Koordinierungsstelle
für IT-Standards

IT-Planungsrat

Spezifikation
XTA 2 (Version 3)

31. Januar 2017 / final

Inhaltsverzeichnis

1 Einleitung

Das vorliegende Dokument spezifiziert den vom IT-Planungsrat beauftragten Interoperabilitätsstandard XTA 2. [1]

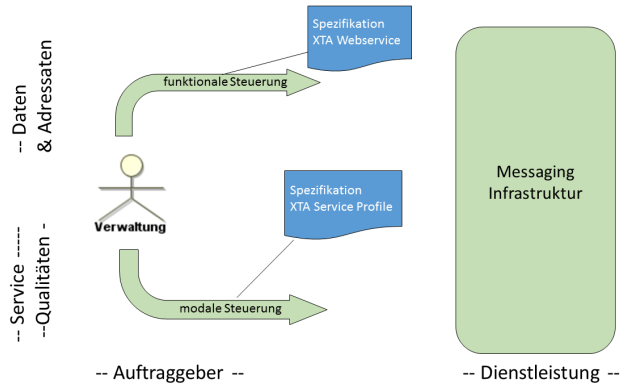
XTA 2 standardisiert die elektronische Übermittlung von Daten im E-Government durch zwei Ansätze auf unterschiedlichen Ebenen:

Durch das Modul der Service Profile wird ein Werkzeug angeboten, mit dem die *Anforderungen* an Datenschutz und Datensicherheit, z. B. bzgl. der Integrität oder Authentizität, für einen Transport definiert und damit einheitlich konfigurierbar gemacht werden können. Der öffentlichen Verwaltung soll so eine *modale* Steuerung ermöglicht werden, siehe auch Abbildung 1.1, „Die zwei Dimensionen der Steuerung der IT-Dienstleistung“.

Durch das Modul des XTA-Webservice (XTA-WS) wird die *Übermittlung* von Daten, also der Transport selbst, standardisiert: Durch die Spezifikation von Webservices wird eine Vereinheitlichung der Schnittstellen zwischen Fachverfahren und Transportverfahren (auch innerhalb eines Landes und Rechenzentrums) erreicht. Die öffentliche Verwaltung erhält so die Möglichkeit der *funktionalen* Steuerung, siehe Abbildung 1.1, „Die zwei Dimensionen der Steuerung der IT-Dienstleistung“.

Die beiden Module "Service Profile" und "XTA-WS" beziehen sich konzeptionell stark aufeinander, es ist aber dennoch möglich, sie unabhängig voneinander einzusetzen.

Abbildung 1.1. Die zwei Dimensionen der Steuerung der IT-Dienstleistung



[1] Der IT-Planungsrat hat im März 2012 die Entwicklung des Interoperabilitätsstandards XTA beauftragt, siehe Beschlüsse 2012/15 und 2012/23 sowie Top 23 (Sachstandsbericht) der 15. Sitzung im Oktober 2014. Ergänzend sei auf den Beschluss des AK I der Innenministerkonferenz vom Oktober 2011 verwiesen, in dem betont wird, dass eine zeitnahe Standardisierung in den Strukturen des IT-Planungsrates sowie die Anwendung einer funktionsfähigen allgemeinen Schnittstelle für erforderlich gehalten wird.

1.1 Hintergrund

Für nahezu alle Projekte und Vorhaben des E-Government ist die Gewährleistung der sicheren Datenübermittlung eine notwendige Voraussetzung. E-Government kann ohne eine Infrastruktur für die sichere, zuverlässige und interoperable Übermittlung von Nachrichten nicht funktionieren. Die in diesem Kontext entstehenden Fragestellungen sind fachunabhängig oder zumindest fachübergreifend und insoweit im Zuständigkeitsbereich des IT-Planungsrats zu behandeln.

In diesem Abschnitt werden die Ziele von XTA 2 und die Rahmenbedingungen seiner Entwicklung erläutert und in Beziehung zur OSCI-Infrastruktur gesetzt.

Zu den Lösungen des IT-Planungsrats im Kontext der sicheren Datenübermittlung gehört der Standard OSCI-Transport. Er ist entwickelt worden, um sichere und zuverlässige Datenübermittlungen über das grundsätzlich unsichere Internet zu erlauben. Die Nutzung dieses Standards generiert auch in sicheren Netzen Mehrwerte, wie zum Beispiel Ende–zu–Ende Sicherheit und -Adressierung, oder auch Nachweise zur Integrität der Nachrichten, die durch die Netzebene allein nicht abgedeckt werden.

Im Rahmen der Umsetzung dieses Standards auf allen Verwaltungsebenen und in verschiedenen fachlichen Bereichen ist eine OSCI-Infrastruktur entstanden, der auch Komponenten wie das DVDV zuzurechnen sind. Durch die Vielzahl der Einsatzgebiete und durch wechselnde rechtliche Rahmenbedingungen sind die Anforderungen an den Schnittstellen zwischen den Fachverfahren der öffentlichen Verwaltung und der Transportinfrastruktur der öffentlichen Verwaltung stark gestiegen.

Es hat sich gezeigt, dass in manchen komplexen E-Government-Anwendungen eine Aufteilung zwischen fachlichen IT-Verfahren und einer spezialisierten Transport-Infrastruktur sinnvoll sein kann. In solchen Fällen wird eine standardkonforme Kommunikation zumeist nur zwischen den beauftragten Rechenzentren der öffentlichen Verwaltung gewährleistet, während die Kommunikation zwischen den Transportverfahren und den Fachanwendungen über proprietäre Schnittstellen erfolgt. Dies führt zu erhöhten Aufwendungen bei den Beteiligten und bei den Herstellern überregionaler Fachanwendungen, da diese unterschiedliche Schnittstellen unterstützen müssen. Auch kann eine datenschutzgerechte Umsetzung der Kommunikation zwischen den Fachverfahren aufgrund dieser individuellen Schnittstellen nicht einheitlich umgesetzt werden.

Diese Lücken sollen durch XTA 2 geschlossen werden, indem die (funktionalen) Schnittstellen zwischen Fach- und Transportverfahren definiert werden (siehe Kapitel 5, *XTA Webservice (2.1.1)*) und indem durch die Service Profile ein Instrument zur Verfügung gestellt wird, durch das die (modalen) Anforderungen an den Transport einheitlich definiert und konfiguriert werden können (siehe Kapitel 4, *XTA Service Profile (1.1)*).

1.1.1 Interoperabilität der Datenübermittlung im E-Government

Erfolgreiche Projekte des E-Government zeichnen sich dadurch aus, dass gemeinsame Verfahrensweisen und Technologien vereinbart und durch Vorgabe offener Standards flächendeckend durchgesetzt werden können. So kann die für den reibungslosen Datenaustausch notwendige Interoperabilität bei allen betroffenen IT-Verfahren gewährleistet werden.

In den so entstehenden Informationsverbünden können stets drei Ebenen unterschieden werden.

Netzebene
: Die *physische* Übermittlung von Daten im Kontext des E-Government kann über verwaltungseigene Netze, zu denen auch das Verbindungsnetz gemäß IT-NetzG gehört, oder über das Internet erfolgen. Die Interoperabilität der Datenübermittlung auf Netzebene ist auch ohne besondere Vorgaben des Bundes oder der Länder gewährleistet, weil sowohl das Internet als auch die verwaltungseigenen Netze die gleichen Industriestandards umsetzen.

Transportebene
: Aus der Existenz elektronischer Netze folgt noch nicht die Erreichbarkeit der daran angeschlossenen IT-Verfahren. Es bedarf weitergehender Regelungen zum technischen Transport der Nachrichten.

Dies betrifft Fragenstellungen der Adressierung auf Basis verwaltungseigener Verzeichnisdienste, der zuverlässigen Zustellung, die Gewährleistung von Authentizität und Integrität sowie die Behandlung von Fehlern. Der Transportebene sind auch Mechanismen zuzuordnen, die eine sichere Aufbewahrung von Nachrichten für den Fall realisieren, dass Empfänger nicht jederzeit erreichbar sind.

Anwendungsebene

Durch die Vorgabe von Fachstandards wird sichergestellt, dass der Inhalt der übermittelten Nachrichten von allen beteiligten IT-Verfahren gleich interpretiert wird. So wird beispielsweise durch den Fachstandard OSCI-XMeld gleichsam eine gemeinsame Sprache für Sachverhalte des Meldewesens vereinbart, die jedes IT-Verfahren im Meldewesen verstehen muss.

Wenn in diesem Sinne - wie üblich - drei Ebenen unterschieden sind, kann der Standard XTA 2 zugeordnet werden: Er adressiert die Fragestellungen der *Transportebene*.

1.1.2 Sicherer und zuverlässiger Nachrichtentransport

Die OSCI-nutzenden Informationsverbünde sind flächendeckend und auf allen drei Verwaltungsebenen im Einsatz. In den Informationsverbünden, für die XTA 2 relevant ist, werden jährlich ca. 100 Mio. Nachrichten zwischen insgesamt über 100.000 Kommunikationspartnern ausgetauscht:

Die OSCI-Infrastruktur der Innenverwaltung deckt das Melde-, Ausländer- und das Personenstandswesen ab. In der Justiz wird OSCI-Transport in besonders starkem Maße auch für die Datenübermittlung mit Kommunikationspartnern außerhalb der öffentlichen Verwaltung genutzt.

Eine wesentliche Erkenntnis aus der Praxis ist, dass die Organisation des zuverlässigen und sicheren Nachrichtentransports eine sehr komplexe Aufgabe ist:

- Es müssen elektronische Adressen aus Verzeichnisdiensten ermittelt, Sicherheitsmechanismen konfiguriert und der Versand nachverfolgt werden;
- Störungen, wie der Verlust einer Nachricht müssen erkannt und behoben werden;
- es sind Fristen zu überwachen und Eskalationsmechanismen zu bestimmen;
- in Übereinstimmung mit den rechtlichen Rahmenbedingungen muss festgelegt werden, ob Daten über verwaltungseigene Netze oder über das Internet zu transportieren sind.

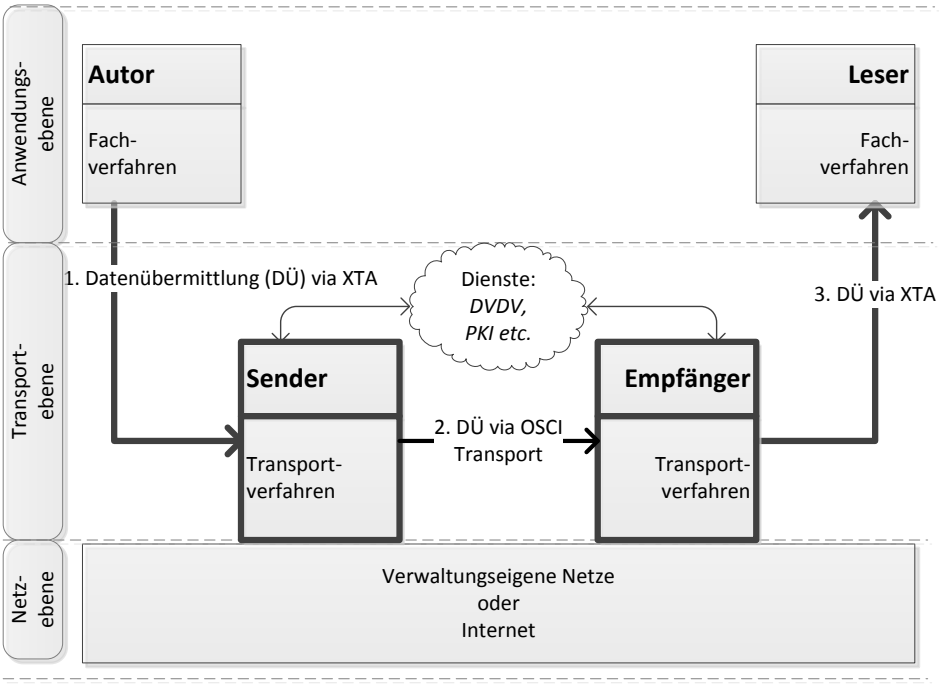
Diese Aufgaben werden wegen ihrer Komplexität häufig aus den IT-Fachverfahren ausgelagert und an spezialisierte IT-Transportverfahren oder an eigene Organisationseinheiten delegiert.

Die Aufgabe des Transports elektronischer Nachrichten ist hierbei vergleichbar mit dem Transport von Briefen oder Paketen, die an große, leistungsfähige Organisationseinheiten wie die Deutsche Post oder Paketdienste ausgelagert worden sind.

Das bedeutet, dass es neben den fachlich zuständigen Stellen, die für die fachlich korrekte Erstellung und Verarbeitung der Nachrichten zuständig sind, *Transporteure* gibt, deren Aufgabe darin besteht, Nachrichten unabhängig von deren Inhalt entsprechend der jeweiligen Rahmenbedingungen sicher und zuverlässig zu transportieren. Die in vielen Bundesländern eingerichteten Clearing- oder Vermittlungsstellen sind Transporteure. Sie betreiben für viele angeschlossene Kommunen Transportverfahren für unterschiedliche XÖV-Standards. Daneben gibt es auf kommunaler Ebene Softwareprodukte, die jeweils einem Fachverfahren vorgeschaltet oder in dieses integriert sind.

Dieses in Abbildung 1.2, „Rollenmodell in der OSCI Infrastruktur, der Regelungsgegenstand des Standards XTA 2“ dargestellte Modell hat sich in den vergangenen Jahren prinzipiell als praxistauglich und effizient erwiesen. In der Abbildung wird der Regelungsgegenstand von XTA 2 gezeigt: Es geht um die Vereinheitlichung der Schnittstellen zwischen IT-Fachverfahren (Autor bzw. Leser einer Nachricht) und den Transporteuren (Sender bzw. Empfänger einer Nachricht).

Abbildung 1.2. Rollenmodell in der OSCI Infrastruktur, der Regelungsgegenstand des Standards XTA 2



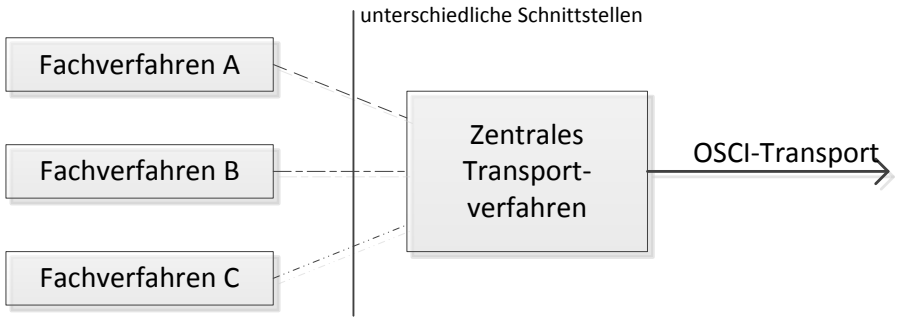
Derzeit sind die Schnittstellen zwischen den Fach- und den Transportverfahren nicht standardisiert, was auf beiden Seiten zu erheblichen Unsicherheiten und Mehraufwänden führt.

Die derzeit aufgrund fehlender Standards auf beiden Seiten entstehenden Aufwände werden nachfolgend dargestellt.

1.1.2.1 Aufwand auf Seiten zentral betriebener Transportverfahren

Insbesondere die Betreiber der in vielen Ländern eingerichteten, zentralen Transportverfahren (z.B. bei Clearingstellen) sehen sich mit vielen unterschiedlichen Schnittstellen der auf kommunaler Ebene betriebenen Fachverfahren konfrontiert, wie in Abbildung 1.3, „Zentrale Transportverfahren: Heterogene Anbindung von kommunalen Fachverfahren“ dargestellt. Dies führt zu hohen Aufwänden und Kosten auf Seiten der Transportverfahren.

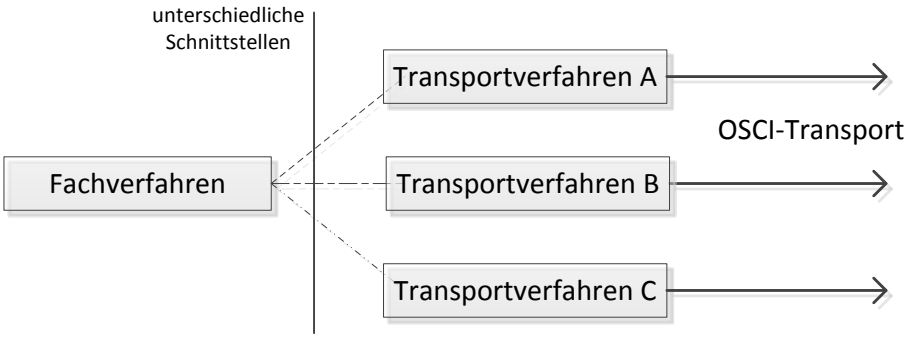
Abbildung 1.3. Zentrale Transportverfahren: Heterogene Anbindung von kommunalen Fachverfahren



1.1.2.2 Aufwand auf Seiten überregional eingesetzter Fachverfahren

Eine ähnliche Situation ergibt sich für Hersteller überregional eingesetzter Fachverfahren. Diese müssen häufig an die in den Ländern befindlichen Transportverfahren angebunden werden. Sofern diese Schnittstellen nicht vereinheitlicht sind, ist das Fachverfahren um die Anbindung an das entsprechende Transportverfahren des jeweiligen Landes zu erweitern. Dargestellt ist dieser Effekt in Abbildung 1.4, „Fachverfahren: Heterogene Anbindung von Transportverfahren in den Ländern“. Dies führt zu hohen Aufwänden und Kosten auf Seiten der überregional eingesetzten Fachverfahren.

Abbildung 1.4. Fachverfahren: Heterogene Anbindung von Transportverfahren in den Ländern



1.1.2.3 Ziele von XTA 2

Um die dargestellte Situation zu verbessern, soll die Anbindung von Fachverfahren an Transportverfahren vereinheitlicht werden:

Es werden einheitliche Vorgaben für Transportverfahren mit ihren Schnittstellen definiert. Es wird definiert, welche Funktionen jedes konforme Transportverfahren bieten muss. Zusätzlich wird ein Werkzeug zur Verfügung gestellt, durch das die jeweiligen Vorgaben für den Transport einheitlich umgesetzt werden können. Hersteller von Fachverfahren erhalten so eine verlässliche Aussage dahingehend, welche Funktionen sie auslagern (delegieren) können und sie erhalten ein performantes Werkzeug zur Umsetzung.

Darüber hinaus wird das technische Format der Schnittstelle zwischen Fach- und Transportverfahren festgelegt.

Damit wird durch XTA 2 bestimmt, was ein Transportverfahren ist und wie dieses an Fachverfahren anzubinden ist. Dies ist in Abbildung 1.2, „Rollenmodell in der OSCI Infrastruktur, der Regelungsgegenstand des Standards XTA 2“ gekennzeichnet.

So sollen Aufwände und Kosten reduziert werden, die derzeit aufgrund uneinheitlicher Funktionalitäten und Schnittstellen bei den Herstellern und Betreibern sowohl von Fach- als auch von Transportverfahren entstehen.

Zusammenfassend kann das Ziel folgendermaßen formuliert und präzisiert werden:

Durch den Standard XTA 2 werden die Voraussetzungen dafür geschaffen, dass auf der gesamten Strecke des Datenaustausches zwischen Fachverfahren (also zwischen den beteiligten Fachbehörden) die Anforderungen der Verwaltung durch die Verwaltung definiert und verbindlich vorgegeben werden können.

- Dies gilt für alle Kontexte, in denen Fachverfahren im Dienste der Verwaltung kommunizieren: länderübergreifend, landesintern und auch zwischen Land und Bund.

- Die Anforderungen, von denen hier die Rede ist, sind sehr umfassend zu verstehen. Sie betreffen nicht nur die Umsetzung der entsprechenden Prozesse, sondern auch deren Leistungsfähigkeit sowie die Gesichtspunkte von Datensicherheit und Datenschutz.
- Wenn gesagt wird, dass Anforderungen definiert und verbindlich vorgegeben werden, so ist darin eingeschlossen, dass die Einhaltung dieser Vorgaben auch überprüft werden können.
- Und noch eine *Abgrenzung*: Es ist *nicht* das Ziel, Vorgaben für die konkrete Verfügbarkeit der Transportinfrastruktur innerhalb der Länder festzuschreiben oder übergreifende Service Level Agreements inhaltlich zu definieren.

1.1.2.4 Bezug zum Standard OSCI-Transport 1.2

OSCI-Transport hat seine Eignung für den zuverlässigen und sicheren Transport elektronischer Nachrichten im Kontext des E-Government hinreichend unter Beweis gestellt. Es bleibt ein unverzichtbarer Bestandteil der seit Jahren erfolgreich betriebenen Infrastruktur. Ebenso unverzichtbar ist die konzeptionelle Trennung zwischen Fach- und Transportaufgaben, die in der Praxis häufig durch dedizierte Transportverfahren oder Organisationseinheiten (Clearingstellen) umgesetzt wird. XTA 2 ergänzt die OSCI-Infrastruktur dahingehend, dass die zwischen Transportverfahren erreichte Standardisierung bis zu den Fachverfahren verlängert wird.

Die im Standard XTA 2 definierte Schnittstelle zwischen Fach- und Transportverfahren wird als XTA-Webservice (XTA-WS) bezeichnet. Er soll die derzeit in Betrieb befindlichen, proprietären Anbindungen von Fach- an Transportverfahren ersetzen. Der XTA-WS ist kein Ersatz des zwischen Transportverfahren eingesetzten Standards OSCI-Transport 1.2, sondern eine Ergänzung.

1.2 Überblick über den Inhalt des Standards XTA 2

Die XTA-Spezifikation besteht aus verpflichtenden und optionalen Teilen. Verpflichtende Teile müssen vollständig umgesetzt werden. Sie stellen den Regelfall dar und werden daher nicht besonders gekennzeichnet. Optionale Teile stellen eine Ausnahme dar, sie müssen nicht verpflichtend umgesetzt werden und werden explizit durch die Bezeichnung „Optionaler Teil“ in der Abschnittsüberschrift gekennzeichnet. Alle Teile der XTA-Spezifikation, die nicht explizit als optionaler Teil gekennzeichnet sind, müssen dementsprechend verpflichtend umgesetzt werden.

Der Standard XTA 2 besteht aus folgenden Modulen:

Modell der Rollen und Verantwortlichkeiten ("Rollenmodell"): Verantwortlichkeiten und Aufgaben im Transport

Das Modell der Rollen und Verantwortlichkeiten bildet das Fundament im Standard XTA 2, in dem es die Aufgaben und Verantwortlichkeiten aller Akteure, die vom Transport von Fachdaten berührt sind, allgemeingültig beschreibt. Dies betrifft die Aufgaben und Verantwortlichkeiten:

1. der Behörden, die in den IT-Fachverfahren die Fachdaten erstellen und sie für den Transport zur Verfügung stellen („Autoren“);
2. der Vermittlungsstellen (auch Clearingstellen oder Nachrichtenbroker genannt), die die Daten von den Behörden entgegen nehmen und sie entsprechend der rechtlichen und fachlichen Vorgaben aufbereiten und versenden („Sender“);
3. der Vermittlungsstellen auf der Gegenseite, die die Nachrichten vom Sender entgegen nehmen („Empfänger“);
4. und schließlich die Aufgaben und Verantwortlichkeiten der Behörden, an die die Fachdaten adressiert wurden („Leser“) und die diese verarbeiten.

Hierbei ist es in der konkreten Ausgestaltung grundsätzlich denkbar, dass einzelne Rollen zusammenfallen.

Die Betrachtung des Gesamtszenarios führte zu einer Zusammenstellung von ca. 40 "Sätzen", in denen jeweils die einzelnen Verantwortlichkeiten und Aufgaben definiert werden. Sie werden in XTA 2 als *Rollenmodell* bezeichnet.

Service Profile: Parametrisierung der Anforderungen an den Transport

Die Service Profile sind ein Werkzeug für die Standardisierung der Anforderungen an den Transport. Durch ein Baukastenprinzip mit vorgefertigten Blaupausen soll erreicht werden, dass in den unterschiedlichen fachlichen Kontexten in vergleichbaren Einsatzszenarien mit gleichen Transportanforderungen die Umsetzung dieser Anforderungen auf dieselbe Weise erfolgt und so überprüfbar wird.

Inhaltlich ist dies keine neue Aufgabe für die einzelnen fachlichen Bereiche. Durch die Nutzung der zentral abgelegten Blaupausen kann sich jedoch zukünftig der Aufwand bei der Definition eines Kommunikationsszenarios reduzieren.

XTA-WS: Einheitliche Webserviceschnittstellen zwischen Fachverfahren und Transportverfahren

Während die Service Profile die *Ausgestaltung*, die Art und Weise, von definierten Qualitäten eines Transports standardisieren und damit die modale Steuerung unterstützen, bietet das Modul XTA-WS eine Unterstützung für die funktionale Steuerung. Die Webserviceschnittstellen zwischen Fachverfahren und Transportverfahren wurden als OSCI2-Profilierung umgesetzt. In der Spezifikation sind die Anforderungen an eine XÖV-konforme Dokumentation berücksichtigt.

Für die Webserviceschnittstellen wurde der sukzessive Abruf von Nachrichten aus dem Postfach als optionaler Teil der XTA-Spezifikation definiert.

1.3 Auslieferungsumfang des Standards

Der Standard besteht aus einer Reihe von Komponenten, die von der KoSIT gleichzeitig zur Release-Freigabe bereitgestellt werden:

- **Spezifikation:** Die Spezifikation (dieses Dokument) steht im PDF-Format zur Verfügung.
- **WSDL-Dateien:** Zum XTA Webservice werden die nötigen WSDL-Dateien bereitgestellt.
- **Schema-Dateien:** Die Datenstrukturen zu den XTA Service Profilen sowie zum XTA-Webservice werden als XML-Schemata zur Verfügung gestellt (inhaltlich identisch zur Darstellung im Spezifikationsdokument).
- **Codelisten:** Die in XTA 2 definierten Codelisten sind als XML-Instanzen im Format OASIS-Genericode verfügbar.

1.4 Überblick über das Dokument

Das vorliegende Dokument enthält die überarbeitete und ergänzte Fassung des Standards XTA 2, der seit November 2013 im Auftrag des IT-Planungsrats erarbeitet wird. Änderungen gegenüber der vorhergehenden Version 2.0 sind der Versionshistorie auf Seite 201 zu entnehmen.

In Kapitel 1, *Einleitung*, sind Motivation und Ziele des Standards XTA 2 beschrieben, sowie seine Bestandteile benannt.

In Kapitel 2, *Allgemeines*, sind übergreifende Themen behandelt: Zunächst werden grundlegende Begriffe eingeführt, die an vielen Stellen im vorliegenden Dokument verwendet werden. Die Verwendung von Quittungsmechanismen in XTA 2 und die Einbindung externer Standards ist hier, weil kapitelübergreifend relevant, ebenfalls aufgenommen.

In Kapitel 3, *Kooperation beim Datenaustausch: Anwendungsfälle*, wird aus der Vogelperspektive ein Blick auf die Prozesse von Beauftragung und Durchführung des Transports von Nachrichten geworfen: Die Aufgaben und Verantwortlichkeiten der beteiligten Akteure werden in Form von Anwendungsfällen

(Use Cases) beschrieben und visualisiert. In diesen Anwendungsfällen sind organisatorische und technische Aspekte berücksichtigt, und ihre Beschreibungen bilden die inhaltliche Grundlage für die weiteren Kapitel. Eine alternative Darstellung des "Modells der Rollen und Verantwortlichkeiten" in Form von Sätzen ist in Abschnitt 2.2 auf Seite 14 enthalten.

In Kapitel 4, *XTA Service Profile (1.1)*, wird das Instrument der Service Profile beschrieben, mit dessen Hilfe die Anforderungen von Verwaltung und Justiz an die Transport- und Fachverfahren für die unterschiedlichen Kommunikationsszenarien sinnvoll gebündelt werden können. Auf diese Weise lassen sich die Leistungen der Transportinfrastruktur durch deren Auftraggeber einheitlich steuern und überprüfen. Für die Umsetzung der XTA Service Profile wird zusätzlich auf die angelegten XSD-Dateien verwiesen, die als technisches Hilfsmittel für die Erstellung von Profilobjekten dienen sollen.

In Kapitel 5, *XTA Webservice (2.1.1)*, ist die Spezifikation des XTA-WS enthalten, der für Fachverfahren einen einheitlichen technischen Zugang zu Leistungen der Transportverfahren bietet. Um die inhaltliche Verbindung zur Darstellung der Anwendungsfälle auf Seite 27 herzustellen und die Herleitung der Methoden und Funktionen deutlich zu machen, wurden Beispielszenarien aufgenommen. Für die Umsetzung des XTA-WS wird zusätzlich auf die angelegten WSDL/XSD-Dateien verwiesen, die ihn als technische Artefakte spezifizieren.

2 Allgemeines

2.1 Grundlegende Begriffe

2.1.1 Definitionen zu IT-Verfahren

2.1.1.1 Transportverfahren

Als Transportverfahren werden in einer XTA-Infrastruktur IT-Verfahren bezeichnet, die in der Lage sind, Nachrichten zu versenden und zu empfangen oder auch an sonstigen Aspekten der Übermittlung mitzuwirken – unabhängig davon, welcher Fachdomäne der Inhalt der Nachricht angehört.

2.1.1.2 Fachverfahren

Fachverfahren sind im Sinne der XTA-Spezifikation die IT-Verfahren, die in Behörden für die Vorgangsbearbeitung der jeweiligen Fachdomäne (z.B. Personenstandswesen, Pass- und Ausweisbehörde) eingesetzt werden.

2.1.2 Definitionen zu Nachrichtenstrukturen

2.1.2.1 XÖV-Nachricht

„XÖV-Nachricht“ bezeichnet eine XML-Instanz oder einen entsprechenden Nachrichtentyp gemäß der Spezifikation eines XÖV-Fachstandards. Die XÖV-Standards der Innenverwaltung definieren ihre Nachrichten bestehend aus Nachrichtenkopf (Angaben zum Autor, Angaben zum Leser, Erstellungszeitpunkt dieser Nachricht, Identifikation dieser Nachricht) und Fachdaten. Eine XÖV-Nachricht ist im Kontext der Nachrichtenübermittlung eine Fachnachricht. Als solche kann sie im Payload einer Transportnachricht Gegenstand der Nachrichtenübermittlung sein.

2.1.2.2 Transportstandard

In einer XTA-Infrastruktur werden als Transportstandards Technologien bezeichnet, die sich damit befassen, wie eine Nachricht von A nach B zu *übermitteln* ist (im Gegensatz zu Fachstandards, die sich mit Struktur und Inhalt der Nachrichten befassen). Zu einem Transportstandard gehört eine Aussage zum Format der Transportnachrichten (z. B. ein auf SOAP basierendes Format, vgl. Abschnitt A.2.15 auf Seite 178) und zum Protokoll der Kommunikation (z.B. HTTP, vgl. Abschnitt A.2.16 auf Seite 179).

2.1.2.3 Transportauftrag

Der Transportauftrag enthält alle erforderlichen Angaben, um die Daten gemäß der Intention des Autors zu den Adressaten zu transportieren. Als Ersteller des Transportauftrags muss der Autor sicherstellen,

dass alle Informationen, die sowohl im Transportauftrag als auch in der Fachnachricht enthalten sind, konsistent sind. Über den Transportauftrag wird auch beispielsweise die Qualität der Protokollierung der beteiligten Systeme durch die Angabe des anzuwendenden Service Profils gesteuert. Jeder Transportauftrag ist eindeutig identifizierbar. Der Transportauftrag wird durch das Objekt *MessageMetaData* im XTA Webservice repräsentiert (vgl. Abschnitt 5.4.2.3.1 auf Seite 122). Vgl. auch Abschnitt 3.3.4 auf Seite 46.

2.1.2.4 Payload

Der Payload ist der fachliche Inhalt der Transportnachricht, der vom Autor für den Leser erstellt wird. Er umfasst die Gesamtheit der vom Autor für den Leser bestimmten Informationen. Oft bildet eine Fachnachricht den Payload einer Transportnachricht.

Der Payload kann vom Autor für den Leser verschlüsselt werden. Deswegen muss der Sender seine Aufgaben mit ausschließlicher Kenntnis des Transportauftrags (ohne Payload) erfüllen können.

2.1.2.5 Nachricht

Eine Nachricht kann entweder eine Transportnachricht oder eine Fachnachricht sein. Der Terminus „Nachricht“ kann auch unbestimmt verwendet werden.

2.1.2.5.1 Transportnachricht

Die *Transportnachricht* besteht aus dem Transportauftrag mit dem zugehörigen Payload (vgl. Abschnitt 3.3.1 auf Seite 45).

2.1.2.5.2 Fachnachricht

Die *Fachnachricht* ist ein Informationsobjekt, welches vom Autor für den Leser erstellt worden ist. Beispielsweise sind XÖV-Nachrichten Fachnachrichten.

Fachnachrichten können im Rahmen einer Datenübermittlung als Payload Bestandteil der Transportnachricht sein.

2.1.2.6 SOAP

Format für die Request-Response-Kommunikation von Webservices. Vielfach als Grundlage des Formats für Transportnachrichten verwendet (vgl. Abschnitt A.2.15 auf Seite 178). Eine SOAP-Message hat eine XML-Struktur und ist unterteilt in Header und Body. Der Body ist dafür gedacht, den zu transportierenden Content – den Payload – aufzunehmen. Der Header nimmt die Nutzungsdaten auf.

2.1.2.7 Webservice

Ein Webservice ist ein in der Sprache WSDL definierter Service, der z. B. von Remote-Systemen per SOAP-Protokoll angesprochen werden kann. Er funktioniert dann als Request / Response von SOAP-Messages.

2.1.3 Begriffe zu Datenschutz und Datensicherheit

2.1.3.1 Integrität

Sicherstellung der Korrektheit (Unversehrtheit) von Daten und der korrekten Funktionsweise von Systemen. Integrität meint, dass die Daten vollständig und unverändert sind.

In der Informationstechnik wird der Begriff in der Regel aber weiter gefasst und auf "Informationen" angewendet. Der Begriff "Information" wird dabei für "Daten" verwendet, denen je nach Zusammenhang

eine bestimmte Semantik wie z. B. Autor oder Zeitpunkt der Erstellung zugeordnet werden kann. Der Verlust der Integrität von Informationen kann daher bedeuten, dass diese unerlaubt verändert, Angaben zum Autor verfälscht oder Zeitangaben zur Erstellung manipuliert wurden. Die Integrität ist Schutzziel der IT-Sicherheit.

Integritätssicherung auf der Transportstrecke meint nicht die nachweisbare Abgabe einer Willenserklärung, sondern die mathematische Nachprüfbarkeit der Unverfälschtheit der Nachricht.

2.1.3.2 Intervenierbarkeit

Durch Intervenierbarkeit wird sichergestellt, dass Betroffene bzw. Akteure mit legitimierten Eingriffsbefugnissen auf ein Verfahren mit Personenbezug wirksam eingreifen und bestehende Verfahren ändern können. Die Intervenierbarkeit ist Schutzziel des Datenschutzes.

Weil es sich bei XTA 2 um eine Infrastruktur handelt, die nur mittelbar in einer Beziehung zum betroffenen Bürger steht, stehen Maßnahmen zur Steuerung der Infrastruktur und Maßnahmen zur Steuerung des unmittelbaren Workflows des Kommunikationsvollzugs im Vordergrund.

2.1.3.3 Nichtverkettbarkeit

Durch die Nichtverkettbarkeit wird sichergestellt, dass die Datenverarbeitung bzw. der Transport von Nachrichten der Zweckbestimmung des Verfahrens insgesamt folgt. Es soll eine vorsätzliche oder fahrlässige oder funktional-fehlerhafte Datenverarbeitung, die ein Risiko für die Einhaltung der Zweckbindung darstellt, wesentlich erschwert werden.

Die Nichtverkettbarkeit im Kontext von IT-Infrastrukturen wird insbesondere durch organisatorische – und nicht technische - Maßnahmen erreicht. Die Nichtverkettbarkeit ist Schutzziel des Datenschutzes.

2.1.3.4 Transparenz

Es soll erreicht werden, dass alle Beteiligten nachweisen können, dass sie den rechtlichen Anforderungen (gemäß des anzuwendenden Schutzprofils) genügt haben. Transparenz soll die Prüffähigkeit des gesamten Verfahrens sowie einzelner Komponenten sicherstellen. Die Transparenz ist Schutzziel des Datenschutzes.

Der Nachweis der Rechtskonformität bzw. die Prüfbarkeit ist für folgende Bereiche relevant:

- im Binnenverhältnis Autor-Sender und Leser-Empfänger
- für die funktionale Aufsicht bzgl. der deutschlandweiten (europaweiten) Sicherstellung von Interoperabilität und Effizienz durch den IT-Planungsrat
- für die Prüfbarkeit von Informationssicherheit und Datenschutz durch die Aufsichtsbehörden.

Die Sicherung der Transparenz ist Voraussetzung für ein Qualitätsmanagement durch den Auftraggeber (IT-Planungsrat), zu der auch eine externe Auditierung des Verfahrens zählen kann.

2.1.3.5 Verfügbarkeit

Die Verfügbarkeit von Dienstleistungen und Funktionen eines IT-Systems, von IT-Anwendungen oder von IT-Netzen oder auch die Verfügbarkeit von Informationen ist gegeben, wenn diese von den Anwendern stets wie vorgesehen genutzt werden können. Die Verfügbarkeit ist Schutzziel der IT-Sicherheit.

2.1.3.6 Vertraulichkeit

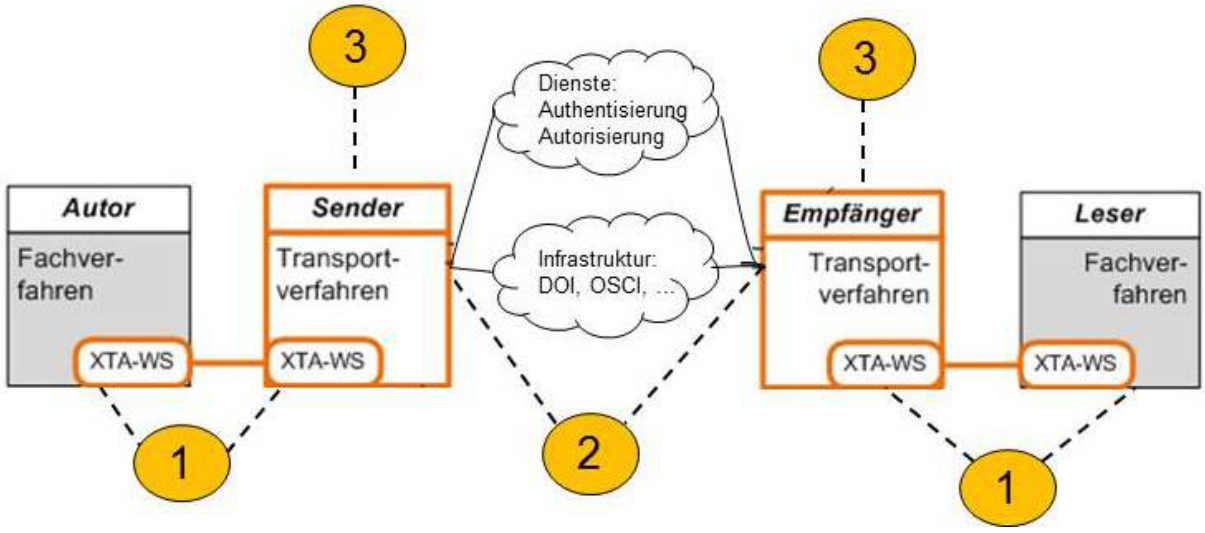
Vertraulichkeit ist der Schutz vor unbefugter Preisgabe von Informationen. Vertrauliche Daten und Informationen dürfen ausschließlich Befugten in der zulässigen Weise zugänglich sein. Die Vertraulichkeit ist Schutzziel der IT-Sicherheit.

2.2 Modell der Rollen und Verantwortlichkeiten

2.2.1 Überblick

Die Infrastruktur im XTA-Kontext lässt sich als ein System aus Fach- und Transportverfahren darstellen, die bei den Prozessen der Nachrichtenübermittlung kooperieren. In der nachfolgenden Abbildung wird sie dargestellt als Interaktion von vier Rollen mit zwei Typen von Schnittstellen (1 und 2). Die Transportverfahren (3) sind an beiden Schnittstellen beteiligt.

Abbildung 2.1. Infrastruktur der Nachrichtenübermittlung: Kooperation von Fach- und Transportverfahren



Von solchen Gegebenheiten der Implementierung soll in dem vorliegenden Kapitel abstrahiert werden, wenn die Infrastruktur aufgeteilt in Rollen analysiert wird. Die Rollen *Autor* und *Leser* sind dabei der Infrastrukturkomponente Fachverfahren zugeordnet, die Rollen Sender und Empfänger der Infrastrukturkomponente Transportverfahren .

