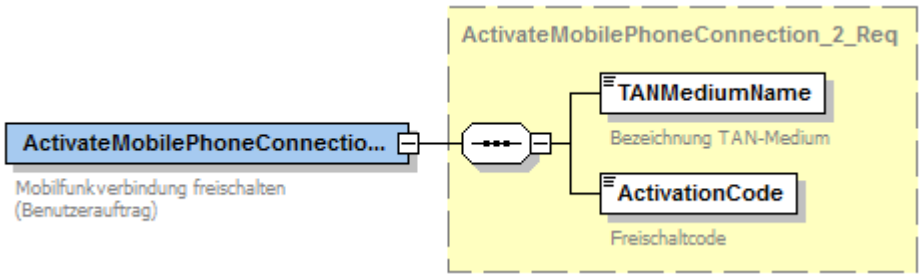


Abbildung 90: Benutzerauftrag Mobilfunkverbindung *freischalten*

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

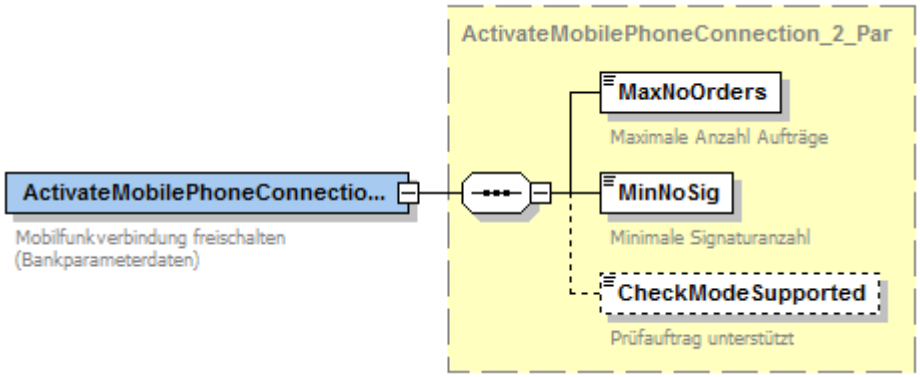


Abbildung 91: Bankparameterdaten Mobilfunkverbindung *freischalten*

III.7.4.10 Mobilfunkverbindung ändern

Mit diesem Geschäftsvorfall kann ein Benutzer seine Mobilfunkverbindung ändern. (siehe [PINTAN], Abschnitt *II.8.2.7 Mobilfunkverbindung ändern*).

a) Benutzerauftrag

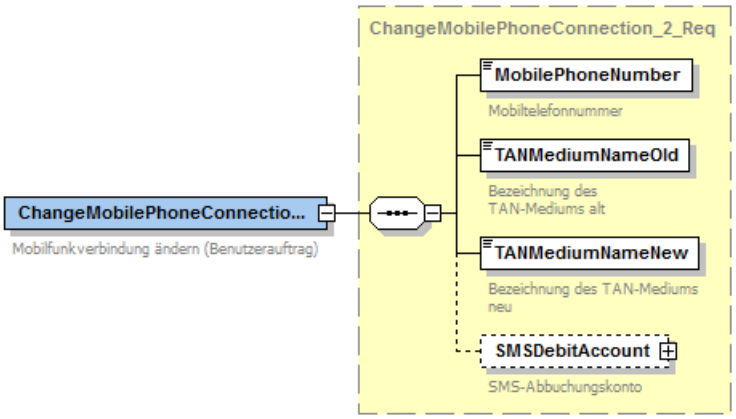


Abbildung 92: Benutzerauftrag Mobilfunkverbindung ändern

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

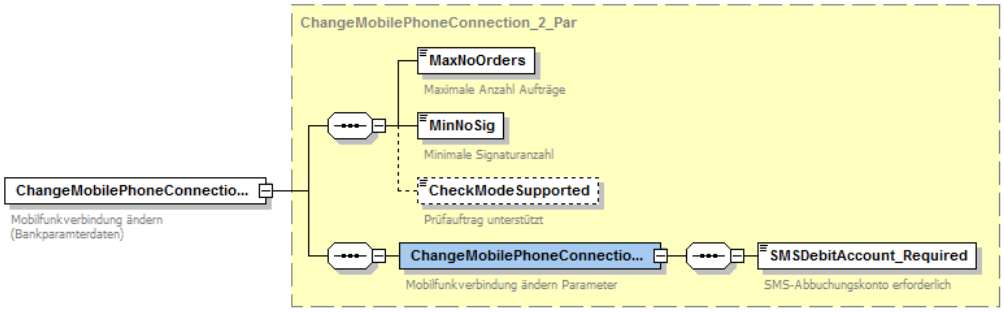


Abbildung 93: Bankparameterdaten Mobilfunkverbindung ändern

III.7.4.11 Deaktivieren / Löschen von TAN-Medien

Mit diesem Geschäftsvorfall kann ein Benutzer sein TAN-Medium (TAN-Generator bzw. Mobilfunkverbindung) deaktivieren oder löschen. (siehe [PINTAN], Abschnitt *II.8.2.8 Deaktivieren / Löschen von TAN-Medien*).

a) Benutzerauftrag

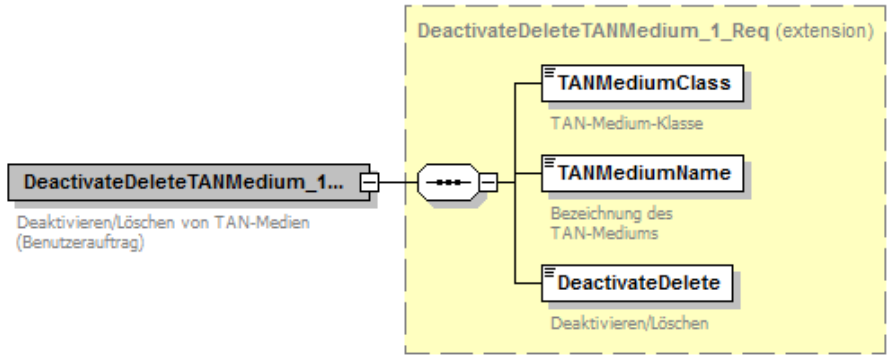


Abbildung 94: Benutzerauftrag *Deaktivieren / Löschen von TAN-Medien*

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

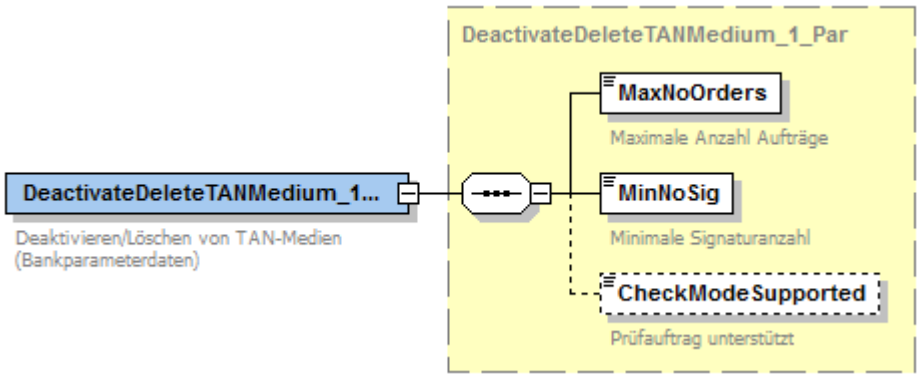


Abbildung 95: Bankparameterdaten *Deaktivieren / Löschen von TAN-Medien*

III.7.5 Abonnement

III.7.5.1 Abonnement einreichen

Mit diesem Auftrag kann ein Benutzer eine Subscription (Abonnement) einrichten (siehe [Formals], Abschnitt *III.8 Das Publish/Subscribe-Verfahren*).

a) Benutzerauftrag

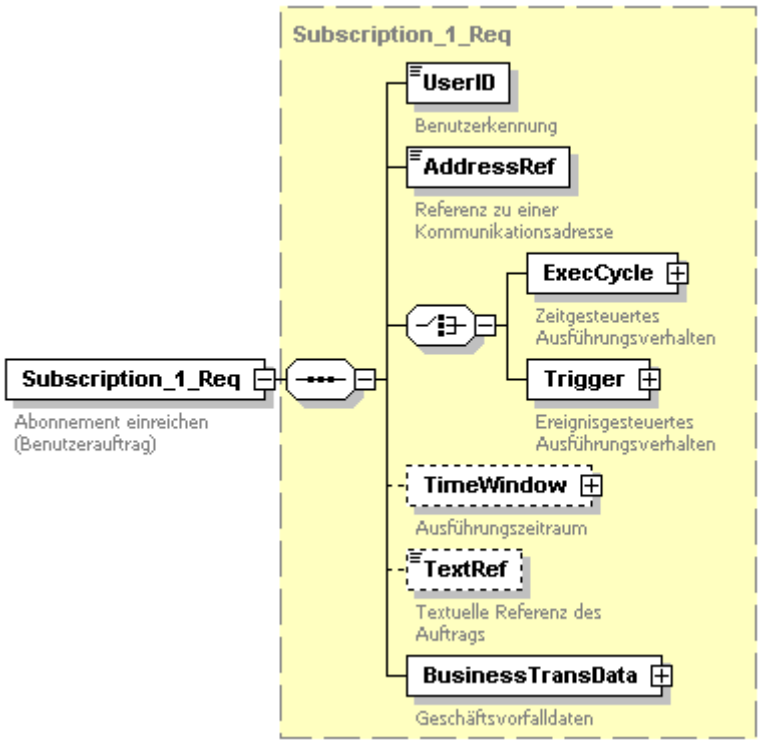


Abbildung 96: Benutzerauftrag Abonnement einreichen

b) Kreditinstitutsrückmeldung

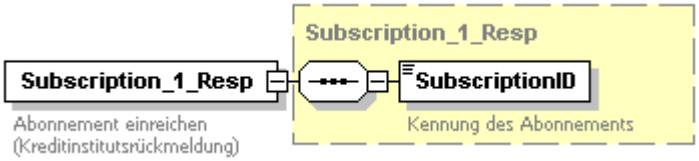


Abbildung 97: Kreditinstitutsrückmeldung Abonnement einreichen

c) Bankparameterdaten

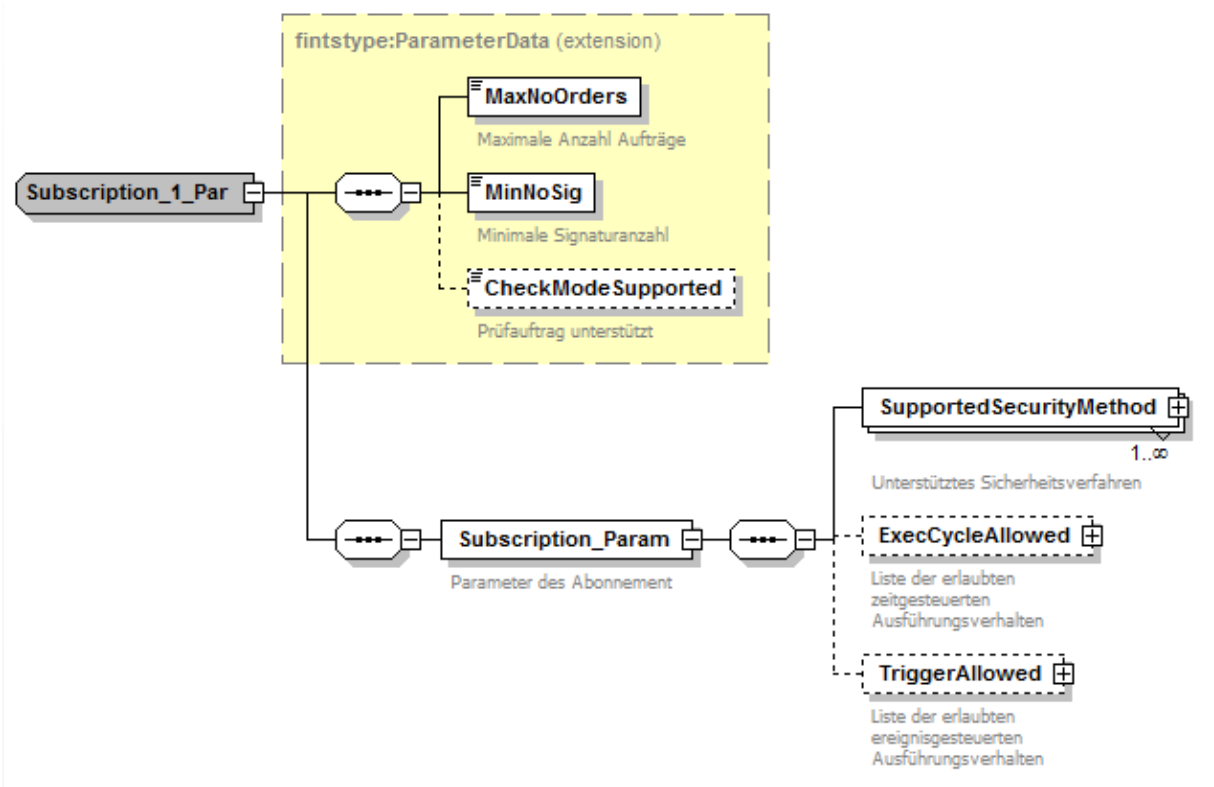


Abbildung 98: Bankparameterdaten Abonnement einreichen

Die Parameter dieses Geschäftsvorfalls enthalten insbesondere die Liste, der in einem bestimmten Sicherheitsverfahren abonnierbaren Geschäftsvorfälle, in den Elementen *SupportedSecurityMethod*:

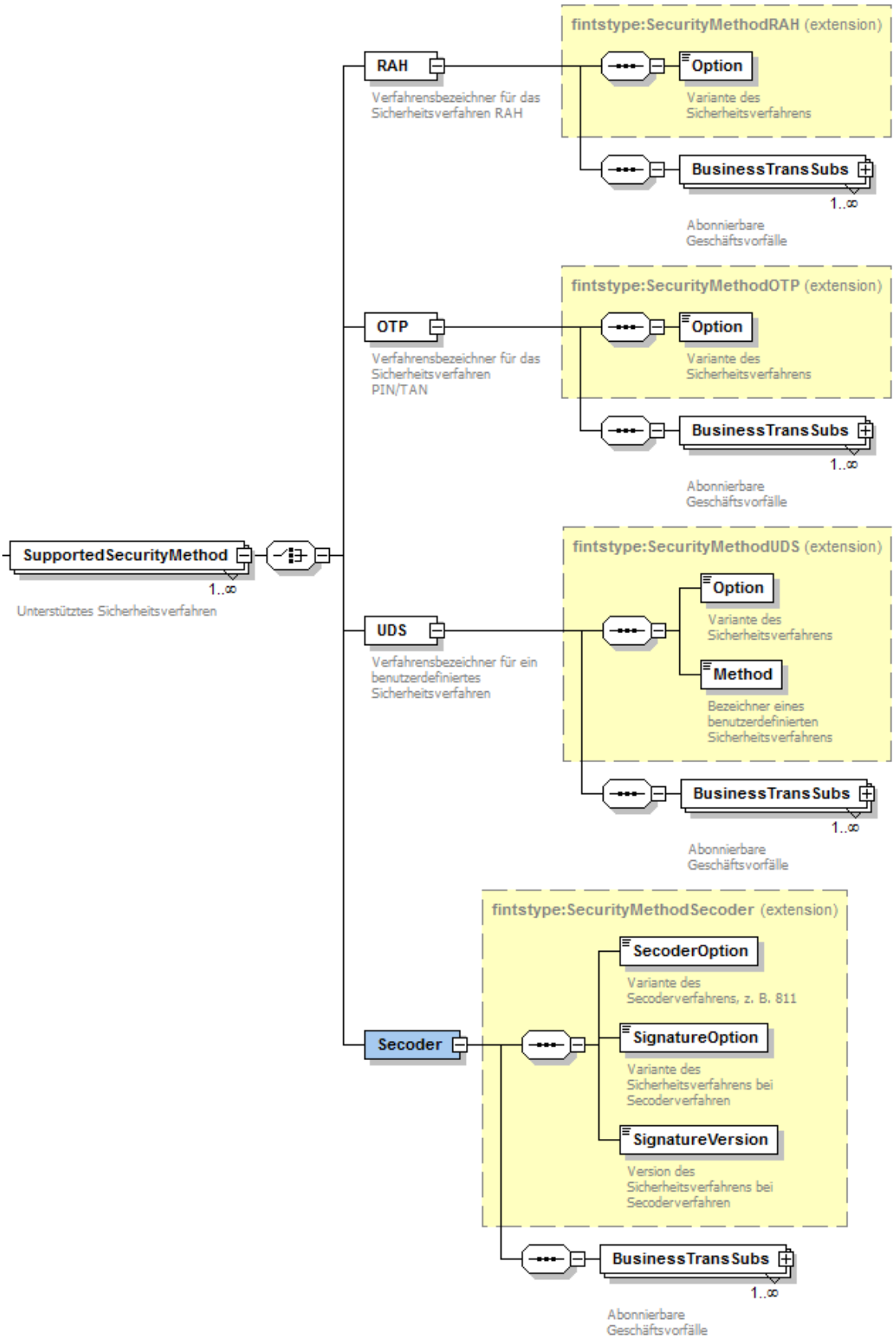


Abbildung 99: Unterstützte Sicherheitsverfahren

III.7.5.2 Abonnement löschen

Mit diesem Auftrag kann ein Benutzer ein Abonnement löschen (siehe [Formals], Abschnitt *III.8 Das Publish/Subscribe-Verfahren*).

a) Benutzerauftrag

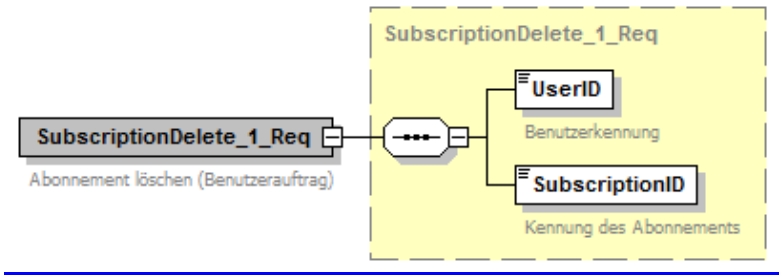


Abbildung 100: Benutzerauftrag Abonnement löschen

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

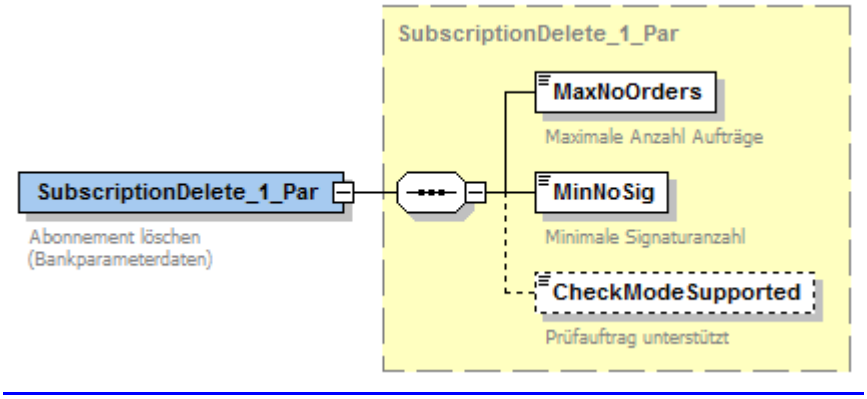


Abbildung 101: Bankparameterdaten Abonnement löschen

III.7.5.3 Abonnementsinformationen anfordern

Mit diesem Auftrag kann ein Benutzer Informationen über Abonnements abrufen (siehe [Formals], Abschnitt *III.8 Das Publish/Subscribe-Verfahren*).

a) Benutzerauftrag

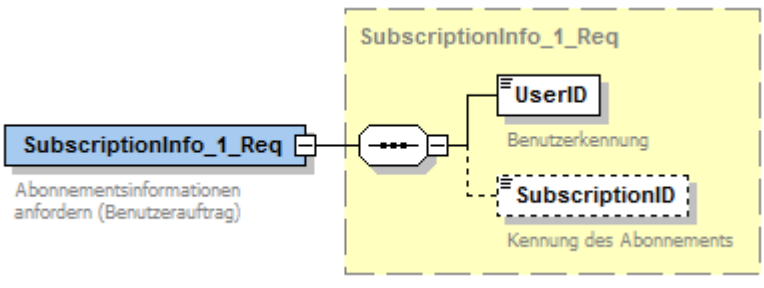


Abbildung 102: Benutzerauftrag Abonnementsinformationen anfordern

b) Kreditinstitutsrückmeldung

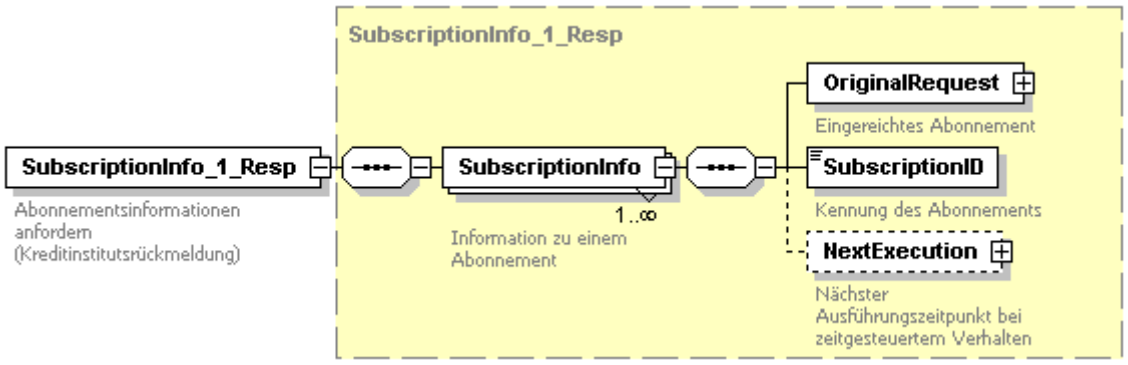


Abbildung 103: Kreditinstitutsrückmeldung Abonnementsinformationen anfordern

c) Bankparameterdaten

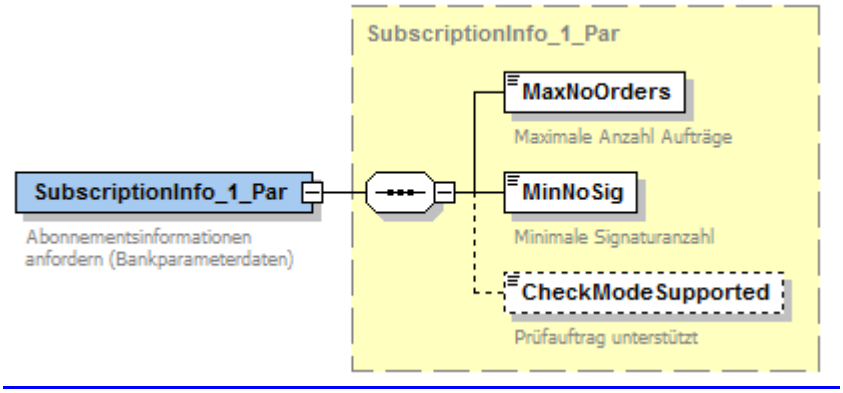


Abbildung 104: Bankparameterdaten Abonnementsinformationen anfordern

III.7.6 Adressenregistrierung

Für asynchrone Kommunikation werden Adressen des Benutzers verwendet, die dieser mit den hier aufgeführten Geschäftsvorfällen registrieren und verwalten kann.

