

# **OSCI-Transport 1.2**

**– Spezifikation –**

**Status: FINAL**

**OSCI Leitstelle**

Bremen, den 6. Juni 2002

Die folgenden Firmen und Institutionen waren an der Erstellung dieser Spezifikation beteiligt:

- Bremen Online Services GmbH & Co. KG, Bremen
- cit GmbH, Dettingen unter Teck (MediaKomm Esslingen)
- OSCI Leitstelle, Bremen (Herausgeber)  
Am Fallturm 9  
28359 Bremen  
Internet: <http://www.osci.de>
- PPI Financial Systems GmbH, Hamburg
- SAP AG, Walldorf
- Stadtverwaltung Hagen

Die Erstellung dieses Dokuments wurde im Rahmen des MEDIA@Komm-Projektes durch das Bundesministerium für Wirtschaft und Technologie gefördert.

## Copyright

Die vorliegende Spezifikation für eine automatisiert nutzbare Schnittstelle für die Abwicklung von Geschäftsvorfällen zwischen Bürgern und Kommunen wurde im Auftrag der OSCI-Leitstelle als Herausgeber entwickelt.

Diese Spezifikation ist urheberrechtlich geschützt. Alle Nutzungsrechte liegen bei dem Herausgeber. Herstellern wird zur Implementation von Bürger-, Kommunal-, Intermediär- oder Dienstleistersystemen unentgeltlich ein einfaches Nutzungsrecht eingeräumt. Im Rahmen des genannten Zwecks darf dieses Dokument in unveränderter Form vervielfältigt und zu den nachstehenden Bedingungen verbreitet werden.

Umgestaltungen, Bearbeitungen, Übersetzungen und jegliche Änderungen sind nur nach Rücksprache mit dem Herausgeber zulässig. Kennzeichnungen, Copyright-Vermerke und Eigentumsangaben sind beizubehalten.

Haftung für Mängel dieses Dokuments wird nur bei Vorsatz und grober Fahrlässigkeit übernommen. Hersteller der oben genannten Systeme sind gebeten, Fehler, Unklarheiten oder Interpretationsfreiräume dieser Spezifikation, die die ordnungsgemäße Funktion oder die Interoperabilität behindern, dem Herausgeber zu melden.

Eine Weitergabe dieses Dokuments an Dritte darf nur unentgeltlich, in unveränderter Form und zu den vorstehenden Bedingungen erfolgen.

## Konventionen zur Textauszeichnung

In dieser Spezifikation gelten die folgenden Konventionen:

- Normative Absätze sind hellgrau unterlegt. Beispiel:

Dieser Absatz ist normativ.

In Zweifelsfällen gelten die Festlegungen in Schemas dieser Spezifikation vor normativen Textpassagen dieser Spezifikation. Diese gelten wiederum vor normativen Teilen referenzierter Dokumente und diese schließlich vor nicht normativen Teilen dieser Spezifikation.

- Hervorhebungen sind *kursiv* gesetzt.
- Jede Art von Code ist in `Schreibmaschinenschrift` gesetzt.

## Inhaltsverzeichnis

1. Einleitung.....	5
2. Konventionen zur Notation .....	7
3. Nachrichtenaustausch mittels OSCI-Transport .....	8
3.1 Am Nachrichtenaustausch beteiligte Instanzen.....	8
3.2 Ebenen der Kommunikation .....	8
3.3 Dialoge.....	10
3.4 Zustellungen .....	10
3.5 Szenarien.....	13
3.5.1 One-Way-Message, aktiver Empfänger, Protokollierung.....	13
3.5.2 One-Way-Message, passiver Empfänger, Protokollierung.....	14
3.5.3 Request-Response, passiver Empfänger, Protokollierung .....	15
3.5.4 Request-Response, passiver Empfänger, keine Protokollierung.....	16
4. Sicherheitsmechanismen .....	18
4.1 Digitale Signaturen .....	18
4.1.1 Signieren von Inhaltsdaten .....	19
4.1.2 Signaturprüfung von Inhaltsdaten .....	19
4.1.3 Signieren von Aufträgen und Auftragsantworten .....	19
4.1.4 Signaturprüfung von Aufträgen .....	19
4.1.5 Signaturprüfung von Auftragsantworten .....	19
4.2 Ver- und Entschlüsselung.....	20
4.2.1 Verschlüsselung von Inhaltsdaten .....	21
4.2.2 Entschlüsselung von Inhaltsdaten .....	21
4.2.3 Verschlüsselung von Aufträgen .....	21
4.2.4 Entschlüsselung von Aufträgen .....	21
4.2.5 Verschlüsselung von Auftragsantworten.....	21
4.2.6 Entschlüsselung von Auftragsantworten.....	22
4.3 Protokollierung von Zeitpunkten .....	22
4.4 Zertifikatsprüfungen .....	23
4.5 Challenge-Response-Verfahren .....	23
4.6 Benutzerauthentisierung im Rahmen der expliziten Dialoginitialisierung .....	24
4.7 Prüfung der Messageld .....	24
4.8 Quittungsmechanismen .....	25
5. Reaktionsvorschriften und Rückmeldungen .....	26
5.1 Fehlermeldungen auf Nachrichtenebene.....	27
5.2 Rückmeldungen auf Auftragsebene .....	28
6. Nachrichtenaufbau .....	31
6.1 Versionierung.....	31
6.2 Bedeutung der Elemente mit Namespace <a href="http://www.osci.de/2002/04/osci">http://www.osci.de/2002/04/osci</a> .....	31
6.3 Globale Typdefinitionen .....	35

6.4	Ausprägung von XML-Signature.....	43
6.5	Ausprägung von XML-Encryption.....	46
6.6	Aufträge und Auftragsantworten.....	48
6.6.1	Dialoginitialisierungsauftrag.....	48
6.6.2	Dialoginitialisierungsantwort.....	52
6.6.3	Dialogendeauftrag.....	56
6.6.4	Dialogendeantwort.....	60
6.6.5	MessageId-Anforderungsauftrag.....	63
6.6.6	MessageId-Anforderungsantwort.....	67
6.6.7	Zustellungsauftrag.....	71
6.6.8	Zustellungsantwort.....	76
6.6.9	Zustellungsabholauftrag.....	80
6.6.10	Zustellungsabholantwort.....	85
6.6.11	Laufzettelabholauftrag.....	91
6.6.12	Laufzettelabholantwort.....	96
6.6.13	Weiterleitungsauftrag.....	101
6.6.14	Weiterleitungsantwort.....	105
6.6.15	Annahmearauftrag.....	110
6.6.16	Annahmeantwort.....	115
6.6.17	Abwicklungsauftrag.....	119
6.6.18	Abwicklungsantwort.....	125
6.6.19	Bearbeitungsauftrag.....	132
6.6.20	Bearbeitungsantwort.....	136
6.7	OSCI-Nachrichten.....	140
6.7.1	Verschlüsselte Auftragsdaten.....	141
6.7.2	Unverschlüsselte Auftragsdaten.....	143
6.7.3	Fehlernachricht.....	143
7.	Konformitätskatalog.....	146
8.	Literaturverzeichnis.....	147
8.1	Normative Referenzen.....	147
8.2	Nicht-normative Referenzen.....	148
9.	Glossar.....	149
10.	Abbildungsverzeichnis.....	153

## 1. Einleitung

OSCI-Transport ist ein Standard, mit dem prinzipiell beliebige Informationen automatisiert übertragen werden können. Die Informationen können (signaturgesetzkonform) signiert und durch Verschlüsselung vor Ausspähung geschützt werden. Daneben bietet OSCI-Transport weitere Sicherheitsmechanismen, die insbesondere die Nachvollziehbarkeit der Kommunikation betreffen.

OSCI-Transport bildet den Teil A von OSCI (Online Service Computer Interface), einem Standard für den E-Government-Bereich, der zurzeit in einem breit getragenen gemeinsamen Diskussions- und Kooperationsprozess von Bund, Ländern und Kommunen unter Federführung der Bremer Leitstelle OSCI im Auftrag des KoopA-ADV entwickelt wird. Teil A betrifft dabei die Ebene der Transport- und Sicherheitsfunktionen. Die fachlich/inhaltliche Ebene ist Gegenstand von Teil B.

Wesentliche Designkriterien für OSCI-Transport in der vorliegenden Version waren (vgl. auch [OSCI-ANF]):

- Aufsetzen auf offene Standards  
Insbesondere bilden die folgenden Standards aus dem Umfeld Web-Services die Basis für OSCI-Transport:
  - SOAP [SOAP 1]
  - XML-Signature [XDSIG]
  - XML-Encryption [XENC]
- Technikunabhängigkeit  
OSCI-Nachrichten können mit einem beliebigen technischen Kommunikationsprotokoll übertragen werden.  
Weiterhin stellt OSCI-Transport keine spezifischen Anforderungen an die Plattformen, auf denen Softwaresysteme eingesetzt werden, die OSCI-Nachrichten verarbeiten, oder an die Programmiersprachen, in denen diese Systeme entwickelt werden.
- Skalierbarkeit der Sicherheitsniveaus  
Beispielsweise können zur Unterstützung einfacher Geschäftsvorfälle fortgeschrittene Signaturen eingesetzt werden, während für Vorgänge mit Schriftformerfordernis qualifizierte bzw. akkreditierte elektronische Signaturen zur Anwendung kommen können.

Diese Spezifikation wendet sich an Softwareentwickler. Zum Verständnis sind Kenntnisse in XML [XML], Namespaces [XNAMEESP] und XML-Schema [XSCHEMA] erforderlich.

Um diese Spezifikation möglichst schlank und übersichtlich zu halten, werden relevante Standards im Regelfall lediglich referenziert (und nicht paraphrasiert). Normative und nicht-normative Teile dieses Dokuments werden durch die Textauszeichnung deutlich voneinander abgegrenzt.

Diese Spezifikation gliedert sich in die folgenden Kapitel:

Zunächst werden in Kapitel 2 einige Festlegungen zur Notation getroffen.

In Kapitel 3 werden dann grundlegende Konzepte von OSCI-Transport erläutert.

Kapitel 4 stellt die Sicherheitsmechanismen von OSCI-Transport dar.

In Kapitel 5 werden Reaktionsvorschriften festgelegt und die möglichen Rückmeldungen aufgeführt.

Kapitel 6 beschreibt den Aufbau von OSCI-Nachrichten, enthält die relevanten Schemas und gibt Belegungsrichtlinien.

Abschließend werden in Kapitel 7 Bedingungen aufgeführt, die Softwaresysteme erfüllen müssen, um konform zu dieser Spezifikation zu sein.

## 2. Konventionen zur Notation

Wenn nicht explizit anders angegeben, wird in dieser Spezifikation von folgendem Namespace-Binding ausgegangen:

<b>Namespace-Praefix</b>	<b>Namespace</b>
ds	<a href="http://www.w3.org/2000/09/xmldsig#">http://www.w3.org/2000/09/xmldsig#</a>
osci	<a href="http://www.osci.de/2002/04/osci">http://www.osci.de/2002/04/osci</a>
soap	<a href="http://schemas.xmlsoap.org/soap/envelope">http://schemas.xmlsoap.org/soap/envelope</a>
xenc	<a href="http://www.w3.org/2001/04/xmlenc#">http://www.w3.org/2001/04/xmlenc#</a>
xsd	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>

XML-Attribute treten grundsätzlich unqualifiziert auf, sofern nicht explizit ein Namespace-Präfix angegeben ist.

### 3. Nachrichtenaustausch mittels OSCI-Transport

In diesem Kapitel werden die grundlegenden Konzepte von OSCI-Transport beschrieben.

Zunächst werden in Abschnitt 3.1 die Instanzen aufgeführt, die am Nachrichtenaustausch mittels OSCI-Transport beteiligt sind.

Dann wird in Abschnitt 3.2 dargestellt, auf welchen Ebenen bei OSCI-Transport Informationen ausgetauscht werden.

Abschnitt 3.3 erläutert das Konzept des Dialogs, Abschnitt 3.4 das der Zustellung.

Abschließend werden in Abschnitt 3.5 vier Grundszenarien für den Nachrichtenaustausch mittels OSCI-Transport vorgestellt.

#### 3.1 Am Nachrichtenaustausch beteiligte Instanzen

OSCI-Transport ist ein Protokoll, mit dem prinzipiell beliebige Informationen zwischen Benutzern ausgetauscht werden können.

Der Begriff eines *Benutzers* ist dabei sehr weit gefasst. Bei einem Benutzer kann es sich beispielsweise um eine Person, eine Gruppe von Personen oder auch um ein Softwaresystem handeln. OSCI-Transport verfügt über keine eigene Benutzerverwaltung, sondern arbeitet mit einer offenen Benutzergruppe. Die einzige Voraussetzung, die eine Instanz erfüllen muss, um bei OSCI-Transport als Benutzer zu fungieren, ist der Besitz eines Chiffrierzertifikats. Chiffrierzertifikate werden zur OSCI-Transport-internen Adressierung verwendet.

Zwei Benutzer kommunizieren bei OSCI-Transport niemals direkt, sondern immer über eine weitere Instanz, die als *Intermediär* bezeichnet wird. Der Intermediär tritt als Vermittler in der Kommunikation auf, protokolliert den Nachrichtenfluss, prüft Zertifikate und kann weitere Mehrwertdienste erbringen.

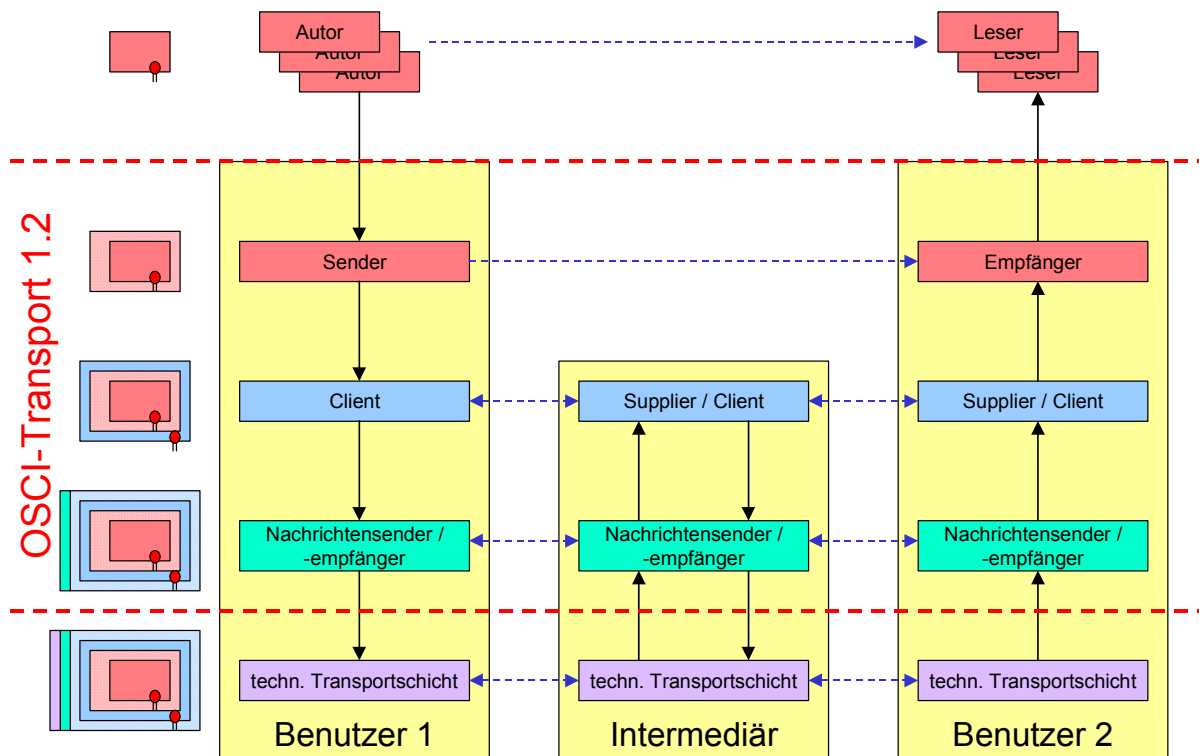
Der Intermediär ist von seiner Rolle her neutral. Er kann jedoch physisch durchaus bei einem Benutzer lokalisiert sein.

Bestimmte Benutzer treten bei OSCI-Transport in der Rolle eines *Dienstanbieters* auf. Neben einem Chiffrierzertifikat benötigt ein Dienstanbieter einen URL, unter dem er (prinzipiell jederzeit) online erreichbar ist.

#### 3.2 Ebenen der Kommunikation

Bei OSCI-Transport werden Informationen auf drei Ebenen ausgetauscht. Je nach Ebene treten die am Nachrichtenaustausch beteiligten Instanzen in unterschiedlichen Rollen auf. Im Folgenden werden die einzelnen Ebenen, die Rollen, welche die beteiligten Instanzen jeweils einnehmen, und die ausgetauschten Daten beschrieben (vgl. auch Abbildung 1).





**Abbildung 1: Ebenen der Kommunikation**

- Geschäftsvorfallsebene:** Ein oder mehrere *Autoren* stellen einem oder mehreren *Lesern* Informationen (sogenannte *Inhaltsdaten*, vgl. [OSCI-ANF]) zur Verfügung. OSCI-Transport stellt keine Anforderungen an Art oder Struktur der Inhaltsdaten. Prinzipiell kann es sich um beliebige Daten handeln. Inhaltsdaten werden in einer *Zustellung* von einem *Sender* an einen *Empfänger* übertragen. Die Rollen *Sender* und *Empfänger* werden beide von Benutzern wahrgenommen. Der *Intermediär* tritt auf Geschäftsvorfallsebene nicht in Erscheinung.

OSCI-Transport sagt nichts aus über die Beziehung von *Autoren* zu *Sendern* oder von *Lesern* zu *Empfängern*. Auch liegt der gegebenenfalls notwendige *Datentransport* von den *Autoren* zum *Sender* und vom *Empfänger* zu den *Lesern* außerhalb des Anwendungsbereichs dieser Spezifikation.
- Auftragsebene:** Ein *Client* richtet einen *Auftrag* an einen *Supplier*. Der *Auftrag* enthält eine *Arbeitsanweisung* und weitere Informationen, die der *Supplier* benötigt, um die *Anweisung* auszuführen und danach eine *Auftragsantwort* an den *Client* zu schicken. In *Aufträgen* und *Auftragsantworten* können insbesondere *Zustellungen* übertragen werden. Im Regelfall treten *Benutzer* als *Clients* und der *Intermediär* als *Supplier* auf. Es gibt jedoch auch Szenarien, in denen der *Intermediär* die Rolle eines *Clients* und ein *Benutzer* (der dann ein *Dienstleister* ist) die eines *Suppliers* übernimmt.
- Nachrichtenebene:** Ein *Nachrichtensender* schickt eine *OSCI-Nachricht* an einen *Nachrichtenempfänger*. Sowohl *Benutzer* als auch der *Intermediär* können beide Rollen wahrnehmen. Einer der Kommunikationspartner (*Nachrichtensender* oder *-empfänger*) muss dabei ein *Intermediär* sein; *Benutzer* kommunizieren niemals direkt.

Eine OSCI-Nachricht umfasst entweder genau einen Auftrag oder genau eine Auftragsantwort oder genau eine Fehlermeldung. Der Auftrag / die Auftragsantwort kann verschlüsselt sein. In diesem Fall enthält die OSCI-Nachricht zusätzlich Informationen, die der Nachrichtenempfänger benötigt, um den Auftrag / die Auftragsantwort zu entschlüsseln.

### 3.3 Dialoge

Der Begriff des *Dialogs* betrifft die Auftragsebene von OSCI-Transport. Auftrag-Antwort-Paare werden zu Dialogen zusammengefasst. Jeder Dialog umfasst mindestens ein Auftrag-Antwort-Paar.

Ein Dialog besteht zwischen genau einem Client und einem Supplier. Er kann auf zwei Arten zustande kommen:

- *explizit* durch einen speziellen Dialoginitialisierungsauftrag des Clients an den Supplier. Der Dialog besteht dann höchstens so lange, bis der Supplier die Antwort auf einen speziellen Dialogendeauftrag abgeschickt hat. Wenn der Supplier einen Fehler meldet, der zum Dialogabbruch führt, wird der Dialog vorzeitig beendet.
- *implizit* durch einen vom Dialoginitialisierungsauftrag verschiedenen Auftrag. Der Dialog umfasst dann genau diesen Auftrag und die zugehörige Antwort.

Innerhalb eines expliziten Dialogs darf ein Client einen Folgeauftrag erst dann abschicken, wenn er die Antwort des Suppliers auf den vorigen Auftrag empfangen hat.

Ist ein expliziter Dialog geöffnet und erhält der Supplier nicht innerhalb einer gewissen (von ihm frei festzulegenden) Zeitspanne einen weiteren Auftrag des Clients, so schließt er den Dialog. Er geht dann davon aus, dass der Client die zuletzt gesendete Auftragsantwort nicht oder nicht unversehrt erhalten hat.

Ein Dialog wird identifiziert durch die ihm zugeordnete *ConversationId*. Die *ConversationId* wird vom Supplier vergeben. Sie ist in dem Sinne eindeutig, dass ein Supplier jede *ConversationId* nur ein einziges Mal vergibt. Die von verschiedenen Suppliers vergebenen *ConversationIds* müssen jedoch nicht notwendig alle paarweise verschieden sein.

Innerhalb eines Dialogs nummerieren Client und Supplier ihre Nachrichten unabhängig voneinander bei 0 beginnend durch. Die Nummer des jeweiligen Auftrags / der jeweiligen Auftragsantwort wird als *SequenceNumber* bezeichnet.

Die Begriffe *ConversationId* und *SequenceNumber* sind aus [EBXML] übernommen.

Durch Angabe des Suppliers, der *ConversationId*, der *SequenceNumber* lässt sich sowohl jeder Auftrag als auch jede Auftragsantwort eindeutig identifizieren.

### 3.4 Zustellungen

Eine *Zustellung* umfasst alle Daten, die von einem Benutzer, der als Sender fungiert, an einen Benutzer übertragen werden, der als Empfänger auftritt. Insbesondere enthält eine *Zustellung* also die Inhaltsdaten.

Für Inhaltsdaten, die in XML formuliert sind, bietet OSCI-Transport sogenannte *Inhaltsdatencontainer*. Andere Inhaltsdaten werden in *Attachments* verpackt.

Inhaltsdatencontainer können durch die Autoren signiert werden. Das Signieren von *Attachments* ist über eine Indirektion ebenfalls möglich: Hierzu wird ein Hashwert über das

Attachment gebildet und in einen Inhaltsdatencontainer eingestellt, der dann mit einer Signatur versehen wird.

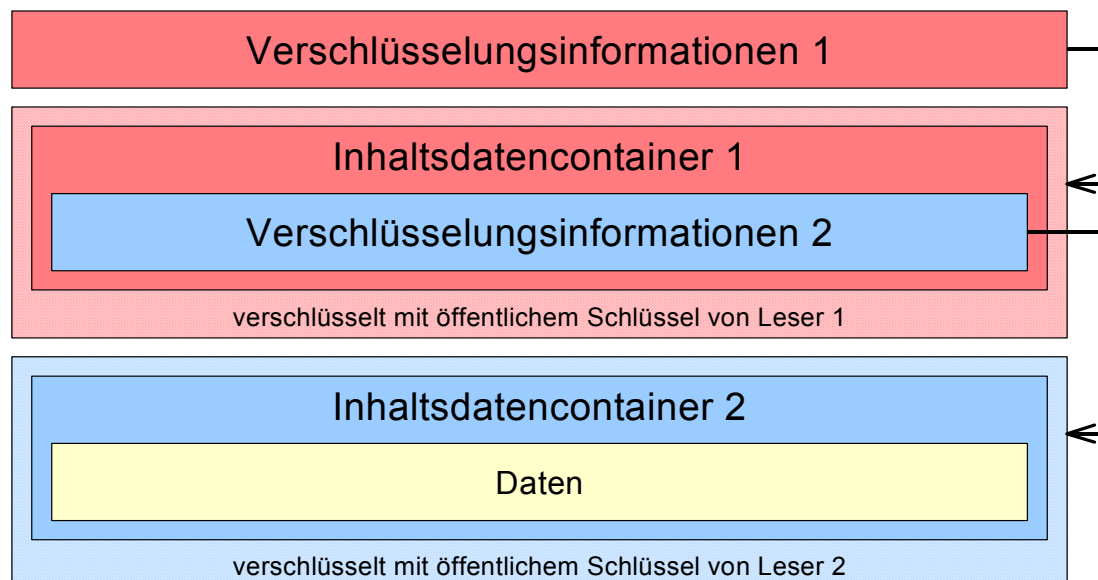
OSCI-Transport bietet auf Ebene der Inhaltsdaten eine Ende-zu-Ende-Verschlüsselung. Das bedeutet, Inhaltsdatencontainer und Attachments können so verschlüsselt werden, dass sie nur durch die Leser eingesehen werden können.

Neben den Inhaltsdaten kann ein Inhaltsdatencontainer Verschlüsselungsinformationen zu weiteren Inhaltsdatencontainern und Attachments in der selben Zustellung enthalten.

Um beispielsweise sicherzustellen, dass zwei Leser nur gemeinsam Zugriff auf die übertragenen Daten erhalten (Mehrfachverschlüsselung, „Vier-Augen-Prinzip“), kann man folgendermaßen vorgehen:<sup>1</sup>

Die eigentlichen Daten werden in einen Inhaltsdatencontainer 2 eingestellt. Dieser wird mit dem öffentlichen Schlüssel von Leser 2 verschlüsselt. Die zugehörigen Verschlüsselungsinformationen (insbesondere der verschlüsselte Sitzungsschlüssel) werden dann in einen Inhaltsdatencontainer 1 verpackt, der mit dem öffentlichen Schlüssel von Leser 1 verschlüsselt wird. In die Zustellung werden abschließend die Verschlüsselungsinformationen zu Inhaltsdatencontainer 1 und die beiden verschlüsselten Inhaltsdatencontainer aufgenommen. In Abbildung 2 ist dies schematisch dargestellt.

### Vier-Augen-Prinzip



**Abbildung 2: Mögliche Realisierung des Vier-Augen-Prinzips**

<sup>1</sup> Die Realisierung einer Mehrfachverschlüsselung bzw. eines "Vier-Augen-Prinzips" wird durch das jeweilige Fachverfahren abgebildet. Das dargestellte Verfahren stellt lediglich ein Beispiel zur Mehrfachverschlüsselung mit Mitteln aus OSCI-Transport dar.

Leser 2 kann nun die Daten in Inhaltsdatencontainer 2 nicht ohne Hilfe von Leser 1 einsehen. Zwar ist der Sitzungsschlüssel, mit dem Inhaltsdatencontainer 2 verschlüsselt ist, mit seinem öffentlichen Schlüssel verschlüsselt. Der verschlüsselte Sitzungsschlüssel befindet sich aber in Inhaltsdatencontainer 1, der nur von Leser 1 entschlüsselt werden kann.

Auch Leser 1 benötigt die Hilfe von Leser 2, um die Daten in Inhaltsdatencontainer 2 einsehen zu können. Zwar befindet sich der Sitzungsschlüssel, mit dem Inhaltsdatencontainer 2 verschlüsselt ist, in Inhaltsdatencontainer 1. Der Sitzungsschlüssel ist aber mit dem öffentlichen Schlüssel von Leser 2 verschlüsselt, so dass er nur von Leser 2 entschlüsselt werden kann.

Leser 1 und Leser 2 müssen zusammenarbeiten, um die Daten einsehen zu können. Zunächst muss Leser 1 den Inhaltsdatencontainer 1 entschlüsseln, damit Leser 2 an die Verschlüsselungsinformationen zu Inhaltsdatencontainer 2 gelangen kann. Leser 2 entschlüsselt dann den Inhaltsdatencontainer 2.

Auch Regelmechanismen lassen sich auf diese Weise in OSCI-Transport abbilden. Hier erhalten die Leser gleichzeitig mit den Informationen, die sie zur Entschlüsselung von Inhaltsdatencontainern bzw. Attachments benötigen, Regeln, die den Zugriff auf diese Inhaltsdatencontainer bzw. Attachments einschränken. Beispielsweise könnte in einem Inhaltsdatencontainer 1 die Verschlüsselungsinformationen zu zwei Inhaltsdatencontainern 2 und 3 sowie die folgende Regel abgelegt sein: „Inhaltsdatencontainer 2 und 3 erst zu einem bestimmten Zeitpunkt öffnen“. Da einem Leser die Verschlüsselungsinformationen zu den Inhaltsdatencontainern 2 und 3 erst nach Öffnen von Inhaltsdatencontainer 1 zur Verfügung stehen, muss er Kenntnis von den Regeln erhalten, bevor er die Inhaltsdatencontainer 2 und 3 entschlüsseln kann. In Abbildung 3 ist dies schematisch dargestellt.

### Regelmechanismus

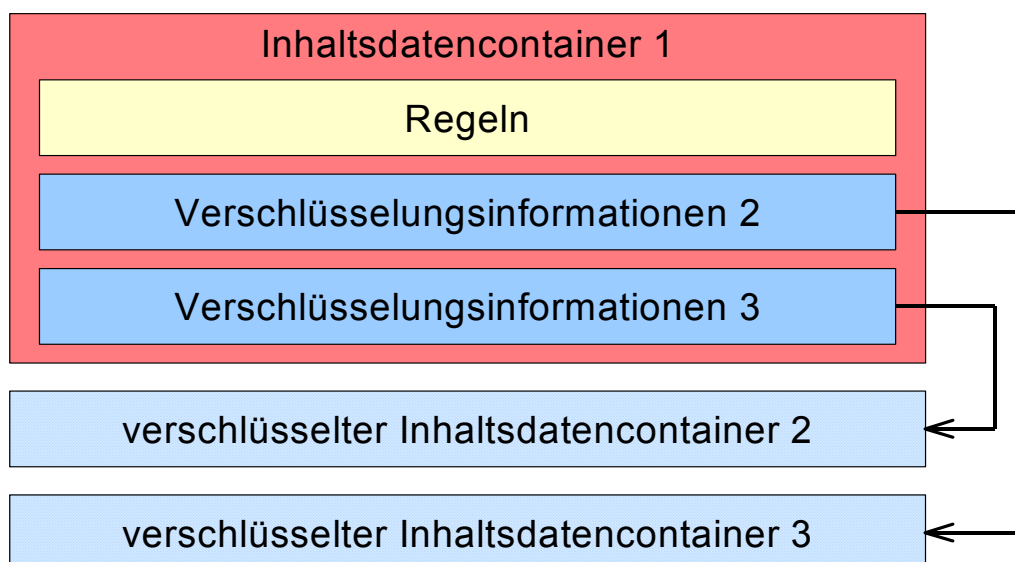


Abbildung 3: Mögliche Realisierung eines Regelmechanismus

Einer Zustellung kann eine sogenannte *MessageId* zugeordnet werden. Dieser Begriff wurde aus [EBXML] übernommen.

Die *MessageId* bezieht sich auf genau eine Zustellung auf dem Weg vom Sender über den Intermediär zum Empfänger. Sie ist weltweit und zeitlich unbegrenzt eindeutig.

Sie wird auf Anforderung des Senders vom Intermediär erzeugt.