Der vorliegende Abschnitt charakterisiert die Rollen und grenzt sie voneinander ab, wie das durch die an der XTA-Standardisierung beteiligten Organisationen abgestimmt worden ist.

Die Definition und Abgrenzung der Rollen geschieht in Form von *Sätzen*. Im Abschnitt 2.2.2.1 auf Seite 14 werden zu jeder der vier Rollen Sätze formuliert, welche entsprechende Festlegungen treffen, also normativ zu verstehen sind.

Die Sätze sind zu den einzelnen Themenbereichen gruppiert. Zu einigen Sätzen sind für ein besseres Verständnis Erläuterungen oder Anmerkungen hinzugefügt.

2.2.2 Die Rollen

2.2.2.1 Der Autor

2.2.2.1.1 Aufgabe des Autors

A 1.1 Der Autor ist fachlich zuständig, d.h. er ist für den Inhalt der zu transportierenden Nachricht, also die Fachnachricht, verantwortlich.

A 1.2 Der Autor erstellt die Fachnachricht gemäß den Regeln des zu Grunde liegenden Standards (z.B. OSCI-XMeld) in einer bestimmten Version.

Anmerkung:

- *Der vollständige Inhalt der vom Autor erstellten Nachricht (der Fachnachricht) ist für den Leser relevant. Und alles, was für den Leser relevant ist, sollte in der Nachricht enthalten sein. Dies betrifft auch die Informationen, die im Nachrichtenkopf einer XÖV-Nachricht (vergleichbar dem Inhalt eines Briefkopfes), enthalten sind, wie z.B. der AGS von Autor und Leser sowie die Nachrichten-Identifikation.*

A 1.3 Der Autor ist verantwortlich dafür, dass die Fachnachricht entsprechend spezifikationskonform ist. Das schließt ein, dass sie valide bezüglich des für den Standard (in der entsprechenden Version) gültigen Schemas ist. Der Autor kann, bei entsprechender vertraglicher Regelung, die Ausführung der Schemavalidierung an den Sender delegieren.

2.2.2.1.2 Zuständigkeitsprüfung des Lesers durch den Autor

A 2.1 Der Autor ist für die fachliche Adressierung des Lesers zuständig.

A 2.2 Der Autor kann prüfen, ob der Leser in einem bestimmten fachlichen Kontext elektronisch erreichbar (über beispielsweise DVDV oder SAFE) ist. (Hiermit ist nicht die Prüfung gemeint, ob der Leser aktuell verfügbar ist.) Hierbei ist der Sender einbezogen bzw. er stellt eine entsprechende Funktionalität zur Verfügung. Diese Prüfung erfolgt durch qualitätsgesicherte Verzeichnisse der öffentlichen Verwaltung.

A 2.3 Der Autor muss benötigte Attribute für die elektronische Kommunkation mit dem Leser abrufen können, sofern dies im fachlichen Kontext notwendig ist. Hierbei ist der Sender einbezogen bzw. er stellt eine entsprechende Funktionalität zur Verfügung.

2.2.2.1.3 Signatur

A 3.1 Der Autor kann die Fachnachricht oder Teile von dieser signieren.

Anmerkungen:

- *Die XhD-Spezifikation ist ein Beispiel, in der Teile einer Fachnachricht signiert werden.*

A 3.2 Der Autor ist zuständig für die Signatur der Fachnachricht, nicht der Sender. Ein Sender kann im Auftrag des Autors diese Aufgabe wahrnehmen. Der Autor muss immer über die Signatur identifizierbar bleiben.

2.2.2.1.4 Verschlüsselung

A 4.1 Der Autor kann die Fachnachricht oder Teile von dieser verschlüsseln.

A 4.2 Der Autor ist zuständig für die Verschlüsselung der Fachnachricht, nicht der Sender. Ein Sender kann im Auftrag des Autors diese Aufgabe wahrnehmen.

2.2.2.1.5 Kommunikationszenario

A 5.1 Die Wahl des Kommunikationsszenarios (synchron, asynchron) ist durch rechtliche Normen und Regeln in einer fachlichen Spezifikation vorgegeben.

A 5.2 Der Autor erteilt den Transportauftrag. Dabei übergibt er die Fachnachricht an den Sender.

A 5.3 Der Autor muss sicherstellen, dass er den korrekten Sender adressiert.

A 5.4 Der Autor muss die Konsistenz des Transportauftrags mit der Fachnachricht sicherstellen, insofern Informationen (z. B. Identität des Lesers) sowohl in der Fachnachricht als auch im Transportauftrag enthalten sein sollen.

2.2.2.1.6 Service Qualität

A 6.1 Der Autor wählt eine bestimmte Service Qualität aus den vom Sender unter Berücksichtigung der rechtlichen und fachlichen Vorgaben angebotenen Optionen aus.

Anmerkungen:

- *Die benötigte Service Qualität muss durch den Autor mit dem Sender vertraglich vereinbart sein.*
- *Beispiele für Service Qualitäten sind "Zeit bis Abschluss Geschäftsprozess" und "Anforderung von Transportquittungen".*
- *Ein weiterer Ausgangspunkt für die Definition von Service Qualitäten sind die fachlich festgelegten Schutzbedarfe (aus Vertraulichkeit, Integrität, Verfügbarkeit) und den daraus abgeleiteten Anforderungen Transparenz (Überprüfbarkeit), Intervenierbarkeit (Changemanagement) und Nichtverkettbarkeit.*

2.2.2.1.7 Eindeutige Identifizierung des Transportauftrages

A 7.1 Der Autor ist verantwortlich für die Erzeugung einer Identifizierung des Transportauftrags (MessageID).

Anmerkung:

- *Diese eindeutige Identifizierung eines Transportauftrags meint nicht die eindeutige Identifizierung einer Fachnachricht. Wenn eine Fachnachricht mehrfach versendet werden muss, erhält sie mit jedem Versand eine neue eindeutige Transport-Identifizierung.*
- *Diese eindeutige identifizierung soll bis zum Leser durchgereicht werden.*

A 7.2 Der Autor lässt die MessageID vor der Beauftragung des Transports durch den Sender erstellen, um sie dann für den Transport dem Sender zu übergeben.

A 7.3 Die MessageID ist im Transportauftrag enthalten. Die MessageID soll eine durchgehende ID für den gesamten Transportauftrag, für die vollständige Zustellungskette sein.

2.2.2.1.8 Überwachung der Übermittlung der Nachricht

A 8.1 Der Autor ist für die Überwachung der Übermittlung und die Einhaltung der (rechtlich- organisatorischen Vorgaben für) Übermittlungfristen der Fachnachricht an den Empfänger bzw. Leser zuständig.

A 8.2 Der Autor ist für das Zurückziehen eines offenen Transportauftrags zuständig.

Anmerkungen:

- *Ein Transportauftrag gilt als "offen", wenn der Sender den Auftrag noch nicht an den Empfänger weitergegeben hat. Der Transportauftrag befindet sich also noch beim Sender.*
- *Für das Zurückziehen stellt der Sender eine Funktionalität zur Verfügung.*

2.2.2.1.9 Aufbewahrung

A 9.1 Der Autor ist für die Aufbewahrung der versandten Nachrichten und der relevanten Transportinformation zuständig und dafür, dass fristgerecht gelöscht wird. Für die Aufbewahrung kann er sich eines Dienstleisters bedienen.

A 9.2 Der Autor legt fest, wie lange die Nachrichten und die Protokolle der Nutzungsdaten beim Sender gespeichert werden. Die Löschfristen werden vereinbart.

Anmerkungen:

- *Die Rahmenbedingungen hierfür werden durch rechtliche und fachliche Vorgaben gesetzt.*

2.2.2.1.10 Vertraglicher und rechtlicher Rahmen

A 10.1 Das rechtliche Verhältnis zwischen Autor und Sender muss geklärt sein.

2.2.2.2 Der Sender

2.2.2.2.1 Aufgabe des Senders

B 1.1 Der Sender ist gemäß Transportauftrag des Autors für die Abwicklung des Transports zuständig. Der Sender unterhält dafür die Infrastruktur und gibt dem Autor einen entsprechenden Zugang.

Anmerkungen:

- *Der Autor entscheidet über Eigenschaften bzgl. der Service Qualität. Hierbei müssen z.B. Vorgaben des Landes berücksichtigt werden. Der Sender muss nachweisen, dass er diese gewährleisten kann.*
- *Der Sender trägt die Verantwortung bei Entscheidungen, die das Transportprotokoll (HTTPS, OSCI-Transport,...) betreffen. Die Anforderungen werden im Rahmen der Service Qualität definiert.*

2.2.2.2.2 Prüfung der Identität des Autors durch den Sender

B 2.1 Der Sender ist für die Authentifizierung des Autors zuständig, d. h. er prüft die Identität des Autors.

Anmerkung:

- *Der Sender überprüft, ob ihm die Authentisierungsinformationen des Autors bekannt sind.*

B 2.2 Der Sender ist verpflichtet, die Angaben der Authentifizierung auf Konsistenz mit den Absenderangaben des Transportauftrags zu prüfen.

Anmerkung:

- *Durch diese Prüfung wird geklärt, ob die authentisierte Behörde in diesem Fachkontext mit dieser Behördenidentität (=z.B. ags:12343123 für Oldenburg im Meldewesen) auftreten darf.*

B 2.3 Abgrenzung: Der Sender ist nicht dafür zuständig, die fachliche Zuständigkeit des Autors für den Inhalt der Fachnachricht zu prüfen. Dies geschieht durch den Leser.

2.2.2.2.3 Kommunikation des Senders mit dem Leser

B 3.1 Der Sender prüft, ob für den Leser ein Zugang eröffnet ist. Der Sender ermittelt die technische Adresse des Lesers anhand dessen fachlicher Adresse. Er verwendet hierfür ein Verzeichnis wie DVDV oder S.A.F.E.

B 3.2 Der Sender stellt, falls nötig, dem Autor die technischen Attribute des Lesers zur Verfügung.

2.2.2.2.4 Transportkanal

B 4.1 Der Sender entscheidet über den zu nutzenden Transportkanal (technische Alternativen im Rahmen seiner Vereinbarung mit dem Autor).

Anmerkung: Der Sender entscheidet zwischen technischen Alternativen (z.B. landesinterne Infrastruktur, OSCI-DVDV-Infrastruktur), die in der Vereinbarung mit dem Autor als mögliche Alternativen für eine bestimmte Qualität (Quittungen, Zustellfristen, ...) benannt sind. Technisch erfolgt diese Vereinbarung über Service Profile (siehe Kapitel 4, XTA Service Profile (1.1)).

2.2.2.2.5 Verschlüsselung und Signatur

B 5.1 Der Sender bringt je nach Policy für den jeweiligen Transportkanal ggf. die Transportsignatur an.

B 5.2 Der Sender verschlüsselt die Nachricht je nach Policy für den jeweiligen Transportkanal ggf. für den Empfänger.

2.2.2.2.6 Zustellung durch den Sender

B 6.1 Der Sender führt den Transport der übergebenen Fachnachricht zum Empfänger durch.

2.2.2.2.7 Protokollierung durch den Sender

B 7.1 Der Sender erstellt Transportprotokolle und stellt sie dem Autor zur Verfügung. Der Sender hält die Transportprotokolle zu Nachweiszwecken vor.

B 7.2 Der Sender trägt Hindernisse für die Auftragserfüllung in die Protokolle ein.

2.2.2.2.8 Daten löschen

B 8.1 Der Sender löscht alle zum Transportauftrag gehörenden Daten, sobald sie nicht mehr benötigt werden.

2.2.2.2.9 Service Qualität

B 9.1 Der Sender muss den Transportauftrag entsprechend der Vorgaben des Autors behandeln.

2.2.2.2.10 Eindeutige Identifizierung des Transportauftrags

B 10.1 Der Sender ist verantwortlich dafür, die Identifizierung des Transportauftrags (MessageID) eindeutig zu erzeugen.

2.2.2.3 Der Empfänger

2.2.2.3.1 Aufgabe des Empfängers

C 1.1 Der Empfänger ist vom Leser mit der Entgegennahme von Nachrichten beauftragt.

Anmerkungen:

- *Ein Intermediär ist Bestandteil der Empfänger-Infrastruktur.*

C 1.2 Der Empfänger unterhält für die Entgegennahme von Nachrichten die Infrastruktur.

C 1.3 Der Empfänger stellt dem Leser die entgegengenommenen Nachrichten zur Verfügung.

2.2.2.3.2 Prüfung Identität Leser

C 2.1 Der Empfänger ist für die Authentifizierung des Lesers zuständig, d. h. er hat die Identität des Lesers zu prüfen.

Anmerkung:

- *Die Prüfung dient der Zugriffskontrolle im Kontext der Autorisierung nach Absprache mit dem Leser.*

C 2.2 Der Empfänger ist für die Prüfung zuständig, ob die Identität des Lesers (Authentisierung gegenüber dem Empfänger) konsistent ist mit der Identität des Lesers für die Fachkommunikation im Rahmen des Transportauftrags.

Anmerkungen:

- *Falls der Empfänger die Fachnachricht keinem Leser zuordnen kann, schickt er dem Sender eine Fehlermeldung (ggf. eine administrativ-technische RTS-Nachricht). Entsprechend kann der Erfolgsfall durch eine Quittung (auf Transportebene) bestätigt werden.*
- *Die Prüfung ist ein Aspekt der Service Qualität des Empfängers.*

C 2.3 Abgrenzung: Der Empfänger ist nicht dafür zuständig, die fachliche Zuständigkeit des Lesers für den Inhalt der Fachnachricht zu prüfen. (Dies geschieht durch den Leser).

2.2.2.3.3 Prüfung der Berechtigung des Autors durch den Empfänger

C 3.1 Der Empfänger prüft in Abhängigkeit vom Nutzungsszenrario, ob der Autor berechtigt ist, diese Fachnachricht zu senden.

C 3.2 Abgrenzung: Der Empfänger ist nicht zuständig, die fachliche Zuständigkeit des Autors für den Inhalt der Fachnachricht zu prüfen. (Das prüft der Leser).

2.2.2.3.4 Entschlüsselung und Signaturprüfung

C 4.1 Der Empfänger prüft je nach Policy für den jeweiligen Transportkanal ggf. die Transportsignatur.

C 4.2 Der Empfänger entschlüsselt je nach Policy für den jeweiligen Transportkanal ggf. die Transportverschlüsselung für den Leser.

2.2.2.3.5 Zustellung durch den Empfänger

C 5.1 Der Empfänger nimmt die transportierte Nachricht entgegen und stellt sie dem Leser zur Verfügung.

Anmerkungen:

- *Die Zustellung kann asynchron oder synchron erfolgen. Synchrone Kommunikation ist dadurch charakterisiert, dass der Prozess blockiert bis der Leser die Reaktion geliefert hat.*

C 5.2 Der Empfänger stellt die Empfangsquittung für den Sender / Autor zur Verfügung, falls diese Service Qualität vom Autor angefordert worden ist.

C 5.3 Fehlerbehandlungen und Ausnahmeregelungen erfolgen entsprechend der fachlichen Spezifikationen und der vertraglichen Vereinbarungen. Falls der Leser nicht ermittelt werden kann, erfolgt keine Zustellung. Dies wird protokolliert und der Autor erhält eine entsprechende Benachrichtigung.

2.2.2.3.6 Protokollierung durch den Empfänger

C 6.1 Der Empfänger erstellt Transportprotokolle und stellt diese dem Leser zur Verfügung. Er hält diese entsprechend der fachlichen und vertraglichen Vorgaben vor.

C 6.2 Der Empfänger trägt Hindernisse für die Auftragserfüllung in die Protokolle ein.

2.2.2.3.7 Daten löschen

C 7.1 Der Empfänger löscht alle zum Transportauftrag gehörenden Daten, sobald sie nicht mehr benötigt werden.

Anmerkung:

- *Im Rahmen der Definition der Schutzbedarfe wird festgelegt, wann wie sicher gelöscht werden muss.*

2.2.2.3.8 Service Qualität

C 8.1 Der Empfänger erfüllt die Verpflichtungen, die mit dem Leser vereinbart sind.

2.2.2.4 Der Leser

2.2.2.4.1 Aufgabe des Lesers

D 1.1 Der Leser ist fachlich zuständig: Er ist für die Auswertung des Inhalts der empfangenen Fachnachricht verantwortlich.

D 1.2 Der Leser prüft, ob die transportierte Fachnachricht spezifikationskonform ist.

D 1.3 Der Leser prüft, ob die Fachnachricht valide bezüglich des für den Standard gültigen Schemas ist. Der Leser kann diese Aufgabe durch entsprechende vertragliche Regelungen an den Empfänger delegieren.

2.2.2.4.2 Zuständigkeitsprüfung des Autors durch den Leser

D 2.1 Der Leser prüft seine eigene Zuständigkeit.

Anmerkungen:

- *Hintergrund für diese Prüfung ist der Wunsch, eine falsche Adressierung auszuschließen.*
- *Der Umgang mit falsch adressierten Nachrichten ist gesondert geregelt.*

D 2.2 Der Leser prüft Zuständigkeit und Berechtigung des Autors.

Anmerkungen:

- *Die Prüfung erfolgt, weil der Leser aus dem Ergebnis ableiten kann, wie er die erhaltenen Informationen verarbeitet, ob z.B. ein Register fortgeschrieben werden muss.*
- *Diese Prüfung kann an den Empfänger delegiert werden.*

D 2.3 Der Leser kann ggf. benötigte Attribute über den Empfänger für die Überprüfung des Autors abrufen.

2.2.2.4.3 Signatur

D 3.1 Der Leser ist zuständig für die Prüfung der (Autor-)Signatur der Fachnachricht.

D 3.2 Der Leser bewertet das Ergebnis der Prüfung der (Autor-)Signatur. Dies geschieht auch dann, wenn ein Dritter die (technische) Prüfung der Signatur durchgeführt hat.

Anmerkungen:

- *Die technische Prüfung kann grundsätzlich delegiert werden.*
- *Der Leser benötigt für seine Prüfung eine Liste der Signatur-Zertifikate, die ihm übermittelt werden müssen.*

2.2.2.4.4 Entschlüsselung

D 4.1 Der Leser ist für die Entschlüsselung der erhaltenen Fachnachricht zuständig.

Anmerkung:

- *Die Entschlüsselung kann grundsätzlich delegiert werden.*

2.2.2.4.5 Kommunikationszenario

D 5.1 Asynchrones Kommunikationszenario: Der Leser ist verpflichtet, Nachrichten und Transportinformation vom Empfänger abzurufen oder entgegenzunehmen.

Anmerkung:

- *Eine "Entgegennahme" setzt voraus, dass eine direkte Zustellung ("Push") eingerichtet ist.*

D 5.2 Synchrones Kommunikationsszenario: Der Leser bedient die Anfrage des Autors unmittelbar.

2.2.2.4.6 Service Qualität

D 6.1 Der Leser reagiert gemäß der vom Autor ausgewählten Service Qualität.

2.2.2.4.7 Eindeutige Identifizierung des Transportauftrags

D 7.1 Der Leser verwendet eine MessageID, um Informationen aus der Transport-Historie (also aus der Bearbeitung des Transportauftrags) auf die Fachnachricht zu beziehen.

2.2.2.4.8 Überwachung des Empfangs durch den Leser

D 8.1 Der Leser ist für die Auswertung der Transportinformationen verantwortlich.

D 8.2 Der Leser überprüft die Identität des Empfängers.

D 8.3 Der Leser ist dafür verantwortlich, im Rahmen seiner vertraglichen und rechtlichen Verpflichtungen für den Autor erreichbar zu sein. Dies impliziert die technische Erreichbarkeit für den Empfänger bei synchronen Transportszenarien.

2.2.2.4.9 Aufbewahrung

D 9.1 Der Leser ist für die Aufbewahrung der empfangenen Nachrichten und der relevanten Transportinformation zuständig und dafür, dass fristgerecht gelöscht wird. Für die Aufbewahrung kann er sich eines Dienstleisters bedienen. Die Löschfristen sind vertraglich festzulegen.

D 9.2 Der Leser legt im Rahmen der fachlichen, vertraglichen und rechtlichen Vorgaben fest, wie lange beim Empfänger die Nachrichten und die Protokolle der Nutzungsdaten gespeichert werden.

2.2.2.4.10 Vertraglicher und rechtlicher Rahmen

D 10.1 Das rechtliche Verhältnis zwischen Leser und Empfänger muss geklärt sein.

2.3 Quittungen in XTA 2

In einer XTA-konformen Nachrichteninfrastruktur ist das Instrument von *Quittungen* vorgesehen.

Durch Quittungen soll der Sender in die Lage versetzt werden, Auskunft über Ereignisse geben zu können, an denen er nicht notwendigerweise direkt beteiligt ist. Gedacht ist hier an die Ereignisse bei der Abarbeitung eines Transportauftrags innerhalb der entsprechenden Transportinfrastruktur. Für den Auftraggeber eines Transports ist in vielen Fällen diese Auskunftsfähigkeit seines Transport-Dienstleisters von großer Wichtigkeit.

Um dafür die Voraussetzungen zu schaffen, sieht XTA 2 die Möglichkeit vor, gezielt die Erstellung und Zustellung von Quittungen zu beauftragen. Eine Quittung wird, falls angefordert, durch einen Knoten der Infrastruktur an den Sender geschickt (oder zum Abholen vorgehalten), sobald der Knoten einen bestimmten Bearbeitungsschritt bei Nachrichtenempfang bzw. -weiterleitung erfolgreich abgeschlossen hat. Der Sender wird diese Quittungen entgegennehmen und im entsprechenden Report einen Eintrag machen (siehe zum Transportprotokoll den Abschnitt 5.5.2.4 auf Seite 156). Auf diesen Report wird der Autor bei Bedarf zugreifen, um die korrekte Abarbeitung seines Transportauftrags zu prüfen und Dritten gegenüber nachzuweisen.

In einer XTA-konformen Infrastruktur vorgesehene Quittungsarten: XTA 2 erfindet nicht Arten von Quittungen neu, sondern orientiert sich an internationalen Standards (Stichwort *Receipt*) bzw. deren Einfluss auf die Fortschreibung des Standards OSCI 2[1]. In Tabelle 2.1, „Arten von Quittungen“ sind die Arten von Quittungen aufgeführt, die in einer OSCI-Architektur unterschieden werden. Auf einige von ihnen wird in verschiedenen Abschnitten der vorliegenden Spezifikation Bezug genommen.

Tabelle 2.1. Arten von Quittungen

Quittungsart	zu erstellen durch Rolle	Bedingung, unter der die Quittung abzusetzen ist (ist identisch mit dem durch die Quittung angezeigte Ereignis)
Submission	Sender	Die Nachricht wurde erfolgreich versendet.
Relay	Relay (Zwischenstation / aktiver Knoten auf der Transportstrecke)	Die Nachricht wurde erfolgreich weitergeleitet.

[1] vgl. Spezifikation OSCI 2.0.2 (Angaben zur Quelle siehe Abschnitt 2.4.1, „OSCI-2.0.2“), dort *Figure 1*: *Actors and nodes involved in the message flow*; und Abschnitt *8.4.2.1 Delivery Attributes*, darin zum Objekt *oscimeta:ReceiptRequests*

Quittungsart	zu erstellen durch Rolle	Bedingung, unter der die Quittung abzusetzen ist (ist identisch mit dem durch die Quittung angezeigte Ereignis)
Delivery	Sender / Relay	Die Nachricht wurde an den Empfänger ausgeliefert bzw. befindet sich - in asynchronen Kommunikationsszenarien - im Zugriffsbereich (Postkorb) des Empfängers.
Fetch	Empfänger	Die Nachricht wurde aus dem Postkorb abgeholt.
Reception	Leser	Die Nachricht wurde abgeholt durch den Leser bzw. diesem zugestellt und liegt diesem entschlüsselt vor.

Die Rolle, durch die die entsprechende Quittung ggf. zu erzeugen ist, ist zu verstehen wie im Rollenmodell eingeführt, d.h. dass die Aktivität durch den Rolleninhaber (z.B. eine Fachbehörde als *Leser*) delegierbar ist an einen IT-Dienstleister (der ansonsten bspw. die Rolle *Empfänger* versieht).

Quittungen werden standardmäßig an den Sender ausgeliefert. Ausnahme: Der Sender verschickt nicht Quittungen an sich selber, denn das würde keinen Unterschied machen. Stattdessen trägt er den Sachverhalt, um den es geht (z.B. den Sachverhalt *Submission*), direkt in den *TransportReport* ein.

Verwendung der Quittung *Reception* in XTA 2: Dieser Quittungstyp wird im XTA-Kontext nicht genutzt. Begründung: Im Zusammenhang der Prozesse, die von XTA 2 durch eine einheitliche Lösung unterstützt werden, ist für den Zweck der Quittung *Reception* eine entsprechende Lösung auf Ebene des Fachstandards die bessere Wahl, falls dort ein entsprechender Bedarf gesehen wird. Die sonstigen Quittungen (*Submission*, *Relay*, *Delivery* und *Fetch*) werden in XTA 2 als genuine Quittungen auf Transportebene gesehen und entsprechend eingesetzt.

Zur Erstellung und Bewertung von Quittungen gelten in XTA 2 die folgenden Regeln:

Wie werden die von den Transportverfahren in einem bestimmten Kontext **auszuliefernden Quittungen** festgelegt?

- **Festlegung im Service Profil** (wie beschrieben in **Kapitel 4 auf Seite 49**): Die für einen bestimmten Geschäftsprozess auszuliefernden Quittungen sind im Service Profil spezifiziert, welches diesen Geschäftsprozess abdeckt. Damit ist für diesen Kontext festgelegt – ganz abgesehen von der indivuellen Situation und den speziellen Wünschen eines Autors, der einen bestimmten Transport beauftragt - welche Quittungen welcher Knoten der Transportstrecke an wen zu übermitteln hat.
- **Festlegung im Transportauftrag (**wie beschrieben zum Objekt ReceiptRequests in **Abschnitt 5.4.2.3.2.3 auf Seite 128**): Der Autor kann in den individuellen Transportauftrag von ihm gewünschte Quittungen eintragen (vgl. die optionalen Kindelemente von *oscimeta:MessageMetaData/DeliveryAttributes/ReceiptRequests*). Falls im MessageMetaData-Header des Transportauftrags dergestalt Quittungsanforderungen eingestellt sind, so sollen diese - über die im Service Profil festgelegten hinaus - von den beauftragten Transportverfahren geliefert werden, soweit diese dazu in der Lage sind.

Zum Aspekt *Festlegung im Transportauftrag*: Wann und durch wen sind die **Elemente des MessageMetaData-Headers unterhalb von ReceiptRequests** (wie beschrieben in **Abschnitt 5.4.2.3.2.3 auf Seite 128**) zu befüllen?

- Diese Elemente sind zu füllen, bevor die Nachricht durch den Sender abgesendet wird (sie also in die Transportinfrastruktur entlassen wird). Dies wird geschehen, wie bereits erwähnt, (a) gemäß der Vorschriften im Service Profil für den gegebenen Kontext und (b) ggf. zusätzlich durch die individuell vom Autor gewünschten Quittungen.
- Der Autor ist dafür verantwortlich, dass die entsprechenden Daten eingetragen sind. Das bedeutet, dass entweder das Fachverfahren die Daten vor Übergabe des Transportauftrags an den Sender einträgt. Oder aber diese Aufgabe wird delegiert im Sinne einer technischen Dienstleistung. Dann wird durch das Transportverfahren eine technische Funktionalität bereitgestellt ('*Aufbereitung des MessageMetaHeaders gemäß Service Profil*'), die das Fachverfahren von diesem Aufwand entlastet.

Implementierung: Wie wird die Übermittlung von Quittungen innerhalb der Transportinfrastruktur umgesetzt?

- Welche Arten von Quittungen in einer XTA-Umgebung übermittelt und vom Sender in den Report eingetragen werden können, hängt von der Leistungsfähigkeit der Sender-Empfänger-Messaging-Technologie ab. Beispielsweise unterstützt OSCI 2 die Receipts *Delivery*, *Fetch*, *Reception*; OSCI 1.2 hingegen nur die Receipts *Delivery*, *Fetch*.
- Dem entspricht eine Konsistenzanforderung an die Service Profile: Die für einen Geschäftsprozess vorgesehenen Quittungsarten müssen zu den für diesen Geschäftsprozess festgelegten Messaging-Technologien passen, also von diesen unterstützt werden.

Fehlerfall: Was hat ein Transportverfahren zu unternehmen, wenn es eine beauftragte Quittung nicht ausliefern kann?

- Im Kontext von Quittungen, die im Service Profil spezifiziert sind, ist wie folgt zu verfahren: Falls der Sender (oder ein anderen Knoten der Transportstrecke) die Ausführung einer der Quittungsanforderungen nicht gewährleisten kann, nimmt er den Transportauftrag nicht an, sondern muss mit einer entsprechenden Exception reagieren.
- Im Kontext von Quittungen, die ergänzend individuell vom Autor in den Transportauftrag eingetragen wurden, gilt hingegen: Hier sind, falls sie in der Infrastruktur nicht umsetzbar sind, die Quittungsanforderungen zu ignorieren. Der Sender (oder ein anderen Knoten der Transportstrecke) fährt also fort den regulären Prozess auszuführen und reagiert nicht mit einer Exception.

Quittungen und XTA-Konformität: Welche Quittungen muss ein Transportverfahren unterstützen?

- Welche Quittungen ein XTA-konformes Transportverfahren unterstützen muss, hängt von den Service Profilen ab, in Bezug auf die es XTA-Konformität beansprucht. Die in den entsprechenden Service Profilen festgelegten Quittungstypen muss dieses Transportverfahren unterstützen.

Ausgebliebene Quittungen: Wie ist der Erfolg der Durchführung des Transportauftrags zu bewerten, falls eine vorgesehene Quittung nicht ausgeliefert wurde?:

- Falls zum Prozess eine Quittungsanforderung im Service Profil festgelegt ist und diese nicht erfüllt ist, dann darf der Ampelstatus im TransportReport nicht auf grün gehen. Wenn die durch das Service Profil abgesteckte Zustellfrist verstrichen ist, geht der Ampelstatus auf rot, denn unter diesen Bedingungen ist keine regelkonforme Zustellung erfolgt.
- Eine Nachricht gilt als zugestellt, sobald auch die letzte geforderte Quittung des Service Profils eingegangen ist.
- Wenn optionale Quittungen fehlen (die vom Autor zusätzlich angefordert wurden), beeinflussen sie nicht den Ampelstatus im TransportReport.

2.4 Eingebundene externe Modelle

Folgende externe Modelle werden in dieser Spezifikation verwendet:

2.4.1 OSCI-2.0.2

Der Auslieferungsumfang des Standards ist zusammen mit weitergehenden Informationen und Materialien verfügbar unter der Überschrift *OSCI 2: Spezifikation und Schema (Version OSCI 2.0.2)* auf der Seite http://www.xoev.de/de/download#OSCI-Transport2

Präfix: osci

Version: 2.0.2

Namensraum: http://www.osci.eu/ws/2008/05/transport

SchemaLocation-Basis: http://www.osci.eu/ws/2014/10/transport/

Folgende Schema-Dateien werden verwendet:

- OSCI2_02.xsd

2.4.2 OSCI-2.0.2-MessageMetaData

Dieser Standard ist eine Komponente des Standards OSCI-2.0.2. Sein Auslieferungsumfang ist an derselben Stelle verfügbar, nämlich unter der Überschrift *OSCI 2: Spezifikation und Schema (Version OSCI 2.0.2)* auf der Seite http://www.xoev.de/de/download#OSCI-Transport2

Präfix: oscimeta

Version: 2.0.2

Namensraum: http://www.osci.eu/ws/2014/10/transport

SchemaLocation-Basis: http://www.osci.eu/ws/2014/10/transport/

Folgende Schema-Dateien werden verwendet:

- OSCI_MessageMetaData_V2.02.xsd

2.4.3 SOAP-Message-Security-1.0

Informationen über den Standard sind verfügbar unter: https://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf

Präfix: wsse

Version: 1.0

Namensraum: http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd

SchemaLocation-Basis: http://docs.oasis-open.org/wss/2004/01

Folgende Schema-Dateien werden verwendet:

- oasis-200401-wss-wssecurity-secext-1.0.xsd

2.4.4 WS-Addressing

Präfix: ws-addressing

Version: 1.0

Namensraum: http://www.w3.org/2005/08/addressing

SchemaLocation-Basis: http://www.w3.org/2005/08/addressing/

Folgende Schema-Dateien werden verwendet:

- ws-addr.xsd

2.4.5 XML-Encryption

Präfix: xml-encryption

Version: 1.0

Namensraum: http://www.w3.org/2001/04/xmlenc#

SchemaLocation-Basis: http://xoev.de/transport/xta/211/

Folgende Schema-Dateien werden verwendet:

- xenc-schema.xsd

2.4.6 XML-Signature

Präfix: xmldsig-core

Version: 1.1

Namensraum: http://www.w3.org/2000/09/xmldsig#

SchemaLocation-Basis: http://www.w3.org/TR/xmldsig-core/

Folgende Schema-Dateien werden verwendet:

- xmldsig-core-schema.xsd

2.4.7 XÖV-Basisdatentypen-V1.1

Präfix: XOEV-Basisdatentypen

Version: 1.1

Namensraum: http://xoev.de/schemata/basisdatentypen/1_1

SchemaLocation-Basis: http://xoev.de/schemata/basisdatentypen/1_1/

Folgende Schema-Dateien werden verwendet:

- xoev-basisdatentypen.xsd

3 Kooperation beim Datenaustausch: Anwendungsfälle

3.1 Einleitung

In den Prozessen der Datenübermittlung kooperieren die Infrastrukturkomponenten *Fachverfahren* und *Transportverfahren*. Fachverfahren sind dabei die IT-Verfahren, die in Behörden für die Vorgangsbearbeitung der jeweiligen Fachdomäne (z.B. Personenstandswesen, Pass- und Ausweisbehörde) eingesetzt werden. Transportverfahren haben die Funktion, Nachrichten zu senden, zu empfangen und an weiteren Aspekten der Übermittlung mitzuwirken. Dies geschieht unabhängig von der jeweiligen Fachdomäne.

Fach- und Transportverfahren werden häufig von getrennten Organisationen (z.B. Behörde und Rechenzentrum) betrieben, die für die Zwecke der Nachrichtenübermittlung vereinbarte Dienstleistungsbeziehungen eingehen.

In diesem Kapitel werden die Anwendungsfälle, die beim Datenaustausch notwendig sind, beschrieben. Hierbei wird von den IT-Verfahren abstrahiert. Stattdessen werden die Aufgaben und Prozesse auf der Basis der Rollen analysiert, in denen die Akteure der Prozesse kooperieren. Daher werden nicht die Fachverfahren, sondern die Rollen "Autor" und "Leser" benannt und statt der Transportverfahren die Rollen "Sender" und "Empfänger".

Der Darstellung der Anwendungsfälle liegen die Fragen zugrunde, welche Aufgabe und Zuständigkeitsbereiche an Erstellung, Transport und Verarbeitung von Nachrichten geknüpft sind. Hiermit verbundene Kompetenzen, Rechten und Pflichten werden berücksichtigt.

Die Anwendungsfälle werden als Use Case Modell dargestellt. Für die Visualisierung wird die UML-Notation gewählt, die es gestattet, die Beteiligung von Akteuren an Anwendungsfällen, die Beziehung eines Anwendungsfalls zu anderen Anwendungsfällen und auch zu Informationsobjekten (Klassen oder Objekten) darzustellen.

Ziel ist es hierbei, eine Sicht zu entwickeln, die sowohl fachlich-organisatorische als auch technische Aspekte berücksichtigt, sie aber voneinander abgrenzt und in einen Zusammenhang stellt.

So können technische Komponenten wie der XTA-Webservice (XTA-WS) in ihrer Funktionalität klarer bestimmt werden. Der XTA-WS muss aus dieser Sichtweise alles anbieten, was die Interaktion der Rollen "Autor" und "Leser" mit den Rollen "Sender" bzw. "Empfänger" unterstützt. Wenn also der Autor einen Transportauftrag erteilen können soll, muss beispielsweise eine entsprechende Operation bzw. Methode im XTA-WS angeboten werden.

Ergänzend zum hier dargestellten Use Case Modell wird auf das Modell der Rollen und Verantwortlichkeiten verwiesen (siehe Abschnitt 2.2 auf Seite 14).

Grundlage dieser Darstellung ist der rechtliche Rahmen, der durch den Gesetz- oder Verordnungsgeber vorgegeben wird und der von den Akteuren zu beachten ist. Die rechtlichen Rahmenbedingungen können Vorgaben für die Qualität der Datenübermittlung enthalten.

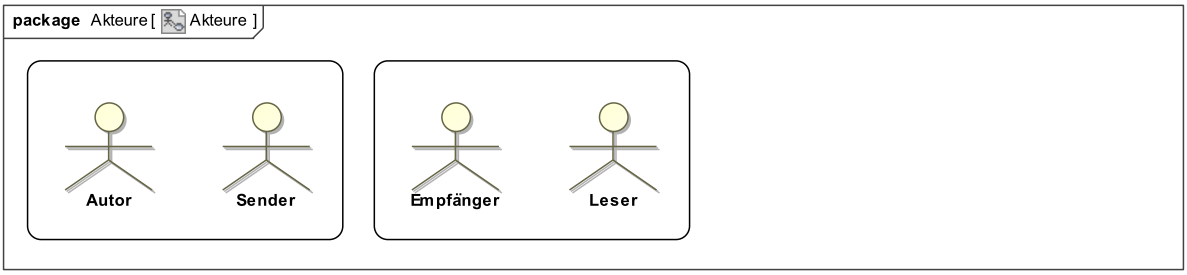
3.2 Anwendungsfälle beim Datenaustausch

Das Kapitel beginnt mit der Vorstellung der an den Anwendungsfällen beteiligten Akteure und mit einer Überblicksdarstellung, in der die Handlungsfelder skizziert sind. Jeder hier genannte Anwendungsfall wird weiter unten detailliert dargestellt.

3.2.1 Akteure

Alle beteiligten Akteure setzen in den Prozessen die jeweils anzuwendenden rechtlichen Regelungen um.

Abbildung 3.1. Anwendungsfalldiagramm "Akteure"



3.2.1.1 Autor

Der Autor ist als Fachverantwortlicher für die fachliche Erstellung der zu transportierenden Nachricht zuständig. Er ist außerdem Auftraggeber des Transports, für den er die rechtlich-organisatorischen Rahmenbedingungen vorgibt, und er überwacht die Erfüllung seiner Transportanforderungen. Als Zuständiger für die fachliche Erstellung und als Auftraggeber des Transports ist der Autor für die Konsistenz aller Informationen verantwortlich, die ggf. sowohl im Transportauftrag als auch in der Fachnachricht enthalten sind.

Eine detaillierte Darstellung der Aufgaben und Verantwortlichkeiten des Autors wird in Abschnitt 2.2.2.1 ab Seite 14 gegeben.

3.2.1.2 Sender

Der Sender ist gemäß des Transportauftrags des Autors zuständig für die Abwicklung des Transports und aller damit zusammenhängenden Leistungen wie Adressierung, Transport-Verschlüsselung und Transport-Signatur sowie Protokollierung.

Detaillierter werden seine Aufgaben und Verantwortlichkeiten in Abschnitt 2.2.2.2 ab Seite 17 dargestellt.

3.2.1.3 Empfänger

Der Empfänger ist in seiner Aufgabe als Transporteur vom Leser beauftragt, Nachrichten entgegenzunehmen, sie vorzuhalten und sie dem Leser ggf. direkt zuzustellen. Der Empfänger ist außerdem beauftragt, hiermit verbundene Aufgaben auszuführen. Dies sind insbesondere die Prüfungen zur Identität und von Zertifikaten und Aufgaben der Protokollierung.

Im Detail werden die Aufgaben und Verantwortlichkeiten des Empfängers in Abschnitt 2.2.2.3 ab Seite 18 dargestellt.

3.2.1.4 Leser

Als Fachverantwortlicher ist der Leser zuständig für die Entgegennahme und fachliche Auswertung der transportierten Nachricht. Zu den Auswertungen gehören insbesondere die Prüfungen in Bezug auf die Autorenschaft.

Eine detaillierte Darstellung seiner Aufgaben und Verantwortlichkeiten wird in Abschnitt 2.2.2.4 ab Seite 19 gegeben.

3.2.2 Anwendungsfälle im Überblick

In Abbildung 3.2, „Anwendungsfalldiagramm "Anwendungsfälle im Überblick"“ wird ein Überblick gegeben über die Handlungsfelder, die vom Standard XTA 2 berührt sind.

Jedem Anwendungsfall aus diesem Diagramm entspricht weiter unten eine detailliertere Darstellung in einem eigenen Diagramm.

