

---

# Spezifikation XTA 2 (Version 4)

*16. März 2020 / Final Draft*

---



# Inhaltsverzeichnis

Einleitung .....	1
1 XTA Service–Profile (1.1) .....	3
1.1 Struktur der Profile .....	3
1.1.1 Aufbau des Serviceprofils .....	3
1.1.2 Aufbau des Schutzprofils .....	7
1.1.3 Aufbau des Infrastrukturprofils .....	13
1.1.4 Aufbau des Kryptographieprofils .....	18
1.1.5 Basistypen für Profile .....	22
1.2 Protokollierung der Servicequalitäten .....	24
1.2.1 Basistypen für den Service Report .....	24
1.2.2 Nachweise für den Service Report .....	27
2 XTA Webservice (2.2) .....	37
2.1 Methoden .....	37
2.1.1 Die Schnittstellentypen der Transportebene .....	37
2.1.2 Die Schnittstellentypen der Applikationsebene .....	56
2.2 Das XTA-WS-Informationsmodell .....	60
2.2.1 Datentypen der Informationsobjekte des XTA Webservice .....	60
2.2.2 Globale Elemente der Informationsobjekte des XTA Webservice .....	79
2.3 XTA Webservice SOAP Exceptions .....	83
2.3.1 Die Exceptions des XTA Webservice .....	83
2.3.2 Struktur von Exception und Fehlernummer .....	85
2.3.3 Exceptions als XML-Instanzen .....	86
A Codelisten .....	87
A.1 Übersicht .....	87
A.2 Details .....	88
A.2.1 Abgabestation .....	88
A.2.2 Abweichverhalten .....	89
A.2.3 Bezeichner für kryptographische Verfahren .....	90
A.2.4 Ereignisart .....	92
A.2.5 Geltungsbereich Infrastruktur-Parameter .....	94
A.2.6 Geltungsbereich Schutzprofil-Parameter .....	95
A.2.7 Kanal .....	96
A.2.8 Kommunikation Typ .....	97
A.2.9 Nachweis Verlässlichkeit .....	98
A.2.10 Qualität Authentizität .....	99
A.2.11 Qualität Kryptographie .....	100
A.2.12 Qualität Löschen .....	101
A.2.13 Qualität Protokollierung .....	102
A.2.14 Qualität Unveränderbarkeit .....	103
A.2.15 Qualität Verfügbarkeit .....	104
A.2.16 Qualität Vertraulichkeit .....	105
A.2.17 Schlüssellängen für asymmetrische kryptographische Verfahren .....	106
A.2.18 Startzeitpunkt Löschfrist .....	107
A.2.19 Technische Quittungen .....	108
A.2.20 Transportnachrichten Format .....	109
A.2.21 Transportprotokoll .....	110
A.2.22 Verzeichnis für Adressierung .....	111
A.2.23 Verzeichnis für Identifizierung .....	112
A.2.24 XTA-Rolle .....	113
A.2.25 XTA-WS Fehlernummer .....	114

A.2.26 Zertifikat Medium .....	116
A.2.27 Zertifikat Niveau .....	117
A.2.28 Zertifikat Quelle .....	118
A.2.29 Zertifikat Status .....	119
A.2.30 Zustellfrist .....	120
B Eingebundene externe Modelle/Standards .....	121
B.1 SOAP-Message-Security-1.0 .....	121
B.2 WS-Addressing .....	121
B.3 XML-Signature .....	121
B.4 XOEV-Bibliothek .....	121
C Versionshistorie .....	123
C.1 Release XTA 2 Version 4 (Final Draft) .....	123

# Einleitung



Diese Spezifikation, wendet sich an Entwicklerinnen und Entwickler, die die von XTA bereitgestellte Funktionalität in ihren Anwendungen implementieren.

XTA standardisiert den Austausch von Nachrichten zwischen Fach- und Transportverfahren und unterstützt zudem die automatisierte, fachunabhängige Weiterverarbeitung von Nachrichten. Hierfür werden unter anderem die folgenden, zentralen Strukturen definiert:

- das einheitliche Nachrichtenformat ([Message](#))
- die generischen Attribute für Nachrichtenteile ([ContentType](#))
- die strukturierten Steuerungsdaten des Transportauftrags einer Nachricht ([MessageMetaData](#))
- die strukturierte Beschreibung der nachrichtenübergreifenden fachlichen Anforderungen an die Kommunikation (Servicequalitäten) im Service-Profil ([ServiceProfilType](#))
- einheitliche, fachunabhängige Berichtsformate ([TransportReport](#), [ServiceReport](#)) und Ereignisse ([RecordType](#), [EreignisType](#))

Zudem werden die Schnittstellen für die Kommunikation zwischen Fach- und Transportverfahren spezifiziert (SOAP-Methoden, [Seite 37](#)).

Eine strukturierte Beschreibung der fachlichen Kommunikation (fachlicher Dienst) zwischen dem Autor und Leser einer Fachnachricht sowie zugehörige Anforderungen an den Transport können in einem Service-Profil definiert werden. Die Verwendung ist in XTA optional und bei Bedarf kann eine Referenz auf ein Service-Profil im Transportauftrag übergeben werden. Ein Serviceprofil (siehe [ServiceProfilType](#)):

- enthält Informationen zum Profil (Profilkopf)
- beschreibt die Menge der fachlichen Dienste, auf die das Serviceprofil angewendet werden soll (Servicekategorie) und
- verweist auf weitere Profile, in denen die Anforderungen an den Transport (Servicequalitäten) einschließlich einer Vorgabe für das Verhalten bei Abweichungen festgelegt sind.

Service-Profile sind nur dann verpflichtend anzuwenden, wenn eine entsprechende fachliche Vorgabe besteht. Sie können auch unabhängig von XTA zur strukturierten Vorgabe von Anforderungen verwendet werden.

Fachverfahren fügen ihre Nachricht in die XTA-Nachrichtenstruktur ein. Sie befüllen die generischen Attribute der Nachrichtenteile nach fachlichen Vorgaben, um ihre Weiterverarbeitung zu unterstützen. Fach- und Transportverfahren tragen alle transportrelevanten Informationen, ebenfalls nach fachlicher Vorgabe, in die Metadaten der XTA-Nachricht ein. Sie nutzen zudem die standardisierten Strukturen, um Ereignisse im Transport zu dokumentieren sowie über den Transport und die Einhaltung der Servicequalitäten zu berichten.

XTA ist unabhängig von konkreten Transportverfahren sowie Standards auf Transportebene und Fachebene. Die Verwendung von XTA ermöglicht eine standardisierte Verbindung zwischen beiden Ebenen und damit eine Fokussierung auf die jeweiligen, domänenspezifischen Aufgaben.



# 1 XTA Service–Profile (1.1)



Anforderungen an den Transport von Nachrichten können in XTA formal beschrieben werden. Die Anforderungen werden gemäß ihres Charakters in unterschiedlichen Profilen dokumentiert.

- Das Serviceprofil legt fest, für welche Kombination aus Diensten (Services) und Kommunikationsarten das Serviceprofil angewendet werden soll und welche Anforderungsprofile auf diese Kombination anzuwenden sind.
- Das Schutzprofil beschreibt die Anforderungen an Datenschutz und Datensicherheit.
- Die Anforderungen an die Infrastrukturkomponenten für den Transport werden im Infrastrukturprofil festgelegt.
- Das Kryptographieprofil enthält die Anforderungen an zu verwendende kryptographische Verfahren.

Zusätzlich zu den Anforderungen an den Transport von Nachrichten wird ebenfalls festgelegt, ob der Nachrichtentransport fortgesetzt oder abgebrochen werden soll, wenn eine spezifische geforderte Eigenschaft von System nicht erbracht werden kann.

Der Inhalt und die Struktur der Protokollierung über die Erbringung und Nichterbringung geforderter Qualitäten werden über Protokollereignisse und den Servicereport definiert.

In der vorliegenden Version sind die Profile und die zugehörige Protokollierung optional und müssen nicht vom implementierten XTA System les- und interpretierbar sein.

## 1.1 Struktur der Profile

### 1.1.1 Aufbau des Serviceprofils

#### 1.1.1.1 serviceProfil

Globales Element: *serviceProfil*

Informationen zur Identität und Gültigkeitsdauer des Service–Profils stehen im Profilkopf. Die Servicekategorie beschreibt den fachlichen Dienst (nicht den Transportdienst) und legt damit den Anwendungsbereich des Service–Profils fest. Für diesen Anwendungsbereich werden Profile referenziert, welche die geforderten Servicequalitäten und zugehörige Vorgaben bei Abweichungen enthalten.

Dieser Typ ist eine Erweiterung des Basistyps *ServiceProfilType* (siehe [Abschnitt 1.1.1.2 auf Seite 3](#)).

#### 1.1.1.2 ServiceProfilType

Typ: *ServiceProfilType*

Die Grundstruktur für alle Serviceprofile.

Kindelemente von <i>ServiceProfilType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
profilkopf		1		

Dieses Element wird gefüllt mit Informationen zu Identität und Gültigkeit der vorliegenden Profil-Instanz.

Kindelemente von <i>ServiceProfilType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Wenn das Profil nicht mehr gültig ist, darf das Profil nicht angewendet werden. Der Transport muss abgebrochen werden. Die verletzte Vorgabe muss mit dem Ereignis <i>ServiceQualitaetVerletzt</i> im <i>ServiceReport</i> dokumentiert werden. Im <i>TransportReport</i> muss für das nichtanwendbare Service-Profil eine Fehlermeldung <i>ParameterIsNotValidException</i> protokolliert werden.				
<b>identifikation</b>	<i>xtasp:IdentifikationType</i>	1	1.1.5.2	23
Identität und Herkunft der vorliegenden Profil-Instanz (informativ)				
<b>dokumentation</b>	<i>oscimeta:NonEmptyStringType</i>	1	2.2.1.40	75
Dokumentation der vorliegenden Profil-Instanz				
<b>gueltigkeit</b>	<i>xtasp:InstanzGueltigkeitType</i>	1	1.1.5.3	23
Gültigkeit der vorliegenden Profil-Instanz				
<b>servicekategorie</b>	<i>xtasp:ServicekategorieType</i>	1	1.1.1.3.1	4
Beschreibt die Eigenschaften eines oder mehrerer fachlicher Dienste, auf die das vorgegebene Service-Profil angewendet werden soll.				
Wird mindestens eine Eigenschaft nicht erfüllt, kann das vorgegeben Service-Profil nicht angewendet werden. Der Transport muss abgebrochen werden. Jede Eigenschaft, die nicht erfüllt wurde, muss jeweils durch ein Ereignis <i>ServiceQualitaetVerletzt</i> im <i>ServiceReport</i> dokumentiert werden. Im <i>TransportReport</i> muss für das nichtanwendbare Service-Profil eine Fehlermeldung <i>ParameterIsNotValidException</i> protokolliert werden.				
<b>refSchutzprofil</b>	<i>xtasp:InstanzIdentifikationType</i>	1	1.1.5.1	22
Referenz auf eine existierende Schutzprofil-Instanz, die Struktur selbst ist in <a href="#">schutzProfil</a> definiert.				
<b>refInfrastrukturprofil</b>	<i>xtasp:InstanzIdentifikationType</i>	1..n	1.1.5.1	22
Referenz auf eine existierende Infrastrukturprofil-Instanz, die Struktur ist in <a href="#">infrastrukturProfil</a> definiert.				
Wenn mehrere Referenzen angegeben werden, liegt es in der Zuständigkeit der Rolle Sender, das für einen gegebenen Transportauftrag geeignete Infrastrukturprofile aus den referenzierten Infrastrukturprofilen auszuwählen. Die Auswahl muss im <i>ServiceReport</i> durch das Ereignis <i>AuswahlProfil</i> protokolliert werden.				
<b>refKryptographieProfil</b>	<i>xtasp:InstanzIdentifikationType</i>	0..n	1.1.5.1	22
Referenz auf eine existierende Kryptographieprofil-Instanz, die Struktur ist in <a href="#">kryptographieProfil</a> definiert.				
Wenn mehrere Referenzen angegeben werden, liegt es in der Zuständigkeit der Rolle Sender, das für einen gegebenen Transportauftrag geeignete Kryptographieprofil aus den referenzierten Kryptographieprofilen auszuwählen. Die Auswahl muss im <i>ServiceReport</i> durch das Ereignis <i>AuswahlProfil</i> protokolliert werden.				

### 1.1.1.3 Datentypen des Serviceprofils

#### 1.1.1.3.1 ServicekategorieType

Typ: *ServicekategorieType*

Grundstruktur für alle Servicekategorien.

Kindelemente von <i>ServicekategorieType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>service</b>		1..n		
Eindeutige Identifikation eines fachlichen Dienstes, durch eine eindeutige Ressourcenkennung (URI) und zugeordneten fachlichen Nachrichtentypen.				
<b>bezeichnung</b>	<i>xs:anyURI</i>	1		
In diesem Element steht bzw. ist einzutragen die Bezeichnung des Service. Es ist jeweils die fachliche Dienst-Bezeichnung einzutragen; sie ist im Fachstandard spezifiziert.				



Kindelemente von <i>ServicekategorieType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Für Fachstandards mit DVDV-Bezug ist diese Bezeichnung die URL der Service-WSDL. Diese muss für Fachstandards im DVDV-Umfeld in der Spezifikation des Fachstandards eingetragen sein.				
<b>nachrichtentypListen</b>		1		
Unterhalb dieses Elements werden die Gruppen von Nachrichten eingetragen, die zu diesem Dienst (Service) gehören.				
<b>nachrichtentypListe</b>	<i>xtasp:NachrichtenformatListType</i>	1..n	1.1.1.3.2	5
Dies ist eine Liste von Nachrichtentypen, die einem gemeinsamen Namespace zugeordnet sind.				
<b>kommunikationstyp</b>	<i>xtacore:Code.Kommunikationstyp</i>	1	1.1.1.3.3.2	6
Angabe zur Art der fachlichen Kommunikation (synchron oder asynchron)				
<b>hatSynchroneFachantwort</b>	<i>xs:boolean</i>	1		
Festlegung für eine fachliche Kommunikation, ob eine fachliche Antwortnachricht erwartet wird. Für synchrone Szenarien mit Fachantwort ist hier „true“ zu wählen, für asynchrone Szenarien und für synchrone Szenarien ohne Fachantwort „false“.				
<b>verfuegbarkeit</b>	<i>xtasp:Code.QualitaetVerfuegbarkeit</i>	1	1.1.1.3.3.3	6
Hier wird die für Verfügbarkeit des fachlichen Dienstes zur Bearbeitung von Transportaufträge eingetragen. Diese wirkt sich unmittelbar auf die Anforderung zur Verfügbarkeit der Transportverfahren aus, mit denen diese Verfügbarkeit realisiert wird. Die Verfügbarkeit ist dabei die Wahrscheinlichkeit, dass der Transportauftrag innerhalb des vereinbarten Zeitraums ausgeführt wird.				
<b>zustellfrist</b>	<i>xtacore:Code.Zustellfrist</i>	1	1.1.1.3.3.5	6
Die Zustellfrist innerhalb derer der fachliche Dienst die Zustellung von Fachnachrichten und damit die über diesen Dienst erteilten Transportaufträge abschließen muss. Die Zeitspanne beginnt ab dem Zeitpunkt der Erteilung des Transportauftrags.				

### 1.1.1.3.2 NachrichtenformatListType

Typ: *NachrichtenformatListType*

Eine Liste von Nachrichtenformaten, die einem gemeinsamen Namespace zugeordnet sind.

Kindelemente von <i>NachrichtenformatListType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>payloadSchema</b>	<i>oscimeta:NonEmptyURIType</i>	1	2.2.1.41	75
Hier ist der Namensraum des Fachstandards in einer bestimmten Version einzutragen, dem die Nachrichtenformate zugeordnet sind.				
<b>nachrichtenformat</b>	<i>xs:string</i>	1..n		
Pro Element ist ein Nachrichtenformat aus dem Fachstandard einzutragen.				

### 1.1.1.3.3 Codelisten des Serviceprofils

#### 1.1.1.3.3.1 Code.ZertifikatQuelle

Codelisten	
-beschreibung	Diese Codeliste enthält gültige Werte für die Angabe des Herausgebers von Zertifikaten in einer Public Key Infrastruktur.

Codelisten	
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 118</a>
-kennung	urn:xoev-de:xta:serviceprofile:codeliste:zertifikat.quelle
-version	1.0

#### 1.1.1.3.3.2 Code.Kommunikationstyp

Codelisten	
-beschreibung	Diese Codeliste definiert mögliche Kommunikationsszenarien.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 97</a>
-kennung	urn:xoev-de:xta:core:codeliste:kommunikationstyp
-version	1.0

#### 1.1.1.3.3.3 Code.QualitaetVerfuegbarkeit

Codelisten	
-beschreibung	Diese Codeliste definiert Verfügbarkeitsstufen.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 104</a>
-kennung	urn:xoev-de:xta:serviceprofile:codeliste:verfuegbarkeit.qualitaet
-version	1.0

#### 1.1.1.3.3.4 Code.StartzeitpunktLoeschfrist

Codelisten	
-beschreibung	Diese Codeliste beschreibt Startzeitpunkte für Löschrufen.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 107</a>
-kennung	urn:xoev-de:xta:core:codeliste:startzeitpunkt-loeschfrist
-version	1.0

#### 1.1.1.3.3.5 Code.Zustellfrist

Codelisten	
-beschreibung	Diese Codeliste enthält Werte, die als Zustellfrist verwendet werden können.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 120</a>
-kennung	urn:xoev-de:xta:core:codeliste:zustellfrist
-version	1.0

#### 1.1.1.3.3.6 Code.Abgabestation

Codelisten	
-beschreibung	Diese Codeliste beschreibt die Knoten der Infrastruktur, an denen eine Nachricht final abgeliefert werden kann. So lässt sich bspw. steuern, ob direkt zuzustellen ist oder ob eine Ablage ins Postfach vorgesehen ist.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 88</a>
-kennung	urn:xoev-de:xta:core:codeliste:abgabestation
-version	1.0

### 1.1.1.3.3.7 Code.Abweichverhalten

Codelisten	
-beschreibung	Diese Codeliste enthält die verschiedenen Vorgaben für das Abweichverhalten, das anzuwenden ist, wenn die zugehörige Vorgabe nicht erfüllt wird. Die Vorgabe wird im Service Profil festgelegt.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 89</a>
-kennung	urn:xoev-de:xta:core:codeliste:abweichverhalten
-version	1.0

## 1.1.2 Aufbau des Schutzprofils

### 1.1.2.1 schutzProfil

Globales Element: *schutzProfil*

Das Schutzprofil beschreibt Schutzziele des Datenschutzes zusammenfassend im Element *schutzkategorie* und der Datensicherheit im Element *sicherheitskategorie*.

Dieser Typ ist eine Erweiterung des Basistyps *SchutzProfilType* (siehe [Abschnitt 1.1.2.2 auf Seite 7](#)).

### 1.1.2.2 SchutzProfilType

Typ: *SchutzProfilType*

Grundstruktur für alle Schutzprofile

Kindelemente von <i>SchutzProfilType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>profilkopf</b>		1		
Dieses Element wird gefüllt mit Informationen zu Identität und Gültigkeit der vorliegenden Profil-Instanz.				
<b>identifikation</b>	<i>xtasp:IdentifikationType</i>	1	<a href="#">1.1.5.2</a>	<a href="#">23</a>
Identität und Herkunft der vorliegenden Profil-Instanz (informativ)				
<b>dokumentation</b>	<i>oscimeta:NonEmptyStringType</i>	1	<a href="#">2.2.1.40</a>	<a href="#">75</a>
Dokumentation der vorliegenden Profil-Instanz				
<b>gueltigkeit</b>	<i>xtasp:InstanzGueltigkeitType</i>	1	<a href="#">1.1.5.3</a>	<a href="#">23</a>
Gültigkeit der vorliegenden Profil-Instanz				
<b>schutzkategorie</b>	<i>xtasp:SchutzkategorieType</i>	1	<a href="#">1.1.2.3.1</a>	<a href="#">7</a>
In den Bereich der Schutzkategorie sind die Schutzziele des Datenschutzes (siehe XTA-Spezifikationsdokument Rahmenbedingungen) einsortiert, soweit für die XTA Service Profile relevant. Sie sind hier gruppiert unter die Oberbegriffe der Transparenz und der Intervenierbarkeit.				
<b>sicherheitskategorie</b>	<i>xtasp:SicherheitskategorieType</i>	1	<a href="#">1.1.2.3.2</a>	<a href="#">9</a>
Unter dem Bereich der Sicherheitskategorie stehen die Servicequalitäten der Datensicherheit, soweit für die XTA Service Profile relevant. Sie sind hier zugeordnet den Oberbegriffen Vertraulichkeit und Integrität.				

## 1.1.2.3 Datentypen des Schutzprofils

### 1.1.2.3.1 SchutzkategorieType

Typ: *SchutzkategorieType*

Bei der Erstellung einer Schutzprofil-Instanz für die Servicequalitäten ist jeweils eine Ausprägung und / oder ein Geltungsbereich anzugeben.

Kindelemente von <i>SchutzkategorieType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>transparenz</b>		1		
<p>Bei Transparenz geht es um Nachvollziehbarkeit. Wenn die Prozesse der Nachrichtenübermittlung transparent ausgestaltet sind, bieten sie Aufsichtsbehörden, dem Auftraggeber oder sonstigen Betroffenen (beispielsweise in den Rollen Autor, Sender, Empfänger, Leser) die Möglichkeit, Vorgehen und Ereignisse nachzuvollziehen bzw. zu überprüfen.</p> <p>Einschbare Protokolle und zugestellte Quittungen sind das XTA-Angebot, Transparenz zu unterstützen.</p>				
<b>protokollierung</b>		1..n		
<p>Protokolle / Reports gemäß XTA-Vorgaben dokumentieren die Bearbeitungsschritte und die Ergebnisse im Rahmen der Abarbeitung eines Transportauftrags. Sie werden durch die Knoten geführt, die an dieser Abarbeitung beteiligt sind und die entsprechenden Zugriff auf die MessageID des Transportauftrags haben.</p> <p>Dabei bedeutet „Protokollierung durch Rolle Autor“ nicht notwendigerweise, dass dies durch das Fachverfahren geschieht. Diese Aufgabe kann an das Transportverfahren delegiert sein.</p>				
<b>rolle</b>	<i>xtacore:Code.XTARolle</i>	1	1.1.4.3.9	22
<p>Hier ist einzutragen, in Bezug auf welche Rolle in der XTA-Infrastruktur hier Festlegungen zur Protokollierung eingetragen werden soll.</p>				
<b>qualitaet</b>	<i>xtasp:Code.QualitaetProtokollierung</i>	1	1.1.2.3. 3.4	12
<p>Hier ist einzutragen, ob durch den betreffenden Knoten ein Protokoll / ein Report zu führen ist und mit welchem Absicherungs niveau dies geschehen soll.</p>				
<b>abweichverhalten</b>	<i>xtacore:Code.Aabweichverhalten</i>	1	1.1.1.3. 3.7	7
<p>Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.</p>				
<b>technischeQuittung</b>		1..n		
<p>Unterhalb dieses Elements wird eingetragen, welche Quittungen in einem bestimmten Kontext gefordert sind.</p> <p>In einer XTA-Infrastruktur ist die Möglichkeit vorgesehen, dass Knoten der Infrastruktur durch Quittungen über entfernte Ereignisse der Abarbeitung eines Transportauftrags informiert werden. (vgl. XTA-Spezifikationsdokument Rahmenbedingungen)</p>				
<b>artDerQuittung</b>	<i>xtacore:Code.TechnischeQuittungen</i>	1	1.1.2.3. 3.5	12
<p>Hier ist eine der Quittungsarten einzutragen, welche in einer XTA-Infrastruktur vorgesehen sind (siehe XTA-Spezifikationsdokument Rahmenbedingungen). Pro Quittungsart, die im Schutzprofil vorgeschrieben werden soll, ist ein eigenes Element zu erstellen.</p>				
<b>abweichverhalten</b>	<i>xtacore:Code.Aabweichverhalten</i>	1	1.1.1.3. 3.7	7
<p>Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.</p>				
<b>intervenierbarkeit</b>		1		
<p>Wenn ein Verfahren intervenierbar ausgestaltet ist, bietet es der betroffenen Person oder Organisation die Möglichkeit, ihre Rechte in einem definierten Vorgehen kontrolliert zu wahren bzw. durchzusetzen.</p>				
<b>definiertesLoeschen</b>		1		
<p>Das Löschen der Daten und Protokolle erfolgt mit einer vorgegebenen Qualität, welche hier festgelegt wird. Der <i>Zeitpunkt</i> des Löschens wird an anderer Stelle (im Service Profil) geregelt. Diese Servicequalität ist nicht</p>				

Kindelemente von <i>SchutzkategorieType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
transportrelevant, da das Löschen der Daten nach dem Transport erfolgt. Dementsprechend muss eine Abweichung von dieser Servicequalität nicht im TransportReport eingetragen werden.				
<b>qualitaet</b>	<i>xtacore:Code.QualitaetLoeschen</i>	1	1.1.2.3. 3.3	12
In diesem Element wird die geforderten Ausprägung der Servicequalität „Löschen von personen- oder organisationsbezogenen Daten“ hinterlegt. Aus dieser geht hervor, auf welche Weise diese Daten zum vorgegebenen Zeitpunkt zu löschen sind.				
<b>abweichverhalten</b>	<i>xtacore:Code.Aabweichverhalten</i>	1	1.1.1.3. 3.7	7
Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.				

### 1.1.2.3.2 SicherheitskategorieType

Typ: *SicherheitskategorieType*

Unter dem Bereich der Sicherheitskategorie stehen die Servicequalitäten der Datensicherheit, soweit für die XTA Service Profile relevant. Sie sind hier den Oberbegriffen Vertraulichkeit und Integrität zugeordnet.

Gegebenenfalls ist bei der Erstellung einer Schutzprofil-Instanz eine Ausprägung (z.B. 'Vertraulichkeit hoch') und / oder ein Geltungsbereich (z.B. 'geltend für die Nachrichtenkommunikation auf der Strecke 'Sender-Empfänger') anzugeben, wofür bei den Unterelementen jeweils die benötigten Codelisten hinterlegt sind.

Kindelemente von <i>SicherheitskategorieType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>vertraulichkeit</b>		1..n		
Die Vertraulichkeit einer Nachricht oder von Daten ist die Eigenschaft, nur für einen abgegrenzten Empfängerkreis vorgesehen zu sein. Diese Servicequalität wird hier definiert bezogen auf einen bestimmten Geltungsbereich der Nachrichtenkommunikation (Element <i>geltungsbereich</i> ). Außerdem wird die Anforderung formuliert bezogen auf ein gefordertes Niveau der Vertraulichkeit (Element <i>qualitaet</i> ).				
Wenn mehrere Vorgaben zur Vertraulichkeit festgelegt werden, liegt es in der Zuständigkeit der Rolle Sender, die für einen gegebenen Transportauftrag geeignete Vorgabe unter Berücksichtigung des Geltungsbereichs auszuwählen. Die Auswahl muss im ServiceReport durch das Ereignis <i>AuswahlServicequalitaet</i> protokolliert werden.				
<b>geltungsbereich</b>	<i>xtasp:Code.GeltungsbereichSchutzprofil-Parameter</i>	1	1.1.2.3. 3.1	11
Hier ist einzutragen, für welche Teilstrecken der Nachrichtenkommunikation eine Vertraulichkeit gefordert werden soll.				
<b>qualitaet</b>	<i>xtasp:Code.QualitaetVertraulichkeit</i>	1	1.1.2.3. 3.7	12
Hier ist einzutragen, welches Niveau von Vertraulichkeit auf der entsprechenden Strecke gefordert ist.				
<b>abweichverhalten</b>	<i>xtacore:Code.Aabweichverhalten</i>	1	1.1.1.3. 3.7	7
Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.				
<b>integritaet</b>		1		
Unter Integrität werden die Unversehrtheit von Daten und das korrekte Funktionieren von Systemen verstanden.				

Kindelemente von <i>SicherheitskategorieType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>nachweisBetreiber</b>	<i>xtasp:Code.NachweisVerlaesslichkeit</i>	1	1.1.2.3. 3.8	12
<p>Organisatorische Anforderung an die Erklärung des Betreibers eines Fach- oder Transportverfahrens über seine Verlässlichkeit. Die Erklärung ist an einem definierten Ort zu hinterlegen.</p> <p>Die Einhaltung kann technisch nicht überprüft werden, der Ort, an dem die Erklärung zum Zeitpunkt des Transportes hinterlegt wurde, muss jedoch im Service-Profil durch das Ereignis <a href="#">ErklaerungHinterlegt</a> dokumentiert werden.</p>				
<b>nachweisHersteller</b>	<i>xtasp:Code.NachweisVerlaesslichkeit</i>	1	1.1.2.3. 3.8	12
<p>Organisatorische Anforderung an die Erklärung des Herstellers über die Verlässlichkeit der von ihm vorgelegten Software hins. Betriebs- und Prozessintegrität. Die Erklärung ist an einem definierten Ort zu hinterlegen.</p> <p>Die Einhaltung kann technisch nicht überprüft werden, der Ort, an dem die Erklärung zum Zeitpunkt des Transportes hinterlegt wurde, muss jedoch im Service-Profil durch das Ereignis <a href="#">ErklaerungHinterlegt</a> dokumentiert werden.</p>				
<b>authentizitaet</b>		1..n		
<p>Die Kommunikationsteilnehmer müssen sich der Identität des Kommunikationspartners vergewissern. In welchem Maß (Element <i>qualitaet</i>) und in welchem Kontext (Element <i>geltungsbereich</i>) das zu geschehen hat: dafür ist die Servicequalität der Authentizität zu definieren.</p> <p>Wenn mehrere Vorgaben zur Authentizität festgelegt werden, liegt es in der Zuständigkeit der Rolle Sender, die für einen gegebenen Transportauftrag geeignete Vorgabe unter Berücksichtigung des Geltungsbereichs auszuwählen. Die Auswahl muss im ServiceReport durch das Ereignis <a href="#">AuswahlServicequalitaet</a> protokolliert werden.</p>				
<b>geltungsbereich</b>	<i>xtasp:Code.GeltungsbereichSchutzprofil-Parameter</i>	1	1.1.2.3. 3.1	11
<p>Hier ist einzutragen, für welche Teilstrecken der Nachrichtenkommunikation die Authentizität abzusichern ist.</p>				
<b>qualitaet</b>	<i>xtasp:Code.QualitaetAuthentizitaet</i>	1	1.1.2.3. 3.2	11
<p>Hier ist einzutragen, welches Niveau von Authentizität auf der entsprechenden Strecke gefordert ist.</p>				
<b>abweichverhalten</b>	<i>xtacore:Code.Aabweichverhalten</i>	1	1.1.1.3. 3.7	7
<p>Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.</p>				
<b>unveraenderbarkeit</b>		1..n		
<p>Die Servicequalität der Unveränderbarkeit ist die Anforderung, dass die Daten im Zuge ihrer Übertragung durch die Messaging Infrastruktur nicht verändert werden können. In welchem Maß (Element <i>qualitaet</i>) und in welchem Kontext (Element <i>geltungsbereich</i>) diese Anforderung besteht wird in einer Instanz dieses Typs definiert.</p> <p>Wenn mehrere Vorgaben zur Unveränderbarkeit festgelegt werden, liegt es in der Zuständigkeit der Rolle Sender, die für einen gegebenen Transportauftrag geeignete Vorgabe unter Berücksichtigung des Geltungsbereichs auszuwählen. Die Auswahl muss im ServiceReport durch das Ereignis <a href="#">AuswahlServicequalitaet</a> protokolliert werden.</p>				
<b>geltungsbereich</b>	<i>xtasp:Code.GeltungsbereichSchutzprofil-Parameter</i>	1	1.1.2.3. 3.1	11
<p>Hier ist einzutragen, für welche Teilstrecken der Nachrichtenkommunikation die Unveränderbarkeit abzusichern ist.</p>				

Kindelemente von <i>SicherheitskategorieType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>qualitaet</b>	<i>xtasp:Code.QualitaetUnveraenderbarkeit</i>	1	1.1.2.3. 3.6	12
Hier ist einzutragen, welches Niveau von Unveränderbarkeit auf der entsprechenden Strecke gefordert ist.				
<b>abweichverhalten</b>	<i>xtacore:Code.Aabweichverhalten</i>	1	1.1.1.3. 3.7	7
Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.				
<b>zertifikat</b>		1..n		
Für die durch die Beteiligten zu verwendenden Zertifikate lassen sich hier Vorgaben formulieren, indem das geforderte Zertifikatsniveau und das geforderte Zertifikatsmedium angegeben wird.				
Zur Quelle der Zertifikate werden die (spezifisch für den Service zu treffenden) Festlegungen nicht hier formuliert, sondern in der Servicekategorie.				
Wenn mehrere Vorgaben zu den zu verwendenden Zertifikaten festgelegt werden, liegt es in der Zuständigkeit der Rolle Sender, die für einen gegebenen Transportauftrag geeignete Vorgabe unter Berücksichtigung des Niveaus und des Mediums auszuwählen. Die Auswahl muss im ServiceReport durch das Ereignis AuswahlServicequalitaet protokolliert werden.				
<b>rolle</b>	<i>xtacore:Code.XTARolle</i>	1	1.1.4.3.9	22
Hier ist der Beteiligte in einer XTA-Kommunikationsinfrastruktur zu nennen, in Bezug auf dessen Zertifikate hier Anforderungen definiert werden.				
<b>niveau</b>	<i>xtacore:Code.ZertifikatNiveau</i>	1	1.1.2.3. 3.10	13
Hier ist das geforderte Niveau der Zertifikate zu benennen.				
<b>medium</b>	<i>xtacore:Code.ZertifikatMedium</i>	1	1.1.2.3. 3.9	13
Hier ist das geforderte Medium der Zertifikate zu benennen.				
<b>abweichverhalten</b>	<i>xtacore:Code.Aabweichverhalten</i>	1	1.1.1.3. 3.7	7
Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.				

### 1.1.2.3.3 Code-Datentypen des Schutzprofils

#### 1.1.2.3.3.1 Code.GeltungsbereichSchutzprofil-Parameter

Codelisten	
-beschreibung	Diese Codeliste definiert Kommunikationsstrecken auf die sich ein Schutzprofil beziehen kann.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 95</a>
-kennung	urn:xoev-de:xta:serviceprofile:codeliste:geltungsbereich.schutzprofil-parameter
-version	1.0

#### 1.1.2.3.3.2 Code.QualitaetAuthentizitaet

Codelisten	
-beschreibung	Diese Codeliste definiert Niveaus, mit dem die Authentizität der Nachrichtenkommunikation abgesichert werden soll.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 99</a>

Codelisten	
-kennung	urn:xoev-de:xta:serviceprofile:codeliste:authentizitaetqualitaet
-version	1.0

#### 1.1.2.3.3.3 Code.QualitaetLoeschen

Codelisten	
-beschreibung	Diese Codeliste definiert Anforderungen an das Löschen.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 101</a>
-kennung	urn:xoev-de:xta:core:codeliste:loeschen.qualitaet
-version	1.0

#### 1.1.2.3.3.4 Code.QualitaetProtokollierung

Codelisten	
-beschreibung	Diese Codeliste legt Ausprägungen der Protokollierung fest.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 102</a>
-kennung	urn:xoev-de:xta:serviceprofile:codeliste:protokollierung.qualitaet
-version	1.0

#### 1.1.2.3.3.5 Code.TechnischeQuittungen

Codelisten	
-beschreibung	Diese Codeliste enthält die Schlüssel für die Arten technischer Quittungen, welche in einer XTA-Infrastruktur vorgesehen sind (siehe XTA-Spezifikationsdokument Rahmenbedingungen).
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 108</a>
-kennung	urn:xoev-de:xta:core:codeliste:technische.quittungen
-version	1.0

#### 1.1.2.3.3.6 Code.QualitaetUnveraenderbarkeit

Codelisten	
-beschreibung	Diese Codeliste definiert den Umfang, in dem die Unveränderbarkeit (Integrität) geschützt wird.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 103</a>
-kennung	urn:xoev-de:xta:serviceprofile:codeliste:unveraenderbarkeit.qualitaet
-version	1.0

#### 1.1.2.3.3.7 Code.QualitaetVertraulichkeit

Codelisten	
-beschreibung	Diese Codeliste definiert Vertraulichkeitsniveaus.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 105</a>
-kennung	urn:xoev-de:xta:serviceprofile:codeliste:vertraulichkeit.qualitaet
-version	1.0

#### 1.1.2.3.3.8 Code.NachweisVerlaesslichkeit

Codelisten	
-beschreibung	Die Codeliste definiert die Stufen der Verlässlichkeit einer Organisation.