### 3.5 Szenarien

OSCI-Transport unterstützt vier verschiedene Szenarien zum Austausch von Inhaltsdaten zwischen Benutzern. Die einzelnen Szenarien unterscheiden sich in drei Gesichtspunkten:

- Wird lediglich eine Zustellung von einem Sender an einen Empfänger übertragen (One-Way-Message) oder reagiert der Empfänger unmittelbar auf den Empfang einer Zustellung mit einer weiteren Zustellung an den Sender (Request-Response)?
- Muss der Empfänger aktiv eine Zustellung beim Intermediär abholen (aktiver Empfänger) oder wird sie ohne sein Zutun automatisch an ihn weitergeleitet (passiver Empfänger)?
- Muss der Inhaltsdatenaustausch nachvollziehbar protokolliert werden oder ist dies nicht erforderlich?

Das zu verwendende Kommunikationsszenario vom Client festgelegt.

Im folgenden werden die Szenarien dargestellt. Die dabei erwähnten Auftrags- und Auftragsantworttypen werden in Abschnitt 6.6 ausführlich erläutert.

#### 3.5.1 One-Way-Message, aktiver Empfänger, Protokollierung

In diesem Szenario werden Inhaltsdaten von einem Benutzer 1 an einen Benutzer 2 übertragen. Der Benutzer 2 muss dabei selbst aktiv werden, um die Zustellung zu erhalten. Die Übertragung wird nachvollziehbar protokolliert.

Dieses Szenario bietet sich an, wenn nicht sichergestellt werden kann, dass Benutzer 2 ständig erreichbar ist. Benutzer 2 braucht in diesem Szenario nicht unbedingt über einen URL zu verfügen. Er muss also kein Dienstanbieter sein.

Der grobe Ablauf stellt sich folgendermaßen dar:

1. Benutzer 1 sendet einen *MessageId*-Anforderungsauftrag an den Intermediär.
2. Der Intermediär erzeugt eine neue *MessageId* und sendet sie in einer *MessageId*-Anforderungsantwort an Benutzer 1.
3. Benutzer 1 sendet einen Zustellungsauftrag an den Intermediär. Der Zustellungsauftrag enthält eine Zustellung, die durch die soeben erhaltene *MessageId* identifiziert wird.
4. Der Intermediär erzeugt einen Laufzettel zu der soeben erhaltenen Zustellung und protokolliert auf diesem den Empfang. Er sendet eine Zustellungsantwort an Benutzer 1.
5. Benutzer 2 sendet einen Dialoginitialisierungsauftrag an den Intermediär.
6. Der Intermediär reagiert mit einer Dialoginitialisierungsantwort an Benutzer 2.
7. Benutzer 2 sendet einen Abholauftrag an den Intermediär, mit dem er für ihn vorliegende Zustellungen anfordert.
8. Der Intermediär sendet eine Abholantwort, welche die Zustellung von Benutzer 1 enthält, an Benutzer 2 und protokolliert die Weiterleitung der Zustellung auf dem Laufzettel.
9. Benutzer 2 sendet einen Dialogendeauftrag an den Intermediär.

10. Der Intermediär protokolliert den Empfang der Zustellung durch Benutzer 2 auf dem Laufzettel und sendet eine Dialogendeantwort an Benutzer 2.

Abbildung 4 enthält eine schematische Darstellung der wesentlichen Schritte dieses Ablaufs.

### One-Way-Message, aktiver Empfänger

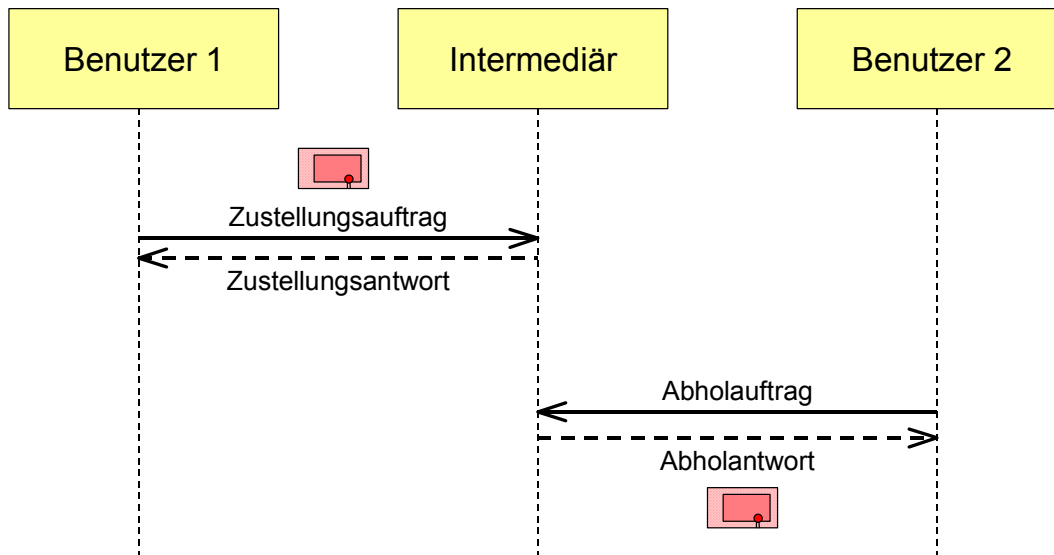


Abbildung 4: One-Way-Message, aktiver Empfänger

#### 3.5.2 One-Way-Message, passiver Empfänger, Protokollierung

In diesem Szenario werden Inhaltsdaten von einem Benutzer 1 an einen Benutzer 2 übertragen. Die Zustellung wird ohne Zutun von Benutzer 2 an ihn weitergeleitet. Die Übertragung wird nachvollziehbar protokolliert.

Dieses Szenario bietet sich an, wenn Benutzer 2 die Zustellung ohne Verzögerung erhalten soll. Benutzer 2 muss in diesem Szenario ständig unter einem URL erreichbar sein. Es handelt sich hier bei Benutzer 2 also um einen Dienstanbieter.

Der grobe Ablauf stellt sich folgendermaßen dar:

1. Benutzer 1 sendet einen Messageld-Anforderungsauftrag an den Intermediär.
2. Der Intermediär erzeugt eine neue Messageld und sendet sie in einer Messageld-Anforderungsantwort an Benutzer 1.
3. Benutzer 1 sendet einen Weiterleitungsauftrag an den Intermediär. Der Weiterleitungsauftrag enthält eine Zustellung, die durch die soeben erhaltene Messageld identifiziert wird.
4. Der Intermediär erzeugt einen Laufzettel zu der soeben erhaltenen Zustellung und protokolliert auf diesem den Empfang. Er sendet einen Annahmearauftrag mit der

Zustellung an Benutzer 2 und protokolliert die Weiterleitung der Zustellung auf dem Laufzettel.

5. Benutzer 2 sendet eine Annahmearbeit an den Intermediär und bestätigt damit, dass er die Zustellung erhalten hat.
6. Der Intermediär protokolliert den Empfang der Zustellung durch Benutzer 2 auf dem Laufzettel und sendet eine Weiterleitungsantwort an Benutzer 1.

Abbildung 5 enthält eine schematische Darstellung der wesentlichen Schritte dieses Ablaufs.

### One-Way-Message, passiver Empfänger

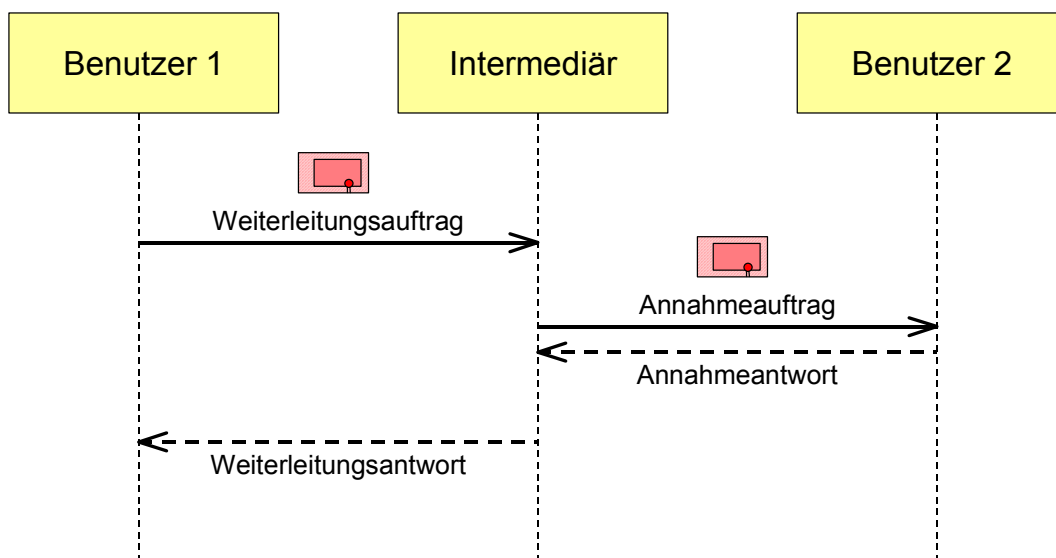


Abbildung 5: One-Way-Message, passiver Empfänger

#### 3.5.3 Request-Response, passiver Empfänger, Protokollierung

In diesem Szenario wird zunächst eine Zustellung 1 von einem Benutzer 1 an einen Benutzer 2 übertragen, dann eine Zustellung 2 von Benutzer 2 an Benutzer 1. Die Übertragung wird nachvollziehbar protokolliert.

Dieses Szenario bietet sich an, wenn Benutzer 2 unmittelbar auf die von Benutzer 1 geschickte Zustellung reagieren soll. Benutzer 2 muss in diesem Szenario ständig unter einem URL erreichbar sein. Es handelt sich hier bei Benutzer 2 also um einen Dienstanbieter.

Der grobe Ablauf stellt sich folgendermaßen dar:

1. Benutzer 1 sendet einen Messaged-Anforderungsauftrag an den Intermediär.
2. Der Intermediär erzeugt eine neue Messaged und sendet sie in einer Messaged-Anforderungsantwort an Benutzer 1.
3. Benutzer 1 sendet einen Dialoginitialisierungsauftrag an den Intermediär.

4. Der Intermediär reagiert mit einer Dialoginitialisierungsantwort an Benutzer 1.
5. Benutzer 1 sendet einen Abwicklungsauftrag mit Zustellung 1 an den Intermediär.
6. Der Intermediär erzeugt einen Laufzettel 1 zu Zustellung 1 und protokolliert auf diesem den Empfang. Er sendet einen Bearbeitungsauftrag mit Zustellung 1 an Benutzer 2 und protokolliert die Weiterleitung von Zustellung 1 auf Laufzettel 1.
7. Benutzer 2 sendet einen Messageld-Anforderungsauftrag an den Intermediär.
8. Der Intermediär erzeugt eine neue Messageld und sendet sie in einer Messageld-Anforderungsantwort an Benutzer 2.
9. Benutzer 2 sendet eine Bearbeitungsantwort mit Zustellung 2 an den Intermediär.
10. Der Intermediär erzeugt einen Laufzettel 2 zu Zustellung 2 und protokolliert auf diesem den Empfang. Weiterhin protokolliert er den Empfang von Zustellung 1 durch Benutzer 2 auf Laufzettel 1. Er sendet eine Abwicklungsantwort mit Zustellung 2 an Benutzer 1.
11. Benutzer 1 sendet einen Dialogendeauftrag an den Intermediär.
12. Der Intermediär protokolliert den Empfang von Zustellung 2 durch Benutzer 2 auf Laufzettel 2 und sendet eine Dialogendeantwort an Benutzer 1.

Abbildung 6 enthält eine schematische Darstellung der wesentlichen Schritte dieses Ablaufs.

### Request-Response, passiver Empfänger

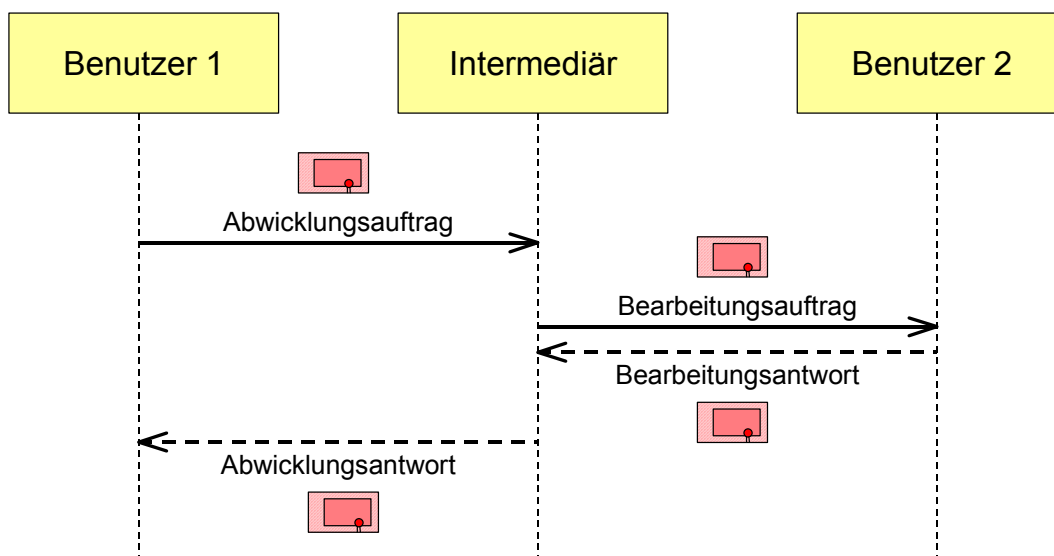


Abbildung 6: Request-Response, passiver Empfänger

#### 3.5.4 Request-Response, passiver Empfänger, keine Protokollierung

Wie im vorigen Szenario wird hier zunächst eine Zustellung 1 von einem Benutzer 1 an einen Benutzer 2 übertragen, dann eine Zustellung 2 von Benutzer 2 an Benutzer 1. Dabei wird die Übertragung allerdings *nicht* protokolliert.



Dieses einfache Szenario ist für Fälle gedacht, in denen keine der Parteien nach erfolgter Kommunikation nachweisen muss, dass sie eine der Zustellungen abgeschickt oder erhalten hat, und eine mögliche Doppeleinreichung einer der Zustellungen kein Problem darstellt.

Der grobe Ablauf stellt sich folgendermaßen dar:

1. Benutzer 1 sendet einen Dialoginitialisierungsauftrag an den Intermediär.
2. Der Intermediär reagiert mit einer Dialoginitialisierungsantwort an Benutzer 1.
3. Benutzer 1 sendet einen Abwicklungsauftrag mit Zustellung 1 an den Intermediär.
4. Der Intermediär sendet einen Bearbeitungsauftrag mit Zustellung 1 an Benutzer 2.
5. Benutzer 2 sendet eine Bearbeitungsantwort mit Zustellung 2 an den Intermediär.
6. Der Intermediär sendet eine Abwicklungsantwort mit Zustellung 2 an Benutzer 1.
7. Benutzer 1 sendet einen Dialogendeauftrag an den Intermediär.
8. Der Intermediär reagiert mit einer Dialogendeantwort an Benutzer 1.

## 4. Sicherheitsmechanismen

OSCI-Transport verfügt über eine Reihe von Mechanismen zur Absicherung des Datenaustauschs. In diesem Kapitel werden die einzelnen Mechanismen beschrieben. Inhalt und Überschriften der einzelnen Abschnitte orientieren sich an [OSCI-ANF, Kapitel 6]. Anstelle des dort verwendeten allgemeineren Begriffs der Nutzungsdaten wird hier allerdings von Aufträgen und Auftragsantworten gesprochen, da diese oft unterschiedlich zu behandeln sind.

### 4.1 Digitale Signaturen

Sowohl Inhalts- als auch Auftragsdaten können digital signiert werden. Dabei wird nach [XDSIG] vorgegangen.

Die Erzeugung einer Signatur erfolgt, indem zunächst zu jedem XML-Element, das durch die Signatur abgesichert werden soll, ein Hashwert berechnet wird. Die Hashwerte werden in einem speziellen XML-Element gesammelt. Über dieses Element wird ein weiterer Hashwert berechnet, der zunächst speziell formatiert und dann mit dem privaten Schlüssel des Signierenden verschlüsselt wird.

Die Prüfung einer Signatur erfolgt, indem der Hashwert über das XML-Element, in dem die Hashwerte zu den abgesicherten Elementen gesammelt worden sind, erneut berechnet und formatiert wird. Das Ergebnis muss mit der entschlüsselten Signatur übereinstimmen. Zusätzlich werden alle Hashwerte über die abgesicherten Elemente erneut berechnet und mit den signierten Hashwerten verglichen.

In der folgenden Tabelle sind die Algorithmen aufgeführt, die bei OSCI-Transport zur Erzeugung von Hashwerten verwendet werden dürfen.

Hashalgorithmus	Algorithmus-Identifizier
SHA-1	<a href="http://www.w3.org/2000/09/xmlldsig#sha1">http://www.w3.org/2000/09/xmlldsig#sha1</a> [XDSIG, Abschnitt 6.2.1]
RIPEND-160	<a href="http://www.osci.de/2002/04/osci#ripemd160">http://www.osci.de/2002/04/osci#ripemd160</a>

Die Signieralgorithmen, die von OSCI-Transport unterstützt werden, sind in der folgenden Tabelle aufgelistet.

Signieralgorithmus	Algorithmus-Identifizier
Signaturschema RSASSA-PKCS1-v1_5 [PKCS 1] mit Hashalgorithmus SHA-1. Modullänge des RSA-Schlüsselpaares mind. 1024 Bit	<a href="http://www.w3.org/2000/09/xmlldsig#rsa-sha1">http://www.w3.org/2000/09/xmlldsig#rsa-sha1</a> [XDSIG, Abschnitt 6.4.2]
Signaturschema RSASSA-PKCS1-v1_5 [PKCS 1] mit Hashalgorithmus RIPEND-160. Modullänge des RSA-Schlüsselpaares mind. 1024 Bit	<a href="http://www.osci.de/2002/04/osci#rsa-ripemd160">http://www.osci.de/2002/04/osci#rsa-ripemd160</a>

#### 4.1.1 Signieren von Inhaltsdaten

Signaturen auf Geschäftsvorfallsebene werden gegebenenfalls durch die Autoren erzeugt.

#### 4.1.2 Signaturprüfung von Inhaltsdaten

Leser können im eigenen Interesse Signaturen auf Geschäftsvorfallsebene prüfen, sind dazu jedoch nicht verpflichtet. Der öffentliche Schlüssel, der zur Signaturprüfung verwendet wird, stammt aus einem Zertifikat, das zusammen mit den signierten Daten übermittelt wird. Die Reaktion auf eine fehlende oder fehlerhafte Signatur liegt im Ermessen der Leser.

#### 4.1.3 Signieren von Aufträgen und Auftragsantworten

Aufträge und Auftragsantworten können (unabhängig von bzw. zusätzlich zu einer eventuell vorhandenen Signatur auf Inhaltsdatenebene) signiert werden, um ihre Authentizität und Integrität zu gewährleisten.

Signaturen von Aufträgen werden gegebenenfalls durch den Client erzeugt.

Signaturen von Auftragsantworten werden gegebenenfalls durch den Supplier erzeugt.

#### 4.1.4 Signaturprüfung von Aufträgen

Wenn ein Auftrag signiert ist, ist der Supplier zur Signaturprüfung verpflichtet. Der öffentliche Schlüssel, der zur Signaturprüfung verwendet wird, stammt aus einem Zertifikat, das im Auftrag übermittelt wird. Der Supplier ist berechtigt, nicht signierte Aufträge abzulehnen. Aufträge, deren Signatur fehlerhaft ist, sind abzulehnen. Die Ablehnung wird dem Client unmittelbar in der Auftragsantwort mitgeteilt. Ebenso sind Aufträge abzulehnen, für die der Supplier feststellt, dass sie mit einem privaten Schlüssel signiert worden sind, für den aus dem Zertifikat hervorgeht, dass er nicht zum Signieren verwendet werden darf.

#### 4.1.5 Signaturprüfung von Auftragsantworten

Die Signatur einer Auftragsantwort kann durch den Client geprüft werden.

Der öffentliche Schlüssel, der gegebenenfalls zur Signaturprüfung verwendet wird, stammt aus einem Zertifikat, das in der Auftragsantwort übermittelt wird.

Zwei Fälle sind zu unterscheiden:

- Der Client ist ein Intermediär. Dies ist der Fall, wenn er im Rahmen der Abarbeitung eines Weiterleitungs- oder Abwicklungsauftrags eines Benutzers einen Annahme- bzw. Bearbeitungsauftrag an einen Dienstleister gerichtet hat.
- Der Client ist ein Benutzer. Dies trifft in allen übrigen Fällen zu.

Ein Intermediär, der als Client agiert, muss die Signaturen von Auftragsantworten prüfen. Stellt er fest, dass die Signatur einer Auftragsantwort fehlerhaft oder mit einem privaten Schlüssel erzeugt worden ist, für den aus dem Zertifikat hervorgeht, dass er nicht zum Signieren verwendet werden darf, so hat er die Auftragsantwort als manipuliert anzusehen. Er gibt dann dem Benutzer, der ihm den Weiterleitungs- oder Abwicklungsauftrag gesendet hat, eine entsprechende Rückmeldung.

Fehlt die Signatur, so erhält der Benutzer eine entsprechende Warnung.

Ein Benutzer, der als Client agiert, kann im eigenen Interesse die Signaturen von Auftragsantworten prüfen, ist hierzu jedoch nicht verpflichtet. Stellt er fest, dass die Signatur einer Auftragsantwort fehlerhaft oder mit einem privaten Schlüssel erzeugt worden ist, für

den aus dem Zertifikat hervorgeht, dass er nicht zum Signieren verwendet werden darf, und ist ein expliziter Dialog geöffnet, so darf der Benutzer keinen weiteren Auftrag (auch keinen Dialogendeauftrag) in diesem Dialog mehr schicken.

Der Benutzer muss davon ausgehen, dass er nicht die Auftragsantwort erhalten hat, die der Supplier an ihn geschickt hat. Erhält der Supplier nicht innerhalb einer gewissen Zeitspanne einen weiteren Auftrag des Benutzers, so schließt er den Dialog und geht davon aus, dass der Benutzer die zuletzt gesendete Auftragsantwort nicht oder nicht unversehrt erhalten hat (vgl. Abschnitt 3.3).

Die Reaktion auf eine fehlende Signatur von Auftragsantworten liegt im Ermessen des Benutzers.

## 4.2 Ver- und Entschlüsselung

Sowohl Inhaltsdaten als auch Aufträge und Auftragsantworten können verschlüsselt werden. Auftragsantworten auf Dialoginitialisierungen sind zwingend zu verschlüsseln (siehe Abschnitt 4.6).

Die Verschlüsselung erfolgt gemäß [XENC], wobei ein Hybridverschlüsselungsalgorithmus eingesetzt wird.

Zunächst wird ein zufälliger Sitzungsschlüssel für einen symmetrischen Verschlüsselungsalgorithmus erzeugt. Mit diesem Schlüssel werden die eigentlichen Daten verschlüsselt. In einem zweiten Schritt wird dann der Sitzungsschlüssel mit dem öffentlichen Chiffrierschlüssel des Adressaten<sup>2</sup> verschlüsselt. Verschlüsselte Daten und verschlüsselter Sitzungsschlüssel werden an den Adressaten weitergeleitet.

Der Adressat entschlüsselt zunächst den Sitzungsschlüssel mit seinem privaten Chiffrierschlüssel. Mit dem entschlüsselten Sitzungsschlüssel kann er dann die eigentlichen Daten entschlüsseln.

Die symmetrischen Verschlüsselungsalgorithmen, die bei OSCI-Transport zur Verschlüsselung der eigentlichen Daten verwendet werden können, sind in der folgenden Tabelle zusammengestellt.

Verschlüsselungs- algorithmus	Algorithmus-Identifizier
Two-Key-Triple-DES	<a href="http://www.w3.org/2001/04/xmlenc#tripleDES-cbc">http://www.w3.org/2001/04/xmlenc#tripleDES-cbc</a> [XENC, Abschnitt 5.2.1]
AES-128	<a href="http://www.w3.org/2001/04/xmlenc#aes128-cbc">http://www.w3.org/2001/04/xmlenc#aes128-cbc</a> [XENC, Abschnitt 5.2.2]
AES-192	<a href="http://www.w3.org/2001/04/xmlenc#aes192-cbc">http://www.w3.org/2001/04/xmlenc#aes192-cbc</a> [XENC, Abschnitt 5.2.2]
AES-256	<a href="http://www.w3.org/2001/04/xmlenc#aes256-cbc">http://www.w3.org/2001/04/xmlenc#aes256-cbc</a> [XENC, Abschnitt 5.2.2]

[XENC] umfasst Vorgaben, wie diese Algorithmen benutzt werden müssen: Insbesondere sind sie jeweils im CBC-Modus einzusetzen, es ist stets ein mit Zufallszahlen gefüllter

<sup>2</sup> Der Adressat ist bei Aufträgen der Supplier, bei Auftragsantworten der Client und bei Inhaltsdaten ein Leser.

Initialisierungsvektor zu verwenden und das Padding ist auf eine bestimmte Weise vorzunehmen (ein Padding nach [PKCS 5] stellt einen Spezialfall des in [XENC] beschriebenen Verfahrens dar).

Es ist sicherzustellen, dass für Daten, die an unterschiedliche Adressaten gehen, nicht derselbe Sitzungsschlüssel verwendet wird.

Die Verschlüsselung des Sitzungsschlüssels erfolgt mittels RSAES-PKCS1-v1\_5 [PKCS 1] (Algorithmus-Identifizier: [http://www.w3.org/2001/04/xmlenc#rsa-1\\_5](http://www.w3.org/2001/04/xmlenc#rsa-1_5) [XENC, Abschnitt 5.4.1]). Die Modullänge des verwendeten RSA-Schlüsselpaares muss dabei mindestens 1024 Bit betragen.

#### 4.2.1 Verschlüsselung von Inhaltsdaten

Inhaltsdaten werden ggf. durch die Autoren verschlüsselt. Die öffentlichen Schlüssel der Leser, die zum Verschlüsseln verwendet werden, stammen aus Zertifikaten. Wie die Autoren an diese Zertifikate gelangen, wird in dieser Spezifikation nicht beschrieben.

#### 4.2.2 Entschlüsselung von Inhaltsdaten

Die Leser ermitteln anhand von Zertifikaten, die den verschlüsselten Inhaltsdaten beigelegt sind, welche ihrer privaten Schlüssel zur Entschlüsselung benötigt werden.

Die Reaktion auf Inhaltsdaten, die nicht entschlüsselt werden können, liegt im Ermessen der Leser.

#### 4.2.3 Verschlüsselung von Aufträgen

Aufträge werden ggf. durch den Client verschlüsselt. Der öffentliche Schlüssel des Suppliers, der zum Verschlüsseln verwendet wird, stammt aus einem Zertifikat. Wie der Client an dieses Zertifikat gelangt, wird in dieser Spezifikation nicht beschrieben.

#### 4.2.4 Entschlüsselung von Aufträgen

Der Supplier ermittelt anhand eines Zertifikats, das den verschlüsselten Daten beigelegt ist, welcher seiner privaten Schlüssel zur Entschlüsselung benötigt wird.

Geht aus dem Zertifikat hervor, dass der enthaltene öffentliche Schlüssel nicht zur Verschlüsselung verwendet werden darf, so ist der Supplier nicht verpflichtet, den Auftrag zu entschlüsseln. Der Supplier braucht in diesem Fall nicht zu antworten.

Ist ein Supplier nicht in der Lage, einen Auftrag zu entschlüsseln, so ist er ebenfalls nicht verpflichtet, den Auftrag zu beantworten.

Der Grund hierfür ist, dass der Client, der den Auftrag gesendet hat, in diesem Fall mit Mitteln von OSCI-Transport nicht ermittelt werden kann.

#### 4.2.5 Verschlüsselung von Auftragsantworten

Auftragsantworten werden ggf. durch den Supplier verschlüsselt. Der öffentliche Schlüssel des Clients, der zum Verschlüsseln verwendet wird, stammt aus einem Zertifikat. Der Supplier entnimmt dieses Zertifikat

- dem Dialoginitialisierungsauftrag, falls der Auftrag innerhalb eines expliziten Dialogs gesendet worden ist, oder
- andernfalls dem Auftrag, auf den sich die zu verschlüsselnde Auftragsantwort bezieht.

#### 4.2.6 Entschlüsselung von Auftragsantworten

Der Client ermittelt anhand eines Zertifikats, das den verschlüsselten Daten beigelegt ist, welcher seiner privaten Schlüssel zur Entschlüsselung benötigt wird.

Kann ein Client einen Auftrag nicht entschlüsseln und ist ein expliziter Dialog geöffnet, so darf der Client keinen weiteren Auftrag (auch keinen Dialogendeauftrag) in diesem Dialog mehr schicken.

Andernfalls würde der Supplier davon ausgehen, dass der Client die Auftragsantwort erhalten hat und interpretieren konnte.

#### 4.3 Protokollierung von Zeitpunkten

Der Intermediär führt zu jeder Zustellung einen sogenannten *Laufzettel*. Auf dem Laufzettel hält er insbesondere die folgenden Zeitpunkte fest:

##### 1. Zeitpunkt der Einreichung beim Intermediär

Aus technischer Sicht ist dies der Zeitpunkt, zu dem der Intermediär feststellt, dass er einen Zustellungs-, Weiterleitungs- oder Abwicklungsauftrag empfangen hat.

##### 2. Zeitpunkt der Weiterleitung an den Empfänger

Aus technischer Sicht ist dies der Zeitpunkt, zu dem der Intermediär die Nachricht an den Empfänger, welche die Zustellung enthält, vollständig aufgebaut hat. Insbesondere kann dieser Zeitpunkt vom Zeitpunkt der technischen Weiterleitung abweichen.

##### 3. Zeitpunkt der Empfangsbestätigung durch den Empfänger

Aus technischer Sicht ist dies genau einer von zwei Zeitpunkten:

1. die Zustellung ist mit einem Annahme- oder Bearbeitungsauftrag geschickt worden: der Zeitpunkt, zu dem der Intermediär feststellt, dass der Empfänger eine positive Rückmeldung geschickt hat;
2. die Zustellung ist mit einer Zustellungsabholantwort geschickt worden: der Zeitpunkt, zu dem der Intermediär feststellt, dass er einen weiteren Auftrag innerhalb des selben expliziten Dialogs erhalten hat, in dem die Zustellung übermittelt worden ist.

Auf Anforderung des Senders kann der Intermediär kryptographische Zeitstempel zu diesen Zeitpunkten erzeugen.

Im Fall 1 wird der Zeitstempel über den Zustellungs-, Weiterleitungs- oder Abwicklungsauftrag des Senders gebildet.

Im Fall 2 wird der Zeitstempel über den Bearbeitungs- oder Annahmearauftrag, bzw. die Zustellungsabholantwort des Intermediärs gebildet.

Im Fall 3.1 wird der Zeitstempel über die empfangene Auftragsantwort des Empfängers gebildet.

Im Fall 3.2 wird der Zeitstempel über den empfangenen weiteren Auftrag des Empfängers gebildet.

Zum Format kryptographischer Zeitstempel macht diese Spezifikation keinerlei Vorgaben.

Es wird jedoch darauf hingewiesen, dass insbesondere Zeitstempel nach [ISIS-MTT, Part 4] verwendet werden können.

#### 4.4 Zertifikatsprüfungen

Als Zertifikatsformat unterstützt OSCI-Transport X.509v3 in der in [ISIS-MTT, Part 1] beschriebenen Ausprägung.

Der Intermediär hat sämtliche Zertifikate zu prüfen, die sich in einer empfangenen Nachricht auf Auftragsebene befinden. Weiterhin hat der Intermediär alle Zertifikate der zugehörigen Zertifikatsketten zu prüfen.