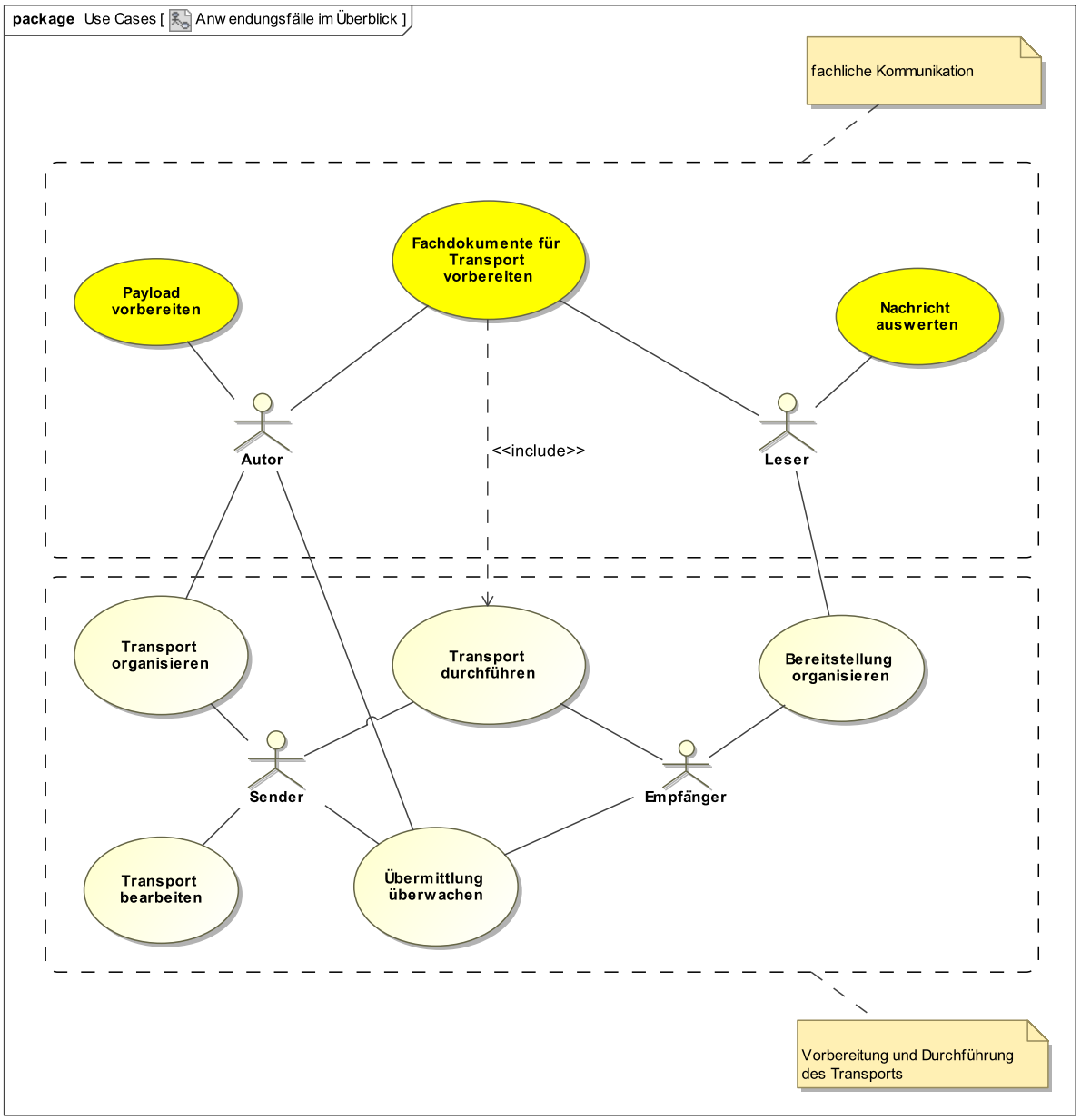
Diese Anwendungsfall bezogene Darstellung ergänzt die lineare Sicht, die der Architektur-Abbildung und dem Rollenmodell zugrunde liegt (siehe Abbildung 2.1, „Infrastruktur der Nachrichtenübermittlung: Kooperation von Fach- und Transportverfahren“. Die im Rollenmodell benannten Kernprozesse für die Akteure Autor, Sender, Empfänger und Leser werden in den Anwendungsfalldiagrammen um detailliertere organisatorische und technische Aktivitäten ergänzt.

Zur Verwendung der Anwendungsfalldiagramme: Es werden sowohl die aktiven Akteure als auch die indirekt betroffenen oder eher passiven Akteure dargestellt und benannt. Beispiel: Beteiligte Akteure im Anwendungsfall "Transport durchführen" sind der Sender und auch der Empfänger.

In der Dokumentation werden Anwendungsfalldiagramme auch als "Use Cases" (UC) bezeichnet.

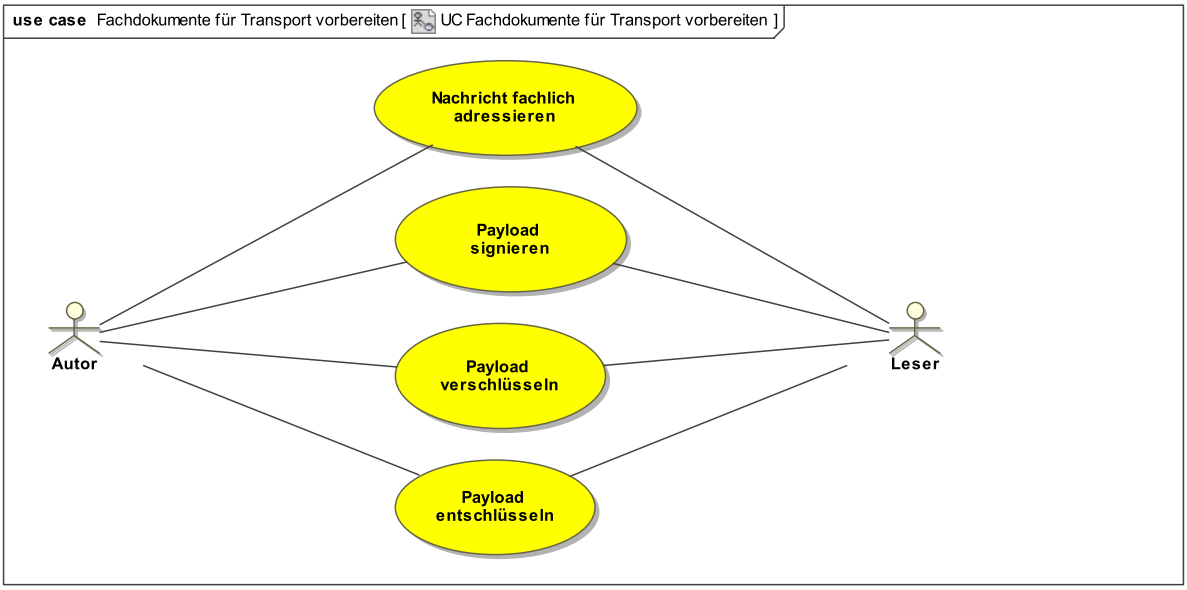
Die in der Dokumentation der Anwendungsfälle verwendeten Fachbegriffe werden im Glossar erläutert.

Abbildung 3.2. Anwendungsfalldiagramm "Anwendungsfälle im Überblick"



3.2.3 UC Fachdokumente für Transport vorbereiten

Abbildung 3.3. Anwendungsfalldiagramm "UC Fachdokumente für Transport vorbereiten"



Der Nachrichtenaustausch ist in fachliche Prozesse eingebettet, an denen Beteiligte organisationsübergreifend mitgewirken können. Diese werden an definierten Stellen in den Prozess einbezogen und ggf. zu Folgeprozessen innerhalb ihrer Zuständigkeit veranlasst.

Der Austausch von Nachrichten wird realisiert auf der Basis technischer Integration von IT-Systemen.

3.2.3.1 Enthaltene Anwendungsfälle

3.2.3.1.1 Nachricht fachlich adressieren

Beschreibung
Der Autor ist für die fachliche Adressierung des Lesers zuständig. Der Autor kann prüfen, ob der Leser in einem bestimmten fachlichen Kontext grundsätzlich elektronisch erreichbar ist. (Hiermit ist nicht die Prüfung gemeint, ob der Leser aktuell erreichbar / verfügbar ist.) Der Sender stellt hierfür eine entsprechende Funktionalität zur Verfügung. Die Prüfung erfolgt durch qualitätsgesicherte Verzeichnisse der öffentlichen Verwaltung (z.B. DVDV, S.A.F.E.). Der Autor muss benötigte Attribute für die elektronische Kommunkation mit dem Leser abrufen können, sofern dies im fachlichen Kontext notwendig ist. Hierbei stellt der Sender eine entsprechende Funktionalität zur Verfügung.

3.2.3.1.2 Payload signieren

Beschreibung
Der Autor kann die zu transportierende Nachricht oder Teile der Nachricht signieren. Der Autor ist zuständig für die Signatur der Nachricht, nicht der Sender. Ein Sender kann im Auftrag des Autors diese Aufgabe wahrnehmen. Der Autor muss immer über die Signatur identifizierbar bleiben.

3.2.3.1.3 Payload verschlüsseln

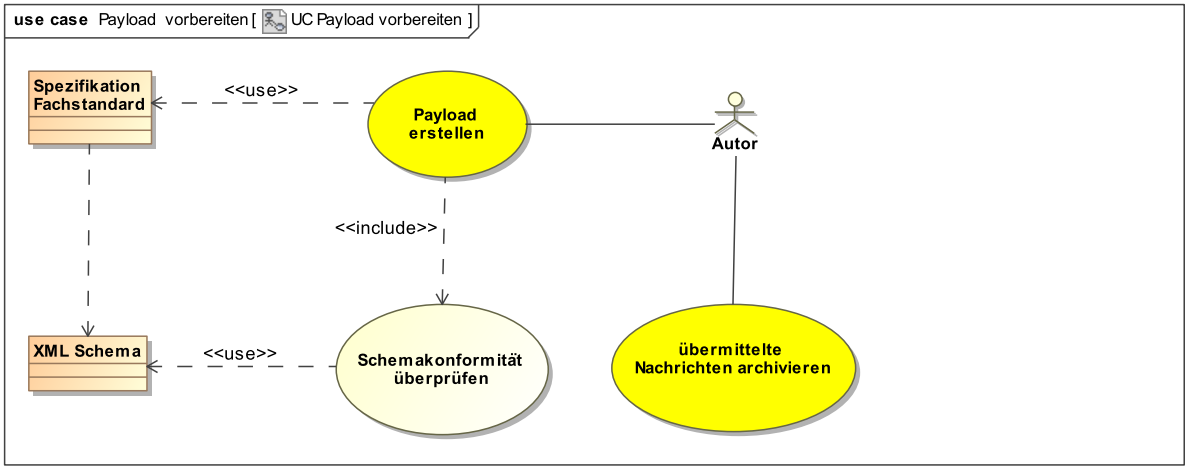
Beschreibung
Der Autor kann die zu transportierende Nachricht oder Teile der Nachricht verschlüsseln. Der Autor ist zuständig für die Verschlüsselung der Nachricht, nicht der Sender. Ein Sender kann im Auftrag des Autors diese Aufgabe wahrnehmen.

3.2.3.1.4 Payload entschlüsseln

Beschreibung
Der Leser kann die zu transportierende Nachricht oder Teile der Nachricht entschlüsseln. Der Leser ist zuständig für die Entschlüsselung der Nachricht, nicht der Empfänger. Ein Empfänger kann im Auftrag des Lesers diese Aufgabe wahrnehmen.

3.2.4 UC Payload vorbereiten

Abbildung 3.4. Anwendungsfalldiagramm "UC Payload vorbereiten"



Der Nachrichtenaustausch ist in organisatorisch-fachliche Prozesse eingebunden, die unterschiedliche Akteure betreffen. Diese Partner werden an definierten Stellen in den Prozess einbezogen und ggf. zu Folgeprozessen innerhalb ihrer Zuständigkeit veranlasst.

Der Use Case fasst Aktivitäten des Autors, der fachliche Prozesse im Rahmen seiner Zuständigkeit verfolgt und an definierten Schnittstellen Nachrichten- und Daten für den Austausch mit Kooperationspartnern (Leser) erstellt, zusammen.

Bei entsprechenden vertraglichen Regelungen zwischen Autor und Sender kann die Durchführung bestimmter Aufgaben an den Sender delegiert werden.

3.2.4.1 Enthaltene Anwendungsfälle

3.2.4.1.1 Payload erstellen

Eingebundene Use Cases		
Use Case	**Ref.**	**Seite**

Eingebundene Use Cases		
Schemakonformität überprüfen	3.2.4.1.2	33
Verwendete Artefakte		
Artefakt	**Ref.**	**Seite**
Spezifikation Fachstandard	3.3.2	45

Beschreibung

Aktivitäten des Autors, der im Rahmen seiner fachlichen Zuständigkeit die zu übermittelnde Nachricht vorbereitet:

Der Autor ist fachlich zuständig, d.h. er ist für den Inhalt der zu transportierenden Nachricht verantwortlich.

Der Autor erstellt den Inhalt der zu transportierenden Nachricht. Er erstellt die zu transportierende Nachricht gemäß den Regeln des zu Grunde liegenden Standards (z.B. OSCI-XMeld) in einer bestimmten Version.

Amerkung:

- *Der vollständige Inhalt der vom Autor erstellten Nachricht ist für den Leser relevant. Und alles, was für den Leser relevant ist, sollte in der Nachricht enthalten sein. Dies betrifft auch die Informationen, die im Nachrichtenkopf einer XÖV-Nachricht (etwa vergleichbar dem Inhalt eines Briefkopfes), enthalten sind, wie z.B. der AGS von Absender und Empfänger sowie die Nachrichten-Identifizierung.*

Der Autor ist verantwortlich dafür, dass die Nachricht spezifikationskonform ist. Das schließt ein, dass die Nachricht valide bezüglich des für den Standard (in der entsprechenden Version) gültigen Schemas ist.

3.2.4.1.2 Schemakonformität überprüfen

Verwendete Artefakte		
Artefakt	**Ref.**	**Seite**
XML Schema	3.3.10	47

Beschreibung

Der Autor ist verantwortlich für die Schemakonformität der Nachricht. Er stellt sicher, dass die Nachricht valide bezüglich der XML-Schema-Definition, die zur Spezifikation gehört, ist. Der Autor kann, bei entsprechender vertraglicher Regelung, diese Aufgabe an den Sender delegieren.

3.2.4.1.3 übermittelte Nachrichten archivieren

Beschreibung

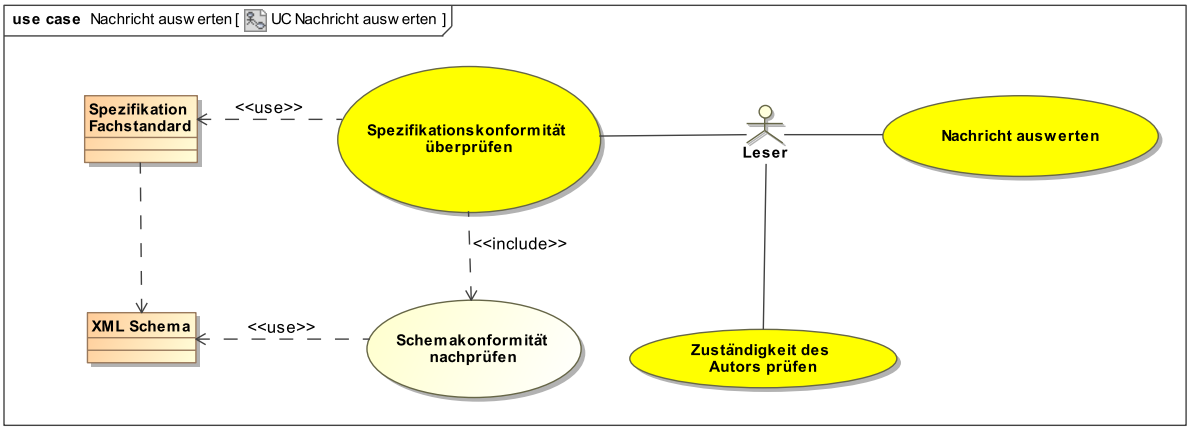
Entsprechend der rechtlichen Vorgaben werden die Nachrichten archiviert:

Der Autor ist für die Aufbewahrung der versandten Nachrichten und der relevanten Transportinformation zuständig und dafür, dass fristgerecht gelöscht wird. Für die Aufbewahrung kann er sich eines Dienstleisters bedienen.

Der Autor legt fest, wie lange beim Sender die Nachrichten und die Protokolle der Nutzungsdaten gespeichert werden. Die Löschfristen werden vertraglich vereinbart.

3.2.5 UC Nachricht auswerten

Abbildung 3.5. Anwendungsfalldiagramm "UC Nachricht auswerten"



Der Leser nimmt die übermittelte Nachricht an den definierten fachlichen Schnittstellen entgegen und führt eine inhaltliche Analyse mit den daraus folgenden Aktivitäten durch.

3.2.5.1 Enthaltene Anwendungsfälle

3.2.5.1.1 Spezifikationskonformität überprüfen

Eingebundene Use Cases		
Use Case	**Ref.**	**Seite**
Schemakonformität nachprüfen	3.2.5.1.2	34
Verwendete Artefakte		
Artefakt	**Ref.**	**Seite**
Spezifikation Fachstandard	3.3.2	45
Beschreibung		
Der Leser prüft die Nachricht gegen die Regeln des zugehörigen Fachstandards.		

3.2.5.1.2 Schemakonformität nachprüfen

Verwendete Artefakte		
Artefakt	**Ref.**	**Seite**
XML Schema	3.3.10	47
Beschreibung		
Der Leser überprüft, ob die Nachricht valide bzgl. der XML-Schema-Definition, die zur Spezifikation gehört, ist. Der Leser kann diese Aufgabe durch entsprechende vertragliche Regelungen an den Empfänger delegieren.		

3.2.5.1.3 Zuständigkeit des Autors prüfen

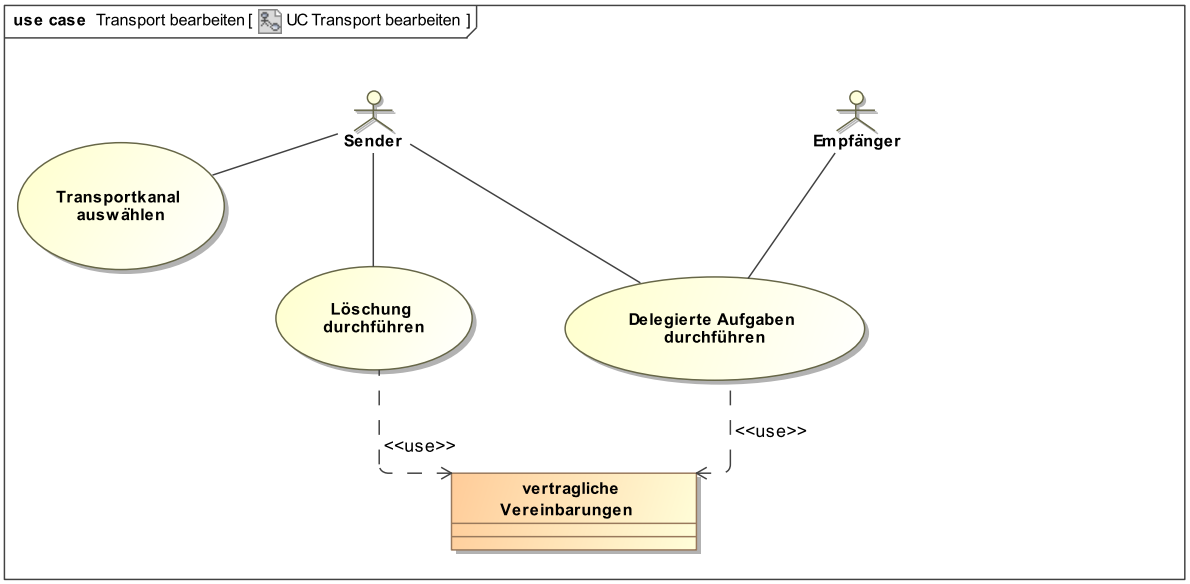
Beschreibung
Der Leser prüft Zuständigkeit und Berechtigung des Autors. Die Prüfung erfolgt, weil der Leser aus dem Ergebnis ableiten kann, wie er die erhaltenen Informationen verarbeitet, ob z.B. ein Register fortgeschrieben werden muss. Diese Prüfung kann an den Empfänger delegiert werden.

3.2.5.1.4 Nachricht auswerten

Beschreibung
Aktivitäten des Lesers, der im Rahmen seiner fachlichen Zuständigkeit die übermittelte Nachricht inhaltlich analysiert und entsprechend weiterbearbeitet. Die Regeln des entsprechenden Fachstandards unterstützen den Leser, den fachlichen Inhalt der Nachricht zu identifizieren und zu interpretieren.

3.2.6 UC Transport bearbeiten

Abbildung 3.6. Anwendungsfalldiagramm "UC Transport bearbeiten"



Vor und nach dem eigentlichen Transport der Daten sind Sender und Empfänger für weitere, hiermit eng verbundene Aufgaben zuständig. Dazu zählen Aufgaben, die als Delegation vom Autor bzw. Leser durch entsprechende vertragliche Vereinbarung übernommen wurden, wie das Anbringen oder Prüfen von Signaturen, Virenprüfungen oder die Auswahl alternativer Transportinfrastrukturen.

3.2.6.1 Enthaltene Anwendungsfälle

3.2.6.1.1 Delegierte Aufgaben durchführen

Verwendete Artefakte		
Artefakt	**Ref.**	**Seite**
vertragliche Vereinbarungen	3.3.9	47
Beschreibung		
Sender und Empfänger führen Aufgaben durch, die sie durch vertraglich geregelte Delegation vom Autor bzw. Leser erhalten haben. Dies kann bspw. das Anbringen einer Signatur oder das Verschlüsseln auf Inhaltsebene sein oder auch die Durchführung von Schemaprüfungen.		

3.2.6.1.2 Transportkanal auswählen

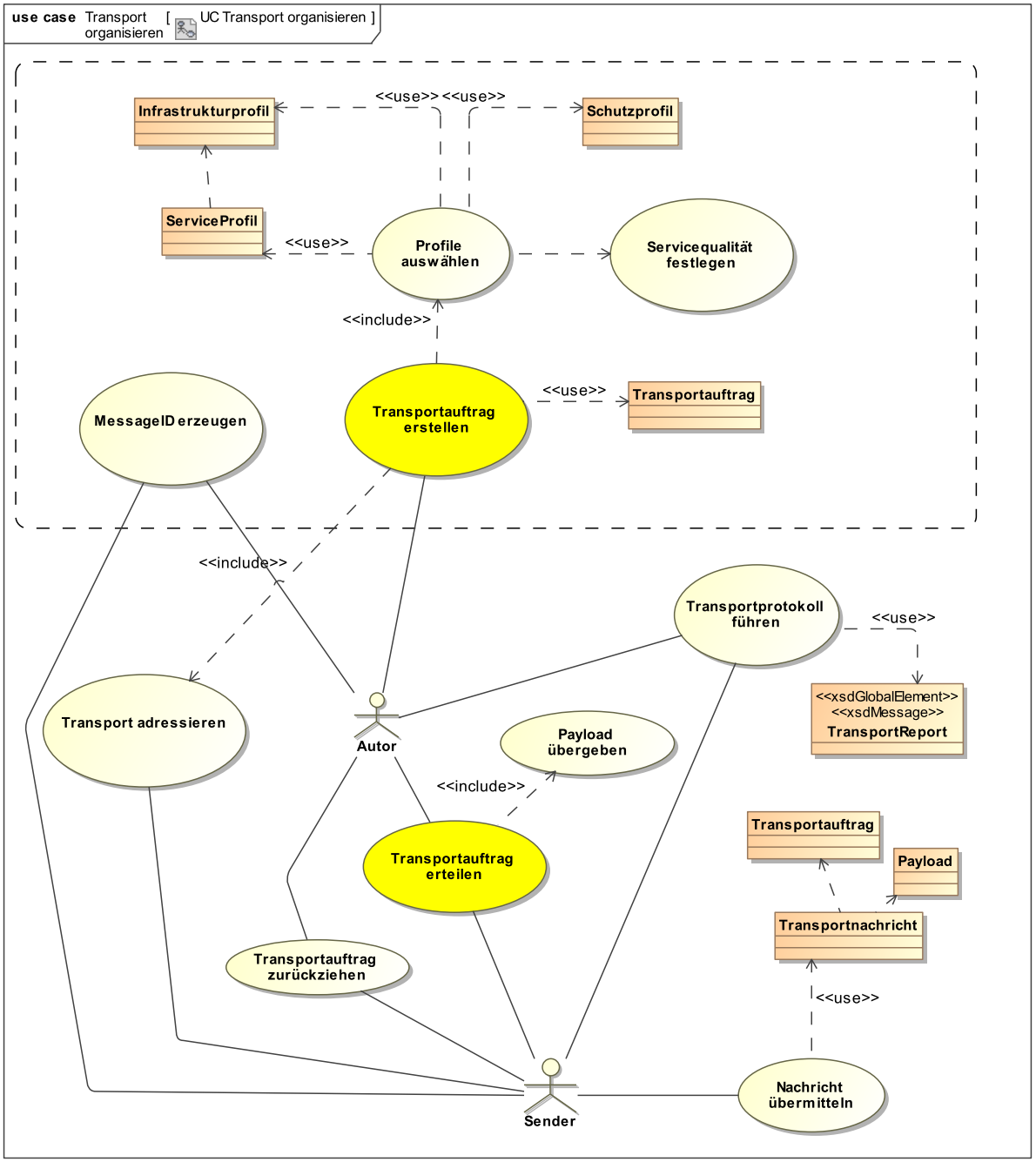
Beschreibung
Für die Übermittlung an den Empfänger wählt der Sender den passenden Transportkanal aus. Diese Wahl trifft er auf der Basis der Angaben im Transportauftrag sowie weiterer Kontextinformationen, die z.B. für den länderinternen Datentransport gelten.

3.2.6.1.3 Löschung durchführen

Verwendete Artefakte		
Artefakt	**Ref.**	**Seite**
vertragliche Vereinbarungen	3.3.9	47
Beschreibung		
Im Rahmen der Vereinbarungen und Bestimmungen löscht der Sender aufbewahrte Dokumente und Protokolle innerhalb bestimmter Fristen.		

3.2.7 UC Transport organisieren

Abbildung 3.7. Anwendungsfalldiagramm "UC Transport organisieren"



Zur Vorbereitung des Transports gehört die Zusammenstellung und Erteilung des Transportauftrags. Den hierfür notwendigen rechtlichen Rahmen gibt der Gesetz- oder Verordnungsgeber vor. Die rechtlichen Rahmenbedingungen können Einfluss auf die Qualität der Datenübermittlung haben. Bei der Erteilung des Transportauftrags wird der Payload übergeben und die Übermittlung der Nachricht initiiert.

3.2.7.1 Enthaltene Anwendungsfälle

3.2.7.1.1 Servicequalität festlegen

Beschreibung
Das Festlegen der Service Qualität bedeutet, dass die Parameter des Transports aus Sicht der Anforderungen der Fachverantwortlichen definiert werden. Sofern rechtliche Regelungen zur Qualität der Datenübermittlung existieren, sind die Parameter des Transports durch die Fachverantwortlichen entsprechend zu formulieren. Die Festlegung der benötigten Service Qualität erfolgt durch die Auswahl von Profilen. Wichtige Parameter der Service Qualität sind Integrität, Vertraulichkeit, Verfügbarkeit, Transparenz und Intervenierbarkeit.

3.2.7.1.2 Profile auswählen

Verwendete Artefakte		
Artefakt	**Ref.**	**Seite**
Infrastrukturprofil	3.3.7	47
Schutzprofil	3.3.6	47
ServiceProfil	3.3.5	46
Beschreibung		
Die Profile, die die Anforderungen an die Transportdurchführung erfüllen, werden ausgewählt und im Transportauftrag referenziert. Wenn kein geeignetes Profil zur Verfügung steht, muss es in einem definierten Prozess erstellt werden. Die Beschreibungen zu den Profilarten - Modell und Prozesse - sind dem Abschnitt Kapitel 4 auf Seite 49 zu entnehmen.		

3.2.7.1.3 Transportauftrag erstellen

Eingebundene Use Cases		
Use Case	**Ref.**	**Seite**
Transport adressieren	3.2.7.1.5	39
Profile auswählen	3.2.7.1.2	38
Verwendete Artefakte		
Artefakt	**Ref.**	**Seite**
Transportauftrag	3.3.4	46
Beschreibung		
Der Transport einer Nachricht wird durch das Zusammenstellen der Daten des Transportauftrags vorbereitet. Hierzu gehören insbesondere die Adressierung von Autor und Leser, die Festlegung des Service Profils und die ID des Transportauftrags (MessageID). Der Eintrag der Autor bzw. Leser-Identifikation muss mit dem entsprechenden Eintrag in der Fachnachricht übereinstimmen. Als Zuständiger für die fachliche Erstellung und als Auftraggeber des Transports ist der Autor für die Konsistenz aller Informationen verantwortlich, die sowohl im Transportauftrag als auch in der Fachnachricht enthalten sind. Die Parameter des Transportauftrags werden vorgehalten und bei Erteilung des Auftrags übergeben.		

3.2.7.1.4 MessageID erzeugen

Beschreibung
Jeder Transportauftrag ist über seine MessageID identifzierbar. Diese wird (veranlasst durch den Autor) durch den Sender erzeugt und vom Autor in die Transportauftragsdaten eingetragen.

3.2.7.1.5 Transport adressieren

Beschreibung
Für die Erteilung des Transportauftrags muss der Leser der Nachricht adressiert werden können. Außerdem sollte verifiziert sein, dass er zum Empfang des vorliegenden Nachrichtentyps einen entsprechenden technischen Dienst eingerichtet hat. Die hierfür notwendigen technischen Parameter für die Adressierung des Lesers werden vom Sender ermittelt und in die Daten des Transportauftrags eingetragen.

3.2.7.1.6 Transportauftrag erteilen

Eingebundene Use Cases		
Use Case	**Ref.**	**Seite**
Payload übergeben	3.2.7.1.7	39
Beschreibung		
Der Auftrag zum Transport einer Nachricht wird vom Autor durch den Aufruf der entsprechenden Operation der Sende-Schnittstelle des XTA-Webservice dem Sender erteilt. Diesem Aufruf werden die Daten des Transportauftrags und der zu transportierende Payload mitgegeben. Als Zuständiger für die fachliche Erstellung und als Auftraggeber des Transports ist der Autor für die Konsistenz aller Informationen verantwortlich, die sowohl im Transportauftrag als auch in der Fachnachricht enthalten sind. Damit beginnt die Ausführung durch den Sender innerhalb der Transportinfrastruktur. Der Sender initiiert den Transport zum Empfänger gemäß der Parameter des Transportauftrags.		

3.2.7.1.7 Payload übergeben

Beschreibung
Die zu transportierende Nachricht, die ggf. signiert und / oder verschlüsselt ist, wird durch den Autor mit der Erteilung des Transportauftrags übergeben und ist damit der Payload des Transports. Die Übergabe geschieht als Parameter beim Aufruf einer XTA-WS-Methode.

3.2.7.1.8 Nachricht übermitteln

Verwendete Artefakte		
Artefakt	**Ref.**	**Seite**
Transportnachricht	3.3.1	45
Beschreibung		
Der Sender wertet die Parameter des Transportauftrags aus, um die Nachricht auf dem vorgesehenen Weg dem Empfänger zuzuleiten.		

3.2.7.1.9 Transportprotokoll führen

Verwendete Artefakte		
Artefakt	**Ref.**	**Seite**
TransportReport	5.5.2.4	156
Beschreibung		
Das Transportprotokoll wird vom Sender geführt, der Ereignisse, Warnungen und Fehler einträgt. Das Transportprotokoll ist jederzeit vom Autor einsehbar.		

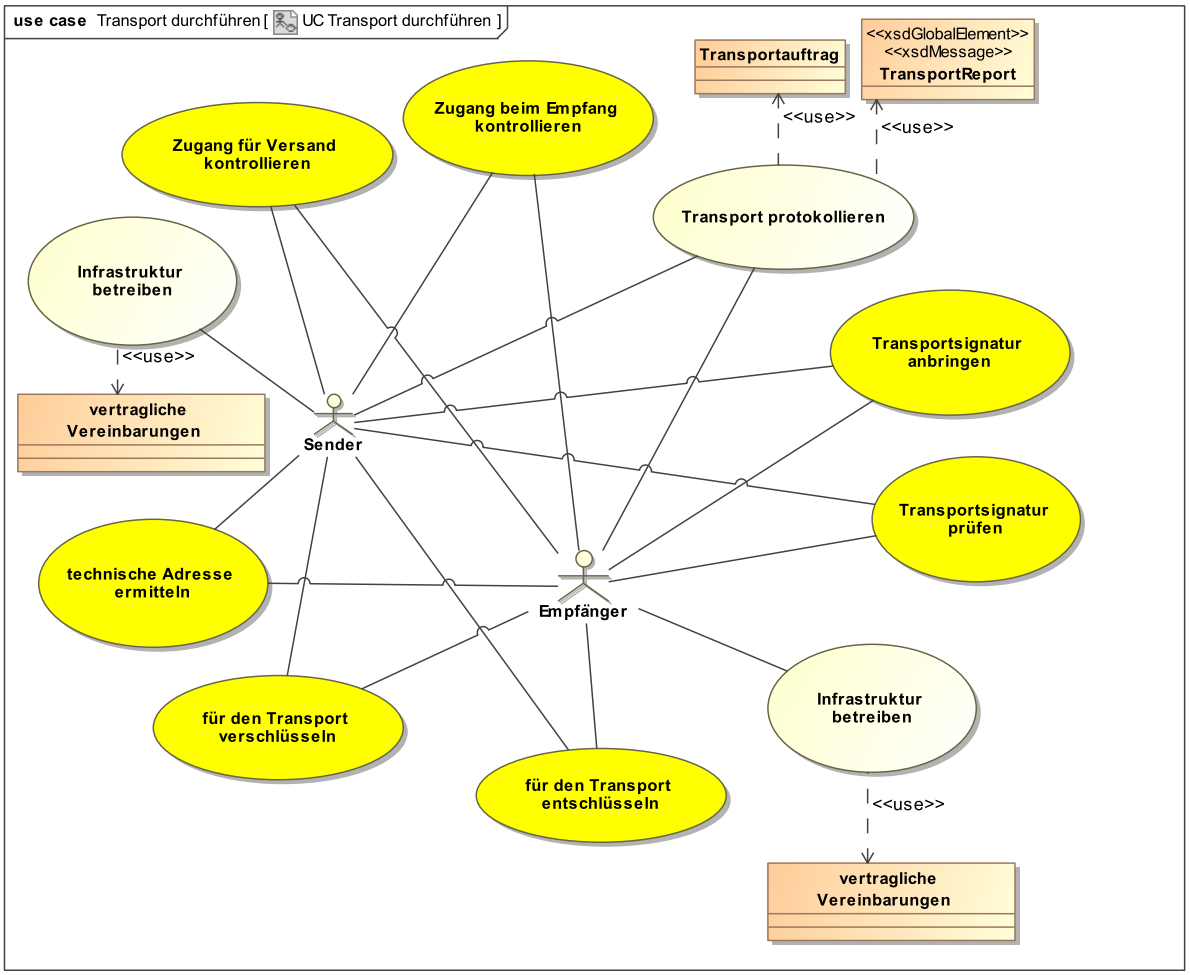
Beschreibung
Das Transportprotokoll wird durch das Objekt TransportReport realisiert.

3.2.7.1.10 Transportauftrag zurückziehen

Beschreibung
Transportaufträge, für die bei Erteilung eine Auslieferungsverzögerung eingetragen wurde und die noch nicht vom Sender ausgeführt sind (die Nachricht ist noch nicht abgesendet, der Transportauftrag also noch offen), können bei Bedarf zurückgezogen werden. Dies ist Aufgabe des Autors. Für das Zurückziehen stellt der Sender dem Autor eine Funktionalität zur Verfügung.

3.2.8 UC Transport durchführen

Abbildung 3.8. Anwendungsfalldiagramm "UC Transport durchführen"



Der Austausch von Nachrichten wird durch die Ausführung von Request/Response-Protokollen wie z.B. SOAP über HTTP realisiert. Transport-Signaturen und -Verschlüsselung werden entsprechend angewendet.

3.2.8.1 Enthaltene Anwendungsfälle

3.2.8.1.1 Infrastruktur betreiben

Verwendete Artefakte		
Artefakt	**Ref.**	**Seite**
vertragliche Vereinbarungen	3.3.9	47

Beschreibung

Der Sender ist gemäß Transportauftrag für die Abwicklung des Transports zuständig. Der Sender unterhält dafür die Infrastruktur und gibt dem Autor einen entsprechenden Zugang.

Der Empfänger ist vom Leser mit der Entgegennahme von Nachrichten beauftragt. Anmerkung: Ein Intermediär ist Bestandteil der Empfänger-Infrastruktur.

Der Empfänger unterhält für die Entgegennahme von Nachrichten die Infrastruktur.

3.2.8.1.2 Transportsignatur anbringen

Beschreibung

Der Sender bringt je nach geltender Policy, die im Service Profil abgebildet ist, für den jeweiligen Transportkanal ggf. die Transportsignatur an.

3.2.8.1.3 für den Transport verschlüsseln

Beschreibung

Der Sender verschlüsselt die zu transportierende Nachricht je nach Policy für den jeweiligen Transportkanal ggf. für den Empfänger.

3.2.8.1.4 technische Adresse ermitteln

Beschreibung

Der Sender prüft, ob für den Leser ein Zugang eröffnet ist. Der Sender ermittelt die technische Adresse des Lesers anhand dessen fachlicher Adresse. Er verwendet hierfür ein Verzeichnis wie DVDV oder S.A.F.E.

Der Sender stellt, falls nötig, dem Autor die technischen Attribute des Lesers zur Verfügung.

3.2.8.1.5 Zugang für Versand kontrollieren

Beschreibung

Der Zugang zur Transportinfrastruktur und der Zugang zu den angeforderten Diensten wird von den Transporteuren kontrolliert:

Prüfung der Identität des Autors durch den Sender:

- Der Sender ist für die Authentifizierung des Autors zuständig, d. h. er prüft die Identität des Autors. Anmerkung: Der Sender überprüft, ob ihm die Authentisierungsinformationen des Autors bekannt sind.
- Der Sender ist verpflichtet, die Angaben der Authentifizierung auf Konsistenz mit den Absenderangaben des Transportauftrages zu prüfen. Anmerkung: Durch diese Prüfung wird geklärt, ob die authentisierte Behörde in diesem Fachkontext mit dieser Behördenidentität (z.B. AGS:12343123 für Oldenburg im Meldewesen) auftreten darf.

Beschreibung
• Abgrenzung: Der Sender ist nicht dafür zuständig, die fachliche Zuständigkeit des Autors für den Inhalt der Nachricht zu prüfen. Dies geschieht durch den Leser.

3.2.8.1.6 Transport protokollieren

Verwendete Artefakte		
Artefakt	**Ref.**	**Seite**
Transportauftrag	3.3.4	46
TransportReport	5.5.2.4	156
Beschreibung		
Sender und Empfänger protokollieren die Ereignisse entsprechend der Vorgaben aus dem Transportauftrag.		

3.2.8.1.7 Zugang beim Empfang kontrollieren

Beschreibung
Der Zugang zur Transportinfrastruktur und der Zugang zu den angeforderten Diensten wird von den Transporteuren kontrolliert: Prüfung der Identität des Lesers durch den Empfänger: • Der Empfänger ist für die Authentifizierung des Lesers zuständig, d. h. er hat die Identität des Lesers zu prüfen. Anmerkung: Die Prüfung dient der Zugriffskontrolle im Kontext der Autorisierung nach Absprache mit dem Leser. • Der Empfänger ist für die Prüfung zuständig, ob die Identität des Lesers (Authentisierung gegenüber dem Empfänger) konsistent ist mit der Identität des Lesers für die Fachkommunikation im Rahmen desTransportauftrags. • Abgrenzung: Der Empfänger ist nicht dafür zuständig, die fachliche Zuständigkeit des Lesers für den Inhalt der Nachricht zu prüfen. Dies geschieht durch den Leser.