III.7.6.1 Adresse registrieren

Mit diesem Auftrag kann ein Benutzer eine Adresse registrieren (siehe [Formals], Abschnitt *III.4 Adressregistrierung*)

a) Benutzerauftrag

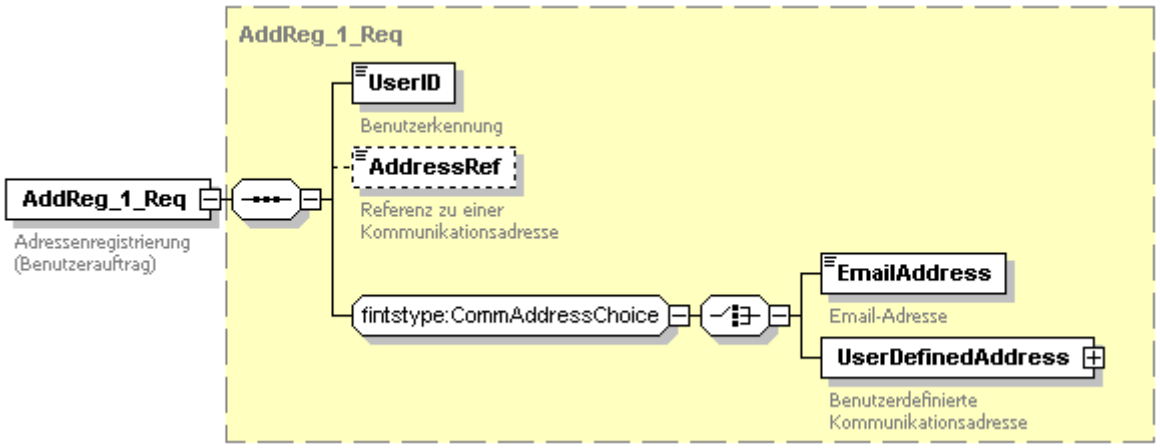


Abbildung 105: Benutzerauftrag Adresse registrieren

b) Kreditinstitutsrückmeldung

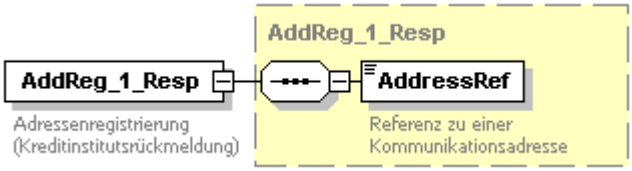


Abbildung 106: Kreditinstitutsrückmeldung Adresse registrieren

c) Bankparameterdaten

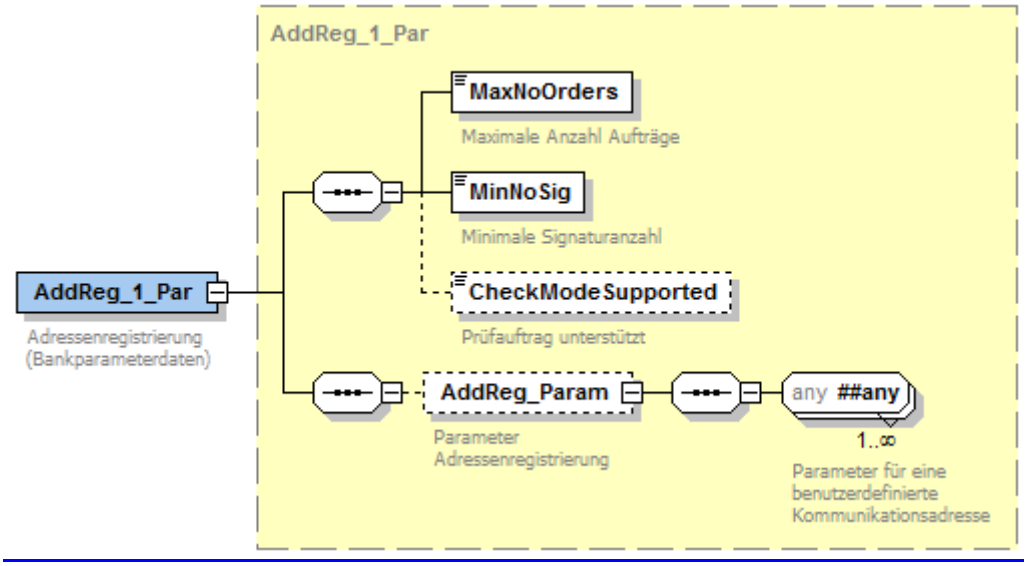


Abbildung 107: Bankparameterdaten Adresse registrieren

III.7.6.2 Adressregistrierungsinformationen holen

Mit diesem Auftrag kann ein Benutzer sich die Adressinformationen geben lassen (siehe [Formals], Abschnitt *III.4 Adressregistrierung*).

a) Benutzerauftrag

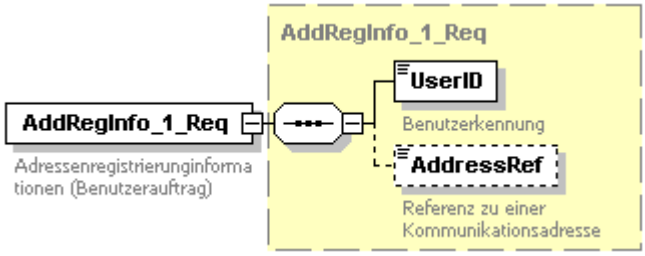


Abbildung 108: Benutzerauftrag Adressregistrierungsinformationen holen

b) Kreditinstitutsrückmeldung

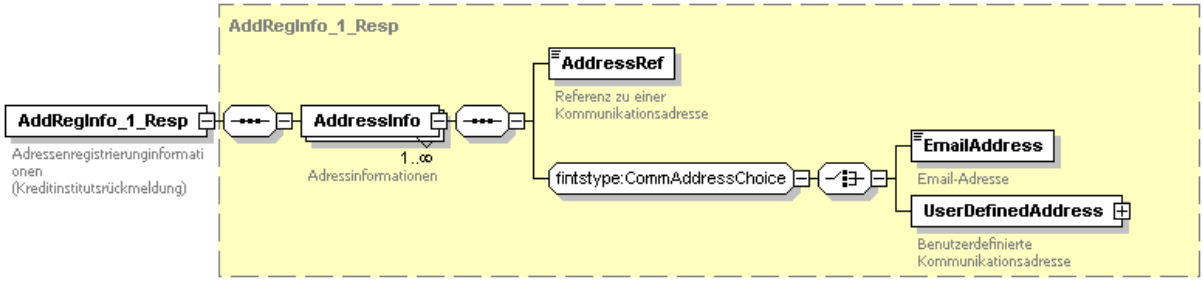


Abbildung 109: Kreditinstitutsrückmeldung Adressregistrierungsinformationen holen

c) Bankparameterdaten

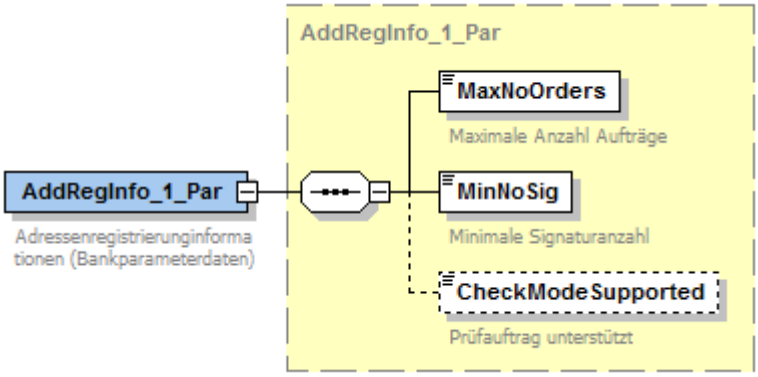


Abbildung 110: Bankparameterdaten Adressregistrierungsinformationen holen

III.7.6.3 Adressregistrierung löschen

Mit diesem Auftrag kann ein Benutzer die Adressinformationen löschen (siehe [Formals], Abschnitt *III.4 Adressregistrierung*).

a) Benutzerauftrag

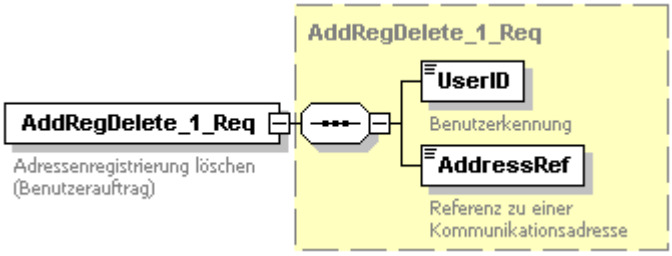


Abbildung 111: Benutzerauftrag Adressregistrierung löschen

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

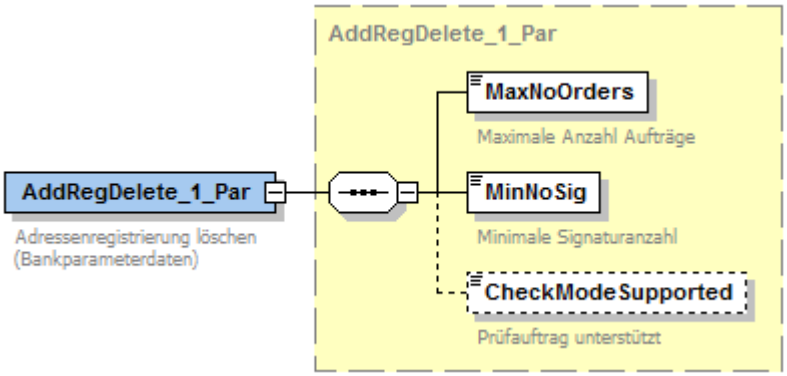


Abbildung 112: Bankparameterdaten Adressregistrierung löschen

III.7.7 Bestätigungen

Für bestimmte Aufträge kann ein Bestätigungsverfahren wie eine Quittung (Zugangsgerät) oder Willenserklärung (natürliche Person) vorgeschrieben sein, der Erhalt der Kreditinstitutsnachricht muss dann mit dem entsprechenden hier gezeigten administrativen Auftrag bestätigt werden.

III.7.7.1 Quittung

Mit diesem Auftrag bestätigt das Zugangsgerät eines Benutzers den Erhalt einer Antwort (siehe [Formals], Abschnitt *III.5 Quittierung von Aufträgen*).

a) Benutzerauftrag

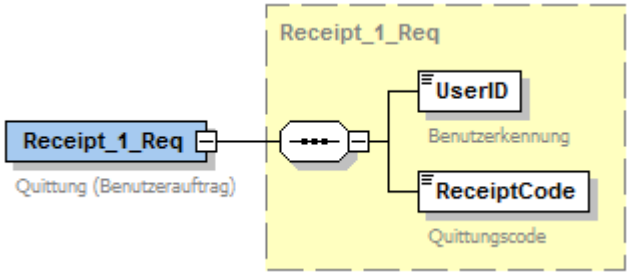


Abbildung 113: Benutzerauftrag Quittung

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

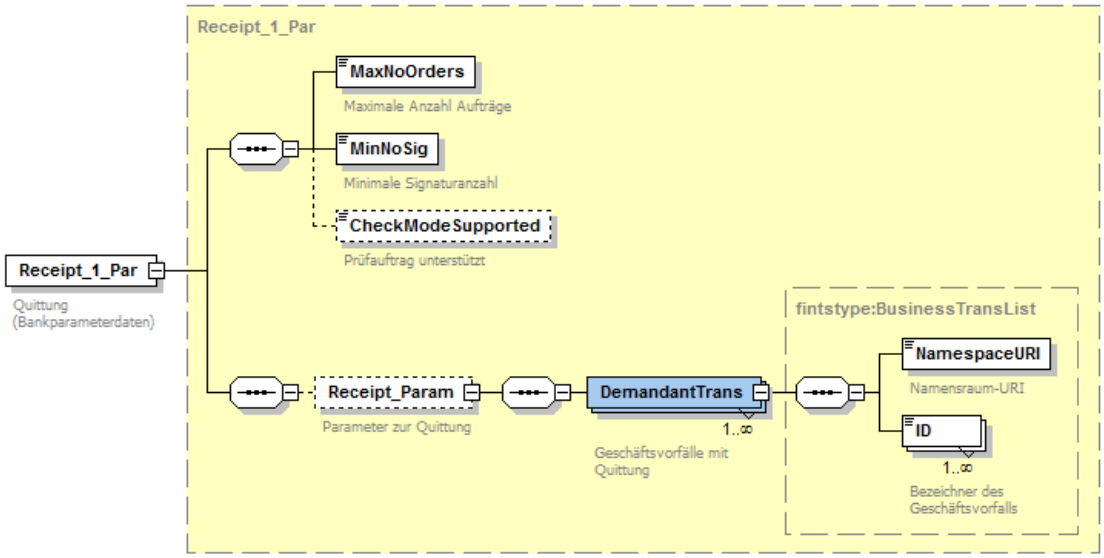


Abbildung 114: Bankparameterdaten Quittung

III.7.7.2 Willenserklärung

Mit diesem Auftrag bestätigt ein Benutzer (als Person) den Erhalt einer Antwort (siehe [Formals], Abschnitt *III.5.2 Willenserklärung*).

a) Benutzerauftrag

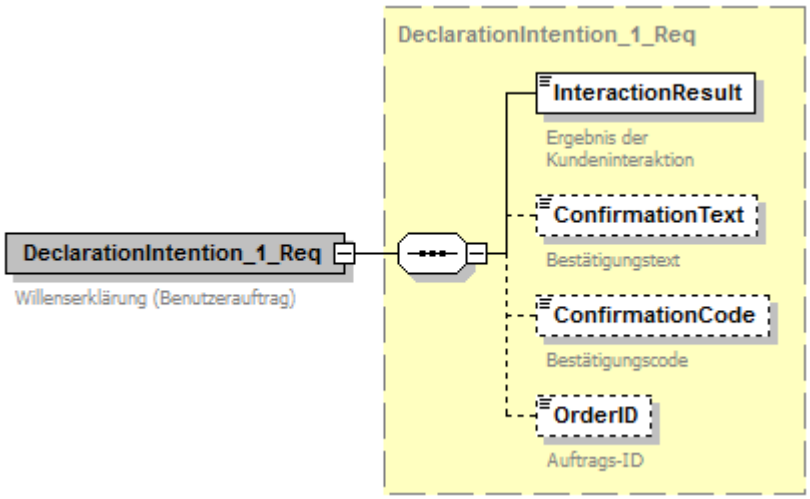


Abbildung 115: Benutzerauftrag Willenserklärung

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

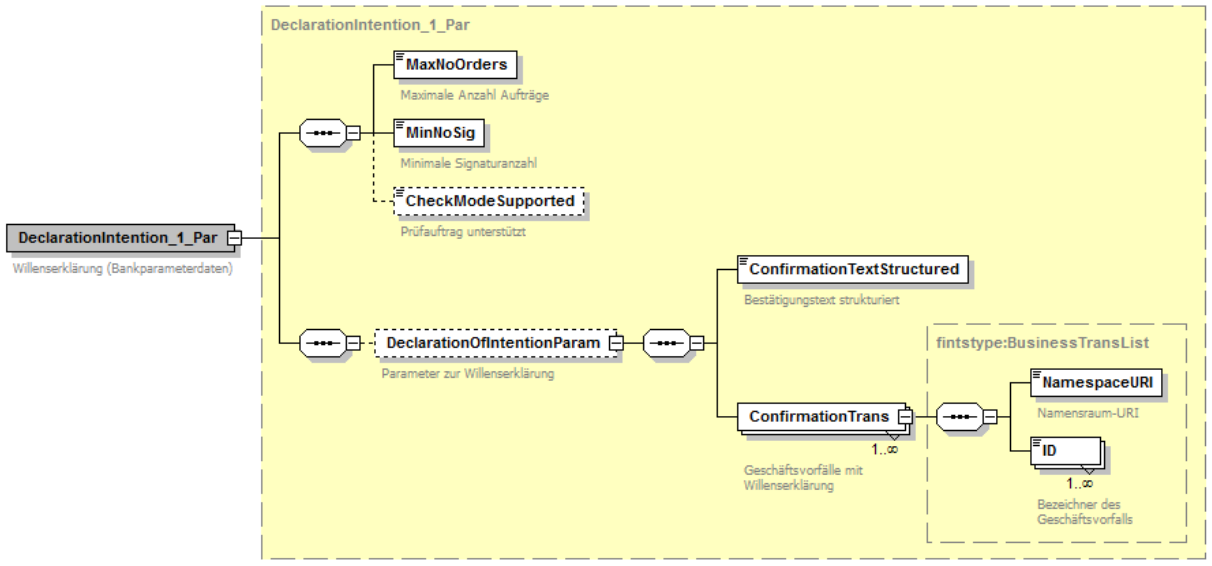


Abbildung 116: Bankparameterdaten Willenserklärung

III.7.8 Verteile Signaturen

Das Verfahren zu verteilten Signaturen ermöglicht es, einen Auftrag beim Kreditinstitut lagern zu lassen und die benötigten Signaturen sukzessive nachzureichen, bis der Auftrag schließlich verarbeitungsfähig ist.

III.7.8.1 Auftrag mit verteilten Signaturen einreichen

Mit diesem Auftrag kann ein Benutzer einen Auftrag einreichen, der später noch von anderen Benutzern signiert werden muss (siehe [Formals], Abschnitt *III.7 Verteilte Signaturen*).

a) Benutzerauftrag

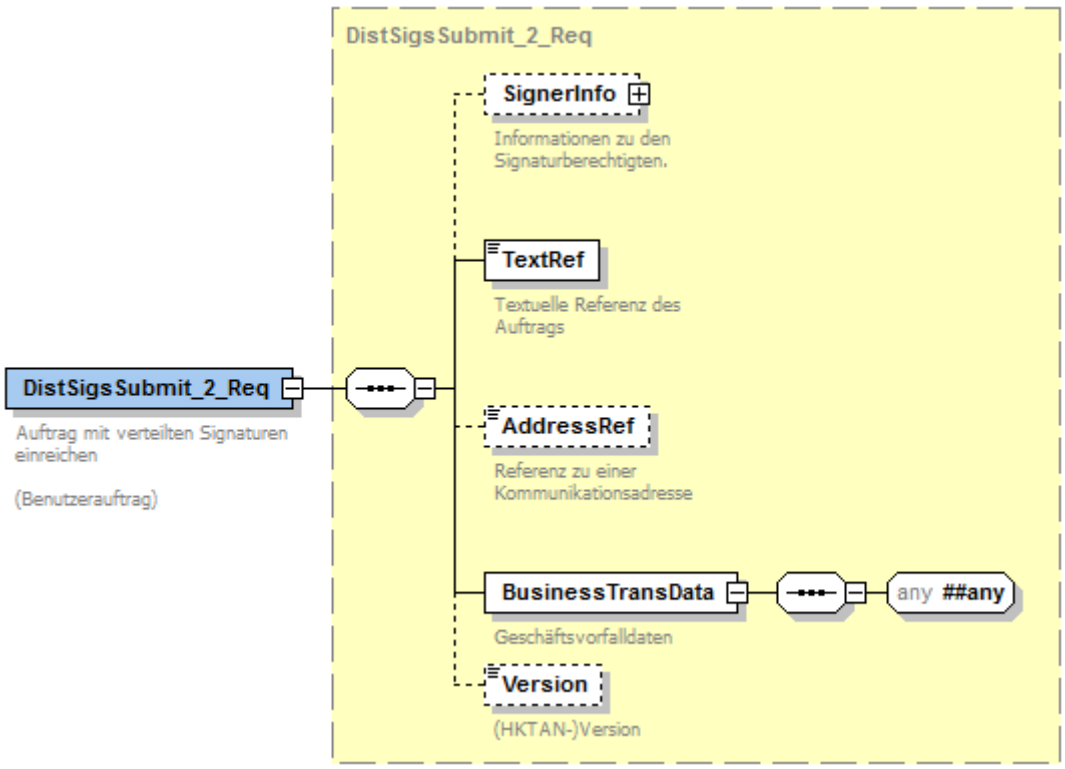


Abbildung 117: Benutzerauftrag Auftrag mit verteilten Signaturen einreichen

Informationen zu den Signaturberechtigten

Wenn die Liste der Signierenden vorhanden ist, wird ihre Interpretation durch eine bilaterale Vereinbarung zwischen Kreditinstitut und Benutzer festgelegt. Ansonsten legt das Kreditinstitut die Signierenden fest.