Die folgenden Prüfungen sind jeweils vorzunehmen:

- Mathematische Prüfung der Signatur des Zertifikats: Der Hashwert über das Zertifikat wird erneut berechnet und formatiert. Das Ergebnis muss mit der entschlüsselten Signatur des Zertifikats übereinstimmen.
- Offline-Gültigkeitsprüfung: Geprüft wird, dass der Zeitpunkt der Prüfung innerhalb des Gültigkeitszeitraums liegt, der im Zertifikat angegeben ist.
- Online-Gültigkeitsprüfung: Geprüft wird, dass das Zertifikat zum Zeitpunkt der Prüfung nicht widerrufen ist.

Das Ergebnis jeder dieser Prüfungen vermerkt der Intermediär in einem Prüfprotokoll. Ist der Intermediär nicht in der Lage, eine der Prüfungen durchzuführen (weil ihm beispielsweise die Zertifizierungsinstantz unbekannt ist), so vermerkt er das ebenfalls im Prüfprotokoll.

Innerhalb eines Dialoges braucht jedes Zertifikat nur einmal geprüft zu werden.

Der Intermediär muss einen Zustellungs-, Weiterleitungs- oder Abwicklungsauftrag ablehnen, wenn eines der Chiffrierzertifikate der Leser oder das Chiffrierzertifikat des Benutzers, der auf Auftragsebene als Supplier fungiert, widerrufen ist. Der Client erhält eine entsprechende Rückmeldung unmittelbar in der Antwort auf den entsprechenden Auftrag.

Stellt der Intermediär fest, dass das Chiffrierzertifikat des Clients widerrufen worden ist, so darf er seine Auftragsantwort nicht mit diesem Zertifikat verschlüsseln. Der Client erhält in diesem Fall eine unverschlüsselte Auftragsantwort, die lediglich auf diesen Umstand hinweist.

Insbesondere werden also Dialoginitialisierungsaufträge eines Clients abgelehnt, für dessen Chiffrierzertifikat dem Supplier bekannt ist, dass es widerrufen worden ist.

Sender und Autoren sollten vor dem Verschlüsseln mindestens die offline möglichen Prüfungen für das Verschlüsselungszertifikat des jeweiligen Empfängers / der jeweiligen Leser vornehmen. Ergibt eine dieser Prüfungen ein negatives Ergebnis, so darf dieses Zertifikat nicht zur Verschlüsselung verwendet werden.

Empfänger und Leser sollten nach dem Empfang unterschriebener Daten mindestens die offline möglichen Prüfungen für die Signierzertifikate des jeweiligen Senders / der jeweiligen Autoren vornehmen. Die Reaktion auf ein negatives Ergebnis einer dieser Prüfungen liegt im Ermessen des Empfängers / der Leser.

#### 4.5 Challenge-Response-Verfahren

OSCI-Transport verwendet ein Challenge-Response-Verfahren, um sicherzustellen, dass die an einem Dialog beteiligten Instanzen (Client und Supplier) während eines Dialogs nicht wechseln können.

Der Client schickt in jedem Auftrag einen frei gewählten Wert als Challenge mit. Der Supplier muss in seiner Auftragsantwort diesen Wert als Response wiederholen.

Empfängt ein Client einen ungültigen Response-Wert, und ist ein expliziter Dialog geöffnet, so darf der Client keinen weiteren Auftrag (auch keinen Dialogendeauftrag) in diesem Dialog mehr schicken.

Der Client muss davon ausgehen, dass er nicht die Auftragsantwort erhalten hat, die der Supplier an ihn geschickt hat. Erhält der Supplier nicht innerhalb einer gewissen (von ihm frei zu bestimmenden) Zeitspanne einen weiteren Auftrag des Clients, so schließt er den Dialog. Er geht davon aus, dass der Client die zuletzt gesendete Auftragsantwort nicht oder nicht unversehrt erhalten hat (vgl. Abschnitt 3.3).

Außerhalb eines expliziten Dialogs liegt die Reaktion auf einen ungültigen Response-Wert im Ermessen des Clients.

Der Supplier schickt in jeder Auftragsantwort innerhalb eines expliziten Dialogs (beginnend mit der Antwort auf den Dialoginitialisierungsauftrag und endend mit der Antwort auf den Auftrag, der dem Dialogendeauftrag vorangeht) einen frei gewählten Wert als Challenge mit. Der Client muss in seinem jeweils folgenden Auftrag diesen Wert als Response wiederholen.

Empfängt der Supplier einen ungültigen Response-Wert, so hat er den entsprechenden Auftrag abzulehnen.

Damit dieses Verfahren wirksam ist, muss das jeweilige Challenge in jeder Nachricht gewechselt werden. Es darf sich nicht auf einfache Weise aus Informationen herleiten lassen, die unberechtigten Benutzern zugänglich sind (z.B. Zeitpunkt).

Neben dem Response-Wert des Clients prüft der Supplier die ConversationId und die SequenceNumber. Empfängt der Supplier einen Auftrag mit einer unerwarteten ConversationId oder Auftragsnummer, so hat er den Auftrag abzulehnen.

#### **4.6 Benutzerauthentisierung im Rahmen der expliziten Dialoginitialisierung**

Im Rahmen der expliziten Dialoginitialisierung prüft der Supplier, ob der Client im Besitz des privaten Schlüssels ist, der zum Chiffrierzertifikat gehört, das ihm der Client im Dialoginitialisierungsauftrag mitgeschickt hat.

Dazu verschlüsselt der Supplier seine Auftragsantwort mit dem öffentlichen Schlüssel aus diesem Zertifikat.

Wenn der Client nicht im Besitz des zugehörigen privaten Schlüssels ist, kann er die Dialoginitialisierungsantwort nicht entschlüsseln. Er kann somit insbesondere nicht an den Challenge-Wert in der Dialoginitialisierungsantwort gelangen. Da der Intermediär (wie in Abschnitt 4.5 beschrieben) jeden Folgeauftrag innerhalb des Dialogs ablehnen muss, der kein gültiges Tripel aus ConversationId, SequenceNumber und Response enthält, ist der Client nicht in der Lage, Folgeaufträge in diesem Dialog einzureichen.

Weiterhin benutzt der Supplier innerhalb eines expliziten Dialogs zum Verschlüsseln seiner Auftragsantworten an den Client stets den öffentlichen Schlüssel aus dem Chiffrierzertifikat, das ihm vom Client im Dialoginitialisierungsauftrag mitgeteilt worden ist.

So wird verhindert, dass ein Client sein Chiffrierzertifikat im Lauf des Dialogs wechselt.

#### **4.7 Prüfung der MessageId**

Die Prüfung der MessageId dient dazu, die Doppeleinreichung von Zustellungen zu vermeiden.



In Zustellungs- und Weiterleitungsaufträgen muss ein Client zusammen mit der Zustellung eine Messageld angeben. In Abwicklungsaufträgen ist die Angabe einer Messageld optional.

Wird in den genannten Aufträgen eine Messageld angegeben, so muss der Intermediär prüfen, ob

- er die Messageld erzeugt hat
- die Messageld schon einmal verwendet worden ist

Liefert eine dieser Prüfungen ein negatives Ergebnis, so hat der Intermediär den Auftrag abzulehnen, mit dem der Sender die Zustellung eingereicht hat.

Aus praktischen Gründen darf der Intermediär den Auftrag auch ablehnen, wenn die Erzeugung der angegebenen Messageld schon längere Zeit (in der Größenordnung eines Timeouts während eines expliziten Dialogs) zurückliegt.

Gibt ein Sender in einem Abwicklungsauftrag an den Intermediär keine Messageld an, so darf der Empfänger den resultierenden Bearbeitungsauftrag des Intermediärs mit dem Hinweis ablehnen, dass er die Annahme der Zustellung ohne Messageld verweigert. Der Sender erhält eine entsprechende Rückmeldung in der Abwicklungsantwort des Intermediärs.

#### 4.8 Quittungsmechanismen

OSCI-Transport kennt zwei explizite und zwei implizite Quittungsmechanismen.

Die expliziten Quittungsmechanismen sind:

- Sender und Empfänger einer Zustellung können jederzeit Kopien des zugehörigen Laufzettels beim Intermediär anfordern.
- In jeder Antwort auf einen Abholauftrag werden die Selektionskriterien, die der Client in seinem Auftrag angegeben hat, wiederholt.

Die impliziten Quittungsmechanismen sind:

- Durch das Senden einer Auftragsantwort mit einem Rückmeldecode, der Erfolg anzeigt, bestätigt ein Supplier, dass er den zugehörigen Auftrag empfangen hat und ausführen konnte.
- Durch das Senden eines Auftrags mit einem Response-Wert, der mit dem Challenge-Wert aus der vorangegangenen Auftragsantwort übereinstimmt, und einer SequenceNumber, die um 1 größer als die SequenceNumber des vorangegangenen Auftrags ist, bestätigt ein Client, dass er die Antwort auf den vorangegangenen Auftrag erhalten hat.

## 5. Reaktionsvorschriften und Rückmeldungen

Wenn ein Supplier eine Nachricht erhält, muss er die folgenden Schritte ausführen:

1. Schema- und andere syntaktische Prüfungen auf Nachrichtenebene vornehmen.
2. Gegebenenfalls die Auftragsdaten entschlüsseln.
3. Schema- und andere syntaktische Prüfungen auf Auftragsebene vornehmen.
4. ConversationId, SequenceNumber und Response ermitteln und prüfen.
5. Wenn die Auftragsantwort verschlüsselt werden soll und das Chiffrierzertifikat des Clients in diesem Dialog noch nicht geprüft worden ist, dieses ermitteln und prüfen.
6. Gegebenenfalls die Signatur prüfen.
7. Die übrigen im Auftrag enthaltenen Zertifikate prüfen, die in diesem Dialog noch nicht geprüft worden sind.
8. Den Auftrag ausführen.

Dabei muss er die Schritte 5 bis 7 nur dann zwingend ausführen, wenn er ein Intermediär ist.

Jeder dieser Schritte kann entweder erfolgreich durchgeführt werden, mit einer Warnung enden oder fehlschlagen. Wenn einer dieser Schritte fehlschlägt, muss der Supplier den Prozess an der entsprechenden Stelle abbrechen. Wenn Warnungen auftreten, kann der Supplier fortfahren.

In seiner Auftragsantwort an den Client übermittelt der Supplier Rückmeldungen, aus denen hervorgeht, welche Schritte er ausführen konnte, welche Warnungen dabei aufgetreten sind und welcher Schritt gegebenenfalls fehlgeschlagen ist.

Um eine automatisierte Auswertung der Rückmeldungen durch den Client zu ermöglichen, umfasst jede Rückmeldung einen vierstelligen numerischen Code, der nach dem folgenden Muster gebildet wird:

- Die erste Ziffer des Codes gibt die Art der Rückmeldung an. Dabei steht eine 0 für eine Erfolgsmeldung, eine 3 für eine Warnung und eine 9 für eine Fehlermeldung.
- Die zweite Ziffer bezeichnet den Schritt, für den die Rückmeldung gegeben wird (wenn der Schritt unbekannt ist, ist diese Ziffer 0).
- Die letzten beiden Ziffern des Codes bilden eine Ordnungsnummer.

Die Rückmeldungen werden in der Reihenfolge ihrer Erzeugung in die Auftragsantwort eingestellt.

Anhand der letzten Rückmeldung in der Auftragsantwort kann der Client also erkennen, ob der Supplier den Auftrag fehlerfrei ausführen konnte, oder ob Fehler aufgetreten sind:

- Handelt es sich bei der letzten Rückmeldung um eine Erfolgsmeldung oder Warnung, so wurde der Auftrag ausgeführt. In allen Schritten können dann höchstens Warnungen aufgetreten sein.
- Handelt es sich dagegen um eine Fehlermeldung, so ist ein Fehler aufgetreten.

Im Regelfall teilt der Supplier dem Client Rückmeldungen auf Auftragsebene mit (siehe Abschnitt 5.2). Nur wenn es nicht möglich ist, dem Client Rückmeldungen auf Auftragsebene zu geben, können Rückmeldungen auf Nachrichtenebene erfolgen (siehe Abschnitt 5.1).

Schlägt einer der Schritte 1 bis 5 fehl, so ist der Supplier nicht verpflichtet, eine Antwort zu senden.

Der Grund hierfür ist, dass es in diesen Fällen unmöglich ist, den Client, der den Auftrag gesendet hat, mit Mitteln von OSCI-Transport zu ermitteln.

Sofern allerdings das zugrundeliegende technische Protokoll dennoch ermöglicht, eine Rückmeldung zu schicken (dies ist beispielsweise für HTTP der Fall), sollte der Supplier dies tun. Der Client erhält dann eine Fehlermeldung auf Nachrichtenebene.

Um die Interoperabilität zu gewährleisten, dürfen Änderungen und Ergänzungen der Rückmeldungs-codes nur in Abstimmung mit dem Herausgeber erfolgen.

## 5.1 Fehlermeldungen auf Nachrichtenebene

Auf Nachrichtenebene erfolgen Rückmeldungen nur dann, wenn entweder der Client, der den Auftrag gesendet hat, nicht ermittelt werden konnte, oder ein interner Fehler beim Supplier aufgetreten ist, der es unmöglich macht, die Auftragsantwort zu signieren, zu verschlüsseln oder in eine OSCI-Nachricht einzufügen. Es handelt sich bei Rückmeldungen auf Nachrichtenebene also stets um Fehlermeldungen.

Rückmeldungen auf Nachrichtenebene werden in einem `soap:Fault`-Element im SOAP-Body übertragen (vgl. [SOAP 1, Abschnitt 4.4]). Das `soap:Fault`-Element enthält genau drei Elemente: `faultcode`, `faultstring` und `detail`.

Die möglichen Werte von `faultcode` sind in [SOAP 1, Abschnitt 4.4.1] beschrieben.

Von diesen Werten verwendet OSCI-Transport nur `soap:Client` und `soap:Server`.

Im Element `faultstring` wird eine Klartextbeschreibung des Fehlers gegeben. Die genaue Formulierung hängt von der verwendeten Landessprache ab und ist dem Supplier überlassen.

Das Element `detail` enthält ein Element `osci:Code`, das den numerischen Code umfasst, der den aufgetretenen Fehler bezeichnet.

Die Belegung von `faultcode` und `osci:Code` in den einzelnen Fehlersituationen geht aus der folgenden Tabelle hervor.

Schritt	Fehlersituation	faultcode	osci:Code
1	Empfangene Nachricht stellt keine gültige OSCI-Nachricht dar	<code>soap:Client</code>	9100
2	Supplier verfügt nicht über den privaten Schlüssel zum Chiffrierzertifikat auf Nachrichtenebene	<code>soap:Client</code>	9200
2	Chiffrierzertifikat auf Nachrichtenebene ist widerrufen	<code>soap:Client</code>	9201
2	Auftragsdaten konnten nicht entschlüsselt werden	<code>soap:Client</code>	9202
3	Auftragsdaten stellen keinen gültigen Auftrag dar	<code>soap:Client</code>	9300
4	ConversationId oder SequenceNumber oder Response nicht korrekt	<code>soap:Client</code>	9400

Schritt	Fehlersituation	faultcode	osci:Code
5	Chiffrierzertifikat des Clients ist nicht in den Auftragsdaten enthalten	soap:Client	9500
5	Signatur über das Chiffrierzertifikat des Clients ist fehlerhaft	soap:Client	9501
5	Chiffrierzertifikat des Clients ist widerrufen	soap:Client	9502
5	Interner Fehler des Suppliers beim Ermitteln oder Prüfen des Chiffrierzertifikats	soap:Server	9503
unbekannt	Interner Fehler beim Supplier	soap:Server	9000

Nur wenn das Element `faultcode` den Wert `soap:Server` hat, darf der Client versuchen, die Nachricht unverändert erneut zu senden.

## 5.2 Rückmeldungen auf Auftragsebene

Im Regelfall erfolgen Rückmeldungen des Suppliers an den Client auf Auftragsebene.

Die Rückmeldungen werden in einem speziellen Element `osci:Feedback` übertragen. Dessen innere Elemente werden wie folgt belegt:

- Das Element `osci:Code` enthält den numerischen Code, der die Rückmeldung bezeichnet.
- Das Element `osci:Text` enthält einen Klartext zur Rückmeldung. Die genaue Formulierung hängt von der verwendeten Landessprache ab und ist dem Supplier überlassen.

Die möglichen Werte von `osci:Code` sind in der folgenden Tabelle zusammengefasst.

Schritt	Situation	osci:Code
5	Chiffrierzertifikat des Clients ist zeitlich ungültig	3500
5	Prüfung des Chiffrierzertifikats des Clients konnte nicht vollständig durchgeführt werden	3501
6	Auftrag ist nicht signiert, obwohl dieser Supplier für diesen Auftragsstyp eine Signatur verlangt	9600
6	Signatur über Auftrag ist fehlerhaft	9601
6	Es sind nicht alle geforderten Elemente des Auftrags signiert	9602
6	Interner Fehler des Suppliers beim Prüfen der Signatur	9603
7	Signierzertifikat des Clients ist zeitlich ungültig	3700
7	Signatur über das Signierzertifikat des Clients ist fehlerhaft	9700
7	Signierzertifikat des Clients ist widerrufen	9701
7	Signierzertifikat eines Autors ist zeitlich ungültig	3701
7	Signatur über das Signierzertifikat eines Autors ist fehlerhaft	3702
7	Signierzertifikat eines Autors ist widerrufen	3703
7	Chiffrierzertifikat eines Autors ist zeitlich ungültig	3704

Schritt	Situation	osci:Code
7	Signatur über das Chiffrierzertifikat eines Autors ist fehlerhaft	9704
7	Chiffrierzertifikat eines Autors ist widerrufen	9705
7	Chiffrierzertifikat des Empfängers ist zeitlich ungültig	3705
7	Signatur über das Chiffrierzertifikat des Empfängers ist fehlerhaft	9706
7	Chiffrierzertifikat des Empfängers ist widerrufen	9707
7	Chiffrierzertifikat eines Lesers ist zeitlich ungültig	3706
7	Signatur über das Chiffrierzertifikat eines Lesers ist fehlerhaft	9708
7	Chiffrierzertifikat eines Lesers ist widerrufen	9709
7	Zertifikatsprüfung konnte nicht vollständig durchgeführt werden	3707
7	Interner Fehler des Suppliers bei der Zertifikatsprüfung	9710
8	Annahme der Zustellung ohne Messageld verweigert	9800
8	Illegales Messageld (Messageld wurde nicht von diesem Supplier erzeugt oder Messageld wurde schon einmal verwendet)	9801
8	Es liegen weitere Zustellungen für diesen Client vor	3800
8	Zu den im Auftrag angegebenen Kriterien liegen weitere Laufzettel vor	3801
8	Auftrag nicht innerhalb eines expliziten Dialogs gesendet, obwohl dies für diesen Auftragsstyp gefordert ist	9802
8	Zu den im Auftrag angegebenen Kriterien ist keine Zustellung vorhanden	9803
8	Zu den im Auftrag angegebenen Kriterien ist kein Laufzettel vorhanden	9804
8	Empfänger hat Zustellung des Senders nicht angenommen	9805
8	Empfänger hat Zustellung des Senders angenommen, seine Auftragsantwort enthält aber keine Zustellung an den Sender	9806
8	Signatur des Empfängers über die Annahme- bzw. Bearbeitungsantwort fehlerhaft	9807
8	Signatur des Empfängers über die Annahme- bzw. Bearbeitungsantwort fehlt	3802
8	Interner Fehler des Empfängers	9808
8	Interner Fehler des Intermediärs bei Weiterleitungs- oder Abwicklungsauftrag, Zustellung hat den Empfänger nicht erreicht	9809
8	Interner Fehler des Intermediärs bei Weiterleitungs- oder Abwicklungsauftrag, Zustellung hat den Empfänger erreicht	9810
8	Interner Fehler des Suppliers bei der Ausführung des Auftrags	9811
8	Auftrag ausgeführt, Dialog beendet	0800

<b>Schritt</b>	<b>Situation</b>	<b>osci:Code</b>
8	Auftrag ausgeführt, Dialog weiterhin geöffnet	0801

## 6. Nachrichtenaufbau

In diesem Abschnitt wird der Nachrichtenaufbau bei OSCI-Transport beschrieben.

Zunächst wird in Abschnitt 6.1 der Namespace angegeben, der durch diese Spezifikation definiert wird.

Abschnitt 6.2 stellt dann die inhaltliche Bedeutung der einzelnen Elemente dar, die in diesem Namespace definiert werden.

Abschnitt 6.3 enthält ein Schema, das in späteren Abschnitten weiter ausgeprägt wird.

Die Ausprägungen von XML-Signature und XML-Encryption für OSCI-Transport finden sich in den Abschnitten 6.4 und 6.5.

Der Aufbau der einzelnen Auftrags- und Auftragsantworttypen, die es in OSCI-Transport gibt, wird in Abschnitt 6.6 beschrieben.

Abschließend wird in Abschnitt 6.7 der Aufbau von OSCI-Nachrichten erläutert.

### 6.1 Versionierung

OSCI-Transport in der Version 1.2 ist der folgende Namespace zugeordnet:

```
http://www.osci.de/2002/04/osci
```

Anhand dieses Namespaces kann ein Supplier, Sender oder Leser die Version von OSCI-Transport ermitteln, die ein Client, Empfänger oder Autor verwendet.

Der oben angegebene Namespace ist an die Namespaces der Standards gekoppelt, auf denen OSCI-Transport basiert:

- <http://schemas.xmlsoap.org/soap/envelope/> (SOAP)
- <http://www.w3.org/2000/09/xmldsig#> (XML-Signature)
- <http://www.w3.org/2001/04/xmlenc#> (XML-Encryption)

Erscheint für einen dieser Standards eine Folgeversion mit einem geänderten Namespace, und soll OSCI-Transport diese Version unterstützen, so wird ein neuer Namespace veröffentlicht.

Ausdrücklich angestrebt ist, die konsolidierten Versionen von SOAP 1.2 und XML-Encryption zu unterstützen.

Weiterhin wird der Namespace gewechselt, wenn sich Rückmeldungs\_codes ändern.

### 6.2 Bedeutung der Elemente mit Namespace <http://www.osci.de/2002/04/osci>

Die folgende Tabelle gibt zu jedem Element mit Namespace

<http://www.osci.de/2002/04/osci> dessen inhaltliche Bedeutung an.

Der Übersichtlichkeit halber werden die Namen der Elemente ohne Namespace-Präfixe aufgeführt.

Name	Bedeutung
acceptDelivery	zeigt einen Annahmeftrag an
CertType	Attribut Type gibt den Typ des Zertifikats an: unknown (unbekannt) / advanced (fortgeschritten) / qualified (qualifiziert)

Name	Bedeutung
Challenge	Challenge (siehe Abschnitt 4.5)
CipherCertificateAddressee	Wenn im Auftrag oder in der Auftragsantwort eine Zustellung mitgeführt wird: Chiffrierzertifikat des Empfängers. Andernfalls: Chiffrierzertifikat des Benutzers als Adressat des Auftrags oder der Auftragsantwort
CipherCertificateIntermediary	Chiffrierzertifikat des Intermediärs.  Ein Intermediär muss dieses Element insbesondere bei jedem Auftrag angeben, auf den er eine verschlüsselte Antwort erwartet und der außerhalb eines expliziten Dialogs gesendet wird. Innerhalb eines expliziten Dialogs verwendet der Supplier zum Verschlüsseln seiner Auftragsantworten stets das Chiffrierzertifikat, das der Intermediär im Element <code>CipherCertificateIntermediary</code> seines Dialoginitialisierungsauftrags angegeben hat.
CipherCertificateOriginator	Wenn im Auftrag oder in der Auftragsantwort eine Zustellung mitgeführt wird: Chiffrierzertifikat des Senders. Andernfalls: Chiffrierzertifikat des Benutzers als Absender des Auftrags oder der Auftragsantwort.  Ein Benutzer muss dieses Element insbesondere bei jedem Auftrag angeben, der außerhalb eines expliziten Dialogs gesendet wird und auf den er eine verschlüsselte Antwort erwartet. Innerhalb eines expliziten Dialogs verwendet der Intermediär zum Verschlüsseln seiner Auftragsantworten stets das Chiffrierzertifikat, das der Benutzer im Element <code>CipherCertificateOriginator</code> seines Dialoginitialisierungsauftrags angegeben hat.
CipherCertificateOtherAuthor	Chiffrierzertifikat eines Autors, bei dem es sich nicht um den Sender der Zustellung handelt
CipherCertificateOtherReader	Chiffrierzertifikat eines Lesers, bei dem es sich nicht um den Empfänger der Zustellung handelt
ClientSignature	Signatur eines Clients über den Auftrag
Code	Rückmeldungscode (siehe Kapitel 5)
ControlBlock	Prüfblock. Enthält Informationen, anhand derer Client und Supplier die Dialogintegrität sicherstellen können
Content	nimmt die eigentlichen Inhaltsdaten auf. Kann insbesondere Verweise auf Attachments aufnehmen
ContentContainer	Inhaltsdatencontainer
ContentPackage	Umfasst beliebig viele Inhaltsdatencontainer. Einige von diesen können verschlüsselt sein und werden dann in Elementen <code>xenc:EncryptedData</code> abgelegt.
ContentReceiver	Attribut URI gibt den URL des Diensteanbieters an



Name	Bedeutung
ConversationId	ConversationId (siehe Abschnitt 3.3)
Creation	Zeitstempel <i>Einreichung beim Intermediär</i>
CRL	Enthält das Datum der CRL, die zur Online-Gültigkeitsprüfung verwendet worden ist
DesiredLanguages	Sprachen, die der Client für textuelle Rückmeldungen bevorzugt. Wenn möglich, sollte ein Supplier diese Angaben berücksichtigen
exitDialog	zeigt einen Dialogendeauftrag an
Feedback	Rückmeldung eines Suppliers an einen Client (siehe Abschnitt 5.2)
fetchDelivery	zeigt einen Zustellauftrag an
fetchProcessCard	zeigt einen Laufzettelabholauftrag an
forwardDelivery	zeigt einen Weiterleitungsauftrag an
Forwarding	Zeitstempel <i>Weiterleitung an Empfänger</i>
getMessageId	zeigt einen Messageld-Anforderungsauftrag an
initDialog	zeigt einen Dialoginitialisierungsauftrag an
Inspection	Eintrag im Prüfprotokoll für ein Zertifikat
InspectionReport	Prüfprotokoll für die Zertifikate, die in einem Auftrag enthalten sind
IntermediaryCertificates	Zertifikate des Intermediärs
LDAP	Zeigt an, dass die Online-Gültigkeitsprüfung mittels LDAP ausgeführt wurde
MathResult	Attribut <i>Result</i> gibt das Ergebnis der Prüfung der Signatur des Zertifikats
mediateDelivery	zeigt einen Abwicklungsauftrag an
MessageId	Messageld (siehe Abschnitt 3.4)
NonIntermediaryCertificates	Zertifikate von Benutzern, Autoren und Lesern
OCSP	Enthält die OCSP-Antwort im Rahmen der Online-Gültigkeitsprüfung für ein Zertifikat
OfflineResult	Attribut <i>Result</i> gibt das Ergebnis der Offline-Gültigkeitsprüfung für das Zertifikat an
OnlineResult	Attribut <i>Result</i> gibt das Ergebnis der Online-Gültigkeitsprüfung für das Zertifikat an
ProcessCard	Laufzettel, in dem der Intermediär den Weg einer Zustellung vom Sender zum Empfänger protokolliert

Name	Bedeutung
ProcessCardBundle	Laufzettelcontainer
processDelivery	zeigt einen Bearbeitungsauftrag an
QualityOfTimestamp	Gibt die gewünschte Qualität eines Zeitstempels auf dem Laufzettel an. Bezieht sich abhängig vom Wert des Attributs actor entweder auf den Zeitstempel <i>Zeitpunkt der Einreichung beim Intermediär</i> (actor="http://www.osci.de/2002/04/osci#creation") oder den Zeitstempel <i>Zeitpunkt der Empfangsbestätigung durch den Empfänger</i> (actor="http://www.osci.de/2002/04/osci#reception"). Die gewünschte Qualität wird im Attribut Quality angegeben. Mögliche Werte sind: http://www.osci.de/2002/04/osci#cryptographic (Kryptographischer Zeitstempel erforderlich) oder http://www.osci.de/2002/04/osci#plain (Systemzeit genügt)
Quantity	Das Attribut Limit dieses Elements gibt die maximale Anzahl an Laufzetteln an, die zurückgegeben werden soll
RecentModification	Zeitpunkt der letzten Änderung eines Laufzettels
Reception	Zeitstempel <i>Empfangsbestätigung durch Empfänger</i>
ReceptionOfDelivery	Zeitpunkt der Einreichung beim Intermediär
ReplyProcessCardBundle	Laufzettelcontainer zur Zustellung in der Abwicklungsantwort
RequestProcessCardBundle	Laufzettelcontainer zur Zustellung im Abwicklungsauftrag
Response	Response (siehe Abschnitt 4.5)
responseToAcceptDelivery	zeigt eine Annahmearantwort an
responseToExitDialog	zeigt eine Dialogendeantwort an
responseToFetchDelivery	zeigt eine Zustellungsabholantwort an
responseToFetchProcessCard	zeigt eine Laufzettelabholantwort an
responseToForwardDelivery	zeigt eine Weiterleitungsantwort an
responseToGetMessageId	zeigt eine Messageld-Anforderungsantwort an
responseToInitDialog	zeigt eine Dialoginitialisierungsantwort an
responseToMediateDelivery	zeigt eine Abwicklungsantwort an

Name	Bedeutung
responseToProcessDelivery	zeigt eine Bearbeitungsantwort an
responseToStoreDelivery	zeigt eine Zustellungsantwort an
SelectionRule	Auswahlkriterium für Zustellungen oder Laufzettel
SequenceNumber	SequenceNumber (siehe Abschnitt 3.3)
SignatureCertificateAddressee	Wenn im Auftrag oder in der Auftragsantwort eine Zustellung mitgeführt wird: Signierzertifikat des Empfängers. Andernfalls: Signierzertifikat des Benutzers als Adressat des Auftrags oder der Auftragsantwort
SignatureCertificateIntermediary	Signierzertifikat des Intermediärs.  Dieses Element ist in jedem signierten Auftrag und jeder signierten Auftragsantwort des Intermediärs vorhanden
SignatureCertificateOriginator	Wenn im Auftrag oder in der Auftragsantwort eine Zustellung mitgeführt wird: Signierzertifikat des Senders. Andernfalls: Signierzertifikat des Benutzers als Absender des Auftrags oder der Auftragsantwort.  Dieses Element ist in allen Aufträgen und Auftragsantworten vorhanden, die Zustellungen mit signierten Inhaltsdaten enthalten. Weiterhin findet sich dieses Element in jedem signierten Auftrag und jeder signierten Auftragsantwort eines Benutzers.
SignatureCertificateOtherAuthor	Signierzertifikat eines Autors, bei dem es sich nicht um den Sender einer Zustellung handelt
SignatureCertificateOtherReader	Signierzertifikat eines Lesers, bei dem es sich nicht um den Empfänger einer Zustellung handelt
storeDelivery	zeigt einen Zustellungsauftrag an
Subject	Betreff einer Zustellung
SupplierSignature	Signatur eines Suppliers über die Auftragsantwort
Text	Rückmeldungstext (siehe Abschnitt 5.2)
Timestamp	Zeitpunkt der Zertifikatsprüfung
X509IssuerName	Herausgeber des geprüften Zertifikats
X509SerialNumber	Seriennummer des geprüften Zertifikats