3.2.8.1.8 Transportsignatur prüfen

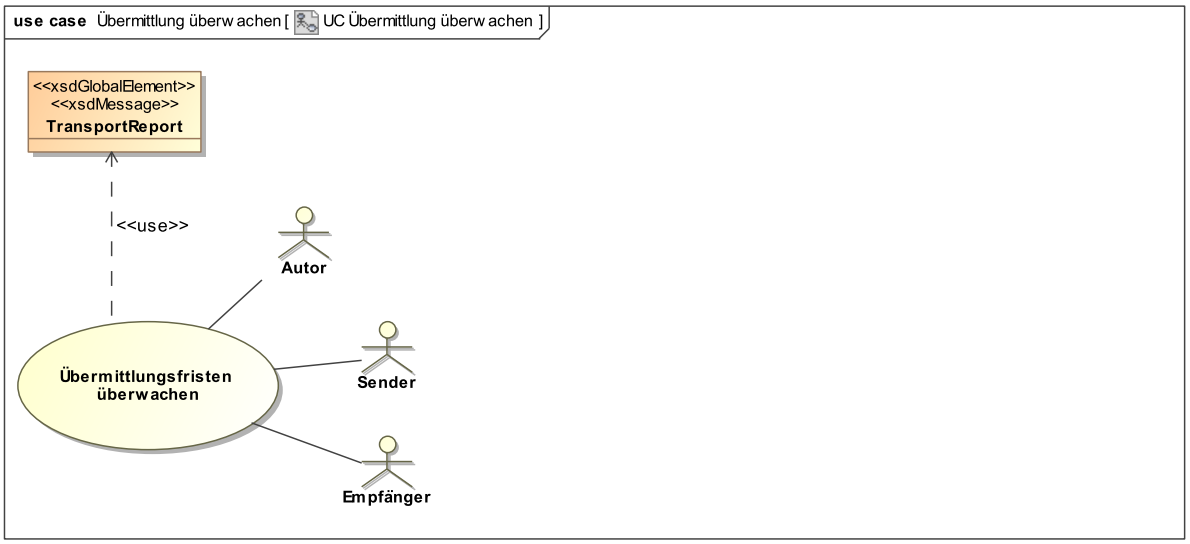
Beschreibung
Der Empfänger prüft je nach geltender Policy, die im Service Profil abgebildet ist, für den jeweiligen Transportkanal ggf. die Transportsignatur.

3.2.8.1.9 für den Transport entschlüsseln

Beschreibung
Der Empfänger entschlüsselt je nach Policy für den jeweiligen Transportkanal ggf. die Transportverschlüsselung für den Leser.

3.2.9 UC Übermittlung überwachen

Abbildung 3.9. Anwendungsfalldiagramm "UC Übermittlung überwachen"



Der Autor überwacht Erfolg und Fristen der Zustellung seiner Transportaufträge an Empfänger und Leser.

Dies geschieht durch Auswertung der Protokolleinträge.

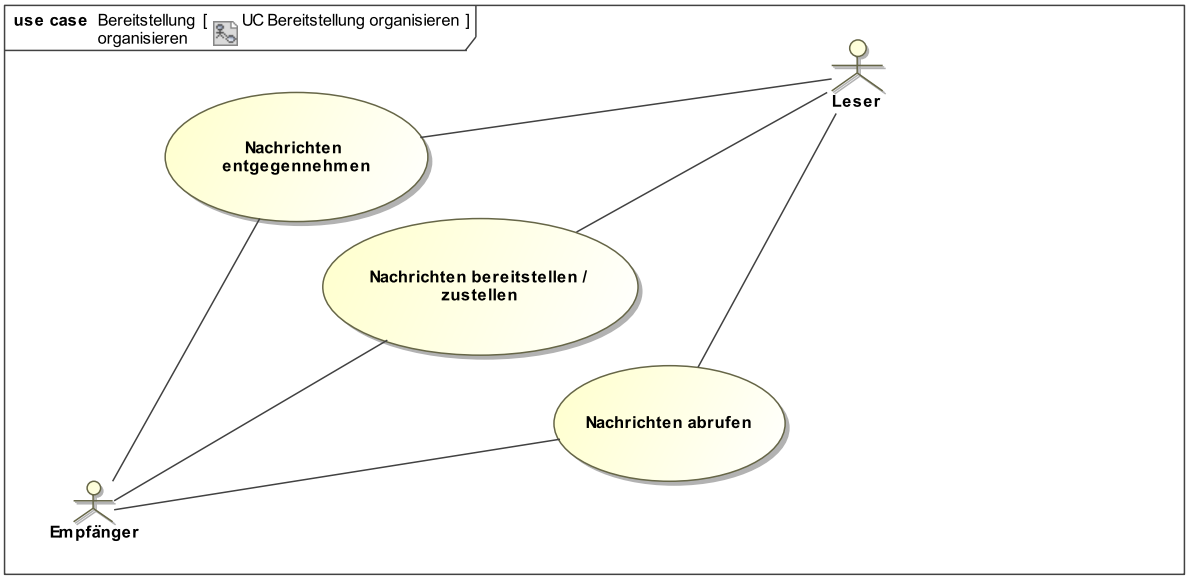
3.2.9.1 Enthaltene Anwendungsfälle

3.2.9.1.1 Übermittlungsfristen überwachen

Verwendete Artefakte		
Artefakt	**Ref.**	**Seite**
TransportReport	5.5.2.4	156
Beschreibung		
Der Autor ist für die Überwachung der Übermittlung und für die Einhaltung der (rechtlich- organisatorischen Vorgaben für) Übermittlungfristen der Nachricht an den Empfänger bzw. Leser zuständig. Für diese Aufgabe nimmt er Einsicht in die vom Sender zu führenden Protokolle.		

3.2.10 UC Bereitstellung organisieren

Abbildung 3.10. Anwendungsfalldiagramm "UC Bereitstellung organisieren"



Der Empfänger stellt dem Leser die nötigen Dienste nach Bedarf bereit. Dazu zählt die Nachrichtenkommunikation im engeren Sinne (Nachrichten bereitstellen oder zustellen), aber auch Kontroll- und Prüfdienste, die die Kommunikation absichern.

3.2.10.1 Enthaltene Anwendungsfälle

3.2.10.1.1 Nachrichten entgegennehmen

Beschreibung
Der Empfänger ist vom Leser mit der Entgegennahme von Nachrichten beauftragt. Der Empfänger nimmt die Nachrichten vom Sender an der Schnittstelle zur Transportinfrastruktur entgegen und verfährt mit ihr entsprechend der Vorgaben des Transportauftrags.

3.2.10.1.2 Nachrichten bereitstellen / zustellen

Beschreibung
Je nach Vorgaben im Transportauftrag verfährt der Empfänger mit den entgegengenommenen Nachrichten: Er stellt sie für den Abruf durch den Leser bereit bzw. stellt sie direkt zu (in synchronen Kommunikationsszenarien).

3.2.10.1.3 Nachrichten abrufen

Beschreibung
Der Leser ruft die bereitgehaltenen Nachrichten vom Empfänger ab. Asynchrones Kommunikationszenario: Der Leser ist verpflichtet, Nachrichten und Transportinformationvom Empfänger abzurufen oder entgegenzunehmen.

Beschreibung
Synchrones Kommunikationsszenario: Der Leser bedient die Anfrage des Autors unmittelbar.

3.3 Zentrale Artefakte beim Nachrichtenaustausch

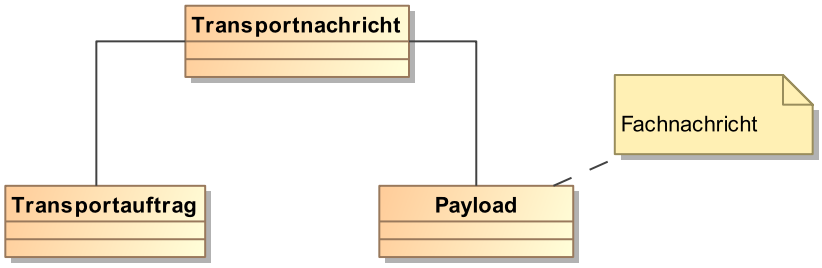
3.3.1 Transportnachricht

Artefakt: *Transportnachricht*

Eine Transportnachricht besteht aus dem Transportauftrag (Abschnitt 3.3.4 auf Seite 46) und dem zugehörigen Payload (Abschnitt 3.3.3 auf Seite 45), wie dargestellt in Abbildung 3.11, „Die Struktur der Transportnachricht“.

Im XTA-Umfeld verwendete Standardformate für Transportnachrichten sind aufgelistet in Abschnitt A.2.15, „Schlüsseltabelle Transportnachrichten Format“

Abbildung 3.11. Die Struktur der Transportnachricht



Attribute von *Transportnachricht*		
Attribut	**Typ**	**Häufigkeit**
	Payload	1
	Transportauftrag	1

3.3.2 Spezifikation Fachstandard

Artefakt: *Spezifikation Fachstandard*

Die Spezifikation des Fachstandards ist ein Regelwerk, das die kollaborativen Prozesse im Kontext des Nachrichtenaustauschs definiert.

Die Spezifikation liegt immer in einer anzuwendenden Version vor, durch die die Regeln des Datenaustausches für Syntax und Semantik definiert sind.

3.3.3 Payload

Artefakt: *Payload*

Der Payload ist der fachliche Inhalt der Transportnachricht, der vom Autor für den Leser erstellt wird. Er umfasst die Gesamtheit der vom Autor für den Leser bestimmten Informationen. Wenn eine XÖV-Nachricht zu übermitteln ist (vgl. Abschnitt 2.1.2.1 auf Seite 11), ist der Payload die (komplette) XÖV-Nachricht.

Der Payload kann vom Autor für den Leser verschlüsselt werden. Deswegen muss der Sender seine Aufgaben mit ausschließlicher Kenntnis des Transportauftrags (ohne Payload) erfüllen können.

Attribut von *Payload*		
Attribut	**Typ**	**Häufigkeit**
	Transportnachricht	1

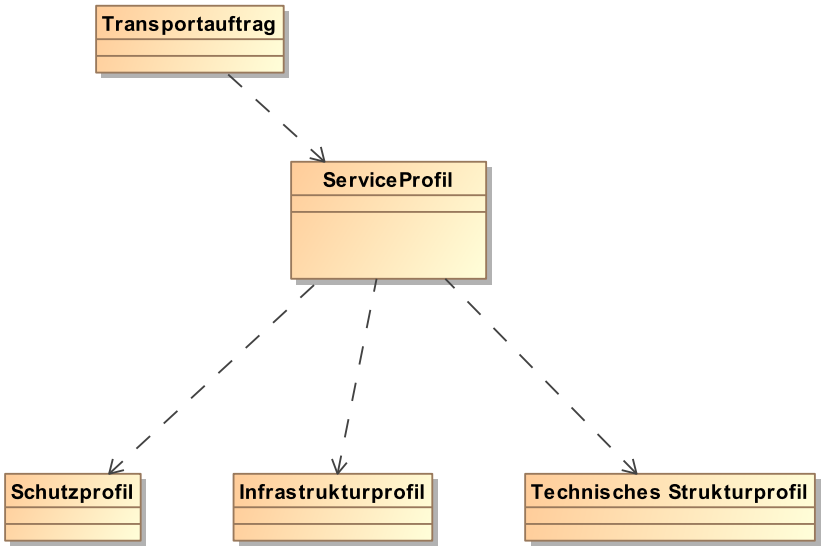
3.3.4 Transportauftrag

Artefakt: *Transportauftrag*

Der Transportauftrag enthält alle erforderlichen Angaben, um die fachliche Nachricht gemäß der Intention des Autors zum Empfänger zu transportieren. Der Autor muss die Konsistenz von Transportauftrag und Fachnachricht sicherstellen, insofern Informationen (z. B. Identität des Lesers für diese Fachnachricht) sowohl in der Fachnachricht als auch im Transportauftrag dargestellt werden. Über den Transportauftrag wird die Qualität der Protokollierung der beteiligten Systeme unter Angabe des anzuwendenden Schutzprofils (vgl. Abschnitt 4.4.1 auf Seite 57) gesteuert. Eine Darstellung des Zusammenhangs des Transportauftrags zu den benötigten Profilen ist dargestellt in Abbildung 3.12, „Transportauftrag mit Referenzen auf die Profile“

Jeder Transportauftrag ist eindeutig identifizierbar. Der Transportauftrag wird durch das Objekt MessageMetaData des XTA-Webservice repräsentiert (vgl. Abschnitt 5.4.2.3.1 auf Seite 122.

Abbildung 3.12. Transportauftrag mit Referenzen auf die Profile



Attribut von *Transportauftrag*		
Attribut	**Typ**	**Häufigkeit**
	Transportnachricht	1

3.3.5 ServiceProfil

Artefakt: *ServiceProfil*

Ein Service Profil ist für die einzelnen fachlichen Domänen zugeschnitten. Es beschreibt für gegebene Schutzprofile und andere Anforderungen die benötigte Konfiguration der Infrastrukturkomponenten des vorgegebenen Infrastrukturprofils (siehe Abschnitt 4.4.6 auf Seite 65).

3.3.6 Schutzprofil

Artefakt: *Schutzprofil*

Ein Schutzprofil wird auf der Basis einer Schutzbedarfsfeststellung ausgewählt. Es bündelt technische und organisatorische Anforderungen, die erfüllt sein müssen, um den Schutzbedarf abzudecken, siehe auch Abschnitt 4.4.1 auf Seite 57.

3.3.7 Infrastrukturprofil

Artefakt: *Infrastrukturprofil*

Ein Infrastrukturprofil ist eine Zusamenstellung von durch den IT-Planungsrat betriebenen IT-Komponenten (siehe Abschnitt 4.4.2 auf Seite 59).

3.3.8 Technisches Strukturprofil

Artefakt: *Technisches Strukturprofil*

Im Technischen Strukturprofil sind Regeln festgelegt für die Implementierung (Kryptographie und Containeraufbau) im Rahmen des verwendeten Transportnachrichten-Formats (siehe Abschnitt 4.4.3 auf Seite 60).

3.3.9 vertragliche Vereinbarungen

Artefakt: *vertragliche Vereinbarungen*

Sender und Empfänger führen ihre Dienste im Auftrag der Fachverantwortlichen (Autor und Leser) aus. Dienstleistungsverträge mit entsprechenden Vereinbarungen zu den Auftragskonditionen regeln den Ablauf der Kooperation.

3.3.10 XML Schema

Artefakt: *XML Schema*

Das in der Sprache XML Schema definierte Artefakt zur Version des Fachstandards enthält einen Teil der Regeln der Spezifikation. Es lässt sich verwenden, um mechanisch die Konformität der zu übermittelnden Nachricht mit diesen Regeln zu prüfen

4 XTA Service Profile (1.1)

4.1 Steuern durch Service Profile

Die Verwaltung muss stets gewährleisten, dass die von ihr beauftragte IT-Infrastruktur für Nachrichtenübermittlung ordnungsgemäß eingesetzt wird. Sie muss in der Lage sein diesen Einsatz zu steuern, also wirksam vorzugeben wie er auszuführen ist.

Viele Punkte sind davon betroffen. Hier sollen einleitend nur einige Aspekte hervorgehoben werden:

- **Datenschutz**: Die Nachrichtenübermittlungen müssen auf allen Strecken vom Autor bis zum Leser so ausgeführt werden, dass auf personen- und organisationsbezogene Daten nur konform mit den rechtlichen Grundlagen zugegriffen wird und nur durch solche Teilnehmer, die dazu autorisiert sind.
- **Datensicherheit**: Zu übermittelnde Daten müssen gegen unbefugte Einsichtnahme und Verfälschung abgesichert werden. Die elektronischen Identitäten müssen passend abgesichert sein.
- **Infrastruktur**: Es ist zu bestimmen, welche Komponenten der verfügbaren Infrastruktur in welcher Konstellation für den Geschäftsprozess zu verwenden sind.
- **Art der Kommunikation**: Es ist zu steuern, wie das interaktive Muster eines Prozesses der Nachrichtenübermittlung aussehen soll. Handelt es sich um einen synchronen oder asynchrone Interaktion, welche Art Antwort wird erwartet?
- **Erforderliches Leistungsniveau**: Was ist die Verfügbarkeit der Dienste, die zu gewährleisten ist? Welche Fristen der Zustellung sind einzuhalten?
- **Aufbau der Transportnachrichten**: Wie sind die Transportnachrichten aufzubauen und wie sind in diesem Zusammenhang die kryptographischen Mittel der verschlüsselung und der Signierung einzusetzen?

Zu jedem dieser Aspekte liegt eine Vielzahl von Eigenschaften vor, die für die Durchführung einer Nachrichtenübermittlung zu bestimmen sind. Die Eigenschaften, von denen hier die Rede ist, werden die *Service Qualitäten* der Nachrichtenübermittlung genannt.

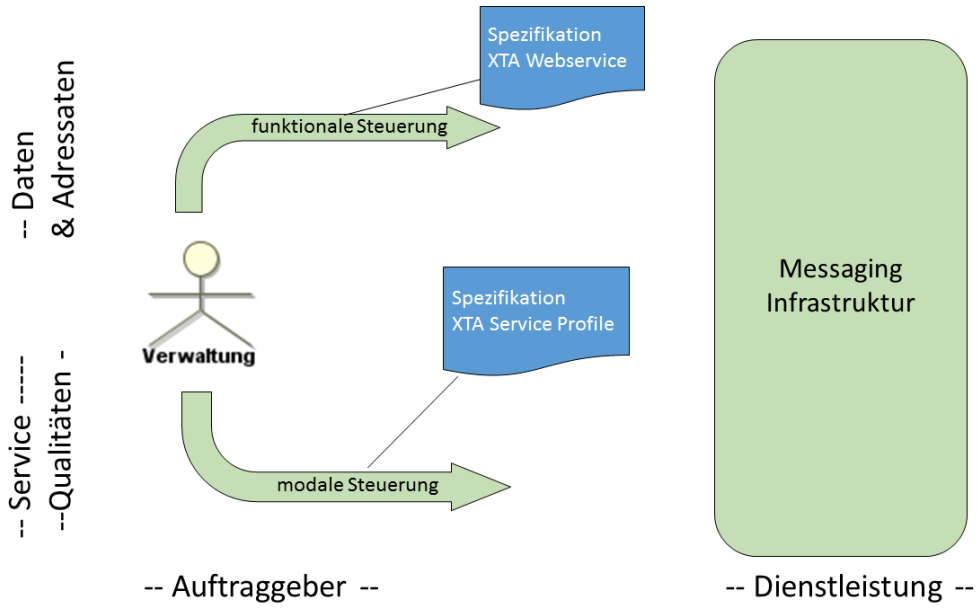
Um ihre Prozesse der Nachrichtenübermittlung im beschriebenen Sinne zu steuern, muss die Verwaltung also vorgeben, welche Service Qualitäten in welcher Ausprägung auszuführen sind. Und sie muss die Einhaltung dieser Service Qualitäten im Nachhinein überprüfen können.

Bei der Bestimmung von Service Qualitäten, von der im vorliegenden Kapitel die Rede ist, geht es um die *Modalitäten* des Einsatzes der IT-Infrastruktur für Nachrichtenübermittlung. Diese sollten nicht in Vergessenheit geraten gegenüber den *funktionalen* Eigenschaften dieses Einsatzes ('Nachricht hat mit den richtigen Daten den korrekten Adressaten erreicht'). Die beiden Dimensionen der Steuerung sind in Abbildung 4.1, „Die zwei Dimensionen der Steuerung der IT-Dienstleistung“ dargestellt.

Klare Vorgaben in Bezug auf Service Qualitäten zu machen ist in der aktuellen Praxis nur eingeschränkt möglich, verschiedene Missstände lassen dies deutlich werden:

- Es ist eine große Vielfalt an Vorgaben anzutreffen, auf deren Basis IT-Dienstleister die Nachrichtenübermittlung zu steuern versuchen.
- Vorgaben sind selten explizit und nachvollziehbar formuliert. Vielmehr sind sie auf unterschiedliche Weise in die Transportverfahren eingebaut.
- Fachlich-rechtliche und technische Vorgaben werden in der Kooperation von Fachstandard und IT-Infrastruktur vermischt, also die jeweils zuständigen Rollen und Kompetenzen nicht differenziert und zugewiesen.
- Viele der Service Qualitäten werden mittels handgeschriebener OSCI-Transport-Profile innerhalb der Fachstandards dokumentiert.

Abbildung 4.1. Die zwei Dimensionen der Steuerung der IT-Dienstleistung



Um in diesem Bereich Effizienz und Rechtskonformität zu verbessern, wird im vorliegenden Kapitel das Instrument der *XTA Service Profile* spezifiziert.

XTA Service Profile sind XML-Dokumente, die auf einer in XTA 2 standardisierten XML Schema-Definition basieren. Sie werden genutzt, um die benötigten Service Qualitäten eindeutig, einheitlich und medienbruchfrei auswertbar zu definieren.

Dieses Kapitel führt das Konzept der XTA Service Profile ein und gibt anschließend die Definitionen im Detail. Es werden in Abschnitt 4.2 auf Seite 51 zunächst die wichtigsten Leistungsmerkmale des Profilkonzepts erklärt. In Abschnitt 4.3 auf Seite 52 wird erläutert, wie Service Profile eines Fachstandards bereitgestellt werden und welche Beiträge dazu der Fachstandard leistet aufbauend auf Angeboten, die in der Zuständigkeit des IT-Planungsrates vorbereitet und verfügbar gemacht werden. Was der fachliche Inhalt der Profile ist, wird erläutert in Abschnitt 4.4 auf Seite 56, und schließlich in Abschnitt 4.5 auf Seite 66, wie Profilobjekte zur Laufzeit genutzt werden durch die verschiedenen Rollen, die am Nachrichtenaustausch in der Infrastruktur beteiligt sind. Nach diesen Vorbereitungen wird im umfangreicheren Abschnitt 4.6, „Struktur der Profile“ das Datenmodell definiert und im Detail erläutert.

4.2 Ziele des XTA Profilkonzepts

Um die Verwaltung in die Lage zu versetzen, die Modalitäten des Einsatzes ihrer IT-Infrastruktur für Nachrichtenübermittlung zu steuern, stellt der Standard XTA 2 das Instrument der **XTA Service Profile** bereit.

Drei Ziele sollen damit erreicht werden:

1. **Einheitlichkeit und Eindeutigkeit**: Die Bestimmung der Service Qualitäten soll auf der Basis eines Standards vorgenommen werden können.
2. **Steuerbarkeit und Überprüfbarkeit**: Die Einhaltung von rechtlichen Anforderungen soll nachvollziehbar gesteuert und überprüft werden können.
3. **Abtrennung von der Technik**: Im Bereich der Service Qualitäten soll die Festlegung der Transporttechnik aus der Fachlichkeit herausgelöst werden.

(1) Standard für die Festlegung von Service Qualitäten

Das Konzept der XTA Service Profile definiert, wie geforderte Service Qualitäten eindeutig zu formulieren sind. Es stellt hierfür eine standardisierte begriffliche Struktur bereit, d.h. definierte Begriffe für Service Qualitäten mit definierten Ausprägungen. Außerdem gibt das Konzept vor, durch wen die Profile zu erstellen sind und wie diese Informationen verfügbar gemacht werden. Im Ergebnis sind die Voraussetzungen geschaffen, dass Anforderungen in Bezug auf die Service Qualitäten einheitlich formuliert und gebündelt abgelegt werden können.

Das schafft eine deutliche Verbesserung gegenüber der aktuellen Praxis, welche gekennzeichnet ist durch folgende Merkmale:

- Die Anforderungen an die Übermittlung von Daten im E-Government werden naturgemäß im jeweiligen rechtlichen Rahmen festgelegt: Der Gesetz- oder Verordnungsgeber legt fest, unter welchen Umständen eine Datenübermittlung zulässig ist und welche Qualität bzgl. Leistungsfähigkeit, Datensicherheit und Datenschutz und anderen Modalitäten erwartet wird.
- Diese Aspekte werden innerhalb juristischer Texte formuliert, zur Umsetzung müssen sie interpretiert und aus der juristischen in eine technische Sprache übersetzt werden. Diese Herleitungen und Übersetzungen sind i.d.R. nicht ohne weiteres nachvollziehbar und können bei gleichen oder sehr ähnlichen Ausgangssituationen stark voneinander abweichen.
- Die Erwartung der Verwaltung, dass die Umsetzung und Einhaltung dieser Anforderungen leicht nachvollziehbar und überprüfbar sind, kann so nicht erfüllt werden. Das hat die Konsequenz, dass es keine einfach zu kontrollierenden Bedingungen zwischen den Kommunikationsendpunkten gibt.

Hier schaffen die standardisierten XTA Service Profile Abhilfe, indem sie eine einheitliche Sprache zur Formulierung der in einem bestimmten Kontext geforderten Service Qualitäten an die Hand geben. Die Fachseite kann nun die durch den jeweiligen rechtlichen Rahmen vorgegebenen Anforderungen nachvollziehbar formulieren, und die Seite der IT-Dienstleister erhält Orientierung durch eindeutige Vorgaben.

Ein gewünschter Effekt dieser Einheitlichkeit ist, dass sie zu größerer Einfachheit führt :

- Die Definition der Service Profile geht mit einer Standardisierung der durch den rechtlichen Rahmen vorgegebenen Attribute einher, so dass die Anzahl der Service Profile gering und damit der Aufwand der Pflege der Profile überschaubar bleibt. Die Idee ist, dass der Gesetz- oder Verordnungsgeber für ein konkretes Übertragungsszenario aus einer definierten Menge das passende Profil auswählt, in dem alle für diesen Kontext geforderten Service Qualitäten festgelegt sind.
- Es wird erwartet, dass sich durch die Service Profile die heute bestehende Vielfalt von Anforderungen stark reduzieren lässt. Das gilt einerseits für die Festlegung von Anforderungen an eine zu beauftragende Datenübermittlung. Das gilt aber auch für die Konfiguration der technischen Umsetzung die-

ser Anforderungen; auch hier sind wiederkehrende Standardkonfigurationen zu erwarten, deren Abruf durch die Transportverfahren ein großes Potential an Reduzierung von Komplexität bedeuten kann.

(2) Steuerung und Überprüfung der Einhaltung von rechtlichen Anforderungen

XTA Service Profile machen nachvollziehbar, wie rechtliche Anforderungen in definierte Service Qualitäten überführt werden. Dies wird nachvollziehbar, weil die Definition eines Service Profils für einen Fachstandard in jedem inhaltlichen Bereich des Profils eine begründete Auswahl trifft unter den durch den Standard angebotenen Alternativen.

Steuerung und Überprüfung der Einhaltung von Service Qualitäten bedeutet dann gleichzeitig die Steuerung und Überprüfung der entsprechenden rechtlichen Anforderungen.

(3) Entkopplung der Transporttechnik von der Fachlichkeit

Bisher werden die technischen Parameter, die für den Transport benötigt werden, vielfach durch den Fachstandard, wie z. B. im OSCI-Transport-Profil für XMeld (vgl. *OSCI-XMeld 2.0 vom 31.07.2014, Anhang C*) vorgegeben. Dies widerspricht der Entkopplung der Transporttechnik von der Fachlichkeit.

XTA Service Profile sollen eine saubere Trennung der technischen Transportimplementierung von der Fachlichkeit möglich machen:

- Für einen fachlichen Prozess der Nachrichtenübermittlung werden die Anforderungen an die Transportinfrastruktur in Form technikneutral formulierter Service Qualitäten spezifiziert. Hier ist noch nicht von der technischen Konfiguration der Transportinfrastruktur die Rede.
- Zu dieser Konfiguration gelangt man durch einen separaten Schritt, der explizit kenntlich macht, wie gemäß eines bestimmten Standes der Technik die Service Qualität in die passende technische Konfiguration von Nachrichtenaufbau und -übermittlung überführt wird. Mehr dazu in Abschnitt 4.4.3, „Technische Strukturprofile“.

4.3 Umsetzung und Zusammenwirken mit den Fachstandards

Die Struktur der Profile ist vom Standard XTA 2 vorgegeben. Diese Vorgaben sind die Syntax und die Semantik der einheitlichen 'Sprache', in der die geforderten Service Qualitäten zu formulieren sind, wie oben in Abschnitt 4.2, „Ziele des XTA Profilkonzepts“ angekündigt.

Es sind die fünf Profilarten *Schutzprofil*, *Infrastrukturprofil*, *Technisches Strukturprofil*, *Kryptographieprofil* und *Service Profil* definiert:

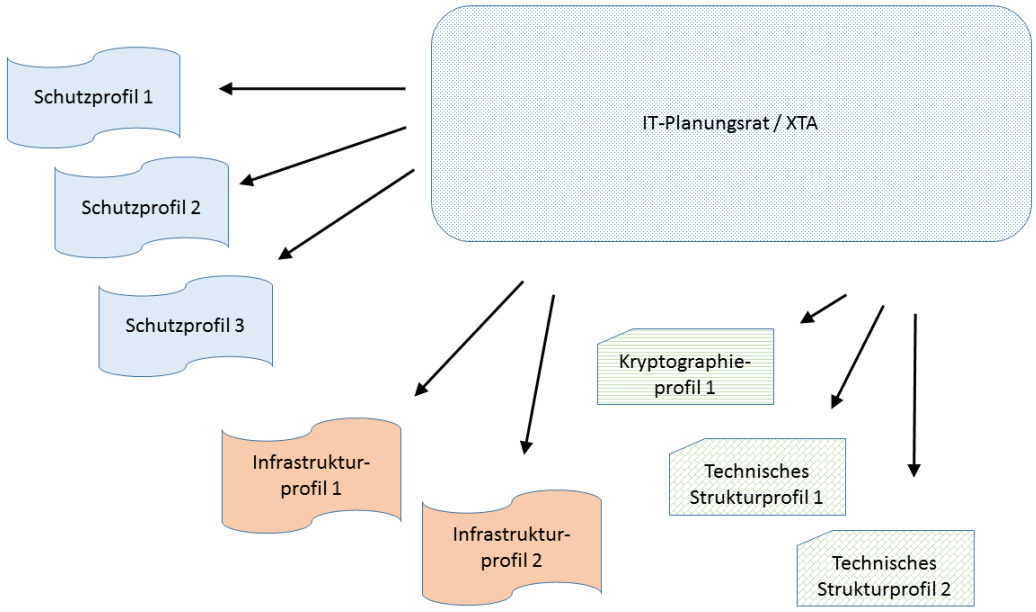
- *Schutzprofile*: Diese fassen Service Qualitäten zusammen, die sich auf IT-Sicherheit und Datenschutz beziehen.
- *Infrastrukturprofile*: Diese fassen Komponenten der Infrastruktur zusammen, die im Zuständigkeitsbereich des IT-Planungsrat betrieben werden.
- *Technische Strukturprofile*: Diese enthalten Festlegungen zur technischen und kryptographischen Konfiguration der Nachrichten.
- *Kryptographieprofil*: Dieses definiert (pauschal und serviceübergreifend) die zu verwendenden kryptographischen Mittel.
- *Service Profile*: Diese enthalten Referenzen auf Profilobjekte von drei anderen Profilarten (Schutz-, Infrastruktur- und Technisches Strukturprofil). Außerdem enthalten sie Einträge zu weiteren Service Qualitäten, die sich direkt aus fachlichen Eigenschaften des Dienstes, um den es geht, ergeben.

Die Datentypen zur Bildung der fünf Profilarten werden weiter unten in Abschnitt 4.6 auf Seite 69 spezifiziert und sind in Form von W3C XML Schema Definitionen abgebildet. Die entsprechenden XSD-Dateien sind im Auslieferungsumfang des Standards XTA 2 enthalten.

Wie und durch wen sind auf der Basis dieser Definitionen die benötigten Profilobjekte zu erstellen und bereitzustellen? Hier spielen Veröffentlichungen im Zuständigkeitsbereich des IT-Planungsrat zusammen mit der Erstellung von Produkten, die durch die Fachstandards auszuliefern sind.

Abbildung 4.2, „Bereitstellung von Standard-Profilobjekten im Zuständigkeitsbereich des IT-Planungsrats“ fokussiert auf die Seite 'IT-Planungsrat'. Im Rahmen seines Zuständigkeitsbereichs werden die benötigten Objekte der Profilarten *Schutzprofil*, *Infrastrukturprofil*, *Technisches Strukturprofil* und *Kryptographieprofil* produziert und bereitgestellt.

Abbildung 4.2. Bereitstellung von Standard-Profilobjekten im Zuständigkeitsbereich des IT-Planungsrats



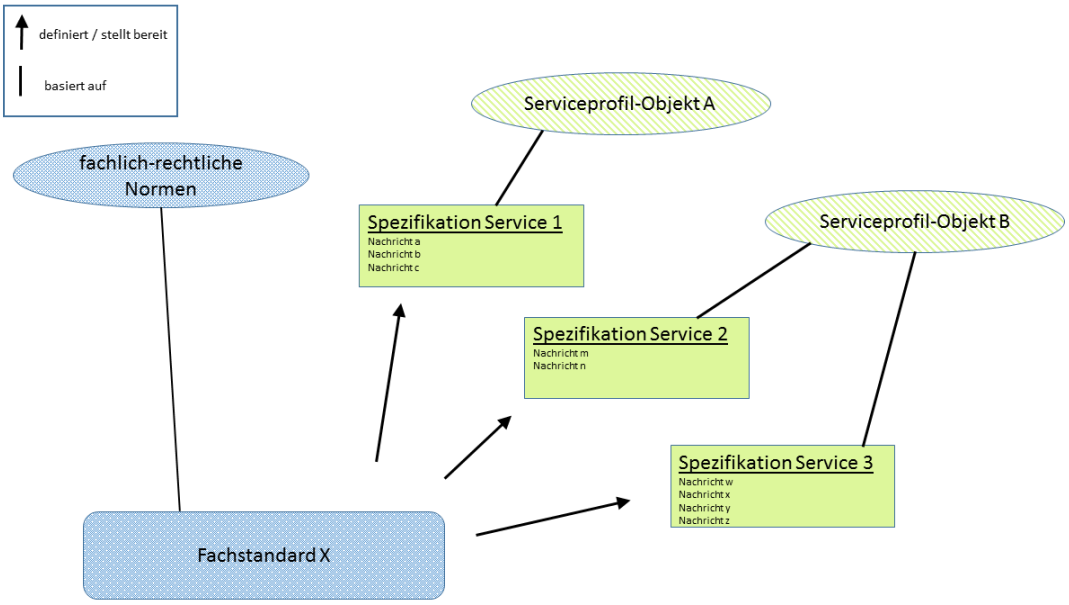
Folgende Punkte sind dazu hervorzuheben:

- Ein **Profilobjekt** ist die Instantiierung einer Profilart, wie sie im Standard XTA 2 definiert ist (eine XML-Instanz z. B. der in XTA 2 enthaltenen XSD-Profildefinition 'Schutzprofil'). Profilobjekte werden auf der Grundlage des Standards XTA 2 als eigenständige Produkte erstellt und herausgegeben. In der Abbildung sind "Schutzprofil 1" oder "Technisches Strukturprofil 2" solche Profilobjekte.
- Im Zuständigkeitsbereich des **IT-Planungsrats** werden autorisierte Profilobjekte produziert und bereitgestellt. In der Abbildung werden diese beispielhaft bezeichnet als Schutzprofile 1-3, Infrastrukturprofile 1 und 2, Technische Strukturprofile 1 und 2 sowie als Kryptographieprofil 1.
- XTA 2 wird in der Abbildung - ergänzend zum IT-Planungsrat - genannt, weil der **Betreiber des Standards XTA 2** sich oft im Auftrag des IT-Planungsrats um die Erledigung der entsprechenden Aufgaben kümmern wird.
- Die in der Abbildung dargestellten Profilobjekte werden in der Überschrift der Abbildung **Standardprofile** bzw. **Standard-Profilobjekte** genannt. Die Intention ist, dass es von ihnen nur einige wenige gibt, die die wichtigsten Kombinationen von angeforderten Service Qualitäten und sonstigen Eigenschaften bündeln. Sie sollen den Bedarf für die allermeisten Fälle abdecken.

Für Objekte der erwähnten vier Profilarten wird also in der Zuständigkeit des IT-Planungsrats gesorgt. In der Zuständigkeit des jeweiligen *Fachstandards* liegt es hingegegen, die nötigen Objekte der Profilart *Service Profil* zu produzieren und bereitzustellen.

Jeder Service, der Bestandteil des Fachstandards ist, muss auf genau *ein* Serviceprofil-Objekt verweisen. Umgekehrt kann ein ServiceProfil-Objekt aber von mehreren Services verwendet werden, wie Abbildung 4.3, „Bereitstellung von Profilobjekten in der Zuständigkeit des Fachstandards“ verdeutlicht.

Abbildung 4.3. Bereitstellung von Profilobjekten in der Zuständigkeit des Fachstandards



Die folgenden Bemerkungen fassen die wichtigsten Punkte dazu zusammen:

- Ein **Service** innerhalb eines Fachstandards ist das, was in dem Fachstandard als nach fachlichen Kriterien zusammenhängender Dienst festgelegt und abgegrenzt ist. Die passende Granularität ist hier also nicht der Fachstandard als Ganzes, sondern seine Einteilung in fachliche Dienste, deren jeder einen oder mehrere Nachrichtentypen des Fachstandards abdeckt. Oft wird zur Laufzeit ein solcher Dienst (in der Webservice-Sprache: „Service“) auf eine WSDL in der Infrastruktur (z. B. DVDV) abgebildet.
- Jeder Fachstandard ist für die **Erstellung der Serviceprofil-Objekte** (XML-Instanzen) zuständig, durch die der Einsatz der Transportinfrastruktur für seine Zwecke konfiguriert werden soll. Die Serviceprofil-Objekte des Fachstandards sind Bestandteil seines Auslieferungsumfangs. In der Abbildung definiert der Fachstandard X drei Services und zu den Services passende Serviceprofil-Objekte.
- Im **Serviceprofil-Objekt A** ist für die Nachrichtenübermittlung festgelegt, welche Service Qualitäten für den **Service 1** in welcher Ausprägung auszuführen sind und wie die Implementierung dieser Service Qualität zu konfigurieren ist. Analoge Inhalte bietet **Serviceprofil-Objekt B** für **Service 2** und **Service 3**.
- Verantwortlich für die Erstellung und Pflege der Serviceprofil-Objekte sind die **Fachgremien des entsprechenden Fachstandards**. Diese Aufgabe erledigen sie, indem sie die bereitgestellten Datentypen der Service Profile wie ein Formular verwenden, das auszufüllen ist. Zu jedem der zu konfigurierenden Service Qualitäten werden dabei die passenden Einträge aus den im Standard enthaltenen Codelisten ausgewählt und eingetragen.

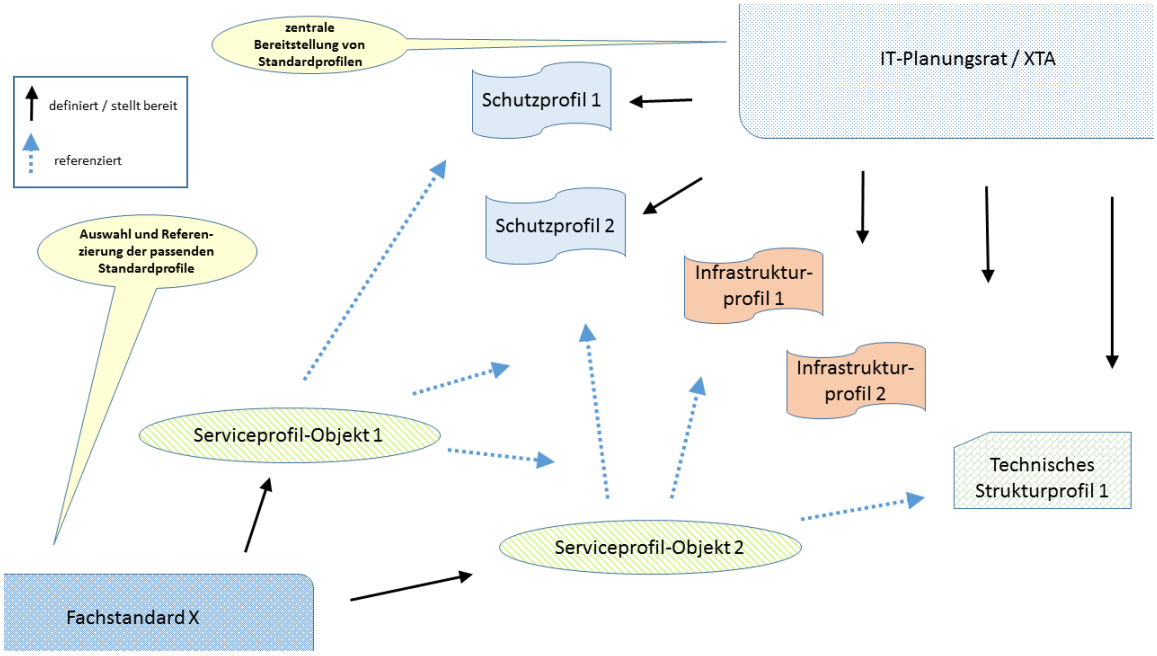
Eine Anmerkung zur Historie: Service Profile lösen im Auslieferungsumfang des Fachstandards das OSCI-Transport-Profil ab. Ein OSCI-Transport-Profil wurde in der Vergangenheit gebraucht, um den

Einsatz der Transporttechnologie OSCI-Transport für die Zwecke des entsprechenden Fachstandards zu konfigurieren. Diese Aufgabe wird, neben anderen, von den XTA Service Profilen übernommen.

Wie spielen nun die Veröffentlichungen im Zuständigkeitsbereich des IT-Planungsrat zusammen mit der Bereitstellung von Serviceprofil-Objekten durch die Fachstandards?