b) Kreditinstitutsrückmeldung

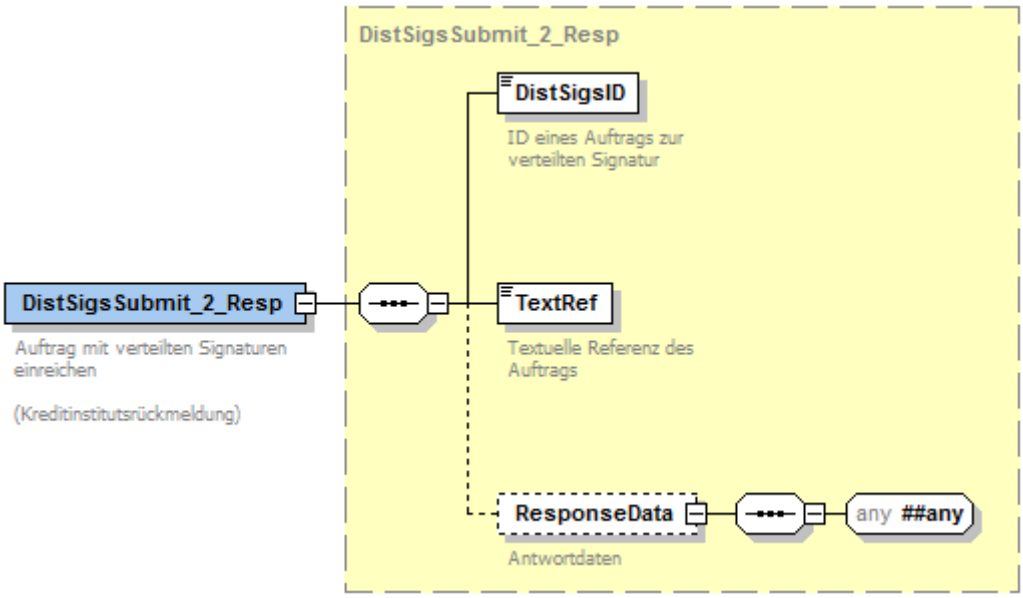


Abbildung 118: Kreditinstitutsrückmeldung Auftrag mit verteilten Signaturen einreichen

c) Bankparameterdaten

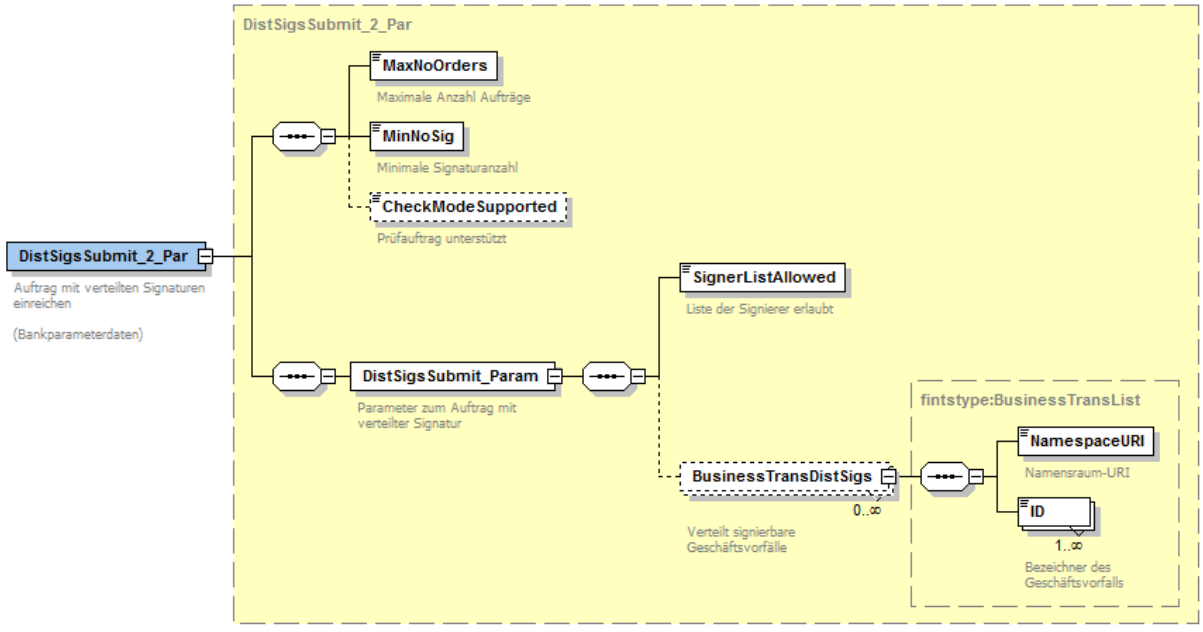


Abbildung 119: Bankparameterdaten Auftrag mit verteilten Signaturen einreichen

III.7.8.2 Informationen zu Auftrag mit verteilten Signaturen

Mit diesem Auftrag kann ein Benutzer sich Informationen zu Aufträgen geben lassen, für die er weitere Signaturen erstellen möchte, insbesondere die zu signierenden Daten selbst (siehe [Formals], Abschnitt *III.7 Verteilte Signaturen*).

a) Benutzerauftrag

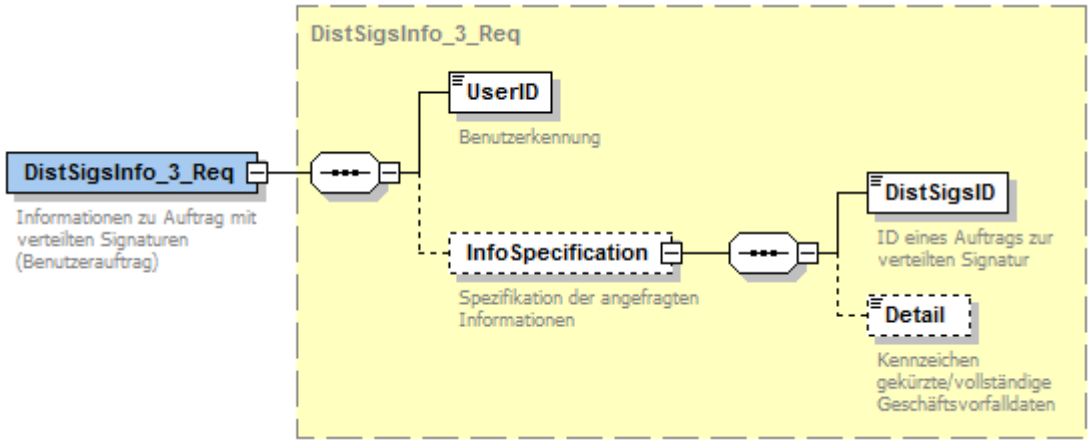


Abbildung 120: Benutzerauftrag Informationen zu Auftrag mit verteilten Signaturen

Detail

Es kann zwischen der gekürzten und der vollen Darstellung des zu signierenden Geschäftsvorfalls gewählt werden. Die zugehörigen Belegungen des Feldes sind *abbreviated* und *complete*.

b) Kreditinstitutsrückmeldung

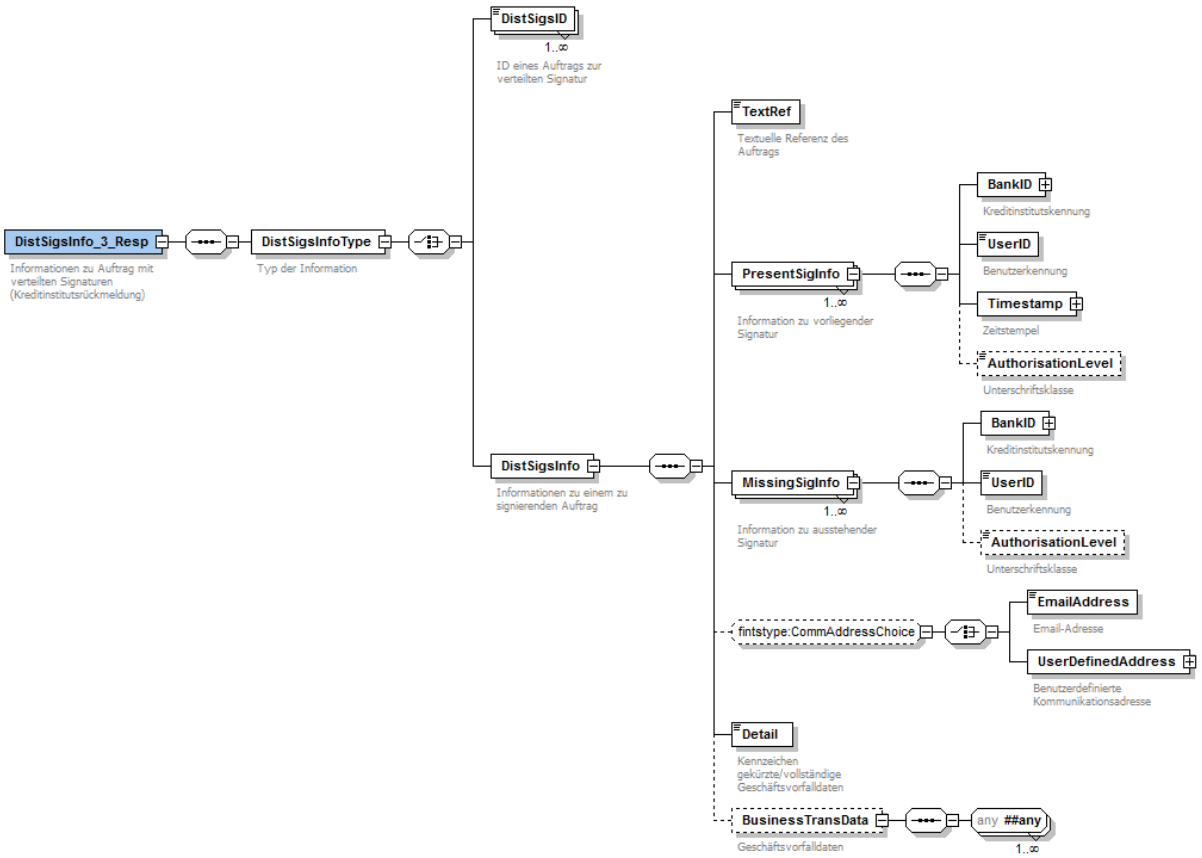


Abbildung 121: Kreditinstitutsrückmeldung Informationen zu Auftrag mit verteilten Signaturen

Geschäftsvorfalldaten

Auftrag oder Auftragsteil, den der Benutzer mit dem Folgeauftrag ‚Auftrag mit verteilter Signatur signieren' signieren soll.

c) Bankparameterdaten

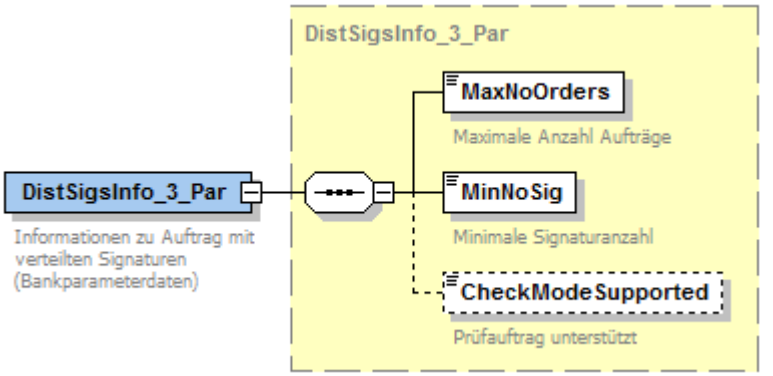


Abbildung 122: Bankparameterdaten Informationen zu Auftrag mit verteilten Signaturen

III.7.8.3 Auftrag mit verteilten Signaturen signieren

Mit diesem Auftrag kann ein Benutzer einen eingereichten Auftrag mit einer weiteren Signatur versehen (siehe [Formals], Abschnitt *III.9 Verteilte Signaturen*).

a) Benutzerauftrag

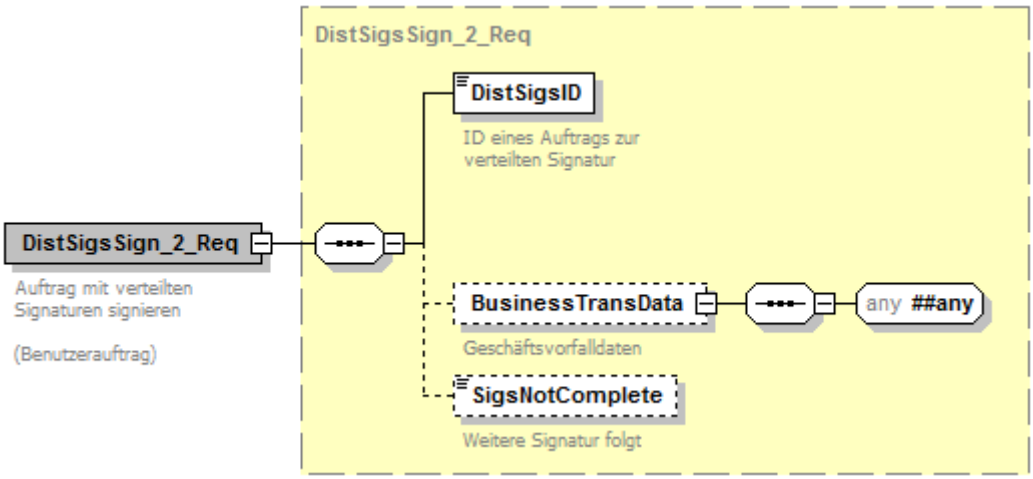


Abbildung 123: Benutzerauftrag Auftrag mit verteilten Signaturen signieren

Auftrag/Auftragsteil

Auftrag oder Auftragsteil, der dem Benutzer bei einer Anforderung zugesandt wurde. Dieser wird vom Benutzer signiert.

b) Kreditinstitutsrückmeldung

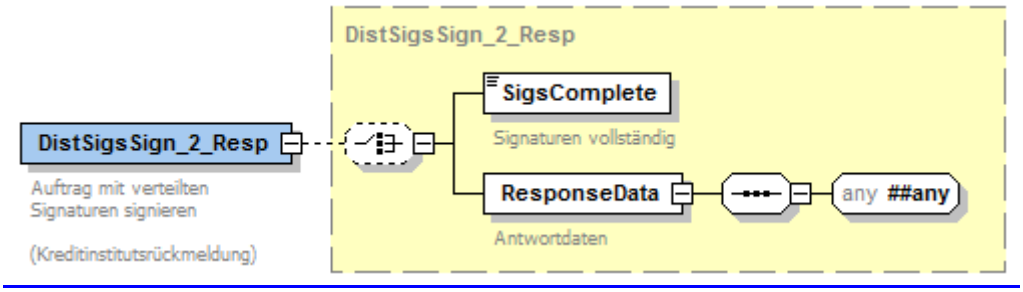


Abbildung 124: Kreditinstitutsrückmeldung Auftrag mit verteilten Signaturen signieren

c) Bankparameterdaten

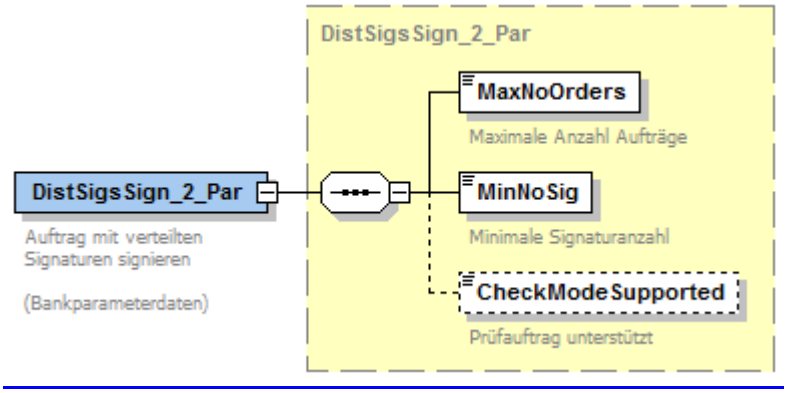


Abbildung 125: Bankparameterdaten Auftrag mit verteilten Signaturen signieren

III.7.8.4 Auftrag mit verteilten Signaturen löschen

Mit diesem Auftrag kann ein Benutzer einen Auftrag löschen, der zur verteilten Signatur eingereicht wurde (siehe [Formals], Abschnitt *III.9 Verteilte Signaturen*).

a) Benutzerauftrag

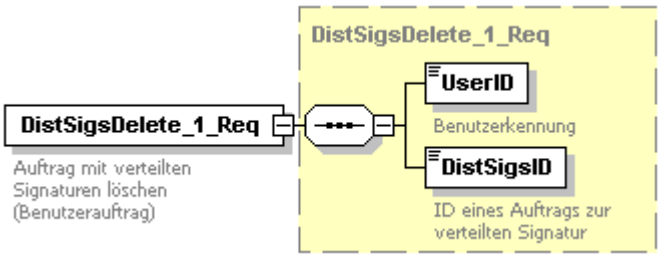


Abbildung 126: Benutzerauftrag Auftrag mit verteilten Signaturen löschen

b) Kreditinstitutsrückmeldung

Die Kreditinstitutsrückmeldung ist leer.

c) Bankparameterdaten

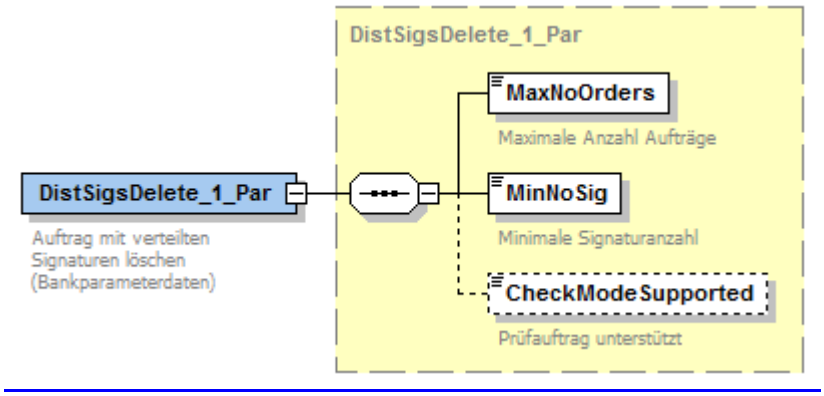


Abbildung 127: Bankparameterdaten Auftrag mit verteilten Signaturen löschen

III.7.9 Statusprotokoll

Im Statusprotokoll kann das Kreditinstitut optional den Bearbeitungsstatus und die Rückmeldungen der vom Benutzer eingereichten Nachrichten und Aufträge verwalten.

III.7.9.1 Statusprotokoll

Mit diesem Auftrag kann ein Benutzer sich das Statusprotokoll anzeigen lassen (siehe [Formals], Abschnitt *III.2 Statusprotokoll*).

a) Benutzerauftrag

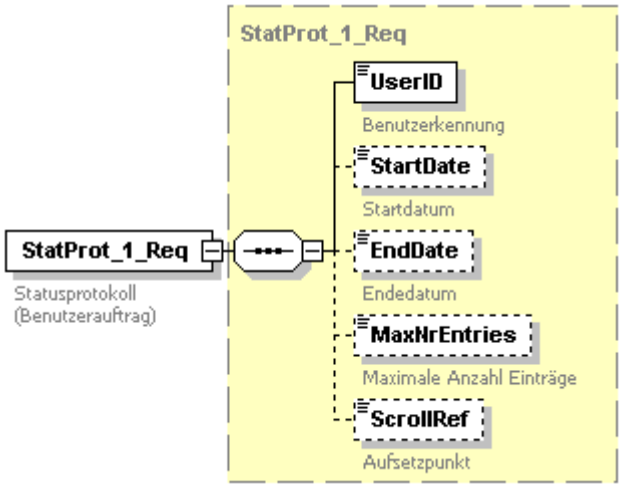


Abbildung 128: Benutzerauftrag Statusprotokoll

b) Kreditinstitutsrückmeldung

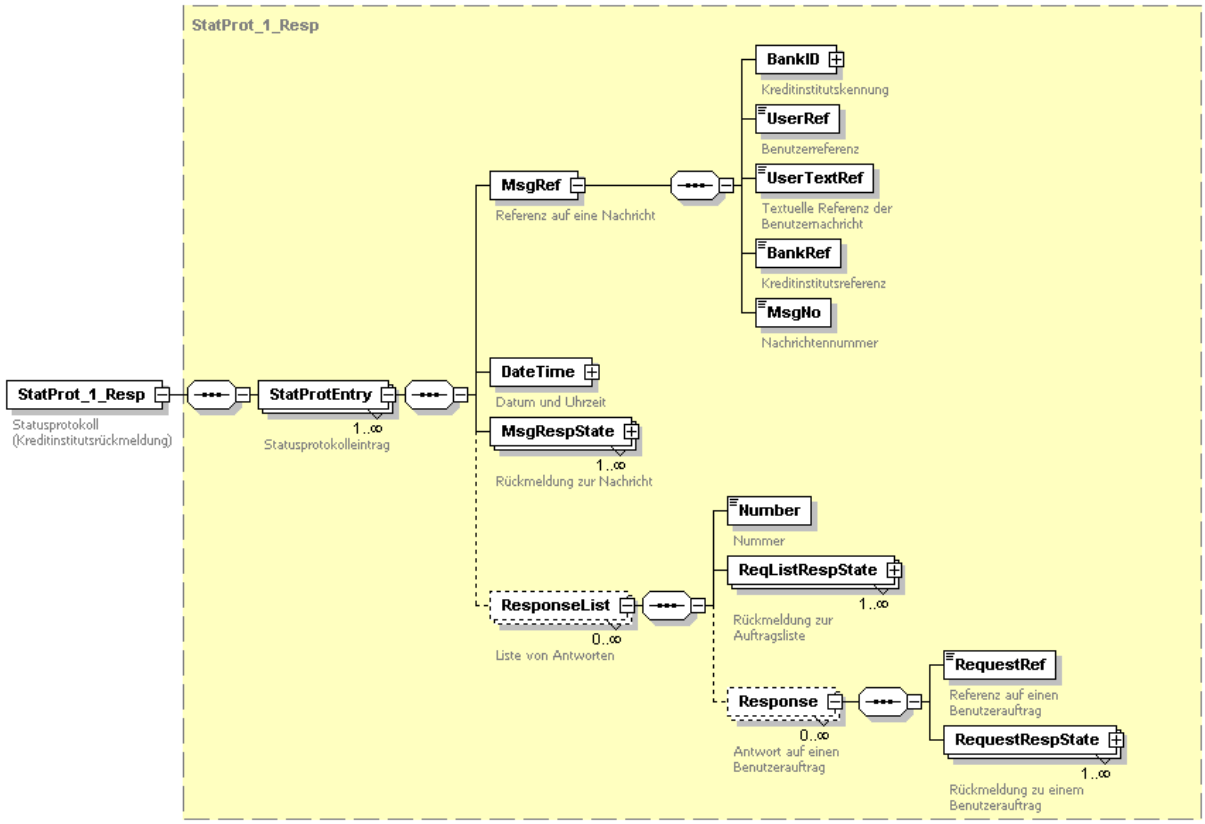


Abbildung 129: Kreditinstitutsrückmeldung Statusprotokoll

Nummer

Das Element bezeichnet die Nummer der Antwortliste der Kreditinstitutsnachricht, beginnend mit Eins.

c) Bankparameterdaten

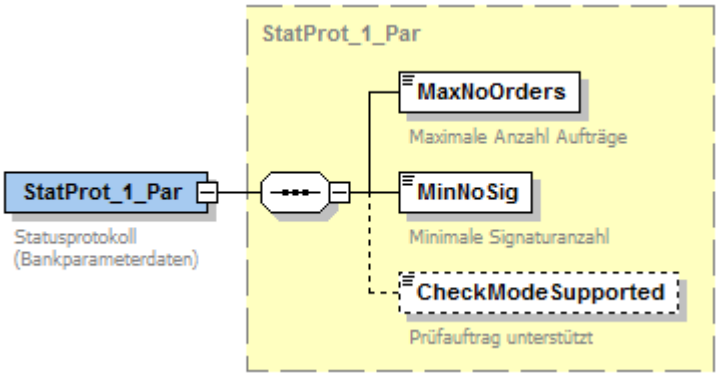


Abbildung 130: Bankparameterdaten Statusprotokoll

IV. SIGNIERTE NACHRICHTENTEILE

IV.1 Überblick

Die FinTS-Syntax spezifiziert drei unterschiedliche Nachrichtentypen, die für verschiedene Sicherheitsverfahren zum Einsatz kommen. Abschnitt *IV.2 Signaturtypen* beschreibt diese Typen.