### 6.3 Globale Typdefinitionen

Für die globalen Typdefinitionen und abstrakten Typschablonen gilt das nachfolgende Schema:

```
<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
```

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:osci="http://www.osci.de/2002/04/osci"
attributeFormDefault="unqualified"
elementFormDefault="qualified">

<xsd:import namespace="http://www.w3.org/XML/1998/namespace"
  schemaLocation="http://www.w3.org/2001/xml.xsd" />
<xsd:import namespace="http://www.w3.org/2000/09/xmldsig#"
  schemaLocation="http://www.w3.org/TR/2001/CR-xmldsig-core-
20010419/xmldsig-core-schema.xsd" />
<xsd:import namespace="http://www.w3.org/2001/04/xmlenc#"
  schemaLocation="http://www.w3.org/TR/xmlenc-core/xenc-schema.xsd" />
<xsd:import namespace="http://schemas.xmlsoap.org/soap/envelope/"
  schemaLocation="http://schemas.xmlsoap.org/soap/envelope/" />

<xsd:annotation>
  <xsd:documentation xml:lang="de">
    OSCI 1.2 - Allgemeine Typen und Strukturen
    $RCSfile: order.xsd,v $, $Revision: 1.20 $
  </xsd:documentation>
</xsd:annotation>

<!-- ### global simple types ### -->

<xsd:simpleType name="LanguagesListType">
  <xsd:list itemType="xsd:language" />
</xsd:simpleType>

<xsd:simpleType name="MessageIdType">
  <xsd:restriction base="xsd:base64Binary">
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="Number">
  <xsd:restriction base="xsd:NMTOKEN">
    <xsd:pattern value="\d+" />
  </xsd:restriction>
</xsd:simpleType>

<!-- ### global complex types and templates ### -->

<!-- ### common types ### -->

<xsd:complexType name="CertificateType">
  <xsd:complexContent>
    <xsd:restriction base="ds:KeyInfoType">
      <xsd:sequence>
```

```
        <xsd:choice>
          <xsd:element ref="ds:X509Data" />
        </xsd:choice>
      </xsd:sequence>
      <xsd:attribute name="Id" type="xsd:ID" use="optional" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="CertTypeType">
  <xsd:attribute name="Type" use="required">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="advanced" />
        <xsd:enumeration value="qualified" />
        <xsd:enumeration value="unknown" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="FeedbackEntryType">
  <xsd:sequence>
    <xsd:element name="Code" type="osci:Number" />
    <xsd:element name="Text" type="xsd:string" />
  </xsd:sequence>
  <xsd:attribute ref="xml:lang" use="optional" />
</xsd:complexType>

<xsd:complexType name="FeedbackType">
  <xsd:sequence>
    <xsd:element name="Entry" type="osci:FeedbackEntryType"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="InspectionType">
  <xsd:sequence>
    <xsd:element name="Timestamp" type="osci:TimestampType" />
    <xsd:element name="X509IssuerName" type="xsd:string" />
    <xsd:element name="X509SerialNumber" type="xsd:integer" />
    <xsd:element name="CertType" type="osci:CertTypeType" />
    <xsd:element name="MathResult" type="osci:MathResultType" />
    <xsd:element name="OfflineResult" type="osci:OfflineResultType" />
    <xsd:element name="OnlineResult" type="osci:OnlineResultType"
      minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="InspectionReportType">
```

```
<xsd:sequence>
  <xsd:element name="Inspection" type="osci:InspectionType"
    minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="MathResultType">
  <xsd:attribute name="Result" use="required">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="ok" />
        <xsd:enumeration value="corrupted" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="OfflineResultType">
  <xsd:attribute name="Result" use="required">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="valid" />
        <xsd:enumeration value="invalid" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="OnlineResultType">
  <xsd:choice>
    <xsd:element name="OSCP" type="xsd:base64Binary" />
    <xsd:element name="CRL" type="xsd:dateTime" />
    <xsd:element name="LDAP">
      <xsd:complexType />
    </xsd:element>
  </xsd:choice>
  <xsd:attribute name="Result" use="required">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="ok" />
        <xsd:enumeration value="revoked" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="TimestampType">
  <xsd:choice>
    <xsd:element name="Plain" type="xsd:dateTime" />
    <xsd:element name="Cryptographic">
```

```

    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base="xsd:base64Binary">
          <xsd:attribute name="Algorith" type="xsd:anyURI" />
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="ContentType">
  <xsd:sequence>
    <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##any" />
</xsd:complexType>

<xsd:complexType name="ContentContainerType">
  <xsd:sequence>
    <xsd:element ref="ds:Signature"
      minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="Content" type="osci:ContentType" minOccurs="0" />
    <xsd:element ref="xenc:EncryptedData"
      minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>

<!-- ### common templates ### -->

<xsd:complexType name="ProcessCardBundleTemplate" abstract="true">
  <xsd:sequence>
    <xsd:element name="MessageId" type="osci:MessageIdType" />
    <xsd:element name="ProcessCard" type="osci:ProcessCardTemplate" />
    <xsd:element name="InspectionReport"
      type="osci:InspectionReportType" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ProcessCardTemplate" abstract="true">
  <xsd:sequence>
    <xsd:element name="Creation" type="osci:TimestampType"
      minOccurs="0" />
    <xsd:element name="Forwarding" type="osci:TimestampType"
      minOccurs="0" />
    <xsd:element name="Reception" type="osci:TimestampType"
      minOccurs="0" />
    <xsd:element name="Subject" type="xsd:string"
      minOccurs="0" />
  </xsd:sequence>

```

```
<xsd:attribute name="RecentModification" type="xsd:dateTime"
  use="required"/>
</xsd:complexType>

<!-- ### SOAP header and body block types ### -->

<xsd:complexType name="SignatureBlockType">
  <xsd:complexContent>
    <xsd:extension base="osci:DefaultHeaderBlockTemplate">
      <xsd:sequence>
        <xsd:element ref="ds:Signature" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="DesiredLanguagesType">
  <xsd:complexContent>
    <xsd:extension base="osci:DefaultHeaderBlockTemplate">
      <xsd:attribute name="LanguagesList" use="required">
        <xsd:simpleType>
          <xsd:restriction base="osci:LanguagesListType">
            <xsd:minLength value="1" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="QualityOfTimestampType">
  <xsd:complexContent>
    <xsd:extension base="osci:DefaultHeaderBlockTemplate">
      <xsd:attribute name="Service" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="creation" />
            <xsd:enumeration value="reception" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="Quality" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="plain" />
            <xsd:enumeration value="cryptographic" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```



```

    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="ContentPackageType">
    <xsd:complexContent>
      <xsd:extension base="osci:DefaultBodyBlockTemplate">
        <xsd:sequence>
          <xsd:element ref="xenc:EncryptedData"
            minOccurs="0" maxOccurs="unbounded" />
          <xsd:element name="ContentContainer"
            type="osci:ContentContainerType"
            minOccurs="0" maxOccurs="unbounded" />
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <!-- ### SOAP header and body templates ### -->

  <xsd:complexType name="GeneralHeaderBlockTemplate" abstract="true">
    <xsd:attribute name="Id" type="xsd:ID" use="required" />
    <xsd:attribute ref="soap:mustUnderstand" fixed="1" use="required" />
  </xsd:complexType>

  <xsd:complexType name="DefaultHeaderBlockTemplate" abstract="true">
    <xsd:complexContent>
      <xsd:extension base="osci:GeneralHeaderBlockTemplate">
        <xsd:attribute ref="soap:actor"
          fixed="http://schemas.xmlsoap.org/soap/actor/next"
          use="required" />
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="GeneralBodyBlockTemplate" abstract="true">
  </xsd:complexType>

  <xsd:complexType name="DefaultBodyBlockTemplate" abstract="true">
    <xsd:complexContent>
      <xsd:extension base="osci:DefaultBodyBlockTemplate">
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="ControlBlockTemplate" abstract="true">
    <xsd:complexContent>
      <xsd:extension base="osci:DefaultHeaderBlockTemplate">
        <xsd:sequence>
          <xsd:element name="Response" type="xsd:string" minOccurs="0" />
          <xsd:element name="Challenge" type="xsd:string" minOccurs="0" />
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

```

```

    </xsd:sequence>
    <xsd:attribute name="ConversationId" type="osci:Number"
      use="optional" />
    <xsd:attribute name="SequenceNumber" type="osci:Number"
      use="optional" />
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="IntermediaryCertificatesTemplate" abstract="true">
  <xsd:complexContent>
    <xsd:extension base="osci:GeneralHeaderBlockTemplate">
      <xsd:sequence>
        <xsd:element name="CipherCertificateIntermediary"
          type="osci:CertificateType" minOccurs="0" />
        <xsd:element name="SignatureCertificateIntermediary"
          type="osci:CertificateType" minOccurs="0" />
      </xsd:sequence>
      <xsd:attribute ref="soap:actor"
        fixed="http://www.w3.org/2001/12/soap-envelope/actor/none"
        use="required" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="NonIntermediaryCertificatesTemplate"
  abstract="true">
  <xsd:complexContent>
    <xsd:extension base="osci:GeneralHeaderBlockTemplate">
      <xsd:sequence>
        <xsd:element name="CipherCertificateOriginator"
          type="osci:CertificateType" minOccurs="0" />
        <xsd:element name="CipherCertificateOtherAuthor"
          type="osci:CertificateType"
          minOccurs="0" maxOccurs="unbounded" />
        <xsd:element name="CipherCertificateAddressee"
          type="osci:CertificateType" minOccurs="0" />
        <xsd:element name="CipherCertificateOtherReader"
          type="osci:CertificateType"
          minOccurs="0" maxOccurs="unbounded" />
        <xsd:element name="SignatureCertificateOriginator"
          type="osci:CertificateType" minOccurs="0" />
        <xsd:element name="SignatureCertificateOtherAuthor"
          type="osci:CertificateType"
          minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
      <xsd:attribute ref="soap:actor"
        fixed="http://www.w3.org/2001/12/soap-envelope/actor/none"
        use="required" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

    </xsd:complexContent>
  </xsd:complexType>

</xsd:schema>

```

## 6.4 Ausprägung von XML-Signature

Für die Verwendung von XML-Signature bei OSCI-Transport gelten die folgenden Restriktionen:

```

<xsd:schema targetNamespace="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.w3.org/2001/04/xmlenc#"
    schemaLocation="oscienc.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Einschraenkung von XML-Signature Auftragsebene
      $RCSfile: oscisig.xsd,v $, $Revision: 1.7 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### redefinitions ### -->

  <xsd:redefine schemaLocation="http://www.w3.org/TR/2001/CR-xmldsig-core-
20010419/xmldsig-core-schema.xsd">

    <xsd:complexType name="KeyInfoType">
      <xsd:complexContent>
        <xsd:restriction base="ds:KeyInfoType">
          <xsd:choice>
            <xsd:element ref="xenc:EncryptedKey" />
            <xsd:element ref="ds:RetrievalMethod" />
            <xsd:element ref="ds:X509Data" />
          </xsd:choice>
          <xsd:attribute name="Id" type="xsd:ID" use="optional" />
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="ReferenceType">
      <xsd:complexContent>
        <xsd:restriction base="ds:ReferenceType">
          <xsd:sequence>
            <xsd:element ref="ds:Transforms" />

```

```
        <xsd:element ref="ds:DigestMethod" />
        <xsd:element ref="ds:DigestValue" />
    </xsd:sequence>
    <xsd:attribute name="URI" type="xsd:anyURI" use="optional" />
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="SignatureType">
    <xsd:complexContent>
        <xsd:restriction base="ds:SignatureType">
            <xsd:sequence>
                <xsd:element ref="ds:SignedInfo" />
                <xsd:element ref="ds:SignatureValue" />
                <xsd:element ref="ds:KeyInfo" />
                <xsd:element ref="ds:Object"
                    minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="SignatureValueType">
    <xsd:simpleContent>
        <xsd:restriction base="ds:SignatureValueType">
            <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
        </xsd:restriction>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="RetrievalMethodType">
    <xsd:complexContent>
        <xsd:restriction base="ds:RetrievalMethodType">
            <xsd:attribute name="URI" type="xsd:anyURI" use="required" />
            <xsd:attribute name="Type">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:anyURI">
                        <xsd:enumeration
                            value="http://www.w3.org/2000/09/xmldsig#X509Data" />
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="X509DataType">
    <xsd:complexContent>
        <xsd:restriction base="ds:X509DataType">
            <xsd:sequence maxOccurs="1">
```

```
        <xsd:element name="X509Certificate" type="xsd:base64Binary" />
    </xsd:sequence>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="CanonicalizationMethodType">
  <xsd:complexContent>
    <xsd:restriction base="ds:CanonicalizationMethodType">
      <xsd:attribute name="Algorithm" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:anyURI">
            <xsd:enumeration
              value="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="DigestMethodType">
  <xsd:complexContent>
    <xsd:restriction base="ds:DigestMethodType">
      <xsd:attribute name="Algorithm" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:anyURI">
            <xsd:enumeration
              value="http://www.w3.org/2000/09/xmlsig#sha1" />
            <xsd:enumeration
              value="http://www.osci.de/2002/04/osci#ripemd160" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="SignatureMethodType">
  <xsd:complexContent>
    <xsd:restriction base="ds:SignatureMethodType">
      <xsd:attribute name="Algorithm" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:anyURI">
            <xsd:enumeration
              value="http://www.w3.org/2000/09/xmlsig#rsa-sha1" />
            <xsd:enumeration
              value="http://www.osci.de/2002/04/osci#rsa-ripemd160" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
```

```

        </xsd:attribute>
        </xsd:restriction>
        </xsd:complexContent>
    </xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

Dieses Schema lässt insbesondere zu, eine Signatur vom Typ XAdES (vgl. [XADES]) als Signatur für Inhaltsdaten zu verwenden.

Auf Geschäftsvorfallebene dürfen sich die Signaturen in den Elementen `ds:Signature` auf beliebig Elemente im umschließenden Inhaltsdatencontainer beziehen.

Die Signatur eines Auftrags bezieht sich auf alle Elemente, die parallel zu `osci:ClientSignature` im `soap:Header` liegen, sowie auf das Element `soap:Body`.

Die Signatur einer Auftragsantwort bezieht sich auf alle Elemente, die parallel zu `osci:SupplierSignature` im `soap:Header` liegen, sowie auf das Element `soap:Body`.

Das Element `ds:KeyInfo` innerhalb des Elements `ds:Signature` verweist auf ein Zertifikat innerhalb des selben Auftrags bzw. der selben Auftragsantwort.

## 6.5 Ausprägung von XML-Encryption

Für die Verwendung von XML-Encryption bei OSCI-Transport gelten die folgenden Restriktionen:

```

<xsd:schema targetNamespace="http://www.w3.org/2001/04/xmlenc#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  elementFormDefault="qualified">

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Einschraenkung von XML Encryption Auftragsebene
      $RCSfile: oscienc.xsd,v $, $Revision: 1.6 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### redefinitions ### -->

  <xsd:redefine
    schemaLocation="http://www.w3.org/TR/xmlenc-core/xenc-schema.xsd">

    <xsd:complexType name="EncryptionMethodType">
      <xsd:complexContent>
        <xsd:restriction base="xenc:EncryptionMethodType">
          <xsd:sequence>
            <xsd:element name="KeySize" minOccurs="0"

```

```

        type="xenc:KeySizeType" />
    </xsd:sequence>
    <xsd:attribute name="Algorithm" use="required">
        <xsd:simpleType>
            <xsd:restriction base="xsd:anyURI">
                <xsd:enumeration
                    value="http://www.w3.org/2001/04/xmlenc#tripleledes-cbc" />
                <xsd:enumeration
                    value="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
                <xsd:enumeration
                    value="http://www.w3.org/2001/04/xmlenc#aes192-cbc" />
                <xsd:enumeration
                    value="http://www.w3.org/2001/04/xmlenc#aes256-cbc" />
                <xsd:enumeration
                    value="http://www.w3.org/2001/04/xmlenc#rsa-1_5" />
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="CipherReferenceType">
    <xsd:complexContent>
        <xsd:restriction base="xenc:CipherReferenceType">
            <xsd:attribute name="URI" type="xsd:anyURI" use="required"/>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="EncryptedDataType">
    <xsd:complexContent>
        <xsd:restriction base="xenc:EncryptedDataType">
            <xsd:sequence>
                <xsd:element name="EncryptionMethod"
                    type="xenc:EncryptionMethodType" minOccurs="0" />
                <xsd:element ref="ds:KeyInfo" minOccurs="0" />
                <xsd:element ref="xenc:CipherData" minOccurs="1" />
            </xsd:sequence>
            <xsd:attribute name="MimeType" type="xsd:string" use="optional"
/>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="EncryptedKeyType">
    <xsd:complexContent>
        <xsd:restriction base="xenc:EncryptedKeyType">
            <xsd:sequence>
                <xsd:element name="EncryptionMethod"

```

```

        type="xenc:EncryptionMethodType" minOccurs="1" />
        <xsd:element ref="ds:KeyInfo" minOccurs="1" />
        <xsd:element ref="xenc:CipherData" minOccurs="1" />
    </xsd:sequence>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

Bei der Verschlüsselung von Inhaltsdaten dürfen sich die Verschlüsselungsinformationen in den Elementen `xenc:EncryptedData` auf beliebige Inhaltsdatencontainer und Attachments beziehen, die sich in der selben Zustellung wie das Element `xenc:EncryptedData` befinden.

## 6.6 Aufträge und Auftragsantworten

Clients können bei OSCI-Transport Aufträge der folgenden Typen senden:

- *Dialoginitialisierungsauftrag*: Ein Client eröffnet einen expliziten Dialog mit einem Supplier.
- *Dialogendeauftrag*: Ein Client beendet einen expliziten Dialog mit einem Supplier.
- *MessageId-Anforderungsauftrag*: Ein Benutzer fordert eine MessageId bei einem Intermediär an.
- *Zustellungsauftrag*: Ein Benutzer sendet eine Zustellung an einen Intermediär, damit dieser sie zur Abholung durch einen anderen Benutzer bereithält.
- *Zustellungsabholauftrag*: Ein Benutzer holt bei einem Intermediär eine Zustellung ab, die ein anderer Benutzer zuvor mittels eines Zustellungsauftrags eingereicht hat.
- *Laufzettelabholauftrag*: Ein Benutzer holt einen oder mehrere Laufzettel bei einem Intermediär ab. In einem Laufzettel wird der Weg einer Zustellung von einem Sender zu einem Empfänger protokolliert.
- *Weiterleitungsauftrag*: Ein Benutzer sendet eine Zustellung an einen Intermediär, damit dieser sie an einen Dienstanbieter weiterleitet.
- *Annahmearauftrag*: Ein Intermediär sendet eine Zustellung an einen Dienstanbieter.
- *Abwicklungsauftrag*: Ein Benutzer sendet eine Zustellung an einen Intermediär, damit dieser sie an einen Dienstanbieter weiterleitet. Der Benutzer erwartet in der Antwort des Intermediärs eine Zustellung des Dienstanbieters.
- *Bearbeitungsauftrag*: Ein Intermediär fordert einen Dienstanbieter auf, eine Zustellung eines Benutzers zu bearbeiten und als Reaktion eine weitere Zustellung zu liefern.

Ein Supplier reagiert auf jeden Auftrag eines Clients mit einem entsprechenden Auftragsantworttyp.

### 6.6.1 Dialoginitialisierungsauftrag

Ein Dialoginitialisierungsauftrag wird als SOAP-Envelope realisiert.

Das folgende Beispiel erläutert den Aufbau eines Dialoginitialisierungsauftrags:



```

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

<!-- XML-Beispiel OSCI 1.2 Dialoginitialisierungsauftrag -->
<!-- $RCSfile: InitDialog.xml,v $, $Revision: 1.8 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    soapInitDialog.xsd
    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">
  <soap:Header>
    <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
      <osci:Challenge>AAQCAEMmCZtuMFQxDSa</osci:Challenge>
    </osci:ControlBlock>
    <osci:DesiredLanguages Id="XREF-0002" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
      LanguagesList="de en-US" />
    <osci:NonIntermediaryCertificates Id="XREF-0010"
      soap:mustUnderstand="1"
      soap:actor="http://www.w3.org/2001/12/soap-envelope/actor/none">
      <osci:CipherCertificateOriginator>
        <ds:X509Data>
          <ds:X509Certificate>
            Chiffrierzertifikat/Benutzer/Base64codiert
          </ds:X509Certificate>
        </ds:X509Data>
      </osci:CipherCertificateOriginator>
    </osci:NonIntermediaryCertificates>
  </soap:Header>
  <soap:Body Id="XREF-0100">
    <osci:initDialog />
  </soap:Body>
</soap:Envelope>

```

Ein Dialoginitialisierungsauftrag genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Dialoginitialisierungsaufträge:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"

```

```

        elementFormDefault="qualified">

<xsd:import namespace="http://www.osci.de/2002/04/osci"
  schemaLocation="InitDialog.xsd" />

<xsd:annotation>
  <xsd:documentation xml:lang="de">
    OSCI 1.2 - Dialoginitialisierungsauftrag SOAP-Envelope
    $RCSfile: soapInitDialog.xsd,v $, $Revision: 1.6 $
  </xsd:documentation>
</xsd:annotation>

<!-- ### restrictions ### -->

<xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

  <!-- ### Envelope, Header und Body ### -->

  <xsd:complexType name="Envelope" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Envelope">
        <xsd:sequence>
          <xsd:element ref="soap:Header" minOccurs="1" />
          <xsd:element ref="soap:Body" minOccurs="1" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Header" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Header">
        <xsd:sequence>
          <xsd:element ref="osci:ControlBlock" />
          <xsd:element ref="osci:ClientSignature" minOccurs="0" />
          <xsd:element ref="osci:DesiredLanguages" />
          <xsd:element ref="osci:NonIntermediaryCertificates" />
          <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
          processContents="lax" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Body" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Body">
        <xsd:sequence>
          <xsd:element ref="osci:initDialog" />

```

```

        </xsd:sequence>
        <xsd:attribute name="Id" type="xsd:ID" use="required" />
    </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- Schema für Dialoginitialisierungsaufträge:

```

<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Dialoginitialisierungsauftrag
      $RCSfile: InitDialog.xsd,v $, $Revision: 1.5 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### derived types ### -->

  <xsd:complexType name="ControlBlockType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ControlBlockTemplate">
        <xsd:sequence>
          <xsd:element name="Challenge" type="xsd:string" minOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="ConversationId" type="osci:Number"
          use="prohibited" />
        <xsd:attribute name="SequenceNumber" type="osci:Number"
          use="prohibited" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="NonIntermediaryCertificatesType">
    <xsd:complexContent>
      <xsd:restriction base="osci:NonIntermediaryCertificatesTemplate">
        <xsd:sequence>

```

```

        <xsd:element name="CipherCertificateOriginator"
            type="osci:CertificateType" />
        <xsd:element name="SignatureCertificateOriginator"
            type="osci:CertificateType" minOccurs="0" />
    </xsd:sequence>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="initDialogType">
    <xsd:complexContent>
        <xsd:extension base="osci:DefaultBodyBlockTemplate">
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="ClientSignature" type="osci:SignatureBlockType" />

<xsd:element name="DesiredLanguages" type="osci:DesiredLanguagesType" />

<xsd:element name="NonIntermediaryCertificates"
    type="osci:NonIntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

<xsd:element name="initDialog" type="osci:initDialogType" />

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)

## 6.6.2 Dialoginitialisierungsantwort

Eine Dialoginitialisierungsantwort wird als SOAP-Envelope realisiert.

Das folgende Beispiel erläutert den Aufbau einer Dialoginitialisierungsantwort:

```

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

<!-- XML-Beispiel OSCI 1.2 Dialoginitialisierungsantwort -->
<!-- $RCSfile: ResponseToInitDialog.xml,v $, $Revision: 1.8 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"

```

```

xmlns:osci="http://www.osci.de/2002/04/osci"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
                    soapResponseToInitDialog.xsd
                    http://www.w3.org/2000/09/xmlnsig# oscisig.xsd
                    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">
<soap:Header>
  <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
    soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
    ConversationId="87634586123">
    <osci:Response>AAQCAEMmCZtuMFQxDSa</osci:Response>
    <osci:Challenge>345nchfcfoqc5dfg</osci:Challenge>
  </osci:ControlBlock>
</soap:Header>
<soap:Body Id="XREF-0100">
  <osci:responseToInitDialog>
    <osci:Feedback>
      <osci:Entry xml:lang="de">
        <osci:Code>0801</osci:Code>
        <osci:Text>Auftrag ausgeführt, Dialog weiterhin
geöffnet</osci:Text>
      </osci:Entry>
    </osci:Feedback>
  </osci:responseToInitDialog>
</soap:Body>
</soap:Envelope>

```

Eine Dialoginitialisierungsantwort genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Dialoginitialisierungsantworten:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmlnsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.osci.de/2002/04/osci"
    schemaLocation="ResponseToInitDialog.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Dialoginitialisierungsantwort SOAP-Envelope
      $RCSfile: soapResponseToInitDialog.xsd,v $, $Revision: 1.6 $
    </xsd:documentation>
  </xsd:annotation>

```

```

<!-- ### restrictions ### -->

<xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

  <!-- ### Envelope, Header und Body ### -->

  <xsd:complexType name="Envelope" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Envelope">
        <xsd:sequence>
          <xsd:element ref="soap:Header" minOccurs="1" />
          <xsd:element ref="soap:Body" minOccurs="1" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Header" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Header">
        <xsd:sequence>
          <xsd:element ref="osci:ControlBlock" />
          <xsd:element ref="osci:SupplierSignature" minOccurs="0" />
          <xsd:element ref="osci:IntermediaryCertificates"
            minOccurs="0" />
          <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
            processContents="lax" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Body" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Body">
        <xsd:sequence>
          <xsd:element ref="osci:responseToInitDialog" />
        </xsd:sequence>
        <xsd:attribute name="Id" type="xsd:ID" use="required" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- Schema für Dialoginitialisierungsantworten:

```
<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Dialoginitialisierungsantwort
      $RCSfile: ResponseToInitDialog.xsd,v $, $Revision: 1.5 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### derived types ### -->

  <xsd:complexType name="ControlBlockType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ControlBlockTemplate">
        <xsd:sequence>
          <xsd:element name="Response" type="xsd:string" minOccurs="1" />
          <xsd:element name="Challenge" type="xsd:string" minOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="ConversationId" type="osci:Number"
          use="required" />
        <xsd:attribute name="SequenceNumber" type="osci:Number"
          use="prohibited" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="IntermediaryCertificatesType">
    <xsd:complexContent>
      <xsd:restriction base="osci:IntermediaryCertificatesTemplate">
        <xsd:sequence>
          <xsd:element name="SignatureCertificateIntermediary"
            type="osci:CertificateType" minOccurs="0" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="responseToInitDialogType">
    <xsd:complexContent>
      <xsd:extension base="osci:DefaultBodyBlockTemplate">
        <xsd:sequence>
```

```

        <xsd:element name="Feedback" type="osci:FeedbackType" />
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="SupplierSignature" type="osci:SignatureBlockType" />

<xsd:element name="IntermediaryCertificates"
    type="osci:IntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

<xsd:element name="responseToInitDialog"
    type="osci:responseToInitDialogType" />

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)

### 6.6.3 Dialogendeauftrag

Ein Dialogendeauftrag wird als SOAP-Envelope realisiert.

Das folgende Beispiel erläutert den Aufbau eines Dialogendeauftrags:

```

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

<!-- XML-Beispiel OSCI 1.2 Dialogendeauftrag -->
<!-- $RCSfile: ExitDialog.xml,v $, $Revision: 1.9 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:osci="http://www.osci.de/2002/04/osci"
    xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
        soapExitDialog.xsd
        http://www.w3.org/2000/09/xmldsig# oscisig.xsd
        http://www.w3.org/2001/04/xmlenc# oscienc.xsd">
    <soap:Header>
        <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
            soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
            ConversationId="87634586123" SequenceNumber="5">
            <osci:Response>AAQCAEMmCZtuMFQxDSa</osci:Response>

```



```

    <osci:Challenge>345nchfcfoqc5dfg</osci:Challenge>
  </osci:ControlBlock>
  <osci:DesiredLanguages Id="XREF-0002" soap:mustUnderstand="1"
    soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
    LanguagesList="de en-US" />
</soap:Header>
<soap:Body Id="XREF-0100">
  <osci:exitDialog />
</soap:Body>
</soap:Envelope>

```

Ein Dialogendeauftrag genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Dialogendeaufträge:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.osci.de/2002/04/osci"
    schemaLocation="ExitDialog.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Dialogendeauftrag SOAP-Envelope
      $RCSfile: soapExitDialog.xsd,v $, $Revision: 1.6 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### restrictions ### -->

  <xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

    <!-- ### Envelope, Header und Body ### -->

    <xsd:complexType name="Envelope" >
      <xsd:complexContent>
        <xsd:restriction base="soap:Envelope">
          <xsd:sequence>
            <xsd:element ref="soap:Header" minOccurs="1" />
            <xsd:element ref="soap:Body" minOccurs="1" />
          </xsd:sequence>
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>

```

```

<xsd:complexType name="Header" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Header">
      <xsd:sequence>
        <xsd:element ref="osci:ControlBlock" />
        <xsd:element ref="osci:ClientSignature" minOccurs="0" />
        <xsd:element ref="osci:DesiredLanguages" />
        <xsd:element ref="osci:NonIntermediaryCertificates"
          minOccurs="0"/>
        <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
          processContents="lax" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="Body" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Body">
      <xsd:sequence>
        <xsd:element ref="osci:exitDialog" />
      </xsd:sequence>
      <xsd:attribute name="Id" type="xsd:ID" use="required" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- Schema für Dialogendeaufträge:

```

<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Dialogendeauftrag
      $RCSfile: ExitDialog.xsd,v $, $Revision: 1.5 $
    </xsd:documentation>
  </xsd:annotation>

```

```
</xsd:documentation>
</xsd:annotation>

<!-- ### derived types ### -->

<xsd:complexType name="ControlBlockType">
  <xsd:complexContent>
    <xsd:restriction base="osci:ControlBlockTemplate">
      <xsd:sequence>
        <xsd:element name="Response" type="xsd:string" minOccurs="1" />
        <xsd:element name="Challenge" type="xsd:string" minOccurs="1" />
      </xsd:sequence>
      <xsd:attribute name="ConversationId" type="osci:Number"
        use="required" />
      <xsd:attribute name="SequenceNumber" type="osci:Number"
        use="required" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="NonIntermediaryCertificatesType">
  <xsd:complexContent>
    <xsd:restriction base="osci:NonIntermediaryCertificatesTemplate">
      <xsd:sequence>
        <xsd:element name="SignatureCertificateOriginator"
          type="osci:CertificateType" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="exitDialogType">
  <xsd:complexContent>
    <xsd:extension base="osci:DefaultBodyBlockTemplate">
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="ClientSignature" type="osci:SignatureBlockType" />

<xsd:element name="DesiredLanguages" type="osci:DesiredLanguagesType" />

<xsd:element name="NonIntermediaryCertificates"
  type="osci:NonIntermediaryCertificatesType" />
```

```

<!-- ### global elements SOAP-Body ### -->

<xsd:element name="exitDialog" type="osci:exitDialogType" />

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)

#### 6.6.4 Dialogendeantwort

Eine Dialogendeantwort wird als SOAP-Envelope realisiert.

Das folgende Beispiel erläutert den Aufbau einer Dialogendeantwort:

```

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

<!-- XML-Beispiel OSCI 1.2 Dialogendeantwort -->
<!-- $RCSfile: ResponseToExitDialog.xml,v $, $Revision: 1.8 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    soapResponseToExitDialog.xsd
    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">

  <soap:Header>
    <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
      ConversationId="87634586123" SequenceNumber="6">
      <osci:Response>345nchfcfoqc5dfg</osci:Response>
    </osci:ControlBlock>
  </soap:Header>
  <soap:Body Id="XREF-0100">
    <osci:responseToExitDialog>
      <osci:Feedback>
        <osci:Entry xml:lang="de">
          <osci:Code>0800</osci:Code>
          <osci:Text>Auftrag ausgeführt, Dialog beendet</osci:Text>
        </osci:Entry>
      </osci:Feedback>
    </osci:responseToExitDialog>
  </soap:Body>
</soap:Envelope>

```

Eine Dialogendeantwort genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Dialogendeantworten:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.osci.de/2002/04/osci"
    schemaLocation="ResponseToExitDialog.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Dialogendeantwort SOAP-Envelope
      $RCSfile: soapResponseToExitDialog.xsd,v $, $Revision: 1.6 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### restrictions ### -->

  <xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

    <!-- ### Envelope, Header und Body ### -->

    <xsd:complexType name="Envelope" >
      <xsd:complexContent>
        <xsd:restriction base="soap:Envelope">
          <xsd:sequence>
            <xsd:element ref="soap:Header" minOccurs="1" />
            <xsd:element ref="soap:Body" minOccurs="1" />
          </xsd:sequence>
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="Header" >
      <xsd:complexContent>
        <xsd:restriction base="soap:Header">
          <xsd:sequence>
            <xsd:element ref="osci:ControlBlock" />
            <xsd:element ref="osci:SupplierSignature" minOccurs="0" />
            <xsd:element ref="osci:IntermediaryCertificates"
              minOccurs="0" />
            <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
              processContents="lax" />
          </xsd:sequence>
        </xsd:restriction>
      </xsd:complexContent>

```

```

</xsd:complexType>

<xsd:complexType name="Body" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Body">
      <xsd:sequence>
        <xsd:element ref="osci:responseToExitDialog" />
      </xsd:sequence>
      <xsd:attribute name="Id" type="xsd:ID" use="required" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- Schema für Dialogeantworten:

```

<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Dialogeantwort
      $RCSfile: ResponseToExitDialog.xsd,v $, $Revision: 1.5 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### derived types ### -->

  <xsd:complexType name="ControlBlockType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ControlBlockTemplate">
        <xsd:sequence>
          <xsd:element name="Response" type="xsd:string" minOccurs="1" />
          <xsd:element name="Challenge" type="xsd:string"
            minOccurs="0" maxOccurs="0" />
        </xsd:sequence>
        <xsd:attribute name="ConversationId" type="osci:Number"
          use="required" />
        <xsd:attribute name="SequenceNumber" type="osci:Number"

```

```

        use="required" />
    </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="IntermediaryCertificatesType">
    <xsd:complexContent>
        <xsd:restriction base="osci:IntermediaryCertificatesTemplate">
            <xsd:sequence>
                <xsd:element name="SignatureCertificateIntermediary"
                    type="osci:CertificateType" minOccurs="0" />
            </xsd:sequence>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="responseToExitDialogType">
    <xsd:complexContent>
        <xsd:extension base="osci:DefaultBodyBlockTemplate">
            <xsd:sequence>
                <xsd:element name="Feedback" type="osci:FeedbackType" />
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="SupplierSignature" type="osci:SignatureBlockType" />

<xsd:element name="IntermediaryCertificates"
    type="osci:IntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

<xsd:element name="responseToExitDialog"
    type="osci:responseToExitDialogType" />

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)

### 6.6.5 Messaged-Anforderungsauftrag

Ein Messaged-Anforderungsauftrag wird als SOAP-Envelope realisiert.