Die Antwort ist: Bei der Erstellung eines benötigten Serviceprofil-Objekts durch einen Fachstandard werden von den schon existierenden Standard-Profilobjekten (Schutzprofile, Infrastrukturprofile, Technische Strukturprofile) einfach die passenden ausgewählt und per Referenzierung in des Serviceprofil-Objekt eingebunden. Diesen Zusammenhang verdeutlicht Abbildung 4.4, „Einbindung von Standard-Profilobjekten per Referenz“.

Abbildung 4.4. Einbindung von Standard-Profilobjekten per Referenz



- Im Zuständigkeit des **IT-Planungsrats** werden autorisierte Standard-Profilobjekte produziert und als XML-Instanzen bereitgestellt (in der Abbildung sind beispielhaft fünf Standard-Profilobjekte dargestellt).
- In der Zuständigkeit des **Fachstandards** sind zwei Serviceprofil-Objekte definiert und bereitgestellt worden.
- **Einbindung Standard-Profilobjekte**: Für die Definition von Serviceprofil-Objekt 1 sind aus den vorgebenen Profilobjekten das Schutzprofil 1, das Infrastrukturprofil 1 und das Technische Strukturprofil 1 ausgewählt worden. Im resultierenden Serviceprofil-Objekt sind im Ergebnis neben weiteren Werten Referenzen eingetragen auf die ausgewählten der vorgegebenen Standardprofile.

Es war bisher von Erstellung und Bereitstellung von Profilobjekten die Rede. Welche technische Realisierung ist hierfür vorgesehen? Alle Profilobjekte werden als XML-Instanzen der definierten Profilarten mit ihren Datentypen erstellt und bereitgestellt. Für diesen Zweck sind im Modell der Service Profile, siehe Abschnitt 4.6 auf Seite 69 die nötigen Wurzelelemente angelegt:

- Schutzprofile werden in Form von XML-Dateien bereitgestellt, deren Wurzelelement festgelegt ist durch das globale Element *schutzProfil*, definiert in Abschnitt 4.6.3.1 auf Seite 103. Bereitgestellt werden die zur Verfügung zu stellenden Schutzprofile im Zuständigkeitsbereich des IT-Planungsrats.

- Infrastrukturprofile werden in Form von XML-Dateien bereitgestellt, deren Wurzelelement festgelegt ist durch das Element *infrastrukturProfil*, definiert in Abschnitt 4.6.3.2 auf Seite 103. Bereitgestellt werden die benötigten Infrastrukturprofile im Zuständigkeitsbereich des IT-Planungsrats.
- Technische Strukturprofile werden in Form von XML-Dateien bereitgestellt, deren Wurzelelement festgelegt ist durch das Element *technischesStrukturprofil*, definiert in Abschnitt 4.6.3.3 auf Seite 103. Bereitgestellt werden die benötigten Technischen Strukturprofile im Zuständigkeitsbereich des IT-Planungsrats.
- Kryptographieprofile werden in Form einer XML-Dateien bereitgestellt, deren Wurzelelement festgelegt ist durch das Element *kryptographieprofil*, definiert in Abschnitt 4.6.3.4 auf Seite 103. Bereitgestellt wird das benötigte Standard-Kryptographieprofil im Zuständigkeitsbereich des IT-Planungsrats.
- Service Profile werden in Form von XML Dateien erstellt, deren Wurzelelement festgelegt ist durch das Element *serviceProfil*, definiert in Abschnitt 4.6.3.5 auf Seite 104. Verantwortlich für die Erstellung der Service Profile sind die jeweiligen Fachgremien. Ein Serviceprofil-Objekt pro im Fachstandard spezifiziertem Service zählt zum Auslieferungsumfang des Fachstandards.

Die KoSIT stellt auf ihren Webseiten prototypische Serviceprofil-Objekte zur Verfügung - als Ergänzung zum Publikationsumfang des Standards XTA 2 (der in Abschnitt 1.3 auf Seite 8 beschrieben wird).

Diese prototypischen Serviceprofil-Objekte sollen den Fachgremien als Vorlagen dienen, an denen sie sich orientieren können, um die Profile zu erstellen, die durch sie für ihre Services zu autorisieren sind.

Die prototypischen Serviceprofil-Objekte verweisen notwendigerweise auf Schutzprofil-, Infrastrukturprofil- und Technische Strukturprofil-Objekte (vgl. Abschnitt 4.4.6, „Aggregation im Service Profil“). Ein Standard-Kryptographieprofil-Objekt wird ebenfalls benötigt (wenn es auch nicht referenziert zu werden braucht). Profilobjekte der letztgenannten vier Arten stehen ebenfalls bereit, entweder in prototypischer Fassung auf den KoSIT-Webseiten (www.xoev.de) oder auch als offiziell durch die KoSIT bzw. den IT-Planungsrat autorisierte Artefakte auf der von der KoSIT betriebenen Distributionsplattform XRepository (www.xrepository.de).

4.4 Komponenten und Inhalt der Profilarten

Das Modell der Service Profile umfasst die genannten fünf Profilarten. Um diese zusammenzusetzen, wird für jede Informationsart - für jede Kategorie von Service Qualitäten - ein Informationsbaustein bereitgehalten, der die entsprechenden Attribute enthält.

Abbildung 4.5, „Komponenten und Bezugsobjekte eines Service Profils“ gibt einen Überblick über die Bausteine, die zur Verfügung stehen.

Die Service Qualitäten der Kommunikationskategorie (Baustein 'Kommunikationsdaten' in der Abbildung) und die Service Qualitäten der Servicekategorie (Baustein 'Servicedaten' in der Abbildung) sind direkt in das Service Profil einzutragen.

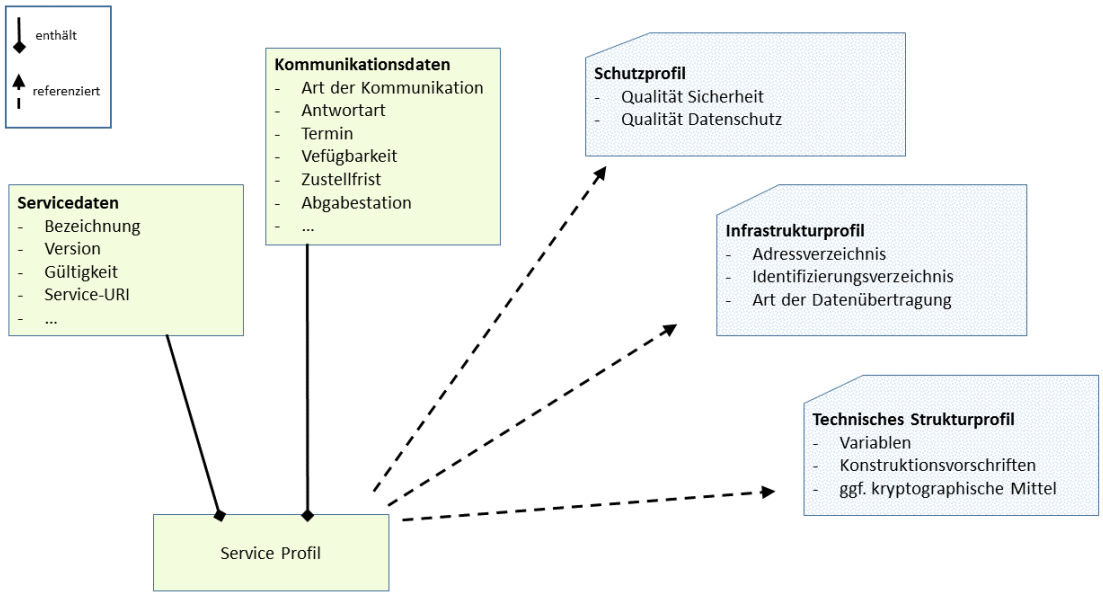
Auf der anderen Seite sind die Service Qualitäten der Schutzkategorie (Baustein 'Schutzprofil' in der Abbildung), der Infrastrukturkategorie (Baustein 'Infrastrukturprofil' in der Abbildung) und der Technischen Strukturkategorie (Baustein 'Technisches Strukturprofil' in der Abbildung) in eigenständigen Profilobjekten enthalten.

Jeder dieser Bausteine wird in den folgenden Abschnitten grob charakterisiert. Zunächst sollen die Profilarten *Schutzprofil*, *Infrastrukturprofil*, *Technisches Strukturprofil* und *Kryptographieprofil* betrachtet werden. Anschließend werden die beiden relevanten Komponenten des Service Profils (*Servicekategorie* und *Kommunikationskategorie*) beschrieben und die Art und Weise, wie all dieses dann in einem ServiceProfil-Objekt zusammenkommt.

Das wird soweit nur eine Beschreibung im Prinzip sein, um dem Leser Orientierung zu geben. Auf den entsprechenden Unterabschnitt der normativen Beschreibung in Abschnitt 4.6 auf Seite 69, wo Auf-

bau und Definition des Bausteins im Detail dargestellt werden, wird in jedem der folgenden Abschnitte verwiesen.

Abbildung 4.5. Komponenten und Bezugsobjekte eines Service Profils



4.4.1 Schutzprofile

Ein Schutzprofil hat Anforderungen aus den Bereichen Datenschutz und Datensicherheit zum Inhalt. Es geht um Service Qualitäten, die mit dem Schutz von Daten vor unberechtigtem Zugriff zu tun haben oder mit Absicherung von Daten gegen Manipulation oder Beseitigung.

Die geforderten Ausprägungen dieser Service Qualitäten werden in ein Schutzprofil-Objekt in dessen Unterbereiche *Schutzkategorie* (für die Anforderungen in puncto Datenschutz) bzw. *Sicherheitskategorie* (für die Anforderungen in puncto Datensicherheit) eingetragen.

Die Service Qualitäten, um die es hier geht, entsprechen eingeführten Grundbegriffen und der Beschlusslage aus der Standardisierung im Bereich des Datenschutzes in Deutschland.[1]

Sicherheitskategorie

In den Unterbereich der Sicherheitskategorie sind die Begriffe *Vertraulichkeit*, *Unveränderbarkeit* und *Authentizität* einer Nachrichtenkommunikation eingeordnet.

Anforderungen an eine Service Qualität „Vertraulichkeit“ beispielsweise haben dabei die folgende Form: Welches Niveau (hoch / normal / niedrig) der Absicherung von Vertraulichkeit wird gefordert auf welcher Teilstrecke der Nachrichtenkommunikation (z. B. auf der Strecke Autor-Sender)?

Analog sind die Anforderungen zu den Service Qualitäten *Unveränderbarkeit* und *Authentizität* einer Nachrichtenkommunikation gefasst.

[1] vgl. „Das Standard-Datenschutzmodell. Konzept zur Datenschutzberatung und -prüfung auf der Basis einheitlicher Gewährleistungsziele“ (in der Version 0.8 beschlossen von der 88. Konferenz der Datenschutzbeauftragten des Bundes und der Länder am 08./09. Oktober 2014 in Hamburg)

Schutzkategorie

Aus der Schutzkategorie seien hier beispielhaft Maßnahmen der Transparenz, wie zum Beispiel die Service Qualitäten *Protokollierung* oder *Technische Quittungen* genannt. Unter Transparenz wird hier die Nachvollziehbarkeit bzw. Nachprüfbarkeit der Nachrichtenkommunikation für einen Auftraggeber oder eine potentielle Aufsichtsbehörde verstanden. Protokollierung und Technische Quittungen sind dafür einige von mehreren Mitteln.

Die Service Qualität *Protokollierung* beschreibt in diesem Kontext, auf welchem Schutzniveau die im Service vorgesehenen Protokolle zu führen sind (z. B. als Standardprotokoll, oder als durch Signatur abgesicherte Protokolle usw.).

Die Service Qualität *Technische Quittungen* würde aussagen, welche Arten technischer Quittungen (z. B. nur Empfangsbestätigung, oder auch Bestätigung der erfolgreichen Weiterleitung an den nächsten Knoten usw.) durch die Knoten der Transportinfrastruktur abzuliefern sind.

Analog werden weitere Service Qualitäten im Rahmen der Schutzkategorie behandelt.

Wiederkehrende Parameter

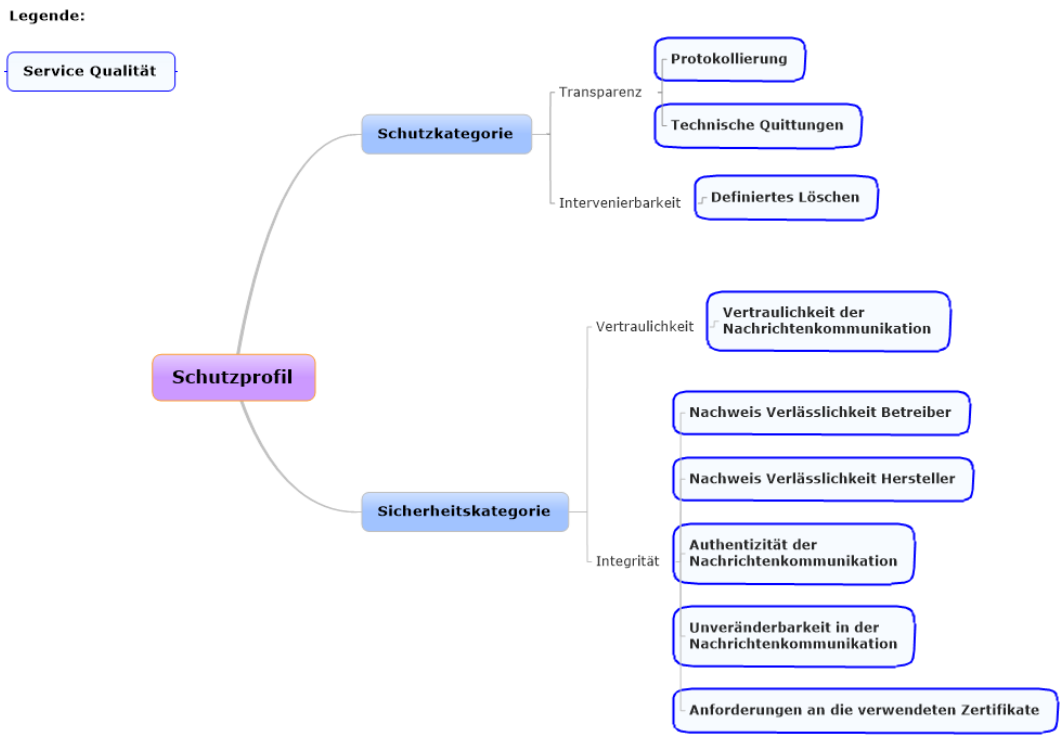
Wiederkehrende Parameter im Datenmodell, die verwendet werden, um die geforderten Service Qualitäten an geeigneten Stellen näher zu spezifizieren, sind:

- ein *Qualifizierer* zur Angabe des geforderten Niveaus bzw. der geforderten Ausprägung dieser Service Qualität (z.B. niedrig, normal, hoch),
- der *Geltungsbereich* der angibt, auf welcher Teilstrecke der Nachrichtenkommunikation (z.B. auf der Teilstrecke Autor-Sender) diese Service Qualität in der genannten Ausprägung erfüllt sein muss und
- eine *Rolle* aus dem XTA-Rollenmodell: Auf welche Rolle / welchen Knoten der Transportinfrastruktur (Autor, Sender, ...) bezieht sich die Forderung dieser Service Qualität in der genannten Ausprägung?

In einem Service Profil wird auf genau *ein* Schutzprofil (genauer: Schutzprofil-Instanz) referenziert. Dieses Schutzprofil muss die Anforderungen zu den genannten Service Qualitäten vollständig enthalten. Das Modell des Schutzprofils ist so strukturiert, dass für eine größere Anzahl unterschiedlicher Rollen und Teilstrecken der Nachrichtenkommunikation jeweils die geforderte Ausprägung einer Service Qualität spezifiziert werden kann.

Einen Überblick bietet Abbildung 4.6, „Die 9 Service Qualitäten des Schutzprofils“. Die Service Qualitäten, die für ein Schutzprofil auszuprägen sind, werden in der Abbildung im Zusammenhang gezeigt, aber nicht weiter erläutert und beschrieben. Die benötigten Definitionen und Parameter zu jeder dieser Service Qualitäten werden im Detail beschrieben (normative Darstellung) im Datenmodell in Abschnitt 4.6.1.1 auf Seite 69.

Abbildung 4.6. Die 9 Service Qualitäten des Schutzprofils



4.4.2 Infrastrukturprofile

Jede Fachlichkeit legt, soweit sie es für erforderlich hält, für den sie betreffenden Nachrichtenaustausch fest, welche der zur Verfügung stehenden Infrastrukturkomponenten in welchem Kontext der Nachrichtenübertragung genutzt werden dürfen bzw. zu nutzen sind.

Hierbei kann es um Infrastrukturkomponenten gehen, die in der Zuständigkeit des IT-Planungsrats betrieben werden, aber auch um solche, die sich unter der Regie eines Landes oder einer kommunalen Organisation befinden.

Der Begriff *Infrastrukturkomponente* wird hier sehr weit verstanden. Messaging-Technologien fallen darunter, auch verfügbare Verzeichnisdienste, sonstige Spezifikationen, Technologien oder Applikationen, die eine Übertragungsinfrastruktur prägen: alles kann in diesem Sinne als Infrastrukturkomponente bezeichnet werden, so wie der Begriff im vorliegenden Profilkonzept verwendet wird.

Eine Infrastrukturkomponente wird in einer Infrastruktur für Nachrichtenübertragung immer in sinnvoller Kombination mit anderen Infrastrukturkomponenten eingesetzt. Insgesamt ergibt sich ein sinnvolles Zusammenwirken eines definierten Sets von Infrastrukturkomponenten. Eine solche definierte Zusammenstellung von Infrastrukturkomponenten wird nach vorliegendem Profilkonzept durch ein *Infrastrukturprofil* repräsentiert.

Ein Infrastrukturprofil ist nichts anderes als eine strukturierte (und als XML-Instanz abgelegte) Liste der Bezeichnungen der betreffenden Infrastrukturkomponenten.

Auf ein oder mehrere solcher Infrastrukturprofile wird in einem Service Profil referenziert. Wenn auf mehr als eines referenziert wird, dann liegt es in der Zuständigkeit des Betreibers der Rolle *Sender*, das für einen gegebenen Transportauftrag geeignete aus der Menge dieser Möglichkeiten auszuwählen.

Ein referenziertes Infrastrukturprofil wird angewendet auf den Umfang des Geltungsbereiches (z. B. bundesweit oder landesintern), der innerhalb des Infrastrukturprofils - als einer von mehreren Parametern - genannt wird.

Die Bestimmungen der Infrastrukturkomponenten, aus denen ein Infrastrukturprofil besteht, bezeichnen wir hier als die Service Qualitäten der XTA-Infrastrukturkategorie.

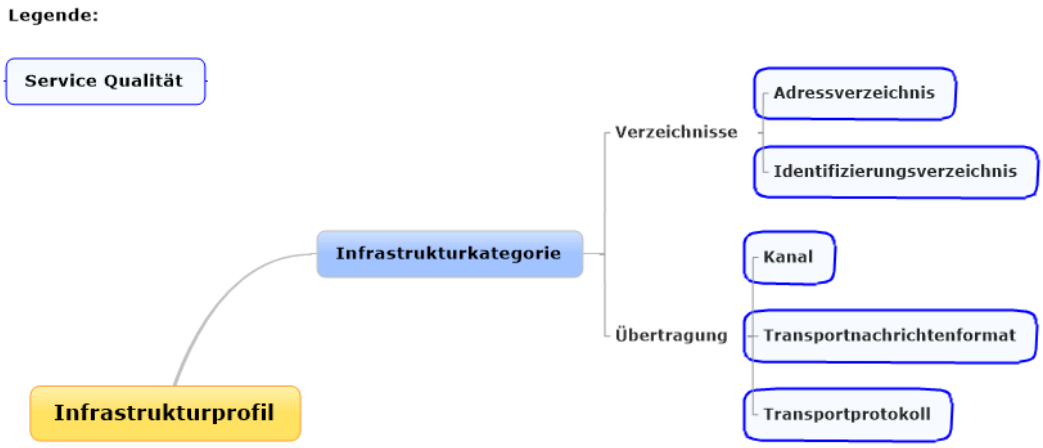
Diese Service Qualitäten werden in Abbildung 4.7, „Die 5 Service Qualitäten des Infrastrukturprofils“ im Überblick gezeigt. Zu jeder dieser Service Qualitäten werden in einer Infrastrukturprofil-Instanz die Ausprägung genannt, die gefordert sind.

Beispiele:

- Service Qualität 'Adressverzeichnis' bedeutet: Welche Infrastrukturkomponente soll die Rolle 'Adressverzeichnis' abdecken?
- Service Qualität 'Transportprotokoll' bedeutet: Welche Messaging-Technologie / welches Transportprotokoll ist für die Datenübertragung zu verwenden?

Die weiter unten gegebene detaillierte normative Darstellung in Abschnitt 4.6.1.2 auf Seite 76 führt jede der hier genannten Service Qualitäten ein und definiert, welche Ausprägungen vorgesehen sind.

Abbildung 4.7. Die 5 Service Qualitäten des Infrastrukturprofils



4.4.3 Technische Strukturprofile

Oben wurde erläutert, dass in den Service Profilen Anforderungen an die Transportinfrastruktur formuliert werden. Dabei wurde betont, dass in dem Zusammenhang die Begriffsebene der fachlichen Anforderung eingehalten ist. Es wird also strikt eine bestimmte Service Qualität (z.B. Vertraulichkeit) in einer bestimmten Ausprägung ("Vertraulichkeit der Stufe 'hoch' auf der Strecke 'Autor-Leser') gefordert. Wie die Ausprägung der Service Qualität technisch umzusetzen ist, ist stark kontextabhängig (verwendete Infrastruktur, Stand der Technik) und wird soweit absichtlich offen gelassen.

An irgendeiner Stelle sind aber natürlich zu den fachlichen Anforderungen die entsprechenden technischen Parameter der Implementierung zu nennen. Und dies soll nicht individuell dem einzelnen Hersteller oder Betreiber eines Transportverfahrens überlassen bleiben, sondern es soll nachvollziehbar und

einheitlich geschehen. Deswegen wird auch dies im Rahmen von zentral zugänglichen Profilobjekten festgelegt.

Die Regeln, nach denen die technischen Parameter determiniert werden, sind durch das *Technische Strukturprofil* abgebildet. Wie das Technische Strukturprofil im Einzelnen aufgebaut ist, wird in Abschnitt 4.6.1.3.1 auf Seite 80 definiert .

Die Regeln, aus denen ein Technisches Strukturprofil besteht, haben die Form "Eine *Service Qualität X* ist in der Ausführung des Nachrichtentransports durch die *technische Konfiguration Y* zu implementieren."

Service Qualität X: Die Service Qualitäten, um deren technische Umsetzung es geht, sind diejenigen der Schutz- bzw. Sicherheitskategorie (siehe Abbildung 4.6, „Die 9 Service Qualitäten des Schutzprofils“).

Technische Konfiguration Y: Die technische Konfiguration, die die Service Qualitäten umsetzen soll, wird in einem Technischen Strukturprofil in Bezug auf zwei Aspekte spezifiziert:

- *Aufbau der Container der Transportnachricht*. Zum einen wird der geforderte Container-Aufbau spezifiziert: Welche Angaben muss das Fachverfahren bei der Übergabe des Transportauftrags an das Transportverfahren wie in den vorgesehenen Containern des Transportauftrags hinterlegen?

  Der vordefinierte Aufbau der Container der entsprechenden Transportnachrichten weist nämlich oft erhebliche Freiheitsgrade auf. Um die Interoperabilität in einem bestimmten Bereich zu gewährleisten, ist es daher hilfreich, aus den vielseitigen Möglichkeiten, die ein Container eines Transportnachrichten-Formats bietet (betrifft XTA-WS gemäß Abschnitt 5.5.2.1, „GenericContentContainer“, betrifft analoge Container in OSCI-Transport 1.2 bzw. OSCI 2) eine ganz bestimmte Möglichkeit auszuwählen und die Implementierung darauf einzuschränken.

  Dies geschieht durch entsprechende Regeln des Technischen Strukturprofils.
- *Die Kryptographische Verarbeitung*. Für die Service Qualitäten aus dem Schutzprofil, die mit kryptographischen Mitteln umzusetzen sind (z. B. Vertraulichkeit, Authentizität und Integrität), ist anzugeben, welche kryptographischen Mittel dies sein sollen (z. B. Signatur oder Verschlüsselung), welche Algorithmen und Schlüssellängen hierfür vorgegeben sind und auf welche Objekte sie an welcher Stelle anzuwenden sind.

  Für diese Zwecke sind in erster Linie die allgemeinen Krypto-Suiten maßgeblich, die im übergreifend bereitgestellen Kryptographie-Profil spezifiziert sind (siehe Abschnitt 4.4.4 auf Seite 63). Aber in einigen Fachkontexten bedarf es spezieller Regelungen: In diesem Fall werden diese *besonderen* kryptographischen Mittel ins Technischen Strukturprofil eingetragen. Die entsprechenden Regeln der allgemeinen Krypto-Suite (laut Kryptographie-Profil) werden dadurch in ihrer Wirksamkeit für den Geltungsbereich dieses Technischen Strukturprofils ersetzt; die Eintragungen im Technischen Strukturprofil haben dann also Vorrang.

Wie bildet ein Technisches Strukturprofil seine Regeln ab?

Aufbau der Regeln: Die Regeln sind aus zwei Abschnitten aufgebaut. Der erste ist der Deklaration von Variablen gewidmet, der zweite definiert Anweisungen bzw. Konstruktionsvorschriften, in denen die Variablen verwendet werden (vgl. Abschnitt 4.6.1.3.2 auf Seite 80). Dies wird in den folgenden Unterabschnitten erläutert (siehe Abschnitt 4.4.3.1 auf Seite 62 sowie Abschnitt 4.4.3.2 auf Seite 62).

Krypto-Suiten: Bei der Ausführung der Regeln tritt die Anwendung von Krypto-Suiten hinzu. Solche Krypto-Suiten müssen als Vorbedingung bereitgestellt sein; das gilt unabhängig davon, ob sie innerhalb des Technischen Strukturprofils oder im Kryptographieprofil (beides ist möglich) definiert sind.

Was Krypto-Suiten beitragen, wird in einem separaten Unterkapitel eingeführt und erläutert (siehe Abschnitt 4.4.4 auf Seite 63). Dort wird auch beschrieben, unter welchen Bedingungen Krypto-Suiten innerhalb eines Technischen Strukturprofils zu definieren sind und wie sie dann anzuwenden sind.

4.4.3.1 Deklaration von Variablen

Im Bereich der Variablendeklaration des Technischen Strukturprofils werden die von der Fachlichkeit für die Regeln des Technischen Strukturprofils anzugebenden Informationen bzw. Informationsblöcke benannt und mit einem Namen und einem Typ versehen (vgl. Abschnitt 4.6.1.3.3, „Variables“).

Einige der Informationen bzw. Informationsblöcke werden vom Fachverfahren bereitgestellt (die zu versendende Fachnachricht, Identität des Empfängers, die Bezeichnung des verwendeten Service Profils u.a.), andere werden a priori für diesen Kontext von der Fachlichkeit allgemeingültig festgelegt (z. B. Konstanten wie die Festlegung des Bezeichners für den OSCI 1.2 - Container).

Ein solcher Abschnitt mit Variablendeklarationen muss immer vorhanden sein. Er besteht aus einer Liste von Deklarationen, die die Objekte benennen, die für den Aufbau und die Bearbeitung eines Transportauftrags benötigt werden.

Es sind drei Arten von Variablen-Deklarationen angelegt:

- für **Konstanten** (vgl. Abschnitt 4.6.1.3.4, „Constant“)

  Die Deklaration einer Konstanten besteht aus dem Namen der Konstanten und ihrem Wert, der im Anschluss nicht mehr geändert werden darf.
- für **XML Dokumente** (vgl. Abschnitt 4.6.1.3.5, „DocumentRef“)

  Soll auf relevante Informationen zurückgegriffen werden, die in einem XML-Dokument bereitgestellt sind, so wird das XML-Dokument referenziert. Ihm wird ein Name zugewiesen, zusätzlich wird ein Typ in Form eines XML Schemas angegeben.
- für **Container oder Nachrichtenfragmente** (vgl. Abschnitt 4.6.1.3.6, „MessagePart“)

  Es werden Variablen eingeführt, die sich auf Objekte (Container oder Nachrichtenfragmente) beziehen, die für den Transport verwendet werden sollen.

  Hierzu gehören bei Verwendung der XTA-WS-Schnittstelle stets die Komponenten der XTA-Nachrichten. Wichtig sind z. B. in XTA 2 bzw. OSCI 2 die Container gemäß Abschnitt 5.5.2.1, „GenericContent-Container“ und gemäß Abschnitt 5.4.2.3.1, „Der Transportauftrag: Header-Block *MessageMetaData*“.

  An den Stellen einer Schnittstelle, an denen ein OSCI-Nachrichtenformat im Einsatz ist, zählen dazu aber auch z. B. die OSCI 1.2 - Nachrichten bzw. deren Teile.

  Über die Deklarationen werden die Elemente der Nachrichten referenzierbar. Die Bezeichner werden dann im Anschluss verwendet, um mittels XPath-Ausdrücken auf bestimmte Stellen zu zeigen und dann zu beschreiben, wie diese Komponenten zu aggregieren bzw. kryptographisch zu behandeln sind.

4.4.3.2 Konstruktionsvorschriften

In den Konstruktionsvorschriften (siehe Abschnitt 4.6.1.3.7, „ProcessingList“) wird spezifiziert, was jede XTA-Rolle im Detail informationstechnisch zu tun hat. Dies betrifft die Art der kryptographischen Verarbeitung einerseits und den Containeraufbau für die Objekte im Transportauftrag andererseits.

Beispielsweise wird für den Autor festgelegt, wie er aus seinen fachlichen Daten einen korrekten Transportauftrag aggregiert. Für den Sender ergibt sich, wo er welche Informationen aus dem Transportauftrag in welcher Form zu extrahieren und sie dann in welcher Form weiterzuverarbeiten hat.

Definiert wird eine Sequenz von Konstruktionsvorschriften, auch etwas pauschaler "Anweisungen" genannt. Jede dieser Anweisungen ist als ein separates Objekt definiert (siehe Abschnitt 4.6.1.3.7, „ProcessingList“).

Anweisungen zum Aufbau einer Nachrichtenstruktur: Die Anweisungsart, die in Abschnitt 4.6.1.3.8, „XML_Injection“ beschrieben wird, dient als Anleitung zum Aufbau einer Nachrichtenstruktur. Eine solche Anweisung ist eine Konstruktionsvorschrift, die in eine Nachricht bzw. ein XML-Dokument an einer durch

einen XPath-Ausdruck referenzierten Stelle einen Wert einfügt bzw. das entsprechende Objekt mit dem entsprechenden Wert erzeugt, falls der entsprechende Knoten noch nicht besteht.

Anweisungen zur kryptographischen Verarbeitung: Andere Anweisungen sind Vorschriften, die eine im Service Profil geforderte Service Qualität in eine technische Konfiguration umsetzen (siehe Abschnitt 4.6.1.3.9, „ProcessingInstruction“). Diese Anweisungen werden eingesetzt, um (a) die Signatur über einem Nachrichtenabschnitt zu erzeugen oder (b) diesen zu verschlüsseln, bevor er in einen Container eingefügt wird. Jeweils wird (zusätzlich zur Angabe, ob signiert oder verschlüsselt werden soll) die Service Qualität (z. B. 'Vertraulichkeit') mit ihren Qualifizierern (z. B. 'hoch') angegeben, die umzusetzen ist. Die in diesem Kontext anzuwendende Krypto-Suite wird aus der Liste der gültigen Krypto-Suiten der angegebenen Stärke ausgewählt.

Noch eine Anmerkung zum besseren Verständnis des konditionalen Charakters der Anweisungen: Sie definieren immer *allgemeine Regeln*, z. B. Regeln der Form "Der Nachrichtenabschnitt x ist im Scope der Anwendung dieser TechnischesStrukturprofil-Instanz zur Umsetzung der Service Qualität 'Vertraulichkeit' gemäß Schutzniveau z zu verschlüsseln." Das hat den Vorteil, dass eine solche TechnischesStrukturprofil-Instanz serviceübergreifend einsetzbar ist. Die entsprechende Regel kommt nur zur praktisch wirksamen Anwendung im Fall eines Service, dessen zugeordnetes Schutzprofil tatsächlich die Vertraulichkeit der Stufe z vorschreibt.

4.4.4 Kryptographieprofile

Oben wurde erläutert, dass in den Infrastruktur- und Schutzprofilen Anforderungen an die Kryptographie formuliert werden. Im vorliegenden Abschnitt wird dargestellt, wie das Kryptographieprofil die Implementierung dieser Anforderungen spezifiziert.

Allgemeine Krypto-Suiten

Eine allgemeine Krypto-Suite enthält für jeden Eintrag eines Sets vorgegebener kryptographischer Funktionen (Digests, Signaturen, symmetrische und asymmetrische Verschlüsselung, Cipher-Suiten) die zulässigen Algorithmen und Schlüssellängen.

Dabei behandelt die Krypto-Suite die unterschiedlichen möglichen Schutzniveaus (vgl. Abschnitt 4.6.1.3.10.1 auf Seite 88) separat, d.h. sie definiert für das Set vorgegebener kryptographischer Funktionen die zulässigen Algorithmen und Schlüssellängen und ordnet dieses Set einem Schutzniveau (z. B. dem Schutzniveau "hoch", vgl. Abschnitt A.2.8 auf Seite 171) zu.

Durch diese Zuordnungen, die sich naturgemäß ändern müssen mit dem sich wandelnden Stand der Technik, enthält die Krypto-Suite Regeln für die kryptographische Abbildung bestimmter Schutz-Anforderungen.

Kryptographieprofil

Ein Kryptographieprofil (siehe Abschnitt 4.6.1.4.1 auf Seite 89) gemäß XTA 2 besteht aus der Definition eines Sets *allgemeiner Krypto-Suiten*. Es definiert jeweils eine Krypto-Suite für jede relevante Transporttechnologie und für die kryptographische Behandlung des Payload. Für jeden dieser Kontexte wird also im Kryptographieprofil separat definiert, welches die zulässigen Algorithmen und Schlüssellängen für die kryptographischen Funktionen sind.

Vorgesehen als Kontexte, für die Definitionen von Krypto-Suiten gebraucht werden, sind aus der Sicht des Standards XTA 2 mindestens:

- die Transporttechnologie (Transportnachrichtenformat) *OSCI 1.2*
- die Transporttechnologie (Transportnachrichtenformat) *OSCI 2*
- die Transporttechnologie (Verschlüsselung einer Netzwerkverbindung) *TLS* [2]
- die Nachricht (Fachnachricht als beispielsweise XML oder PDF) im *Payload*

[2] Zu TLS (Transport Layer Security) werden die offiziellen Krypto-Suiten des BSI unterstützt.

Diese Liste ist erweiterbar. Stets werden innerhalb der Krypto-Suite die zulässigen Algorithmen unter Angabe von Typ und Ablaufdatum definiert; optional können auch Schlüssellängen und Algorithmen für die Schlüsselverschlüsselung definiert werden.

Festlegung und Bereitstellung

Das Set der vorgegebenen allgemeinen Krypto-Suiten wird im Zuständigkeitsbereich des IT-Planungsrat festgelegt und als Kryptographieprofil-Instanz als Standard zur Verfügung gestellt.

Wenn in einem bestimmten fachlichen Kontext von diesen Vorgaben abgewichen werden soll, dann sieht das Modell der XTA Service Profile auch hierfür einen Mechanismus vor. Die entsprechenden Regeln können nämlich durch Einträge in Instanzen eines Technischen Strukturprofils überschrieben werden (siehe Abschnitt 4.6.1.3.2 auf Seite 80), sofern in einem bestimmten fachlichen Kontext nicht die standardmäßige Krypto-Suite genutzt werden darf.

Diese Krypto-Suiten sind dann vom Fachstandard zu definieren und fortzuschreiben. Sie haben dann in ihren Kontexten Vorrang vor den Regelungen der standardmäßigen Krypto-Suite.

Anwendung von Krypto-Suiten:

Verwendet werden die Einträge der Krypto-Suiten bei der Interpretation eines Technischen Strukturprofils: Wird hier eine kryptographische Funktion aufgerufen (siehe Abschnitt 4.6.1.3.9 auf Seite 87), dann findet sich darin die Angabe der Qualität der kryptographischen Verarbeitung (vgl. Abschnitt 4.6.1.3.10.1 auf Seite 88). Es sind dann die für diese Qualität zugelassenen Algorithmen auszulesen und zu nutzen.

Dies geschieht naturgemäß entsprechend der Anforderung der XTA-Rolle (vgl. Abschnitt 4.6.1.3.10.2 auf Seite 89) und der Aufgaben, die sich dieser Rolle stellen. Mehr zu den Bearbeitungsschritten aus Sicht der unterschiedllichen Rollen wird erläutert in Abschnitt 4.5 auf Seite 66

4.4.5 Service Qualitäten der Kommunikations- und der Servicekategorie

In den Bausteinen *Kommunikationskategorie* und *Servicekategorie* des Service Profils werden Aspekte definiert, die sich individuell auf Eigenschaften und Anforderungen des Service beziehen, um den es geht. Dies sind also Aspekte, die nicht abgetrennt im Rahmen von Standardprofilen wie Schutzprofil oder Infrastrukturprofil behandelt werden können.

Die Service Qualitäten aus diesen beiden Bausteinen werden in Abbildung 4.8, „Die 8 Service Qualitäten aus Kommunikations- und Servicekategorie“ im Überblick gezeigt. Es geht um die folgenden Themen.

Kommunikationskategorie

In diesem Baustein wird die Art der Interaktion beschrieben, wie sie im Rahmen des Service ausgeführt werden soll.

Beispielsweise gehören hierzu die Angaben, ob die Kommunikation synchron oder asynchron stattfindet, ob direkte Zustellung oder Ablage ins Postfach ansteht, außerdem geht es um Parameter des beauftragten Leistungsniveaus wie Verfügbarkeit, beauftragte Zustellfristen und mehr.

Auch hierzu sind die Service Qualität, um die es geht, in der Abbildung dargestellt. Details und die normativen Festlegungen sind Abschnitt 4.6.1.5.2 auf Seite 95 zu entnehmen.

Servicekategorie

In dieser Kategorie werden zwei Service Qualitäten abgehandelt:

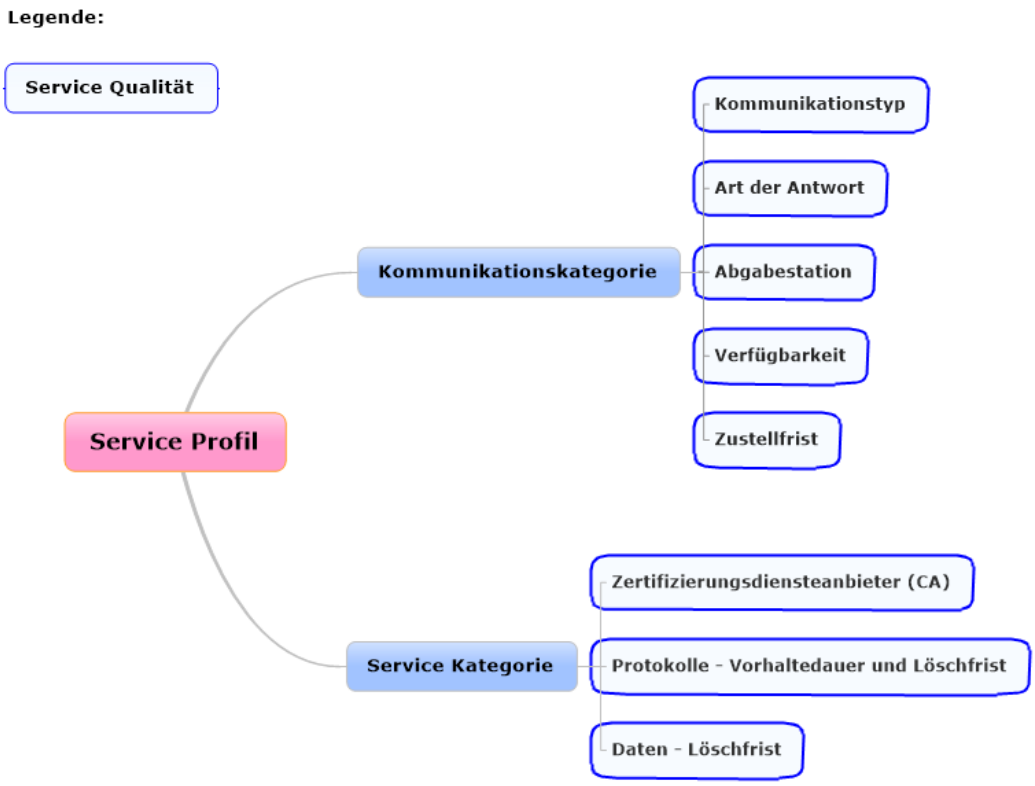
- Die erste Service Qualität betrifft Festlegungen, welche Zertifizierungsdiensteanbieter (CA) zugelassen sind, so dass sie durch die IT-Dienstleister im Rahmen der Messaging-Infrastruktur mit Zertifikaterstellung und Zertifikatprüfung betraut werden können.