In FinTS-Nachrichten können Nachrichten auf zwei Ebenen signiert werden. Die Botensignatur bezieht sich auf alle Aufträge im Nachrichtenkörper. Eine Auftragssignatur hingegen bezieht sich auf Geschäftsvorfälle bzw. administrative Aufträge eines Auftragsteils.

Belegungshinweise und Beispiele für Boten- und Auftragssignaturen finden sich in den Abschnitten *IV.3 Botensignatur* bzw. *IV.4 Auftragssignatur*.

Bei der Angabe von Grammatiken für die Bildung von Lokalisierungspfaden gelten die Festlegungen in *II.8 Referenzierung mit XPath-Ausdrücken*.

IV.2 Signaturtypen

Auf beiden Nachrichtenebenen können drei verschiedene Signaturformen auftreten. Die vier Signaturformen werden für Benutzernachrichten durch die Gruppe *SigChoiceReq* modelliert. Das Inhaltsmodell von *SigChoiceReq* ist so definiert, dass wahlweise [XML Signature]-konform (siehe auch Sicherheitsverfahren [HBCI], Abschnitt *II.5.1 Signatur-Segment bzw. Abschnitt III Secoder-Integration*), nach dem PIN/TAN-Verfahren (siehe auch Sicherheitsverfahren [PINTAN], Abschnitt *II.5.1 PIN/TAN-Signatur*) oder nach einem benutzerdefinierten Verfahren signiert werden kann.

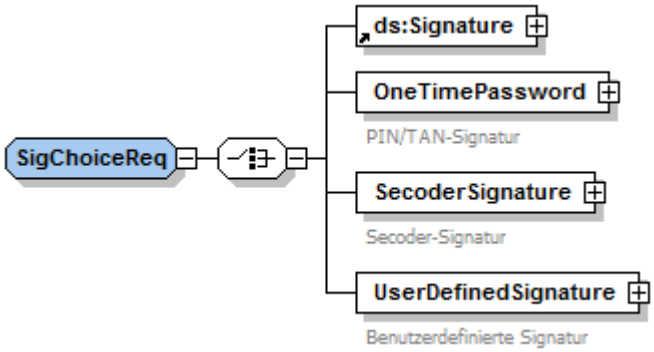


Abbildung 131: Signatur einer Benutzernachricht

Für die Kreditinstitutsnachrichten existieren dieselben Signaturformen. Analog zur Auftragssignatur enthält die Gruppe *SigChoiceResp* eine Auswahlliste mit den vier Signaturformen:

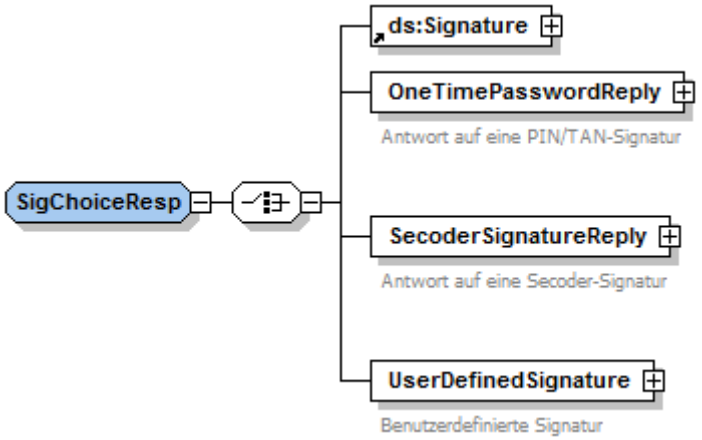


Abbildung 132: Signatur einer Kreditinstitutsnachricht

Die folgenden Abschnitte beschreiben das W3C-konforme Signaturverfahren (HBCI mit oder ohne Secoder), das PIN/TAN-Verfahren und die benutzerdefinierte Signatur.

IV.2.1 [HBCI]-Verfahren: W3C-konforme XML-Signatur

Eine XML-Signatur nach Spezifikation des W3C hat den folgenden Aufbau:

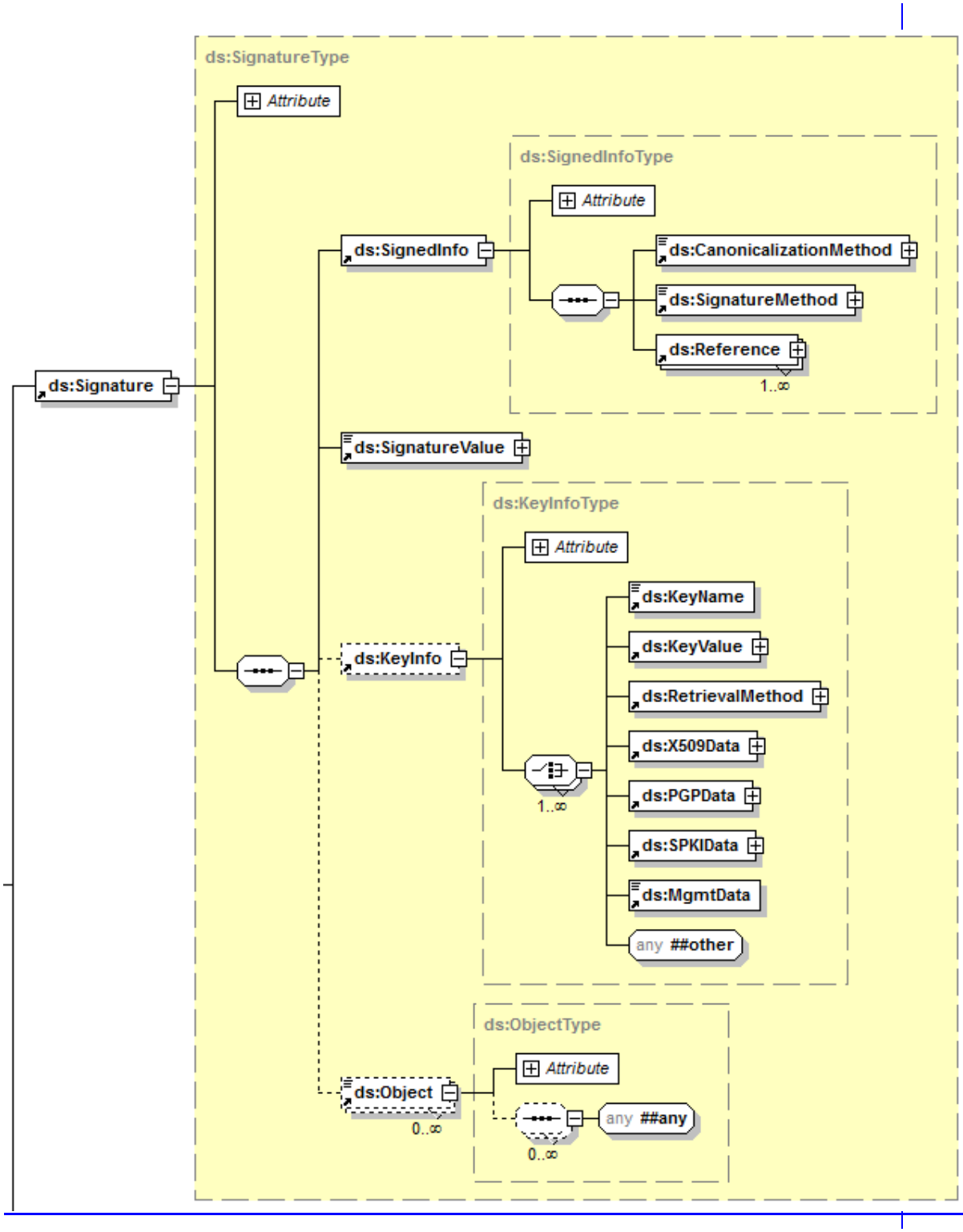


Abbildung 133: Element Signature

Signaturen für Benutzernachrichten und Kreditinstitutsnachrichten sind gleich aufgebaut. Zur Verwendung und Belegung der Signaturen siehe auch [HBCI], Abschnitt *II.5.1 Signatur-Segment*.

- **Namespace**

  Die Elemente dieser Signatur liegen im Namensraum *http://www.w3.org/2000/09/xmldsig#*, der an das Präfix *ds* gebunden wird.

- **Kanonisierung**

  Das Element *CanonicalizationMethod* enthält die Angabe des verwendeten Algorithmus zur Kanonisierung des *SignedInfo*-Elements.

  Für FinTS-Anwendungen ist in Boten- und Auftragssignaturen das Kanonisierungsverfahren [Exclusive Canonical XML] vorgeschrieben.

  Erläuterung:

  Die Kanonisierung eines XML-Dokuments ist die Überführung eines XML-Fragments bzw. einer Knotenmenge in einheitlicher Form in einen Bytestrom. Diese einheitliche Form ermöglicht die Prüfung einer Signatur. Manipuliert beispielsweise eine verarbeitende XML-Anwendung ein XML-Dokument durch Einfügen von Leerzeichen in Start- und Endtags oder durch Änderung der Attributreihenfolge, ist die Semantik davon nicht betroffen, die Signaturen jedoch verlieren ihre Gültigkeit. Dieses lässt sich vermeiden, wenn die Signatur auf ein kanonisiertes XML-Fragment gebildet wird.

  Das Verfahren [Exclusive Canonical XML] normalisiert ein XML-Fragment ohne den dazugehörigen Kontext.

  Beispiel:

  Zum Kanonisieren des Elements *SignedInfo* nach dem Verfahren [Exclusive Canonical XML] ist das Attribut *Algorithm* des Elements *CanonicalizationMethod* mit dem Wert *http://www.w3.org/2001/10/xml-exc-c14n#* zu belegen:

```
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
  <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
    PrefixList="ds fintstype fintstrans"/>
</ds:CanonicalizationMethod>
```

  Im Attribut *PrefixList* des Elements *InclusiveNamespaces* werden Namensraum-Präfixe aufgelistet, die in den Lokalisierungspfaden der verwendeten XPath-Ausdrücke, ansonsten aber nicht als Element-Präfix innerhalb der Signatur verwendet werden. Mit dieser Auflistung werden sie bei einer Kanonisierung des *SignedInfo* mit in die kanonisierte Darstellung übernommen, so dass die XPath-Ausdrücke gültig bleiben.

- **Signaturverfahren und Signaturwert**

  Das Signaturverfahren (*SignatureMethod*) im *SignedInfo*-Element legt den Algorithmus fest, der zur Berechnung des Signaturwerts (*SignatureValue*) verwendet wird.

  Die zulässigen Verfahren sind durch die Sicherheitsprofile des Sicherheitsverfahrens [HBCI] festgelegt. Der Signaturwert einer XML-Signatur wird durch Anwendung des Signaturverfahrens auf das kanonisierte *SignedInfo*-Element berechnet und in base64-Codierung in das Element eingestellt.

- **Signierte Inhalte**

  Die durch die XML-Signatur signierten Inhalte werden durch die Referenzen (*Reference*) im *SignedInfo*-Element bestimmt. In FinTS ist genau ein Reference-Element pro Signatur zu verwenden. Die Referenzierung mehrerer Dokumententeile

in einer Signatur wird durch den Vereinigungsoperator im XPath-Ausdruck ermöglicht.

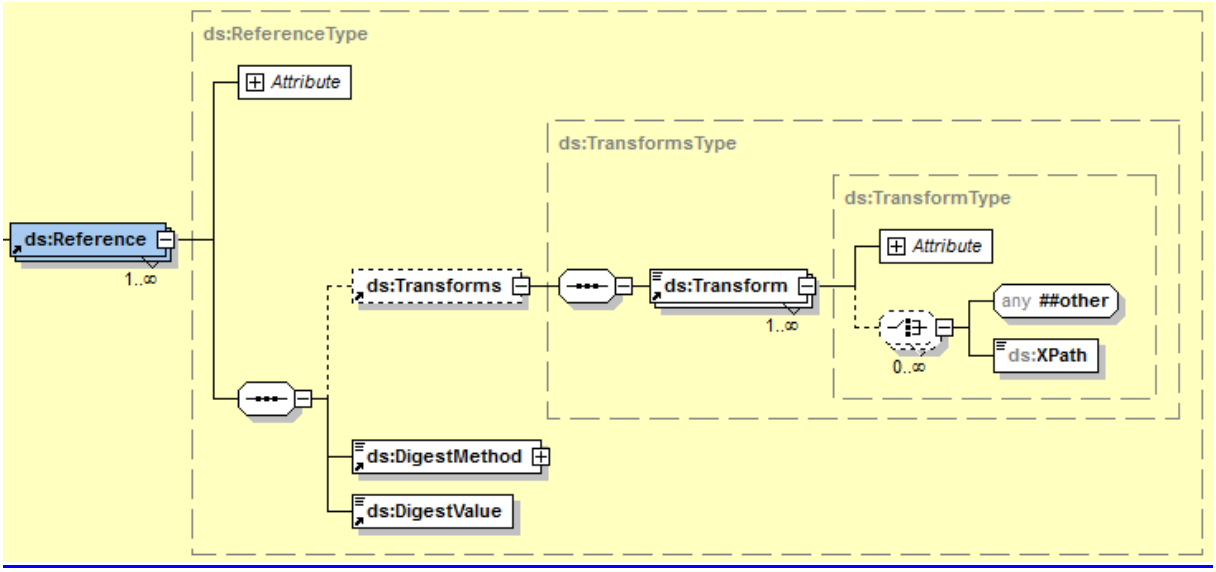


Abbildung 134: Element Reference

Das *Reference*-Element enthält einen Bezeichner (*DigestMethod*) für einen Algorithmus zur Berechnung des Hashwertes, den Hashwert (*DigestValue*), sowie ein *Transforms*-Element, welches die zu signierenden Daten referenziert.

Die zulässigen Hashalgorithmen sind durch die Sicherheitsprofile des Sicherheitsverfahrens festgelegt (siehe [HBCI], Abschnitt *II.1.1 Sicherheitsprofile*). Der Hashwert einer Referenz wird durch Anwendung des Hashalgorithmus auf das Ergebnis der *Transform*-Kette gebildet.

Die Transformation wird durch eine geordnete Abfolge einzelner Transformationsschritte (*Transform*) beschrieben. In FinTS sind genau zwei *Transform*-Elemente verpflichtend:

1. Für den ersten Transformationsschritt ist ausschließlich das Verfahren [XPath Filter], mit genau einem *XPath*-Ausdruck des Filtertyps *intersect* erlaubt. Der einzustellende Ausdruck für das Element *XPath* ist nach den allgemeinen Regeln für Lokalisierungspfade in *II.8 Referenzierung mit XPath-Ausdrücken* zu bilden.

   In den Abschnitten *IV.3 Botensignatur* und *IV.4 Auftragssignatur* wird festgelegt und anhand von Beispielen erläutert, wie diese Regeln für die Verwendung in Boten- und Auftragssignaturen anzuwenden sind.

2. Das Ergebnis des ersten Transformationsschrittes wird im zweiten Schritt in die kanonische Form überführt. Hierzu ist in Boten- und Auftragssignaturen die Verwendung von [Exclusive Canonical XML] Pflicht.

Das *URI*-Attribut des *Reference*-Elements ist in FinTS immer mit der leeren Zeichenkette “” zu belegen

```
<ds:Reference URI="">
  ...
</ds:Reference>
```

- **Schlüsselinformationen**

  Das Element *KeyInfo* in Abbildung 133: Element Signature enthält in Form eines *RAHKeyInfo*-Elements die Informationen über den Signaturschlüssel gemäß Sicherheitsverfahren [HBCI].

  Abbildung 135 zeigt die Schlüsselinformationen für eine XML-Signatur im Sicherheitsverfahren RAH.

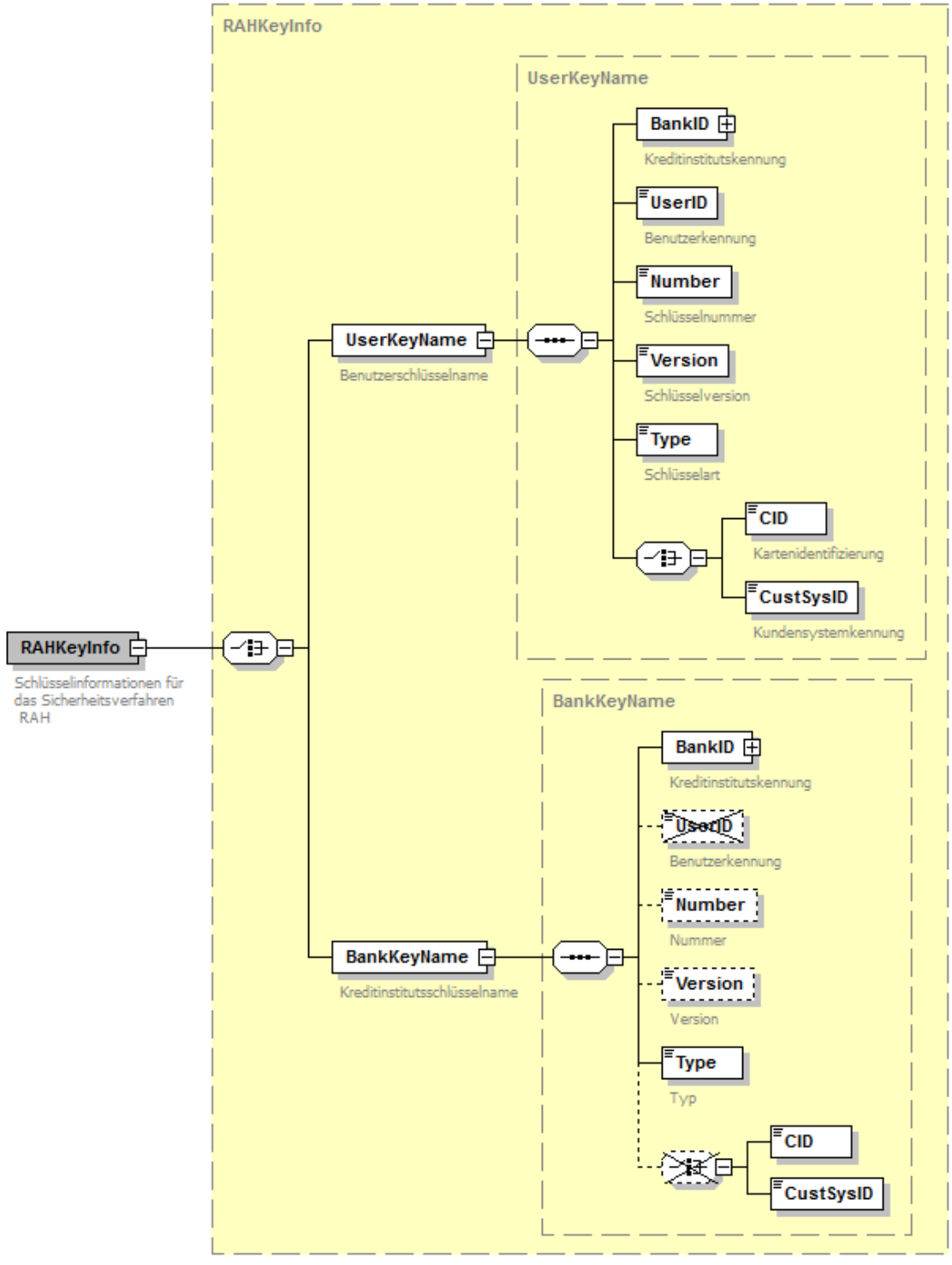


Abbildung 135: Element RAHKeyInfo

Benutzerschlüsselname, Kreditinstitutsschlüsselname

In Benutzernachrichten wird der Benutzerschlüsselname belegt, in Kreditinstitutsnachrichten der Kreditinstitutsschlüsselname.

Kartenidentifizierung

Dieses Feld wird in Kreditinstitutsnachrichten leer gelassen.

Typ

Dieses Element wird mit dem logischen Typ des für die jeweilige Operation notwendigen Schlüssels belegt.

- **Signatureigenschaften**

Unter dem Element *Object* können beliebige Zusatzinformationen zur Signatur als *SignatureProperties* transportiert werden. In FinTS werden dort innerhalb eines *SignatureProperty*-Elements die FinTS-spezifischen Signatureigenschaften in Form eines *RAHProperty*-Elements hinterlegt.

Beispiel:

```
<ds:Signature Id="28099950001Ben-000000012-1228743211628"
   xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
...
  <ds:Object>
    <ds:SignatureProperties>
      <ds:SignatureProperty Target="#28099950001Ben-000000012-1228743211628"">
        <fintsmsg:RAHProperty>
          ...
        </fintsmsg:RAHProperty>
      </ds:SignatureProperty>
    </ds:SignatureProperties>
  </ds:Object>
</ds:Signature>
```

Das Pflichtattribut *Target* verweist auf die Signatur zu der die Signatureigenschaften gehören, in FinTS somit auf das *Signature*-Element, in das die Properties eingebettet sind.

Verweisziel des *Target*-Attributs ist das *Id*-Attribut der Signatur. Diese *ID* muss im gesamten Dokument eindeutig sein.

Damit auch im Intermediärszenario keine Kollisionen zwischen den - von verschiedenen Benutzern erstellten - IDs auftreten, wird für Kundensysteme folgende Belegungsrichtlinie empfohlen: die Id wird gebildet durch Konkatenation von Länderkennzeichen, Kreditinstitutscode, Benutzerkennung, Kundensystemkennung und Signatur-ID.

Das Kreditinstitut muss in seinen Nachrichten ebenfalls auf Eindeutigkeit der IDs innerhalb einer Nachricht achten.

Abbildung 136 zeigt die spezifischen Parameter für eine XML-Signatur im Sicherheitsverfahren RAH.