Das folgende Beispiel erläutert den Aufbau eines MessageId-Anforderungsauftrags:

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

<!-- XML-Beispiel OSCI 1.2 Message-ID-Anforderungsauftrag -->
<!-- $RCSfile: GetMessageId.xml,v $, $Revision: 1.9 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    soapGetMessageId.xsd
    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">

  <soap:Header>
    <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
      ConversationId="87634586123" SequenceNumber="5">
      <osci:Response>AAQCAEMmCZtuMFQxDSa</osci:Response>
      <osci:Challenge>345nchfcfoqc5dfg</osci:Challenge>
    </osci:ControlBlock>
    <osci:DesiredLanguages Id="XREF-0002" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
      LanguagesList="de en-US" />
  </soap:Header>
  <soap:Body Id="XREF-0100">
    <osci:getMessageId />
  </soap:Body>
</soap:Envelope>
```

Ein MessageId-Anforderungsauftrag genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für MessageId-Anforderungsaufträge:

```
<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.osci.de/2002/04/osci"
    schemaLocation="GetMessageId.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - MessageId-Anforderungsauftrag SOAP-Envelope
    </xsd:documentation>
  </xsd:annotation>
```



```

    $RCSfile: soapGetMessageId.xsd,v $, $Revision: 1.7 $
  </xsd:documentation>
</xsd:annotation>

<!-- ### restrictions ### -->

<xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

  <!-- ### Envelope, Header und Body ### -->

  <xsd:complexType name="Envelope" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Envelope">
        <xsd:sequence>
          <xsd:element ref="soap:Header" minOccurs="1" />
          <xsd:element ref="soap:Body" minOccurs="1" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Header" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Header">
        <xsd:sequence>
          <xsd:element ref="osci:ControlBlock" />
          <xsd:element ref="osci:ClientSignature" minOccurs="0" />
          <xsd:element ref="osci:DesiredLanguages" />
          <xsd:element ref="osci:NonIntermediaryCertificates"
            minOccurs="0" />
          <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
            processContents="lax" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Body" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Body">
        <xsd:sequence>
          <xsd:element ref="osci:getMessageId" />
        </xsd:sequence>
        <xsd:attribute name="Id" type="xsd:ID" use="required" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

</xsd:redefine>

```

```
</xsd:schema>
```

- Schema für Messaged-Anforderungsaufträge:

```
<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - MessageId-Anforderungsauftrag
      $RCSfile: GetMessageId.xsd,v $, $Revision: 1.6 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### derived types ### -->

  <xsd:complexType name="ControlBlockType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ControlBlockTemplate">
        <xsd:sequence>
          <xsd:element name="Response" type="xsd:string" minOccurs="0" />
          <xsd:element name="Challenge" type="xsd:string" minOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="ConversationId" type="osci:Number"
          use="optional" />
        <xsd:attribute name="SequenceNumber" type="osci:Number"
          use="required" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="NonIntermediaryCertificatesType">
    <xsd:complexContent>
      <xsd:restriction base="osci:NonIntermediaryCertificatesTemplate">
        <xsd:sequence>
          <xsd:element name="CipherCertificateOriginator"
            type="osci:CertificateType" minOccurs="0" />
          <xsd:element name="SignatureCertificateOriginator"
            type="osci:CertificateType" minOccurs="0" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
```

```

    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="getMessageIdType">
    <xsd:complexContent>
      <xsd:extension base="osci:DefaultBodyBlockTemplate">
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>

    <!-- ### global types ### -->

    <!-- ### global elements SOAP-Header ### -->

    <xsd:element name="ControlBlock" type="osci:ControlBlockType" />

    <xsd:element name="ClientSignature" type="osci:SignatureBlockType" />

    <xsd:element name="DesiredLanguages" type="osci:DesiredLanguagesType" />

    <xsd:element name="NonIntermediaryCertificates"
      type="osci:NonIntermediaryCertificatesType" />

    <!-- ### global elements SOAP-Body ### -->

    <xsd:element name="getMessageId" type="osci:getMessageIdType" />

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)

### 6.6.6 Messageld-Anforderungsantwort

Eine Messageld-Anforderungsantwort wird als SOAP-Envelope realisiert.

Das folgende Beispiel erläutert den Aufbau einer Messageld-Anforderungsantwort:

```

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

<!-- XML-Beispiel OSCI 1.2 Message-ID-Anforderungsantwort -->
<!-- $RCSfile: ResponseToGetMessageId.xml,v $, $Revision: 1.9 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    soapResponseToGetMessageId.xsd
    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">

```

```

<soap:Header>
  <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
    soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
    ConversationId="87634586123" SequenceNumber="6">
    <osci:Response>345nchfcfoqc5dfg</osci:Response>
    <osci:Challenge>n5vl2dfie9uf6d</osci:Challenge>
  </osci:ControlBlock>
</soap:Header>
<soap:Body Id="XREF-0100">
  <osci:responseToGetMessageId>
    <osci:Feedback>
      <osci:Entry xml:lang="de">
        <osci:Code>0801</osci:Code>
        <osci:Text>Auftrag ausgeführt, Dialog weiterhin
geöffnet</osci:Text>
      </osci:Entry>
    </osci:Feedback>
    <osci:MessageId>MessageId/base64codiert</osci:MessageId>
  </osci:responseToGetMessageId>
</soap:Body>
</soap:Envelope>

```

Eine Messageld-Anforderungsantwort genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Messageld-Anforderungsantworten:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.osci.de/2002/04/osci"
    schemaLocation="ResponseToGetMessageId.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - MessageId-Anforderungsantwort SOAP-Envelope
      $RCSfile: soapResponseToGetMessageId.xsd,v $, $Revision: 1.7 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### restrictions ### -->

  <xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

  <!-- ### Envelope, Header und Body ### -->

```

```

<xsd:complexType name="Envelope" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Envelope">
      <xsd:sequence>
        <xsd:element ref="soap:Header" minOccurs="1" />
        <xsd:element ref="soap:Body" minOccurs="1" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="Header" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Header">
      <xsd:sequence>
        <xsd:element ref="osci:ControlBlock" />
        <xsd:element ref="osci:SupplierSignature" minOccurs="0" />
        <xsd:element ref="osci:IntermediaryCertificates"
          minOccurs="0" />
        <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
          processContents="lax" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="Body" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Body">
      <xsd:sequence>
        <xsd:element ref="osci:responseToGetMessageId" />
      </xsd:sequence>
      <xsd:attribute name="Id" type="xsd:ID" use="required" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- Schema für Messageld-Anforderungsantworten:

```

<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"

```

```
xmlns:osci="http://www.osci.de/2002/04/osci"
attributeFormDefault="unqualified"
elementFormDefault="qualified">

<xsd:include schemaLocation="order.xsd" />

<xsd:annotation>
  <xsd:documentation xml:lang="de">
    OSCI 1.2 - MessageId-Anforderungsantwort
    $RCSfile: ResponseToGetMessageId.xsd,v $, $Revision: 1.6 $
  </xsd:documentation>
</xsd:annotation>

<!-- ### derived types ### -->

<xsd:complexType name="ControlBlockType">
  <xsd:complexContent>
    <xsd:restriction base="osci:ControlBlockTemplate">
      <xsd:sequence>
        <xsd:element name="Response" type="xsd:string" minOccurs="1" />
        <xsd:element name="Challenge" type="xsd:string" minOccurs="0" />
      </xsd:sequence>
      <xsd:attribute name="ConversationId" type="osci:Number"
        use="required" />
      <xsd:attribute name="SequenceNumber" type="osci:Number"
        use="required" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="IntermediaryCertificatesType">
  <xsd:complexContent>
    <xsd:restriction base="osci:IntermediaryCertificatesTemplate">
      <xsd:sequence>
        <xsd:element name="SignatureCertificateIntermediary"
          type="osci:CertificateType" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="responseToGetMessageIdType">
  <xsd:complexContent>
    <xsd:extension base="osci:DefaultBodyBlockTemplate">
      <xsd:sequence>
        <xsd:element name="Feedback" type="osci:FeedbackType" />
        <xsd:element name="MessageId" type="osci:MessageIdType"
          minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```

    </xsd:complexContent>
</xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="SupplierSignature" type="osci:SignatureBlockType" />

<xsd:element name="IntermediaryCertificates"
  type="osci:IntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

<xsd:element name="responseToGetMessageId"
  type="osci:responseToGetMessageIdType" />

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)

### 6.6.7 Zustellungsauftrag

Ein Zustellungsauftrag wird entweder als SOAP-Envelope realisiert oder (wenn er Attachments enthält) als SOAP-Message-Package im Sinne von [SOAP A].

Das folgende Beispiel erläutert den Aufbau eines Zustellungsauftrags:

```

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

<!-- XML-Beispiel OSCI 1.2 Zustellungsauftrag -->
<!-- $RCSfile: StoreDelivery.xml,v $, $Revision: 1.9 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    soapStoreDelivery.xsd
    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">
  <soap:Header>
    <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
      ConversationId="87634586123" SequenceNumber="5">
      <osci:Response>AAQCAEMmCZtuMFQxDSa</osci:Response>
      <osci:Challenge>345nchfcfoqc5dfg</osci:Challenge>
    </osci:ControlBlock>

```

```

<osci:DesiredLanguages Id="XREF-0002" soap:mustUnderstand="1"
  soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
  LanguagesList="de en-US" />
<osci:QualityOfTimestamp Id="XREF-0003" soap:mustUnderstand="1"
  soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
  Service="creation" Quality="plain" />
<osci:QualityOfTimestamp Id="XREF-0004" soap:mustUnderstand="1"
  soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
  Service="reception" Quality="plain" />
<osci:storeDelivery Id="XREF-0005" soap:mustUnderstand="1"
  soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
  <osci:MessageId>MessageId/Base64codiert</osci:MessageId>
  <osci:Subject>Betreff der Zustellung</osci:Subject>
</osci:storeDelivery>
<osci:NonIntermediaryCertificates Id="XREF-0010"
  soap:mustUnderstand="1"
  soap:actor="http://www.w3.org/2001/12/soap-envelope/actor/none">
  <osci:CipherCertificateAddressee>
    <ds:X509Data>
      <ds:X509Certificate>
        Chiffrierzertifikat/Empfaenger/Base64codiert
      </ds:X509Certificate>
    </ds:X509Data>
  </osci:CipherCertificateAddressee>
</osci:NonIntermediaryCertificates>
</soap:Header>
<soap:Body Id="XREF-0100">
  <osci:ContentPackage>
    <osci:ContentContainer>
      <osci:Content />
    </osci:ContentContainer>
  </osci:ContentPackage>
</soap:Body>
</soap:Envelope>

```

Ein Zustellungsauftrag (bzw. sein Root-Body-Part) genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Zustellungsaufträge:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.osci.de/2002/04/osci"
    schemaLocation="StoreDelivery.xsd" />

```



```

<xsd:annotation>
  <xsd:documentation xml:lang="de">
    OSCI 1.2 - Zustellungsauftrag SOAP-Envelope
    $RCSfile: soapStoreDelivery.xsd,v $, $Revision: 1.6 $
  </xsd:documentation>
</xsd:annotation>

<!-- ### restrictions ### -->

<xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

  <!-- ### Envelope, Header und Body ### -->

  <xsd:complexType name="Envelope" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Envelope">
        <xsd:sequence>
          <xsd:element ref="soap:Header" minOccurs="1" />
          <xsd:element ref="soap:Body" minOccurs="1" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Header" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Header">
        <xsd:sequence>
          <xsd:element ref="osci:ControlBlock" />
          <xsd:element ref="osci:ClientSignature" minOccurs="0" />
          <xsd:element ref="osci:DesiredLanguages" />
          <xsd:element ref="osci:QualityOfTimestamp"
            minOccurs="2" maxOccurs="2" />
          <xsd:element ref="osci:storeDelivery" />
          <xsd:element ref="osci:NonIntermediaryCertificates" />
          <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
            processContents="lax" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Body" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Body">
        <xsd:sequence>
          <xsd:element ref="osci:ContentPackage" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

```

```

        <xsd:attribute name="Id" type="xsd:ID" use="required" />
    </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- **Schema für Zustellungsaufträge:**

```

<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Zustellungsauftrag
      $RCSfile: storeDelivery.xsd,v $, $Revision: 1.4 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### derived types ### -->

  <xsd:complexType name="ControlBlockType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ControlBlockTemplate">
        <xsd:sequence>
          <xsd:element name="Response" type="xsd:string" minOccurs="0" />
          <xsd:element name="Challenge" type="xsd:string" minOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="ConversationId" type="osci:Number"
          use="optional" />
        <xsd:attribute name="SequenceNumber" type="osci:Number"
          use="required" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="storeDeliveryType">
    <xsd:complexContent>
      <xsd:extension base="osci:DefaultHeaderBlockTemplate">
        <xsd:sequence>

```

```
<xsd:element name="MessageId" type="osci:MessageIdType" />
  <xsd:element name="Subject" type="xsd:string" minOccurs="0" />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="NonIntermediaryCertificatesType">
  <xsd:complexContent>
    <xsd:restriction base="osci:NonIntermediaryCertificatesTemplate">
      <xsd:sequence>
        <xsd:element name="CipherCertificateOriginator"
          type="osci:CertificateType" minOccurs="0" />
        <xsd:element name="CipherCertificateOtherAuthor"
          type="osci:CertificateType"
          minOccurs="0" maxOccurs="unbounded" />
        <xsd:element name="CipherCertificateAddressee"
          type="osci:CertificateType" minOccurs="1" />
        <xsd:element name="CipherCertificateOtherReader"
          type="osci:CertificateType"
          minOccurs="0" maxOccurs="unbounded" />
        <xsd:element name="SignatureCertificateOriginator"
          type="osci:CertificateType" minOccurs="0" />
        <xsd:element name="SignatureCertificateOtherAuthor"
          type="osci:CertificateType"
          minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="ClientSignature" type="osci:SignatureBlockType" />

<xsd:element name="DesiredLanguages" type="osci:DesiredLanguagesType" />

<xsd:element name="QualityOfTimestamp"
  type="osci:QualityOfTimestampType" />

<xsd:element name="storeDelivery" type="osci:storeDeliveryType" />

<xsd:element name="NonIntermediaryCertificates"
  type="osci:NonIntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->
```



```

        <osci:Subject>Betreff der Zustellung</osci:Subject>
    </osci:ProcessCard>
    <osci:InspectionReport />
</osci:ProcessCardBundle>
</osci:responseToStoreDelivery>
</soap:Header>
<soap:Body Id="XREF-0100" />
</soap:Envelope>

```

Ist der Intermediär in der Lage, zu der mittels *Zustellungsauftrag* eingereichten Zustellung einen Laufzettel zu erzeugen, so wird dieser innerhalb der *Zustellungsantwort* an den Benutzer gesendet.

Eine Zustellungsantwort genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Zustellungsantworten:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.osci.de/2002/04/osci"
    schemaLocation="ResponseToStoreDelivery.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Zustellungsantwort SOAP-Envelope
      $RCSfile: soapResponseToStoreDelivery.xsd,v $, $Revision: 1.5 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### restrictions ### -->

  <xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

    <!-- ### Envelope, Header und Body ### -->

    <xsd:complexType name="Envelope" >
      <xsd:complexContent>
        <xsd:restriction base="soap:Envelope">
          <xsd:sequence>
            <xsd:element ref="soap:Header" minOccurs="1" />
            <xsd:element ref="soap:Body" minOccurs="1" />
          </xsd:sequence>
        </xsd:restriction>
      </xsd:complexContent>

```

```

</xsd:complexType>

<xsd:complexType name="Header" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Header">
      <xsd:sequence>
        <xsd:element ref="osci:ControlBlock" />
        <xsd:element ref="osci:SupplierSignature" minOccurs="0" />
        <xsd:element ref="osci:responseToStoreDelivery" />
        <xsd:element ref="osci:IntermediaryCertificates"
          minOccurs="0" />
        <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
          processContents="lax" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="Body" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Body">
      <xsd:attribute name="Id" type="xsd:ID" use="required" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- Schema für Zustellungsantworten:

```

<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Zustellungsantwort
      $RCSfile: ResponseToStoreDelivery.xsd,v $, $Revision: 1.4 $
    </xsd:documentation>
  </xsd:annotation>

```

```
<!-- ### derived types ### -->

<xsd:complexType name="ControlBlockType">
  <xsd:complexContent>
    <xsd:restriction base="osci:ControlBlockTemplate">
      <xsd:sequence>
        <xsd:element name="Response" type="xsd:string" minOccurs="1" />
        <xsd:element name="Challenge" type="xsd:string" minOccurs="0" />
      </xsd:sequence>
      <xsd:attribute name="ConversationId" type="osci:Number"
        use="required" />
      <xsd:attribute name="SequenceNumber" type="osci:Number"
        use="required" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ProcessCardBundleType">
  <xsd:complexContent>
    <xsd:restriction base="osci:ProcessCardBundleTemplate">
      <xsd:sequence>
        <xsd:element name="MessageId" type="osci:MessageIdType" />
        <xsd:element name="ProcessCard" type="osci:ProcessCardType" />
        <xsd:element name="InspectionReport"
          type="osci:InspectionReportType" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ProcessCardType">
  <xsd:complexContent>
    <xsd:restriction base="osci:ProcessCardTemplate">
      <xsd:sequence>
        <xsd:element name="Creation" type="osci:TimestampType" />
        <xsd:element name="Subject" type="xsd:string" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="responseToStoreDeliveryType">
  <xsd:complexContent>
    <xsd:extension base="osci:DefaultHeaderBlockTemplate">
      <xsd:sequence>
        <xsd:element name="Feedback" type="osci:FeedbackType" />
        <xsd:element name="ProcessCardBundle"
          type="osci:ProcessCardBundleType" minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```

    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="IntermediaryCertificatesType">
  <xsd:complexContent>
    <xsd:restriction base="osci:IntermediaryCertificatesTemplate">
      <xsd:sequence>
        <xsd:element name="SignatureCertificateIntermediary"
          type="osci:CertificateType" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="SupplierSignature" type="osci:SignatureBlockType" />

<xsd:element name="responseToStoreDelivery"
  type="osci:responseToStoreDeliveryType" />

<xsd:element name="IntermediaryCertificates"
  type="osci:IntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)

### 6.6.9 Zustellungsabholauftrag

Ein Zustellungsabholauftrag wird als SOAP-Envelope realisiert.

Das folgende Beispiel erläutert den Aufbau eines Zustellungsabholauftrags:

```

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

<!-- XML-Beispiel OSCI 1.2 Zustellungsabholauftrag -->
<!-- $RCSfile: FetchDelivery.xml,v $, $Revision: 1.7 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:osci="http://www.osci.de/2002/04/osci"

```



```

xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
                    soapFetchDelivery.xsd
                    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
                    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">
<soap:Header>
  <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
    soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
    ConversationId="87634586123" SequenceNumber="5">
    <osci:Response>AAQCAEMmCZtuMFQxDSa</osci:Response>
    <osci:Challenge>345nchfcfoqc5dfg</osci:Challenge>
  </osci:ControlBlock>
  <osci:DesiredLanguages Id="XREF-0002" soap:mustUnderstand="1"
    soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
    LanguagesList="de en-US" />
  <osci:fetchDelivery Id="XREF-0005" soap:mustUnderstand="1"
    soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
    <osci:SelectionRule>
      <osci:ReceptionOfDelivery>2002-04-01T12:00:00.000-
01:00</osci:ReceptionOfDelivery>
    </osci:SelectionRule>
  </osci:fetchDelivery>
</soap:Header>
<soap:Body Id="XREF-0100" />
</soap:Envelope>

```

Der Benutzer hat optional die Möglichkeit, mittels des Elements `osci:SelectionRule` Auswahlkriterien anzugeben, mit denen eine angeforderte Zustellung näher spezifiziert werden kann. Als Auswahlkriterien kann der Benutzer wahlweise einen Empfangszeitpunkt oder eine Messageld angeben.

Der Intermediär ermittelt eine ggf. in einer *Zustellungsabholantwort* an den Benutzer zu sendende Zustellung anhand der nachfolgenden Regeln:

1. Wird das Element `osci:MessageId` eingestellt, so wird diejenige Zustellung zurückgegeben, welche der in `osci:MessageId` angegebenen Messageld entspricht.
2. Wird das Element `osci:ReceptionOfDelivery` eingestellt, so wird die älteste Zustellung (Zeitpunkt der Einreichung beim Intermediär) zurückgegeben, deren Zeitpunkt der Einreichung beim Intermediär nach dem in `osci:ReceptionOfDelivery` angegebenen Empfangszeitpunkt liegt.
3. Ist weder das Element `osci:ReceptionOfDelivery` noch das Element `osci:MessageId` vorhanden, so wird die Zustellung mit dem ältesten Zeitpunkt der Einreichung beim Intermediär zurückgesendet.

Ein Zustellungsabholauftrag genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Zustellungsabholaufträge:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"

```

```

xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:osci="http://www.osci.de/2002/04/osci"
attributeFormDefault="unqualified"
elementFormDefault="qualified">

<xsd:import namespace="http://www.osci.de/2002/04/osci"
  schemaLocation="FetchDelivery.xsd" />

<xsd:annotation>
  <xsd:documentation xml:lang="de">
    OSCI 1.2 - Zustellungsabholauftrag SOAP-Envelope
    $RCSfile: soapFetchDelivery.xsd,v $, $Revision: 1.4 $
  </xsd:documentation>
</xsd:annotation>

<!-- ### restrictions ### -->

<xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

  <!-- ### Envelope, Header und Body ### -->

  <xsd:complexType name="Envelope" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Envelope">
        <xsd:sequence>
          <xsd:element ref="soap:Header" minOccurs="1" />
          <xsd:element ref="soap:Body" minOccurs="1" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Header" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Header">
        <xsd:sequence>
          <xsd:element ref="osci:ControlBlock" />
          <xsd:element ref="osci:ClientSignature" minOccurs="0" />
          <xsd:element ref="osci:DesiredLanguages" />
          <xsd:element ref="osci:fetchDelivery" />
          <xsd:element ref="osci:NonIntermediaryCertificates"
            minOccurs="0" />
          <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
            processContents="lax" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

```

```

<xsd:complexType name="Body" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Body">
      <xsd:attribute name="Id" type="xsd:ID" use="required" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- **Schema für Zustellungsabholaufträge:**

```

<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Zustellungsabholauftrag
      $RCSfile: fetchDelivery.xsd,v $, $Revision: 1.2 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### derived types ### -->

  <xsd:complexType name="ControlBlockType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ControlBlockTemplate">
        <xsd:sequence>
          <xsd:element name="Response" type="xsd:string" minOccurs="1" />
          <xsd:element name="Challenge" type="xsd:string" minOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="ConversationId" type="osci:Number"
          use="required" />
        <xsd:attribute name="SequenceNumber" type="osci:Number"
          use="required" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="fetchDeliveryType">

```

```

<xsd:complexContent>
  <xsd:extension base="osci:DefaultHeaderBlockTemplate">
    <xsd:sequence>
      <xsd:element name="SelectionRule" minOccurs="0">
        <xsd:complexType>
          <xsd:choice>
            <xsd:element name="ReceptionOfDelivery"
              type="xsd:dateTime" />
            <xsd:element name="MessageId" type="osci:MessageIdType" />
          </xsd:choice>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="NonIntermediaryCertificatesType">
  <xsd:complexContent>
    <xsd:restriction base="osci:NonIntermediaryCertificatesTemplate">
      <xsd:sequence>
        <xsd:element name="SignatureCertificateOriginator"
          type="osci:CertificateType" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="ClientSignature" type="osci:SignatureBlockType" />

<xsd:element name="DesiredLanguages" type="osci:DesiredLanguagesType" />

<xsd:element name="fetchDelivery" type="osci:fetchDeliveryType" />

<xsd:element name="NonIntermediaryCertificates"
  type="osci:NonIntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)

### 6.6.10 Zustellungsabholantwort

Eine Zustellungsabholantwort wird entweder als SOAP-Envelope realisiert oder (wenn sie Attachments enthält) als SOAP-Message-Package im Sinne von [SOAP A].

Das folgende Beispiel erläutert den Aufbau einer Zustellungsabholantwort:

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

<!-- XML-Beispiel OSCI 1.2 Zustellungsabholantwort -->
<!-- $RCSfile: ResponseToFetchDelivery.xml,v $, $Revision: 1.7 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    soapResponseToFetchDelivery.xsd
    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">
  <soap:Header>
    <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
      ConversationId="87634586123" SequenceNumber="6">
      <osci:Response>n5vl2dfie9uf6d</osci:Response>
      <osci:Challenge>fie9uf6dGGFRGm3</osci:Challenge>
    </osci:ControlBlock>
    <osci:responseToFetchDelivery Id="XREF-0005" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
      <osci:Feedback>
        <osci:Entry xml:lang="de">
          <osci:Code>0801</osci:Code>
          <osci:Text>Auftrag ausgeführt, Dialog weiterhin
geöffnet</osci:Text>
        </osci:Entry>
      </osci:Feedback>
      <osci:fetchDelivery>
        <osci:SelectionRule>
          <osci:ReceptionOfDelivery>2002-04-01T12:00:00.000-
01:00</osci:ReceptionOfDelivery>
        </osci:SelectionRule>
      </osci:fetchDelivery>
      <osci:ProcessCardBundle>
        <osci:MessageId>MessageId/Base64codiert</osci:MessageId>
        <osci:ProcessCard RecentModification="2002-04-24T12:15:01.516-
01:00">
          <osci:Creation>
            <osci:Plain>2002-04-23T23:41:05.527-01:00</osci:Plain>
          </osci:Creation>
          <osci:Forwarding>
```

```

    <osci:Plain>2002-04-24T12:15:01.516-01:00</osci:Plain>
  </osci:Forwarding>
  <osci:Subject>Betreff der Zustellung</osci:Subject>
</osci:ProcessCard>
<osci:InspectionReport>
  <osci:Inspection>
    <osci:Timestamp>
      <osci:Plain>2002-04-24T12:14:32.291-01:00</osci:Plain>
    </osci:Timestamp>
    <osci:X509IssuerName>Otto Mustermann</osci:X509IssuerName>
    <osci:X509SerialNumber>1625843</osci:X509SerialNumber>
    <osci:CertType Type="qualified" />
    <osci:MathResult Result="ok" />
    <osci:OfflineResult Result="valid" />
    <osci:OnlineResult Result="ok">
      <osci:LDAP />
    </osci:OnlineResult>
  </osci:Inspection>
</osci:InspectionReport>
</osci:ProcessCardBundle>
</osci:responseToFetchDelivery>
<osci:NonIntermediaryCertificates Id="XREF-0020"
  soap:mustUnderstand="1"
  soap:actor="http://www.w3.org/2001/12/soap-envelope/actor/none">
  <osci:CipherCertificateAddressee Id="XREF-4444">
    <ds:X509Data>
      <ds:X509Certificate>
        Chiffrierzertifikat/Empfaenger/Base64codiert
      </ds:X509Certificate>
    </ds:X509Data>
  </osci:CipherCertificateAddressee>
</osci:NonIntermediaryCertificates>
</soap:Header>
<soap:Body Id="XREF-0100">
  <osci:ContentPackage>
    <xenc:EncryptedData>
      <xenc:EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc" />
      <ds:KeyInfo>
        <xenc:EncryptedKey>
          <xenc:EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5">
            <xenc:KeySize>1024</xenc:KeySize>
          </xenc:EncryptionMethod>
        <ds:KeyInfo>
          <ds:RetrievalMethod URI="#XREF-4444" />
        </ds:KeyInfo>
      </xenc:EncryptedKey>
    </xenc:EncryptedData>
  </osci:ContentPackage>

```

```

<xenc:CipherValue>Verschluesselter/Containerschluessel/Base64codiert</xenc:
CipherValue>
    </xenc:CipherData>
    </xenc:EncryptedKey>
</ds:KeyInfo>
<xenc:CipherData>

<xenc:CipherValue>Verschluesselter/Inhaltsdatencontainer/Base64codiert</xen
c:CipherValue>
    </xenc:CipherData>
    </xenc:EncryptedData>
</osci:ContentPackage>
</soap:Body>
</soap:Envelope>

```

Die Auswertung der Auswahlkriterien des Elements `SelectionRule` im *Zustellungsabholauftrag* ist dort (Abschnitt 6.6.9) beschrieben.