- Die zweite Service Qualität bietet die Möglichkeit, für den Service Vorgaben zu den Aufbewahrungs- und Löschfristen von möglicherweise zu erstellenden Protokollen zu formulieren.

Außerdem stehen unterhalb der Servicekategorie Informationen zur *Identität des Service*: Dazu gehört die Bezeichnung des Service, auf den sich das vorliegende Service Profil bezieht, und die Liste der durch diesen Service abgedeckten Nachrichtentypen. Diese Informationen geben keine Service Qualitäten vor, sondern machen lediglich den Bezug des Service Profils zum betreffenden Service explizit.

Die Service Qualitäten der Servicekategorie sind in der Abbildung ohne weitere Erläuterung aufgelistet. Details zu Definitionen und normativen Festlegungen in diesem Zusammenhang sind Abschnitt 4.6.1.5.3 auf Seite 96 zu entnehmen.

Abbildung 4.8. Die 8 Service Qualitäten aus Kommunikations- und Servicekategorie



4.4.6 Aggregation im Service Profil

Schutzprofile, Infrastrukturprofile und Technische Strukturprofile sind eigenständige Profilobjekte, die separat erstellt und als Standardangebote bereitgestellt werden, wie oben erläutert in Abbildung 4.2, „Bereitstellung von Standard-Profilobjekten im Zuständigkeitsbereich des IT-Planungsrats“.

Das Serviceprofil-Objekt muss nun die gewünschten Profilobjekte referenzieren, d. h. dass Referenzen auf die richtigen Objekte der Standardangebote in das Serviceprofil-Objekt bei dessen Erstellung einzutragen sind. Und zwar sind dies: Referenzen auf genau ein Schutzprofil, auf ein oder mehrere Infrastrukturprofile und auf genau ein Technisches Strukturprofil.

Zuvor sind naturgemäß die nach fachlich-rechtlichen bzw. technischen Kriterien richtigen Profilobjekte auszuwählen:

- Die Auswahl des richtigen *Schutzprofils* basiert auf der Feststellung des Schutzniveaus für den betrachteten Service. Diese Auswahl wird im Hinblick auf den Service in der Verantwortung der Fachlichkeit auf der Basis einer Schutzbedarf-Feststellung nach BSI IT-Grundschutz durchgeführt[3]. Es ist anschließend das Schutzprofil-Objekt zu referenzieren, das das identifzierte Schutzniveau abbildet.
- Die Auswahl der richtigen *Infrastrukturprofile* ergibt sich aus den Infrastruktur-Festlegungen der Fachlichkeit, die zum Datenaustausch im Rahmen dieses Service vorliegen. Falls mehrere Infrastrukturen verwendet werden sollen (z. B. differenziert für länderübergreifenden und landesinternen Datenaustausch), sind mehre Profilobjekte zu referenzieren. Die Rolle *Sender* wird dann für einen gegebenen Transportauftrag die passende Infrastruktur unter diesen Möglichkeiten auswählen.
- Die Auswahl des richtigen *Technischen Strukturprofils* erfolgt hauptsächlich nach Kriterien der Einheitlichkeit und der Interoperabilität. In bestimmten Fällen wird es auch hierzu technische Festlegungen der Fachlichkeit geben, die die Auswahl beeinflusst.

Das Service Profil enthält im Ergebnis einerseits die genannten Referenzen, andererseits enthält es unmittelbar die Informationen zu *Kommunikationskategorie* und *Servicekategorie*. Diese Informationen werden von der Fachlichkeit direkt ins Service Profil eingetragen. Zusätzlich enthält ein Service Profil noch einige identifizierende Merkmale wie seine Bezeichnung und Versionsnummer.

Das *Kryptographieprofil* mit seinen Definitionen der einzusetzenden kryptographischen Mittel wird nicht aus dem Service Profil referenziert, sondern ist als allgemeingültiges Regelwerk wirksam.

Es ergibt sich insgesamt der folgende Aufbau eines Service Profils, wie dargestellt in Tabelle 4.1, „Komponenten des Service Profils - Aggregation im Überblick“.

Dieser Aufbau entspricht dem oben in Abbildung 4.5, „Komponenten und Bezugsobjekte eines Service Profils“ verdeutlichten.

Details und normative Darstellung sind Abschnitt 4.6.1.5.1 auf Seite 94 zu entnehmen.

Tabelle 4.1. Komponenten des Service Profils - Aggregation im Überblick

Eigenschaft	Wert
Schutzprofil	Bezeichnung des referenzierten Schutzprofils.
Infrastrukturprofil	Liste von Bezeichnungen der referenzierten Infrastrukturprofile.
Technisches Strukturprofil	Bezeichnung des referenzierten Technischen Strukturprofils.
Kommunikationskategorie	Die ins Service Profil eingetragenen Service Qualitäten der Kommunikationskategorie.
Servicekategorie	Die ins Service Profil eingetragenen Service Qualitäten der eingebettete Servicekategorie.

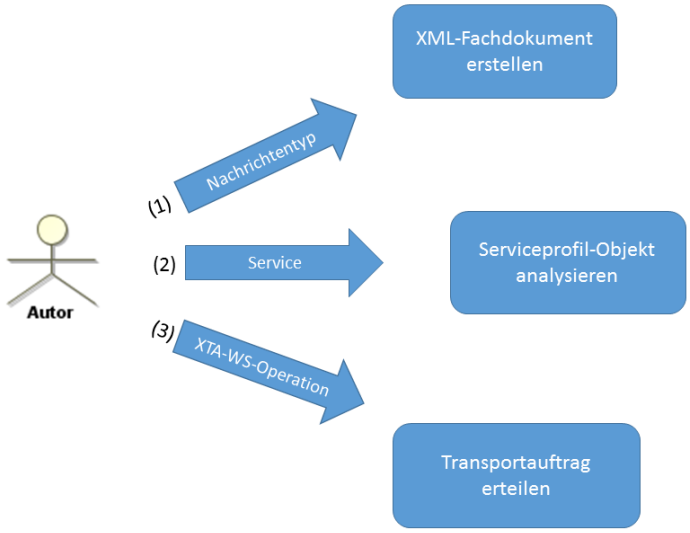
4.5 Anwendung eines Service Profils

Die Profilobjekte stellen Handlungs- und Konstruktionsvorschriften dar, die von den Knoten der Infrastruktur einzusetzen sind, um einen Transportauftraug zu erstellen, zu erteilen oder auszuführen.

Wie das laufen kann und welche Objekte an welchen Stellen zu Rate gezogen werden, soll mit den folgenden Darstellungen illustriert werden.

[3]Die Schutzbedarfsstufe "sehr hoch" ist nach BSI IT-Grundschutz prinzipiell vorgesehen. Sie wird aber in XTA 2 nicht angewendet, weil die Umsetzung von "sehr hoch" i.d.R. individuelle Maßnahmen bedingt und daher nicht für Standardisierung geeignet ist.

Abbildung 4.9. Anwendung eines Serviceprofil-Objekts durch den Autor



In Abbildung 4.9, „Anwendung eines Serviceprofil-Objekts durch den Autor“ sind die grundlegenen Schritte dargestellt, die der Autor bei der Erstellung und Erteilung eines Transportauftrags durchläuft, um die Informationen aus dem passenden Service Profil zu nutzen.

Autor: Bereitstellung und Erteilung des Transportauftrags

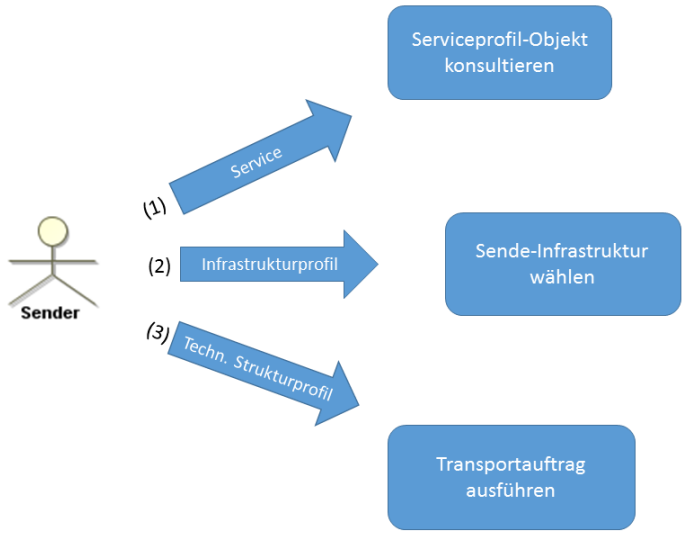
1. Der Autor erstellt die zu transportierende Fachnachricht gemäß den Vorschriften der Spezifikation des Fachstandards.
2. Dann bestimmt er den bei der empfangenden Behörde anzusprechenden Service. Welchem Service der Nachrichtentyp zugeordnet ist, lässt sich ebenfalls der Spezifikation des Fachstandards entnehmen. Aus dem Service und dem Typ der Fachnachricht ergibt sich das zu verwendende Service Profil (Serviceprofil-Objekt), welches durch das Fachverfahren des Autors auszuwerten ist, um die Regeln für die weiteren Schritte zu entnehmen.
3. Der Autor liest (nachdem das referenzierte Infrastrukturprofil ausgewertet ist und u. a. das passende Transportnachrichtenformat identifziert ist) aus dem im Service Profil referenzierten Technischen Strukturprofil die für ihn relevanten Anweisungen aus den verschiedenen Abschnitten aus.

   (a) Er ermittelt, welche fachlichen Informationen für den Aufbau der Transportnachricht benötigt werden und in Form welcher Container sie einzutragen sind.

   (b) Ebenso ermittelt er daraus – in Kombination mit den im Schutzprofil benannten Service Qualitäten –, welche Abschnitte des Transportauftrags und des Payloads ggf. zu signieren oder zu verschlüsseln sind. Für die kryptographischen Funktionen werden die im Kryptographieprofil enthaltenen Informationen zu Algorithmen verwendet.

   Die erstellte Transportnachricht wird an den Sender übermittelt.

Abbildung 4.10. Anwendung eines Serviceprofil-Objekts durch den Sender



Analog sieht es für den Sender aus wie dargestellt in Abbildung 4.10, „Anwendung eines Serviceprofil-Objekts durch den Sender“

Sender: Entgegennahme Transportauftrag und Absenden über die Infrastruktur

1. Der Sender nimmt den Transportauftrag und die Fachnachricht entgegen, identifiziert darin den angesprochenen Service und konsultiert das entsprechende Service Profil (Serviceprofil-Objekt).
2. Aus den im Service Profil angegebenen Infrastrukturprofilen wählt der Sender das passende aus (es wird von ihm technisch unterstützt, und es ist das für den Kontext, z. B. landesinterne Übermittlung, vorgesehene).
3. Der Sender interpretiert - wie das der Autor vorher getan hat - die für ihn relevanten Angaben aus dem im Service Profil referenzierten Technischen Strukturprofil:

   (a) An welchen Stellen muss der Autor welche Informationen in den Transportauftrag eingetragen haben? Wie ist der erwartete Aufbau der Transportnachricht?

   (b) In Kombination mit den Service Qualitäten: Was ist ggf. signiert, was verschlüsselt? Für die kryptographischen Funktionen werden die beiliegenden Algorithmen verwendet.

   Die fertige Transportnachricht wird unter Verwendung der Komponenten des gewählten Infrastrukturprofils übermittelt.

Empfänger: Durchführung Empfang

Der Empfänger liest dieselben Abschnitte und Informationen wie der Sender. Allerdings arbeitet er die Prozessschritte umgekehrt ab in dem Sinne, dass er im Falle einer Signatur eine Signaturprüfung durchführt, im Falle einer Verschlüsselung eine Entschlüsselung usw. Am Ende stehen ihm die vom Autor bereitgestellten Daten zur Verfügung, die dem Leser zur Verfügung gestellt werden.

Leser: Abruf und Darstellung der Nachrichten

Der Leser liest dieselben Abschnitte und Informationen wie der Autor. Die Prozessschritte erfolgen auch hier umgekehrt in dem Sinne, dass er im Falle einer Signatur eine Signaturprüfung durchführt, im Falle einer Verschlüsselung eine Entschlüsselung usw. Am Ende liegt ihm die vom Autor bereitgestellte Fachnachricht vor.

4.6 Struktur der Profile

In diesem Abschnitt werden Aufbauarchitektur und Semantik der Profile und ihrer Komponenten beschrieben. Die vier Profilarten werden behandelt: Service Profile sowie Schutz- und Infrastrukturprofile, auf die ein Service Profil Bezug nehmen muss. Außerdem die Technischen Strukturprofile, die die passende technische Konfiguration der Leistungserbringung enthalten.

4.6.1 Datentypen

Aus diesen Bausteinen sind die Profile zusammengesetzt.

4.6.1.1 Datentypen des Schutzprofils

Die Typen, die hier dargestellt werden, werden im Kontext einer XML-Instanz angewendet, die auf dem globalen Element schutzProfil basiert.

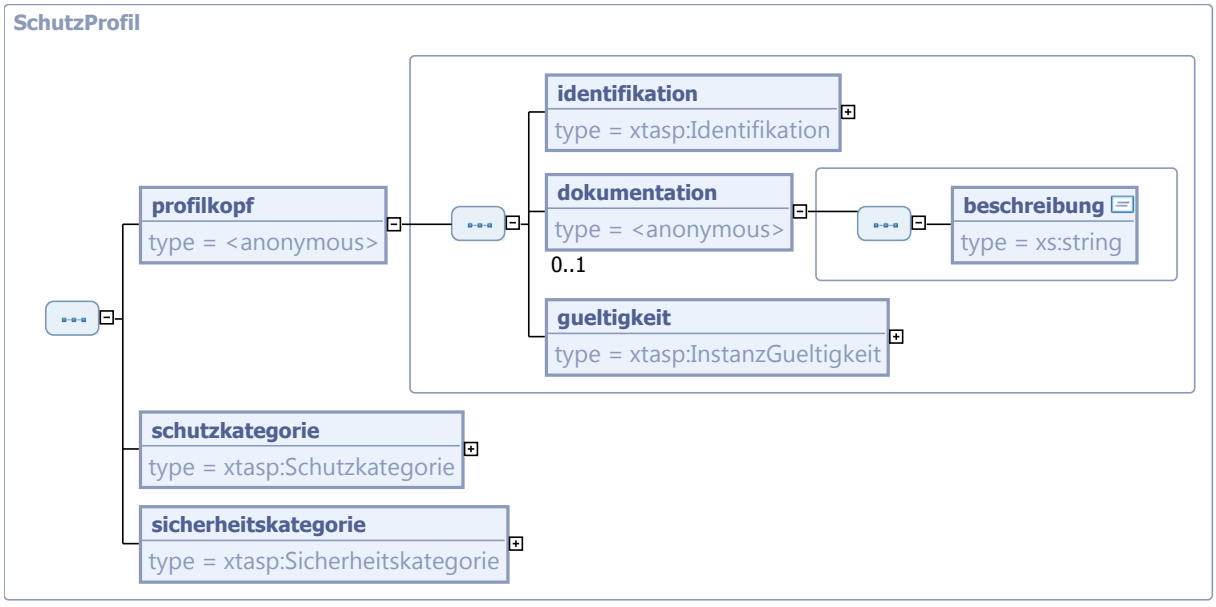
4.6.1.1.1 SchutzProfil

Typ: *SchutzProfil*

Das Schutzprofil deckt die Themen Datenschutz und Datensicherheit ab.

Es führt ausgewählte Parameter des Datenschutzes (Element *schutzkategorie*) und der Datensicherheit (Element *sicherheitskategorie*) zu einem Profil zusammen.

Abbildung 4.11. SchutzProfil



Kindelemente von *SchutzProfil*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
profilkopf		**1**		
Dieses Element wird gefüllt mit Informationen zu Identität und Gültigkeit der vorliegenden Profil-Instanz.				
identifikation	*xtasp:Identifikation*	**1**	4.6.2.2	101
Unterhalb dieses Elements werden die Parameter zu Identität und Herkunft der vorliegenden Profil-Instanz gefüllt.				

Kindelemente von *SchutzProfil*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
dokumentation		**0..1**		
In diesem Bereich kann Dokumentation zum Profil eingetragen werden.				
beschreibung	*xs:string*	**1**		
In dieses Element können Erläuterungen zum Profil eingetragen werden.				
gueltigkeit	*xtasp:InstanzGueltigkeit*	**1**	4.6.2.3	102
Unterhalb dieses Elements werden Parameter zur Gültigkeit der vorliegenden Profil-Instanz gefüllt.				
schutzkategorie	*xtasp:Schutzkategorie*	**1**	4.6.1.1.2	70
In den Bereich der Schutzkategorie sind die Schutzziele des Datenschutzes (siehe Abschnitt 2.1.3, „Begriffe zu Datenschutz und Datensicherheit“) einsortiert, soweit für die XTA Service Profile relevant. Sie sind hier gruppiert unter die Oberbegriffe der Transparenz und der Intervenierbarkeit.				
sicherheitskategorie	*xtasp:Sicherheitskategorie*	**1**	4.6.1.1.3	71
Unter dem Bereich der Sicherheitskategorie stehen die Service Qualitäten der Datensicherheit, soweit für die XTA Service Profile relevant. Sie sind hier zugeordnet den Oberbegriffen Vertraulichkeit und Integrität.				

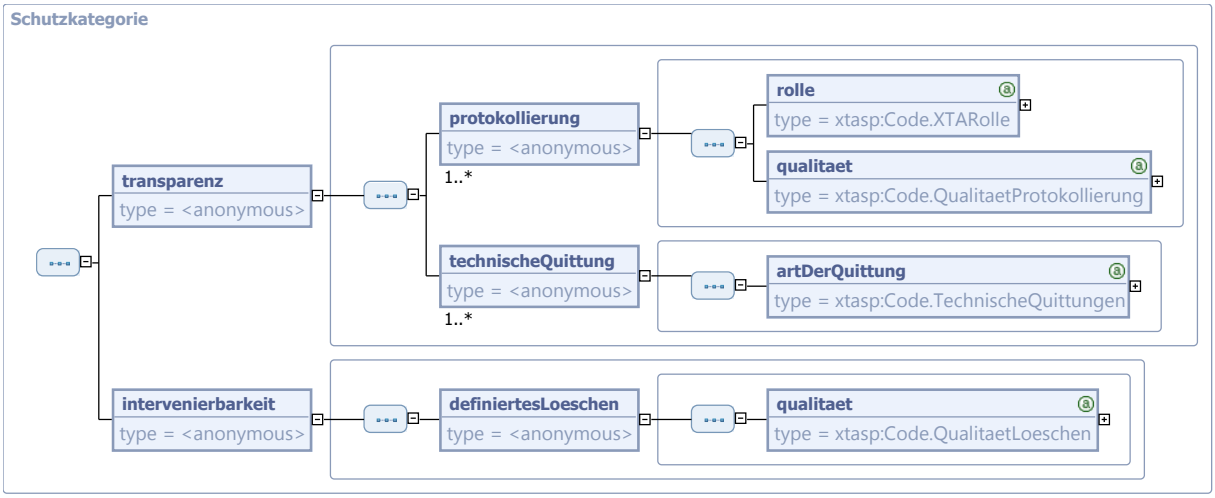
4.6.1.1.2 Schutzkategorie

Typ: *Schutzkategorie*

In den Bereich der Schutzkategorie sind die Schutzziele des Datenschutzes (siehe Abschnitt 2.1.3, „Begriffe zu Datenschutz und Datensicherheit“) einsortiert, soweit für die XTA Service Profile relevant. Sie sind hier gruppiert unter die Oberbegriffe der Transparenz und der Intervenierbarkeit.

Bei der Erstellung einer Schutzprofil-Instanz für die Service Qualitäten ist jeweils eine Ausprägung und / oder ein Geltungsbereich anzugeben. Bei den entsprechenden Unterelementen ist jeweils die benötigte Codeliste hinterlegt.

Abbildung 4.12. Schutzkategorie



Kindelemente von *Schutzkategorie*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
transparenz		**1**		
Bei Transparenz geht es um Nachvollziehbarkeit. Wenn die Prozesse der Nachrichtenübermittlung transparent ausgestaltet sind, bieten sie Aufsichtsbehörden, dem Auftraggeber oder sonstigen Betroffenen (beispielsweise in				

Kindelemente von *Schutzkategorie*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
den Rollen Autor, Sender, Empfänger, Leser) die Möglichket, Vorgehen und Ereignisse nachzuvollziehen bzw. zu überprüfen. Einsehbare Protokolle und zugestellte Quittungen sind das XTA-Angebot, Transparenz zu unterstützen.				
protokollierung		**1..n**		
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. Dabei bedeutet „Protokollierung durch Rolle Autor“ nicht notwendigerweise, dass dies durch das Fachverfahren geschieht. Diese Aufgabe kann an das Transportverfahren delegiert sein.				
rolle	*xtasp:Code.XTARolle*	**1**	4.6.1.3. 10.2	89
Hier ist einzutragen, in Bezug auf welche Rolle in der XTA-Infrastruktur hier Festlegungen zur Protokollierung eingetragen werden solll.				
qualitaet	*xtasp:Code.QualitaetProtokollierung*	**1**	4.6.1.1. 4.4	74
Hier ist einzutragen, ob durch den betreffenden Knoten ein Protokoll / ein Report zu führen ist und mit welchem Absicherungsniveau dies geschehen soll.				
technischeQuittung		**1..n**		
Unterhalb dieses Elements wird eingetragen, welche Quittungen in einem bestimmten Kontext gefordert sind. In einer XTA-Infrastruktur ist die Möglichkeit vorgesehen, dass Knoten der Infrastruktur durch Quittungen über entfernte Ereignisse der Abarbeitung eines Transportauftrags informiert werden. (vgl. Abschnitt 2.3 auf Seite 21)				
artDerQuittung	*xtasp:Code.TechnischeQuittungen*	**1**	4.6.1.1. 4.5	75
Hier ist eine der Quittungsarten einzutragen, welche in einer XTA-Infrastruktur vorgesehen sind (siehe Abschnitt 2.3, „Quittungen in XTA 2“). Pro Quittungsart, die im Schutzprofil vorgeschrieben werden soll, ist ein eigenes Element zu erstellen.				
intervenierbarkeit		**1**		
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.				
definiertesLoeschen		**1**		
Das Löschen der Daten und Protokolle erfolgt mit einer vorgegebenen Qualität, welche hier festgelegt wird. Der *Zeitpunkt* des Löschens wird an anderer Stelle (im Service Profil) geregelt.				
qualitaet	*xtasp:Code.QualitaetLoeschen*	**1**	4.6.1.1. 4.3	74
In diesem Element wird die geforderten Ausprägung der Service Qualität „Löschen von personen- oder organisationsbezogenen Daten“ hinterlegt. Aus dieser geht hervor, auf welche Weise diese Daten zum vorgegebenen Zeitpunkt zu löschen sind.				

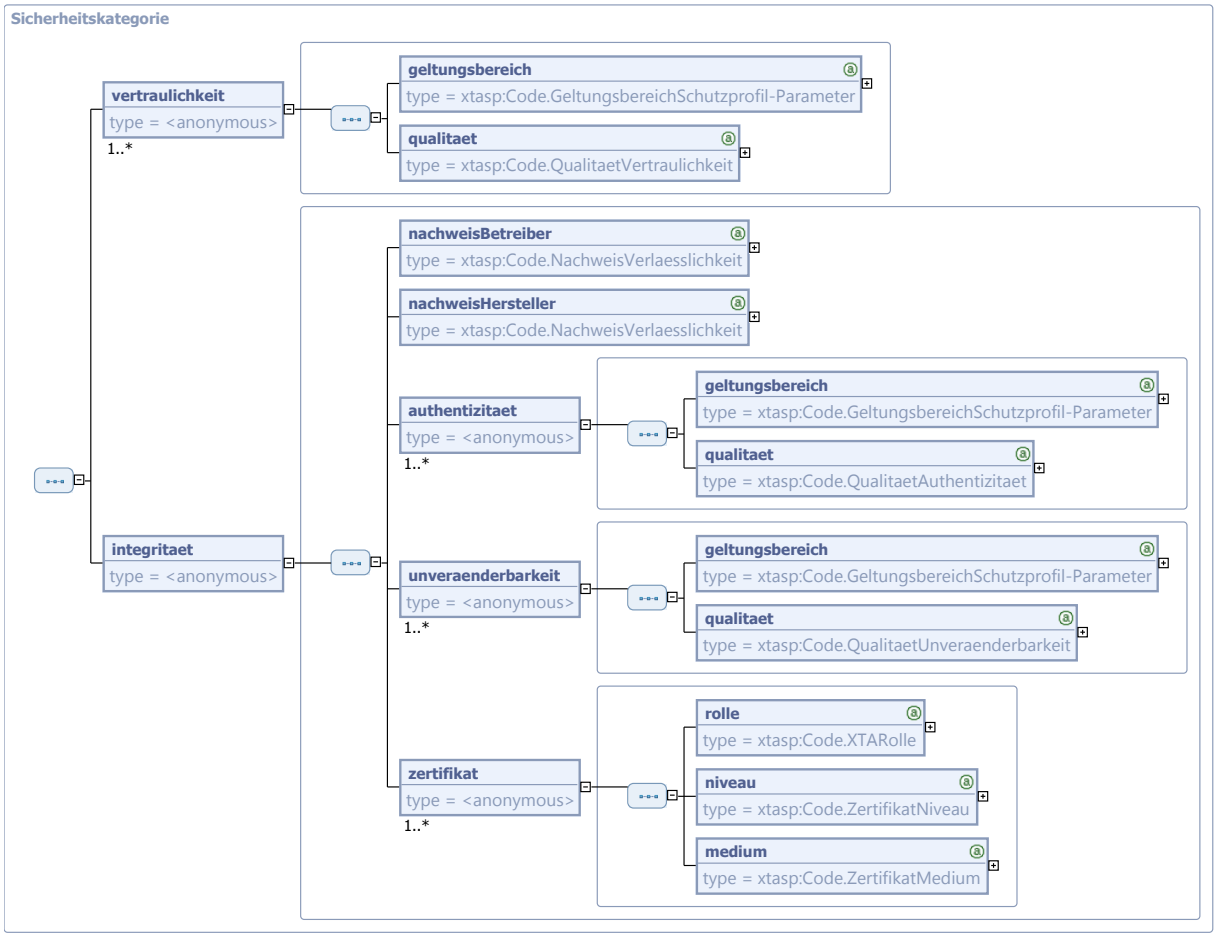
4.6.1.1.3 Sicherheitskategorie

Typ: *Sicherheitskategorie*

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

Ggf. 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.

Abbildung 4.13. Sicherheitskategorie



Kindelemente von *Sicherheitskategorie*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
vertraulichkeit		**1..n**		
Die Vertraulichkeit einer Nachricht oder von Daten ist die Eigenschaft, nur für einen abgegrenzten Empfängerkreis vorgesehen zu sein. Diese Service Qualität wird hier definiert bezogen auf einen bestimmten Geltungsbereich der Nachrichtenkommunikation (Element *geltungsbereich*). Außerdem wird die Anforderung formuliert bezogen auf ein gefordertes Niveau der Vertraulichkeit (Element qualitaet).				
geltungsbereich	*xtasp:Code.GeltungsbereichSchutzprofil-Parameter*	**1**	4.6.1.1.4.1	74
Hier ist einzutragen, für welche Teilstrecken der Nachrichtenkommunikation eine Vertraulichkeit gefordert werden soll.				
qualitaet	*xtasp:Code.QualitaetVertraulichkeit*	**1**	4.6.1.1.4.7	75
Hier ist einzutragen, welches Niveau von Vertraulichkeit auf der entsprechenden Strecke gefordert ist.				
integritaet		**1**		

Kindelemente von *Sicherheitskategorie*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
Unter Integrität werden die Unversehrtheit von Daten und das korrekte Funktionieren von Systemen verstanden.				
nachweisBetreiber	*xtasp:Code.NachweisVerlaesslichkeit*	**1**	4.6.1.1. 4.8	75
Diese Service Qualität besagt, welche Anforderungen an die durch einen Betreiber eines Fach- oder Transportverfahrens beigebrachten Nachweise gestellt werden, durch die er seiner Pflicht nachkommen kann, Verlässlichkeit hins. Betriebs- und Prozessintegrität zu belegen.				
nachweisHersteller	*xtasp:Code.NachweisVerlaesslichkeit*	**1**	4.6.1.1. 4.8	75
Diese Service Qualität besagt, welche Anforderungen an die durch einen Hersteller beigebrachten Nachweise gestellt werden, durch die er seiner Pflicht nachkommen kann, die Verlässlichkeit der von ihm vorgelegten Software hins. Betriebs- und Prozessintegrität zu belegen.				
authentizitaet		**1..n**		
Die Kommunikationsteilnehmer müssen sich der Identität des Kommunikationspartners vergewissern. In welchem Maß (Element qualitaet) und in welchem Kontext (Element *geltungsbereich*) das zu geschehen hat: dafür ist die Service Qualität der Authentizität zu definieren.				
geltungsbereich	*xtasp:Code. GeltungsbereichSchutzprofil-Parameter*	**1**	4.6.1.1. 4.1	74
Hier ist einzutragen, für welche Teilstrecken der Nachrichtenkommunikation die Authentizität abzusichern ist.				
qualitaet	*xtasp:Code.QualitaetAuthentizitaet*	**1**	4.6.1.1. 4.2	74
Hier ist einzutragen, welches Niveau von Authentizität auf der entsprechenden Strecke gefordert ist.				
unveraenderbarkeit		**1..n**		
Die Service Qualität der Unveränderbarkeit ist die Anforderung, dass die Daten im Zuge ihrer Übertragung durch die Messaging Infrastruktur nicht verändert werden können. In welchem Maß (Element *qualitaet*) und in welchem Kontext (Element *geltungsbereich*) diese Anforderung besteht wird in einer Instanz dieses Typs definiert.				
geltungsbereich	*xtasp:Code. GeltungsbereichSchutzprofil-Parameter*	**1**	4.6.1.1. 4.1	74
Hier ist einzutragen, für welche Teilstrecken der Nachrichtenkommunikation die Unveränderbarkeit abzusichern ist.				
qualitaet	*xtasp:Code.QualitaetUnveraenderbarkeit*	**1**	4.6.1.1. 4.6	75
Hier ist einzutragen, welches Niveau von Unveränderbarkeit auf der entsprechenden Strecke gefordert ist.				
zertifikat		**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 formulilert, sondern in der Serivcekategorie.				
rolle	*xtasp:Code.XTARolle*	**1**	4.6.1.3. 10.2	89
Hier ist der Beteiligte in einer XTA-Kommunikationsinfrastruktur zu nennen, in Bezug auf dessen Zertifikate hier Anforderungen definiert werden.				
niveau	*xtasp:Code.ZertifikatNiveau*	**1**	4.6.1.1. 4.10	76

Kindelemente von *Sicherheitskategorie*					
Kindelement		**Typ**	**Anz.**	**Ref.**	**Seite**
	Hier ist das geforderte Nivau der Zertifikate zu benennen.				
	medium	*xtasp:Code.ZertifikatMedium*	**1**	4.6.1.1.4.9	75
	Hier ist das geforderte Medium der Zertifikate zu benennen.				

4.6.1.1.4 Code-Datentypen des Schutzprofils

4.6.1.1.4.1 Code.GeltungsbereichSchutzprofil-Parameter

Code	**Code.GeltungsbereichSchutzprofil-Parameter**
Beschreibung	Diese Codeliste enthält die Schlüssel für die Kommunikationsstrecken, die in einer XTA-Infrastruktur betrachtet werden können bzw. die für die Schutzprofile von Interesse sind (z.B. die Strecke 'Autor-Sender' oder die Strecke 'Autor-Leser'). Sie werden verwendet, um den Geltungsbereich der Ausprägung einer Service Qualität der Nachrichtenkommunikation zu benennen. Beispielsweise kann in einem Schutzprofil die Service Qualität 'Vertraulichkeit hoch' für die Strecke (=den Geltungsbereich) 'Autor-Leser' gefordert werden.
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 166
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:geltungsbereich.schutzprofil-parameter
Codelisten-Version	1.0

4.6.1.1.4.2 Code.QualitaetAuthentizitaet

Code	**Code.QualitaetAuthentizitaet**
Beschreibung	Diese Codeliste enthält die Schlüssel für die zur Verfügung stehenden Niveaus, auf denen sich die Authentizität der Nachrichtenkommunikation auf einer entsprechenden Strecke abgesichert werden kann.
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 170
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:authentizitaetqualitaet
Codelisten-Version	1.0

4.6.1.1.4.3 Code.QualitaetLoeschen

Code	**Code.QualitaetLoeschen**
Beschreibung	Diese Codeliste enthält die Schlüssel für die Ausprägungen der Service Qualität „Löschen von personen- oder organisationsbezogenen Daten“. Sie enthält mögliche Arten der Löschung dieser Daten zum vorgegebenen Zeitpunkt.
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 172
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:loeschen.qualitaet
Codelisten-Version	1.0

4.6.1.1.4.4 Code.QualitaetProtokollierung

Code	**Code.QualitaetProtokollierung**
Beschreibung	Diese Codeliste enthält die Schlüssel für Ausprägungen der Protokollführung. Es wird festgelegt, ob ein Protokoll zu führen ist und unter welchem Absicherungsniveau dies ggf. zu geschehen hat.

Code	Code.QualitaetProtokollierung
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 173
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:protokollierung.qualitaet
Codelisten-Version	1.0

4.6.1.1.4.5 Code.TechnischeQuittungen

Code	Code.TechnischeQuittungen
Beschreibung	Diese Codeliste enthält die Schlüssel für die Arten technischer Quittungen, welche in einer XTA-Infrastruktur vorgesehen sind (siehe Abschnitt 2.3, „Quittungen in XTA 2“).
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 177
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:technische.quittungen
Codelisten-Version	1.0

4.6.1.1.4.6 Code.QualitaetUnveraenderbarkeit

Code	Code.QualitaetUnveraenderbarkeit
Beschreibung	Diese Codeliste enthält die Schlüssel für die verschiedenen Niveaus von abgesicherter Unveränderbarkeit, die auf entsprechenden Strecken gefordert sein können.
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 174
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:unveraenderbarkeit.qualitaet
Codelisten-Version	1.0

4.6.1.1.4.7 Code.QualitaetVertraulichkeit

Code	Code.QualitaetVertraulichkeit
Beschreibung	Diese Codeliste enthält die Schlüssel für die zur Verfügung stehenden Niveaus der Service Qualität der Vertraulichkeit der Nachrichtenkommunikation auf einer bestimmten Strecke der Messaging Infrastruktur.
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 176
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:vertraulichkeit.qualitaet
Codelisten-Version	1.0

4.6.1.1.4.8 Code.NachweisVerlaesslichkeit

Code	Code.NachweisVerlaesslichkeit
Beschreibung	Diese Codeliste beschreibt Arten, wie eine Organisation (Hersteller oder Betreiber einer Software) ihre Verlässlichkeit belegen kann. Kraft dieser Verlässlichkeit steht sie ein für die Unveränderbarkeit der durch die genannte Software unterstützten Prozesse (abgesehen von Kontexten, die der definierten Intervenierbarkeit dienen). Auch steht sie dafür ein, dass es ein effektives Changemanagement für die entsprechenden Prozesse gibt und diese Prozesse auch nicht umgangen werden können.
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 169
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:nachweis.verlaesslichkeit
Codelisten-Version	1.0

4.6.1.1.4.9 Code.ZertifikatMedium

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

Code	Code.ZertifikatMedium
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 184
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:zertifikat.medium
Codelisten-Version	1.0

4.6.1.1.4.10 Code.ZertifikatNiveau

Code	Code.ZertifikatNiveau
Beschreibung	Diese Codeliste enthält die Schlüssel für die definierten Niveaus eines Zertifikats.
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 185
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:zertifikat.niveau
Codelisten-Version	1.0

4.6.1.2 Datentypen des Infrastrukturprofils

Die Typen, die hier dargestellt werden, werden im Kontext einer XML-Instanz angewendet, die auf dem globalen Element infrastrukturProfil basiert.

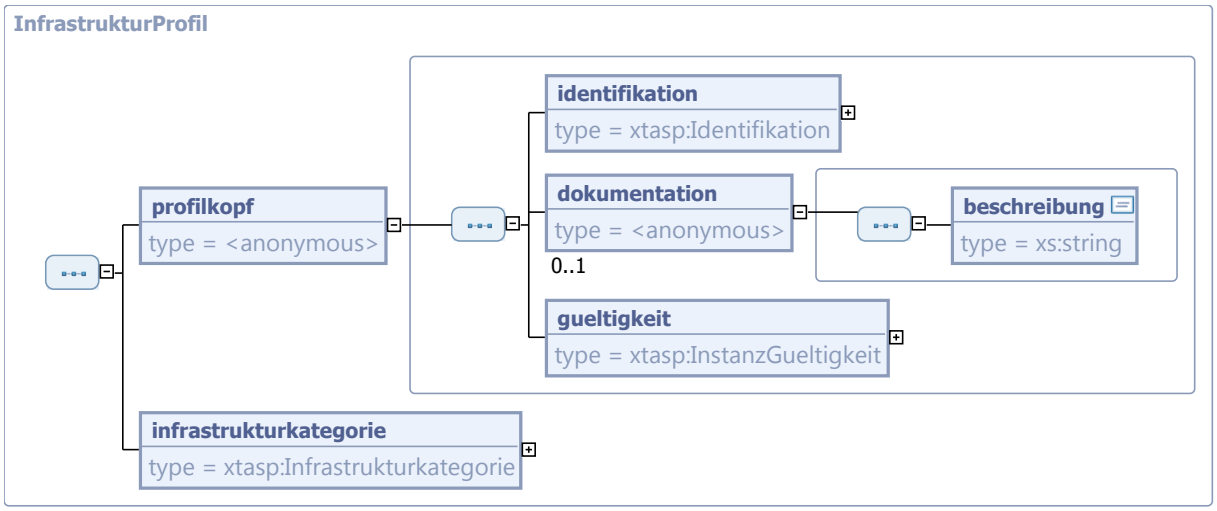
4.6.1.2.1 InfrastrukturProfil

Typ: *InfrastrukturProfil*

In eine Instanz dieses Typs sind Bezeichnungen von Infrastruktur-Komponenten einzutragen, die bei der Umsetzung eines Service Profils einzusetzen sind.

Die möglichen Ausprägungen der entsprechenden Service Qualitäten sind durch Codelisten hinterlegt.

Abbildung 4.14. InfrastrukturProfil



Kindelemente von *InfrastrukturProfil*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
profilkopf		**1**		
Dieses Element wird gefüllt mit Informationen zu Identität und Gültigkeit der vorliegenden Profil-Instanz.				

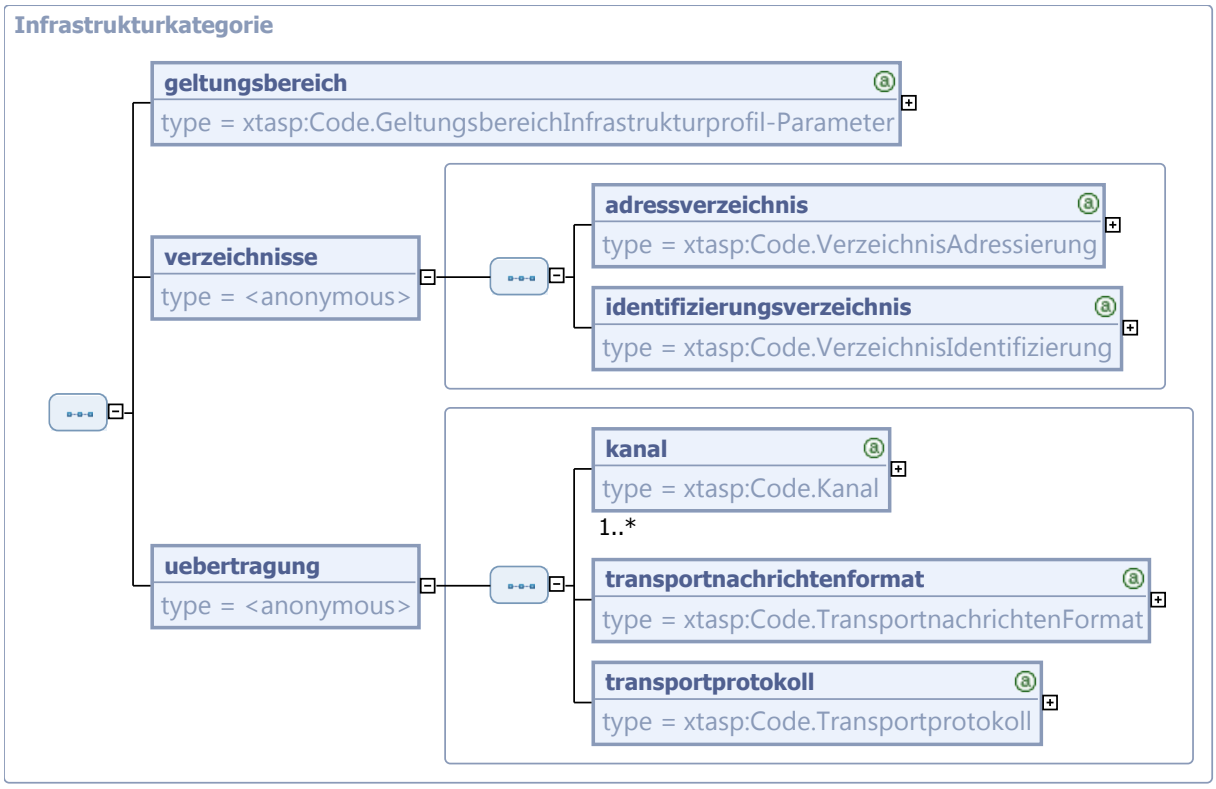
Kindelemente von *InfrastrukturProfil*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
identifikation	*xtasp:Identifikation*	**1**	4.6.2.2	101
Unterhalb dieses Elements werden die Parameter zu Identität und Herkunft der vorliegenden Profil-Instanz gefüllt.				
dokumentation		**0..1**		
In diesem Bereich kann Dokumentation zum Profil eingetragen werden.				
beschreibung	*xs:string*	**1**		
In dieses Element können Erläuterungen zum Profil eingetragen werden.				
gueltigkeit	*xtasp:InstanzGueltigkeit*	**1**	4.6.2.3	102
Unterhalb dieses Elements werden Parameter zur Gültigkeit der vorliegenden Profil-Instanz gefüllt.				
infrastrukturkategorie	*xtasp:Infrastrukturkategorie*	**1**	4.6.1.2.2	77
Unter diese Kategorie fallen alle Service Qualitäten, die dazu dienen, die Infrastrukturkomponenten für einen Transport festzulegen.				