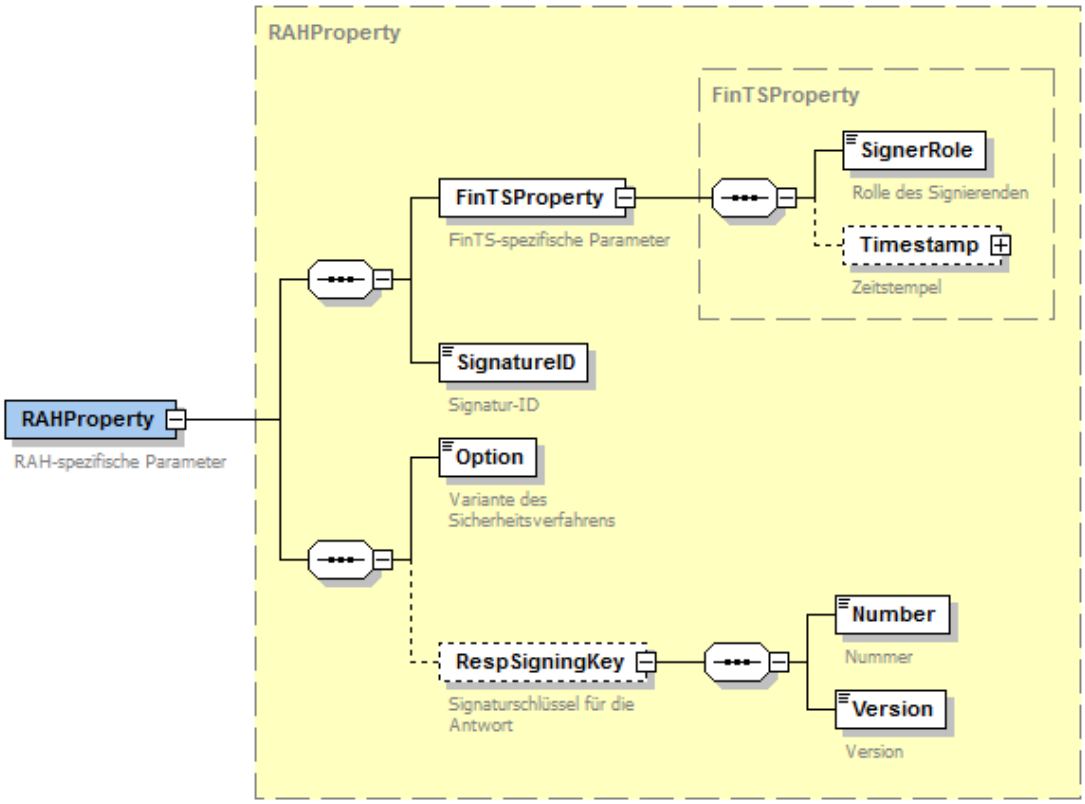


Abbildung 136: Element RAHProperty

Signaturschlüssel für die Antwort

In der Benutzernachricht ist dieses Element Pflicht, sofern das Kreditinstitut seine Nachrichten signiert. In der Kreditinstitutsnachricht entfällt es.

Variante des Sicherheitsverfahrens

In diesem Element stellt der Signierende die Variante des aktuell von ihm verwendeten RAH-Verfahrens (siehe [HBCI], Abschnitt *II.1.1 Sicherheitsprofile*) ein. Ein Benutzer oder Kreditinstitut hat alle Signaturen oder Verschlüsselungen in einem Dokument im selben Sicherheitsprofil vorzunehmen.

IV.2.2 Secoder-Verfahren

Für die Signatur einer Benutzernachricht im Verfahren [HBCI] in Verbindung mit einem Secoder wird das Element *Secoder* verwendet. Zur Verwendung und Belegung der Signatur siehe auch [HBCI], Abschnitt *III Secoder-Integration*.

Die Bildung der Signaturen in Verbindung mit einem Secoder entspricht exakt den in Abschnitt IV.2.1 beschriebenen RAH-Signaturen.

Die Eigenschaften des Secoder-Sicherheitsverfahrens werden in den Eigenschaften *SecoderProperty* beschrieben:

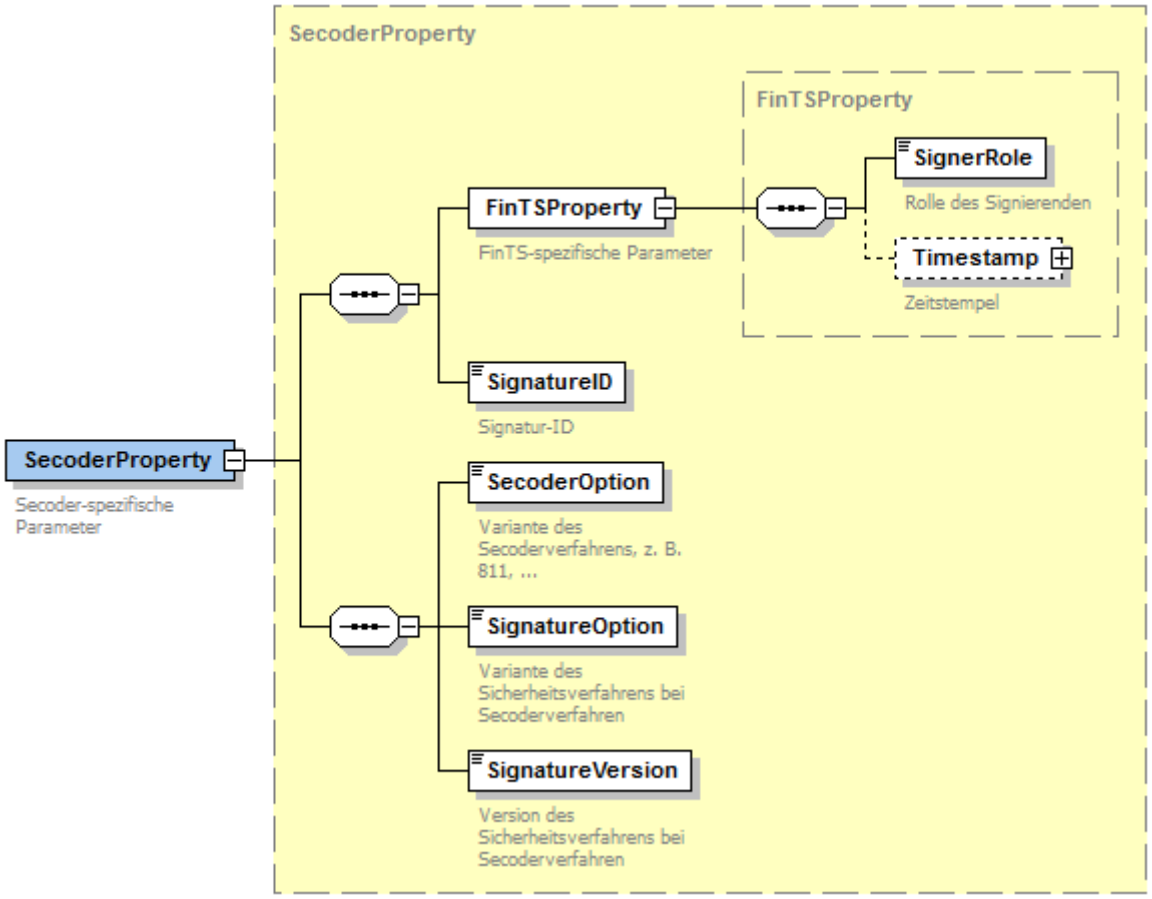


Abbildung 137: Element *SecoderProperty*

SecoderOption

In [HBCI] werden im *Abschnitt III Secoderintegration* die einzelnen Secoderverfahren beschrieben. Derzeit ist ausschließlich das Verfahren 811 für fortgeschrittene Signaturen mit RAH-Verfahren möglich.

IV.2.3 PIN/TAN-Verfahren

Für die Signatur einer Benutzernachricht im Verfahren PIN/TAN wird das Element *OneTimePassword* verwendet. Zur Verwendung und Belegung der Signatur siehe auch [PINTAN], Abschnitt *II.5.1 PIN/TAN-Signatur*.

Als allgemeiner Fall eines Challenge-Response-Verfahrens wie chipTAN und mobileTAN ist in [PINTAN], Abschnitt *II.2ff* das Zwei-Schritt-TAN-Verfahren beschrieben. Die Elemente zur Anforderung und Bereitstellung der Challenge-Informationen werden ebenfalls in der *PIN/TAN-Signatur* transportiert.

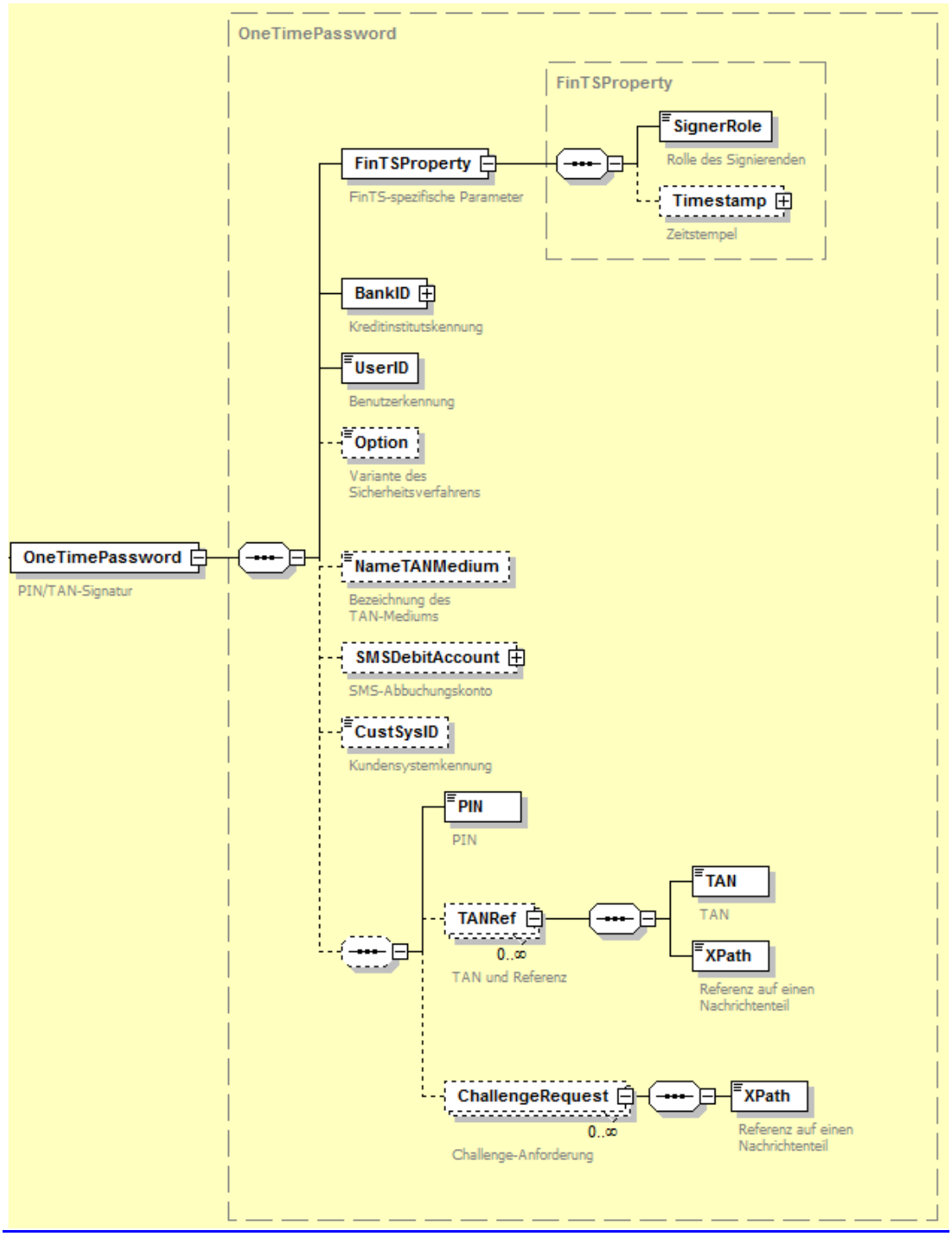


Abbildung 138: Element OneTimePassword

Referenz auf die mit PIN und TAN signierten Nachrichtenteile

In den Abschnitten *IV.3 Botensignatur* und *IV.4 Auftragssignatur* wird festgelegt und anhand von Beispielen gezeigt, wie die XPath-Referenzen in Boten- und Auftragssignaturen zu verwenden sind.

Die Signatur der Kreditinstitutsnachricht wird im Element *OneTimePasswordReply* übertragen. Es gelten die Festlegungen aus [PINTAN], Abschnitt *II.5.2 Antwort auf eine PIN/TAN-Signatur*.

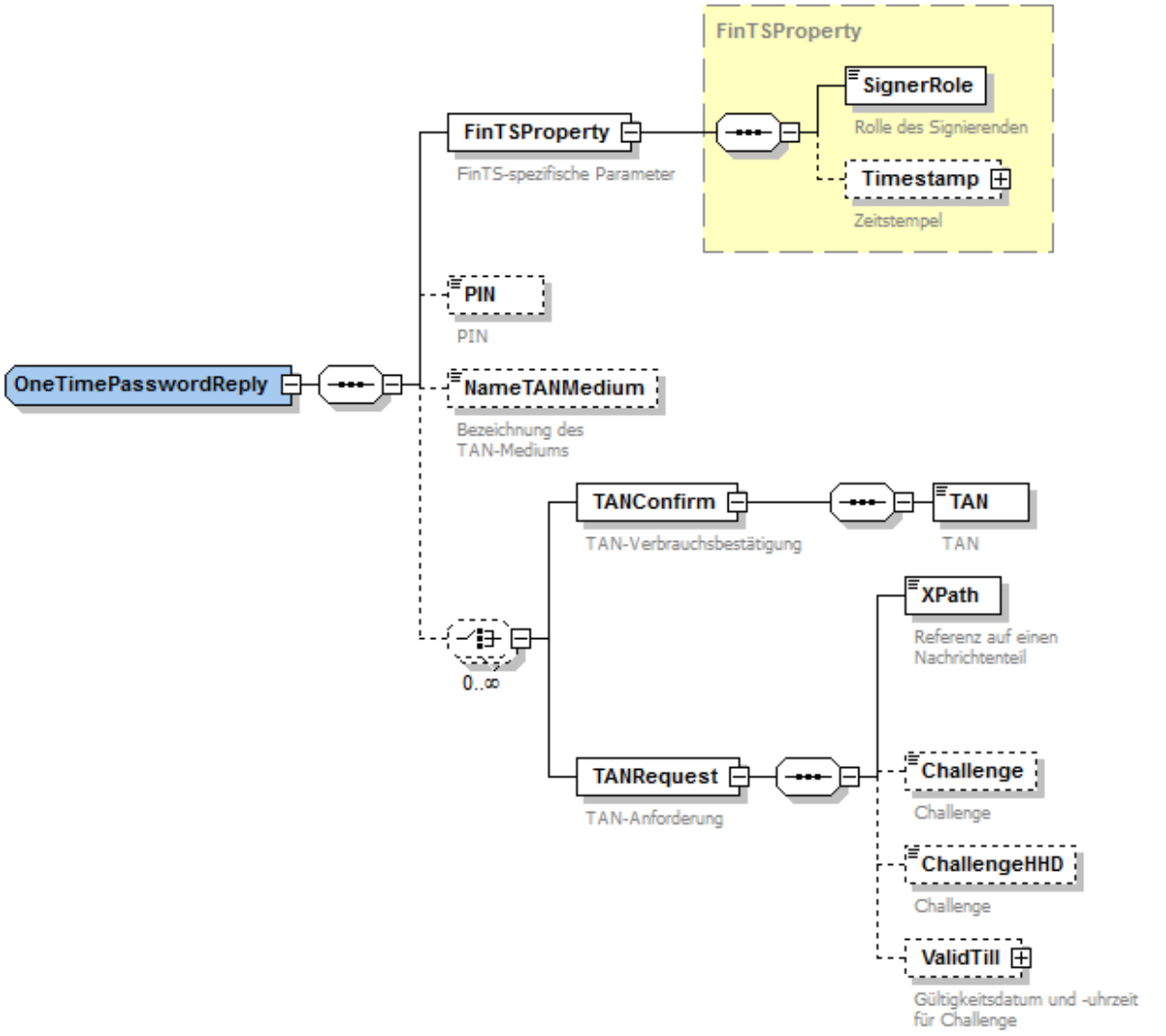


Abbildung 139: Element OneTimePasswordReply

IV.2.4 Benutzerdefinierte Signatur

Eine benutzerdefinierte Signatur besteht aus einem festen Rahmen *UserDefinedSignature*, welcher ein beliebiges (benutzerdefiniertes) Inhaltsmodell aufweist. Mit „benutzerdefiniert“ ist hier eine bilaterale Vereinbarung zwischen einem Kreditinstitut und Benutzern oder Benutzergruppen gemeint bzw. die Definition und Veröffentlichung eines Verfahrens durch ein Kreditinstitut oder einen Verband.

Zur Validierung der XML-Struktur der benutzerdefinierten Signatur kann analog zur Einbettung von Fremdformaten in eine FinTS-Nachricht (siehe *II.6.2 Integration fremder Transaktionsformate in FinTS*) ein XML-Schema angegeben werden.

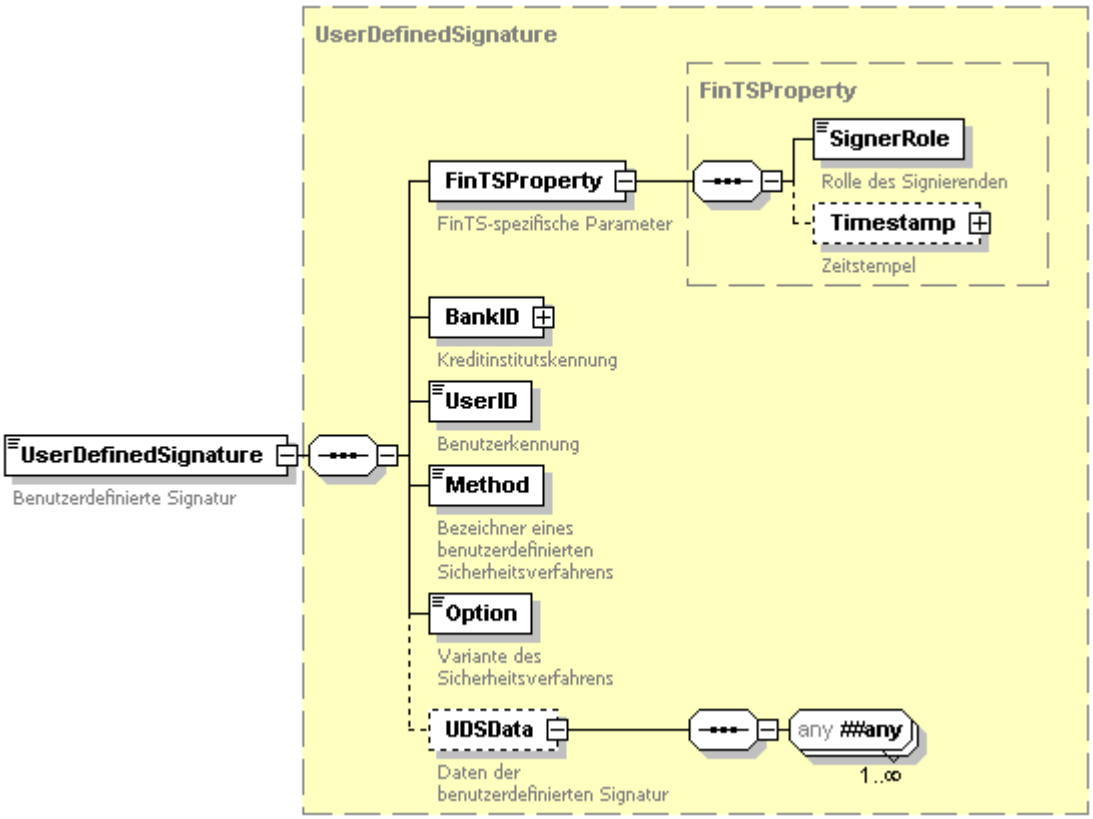


Abbildung 140: Element UserDefinedSignature

Zum festen Rahmen gehören *BankID* und *UserID*, die den signierenden Benutzer identifizieren, sowie ein *FinTSProperty*-Element mit der Rolle des Signierenden (*SignerRole*) sowie dem Signaturzeitpunkt (*Timestamp*). Weiterhin kann das verwendete Verfahren anhand der Elemente *Method* und *Option* identifiziert werden, ohne Kenntnis des benutzerdefinierten Inhaltsmodells besitzen zu müssen. Im Element *UDSData* können beliebige, bilateral vereinbarte Daten übertragen werden.

Bei diesem Signaturverfahren wird sowohl für Signaturen der Benutzernachricht als auch für Signaturen der Kreditinstitutsnachricht das Element *UserDefinedSignature* verwendet.

Beispiel:

```
<fintsmsg:ReqMsg xmlns=http://www.fints.org/spec/xmlschema/4.1/types
    xmlns:fintsmsg="http://www.fints.org/spec/xmlschema/4.1/messages"
    xsi:schemaLocation="http://www.fints.org/spec/xmlschema/4.1/messages
      http://www.fints.org/spec/xmlschema/4.1/messages/message.xsd
      http://www.fints.org/spec/xmlschema/4.1/types
      http://www.fints.org/spec/xmlschema/4.1/types/types.xsd">
    ...
    <fintsmsg:UserDefinedSignature>
        <fintsmsg:FinTSProperty>
            <fintsmsg:SignerRole>ISS</fintsmsg:SignerRole>
            <fintsmsg:Timestamp>
                <Date>2013-08-28</Date>
                <Time>14:00:08</Time>
            </fintsmsg:Timestamp>
        </fintsmsg:FinTSProperty>
        <fintsmsg:BankID>
            <CountryCode>280</CountryCode>
            <BankCode>99950001</BankCode>
        </fintsmsg:BankID>
        <fintsmsg:UserID>UserID</fintsmsg:UserID>
        <fintsmsg:Method>JUnitSignature</fintsmsg:Method>
        <fintsmsg:Option>123</fintsmsg:Option>
        <fintsmsg:UDSData>
            <MySignature xmlns="mySig-Namensraum-URI" xsi:schemaLocation
="mySig-Namensraum-URI mySig.xsd">
            </MySignature>
        </fintsmsg:UDSData>
    </fintsmsg:UserDefinedSignature>
    ...
</fintsmsg:ReqMsg>
```

Bei benutzerdefinierten Signaturen wird die Festlegung eines Schemas sowie eines Namensraumes für das Schema empfohlen. Im Beispiel wird bei der benutzerdefinierten Signatur das Schema *mySig.xsd* zur Validierung angegeben.

IV.3 Botensignatur

Ein Kreditinstitut kann alle acht Nachrichtentypen mit einer Botensignatur versehen. Ein Benutzer signiert dagegen lediglich die Nachrichtentypen *StandardReq*, *SynchronisationReq*, *UserKeyTransmissionReq*, *KeyChangeReq* und *KeyBlockReq*.