Codelisten	
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 98</a>
-kennung	urn:xoev-de:xta:serviceprofile:codeliste:nachweis.verlaesslichkeit
-version	1.0

#### 1.1.2.3.3.9 Code.ZertifikatMedium

Codelisten	
-beschreibung	Diese Codeliste benennt die verschiedenen Medien, die ein Zertifikat tragen können.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 116</a>
-kennung	urn:xoev-de:xta:core:codeliste:zertifikat.medium
-version	1.0

#### 1.1.2.3.3.10 Code.ZertifikatNiveau

Codelisten	
-beschreibung	Diese Codeliste enthält die Schlüssel für die definierten Niveaus eines Zertifikats.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 117</a>
-kennung	urn:xoev-de:xta:core:codeliste:zertifikat.niveau
-version	1.0

## 1.1.3 Aufbau des Infrastrukturprofils

### 1.1.3.1 infrastrukturProfil

Globales Element: *infrastrukturProfil*

Legt Eigenschaften der für einen Transport benötigten bzw. einzusetzenden Infrastrukturkomponenten zusammenfassend im Element *infrastrukturkategorie*.

Dieser Typ ist eine Erweiterung des Basistyps *InfrastrukturProfilType* (siehe [Abschnitt 1.1.3.2 auf Seite 13](#)).

### 1.1.3.2 InfrastrukturProfilType

Typ: *InfrastrukturProfilType*

Grundstruktur für alle Infrastrukturprofile

Kindelemente von <i>InfrastrukturProfilType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>profilkopf</b>		1		
Dieses Element wird gefüllt mit Informationen zu Identität und Gültigkeit der vorliegenden Profil-Instanz.				
<b>identifikation</b>	<i>xtasp:IdentifikationType</i>	1	<a href="#">1.1.5.2</a>	<a href="#">23</a>
Identität und Herkunft der vorliegenden Profil-Instanz (informativ)				
<b>dokumentation</b>	<i>oscimeta:NonEmptyStringType</i>	1	<a href="#">2.2.1.40</a>	<a href="#">75</a>
Dokumentation der vorliegenden Profil-Instanz				
<b>gueltigkeit</b>	<i>xtasp:InstanzGueltigkeitType</i>	1	<a href="#">1.1.5.3</a>	<a href="#">23</a>
Gültigkeit der vorliegenden Profil-Instanz				
<b>infrastrukturkategorie</b>	<i>xtasp:InfrastrukturkategorieType</i>	1	<a href="#">1.1.3.3.1</a>	<a href="#">14</a>

Kindelemente von <i>InfrastrukturProfilType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Unter diese Kategorie fallen alle Servicequalitäten, die dazu dienen, die Infrastrukturkomponenten für einen Transport festzulegen.				

### 1.1.3.3 Datentypen des Infrastrukturprofils

#### 1.1.3.3.1 InfrastrukturkategorieType

Typ: *InfrastrukturkategorieType*

Beschreibt die für einen Transport benötigten bzw. einzusetzenden Verzeichnisdienste und Übertragungswege.

Kindelemente von <i>InfrastrukturkategorieType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>geltungsbereich</b>	<i>xtasp:Code.GeltungsbereichInfrastrukturprofil-Parameter</i>	1	1.1.3.3.2.1	17
Geltungsbereich für das vorliegende Infrastrukturprofil				
<b>kanal</b>		1		
Vorgaben zum Übertragungskanal				
<b>bezeichnung</b>	<i>xtacore:Code.Kanal</i>	1..n	1.1.3.3.2.4	17
Kommunikationskanal zwischen Sender und Empfänger (Art der technischen Verbindung oder das Netz, über das kommuniziert wird). Wenn mehrere Kanäle vorgegeben werden, liegt es in der Zuständigkeit der Rolle Sender, die für einen gegebenen Transportauftrag geeignete Vorgabe auszuwählen. Die Auswahl muss im ServiceReport durch das Ereignis AuswahlServicequalitaet protokolliert werden. Wenn keiner der angegebenen Kanäle verwendet werden kann, wurde die Servicequalität nicht erfüllt.				
<b>abweichverhalten</b>	<i>xtacore:Code.Aabweichverhalten</i>	1	1.1.1.3.3.7	7
Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.				
<b>transportnachrichtenformat</b>		1		
Vorgaben zum Format der Transportnachrichten				
<b>bezeichnung</b>	<i>xtacore:Code.TransportnachrichtenFormat</i>	1	1.1.3.3.2.5	18
In diesem Element wird das Nachrichtenformat eingetragen, in dem die Daten zwischen Sender und Empfänger zu übertragen sind.				
<b>abweichverhalten</b>	<i>xtacore:Code.Aabweichverhalten</i>	1	1.1.1.3.3.7	7
Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.				
<b>transportprotokoll</b>		1		
Vorgaben zum Transportprotokoll				
<b>bezeichnung</b>	<i>xtacore:Code.Transportprotokoll</i>	1	1.1.3.3.2.6	18

Kindelemente von <i>InfrastrukturkategorieType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Dieses Element legt das Protokoll fest, das für die Kommunikation der Nachrichten zwischen Sender und Empfänger zu verwenden ist.				
<b>abweichverhalten</b>	<i>xtacore:Code.Abweichverhalten</i>	1	1.1.1.3. 3.7	7
Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.				
<b>adressierung</b>		1		
Vorgaben zur Adressierung				
<b>adressverzeichnis</b>	<i>xtacore:Code.VerzeichnisAdressierung</i>	1	1.1.3.3. 2.2	17
Hier wird die Bezeichnung des Verzeichnisdienstes eingetragen, der für jeden Teilnehmer des Nachrichtenaustauschs die Parameter für die technische Adressierung von Teilnehmern bereitstellt.				
<b>abweichverhalten</b>	<i>xtacore:Code.Abweichverhalten</i>	1	1.1.1.3. 3.7	7
Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.				
<b>identifizierung</b>		1		
Vorgaben zur Identifizierung				
<b>identifizierungsverzeichnis</b>	<i>xtasp:Code.VerzeichnisIdentifizierung</i>	1	1.1.3.3. 2.3	17
Hier ist der Verzeichnisdienst genannt, der zu allen Teilnehmern die Parameter und Entitäten für Identität und Identitätsnachweis bereitstellt.				
<b>abweichverhalten</b>	<i>xtacore:Code.Abweichverhalten</i>	1	1.1.1.3. 3.7	7
Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.				
<b>zertifikate</b>		1		
Rollenbezogene Vorgaben zu Zertifikaten				
<b>rolle</b>	<i>xtacore:Code.XTARolle</i>	1	1.1.4.3.9	22
Hier wird die Rolle in der XTA-Infrastruktur (Autor, Sender, ...) benannt, auf die sich die Vorgabe bezieht.				
<b>quelle</b>	<i>xtasp:Code.ZertifikatQuelle</i>	1	1.1.1.3. 3.1	5
Hier ist eine Vorgabe in Bezug auf die Quelle der zugeordneten Zertifikate einzutragen.				
<b>abweichverhalten</b>	<i>xtacore:Code.Abweichverhalten</i>	1	1.1.1.3. 3.7	7
Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.				
<b>protokollfuehrung</b>		1		
Festlegungen zur Protokollführung in Transportprozessen				
Thema ist hier Vorhaltung und Löschung der Protokollinhalte. Die <i>Qualität</i> der Löschung (wie sie durchzuführen ist) wird an anderer Stelle (im Schutzprofil) geregelt.				
Diese Festlegungen gelten für alle Transporteure in den Rollen Sender und Empfänger.				
<b>vorhaldedauer</b>	<i>xs:positiveInteger</i>	1		

Kindelemente von <i>InfrastrukturkategorieType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Es ist die Anzahl Tage einzutragen, für die die Protokolle mindestens vorzuhalten sind.				
<b>startzeitpunktLoeschfrist</b>	<i>xtacore:Code.StartzeitpunktLoeschfrist</i>	1	1.1.1.3. 3.4	6
<b>loeschfrist</b>	<i>xs:positiveInteger</i>	1		
Hier ist die Höchstspeicherzeit für ggf. erstellte Protokolle einzutragen. Es ist die Anzahl Tage einzutragen, nach deren Verstreichen die Protokolle spätestens zu löschen sind.				
<b>abweichverhalten</b>	<i>xtacore:Code.Aabweichverhalten</i>	1	1.1.1.3. 3.7	7
Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.				
<b>datenhaltung</b>		1		
Festlegungen zum Umgang mit Daten - also mit den Fachnachrichten - in einer XTA-Infrastruktur eingetragen. Diese Festlegungen gelten für alle Transporteure in den Rollen Sender und Empfänger.				
<b>loeschfrist</b>	<i>xs:positiveInteger</i>	1		
Generell gilt, dass durch den Sender erfolgreich übermittelte Daten auf Seiten des Senders anschließend zu löschen sind ("erfolgreich übermittelt" bedeutet, dass die Daten im Zugriffsbereich des Empfängers angekommen sind). Dieses Element dient der Festlegung, wann (a) nicht-vermittelte Daten durch den Sender bzw. den Empfänger zu löschen sind und wann (b) vermittelte Daten vom Empfänger zu löschen sind. Es ist in das Element die Höchstspeicherzeit für die vorgehaltenen Daten einzutragen. Es ist die Anzahl Tage einzutragen, nach deren Verstreichen die Daten spätestens zu löschen sind.				
<b>abweichverhalten</b>	<i>xtacore:Code.Aabweichverhalten</i>	1	1.1.1.3. 3.7	7
Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.				
<b>abgabestation</b>		1		
Vorgaben zum Infrastrukturknoten, der das Ende des Übertragungsweges der Fachnachricht ist				
<b>abweichverhalten</b>	<i>xtacore:Code.Aabweichverhalten</i>	1	1.1.1.3. 3.7	7
Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.				
<b>zielknoten</b>	<i>xtacore:Code.Abgabestation</i>	1	1.1.1.3. 3.6	6
Knoten der Infrastruktur, an dem die fachliche Nachricht final abzuliefern ist				
<b>maxNachrichtengroesse</b>		0..1		
Größenbeschränkung für den Payload der Transportnachricht Falls eine Prüfung durchgeführt wird, muss die Auswahl im ServiceReport durch das Ereignis AuswahlServicequalitaet protokolliert werden. Hierbei ist der Name des Elementes und die ermittelte Nachrichtengröße eingetragen werden.				
<b>maxNachrichtengroesse</b>	<i>xs:positiveInteger</i>	1		
Dieses Element ermöglicht es, eine Größenbeschränkung für den gesamten übergebenen Payload der Transportnachricht festzulegen. Die Größenbeschränkung ist als positive ganze Zahl anzugeben und bezeichnet eine Anzahl Megabyte (MB).				

Kindelemente von <i>InfrastrukturkategorieType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
	Größenbeschränkungen können sich zum Beispiel aus den technischen Grenzen der Fachverfahren und aus den technischen Grenzen der Infrastruktur ergeben.			
<b>abweichverhalten</b>	<i>xtacore:Code.Abweichverhalten</i>	<b>1</b>	<a href="#">1.1.1.3.3.7</a>	<a href="#">7</a>
	Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.			

### 1.1.3.3.2 Code-Datentypen des Infrastrukturprofils

#### 1.1.3.3.2.1 Code.GeltungsbereichInfrastrukturprofil-Parameter

Codelisten	
-beschreibung	Diese Codeliste definiert Kommunikationsstrecken für den Geltungsbereich eines Infrastrukturprofils.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 94</a>
-kennung	urn:xoev-de:xta:serviceprofile:codeliste:geltungsbereich.infrastrukturprofil-parameter
-version	1.0

#### 1.1.3.3.2.2 Code.VerzeichnisAdressierung

Codelisten	
-beschreibung	Diese Codeliste enthält die Schlüssel für Verzeichnislösungen zur Bereitstellung von Parametern für die technische Adressierung von Teilnehmern.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 111</a>
-kennung	urn:xoev-de:xta:core:codeliste:verzeichnis.adressierung
-version	1.0

#### 1.1.3.3.2.3 Code.VerzeichnisIdentifizierung

Codelisten	
-beschreibung	Diese Codeliste enthält die Schlüssel für Verzeichnislösungen zur Verwaltung von elektronischen Identitäten (Bezeichnungen und kryptographische Token für Identität und Identitätsnachweis).
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 112</a>
-kennung	urn:xoev-de:xta:serviceprofile:codeliste:verzeichnis.identifizierung
-version	1.0

#### 1.1.3.3.2.4 Code.Kanal

Codelisten	
-beschreibung	Diese Codeliste definiert die möglichen Netze über die eine Nachricht zwischen Sender und Empfänger übermittelt werden darf.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 96</a>
-kennung	urn:xoev-de:xta:core:codeliste:kanal
-version	1.0

### 1.1.3.3.2.5 Code.TransportnachrichtenFormat

Codelisten	
-beschreibung	Diese Codeliste nennt verfügbare Nachrichtenformate für Transportnachrichten.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 109</a>
-kennung	urn:xoev-de:xta:core:codeliste:transportnachrichten.format
-version	1.0

### 1.1.3.3.2.6 Code.Transportprotokoll

Codelisten	
-beschreibung	Diese Codeliste nennt verfügbare Protokolle, die die Kommunikation von Daten zwischen Partnern festlegen.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 110</a>
-kennung	urn:xoev-de:xta:core:codeliste:transportprotokoll
-version	1.0

## 1.1.4 Aufbau des Kryptographieprofils

### 1.1.4.1 kryptographieProfil

Globales Element: *kryptographieProfil*

Legt Anforderungen an die kryptographische Absicherung von Nachrichten zusammenfassend im Element *kryptoSuiten* fest.

Dieser Typ ist eine Erweiterung des Basistyps *KryptographieProfilType* (siehe [Abschnitt 1.1.4.2 auf Seite 18](#)).

### 1.1.4.2 KryptographieProfilType

Typ: *KryptographieProfilType*

Grundstruktur für alle Kryptographieprofile

Kindelemente von <i>KryptographieProfilType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>profilkopf</b>		<b>1</b>		
Dieses Element wird gefüllt mit Informationen zu Identität und Gültigkeit der vorliegenden Profil-Instanz. Hier ist die Versionsnummer der Kryptographieprofil-Instanz einzutragen.				
<b>identifikation</b>	<i>xtasp:IdentifikationType</i>	<b>1</b>	<a href="#">1.1.5.2</a>	<a href="#">23</a>
Identität und Herkunft der vorliegenden Profil-Instanz (informativ)				
<b>dokumentation</b>	<i>oscmeta:NonEmptyStringType</i>	<b>1</b>	<a href="#">2.2.1.40</a>	<a href="#">75</a>
Dokumentation der vorliegenden Profil-Instanz				
<b>gultigkeit</b>	<i>xtasp:InstanzGultigkeitType</i>	<b>1</b>	<a href="#">1.1.5.3</a>	<a href="#">23</a>
Gültigkeit der vorliegenden Profil-Instanz				
<b>datumVeroeffentlichung</b>	<i>xs:date</i>	<b>1</b>		
Hier ist das Datum einzutragen, an dem die Kryptographieprofil-Instanz veröffentlicht wurde.				
<b>kryptoSuiten</b>	<i>xtasp:CryptoSuitesType</i>	<b>1</b>	<a href="#">1.1.4.3.1</a>	<a href="#">19</a>

Kindelemente von <i>KryptographieProfilType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Unterhalb dieses Elements wird der Inhalt des Kryptographieprofils dargestellt. Dies sind die Definitionen der bei der Umsetzung der Anforderungen aus Infrastruktur- und Schutzprofilen zu verwendenden kryptographischen Mittel.				
Im Zusammenhang wird dies näher erläutert im XTA-Spezifikationsdokument Rahmenbedingungen.				

### 1.1.4.3 Datentypen des Kryptographieprofils

Die Typen, die hier dargestellt werden, werden im Kontext einer XML-Instanz angewendet, die auf dem globalen Element [kryptographieProfil](#) basiert.

#### 1.1.4.3.1 CryptoSuitesType

Typ: *CryptoSuitesType*

Dieser Typ bietet den Rahmen, um ein Set von Krypto-Suiten zu definieren, die für die kryptographische Umsetzung der Anforderungen aus Schutz- und Infrastrukturprofilen benötigt werden.

Kindelement von <i>CryptoSuitesType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>cryptoSuite</b>		1..n		
Ein Element bildet die Definition einer Krypto-Suite ab. Eine Krypto-Suite ist eine Zusammenstellung von kryptographischen Mitteln (Algorithmen und Schlüssellängen), jeweils zugeordnet einem bestimmten Schutzniveau einer kryptographisch abzusichernden Kommunikation.				
Wenn mehrere Verfahren vorgegeben werden, liegt es in der Zuständigkeit der Rolle Sender, das für einen gegebenen Transportauftrag geeignete Verfahren unter Berücksichtigung der Qualitätsanforderung in <i>cryptoQuality</i> auszuwählen. Die Auswahl muss im ServiceReport durch das Ereignis AuswahlServicequalitaet protokolliert werden.				
Wenn keine der angegebenen Krypto-Suiten verwendet werden konnte, ist die Servicequalität nicht erfüllt.				
<b>target</b>	<i>xs:string</i>	1		
Auf die Umsetzung dieses Formats oder Protokolls bezieht sich diese Krypto-Suite. Vorgesehen sind Krypto-Suiten für die Umsetzung von W3C (z.B. für OSCI 1.2, OSCI 2), TLS und des Payloads eines Transportnachrichtenformats.				
<b>cryptoQuality</b>	<i>xtacore:Code.QualitaetKryptographie</i>	1	1.1.4.3.8	22
Eine Krypto-Qualität benennt eine Abstufung (z. B. hoch, normal, niedrig) des geforderten Schutzniveaus einer kryptographisch zu sichernden Kommunikation.				
Das hier einzustellende Niveau korrespondiert mit den Elementen namens <i>qualitaet</i> aus dem Schutzprofil in den Kontexten der Servicequalitäten Authentizität, Unveränderbarkeit und Vertraulichkeit (vgl. <a href="#">Abschnitt 1.1.2.3.1</a> , „ <i>SchutzkategorieType</i> “ und <a href="#">Abschnitt 1.1.2.3.2</a> , „ <i>SicherheitskategorieType</i> “).				
<b>cryptoAssignment</b>		1..n		
Unterhalb dieses Elements wird einem bestimmten Schutzniveau einer kryptographisch abzusichernden Kommunikation (gelistete Krypto-Qualität, festgelegt durch das Element <i>cryptoQuality</i> ) ein Set von Algorithmen und Schlüssellängen zugeordnet.				
Wenn mehrere Algorithmen vorgegeben werden, liegt es in der Zuständigkeit der Rolle Sender, den für einen gegebenen Transportauftrag geeigneten Algorithmus unter Berücksichtigung der Qualitätsanforderung in <i>cryptoQuality</i> auszuwählen. Die Auswahl muss im ServiceReport durch das Ereignis AuswahlServicequalitaet protokolliert werden.				
Wenn keiner der angegebenen Algorithmen verwendet werden konnte, ist die Servicequalität nicht erfüllt.				
Auswahl: <b>w3cCipherSuite</b>	<i>xtasp:W3CCipherSuiteType</i>	1	1.1.4.3.5	21

Kindelement von <i>CryptoSuitesType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Auswahl: <b>cipherSuite</b>	<i>xtasp:CipherSuiteType</i>	1	1.1.4.3.4	20
<b>abweichverhalten</b>	<i>xtacore:Code.Aabweichverhalten</i>	1	1.1.1.3.3.7	7
Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.				

### 1.1.4.3.2 Algorithm

Typ: *Algorithm*

Dieser Typ beinhaltet Parameter für Identifikation und Beschreibung eines kryptographischen Algorithmus.

Kindelemente von <i>Algorithm</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>algorithmidentifizier</b>	<i>xtasp:Code.KryptographischeAlgorithmen</i>	1	1.1.4.3.6	21
Hier steht die Zeichenfolge, die zur Identifizierung des Algorithmus dient.				
<b>notAfter</b>	<i>xs:date</i>	0..1		
Hier ist das Datum zu nennen, bis zu dem der Algorithmus gültig ist.				
<b>remark</b>	<i>xs:string</i>	0..1		
Hier besteht die Möglichkeit, eine Bemerkung zu diesem Algorithmus einzutragen.				

### 1.1.4.3.3 AlgorithmAsymmetricEncryption

Typ: *AlgorithmAsymmetricEncryption*

Dieser Typ bildet einen Algorithmus für asymmetrische Verschlüsselung ab. Er ergänzt den Typ *Algorithmus* um einen weiteren Parameter.

Dieser Typ ist eine Erweiterung des Basistyps *Algorithm* (siehe [Abschnitt 1.1.4.3.2 auf Seite 20](#)).

Kindelement von <i>AlgorithmAsymmetricEncryption</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>keySize</b>	<i>xtasp:Code.KryptographischeSchluessellaenge</i>	1	1.1.4.3.7	21
Hier ist die zu verwendende Schlüssellänge einzutragen.				

### 1.1.4.3.4 CipherSuiteType

Typ: *CipherSuiteType*

Dieser Typ bildet eine Algorithmus-Deklaration für Cipher-Suiten ab.

Dieser Typ ist eine Erweiterung des Basistyps *Algorithm* (siehe [Abschnitt 1.1.4.3.2 auf Seite 20](#)).

Kindelement von <i>CipherSuiteType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>source</b>	<i>xs:string</i>	0..1		



Kindelement von <i>CipherSuiteType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Die Quelle beschreibt die Herkunft des Algorithmus. Hier kann z. B. „WS-Security policy Spec“ oder „BSI Algo Catalog“ stehen.				

### 1.1.4.3.5 W3CCipherSuiteType

Typ: *W3CCipherSuiteType*

Dieser Typ bildet eine Algorithmus-Deklaration für Cipher-Suiten ab.

Dieser Typ ist eine Erweiterung des Basistyps *Algorithm* (siehe [Abschnitt 1.1.4.3.2 auf Seite 20](#)).

Kindelemente von <i>W3CCipherSuiteType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>source</b>	<i>xs:string</i>	<b>0..1</b>		
Die Quelle beschreibt die Herkunft des Algorithmus. Hier kann z. B. „WS-Security policy Spec“ oder „BSI Algo Catalog“ stehen.				
<b>digest</b>	<i>xtasp:Algorithm</i>	<b>0..n</b>	<a href="#">1.1.4.3.2</a>	<a href="#">20</a>
Algorithmus-Deklaration für Digests ohne Erweiterungen.				
<b>signature</b>	<i>xtasp:Algorithm</i>	<b>0..n</b>	<a href="#">1.1.4.3.2</a>	<a href="#">20</a>
Algorithmus-Deklaration für Signaturen ohne Erweiterungen.				
<b>symmetricEncryption</b>	<i>xtasp:Algorithm</i> (Basistyp)	<b>0..n</b>	<a href="#">1.1.4.3.2</a>	<a href="#">20</a>
Algorithmus-Deklaration für symmetrische Verschlüsselung inkl. Schlüsselverschlüsselung.				
<b>keyEncryption</b>	<i>xtasp:AlgorithmAsymmetricEncryption</i>	<b>0..n</b>	<a href="#">1.1.4.3.3</a>	<a href="#">20</a>
Hier sind Algorithmen und Parameter zur Schlüsselverschlüsselung aufgeführt.				
<b>asymmetricEncryption</b>	<i>xtasp:AlgorithmAsymmetricEncryption</i>	<b>0..n</b>	<a href="#">1.1.4.3.3</a>	<a href="#">20</a>
Algorithmus-Deklaration für asymmetrische Verschlüsselung inkl. Schlüssellänge.				

### 1.1.4.3.6 Code.KryptographischeAlgorithmen

Codelisten	
-beschreibung	Diese Codeliste definiert die in den Service Profilen zu verwendenden eindeutigen Bezeichner kryptographischer Verfahren. Es handelt sich um Bezeichner nach "XML Security Algorithm Cross-Reference" der W3 Gruppe, verfügbar unter <a href="http://www.w3.org/TR/xmlsec-algorithms/">http://www.w3.org/TR/xmlsec-algorithms/</a> . In diesem Dokument der W3 Gruppe ist jedem Verfahren eine eindeutige URI zugeordnet, diese URI wird in XTA als Bezeichner für kryptographische Verfahren verwendet und ist in der Spalte „Wert“ aufgeführt. Die Spalte „Schlüssel“ enthält den Namen des Verfahrens, wie er im Dokument der W3 verwendet wird.
-nutzung	Typ: 3, siehe Beschreibung
-kennung	urn:xoev-de:xta:serviceprofile:codeliste:kryptographische-algorithmen
-version	unbestimmt

### 1.1.4.3.7 Code.KryptographischeSchluessellaenge

Codelisten	
-beschreibung	Diese Codeliste enthält die zu verwendenden minimalen Schlüssellängen für asymmetrische Verschlüsselungsverfahren.

Codelisten	
-nutzung	Typ: 3, siehe Beschreibung
-kennung	urn:xoev-de:xta:serviceprofile:codeliste:kryptographische-schluessellaengen
-version	unbestimmt

### 1.1.4.3.8 Code.QualitaetKryptographie

Codelisten	
-beschreibung	Diese Codeliste enthält die Schlüssel zur Angabe des geforderten Schutzniveaus einer kryptographisch zu sichernden Kommunikation.  Die hier verwendeten Abstufungen basieren auf den vom BSI im Kontext der Schutzbedarf-Feststellung definierten Begriffen zum IT-Grundschutz .
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 100</a>
-kennung	urn:xoev-de:xta:core:codeliste:kryptographie.qualitaet
-version	1.0

### 1.1.4.3.9 Code.XTARolle

Codelisten	
-beschreibung	Diese Codeliste benennt die Rollen, die in einer XTA-Kommunikationsinfrastruktur am Prozess der Nachrichtenübermittlung beteiligt sind.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 113</a>
-kennung	urn:xoev-de:xta:core:codeliste:xta-rolle
-version	1.0

## 1.1.5 Basistypen für Profile

Hier werden die Typen dargestellt, die in mehr als einer Profilart angewendet werden.

### 1.1.5.1 InstanzIdentifikationType

Typ: *InstanzIdentifikationType*

Typ für die Identifikation einer Profil-Instanz.

Lässt sich sowohl einsetzen, um innerhalb einer Profilinstanz ihre Identität einzutragen als auch für die Referenzierung auf eine Profilinstanz.

Kindelemente von <i>InstanzIdentifikationType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>version</b>	<i>xs:string</i>	<b>1</b>		
Hier wird die Versionsbezeichnung der Profil-Instanz eingetragen. Die Versionsbezeichnung ist aus Zahlen, ggf. kombiniert mit Punkten, zu bilden. Weitere Zeichen (wie z. B. Unterstriche) sind nicht vorgesehen. Beispiel für eine Versionsbezeichnung: „1.1“				
<b>uri</b>	<i>xs:anyURI</i>	<b>1</b>		
Hier ist die URI einzutragen, durch die eine Profil-Instanz identifiziert wird. Die Bezeichnung hat die Form einer URN.				

Kindelemente von <i>InstanzIdentifikationType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Dieser Bezeichner wird versionsübergreifend interpretiert, er darf also keine Angabe einer Versionsnummer enthalten.				
Es sind die Bildungsregeln laut XÖV-Handbuch anzuwenden. Beispiel: <i>urn:xoev-de:xta:schutzprofile:fensterbriefumschlag</i>				

### 1.1.5.2 IdentifikationType

Typ: *IdentifikationType*

Dieser Typ wird verwendet zur Darstellung der Parameter zu Identität und Herkunft eines Profils (versionsübergreifend verstanden) bzw. einer Profil-Instanz.

Dieser Typ ist eine Erweiterung des Basistyps *InstanzIdentifikationType* (siehe [Abschnitt 1.1.5.1 auf Seite 22](#)).

Kindelemente von <i>IdentifikationType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>uriDerVersion</b>	<i>xs:anyURI</i>	1		
Hier ist eine URI einzutragen, durch die diese Profil-Instanz identifiziert wird. Die Bezeichnung hat die Form einer URN mit angehängter Versionsnummer.				
Beispiel: <i>urn:xoev-de:xta:schutzprofile:fensterbriefumschlag_1.0.1</i>				
<b>bezeichnungKurz</b>	<i>xs:string</i>	1		
Hier wird die Kurzbezeichnung des Profils eingetragen (versionsübergreifend). Sie soll nach Möglichkeit aus einem oder zwei Worten bestehen.				
<b>bezeichnungLang</b>	<i>xs:string</i>	1		
Hier ist die vollständige Bezeichnung des Profils (versionsübergreifend) einzutragen. Sie kann aus mehreren Begriffen bestehen.				
<b>herausgeber</b>		1		
Unterhalb dieses Element sind die Daten des Herausgebers des Profils eingetragen.				
<b>bezeichnungKurz</b>	<i>xs:string</i>	1		
Hier wird die Kurzbezeichnung des Herausgebers eingetragen.				
<b>bezeichnungLang</b>	<i>xs:string</i>	1		
Hier ist die vollständige Bezeichnung des Herausgebers einzutragen.				

### 1.1.5.3 InstanzGueltigkeitType

Typ: *InstanzGueltigkeitType*

Typ für die Angabe der Gültigkeit (von, bis) einer Profil-Instanz.

Kindelemente von <i>InstanzGueltigkeitType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>gueltigkeitVon</b>	<i>xs:date</i>	1		
Hier ist, falls eine solche Festlegung vorgesehen ist, der Tag einzutragen, an dem die Gültigkeit der Profil-Instanz beginnt.				
<b>gueltigkeitBis</b>	<i>xs:date</i>	0..1		

Kindelemente von <i>InstanzGueltigkeitType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Hier ist, falls eine solche Festlegung vorgesehen ist, der letzte Tag der Gültigkeit der Profil-Instanz einzutragen.				

## 1.2 Protokollierung der Servicequalitäten

### 1.2.1 Basistypen für den Service Report

In diesem Abschnitt werden die Grundstrukturen eines Ereignisses und von Nachweisen festgelegt.

#### 1.2.1.1 EreignisType

Typ: *EreignisType* (abstrakt)

Die Grundstruktur für alle Ereignisse. Als Grundbestandteil wird der *EreignisType* nie direkt verwendet. An seiner Stelle wird immer ein daraus abgeleiteter, konkreter Nachweis verwendet.

Kindelemente von <i>EreignisType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>TimeStamp</b>	<i>xs:dateTime</i>	1		
Der Zeitpunkt, an dem das Ereignis protokolliert wird. Datum/Uhrzeit sollen auf eine 1/1000 Sekunde genau dokumentiert werden. Es ist sicherzustellen, dass für die beteiligten Systeme eine Zeitsynchronisierung durchgeführt wird.				
<b>wsa:MessageID (ref)</b>		1	B.2	121
Der eindeutige Identifikator (auch messageID) des Transportauftrags. Er ist über die gesamte Datenübermittlung die eindeutige Referenz des Transportauftrages.				
<b>xtaServerIdentitaet</b>	<i>xs:token</i>	1		
Die Identität des protokollierenden Prozesses als Software-Instanz, z.B. die Server-IP-Adresse oder die URI seines XTA-WS.				
<b>xtaRolle</b>	<i>xtacore:Code.XTARolle</i>	1	1.1.4.3.9	22
Die XTA-Rolle (Autor, Sender, Empfänger oder Leser) des Prozesses, die das Ereignis erstellt.				
<b>ereignis</b>	<i>xta:Code.EreignisArtType</i>	1	2.2.1.6	62
Ereignisart des Ereignisses, das durch den Protokolleintrag dokumentiert wird.				
<b>istFehler</b>	<i>xs:boolean</i>	1		
Gibt an, ob das Ereignis einen Fehlerfall dokumentiert.				
Erhält den Wert <code>true</code> , falls ein Fehlerfall dokumentiert wird. In diesem Fall müssen im Element <code>fehler</code> zusätzliche Informationen zur Fehlersituation eingetragen werden.				
Der Wert wird auf <code>false</code> gesetzt, wenn ein erfolgreicher Bearbeitungsschritt dokumentiert wird. In diesem Fall wird dem Element <code>fehler</code> kein Wert zugewiesen.				
Dieses Element/Attribut hat den Wert „false“, sofern kein anderer Wert übermittelt wird (default-Wert).				
<b>fehler</b>	<i>xs:string</i>	0..1		
Zusatzinformation zur Fehlersituation.				

#### 1.2.1.2 EreignisListType

Typ: *EreignisListType*

Liste der Ereignisse.

Kindelement von <i>EreignisListType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>Ereignis</b>	<i>xtaevt:EreignisType</i>	<b>0..n</b>	<a href="#">1.2.1.1</a>	<a href="#">24</a>
Hier werden Ereignisse eingetragen.				

### 1.2.1.3 KryptographieNachweis

Typ: *KryptographieNachweis*

Enthält alle Nachweise für Ereignisse mit Bezug zu kryptographischen Verfahren.

Dieser Typ ist eine Erweiterung des Basistyps *ZertifikatNachweis* (siehe [Abschnitt 1.2.1.9 auf Seite 27](#)).

Kindelemente von <i>KryptographieNachweis</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>algorithmenbezeichnung</b>	<i>xoenv-1c:String.Latin</i>	<b>1</b>	<a href="#">B.4</a>	<a href="#">121</a>
Bezeichnung des kryptographischen Verfahrens.				
<b>schluessellaenge</b>	<i>xs:integer</i>	<b>1</b>		
Schlüssellänge für das verwendete kryptographische Verfahren.				
<b>zertifikatNiveau</b>	<i>xtacore:Code.ZertifikatNiveau</i>	<b>1</b>	<a href="#">1.1.2.3.3.10</a>	<a href="#">13</a>
Niveau des elektronischen Zertifikates, welches für die Transportverschlüsselung verwendet wurde.				
<b>zertifikatMedium</b>	<i>xtacore:Code.ZertifikatMedium</i>	<b>1</b>	<a href="#">1.1.2.3.3.9</a>	<a href="#">13</a>
Art des Trägermediums des elektronischen Zertifikates, welches für die Transportverschlüsselung verwendet wurde.				

### 1.2.1.4 QuittungstypNachweis

Typ: *QuittungstypNachweis*

Enthält alle Nachweise für Ereignisse mit Bezug zu Quittungstypen.