Bezugnehmend auf diese Auswahlregeln im *Zustellungsabholauftrag* gilt für die Rückmeldungen in der *Zustellungsabholantwort* ergänzend:

- Liegen beim Intermediär (gemäß der Punkte 1.-3.) keine Zustellungen für den Benutzer vor, wird keine Zustellung in der *Zustellabholantwort* gesendet und der Benutzer erhält eine dementsprechende Rückmeldung (siehe Kapitel 5).
- Hält der Intermediär weitere Zustellungen für den Benutzer bereit, erhält der Benutzer hierüber eine Rückmeldung (siehe Kapitel 5). Das Vorliegen weiterer Zustellungen ist dabei nicht an die Auswahlkriterien im Element `osci:SelectionRule` des *Zustellungsabholauftrags* gebunden.

Eine Zustellungsabholantwort (bzw. ihr Root-Body-Part) genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Zustellungsabholantworten:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.osci.de/2002/04/osci"
    schemaLocation="ResponseToFetchDelivery.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Zustellungsabholantwort SOAP-Envelope
      $RCSfile: soapResponseToFetchDelivery.xsd,v $, $Revision: 1.5 $
    </xsd:documentation>
  </xsd:annotation>

```

```

<!-- ### restrictions ### -->

<xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

<!-- ### Envelope, Header und Body ### -->

<xsd:complexType name="Envelope" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Envelope">
      <xsd:sequence>
        <xsd:element ref="soap:Header" minOccurs="1" />
        <xsd:element ref="soap:Body" minOccurs="1" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="Header" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Header">
      <xsd:sequence>
        <xsd:element ref="osci:ControlBlock" />
        <xsd:element ref="osci:SupplierSignature" minOccurs="0" />
        <xsd:element ref="osci:responseToFetchDelivery" />
        <xsd:element ref="osci:IntermediaryCertificates"
          minOccurs="0" />
        <xsd:element ref="osci:NonIntermediaryCertificates"
          minOccurs="0" />
        <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
          processContents="lax" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="Body" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Body">
      <xsd:sequence>
        <xsd:element ref="osci:ContentPackage" minOccurs="0" />
      </xsd:sequence>
      <xsd:attribute name="Id" type="xsd:ID" use="required" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

```



```
</xsd:schema>
```

- Schema für Zustellungsabholantworten:

```
<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Zustellungsabholantwort
      $RCSfile: ResponseToFetchDelivery.xsd,v $, $Revision: 1.3 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### derived types ### -->

  <xsd:complexType name="ControlBlockType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ControlBlockTemplate">
        <xsd:sequence>
          <xsd:element name="Response" type="xsd:string" minOccurs="1" />
          <xsd:element name="Challenge" type="xsd:string" minOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="ConversationId" type="osci:Number"
          use="required" />
        <xsd:attribute name="SequenceNumber" type="osci:Number"
          use="required" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="ProcessCardBundleType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ProcessCardBundleTemplate">
        <xsd:sequence>
          <xsd:element name="MessageId" type="osci:MessageIdType" />
          <xsd:element name="ProcessCard" type="osci:ProcessCardType" />
          <xsd:element name="InspectionReport"
            type="osci:InspectionReportType" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
```

```
</xsd:complexType>

<xsd:complexType name="ProcessCardType">
  <xsd:complexContent>
    <xsd:restriction base="osci:ProcessCardTemplate">
      <xsd:sequence>
        <xsd:element name="Creation" type="osci:TimestampType" />
        <xsd:element name="Forwarding" type="osci:TimestampType" />
        <xsd:element name="Subject" type="xsd:string" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="responseToFetchDeliveryType">
  <xsd:complexContent>
    <xsd:extension base="osci:DefaultHeaderBlockTemplate">
      <xsd:sequence>
        <xsd:element name="Feedback" type="osci:FeedbackType" />
        <xsd:element name="fetchDelivery">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:any namespace="http://www.osci.de/2002/04/osci"
                minOccurs="0" maxOccurs="unbounded" processContents="strict" />
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="ProcessCardBundle"
          type="osci:ProcessCardBundleType" minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="IntermediaryCertificatesType">
  <xsd:complexContent>
    <xsd:restriction base="osci:IntermediaryCertificatesTemplate">
      <xsd:sequence>
        <xsd:element name="SignatureCertificateIntermediary"
          type="osci:CertificateType" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="NonIntermediaryCertificatesType">
  <xsd:complexContent>
    <xsd:extension base="osci:NonIntermediaryCertificatesTemplate" />
  </xsd:complexContent>
</xsd:complexType>
```

```

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="SupplierSignature" type="osci:SignatureBlockType" />

<xsd:element name="responseToFetchDelivery"
  type="osci:responseToFetchDeliveryType" />

<xsd:element name="IntermediaryCertificates"
  type="osci:IntermediaryCertificatesType" />

<xsd:element name="NonIntermediaryCertificates"
  type="osci:NonIntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

<xsd:element name="ContentPackage" type="osci:ContentPackageType" />

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)
- Ausprägung von XML-Encryption(siehe Abschnitt 6.5)

### 6.6.11 Laufzettelabholauftrag

Ein Laufzettelabholauftrag wird als SOAP-Envelope realisiert.

Das folgende Beispiel erläutert den Aufbau eines Laufzettelabholauftrags:

```

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

<!-- XML-Beispiel OSCI 1.2 Laufzettelabholauftrag -->
<!-- $RCSfile: FetchProcessCard.xml,v $, $Revision: 1.6 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    soapFetchProcessCard.xsd
    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">
  <soap:Header>
    <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
      ConversationId="87634586123" SequenceNumber="5">

```

```

    <osci:Response>AAQCAEMmCZtuMFQxDSa</osci:Response>
    <osci:Challenge>345nchfcfoqc5dfg</osci:Challenge>
</osci:ControlBlock>
    <osci:DesiredLanguages Id="XREF-0002" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
      LanguagesList="de en-US" />
</soap:Header>
<soap:Body Id="XREF-0100">
  <osci:fetchProcessCard>
    <osci:SelectionRule>
      <osci:RecentModification>2002-04-01T12:00:00.000-
01:00</osci:RecentModification>
    </osci:SelectionRule>
    <osci:Quantity Limit="1" />
  </osci:fetchProcessCard>
</soap:Body>
</soap:Envelope>

```

Das Element `osci:SelectionRule` wird im *Laufzettelabholauftrag* dazu verwendet, die angeforderten Laufzettel näher zu spezifizieren.

Der Benutzer kann wahlweise einen Empfangszeitpunkt, einen Änderungszeitpunkt oder eine unbegrenzte Anzahl von Messagelds angeben.

Der Intermediär ermittelt die ggf. in einer *Laufzettelabholantwort* an den Benutzer zu sendenden Laufzettel anhand der nachfolgenden Regeln:

1. Zu jedem eingestellten Element `osci:MessageId`, wird derjenige Laufzettel zurückgegeben, welcher der in `osci:MessageId` angegebenen Messageld entspricht.
2. Wird das Element `osci:ReceptionOfDelivery` eingestellt, so werden alle Laufzettel zu Zustellungen zurückgegeben, deren Zeitpunkt der Einreichung beim Intermediär jünger als der in `osci:ReceptionOfDelivery` angegebene Empfangszeitpunkt ist.
3. Wird das Element `osci:RecentModification` eingestellt, so werden alle Laufzettel zurückgegeben, deren letzter Änderungszeitpunkt jünger als der in `osci:RecentModification` angegebene Zeitpunkt ist.
4. Ist keines der Elemente `osci:ReceptionOfDelivery`, `osci:RecentModification` oder `osci:MessageId` angegeben, so werden alle für den Benutzer vorliegenden Laufzettel gesendet.

Der Benutzer hat weiterhin optional die Möglichkeit die Menge zurückzuliefernder Laufzettel zahlenmäßig zu begrenzen.

5. Hat der Benutzer ein Element `osci:Quantity` angegeben, so wird in einer *Laufzettelabholantwort* von den gemäß der Punkte 1.-4. ermittelten Laufzetteln maximal die im Attribut `Limit` bezifferte Anzahl an den Benutzer gesendet.

Ein Laufzettelabholauftrag genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Laufzettelabholaufträge:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"

```

```

xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:osci="http://www.osci.de/2002/04/osci"
attributeFormDefault="unqualified"
elementFormDefault="qualified">

<xsd:import namespace="http://www.osci.de/2002/04/osci"
  schemaLocation="FetchProcessCard.xsd" />

<xsd:annotation>
  <xsd:documentation xml:lang="de">
    OSCI 1.2 - Laufzettelsabholauftrag SOAP-Envelope
    $RCSfile: soapFetchProcessCard.xsd,v $, $Revision: 1.4 $
  </xsd:documentation>
</xsd:annotation>

<!-- ### restrictions ### -->

<xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

  <!-- ### Envelope, Header und Body ### -->

  <xsd:complexType name="Envelope" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Envelope">
        <xsd:sequence>
          <xsd:element ref="soap:Header" minOccurs="1" />
          <xsd:element ref="soap:Body" minOccurs="1" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Header" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Header">
        <xsd:sequence>
          <xsd:element ref="osci:ControlBlock" />
          <xsd:element ref="osci:ClientSignature" minOccurs="0" />
          <xsd:element ref="osci:DesiredLanguages" />
          <xsd:element ref="osci:NonIntermediaryCertificates"
            minOccurs="0" />
          <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
            processContents="lax" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Body" >

```

```

    <xsd:complexContent>
      <xsd:restriction base="soap:Body">
        <xsd:sequence>
          <xsd:element ref="osci:fetchProcessCard" />
        </xsd:sequence>
        <xsd:attribute name="Id" type="xsd:ID" use="required" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- Schema für Laufzettelabholaufträge:

```

<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Laufzettelabholauftrag
      $RCSfile: FetchProcessCard.xsd,v $, $Revision: 1.3 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### derived types ### -->

  <xsd:complexType name="ControlBlockType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ControlBlockTemplate">
        <xsd:sequence>
          <xsd:element name="Response" type="xsd:string" minOccurs="1" />
          <xsd:element name="Challenge" type="xsd:string" minOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="ConversationId" type="osci:Number"
          use="required" />
        <xsd:attribute name="SequenceNumber" type="osci:Number"
          use="required" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

```

```
<xsd:complexType name="NonIntermediaryCertificatesType">
  <xsd:complexContent>
    <xsd:restriction base="osci:NonIntermediaryCertificatesTemplate">
      <xsd:sequence>
        <xsd:element name="SignatureCertificateOriginator"
          type="osci:CertificateType" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="fetchProcessCardType">
  <xsd:complexContent>
    <xsd:extension base="osci:DefaultBodyBlockTemplate">
      <xsd:sequence>
        <xsd:element name="SelectionRule" minOccurs="0">
          <xsd:complexType>
            <xsd:choice>
              <xsd:element name="ReceptionOfDelivery"
                type="xsd:dateTime" />
              <xsd:element name="RecentModification"
                type="xsd:dateTime" />
              <xsd:element name="MessageId" type="osci:MessageIdType"
                maxOccurs="unbounded" />
            </xsd:choice>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="Quantity" minOccurs="0">
          <xsd:complexType>
            <xsd:attribute name="Limit" type="xsd:positiveInteger"
              use="required" />
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="ClientSignature" type="osci:SignatureBlockType" />

<xsd:element name="DesiredLanguages" type="osci:DesiredLanguagesType" />

<xsd:element name="NonIntermediaryCertificates"
```

```

    type="osci:NonIntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

<xsd:element name="fetchProcessCard" type="osci:fetchProcessCardType" />

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)

### 6.6.12 Laufzettelabholantwort

Eine Laufzettelabholantwort wird als SOAP-Envelope realisiert.

Das folgende Beispiel erläutert den Aufbau einer Laufzettelabholantwort:

```

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

<!-- XML-Beispiel OSCI 1.2 Laufzettelabholantwort -->
<!-- $RCSfile: ResponseToFetchProcessCard.xml,v $, $Revision: 1.7 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    soapResponseToFetchProcessCard.xsd
    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">
  <soap:Header>
    <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
      ConversationId="87634586123" SequenceNumber="6">
      <osci:Response>n5vl2dfie9uf6d</osci:Response>
      <osci:Challenge>fie9uf6dGGFRGm3</osci:Challenge>
    </osci:ControlBlock>
  </soap:Header>
  <soap:Body Id="XREF-0100">
    <osci:responseToFetchProcessCard>
      <osci:Feedback>
        <osci:Entry xml:lang="de">
          <osci:Code>3801</osci:Code>
          <osci:Text>Zu den im Auftrag angegebenen Kriterien liegen weitere
Laufzettel vor</osci:Text>
        </osci:Entry>
        <osci:Entry xml:lang="de">
          <osci:Code>0801</osci:Code>
          <osci:Text>Auftrag ausgeführt, Dialog weiterhin
geöffnet</osci:Text>
        </osci:Entry>
      </osci:Feedback>
    </osci:responseToFetchProcessCard>
  </soap:Body>
</soap:Envelope>

```



```

    </osci:Feedback>
    <osci:fetchProcessCard>
      <osci:SelectionRule>
        <osci:RecentModification>2002-04-01T12:00:00.000-
01:00</osci:RecentModification>
      </osci:SelectionRule>
      <osci:Quantity Limit="1" />
    </osci:fetchProcessCard>
    <osci:ProcessCardBundle>
      <osci:MessageId>MessageId/1/Base64codiert</osci:MessageId>
      <osci:ProcessCard
        RecentModification="2002-04-24T12:15:01.516-01:00">
        <osci:Creation>
          <osci:Plain>2002-04-23T23:41:05.527-01:00</osci:Plain>
        </osci:Creation>
        <osci:Forwarding>
          <osci:Plain>2002-04-24T12:15:01.516-01:00</osci:Plain>
        </osci:Forwarding>
        <osci:Reception>
          <osci:Plain>2002-04-24T12:17:12.169-01:00</osci:Plain>
        </osci:Reception>
        <osci:Subject>Betreff der Zustellung</osci:Subject>
      </osci:ProcessCard>
      <osci:InspectionReport />
    </osci:ProcessCardBundle>
  </osci:responseToFetchProcessCard>
</soap:Body>
</soap:Envelope>

```

Die Auswertung der Auswahlkriterien des Elements `SelectionRule` im *Laufzettelabholauftrag* ist in Abschnitt 6.6.11 beschrieben.

Bezugnehmend auf diese Auswahlregeln im *Laufzettelabholauftrag* gilt für die Rückmeldungen in der *Laufzettelabholantwort* ergänzend:

- Liegen gemäß der Punkte 1.-4. keine Laufzettel vor, so wird dieses dem Benutzer in einer Rückmeldung mitgeteilt.
- Liegen gemäß der Punkte 1.-5. mehr Laufzettel als in `Limit` angegeben vor, so wird dieses dem Benutzer in einer Rückmeldung mitgeteilt.

Eine Laufzettelabholantwort genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Laufzettelabholantworten:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

```

```

<xsd:import namespace="http://www.osci.de/2002/04/osci"
  schemaLocation="ResponseToFetchProcessCard.xsd" />

<xsd:annotation>
  <xsd:documentation xml:lang="de">
    OSCI 1.2 - Laufzettelabholantwort SOAP-Envelope
    $RCSfile: soapResponseToFetchProcessCard.xsd,v $, $Revision: 1.4 $
  </xsd:documentation>
</xsd:annotation>

<!-- ### restrictions ### -->

<xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

  <!-- ### Envelope, Header und Body ### -->

  <xsd:complexType name="Envelope" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Envelope">
        <xsd:sequence>
          <xsd:element ref="soap:Header" minOccurs="1" />
          <xsd:element ref="soap:Body" minOccurs="1" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Header" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Header">
        <xsd:sequence>
          <xsd:element ref="osci:ControlBlock" />
          <xsd:element ref="osci:SupplierSignature" minOccurs="0" />
          <xsd:element ref="osci:IntermediaryCertificates"
            minOccurs="0" />
          <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
            processContents="lax" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Body" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Body">
        <xsd:sequence>
          <xsd:element ref="osci:responseToFetchProcessCard" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

```

```

        <xsd:attribute name="Id" type="xsd:ID" use="required" />
    </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- Schema für Laufzettelabholantworten:

```

<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Laufzettelabholantwort
      $RCSfile: ResponseToFetchProcessCard.xsd,v $, $Revision: 1.4 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### derived types ### -->

  <xsd:complexType name="ControlBlockType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ControlBlockTemplate">
        <xsd:sequence>
          <xsd:element name="Response" type="xsd:string" minOccurs="1" />
          <xsd:element name="Challenge" type="xsd:string" minOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="ConversationId" type="osci:Number"
          use="required" />
        <xsd:attribute name="SequenceNumber" type="osci:Number"
          use="required" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="ProcessCardBundleType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ProcessCardBundleTemplate">
        <xsd:sequence>

```

```

    <xsd:element name="MessageId" type="osci:MessageIdType" />
    <xsd:element name="ProcessCard" type="osci:ProcessCardType" />
    <xsd:element name="InspectionReport"
      type="osci:InspectionReportType" />
  </xsd:sequence>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ProcessCardType">
  <xsd:complexContent>
    <xsd:restriction base="osci:ProcessCardTemplate">
      <xsd:sequence>
        <xsd:element name="Creation" type="osci:TimestampType" />
        <xsd:element name="Forwarding" type="osci:TimestampType"
          minOccurs="0" />
        <xsd:element name="Reception" type="osci:TimestampType"
          minOccurs="0" />
        <xsd:element name="Subject" type="xsd:string" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="responseToFetchProcessCardType">
  <xsd:complexContent>
    <xsd:extension base="osci:DefaultBodyBlockTemplate">
      <xsd:sequence>
        <xsd:element name="Feedback" type="osci:FeedbackType" />
        <xsd:element name="fetchProcessCard">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:any namespace="http://www.osci.de/2002/04/osci"
                maxOccurs="unbounded" processContents="strict"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="ProcessCardBundle"
          type="osci:ProcessCardBundleType"
          minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="IntermediaryCertificatesType">
  <xsd:complexContent>
    <xsd:restriction base="osci:IntermediaryCertificatesTemplate">
      <xsd:sequence>
        <xsd:element name="SignatureCertificateIntermediary"

```

```

        type="osci:CertificateType" minOccurs="0" />
    </xsd:sequence>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="SupplierSignature" type="osci:SignatureBlockType" />

<xsd:element name="IntermediaryCertificates"
    type="osci:IntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

<xsd:element name="responseToFetchProcessCard"
    type="osci:responseToFetchProcessCardType" />

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)

### 6.6.13 Weiterleitungsauftrag

Ein Weiterleitungsauftrag wird entweder als SOAP-Envelope realisiert oder (wenn er Attachments enthält) als SOAP-Message-Package im Sinne von [SOAP A].

Das folgende Beispiel erläutert den Aufbau eines Weiterleitungsauftrags:

```

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

<!-- XML-Beispiel OSCI 1.2 Weiterleitungsauftrag -->
<!-- $RCSfile: ForwardDelivery.xml,v $, $Revision: 1.7 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:osci="http://www.osci.de/2002/04/osci"
    xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
        soapForwardDelivery.xsd
            http://www.w3.org/2000/09/xmldsig# oscisig.xsd
            http://www.w3.org/2001/04/xmlenc# oscienc.xsd">
    <soap:Header>
        <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
            soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
            ConversationId="87634586123" SequenceNumber="5">

```

```

    <osci:Response>hZGr4fP+IGrt5</osci:Response>
    <osci:Challenge>345nchfcfoqc5dfg</osci:Challenge>
</osci:ControlBlock>
<osci:DesiredLanguages Id="XREF-0002" soap:mustUnderstand="1"
  soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
  LanguagesList="de en-US" />
<osci:QualityOfTimestamp Id="XREF-0003" soap:mustUnderstand="1"
  soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
  Service="creation" Quality="plain" />
<osci:QualityOfTimestamp Id="XREF-0004" soap:mustUnderstand="1"
  soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
  Service="reception" Quality="plain" />
<osci:forwardDelivery Id="XREF-0005" soap:mustUnderstand="1"
  soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
  <osci:ContentReceiver
    URI="http://www.osci.de/2002/04/osci#deployer" />
    <osci:MessageId>MessageId/base64codiert</osci:MessageId>
    <osci:Subject>Betreff der Zustellung</osci:Subject>
  </osci:forwardDelivery>
</soap:Header>
<soap:Body Id="XREF-0100">
  <osci:ContentPackage>
    <osci:ContentContainer>
      <osci:Content />
    </osci:ContentContainer>
  </osci:ContentPackage>
</soap:Body>
</soap:Envelope>

```

Ein Weiterleitungsauftrag (bzw. sein Root-Body-Part) genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Weiterleitungsaufträge:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.osci.de/2002/04/osci"
    schemaLocation="ForwardDelivery.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Weiterleitungsauftrag SOAP-Envelope
      $RCSfile: soapForwardDelivery.xsd,v $, $Revision: 1.4 $
    </xsd:documentation>
  </xsd:annotation>

```

```

</xsd:annotation>

<!-- ### restrictions ### -->

<xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

<!-- ### Envelope, Header und Body ### -->

<xsd:complexType name="Envelope" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Envelope">
      <xsd:sequence>
        <xsd:element ref="soap:Header" minOccurs="1" />
        <xsd:element ref="soap:Body" minOccurs="1" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="Header" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Header">
      <xsd:sequence>
        <xsd:element ref="osci:ControlBlock" />
        <xsd:element ref="osci:ClientSignature" minOccurs="0" />
        <xsd:element ref="osci:DesiredLanguages" />
        <xsd:element ref="osci:QualityOfTimestamp"
          minOccurs="2" maxOccurs="2" />
        <xsd:element ref="osci:forwardDelivery" />
        <xsd:element ref="osci:NonIntermediaryCertificates"
          minOccurs="0" />
        <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
          processContents="lax" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="Body" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Body">
      <xsd:sequence>
        <xsd:element ref="osci:ContentPackage" />
      </xsd:sequence>
      <xsd:attribute name="Id" type="xsd:ID" use="required" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

```

```
</xsd:redefine>
```

```
</xsd:schema>
```

- Schema für Weiterleitungsaufträge:

```
<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Weiterleitungsauftrag
      $RCSfile: ForwardDelivery.xsd,v $, $Revision: 1.4 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### derived types ### -->

  <xsd:complexType name="ControlBlockType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ControlBlockTemplate">
        <xsd:sequence>
          <xsd:element name="Response" type="xsd:string" minOccurs="0" />
          <xsd:element name="Challenge" type="xsd:string" minOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="ConversationId" type="osci:Number"
          use="optional" />
        <xsd:attribute name="SequenceNumber" type="osci:Number"
          use="required" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="forwardDeliveryType">
    <xsd:complexContent>
      <xsd:extension base="osci:DefaultHeaderBlockTemplate">
        <xsd:sequence>
          <xsd:element name="ContentReceiver">
            <xsd:complexType>
              <xsd:attribute name="URI" type="xsd:anyURI" />
            </xsd:complexType>
          </xsd:element>
```



```

        <xsd:element name="MessageId" type="osci:MessageIdType" />
        <xsd:element name="Subject" type="xsd:string" minOccurs="0" />
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="NonIntermediaryCertificatesType">
    <xsd:complexContent>
        <xsd:extension base="osci:NonIntermediaryCertificatesTemplate" />
    </xsd:complexContent>
</xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="ClientSignature" type="osci:SignatureBlockType" />

<xsd:element name="DesiredLanguages" type="osci:DesiredLanguagesType" />

<xsd:element name="QualityOfTimestamp"
    type="osci:QualityOfTimestampType" />

<xsd:element name="forwardDelivery" type="osci:forwardDeliveryType" />

<xsd:element name="NonIntermediaryCertificates"
    type="osci:NonIntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

<xsd:element name="ContentPackage" type="osci:ContentPackageType" />

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)
- Ausprägung von XML-Encryption(siehe Abschnitt 6.5)

#### 6.6.14 Weiterleitungsantwort

Eine Weiterleitungsantwort wird als SOAP-Envelope realisiert.

Das folgende Beispiel erläutert den Aufbau einer Weiterleitungsantwort:

```

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

<!-- XML-Beispiel OSCI 1.2 Weiterleitungsantwort -->
<!-- $RCSfile: ResponseToForwardDelivery.xml,v $, $Revision: 1.6 $ -->

```

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    soapResponseToForwardDelivery.xsd
    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">
  <soap:Header>
    <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
      ConversationId="87634586123" SequenceNumber="6"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
      <osci:Response>345nchfcfoqc5dfg</osci:Response>
      <osci:Challenge>n5vl2dfie9uf6d</osci:Challenge>
    </osci:ControlBlock>
    <osci:responseToForwardDelivery Id="XREF-0005" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
      <osci:Feedback>
        <osci:Entry xml:lang="de">
          <osci:Code>0801</osci:Code>
          <osci:Text>Auftrag ausgeführt, Dialog weiterhin
geöffnet</osci:Text>
        </osci:Entry>
      </osci:Feedback>
      <osci:ProcessCardBundle>
        <osci:MessageId>MessageId/Base64codiert</osci:MessageId>
        <osci:ProcessCard
          RecentModification="2002-04-24T12:15:01.516-01:00">
          <osci:Creation>
            <osci:Plain>2002-04-23T23:41:05.527-01:00</osci:Plain>
          </osci:Creation>
          <osci:Forwarding>
            <osci:Plain>2002-04-24T12:15:01.516-01:00</osci:Plain>
          </osci:Forwarding>
          <osci:Reception>
            <osci:Plain>2002-04-24T12:17:12.169-01:00</osci:Plain>
          </osci:Reception>
          <osci:Subject>Betreff der Zustellung</osci:Subject>
        </osci:ProcessCard>
        <osci:InspectionReport />
      </osci:ProcessCardBundle>
    </osci:responseToForwardDelivery>
  </soap:Header>
  <soap:Body Id="XREF-0100" />
</soap:Envelope>

```

Ist der Intermediär in der Lage zu der mittels *Weiterleitungsauftrag* eingereichten Zustellung einen Laufzettel zu erzeugen, so wird dieser innerhalb der *Weiterleitungsantwort* an den Benutzer gesendet.

Eine Weiterleitungsantwort genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Weiterleitungsantworten:

```
<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.osci.de/2002/04/osci"
    schemaLocation="ResponseToForwardDelivery.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Weiterleitungsantwort SOAP-Envelope
      $RCSfile: soapResponseToForwardDelivery.xsd,v $, $Revision: 1.4 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### restrictions ### -->

  <xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/"

  <!-- ### Envelope, Header und Body ### -->

  <xsd:complexType name="Envelope" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Envelope">
        <xsd:sequence>
          <xsd:element ref="soap:Header" minOccurs="1" />
          <xsd:element ref="soap:Body" minOccurs="1" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Header" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Header">
        <xsd:sequence>
          <xsd:element ref="osci:ControlBlock" />
          <xsd:element ref="osci:SupplierSignature" minOccurs="0" />
          <xsd:element ref="osci:responseToForwardDelivery" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
```

```

        <xsd:element ref="osci:IntermediaryCertificates"
            minOccurs="0" />
        <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
            processContents="lax" />
    </xsd:sequence>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="Body" >
    <xsd:complexContent>
        <xsd:restriction base="soap:Body">
            <xsd:attribute name="Id" type="xsd:ID" use="required" />
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- Schema für Weiterleitungsantworten:

```

<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    xmlns:osci="http://www.osci.de/2002/04/osci"
    attributeFormDefault="unqualified"
    elementFormDefault="qualified">

    <xsd:include schemaLocation="order.xsd" />

    <xsd:annotation>
        <xsd:documentation xml:lang="de">
            OSCI 1.2 - Weiterleitungsantwort
            $RCSfile: ResponseToForwardDelivery.xsd,v $, $Revision: 1.3 $
        </xsd:documentation>
    </xsd:annotation>

    <!-- ### derived types ### -->

    <xsd:complexType name="ControlBlockType">
        <xsd:complexContent>
            <xsd:restriction base="osci:ControlBlockTemplate">
                <xsd:sequence>
                    <xsd:element name="Response" type="xsd:string" minOccurs="1" />
                    <xsd:element name="Challenge" type="xsd:string" minOccurs="0" />
                </xsd:sequence>
            </xsd:restriction>
        </xsd:complexContent>
    </xsd:complexType>

```

```

    </xsd:sequence>
    <xsd:attribute name="ConversationId" type="osci:Number"
      use="required" />
    <xsd:attribute name="SequenceNumber" type="osci:Number"
      use="required" />
  </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ProcessCardBundleType">
  <xsd:complexContent>
    <xsd:restriction base="osci:ProcessCardBundleTemplate">
      <xsd:sequence>
        <xsd:element name="MessageId" type="osci:MessageIdType" />
        <xsd:element name="ProcessCard" type="osci:ProcessCardType" />
        <xsd:element name="InspectionReport"
          type="osci:InspectionReportType" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ProcessCardType">
  <xsd:complexContent>
    <xsd:restriction base="osci:ProcessCardTemplate">
      <xsd:sequence>
        <xsd:element name="Creation" type="osci:TimestampType" />
        <xsd:element name="Forwarding"
          type="osci:TimestampType" minOccurs="0" />
        <xsd:element name="Reception" type="osci:TimestampType"
          minOccurs="0" />
        <xsd:element name="Subject" type="xsd:string" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="responseToForwardDeliveryType">
  <xsd:complexContent>
    <xsd:extension base="osci:DefaultHeaderBlockTemplate">
      <xsd:sequence>
        <xsd:element name="Feedback" type="osci:FeedbackType" />
        <xsd:element name="ProcessCardBundle"
          type="osci:ProcessCardBundleType" minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="IntermediaryCertificatesType">

```

```

<xsd:complexContent>
  <xsd:restriction base="osci:IntermediaryCertificatesTemplate">
    <xsd:sequence>
      <xsd:element name="SignatureCertificateIntermediary"
        type="osci:CertificateType" minOccurs="0" />
    </xsd:sequence>
  </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="SupplierSignature" type="osci:SignatureBlockType" />

<xsd:element name="responseToForwardDelivery"
  type="osci:responseToForwardDeliveryType" />

<xsd:element name="IntermediaryCertificates"
  type="osci:IntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)

### 6.6.15 Annahmearauftrag

Ein Annahmearauftrag wird entweder als SOAP-Envelope realisiert oder (wenn er Attachments enthält) als SOAP-Message-Package im Sinne von [SOAP A].

Das folgende Beispiel erläutert den Aufbau eines Annahmearauftrags:

```

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

<!-- XML-Beispiel OSCI 1.2 Annahmearauftrag -->
<!-- $RCSfile: AcceptDelivery.xml,v $, $Revision: 1.7 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    soapAcceptDelivery.xsd
    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">

```