4.6.1.2.2 Infrastrukturkategorie

Typ: *Infrastrukturkategorie*

Eine Instanz dieser Kategorie definiert die für einen Transport benötigten bzw. einzusetzenden Infrastrukturkomponenten.

Abbildung 4.15. Infrastrukturkategorie



Kindelemente von *Infrastrukturkategorie*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
geltungsbereich	*xtasp:Code. GeltungsbereichInfrastrukturprofil-Parameter*	**1**	4.6.1.2.3.1	78
Hier wird der Geltungsbereich für das vorliegende Infrastrukturprofil genannt.				
verzeichnisse		**1**		
Unterhalb dieses Elements sind die Verzeichnisdienste genannt, die zum vorliegenden Infrastrukturprofil gehören.				
adressverzeichnis	*xtasp:Code.VerzeichnisAdressierung*	**1**	4.6.1.2.3.2	78
Hier wird die Bezeichnung des Verzeichnisdienstes eingetragen, der für jeden Teilnehmer des Nachrichtenaustauschs die Parameter für die technische Adressierung von Teilnehmern bereitstellt.				
identifizierungsverzeichnis	*xtasp:Code.VerzeichnisIdentifizierung*	**1**	4.6.1.2.3.3	79
Hier ist der Verzeichnisdienst genannt, der zu allen Teilnehmern die Parameter und Entitäten für Identität und Identitätsnachweis bereitstellt.				
uebertragung		**1**		
Unterhalb dieses Elements stehen die Infrastrukturkomponenten zum Thema Übertragungstechnologien.				
kanal	*xtasp:Code.Kanal*	**1..n**	4.6.1.2.3.4	79
Hier wird für die Kommunikation von Sender und Empfänger der Kanal eingetragen, über den der Nachrichtenaustausch ausgeführt wird. Unter Kanal wird die Art der technischen Verbindung verstanden oder das Netz, über das kommuniziert wird.				
transportnachrichtenformat	*xtasp:Code.TransportnachrichtenFormat*	**1**	4.6.1.2.3.5	79
In diesem Element wird das Nachrichtenformat eingetragen, in dem die Daten zwischen Sender und Empfänger zu übertragen sind.				
transportprotokoll	*xtasp:Code.Transportprotokoll*	**1**	4.6.1.2.3.6	79
Dieses Element legt das Protokoll fest, das für die Kommunikaiton der Nachrichten zwischen Sender und Empfänger zu verwenden ist.				

4.6.1.2.3 Code-Datentypen des Infrastrukturprofils

4.6.1.2.3.1 Code.GeltungsbereichInfrastrukturprofil-Parameter

Code	**Code.GeltungsbereichInfrastrukturprofil-Parameter**
Beschreibung	Diese Codeliste enthält die Schlüssel für den Geltungsbereich eines Infrastrukturprofils. Jeder Eintrag der Codeliste nennt einen (fachneutral bezeichneten) Ausschnitt des Nachrichtenaustauschs der öffentlichen Verwaltung.
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 165
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:geltungsbereich.infrastrukturprofil-parameter
Codelisten-Version	1.0

4.6.1.2.3.2 Code.VerzeichnisAdressierung

Code	**Code.VerzeichnisAdressierung**
Beschreibung	Diese Codeliste enthält die Schlüssel für Verzeichnislösungen zur Bereitstellung von Parametern für die technische Adressierung von Teilnehmern.

Code	Code.VerzeichnisAdressierung
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 180
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:verzeichnis.adressierung
Codelisten-Version	1.0

4.6.1.2.3.3 Code.VerzeichnisIdentifizierung

Code	Code.VerzeichnisIdentifizierung
Beschreibung	Diese Codeliste enthält die Schlüssel für Verzeichnislösungen zur Verwaltung von elektronischen Identitäten (Bezeichnungen und kryptographische Token für Identität und Identitätsnachweis).
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 181
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:verzeichnis.identifzierung
Codelisten-Version	1.0

4.6.1.2.3.4 Code.Kanal

Code	Code.Kanal
Beschreibung	Diese Codeliste enthält die Schlüssel für die Beschreibung des Kanals der Kommunikation von Sender und Empfänger, d. h. die Art der Verbindung oder das Netzsegment, über das sie kommunizieren.
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 167
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:kanal
Codelisten-Version	1.0

4.6.1.2.3.5 Code.TransportnachrichtenFormat

Code	Code.TransportnachrichtenFormat
Beschreibung	Diese Codeliste nennt verfügbare Nachrichtenformate für Transportnachrichten (normalerweise basierend auf dem XML-Format SOAP).
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 178
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:transportnachrichten.format
Codelisten-Version	1.0

4.6.1.2.3.6 Code.Transportprotokoll

Code	Code.Transportprotokoll
Beschreibung	Diese Codeliste nennt verfügbare Protokolle, dier die Kommunikaiton von Daten zwischen Partnern festlegen.
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 179
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:transportprotokoll
Codelisten-Version	1.0

4.6.1.3 Datentypen des Technischen Strukturprofils

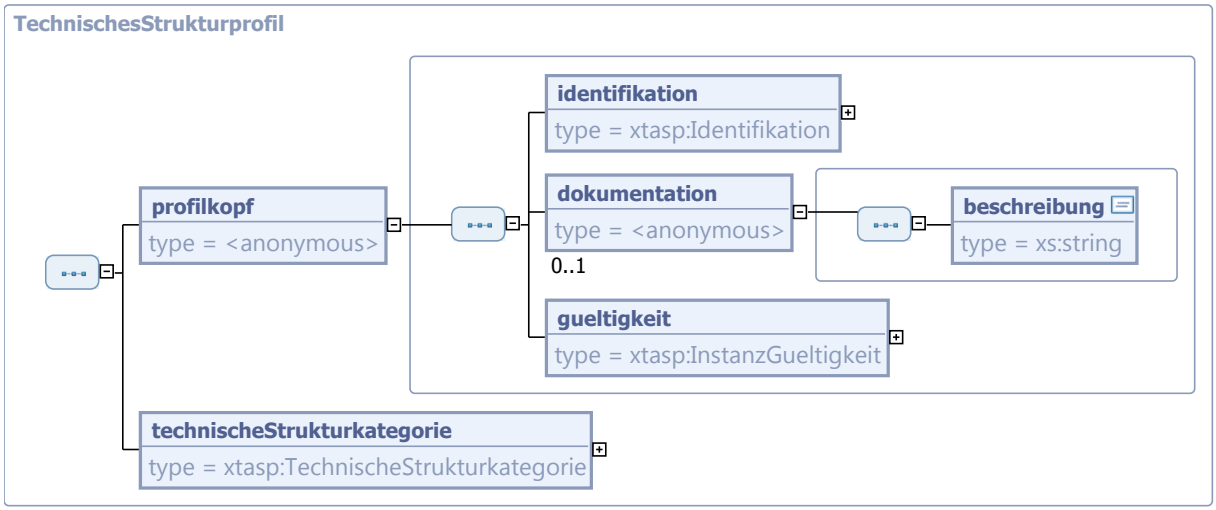
Die Typen, die hier dargestellt werden, werden im Kontext einer XML-Instanz angewendet, die auf dem globalen Element technischesStrukturprofil basiert.

4.6.1.3.1 TechnischesStrukturprofil

Typ: *TechnischesStrukturprofil*

In einer Instanz dieses Typs wird – in Form von Regeln – zu den Ausprägungen der Service Qualitäten die technische Konfiguration der vorgesehenen Implementierung genannt.

Abbildung 4.16. TechnischesStrukturprofil



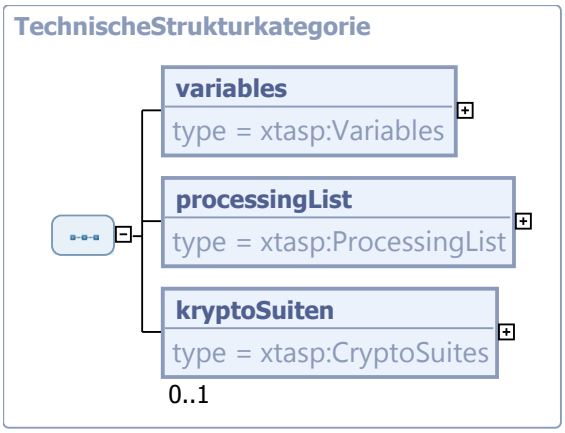
Kindelemente von *TechnischesStrukturprofil*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
profilkopf		**1**		
Dieses Element wird gefüllt mit Informationen zu Identität und Gültigkeit der vorliegenden Profil-Instanz.				
identifikation	*xtasp:Identifikation*	**1**	4.6.2.2	101
Unterhalb dieses Elements werden die Parameter zu Identität und Herkunft der vorliegenden Profil-Instanz gefüllt.				
dokumentation		**0..1**		
In diesem Bereich kann Dokumentation zum Profil eingetragen werden.				
beschreibung	*xs:string*	**1**		
In dieses Element können Erläuterungen zum Profil eingetragen werden.				
gueltigkeit	*xtasp:InstanzGueltigkeit*	**1**	4.6.2.3	102
Unterhalb dieses Elements werden Parameter zur Gültigkeit der vorliegenden Profil-Instanz gefüllt.				
technischeStrukturkategorie	*xtasp:TechnischeStrukturkategorie*	**1**	4.6.1.3.2	80
In diesem Objekt ist der Inhalt, also die Regeln des Technischen Strukturprofils, dargestellt.				

4.6.1.3.2 TechnischeStrukturkategorie

Typ: *TechnischeStrukturkategorie*

Eine Instanz dieses Typs enthält ein Set von Regeln. Dies sind die Regeln, die das Technische Strukturprofil für die technische Umsetzung der Service Qualitäten vorgibt.

Abbildung 4.17. TechnischeStrukturkategorie



Kindelemente von *TechnischeStrukturkategorie*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
variables	*xtasp:Variables*	**1**	4.6.1.3.3	81
In diesem Bereich werden die von der Fachlichkeit für die Regeln des Technischen Strukturprofils anzugebenden Informationen bzw. Informationsblöcke benannt und mit einem Namen und einem Typ versehen. Dies geschieht, damit in den zu definierenden Regeln (Bereich *processingList*) auf diese Informationen Bezug genommen werden kann. Ein solcher Abschnitt mit Variablendeklarationen muss immer vorhanden sein. Er besteht aus einer Liste einzelner Deklarationen.				
processingList	*xtasp:ProcessingList*	**1**	4.6.1.3.7	85
Hier sind die Regeln (Konstruktionsvorschriften) angegeben, welche das Technische Strukturprofil vorgibt. Es wird - unter Verwendung der im Bereich *variables* deklarierten Bezeichner - spezifiziert, was jede XTA-Rolle im Detail informationstechnisch zu tun hat: Dies betrifft die Art der kryptographischen Verarbeitung einerseits und andererseits den Containeraufbau für die Objekte im Transportauftrag.				
kryptoSuiten	*xtasp:CryptoSuites*	**0..1**	4.6.1.4.2	90
Hier sind bei Bedarf die Krypto-Zuordnungen einzutragen (Qualität und Krypto-Suiten), die dieses technische Strukturprofil direkt vorschreibt, um dadurch die Vorgaben des standardmäßigen Kryptographieprofils zu überschreiben. Eine Krypto-Suite enthält Algorithmen und Schlüssellängen für einschlägige kryptographische Funktionen.				

4.6.1.3.3 Variables

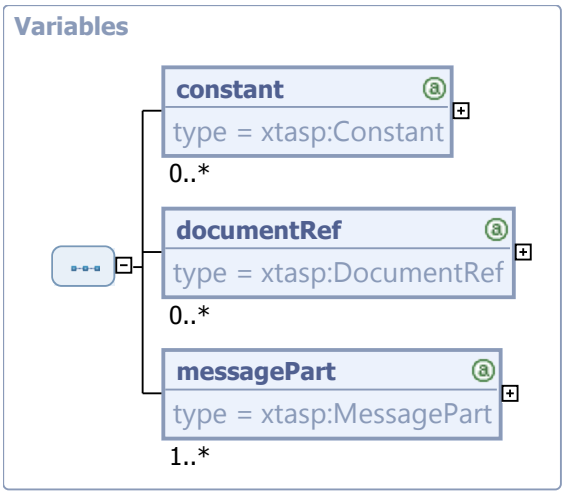
Typ: *Variables*

Eine Instanz dieses Typs deklariert die von der Fachlichkeit für die Regeln des Technischen Strukturprofils anzugebenden Objekte und versieht sie mit einem Namen und einem Typ.

Es werden drei Arten von Variablen unterschieden, die den drei unterschiedlilchen Kindelementen entsprechen.

Jede gelistete Variable hat einen Bezeichner, z. B. „contentPackage“. In den Konstruktionsvorschriften wird der Wert der Variablen verwendet, indem der Bezeichner der Variablen mit vorangestelltem Zeichen „$“ geschrieben wird („$contentPackage“).

Abbildung 4.18. Variables



Kindelemente von *Variables*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
constant	*xtasp:Constant*	**0..n**	4.6.1.3.4	82
Hier ist die Deklaration einer Konstanten einzutragen. Sie besteht aus dem Namen der Konstanten und ihrem Wert, der im Anschluss nicht mehr geändert werden darf.				
documentRef	*xtasp:DocumentRef*	**0..n**	4.6.1.3.5	83
Pro Element wird eine Variable definiert, die dafür vorgesehen ist, ein XML-Dokument zu referenzieren.				
messagePart	*xtasp:MessagePart*	**1..n**	4.6.1.3.6	84
Jedes Element deklariert einen Container oder ein Nachrichtenfragment, das für den Transport zu verwenden ist. Hierzu gehören bei Verwendung der XTA-WS-Schnittstelle stets die Komponenten der XTA-Nachrichten, es zählen dazu aber auch z. B. die OSCI 1.2 - Nachrichten. Auf der Basis einer solchen Deklaration werden die Elemente des Containers oder Nachrichtenfragments referenzierbar. Der Bezeichner wird im Anschluss verwendet, um mittels XPath-Ausdrücken zu beschreiben, wie die Komponenten zu aggregieren bzw. kryptographisch zu behandeln sind.				

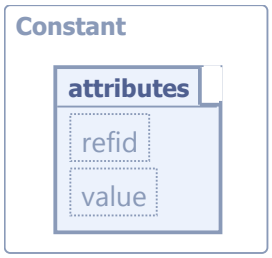
4.6.1.3.4 Constant

Typ: *Constant*

In einer Instanz dieses Typs ist die Deklaration einer Konstanten enthalten. Sie besteht aus dem Namen der Konstanten und ihrem Wert. Im folgenden Beispiel wird eine Konstante deklariert mit dem Bezeichner „containerName“ und dem Wert „XMELD_DATA“:

```
<constant refid="containerName" value="XMELD_DATA"/>
```

Abbildung 4.19. Constant



Kindelemente von *Constant*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
refid	*xs:string*	**1**		
Hier wird der festgelegte Bezeichner der Konstanten eingetragen, unter dem diese später referenziert werden kann.				
value	*xs:string*	**1**		
Hier ist der Wert der Konstanten eingetragen.				

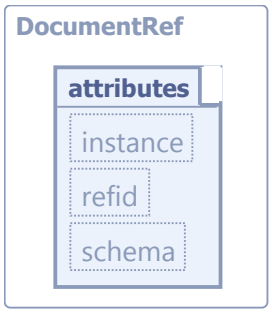
4.6.1.3.5 DocumentRef

Typ: *DocumentRef*

Eine Instanz dieses Typs ist eine Variable, die dafür vorgesehen ist, ein XML-Dokument (XML-Instanz) zu referenzieren. Diesem wird ein Bezeichner zugewiesen, zusätzlich wird ein Typ in Form eines XML Schemas angegeben. Im folgenden Beispiel wird als XML-Dokument (XML-Instanz) das Service Profil referenziert, welches bei einem Transport berücksichtigt werden soll.

```
<documentRef refid="ServiceProfil" instance="ServiceProfil"
 schema="ServiceProfil.xsd"/>
```

Abbildung 4.20. DocumentRef



Kindelemente von *DocumentRef*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
instance	*xs:string*	**1**		
Mit diesem Attribut wird auf eines der globalen Elemente aus der im Attribut *schema* bezeichneten XSD zu referenziert. Der Elementname ist einzutragen.				
refid	*xs:string*	**1**		
Hier wird der Bezeichner eingetragen, der das XML-Dokument (XML-Instanz) identifiziert.				

Kindelemente von *DocumentRef*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
schema	*xs:string*	**0..1**		
Hier wird der Dateiname des XML-Schema eingetragen, der dem XML-Dokument (XML-Instanz) als Typ zugeordnet ist.				

4.6.1.3.6 MessagePart

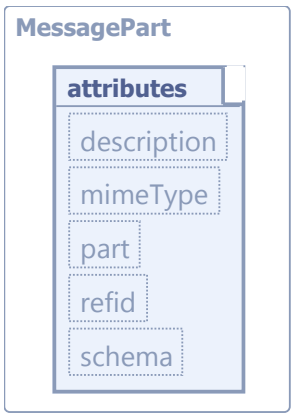
Typ: *MessagePart*

Eine Instanz dieses Typs definiert einen der Container oder Nachrichtenfragmente, die für den Transport verwendet werden sollen. Dies kann ein XML-Dokument sein, aber auch ein Anhang in einem anderen technischen Format.

Das folgende Beispiel definiert den Metadatencontainer, wie er in XTA Nachrichten verwendet wird, unter dem Namen „messagemetadata“:

```
<messagePart refid="messagemetadata"
 schema="OSCI_MessageMetaData_V2.02.xsd" part="MessageMetadata"/>
```

Abbildung 4.21. MessagePart



Kindelemente von *MessagePart*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
description	*xs:string*	**0..1**		
Dieses Attribut kann verwendet werden, um eine Erläuterung zu dem zugefügten Artefakt einzutragen.				
mimeType	*xs:string*	**0..1**		
Hier kann der Mimetype des referenzierten Artefakts eingetragen werden.				
part	*xs:string*	**0..1**		
Hier wird ein bestimmtes globales Element aus dem referenzierten Schema genannt.				
refid	*xs:string*	**1**		
Hier wird ein Bezeichner für den Container oder das Nachrichtenfragment eingetragen, so dass später referenziert werden kannn.				
schema	*xs:string*	**0..1**		
Hier kann ein dem XML-Dokument zugeordnetes XML Schema genannt werden.				

4.6.1.3.7 ProcessingList

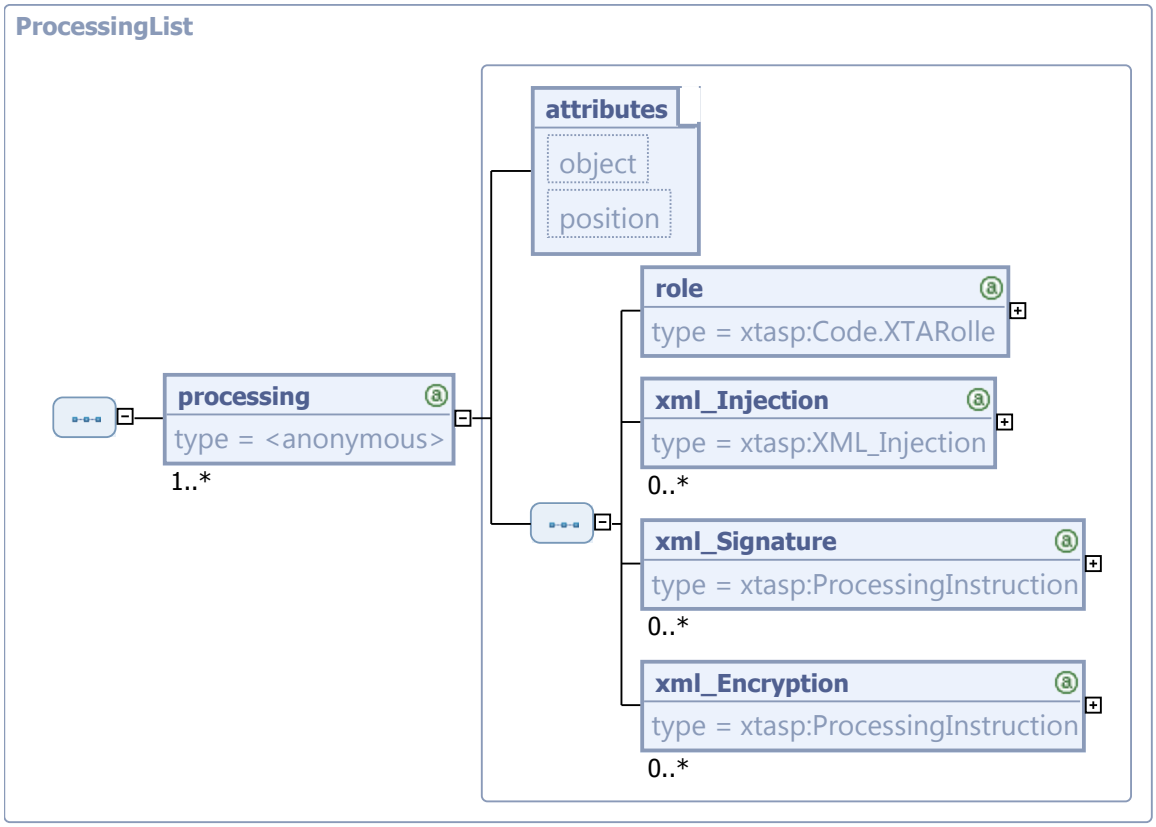
Typ: *ProcessingList*

Eine Instanz dieses Typs enthält die Konstruktionsvorschriften, welche das Technische Strukturprofil vorgibt. Es wird spezifiziert, was jede XTA-Rolle im Detail informationstechnisch zu tun hat.

Diese Konstruktionsvorschriften beschreiben für jede Rolle, wo welche Daten zu hinterlegen sind und welche Daten, abhängig von den Service Qualitäten des Service Profils, auf welche Weise kryptographisch zu behandeln sind bzw. behandelt wurden.

Die Reihenfolge der Abarbeitung wird auf zwei Ebenen über Positionsattribute vorgegeben.

Abbildung 4.22. ProcessingList



Kindelement von *ProcessingList*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
processing		**1..n**		
Pro Element ist eine Menge von Anweisungen bzw. Konstruktionsvorschriften für eine bestimmte XTA-Rolle enthalten.				
object	*xs:string*	**1**		
Hier ist das Nachrichtenelement genannt, auf das sich die aufgeführte Sequenz von Anweisungen bzw. Konstruktionsvorschriften bezieht. Die ausführende Rolle muss die Instanz des Nachrichtenelements erzeugen, wenn diese noch nicht existiert. Es kann eine bereits definierte Variable angegeben werden. Der Wert der Variablen wird benannt, indem ein Zeichen "$" dem Variablennamen vorangestellt wird, z. B. bezeichnet "$genericContentContainer" den aktuellen Wert der Variablen genericContentContainer.				

Kindelement von *ProcessingList*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
position	*xs:positiveInteger*	**1**		
Die Instanzen des Elements müssen sequentiell abgearbeitet werden gemäß aufsteigender Sortierung über dieses Attribut. Von der Menge der Elemente unterhalb der processingList wird gefordert, dass jede Ausprägung dieses Attributs eindeutig ist. Darüber hinausgehend wird nichts gefordert, z. B. muss die Sequenz der Belegungen nicht lückenlos sein.				
role	*xtasp:Code.XTARolle*	**1**	4.6.1.3.10.2	89
Für diese XTA-Rolle gelten die aufgeführten Anweisungen.				
xml_Injection	*xtasp:XML_Injection*	**0..n**	4.6.1.3.8	86
Eine Anweisung dieser Art ist eine Konstruktionsvorschrift. Sie fügt in das referenzierte Nachrichtenfragment an einer durch einen XPath Ausdruck referenzierten Stelle einen Wert ein. Besteht der referenzierte Knoten nicht, ist er zu erzeugen.				
xml_Signature	*xtasp:ProcessingInstruction*	**0..n**	4.6.1.3.9	87

Diese Anweisung erzeugt die Signatur über einem Nachrichtenabschnitt. Das Anbringen einer Signatur dient oftmals zur Umsetzung einer Service Qualität, die in diesem Fall mitgegeben wird.
Das folgende Beispiel signiert in einem XML-Dokument den Teil, auf den der in der Variablen „$xhdContent“ gespeicherte XPath-Ausdruck verweist. Diese Signatur erfolgt dann und nur dann, wenn in dem Serviceprofil die Service Qualität „Unveränderbarkeit“ mit der Qualität „hoch“ gefordert wird.
`<xml_Signature serviceQuality="Unveränderbarkeit">`
`  <path>$xhdContent</path>`
`  <cryptoQuality="hoch"/>`
`<XML_Signature>`
Die anzuwendende Krypto-Suite wird aus der Liste der gültigen Krypto-Suiten der angegebenen Stärke (hier „hoch“) angewählt.				

xml_Encryption	*xtasp:ProcessingInstruction*	**0..n**	4.6.1.3.9	87

Eine Anweisung dieser Art führt die Verschlüsselung eines bestimmten Elements in einer Nachricht oder einem Nachrichtenelement aus. Eine solche Verschlüsselung dient oft zur Umsetzung einer Service Qualität, die dann entsprechend mitzugeben ist.
Das folgende Beispiel verschlüsselt in einem XML-Dokument den Teil, auf den der in der Variablen „$xhdContent/Nutzdaten“ gespeicherte XPath-Ausdruck verweist. Diese Verschlüsselung erfolgt dann und nur dann, wenn in dem Service-Profil die Service Qualität „Vertraulichkeit“ mit der Qualität „hoch“ gefordert wird.
`<xml_Encryption serviceQuality="Vertraulichkeit">`
`  <path>$xhDContent/Nutzdaten</path>`
`  <cryptoQuality="hoch"/>`
`<xml_Encryption>`
Die anzuwendende Krypto-Suite wird aus der Liste der gültigen Krypto-Suiten der angegebenen Stärke (hier „hoch“) ausgewählt.				

4.6.1.3.8 XML_Injection

Typ: *XML_Injection*

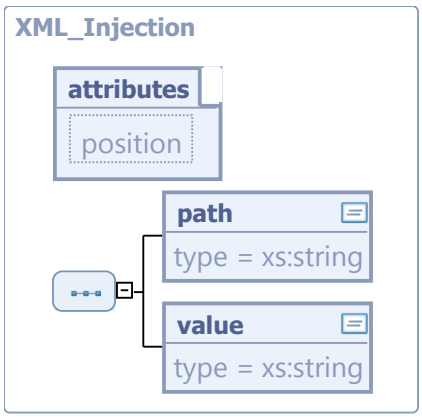
Eine Instanz dieses Typs ist eine Anweisung (hier eine Konstruktionsvorschrift). Sie fügt in das referenzierte Nachrichtenfragment (Attribut *object*) an einer durch einen XPath Ausdruck referenzierten Stel-

le (Element *path*) einen Wert (Element *value*) ein. Besteht das referenzierte Element nicht, ist es zu erzeugen.

Das folgende Beispiel erzeugt in einem *GenericContentContainer* einer XTA Nachricht eine Message mit der Referenz „xhdContent“ und speichert darin das in der Variable „$xhdContent“ gespeicherte XML.

```
<xml_Injection>
      <path>ContentContainer/Message[id="xhdContent"]</path>
      <value>$xhdContent></value>
<xml_Injection>
```

Abbildung 4.23. XML_Injection



Kindelemente von *XML_Injection*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
position	*xs:positiveInteger*	**1**		
Die Elemente mit Attribut position unterhalb eines Elements processing müssen sequentiell abgearbeitet werden gemäß aufsteigender Sortierung über dieses Attribut. Von der Menge der Instanzen unterhalb eines processing-Elementes wird gefordert, dass jede Ausprägung dieses Attributs eindeutig ist. Darüber hinausgehend wird nichts gefordert, z. B. muss die Sequenz der Belegungen nicht lückenlos sein.				
path	*xs:string*	**1**		
Hier wird eine bestimmte Stelle in dem referenzierten XML-Objekt referenziert. Es ist ein XPath-Ausdruck einzutragen.				
value	*xs:string*	**1**		
Hier steht der Wert, der an der referenzierten Stelle / in das referenzierte Objekt einzutragen ist.				

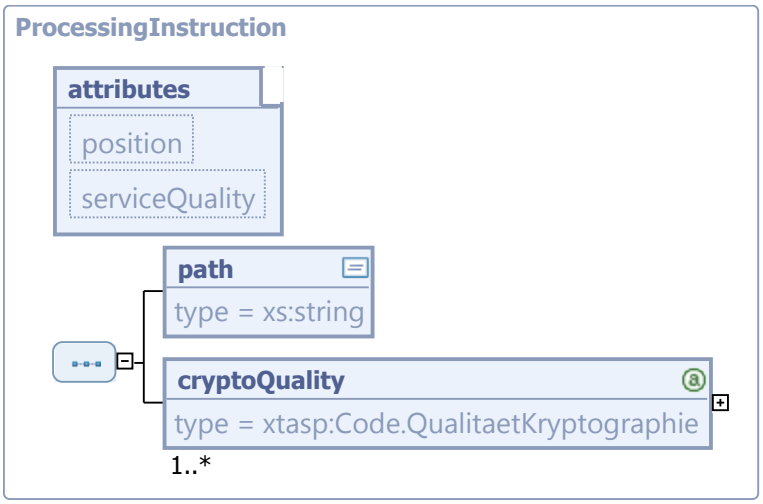
4.6.1.3.9 ProcessingInstruction

Typ: *ProcessingInstruction*

Eine Instanz dieses Typs ist eine Anweisung, die eine kryptographische Operation (Signierung oder Verschlüsselung) über einem bestimmten Element in einer Nachricht oder einem Nachrichtenelement nennt. Diese Operation dient zur Umsetzung einer Service Qualität, die entsprechend einzutragen ist.

Die anzuwendende Krypto-Suite wird aus der Liste der gültigen Krypto-Suiten der angegebenen Stärke ausgewählt.

Abbildung 4.24. ProcessingInstruction



Kindelemente von *ProcessingInstruction*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
position	*xs:positiveInteger*	**1**		
Die Elemente mit Attribut position unterhalb eines Elements processing müssen sequentiell abgearbeitet werden gemäß aufsteigender Sortierung über dieses Attribut. Von der Menge der Instanzen unterhalb eines processing-Elementes wird gefordert, dass jede Ausprägung dieses Attributs eindeutig ist. Darüber hinausgehend wird nichts gefordert, z. B. muss die Sequenz der Belegungen nicht lückenlos sein.				
serviceQuality	*xs:string*	**0..1**		
Hier wird die Service Qualität benannt, deren Umsetzung die Anweisung betreibt.				
path	*xs:string*	**1**		
Hier wird eine bestimmte Stelle in dem referenzierten XML-Objekt referenziert. Es ist ein XPath-Ausdruck einzutragen.				
cryptoQuality	*xtasp:Code.QualitaetKryptographie*	**1..n**	4.6.1.3.10.1	88
Hier wird die kryptograpshische Qualität (z. B. hoch, normal, niedrig) benannt, auf deren Umsetzung sich die Anweisung bezieht (in Bezug auf die Service Qualität).				

4.6.1.3.10 Code-Datentypen des Technischen Strukturprofils

4.6.1.3.10.1 Code.QualitaetKryptographie

Code	Code.QualitaetKryptographie
Beschreibung	Diese Codeliste enthält die Schlüssel für die Abstufungen 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 .
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 171

Code	Code.QualitaetKryptographie
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:kryptographie.qualitaet
Codelisten-Version	1.0

4.6.1.3.10.2 Code.XTARolle

Code	Code.XTARolle
Beschreibung	Diese Codeliste benennt die Rollen, die in einer XTA-Kommunikationsinfrastruktur am Prozess der Nachrichtenübermittlung beteiligt sind. Ein Knoten Relay ist nicht in den Einträgen der Codeliste aufgeführt: Ein Relay ist keine eigene Rolle, sondern entweder der XTA-Rolle Sender oder der XTA-Rolle Empfänger zugeordnet.
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 182
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:xta-rolle
Codelisten-Version	1.0

4.6.1.4 Datentypen des Kryptographieprofils

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

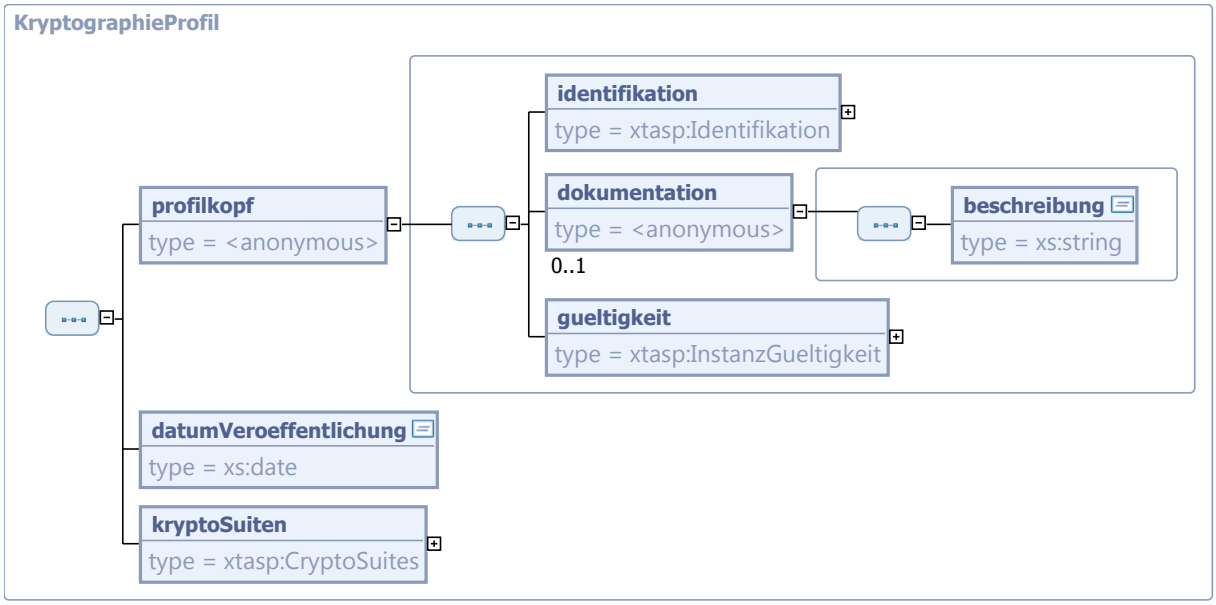
4.6.1.4.1 KryptographieProfil

Typ: *KryptographieProfil*

Eine Instanz dieses Typs bildet für die benötigten Formate und Schutzniveaus einer kryptographisch abzusichernden Kommunikation die Algorithmen und sonstigen kryptographischen Parameter ab, die bei der Umsetzung der Anforderungen aus Infrastruktur- und Schutzprofil einzusetzen sind.

Im Zusammenhang wird dies näher erläutert in Abschnitt 4.4.4 auf Seite 63.

Abbildung 4.25. KryptographieProfil



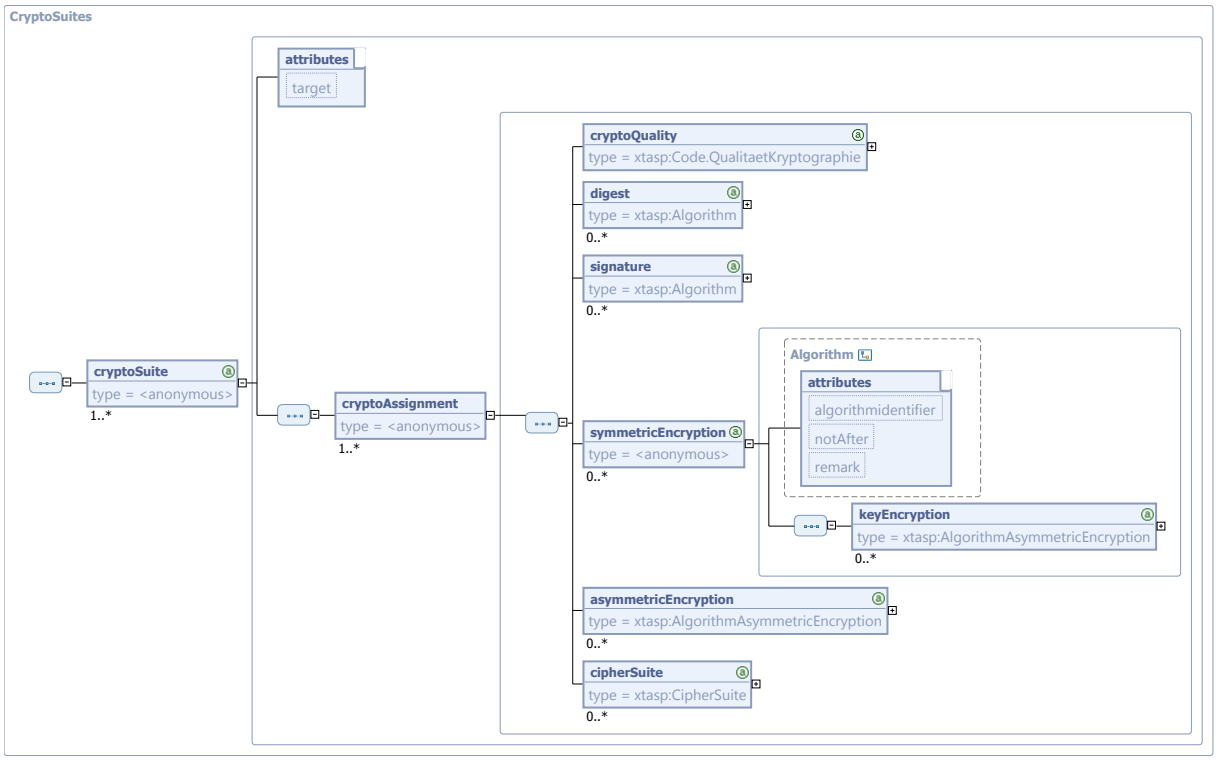
Kindelemente von *KryptographieProfil*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
profilkopf		**1**		
Dieses Element wird gefüllt mit Informationen zu Identität und Gültigkeit der vorliegenden Profil-Instanz. Hier ist die Versionsnummer der Kryptographieprofil-Instanz einzutragen.				
identifikation	*xtasp:Identifikation*	**1**	4.6.2.2	101
Unterhalb dieses Elements werden die Parameter zu Identität und Herkunft der vorliegenden Profil-Instanz gefüllt.				
dokumentation		**0..1**		
In diesem Bereich kann Dokumentation zum Profil eingetragen werden.				
beschreibung	*xs:string*	**1**		
In dieses Element können Erläuterungen zum Profil eingetragen werden.				
gueltigkeit	*xtasp:InstanzGueltigkeit*	**1**	4.6.2.3	102
Unterhalb dieses Elements werden Parameter zur Gültigkeit der vorliegenden Profil-Instanz gefüllt.				
datumVeroeffentlichung	*xs:date*	**1**		
Hier ist das Datum einzutragen, an dem die Kryptographieprofil-Instanz veröffentlicht wurde.				
kryptoSuiten	*xtasp:CryptoSuites*	**1**	4.6.1.4.2	90
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 in Abschnitt 4.4.4 auf Seite 63.				

4.6.1.4.2 CryptoSuites

Typ: *CryptoSuites*

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.