Kindelement von <i>QuittungstypNachweis</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>quittungstyp</b>	<i>xtacore:Code.TechnischeQuittungen</i>	<b>1</b>	<a href="#">1.1.2.3.3.5</a>	<a href="#">12</a>
Art der technischen Quittung im Rahmen des Nachrichtentransports.				

### 1.2.1.5 ServiceUriNachweis

Typ: *ServiceUriNachweis*

Enthält alle Nachweise für Ereignisse mit Bezug zu Service URIs.

Kindelemente von <i>ServiceUriNachweis</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>ServiceURI</b>	<i>xs:anyURI</i>	1		
Die eindeutige Ressourcenkennung, unter der der Zustellungsdienst für synchrone Nachrichten beim Leser aufgerufen wurde.				
<b>kanal</b>	<i>xtacore:Code.Kanal</i>	1	1.1.3.3. 2.4	17
Verwendeter Kommunikationskanal zwischen Sender und Empfänger.				

### 1.2.1.6 TeilnehmeridentifikationNachweis

Typ: *TeilnehmeridentifikationNachweis*

Enthält alle Nachweise für Ereignisse mit Bezug zur Identifikation von Teilnehmern.

Kindelement von <i>TeilnehmeridentifikationNachweis</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>teilnehmer</b>	<i>oscimeta:PartyType</i>	1	2.2.1.44	75
Die fachliche Identität des Autors oder Lesers.				

### 1.2.1.7 TransportNachweis

Typ: *TransportNachweis*

Enthält alle Nachweise für Ereignisse mit Bezug zu den technischen Merkmalen des Transports.

Kindelemente von <i>TransportNachweis</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>typKommunikation</b>	<i>xtacore:Code.Kommunikationstyp</i>	1	1.1.1.3. 3.2	6
Typ der Kommunikation (entweder asynchron oder synchron).				
<b>abgabestation</b>	<i>xtacore:Code.Abgabestation</i>	1	1.1.1.3. 3.6	6
Eindeutige Ressourcenkennung des Systems, von dem die Nachricht entgegengenommen wurde.				
<b>formatTransportnachricht</b>	<i>xtacore:Code.TransportnachrichtenFormat</i>	1	1.1.3.3. 2.5	18
Format der Transportnachricht, auszuwählen aus der Liste der verfügbaren Nachrichtenformate für Transportnachrichten.				
<b>kanal</b>	<i>xtaevt:VerzeichnisdienstNachweis</i>	1	1.2.1.8	26
<b>transportprotokoll</b>	<i>xtacore:Code.Transportprotokoll</i>	1	1.1.3.3. 2.6	18
Verwendetes Transportprotokoll zwischen Sender und Empfänger.				

### 1.2.1.8 VerzeichnisdienstNachweis

Typ: *VerzeichnisdienstNachweis*

Enthält alle Nachweise für Ereignisse mit Bezug zu Verzeichnisdiensten.

Kindelemente von <i>VerzeichnisdienstNachweis</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>nameVerzeichnisdienst</b>	<i>xtacore:Code.VerzeichnisAdressierung</i>	1	<a href="#">1.1.3.3.2.2</a>	<a href="#">17</a>
Eindeutiger Name des Verzeichnisdienstes, über den die Parameter für die technische Adressierung von Teilnehmern bezogen werden.				
<b>adresseVerzeichnisdienst</b>	<i>xs:anyURI</i>	1		
Die eindeutige Ressourcenkennung, unter der der festgelegte Verzeichnisdienst ( <i>nameVerzeichnisdienst</i> ) zu erreichen ist.				

### 1.2.1.9 ZertifikatNachweis

Typ: *ZertifikatNachweis*

Enthält alle Nachweise für Ereignisse mit Bezug zu elektronischen Zertifikaten.

Kindelemente von <i>ZertifikatNachweis</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>zertifizierungsdiensteanbieter</b>	<i>xoev-lc:String.Latin</i>	1	<a href="#">B.4</a>	<a href="#">121</a>
Bezeichnung des Zertifizierungsdiensteanbieters.				
<b>zertifikatSeriennummer</b>	<i>xoev-lc:String.Latin</i>	1	<a href="#">B.4</a>	<a href="#">121</a>
Seriennummer des verwendeten Zertifikats.				
<b>zertifikatCommonName</b>	<i>xoev-lc:String.Latin</i>	1	<a href="#">B.4</a>	<a href="#">121</a>
Attribut „Common Name“ aus dem verwendeten Zertifikat.				
<b>zertifikatStatus</b>	<i>xta:Code.ZertifikatStatus</i>	1	<a href="#">2.2.1.3</a>	<a href="#">61</a>
Status des geprüften Zertifikats.				

## 1.2.2 Nachweise für den Service Report

Aufbauend auf den Grundstrukturen, werden in diesem Abschnitt die Nachweise beschrieben, die im ServiceReport verwendet werden. Nachweise müssen eingetragen werden, sobald ein Ereignis der Schlüsseltabelle Ereignisart (siehe [Abschnitt A.2.4 auf Seite 92](#)) eintritt. Der einzutragende Nachweis trägt denselben Namen wie das Ereignis.

### 1.2.2.1 AbbruchZustellversuche

Typ: *AbbruchZustellversuche*

Ist vom Sender zu protokollieren, wenn dieser die Zustellversuche abbricht, weil die maximale Anzahl an Zustellversuchen überschritten wurde.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>AbbruchZustellversuche</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>anzahlZustellversuche</b>	<i>xs:integer</i>	1		
Vorgabe für die maximale Anzahl von Versuchen, eine Nachricht erfolgreich zu übermitteln.				

### 1.2.2.2 AbfrageVerzeichnisdienst

Typ: *AbfrageVerzeichnisdienst*

Ist vom Sender zu protokollieren, wenn dieser einen Verzeichnisdienst abfragt.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>AbfrageVerzeichnisdienst</i>				
Kindelement	Typ	Anz.	Ref.	Seite
verzeichnisdienst	<i>xtaevt:VerzeichnisdienstNachweis</i>	1	1.2.1.8	26

### 1.2.2.3 Abholbestaetigung

Typ: *Abholbestaetigung*

Ist vom Empfänger zu protokollieren, wenn der Leser die Abholung einer Nachricht bestätigt.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>Abholbestaetigung</i>				
Kindelement	Typ	Anz.	Ref.	Seite
teilnehmer	<i>xtaevt:TeilnehmeridentifikationNachweis</i>	1	1.2.1.6	26

### 1.2.2.4 AbholenDurchLeser

Typ: *AbholenDurchLeser*

Ist vom Empfänger zu protokollieren, wenn ein Leser eine Nachricht abholt.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>AbholenDurchLeser</i>				
Kindelement	Typ	Anz.	Ref.	Seite
teilnehmer	<i>xtaevt:TeilnehmeridentifikationNachweis</i>	1	1.2.1.6	26

### 1.2.2.5 AufloesungTransportverschluesselung

Typ: *AufloesungTransportverschluesselung*

Ist vom Empfänger zu protokollieren, wenn dieser eine Transportverschlüsselung auflöst.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>AufloesungTransportverschluesselung</i>				
Kindelement	Typ	Anz.	Ref.	Seite
kryptographiemerkmale	<i>xtaevt:KryptographieNachweis</i>	1	1.2.1.3	25

### 1.2.2.6 AuswahlProfil

Typ: *AuswahlProfil*



Ist vom Sender oder Empfänger zu protokollieren, wenn dieser ein Profil auswählt, dass auf den Transport angewendet werden soll.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>AuswahlProfil</i>				
Kindelement	Typ	Anz.	Ref.	Seite
profil	<i>xs:anyURI</i>	1		
Die eindeutige URI des ausgewählten Service-Profil.				

### 1.2.2.7 AuswahlServicequalitaet

Typ: *AuswahlServicequalitaet*

Ist vom Sender oder Empfänger zu protokollieren, wenn dieser ein Profil auswählt, dass auf den Transport angewendet werden soll.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelemente von <i>AuswahlServicequalitaet</i>				
Kindelement	Typ	Anz.	Ref.	Seite
servicequalitaet	<i>oscimeta:NonEmptyStringType</i>	1	<a href="#">2.2.1.40</a>	<a href="#">75</a>
Name der Servicequalität, deren Wert bei der Durchführung vom Sender oder Empfänger ausgewählt wird.				
wert	<i>oscimeta:NonEmptyStringType</i>	1	<a href="#">2.2.1.40</a>	<a href="#">75</a>
Der Wert, der für die angegebene Servicequalität ausgewählt wurde.				

### 1.2.2.8 ErklaerungHinterlegt

Typ: *ErklaerungHinterlegt*

Ist vom Sender oder Empfänger zu protokollieren, um eine Verbindung zu dem Dokument herzustellen, mit dem die organisatorische Vorgabe umgesetzt wurde.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>ErklaerungHinterlegt</i>				
Kindelement	Typ	Anz.	Ref.	Seite
ort	<i>oscimeta:NonEmptyStringType</i>	1	<a href="#">2.2.1.40</a>	<a href="#">75</a>
Eindeutige, nachvollziehbare Bezeichnung des Ortes, an dem die Erklärung hinterlegt ist.				

### 1.2.2.9 NachrichtEmpfangen

Typ: *NachrichtEmpfangen*

Ist vom Empfänger zu protokollieren, wenn dieser eine Nachricht annimmt.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>NachrichtEmpfangen</i>				
Kindelement	Typ	Anz.	Ref.	Seite
transporteigenschaften	<i>xtaevt:TransportNachweis</i>	1	<a href="#">1.2.1.7</a>	<a href="#">26</a>

### 1.2.2.10 NachrichtengroesseGeprueft

Typ: *NachrichtengroesseGeprueft*

Ist vom Sender oder Empfänger zu protokollieren, wenn dieser die ermittelte Größe einer Nachricht gegen eine vorgegebene Größenbeschränkung für Nachrichten prüft.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelemente von <i>NachrichtengroesseGeprueft</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>maxNachrichtengroesse</b>	<i>xs:positiveInteger</i>	1		
Vorgegebene Größenbeschränkung für Nachrichten, gegen die geprüft wurde, in Megabyte (MB) als positive ganze Zahl.				
<b>istNachrichtengroesse</b>	<i>xs:positiveInteger</i>	1		
Ermittelte Größe der Nachricht in Megabyte (MB) als positive ganze Zahl.				

### 1.2.2.11 NachrichtLoeschen

Typ: *NachrichtLoeschen*

Ist vom Sender oder Empfänger zu protokollieren, wenn dieser eine Nachricht löscht. Bei wem dieses Ereignis ausgelöst wird, wird im Element XTA-Rolle dokumentiert. Dieses Ereignis ist nicht transportrelevant und führt daher nicht zu einem Eintrag im TransportReport.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>NachrichtLoeschen</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>loeschqualitaet</b>	<i>xtacore:Code.QualitaetLoeschen</i>	1	<a href="#">1.1.2.3.3.3</a>	<a href="#">12</a>
Niveaustufe, auf dem das Löschen der Daten vorgenommen wurde.				

### 1.2.2.12 NachrichtVersenden

Typ: *NachrichtVersenden*

Ist vom Sender zu protokollieren, wenn dieser eine Nachricht versendet.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelemente von <i>NachrichtVersenden</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>adresseEmpfaenger</b>	<i>xs:string</i>	1		
Eindeutige, technische Adressierung des Empfängers.				
<b>transporteigenschaften</b>	<i>xtaevt:TransportNachweis</i>	1	<a href="#">1.2.1.7</a>	<a href="#">26</a>

### 1.2.2.13 NotifikationGesendet

Typ: *NotifikationGesendet*

Ist vom Empfänger zu protokollieren, wenn dieser eine Notifikation an den Leser sendet.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>NotifikationGesendet</i>				
Kindelement	Typ	Anz.	Ref.	Seite
serviceUri	<i>xtaevt:ServiceUriNachweis</i>	1	1.2.1.5	25

### 1.2.2.14 OnlineZertifikatspruefung

Typ: *OnlineZertifikatspruefung*

Ist vom Empfänger zu protokollieren, wenn dieser eine Notifikation an den Leser sendet.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>OnlineZertifikatspruefung</i>				
Kindelement	Typ	Anz.	Ref.	Seite
zertifikatEigenschaften	<i>xtaevt:ZertifikatNachweis</i>	1	1.2.1.9	27

### 1.2.2.15 ProtokollAbgerufen

Typ: *ProtokollAbgerufen*

Ist vom Sender oder Empfänger zu protokollieren, wenn eine Nachricht den Schutzbedarf „hoch“ hat und sie Gegenstand eines Service Report ist. Bei wem dieses Ereignis ausgelöst wird, wird im Element XTA-Rolle dokumentiert.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>ProtokollAbgerufen</i>				
Kindelement	Typ	Anz.	Ref.	Seite
teilnehmer	<i>xtaevt:TeilnehmeridentifikationNachweis</i>	1	1.2.1.6	26

### 1.2.2.16 ProtokollierungGeloescht

Typ: *ProtokollierungGeloescht*

Ist vom Sender oder Empfänger zu protokollieren, wenn Nachweise nach Erreichen der Löschfrist automatisch vom XTA-Server gelöscht werden. Bei wem dieses Ereignis ausgelöst wird, wird im Element XTA-Rolle dokumentiert.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelemente von <i>ProtokollierungGeloescht</i>				
Kindelement	Typ	Anz.	Ref.	Seite
loeschfrist	<i>xs:positiveInteger</i>	1		
Die Anzahl Tage, nach deren Verstreichen das Protokoll spätestens gelöscht werden musste (Höchstspeicherzeit).				
startzeitpunkt	<i>xtacore:Code.StartzeitpunktLoeschfrist</i>	1	1.1.1.3. 3.4	6

### 1.2.2.17 ProtokollqualitaetNachweis

Typ: *ProtokollqualitaetNachweis*

Ist vom Sender oder Empfänger zu protokollieren, wenn Protokolle / Reports die Bearbeitungsschritte und die Ergebnisse im Rahmen der Abarbeitung eines Transportauftrags gemäß der Vorgaben zur Transparenz im Service-Profil dokumentieren.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>ProtokollqualitaetNachweis</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>protokoll</b>		1		
<b>rolle</b>	<i>xtacore:Code.XTARolle</i>	1	<a href="#">1.1.4.3.9</a>	<a href="#">22</a>
Hier ist einzutragen, in Bezug auf welche Rolle in der XTA-Infrastruktur hier Festlegungen zur Protokollierung eingetragen werden soll.				
<b>qualitaet</b>	<i>xtasp:Code.QualitaetProtokollierung</i>	1	<a href="#">1.1.2.3.3.4</a>	<a href="#">12</a>
Hier ist einzutragen, ob durch den betreffenden Knoten ein Protokoll / ein Report zu führen ist und mit welchem Absicherungs niveau dies geschehen soll.				
<b>abweichverhalten</b>	<i>xtacore:Code.Aabweichverhalten</i>	1	<a href="#">1.1.1.3.3.7</a>	<a href="#">7</a>
Hier wird festgelegt was geschehen soll, wenn von der zugehörigen Vorgabe abgewichen wird. Das Abweichverhalten muss aus der vorgegebenen Codeliste gewählt werden.				

### 1.2.2.18 PruefungTransportsignatur

Typ: *PruefungTransportsignatur*

Ist vom Sender zu protokollieren, wenn dieser eine Transportsignatur prüft.

Kindelement von <i>PruefungTransportsignatur</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>kryptographiemerkmale</b>	<i>xtaevt:KryptographieNachweis</i>	1	<a href="#">1.2.1.3</a>	<a href="#">25</a>

### 1.2.2.19 Quittungsempfang

Typ: *Quittungsempfang*

Ist vom Sender zu protokollieren, wenn dieser eine Quittung empfängt.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>Quittungsempfang</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>quittungstyp</b>	<i>xtaevt:QuittungstypNachweis</i>	1	<a href="#">1.2.1.4</a>	<a href="#">25</a>

### 1.2.2.20 Quittungsversand

Typ: *Quittungsversand*

Ist vom Empfänger zu protokollieren, wenn dieser eine Quittung versendet.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>Quittungsversand</i>				
Kindelement	Typ	Anz.	Ref.	Seite
quittungstyp	<i>xtaevt:QuittungstypNachweis</i>	1	1.2.1.4	25

### 1.2.2.21 ServiceQualitaetVerletzt

Typ: *ServiceQualitaetVerletzt*

Ist vom Sender oder Empfänger zu protokollieren, wenn eine Servicequalität nicht erbracht werden kann. Bei wem dieses Ereignis ausgelöst wird, wird im Element XTA-Rolle dokumentiert.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelemente von <i>ServiceQualitaetVerletzt</i>				
Kindelement	Typ	Anz.	Ref.	Seite
servicequalitaet	<i>oscimeta:NonEmptyStringType</i>	1	2.2.1.40	75
zielwert	<i>oscimeta:NonEmptyStringType</i>	1	2.2.1.40	75
aktuellerWert	<i>oscimeta:NonEmptyStringType</i>	1	2.2.1.40	75
abweichverhalten	<i>xtacore:Code.Aabweichverhalten</i>	1	1.1.1.3. 3.7	7

### 1.2.2.22 TransportauftragsIDErstellen

Typ: *TransportauftragsIDErstellen*

Ist vom Sender zu protokollieren, wenn dieser eine TransportauftragsID erstellt.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>TransportauftragsIDErstellen</i>				
Kindelement	Typ	Anz.	Ref.	Seite
teilnehmer	<i>xtaevt:TeilnehmeridentifikationNachweis</i>	1	1.2.1.6	26

### 1.2.2.23 TransportsignaturAnbringen

Typ: *TransportsignaturAnbringen*

Ist vom Sender zu protokollieren, wenn dieser eine Transportsignatur anbringt.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>TransportsignaturAnbringen</i>				
Kindelement	Typ	Anz.	Ref.	Seite
kryptographiemerkmale	<i>xtaevt:KryptographieNachweis</i>	1	1.2.1.3	25

### 1.2.2.24 TransportverschlüsselungAnbringen

Typ: *TransportverschlüsselungAnbringen*

Ist vom Sender zu protokollieren, wenn dieser eine Transportverschlüsselung anbringt.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>TransportverschlüsselungAnbringen</i>				
Kindelement	Typ	Anz.	Ref.	Seite
kryptographiemerkmale	<i>xtaevt:KryptographieNachweis</i>	1	1.2.1.3	25

### 1.2.2.25 VerifikationAutor

Typ: *VerifikationAutor*

Ist vom Empfänger zu protokollieren, wenn dieser den Autor einer Nachricht verifiziert.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelemente von <i>VerifikationAutor</i>				
Kindelement	Typ	Anz.	Ref.	Seite
teilnehmer	<i>xtaevt:TeilnehmeridentifikationNachweis</i>	1	1.2.1.6	26
Eindeutiges Identifikationsmerkmal einer XTA-Rolle.				
verzeichnisdienst	<i>xtaevt:VerzeichnisdienstNachweis</i>	1	1.2.1.8	26
Eindeutiger Name des Verzeichnisdienstes, über den die Parameter für die technische Adressierung von Teilnehmern bezogen werden.				

### 1.2.2.26 VersandauftragErhalten

Typ: *VersandauftragErhalten*

Ist vom Sender zu protokollieren, wenn dieser einen Versandauftrag erhalten hat.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelemente von <i>VersandauftragErhalten</i>				
Kindelement	Typ	Anz.	Ref.	Seite
typKommunikation	<i>xtacore:Code.Kommunikationstyp</i>	1	1.1.1.3. 3.2	6
Typ der Kommunikation (entweder asynchron oder synchron).				
schutzbedarf	<i>xtacore:Code.QualitaetKryptographie</i>	1	1.1.4.3.8	22
Geforderte Niveaustufe der kryptographischen Absicherung.				
teilnehmer	<i>xtaevt:TeilnehmeridentifikationNachweis</i>	1	1.2.1.6	26

### 1.2.2.27 VersandauftragZurueckziehen

Typ: *VersandauftragZurueckziehen*

Ist vom Sender zu protokollieren, wenn der Autor einen erteilten Versandauftrag zurückzieht.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>VersandauftragZurueckziehen</i>				
Kindelement	Typ	Anz.	Ref.	Seite
teilnehmer	<i>xtaevt:TeilnehmeridentifikationNachweis</i>	1	1.2.1.6	26

### 1.2.2.28 WeiterleitungAnLeser

Typ: *WeiterleitungAnLeser*

Ist vom Empfänger zu protokollieren, wenn dieser eine synchrone Nachricht an den Leser weiterleitet.

Kindelement von <i>WeiterleitungAnLeser</i>				
Kindelement	Typ	Anz.	Ref.	Seite
serviceUri	<i>xtaevt:ServiceUriNachweis</i>	1	1.2.1.5	25

### 1.2.2.29 Zustellfrist

Typ: *Zustellfrist*

Ist vom Sender zu protokollieren, wenn der Autor dem Sender eine Zustellfrist vorgibt.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>Zustellfrist</i>				
Kindelement	Typ	Anz.	Ref.	Seite
zustellfrist	<i>xtacore:Code.Zustellfrist</i>	1	1.1.1.3. 3.5	6
Datum, bis zu dem die Nachricht erfolgreich zugestellt worden sein muss.				

### 1.2.2.30 Zustellungstermin

Typ: *Zustellungstermin*

Ist vom Sender zu protokollieren, wenn der Autor dem Sender einen Zustellungstermin vorgibt.

Dieser Typ ist eine Erweiterung des Basistyps *EreignisType* (siehe [Abschnitt 1.2.1.1 auf Seite 24](#)).

Kindelement von <i>Zustellungstermin</i>				
Kindelement	Typ	Anz.	Ref.	Seite
zustellungstermin	<i>xs:dateTime</i>	1		
Datum, an dem die Nachricht zugestellt werden soll.				





## 2 XTA Webservice (2.2)



Die Kommunikation zu den WebServices wird über SSL mit Client-Authentication abgesichert.

### 2.1 Methoden

#### 2.1.1 Die Schnittstellentypen der Transportebene

##### 2.1.1.1 Schnittstellentyp transportPortType

Der transportType gruppiert Methoden die nicht unmittelbar zur Steuerung von Transportaufträgen verwendet werden.

Es handelt sich um unterstützende Funktionen rund um den Transport von Nachrichten.

##### 2.1.1.1.1 Methode checkAccountActive

Die Methode checkAccountActive prüft, ob der Webservice verfügbar ist und ob der angegebene Account eingerichtet ist.

##### 2.1.1.1.1.1 Exceptions

- ParameterNotSupportedException wird geworfen, wenn eine Methode mit spezifikationsgemäßen, aber von der Implementation nicht unterstützen Parametern aufgerufen wird.
- PermissionDeniedException signalisiert einen Autorisierungsfehler.
- XTAWSTechnicalProblemException signalisiert einen allgemeinen Fehler während der Verarbeitung.

##### 2.1.1.1.1.2 Input

**Tabelle 2.1. Header**

WSDL Message	Message Part	Part Element
XTAHeader	XTAIdentifier	oscimeta:Identifier (vgl. <a href="#">Seite 72</a> )
Fachliche Identität des Autors oder Lesers.		

**Tabelle 2.2. Body**

WSDL Message	Message Part	Part Element
EmptyBody		

##### 2.1.1.1.1.3 Output

Keine Header-Informationen.

**Tabelle 2.3. Body**

WSDL Message	Message Part	Part Element
EmptyBody		

WSDL Message	Message Part	Part Element
Liefert die Methode ohne Fehler eine leere Nachricht ist der XTA-WS erreichbar und der angegebene Account eingerichtet und aktiviert.		

### 2.1.1.1.2 Methode getServiceReport

Die Methode liefert dem Autor den ServiceReport über die Einhaltung der Servicequalitäten für den Versand der Nachricht. Wird die Funktion vom Leser aufgerufen enthält der ServiceReport einen Bericht über die Einhaltung der Servicequalitäten beim Empfang der Nachricht.

#### 2.1.1.1.2.1 Exceptions

- ParameterNotSupportedException wird geworfen, wenn eine Methode mit spezifikationsgemäßen, aber von der Implementation nicht unterstützen Parametern aufgerufen wird.
- PermissionDeniedException signalisiert einen Autorisierungsfehler.
- TooManyResultsException signalisiert, dass der ServiceReport aufgrund der Selektorbelegung zu viele Einträge hat.
- XTAWSTechnicalProblemException signalisiert einen allgemeinen Fehler während der Verarbeitung.

#### 2.1.1.1.2.2 Input

**Tabelle 2.4. Header**

WSDL Message	Message Part	Part Element
XTAHeader	XTAIdentifizier	oscimeta:Identifizier (vgl. <a href="#">Seite 72</a> )
Fachliche Identität des Autors oder Lesers.		

**Tabelle 2.5. Body**

WSDL Message	Message Part	Part Element
ReportSelector	ReportSelector	xta:ReportSelector (vgl. <a href="#">Seite 82</a> )
Parameter für die Auswahl/Filterung von Einträgen in einem Service Report.		

#### 2.1.1.1.2.3 Output

Keine Header-Informationen.

**Tabelle 2.6. Body**

WSDL Message	Message Part	Part Element
ServiceReport	ServiceReport	xta:ServiceReport (vgl. <a href="#">Seite 81</a> )
ServiceReport zu den durch den Selector spezifizierten Transportaufträgen.		

### 2.1.1.1.3 Methode getTransportReport

Die Bearbeitung von Transportaufträgen wird protokolliert. Dieses Protokoll enthält relevante Information zu einem konkreten Transportauftrag und kann als Nachweis für eien Zustellung verwendet werden. Das Protokoll wird während der Bearbeitung des Transportauftrages sukzessive erstellt und dokumentiert den jeweils aktuellen Bearbeitungsstand des Auftrags.

### 2.1.1.1.3.1 Exceptions

- `InvalidMessageIDException` signalisiert, dass zu der angegebenen Nachrichtenennung keine Protokollinformationen vorliegen.
- `ParameterNotSupportedException` wird geworfen, wenn eine Methode mit spezifikationsgemäßen, aber von der Implementation nicht unterstützen Parametern aufgerufen wird.
- `PermissionDeniedException` signalisiert einen Autorisierungsfehler.
- `XTAWSTechnicalProblemException` signalisiert einen allgemeinen Fehler während der Verarbeitung.

### 2.1.1.1.3.2 Input

**Tabelle 2.7. Header**

WSDL Message	Message Part	Part Element
XTAHeader	XTAIdentifier	oscimeta:Identifier (vgl. <a href="#">Seite 72</a> )
Fachliche Identität des Autors oder Lesers.		

**Tabelle 2.8. Body**

WSDL Message	Message Part	Part Element
MessageID	MessageID	wsa:MessageID (vgl. <a href="#">Seite 121</a> )
NachrichtenID der Nachricht, zu der ein Protokoll abgerufen werden soll.		

### 2.1.1.1.3.3 Output

Keine Header-Informationen.

**Tabelle 2.9. Body**

WSDL Message	Message Part	Part Element
GetTransportReportResponse	GetTransportReportResponse	xta:TransportReport (vgl. <a href="#">Seite 80</a> )
Es wird das Transportprotokoll in einem TransportReport Objekt zurückgegeben.		

### 2.1.1.1.4 Methode `lookupService`

Zur Formulierung eines Transportauftrags wird die Adressinformation des Lesers benötigt. Die Ermittlung der technischen Adresse und wenn notwendig, weiterer für den Versand notwendige Informationen ist Aufgabe dieser Methode.

Zusätzlich wird mit der Methode abgefragt, ob der Leser grundsätzlich anbietet, die Nachricht eines bestimmten Nachrichtentyps anzunehmen. Die Methode greift dabei auf einen Verzeichnisdienst zurück, der diese Informationen enthält.

In der Ausprägung DVDV wird die Adresse des zuständigen Intermediärs inklusive benötigter öffentlicher Schlüssel zusammen mit der Auskunft, ob der Leser einen bestimmten Dienst anbietet, zurückgegeben.

#### 2.1.1.1.4.1 Exceptions

- `ExternalServiceUnavailableException` signalisiert einen Fehler aufgrund von Nichterreichbarkeit des Dienstes.
- `ParameterIsNotValidException` signalisiert fehlende oder fehlerhafte Eingabeparameter.

- `ParameterNotSupportedException` wird geworfen, wenn eine Methode mit spezifikationsgemäßen, aber von der Implementation nicht unterstützten Parametern aufgerufen wird.
- `PermissionDeniedException` signalisiert einen Autorisierungsfehler.
- `XTAWSTechnicalProblemException` signalisiert einen allgemeinen Fehler während der Verarbeitung.

#### 2.1.1.1.4.2 Input

**Tabelle 2.10. Header**

WSDL Message	Message Part	Part Element
XTAHeader	XTAIdentifier	oscimeta:Identifier (vgl. <a href="#">Seite 72</a> )
Fachliche Identität des Autors oder Lesers.		

**Tabelle 2.11. Body**

WSDL Message	Message Part	Part Element
LookupServiceRequest	LookupServiceRequest	xta:LookupServiceRequest (vgl. <a href="#">Seite 79</a> )
Liste von Dienstanfragen.		

#### 2.1.1.1.4.3 Output

Keine Header-Informationen.

**Tabelle 2.12. Body**

WSDL Message	Message Part	Part Element
LookupServiceResponse	LookupServiceResponse	xta:LookupServiceResponse (vgl. <a href="#">Seite 79</a> )
Die Methode liefert Informationen zur Verfügbarkeit des angefragten Dienstes und für die Kommunikation notwendige Zusatzinformationen zurück.		

### 2.1.1.1.5 Wichtige Objekte der transportPort-Schnittstelle

#### 2.1.1.1.5.1 PartyIdentifierType und PartyType

Während im Transportauftrag eine konkrete technische Adresse verwendet wird, arbeitet `lookupService` auf Basis einer auf dem Rollenmodell basierenden Formulierung des Adressaten. Die auf dem XTA Rollenmodell basierende Adressformulierung wird mittels des Typ `PartyIdentifierType` abgebildet.

#### 2.1.1.1.5.2 MessageID

Eine `MessageID` (modelliert im Objekt `MsgIdentificationType`) identifiziert eindeutig einen Transportauftrag.

Die `MessageID` hat das Format einer URN. Sie sollte folgende Anforderungen erfüllen:

- Die `MessageID` muss bei Sender und Empfänger eindeutig sein.
- Die `MessageID` soll erkennen lassen, wer die Nachricht erstellt hat. So kann in Problemfällen der Ersteller und Versender der Nachricht leichter ermittelt werden.

Zur Erfüllung der Anforderungen besteht die `MessageID` aus einer UUID der als Präfix die Kennung des versendenden Systems vorgestellt wird.

Somit ergibt sich der folgende Aufbau der MessageID:

- Präfix: Angabe über die Softwareinstanz, die die MessageID erstellt, z.B. ClearingstelleXY\_Xta\_01 oder ClearingstelleXY\_SAP\_15.
- Identifikator: Dieser muss aus einer UUID generiert sein (siehe RFC4122, z.B. 000ca2fe-f4e1-45c2-8233-3a0eb760bd16)

Die MessageID soll die Form einer URN haben. Als Namespace-ID wird 'de:xta:messageid' verwendet. Wenn wir diese Aspekte alle zusammenfassen, ergibt sich für die MessageID als URN der folgende allgemeine Aufbau: "urn:de:xta:messageid:<Präfix>:<Identifikator>"

Eine                    Beispielinstanz                    wäre:                    "urn:de:xta:messageid:clearingstelleXY\_xt-a\_01:000ca2fe-f4e1-45c2-8233-3a0eb760bd16"

### 2.1.1.2 Schnittstellentyp senderPortType

In diesem Schnittstellentyp sind alle Methoden zum Versand einer Nachricht zusammengefasst.

#### 2.1.1.2.1 Methode cancelMessage

Diese Methode storniert einen bereits erteilten Transportauftrag, der nicht unmittelbar sondern zu einem späteren Zeitpunkt ausgeführt werden sollte.

Das Zurückziehen des Transportauftrags ist nur möglich, wenn folgende Bedingungen erfüllt sind:

- Die zugehörige Fachnachricht gehört dem Autor.
- Die Fachnachricht wurde zuvor vom Autor zum asynchronen Versand unter Angabe des Schalters **NotBefore** übergeben.
- Der Transportauftrag ist noch nicht ausgeführt worden.

Der **TransportReport** wird zum Nachweis des Rückrufs der Nachricht vorgehalten, der Payload wird gelöscht

#### 2.1.1.2.1.1 Exceptions

- CancelDeniedException signalisiert, dass der referenzierte Transportauftrag nicht zurückgezogen werden konnte.
- InvalidMessageIDException signalisiert, dass zu der angegebenen Nachrichtenennung keine Protokollinformationen vorliegen.
- NotImplementedException informiert darüber, dass diese Methode nicht implementiert wurde.
- ParameterNotSupportedException wird geworfen, wenn eine Methode mit spezifikationsgemäßen, aber von der Implementation nicht unterstützen Parametern aufgerufen wird.
- PermissionDeniedException signalisiert einen Autorisierungsfehler.
- XTAWSTechnicalProblemException signalisiert einen allgemeinen Fehler während der Verarbeitung.

#### 2.1.1.2.1.2 Input

**Tabelle 2.13. Header**

WSDL Message	Message Part	Part Element
XTAHeader	XTAIdentifier	oscmeta:Identifier (vgl. <a href="#">Seite 72</a> )
Fachliche Identität des Autors oder Lesers.		

**Tabelle 2.14. Body**

WSDL Message	Message Part	Part Element
MessageID	MessageID	wsa:MessageID (vgl. <a href="#">Seite 121</a> )
NachrichtenID der Nachricht, deren Übermittlung abgebrochen werden soll.		

**2.1.1.2.1.3 Output**

Keine Header-Informationen.

**Tabelle 2.15. Body**

WSDL Message	Message Part	Part Element
EmptyBody		
Der Transportauftrag wurde zurückgezogen. Da der Transport nicht durchgeführt werden konnte, geht der Ampelstatus im TransportReport auf rot.		

**2.1.1.2.2 Methode createMessageID**

Die Methode createMessageID liefert eine eindeutige MessageID. Mittels dieser MessageID wird ein Transportauftrag eindeutig identifiziert.

**2.1.1.2.2.1 Exceptions**

- NotImplementedException informiert darüber, dass diese Methode nicht implementiert wurde.
- ParameterNotSupportedException wird geworfen, wenn eine Methode mit spezifikationsgemäßen, aber von der Implementation nicht unterstützten Parametern aufgerufen wird.
- PermissionDeniedException signalisiert einen Autorisierungsfehler.
- XTAWSTechnicalProblemException signalisiert einen allgemeinen Fehler während der Verarbeitung.

**2.1.1.2.2.2 Input****Tabelle 2.16. Header**

WSDL Message	Message Part	Part Element
XTAHeader	XTAIdentifier	oscimeta:Identifier (vgl. <a href="#">Seite 72</a> )
Fachliche Identität des Autors oder Lesers.		

**Tabelle 2.17. Body**

WSDL Message	Message Part	Part Element
EmptyBody		

**2.1.1.2.2.3 Output**

Keine Header-Informationen.

**Tabelle 2.18. Body**

WSDL Message	Message Part	Part Element
MessageID	MessageID	wsa:MessageID (vgl. <a href="#">Seite 121</a> )

WSDL Message	Message Part	Part Element
Eine eindeutige MessageID.		

### 2.1.1.2.3 Methode sendMessage

Mit dieser Methode wird ein Transportauftrag für den asynchronen Versand einer Nachricht erteilt.