```

<soap:Header>
  <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
    soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
    <osci:Challenge>xaLW8jIJcCAwEA</osci:Challenge>
  </osci:ControlBlock>
  <osci:DesiredLanguages Id="XREF-0002" soap:mustUnderstand="1"
    soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
    LanguagesList="de en-US" />
  <osci:acceptDelivery Id="XREF-0005" soap:mustUnderstand="1"
    soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
    <osci:ProcessCardBundle>
      <osci:MessageId>MessageId/Base64codiert</osci:MessageId>
      <osci:ProcessCard
        RecentModification="2002-04-24T12:15:01.516-01:00">
        <osci:Creation>
          <osci:Plain>2002-04-23T23:41:05.527-01:00</osci:Plain>
        </osci:Creation>
        <osci:Forwarding>
          <osci:Plain>2002-04-24T12:15:01.516-01:00</osci:Plain>
        </osci:Forwarding>
        <osci:Subject>Betreff der Zustellung</osci:Subject>
      </osci:ProcessCard>
      <osci:InspectionReport />
    </osci:ProcessCardBundle>
  </osci:acceptDelivery>
</soap:Header>
<soap:Body Id="XREF-0100">
  <osci:ContentPackage>
    <osci:ContentContainer>
      <osci:Content />
    </osci:ContentContainer>
  </osci:ContentPackage>
</soap:Body>
</soap:Envelope>

```

Ein Annahmearauftrag (bzw. sein Root-Body-Part) genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Annahmearaufträge:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.osci.de/2002/04/osci"
    schemaLocation="AcceptDelivery.xsd" />

```

```

<xsd:annotation>
  <xsd:documentation xml:lang="de">
    OSCI 1.2 - Annahmefauftrag SOAP-Envelope
    $RCSfile: soapAcceptDelivery.xsd,v $, $Revision: 1.4 $
  </xsd:documentation>
</xsd:annotation>

<!-- ### restrictions ### -->

<xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

  <!-- ### Envelope, Header und Body ### -->

  <xsd:complexType name="Envelope" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Envelope">
        <xsd:sequence>
          <xsd:element ref="soap:Header" minOccurs="1" />
          <xsd:element ref="soap:Body" minOccurs="1" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Header" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Header">
        <xsd:sequence>
          <xsd:element ref="osci:ControlBlock" />
          <xsd:element ref="osci:ClientSignature" minOccurs="0" />
          <xsd:element ref="osci:DesiredLanguages" />
          <xsd:element ref="osci:acceptDelivery" />
          <xsd:element ref="osci:IntermediaryCertificates"
            minOccurs="0" />
          <xsd:element ref="osci:NonIntermediaryCertificates"
            minOccurs="0" />
          <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
            processContents="lax" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Body" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Body">
        <xsd:sequence>
          <xsd:element ref="osci:ContentPackage" />

```



```

        </xsd:sequence>
        <xsd:attribute name="Id" type="xsd:ID" use="required" />
    </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- Schema für Annahmeforderungen:

```

<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Annahmeforderung
      $RCSfile: AcceptDelivery.xsd,v $, $Revision: 1.3 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### derived types ### -->

  <xsd:complexType name="ControlBlockType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ControlBlockTemplate">
        <xsd:sequence>
          <xsd:element name="Response" type="xsd:string"
            minOccurs="0" maxOccurs="0"/>
          <xsd:element name="Challenge" type="xsd:string" minOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="ConversationId" type="osci:Number"
          use="prohibited" />
        <xsd:attribute name="SequenceNumber" type="osci:Number"
          use="prohibited" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="ProcessCardBundleType">
    <xsd:complexContent>

```

```
<xsd:restriction base="osci:ProcessCardBundleTemplate">
  <xsd:sequence>
    <xsd:element name="MessageId" type="osci:MessageIdType" />
    <xsd:element name="ProcessCard" type="osci:ProcessCardType" />
    <xsd:element name="InspectionReport"
      type="osci:InspectionReportType" />
  </xsd:sequence>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ProcessCardType">
  <xsd:complexContent>
    <xsd:restriction base="osci:ProcessCardTemplate">
      <xsd:sequence>
        <xsd:element name="Creation" type="osci:TimestampType" />
        <xsd:element name="Forwarding"
          type="osci:TimestampType" minOccurs="1" />
        <xsd:element name="Subject" type="xsd:string" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="acceptDeliveryType">
  <xsd:complexContent>
    <xsd:extension base="osci:DefaultHeaderBlockTemplate">
      <xsd:sequence>
        <xsd:element name="ProcessCardBundle"
          type="osci:ProcessCardBundleType" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="IntermediaryCertificatesType">
  <xsd:complexContent>
    <xsd:restriction base="osci:IntermediaryCertificatesTemplate">
      <xsd:sequence>
        <xsd:element name="CipherCertificateIntermediary"
          type="osci:CertificateType" minOccurs="0" />
        <xsd:element name="SignatureCertificateIntermediary"
          type="osci:CertificateType" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="NonIntermediaryCertificatesType">
  <xsd:complexContent>
```

```

        <xsd:extension base="osci:NonIntermediaryCertificatesTemplate" />
    </xsd:complexContent>
</xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="ClientSignature" type="osci:SignatureBlockType" />

<xsd:element name="DesiredLanguages" type="osci:DesiredLanguagesType" />

<xsd:element name="acceptDelivery" type="osci:acceptDeliveryType" />

<xsd:element name="IntermediaryCertificates"
    type="osci:IntermediaryCertificatesType" />

<xsd:element name="NonIntermediaryCertificates"
    type="osci:NonIntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

<xsd:element name="ContentPackage" type="osci:ContentPackageType" />

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)
- Ausprägung von XML-Encryption(siehe Abschnitt 6.5)

### 6.6.16 Annahmeantwort

Eine Annahmeantwort wird als SOAP-Envelope realisiert.

Das folgende Beispiel erläutert den Aufbau einer Annahmeantwort:

```

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

<!-- XML-Beispiel OSCI 1.2 Annahmeantwort -->
<!-- $RCSfile: ResponseToAcceptDelivery.xml,v $, $Revision: 1.4 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    xmlns:osci="http://www.osci.de/2002/04/osci"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
        soapResponseToAcceptDelivery.xsd
        http://www.w3.org/2000/09/xmldsig# oscisig.xsd
        http://www.w3.org/2001/04/xmlenc# oscienc.xsd">

```

```

<soap:Header>
  <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
    soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
    ConversationId="87634586123" SequenceNumber="0">
    <osci:Response>xaLW8jIJcCAwEA</osci:Response>
  </osci:ControlBlock>
  <osci:responseToAcceptDelivery Id="XREF-0005" soap:mustUnderstand="1"
    soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
    <osci:Feedback>
      <osci:Entry xml:lang="de">
        <osci:Code>0800</osci:Code>
        <osci:Text>Auftrag ausgeführt, Dialog beendet</osci:Text>
      </osci:Entry>
    </osci:Feedback>
  </osci:responseToAcceptDelivery>
</soap:Header>
<soap:Body Id="XREF-0100" />
</soap:Envelope>

```

Eine Annahmeantwort genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Annahmeantworten:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.osci.de/2002/04/osci"
    schemaLocation="ResponseToAcceptDelivery.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Annahmeantwort SOAP-Envelope
      $RCSfile: soapResponseToAcceptDelivery.xsd,v $, $Revision: 1.4 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### restrictions ### -->

  <xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/"

    <!-- ### Envelope, Header und Body ### -->

    <xsd:complexType name="Envelope" >
      <xsd:complexContent>

```

```

    <xsd:restriction base="soap:Envelope">
      <xsd:sequence>
        <xsd:element ref="soap:Header" minOccurs="1" />
        <xsd:element ref="soap:Body" minOccurs="1" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="Header" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Header">
      <xsd:sequence>
        <xsd:element ref="osci:ControlBlock" />
        <xsd:element ref="osci:SupplierSignature" minOccurs="0" />
        <xsd:element ref="osci:responseToAcceptDelivery" />
        <xsd:element ref="osci:NonIntermediaryCertificates"
          minOccurs="0" />
        <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
          processContents="lax" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="Body" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Body">
      <xsd:attribute name="Id" type="xsd:ID" use="required" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- Schema für Annahmeantworten:

```

<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

```

```
<xsd:annotation>
  <xsd:documentation xml:lang="de">
    OSCI 1.2 - Annahmeantwort
    $RCSfile: ResponseToAcceptDelivery.xsd,v $, $Revision: 1.2 $
  </xsd:documentation>
</xsd:annotation>

<!-- ### derived types ### -->

<xsd:complexType name="ControlBlockType">
  <xsd:complexContent>
    <xsd:restriction base="osci:ControlBlockTemplate">
      <xsd:sequence>
        <xsd:element name="Response" type="xsd:string" minOccurs="1" />
        <xsd:element name="Challenge" type="xsd:string" minOccurs="0" />
      </xsd:sequence>
      <xsd:attribute name="ConversationId" type="osci:Number"
        use="required" />
      <xsd:attribute name="SequenceNumber" type="osci:Number"
        use="required" fixed="0" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="responseToAcceptDeliveryType">
  <xsd:complexContent>
    <xsd:extension base="osci:DefaultHeaderBlockTemplate">
      <xsd:sequence>
        <xsd:element name="Feedback" type="osci:FeedbackType" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="NonIntermediaryCertificatesType">
  <xsd:complexContent>
    <xsd:restriction base="osci:NonIntermediaryCertificatesTemplate">
      <xsd:sequence>
        <xsd:element name="SignatureCertificateAddressee"
          type="osci:CertificateType" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->
```

```

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="SupplierSignature" type="osci:SignatureBlockType" />

<xsd:element name="responseToAcceptDelivery"
  type="osci:responseToAcceptDeliveryType" />

<xsd:element name="NonIntermediaryCertificates"
  type="osci:NonIntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)

### 6.6.17 Abwicklungsauftrag

Ein Abwicklungsauftrag wird entweder als SOAP-Envelope realisiert oder (wenn er Attachments enthält) als SOAP-Message-Package im Sinne von [SOAP A].

Das folgende Beispiel erläutert den Aufbau eines Abwicklungsauftrags:

```

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

<!-- XML-Beispiel OSCI 1.2 Abwicklungsauftrag -->
<!-- $RCSfile: MediateDelivery.xml,v $, $Revision: 1.7 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    soapMediateDelivery.xsd
    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">
  <soap:Header>
    <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
      ConversationId="87634586123" SequenceNumber="5"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
      <osci:Response>AAQCAEMmCZtuMFQxDSa</osci:Response>
      <osci:Challenge>345nchfcfoqc5dfg</osci:Challenge>
    </osci:ControlBlock>
    <osci:ClientSignature Id="XREF-0001" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
      <ds:Signature>
        <ds:SignedInfo>
          <ds:CanonicalizationMethod
            Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
          <ds:SignatureMethod

```

```
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
<ds:Reference URI="#XREF-0000">
  <ds:Transforms>
    <ds:Transform
      Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"
    />
  </ds:Transforms>
  <ds:DigestMethod
    Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
  <ds:DigestValue>Hash/XREF0000/Base64codiert</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#XREF-0002">
  <ds:Transforms>
    <ds:Transform
      Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"
    />
  </ds:Transforms>
  <ds:DigestMethod
    Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
  <ds:DigestValue>Hash/XREF0002/Base64codiert</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#XREF-0005">
  <ds:Transforms>
    <ds:Transform
      Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"
    />
  </ds:Transforms>
  <ds:DigestMethod
    Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
  <ds:DigestValue>Hash/XREF0005/Base64codiert</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#XREF-0020">
  <ds:Transforms>
    <ds:Transform
      Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"
    />
  </ds:Transforms>
  <ds:DigestMethod
    Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
  <ds:DigestValue>Hash/XREF0020/Base64codiert</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#XREF-0100">
  <ds:Transforms>
    <ds:Transform
      Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"
    />
  </ds:Transforms>
  <ds:DigestMethod
    Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
  <ds:DigestValue>Hash/XREF0100/Base64codiert</ds:DigestValue>
```



```

        </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>Signatur/Base64codiert</ds:SignatureValue>
    <ds:KeyInfo>
        <ds:RetrievalMethod URI="#XREF-9999" />
    </ds:KeyInfo>
</ds:Signature>
</osci:ClientSignature>
<osci:DesiredLanguages Id="XREF-0002" soap:mustUnderstand="1"
    soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
    LanguagesList="de en-US" />
<osci:mediateDelivery Id="XREF-0005" soap:mustUnderstand="1"
    soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
    <osci:ContentReceiver
        URI="http://www.osci.de/2002/04/osci#deployer" />
</osci:mediateDelivery>
<osci:NonIntermediaryCertificates Id="XREF-0020"
soap:mustUnderstand="1"
    soap:actor="http://www.w3.org/2001/12/soap-envelope/actor/none">
    <osci:SignatureCertificateOriginator Id="XREF-9999">
        <ds:X509Data>
            <ds:X509Certificate>
                Signierzertifikat/Absender/Base64codiert
            </ds:X509Certificate>
        </ds:X509Data>
    </osci:SignatureCertificateOriginator>
</osci:NonIntermediaryCertificates>
</soap:Header>
<soap:Body Id="XREF-0100">
    <osci:ContentPackage>
        <osci:ContentContainer>
            <osci:Content />
        </osci:ContentContainer>
    </osci:ContentPackage>
</soap:Body>
</soap:Envelope>

```

Ein Abwicklungsauftrag (bzw. sein Root-Body-Part) genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Abwicklungsaufträge:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:osci="http://www.osci.de/2002/04/osci"
    attributeFormDefault="unqualified"
    elementFormDefault="qualified">

```

```

<xsd:import namespace="http://www.osci.de/2002/04/osci"
  schemaLocation="MediateDelivery.xsd" />

<xsd:annotation>
  <xsd:documentation xml:lang="de">
    OSCI 1.2 - Abwicklungsauftrag SOAP-Envelope
    $RCSfile: soapMediateDelivery.xsd,v $, $Revision: 1.4 $
  </xsd:documentation>
</xsd:annotation>

<!-- ### restrictions ### -->

<xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

  <!-- ### Envelope, Header und Body ### -->

  <xsd:complexType name="Envelope" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Envelope">
        <xsd:sequence>
          <xsd:element ref="soap:Header" minOccurs="1" />
          <xsd:element ref="soap:Body" minOccurs="1" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Header" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Header">
        <xsd:sequence>
          <xsd:element ref="osci:ControlBlock" />
          <xsd:element ref="osci:ClientSignature" minOccurs="0" />
          <xsd:element ref="osci:DesiredLanguages" />
          <xsd:element ref="osci:QualityOfTimestamp"
            minOccurs="0" maxOccurs="2" />
          <xsd:element ref="osci:mediateDelivery" />
          <xsd:element ref="osci:NonIntermediaryCertificates"
            minOccurs="0" />
          <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
            processContents="lax" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Body" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Body">

```

```

    <xsd:sequence>
      <xsd:element ref="osci:ContentPackage" />
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:ID" use="required" />
  </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- **Schema für Abwicklungsaufträge:**

```

<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Abwicklungsauftrag
      $RCSfile: MediateDelivery.xsd,v $, $Revision: 1.5 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### derived types ### -->

  <xsd:complexType name="ControlBlockType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ControlBlockTemplate">
        <xsd:sequence>
          <xsd:element name="Response" type="xsd:string" minOccurs="1" />
          <xsd:element name="Challenge" type="xsd:string" minOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="ConversationId" type="osci:Number"
          use="required" />
        <xsd:attribute name="SequenceNumber" type="osci:Number"
          use="required" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="mediateDeliveryType">

```

```

<xsd:complexContent>
  <xsd:extension base="osci:DefaultHeaderBlockTemplate">
    <xsd:sequence>
      <xsd:element name="ContentReceiver">
        <xsd:complexType>
          <xsd:attribute name="URI" type="xsd:anyURI" />
        </xsd:complexType>
      </xsd:element>
      <xsd:sequence minOccurs="0">
        <xsd:element name="MessageId" type="osci:MessageIdType" />
        <xsd:element name="Subject" type="xsd:string" minOccurs="0" />
      </xsd:sequence>
    </xsd:sequence>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="NonIntermediaryCertificatesType">
  <xsd:complexContent>
    <xsd:extension base="osci:NonIntermediaryCertificatesTemplate" />
  </xsd:complexContent>
</xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="ClientSignature" type="osci:SignatureBlockType" />

<xsd:element name="DesiredLanguages" type="osci:DesiredLanguagesType" />

<xsd:element name="QualityOfTimestamp"
  type="osci:QualityOfTimestampType" />

<xsd:element name="mediateDelivery" type="osci:mediateDeliveryType" />

<xsd:element name="NonIntermediaryCertificates"
  type="osci:NonIntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

<xsd:element name="ContentPackage" type="osci:ContentPackageType" />

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)
- Ausprägung von XML-Encryption(siehe Abschnitt 6.5)

### 6.6.18 Abwicklungsantwort

Eine Abwicklungsantwort wird als entweder als SOAP-Envelope realisiert oder (wenn sie Attachments enthält) als SOAP-Message-Package im Sinne von [SOAP A].

Das folgende Beispiel erläutert den Aufbau einer Abwicklungsantwort:

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

<!-- XML-Beispiel OSCI 1.2 Abwicklungsantwort -->
<!-- $RCSfile: ResponseToMediateDelivery.xml,v $, $Revision: 1.6 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    soapResponseToMediateDelivery.xsd
    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">
  <soap:Header>
    <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
      ConversationId="87634586123" SequenceNumber="6">
      <osci:Response>345nchfcfoqc5dfg</osci:Response>
      <osci:Challenge>fie9uf6dGGFRGm3</osci:Challenge>
    </osci:ControlBlock>
    <osci:SupplierSignature Id="XREF-0001" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
      <ds:Signature>
        <ds:SignedInfo>
          <ds:CanonicalizationMethod
            Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
          <ds:SignatureMethod
            Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
          <ds:Reference URI="#XREF-0000">
            <ds:Transforms>
              <ds:Transform
                Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"
              />
            </ds:Transforms>
          </ds:Reference>
          <ds:Reference URI="#XREF-0005">
            <ds:Transforms>
              <ds:Transform
                Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"
              />
            </ds:Transforms>
          </ds:Reference>
          <ds:DigestMethod
            Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
          <ds:DigestValue>Hash/XREF0000/Base64codiert</ds:DigestValue>
        </ds:SignedInfo>
      </ds:Signature>
    </osci:SupplierSignature>
  </soap:Header>
  </soap:Envelope>
```

```

        </ds:Transforms>
        <ds:DigestMethod
          Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
        <ds:DigestValue>Hash/XREF0005/Base64codiert</ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="#XREF-0010">
        <ds:Transforms>
          <ds:Transform
            Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"
/>
          </ds:Transforms>
          <ds:DigestMethod
            Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
          <ds:DigestValue>Hash/XREF0010/Base64codiert</ds:DigestValue>
        </ds:Reference>
      <ds:Reference URI="#XREF-0100">
        <ds:Transforms>
          <ds:Transform
            Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"
/>
          </ds:Transforms>
          <ds:DigestMethod
            Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
          <ds:DigestValue>Hash/XREF0100/Base64codiert</ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>Signatur/Base64codiert</ds:SignatureValue>
      <ds:KeyInfo>
        <ds:RetrievalMethod URI="#XREF-9999" />
      </ds:KeyInfo>
    </ds:Signature>
  </osci:SupplierSignature>
  <osci:responseToMediateDelivery Id="XREF-0005" soap:mustUnderstand="1"
  soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
    <osci:Feedback>
      <osci:Entry xml:lang="de">
        <osci:Code>0801</osci:Code>
        <osci:Text>Auftrag ausgeführt, Dialog weiterhin
geöffnet</osci:Text>
      </osci:Entry>
    </osci:Feedback>
  </osci:responseToMediateDelivery>
  <osci:IntermediaryCertificates Id="XREF-0010" soap:mustUnderstand="1"
  soap:actor="http://www.w3.org/2001/12/soap-envelope/actor/none">
    <osci:SignatureCertificateIntermediary Id="XREF-9999">
      <ds:X509Data>
        <ds:X509Certificate>
          Signierzertifikat/Intermediaer/Base64codiert
        </ds:X509Certificate>
      </ds:X509Data>
    </osci:SignatureCertificateIntermediary>
  </osci:IntermediaryCertificates>

```

```

        </osci:SignatureCertificateIntermediary>
    </osci:IntermediaryCertificates>
</soap:Header>
<soap:Body Id="XREF-0100">
    <osci:ContentPackage>
        <osci:ContentContainer>
            <osci:Content />
        </osci:ContentContainer>
    </osci:ContentPackage>
</soap:Body>
</soap:Envelope>

```

Das Element `osci:RequestProcessCardBundle` enthält den aktuellen Laufzettel zu der Zustellung, die der Benutzer innerhalb des *Abwicklungsauftrags* an den Intermediär gesendet hat.

Es ist nur vorhanden, wenn alle folgenden Bedingungen zutreffen:

- Der Benutzer hat im *Abwicklungsauftrag* eine Messageld gesendet.
- Der Intermediär war in der Lage einen Laufzettel zu der Zustellung im *Abwicklungsauftrag* des Benutzers zu erstellen.

Das Element `osci:ReplyProcessCardBundle` enthält den aktuellen Laufzettel zu der Zustellung, die der Dienstanbieter innerhalb der *Bearbeitungsantwort* an den Intermediär gesendet hat.

Es ist nur vorhanden, wenn mindestens die folgenden Bedingungen zutreffen:

- Der Dienstanbieter hat in der *Bearbeitungsantwort* eine Zustellung mit Message-ID gesendet.
- Der Intermediär war in der Lage einen Laufzettel zu der Zustellung des Dienstanbieters zu erstellen.

Eine Abwicklungsantwort (bzw. ihr Root-Body-Part) genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Abwicklungsantworten:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.osci.de/2002/04/osci"
    schemaLocation="ResponseToMediateDelivery.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Abwicklungsantwort SOAP-Envelope
      $RCSfile: soapResponseToMediateDelivery.xsd,v $, $Revision: 1.5 $
    </xsd:documentation>
  </xsd:annotation>

```

```

</xsd:annotation>

<!-- ### restrictions ### -->

<xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

  <!-- ### Envelope, Header und Body ### -->

  <xsd:complexType name="Envelope" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Envelope">
        <xsd:sequence>
          <xsd:element ref="soap:Header" minOccurs="1" />
          <xsd:element ref="soap:Body" minOccurs="1" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Header" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Header">
        <xsd:sequence>
          <xsd:element ref="osci:ControlBlock" />
          <xsd:element ref="osci:SupplierSignature" minOccurs="0" />
          <xsd:element ref="osci:responseToMediateDelivery" />
          <xsd:element ref="osci:IntermediaryCertificates"
            minOccurs="0" />
          <xsd:element ref="osci:NonIntermediaryCertificates"
            minOccurs="0" />
          <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
            processContents="lax" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Body" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Body">
        <xsd:sequence>
          <xsd:element ref="osci:ContentPackage" minOccurs="0" />
        </xsd:sequence>
        <xsd:attribute name="Id" type="xsd:ID" use="required" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

</xsd:redefine>

```



```
</xsd:schema>
```

- Schema für Abwicklungsantworten:

```
<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Abwicklungsantwort
      $RCSfile: ResponseToMediateDelivery.xsd,v $, $Revision: 1.5 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### derived types ### -->

  <xsd:complexType name="ControlBlockType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ControlBlockTemplate">
        <xsd:sequence>
          <xsd:element name="Response" type="xsd:string" minOccurs="1" />
          <xsd:element name="Challenge" type="xsd:string" minOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="ConversationId" type="osci:Number"
          use="required" />
        <xsd:attribute name="SequenceNumber" type="osci:Number"
          use="required" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="RequestProcessCardBundleType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ProcessCardBundleTemplate">
        <xsd:sequence>
          <xsd:element name="MessageId" type="osci:MessageIdType" />
          <xsd:element name="ProcessCard"
            type="osci:RequestProcessCardType" />
          <xsd:element name="InspectionReport"
            type="osci:InspectionReportType" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
```

```
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="RequestProcessCardType">
  <xsd:complexContent>
    <xsd:restriction base="osci:ProcessCardTemplate">
      <xsd:sequence>
        <xsd:element name="Creation" type="osci:TimestampType" />
        <xsd:element name="Forwarding"
          type="osci:TimestampType" minOccurs="1" />
        <xsd:element name="Reception" type="osci:TimestampType"
          minOccurs="1" />
        <xsd:element name="Subject" type="xsd:string" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ReplyProcessCardBundleType">
  <xsd:complexContent>
    <xsd:restriction base="osci:ProcessCardBundleTemplate">
      <xsd:sequence>
        <xsd:element name="MessageId" type="osci:MessageIdType" />
        <xsd:element name="ProcessCard"
          type="osci:ReplyProcessCardType" />
        <xsd:element name="InspectionReport"
          type="osci:InspectionReportType" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ReplyProcessCardType">
  <xsd:complexContent>
    <xsd:restriction base="osci:ProcessCardTemplate">
      <xsd:sequence>
        <xsd:element name="Creation" type="osci:TimestampType" />
        <xsd:element name="Forwarding" type="osci:TimestampType"
          minOccurs="1" />
        <xsd:element name="Subject" type="xsd:string" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="responseToMediateDeliveryType">
  <xsd:complexContent>
    <xsd:extension base="osci:DefaultHeaderBlockTemplate">
      <xsd:sequence>
```

```
<xsd:element name="Feedback" type="osci:FeedbackType" />
<xsd:sequence minOccurs="0">
  <xsd:element name="RequestProcessCardBundle"
    type="osci:RequestProcessCardBundleType" />
  <xsd:element name="ReplyProcessCardBundle"
    type="osci:ReplyProcessCardBundleType" minOccurs="0" />
</xsd:sequence>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="IntermediaryCertificatesType">
  <xsd:complexContent>
    <xsd:restriction base="osci:IntermediaryCertificatesTemplate">
      <xsd:sequence>
        <xsd:element name="SignatureCertificateIntermediary"
          type="osci:CertificateType" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="NonIntermediaryCertificatesType">
  <xsd:complexContent>
    <xsd:extension base="osci:NonIntermediaryCertificatesTemplate" />
  </xsd:complexContent>
</xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="SupplierSignature" type="osci:SignatureBlockType" />

<xsd:element name="responseToMediateDelivery"
  type="osci:responseToMediateDeliveryType" />

<xsd:element name="IntermediaryCertificates"
  type="osci:IntermediaryCertificatesType" />

<xsd:element name="NonIntermediaryCertificates"
  type="osci:NonIntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

<xsd:element name="ContentPackage" type="osci:ContentPackageType" />
```

```
</xsd:schema>
```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)

### 6.6.19 Bearbeitungsauftrag

Ein Bearbeitungsauftrag wird entweder als SOAP-Envelope realisiert oder (wenn er Attachments enthält) als SOAP-Message-Package im Sinne von [SOAP A].

Das folgende Beispiel erläutert den Aufbau eines Bearbeitungsauftrags:

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

<!-- XML-Beispiel OSCI 1.2 Bearbeitungsauftrag -->
<!-- $RCSfile: ProcessDelivery.xml,v $, $Revision: 1.7 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    soapProcessDelivery.xsd
    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">
  <soap:Header>
    <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
      <osci:Challenge>xaLW8jIJcCAwEA</osci:Challenge>
    </osci:ControlBlock>
    <osci:DesiredLanguages Id="XREF-0002" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next"
      LanguagesList="de en-US" />
    <osci:processDelivery Id="XREF-0005" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
    </osci:processDelivery>
  </soap:Header>
  <soap:Body Id="XREF-0100">
    <osci:ContentPackage>
      <osci:ContentContainer>
        <osci:Content />
      </osci:ContentContainer>
    </osci:ContentPackage>
  </soap:Body>
</soap:Envelope>
```

Ein Bearbeitungsauftrag (bzw. sein Root-Body-Part) genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Bearbeitungsaufträge:

```
<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```

xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:osci="http://www.osci.de/2002/04/osci"
attributeFormDefault="unqualified"
elementFormDefault="qualified">

<xsd:import namespace="http://www.osci.de/2002/04/osci"
  schemaLocation="ProcessDelivery.xsd" />

<xsd:annotation>
  <xsd:documentation xml:lang="de">
    OSCI 1.2 - Bearbeitungsauftrag SOAP-Envelope
    $RCSfile: soapProcessDelivery.xsd,v $, $Revision: 1.4 $
  </xsd:documentation>
</xsd:annotation>

<!-- ### restrictions ### -->

<xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

  <!-- ### Envelope, Header und Body ### -->

  <xsd:complexType name="Envelope" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Envelope">
        <xsd:sequence>
          <xsd:element ref="soap:Header" minOccurs="1" />
          <xsd:element ref="soap:Body" minOccurs="1" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Header" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Header">
        <xsd:sequence>
          <xsd:element ref="osci:ControlBlock" />
          <xsd:element ref="osci:ClientSignature" minOccurs="0" />
          <xsd:element ref="osci:DesiredLanguages" />
          <xsd:element ref="osci:processDelivery" />
          <xsd:element ref="osci:IntermediaryCertificates"
            minOccurs="0" />
          <xsd:element ref="osci:NonIntermediaryCertificates"
            minOccurs="0" />
          <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
            processContents="lax" />
        </xsd:sequence>

```

```

        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="Body" >
    <xsd:complexContent>
        <xsd:restriction base="soap:Body">
            <xsd:sequence>
                <xsd:element ref="osci:ContentPackage" />
            </xsd:sequence>
            <xsd:attribute name="Id" type="xsd:ID" use="required" />
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- Schema für Bearbeitungsaufträge:

```

<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    xmlns:osci="http://www.osci.de/2002/04/osci"
    attributeFormDefault="unqualified"
    elementFormDefault="qualified">

    <xsd:include schemaLocation="order.xsd" />

    <xsd:annotation>
        <xsd:documentation xml:lang="de">
            OSCI 1.2 - Bearbeitungsauftrag
            $RCSfile: ProcessDelivery.xsd,v $, $Revision: 1.4 $
        </xsd:documentation>
    </xsd:annotation>

    <!-- ### derived types ### -->

    <xsd:complexType name="ControlBlockType">
        <xsd:complexContent>
            <xsd:restriction base="osci:ControlBlockTemplate">
                <xsd:sequence>
                    <xsd:element name="Response" type="xsd:string"
                        minOccurs="0" maxOccurs="0"/>
                    <xsd:element name="Challenge" type="xsd:string" minOccurs="1" />
                </xsd:sequence>
                <xsd:attribute name="ConversationId" type="osci:Number"

```

```
        use="prohibited" />
        <xsd:attribute name="SequenceNumber" type="osci:Number"
        use="prohibited" />
    </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ProcessCardBundleType">
    <xsd:complexContent>
        <xsd:restriction base="osci:ProcessCardBundleTemplate">
            <xsd:sequence>
                <xsd:element name="MessageId" type="osci:MessageIdType" />
                <xsd:element name="ProcessCard" type="osci:ProcessCardType" />
                <xsd:element name="InspectionReport"
                type="osci:InspectionReportType" />
            </xsd:sequence>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ProcessCardType">
    <xsd:complexContent>
        <xsd:restriction base="osci:ProcessCardTemplate">
            <xsd:sequence>
                <xsd:element name="Creation" type="osci:TimestampType" />
                <xsd:element name="Forwarding"
                type="osci:TimestampType" minOccurs="1" />
                <xsd:element name="Subject" type="xsd:string" minOccurs="0" />
            </xsd:sequence>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="processDeliveryType">
    <xsd:complexContent>
        <xsd:extension base="osci:DefaultHeaderBlockTemplate">
            <xsd:sequence>
                <xsd:element name="ProcessCardBundle"
                type="osci:ProcessCardBundleType" minOccurs="0" />
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="IntermediaryCertificatesType">
    <xsd:complexContent>
        <xsd:restriction base="osci:IntermediaryCertificatesTemplate">
            <xsd:sequence>
                <xsd:element name="CipherCertificateIntermediary"
                type="osci:CertificateType" minOccurs="0" />
            </xsd:sequence>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>
```

```

        <xsd:element name="SignatureCertificateIntermediary"
            type="osci:CertificateType" minOccurs="0" />
    </xsd:sequence>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="NonIntermediaryCertificatesType">
    <xsd:complexContent>
        <xsd:extension base="osci:NonIntermediaryCertificatesTemplate" />
    </xsd:complexContent>
</xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="ClientSignature" type="osci:SignatureBlockType" />

<xsd:element name="DesiredLanguages" type="osci:DesiredLanguagesType" />

<xsd:element name="processDelivery" type="osci:processDeliveryType" />

<xsd:element name="IntermediaryCertificates"
    type="osci:IntermediaryCertificatesType" />

<xsd:element name="NonIntermediaryCertificates"
    type="osci:NonIntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

<xsd:element name="ContentPackage" type="osci:ContentPackageType" />

</xsd:schema>

```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)
- Ausprägung von XML-Encryption(siehe Abschnitt 6.5)

### 6.6.20 Bearbeitungsantwort

Eine Bearbeitungsantwort wird entweder als SOAP-Envelope realisiert oder (wenn sie Attachments enthält) als SOAP-Message-Package im Sinne von [SOAP A].