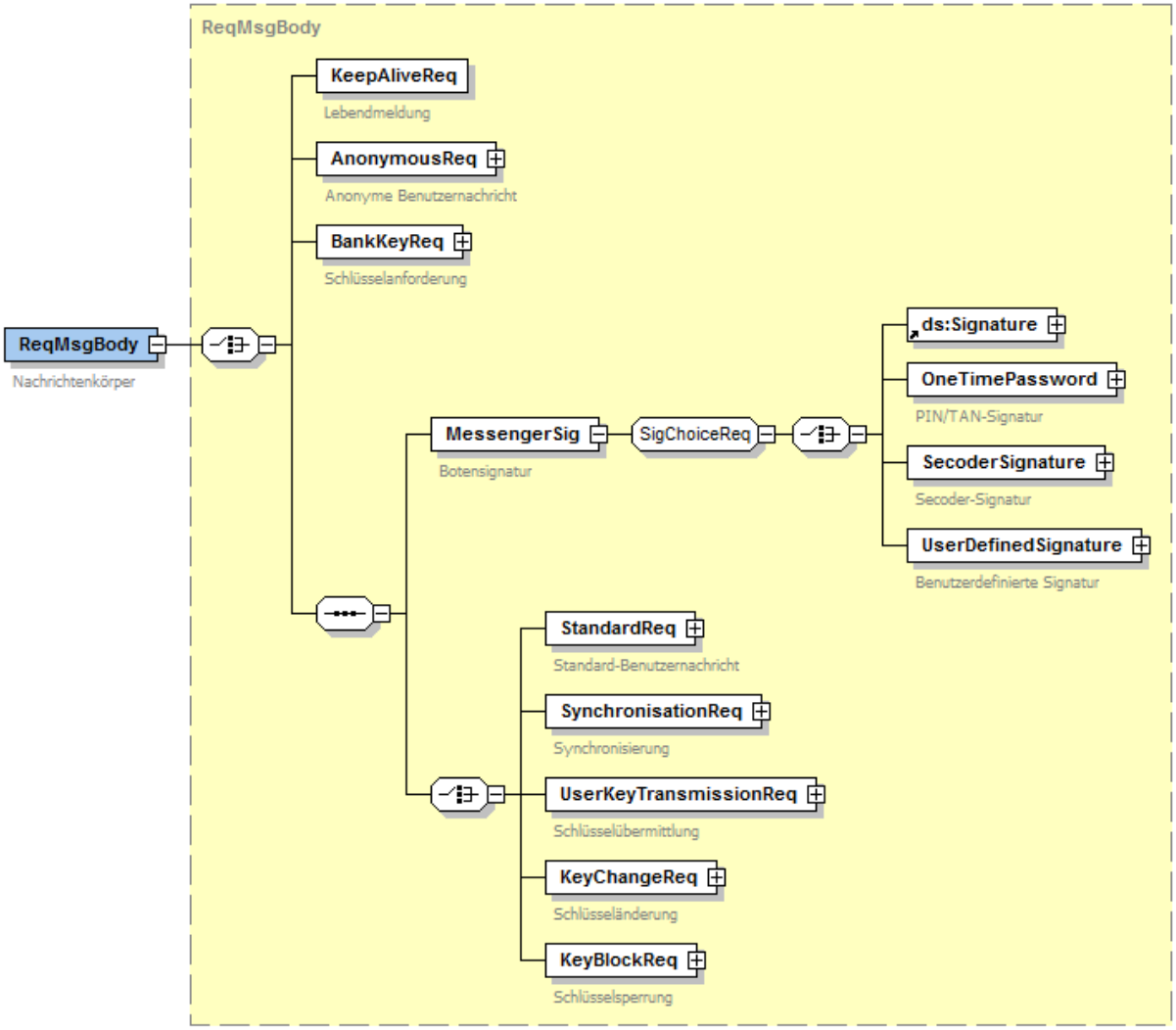


Abbildung 141: Botensignatur im Nachrichtenkörper einer Benutzernachricht

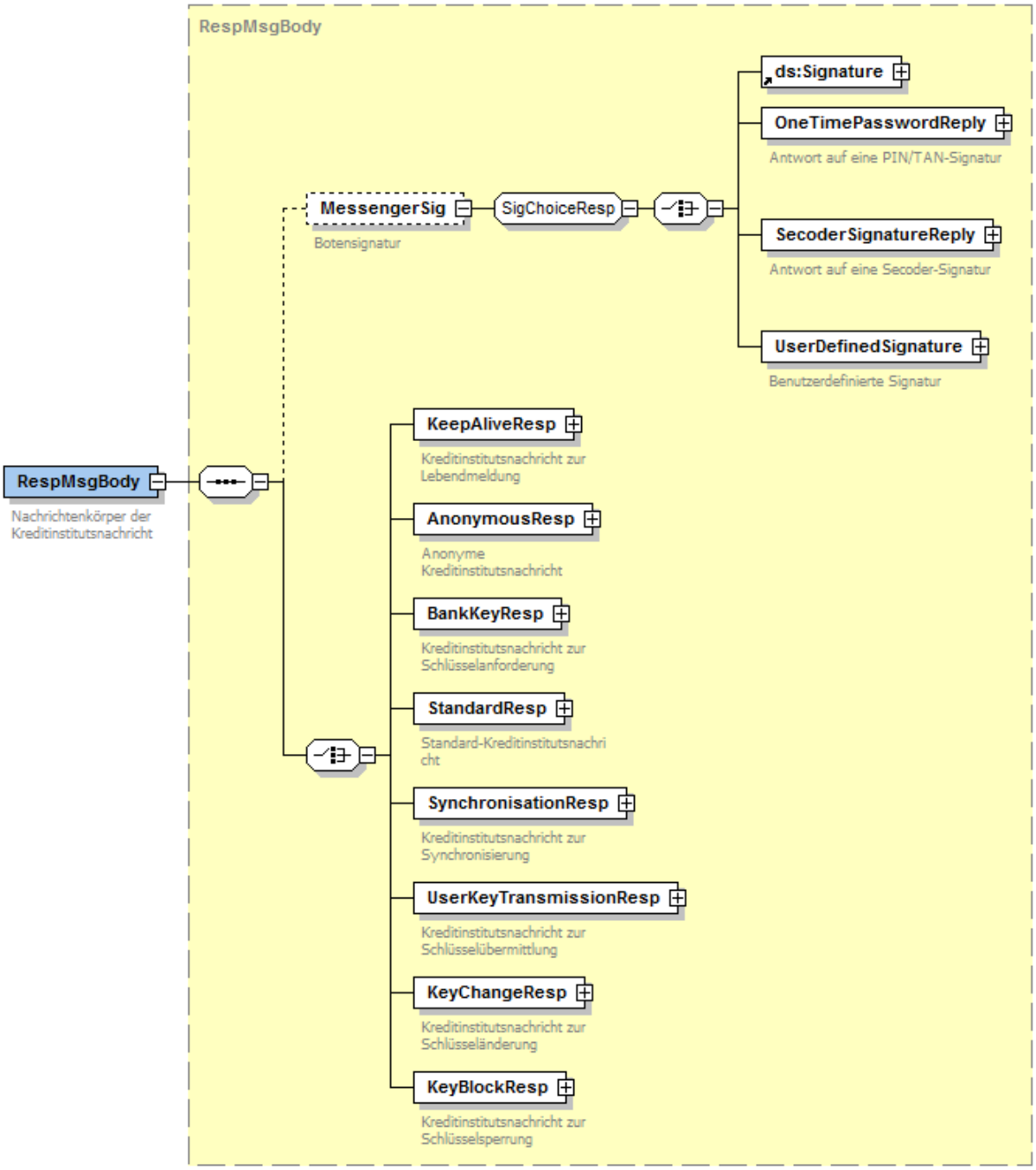


Abbildung 142: Botensignatur im Nachrichtenkörper der Kreditinstitutsnachricht

IV.3.1 W3C-konforme XML-Signatur

Die Referenzen werden bei einer Botensignatur wie folgt verwendet:

Referenz-Ziele	Die folgenden Ziele sind verpflichtend anzugeben: ▮ Nachrichtenkopf (*ReqMsgHeader* in Benutzernachrichten oder *RespMsgHeader* in Kreditinstitutsnachrichten) ▮ Inhalt des Nachrichtenkörpers (je nach Nachrichtentyp: *StandardReq*, *StandardResp*, *SyncReq* etc.) ▮ eigenes *ds:Object*-Element
Gültigkeitsbereich	Wurzelelement der Nachricht ▮ *ReqMsg* in Benutzernachrichten oder ▮ *RespMsg* in Kreditinstitutsnachrichten

Zulässiger Ausdruck	Der XPath-Ausdruck für die [XPath Filter]-Transformation muss der Bildungsregel für *LocationPath* genügen:
	
	`LocationPath ::= SinglePath ('|' SinglePath)*`
	`SinglePath ::= AbsoluteLocationPath`
	`::= | HereLocationPath`

Das folgende Beispiel zeigt eine W3C-konforme XML-Signatur als Botensignatur und die Syntax der XPath-Referenzen für eine FinTS- Benutzernachricht:

```
<fintsmsg:MessengerSig>
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="28099950001
     Ben-900312-122874321105290567 8156">
        <ds:SignedInfo>
            <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/
             xml-exc-c14n#">
                <exc:InclusiveNamespaces xmlns:exc="http://www.w3.org/2001/10/
                 xml-exc-c14n#" PrefixList="ds fintstrans
fintstype"></exc:InclusiveNamespaces>
            </ds:CanonicalizationMethod>
            <ds:SignatureMethod Algorithm="http://www.fints.org/spec/xmlschema/4.1/
             xmldsig#rsa-rsassapss-sha256x2"></ds:SignatureMethod>
          <ds:Reference URI="">
                <ds:Transforms>
                    <ds:Transform Algorithm="http://www.w3.org/2002/06/
                     xmldsig-filter2">
                        <xpf:XPath xmlns:xpf=http://www.w3.org/2002/06/
                         xmldsig-filter2
Filter="intersect">/fintsmsg:ReqMsg/fintsmsg:ReqMsgHeader |
        /fintsmsg:ReqMsg/fintsmsg:ReqMsgBody/fintsmsg:StandardReq |
        here()/../../../../../ds:Object</xpf:XPath>
                    </ds:Transform>
                    <ds:Transform Algorithm="http://www.w3.org/2001/10/
                     xml-exc-c14n#">
                        <exc:InclusiveNamespacesxmlns:exc="http://www.w3.org/2001/10/
                         xml-exc-c14n#"PrefixList="ds fintstrans
fintstype"></exc:InclusiveNamespaces>
                    </ds:Transform>
                </ds:Transforms>
                <ds:DigestMethod
                  algorithm="http://www.w3.org/2001/04/xmlenc#sha256">
                </ds:DigestMethod>
                <ds:DigestValue>DEADBEEF...KIrUsd</ds:DigestValue>
            </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>DEADBEEF...oGrUieN</ds:SignatureValue>
        <ds:KeyInfo>
            <fintsmsg:RAHKeyInfo>
                <UserKeyName mlns="http://www.fints.org/spec/xmlschema/4.1/messages">
                    <BankID>
                        <fintstype:CountryCode>280</fintstype:CountryCode>
                        <fintstype:BankCode>99950001</fintstype:BankCode>
                    </BankID>
                    <UserID>UID926518<</UserID>
                    <Number>1</Number>
                    <Version>1</Version>
                    <Type>S</Type>
                    <CustSysID>1</CustSysID>
                </UserKeyName>
            </fintsmsg:RAHKeyInfo>
        </ds:KeyInfo>
```

```
        <ds:Object>
            <ds:SignatureProperties>
                <ds:SignatureProperty Target="#28099950001Ben-900312-
                    228743211052905678156">
                    <RAHProperty
                        xmlns="http://www.fints.org/spec/xmlschema/4.1/messages">
                        <FinTSProperty>
                            <SignerRole>ISS</SignerRole>
                            <Timestamp>
                                <fintstype:Date>2013-12-17</fintstype:Date>
                                <fintstype:Time>11:13:00</fintstype:Time>
                            </Timestamp>
                        </FinTSProperty>
                        <SignatureID>1052905678156<</SignatureID>
                        <Option>7</Option>
                        <RespSigningKey>
                            <Number>1</Number>
                            <Version>1</Version>
                        </RespSigningKey>
                    </RAHProperty>
                </ds:SignatureProperty>
            </ds:SignatureProperties>
        </ds:Object>
    </ds:Signature>
</fintsmsg:MessengerSig>
```

Das Beispiel zeigt die Referenzierung in einer Botensignatur anhand des in ihr enthaltenen *Reference*-Elements. Darin werden die drei referenzierten Nachrichtenteile durch die Vereinigungsoperation | zusammengefasst. Der erste XPath-Ausdruck referenziert das Element *ReqMsgHeader* der Benutzernachricht, in der die Botensignatur enthalten ist. Die resultierende Knotenmenge wird durch die beiden folgenden Lokalisierungspfade mit den Mengen der Knoten unter *StandardReq* und *ds:Object* vereinigt.

IV.3.2 PIN/TAN-Verfahren

In der Botensignatur einer Benutzernachricht kann optional eine TAN zusammen mit einer Referenz angegeben werden. Wird dieses Element *TANRef* angegeben, gelten die folgenden Festlegungen für das Element *XPath*:

Referenz-Ziel	Wurzelelement des Nachrichtenkörpers (*ReqMsgBody*)
Gültigkeitsbereich	Wurzelelement der Benutzernachricht(*ReqMsg*)
Kontext	Der Kontext zur Auswertung des Lokalisierungspfades ist das Element *XPath*, in dem sich der Ausdruck befindet.

Zulässiger Ausdruck	Der Ausdruck im Element *XPath* muss der Bildungsregel für *LocationPath* genügen:
	
	`LocationPath ::= AbsoluteLocationPath`
	`::= | RelativeLocationPath`

Das folgende Beispiel zeigt eine Botensignatur nach dem PIN/TAN-Verfahren in einer Benutzernachricht:

```
<fintsmsg:MessengerSig xmlns="http://www.fints.org/spec/xmlschema/4.1/types"
  xmlns:fintsmsg="http://www.fints.org/spec/xmlschema/4.1/messages">
    <fintsmsg:OneTimePassword>
        <fintsmsg:FinTSProperty>
            <fintsmsg:SignerRole>ISS</fintsmsg:SignerRole>
            <fintsmsg:Timestamp>
                <Date>2013-08-28</Date>
                <Time>14:03:29</Time>
            </fintsmsg:Timestamp>
        </fintsmsg:FinTSProperty>
        <fintsmsg:BankID>
            <CountryCode>280</CountryCode>
            <BankCode>99950001</BankCode>
        </fintsmsg:BankID>
        <fintsmsg:UserID>Benutzer-0000000012-01</fintsmsg:UserID>
        <fintsmsg:Option>1.1</fintsmsg:Option>
        <fintsmsg:CustSysID>meinPC</fintsmsg:CustSysID>
        <fintsmsg:PIN>12345</fintsmsg:PIN>
    </fintsmsg:OneTimePassword>
</fintsmsg:MessengerSig>
```

Es wird hier implizit die gesamte Nachricht signiert, die Angabe einer TAN sowie eines Referenz-Ziels entfällt.

IV.3.3 Benutzerdefinierte Signatur

Eine benutzerdefinierte Signatur enthält keine im Rahmen des FinTS-Standards spezifizierten Referenzierungen. Die Belegung ist somit identisch bei der Verwendung als Boten- oder Auftragssignatur.

Eine benutzerdefinierte Botensignatur signiert implizit den Nachrichtenkopf und den Inhalt des Nachrichtenkörpers.

IV.4 Auftragssignatur

Mit einer Auftragssignatur werden einzelne Geschäftsvorfälle in einer *RequestList* oder Antworten zu Geschäftsvorfällen in einer *ResponseList* signiert. Eine *RequestList* oder *ResponseList* kann mehrere Auftragssignaturen enthalten, jede Signatur kann einen oder mehrere Geschäftsvorfälle signieren. Die beiden folgenden Abbildungen zeigen die Positionen der Auftragssignaturen in einer Standard-Benutzernachricht und der Standard-Kreditinstitutsnachricht:

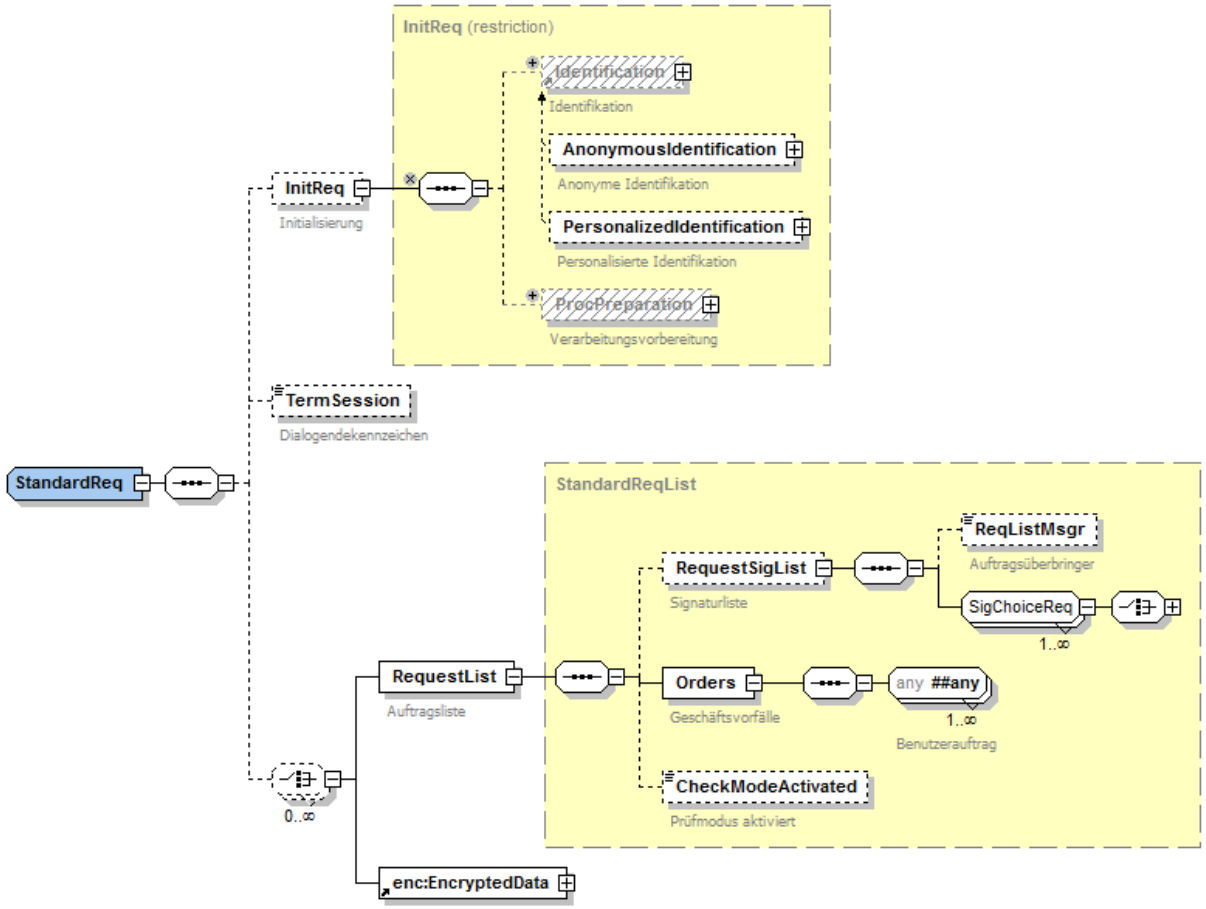


Abbildung 143: Auftragssignatur in einer Benutzernachricht

Auftragsüberbringer

Im Element *ReqListMsgr* wird eine der Signaturen als diejenige des Benutzers gekennzeichnet, für den in der Kreditinstitutsnachricht die entsprechende Liste der Antworten signiert und ggf. verschlüsselt werden soll. Für den hier einzustellenden XPath gelten folgende Regeln:

Referenz-Ziele	Genau einer der Geschwisterknoten ▮ Signature, ▮ OneTimePassword, ▮ Secoder oder ▮ UserDefinedSignature des Elements *ReqListMsgr*.
Gültigkeitsbereich	Das Elternelement *RequestSigList* von *ReqListMsgr*.
Kontext	Der Kontext zur Auswertung des Lokalisierungspfades ist das Element *ReqListMsgr*, in dem sich der Ausdruck befindet.
Zulässiger Ausdruck	Der Ausdruck in *ReqListMsgr* muss der Bildungsregel für *RelativeLocationPath* genügen.