#### 2.1.1.2.3.1 Exceptions

- MessageSchemaViolationException zeigt an, dass die übergebene Fachnachricht nicht schemakonform ist.
- MessageVirusDetectionException signalisiert, dass ein Virensch scanner in der Nachricht Schadcode entdeckt hat.
- NotImplementedException informiert darüber, dass diese Methode nicht implementiert wurde.
- ParameterIsNotValidException signalisiert fehlende oder fehlerhafte Eingabeparameter.
- ParameterNotSupportedException wird geworfen, wenn eine Methode mit spezifikationsgemäßen, aber von der Implementation nicht unterstützten Parametern aufgerufen wird.
- PermissionDeniedException signalisiert einen Autorisierungsfehler.
- SyncAsyncException zeigt an, dass eine Fachnachricht übergeben wurde, die nur für den synchronen Versand gültig ist.
- XTAWSTechnicalProblemException signalisiert einen allgemeinen Fehler während der Verarbeitung.

#### 2.1.1.2.3.2 Input

**Tabelle 2.19. Header**

WSDL Message	Message Part	Part Element
XTAHeader	MessageMetaData	oscmeta:MessageMetaData (vgl. <a href="#">Seite 73</a> )
In dieser Struktur werden die Steuerungsdaten des Transportauftrags definiert.		

**Tabelle 2.20. Body**

WSDL Message	Message Part	Part Element
MessageBody	Message	xta:Message (vgl. <a href="#">Seite 79</a> )
Dieses Objekt beinhaltet die zu übertragende Fachnachricht und eine beliebige Anzahl von Anhängen (Attachments). Die Fachnachricht kann in einem verschlüsselten Container hinterlegt werden. Zu der Fachnachricht kann ein Betreff (Subject) angegeben werden.		

#### 2.1.1.2.3.3 Output

Keine Header-Informationen.

**Tabelle 2.21. Body**

WSDL Message	Message Part	Part Element
EmptyBody		
Der Transportauftrag ist an das System übergeben.		

### 2.1.1.2.4 Methode sendMessageSync

Mit dieser Methode wird ein Transportauftrag für den synchronen Versand einer Nachricht erteilt. Die Methode übermittelt die Nachricht und liefert die eingegangene Antwort. Die Methode "blockiert" den Aufrufer solange, bis die Antwort vom Adressaten zurückgekommen ist.

Die Methode realisiert zwei unterschiedliche Kommunikationsszenarien. In Abhängigkeit der Parameter liefert sendMessageSync eine fachliche Rückantwort oder nicht.

#### 2.1.1.2.4.1 Exceptions

- ExternalServiceUnavailableException signalisiert einen Fehler aufgrund von Nichterreichbarkeit des Dienstes.
- MessageSchemaViolationException zeigt an, dass die übergebene Fachnachricht nicht schemakonform ist.
- MessageVirusDetectionException signalisiert, dass ein Virens Scanner in der Nachricht Schadcode entdeckt hat.
- NotImplementedException informiert darüber, dass diese Methode nicht implementiert wurde.
- ParameterIsNotValidException signalisiert fehlende oder fehlerhafte Eingabeparameter.
- ParameterNotSupportedException wird geworfen, wenn eine Methode mit spezifikationsgemäßen, aber von der Implementation nicht unterstützten Parametern aufgerufen wird.
- PermissionDeniedException signalisiert einen Autorisierungsfehler.
- SyncAsyncException zeigt an, dass eine Fachnachricht übergeben wurde, die nur für den synchronen Versand gültig ist.
- XTAWSTechnicalProblemException signalisiert einen allgemeinen Fehler während der Verarbeitung.

#### 2.1.1.2.4.2 Input

**Tabelle 2.22. Header**

WSDL Message	Message Part	Part Element
XTAHeader	MessageMetaData	oscimeta:MessageMetaData (vgl. <a href="#">Seite 73</a> )
In dieser Struktur werden die Steuerungsdaten des Transportauftrags definiert.		

**Tabelle 2.23. Body**

WSDL Message	Message Part	Part Element
MessageBody	Message	xta:Message (vgl. <a href="#">Seite 79</a> )
Dieses Objekt beinhaltet die zu übertragende Fachnachricht und eine beliebige Anzahl von Anhängen (Attachments). Die Fachnachricht kann in einem verschlüsselten Container hinterlegt werden. Zu der Fachnachricht kann ein Betreff (Subject) angegeben werden.		

#### 2.1.1.2.4.3 Output

**Tabelle 2.24. Header**

WSDL Message	Message Part	Part Element
OptHeaders	MessageMetaDataResponse	oscimeta:MessageMetaData (vgl. <a href="#">Seite 73</a> )
Das Vorhandensein des optionalen Headerelements MessageMetaDataResponse signalisiert, ob sich im Body-Element xta:MessageResponse ein Element Message enthalten ist, in dem eine Antwortnachricht enthalten ist.		



Tabelle 2.25. Body

WSDL Message	Message Part	Part Element
MessageResponseBody	MessageResponseContentContainer	xta:MessageResponse (vgl. Seite 81)
<p>Falls im Fachszenario eine synchrone Fachantwort gefordert wurde, befindet sich diese im Kindelement Message. Sie besteht aus einer Fachnachricht und ggf. einer Anzahl von Anhängen (Attachments). Zu der Fachnachricht kann ein Betreff (Subject) angegeben werden. Im Header muss ein Objekt MessageMetaDataResponse enthalten sein.</p> <p>Falls im Fachszenario keine synchrone Fachantwort gefordert ist, bleibt die Struktur MessageResponse leer, d. h. sie enthält kein Kindelement Message. In diesem Fall darf kein Objekt MessageMetaDataResponse im Header enthalten sein.</p>		

### 2.1.1.2.5 Wichtige Objekte der senderPort-Schnittstelle

#### 2.1.1.2.5.1 Der Transportauftrag: Header-Element *MessageMetaData*

Über die Struktur MessageMetaData (MMD) werden Informationen sowie die Vorgaben für den Transport von Nachrichten ausgetauscht. Der MMD enthält alle erforderlichen Informationen für den Transport der Nachricht, so dass der Versand ohne Zugriff auf die Fachnachricht erfolgen kann. Zusätzlich enthält MessageMetaData weitere Elemente, mit denen bestimmte Aspekte des Transports dokumentiert werden können. Diese Protokollinformationen werden für die Erstellung des TransportReport benötigt.

Der Autor erstellt einen MMD (Autor-MMD) und übergibt diesen zusammen mit der Fachnachricht an den Sender. Der Sender ergänzt diesen Autor-MMD zur Dokumentation des Transports, speichert ihn und verwendet ihn in der Erstellung des senderseitigen TransportReports. XTA enthält keine Vorgaben zur Übermittlung des Autor-MMD an den Empfänger oder Leser. Daher kann nicht davon ausgegangen werden, dass der MMD vom Autor zum Leser übertragen wird.

Empfänger konstruieren nach Erhalt einer Fachnachricht einen zugehörigen MMD für den Leser (Leser-MMD) und dokumentieren darin den Erhalt der Nachricht. Der Empfänger speichert diesen Leser-MMD und verwendet ihn, um den empfängerseitigen TransportReport zu erstellen. XTA enthält keine Vorgaben, dass Empfänger im MMD weitere Informationen eintragen müssen, z.B. aus der erhaltenen Fachnachrichten oder aus einem übermittelten Autor-Sender-MMD.

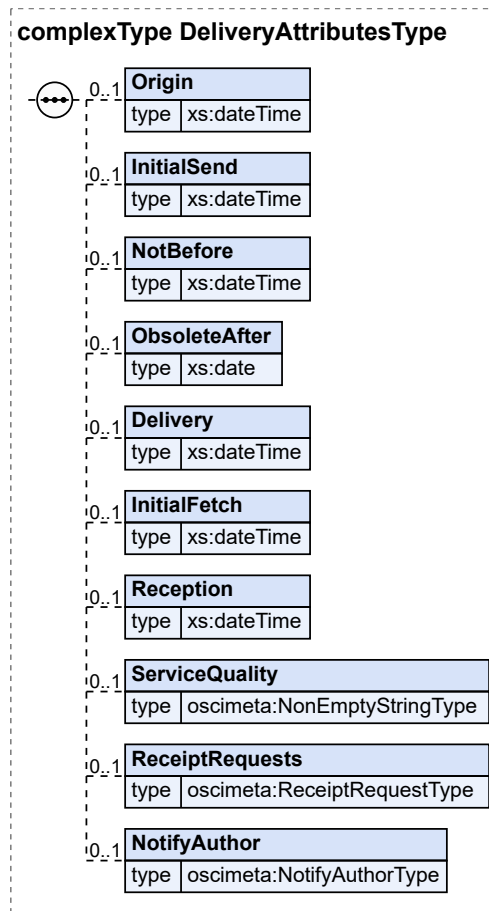
Leser erhalten den vom Empfänger erstellten MMD (Leser-MMD) beim Abruf einer asynchron übermittelten Nachricht oder als Teil einer synchronen Kommunikation. Sofern in einer synchronen Kommunikation vom Leser eine Fachantwort gefordert wird, verwendet der Leser den MMD befüllt ihn als Autor der Rückantwort mit entsprechenden Vorgaben und Informationen zum Transport der Rückantwort.

Wird ein MMD erstellt und befüllt, ist dabei grundsätzlich zu beachten, dass die Informationen ausreichend sind, um die Nachricht zu transportieren, und Informationen aus Fachnachrichten korrekt übernommen werden

Die Steuerung des Transports erfolgt im MMD über das Attribut TestMsg sowie die in der Struktur DeliveryAttributes enthaltenen Angaben. Zusätzlich ist die Angabe der Größe der Nachricht im Element MsgSize für den Transport zwingend erforderlich:

1. /oscmeta:MessageMetaData/@TestMsg: Über dieses Attribut kann eine Nachricht als Testnachricht ausgewiesen werden. Das Attribut ist vom Typ Boolean und enthält voreingestellt den Wert „false“. Um eine Testnachricht zu senden, muss diesem Attribut explizit der Wert „true“ zugewiesen werden.
2. /oscmeta:MessageMetaData/MsgSize: Das Element *MsgSize* enthält die Größe der Nachricht. Ist der Eintrag nicht vorhanden, muss das sendende System einen Wert eintragen.
3. /oscmeta:MessageMetaData/DeliveryAttribute: Die Struktur enthält Informationen und Vorgaben für die Übermittlung einer Nachricht.

Abbildung 2.1. MessageMetaData/DeliveryAttributes



Die nachfolgenden Tabellen beschreiben für die Rollen Autor, Sender und Empfänger diejenigen Elemente in DeliveryAttribute, die von diesen Rollen verwendet werden. Leser verwenden den MDD als Autor einer Rückantwort in der synchronen Kommunikation und verwenden hierfür die Tabelle für Autoren.

Tabelle 2.26. Vom Autor zu füllende Attribute der MessageMetaData-Struktur DeliveryAttributes

Element in DeliveryAttributes	Information / Vorgabe zur Übermittlung einer Fachnachricht
Origin	Optionaler Zeitstempel für die Erzeugung der Nachricht durch den Autor und deren Übermittlung an den Sender.
NotBefore	Optionale Festlegung für asynchrone Nachrichten. Definiert den Zeitpunkt des frühesten zulässigen Transports der Nachricht vom Sender an den Empfänger. Falls dieser Zeitpunkt bei der Bearbeitung durch den Sender bereits in der Vergangenheit liegt, erfolgt der Versand sofort.
ObsoleteAfter	Optionale Definition eines Zeitpunktes für asynchrone Nachrichten, ab dem der Transportauftrag nicht mehr durchgeführt werden darf. Nach dem Erreichen des Zeitpunkts darf der Sender keinen Versand der Nachricht durchführen. Die Durchführung des Transports ist gescheitert und entsprechend zu protokollieren.

Element in DeliveryAttributes	Information / Vorgabe zur Übermittlung einer Fachnachricht
ServiceQuality	Optionale Referenz auf das anzuwendende Service-Profil. Falls von fachlicher Seite ein Service-Profil definiert wurde, muss der Inhalt des Elementes uriDerVersion des Service-Profiles hier eingetragen werden.
ReceiptRequests	Optionaler Container für vom Autor für den Transportauftrag geforderten Quittungen. Durch das Anlegen des leeren Containers zeigt der Autor an, dass Quittungen gefordert werden. Der Autor legt fest, welche Quittungen gefordert werden, in dem er innerhalb des Containers die entsprechenden, ebenfalls leeren Elemente erzeugt.
NotifyAuthor	Optionale Struktur mit der der Autor angegeben kann, ob (Element NotifyMessageSent) und über welchen Dienst (Element NotifyAuthorURI) der Autor über die Zustellung einer Nachricht informiert werden will.

**Tabelle 2.27. Vom Sender zu füllende Attribute der MessageMetaData-Struktur DeliveryAttributes**

Element in DeliveryAttributes	Information / Vorgabe zur Übermittlung einer Fachnachricht
InitialSend	Optionaler Zeitstempel für den Zeitpunkt, an dem der Sender die Nachricht an den Empfänger verschickt hat.

**Tabelle 2.28. Vom Empfänger zu füllende Attribute der MessageMetaData-Struktur DeliveryAttributes**

Element in DeliveryAttributes	Information / Vorgabe zur Übermittlung einer Fachnachricht
Delivery	Optionaler Zeitstempel für den Zeitpunkt, an dem die vom Sender versendete Nachricht beim Empfänger eingegangen ist.
InitialFetch	Optionaler Zeitstempel für den Zeitpunkt, an dem der Leser die Nachricht erstmalig vom Empfänger abrufen.
Reception	Optionaler Zeitstempel für den Zeitpunkt, an dem der Leser das erfolgreiche Abrufen der Nachricht vom Empfänger bestätigt.

#### 2.1.1.2.5.2 Qualitätsanforderungen an den Transport (Delivery Attributes - Service Quality)

Das Element *ServiceQuality* ist vorgesehen, um die beim Transport der Nachricht zu berücksichtigenden Service Qualitäten anzugeben.

In XTA 2 wird das Thema Service Qualitäten durch die Spezifikation der XTA Service–Profile (siehe XTA-Spezifikationsdokument Rahmenbedingungen) behandelt. Die Verwendung des Elements *ServiceQuality* im XTA Webservice ist daher eng an diesen Teil der Spezifikation gebunden.

**Wenn für einen Service/Dienst ein gültiges Service Profile existiert** (beispielsweise per offizieller Definition und Auslieferung durch einen Fachstandard), ist das Element *ServiceQuality* durch den Autor zur Laufzeit bei Erteilung des Transportauftrags *mandatorisch* zu verwenden.

- In dem Element ist eine **Referenz auf die entsprechende ServiceProfil-Instanz** einzutragen. Eine ServiceProfil-Instanz ist eine XML-Instanz auf der Basis des globalen Elements gemäß Spezifikation in [Abschnitt 1.1.1.1 auf Seite 3](#).
- Wenn der Transportauftrag eine Referenz auf eine ServiceProfil-Instanz enthält, sind die darin enthaltenen Vorgaben zum Transport von allen beteiligten Infrastrukturkomponenten zu berücksichtigen. Bei der Entgegennahme des Transportauftrags ist der Sender dazu verpflichtet, ob der Eintrag korrekt vorgenommen ist.

**In allen anderen Kontexten** (es existiert keine vorgegebene ServiceProfilinstanz), **kann** das Element *ServiceQuality* verwendet werden.

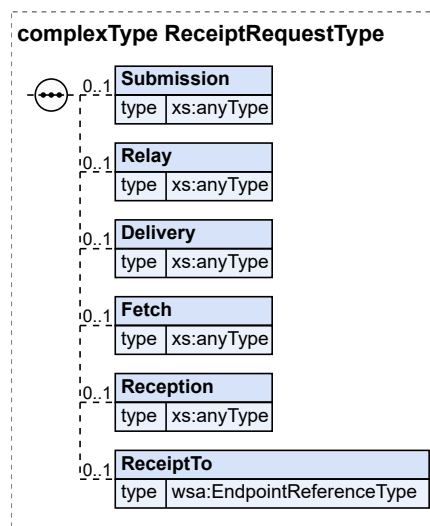
- Es wird dann eine lokal definierte ServiceProfil-Instanz referenziert.
- Alternativ wird das Element *nicht* verwendet (insbesondere dürfen keine informellen Inhalte als Freitext eingetragen werden).

**Allgemein gilt:**

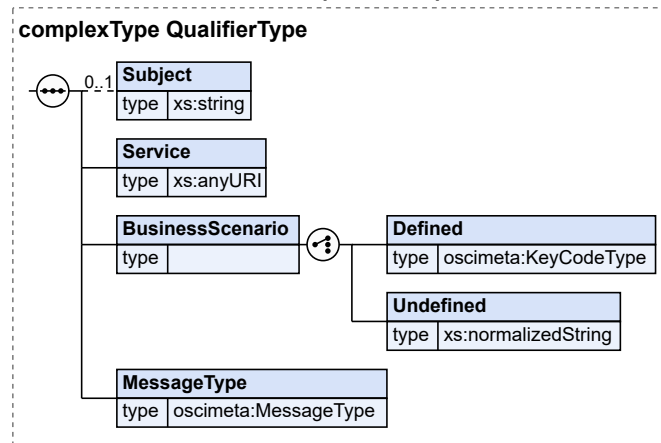
- In allen Fällen, in denen das Element *ServiceQuality* verwendet wird, ist eine Referenz auf eine (global oder lokal definierte) ServiceProfil-Instanz einzutragen.
- Das ist umzusetzen, indem der Inhalt des Elements *uriDerVersion* (vgl. [Abschnitt 1.1.5.2, „IdentifikationType“](#)) einer zu referenzierenden ServiceProfil-Instanz in das Element *ServiceQuality* des in Rede stehenden Transportauftrags eingetragen wird.

**2.1.1.2.5.3 Anforderungen von Quittungen (DeliveryAttributes)**

**ReceiptRequests:** Dieses Objekt gestattet es dem Initiator, im Rahmen des Transportauftrags Quittungen zu beantragen. Grundsätzliches zum Thema Quittungen in XTA ist dem XTA-Spezifikationsdokument Rahmenbedingungen zu entnehmen. Dort ist auch geregelt, wie sich die im Service Profil festgelegten Quittungen (vgl. Element technischeQuittungen in [Abschnitt 1.1.2.3.1, „SchutzkategorieType“](#)) zu den hier dokumentierten Einträgen im Objekt ReceiptRequests des Transportauftrags verhalten.



**2.1.1.2.5.4 Fachlicher Kontext eine Nachricht (Qualifier)**



Dieses Objekt nimmt Informationen auf, die sich auf den fachlichen Charakter der Transaktion beziehen. Es bietet für die Weiterverarbeitung Aspekte der fachlichen Einordnung des Payload an; auch für Transportprozesse, die nicht auf den Payload zugreifen sollen oder können. Das Objekt ist im Abschnitt [Abschnitt 2.2.1.48 auf Seite 77](#) spezifiziert.

### **Element *Subject***

Dieses optionale Element bietet Raum für informatorischen Begleittext.

### **Element *BusinessScenario***

Mittels dieses mandatorischen Elements wird das Geschäftsszenario genannt. Der Zweck der durch den Autor an dieser Stelle einzufüllenden Information des Transportauftrags ist, dass der Leser bereitliegende Transportnachrichten filtern kann ohne ihren Payload lesen bzw. analysieren zu müssen. Er kann die Information beispielsweise als Suchfeld für den Abruf aus dem Postfach einsetzen (das Element beerbt die Funktionalität der RefId aus OSCI 1.2)

Wenn im durch den Sender entgegengenommenen Transportauftrag in diesem Objekt keine korrekten Einträge enthalten sind, wird der Sender den Auftrag abweisen (Exception).

Das Business Scenario wird über eine im Objekt einzubindende Codeliste identifiziert. Die KoSIT gibt für diesen Zweck eine Codeliste heraus, welche Einträge für einschlägige Arten von Fachkontexten auflistet, die den Standard XTA anwenden. Diese Codeliste kann auf Antrag erweitert bzw. geändert werden. Sie ist durch XTA-konforme Systeme für übergreifende Prozesse zu verwenden.

Diese Codeliste ist im XRepository ([www.xrepository.de](http://www.xrepository.de)) unter Nennung ihrer Codelisten-URI *urn:de:xta:codeliste:business.scenario* auffindbar und kann dort im XML-Format OASIS Genericcode in der aktuellen Version abgerufen werden (ggf. sind auch frühere Versionen verfügbar). In die Attribute des vorliegenden Typs sind entsprechend ihre Codelisten-URI und die Nummer der ausgewählten Version einzutragen.

Für lokale Zwecke können XTA-Kommunikationspartner auch eigene Codelisten definieren (welche bilateral abgestimmte Reportformate benennen) und an dieser Stelle einbinden. In die Attribute des vorliegenden Typs werden dann Codelisten-URI und Versionsnummer der selbstdefinierten Codeliste eingetragen.

### **Element *Service***

Hier ist der durch den Leser angebotene Dienst einzutragen, der durch den Transportauftrag in Anspruch genommen werden soll. Der Inhalt dieses Elements ist nicht auf der Basis einer Codeliste eingeschränkt. Dieser Dienst ist mit der Syntax einer URI zu bezeichnen, die vom Betreiber eines Fachstandards bereitgestellt wird. Für Fachstandards aus XÖV ist dies gewöhnlich eine DVDV-bezogene Dienste-URI.

### **Element *MessageType***

In dieses mandatorische Element ist der Nachrichtentyp (Art der Fachnachricht) einzutragen, wie er innerhalb des Geschäftsszenarios (fachliches Nachrichtenformat; Fachstandard) definiert ist. Dies geschieht auf der Basis einer einzubindenden Codeliste. Passende Codelisten werden innerhalb der Geschäftsszenarios (z.B. innerhalb des XÖV-Standards) definiert und sind im XRepository ([www.xrepository.de](http://www.xrepository.de)) verfügbar. In die Attribute des vorliegenden Elements sind die Codelisten-URI (*@listURI*) und die Nummer der ausgewählten Version (*@listVersionID*) einzutragen. Außerdem ist in ein zusätzliches Attribut (*@payloadSchema*) der Namespace der entsprechenden Version des fachlichen Nachrichtenformats (des Fachstandards) einzutragen.

Codelisten-URIs passender einschlägiger Codelisten:

- OSCI-XMeld (ab Version 2.0): *urn:de:xmeld:schluesseltabelle:xmeld.nachrichten*
- XAusländer: *urn:de:xauslaender:codelist:nachrichtentyp*
- XPersonenstand: *urn:xpersonenstand:schluesseltabelle:nachrichtentyp*

Für das Geschäftsszenario XhD wurde durch XTA eine Codeliste definiert und veröffentlicht, welche mittelfristig von einer durch XhD herausgegebene Codeliste abgelöst werden soll:

- XhD: *urn:de:xta:codeliste:xhd-nachrichten*

### 2.1.1.3 Schnittstellentyp recipientPortType

Die Methoden des recipientPortType ermöglichen das Auslesen von empfangenen Nachrichten. Erfolgreich abgeholte Nachrichten sind zu kennzeichnen so dass sie im System als zugestellt markiert werden.

#### 2.1.1.3.1 Methode close

Mit dem Aufruf der Methode close signalisiert der Leser, dass die entsprechende Nachricht ordnungsgemäß abgeholt wurde, Die Nachricht gilt als erfolgreich zugestellt.

##### 2.1.1.3.1.1 Exceptions

- InvalidMessageIDException signalisiert, dass zu der angegebenen Nachrichtenkenung keine Protokollinformationen vorliegen.
- NotImplementedException informiert darüber, dass diese Methode nicht implementiert wurde.
- ParameterNotSupportedException wird geworfen, wenn eine Methode mit spezifikationsgemäßen, aber von der Implementation nicht unterstützen Parametern aufgerufen wird.
- PermissionDeniedException signalisiert einen Autorisierungsfehler.
- XTAWSTechnicalProblemException signalisiert einen allgemeinen Fehler während der Verarbeitung.

##### 2.1.1.3.1.2 Input

**Tabelle 2.29. Header**

WSDL Message	Message Part	Part Element
XTAHeader	XTAIdentifizier	oscimeta:Identifizier (vgl. <a href="#">Seite 72</a> )
Fachliche Identität des Autors oder Lesers.		

**Tabelle 2.30. Body**

WSDL Message	Message Part	Part Element
CloseRequest	FetchRequest	osci:MsgBoxCloseRequest (vgl. <a href="#">Seite 66</a> )
Steuerungsdaten zum Schließen einer MessageBox.		

##### 2.1.1.3.1.3 Output

Keine Header-Informationen.

**Tabelle 2.31. Body**

WSDL Message	Message Part	Part Element
EmptyBody		

#### 2.1.1.3.2 Methode getMessage

Mit dieser Methode wird genau eine den Selektionskriterien (vgl. [Abschnitt 2.2.1.24 auf Seite 68](#)) entsprechende Nachricht vom Empfänger abgeholt.

Entsprechen mehrere Nachrichten den Selektionskriterien liefert die Methode die zuerst beim Empfänger eingegangene Nachricht.

### 2.1.1.3.2.1 Exceptions

- `InvalidMessageIDException` signalisiert, dass zu der angegebenen Nachrichtenennung keine Protokollinformationen vorliegen.
- `NotImplementedException` informiert darüber, dass diese Methode nicht implementiert wurde.
- `ParameterNotSupportedException` wird geworfen, wenn eine Methode mit spezifikationsgemäßen, aber von der Implementation nicht unterstützten Parametern aufgerufen wird.
- `PermissionDeniedException` signalisiert einen Autorisierungsfehler.
- `XTAWSTechnicalProblemException` signalisiert einen allgemeinen Fehler während der Verarbeitung.

### 2.1.1.3.2.2 Input

**Tabelle 2.32. Header**

WSDL Message	Message Part	Part Element
XTAHeader	XTAIdentifizier	oscimeta:Identifizier (vgl. <a href="#">Seite 72</a> )
Fachliche Identität des Autors oder Lesers.		

**Tabelle 2.33. Body**

WSDL Message	Message Part	Part Element
FetchMsgRequest	FetchRequest	osci:MsgBoxFetchRequest (vgl. <a href="#">Seite 66</a> )
Das OSCI-Element <code>MsgBoxFetchRequest</code> enthält die Parameter für die Selektion der abzuholenden Nachrichten.		

### 2.1.1.3.2.3 Output

**Tabelle 2.34. Header**

WSDL Message	Message Part	Part Element
OptHeaders	FetchResponseHeader	osci:MsgBoxResponse (vgl. <a href="#">Seite 68</a> )
Dieser Container enthält die Nutzlast einer Antwort auf <code>MsgBox-Anfragen</code> ( <code>MsgBoxFetchRequest</code> und <code>MsgBoxStatusListRequest</code> ).		
XTAHeader	MessageMetaData	oscimeta:MessageMetaData (vgl. <a href="#">Seite 73</a> )
In dieser Struktur werden die Steuerungsdaten des Transportauftrags definiert.		

**Tabelle 2.35. Body**

WSDL Message	Message Part	Part Element
MessageBody	Message	xta:Message (vgl. <a href="#">Seite 79</a> )
Im Erfolgsfall wird eine Nachricht zusammen mit zugehörigen Metainformationen zurückgeliefert. Hierbei ist zu beachten, dass für eine abgeholte Nachricht der Status auf "abgeholt" geändert wird, nachdem die Transaktion durch Aufruf der Methode <code>close</code> bestätigt worden ist.		

### 2.1.1.3.3 Methode getNextMessage

Mit der Funktion "getNextMessage" kann das Abholen von Nachrichten durch mehrere parallele Abholinstanzen umgesetzt werden. Die erste Nachricht wird mittels der Funktion getMessage abgeholt. Weitere Nachrichten zu den gegebenen Selektionskriterien können dann mittels getNextMessage abgeholt werden.

Die Synchronisation der Abrufe erfolgt über den Parameter MsgBoxResponse. Wenn das parallele Abholen implementiert ist, liefert getMessage den optionalen Parameter zurück. Dieses Objekt ist den Abholinstanzen bekannt zu geben, so dass er beim erstmaligen Aufruf der Funktion getNextMessage mitgegeben werden kann. Der Parameter MsgBoxresponse zeigt an, ob weitere Nachrichten vorliegen.

#### 2.1.1.3.3.1 Exceptions

- NotImplementedException informiert darüber, dass diese Methode nicht implementiert wurde.
- ParameterNotSupportedException wird geworfen, wenn eine Methode mit spezifikationsgemäßen, aber von der Implementation nicht unterstützten Parametern aufgerufen wird.
- PermissionDeniedException signalisiert einen Autorisierungsfehler.
- XTAWSTechnicalProblemException signalisiert einen allgemeinen Fehler während der Verarbeitung.

#### 2.1.1.3.3.2 Input

**Tabelle 2.36. Header**

WSDL Message	Message Part	Part Element
XTAHeader	XTAIdentifier	oscimeta:Identifier (vgl. <a href="#">Seite 72</a> )
Fachliche Identität des Autors oder Lesers.		

**Tabelle 2.37. Body**

WSDL Message	Message Part	Part Element
GetNextRequest	FetchRequest	osci:MsgBoxGetNextRequest (vgl. <a href="#">Seite 67</a> )
MsgBoxNextRequest enthält das Handle zur Steuerung des sukzessiven parallelen Abholens von Nachrichten.		

#### 2.1.1.3.3.3 Output

**Tabelle 2.38. Header**

WSDL Message	Message Part	Part Element
OptHeaders	FetchResponseHeader	osci:MsgBoxResponse (vgl. <a href="#">Seite 68</a> )
Dieser Container enthält die Nutzlast einer Antwort auf MsgBox-Anfragen (MsgBoxFetchRequest und MsgBoxStatusListRequest).		
XTAHeader	MessageMetaData	oscimeta:MessageMetaData (vgl. <a href="#">Seite 73</a> )
In dieser Struktur werden die Steuerungsdaten des Transportauftrags definiert.		

**Tabelle 2.39. Body**

WSDL Message	Message Part	Part Element
MessageBody	Message	xta:Message (vgl. <a href="#">Seite 79</a> )



WSDL Message	Message Part	Part Element
Im Erfolgsfall liefert die Methode getNextMessage die nächste, noch nicht abgeholte Nachricht zusammen mit zugehörigen Metainformationen zu einer vorausgegangenen Abfrage mittels der Methode getMessage zurück. Hierbei ist zu beachten, dass für eine abgeholte Nachricht der Status auf "abgeholt" geändert wird, nachdem die Transaktion durch Aufruf entweder der Methode close oder der Methode getNextMessage bestätigt worden ist.		

#### 2.1.1.3.4 Methode getNextStatusList

Insbesondere bei umfangreichen Datenmengen oder bei parallelem Zugriff mehrerer Leser kann es sinnvoll sein, sich die Liste der MessageIDs und Metadaten für Nachrichten blockweise geben zu lassen. Der Vorgang des Abrufs einer Teilliste erfolgt, indem nur eine bestimmte Anzahl von MessageIDs und Metadaten erwartet und zurückgeliefert wird. Zusätzlich wird eine Ressourcenkennung („Handle“) für einen Iterator geliefert, mit der die weiteren Blöcke („Teillisten“) von MessageIDs mit der Methode getNextStatusList (siehe [Abschnitt 2.1.1.3.4 auf Seite 53](#)) abgeholt werden können.

Die Verwendung der Methode getNextStatusList steht im direkten Zusammenhang mit der Methode getStatusList: Der Leser kann mit dieser Methode Teillisten von MessageIDs von Metadaten vom Empfänger abholen, nachdem er die erste dieser Teillisten mit der Methode getStatusList (siehe [Abschnitt 2.1.1.3.5 auf Seite 54](#)) erhalten hat:

##### 2.1.1.3.4.1 Exceptions

- NotImplementedException informiert darüber, dass diese Methode nicht implementiert wurde.
- ParameterNotSupportedException wird geworfen, wenn eine Methode mit spezifikationsgemäßen, aber von der Implementation nicht unterstützten Parametern aufgerufen wird.
- PermissionDeniedException signalisiert einen Autorisierungsfehler.
- XTAWSTechnicalProblemException signalisiert einen allgemeinen Fehler während der Verarbeitung.

##### 2.1.1.3.4.2 Input

**Tabelle 2.40. Header**

WSDL Message	Message Part	Part Element
XTAHeader	XTAIdentifier	oscimeta:Identifier (vgl. <a href="#">Seite 72</a> )
Fachliche Identität des Autors oder Lesers.		

**Tabelle 2.41. Body**

WSDL Message	Message Part	Part Element
GetNextRequest	FetchRequest	osci:MsgBoxGetNextRequest (vgl. <a href="#">Seite 67</a> )
Abfrage-Daten für noch nicht erhaltene Nachrichten einer Ergebnismenge in einer MessageBox.		

##### 2.1.1.3.4.3 Output

**Tabelle 2.42. Header**

WSDL Message	Message Part	Part Element
OptHeaders	FetchResponseHeader	osci:MsgBoxResponse (vgl. <a href="#">Seite 68</a> )

WSDL Message	Message Part	Part Element
Dieser Container enthält die Nutzlast einer Antwort auf MessageBox-Anfragen (MsgBoxFetchRequest und MsgBoxStatusListRequest).		

Tabelle 2.43. Body

WSDL Message	Message Part	Part Element
FetchListResponse	FetchResponse	osci:MsgStatusList (vgl. <a href="#">Seite 69</a> )
Liste der Ergebnisparameter. Liegen für die Selektionskriterien keine Nachrichten vor, ist die Liste leer. Im Ergebnis-Header (MsgBoxResponse) werden Zusatzinformationen zum Anfragevorgang und seiner Ergebnisliste geliefert.		

### 2.1.1.3.5 Methode getStatusList

Die Methode liefert eine Liste mit Informationen über die beim Empfänger den Selektionskriterien entsprechenden vorliegenden Nachrichten. Neben den Selektionskriterien für die Nachrichten wird die maximale Länger der Ergebnisliste und die Art der Informationen (Nachrichtenattribute oder Metadaten der Fachnachricht) angegeben.

#### 2.1.1.3.5.1 Exceptions

- NotImplementedException informiert darüber, dass diese Methode nicht implementiert wurde.
- ParameterNotSupportedException wird geworfen, wenn eine Methode mit spezifikationsgemäßen, aber von der Implementation nicht unterstützen Parametern aufgerufen wird.
- PermissionDeniedException signalisiert einen Autorisierungsfehler.
- XTAWSTechnicalProblemException signalisiert einen allgemeinen Fehler während der Verarbeitung.

#### 2.1.1.3.5.2 Input

Tabelle 2.44. Header

WSDL Message	Message Part	Part Element
XTAHeader	XTAIdentifier	oscimeta:Identifier (vgl. <a href="#">Seite 72</a> )
Fachliche Identität des Autors oder Lesers.		

Tabelle 2.45. Body

WSDL Message	Message Part	Part Element
FetchListRequest	FetchRequest	osci:MsgBoxStatusListRequest (vgl. <a href="#">Seite 69</a> )
Abfrage-Parameter für Statusinformationen au seiner MessageBox.		

#### 2.1.1.3.5.3 Output

Tabelle 2.46. Header

WSDL Message	Message Part	Part Element
OptHeaders	FetchResponseHeader	osci:MsgBoxResponse (vgl. <a href="#">Seite 68</a> )

WSDL Message	Message Part	Part Element
Dieser Container enthält die Nutzlast einer Antwort auf MessageBox-Anfragen (MsgBoxFetchRequest und MsgBoxStatusListRequest).		