Das folgende Beispiel erläutert den Aufbau einer Bearbeitungsantwort:

```

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

<!-- XML-Beispiel OSCI 1.2 Abwicklungsauftrag -->
<!-- $RCSfile: ResponseToProcessDelivery.xml,v $, $Revision: 1.7 $ -->

```



```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    soapResponseToProcessDelivery.xsd
    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">
  <soap:Header>
    <osci:ControlBlock Id="XREF-0000" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
      <osci:Response>xaLW8jIJcCAwEA</osci:Response>
    </osci:ControlBlock>
    <osci:responseToProcessDelivery Id="XREF-0005" soap:mustUnderstand="1"
      soap:actor="http://schemas.xmlsoap.org/soap/actor/next">
      <osci:Feedback>
        <osci:Entry xml:lang="de">
          <osci:Code>0800</osci:Code>
          <osci:Text>Auftrag ausgeführt, Dialog beendet</osci:Text>
        </osci:Entry>
      </osci:Feedback>
    </osci:responseToProcessDelivery>
  </soap:Header>
  <soap:Body Id="XREF-0100">
    <osci:ContentPackage>
      <osci:ContentContainer>
        <osci:Content />
      </osci:ContentContainer>
    </osci:ContentPackage>
  </soap:Body>
</soap:Envelope>

```

Ein Dienstanbieter muss das Element `osci:MessageId` genau dann einstellen, wenn er im Bearbeitungsauftrag eine Zustellung mit Laufzettel erhalten hat.

Eine Bearbeitungsantwort (bzw. ihr Root-Body-Part) genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für Bearbeitungsantworten:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">
  <xsd:import namespace="http://www.osci.de/2002/04/osci"

```

```

    schemaLocation="ResponseToProcessDelivery.xsd" />

<xsd:annotation>
  <xsd:documentation xml:lang="de">
    OSCI 1.2 - Bearbeitungsantwort SOAP-Envelope
    $RCSfile: soapResponseToProcessDelivery.xsd,v $, $Revision: 1.5 $
  </xsd:documentation>
</xsd:annotation>

<!-- ### restrictions ### -->

<xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

  <!-- ### Envelope, Header und Body ### -->

  <xsd:complexType name="Envelope" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Envelope">
        <xsd:sequence>
          <xsd:element ref="soap:Header" minOccurs="1" />
          <xsd:element ref="soap:Body" minOccurs="1" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Header" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Header">
        <xsd:sequence>
          <xsd:element ref="osci:ControlBlock" />
          <xsd:element ref="osci:SupplierSignature" minOccurs="0" />
          <xsd:element ref="osci:QualityOfTimestamp"
            minOccurs="0" maxOccurs="2" />
          <xsd:element ref="osci:responseToProcessDelivery" />
          <xsd:element ref="osci:NonIntermediaryCertificates"
            minOccurs="0" />
          <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"
            processContents="lax" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Body" >
    <xsd:complexContent>
      <xsd:restriction base="soap:Body">
        <xsd:sequence>
          <xsd:element ref="osci:ContentPackage" minOccurs="0" />

```

```

        </xsd:sequence>
        <xsd:attribute name="Id" type="xsd:ID" use="required" />
    </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- Schema für Bearbeitungsantworten:

```

<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="order.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Bearbeitungsantwort
      $RCSfile: ResponseToProcessDelivery.xsd,v $, $Revision: 1.3 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### derived types ### -->

  <xsd:complexType name="ControlBlockType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ControlBlockTemplate">
        <xsd:sequence>
          <xsd:element name="Response" type="xsd:string" minOccurs="1" />
          <xsd:element name="Challenge" type="xsd:string"
            minOccurs="0" maxOccurs="0" />
        </xsd:sequence>
        <xsd:attribute name="ConversationId" type="osci:Number"
          use="prohibited" />
        <xsd:attribute name="SequenceNumber" type="osci:Number"
          use="prohibited" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="responseToProcessDeliveryType">
    <xsd:complexContent>

```

```
<xsd:extension base="osci:DefaultHeaderBlockTemplate">
  <xsd:sequence>
    <xsd:element name="Feedback" type="osci:FeedbackType" />
    <xsd:sequence minOccurs="0">
      <xsd:element name="MessageId" type="osci:MessageIdType" />
      <xsd:element name="Subject" type="xsd:string" minOccurs="0" />
    </xsd:sequence>
  </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="NonIntermediaryCertificatesType">
  <xsd:complexContent>
    <xsd:extension base="osci:NonIntermediaryCertificatesTemplate" />
  </xsd:complexContent>
</xsd:complexType>

<!-- ### global types ### -->

<!-- ### global elements SOAP-Header ### -->

<xsd:element name="ControlBlock" type="osci:ControlBlockType" />

<xsd:element name="SupplierSignature" type="osci:SignatureBlockType" />

<xsd:element name="QualityOfTimestamp"
  type="osci:QualityOfTimestampType" />

<xsd:element name="responseToProcessDelivery"
  type="osci:responseToProcessDeliveryType" />

<xsd:element name="NonIntermediaryCertificates"
  type="osci:NonIntermediaryCertificatesType" />

<!-- ### global elements SOAP-Body ### -->

<xsd:element name="ContentPackage" type="osci:ContentPackageType" />

</xsd:schema>
```

- Ausprägung von XML-Signature (siehe Abschnitt 6.4)
- Ausprägung von XML-Encryption(siehe Abschnitt 6.5)

## 6.7 OSCI-Nachrichten

OSCI-Transport unterscheidet drei Typen von OSCI-Nachrichten:

- *Verschlüsselte Auftragsdaten* (siehe Abschnitt 6.7.1)
- *Unverschlüsselte Auftragsdaten* (siehe Abschnitt 6.7.2)

- *Fehlernachricht* (siehe Abschnitt 6.7.3)

### 6.7.1 Verschlüsselte Auftragsdaten

Eine OSCI-Nachricht des Typs *Verschlüsselte Auftragsdaten* wird als SOAP-Message-Package im Sinne von [SOAP A] realisiert.

Das folgende Beispiel erläutert den Aufbau einer OSCI-Nachricht dieses Typs.

```
MIME-Version: 1.0
Content-Type: Multipart/Related; boundary=Trenner; type=text/xml

--Trenner
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <Content-ID@message>

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

<!-- XML-Beispiel OSCI 1.2 Verschlüsselte Nachricht -->
<!-- $RCSfile: MessageEncrypted.xml,v $, $Revision: 1.3 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    soapMessageEncrypted.xsd
    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">
  <soap:Body>
    <xenc:EncryptedData MimeType="Multipart/Related">
      <xenc:EncryptionMethod
        Algorithm="http://www.w3.org/2001/04/xmlenc#tripleDES-cbc" />
      <ds:KeyInfo>
        <xenc:EncryptedKey>
          <xenc:EncryptionMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5">
            <xenc:KeySize>1024</xenc:KeySize>
          </xenc:EncryptionMethod>
          <ds:KeyInfo>
            <ds:X509Data>
              <ds:X509Certificate>
                Chiffrierzertifikat/Nachrichtenempfaenger/Base64codiert
              </ds:X509Certificate>
            </ds:X509Data>
          </ds:KeyInfo>
        <xenc:CipherData>
          <xenc:CipherValue>
            Verschlüsselter/Containerschlüssel/Base64codiert
          </xenc:CipherValue>
        </xenc:EncryptedData>
      </xenc:EncryptedData>
    </soap:Body>
  </soap:Envelope>
```

```

        </xenc:CipherData>
      </xenc:EncryptedKey>
    </ds:KeyInfo>
    <xenc:CipherData>
      <xenc:CipherReference URI="cid:Content-ID@attachment" />
    </xenc:CipherData>
  </xenc:EncryptedData>
</soap:Body>
</soap:Envelope>
--Trenner
Content-Type: text/base64
Content-Transfer-Encoding: 7bit
Content-ID: <Content-ID@attachment>

```

Verschlüsseltes SOAP-Message-Package mit Auftragsdaten

--Trenner

Eine OSCI-Nachricht des Typs *Verschlüsselte Auftragsdaten* umfasst genau zwei Body-Parts: den Root-Body-Part und einen Body-Part mit Content-Type `text/base64`, der die Base64-codierten verschlüsselten Auftragsdaten enthält.

Das Attribut `URI` von `xenc:CipherData` verweist auf die Content-ID des Body-Parts mit den verschlüsselten Auftragsdaten.

Der Root-Body-Part einer OSCI-Nachricht des Typs *Verschlüsselte Auftragsdaten* genügt den folgenden Schemas.

- Ausprägung eines SOAP-Envelopes für den Nachrichtentyp *Verschlüsselte Auftragsdaten*:

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.w3.org/2001/04/xmlenc#"
    schemaLocation="http://www.w3.org/TR/xmlenc-core/xenc-schema.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Verschlüsselte Nachricht SOAP-Envelope
      $RCSfile: soapMessageEncrypted.xsd,v $, $Revision: 1.2 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### restrictions ### -->

  <xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

```

```

<!-- ### Envelope, Header und Body ### -->

<xsd:complexType name="Envelope" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Envelope">
      <xsd:sequence>
        <xsd:element ref="soap:Body" minOccurs="1" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="Body" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Body">
      <xsd:choice>
        <xsd:element ref="xenc:EncryptedData" />
      </xsd:choice>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>

```

- Ausprägung von XML-Encryption (siehe Abschnitt 6.5)

### 6.7.2 Unverschlüsselte Auftragsdaten

Eine OSCI-Nachricht des Typs *Unverschlüsselte Auftragsdaten* besteht aus genau einem Auftrag oder genau einer Auftragsantwort (siehe Abschnitt 6.6).

### 6.7.3 Fehlernachricht

Eine OSCI-Nachricht des Typs *Fehlernachricht* wird als SOAP-Envelope realisiert.

Das folgende Beispiel erläutert den Aufbau einer OSCI-Nachricht dieses Typs.

```

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

<!-- XML-Beispiel OSCI 1.2 Fehlernachricht -->
<!-- $RCSfile: MessageFault.xml,v $, $Revision: 1.3 $ -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    soapMessageFault.xsd
    http://www.w3.org/2000/09/xmldsig# oscisig.xsd
    http://www.w3.org/2001/04/xmlenc# oscienc.xsd">

```

```

<soap:Body>
  <soap:Fault>
    <soap:faultcode>soap:Client</soap:faultcode>
    <soap:faultstring>Auftragsdaten konnten nicht entschlüsselt
werden</soap:faultstring>
    <soap:detail>
      <osci:Code>9201</osci:Code>
    </soap:detail>
  </soap:Fault>
</soap:Body>
</soap:Envelope>

```

Eine OSCI-Nachricht des Typs *Fehlernachricht* genügt dem folgenden Schema.

- Ausprägung eines SOAP-Envelopes für den Nachrichtentyp *Fehlernachricht*.

```

<xsd:schema targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.w3.org/2001/04/xmlenc#"
    schemaLocation="http://www.w3.org/TR/xmlenc-core/xenc-schema.xsd" />

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Fehlernachricht SOAP-Envelope
      $RCSfile: soapMessageFault.xsd,v $, $Revision: 1.2 $
    </xsd:documentation>
  </xsd:annotation>

  <!-- ### restrictions ### -->

  <xsd:redefine schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">

    <!-- ### Envelope, Header und Body ### -->

    <xsd:complexType name="Envelope" >
      <xsd:complexContent>
        <xsd:restriction base="soap:Envelope">
          <xsd:sequence>
            <xsd:element ref="soap:Body" minOccurs="1" />
          </xsd:sequence>
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>

```



```
<xsd:complexType name="Body" >
  <xsd:complexContent>
    <xsd:restriction base="soap:Body">
      <xsd:choice>
        <xsd:element name="Fault" type="soap:Fault" />
      </xsd:choice>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

</xsd:redefine>

</xsd:schema>
```

## 7. Konformitätskatalog

Um konform zu dieser Spezifikation zu sein, müssen Softwaresysteme für Benutzer, Dienstanbieter und Intermediäre alle für sie relevanten normativen Teile dieser Spezifikation beachten.

Insbesondere müssen sie also

- die vorgesehenen Namespaces verwenden
- XML-Dokumente aufbauen, die zu den Schemas konform sind, die in dieser Spezifikation angegeben sind
- die Reaktionsvorschriften in Kapitel 5 beachten und die dort angegebenen Rückmeldungscodes benutzen

Softwaresysteme für Intermediäre müssen alle in dieser Spezifikation definierten Auftragstypen in der angegebenen Version unterstützen. Softwaresysteme für Benutzer und Dienstanbieter brauchen nur Unterstützung für diejenigen Auftragstypen zu bieten, die sie für ihren speziellen Einsatzzweck benötigen.

## 8. Literaturverzeichnis

Dieses Literaturverzeichnis gliedert sich in einen Abschnitt mit Referenzen, die im Hinblick auf diese Spezifikation normativen Charakter haben, und einen weiteren Abschnitt mit Literaturhinweisen, die zu Informationszwecken aufgenommen worden sind.

### 8.1 Normative Referenzen

- [ALG 2001] Bundesamt für Sicherheit in der Informationstechnik: Geeignete Kryptoalgorithmen – In Erfüllung der Anforderungen nach §17 (1) SigG vom 16. Mai 2001 in Verbindung mit §17 (2) SigV vom 22. Oktober 1997. Bonn, 5.7.01. Veröffentlicht im Bundesanzeiger Nr. 158 vom 24. August 2001, Seite 18562. Online verfügbar unter <http://www.bsi.bund.de/esig/basics/techbas/krypto/39.pdf>
- [EBXML] OASIS ebXML Messaging Services Technical Committee: Message Service Specification. Version 2.0. 1 April 2002
- [ISIS-MTT] ISIS-MTT Specification – Common ISIS-MTT Specification for PKI Applications. Version 1.0.1, 15 November 2001. Online verfügbar unter [http://www.t7-isis.de/ISIS-MTT/ISIS-MTT-CoreDocuments-Version1\\_0\\_1.pdf](http://www.t7-isis.de/ISIS-MTT/ISIS-MTT-CoreDocuments-Version1_0_1.pdf)
- [OSCI-ANF] datenschutz nord GmbH: OSCI-Transport 1.2: Entwurfsprinzipien, Sicherheitsziele und -mechanismen. Bremerhaven, April 2002
- [PKCS 1] B. Kaliski, J. Staddon: PKCS #1: RSA Cryptography Specifications – Version 2.0. RFC 2437, October 1998. Online verfügbar unter <http://www.ietf.org/rfc/rfc2437.txt>
- [PKCS 5] B. Kaliski: PKCS #5: Password-Based Cryptography Specification – Version 2.0. RFC 2898, September 2000. Online verfügbar unter <http://www.ietf.org/rfc/rfc2898.txt>
- [RFC 2045] N. Freed, N. Borenstein: Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. RFC 2045, November 1996. Online verfügbar unter <http://www.ietf.org/rfc/rfc2045.txt>
- [RFC 2046] N. Freed, N. Borenstein: Part Two: Media Types. RFC 2046, November 1996. Online verfügbar unter <http://www.ietf.org/rfc/rfc2046.txt>
- [RFC 2111] E. Levinson: Content-ID and Message-ID Uniform Resource Locators. RFC 2111, March 1997. Online verfügbar unter <http://www.ietf.org/rfc/rfc2111.txt>
- [RFC 2387] E. Levinson: The MIME Multipart/Related Content-type. The Internet Society, RFC 2387, August 1998. Online verfügbar unter <http://www.ietf.org/rfc/rfc2387.txt>
- [SOAP 1] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Nielsen, S. Thatte: Simple Object Access Protocol (SOAP) 1.1. W3C Note 08 May 2000. Online verfügbar unter <http://www.w3.org/TR/2000/NOTE-SOAP-20000508>. Es handelt sich um „Work in progress“. Für diese Spezifikation maßgebend ist die angegebene Version, die von der aktuellen Version (online verfügbar unter <http://www.w3.org/TR/SOAP>) abweichen kann
- [SOAP A] John J. Barton, Satish Thatte, Henrik Frystyk Nielsen: SOAP Messages with Attachments. W3C Note 11 December 2000. Online verfügbar unter <http://www.w3.org/TR/2001/WD-soap12-part1-20011217/>. Es handelt sich um „Work in progress“. Für diese Spezifikation maßgebend ist die angegebene Version, die von der aktuellen Version (online verfügbar unter <http://www.w3.org/TR/soap12-part1/>) abweichen kann

- [XAdES] XML Advanced Electronic Signatures (XAdES). ETSI TS 101 903 V1.1.1 (2002-02)
- [XDSIG] Mark Bartel, John Boyer, Barb Fox, Brian LaMacchia, Ed Simon: XML-Signature Syntax and Processing. W3C Recommendation 12 February 2002. Online verfügbar unter <http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/>
- [XENC] Takeshi Imamura, Blair Dillaway, Ed Simon: XML Encryption Syntax and Processing. W3C Candidate Recommendation 04 March 2002. Online verfügbar unter <http://www.w3.org/TR/2002/CR-xmlenc-core-20020304/>. Es handelt sich um „Work in progress“. Für diese Spezifikation maßgebend ist die angegebene Version, die von der aktuellen Version (online verfügbar unter <http://www.w3.org/TR/xmlenc-core/>) abweichen kann
- [XML] Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000. Online verfügbar unter <http://www.w3.org/TR/2000/REC-xml-20001006>
- [XNAMESP] Namespaces in XML. W3C Recommendation 14 January 1999. Online verfügbar unter <http://www.w3.org/TR/1999/REC-xml-names-19990114>
- [XSCHEMA 1] XML Schema Part 1: Structures. W3C Recommendation 2 May 2001. Online verfügbar unter <http://www.w3.org/TR/2001/REC-xmldsig-core-20010502/>
- [XSCHEMA 2] XML Schema Part 2: Datatypes. W3C Recommendation 02 May 2001. Online verfügbar unter <http://www.w3.org/TR/2001/REC-xmldsig-core-20010502/>
- [XSLT] XSL Transformations (XSLT) – Version 1.0. W3C Recommendation 16 November 1999. Online verfügbar unter <http://www.w3.org/TR/1999/REC-xslt-19991116.html>

## 8.2 Nicht-normative Referenzen

- [SOAP 12P0] SOAP Version 1.2 Part 0: Primer. W3C Working Draft 17 December 2001. Aktuelle Version online verfügbar unter <http://www.w3.org/TR/soap12-part0/>
- [XSCHEMA 0] XML Schema Part 0: Primer. W3C Recommendation, 2 May 2001. Aktuelle Version online verfügbar unter <http://www.w3.org/TR/xmlschema-0/>

## 9. Glossar

Abwicklungsantwort: → Auftragsantwort auf einen → Abwicklungsauftrag.

Abwicklungsauftrag: → Auftrag, mit dem ein → Benutzer eine → Zustellung an einen → Intermediär übermittelt, damit dieser sie an einen → Dienstanbieter weiterleitet. Im Gegensatz zum → Weiterleitungsauftrag erwartet der Benutzer in der Auftragsantwort des Intermediärs eine Zustellung des Dienstanbieters.

Annahmeantwort: → Auftragsantwort auf einen → Annahmearauftrag.

Annahmearauftrag: → Auftrag, mit dem ein → Intermediär eine → Zustellung an einen → Dienstanbieter übermittelt.

Attachment: Teil einer → Zustellung. Besteht aus beliebigen Daten, die im Regelfall *nicht* in XML formuliert sind. Wird von einem oder mehreren → Autoren befüllt und richtet sich an einen oder mehrere → Leser.

Auftrag: Datenpaket auf → Auftragsebene, das ein → Client an einen → Supplier richtet. Umfasst eine Arbeitsanweisung an den Supplier und weitere Informationen, die der Supplier benötigt, um die Anweisung auszuführen und eine → Auftragsantwort zu schicken. Ein Auftrag kann insbesondere eine → Zustellung enthalten. Er kann durch den Client signiert sein.

Auftragsantwort: Datenpaket auf → Auftragsebene, das ein → Supplier als Reaktion auf einen → Auftrag an einen → Client richtet. In der Auftragsantwort teilt der Supplier das Ergebnis der Bearbeitung des Auftrags mit. Eine Auftragsantwort kann insbesondere eine → Zustellung enthalten. Die Auftragsantwort kann durch den Supplier signiert sein.

Auftragsebene: Mittlere Ebene der Kommunikation mittels OSCI-Transport. Auf Auftragsebene kommunizieren ein → Benutzer und ein → Intermediär; der eine in der Rolle → Client, der andere in der Rolle → Supplier. Im Regelfall ist der Benutzer der Client und der Intermediär der Supplier; es gibt jedoch auch Szenarien, in denen die Rollen vertauscht sind.

Autor: Instanz, die → Inhaltsdaten in → Inhaltsdatencontainern und → Attachments erzeugt.

Bearbeitungsantwort: → Auftragsantwort auf einen → Bearbeitungsauftrag.

Bearbeitungsauftrag: → Auftrag, mit dem ein → Intermediär einen → Dienstanbieter auffordert, eine → Zustellung eines → Benutzers zu bearbeiten und als Reaktion eine weitere Zustellung zu liefern.

Benutzer: Instanz, die OSCI-Transport verwendet, um Informationen mit anderen Benutzern auszutauschen. Bei einem Benutzer kann es sich um eine Person, eine Gruppe von Personen oder auch um ein Softwaresystem handeln. Die einzige Voraussetzung, die ein Benutzer erfüllen muss, ist der Besitz eines gültigen Chiffrierzertifikats.

Challenge: 1. Nach einem Zufallsverfahren gewählter Wert, den ein → Client in seinen Auftrag an einen → Supplier einstellt. Der Supplier muss diesen Wert in seiner Auftragsantwort als → Response wiederholen. 2. Zufällig gewählter Wert, den ein → Supplier innerhalb eines → expliziten Dialogs mit einem → Client in seine Auftragsantwort einstellt. Der Client muss diesen Wert in seinem nächsten Auftrag wiederholen.

Client: Rolle eines → Benutzers oder → Intermediärs auf → Auftragsebene. Der Client ist derjenige Kommunikationspartner, der die Kommunikation anstößt. Er sendet einen → Auftrag an den → Supplier und empfängt dessen → Auftragsantwort.

ConversationId: Kennung eines → Dialogs.

Dialog: Abfolge von Auftrag-Antwort-Paaren. Besteht aus mindestens einem Auftrag-Antwort-Paar. Innerhalb eines Dialogs muss der → Client mit dem Senden seines nächsten → Auftrags warten, bis er die → Auftragsantwort des → Suppliers auf den vorigen Auftrag empfangen hat.

Dialog, expliziter: → Dialog, der durch einen → Dialoginitialisierungsauftrag zustande kommt. Dauert an, bis der → Supplier seine Auftragsantwort auf den → Dialogendeauftrag des → Clients abgeschickt hat.

Dialog, impliziter: Gegensatz zu → expliziter Dialog. Kommt durch einen beliebigen → Auftrag zustande (der kein → Dialoginitialisierungsauftrag ist). Dauert an, bis der → Supplier die → Auftragsantwort abgeschickt hat.

Dialogendeantwort: → Auftragsantwort auf einen → Dialogendeauftrag.

Dialogendeauftrag: → Auftrag, mit dem ein → Client einen → expliziten Dialog mit einem → Supplier beendet.

Dialoginitialisierungsantwort: → Auftragsantwort auf einen → Dialoginitialisierungsauftrag.

Dialoginitialisierungsauftrag: → Auftrag, mit dem ein → Client einen → expliziten Dialog mit einem → Supplier eröffnet.

Dienstanbieter: Spezieller → Benutzer, der (prinzipiell jederzeit) über einen URL erreichbar ist.

Empfänger: Rolle eines → Benutzers auf → Geschäftsvorfallsebene. Ein Empfänger nimmt eine → Zustellung eines → Senders entgegen.

Geschäftsvorfallsebene: Oberste Ebene der Kommunikation mittels OSCI-Transport. Auf Geschäftsvorfallsebene kommunizieren zwei → Benutzer; der eine in der Rolle → Sender, der andere in der Rolle → Empfänger.

Inhaltsdaten: Daten, die ein → Benutzer an einen anderen Benutzer übermitteln möchte. Die Übertragung dieser Daten bietet den Anlass, OSCI-Transport zu verwenden.

Inhaltsdatencontainer: Teil einer → Zustellung. Kann beliebige in XML formulierte Informationen enthalten. Wird von einem oder mehreren → Autoren befüllt und richtet sich an einen oder mehrere → Leser. Kann insbesondere Verschlüsselungsinformationen zu weiteren Inhaltsdatencontainern und/oder → Attachments in der selben Zustellung enthalten. Kann von den Autoren signiert werden.

Intermediär: Instanz, die im Informationsaustausch zwischen → Benutzern vermittelt. Der Intermediär ist von der Rolle her neutral. Er kann aber physisch durchaus bei einem der beiden kommunizierenden Benutzer lokalisiert sein.

Laufzettel: In einem Laufzettel wird der Weg einer → Zustellung von einem → Sender über einen → Intermediär zu einem → Empfänger protokolliert. Der Laufzettel wird vom Intermediär geführt.

Laufzettelabholantwort: → Auftragsantwort auf einen → Laufzettelabholauftrag.

Laufzettelabholauftrag: → Auftrag, mit dem ein → Benutzer einen oder mehrere → Laufzettel bei einem → Intermediär abholt.

Leser: Instanz, an die → Inhaltsdaten in → Inhaltsdatencontainern und → Attachments gerichtet sind.

MessageId: Kennung einer → Zustellung.

MessageId-Anforderungsantwort: → Auftragsantwort auf einen → MessageId-Anforderungsauftrag.

MessageId-Anforderungsauftrag: → Auftrag, mit dem ein → Benutzer eine MessageId bei einem Intermediär anfordert.

Nachrichtenebene: Basisebene der Kommunikation mittels OSCI-Transport. Auf Nachrichtenebene kommunizieren ein → Benutzer und ein → Intermediär; der eine in der Rolle → Nachrichtensender, der andere in der Rolle → Nachrichtenempfänger. Beide Kommunikationspartner können beide Rollen annehmen.

Nachrichtensender: Rolle eines → Benutzers oder → Intermediärs auf → Nachrichtenebene. Ein Nachrichtensender sendet eine → OSCI-Nachricht an einen → Nachrichtenempfänger.

Nachrichtenempfänger: Rolle eines → Benutzers oder → Intermediärs auf → Nachrichtenebene. Ein Nachrichtenempfänger empfängt eine → OSCI-Nachricht von einem → Nachrichtensender.

Nutzungsdaten: Daten, die zusätzlich zu den → Inhaltsdaten ausgetauscht werden und dazu dienen, den Datenfluss zu kontrollieren und zu steuern.

OSCI-Nachricht: Datenpaket auf → Nachrichtenebene, das ein → Nachrichtensender an einen → Nachrichtenempfänger richtet. Umfasst entweder genau einen → Auftrag oder genau eine → Auftragsantwort oder genau eine Fehlermeldung. Der Auftrag / die Auftragsantwort kann verschlüsselt sein. In diesem Fall enthält die OSCI-Nachricht zusätzlich Informationen, die der Nachrichtenempfänger benötigt, um den Auftrag / die Auftragsantwort zu entschlüsseln.

Response: 1. Wert, den ein → Supplier als Reaktion auf das → Challenge im Auftrag des jeweiligen → Clients in seine Auftragsantwort einstellt. 2. Wert, den ein → Client innerhalb eines → expliziten Dialogs mit einem → Supplier als Reaktion auf das → Challenge in der Auftragsantwort des Suppliers in seinen Folgeauftrag einstellt.

Sender: Rolle eines → Benutzers auf → Geschäftsvorfallsebene. Ein Sender richtet eine → Zustellung an einen → Empfänger.

SequenceNumber: Nummer eines Auftrags / einer Auftragsantwort innerhalb eines → Dialogs.

Supplier: Rolle eines → Benutzers oder → Intermediärs auf → Auftragsebene. Der Supplier ist der reagierende Kommunikationspartner. Er empfängt den → Auftrag des → Clients und sendet als Reaktion eine → Auftragsantwort.

Weiterleitungsantwort: → Auftragsantwort auf einen → Weiterleitungsauftrag.

Weiterleitungsauftrag: → Auftrag, mit dem ein → Benutzer eine → Zustellung an einen → Intermediär übermittelt, damit dieser sie an einen → Dienstleister weiterleitet. Im Gegensatz zum → Abwicklungsauftrag erwartet der Benutzer in der Auftragsantwort des Intermediärs *keine* Zustellung des Dienstleisters.

Zustellung: Datenpaket auf → Geschäftsvorfallsebene, das ein → Sender an einen → Empfänger richtet. Kann beliebig viele → Inhaltsdatencontainer und/oder → Attachments enthalten. Einige oder alle Inhaltsdatencontainer und/oder Attachments können verschlüsselt sein. Die Zustellung enthält dann zusätzlich Informationen, die der

Empfänger benötigt, um die Inhaltsdatencontainer und/oder Attachments zu entschlüsseln. Weiterhin kann die Zustellung ein oder mehrere Zertifikate umfassen.

Zustellungsabholantwort: → Auftragsantwort auf einen → Zustellungsabholauftrag.

Zustellungsabholauftrag: → Auftrag, mit dem ein → Benutzer bei einem → Intermediär eine → Zustellung abholt, die ein anderer Benutzer zuvor mittels eines → Zustellungsauftrags eingereicht hat.

Zustellungsantwort: → Auftragsantwort auf einen → Zustellungsauftrag.

Zustellungsauftrag: → Auftrag, mit dem ein → Benutzer eine → Zustellung an einen → Intermediär übermittelt, damit dieser sie zur Abholung durch einen anderen Benutzer bereithält.



## 10. Abbildungsverzeichnis

Abbildung 1: Ebenen der Kommunikation .....	9
Abbildung 2: Mögliche Realisierung des Vier-Augen-Prinzips .....	11
Abbildung 3: Mögliche Realisierung eines Regelmechanismus .....	12
Abbildung 4: One-Way-Message, aktiver Empfänger .....	14
Abbildung 5: One-Way-Message, passiver Empfänger .....	15
Abbildung 6: Request-Response, passiver Empfänger.....	16