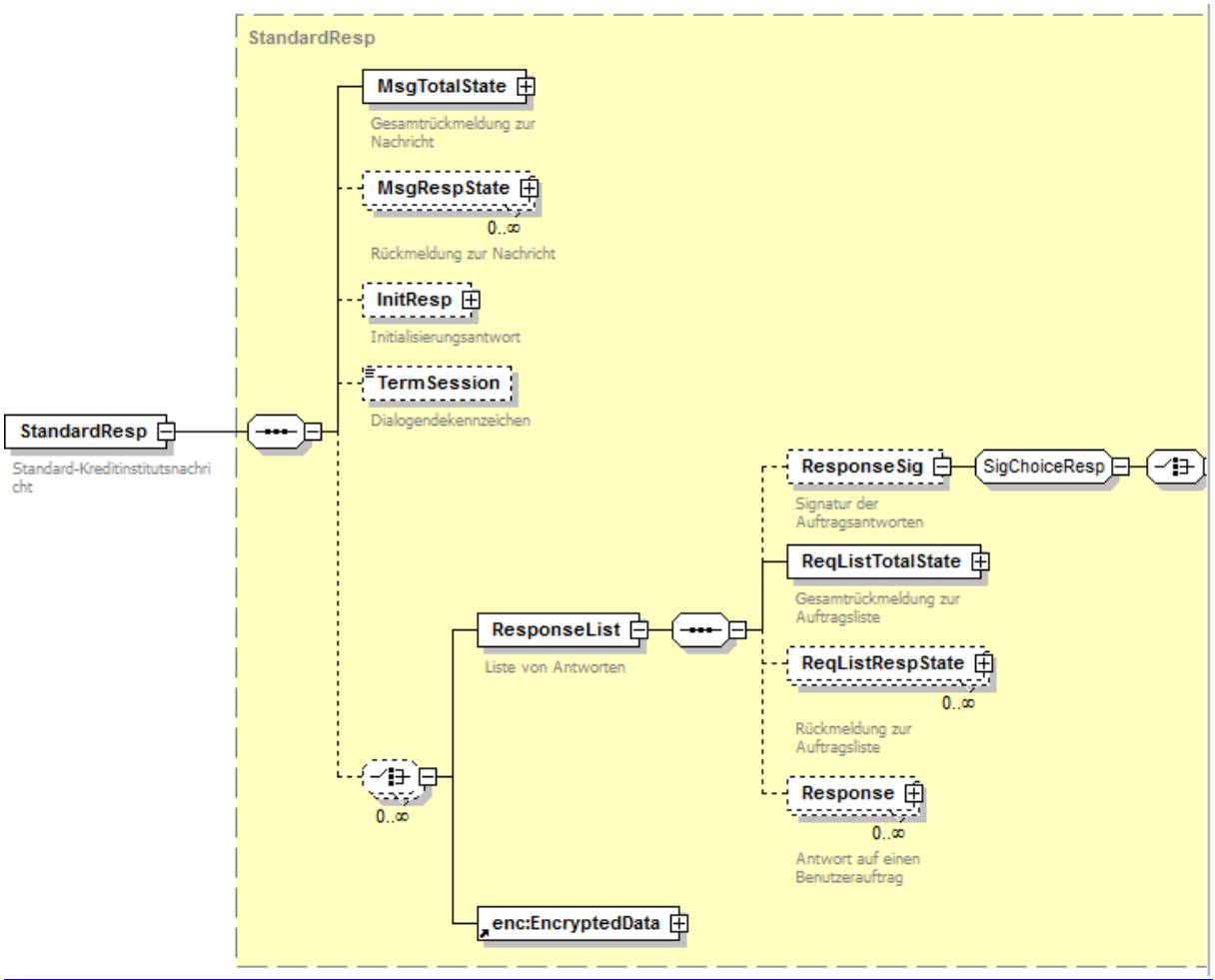


Abbildung 144: Auftragssignatur in einer Kreditinstitutsnachricht

IV.4.1 W3C-konforme XML-Signatur

Die Referenzen werden bei einer Auftragssignatur wie folgt verwendet:

Referenz-Ziele	Es ist immer das eigene *ds:Object*-Element zu signieren In einer Benutzernachricht können darüber hinaus beliebige Benutzeraufträge, also beliebige Kindelemente des Elements *Orders* oder der entsprechenden Elemente der administrativen Nachrichten unterhalb der *RequestList* angegeben werden, in der sich die Signatur befindet. In einer Kreditinstitutsnachricht werden zwingend alle Rückmeldungen zu den Auftragslisten sowie alle Antworten auf die Benutzeraufträge, also alle Kindelemente *ReqListTotalState*, *ReqListRespState* oder *Response* der *ResponseList* angegeben, in der sich die Signatur befindet.
Gültigkeitsbereich	Wurzelelement der Liste von Aufträgen bzw. Antworten in der sich die Signatur befindet: ▮ *RequestList* in Benutzernachrichten oder ▮ *ResponseList* in Kreditinstitutsnachrichten
Zulässiger Ausdruck	Der Ausdruck für die [XPath Filter]-Transformation muss der Bildungsregel für *LocationPath* genügen: `LocationPath ::= HereLocationPath ('|' HereLocationPath)*`

Das folgende Beispiel zeigt die Signatur eines Geschäftsvorfalls (*AcctBal*):

```
<fintsmsg:RequestList xmlns="http://www.fints.org/spec/xmlschema/4.1/types"
xmlns:fintsmsg="http://www.fints.org/spec/xmlschema/4.1/messages"
xmlns:fintstrans="http://www.fints.org/spec/xmlschema/4.1/transactions"
    xmlns:fintstype="http://www.fints.org/spec/xmlschema/4.1/types">
    <fintsmsg:RequestSigList>
        <fintsmsg:ReqListMsgr
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">../ds:Signature</fintsmsg:ReqListMsgr>
        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="travic2">
            <ds:SignedInfo>
                <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#">
                    <exc:InclusiveNamespaces
xmlns:exc="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="ds fintstrans
fintstype"></exc:InclusiveNamespaces>
                </ds:CanonicalizationMethod>
                <ds:SignatureMethod
Algorithm="http://www.fints.org/spec/xmlschema/4.1/xmldsig#rsa-rsassapss-
sha256x2"></ds:SignatureMethod>
                <ds:Reference URI="">
                    <ds:Transforms>
                        <ds:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-
filter2">
                            <xpf:XPath xmlns:xpf="http://www.w3.org/2002/06/xmldsig-
filter2"
Filter="intersect">here()/../../../../../../../fintsmsg:Orders/fintstrans:AcctMvmtsSp
ecifiedPeriod_1_Req | here()/../../../../../../ds:Object</xpf:XPath>
                        </ds:Transform>
                        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
                            <exc:InclusiveNamespaces
xmlns:exc="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="ds fintstrans
fintstype"></exc:InclusiveNamespaces>
                        </ds:Transform>
```

```
                    </ds:Transforms>
                    <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"></ds:DigestMethod>
                    <ds:DigestValue>DIGEST...VALUE=</ds:DigestValue>
                </ds:Reference>
            </ds:SignedInfo>
            <ds:SignatureValue>SIG...VALUE</ds:SignatureValue>
            <ds:KeyInfo>
                <fintsmsg:RAHKeyInfo>
                    <UserKeyName
xmlns="http://www.fints.org/spec/xmlschema/4.1/messages">
                        <BankID>
                            <fintstype:CountryCode>280</fintstype:CountryCode>
                            <fintstype:BankCode>99950001</fintstype:BankCode>
                        </BankID>
                        <UserID>UID926518</UserID>
                        <Number>1</Number>
                        <Version>1</Version>
                        <Type>S</Type>
                        <CustSysID>1</CustSysID>
                    </UserKeyName>
                </fintsmsg:RAHKeyInfo>
            </ds:KeyInfo>
            <ds:Object>
                <ds:SignatureProperties>
                    <ds:SignatureProperty Target="#28099950001Ben-900312-
122874321105270593818 6">
                        <RAHProperty
xmlns="http://www.fints.org/spec/xmlschema/4.1/messages">
                            <FinTSProperty>
                                <SignerRole>ISS</SignerRole>
                                <Timestamp>
                                    <fintstype:Date>2013-08-28</fintstype:Date>
                                    <fintstype:Time>14:14:27</fintstype:Time>
                                </Timestamp>
                            </FinTSProperty>
                            <SignatureID>2</SignatureID>
                            <Option>7</Option>
                            <RespSigningKey>
                                <Number>1</Number>
                                <Version>1</Version>
                            </RespSigningKey>
                        </RAHProperty>
                    </ds:SignatureProperty>
                </ds:SignatureProperties>
            </ds:Object>
        </ds:Signature>
    </fintsmsg:RequestSigList>
    <fintsmsg:Orders>
        <fintstrans:AcctMvmtsSpecifiedPeriod_1_Req>
            ...
        </fintstrans:AcctMvmtsSpecifiedPeriod_1_Req>
    </fintsmsg:Orders>
</fintsmsg:RequestList>
```

IV.4.2 PIN/TAN-Verfahren

In der Auftragssignatur einer Benutzernachricht werden diejenigen Aufträge referenziert, welche mit einer *TAN* signiert werden. Für die Referenzen gelten die folgenden Festlegungen:

Referenz-Ziele	In einer Benutzernachricht verweist jeder *XPath* auf einen beliebigen Benutzerauftrag, also auf ein beliebiges Kindelement des Elements *Orders* oder der entsprechenden Elemente der administrativen Nachrichten unterhalb der *RequestList*, in der sich die Signatur befindet.
Gültigkeitsbereich	Wurzelelement der Liste von Aufträgen (*RequestList*) in der sich die Signatur befindet:
Kontext	Der Kontext zur Auswertung des Lokalisierungspfades ist das Element *XPath* in dem sich der Ausdruck befindet.
Zulässiger Ausdruck	Der Ausdruck in *XPath* muss der Bildungsregel für *RelativeLocationPath* genügen.

Das Beispiel zeigt die Signatur eines Geschäftsvorfalls (*AcctBal*):

```
<RequestList xmlns="http://www.fints.org/spec/xmlschema/4.1/types"
 xmlns:fintstype="http://www.fints.org/spec/xmlschema/4.1/types">
  <RequestSigList>
    <ReqListMsgr>../fintstype:OneTimePassword[1]</ReqListMsgr>
    <OneTimePassword>
      <FinTSProperty>
        <SignerRole>ISS</SignerRole>
        <Timestamp>
          <Date>2003-10-05</Date>
          <Time>15:13:45</Time>
        </Timestamp>
      </FinTSProperty>
      <BankID>
        <CountryCode>280</CountryCode>
        <BankCode>99950001</BankCode>
      </BankID>
      <UserID>UID926518</UserID>
      <CustSysID>KSIdent</CustSysID>
      <PIN>123456</PIN>
      <TANRef>
        <TAN>123456</TAN>
        <XPath>../../../../fintstype:Orders[1]/fintstrans:AcctBal 1 Req[1]
</XPath>
      </TANRef>
    </OneTimePassword>
  </RequestSigList>
  <Orders>
    <fintstrans:AcctBal_1_Req
     xmlns:fintstrans="http://www.fints.org/spec/xmlschema/4.1/transactions"
      ...
    </fintstrans:AcctBal 1 Req>
  </Orders>
</RequestList>
```

IV.4.3 Secoder-Signatur

Die Referenzen werden bei einer Secodersignatur wie folgt verwendet:

Referenz-Ziele	Es ist immer das eigene *ds:Object*-Element zu signieren In einer Benutzernachricht können darüber hinaus beliebige Benutzeraufträge, also beliebige Kindelemente des Elements *Orders* oder der entsprechenden Elemente der administrativen Nachrichten unterhalb der *RequestList* angegeben werden, in der sich die Signatur befindet. In einer Kreditinstitutsnachricht werden zwingend alle …gen zu den Auftragslisten sowie alle Antworten auf …raufträge, also alle Kindelemente …*lState*, *ReqListRespState* oder *Response* der …*st* angegeben, in der sich die Signatur befindet.
Gültigkeitsbereich	…ent der Liste von Aufträgen bzw. Antworten in der …natur befindet: ▮ *RequestList* in Benutzernachrichten oder ▮ *ResponseList* in Kreditinstitutsnachrichten

Zulässiger Ausdruck	Der Ausdruck für die [XPath Filter]-Transformation muss der Bildungsregel für *LocationPath* genügen:
	
	`LocationPath ::= HereLocationPath`
	`('|' HereLocationPath)*`

In Überarbeitung

IV.4.4 Benutzerdefinierte Signatur

Eine benutzerdefinierte Signatur enthält keine im Rahmen des FinTS-Standards spezifizierten Referenzierungen. Die Belegung ist somit identisch bei der Verwendung als Boten- oder Auftragssignatur.

Eine benutzerdefinierte Auftragssignatur signiert in einer Benutzernachricht implizit alle Aufträge einer *RequestList* bzw. in einer Kreditinstitutsnachricht alle Rückmeldungen zur Auftragsliste sowie alle Antworten in einer *ResponseList*.

V. VERSCHLÜSSELTE UND KOMPRIMIERTE NACHRICHTENTEILE

V.1 Aufbau des Verschlüsselungssegments

FinTS-Nachrichten können auf zwei Nachrichtenebenen verschlüsselt werden. Zur Darstellung der Verschlüsselung in XML wird der [XML Encryption] Standard des W3C verwendet. Verschlüsselung ist nur im Sicherheitsverfahren [HBCI] möglich (siehe auch [HBCI], Abschnitt *II.5.2 Verschlüsselungsdaten*).

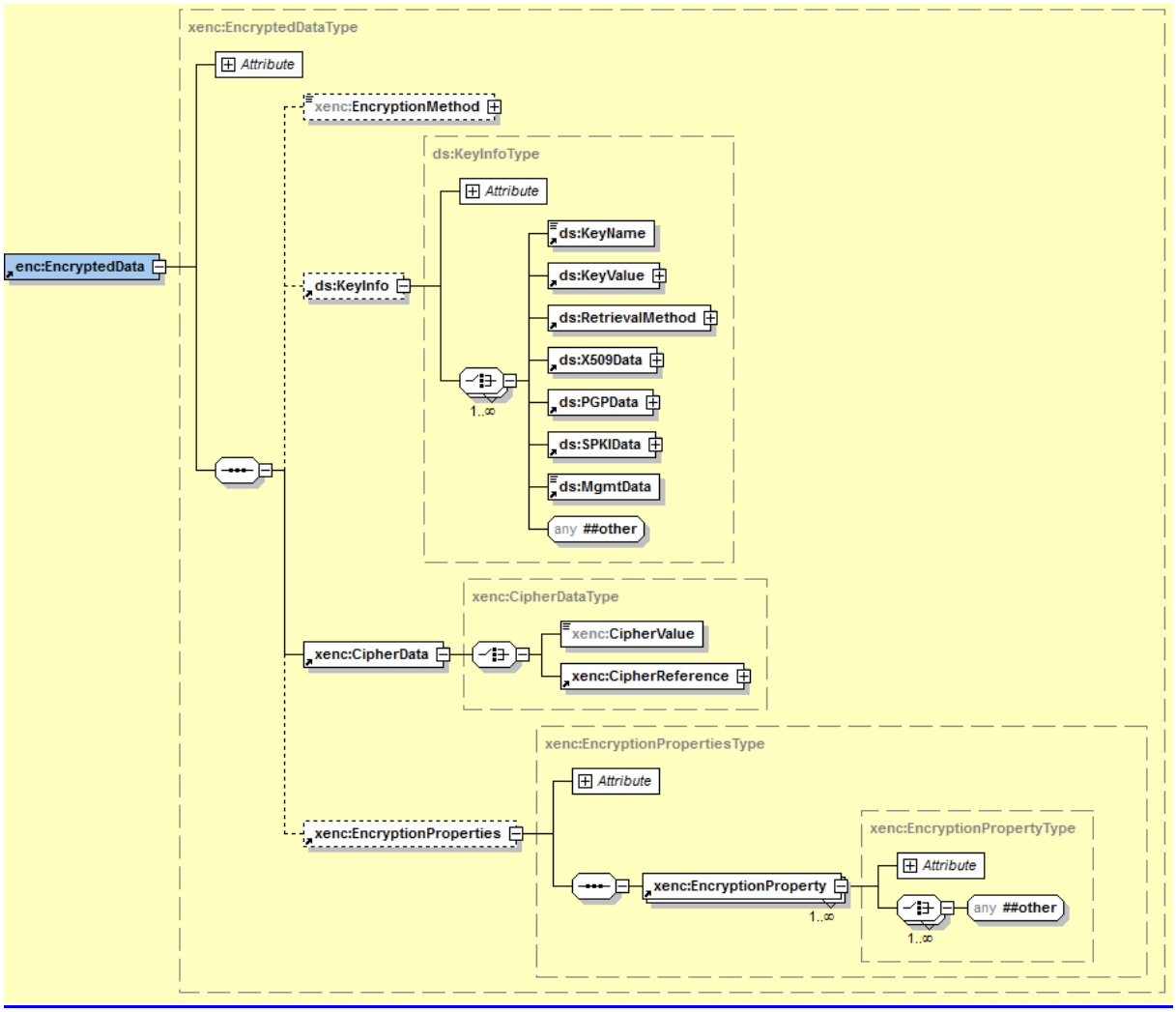


Abbildung 145: Verschlüsselung mit dem EncryptedData-Element

Die Abbildung zeigt das Element *EncryptedData*, welches die nach XML Encryption verschlüsselten Teile einer FinTS-Nachricht enthält.

- **Verfahren zur Nachrichtenverschlüsselung**

  Das Attribut *Algorithm* des Elements *EncryptionMethod* enthält die Beschreibung des Verfahrens zur Verschlüsselung der Nachrichteninhalte. Die zulässigen Verfahren zur Nachrichtenverschlüsselung sind durch die Sicherheitsprofile des Sicherheitsverfahrens [HBCI] festgelegt (siehe [HBCI], Abschnitt *II.1.1 Sicherheitsprofile*). Das Attribut *Type* des *EncryptedData*-Elements muss bei Verschlüsselung in FinTS immer mit dem Wert *http://www.w3.org/2001/04/xmlenc#Element* belegt werden.

Unterhalb des Elements *EncryptionProperty* wird das aktuelle Sicherheitsprofil übertragen. Hier ist bei Verwendung des RAH-Verfahrens das Element *RAH* mit dem entsprechenden Variantenbezeichner einzustellen. Ein Benutzer oder Kreditinstitut hat alle Signaturen oder Verschlüsselungen in einem Dokument im selben Sicherheitsprofil vorzunehmen.

- **Verschlüsselter Nachrichteninhalt**

  Der verschlüsselte Teil der Nachricht befindet sich in dem Element *CipherValue* unter dem Element *CipherData*. Der Inhalt von *CipherValue* wird durch Anwendung des Verschlüsselungsalgorithmus aus *EncryptionMethod* und des Sitzungsschlüssels (siehe folgender Absatz) auf die zu verschlüsselnden Daten berechnet. Er wird in *base64*-Codierung eingestellt. Diese Daten müssen vor der Verschlüsselung mit dem Algorithmus [Canonical XML] kanonisiert werden. Das *CipherReference*-Element und die *EncryptionProperties* werden in FinTS nicht zur Verschlüsselung benutzt.

- **Verschlüsselter Sitzungsschlüssel**

  Da es sich bei den Verschlüsselungen im Sicherheitsverfahren [HBCI] immer um zweistufige Verfahren handelt, enthält das Element *KeyInfo* den zur Nachrichtenverschlüsselung verwendeten Schlüssel (Sitzungsschlüssel) nicht direkt, sondern in verschlüsselter Form als *EncryptedKey*-Element. Dieses wiederum enthält ein Element *EncryptionMethod* mit der Beschreibung des Verfahrens zur Verschlüsselung des Sitzungsschlüssels gemäß den Sicherheitsprofilen des Sicherheitsverfahrens [HBCI], Abschnitt *II.1.1 Sicherheitsprofile*. Außerdem enthält es den verschlüsselten AES-Sitzungsschlüssel im *CipherData*-Element sowie die Daten des zu seiner Verschlüsselung verwendeten AES- bzw. RAH-Schlüssels in Form eines *RAHKeyInfo* (Schlüsselinformationen für das Sicherheitsverfahren RAH) im *KeyInfo*-Element.

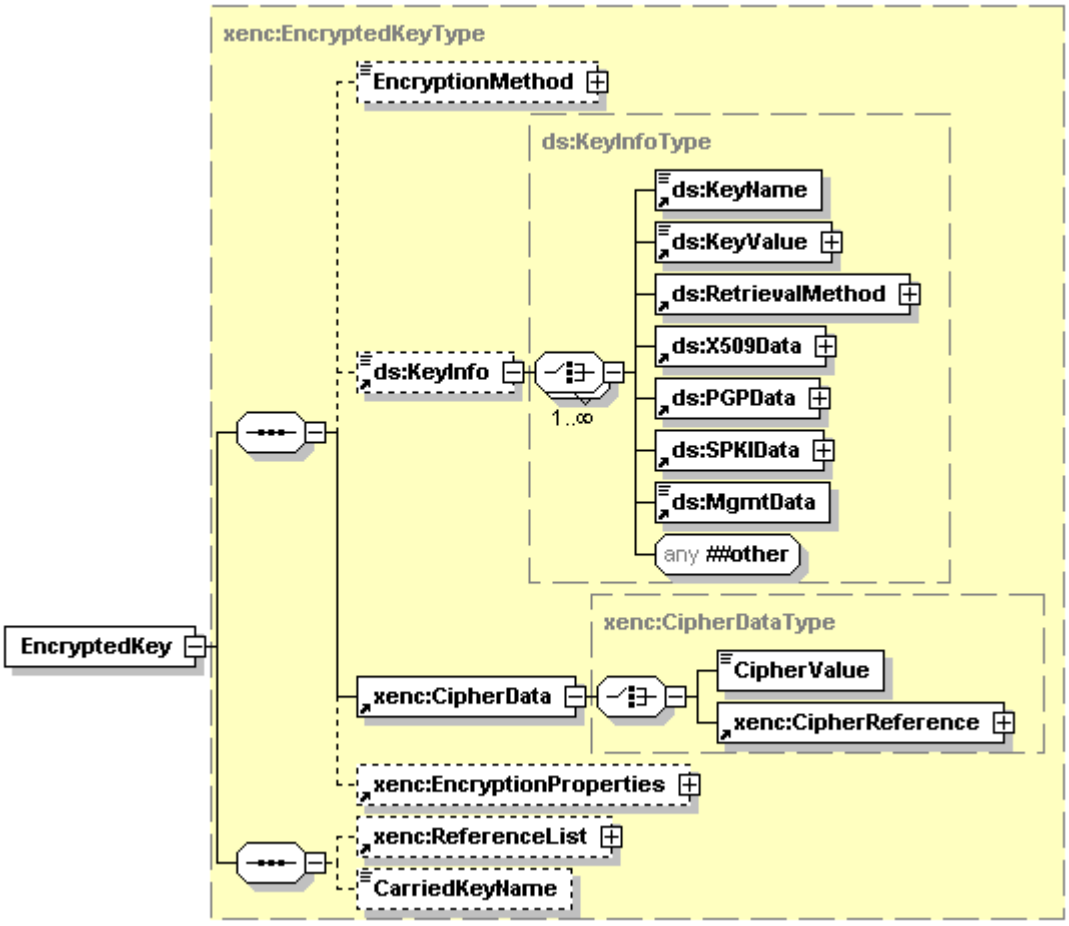


Abbildung 146: Verschlüsselter Sitzungsschlüssel

Schlüsselinformationen für das Sicherheitsverfahren RAH

In Benutzernachrichten wird der **Benutzerschlüsselname** belegt, in Kreditinstitutsnachrichten hingegen der **Kreditinstitutsschlüsselname**.