Tabelle 2.47. Body

WSDL Message	Message Part	Part Element
FetchListResponse	FetchResponse	osci:MsgStatusList (vgl. Seite 69)
Liste der Ergebnisparameter. Liegen für die Selektionskriterien keine Nachrichten vor, ist die Liste leer. Im Ergebnis-Header (MsgBoxResponse) werden Zusatzinformationen zum Anfragevorgang und seiner Ergebnisliste geliefert.		

### 2.1.1.3.6 Wichtige Objekte der recipientPort-Schnittstelle

#### 2.1.1.3.6.1 MsgSelector

Dieses Objekt enthält die Selektionsmöglichkeiten der Pull-Mechanismen auf die OSCI Message-Box. Es ist im Abschnitt [Abschnitt 2.2.1.24 auf Seite 68](#) spezifiziert.

#### 2.1.1.3.6.2 MsgBoxResponse

MsgBoxResponse enthält die Metainformationen zur Anfrage (Request) und ist im Abschnitt [Abschnitt 2.2.1.22 auf Seite 68](#) spezifiziert.

#### 2.1.1.3.6.3 MsgBoxFetchRequest

MsgBoxFetchRequest wird verwendet, um eine Nachricht anzufordern und enthält die erforderlichen Selektionskriterien in MsgSelector sowie das Attribut **@MsgPart**. Das Objekt ist im Abschnitt [Abschnitt 2.2.1.20 auf Seite 67](#) spezifiziert.

#### 2.1.1.3.6.4 Sukzessiver Abruf von Nachrichten aus dem Postfach

Bei einem asynchronen Empfang nimmt der Empfänger die Nachrichten entgegen und hält diese für den Leser für eine Abholung bereit. Der Leser kann die Nachrichten zu einem von ihm bestimmten Zeitpunkt abholen.

In der folgenden Darstellung der Abholung von Nachrichten gibt es nur einen aktiven Leser. Für eine parallele Abholung durch mehrere Leser reserviert ein Leser in Schritt 1 einen Ressourcenhandle. Diesen reicht er an andere Leser weiter, die dann die folgenden Arbeitsschritte 2 und 3 parallel zu ihm durchführen.

##### 1. Abholung der ersten Nachricht mit Kriterien

Der Leser holt die erste Nachricht ab (vgl. Rollenmodell D5.1). Hierbei kann er Selektionskriterien (gelesen, ungelesen; Zeitraum des Empfangs) angeben. Mit der ersten Nachricht bekommt der Leser eine Ressourcenkennung für einen „Iterator“, die er für die weitere Abholung von Nachrichten benötigt. (Wenn mehrere Leser auf die Nachrichten zugreifen wollen, reicht er diese Ressourcenkennung an andere Leser weiter.)

- Abholung der ersten Nachricht (siehe [Abschnitt 2.1.1.3.2 auf Seite 50](#))

##### 2. Überprüfung der Kommunikation

Der Leser überprüft, ob der Transport der Nachricht erfolgreich durchgeführt werden konnte (vgl. Rollenmodell D7.1, D 8.1), z. B. ob die verwendeten Zertifikate gültig waren. Bei positivem Ergebnis kann er die Nachricht des Autors verarbeiten. Im Falle eines Misserfolgs muss er ggf. Eskalationsmaßnahmen ergreifen.

XTA Funktionalitäten:

- Abruf eines Transportprotokolls (siehe [Abschnitt 2.1.1.1.3 auf Seite 38](#))

### 3. Liste der Nachrichten mit Iterator abholen

Der Leser hat für das Abholen der Nachrichten vom Empfänger eine Ressourcenkennung erhalten. Unter Angabe dieser Kennung kann er die nächste Nachricht abholen (vgl. Rollenmodell D5.1). Dabei sollte er die MessageID der zuletzt abgeholten Nachricht mit angeben. Dadurch quittiert er die erfolgreiche Abholung dem Empfänger. Liegt keine weitere Nachricht vor, dann liefert der Empfänger eine entsprechende Meldung zurück.

XTA Funktionalitäten:

- Abholen einer weiteren Nachricht

(siehe [Abschnitt 2.1.1.3.3 auf Seite 52](#))

### 4. Überprüfung der Kommunikation

Der Leser überprüft in derselben Weise wie bereits zuvor, ob der Transport der Nachricht erfolgreich durchgeführt werden konnte (vgl. Rollenmodell D7.1, D 8.1).

### 5. Beenden der Abholung von Nachrichten

Wenn der Leser alle Nachrichten abgeholt hat, soll er dies dem Empfänger mitteilen, indem er abschließend eine Quittung sendet. Diese Information unterstützt den Empfänger bei der Koordination des parallelen Zugriffs.

XTA Funktionalität:

- Quittieren der Abholung (siehe [Abschnitt 2.1.1.3.1 auf Seite 50](#))

## 2.1.2 Die Schnittstellentypen der Applikationsebene

### 2.1.2.1 Schnittstellentyp authorPortType

In diesem Schnittstellentyp sind die Methoden des XTA-WS zusammengefasst, die vom Autor für den Sender zur Verfügung gestellt werden müssen.

#### 2.1.2.1.1 Methode notifyMessageSent

Die Methode notifyMessageSent ist vom Autor bereitzustellen und wird vom Sender aufgerufen, um den Autor über die Zustellung einer Nachricht zu informieren. Die Methode wird nur für Nachrichten aufgerufen, wenn in oscimeta:DeliveryAttributesType die Benachrichtigung gefordert wird.

##### 2.1.2.1.1.1 Exceptions

- Der technische Fehler (SoapFault) InvalidMessageIDException entsteht, wenn die anhand der ID angeforderte Nachricht dem Account nicht bekannt ist.
- NotImplementedException wird zurückgegeben, wenn eine nicht implementierte Methode aufgerufen wird.
- Der technische Fehler (SoapFault) ParameterIsNotValidException wird zurückgegeben, wenn ein Pflicht-Übergabeparameter fehlt oder ein Übergabeparameter fehlerhaft ist.
- Der technische Fehler (SoapFault) PermissionDeniedException entsteht, wenn ein nicht autorisierter Zugriff auf den XTA-WS stattfindet.
- Der technische Fehler (SoapFault) XTAWSTechnicalProblemException entsteht, wenn ein technischer Fehler im XTA-WS aufgetreten ist.

##### 2.1.2.1.1.2 Input

Keine Header-Informationen.

Tabelle 2.48. Body

WSDL Message	Message Part	Part Element
MessageSentReport	MessageSentReport	xta:TransportReport (vgl. <a href="#">Seite 80</a> )
Das Transportprotokoll der Nachricht, über deren Versand mit der Notifikation informiert wird.		

#### 2.1.2.1.1.3 Output

Keine Header-Informationen.

Tabelle 2.49. Body

WSDL Message	Message Part	Part Element
EmptyBody		
Im Erfolgsfall werden keine Informationen vom Autor an den Sender zurückgegeben. Der Sender kann daraus schließen, dass der Autor die Benachrichtigung erhalten hat, dass der Versand der von ihm beauftragten Nachricht abgeschlossen ist.		

#### 2.1.2.1.2 Wichtige Objekte der authorPort-Schnittstelle

#### 2.1.2.2 Schnittstellentyp readerPortType

In diesem Schnittstellentyp sind die Methoden des XTA-WS zusammengefasst, die vom Leser für den Empfänger zur Verfügung gestellt werden müssen.

##### 2.1.2.2.1 Methode deliverMessage

Mit der Methode deliverMessage leitet der Empfänger eine Nachricht synchron an den Leser weiter. Die Antwort des Lesers wird innerhalb derselben Transaktion an den Empfänger weitergereicht.

Die Signatur der Funktion ist in der Datei xta-application.wsdl enthalten.

##### 2.1.2.2.1.1 Exceptions

- Der technische Fehler (SoapFault) MessageSchemaViolationException entsteht, wenn die zum Versand übergebene Fachnachricht nicht konform zur jeweiligen XML Schema-Definition ist. Insbesondere entsteht der Fehler dann, wenn in der übergebenen Nachricht ein fehlerhaftes Encoding eingestellt ist oder wenn das entsprechende Service Profil verletzt ist.
- Der technische Fehler (SoapFault) MessageVirusDetectionException entsteht, wenn in der Fachnachricht schadhafter Code ermittelt wurde.
- NotImplementedException wird zurückgegeben, wenn eine nicht implementierte Methode aufgerufen wird.
- Der technische Fehler (SoapFault) ParameterIsNotValidException wird zurückgegeben, wenn ein Pflicht-Übergabeparameter fehlt oder ein Übergabeparameter fehlerhaft ist.
- ParameterNotSupportedException wird geworfen, wenn eine Methode mit spezifikationsgemäßen, aber von der Implementation nicht unterstützen Parametern aufgerufen wird.
- Der technische Fehler (SoapFault) PermissionDeniedException entsteht, wenn ein nicht autorisierter Zugriff auf den XTA-WS stattfindet.
- Der technische Fehler (SoapFault) SyncAsyncException entsteht, wenn eine Fachnachricht übergeben wurde, die nur für den synchronen Versand gültig ist.
- Der technische Fehler (SoapFault) XTAWSTechnicalProblemException entsteht, wenn ein technischer Fehler im XTA-WS aufgetreten ist.

### 2.1.2.2.1.2 Input

**Tabelle 2.50. Header**

WSDL Message	Message Part	Part Element
XTAHeader	MessageMetaData	oscimeta:MessageMetaData (vgl. <a href="#">Seite 73</a> )
In dieser Struktur werden die Steuerungsdaten des Transportauftrags definiert.		

**Tabelle 2.51. Body**

WSDL Message	Message Part	Part Element
MessageBody	Message	xta:Message (vgl. <a href="#">Seite 79</a> )
Dieses Objekt beinhaltet die zu übertragende Fachnachricht und eine beliebige Anzahl von Anhängen (Attachments). Die Fachnachricht kann in einem verschlüsselten Container hinterlegt werden. Zu der Fachnachricht kann ein Betreff (Subject) angegeben werden.		

### 2.1.2.2.1.3 Output

**Tabelle 2.52. Header**

WSDL Message	Message Part	Part Element
XTAHeader	MessageMetaData	oscimeta:MessageMetaData (vgl. <a href="#">Seite 73</a> )
In dieser Struktur werden die Steuerungsdaten des Transportauftrags definiert.		

**Tabelle 2.53. Body**

WSDL Message	Message Part	Part Element
MessageResponseBody	MessageResponseContentContainer	xta:MessageResponse (vgl. <a href="#">Seite 81</a> )
Im Erfolgsfall muss das Objekt <i>MessageMetaDataResponse</i> mit der Struktur <i>MessageMetaData</i> im Header der Antwortnachricht befüllt werden.		
Als Rückgabewert wird im Erfolgsfall das Element <i>MessageResponse</i> zurückgegeben, das die Antwort des Kommunikationspartners enthält. Dabei sind folgende Fälle zu unterscheiden:		
<ul style="list-style-type: none"> <li>Falls im Fachszenario eine synchrone Fachantwort gefordert ist, ist diese im Kindelement <i>Message</i> enthalten. Sie besteht aus einer Fachnachricht und ggf. einer Anzahl von Anhängen (Attachments). Zu der Fachnachricht kann ein Betreff (Subject) angegeben werden.</li> <li>Falls im Fachszenario <i>keine</i> synchrone Fachantwort gefordert ist, darf das Objekt <i>MessageMetaDataResponse</i> im Header der Antwortnachricht nicht befüllt werden, und die Struktur <i>MessageResponse</i> bleibt leer, d.h. sie enthält kein Kindelement <i>Message</i>.</li> </ul>		

### 2.1.2.2.2 Methode notifyMessageReceived

Mit der Methode `notifyMessageReceived` informiert der Empfänger den Leser über den Eingang einer Nachricht. Die Methode wird nur für die vom Leser mit dem Empfänger vereinbarten Postfächer verwendet. Der Leser wird für jede Nachricht der vereinbarten Postfächer nur einmal über deren Eingang informiert.

### 2.1.2.2.1 Exceptions

- NotImplementedException wird zurückgegeben, wenn eine nicht implementierte Methode aufgerufen wird.
- Der technische Fehler (SoapFault) NotificationForFetchedMessageException wird zurückgegeben, wenn der Empfänger dem Leser MessageMetaData für eine Nachricht übergibt, die der Leser bereits erfolgreich abgeholt hat. Der Fehler kann pro Nachricht mehrfach auftreten, z.B. aufgrund von Latenzen. Leser und Empfänger müssen festlegen, wie damit umgegangen wird, wenn der Fehler mehrfach auftritt.
- Diese Fehlermeldung wird geworfen, wenn ein Parameter nicht korrekt an die Methode übergeben wurde.
- ParameterNotSupportedException wird geworfen, wenn eine Methode mit spezifikationsgemäßen, aber von der Implementation nicht unterstützten Parametern aufgerufen wird.
- Diese Exception wird geworfen, wenn der Account gesperrt oder nicht vorhanden ist.
- Diese Exception wird allgemein geworfen, wenn ein technisches Problem im XTA-WS aufgetreten ist. Sie kann z. B. durch ein Problem beim Zugriff auf die interne Datenbank des XTA-Servers ausgelöst worden sein.

### 2.1.2.2.2 Input

**Tabelle 2.54. Header**

WSDL Message	Message Part	Part Element
XTAHeader	MessageMetaData	oscimeta:MessageMetaData (vgl. <a href="#">Seite 73</a> )
In dieser Struktur werden die Steuerungsdaten des Transportauftrags definiert.		

**Tabelle 2.55. Body**

WSDL Message	Message Part	Part Element
EmptyBody		

### 2.1.2.2.3 Output

**Tabelle 2.56. Header**

WSDL Message	Message Part	Part Element
XTAHeader	MessageMetaData	oscimeta:MessageMetaData (vgl. <a href="#">Seite 73</a> )
In dieser Struktur werden die Steuerungsdaten des Transportauftrags definiert.		

**Tabelle 2.57. Body**

WSDL Message	Message Part	Part Element
EmptyBody		

### 2.1.2.2.3 Wichtige Objekte der readerPort-Schnittstelle

Für den Versand synchroner Nachrichten sind dieselben wichtigen Objekte zu berücksichtigen, wie für den Versand asynchroner Nachrichten (vgl. [Abschnitt 2.1.1.2.5](#), „Wichtige Objekte der sender-Port-Schnittstelle“).

### 2.1.2.2.3.1 MessageMetaDataResponse

Das Mantelobjekt MessageResponse ermöglicht, dass die optionale Verwendung des Message in der synchronen Antwort.

## 2.2 Das XTA-WS-Informationsmodell

Dieser Abschnitt dokumentiert die Objekte XTA Webservices..

### 2.2.1 Datentypen der Informationsobjekte des XTA Webservice

Hier werden die Bausteine beschrieben, aus denen sich die Informationsobjekte der Methodenaufrufe des XTA Webservice zusammensetzen.

#### 2.2.1.1 RecordType

Typ: *RecordType*

Der Typ zur Kennzeichnung und Erläuterung einer Meldung (anwendbar auf Info-, Fehlermeldungen und Warnungen).

Kindelemente von <i>RecordType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>Timestamp</b>	<i>xs:dateTime</i>	1		
Zeitstempel für den Zeitpunkt der Aufzeichnung der Meldung.				
<b>Code</b>	<i>xta:Code.RecordType</i>	1	<a href="#">2.2.1.2</a>	<a href="#">60</a>
Schlüssel, der die Bedeutung der Meldung kodiert. Dieser Schlüssel muss aus einer eingebundenen Codeliste stammen.				
<b>Reason</b>	<i>xs:string</i>	1		
Hier wird zur weiteren Erläuterung der Grund der Meldung als Freitext eingetragen.				

#### 2.2.1.2 Code.RecordType

In diesen Typ ist eine auszuwählende bzw. selbst zu definierende Codeliste einzubinden, die Arten von Meldungen benennt, welche in das Protokoll zur Abarbeitung eines Transportauftrags (TransportReport) eingetragen werden. Dort können die Meldungen als Fehler-, Warn- oder Informationseinträge eingeordnet sein.

In die Attribute des vorliegenden Typs sind die Codelisten-URI und die Nummer der Version der ausgewählten Codeliste einzutragen.

Die KoSIT gibt für den Standard XTA eine Codeliste heraus, welche Einträge für einschlägige Arten von Reports auflistet. Diese Codeliste kann auf Antrag erweitert bzw. geändert werden. Sie ist durch XTA-konforme Systeme für übergreifende Prozesse zu verwenden.

Diese Codeliste ist im XRepository ([www.xrepository.de](http://www.xrepository.de)) unter Nennung ihrer Codelisten-URI *urn:de:xta:codeliste:record.type* auffindbar und kann dort im XML-Format OASIS Genericcode in der aktuellen Version abgerufen werden (ggf. sind auch frühere Versionen verfügbar). In die Attribute des vorliegenden Typs sind entsprechend ihre Codelisten-URI und die Nummer der ausgewählten Version einzutragen.

Für lokale Zwecke können XTA-Kommunikationspartner auch eigene Codelisten definieren (welche bilateral abgestimmte Reportformate benennen) und an dieser Stelle einbinden. In die Attribute des vorlie-



genden Typs werden dann Codelisten-URI und Versionsnummer der selbstdefinierten Codeliste eingetragen.

Codelisten	
-beschreibung	unbestimmt
-nutzung	Typ: 4, siehe Beschreibung
-kennung	unbestimmt
-version	unbestimmt

### 2.2.1.3 Code.ZertifikatStatus

Codelisten	
-beschreibung	Diese Codeliste enthält die Schlüssel für den Status eines Zertifikats.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 119</a>
-kennung	urn:xoev-de:xta:serviceprofile:codeliste:zertifikat.status
-version	1.0

### 2.2.1.4 AdditionalReportListType

Typ: *AdditionalReportListType*

Dieser Typ gestattet das Ablegen weiterer Prüfberichte, welche das XTA-Protokoll (*TransportReport*) ergänzen sollen.

Kindelement von <i>AdditionalReportListType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>Report</b>		1..n		
In diesem Element ist ein zusätzlicher Report abgelegt, der das XTA-Protokoll ( <i>TransportReport</i> ) ergänzt. Die Art des Reports (z. B. „OSCI Process Card“) und der Inhalt des Reports werden bzw. sind in separaten Bereichen dieses Containers eingetragen.				
<b>Key</b>	<i>xta:Code.ReportType</i>	1	<a href="#">2.2.1.5</a>	61
Dieses Element benennt den Typ des Reports, um dem Leser die Interpretation der Reportdaten zu ermöglichen. Die Benennung des Typs des Reports geschieht auf der Basis einer Codeliste.				
<b>Data</b>	<i>xs:base64Binary</i>	1		
Hier wird der zusätzliche Report in einem technisch neutralen Format eingetragen.				

### 2.2.1.5 Code.ReportType

Dieser Typ gestattet die Kennzeichnung der Art eines zusätzlichen Reports. Es wird eine zu wählende Codeliste eingebunden, die mögliche Arten von Reports nennt (spezielles Format, innerhalb oder außerhalb von XTA definiert), die in das XTA-Protokoll (*TransportReport*) eingefügt werden können.

Die KoSIT gibt für den Standard XTA eine Codeliste heraus, welche Einträge für einschlägige Arten von Reports auflistet. Diese Codeliste kann auf Antrag erweitert bzw. geändert werden. Sie ist durch XTA-konforme Systeme für übergreifende Prozesse zu verwenden.

Diese Codeliste ist im XRepository ([www.xrepository.de](http://www.xrepository.de)) unter Nennung ihrer Codelisten-URI *urn:de:xta:codeliste:report.type* auffindbar und kann dort im XML-Format OASIS Genericcode in der aktuellen Ver-

sion abgerufen werden (ggf. sind auch frühere Versionen verfügbar). In die Attribute des vorliegenden Typs sind entsprechend ihre Codelisten-URI und die Nummer der ausgewählten Version einzutragen.

Für lokale Zwecke können XTA-Kommunikationspartner auch eigene Codelisten definieren (welche bilateral abgestimmte Reportformate benennen) und an dieser Stelle einbinden. In die Attribute des vorliegenden Typs werden dann Codelisten-URI und Versionsnummer der selbstdefinierten Codeliste eingetragen.

Codelisten	
-beschreibung	unbestimmt
-nutzung	Typ: 4, siehe Beschreibung
-kennung	unbestimmt
-version	unbestimmt

### 2.2.1.6 Code.EreignisArtType

Codelisten	
-beschreibung	Diese Codeliste definiert die für den ServiceReport zu protokollierenden Ereignisarten.
-nutzung	Typ: 1, Inhalte der Codeliste siehe <a href="#">Seite 92</a>
-kennung	urn:xoev-de:xta:webservice:codeliste:ereignisart
-version	1.0

### 2.2.1.7 MessageStatusType

Typ: *MessageStatusType*

Gibt die Struktur für die Meldungen (Logging-Informationen) über den Transportverlauf vor. Er sieht Meldungszeilen für Infos, Warnungen und Fehler vor.

Kindelemente von <i>MessageStatusType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>Status</b>	<i>xs:integer</i>	<b>1</b>		
Wird durch Sender bzw. Empfänger fortgeschrieben. Wird der TransportReport noch fortgeschrieben, wird er hier mit 0=open markiert. Nach Abschluss des TransportReports wird nach dem Max-Prinzip der höchste Ampelstatus aus den Elementen ErrorList, WarnList, InfoList hier numerisch dargestellt.				
<ul style="list-style-type: none"> <li>• 0=open: Die Nachricht befindet sich noch in der Verarbeitung.</li> <li>• 1=grün: Es sind keine Fehler oder Warnungen aufgetreten.</li> <li>• 2=gelb: Es sind Warnungen, aber keine kritischen Fehler aufgetreten.</li> <li>• 3=rot: Es sind kritische Fehler aufgetreten.</li> </ul>				
<b>ErrorList</b>		<b>0..1</b>		
Liste der Fehlermeldungen.				
<b>Error</b>	<i>xta:RecordType</i>	<b>0..n</b>	<a href="#">2.2.1.1</a>	<a href="#">60</a>
Hier wird die Fehlermeldung mit ihren Parametern eingetragen.				
<b>WarnList</b>		<b>0..1</b>		
Liste der Warnungen.				
<b>Warning</b>	<i>xta:RecordType</i>	<b>0..n</b>	<a href="#">2.2.1.1</a>	<a href="#">60</a>
Hier wird die Warnung mit ihren Parametern eingetragen.				

Kindelemente von <i>MessageStatusType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>InfoList</b>		0..1		
Liste der Informationsmeldungen.				
<b>Info</b>	<i>xta:RecordType</i>	0..n	2.2.1.1	60
Hier wird die Informationsmeldung mit ihren Parametern eingetragen.				

### 2.2.1.8 LookupServiceType

Typ: *LookupServiceType*

Dies ist die Struktur einer Service-Anfrage: Sie enthält die Daten über den Diensteanbieter (Leser) und den Dienst des Lesers, den der Autor in Anspruch nehmen will. Diese Anfrage dient dazu, zu ermitteln, ob der Dienst von diesem Anbieter angeboten wird und über welche technischen Parameter er angesprochen werden kann.

Kindelement von <i>LookupServiceType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>ServiceType</b>	<i>xs:anyURI</i>	1		
Dies ist die Bezeichnung des anzufordernden Dienstes. Sie wird im Format einer URL übergeben, was den Vorteil hat, dass damit auch eine Versionsnummer eingeschlossen ist. Beispiel für Dienstbezeichnungen, wie sie im DVDV verwendet werden: <a href="http://www.osci.de/xmeld181/xmeld181Rueckmeldung.wsdl">http://www.osci.de/xmeld181/xmeld181Rueckmeldung.wsdl</a>				
Abgrenzung: "Dienst" ist das, was gemäß Diensteeinteilung der Fachdomäne im Verzeichnisdienst als Service (im Sinne eines Web Service) eingetragen ist. Dadurch ist die Dienstbezeichnung weniger differenziert als der Nachrichtentyp. Typischerweise sind im Verzeichnisdienst mehrere Nachrichtentypen in einer Service-WSDL zusammengefasst.				

### 2.2.1.9 LookupServiceResultType

Typ: *LookupServiceResultType*

Das Ergebnis zu einer Dienstanfrage, das die Information enthält, ob der Dienst angeboten wird. Außerdem sind die nötigen technischen Parameter für die Erreichbarkeit vorhanden.

Dieser Typ ist eine Erweiterung des Basistyps *LookupServiceType* (siehe [Abschnitt 2.2.1.8 auf Seite 63](#)).

Kindelemente von <i>LookupServiceResultType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>IsServiceAvailableValue</b>	<i>xta:IsServiceAvailableValueType</i>	1	2.2.1.10	64
Enthält das Ergebnis der Dienstanfrage: ob der Dienst angeboten wird oder nicht oder ob diese Information generell nicht bekannt ist.				
<b>ServiceParameter</b>		0..n		
Dieses Element enthält im Erfolgsfall die benötigten technischen Parameter für die elektronische Kommunikation mit dem Leser, z.B. das öffentliche Zertifikat des Lesers zur Inhaltsdatenverschlüsselung. Das Feld ist zu füllen, falls der angefragte Dienst angeboten und in diesem Kontext der Parameter benötigt wird.				
Vom Fachszenario ist zu beschreiben, welche Parameter für die Erreichbarkeit der Dienste in diesem Fachszenario anzuwenden sind.				
<b>ParameterType</b>	<i>xta:Code.ServiceParameterType</i>	1	2.2.1.11	64

Kindelemente von <i>LookupServiceResultType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Dieses Element steht für die Art des Parameters, welche ins passende Kindelement einzutragen bzw. eingetragen ist. Die vorgesehenen Parameterarten werden auf der Basis einer Codeliste interpretiert, welche durch die Attribute <i>listURI</i> und <i>listVersionID</i> referenziert ist.				
<b>Resource</b>	<i>xs:base64Binary</i>	<b>1</b>		
Hier ist der Parameter enthalten bzw. einzutragen in technisch neutraler Darstellung.				

### 2.2.1.10 *IsServiceAvailableValueType*

Typ: *IsServiceAvailableValueType*

Das Feld enthält die benötigten Attribute zum Ergebnis der Dienstanfrage: ob der Dienst angeboten wird oder nicht, oder ob diese Information generell nicht bekannt ist.

Kindelemente von <i>IsServiceAvailableValueType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Auswahl: <b>ServiceIsAvailable</b>	<i>xs:boolean</i>	<b>1</b>		
Der Dienst wird angeboten (true) oder nicht angeboten (false).				
Auswahl: <b>ServiceIsAvailableUnknown</b>	<i>xs:boolean</i>	<b>1</b>		
Es ist nicht bekannt, ob der Dienst angeboten wird oder nicht. In diesem Element/Attribut ist nur Wert „true“ zulässig (fixed-Wert).				

### 2.2.1.11 *Code.ServiceParameterType*

Dieser Typ gestattet die Kennzeichnung der Art eines Parameters für die technische Erreichbarkeit des Dienstes, der adressiert werden soll.

Hier wird eine zu wählende Codeliste eingebunden, die mögliche Parameterarten nennt.

Die KoSIT gibt für den Standard XTA eine Codeliste heraus, welche einschlägige solcher Parameterarten auflistet. Diese Codeliste kann auf Antrag erweitert bzw. geändert werden. Sie ist durch XTA-konforme Systeme für übergreifende Prozesse zu verwenden.

Diese Codeliste ist im XRepository ([www.xrepository.de](http://www.xrepository.de)) unter Nennung ihrer Codelisten-URI *urn:de:xta:codeliste:service.parameter.type* auffindbar und kann dort im XML-Format OASIS Genericode in der aktuellen Version abgerufen werden (ggf. sind auch frühere Versionen verfügbar). In die Attribute des vorliegenden Typs sind entsprechend ihre Codelisten-URI und die Nummer der ausgewählten Version einzutragen.

Für lokale Zwecke können XTA-Kommunikationspartner auch eigene Codelisten definieren (welche bilateral abgestimmte Parameterarten benennen) und an dieser Stelle einbinden. In die Attribute des vorliegenden Typs werden dann Codelisten-URI und Versionsnummer der selbstdefinierten Codeliste eingetragen.

Codelisten	
-beschreibung	unbestimmt
-nutzung	Typ: 4, siehe Beschreibung
-kennung	unbestimmt

Codelisten	
-version	unbestimmt

### 2.2.1.12 ContentType

Typ: *ContentType*

Typ für die technisch neutrale (base64-kodierte) Darstellung von Information. Enthält den base64-kodierten Inhalt (Fachnachricht), der zwischen Webservice-Client und XTA-Server transportiert wird. Die Attribute sind der MIME-Spezifikation (RFC 2183) entnommen.

Die Belegung der Attribute ist für verschiedene Fachlichkeiten unterschiedlich und ist durch den Fachstandard festzulegen, der für die Fachnachricht verantwortlich ist.

Dieser Typ ist eine Erweiterung des Basistyps *xs:base64Binary*.

Kindelemente von <i>ContentType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>category</b>	<i>xta:ContentCategory</i>	1		
Auszeichnung der Inhaltskategorie, z.B. body (Nachricht) oder attachment (Anhang).				
<b>contentDescription</b>	<i>oscimeta:NonEmptyStringType</i>	0..1	2.2.1.40	75
Beschreibung des fachlichen Inhalts, z.B. 'Angebot' oder 'Rechnung'.				
<b>contentType</b>	<i>oscimeta:NonEmptyStringType</i>	1	2.2.1.40	75
Dieses Attribut nennt den MIME-Typ des enthaltenen Inhalts, hat also Einträge wie text/xml, text/plain, application/gzip oder application/pdf. Mandatorisch, weil besonders wichtige Information (wird in E-Mail analog gehandhabt).				
<b>encoding</b>	<i>oscimeta:NonEmptyStringType</i>	0..1	2.2.1.40	75
Der Zeichensatz, der der Kodierung des Inhalts zugrunde gelegen hat.				
<b>filename</b>	<i>oscimeta:NonEmptyStringType</i>	0..1	2.2.1.40	75
Der Dateiname der Datenquelle, falls der Inhalt einer Datei entnommen worden ist. Bsp.: Für die Übermittlung von xdomea-Nachrichten ist dieses Attribut Pflicht.				
<b>id</b>	<i>xs:ID</i>	0..1		
Bietet die Möglichkeit, den Inhalt über z.B. eine laufende Nummer zu referenzieren.				
<b>lang</b>	<i>xs:language</i>	0..1		
Sprache, in der der Inhalt formuliert ist.				
<b>refID</b>	<i>xs:IDREF</i>	0..1		
Bezug auf die ID eines anderen Objekts MessagePart, z.B. zur Zuordnung von Anhängen zu Nachrichten.				
<b>size</b>	<i>xs:positiveInteger</i>	0..1		
Die Größe des Inhalts in Bytes.				

### 2.2.1.13 MsgAttributeListType

Typ: *MsgAttributeListType*

Container für die Attribute/Eigenschaften einer Nachricht in der MessageBox.

Kindelemente von <i>MsgAttributeListType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>wsa:MessageID</b> (ref)		1	B.2	121

Kindelemente von <i>MsgAttributeListType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Eindeutige ID der Nachricht.				
<b>wsa:RelatesTo</b> (ref)		1	B.2	121
Bezug zu weiteren Nachrichten anhand von deren MessageIDs.				
<b>wsa:From</b> (ref)		1	B.2	121
Ursprung der Nachricht (als Endpoint Reference).				
<b>osci:TypeOfBusinessScenario</b> (ref)		1	2.2.1.29	70
Geschäftsszenario der Nachricht für Zuordnung bzw gefilterten Abruf durch den Leser.				
<b>MsgSize</b>	<i>xs:integer</i>	1		
Größe der Nachricht in Kilobytes als positiver Integer.				
<b>ObsoleteAfterDate</b>	<i>xs:date</i>	1		
Verfallsdatum der Nachricht, welches - falls angegeben - den Wert des Elements ObsoleteAfter aus dem SOAP-Header der zugrundeliegenden OSC1.2-Nachricht enthält.				
<b>DeliveryTime</b>	<i>xs:dateTime</i>	1		
Zeitpunkt der Zustellung der Nachricht, welcher - falls angegeben - den Wert des Elements Delivery aus dem SOAP-Header der zugrundeliegenden OSC1.2-Nachricht enthält.				
<b>InitialFetchedTime</b>	<i>xs:dateTime</i>	1		
Zeitpunkt des ersten Abrufs der Nachricht, welcher - falls angegeben - den Wert des Elements InitialFetch aus dem SOAP-Header der zugrundeliegenden OSC1.2-Nachricht enthält.				

### 2.2.1.14 MsgBoxCloseRequest

Globales Element: *MsgBoxCloseRequest*

Element des Typs *MsgBoxCloseRequestType*.

### 2.2.1.15 MsgBoxCloseRequestType

Typ: *MsgBoxCloseRequestType*

Nachricht zum Schließen einer MessageBox nach dem Abruf von Nachrichten durch den Leser, bzw. den Abbruch eines iterativen Nachrichtenabrufes (z.B. durch die WS-Methoden *getMessage* oder *getStatusList*).

Kindelemente von <i>MsgBoxCloseRequestType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>MsgBoxRequestID</b>	<i>xs:anyURI</i>	1		
MsgBoxRequestID der vorangegangenen MsgBoxResponse zur Zuordnung zum ursprünglichen Abhol-Auftrag ( <i>MsgBoxFetchRequest</i> bzw. <i>MsgBoxStatusListRequest</i> ).				
<b>LastMsgReceived</b>	<i>wsa:AttributedURIType</i>	1..n	B.2	121
MessageID der letzten, vom Leser erhaltenen Nachricht als Bestätigung über den Erhalt. Sobald der erfolgreiche Abruf durch den Leser signalisiert wird, setzt der Empfänger/MsgBox den Zeitstempel „Reception“ im Message-MetaData-Header der Nachricht auf seine Systemzeit.				

### 2.2.1.16 MsgBoxFetchRequest

Globales Element: *MsgBoxFetchRequest*

Element des Typs `MsgBoxFetchRequestType`.

### 2.2.1.17 MsgBoxFetchRequestType

Typ: `MsgBoxFetchRequestType`

Dieser Container konkretisiert die Abfrage von Nachrichten aus der `MessageBox` des Lesers.

Dieser Typ ist eine Erweiterung des Basistyps `MsgBoxRequestType` (siehe [Abschnitt 2.2.1.20 auf Seite 67](#)).

Kindelement von <code>MsgBoxFetchRequestType</code>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>MsgPart</b>	<code>xs:NMTOKEN</code>	1		
<p>Der Leser kann über ein NMTOKEN zwischen verschiedenen Formen der Zustellung wählen:</p> <p><i>Envelope</i>: Der vollständige s12:Envelope-Container der gewählten Nachricht wird als Nutzlast der SOAP-Nachricht mitgeliefert. Variante ist die Voreinstellung.</p> <p><i>Header</i>: Der SOAP-Header der Nachricht wird als Nutzlast der Antwort zugestellt.</p> <p><i>Body</i>: Nur die Nutzlast der ursprünglichen SOAP-Nachricht wird als Nutzlast der Antwort zugestellt.</p> <p>Wird dieses Attribut nicht angegeben, ist Body der Default-Wert.</p>				

### 2.2.1.18 MsgBoxGetNextRequest

Globales Element: `MsgBoxGetNextRequest`

Element des Typs `MsgBoxGetNextRequestType`.

### 2.2.1.19 MsgBoxGetNextRequestType

Typ: `MsgBoxGetNextRequestType`

Container für die Abfrage noch nicht erhaltener Nachrichten einer Ergebnismenge in einer `MessageBox`.

Kindelemente von <code>MsgBoxGetNextRequestType</code>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>MsgBoxRequestID</b>	<code>xs:anyURI</code>	1		
<p><code>MsgBoxRequestID</code> der vorangegangenen <code>MsgBoxResponse</code> zur Zuordnung zum ursprünglichen Abhol-Auftrag (<code>MsgBoxFetchRequest</code> bzw. <code>MsgBoxStatusListRequest</code>). Die ID kann an andere Leser weitergereicht werden, falls mehrere Leser auf die Nachrichten zugreifen wollen.</p>				
<b>LastMsgReceived</b>	<code>wsa:AttributedURIType</code>	1..n	<a href="#">B.2</a>	<a href="#">121</a>
<p><code>MessageID</code> der letzten, vom Leser erhaltenen Nachricht als Bestätigung über den Erhalt. Falls das Header-Elemente <code>osci:Reception</code> im SOAP-Header-Block <code>MsgTimeStamps</code> der zugrundeliegenden OSCI 1.2-Nachricht fehlt oder genullt ist, ist hier die Systemzeit einzutragen und in den SOAP-Header-Block in der <code>MessageBox</code> zu übertragen.</p>				

### 2.2.1.20 MsgBoxRequestType

Typ: `MsgBoxRequestType`

Dieses Objekt ermöglicht die Selektion der durch den Leser angeforderten Nachrichten oder Status-Listen.

Kindelement von <i>MsgBoxRequestType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>osci:MsgSelector</b> (ref)		<b>0..1</b>	<a href="#">2.2.1.23</a>	<a href="#">68</a>
Betreff der Nachricht (informativ)				

### 2.2.1.21 MsgBoxResponse

Globales Element: *MsgBoxResponse*

Element des Typs *MsgBoxResponseType*.

### 2.2.1.22 MsgBoxResponseType

Typ: *MsgBoxResponseType*

Dieser Container enthält die Nutzlast einer Antwort auf MessageBox-Anfragen (*MsgBoxFetchRequest* und *MsgBoxStatusListRequest*).

Kindelemente von <i>MsgBoxResponseType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>MsgBoxRequestID</b>	<i>xs:anyURI</i>	<b>1</b>		
UUID-Identifikator für Nachrichten zur Abfrage von Nachrichten und Status-Listen, der von der MessageBox erzeugt wird. Diese ID muss vom Abfragenden für die Abfrage weiterer Nachrichten oder Status-Listen sowie das Beenden der Abfrage genutzt werden.				
Auswahl: <b>NoMessageAvailable</b>	<i>xs:anyURI</i>	<b>1</b>		
Dieses Element ist zu befüllen, falls keine Nachrichten in der MessageBox zur Verfügung stehen. Hier sind URI-Werte gemäß folgender Systematik einzutragen: Keine Treffer für Filterkriterien: <a href="http://www.osci.eu/ws/2008/05/transport/MsgBox/reasons/NoMatch">http://www.osci.eu/ws/2008/05/transport/MsgBox/reasons/NoMatch</a> Fehlerhafte Such- bzw. Filterkriterien: <a href="http://www.osci.eu/ws/2008/05/transport/MsgBox/reasons/SearchArgsInvalid">http://www.osci.eu/ws/2008/05/transport/MsgBox/reasons/SearchArgsInvalid</a> Request-ID nicht bekannt oder verfügbar: <a href="http://www.osci.eu/ws/2008/05/common/urn/MsgBox/reasons/RequestIDInvalid">http://www.osci.eu/ws/2008/05/common/urn/MsgBox/reasons/RequestIDInvalid</a> Weiterer spezifischer Grund: beliebige, vereinbarte weitere URI. Dieser Parameter kann nicht verwendet werden, wenn <i>osci:ItemsPending</i> verwendet wird.				
Auswahl: <b>ItemsPending</b>	<i>xs:nonNegativeInteger</i>	<b>1</b>		
Anzahl der nach der Antwort verbliebenen Nachrichten, ist beim letzten Inkrement einer Ergebnismenge auf Null zu setzen. Dieser Parameter kann nicht verwendet werden, wenn <i>osci:NoMessageAvailable</i> verwendet wird.				

### 2.2.1.23 MsgSelector

Globales Element: *MsgSelector*

Element des Typs *MsgSelectorType*.

### 2.2.1.24 MsgSelectorType

Typ: *MsgSelectorType*

Parameter für die Auswahl/Filterung von Nachrichten oder Listen in einer MessageBox.



Kindelemente von <i>MsgSelectorType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b><i>newEntry</i></b>	<i>xs:boolean</i>	0..1		
Über diesen Wahrheitswert legt der Leser fest, ob nur neue Nachrichten in die Auswahl mit einbezogen werden sollen. Beim Fehlen dieses Attributs wird davon ausgegangen, dass nur neue Nachrichten (ohne Zeitstempel InitialFetch) ausgewählt werden sollen.				
<b><i>wsa:MessageID</i></b> (ref)		0..n	B.2	121
Liste der auszuwählenden Nachrichten anhand deren WS-Adressing MessageID. Das Setzen dieses Selektionskriteriums setzt ggf. andere Kriterien außer Kraft.				
<b><i>wsa:RelatesTo</i></b> (ref)		0..n	B.2	121
Liste der auszuwählenden Nachrichten anhand deren RelatesTo-Eigenschaft, z.B. Vorgängernachrichten.				
<b><i>MsgBoxEntryTimeFrom</i></b>	<i>xs:dateTime</i>	0..1		
Untergrenze für den Eingangszeitpunkt der auszuwählenden Nachrichten. Wenn dieses Element gesetzt ist, werden nur MessageIDs und Metadaten von Nachrichten geliefert, die nach diesem Zeitpunkt empfangen wurden.				
<b><i>MsgBoxEntryTimeTo</i></b>	<i>xs:dateTime</i>	0..1		
Obergrenze für den Eingangszeitpunkt der auszuwählenden Nachrichten. Wenn dieses Element gesetzt ist, werden nur MessageIDs und Metadaten von Nachrichten geliefert, die vor diesem Zeitpunkt empfangen wurden.				
<b><i>Extension</i></b>	<i>xs:anyType</i>	0..1		
Dieses Element ist eine Vorkehrung für die Festlegung weiterer Suchkriterien, zum Beispiel XPath-basierte Abfragen auf Basis der Nachrichten-Metadaten oder Token zur Teilnehmererkennung.				

### 2.2.1.25 *MsgBoxStatusListRequest*

Globales Element: *MsgBoxStatusListRequest*

Element des Typs *MsgBoxStatusListRequestType*.

### 2.2.1.26 *MsgBoxStatusListRequestType*

Typ: *MsgBoxStatusListRequestType*

Container für die Parameter einer Abfrage von Nachrichten-Statusinformationen aus einer MessageBox. Hinweis: Im XTA-Kontext ist für das Attribut ListForm der Wert „MessageMetaData“ anzugeben.

Dieser Typ ist eine Erweiterung des Basistyps *MsgBoxRequestType* (siehe [Abschnitt 2.2.1.20 auf Seite 67](#)).

Kindelemente von <i>MsgBoxStatusListRequestType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b><i>ListForm</i></b>	<i>osci:ListFormEnum</i>	1		
<b><i>maxListItems</i></b>	<i>xs:positiveInteger</i>	1		
Vom Leser vorgebene maximale Anzahl an Nachrichten, die beim Aufruf der Methode getStatusList zurückgeliefert werden sollen. Der Empfänger ist ungeachtet dessen verpflichtet, die vollständige Liste der Ergebnisse vorzuhalten und diese bei nachfolgenden Anfragen mit Bezug auf diese Statusliste bereitzustellen. Diese Obergrenze kann durch den Empfänger auf Ebene der MessageBox-Instanz selbstständig auf jeden Wört größer „0“ (null) beschränkt werden.				

### 2.2.1.27 *MsgStatusList*

Globales Element: *MsgStatusList*

Element des Typs `MsgStatusListType`.

### 2.2.1.28 `MsgStatusListType`

Typ: `MsgStatusListType`

Container für die Inhaltsdaten einer Antwortnachricht beim Aufruf der Methode `getStatusList`. Abhängig vom Attribut `ListForm` der beantworteten Anfrage werden hier entweder Nachrichten-Attribute (`MsgAttributes`) oder -Metadaten (`MessageMetaDate`) der ausgewählten Nachrichten übertragen.

Kindelemente von <code>MsgStatusListType</code>				
Kindelement	Typ	Anz.	Ref.	Seite
<code>MsgAttributes</code>	<code>osci:MsgAttributeListType</code>	0..n	2.2.1.13	65
Liste der Nachrichten-Attribute der ausgewählten Nachrichten (als Antwort auf Anfragen mit <code>ListFormMsgAttributes</code> ).				
<code>oscimeta:MessageMetaData</code> (ref)		0..n	2.2.1.36	73
Liste der Nachrichten-Metadaten der ausgewählten Nachrichten (als Antwort auf Anfragen mit <code>ListFormMessageMetaData</code> ).				

### 2.2.1.29 `TypeOfBusinessScenario`

Globales Element: `TypeOfBusinessScenario`

Element des Typs `TypeOfBusinessScenarioType`.

### 2.2.1.30 `TypeOfBusinessScenarioType`

Typ: `TypeOfBusinessScenarioType`

Geschäftsszenario der Nachricht für Zuordnung bzw gefilterten Abruf durch den Leser. Die Identifikation von Geschäftsszenarien erfolgt über eine von der KoSIT herausgegebene Codeliste, welche XTA-kompatible Fachkontexte enthält. Die Codeliste ist im XRepository unter der URN `urn:de:xta:codelist:business.scenario` verfügbar. Für lokale Geschäftskontexte können Kommunikationspartner auch eigene Codelisten definieren und an dieser Stelle einbinden.

Dieser Typ ist eine Erweiterung des Basistyps `xs:anyURI`.

Kindelement von <code>TypeOfBusinessScenarioType</code>				
Kindelement	Typ	Anz.	Ref.	Seite
<code>IsReferenceParameter</code>	<code>xs:boolean</code>	0..1		
Gemäß WS-Adressing ist hier der Wahrheitswert „true“ zu setzen.				

### 2.2.1.31 `DeliveryAttributesType`

Typ: `DeliveryAttributesType`

Container für Zeitstempel der Transportetappen, erforderliche Servicequalitäten und angeforderte Quitungen.

Kindelemente von <code>DeliveryAttributesType</code>				
Kindelement	Typ	Anz.	Ref.	Seite
<code>Origin</code>	<code>xs:dateTime</code>	0..1		

Kindelemente von <i>DeliveryAttributesType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Zeitstempel für die Erzeugung der Nachricht durch den Autor und deren Übermittlung an den Sender.				
<b>InitialSend</b>	<i>xs:dateTime</i>	0..1		
Zeitstempel für den Zeitpunkt des Versands der Nachricht vom Sender an den Empfänger.				
<b>NotBefore</b>	<i>xs:dateTime</i>	0..1		
Zeitpunkt des frühesten zulässigen Transport der Nachricht vom Sender an den Empfänger, vorgegeben durch den Autor. Fall dieser Zeitpunkt bei der Bearbeitung durch den Sender bereits in der Vergangenheit liegt, erfolgt der Versand sofort.				
<b>ObsoleteAfter</b>	<i>xs:date</i>	0..1		
Datum, ab dem die Nachricht obsolet ist, dieses kann durch Autor festgelegt werden.				
<b>Delivery</b>	<i>xs:dateTime</i>	0..1		
Zeitpunkt des Eingangs bei einem Empfänger oder einer MessageBox.				
<b>InitialFetch</b>	<i>xs:dateTime</i>	0..1		
Zeitpunkt des ersten Abrufes aus einer MessageBox.				
<b>Reception</b>	<i>xs:dateTime</i>	0..1		
Empfangszeitpunkt (lokale Systemzeit) der Nachricht durch den Leser, gesetzt durch den Empfänger bei erfolgreicher Abholung.				
<b>ServiceQuality</b>	<i>oscimeta:NonEmptyStringType</i>	0..1	<a href="#">2.2.1.40</a>	<a href="#">75</a>
Verweis auf Servicequalitäten des Transportauftrags. Wenn vordefinierte ServiceProfile für den entsprechenden Service gegeben sind (beispielsweise per offizieller Definition und Auslieferung durch einen Fachstandard), ist das Element ServiceQuality durch den Autor zur Laufzeit bei Erteilung des Transportauftrags mandatorisch zu verwenden. Es ist in das Element eine Referenz auf die entsprechende ServiceProfil-Instanz einzutragen. Dazu ist der Inhalt des Elements <i>uriDerVersion</i> der Serviceprofil-Instanz anzugeben.  Die Einhaltung der geforderten Servicequalitäten durch alle beteiligten Infrastrukturknoten ist verpflichtend, ebenso die Überprüfung der korrekten Eintragung durch den Sender.  Falls keine Serviceprofil-Instanzen vorgegeben sind, kann das Element genutzt werden, um eine lokal definierte Serviceprofil-Instanz zu referenzieren. Alternativ kann auf die Verwendung des Elements verzichtet werden.				
<b>ReceiptRequests</b>	<i>oscimeta:ReceiptRequestType</i>	0..1	<a href="#">2.2.1.51</a>	<a href="#">78</a>
Container für vom Autor für den Transportauftrag geforderten Quittungen. Für die erforderlichen Quittungen werden hier leere Elemente übertragen.				
<b>NotifyAuthor</b>	<i>oscimeta:NotifyAuthorType</i>	0..1	<a href="#">2.2.1.32</a>	<a href="#">71</a>
Benachrichtigung des Autors über erfolgreiche Zustellung erfordert.				

### 2.2.1.32 NotifyAuthorType

Typ: *NotifyAuthorType*

Angaben zur Autoren-Benachrichtigung nach Zustellung einer Nachricht.

Kindelemente von <i>NotifyAuthorType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>NotifyAuthorURI</b>	<i>xs:anyURI</i>	0..1		
ServiceURI, unter welcher der Autor über die erfolgte Zustellung informiert werden kann.				
<b>NotifyAuthorEnabled</b>	<i>xs:boolean</i>	1		
Benachrichtigung des Autors über erfolgreiche Zustellung erfordert. Der Default-Wert ist FALSCH/FALSE.				

Kindelemente von <i>NotifyAuthorType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Dieses Element/Attribut hat den Wert „false“, sofern kein anderer Wert übermittelt wird (default-Wert).				

### 2.2.1.33 DestinationsType

Typ: *DestinationsType*

Container für Adressaten einer Nachricht.

Kindelemente von <i>DestinationsType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>oscimeta:Reader</b> (ref)		1	2.2.1.50	78
Der Leser der Nachricht. Zusätzlich vom Leser verwendete Zertifikate können in SecurityToken/BinarySecurityToken abgelegt werden.				
<b>oscimeta:OtherDestinations</b> (ref)		0..1	2.2.1.43	75
Weitere Adressaten der Nachricht. Zusätzlich von weiteren Adressaten verwendete Zertifikate können in SecurityToken/BinarySecurityToken abgelegt werden.				

### 2.2.1.34 Identifier

Nachricht: *Identifier*

Container für die Identifikation eines Kommunikationspartners sowie dessen Zertifikate/Tokens.

### 2.2.1.35 KeyCodeType

Typ: *KeyCodeType*

Container für die flexible Übertragung von Schlüssel/Wert-Paaren

Dieser Typ ist eine Einschränkung des Basistyps *Code* (siehe [Abschnitt B.4 auf Seite 121](#)).

Kindelemente von <i>KeyCodeType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>listURI</b>	<i>xs:anyURI</i>	1		
Dokumentation aus <a href="#">B.4</a> : Mit diesem XML-Attribut wird die Kennung der Codeliste übermittelt, in deren Kontext der jeweilige Code zu interpretieren ist. Die Kennung identifiziert die Codeliste, nicht jedoch deren Version eindeutig. Wird bereits im Rahmen des XÖV-Standards eine Kennung vorgegeben (es handelt sich in diesem Fall um einen Code-Typ 1, 2 oder 3) darf auf eine nochmalige Angabe der Kennung bei der Übermittlung eines Codes verzichtet werden. Aus diesem Grund ist das XML-Attribut <i>listURI</i> zunächst als optional deklariert.				
<b>listVersionID</b>	<i>xs:normalizedString</i>	1		
Dokumentation aus <a href="#">B.4</a> : Die konkrete Version der zu nutzenden Codeliste wird mit diesem XML-Attribut übertragen. Analog zum <i>listURI</i> ist die Bestimmung der Version einer Codeliste bei der Übertragung eines Codes zwingend. Die Version kann jedoch ebenfalls bereits im XÖV-Standard festgelegt werden (es handelt sich in diesem Fall um einen Code-Typ 1 oder 2).				
<b>code</b>	<i>xs:token</i>	1		

Kindelemente von <i>KeyCodeType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>name</b>	<i>xs:normalizedString</i>	<b>0..1</b>		
Dokumentation aus <a href="#">B.4</a> : Mit diesem optionalen XML-Element kann die Beschreibung des Codes, wie in der jeweiligen Beschreibungsspalte der Codeliste vorgegeben, übermittelt werden.				

### 2.2.1.36 MessageMetaData

Nachricht: *MessageMetaData*

Container für Metadaten einer Nachricht. Abgebildet werden dabei sowohl Daten zum Transport der Nachricht, als auch über den Inhalt der Nachricht.

Als Transportdaten wird festgehalten, von wem die Nachricht erstellt wurde (Autor), für wen (Leser) und welche Steuerungsinformationen für Transportstationen, Service-Qualitäten und Quittungen vorgegeben wurden.

Als Metainformation zur Nutzlast der Nachricht werden Identifikation und Prozesszusammenhang der Nachricht sowie der fachliche Kontext und Nachrichtentyp eingetragen.

Kindelemente von <i>MessageMetaData</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>TestMsg</b>	<i>xs:boolean</i>	<b>0..1</b>		
Mit diesem Wahrheitswert kann eine Nachricht als Testnachricht ausgewiesen werden. Hier ist der Wert Falsch voreingestellt. Dieses Element/Attribut hat den Wert „false“, sofern kein anderer Wert übermittelt wird (default-Wert).				
<b>DeliveryAttributes</b>	<i>oscimeta:DeliveryAttributesType</i>	<b>1</b>	<a href="#">2.2.1.31</a>	<a href="#">70</a>
Zeitstempel der Transportetappen, erforderliche Servicequalitäten und angeforderte Quittungen				
<b>Originators</b>	<i>oscimeta:OriginatorsType</i>	<b>1</b>	<a href="#">2.2.1.42</a>	<a href="#">75</a>
Autor und Sender der Nachricht, sowie ggf. abweichende Antwort-Adressen.				
<b>Destinations</b>	<i>oscimeta:DestinationsType</i>	<b>1</b>	<a href="#">2.2.1.33</a>	<a href="#">72</a>
Angabe des Lesers sowie weiterer Adressaten.				
<b>MsgIdentification</b>	<i>oscimeta:MsgIdentificationType</i>	<b>1</b>	<a href="#">2.2.1.39</a>	<a href="#">74</a>
Informationen zur Zu- und Einordnung der Nachricht auf Anwendungsebene sowie ihr Bezug zu anderen Nachrichten.				
<b>Qualifier</b>	<i>oscimeta:QualifierType</i>	<b>1</b>	<a href="#">2.2.1.48</a>	<a href="#">77</a>
Container für Metadaten zur Einordnung der Nutzlast auf Anwendungsebene.				
<b>oscimeta:MessageProperties</b> (ref)		<b>0..1</b>	<a href="#">2.2.1.37</a>	<a href="#">73</a>
Erweiterungsobjekt, um zusätzliche Wertepaare in den Nachrichten-Metadaten zu transportieren. Dieses Objekt kann ggf. durch Messaging-Knoten bei Bedarf fortgeschrieben werden.				
<b>MsgSize</b>	<i>xs:positiveInteger</i>	<b>1</b>		
Hier ist die Größe des Objekts Message (globales Element im Body) in Bytes zu erfassen.				

### 2.2.1.37 MessageProperties

Globales Element: *MessageProperties*

Element des Typs `MessagePropertiesType`.

Kindelement von <i>MessageProperties</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>Property</b>	<i>oscimeta:PropertyType</i>	1..n	<a href="#">2.2.1.47</a>	<a href="#">76</a>
Liste von Schlüssel-Wert-Paaren.				

### 2.2.1.38 MessagePropertiesType

Typ: *MessagePropertiesType*

Erweiterungsobjekt, um zusätzliche Wertepaare in den Nachrichten-Metadaten zu transportieren.

Kindelement von <i>MessagePropertiesType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>Property</b>	<i>oscimeta:PropertyType</i>	1..n	<a href="#">2.2.1.47</a>	<a href="#">76</a>
Liste von Schlüssel-Wert-Paaren.				

### 2.2.1.39 MsgIdentificationType

Typ: *MsgIdentificationType*

Informationen zur Zu- und Einordnung der Nachricht auf Anwendungsebene sowie ihr Bezug zu anderen Nachrichten.

Kindelemente von <i>MsgIdentificationType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>wsa:MessageID</b> (ref)		1	<a href="#">B.2</a>	<a href="#">121</a>
Eindeutige, zeitlich und räumlich unbegrenzt gültige Identifikationsnummer eines Transportauftrags, die normalerweise von einem Fachverfahren (Autor, Leser) an ein Transportverfahren (Sender, Empfänger) übergeben wird. Sie entspricht der Syntax einer URN und setzt sich aus einem Präfix des Ausstellers sowie einer UUID gemäß IETF RFC4122 zusammen, z.B.: „urn:de:xta:messageid:clearingstelleXY_xta_01:000ca2fe-f4e1-45c2-8233-3a0eb760bd16“				
Für die Zwecke der Identifikation, Protokollierung und Nachverfolgung muss die MessageID sowohl beim Sender als auch beim Empfänger eindeutig sein, und außerdem die ausstellende Softwareinstanz anhand des Präfixes eindeutig erkennen lassen.				
<b>In-Reply-To</b>	<i>wsa:AttributedURIType</i>	0..n	<a href="#">B.2</a>	<a href="#">121</a>
Verweis/Bezug auf weitere Nachrichten-ID(s) auf Anwendungsebene.				
<b>ProcessRef</b>		0..1		
Zuordnung der Nachricht zu einem Geschäftsprozess, hier kann die ID einer XÖV-Nachricht eingetragen werden. Die konkrete Verwendung des Feldes wird durch das ServiceProfil festgelegt.				
<b>Requester</b>	<i>oscimeta:ProcessIdentifierType</i>	0..1	<a href="#">2.2.1.46</a>	<a href="#">76</a>
Zuordnung zu einem Geschäftsprozess beim Autor.				
<b>Responder</b>	<i>oscimeta:ProcessIdentifierType</i>	0..1	<a href="#">2.2.1.46</a>	<a href="#">76</a>
Zuordnung zu einem Geschäftsprozess beim Leser.				

### 2.2.1.40 NonEmptyStringType

Typ: *NonEmptyStringType*

Eine Zeichenkette mit der Mindestlänge 1. Dies schließt nicht Leerzeichen o.ä. aus, aber den Leerstring.

Dieser Typ ist eine Einschränkung des Basistyps *xs:string*.

Beschränkungen (XML-Schema-Facetten): *minLength:1*

### 2.2.1.41 NonEmptyURIType

Typ: *NonEmptyURIType*

Dieser Typ ist eine Einschränkung des Basistyps *xs:anyURI*.

Beschränkungen (XML-Schema-Facetten): *minLength:1*

### 2.2.1.42 OriginatorsType

Typ: *OriginatorsType*

Autor und Sender der Nachricht, sowie ggf. abweichende Antwort-Adressen.

Kindelemente von <i>OriginatorsType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>Author</b>	<i>oscimeta:PartyType</i>	1	<a href="#">2.2.1.44</a>	<a href="#">75</a>
Der Autor der Nachricht. Zusätzlich vom Autor verwendete Zertifikate können in SecurityToken/BinarySecurityToken abgelegt werden.				
<b>Sender</b>	<i>oscimeta:PartyType</i>	0..1	<a href="#">2.2.1.44</a>	<a href="#">75</a>
Der Sender der Nachricht, z.B. XTA-WS oder OSCI-Gateway. Zusätzlich vom Sender verwendete Zertifikate können in SecurityToken/BinarySecurityToken abgelegt werden.				
<b>ReplyTo</b>	<i>oscimeta:PartyType</i>	0..1	<a href="#">2.2.1.44</a>	<a href="#">75</a>
Der Empfänger der Antwort, falls vom im Element Author abweichend. Zusätzlich für abweichende Empfänger verwendete Zertifikate können in SecurityToken/BinarySecurityToken abgelegt werden.				

### 2.2.1.43 OtherDestinations

Globales Element: *OtherDestinations*

Container für weitere Adressaten einer Nachricht.

Kindelemente von <i>OtherDestinations</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>OtherReaders</b>	<i>oscimeta:PartyIdentifierType</i>	1..n	<a href="#">2.2.1.45</a>	<a href="#">76</a>
Weitere Empfänger der Nachricht.				
<b>CcReaders</b>	<i>oscimeta:PartyType</i>	0..n	<a href="#">2.2.1.44</a>	<a href="#">75</a>
Nachgeordnete Adressaten der Nachricht, analog zu cc/carbon copy in Emails.				

### 2.2.1.44 PartyType

Typ: *PartyType*

Container für die Identifikation eines Kommunikationspartners sowie dessen Zertifikate/Tokens.

Kindelemente von <i>PartyType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>Identifizier</b>	<i>oscimeta:PartyIdentifizierType</i>	1	<a href="#">2.2.1.45</a>	76
Identifikation des Kommunikationspartners.				
<b>oscimeta:SecurityToken</b> (ref)		0..n	<a href="#">2.2.1.52</a>	78
Zertifikate/Tokens des Kommunikationspartners, z.B. X.509.				

### 2.2.1.45 PartyIdentifizierType

Typ: *PartyIdentifizierType*

Container zur eindeutigen Identifikation eines Kommunikationspartners anhand einer Bezeichnung sowie mehrerer Attribute zur Zuordnung eines Kontextes für diese Bezeichnung.

Dieser Typ ist eine Erweiterung des Basistyps *NonEmptyStringType* (siehe [Abschnitt 2.2.1.40 auf Seite 75](#)).

Kindelemente von <i>PartyIdentifizierType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>category</b>	<i>oscimeta:NonEmptyStringType</i>	0..1	<a href="#">2.2.1.40</a>	75
Rolle des identifizierten Kommunikationspartners innerhalb der Domäne (z.B. "Käufer", "Meldebehörde", "Standesamt", etc.). Zur Unterstützung stellt die KoSIT die Codeliste <i>XOEV-Category of Party</i> bereit, die mit der URN „urn:de:xta:codeliste:category.of.party“ im XRepository abrufbar ist.				
<b>name</b>	<i>oscimeta:NonEmptyStringType</i>	0..1	<a href="#">2.2.1.40</a>	75
Optionaler menschenlesbarer Name zur Darstellung in User agents (bekannt z.B. aus eMail).				
<b>type</b>	<i>oscimeta:NonEmptyStringType</i>	1	<a href="#">2.2.1.40</a>	75
Ausweisung des Schemas zur Interpretation des PartyIdentifizierers/Bezeichners, z.B. DVDV oder S.A.F.E. Aus dem jeweiligen Verzeichnisdienst können die Verbindungsparameter entnommen werden. Zur Unterstützung stellt die KoSIT die Codeliste <i>Type of Party Identifizier</i> bereit, die mit der URN „urn:de:xta:codeliste:type.of.party.identifizier“ im XRepository abrufbar ist.				

### 2.2.1.46 ProcessIdentifizierType

Typ: *ProcessIdentifizierType*

Eindeutige Zuordnung zu einem Geschäftsprozess.

Kindelement von <i>ProcessIdentifizierType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>ProccesName</b>	<i>oscimeta:NonEmptyStringType</i>	0..1	<a href="#">2.2.1.40</a>	75
Bezeichnung des Geschäftsprozesses, z.B. "Bestellung", als Kontext einer Vorgangsnummer innerhalb des hier benannten Prozesses.				

### 2.2.1.47 PropertyType

Typ: *PropertyType*



Generischer Typ für Schlüssel-Wert-Paare innerhalb eines definierten Namensraumes oder einer Code-liste.

Kindelemente von <i>PropertyType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>Key</b>	<i>oscimeta:KeyCodeType</i>	1	<a href="#">2.2.1.35</a>	<a href="#">72</a>
Schlüssel des Schlüssel-Wert-Paares.				
<b>Value</b>	<i>oscimeta:NonEmptyStringType</i>	1	<a href="#">2.2.1.40</a>	<a href="#">75</a>
Wert des Schlüssel-Wert-Paares.				

### 2.2.1.48 QualifierType

Typ: *QualifierType*

Container zur Einordnung der Nutzlast mit Anwendungsfall-übergreifend genutzten Metadaten.

Kindelemente von <i>QualifierType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>Subject</b>	<i>xs:string</i>	0..1		
Betreff der Nachricht (informativ)				
<b>Service</b>	<i>xs:anyURI</i>	1		
Spezifischer Service im Kontext eines bestimmten Geschäftsfalls, einzutragen als URI. Im XÖV-Kontext gewöhnlich die aus DVDV bezogene Dienste-URI.				
<b>BusinessScenario</b>		1		
Angabe eines entweder über Codelisten oder freitextlich definierten Geschäftsszenarios. Die KoSIT stellt hierzu unter der URN „urn:de:xta:codeliste:business.scenario“ eine Codeliste im XRepository bereit. Alternativ können XTA-Kommunikationspartner beschließen und an dieser Stelle mitsamt Codelisten-URI und Versionsnummer eintragen.				
Auswahl: <b>Defined</b>	<i>oscimeta:KeyCodeType</i>	1	<a href="#">2.2.1.35</a>	<a href="#">72</a>
Definierte Angabe eines Geschäftsszenarios zum Abgleich mit einer Codeliste.				
Auswahl: <b>Undefined</b>	<i>xs:normalizedString</i>	1		
Freitextliche Angabe eines nicht in einer Codeliste definierten Geschäftsszenarios.				
<b>MessageType</b>	<i>oscimeta:MessageType</i>	1	<a href="#">2.2.1.49</a>	<a href="#">77</a>
Nachrichtentyp, wie innerhalb des Geschäftsszenarios (z.B. Fachstandard) definiert.				

### 2.2.1.49 MessageType

Typ: *MessageType*

Art der Nachricht bzw. der Nutzlast im Kontext des Geschäftsszenarios.

Dieser Typ ist eine Erweiterung des Basistyps *xs:string*.

Kindelement von <i>MessageType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>payloadSchema</b>	<i>oscimeta:NonEmptyURIType</i>	1	<a href="#">2.2.1.41</a>	<a href="#">75</a>
Angabe des für die Nutzlast der Nachricht verwendeten XML Schemas als URI.				

### 2.2.1.50 Reader

Nachricht: *Reader*

Der Leser der Nachricht.

### 2.2.1.51 ReceiptRequestType

Typ: *ReceiptRequestType*

Container für die Anforderung von Quittungen durch den Autor.

Kindelemente von <i>ReceiptRequestType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>Submission</b>	<i>xs:anyType</i>	0..1		
Anforderung einer Quittung über den Erhalt der Nachricht durch den Sender.				
<b>Relay</b>	<i>xs:anyType</i>	0..1		
Anforderung einer Quittung über die Weiterleitung der Nachricht durch Infrastrukturknoten.				
<b>Delivery</b>	<i>xs:anyType</i>	0..1		
Anforderung einer Quittung über den Erhalt der Nachricht durch den Sender.				
<b>Fetch</b>	<i>xs:anyType</i>	0..1		
Anforderung einer Quittung über den Abruf der Nachricht durch den Leser aus dessen MessageBox.				
<b>Reception</b>	<i>xs:anyType</i>	0..1		
Anforderung einer Quittung über den Erhalt der Nachricht durch den Leser.				
<b>ReceiptTo</b>	<i>wsa:EndpointReferenceType</i>	0..1	B.2	121
Hier kann der Autor einen separaten Knoten (Endpoint Reference) angeben, der die Quittungen für den Transportauftrag erhalten soll. Falls hier kein Wert angegeben wird, wird der Wert aus dem Element <i>wsa:From</i> aus dem Header-Block übernommen.				

### 2.2.1.52 SecurityToken

Globales Element: *SecurityToken*

Container für die Übermittlung von Security Tokens sowie deren Verwendungszweck und Verortung in der Nachricht.

Kindelemente von <i>SecurityToken</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>payloadRef</b>	<i>xs:IDREF</i>	0..1		
Dieses Attribut wird in XTA nicht verwendet.				
<b>usage</b>	<i>oscimeta:SecurityTokenEnum</i>	1		
Auswahl: <b>wsse:BinarySecurityToken</b> (ref)		1	B.1	121
Als base64binary kodiertes Security Token (z.B. X.509).				
Auswahl: <b>wsse:SecurityTokenReference</b> (ref)		1	B.1	121
Referenz auf Security Token: Flexibler Adressierungsmechanismus kryptographischer Bestandteile in der Nachricht.				

Kindelemente von <i>SecurityToken</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Auswahl: <b>wsse:UsernameToken</b> (ref)		1	B.1	121
Container für die Angabe eines Security Tokens mit Nutzernamen.				

## 2.2.2 Globale Elemente der Informationsobjekte des XTA Webservice

Die in diesem Abschnitt beschriebenen Objekte sind in den Methodenaufrufen des XTA Webservice eingebunden. Sie sind zusammengesetzt teils aus in XTA definierten Bestandteilen wie beschrieben in [Section 2.2.1 on page 60](#), teils aus Typen der externen Standards, die aufgeführt sind in [Anhang B, Eingebundene externe Modelle/Standards](#)).

### 2.2.2.1 Message

Nachricht: *Message*

Das Element Message nimmt den zu transportierenden oder abzuliefernden Inhalt auf, z.B. eine XÖV-Nachricht mit ihren Anlagen.

Kindelement von <i>Message</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>MessagePart</b>	<i>xta:ContentType</i>	1..n	2.2.1.12	65
Enthält den base64-kodierten Inhalt, der zwischen WebService-Client und XTA-Server transportiert wird. Die Attribute sind der MIME-Spezifikation (RFC 2183) entnommen. In die Attribute wird je nach Kontext Metainformation zur Nachricht eingetragen. Die XÖV-Nachricht und ihre Anlagen können als jeweils ein MessagePart mit entsprechend unterschiedlichen Attributen übermittelt werden.				

### 2.2.2.2 LookupServiceRequest

Nachricht: *LookupServiceRequest*

Dies ist eine Liste von Dienstanfragen.

Jede Anfrage dient dazu, zu ermitteln, ob der Dienst von diesem Anbieter angeboten wird, und über welche technischen Parameter er angesprochen werden kann.

Kindelement von <i>LookupServiceRequest</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>LookupServiceRequestList</b>		1..n		
Dies ist die Struktur für eine Liste von Dienstanfragen.				
<b>LookupService</b>	<i>xta:LookupServiceType</i>	1	2.2.1.8	63
Dies ist eine Service-Anfrage. Sie enthält Daten zum potentiellen Diensteanbieter (Leser) und dem Dienst, der angefragt werden soll. Diese Anfrage dient dazu, zu ermitteln, ob der Dienst von diesem Anbieter angeboten wird, und über welche technischen Parameter er angesprochen werden kann.				