Das Beispiel zeigt den Inhalt des Elements *ReqMsg* nach der Verschlüsselung des *ReqMsgBody*-Elements. Das Element *RegMsgBody* wurde nach der Verschlüsselung durch das Element *EncryptedData* ersetzt.

```
<fintsmsg:ReqMsg xmlns="http://www.fints.org/spec/xmlschema/4.1/types"
xmlns:fintsmsg="http://www.fints.org/spec/xmlschema/4.1/messages"
xmlns:fintstrans="http://www.fints.org/spec/xmlschema/4.1/transactions"
xmlns:fintstype="http://www.fints.org/spec/xmlschema/4.1/types">
    <fintsmsg:ReqMsgHeader>
        <fintsmsg:BankID>
            <CountryCode>280</CountryCode>
            <BankCode>99950001</BankCode>
        </fintsmsg:BankID>
        <fintsmsg:UserRef>USER_REF12345</fintsmsg:UserRef>
        <fintsmsg:UserTextRef>MeineReferenz</fintsmsg:UserTextRef>
        <fintsmsg:MsgNo>1</fintsmsg:MsgNo>
    </fintsmsg:ReqMsgHeader>
    <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
Type="http://www.w3.org/2001/04/xmlenc#Element">
        <xenc:EncryptionMethod
Algorithm="http://www.fints.org/spec/xmlschema/4.1/xmlenc#aes256"></xenc:EncryptionMe
thod>
```

```
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
            <xenc:EncryptedKey>
                <xenc:EncryptionMethod
Algorithm="http://www.w3c.org/2001/04/xmlenc#rsa-1_5"></xenc:EncryptionMethod>
                <ds:KeyInfo>
                    <fintsmsg:RAHKeyInfo>
                        <BankKeyName
xmlns="http://www.fints.org/spec/xmlschema/4.1/messages">
                            <BankID>
                                <fintstype:CountryCode>280</fintstype:CountryCode>
                                <fintstype:BankCode>99950001</fintstype:BankCode>
                            </BankID>
                            <Number>1</Number>
                            <Version>1</Version>
                            <Type>C</Type>
                        </BankKeyName>
                    </fintsmsg:RAHKeyInfo>
                </ds:KeyInfo>
                <xenc:CipherData>
                    <xenc:CipherValue>CIPHER VALUE</xenc:CipherValue>
                </xenc:CipherData>
            </xenc:EncryptedKey>
        </ds:KeyInfo>
        <xenc:CipherData>
            <xenc:CipherValue>CIPHER VALUE</xenc:CipherValue>
        </xenc:CipherData>
        <xenc:EncryptionProperties>
            <xenc:EncryptionProperty>
                <fintsmsg:RAH>
                    <fintstype:Option>7</fintstype:Option>
                </fintsmsg:RAH>
            </xenc:EncryptionProperty>
        </xenc:EncryptionProperties>
    </xenc:EncryptedData>
</fintsmsg:ReqMsg>
```

V.2 Verschlüsselung des Nachrichtenkörpers

Bei der Verschlüsselung des Nachrichtenkörpers wird der gesamte Teilbaum unter dem Wurzelelement des Nachrichtenkörpers (*ReqMsgBody* bzw. *RespMsgBody*) verschlüsselt und durch das *EncryptedData*-Element ersetzt. Bei der Entschlüsselung ist entsprechend das *EncryptedData*-Element durch den entschlüsselten Teilbaum zu ersetzen.

Eine erneute Verschlüsselung des verschlüsselten Teils (*super encryption)* ist in FinTS nicht zulässig, allerdings ist die Komprimierung formal auch eine [XML Encryption], siehe *V.4 Komprimierung*.

Die beiden folgenden Abbildungen zeigen den Teil des Nachrichtenaufbaus zur Verschlüsselung der Nachrichtenkörper einer Benutzernachricht und einer Kreditinstitutsnachricht.

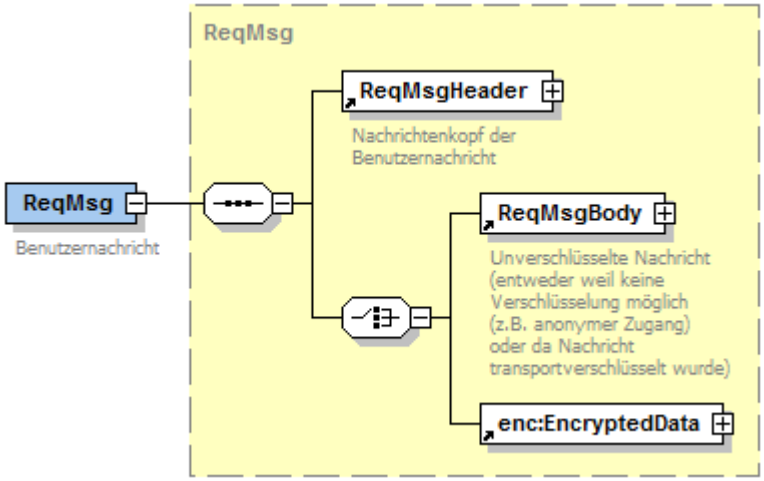


Abbildung 147: Verschlüsselung des Nachrichtenkörpers der Benutzernachricht

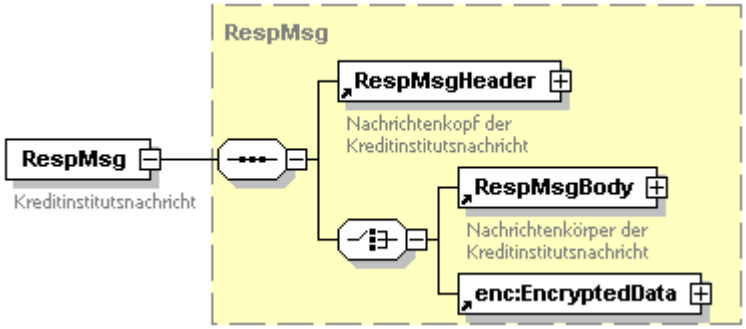


Abbildung 148: Verschlüsselung des Nachrichtenkörpers der Kreditinstitutsnachricht

V.3 Verschlüsselung von Aufträgen und Auftragsantworten

Die Verschlüsselung von Aufträgen und Auftragsantworten ist nur für bestimmte Nachrichtentypen zulässig (siehe auch *III.3 Verschiedene Benutzer- und Antwortnachrichten* und *III.4 Keymanagement-Nachrichten*). Verschlüsselt wird jeweils der gesamte Teilbaum der *RequestList* bzw. *ResponseList*, das *EncryptedData*-Element ersetzt nach der Verschlüsselung den verschlüsselten Teilbaum. Umgekehrt wird bei der Entschlüsselung *EncryptedData* wieder durch den entschlüsselten Teilbaum ersetzt.

Eine erneute Verschlüsselung des verschlüsselten Teils *super encryption* ist in FinTS nicht zulässig, allerdings ist die Komprimierung formal auch eine [XML Encryption] (siehe *V.4 Komprimierung*).

Die normale Benutzernachricht *StandardReq* bzw. ihre Antwort *StandardResp* können mehrere Auftrags- bzw. Antwortteile enthalten. Dabei können verschlüsselte und unverschlüsselte Teile beliebig gemischt werden (vgl. Abbildung 149 und Abbildung 150).

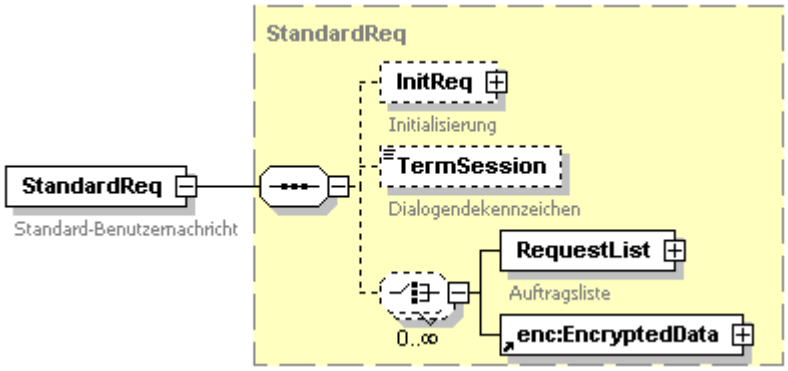


Abbildung 149: Auftragsverschlüsselung in der Standard-Benutzernachricht

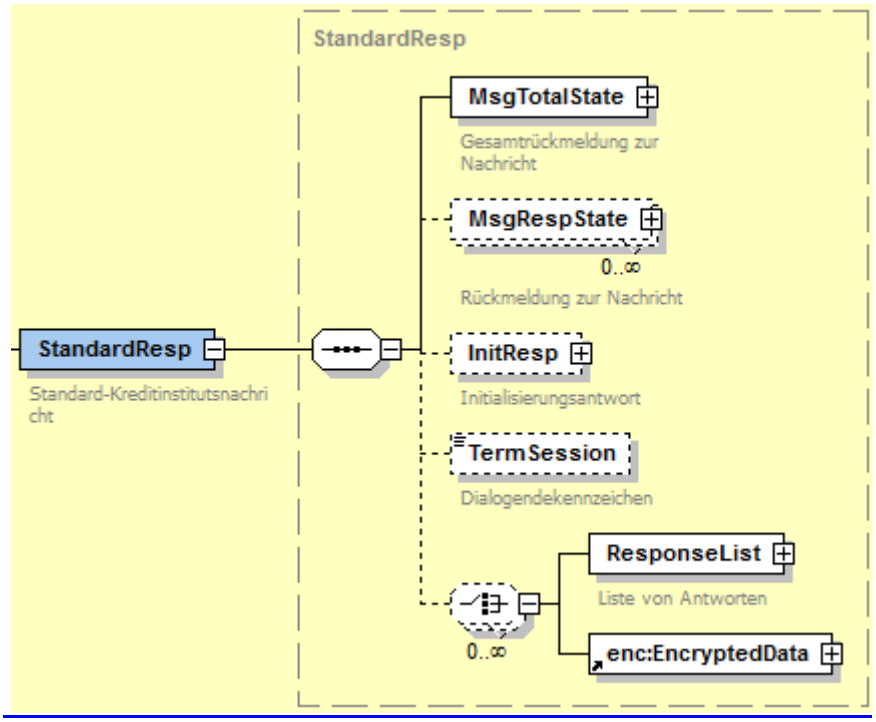


Abbildung 150: Auftragsverschlüsselung in der Standard-Kreditinstitutsnachricht

V.4 Komprimierung

Alle Teile einer FinTS-Nachricht, für die die Syntax eine Verschlüsselung erlaubt, können optional komprimiert werden. Die Komprimierung ist also mit allen Typen von Signaturen, mit verschlüsselten oder unverschlüsselten Nachrichten und auch mit anonymen Nachrichten kombinierbar. In FinTS wird die XML-Kompression als eine spezielle Form der Verschlüsselung behandelt. Daher wird analog zur Verschlüsselung der [XML Encryption]-Standard verwendet.

Bei der Komprimierung ersetzt das *EncryptedData*-Element die komprimierten Nachrichtenteile. Das Attribut *Type* des *EncryptedData*-Elements ist mit dem Wert *http://www.w3.org/2001/04/xmlenc#Element* zu belegen. Das Attribut *Algorithm* gibt das verwendete Komprimierungsverfahren (im Beispiel [RFC1951]) an. Die möglichen Komprimierungsverfahren werden durch das Profil des verwendeten Sicherheitsverfahrens, [HBCI] bzw. [PINTAN], festgelegt. Die, von einem Kreditinstitut unterstützten Komprimierungsverfahren, werden innerhalb der Bankparameterdaten übermittelt. Der komprimierte Teil wird in *base64*-Codierung im Element *CipherValue* abgelegt. Die Elemente *KeyInfo* und *EncryptionProperties* sind nicht vorhanden.

Beispiel:

```
<EncryptedData
 xmlns="http://www.w3.org/2001/04/xmlenc#"
 Type='http://www.w3.org/2001/04/xmlenc#Element'>
  <EncryptionMethod
   Algorithm="http://www.fints.org/spec/xmlschema/4.1/xmlcomp#rfc1951"/>
    <CipherData>
      <CipherValue>DEADBEEFK...VLVLZVL</CipherValue>
    </CipherData>
</EncryptedData>
```

Wenn FinTS-Nachrichtenteile komprimiert und verschlüsselt übertragen werden, sind sie immer zuerst zu komprimieren und danach zu verschlüsseln. Auf der Empfängerseite müssen sie dann erst entschlüsselt und anschließend dekomprimiert werden. Das *EncryptedData*-Element der Verschlüsselung ersetzt dann das *EncryptedData*-Element des komprimierten Nachrichtenteils.

Die mehrfache Verschlüsselung von XML-Fragmenten wird in [XML Encryption] als *super-Encryption* bezeichnet. In FinTS ist keine mehrfache Verschlüsselung zulässig, es sei denn, es handelt sich um eine Komprimierung gefolgt von einer Verschlüsselung.

VI. FINTS-VERSIONSVERWALTUNG

Die mit FinTS V4.1 eingeführte FinTS-Versionverwaltung ermöglicht es dem Kundenprodukt, alle vom Kreditinstitut unterstützten FinTS-Versionen in einer stabilen Ausführungsumgebung des Namespace *0.0* abzufragen und die höchste unterstützte und vom Kreditinstitut favorisierte Version für die Ausführung zu wählen. Details Hierzu befinden sich in *[Formals] im Abschnitt VI FinTS Versionsverwaltung*.

Der neue Namespace *0.0* stellt eine echte Untermenge des Namespace 4.1 dar und beinhaltet nur die notwendigen Syntaxelemente ohne Sicherheit und Parametrisierung für einen Versionsabgleich. Die Grundidee besteht darin, diesen Namespace über alle Versionen bzgl. seiner Protokollstrukturen hinweg unverändert zu lassen. Ausnahme bildet der administrative Geschäftsvorfall *FinTS-Versionsabfrage (FinTSVersionInquiry-1.xsd)*, dessen Aufbau sich theoretisch ändern könnte, was jedoch ebenfalls nicht geplant ist.

a) Benutzerauftrag

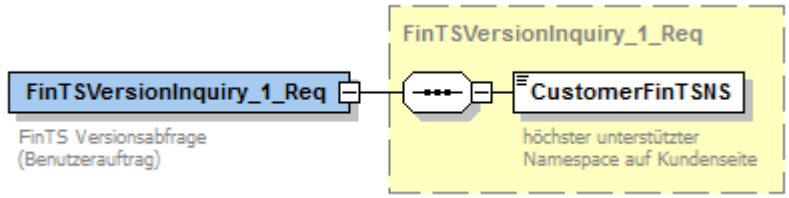


Abbildung 151: Benutzerauftrag FinTS-Versionsabfrage

b) Kreditinstututsrückmeldung

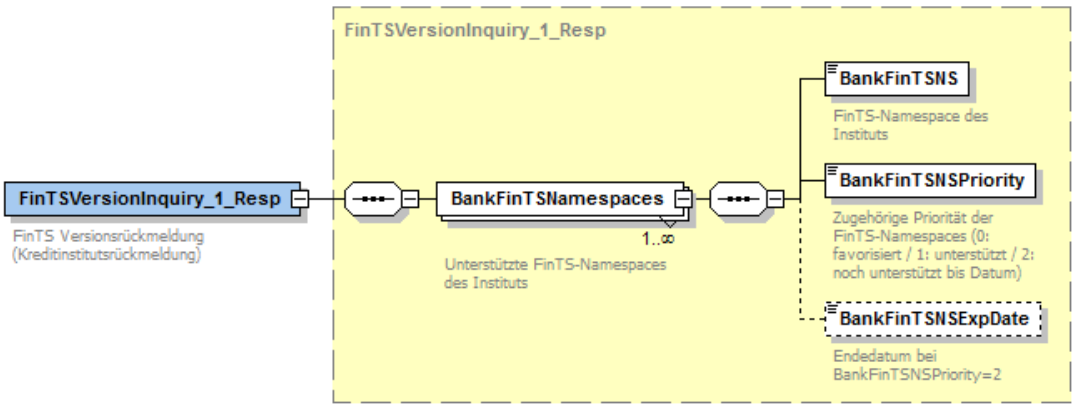


Abbildung 152: Kreditinstitutsrückmeldung FinTS-Versionsabfrage

c) Bankparameterdaten

Zu diesem Auftrag werden keine Bankparameterdaten verwendet. Dies ist eine Spezialfunktion der FinTS-Versionsverwaltung.

VII. ANHANG: KONVENTIONEN ZUR BILDUNG VON ELEMENTNAMEN

Um eine homogene Namensgebung der XML-Deklarationen und -Definitionen zu erreichen, soll die Bildung der qualifizierenden Element- und Typnamen nach generellen Regeln erfolgen. Damit soll ein übergreifend einheitliches Verständnis des semantischen Inhalts von Namen sichergestellt werden.

1. Jedes Element erhält einen eindeutigen Namen aus dem [DataDictionary]. Der Name ist sinnvoll aus der englischsprachigen Bedeutung des Elements abzuleiten und ggf. abzukürzen.
2. Ein Element ist anhand des Namens eindeutig zu identifizieren, d. h. es existieren keine Homonyme.
3. Der Zeichenvorrat für die Bildung der Namen besteht aus den Zeichen A-Z, a-z, 0-9 und dem Unterstrich (_). Insbesondere enthält ein Element- oder Typname keine Leerzeichen.
4. Jedes Kürzel fängt mit einem groß geschriebenen alphabetischen Zeichen an. (Die Zeichenkette „XML“ ist in jeder Kombination von Groß- und Kleinschreibung am Anfang von qualifizierenden Namen verboten)
5. Qualifizierende Namen sind in XML *case sensitive*. Trotzdem dürfen sie nicht in verschiedenen case-sensitiven Schreibweisen verwendet werden. (Die Bedeutung der Elemente *AcctNr* und *acctnr* ist z. B. fachlich identisch, was zu Missverständnissen führen könnte, obwohl die Schreibweisen bei der XML-Verarbeitung unterscheidbar sind.)
6. Als Regel für die Reihenfolge der Kürzel gilt: Kürzel, die den Gegenstand bzw. das Wesen des Datenelements beschreiben, haben möglichst am Ende zu stehen. Kürzel, die dagegen nur beschreibende Funktion haben, werden voran gestellt. Wenn eines der folgenden Kürzel benutzt wird, sollte es das letzte Wort des Elementnamens bilden:
   Acct, *Amt*, *Code*, *Curr*, *Date*, *Id*, *Ref*, *Status*

   Beispiel:

Minimum order amount	*MinOrderAmt*
Currency of exchange	*ExchCurr*
Identification of the party	*PartyId*
Portfolio required	*PortfolioRequired*
Type of denomination allowed	*DenomTypeAllowed*
7. Artikel und Präpositionen (*of*, *in*, *the*) werden nicht im Elementnamen abgebildet.
8. Wenn Datenbeschreibungen aus XML-Fremdformaten (z. B. swiftML) verwendet werden, so werden die dort definierten Namen unverändert übernommen. (Die Fremdformate werden durch einen eigenen Namesraum gekennzeichnet, um Namenskollisionen zu vermeiden.)
9. Allgemein gebräuchliche Akronyme (z. B. ISIN, IBAN) werden stets in dieser Kurzform (alle Buchstaben groß geschrieben) verwendet, ansonsten sind Begriffe auszuschreiben.
10. Eigennamen werden nicht übersetzt.
11. Qualifizierende FinTS-Namen sollten 25 Stellen nicht überschreiten.

12. Damit FinTS-Namen einheitliche und konsistente Namensbestandteile aufweisen, wird ein normierter englischer Wortschatz bei der Namensbildung herangezogen (s. Tabelle).[1]

    Beispiel: Das Adjektiv *allowed* wird bei gleicher Bedeutung anstelle von möglichen Alternativen wie *permissible*, *admissible*, *valid*, *in range* etc. verwendet.

    Begriffe, die in der Liste nicht enthalten sind, werden aus der ausgeschriebenen bzw. sinnvoll abgekürzten, englischen Entsprechung gebildet.

Deutscher Begriff	Englische Entsprechung	Standard-Kürzel
Administration	administration	Admin
Adresse	address	Addr
Aktionär	stockholders	Stockh
alternativ, Alternative	alternative	Alt
änderbar	modifiable	Modif
Änderung	Modification	Mod
Anfrage	request	Req*
Angabe	specification	Spec
Anzahl	quantity	Qty*
~aufdruck (z.B. Adressaufdruck)	printed	Print
ausführen, Ausführung	execute, execution	Exec
Auswahl	selection	Sel
Auszug (z.B. Konto~)	statement	Statemt
Bedingung	condition	Cond*
Begünstigter	beneficiary	Benef
Beschreibung	description	Desc*
Betrag	amount	Amt*
Börse	exchange	Exch
Code	code	Code
Datum	date	Date
Dokument	document	Doc
Durchschnitt, durchschnittlich	average	Avg*
Einzel~ (z.B. ~lastschrift)	single	Single
Einzelheiten	details	Details
Empfänger	beneficiary	Benef
erhältlich	available	Avail
erlaubt	allowed	Allowed
Erläuterung	remark	Rem
Fehler	error	Err
Finanz~	financial	Fin*
Gesamt	total	Total
häufig, Häufigkeit	frequency	Freq*
Identifikation	ID, identifier, identification	Id*

[1] Die mit einem Stern (*) gekennzeichneten Kürzel sind identisch mit den von SWIFT in [swiftML] vorgeschlagenen Abkürzungen.

Deutscher Begriff	Englische Entsprechung	Standard-Kürzel
Information	information	Info*
Institut	institution	Inst
Instrument	instrument	Instr*
Kondition	condition	Cond
Konto	account	Acct*
Kennung	identifier, identification	Id
Kunde	customer	Cust
Kurs	price	Price
Länge	length	Len
Lastschrift	direct debit	DirDeb
Limit	limit	Lim
Liste	list	List
löschen, Löschung	delete, deletion	Del
maximal, Maximum	maximum	Max*
minimal, Minimum	minimum	Min*
möglich	possible	Possible
Name	name	Name
Nachricht	message	Msg
Nummer	number	No
Ort	place	Place
Parameter	parameter	Param
Partei	party	Party
Post~ (z.B. Postanschrift)	mailing	Mail
Post~ (z.B. Postleitzahl)	postal	Post
Produkt	product	Prod
Prozent	percentage	Pct*
Referenz	reference	Ref*
Rückmeldung	response	Resp
Saldo	balance	Bal
Sammel~ (z.B. ~überweisung)	multiple	Mult
Satz	record	Rec
schließen, Schließung	close	Close
Segment	segment	Segm
Signatur	signature	Sig
Sprache	language	Lang
Standard	standard	Std
Status	status	Stat
Stornierung	cancellation	Cancel
Stückelung	denomination	Denom
Synchronisierung	synchronization	Sync
Telefon	telephone	Tel
terminiert	post-dated	Postdated
Transaktion	transaction	Tran*
Typ	type	Type
Überweisung	remittance	Remitt
Unter~ (z.B. Unterkonto)	subsidiary	Sub
unterstützt	supported	Supported
Verfahren	procedure	Proc

Deutscher Begriff	Englische Entsprechung	Standard-Kürzel
Version	version	Ver
Vormerkung	registration	Reg
Wert	value	Val
Währung	currency	Ccy*
wiederkehrend	recurring	Recurr
Zahlung	payment	Paymt
Zahlungsempfänger	payee	Payee
Zeit	time	Time
zeitweilig, vorläufig	temporary	Temp
Zins	interest	Int
Zusatz	extension	Ext