### 2.2.2.3 LookupServiceResponse

Nachricht: *LookupServiceResponse*

Dies ist das Ergebnis zu einer Liste von Dienstanfragen, also eine Liste von Dienstanfrageergebnissen. Die Anfrage wird jeweils zitiert und das zugehörige Ergebnis ausgegeben.

Kindelement von <i>LookupServiceResponse</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>LookupServiceResultList</b>		1		
Die Struktur einer Liste von Dienstanfrageergebnissen.				
<b>LookupServiceResult</b>	<i>xta:LookupServiceResultType</i>	1..n	2.2.1.9	63
Dies ist die Struktur der Liste von Ergebnissen zur Liste von Dienstanfragen.				

### 2.2.2.4 TransportReport

Nachricht: *TransportReport*

Der TransportReport ist die Struktur des durch XTA standardisierten Transportprotokolls. Neben den übermittelten Nachrichten ruft das Fachverfahren (in den Rollen Autor und Leser) über den Webservice-Client Zusatzinformationen über den Transportauftrag und die Transportereignisse vom XTA-WS ab.

Um Autor und Leser die Möglichkeit zu geben, die Abarbeitung ihrer Transportaufträge zu überwachen, erstellen Sender und Empfänger Transportprotokolle, die in einer XML-Struktur des Typs TransportReport dargestellt und für Abruf und Auswertung bereit liegen.

Die Datenstruktur aggregiert die Information zum erteilten Transportauftrag, zum Verlauf des sich anschließenden Transports einschließlich Zertifikatsüberprüfungen mit Ergebnissen.

Kindelemente von <i>TransportReport</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>ReportTime</b>	<i>xs:dateTime</i>	1		
Zeitpunkt der letzten Aktualisierung des Protokolls. Ist bei Fortschreibung des Protokolls zu überschreiben.				
<b>XTAServerIdentity</b>	<i>xs:token</i>	1		
Hier protokolliert der den TransportReport erstellende Prozess seine Identität als Software-Instanz, indem er z.B. die Server-IP-Adresse oder die URI seines XTA-WS einträgt.				
<b>MessageStatus</b>	<i>xta:MessageStatusType</i>	1	2.2.1.7	62
Enthält Information über den Verlauf des Transports. Es werden hier Listen mit aufgetretenen Fehler-, Warnungs- und Informationsmeldungen geführt. Außerdem ist nach Schließung des Transportauftrags im Feld Status eine "Schnell-Info" verfügbar.				
<b>oscimeta:MessageMetaData</b> (ref)		1	2.2.1.36	73
Dieser Container umfasst alle Daten des Transportauftrags, auf dessen Ausführung sich der TransportReport bezieht. Zu den Informationen gehören die Identifizierung von Absender und (einem oder mehreren) Empfängern, Metainformation zu Inhalt und Identität der zu transportierenden Fachnachricht (Payload) sowie weitere Attribute, die Auslieferung, Quittungen und Servicequalität betreffen.				
Weitere Informationen zu diesem Objekt sind in <a href="#">Abschnitt 2.1.1.2.5.1 auf Seite 45</a> zu finden.				
<b>AdditionalReports</b>	<i>xta:AdditionalReportListType</i>	0..1	2.2.1.4	61
Hier sind weitere Prüfberichte abgelegt bzw. abzulegen, welche das XTA-Protokoll ( <i>TransportReport</i> ) ergänzen sollen.				
<b>ds:Signature</b> (ref)		0..1	B.3	121
Falls der TransportReport signiert ist, findet sich hier die Signatur.				

### 2.2.2.5 ServiceReport

Nachricht: *ServiceReport*

Der Service Report ist die Struktur des durch XTA standardisierten Berichts über die technische Abarbeitung eines Transportauftrages. Er wird von Autoren und Lesern dazu verwendet, die Einhaltung des Transportauftrags zu kontrollieren. Entsprechend können Sender und Empfänger mittels eines Service Report nachweisen, dass der Transportauftrag eingehalten wurde. Sender erstellen einen Service Report für Autoren, Empfänger einen Service Report für Leser. Stehen für einen Transportauftrag sowohl der Service Report des Senders als auch des Empfängers zur Verfügung, kann die Einhaltung des Transportauftrags vom Versand bis zum Empfang nachgewiesen bzw. kontrolliert werden.

Mit dem Service Report soll insbesondere auch die rechtskonforme Funktionsweise der den Transport durchführenden Infrastruktur nachvollzogen werden können. Der Service Report ermöglicht Aussagen über die Einhaltung gesetzlicher Anforderungen und die Erfüllung von Vorgaben aus Service Profilen. Er dient ferner als zusätzliches Mittel zur Fehlerverfolgung in Transportprozessen.

Die Service Report-Ereignisse werden vom Sender und Empfänger während der Abarbeitung von Transportaufträgen protokolliert. Aus diesen Ereignissen erstellen Sender oder Empfänger den Service Report, wenn dieser durch den Autor oder den Leser einer Fachnachricht angefordert wird. Autor oder Leser bestimmen beim Abruf, zu welchen Transportaufträgen Service Report-Ereignisse zu einem Service Report zusammengestellt werden sollen. Dazu können die Selektionskriterien gemäß ReportSelector ([Abschnitt 2.2.2.7 auf Seite 82](#)) verwendet werden. Der auf dieser Basis erstellte Service Report enthält immer alle Ereignisse zu den so bestimmten Transportaufträgen. Autoren und Leser fordern einen Service Report über eine entsprechende XTA-Webservice-Methode an.

Kindelemente von <i>ServiceReport</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>ReportTime</b>	<i>xs:dateTime</i>	1		
Zeitpunkt der letzten Aktualisierung des Protokolls. Ist bei Fortschreibung des Protokolls zu überschreiben.				
<b>XTAServerIdentity</b>	<i>xs:token</i>	1		
Hier protokolliert der den TransportReport erstellende Prozess seine Identität als Software-Instanz, indem er z.B. die Server-IP-Adresse oder die URI seines XTA-WS einträgt.				
<b>xta:ReportSelector</b> (ref)		1	<a href="#">2.2.2.7</a>	<a href="#">82</a>
Parameter für die Auswahl/Filterung von Einträgen in einem Service Report.				
<b>ds:Signature</b> (ref)		0..1	<a href="#">B.3</a>	<a href="#">121</a>
Falls der TransportReport signiert ist, findet sich hier die Signatur.				
<b>EreignisList</b>	<i>xtaevt:EreignisListType</i>	0..1	<a href="#">1.2.1.2</a>	<a href="#">24</a>

### 2.2.2.6 MessageResponse

Nachricht: *MessageResponse*

Dieses Element enthält die Antwortnachricht in der synchronen Kommunikation zwischen Empfänger und Leser und entsprechend analog zwischen Sender an den Autor. Ist eine Fachantwort vom Leser gefordert, ist diese im Kindelement *Message* enthalten. Falls keine Fachantwort vom Leser gefordert wird, ist das Kindelement *Message* nicht in diesem Element enthalten.

Kindelement von <i>MessageResponse</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>xta:Message</b> (ref)		0..1	<a href="#">2.2.2.1</a>	<a href="#">79</a>

Kindelement von <i>MessageResponse</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Das Element Message nimmt den zu transportierenden oder abzuliefernden Inhalt auf, z.B. eine XÖV-Nachricht mit ihren Anlagen.				

### 2.2.2.7 ReportSelector

Globales Element: *ReportSelector*

Parameter für die Auswahl/Filterung von Einträgen in einem Service Report.

Kindelemente von <i>ReportSelector</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>AuthorIdentifier</b>	<i>oscimeta:PartyType</i>	0..1	2.2.1.44	75
Eindeutiges Identifikationsmerkmal des Autors. Wird kein Identifikationsmerkmal angegeben, werden alle Einträge zurückgegeben, die dem im Methodenaufruf verwendeten Clientzertifikat zugeordnet werden können.				
<b>ReaderIdentifier</b>	<i>oscimeta:PartyType</i>	0..1	2.2.1.44	75
Eindeutiges Identifikationsmerkmal des Lesers.				
<b>EntryTimeFrom</b>	<i>xs:dateTime</i>	0..1		
Untergrenze für den Eingangszeitpunkt der auszuwählenden Nachrichten. Der Eingangszeitpunkt ist größer oder gleich dieser Untergrenze.				
<b>EntryTimeTo</b>	<i>xs:dateTime</i>	0..1		
Obergrenze für den Eingangszeitpunkt der auszuwählenden Nachrichten. Der Eingangszeitpunkt ist echt kleiner als diese Obergrenze.				
<b>messageID</b>		0..1		
Eine MessageID stellt eine eindeutige Identifizierung eines bestimmten Transportauftrages dar (Transportauftragsnummer). Damit entspricht sie nicht anderen MessageID's, wie z. B. der MessageID aus OSCI-Transport 1.2 oder der ID des XÖV-Dokuments.				
Die MessageID ist eine zeitlich und räumlich unbegrenzt gültige eindeutige Identifikationsnummer (z.B. entsprechend WS-Adressing). Sie gewährleistet im XTA-WS-Kontext die eindeutige Identifikation einer Nachricht und wird normalerweise vom Fachverfahren (Autor/Leser) an das Transportverfahren (Sender/Empfänger) übergeben. Für jeden Transportauftrag wird eine neue MessageID erzeugt, auch wenn dieselbe fachliche Nachricht übertragen werden soll. Muss z. B. eine XÖV Nachricht auf Grund von aufgetretenen Fehlern erneut verschickt werden, dann bleibt die ID der XÖV Nachricht gleich, während die MessageID des Transportauftrags neu berechnet wird.				
Verwendung findet die MessageID bei der Protokollierung und den Statusabfragen. Sie ist aber auch bei der Kommunikation im Fehlerfall sehr wichtig, weil sie die Identifizierung der Nachricht über die gesamte Transportstrecke hinweg erleichtert. Das Transportverfahren kann die benötigten MessageID's zur Verfügung stellen, Fachverfahren können sie auch selber berechnen.				
Diese MessageID gilt sowohl für die Nachricht, wie auch für das dazugehörige Transportprotokoll. Die MessageID hat das Format einer URN. Sie sollte folgende Anforderungen erfüllen:				
<ul style="list-style-type: none"> <li>• Die MessageID muss beim Sender eindeutig sein. Sonst kann das Fachverfahren nicht den zugehörigen TransportReport abholen.</li> <li>• Die MessageID muss beim Empfänger eindeutig sein. Sonst kann das Fachverfahren nicht den zugehörigen TransportReport abholen.</li> <li>• Die MessageID soll erkennen lassen, wer die MessageID erstellt hat. So kann in Problemfällen der Ersteller der Nachricht leichter ermittelt werden.</li> <li>• Die MessageID soll beim Empfänger dieselbe sein wie beim Sender. Nur so kann die Nachricht über den gesamten Transporthweg eindeutig identifiziert werden.</li> </ul>				

Kindelemente von <i>ReportSelector</i>				
Kindelement	Typ	Anz.	Ref.	Seite
Für die Erfüllung der ersten beiden Anforderungen reicht es aus, eine UUID als Identifikator zu verwenden. Wegen der dritten Anforderung wird ein Präfix hinzugefügt, der eindeutig die Softwareinstanz identifiziert, die die MessageID erstellt hat. Somit ergibt sich der folgende Aufbau der MessageID:				
<ul style="list-style-type: none"> <li>• Präfix: Angabe über die Softwareinstanz, die die MessageID erstellt, z.B. ClearingstelleXY_Xta_01 oder ClearingstelleXY_SAP_15.</li> <li>• Identifikator: Dieser muss aus einer UUID generiert sein (siehe RFC4122, z.B. 000ca2fe-f4e1-45c2-8233-3a0eb760bd16)</li> </ul>				
Allgemeinen Aufbau als URN: urn:xoevmmessageid:<Präfix>:<Identifikator>				
Beispiel: urn: xoevmmessageid:Dataport_Xta_01:000ca2fe-f4e1-45c2-8233-3a0eb760bd16				

## 2.3 XTA Webservice SOAP Exceptions

Fehler des XTA Webservice werden als SOAP 1.2 Exceptions geworfen. Es sind eine Reihe solcher Exceptions für den XTA Webservice definiert. Sie werden in [Abschnitt 2.3.1 auf Seite 83](#) aufgezählt und dokumentiert. Sie sind alle von derselben Basisklasse abgeleitet, die in [Abschnitt 2.3.2 auf Seite 85](#) erläutert wird.

### 2.3.1 Die Exceptions des XTA Webservice

Die in diesem Abschnitt beschriebenen Objekte werden in Fehlersituationen als Reaktion auf XTA Webservice-Aufrufe übergeben.

Sie dienen dazu, Informationen innerhalb einer transportierten SOAP-Exception zu verpacken (vgl. [Abschnitt 2.3.3 auf Seite 86](#)).

Die Information ist in einem Fehlercode enthalten, den das Objekt überbringt. Der Name des Objekts stellt einen groben Hinweis auf die Art eines aufgetretenen Fehlers dar, der Fehlercode gibt eine differenziertere Information.

**Umsetzungshinweis für die Implementierung von Webservice-Clients:** Bei fast allen Exceptions macht ein automatisierter erneuter Aufruf der entsprechenden Methode keinen Sinn. Es ist erst eine Fehlerklärung erforderlich. Eine Ausnahme bildet nur die Exception `XTAWSTechnicalProblemException` (vgl. [Abschnitt 2.3.1.11 auf Seite 85](#)). Tritt diese Exception auf, ist typischerweise ein erneuter, automatisierter Aufruf der entsprechenden Methode (im Fall eines Nachrichten-Versands mit neuer MessageID) sinnvoll.

#### 2.3.1.1 `PermissionDeniedException`

Nachricht: *PermissionDeniedException*

Diese Exception wird geworfen, wenn der Account gesperrt oder nicht vorhanden ist.

#### 2.3.1.2 `ParameterIsNotValidException`

Nachricht: *ParameterIsNotValidException*

Diese Fehlermeldung wird geworfen, wenn ein Parameter nicht korrekt an die Methode übergeben wurde.

#### 2.3.1.3 `ParameterNotSupportedException`

Nachricht: *ParameterNotSupportedException*

ParameterNotSupportedException wird geworfen, wenn eine Methode mit spezifikationsgemäßen, aber von der Implementation nicht unterstützen Parametern aufgerufen wird.

#### **2.3.1.4 ExternalServiceUnavailableException**

Nachricht: *ExternalServiceUnavailableException*

ExternalServiceUnavailableException signalisiert einen Fehler aufgrund von Nichterreichbarkeit des Dienstes.

#### **2.3.1.5 InvalidMessageIDException**

Nachricht: *InvalidMessageIDException*

Diese Exception wird geworfen, wenn in einem gegebenen Kontext die anhand der ID bezeichnete Nachricht nicht bekannt ist, also beispielsweise nicht geliefert werden kann.

#### **2.3.1.6 MessageSchemaViolationException**

Nachricht: *MessageSchemaViolationException*

Diese Exception wird geworfen, wenn eine Fachnachricht nicht der jeweiligen Schema-Definition entspricht.

#### **2.3.1.7 SyncAsyncException**

Nachricht: *SyncAsyncException*

Diese Exception wird geworfen falls dem XTA-Webservice

- eine Nachricht, die nur für die synchrone Weiterleitung gültig ist, für die asynchrone Weiterleitung übergeben wurde oder
- eine Nachricht für die synchrone Weiterleitung übergeben wurde, die nur für die asynchrone Weiterleitung gültig ist.

#### **2.3.1.8 MessageVirusDetectionException**

Nachricht: *MessageVirusDetectionException*

Diese Exception wird geworfen, wenn schadhafter Code in einem der übergebenen Container ermittelt wurde.

#### **2.3.1.9 CancelDeniedException**

Nachricht: *CancelDeniedException*

Diese Exception wird geworfen, falls die Methode `cancelMessage` aufgerufen wurde, aber der Transportauftrag aus einem der folgenden Gründe nicht zurückgezogen werden kann:

- Der bei Erteilung des Transportauftrags über den Schalter *NotBefore* gesetzte Termin ist erreicht.
- Der Schalter *NotBefore* wurde bei Erteilung des Transportauftrags nicht gesetzt.

#### **2.3.1.10 NotImplementedException**

Nachricht: *NotImplementedException*

Der technische Fehler (SoapFault) `NotImplementedException` wird zurückgegeben, wenn eine nicht implementierte Methode aufgerufen wird.



### 2.3.1.11 XTAWSTechnicalProblemException

Nachricht: *XTAWSTechnicalProblemException*

Diese Exception wird allgemein geworfen, wenn ein technisches Problem im XTA-WS aufgetreten ist. Sie kann z. B. durch ein Problem beim Zugriff auf die interne Datenbank des XTA-Servers ausgelöst worden sein.

### 2.3.1.12 NotificationForFetchedMessageException

Nachricht: *NotificationForFetchedMessageException*

Der technische Fehler (SoapFault) *NotificationForFetchedMessageException* wird zurückgegeben, wenn der Empfänger dem Leser *MessageMetaData* für eine Nachricht übergibt, die der Leser bereits erfolgreich abgeholt hat. Der Fehler kann pro Nachricht mehrfach auftreten, z.B. aufgrund von Latenzen. Leser und Empfänger müssen festlegen, wie damit umgegangen wird, wenn der Fehler mehrfach auftritt.

### 2.3.1.13 TooManyResultsException

Nachricht: *TooManyResultsException*

Der technische Fehler (SoapFault) *TooManyResultsException* wird zurückgegeben, wenn die Ergebnismenge für die gewählten Parameter eines Abrufs zu groß ist. Der Grenzwert der Ergebnismenge wird außerhalb der XTA-Spezifikation, also beispielsweise vom Betreiber des XTA-Server, festgelegt.

## 2.3.2 Struktur von Exception und Fehlernummer

Die in [Abschnitt 2.3.1 auf Seite 83](#) dargestellten Exceptions sind alle von einer gemeinsamen Basis-Klasse abgeleitet. Sie kapselt die Fehlernummer, die zur näheren Beschreibung einer Fehlersituation verwendet wird.

### 2.3.2.1 ExceptionType

Typ: *ExceptionType*

Dieser Datentyp legt die grundlegende Struktur einer Exception im Rahmen des XTA Webservice fest. Sie kapselt Information zu Identität und Bedeutung eines aufgetretenen Fehlers.

Kindelement von <i>ExceptionType</i>				
Kindelement	Typ	Anz.	Ref.	Seite
<b>errorCode</b>	<i>xta:Code.Fehlernummer</i>	1	<a href="#">2.3.2.2</a>	<a href="#">85</a>
In diesem Element werden Fehlernummer und Fehlertext übermittelt, die einen Fehler näher beschreiben (gemäß verlinkter Codeliste).				
In das Unterelement <i>code</i> ist die Fehlernummer einzutragen, ins Unterelement <i>name</i> die entsprechende textuelle Repräsentation.				

### 2.3.2.2 Code.Fehlernummer

Codelisten	
-beschreibung	Diese Codeliste gibt eine Übersicht über die in XTA-WS zu verwendenden Fehlernummern (Error-Codes) und ordnet sie den Exceptions zu, in deren Kontext sie auftreten können.
-nutzung	Typ: 2, Inhalte der Codeliste siehe <a href="#">Seite 114</a>
-kennung	urn:de:xta:webservice:codeliste:fehlnummer

Codelisten	
-version	1.0

### 2.3.3 Exceptions als XML-Instanzen

Ein Beispiel (Abbildung 2.2, „Exception als SOAP-Text“) soll verdeutlichen, wie eine in XTA Webservice definiertes Exception-Objekt im Rahmen einer transportierten SOAP-Exception eingesetzt wird.

Abbildung 2.2. Exception als SOAP-Text

```

<?xml version="1.0" encoding="UTF-8" ?>
- <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:a="http://www.w3.org/2005/08/addressing">
- <s:Header>
  <a:Action
    s:mustUnderstand="1">http://www.osci.eu/ws/2008/05/transport/urn/messageTypes/MsgBoxFetchRequest</a:Action>
  <a:RelatesTo>urn:uuid:49cfe3ef-caef-4c18-b4d6-63f6f3a3d38a</a:RelatesTo>
</s:Header>
- <s:Body>
- <s:Fault>
- <s:Code>
  <s:Value>s:Receiver</s:Value>
</s:Code>
- <s:Reason>
  <s:Text xml:lang="de-DE">Error occured</s:Text>
</s:Reason>
- <s:Detail>
- <xta:InvalidMessageIDException xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xta="http://xoev.de/transport/xta/211">
- <xta:errorCode listVersionID="1.0" listURI="urn:de:xta:webservice:codeliste:fehlernummer">
  <code>9070</code>
  <name>MessageID für den Account nicht bekannt: urn:xta:messageid:dataport_xta_210:69490cbc-
  67cd-4470-9929-97f0cc2289ef</name>
  </xta:errorCode>
</xta:InvalidMessageIDException>
</s:Detail>
</s:Fault>
</s:Body>
</s:Envelope>

```

Die Mitteilung von Fehlern bei Verarbeitung innerhalb der Methoden des XTA Webservice wird mit dem Element <Fault> realisiert, welches vom SOAP-Standard für Fehlerbehandlung vorgesehen ist. Es ist vom Webservice-Client auszuwerten, so dass der Anwender möglichst genau weiß, welcher Fehler aufgetreten ist bzw. was geändert werden muss, um die Fachnachricht korrigiert übersenden zu können.

- Im Element <Detail> wird die Identität des Fehlers genannt, indem das entsprechende Exception-Objekt übergeben wird gemäß [Abschnitt 2.3.1 auf Seite 83](#). Es muss vom XTA Webservice-Client ausgewertet werden.
- Im Element <errorCode> wird die Fehlernummer (Unterelement *code*) und die entsprechende textuelle Repräsentation (Unterelement *name*) eingetragen. Es muss gefüllt, aber nicht ausgewertet werden.

# A Codelisten



In diesem Abschnitt sind die in XTA verwendeten Codelisten und ihre Inhalte aufgeführt.

## A.1 Übersicht

In der nachstehenden Tabelle werden die folgenden Informationen dargestellt:

### Codeliste

Alle in XTA genutzten Codelisten in alphabetischer Reihenfolge, die in mindestens einem Code-Datentyp genutzt werden (Typ der Codelistennutzung 1 bis 3).<sup>1</sup>

### Version

Die Version der Codeliste.

### Code-Datentyp(en)

Die die jeweilige Codeliste nutzenden Code-Datentypen.<sup>1</sup>

Die Namen der Code-Datentypen und der Codelisten stellen Links zu den jeweiligen Detail-Abschnitten dar.

Codeliste	Version	Code-Datentyp(en)
Abgabestation	1.0	Code.Abgabestation
Abweichverhalten	1.0	Code.Abweichverhalten
Bezeichner für kryptographische Verfahren	1.0	Code.KryptographischeAlgorithmen
Ereignisart	1.0	Code.EreignisArtType
Geltungsbereich Infrastruktur-Parameter	1.0	Code.GeltungsbereichInfrastrukturprofil-Parameter
Geltungsbereich Schutzprofil-Parameter	1.0	Code.GeltungsbereichSchutzprofil-Parameter
Kanal	1.0	Code.Kanal
Kommunikation Typ	1.0	Code.Kommunikationstyp
Nachweis Verlässlichkeit	1.0	Code.NachweisVerlaesslichkeit
Qualität Authentizität	1.0	Code.QualitaetAuthentizitaet
Qualität Kryptographie	1.0	Code.QualitaetKryptographie
Qualität Löschen	1.0	Code.QualitaetLoeschen
Qualität Protokollierung	1.0	Code.QualitaetProtokollierung
Qualität Unveränderbarkeit	1.0	Code.QualitaetUnveraenderbarkeit
Qualität Verfügbarkeit	1.0	Code.QualitaetVerfuegbarkeit
Qualität Vertraulichkeit	1.0	Code.QualitaetVertraulichkeit
Schlüssellängen für asymmetrische kryptographische Verfahren	1.0	Code.KryptographischeSchluessellaenge

<sup>1</sup>Sofern in der Spalte „Code-Datentyp(en)“ kein Eintrag vorhanden ist, bedeutet dies, dass der Standard die jeweilige Codeliste verwendet und dokumentieren möchte. Der die Codeliste nutzende Code-Datentyp ist jedoch nicht im Standard spezifiziert.

Codeliste	Version	Code-Datentyp(en)
Startzeitpunkt Löschfrist	1.0	Code.StartzeitpunktLoeschfrist
Technische Quittungen	1.0	Code.TechnischeQuittungen
Transportnachrichten Format	1.0	Code.TransportnachrichtenFormat
Transportprotokoll	1.0	Code.Transportprotokoll
Verzeichnis für Adressierung	1.0	Code.VerzeichnisAdressierung
Verzeichnis für Identifizierung	1.0	Code.VerzeichnisIdentifizierung
XTA-Rolle	1.0	Code.XTARolle
XTA-WS Fehlernummer	1.0	Code.Fehlernummer
Zertifikat Medium	1.0	Code.ZertifikatMedium
Zertifikat Niveau	1.0	Code.ZertifikatNiveau
Zertifikat Quelle	1.0	Code.ZertifikatQuelle
Zertifikat Status	1.0	Code.ZertifikatStatus
Zustellfrist	1.0	Code.Zustellfrist

## A.2 Details

### A.2.1 Abgabestation

Diese Codeliste beschreibt die Knoten der Infrastruktur, an denen eine Nachricht final abgeliefert werden kann. So lässt sich bspw. steuern, ob direkt zuzustellen ist oder ob eine Ablage ins Postfach vorgesehen ist.

#### A.2.1.1 Metadaten

Metadatenelement	Wert
Name (lang)	Abgabestation
Name (kurz)	Abgabestation
Kennung	urn:xoev-de:xta:core:codeliste:abgabestation
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

#### A.2.1.2 Daten

Code (code)	Wert (beschreibung)
relay	Abgabestation ist der Knoten 'Relay des Empfängers (Postfach)'
empfänger	Abgabestation ist die Rolle 'Empfänger'
leser	Abgabestation ist die Rolle 'Leser'

## A.2.2 Abweichverhalten

Diese Codeliste enthält die verschiedenen Vorgaben für das Abweichverhalten, das anzuwenden ist, wenn die zugehörige Vorgabe nicht erfüllt wird. Die Vorgabe wird im Service Profil festgelegt.

### A.2.2.1 Metadaten

Metadatenelement	Wert
Name (lang)	Abweichverhalten
Name (kurz)	Abweichverhalten
Kennung	urn:xoev-de:xta:core:codeliste:abweichverhalten
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.2.2 Daten

Code (code)	Wert (beschreibung)
akzeptieren	Die Vorgabe ist nicht erfüllt. Der Arbeitsschritt, in dem die Vorgabe nicht erfüllt wurde, gilt als abgeschlossen und die Bearbeitung der Nachricht wird fortgesetzt. Soweit für die Servicequalität nichts anderes angegeben ist, muss zusätzlich zum Eintrag im ServiceReport ein Eintrag im TransportReport mit dem Status 2 erfolgen (gelb) und die WarnList wird um einen entsprechenden Eintrag erweitert (vgl. MessageStatusType).
ablehnen	Die Vorgabe ist nicht erfüllt. Der Arbeitsschritt, in dem die Vorgabe nicht erfüllt wurde, gilt als abgeschlossen und die Bearbeitung der Nachricht wird abgelehnt. Soweit für die Servicequalität nichts anderes angegeben ist, muss zusätzlich zum Eintrag im ServiceReport ein Eintrag im TransportReport mit dem Status 3 erfolgen (rot) und die ErrorList wird um einen entsprechenden Eintrag erweitert (vgl. MessageStatusType).

## A.2.3 Bezeichner für kryptographische Verfahren

Diese Codeliste definiert die in den Service Profilen zu verwendenden eindeutigen Bezeichner kryptographischer Verfahren. Es handelt sich um Bezeichner nach "XML Security Algorithm Cross-Reference" der W3 Gruppe, verfügbar unter <http://www.w3.org/TR/xmlsec-algorithms/>. In diesem Dokument der W3 Gruppe ist jedem Verfahren eine eindeutige URI zugeordnet, diese URI wird in XTA als Bezeichner für kryptographische Verfahren verwendet und ist in der Spalte „Wert“ aufgeführt. Die Spalte „Schlüssel“ enthält den Namen des Verfahrens, wie er im Dokument der W3 verwendet wird.

### A.2.3.1 Metadaten

Metadatenelement	Wert
Name (lang)	Bezeichner für kryptographische Verfahren
Name (kurz)	Bezeichner für kryptographische Verfahren
Kennung	urn:xoev-de:xta:serviceprofile:codeliste:kryptographische-algorithmen
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.3.2 Daten

Code (code)	Wert (beschreibung)
DSA-SHA256	<a href="http://www.w3.org/2009/xmlsig11#dsa-sha256">http://www.w3.org/2009/xmlsig11#dsa-sha256</a>
RSA-SHA256	<a href="http://www.w3.org/2001/04/xmlsig-more#rsa-sha256">http://www.w3.org/2001/04/xmlsig-more#rsa-sha256</a>
RSA-SHA384	<a href="http://www.w3.org/2001/04/xmlsig-more#rsa-sha384">http://www.w3.org/2001/04/xmlsig-more#rsa-sha384</a>
RSA-SHA512	<a href="http://www.w3.org/2001/04/xmlsig-more#rsa-sha512">http://www.w3.org/2001/04/xmlsig-more#rsa-sha512</a>
ECDSA-SHA256	<a href="http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha256">http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha256</a>
ECDSA-SHA384	<a href="http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha384">http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha384</a>
ECDSA-SHA512	<a href="http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha512">http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha512</a>
SHA-256	<a href="http://www.w3.org/2001/04/xmlenc#sha256">http://www.w3.org/2001/04/xmlenc#sha256</a>
SHA-384	<a href="http://www.w3.org/2001/04/xmlsig-more#sha384">http://www.w3.org/2001/04/xmlsig-more#sha384</a>
SHA-512	<a href="http://www.w3.org/2001/04/xmlenc#sha512">http://www.w3.org/2001/04/xmlenc#sha512</a>
AES-256 (CBC mode)	<a href="http://www.w3.org/2001/04/xmlenc#aes256-cbc">http://www.w3.org/2001/04/xmlenc#aes256-cbc</a>
AES256-GCM	<a href="http://www.w3.org/2009/xmlenc11#aes256-gcm">http://www.w3.org/2009/xmlenc11#aes256-gcm</a>
RSA-OAEP-MGF1P	<a href="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p">http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p</a>
RSA-OAEP	<a href="http://www.w3.org/2009/xmlenc11#rsa-oaep">http://www.w3.org/2009/xmlenc11#rsa-oaep</a>
RSA-OAEP-MGF1P-SHA256	<a href="http://www.w3.org/2009/xmlenc11#mgf1sha256">http://www.w3.org/2009/xmlenc11#mgf1sha256</a>
RSA-OAEP-MGF1P-SHA384	<a href="http://www.w3.org/2009/xmlenc11#mgf1sha384">http://www.w3.org/2009/xmlenc11#mgf1sha384</a>

---

Code (code)	Wert (beschreibung)
RSA-OAEP- MGF1P- SHA512	<a href="http://www.w3.org/2009/xmlenc11#mgf1sha512">http://www.w3.org/2009/xmlenc11#mgf1sha512</a>

## A.2.4 Ereignisart

Diese Codeliste definiert die für den ServiceReport zu protokollierenden Ereignisarten.

### A.2.4.1 Metadaten

Metadatenelement	Wert
Name (lang)	Ereignisart
Name (kurz)	Ereignisart
Kennung	urn:xoev-de:xta:webservice:codeliste:ereignisart
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.4.2 Daten

Code (code)	Wert (beschreibung)
transportauftragsIDErstellen	Dieses Ereignis ist vom Sender zu protokollieren, wenn dieser eine TransportauftragsID erstellt.
versandauftragErhalten	Dieses Ereignis ist vom Sender zu protokollieren, wenn dieser einen Versandauftrag erhält.
zustellungstermin	Dieses Ereignis ist vom Sender zu protokollieren, wenn der Autor dem Sender einen Zustellungstermin vorgibt.
zustellfrist	Dieses Ereignis ist vom Sender zu protokollieren, wenn der Autor dem Sender eine Zustellfrist vorgibt.
abfrageVerzeichnisdienst	Dieses Ereignis ist vom Sender zu protokollieren, wenn dieser einen Verzeichnisdienst abfragt.
transportverschlüsselungAnbringen	Dieses Ereignis ist vom Sender zu protokollieren, wenn dieser eine Transportverschlüsselung anbringt.
transportsignaturAnbringen	Dieses Ereignis ist vom Sender zu protokollieren, wenn dieser eine Transportsignatur anbringt.
nachrichtVersenden	Dieses Ereignis ist vom Sender zu protokollieren, wenn dieser eine Nachricht versendet.
versandauftragZurueckziehen	Dieses Ereignis ist vom Sender zu protokollieren, wenn der Autor einen erteilten Versandauftrag zurückzieht.
abbruchZustellversuche	Dieses Ereignis ist vom Sender zu protokollieren, wenn dieser die Zustellversuche abbricht, weil die maximale Anzahl an Zustellversuchen überschritten wurde.
quittungsempfang	Dieses Ereignis ist vom Sender zu protokollieren, wenn dieser eine Quittung empfängt.
nachrichtLoeschen	Dieses Ereignis ist vom Sender oder Empfänger zu protokollieren, wenn diese eine Nachricht löschen. Bei wem dieses Ereignis ausgelöst wird, wird im Element XTA-Rolle dokumentiert.
serviceQualitaetVerletzt	Dieses Ereignis ist vom Sender oder Empfänger zu protokollieren, wenn eine Servicequalität nicht erbracht werden kann. Bei wem dieses Ereignis ausgelöst wird, wird im Element XTA-Rolle dokumentiert.
nachrichtEmpfangen	Dieses Ereignis ist vom Empfänger zu protokollieren, wenn dieser eine Nachricht annimmt.
aufloesungTransportverschlüsselung	Dieses Ereignis ist vom Empfänger zu protokollieren, wenn dieser eine Transportverschlüsselung auflöst.



Code (code)	Wert (beschreibung)
pruefungTrans- portsignatur	Dieses Ereignis ist vom Sender zu protokollieren, wenn dieser eine Transportsignatur prüft.
verifikationAutor	Dieses Ereignis ist vom Empfänger zu protokollieren, wenn dieser den Autor einer Nachricht verifiziert.
onlineZertifikats- pruefung	Dieses Ereignis ist vom Empfänger zu protokollieren, wenn dieser eine Online-Zertifikatsprüfung vornimmt.
weiterleitungAn- Leser	Dieses Ereignis ist vom Empfänger zu protokollieren, wenn dieser eine synchrone Nachricht an den Leser weiterleitet.
abholenDurch- Leser	Dieses Ereignis ist vom Empfänger zu protokollieren, wenn ein Leser eine Nachricht abholt.
abholbestaeti- gung	Dieses Ereignis ist vom Empfänger zu protokollieren, wenn der Leser die Abholung einer Nachricht bestätigt.
quittungsver- sand	Dieses Ereignis ist vom Empfänger zu protokollieren, wenn dieser eine Quittung versendet.
notifikationGe- sendet	Dieses Ereignis ist vom Empfänger zu protokollieren, wenn dieser eine Notifikation an den Leser sendet.
protokollAbgeru- fen	Dieses Ereignis ist vom Sender oder Empfänger zu protokollieren, wenn eine Nachricht den Schutzbedarf <quote>hoch</quote> hat und sie Gegenstand eines Service Report ist. Bei wem dieses Ereignis ausgelöst wird, wird im Element XTA-Rolle dokumentiert.
protokollie- rungGeloesch	Dieses Ereignis ist vom Sender oder Empfänger zu protokollieren, wenn Nachweise nach Erreichen der Löschfrist automatisch vom XTA-Server geloescht werden. Bei wem dieses Ereignis ausgelöst wird, wird im Element XTA-Rolle dokumentiert.

## A.2.5 Geltungsbereich Infrastruktur-Parameter

Diese Codeliste definiert Kommunikationsstrecken für den Geltungsbereich eines Infrastrukturprofils.

### A.2.5.1 Metadaten

Metadatenelement	Wert
Name (lang)	Geltungsbereich Infrastruktur-Parameter
Name (kurz)	Geltungsbereich Infrastruktur-Parameter
Kennung	urn:xoev-de:xta:serviceprofile:codeliste:geltungsbereich.infrastrukturprofil-parameter
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.5.2 Daten

Code (code)	Wert (beschreibung)
bundesweit	Geltungsbereich ist der bundesweite Nachrichtenaustausch.
länderübergreifend	Geltungsbereich ist der länderübergreifende Nachrichtenaustausch.
landesintern	Geltungsbereich ist landesinterner Nachrichtenaustausch.
kein	Es ist kein Geltungsbereich definiert.

## A.2.6 Geltungsbereich Schutzprofil-Parameter

Diese Codeliste definiert Kommunikationsstrecken auf die sich ein Schutzprofil beziehen kann.

### A.2.6.1 Metadaten

Metadatenelement	Wert
Name (lang)	Geltungsbereich Schutzprofil-Parameter
Name (kurz)	Geltungsbereich Schutzprofil-Parameter
Kennung	urn:xoev-de:xta:serviceprofile:codeliste:geltungsbereich.schutzprofil-parameter
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.6.2 Daten

Code (code)	Wert (beschreibung)
autor-leser	Kommunikationsstrecke Autor - Leser
autor-sender	Kommunikationsstrecke Autor - Sender
autor-empfänger	Kommunikationsstrecke Autor - Empfänger
sender-empfänger	Kommunikationsstrecke Sender - Empfänger
empfänger-leser	Kommunikationsstrecke Empfänger - Leser

## A.2.7 Kanal

Diese Codeliste definiert die möglichen Netze über die eine Nachricht zwischen Sender und Empfänger übermittelt werden darf.

### A.2.7.1 Metadaten

Metadatenelement	Wert
Name (lang)	Kanal
Name (kurz)	Kanal
Kennung	urn:xoev-de:xta:core:codeliste:kanal
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.7.2 Daten

Code (code)	Wert (beschreibung)
internet	Übermittlung über das Internet.
verbindungsnetz	Übermittlung über das Verbindungsnetz.
notarnet	Übermittlung über das NotarNet.
rz-intern	Rechenzentrumsinterne Übermittlung.
systemintern	Systeminterne Übermittlung.

## A.2.8 Kommunikation Typ

Diese Codeliste definiert mögliche Kommunikationsszenarien.

### A.2.8.1 Metadaten

Metadatenelement	Wert
Name (lang)	Kommunikation Typ
Name (kurz)	Kommunikation Typ
Kennung	urn:xoev-de:xta:core:codeliste:kommunikationstyp
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.8.2 Daten

Code (code)	Wert (beschreibung)
asynchron	Die Kommunikation ist asynchron.
synchron	Die Kommunikation ist synchron.

## A.2.9 Nachweis Verlässlichkeit

Die Codeliste definiert die Stufen der Verlässlichkeit einer Organisation.

### A.2.9.1 Metadaten

Metadatenelement	Wert
Name (lang)	Nachweis Verlässlichkeit
Name (kurz)	Nachweis Verlässlichkeit
Kennung	urn:xoev-de:xta:serviceprofile:codeliste:nachweis.verlaesslichkeit
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.9.2 Daten

Code (code)	Wert (beschreibung)
eigenerklärung	Als Nachweis zählt eine Betreiber- oder Herstellererklärung mit einem entsprechenden Vertrag. Die Erklärung ist an einem definierten Ort hinterlegt.
zertifizierung	Als Nachweis wird eine Zertifizierung gefordert. Der Betreiber oder Hersteller durchläuft erfolgreich einen Zertifizierungsprozess und hinterlegt den Zertifizierungsnachweis an einem definierten Ort.

## A.2.10 Qualität Authentizität

Diese Codeliste definiert Niveaus, mit dem die Authentizität der Nachrichtenkommunikation abgesichert werden soll.

### A.2.10.1 Metadaten

Metadatenelement	Wert
Name (lang)	Qualität Authentizität
Name (kurz)	Qualität Authentizität
Kennung	urn:xoev-de:xta:serviceprofile:codeliste:authentizitaetqualitaet
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.10.2 Daten

Code (code)	Wert (beschreibung)
normal	Selbstbehauptung und Plausibilitätsprüfung mit vertrauenswürdiger Dritter.
hoch	Prüfung mit Mitteln die eine Akkreditierung erfordern (z.B. Zertifikate).

## A.2.11 Qualität Kryptographie

Diese Codeliste enthält die Schlüssel zur Angabe des geforderten Schutzniveaus einer kryptographisch zu sichernden Kommunikation.

Die hier verwendeten Abstufungen basieren auf den vom BSI im Kontext der Schutzbedarf-Feststellung definierten Begriffen zum IT-Grundschutz .

### A.2.11.1 Metadaten

Metadatenelement	Wert
Name (lang)	Qualität Kryptographie
Name (kurz)	Qualität Kryptographie
Kennung	urn:xoev-de:xta:core:codeliste:kryptographie.qualitaet
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.11.2 Daten

Code (code)	Wert (beschreibung)
normal	Es handelt sich um kryptographisch abzusichernde Kommunikation des geforderten Schutzniveaus 'normal'
hoch	Es handelt sich um kryptographisch abzusichernde Kommunikation des geforderten Schutzniveaus 'hoch'



## A.2.12 Qualität Löschen

Diese Codeliste definiert Anforderungen an das Löschen.

### A.2.12.1 Metadaten

Metadatenelement	Wert
Name (lang)	Qualität Löschen
Name (kurz)	Qualität Löschen
Kennung	urn:xoev-de:xta:core:codeliste:loeschen.qualitaet
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.12.2 Daten

Code (code)	Wert (beschreibung)
normal	Einfaches Löschen.
hoch	Sicheres Löschen.

## A.2.13 Qualität Protokollierung

Diese Codeliste legt Ausprägungen der Protokollierung fest.

### A.2.13.1 Metadaten

Metadatenelement	Wert
Name (lang)	Qualität Protokollierung
Name (kurz)	Qualität Protokollierung
Kennung	urn:xoev-de:xta:serviceprofile:codeliste:protokollierung.qualitaet
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.13.2 Daten

Code (code)	Wert (beschreibung)
niedrig	Optionale Protokollierung.
normal	VerpflichtendeProtokollierung.
hoch	Revisionssichere Protokollierung.

## A.2.14 Qualität Unveränderbarkeit

Diese Codeliste definiert den Umfang, in dem die Unveränderbarkeit (Integrität) geschützt wird.

### A.2.14.1 Metadaten

Metadatenelement	Wert
Name (lang)	Qualität Unveränderbarkeit
Name (kurz)	Qualität Unveränderbarkeit
Kennung	urn:xoev-de:xta:serviceprofile:codeliste:unveraenderbarkeit.qualitaet
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.14.2 Daten

Code (code)	Wert (beschreibung)
normal	Systeminterne Mittel.
hoch	Zusätzliche Schutzmaßnahmen.
niedrig	Optional.

## A.2.15 Qualität Verfügbarkeit

Diese Codeliste definiert Verfügbarkeitsstufen.

### A.2.15.1 Metadaten

Metadatenelement	Wert
Name (lang)	Qualität Verfügbarkeit
Name (kurz)	Qualität Verfügbarkeit
Kennung	urn:xoev-de:xta:serviceprofile:codeliste:verfuegbarkeit.qualitaet
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.15.2 Daten

Code (code)	Wert (beschreibung)
normal	Anforderung normale Verfügbarkeit (98,5 % im Tagesbetrieb)
hoch	Es ist hohe Verfügbarkeit vorgesehen (98,5 % im 7 * 24-Stunden-Betrieb)
sehrhoch	Es ist sehr hohe Verfügbarkeit vorgesehen (99,5 % im 7 * 24 Stunden Betrieb).
extremhoch	Es ist extrem hohe Verfügbarkeit vorgesehen (99,9 % im 7 * 24 Stunden Betrieb).

## A.2.16 Qualität Vertraulichkeit

Diese Codeliste definiert Vertraulichkeitsniveaus.

### A.2.16.1 Metadaten

Metadatenelement	Wert
Name (lang)	Qualität Vertraulichkeit
Name (kurz)	Qualität Vertraulichkeit
Kennung	urn:xoev-de:xta:serviceprofile:codeliste:vertraulichkeit.qualitaet
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.16.2 Daten

Code (code)	Wert (beschreibung)
öffentlich	Keine Vertraulichkeit.
normal	Normale Vertraulichkeit.
hoch	Hohe Vertraulichkeit.

## A.2.17 Schlüssellängen für asymmetrische kryptographische Verfahren

Diese Codeliste enthält die zu verwendenden minimalen Schlüssellängen für asymmetrische Verschlüsselungsverfahren.

### A.2.17.1 Metadaten

Metadatenelement	Wert
Name (lang)	Schlüssellängen für asymmetrische kryptographische Verfahren
Name (kurz)	Schlüssellängen für asymmetrische kryptographische Verfahren
Kennung	urn:xoev-de:xta:serviceprofile:codeliste:kryptographische-schluessellaengen
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.17.2 Daten

Code (code)	Wert (beschreibung)
ECDSAmin	250
ECDSAmittel	256
ECDSAhoch	264
RSAMin	2048
RSAmittel	3072
RSAhoch	4096

## A.2.18 Startzeitpunkt Löschfrist

Diese Codeliste beschreibt Startzeitpunkte für Löschfristen.

### A.2.18.1 Metadaten

Metadatenelement	Wert
Name (lang)	Startzeitpunkt Löschfrist
Name (kurz)	Startzeitpunkt Löschfrist
Kennung	urn:xoev-de:xta:core:codeliste:startzeitpunkt-loeschfrist
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.18.2 Daten

Code (code)	Wert (beschreibung)
erhebung	Die Löschfrist beginnt mit Entgegennahme des Transportauftrags.
vorgangsende	Die Löschfrist beginnt mit Abschluss des Transportauftrags.
beziehungsende	Die Löschfrist beginnt mit Ende der Vertragsbeziehung.

## A.2.19 Technische Quittungen

Diese Codeliste enthält die Schlüssel für die Arten technischer Quittungen, welche in einer XTA-Infrastruktur vorgesehen sind (siehe XTA-Spezifikationsdokument Rahmenbedingungen).

### A.2.19.1 Metadaten

Metadatenelement	Wert
Name (lang)	Technische Quittungen
Name (kurz)	Technische Quittungen
Kennung	urn:xoev-de:xta:core:codeliste:technische.quittungen
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.19.2 Daten

Code (code)	Wert (beschreibung)
submission	Eine Quittung diesen Typs ist durch die Rolle Sender zu erzeugen gdw. die Nachricht erfolgreich versendet wurde.
relay	Eine Quittung diesen Typs ist durch einen Knoten <i>Relay</i> zu erzeugen gdw. die Nachricht erfolgreich weitergeleitet wurde.
delivery	Eine Quittung diesen Typs ist durch die Rolle Sender oder einen Knoten Relay zu erzeugen gdw. die Nachricht an die Rolle Empfänger ausgeliefert wurde bzw. sich - in asynchronen Kommunikationsszenarien - im Zugriffsbereich (Postkorb) des Empfängers befindet.
fetch	Eine Quittung diesen Typs ist durch die Rolle Empfänger zu erzeugen gdw. die Nachricht aus dem Postkorb abgeholt wurde.
keineQuittung	Es werden keine technischen Quittungen verlangt.



## A.2.20 Transportnachrichten Format

Diese Codeliste nennt verfügbare Nachrichtenformate für Transportnachrichten.

### A.2.20.1 Metadaten

Metadatenelement	Wert
Name (lang)	Transportnachrichten Format
Name (kurz)	Transportnachrichten Format
Kennung	urn:xoev-de:xta:core:codeliste:transportnachrichten.format
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.20.2 Daten

Code (code)	Wert (beschreibung)
osci12	OSCI 1.2
osci2	OSCI 2
xta11	XTA 1.1
xta2	XTA 2
systemintern	Es ist ein systeminternes Nachrichtenformat vorgesehen.

## A.2.21 Transportprotokoll

Diese Codeliste nennt verfügbare Protokolle, die die Kommunikation von Daten zwischen Partnern festlegen.

### A.2.21.1 Metadaten

Metadatenelement	Wert
Name (lang)	Transportprotokoll
Name (kurz)	Transportprotokoll
Kennung	urn:xoev-de:xta:core:codeliste:transportprotokoll
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.21.2 Daten

Code (code)	Wert (beschreibung)
http	HTTP
https	HTTPS
ftp	FTP
sftp	SFTP
smtp	SMTP
systemintern	Es ist ein systemintern definiertes Protokoll vorgesehen.

## A.2.22 Verzeichnis für Adressierung

Diese Codeliste enthält die Schlüssel für Verzeichnislösungen zur Bereitstellung von Parametern für die technische Adressierung von Teilnehmern.

### A.2.22.1 Metadaten

Metadatenelement	Wert
Name (lang)	Verzeichnis für Adressierung
Name (kurz)	Verzeichnis für Adressierung
Kennung	urn:xoev-de:xta:core:codeliste:verzeichnis.adressierung
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.22.2 Daten

Code (code)	Wert (beschreibung)
dvdv	Deutsches Verwaltungsdiensteverzeichnis (DVDV)
safe	Verzeichnisdienst SAFE (Secure Access to Federated e-Justice/e-Government)
rz-intern	Es wird als ein rechenzentrumsinternes Verzeichnis eingesetzt.
systemintern	Es wird ein systeminternes Verzeichnis eingesetzt.

## A.2.23 Verzeichnis für Identifizierung

Diese Codeliste enthält die Schlüssel für Verzeichnislösungen zur Verwaltung von elektronischen Identitäten (Bezeichnungen und kryptographische Token für Identität und Identitätsnachweis).

### A.2.23.1 Metadaten

Metadatenelement	Wert
Name (lang)	Verzeichnis für Identifizierung
Name (kurz)	Verzeichnis für Identifizierung
Kennung	urn:xoev-de:xta:serviceprofile:codeliste:verzeichnis.identifizierung
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.23.2 Daten

Code (code)	Wert (beschreibung)
dvdv	Deutsches Verwaltungsdiensteverzeichnis (DVDV)
safe	Verzeichnisdienst SAFE (Secure Access to Federated e-Justice/e-Government)
pki	Public Key Infrastruktur (PKI)
rz-intern	Es wird als ein rechenzentrumsinternes Verzeichnis eingesetzt.
systemintern	Es wird ein systeminternes Verzeichnis eingesetzt.

## A.2.24 XTA-Rolle

Diese Codeliste benennt die Rollen, die in einer XTA-Kommunikationsinfrastruktur am Prozess der Nachrichtenübermittlung beteiligt sind.

### A.2.24.1 Metadaten

Metadatenelement	Wert
Name (lang)	XTA-Rolle
Name (kurz)	XTA-Rolle
Kennung	urn:xoev-de:xta:core:codeliste:xta-rolle
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.24.2 Daten

Code (code)	Wert (beschreibung)
autor	XTA-Rolle 'Autor'
sender	XTA-Rolle 'Sender'
empfänger	XTA-Rolle 'Empfänger'
leser	XTA-Rolle 'Leser'

## A.2.25 XTA-WS Fehlernummer

Diese Codeliste gibt eine Übersicht über die in XTA-WS zu verwendenden Fehlernummern (ErrorCodes) und ordnet sie den Exceptions zu, in deren Kontext sie auftreten können.

### A.2.25.1 Metadaten

Metadatenelement	Wert
Name (lang)	XTA-WS Fehlernummer
Name (kurz)	XTA-WS Fehlernummer
Kennung	urn:de:xta:webservice:codeliste:fehlernummer
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.25.2 Daten

Code (code)	Wert (beschreibung)	Beschreibung
9000	Unspezifizierter Fehler, als Freitext beschreiben	
9010	Authentisierung/Zertifikat ist abgelaufen.	PermissionDeniedException
9011	Account ist gesperrt.	PermissionDeniedException
9012	Account nicht vorhanden.	PermissionDeniedException
9013	Dienst ist nicht gebucht.	PermissionDeniedException
9014	Authentisierung/Zertifikat passt nicht zur Absenderkennung.	PermissionDeniedException
9020	Keine Parameter vorhanden	ParameterIsNotValidException
9021	Keine gültige URI	ParameterIsNotValidException
9022	Ungültige Parameterkombination	ParameterIsNotValidException
9023	Die Nachricht überschreitet die technisch oder organisatorisch festgelegte Größenbeschränkung.	ParameterIsNotValidException
9024	MessageID ist bereits vergeben.	ParameterIsNotValidException
9030	Interner Fehler beim XTA-Server bzw. XTA-Dienstleister	XTAWSTechnicalProblemException
9032	Fehler bei der internen Zustellung	XTAWSTechnicalProblemException
9040	Die Nachricht zur übergebenen MessageMetadata wurde bereits abgeholt und quittiert.	NotificationForFetchedMessage
9050	Fachnachricht ist nicht schemakonform	MessageSchemaViolationException
9051	Fachnachricht trägt ein falsches Encoding.	MessageSchemaViolationException
9052	Nachricht verletzt das entsprechende Service Profil.	MessageSchemaViolationException
9060	Es wurde schadhafter Code ermittelt.	MessageVirusDetectionException
9070	MessageID für den Account nicht bekannt.	InvalidMessageIDException
9080	Der Dienst wird nur asynchron angeboten.	SyncAsyncException
9081	Der Dienst wird nur synchron angeboten.	SyncAsyncException

Code (code)	Wert (beschreibung)	Beschreibung
9090	Die Operation wird nicht angeboten.	NotImplementedException
9100	Der durch den Schalter NotBefore gesetzte Termin ist verstrichen.	CancelDeniedException
9101	Der Schalter NotBefore wurde nicht gesetzt.	CancelDeniedException
9110	Die Ergebnismenge für die gewählten Parameter der Abfrage ist zu groß.	TooManyResultsException
9120	Service nicht erreichbar	ExternalServiceUnavailableException
9121	Dienstverzeichnis nicht erreichbar	ExternalServiceUnavailableException
9122	Identitätsverzeichnis nicht erreichbar	ExternalServiceUnavailableException
9123	PKI nicht erreichbar	ExternalServiceUnavailableException
9124	Leser nicht erreichbar	ExternalServiceUnavailableException
9125	Empfänger nicht erreichbar	ExternalServiceUnavailableException
9200	Parameter wird nicht unterstützt	ParameterNotSupportedException
9201	Testnachrichten werden nicht unterstützt	ParameterNotSupportedException
9202	Sendetermin wird nicht unterstützt	ParameterNotSupportedException
9203	Ablauftermin wird nicht unterstützt	ParameterNotSupportedException
9204	ServiceProfile werden nicht unterstützt	ParameterNotSupportedException
9205	Quittungen werden nicht unterstützt	ParameterNotSupportedException
9206	Callback über NotifyMessageSent wird nicht unterstützt	ParameterNotSupportedException
9207	Zusätzliche SecurityToken werden nicht unterstützt	ParameterNotSupportedException
9208	Abweichende Antwortadressen werden nicht unterstützt	ParameterNotSupportedException
9209	Zusätzliche Empfängeradressen werden nicht unterstützt	ParameterNotSupportedException

## A.2.26 Zertifikat Medium

Diese Codeliste benennt die verschiedenen Medien, die ein Zertifikat tragen können.

### A.2.26.1 Metadaten

Metadatenelement	Wert
Name (lang)	Zertifikat Medium
Name (kurz)	Zertifikat Medium
Kennung	urn:xoev-de:xta:core:codeliste:zertifikat.medium
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.26.2 Daten

Code (code)	Wert (beschreibung)
hardwarebasiert	Es ist ein hardwarebasiertes Medium für das Zertifikat vorgesehen.
softwarebasiert	Es ist ein softwarebasiertes Medium für das Zertifikat vorgesehen.



## A.2.27 Zertifikat Niveau

Diese Codeliste enthält die Schlüssel für die definierten Niveaus eines Zertifikats.

### A.2.27.1 Metadaten

Metadatenelement	Wert
Name (lang)	Zertifikat Niveau
Name (kurz)	Zertifikat Niveau
Kennung	urn:xoev-de:xta:core:codeliste:zertifikat.niveau
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.27.2 Daten

Code (code)	Wert (beschreibung)
fortgeschritten	Vorgesehen sind Zertifikate für fortgeschrittene Signatur.
qualifiziert	Vorgesehen sind Zertifikate für qualifizierte Signatur.

## A.2.28 Zertifikat Quelle

Diese Codeliste enthält gültige Werte für die Angabe des Herausgebers von Zertifikaten in einer Public Key Infrastruktur.

### A.2.28.1 Metadaten

Metadatenelement	Wert
Name (lang)	Zertifikat Quelle
Name (kurz)	Zertifikat Quelle
Kennung	urn:xoev-de:xta:serviceprofile:codeliste:zertifikat.quelle
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.28.2 Daten

Code (code)	Wert (beschreibung)
pki1	PKI1-Verwaltung
doi-hd	DOI CA Hoheitliche Dokumente
testa	DOI (TESTA)
xjustiz-liste	Die Herausgeberschaft ist eingeschränkt auf die Quellen, die in der folgenden Liste genannt werden: <link xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="https://www1.osci.de/sixcms/media.php/13/2009-05-28-zertifikatsherausgeber.pdf">https://www1.osci.de/sixcms/media.php/13/2009-05-28-zertifikatsherausgeber.pdf</link>
beliebig	Keine Einschränkung.

## A.2.29 Zertifikat Status

Diese Codeliste enthält die Schlüssel für den Status eines Zertifikats.

### A.2.29.1 Metadaten

Metadatenelement	Wert
Name (lang)	Zertifikat Status
Name (kurz)	Zertifikat Status
Kennung	urn:xoev-de:xta:serviceprofile:codeliste:zertifikat.status
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.29.2 Daten

Code (code)	Wert (beschreibung)
gültig	Das Zertifikat ist gültig.
abgelaufen	Das Zertifikat ist abgelaufen.
gesperrt	Das Zertifikat ist gesperrt.

## A.2.30 Zustellfrist

Diese Codeliste enthält Werte, die als Zustellfrist verwendet werden können.

### A.2.30.1 Metadaten

Metadatenelement	Wert
Name (lang)	Zustellfrist
Name (kurz)	Zustellfrist
Kennung	urn:xoev-de:xta:core:codeliste:zustellfrist
Herausgeber	Koordinierungsstelle für IT-Standards (KoSIT)
Version	1.0

### A.2.30.2 Daten

Code (code)	Wert (beschreibung)
72h	Zustellung in den Verfügungsbereich des Lesers innerhalb von drei Tagen bzw. 72 Stunden.
24h	Zustellung in den Verfügungsbereich des Lesers innerhalb von einem Tag bzw. 24 Stunden.
unverzüglich	Unverzügliche Zustellung (so schnell wie möglich, ohne dass schuldhaft verzögert wird).

# B Eingebundene externe Modelle/Standards



Folgende externe Modelle werden in dieser Spezifikation verwendet und sind auf den XÖV-Webseiten (siehe <http://www.xoev.de/de/produkte>) oder im XRepository (siehe <http://www.xrepository.de>) veröffentlicht:

## B.1 SOAP-Message-Security-1.0

wsse; Version 1.0

Folgende Datentypen aus dem externen Modell werden in dieser Spezifikation verwendet:

- BinarySecurityToken
- SecurityTokenReference
- UsernameToken

## B.2 WS-Addressing

ws-addressing; Version 1.0

Folgende Datentypen aus dem externen Modell werden in dieser Spezifikation verwendet:

- AttributedURIType
- EndpointReferenceType
- From
- MessageID
- RelatesTo

## B.3 XML-Signature

xmlsig-core; Version 1.1

Folgende Datentypen aus dem externen Modell werden in dieser Spezifikation verwendet:

- Signature

## B.4 XOEV-Bibliothek

XOEV-Bibliothek; Fassung 2017-08-01

Folgende Datentypen aus dem externen Modell werden in dieser Spezifikation verwendet:

- Code
- String.Latin



---

# C Versionshistorie



---

## C.1 Release XTA 2 Version 4 (Final Draft)

### ohne CR: interne Qualitätssicherung der Spezifikation

Im Rahmen einer ausgiebigen Qualitätssicherung wurde das Spezifikationsdokument verschlankt und konsistenter gemacht. Einzelne Inhaltsabschnitte wurden in das separate Dokument Rahmenbedingungen verschoben, darüber hinaus wurde die Spezifikation durch die Entfernung von Grafiken und nicht genutzten Übersichtstabellen weiter komprimiert.

### CR 2014-17 Technical Report

Einführung des ServiceReport (Arbeitstitel „TechnicalReport“) als Container für Ereignisobjekte zur Protokollierung des Transports. Erweiterung von xta-transport.wsdl, transportPortType um Methode getServiceReport zur Abholung eines ServiceReports zu einer Nachricht.

### CR 2015-19 XTA-synchron Request Only

Umsetzung des request-only Szenarios (OSCI Forward Delivery): Ergänzung des Datentyps MessageResponse als Output der Methode sendMessageSync (statt bisherigem GenericContentContainer); außerdem Erweiterung des XTAHeader um Element MessageMetaDataResponse und redaktionelle Anpassungen zur Befüllung der Elemente.

### CR 2015-20 Fehlerbehebung Transport

Einführung neuer Codelist Abweichverhalten sowie des Nachweis-Objekts ServicequalitaetAbweichungNachweis.

### CR 2015-22 Ergänzung CL XhD-Nachrichten

Veröffentlichung einer neuen Codeliste der XhD-Nachrichten im XRepository.

### CR 2016-08 Anpassung der Selektionkriterien XTA-WS-MsgBox an OSCI 2

Erweiterung der Selektoren in den Methoden getMessage und getStatusList in der Dokumentation um Kriterien aus OSCI2 (redaktionell).

### CR 2016-10 Schreibweise Messageld

Die Methode createMessageld wurde umbenannt in createMessageID.

### CR 2016-12 Codeliste Algorithmen-Bezeichner

Einfügen einer neuen Codeliste codelist.KryptographischeAlgorithmen (samt Code-Datentyp Code.KryptographischeAlgorithmen), sowie Typ-Änderung des Elements Algorithm.algorithmidentifier von String auf Code.KryptographischeAlgorithmen. In den Klassen Algorithm, CypherSuiteType und AlgorithmAsymmetricEncryption wurden Attribute in Elemente überführt.

### CR 2016-13 MMD-Erstellung durch den Autor

Redaktionelle Anpassungen der Spezifikation zum Rollenmodell und resultierenden Verantwortlichkeiten hinsichtlich der Erstellung des Transportauftrags.

### CR 2016-15 Security Policy anpassen. Überflüssige Policy Assertions entfernen.

Entfernung der Policies WS-Trust und Soap Security aus beiden WSDL-Dateien.

### CR 2016-22 Größenbeschränkung des ContentContainer streichen

Entfernung der überholten Größenbeschränkung auf 40 MB aus der Dokumentation des ContentContainer (aktueller Name: Message). Aufnahme einer Obergrenze für Nachrichtengrößen (für den Payload) in die ServiceKategorie (maxNachrichtengroesse).

**CR 2016-23 Überprüfung OSCI Parameter**

In xta-transport.wsdl, recipientPortType, close: Ergänzung der Exception ParameterNotValid

**CR 2016-24 Trennung Fachlichkeit vs. Transport im Fachstandard**

Redaktionelle Verfeinerung der Abgrenzung von Fachlichkeit und Transport.

**CR 2016-26 Korrektur der SOAP-Exceptions in cancelMessage und getNextMessage**

xta-transport.wsdl, senderPortType, cancelMessage: Ersetzung der Exception ParameterIsNotValidException durch InvalidMessageIDException. Im recipientPortType, getNextMessage: Entfernung der Exception InvalidMessageIDException.

**CR 2016-27 Dokumentation WSDL-Inhalte**

Der CR verfolgte das Ziel, mehr Informationen zu den WSDL-Policies in der Spezifikation unterzubringen (spezifisch MTOM). Der CR wurde umgesetzt, der übergeordnete Abschnitt jedoch in der 2019 durchgeführten Qualitätssicherung entfernt. Die zusätzlichen Informationen richteten sich nicht an Entwickler und Entwicklerinnen und damit nicht an die Zielgruppe der Spezifikation.

**CR 2016-28 Methodenaufrufe Beispielcode**

Aktualisierung der Beispiel-Methodenaufrufen in der Spezifikation.

**CR 2017-01 Nachricht mit Leser-Notifikation**

Erweiterung von xta-application.wsdl um Methode notifyMessageReceived im readerPortType sowie die Ausnahme NotificationForFetchedMessageException.

**CR 2017-03 Notifikation des Autors durch den Sender nach erfolgreicher Zustellung**

Erweiterung des Webservice um eine Methode zur Benachrichtigung eines Autors über die erfolgte Umsetzung eines Transportauftrages: notifyMessageSent in xta-application.wsdl, authorPortType sowie Erweiterung von DeliveryAttributes um ein Element NotifyMessageSent, um im Transportauftrag eine Benachrichtigung über die Zustellung zu beauftragen.

**CR 2017-04 Attribute für neue und abgeholte Nachrichten**

Redaktionelle Konkretisierung der Vorgaben zur Nutzung des DeliveryAttributes-Attributes @InitialFetch.

**CR 2017-05 Umbenennung WSDL - PortTypes - Methoden**

Umstrukturierung der Webservice-Methoden in xta-transport.wsdl (mit senderPortType, recipientPortType und transportPortType) und xta-application.wsdl (mit authorPortType und readerPortType).

**CR 2017-06 Aufnahme des Fachkontextes XInneres in die Codeliste ""XTA BusinessScenario""**

Aktualisierung der externen Codeliste XTA BusinessScenario (urn:de:xta:codelist:business.scenario) und Ergänzung von Codewerten „XINNERES\_DATA“ sowie „GEWERBE\_DATA“ und Veröffentlichung in der Version 2 mit Gültigkeit ab dem 01.11.2017.

**CR 2017-09 Effiziente Übertragung großer Datenmengen**

Um das Streaming großer Datenmengen zu ermöglichen, wird der Datentyp ContentType als Erweiterung einer xs:base64Binary um das Attribut @ mimeType:expectedContentType="application/octet-stream" erweitert.

**CR 2017-10 Author- oder ReaderIdentifier?**

Umbenennung des XTAHeader Message-Parts AuthorIdentifier in XTAIdentifier, sowie Änderung des Part-Elements von oscimeta:PartyType in oscimeta:Identifier.

**CR 2018-01 DS-GVO Anforderungen und XTA Ereignisse**

(Vorarbeiten für die Umsetzung von CR 2014-17 Technical Report)

**CR 2018-02 XTA-Integration von OSCI-2-Objekten in die XTA Spezifikation**

Bisher importierte Strukturen aus OSCI2 und OSCI2 MessageMetaData werden in XTA integriert, bleiben aber als eigene Schema-Dateien mit den bisherigen Namespaces erhalten.

**CR 2018-04 oscimeta:MessageType**

Datentyp MessageType wurde auf Typ String geändert, mit einem Attribut @payloadSchema (Typ NonEmptyURIType). Analog wurde das Attribut @payloadSchema des Datentyps NachrichtentypListe von Typ String auf Typ NonEmptyURIType umgestellt.



**CR 2019-01 Namespaces entkoppeln (xtacore)**

Verschränkung zwischen Bereichen Webservice und ServiceProfile wird aufgehoben, indem gemeinsam genutzte Datentypen in einen neuen Bereich „core“ verschoben werden. Bisher sind davon nur 15 Codelisten betroffen.

**CR 2019-04 SendMessageSync im ReaderPort**

In xta-application.wsdl wurde im readerPortType die Methode sendMessageSync semantisch akkurat umbenannt in deliverMessage.

**CR 2019-05 NotifyMessageSent um ServiceURI erweitern**

Änderung von NotifyAuthorType: Umbenennung des Elements NotifyMessageSent in NotifyAuthorEnabled (mandatorisches Boolean-Element), außerdem Umbenennung von DeliveryAttributes.NotifyMessageSent in NotifyAuthor.

**CR 2019-06 Enumeration für keysize**

Erzeugung einer neuen externen Codeliste für Schlüssellängen namens kryptographische-schlüssellaenge zur Nutzung im Element keysize des Datentyps AlgorithmAsymmetricEncryption.

**CR 2019-08 Streichung der Inhalte des technischen Strukturprofils**

Streichung der Datentypen TechnischesStrukturprofil sowie TechnischeStrukturkategorie; außerdem Löschung der nun ungenutzten Datentypen Variables, Constant, DocumentRef, MessagePart, ProcessingList, XML\_Injection, ProcessingInstruction, Processing sowie des Elements ServiceProfile.refTechnischesStrukturprofil.

**CR 2019-12 Ergänzung einer Exception für Nichterreichbarkeit (IOException)**

Einführung einer Ausnahme ExternalServiceUnavailableException zur Abbildung von Nichterreichbarkeit sowie Aktualisierung der Codeliste Code.Fehlernummer.

**CR 2019-13 Neue Exception ParameterNotSupportedException**

Ergänzung einer Ausnahme ParameterNotSupportedException als Signal, dass optionale Parameter nicht implementiert sind. Außerdem Ergänzung von Fehlernummern im Zahlenraum 9200-9299.

**CR 2019-15 Einfache Struktur für Payload**

Umbenennung von GenericContentContainer in Message, ausgestaltet als Sequenz von Elementen MessagePart vom Typ ContentType. Erweiterung des Datentyps ContentType um ein Attribut @category (Enumeration mit Werten „body“ und „attachment“) sowie @refID (Typ IDREF) zum Bezug auf andere MessagePart-Elemente. Entsprechende Anpassungen in den WSDL-Dateien.

**CR 2019-16 Korrektur SOAP Action in WSDL**

In beiden WSDL-Dateien wird innerhalb jedes operation-Elements bei den Elementen input, output und fault ein Attribut @wsam:Action hingefügt und nach einer Namenskonvention befüllt. Darüber hinaus sind die Attribute soapAction aus den WSDL-Dateien entfernt worden (in WSDL 1.2 und WSDL nicht mehr unterstützt).

**CR 2019-17 Dokumentation und Verwendung des OptHeaders X509TokenContainer**

Vollständige Streichung des Headers OptHeader sowie des X509TokenContainer (betrifft Methoden sendMessage, sendMessageSync, notifyMessageSent, deliverMessage, notifyMessageReceived). Entfernt wurden außerdem die nun ungenutzten Datentypen X509TokenApplication, X509TokenContainer, X509TokenContainerType und X509TokenInfo.

**CR 2019-20 Autor-Notifikation (notifyMessageSent) liefert TransportReport**

Änderung der Rückgabewerte der Methode notifyMessageSent in xta-application.wsdl: Als Input-Parameter wird xta:TransportReport (ohne Header) übergeben. Aus dem Output-Header der Methode wird der Transportauftrag MessageMetaData gestrichen.