Abbildung 4.26. CryptoSuites



Kindelement von *CryptoSuites*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
cryptoSuite		**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 bestimmen Schutzniveau einer kryptographisch abzusichernden Kommunikation.				
target	*xs:string*	**1**		
Auf die Umsetzung dieses Formats oder Protokolls bezieht sich diese Krypto-Suite. Vorgesehen sind Krypto-Suiten für die Umsetzung von OSCI 1.2, OSCI 2, TLS und des Payloads eines Transportnachrichtenformats.				
cryptoAssignment		**1..n**		
Unterhalb dieses Elements wird einem bestimmten Schutzniveau einer kryptographisch abzusichernden Kommunikation (gelistete Krypto-Qualität, festgelegt durch das Element *cryptoQuality*) ein Set von Algorithmen und Schlüssellängen zugeordnet.				
cryptoQuality	*xtasp:Code.QualitaetKryptographie*	**1**	4.6.1.3. 10.1	88
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 *qualitaet* aus dem Schutzprofil in den Kontexten der Service Qualitäten Authentizität, Unveränderbarkeit und Vertraulichkeit (vgl. Abschnitt 4.6.1.1.2, „Schutzkategorie“ und Abschnitt 4.6.1.1.3, „Sicherheitskategorie“).				
digest	*xtasp:Algorithm*	**0..n**	4.6.1.4.3	92
Algorithmus-Deklaration für Digests ohne Erweiterungen.				
signature	*xtasp:Algorithm*	**0..n**	4.6.1.4.3	92

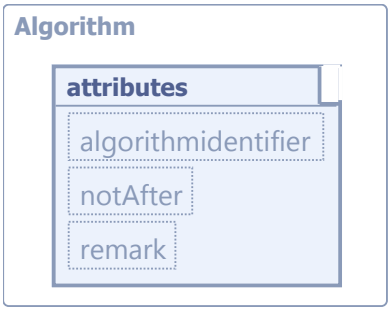
Kindelement von *CryptoSuites*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
Algorithmus-Deklaration für Signaturen ohne Erweiterungen.				
symmetricEncryption		**0..n**		
Algorithmus-Deklaration für symmetrische Verschlüsselung inkl. Schlüsselverschlüsselung.				
keyEncryption	*xtasp:AlgorithmAsymmetricEncryption*	**0..n**	4.6.1.4.4	92
Hier sind Algorithmen und Parameter zur Schlüsselverschlüsselung aufgeführt.				
asymmetricEncryption	*xtasp:AlgorithmAsymmetricEncryption*	**0..n**	4.6.1.4.4	92
Algorithmus-Deklaration für asymmetrische Verschlüsselung inkl. Schlüssellänge.				
cipherSuite	*xtasp:CipherSuite*	**0..n**	4.6.1.4.5	93
Algorithmus-Deklaration für Cipher-Suite.				

4.6.1.4.3 Algorithm

Typ: *Algorithm*

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

Abbildung 4.27. Algorithm



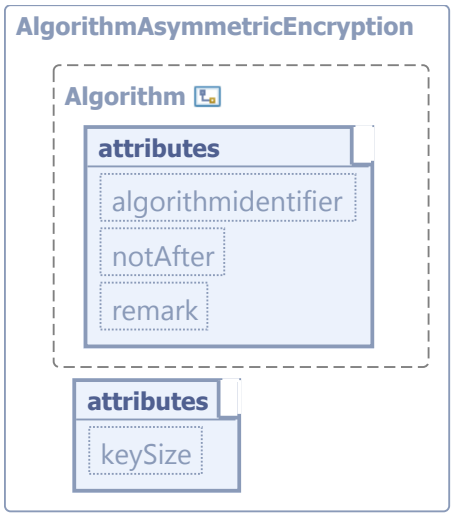
Kindelemente von *Algorithm*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
algorithmidentifier	*xs:string*	**1**		
Hier steht die Zeichenfolge, die zur Identifizierung des Algorithmus dient.				
notAfter	*xs:date*	**0..1**		
Hier ist das Datum zu nennen, bis zu dem der Algorithmus gültig ist.				
remark	*xs:string*	**0..1**		
Hier besteht die Möglichkeit, eine Bemerkung zu diesem Algorithmus einzutragen.				

4.6.1.4.4 AlgorithmAsymmetricEncryption

Typ: *AlgorithmAsymmetricEncryption*

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

Abbildung 4.28. AlgorithmAsymmetricEncryption



Dieser Typ ist eine Erweiterung des Basistyps *Algorithm* (siehe Abschnitt 4.6.1.4.3 auf Seite 92).

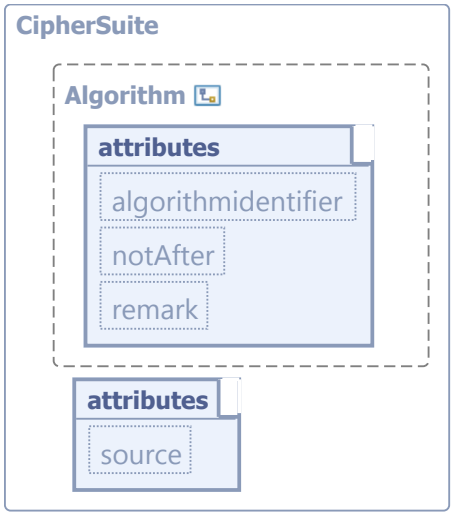
Kindelement von ***AlgorithmAsymmetricEncryption***				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
keySize	*xs:integer*	**1**		
Hier ist die zu verwendende Schlüssellänge einzutragen.				

4.6.1.4.5 CipherSuite

Typ: *CipherSuite*

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

Abbildung 4.29. CipherSuite



Dieser Typ ist eine Erweiterung des Basistyps *Algorithm* (siehe Abschnitt 4.6.1.4.3 auf Seite 92).

Kindelement von *CipherSuite*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
source	*xs:string*	**0..1**		
Die Quelle beschreibt die Herkunft des Algorithmus. Hier kann z. B. „WS-Security policy Spec“ oder „BSI Algo Catalog“ stehen.				

4.6.1.5 Datentypen des Service Profils

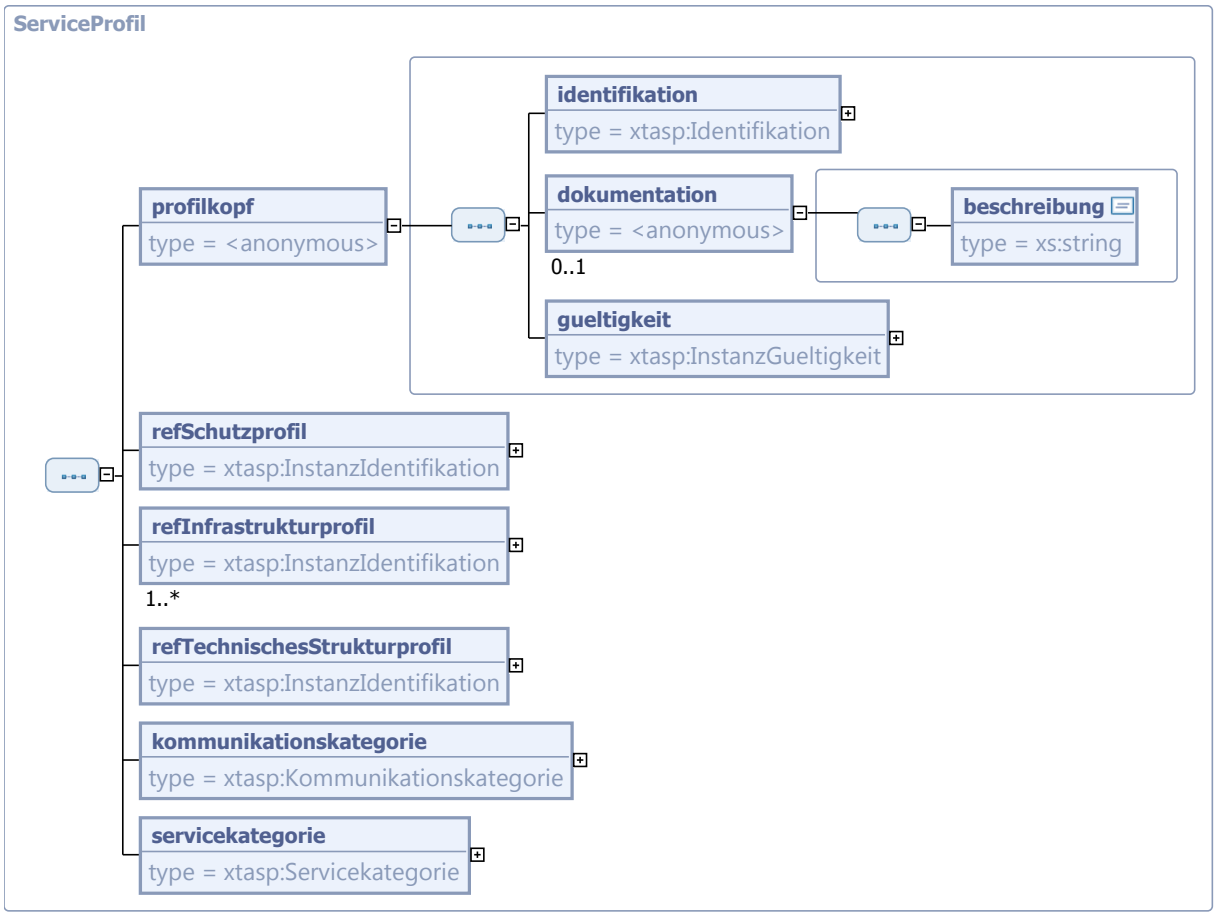
Die Typen, die hier dargestellt werden, werden im Kontext einer XML-Instanz angewendet, die auf dem globalen Element serviceProfil basiert.

4.6.1.5.1 ServiceProfil

Typ: *ServiceProfil*

Eine Instanz dieses Typs stellt ein ServiceProfil-Objekt (eine ServiceProfil-Instanz) dar. Es enthält Ausprägungen zu Service Qualitäten verschiedener Kategorien, welche durch die Messaging-Infrastruktur bei der Ausführung des Service, auf den sich das Profilobjekt bezieht, einzuhalten sind.

Abbildung 4.30. ServiceProfil



Kindelemente von *ServiceProfil*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
profilkopf		**1**		
Dieses Element wird gefüllt mit Informationen zu Identität und Gültigkeit der vorliegenden Profil-Instanz.				
identifikation	*xtasp:Identifikation*	**1**	4.6.2.2	101
Unterhalb dieses Elements werden die Parameter zu Identität und Herkunft der vorliegenden Profil-Instanz gefüllt.				
dokumentation		**0..1**		
In diesem Bereich kann Dokumentation zum Profil eingetragen werden.				
beschreibung	*xs:string*	**1**		
In dieses Element können Erläuterungen zum Profil eingetragen werden.				
gueltigkeit	*xtasp:InstanzGueltigkeit*	**1**	4.6.2.3	102
Unterhalb dieses Elements werden Parameter zur Gültigkeit der vorliegenden Profil-Instanz gefüllt.				
refSchutzprofil	*xtasp:InstanzIdentifikation*	**1**	4.6.2.1	100
Hier ist das Schutzprofil zu nennen, dessen Vorgaben für dieses Service Profil ausgewählt wurden. Es ist auf eine existierende Schutzprofil-Instanz zu referenzieren.				
refInfrastrukturprofil	*xtasp:InstanzIdentifikation*	**1..n**	4.6.2.1	100
Hier ist das Infrastrukturprofil zu nennen, dessen Vorgaben für dieses Service Profil ausgewählt wurden. Es ist auf eine existierende Infrastrukturprofill-Instanz zu referenzieren. Pro Element ist genau ein Infrastrukturprofil-Instanz zu referenzieren. Wenn mehrere Elemente instantiiert sind, dann liegt es in der Zuständigkeit des Betreibers der Rolle Sender, das für einen gegebenen Transportauftrag geeignete aus der Menge dieser Infrastrukturprofile auszuwählen.				
refTechnischesStrukturprofil	*xtasp:InstanzIdentifikation*	**1**	4.6.2.1	100
Hier ist das technische Strukturprofil zu nennen, dessen Vorgaben für dieses Service Profil ausgewählt wurden. Es ist auf eine existierende Profil-Instanz zu referenzieren.				
kommunikationskategorie	*xtasp:Kommunikationskategorie*	**1**	4.6.1.5.2	95
Durch die Kindelemente dieses Elements werden die Service Qualitäten der Kommunikationskategorie festgelegt (Modalitäten von Kommunikation und Zustellung), die dieses Service Profil vorschreibt.				
servicekategorie	*xtasp:Servicekategorie*	**1**	4.6.1.5.3	96
Hier werden die Service Qualitäten der Servicekategorie dieses Service Profils dargestellt.				

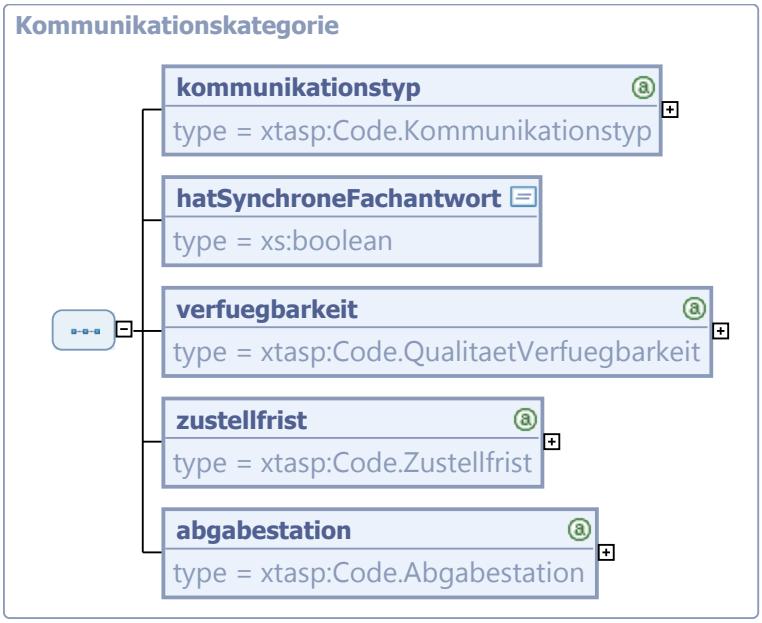
4.6.1.5.2 Kommunikationskategorie

Typ: *Kommunikationskategorie*

Eine Instanz dieses Typs beschreibt die Art der Interaktion, wie sie im Rahmen des Service ausgeführt werden soll. Es geht um Dinge wie asynchron vs. synchron, wie Zustell-Knoten und -Fristen. Jede Eigenschaft entspricht einer Service Qualität, die durch Benennung ihrer Ausprägung zu konfigurieren ist.

Die Möglichkeiten der Ausprägung der Service Qualitäten sind durch Codelisten hinterlegt.

Abbildung 4.31. Kommunikationskategorie



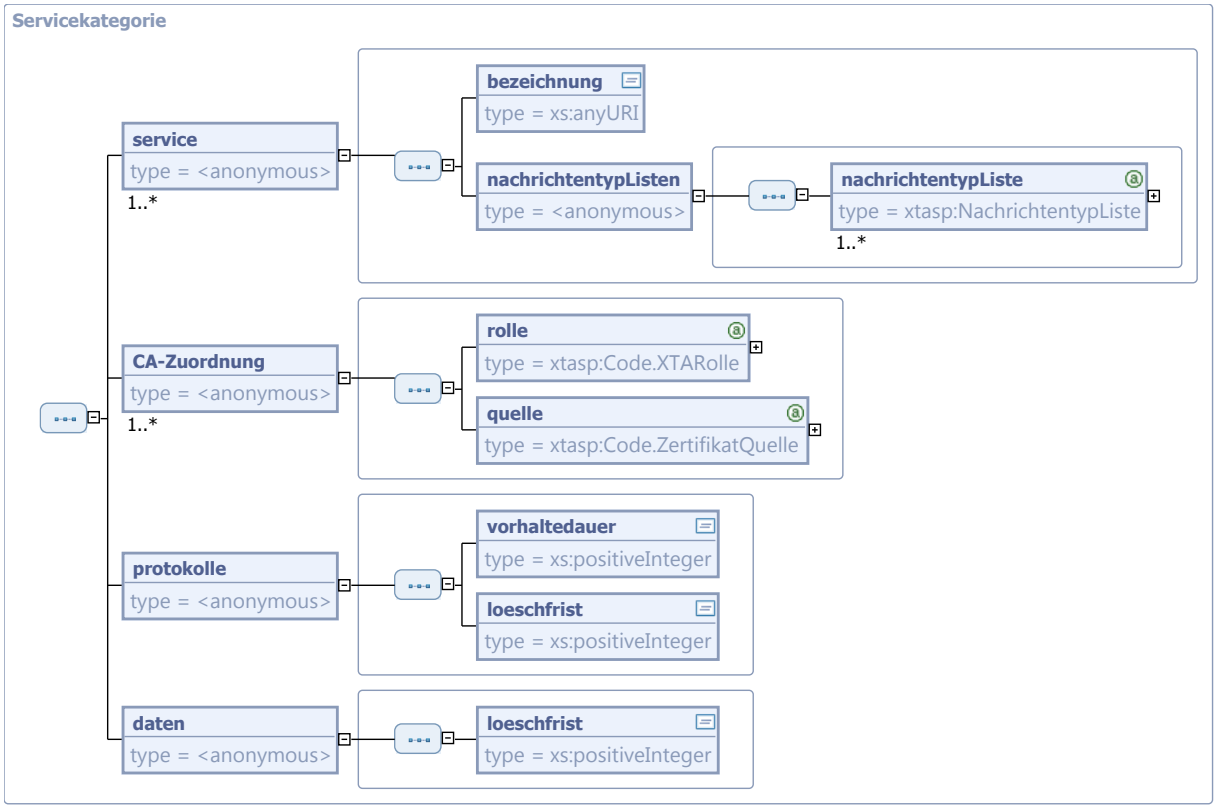
Kindelemente von *Kommunikationskategorie*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
kommunikationstyp	*xtasp:Code.Kommunikationstyp*	**1**	4.6.1.5.5.2	99
Hier wird festgelegt, wie die Grundstruktur (synchron oder asynchron) der Nachrichtenkommunikation im Rahmen des Service auszuführen ist.				
hatSynchroneFachantwort	*xs:boolean*	**1**		
Für synchrone Szenarien muss festgelegt sein, ob eine fachliche Antwort für die übermittelten Nachrichten im Rahmen des Service vorgesehen ist. Eine fachliche Antwort besteht darin, dass in der synchronen Antwort eine Fachnachricht enthalten ist. Für synchrone Szenarien mit Fachantwort ist hier „true“ zu wählen, für asynchrone Szenarien und für synchrone Szenarien ohne Fachantwort „false“.				
verfuegbarkeit	*xtasp:Code.QualitaetVerfuegbarkeit*	**1**	4.6.1.5.5.3	99
Hier wird die für Transportaufträge zum Service vorgesehene Verfügbarkeitsstufe eingetragen. Die Verfügbarkeit ist dabei die Wahrscheinlichkeit, dass der Transportauftrag innerhalb des vereinbarten Zeitraums ausgeführt wird.				
zustellfrist	*xtasp:Code.Zustellfrist*	**1**	4.6.1.5.5.4	100
Hier wird eine Zeitspanne genannt, innerhalb derer die Nachricht - gerechnet ab dem Zeitpunkt der Erteilung des Transportauftrags - zuzustellen ist, also die Zustellfrist von Transportaufträgen zu diesem Service.				
abgabestation	*xtasp:Code.Abgabestation*	**1**	4.6.1.5.5.5	100
Hier wird der Knoten der Infrastruktur bezeichnet, an dem die Nachricht final abzuliefern ist: Soll direkt dem Leser zugestellt werden, soll es eine Ablage in sein Postfach geben oder ist eine andere Variante vorgesehen?				

4.6.1.5.3 Servicekategorie

Typ: *Servicekategorie*

Dieser Typ beschreibt alle Eigenschaften mit ihren Ausprägungen, die den Service ausmachen und nicht einer anderen Kategorie oder einem anderen Profil zuzuordnen sind.

Abbildung 4.32. Servicekategorie



Kindelemente von *Servicekategorie*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
service		**1..n**		
Jedes Element steht für einen der Services, auf die sich das vorliegende Service Profil bezieht. Ein Service umfasst in der Regel einen fachlichen Kontext, welcher durch eine Bezeichnung kenntlich gemacht wird. Nachrichten können gemeinsam in einem Service zusammen empfangen werden. Wenn ein Autor eine Nachricht versenden will, dann muss er außer dem Nachrichtentyp auch die Bezeichnung des Service kennen, in dessen Kontext der Nachrichtentyp im Fachstandard einsortiert ist. Insofern sind Identifikation des Service und seiner Nachrichtentypen hier die wichtigsten Parameter.				
bezeichnung	*xs:anyURI*	**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. 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.				
nachrichtentypListen		**1**		
Unterhalb dieses Elements werden die Gruppen von Nachrichten eingetragen, die zu diesem Dienst (Service) gehören.				
nachrichtentypListe	*xtasp:NachrichtentypListe*	**1..n**	4.6.1.5.4	98

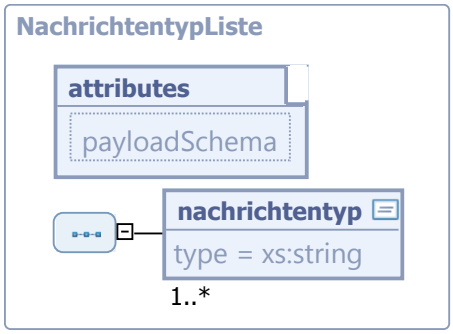
Kindelemente von *Servicekategorie*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
Dies ist eine Liste von Nachrichtentypen, die einem gemeinsamen Namespace zugeordnet sind.				
CA-Zuordnung		**1..n**		
Für die zu verwendenden Zertifikate lassen sich hier Vorgaben eintragen in Bezug auf die zugelassenen Herausgeber.				
rolle	*xtasp:Code.XTARolle*	**1**	4.6.1.3.10.2	89
Hier wird die Rolle in der XTA-Infrastruktur (Autor, Sender, ...) benannt, auf die sich die Vorgabe bezieht.				
quelle	*xtasp:Code.ZertifikatQuelle*	**1**	4.6.1.5.5.1	99
Hier ist eine Vorgabe in Bezug auf die Quelle der zugeordneten Zertifikate einzutragen.				
protokolle		**1**		
Unterhalb dieses Elements werden Festlegungen zum Umgang mit Protokollen in einer XTA-Infrastruktur eingetragen. Es geht dabei immer um Protokolle zu (Ereignissen in) Transportprozessen. Thema ist hier Vorhaltung und Löschung der Protokollinhalte. Die *Qualität* 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.				
vorhaltedauer	*xs:positiveInteger*	**1**		
Es ist die Anzahl Tage einzutragen, für die die Protokolle mindestens vorzuhalten sind.				
loeschfrist	*xs:positiveInteger*	**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.				
daten		**1**		
Es werden unterhalb dieses Elements 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.				
loeschfrist	*xs:positiveInteger*	**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.				

4.6.1.5.4 NachrichtentypListe

Typ: *NachrichtentypListe*

Eine Instanz dieses Typs ist eine Liste von Nachrichtentypen, die einem gemeinsamen Namespace zugeordnet sind.

Abbildung 4.33. NachrichtentypListe



Kindelemente von *NachrichtentypListe*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
payloadSchema	*xs:string*	**1**		
Hier ist der Namensraum des Fachstandards in einer bestimmten Version einzutragen, dem die Nachrichtentypen zugeordnet sind.				
nachrichtentyp	*xs:string*	**1..n**		
Pro Element ist ein Nachrichtentyp (Nachrichtenformat) aus dem Fachstandard einzutragen.				

4.6.1.5.5 Code-Datentypen Service Kategorie und Kommunikationskategorie

4.6.1.5.5.1 Code.ZertifikatQuelle

Code	**Code.ZertifikatQuelle**
Beschreibung	Diese Codeliste nennt Möglichkeiten für festzulegende Quellen (Herausgeber) von Zertifikaten für eine Public Key Infrastruktur.
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 186
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:zertifikat.quelle
Codelisten-Version	1.0

4.6.1.5.5.2 Code.Kommunikationstyp

Code	**Code.Kommunikationstyp**
Beschreibung	Diese Codeliste nennt die Arten, wie der Empfang einer Nachricht durch den Leser mit dem Absenden der Antwortnachricht durch ihn technisch gekoppelt sein kann.
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 168
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:kommunikationstyp
Codelisten-Version	1.0

4.6.1.5.5.3 Code.QualitaetVerfuegbarkeit

Code	**Code.QualitaetVerfuegbarkeit**
Beschreibung	Diese Codeliste beschreibt die für Transportaufträge vorgesehenen Verfügbarkeitsstufen. Die Verfügbarkeit ist dabei die Wahrscheinlichkeit, dass der Transportauftrag innerhalb des vereinbarten Zeitraums ausgeführt wird.
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 175

Code	Code.QualitaetVerfuegbarkeit
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:verfuegbarkeit.qualitaet
Codelisten-Version	1.0

4.6.1.5.5.4 Code.Zustellfrist

Code	Code.Zustellfrist
Beschreibung	Diese Codeliste nennt Zeitintervalle, die als Zustellfrist in Frage kommen. Die vorgegebene Zustellfrist eines Transportauftrags ist das Zeitspanne, innerhalb derer die Nachrichtenzustellung erfolgt sein muss.
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 187
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:zustellfrist
Codelisten-Version	1.0

4.6.1.5.5.5 Code.Abgabestation

Code	Code.Abgabestation
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.
Codelisten-Nutzung	Typ: 1, Inhalte der Codeliste siehe Seite 164
Codelisten-URI	urn:xoev-de:xta:serviceprofile:codeliste:abgabestation
Codelisten-Version	1.0

4.6.2 Übergreifende Typen für Profil-Instanzen

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

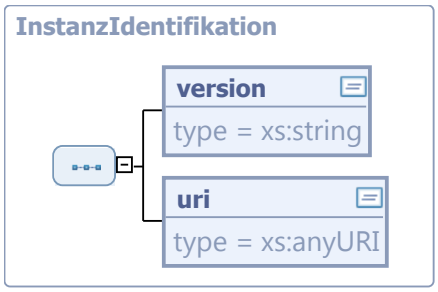
4.6.2.1 InstanzIdentifikation

Typ: *InstanzIdentifikation*

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 Profilnstanz.

Abbildung 4.34. InstanzIdentifikation



Kindelemente von *InstanzIdentifikation*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
version	*xs:string*	**1**		

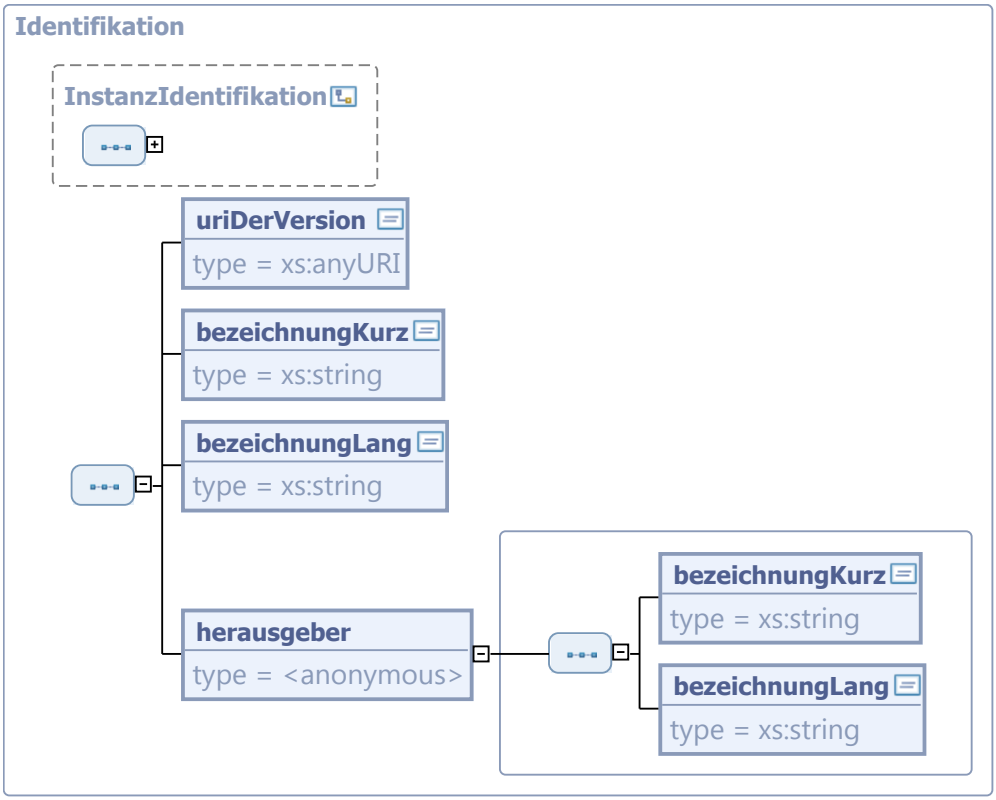
Kindelemente von *InstanzIdentifikation*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
Hier wird die Versionsbezeichnung der Profil-Instanz eingetragen. Die Versionsbezeichnung ist aus Zahlen, ggf. kombiniert mit Punkten, zu bilden. Weitere Zeichen (wie z. B. Unterstriche) sind nicht vorgesehen. Beispiel für eine Versionsbezeichnung: „1.1“				
uri	*xs:anyURI*	**1**		
Hier ist die URI einzutragen, durch die eine Profil-Instanz identifiziert wird. Die Bezeichnung hat die Form einer URN. Dieser Bezeichner wird versionsübergreifend interpretiert, er darf also keine Angabe einer Versionsnummer enthalten. Es sind die Bildungsregeln laut XÖV-Handbuch anzuwenden. Beispiel: *urn:xoev-de:xta:schutzprofile:fensterbriefumschlag*				

4.6.2.2 Identifikation

Typ: *Identifikation*

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

Abbildung 4.35. Identifikation



Dieser Typ ist eine Erweiterung des Basistyps *InstanzIdentifikation* (siehe Abschnitt 4.6.2.1 auf Seite 100).

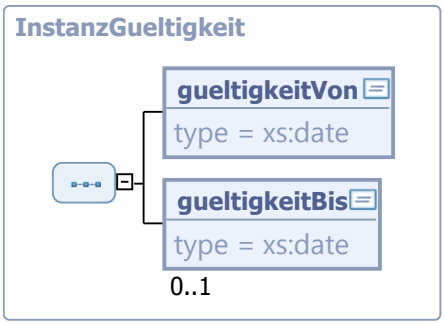
Kindelemente von *Identifikation*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
uriDerVersion	*xs:anyURI*	**1**		
Hier ist eine URI einzutragen, durch die diese Profil-Instanz identifiziert wird. Die Bezeichnung hat die Form einer URN mit angehänger Versionsnummer. Beispiel: *urn:xoev-de:xta:schutzprofile:fensterbriefumschlag_1.0.1*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

4.6.2.3 InstanzGueltigkeit

Typ: *InstanzGueltigkeit*

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

Abbildung 4.36. InstanzGueltigkeit



Kindelemente von *InstanzGueltigkeit*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
gueltigkeitVon	*xs:date*	**1**		
Hier ist, falls eine solche Festlegung vorgesehen ist, der Tag einzutragen, an dem die Gültigkeit der Profil-Instanz beginnt.				
gueltigkeitBis	*xs:date*	**0..1**		
Hier ist, falls eine solche Festlegung vorgesehen ist, der letzte Tag der Gültigkeit der Profil-Instanz einzutragen.				

4.6.3 Globale Elemente für XML-Instanzen

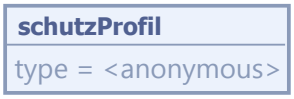
Hier werden die Elemente dargestellt, die als Wurzelelemente für Profilinstanzen dienen.

4.6.3.1 schutzProfil

Globales Element: *schutzProfil*

Das Schutzprofil deckt die Themen Datenschutz und Datensicherheit ab.

Abbildung 4.37. schutzProfil



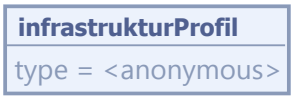
Dieser Typ ist eine Erweiterung des Basistyps *SchutzProfil* (siehe Abschnitt 4.6.1.1.1 auf Seite 69).

4.6.3.2 infrastrukturProfil

Globales Element: *infrastrukturProfil*

In einer Infrastrukturprofil-Instanz sind Bezeichnungen der Infrastruktur-Komponenten eingetragen, die bei der Umsetzung eines Service Profils einzusetzen sind.

Abbildung 4.38. infrastrukturProfil



Dieser Typ ist eine Erweiterung des Basistyps *InfrastrukturProfil* (siehe Abschnitt 4.6.1.2.1 auf Seite 76).

4.6.3.3 technischesStrukturprofil

Globales Element: *technischesStrukturprofil*

Im Technischen Strukturprofil wird – in Form von Regeln – zu den Ausprägungen der Service Qualitäten die technische Konfiguration der vorgesehenen Implementierung genannt.

Abbildung 4.39. technischesStrukturprofil

Dieser Typ ist eine Erweiterung des Basistyps *TechnischesStrukturprofil* (siehe Abschnitt 4.6.1.3.1 auf Seite 80).

4.6.3.4 kryptographieProfil

Globales Element: *kryptographieProfil*

Das Kryptographieprofil stellt allgemeine Krypto-Suiten (als Standard) für die verschiedenen Transporte (abzusichernde Kommunikation) und Inhaltsdaten (Payload) bereit, um ein gefordertes Schutzniveau auf die technische Implementierung abzubilden - gemäß des sich wandelnden Standes der Technik.

Abbildung 4.40. kryptographieProfil

Dieser Typ ist eine Erweiterung des Basistyps *KryptographieProfil* (siehe Abschnitt 4.6.1.4.1 auf Seite 89).

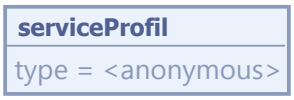
4.6.3.5 serviceProfil

Globales Element: *serviceProfil*

Service Profile nutzen bestehende Datenschutz,- Datensicherheits- und Kommunikationsprofile, um ihre Anforderungen an den Transport ihrer Nachrichten zu beschreiben. Zusätzlich werden noch servicespezifische Angaben gemacht. Zusammen werden daraus die Service Profile gebildet, die mit einem Namen versehen werden.

Ein Serviceprofil stellt die rechtlichen Grundlagen seiner Domäne in Form von Service Qualitäten mit definierter Syntax und Semantik zur Verfügung.

Abbildung 4.41. serviceProfil



Dieser Typ ist eine Erweiterung des Basistyps *ServiceProfil* (siehe Abschnitt 4.6.1.5.1 auf Seite 94).

5 XTA Webservice (2.1.1)

5.1 Überblick

Die Kommunikation zwischen Fachverfahren und Transportverfahren – also aus der Sicht des Rollenmodells die Kommunikation zwischen Autor und Sender - wird über eine Webservice-Schnittstelle realisiert, die XTA-Webservice („XTA-WS“) genannt wird. Diese Schnittstelle ist der einheitliche Zugang eines Fachverfahrens zu einer XTA-konformen Transport-Infrastruktur. Gleichzeitig entkoppelt sie das Fachverfahren von der technischen Komplexität der Transportprozesse. Der XTA-Webservice ist damit unabhängig von den beim Transport verwendeten Kommunikationsprotokollen (z.B. HTTPS oder OSCI-Transport).

Um diesen einheitlichen Zugang zu nutzen, wird in die Fachverfahren des Autors und des Lesers jeweils ein XTA-Webservice-Client eingebunden. Dieser ist in der Lage, die Funktionalitäten des XTA-WS, der von einem XTA-konformen Transportverfahren angeboten wird, aufzurufen.

Der XTA-Webservice wird durch eine oder mehrere WSDL-Dateien technisch beschrieben. Sie sind als Anlage der Spezifikation beigefügt. Der XTA-WS ist separat vom Fachverfahren implementiert und wird von Sender bzw. Empfänger zur Verfügung gestellt und betrieben. Der Betrieb kann zentral durch eine Clearingstelle (Vermittlungsstelle, Nachrichtenbroker) oder lokal durch eine Kommunikationssoftware (z.B. OSCI-Client) erfolgen.

Für durchgängig synchrone Prozesse ist ein Teil der Webservice-Schnittstelle auch durch den Leser zu implementieren. Auf die entsprechenden Operationen wird dann durch den Empfänger zugegriiffen. Dies wird weiter unten im Einzelnen beschrieben (siehe Abschnitt 5.4.2.2 auf Seite 120).

5.2 Rahmenbedingungen für die XTA-WS-Schnittstelle

5.2.1 XTA-WS als OSCI 2 Profilierung

Der XTA-Webservice ist als OSCI 2 Profilierung realisiert. OSCI 2 wurde im Auftrag der öffentlichen Verwaltung auf der Basis internationaler Webservice-Standards entwickelt.

OSCI 2 ist selbst eine Profilierung: Es basiert auf dem WS-Stack, der eine Menge von internationalen, auf Webservices basierenden Protokollstandards darstellt. Der WS-Stack bietet konfigurierbare Bausteine. Durch OSCI 2 wird die Konfiguration vorgenommen.

So wird durch die OSCI 2-Profilierung zum einen die Interoperabilität mit auf Webservice basierenden Lösungen sichergestellt. Zum anderen bewirkt die Profilierung eine Einengung auf die Anforderungen der deutschen Verwaltung an eine vertrauliche, verlässliche und rechtsverbindliche Kommunikation, wie sie u.a. durch das BSI gefordert werden.

In dieser XTA-Dokumentation werden die Methoden, die aus OSCI 2 stammen, dokumentiert, so dass ihre Funktion und Aufgabe innerhalb des XTA-WS deutlich werden. Die mit der Implementierung des XTA-WS betrauten Personen werden (zusätzlich) auf die OSCI 2- Spezifikation (in der Version 2.0.2) verwiesen. (Die im Projekt betrachteten Szenarien waren Anlass für Änderungen an der OSCI 2-Spezifikation, die in der Version OSCI 2.0.2 mündeten.)

5.2.2 Authentifizierung und Autorisierung

Die gegenseitige Authentifizierung und Autorisierung der Kommunikationspartner sind wesentliche Aufgaben, die für die Erreichung des geforderten Sicherheitsniveaus geleistet werden müssen. Im Modell der Rollen und Verantwortlichkeiten werden sie im Rahmen der wechselseitigen Prüfung der Identitäten der Akteure benannt.

So müssen sich der Autor gegenüber seinem Sender und der Leser gegenüber seinem Empfänger authentisieren, wobei die hierfür notwendigen Daten verschlüsselt übertragen werden müssen. Diese beiden Ziele werden durch die Verwendung von TLS erreicht, wobei der Autor bzw. Leser sich durch die Verwendung eines ihm zugeordneten Client-Zertifikats ausweist.

Der Sender bzw. Empfänger überprüft die Berechtigung des Clients zum Zugriff (Authentifizierung). Reicht die Angabe des Zertifikats nicht zur eindeutigen Identifikation aus, muss der Sender bzw. der Empfänger weitere Angaben beim Verbindungsaufbau (im SOAP-Header der Transportnachricht) mitgeben.

Für die Gegenrichtung, also für die Authentisierung des Senders durch den Autor benötigt dieser eine lokale Konfiguration für den Zugriff. Hierfür nutzt der Sender in der TLS Verbindung das ihm zugeordnete Zertifikat. Der Autor vergleicht dieses mit den, in der lokalen Konfiguration gespeicherten Angaben.

Entsprechend prüft der Leser die Authentisierung des Empfängers.

Zur Authentifizierung werden von WS-Interoperability grundsätzlich nur zertifizierte Webservicestandards zugelassen. Die vorliegende XTA-Schnittstellenversion erlaubt ausschließlich die SSL-Client-Authentifizierung.

Ob dem authentifizierten Benutzer der Zugang zum Webservice gewährt werden darf, entscheidet der XTA-WS im Rahmen der Autorisierung.

Die Autorisierung auf einen Account beim XTA-Betreiber erfolgt über das Zertifikat. Die Authentifizierung erfolgt mit Hilfe des Parameters **Identifier** (entspricht der Autor/Leser-Identifikation) im Typ **PartyType** des Objekts **oscimeta:MessageMetaData**, siehe Abschnitt 5.4.1.6.1 auf Seite 117. Es besteht die Möglichkeit, dass verschiedene Kunden gemeinsam ein Zertifikat nutzen und sich nur über die Autor/Leser-Identifikation unterscheiden. In diesem Fall, wenn das Zertifikat nicht genau einen Autor/Leser identifiziert, ist der Parameter AuthorIdentifier mandatorisch zu übergeben.

5.3 Beispielszenarien

Die Funktionen des XTA-Webservice (XTA-WS) sind von den Anforderungen der Rollen (vgl. Abschnitt 2.2 auf Seite 14)) und Anwendungsfälle (vgl. Kapitel 3 auf Seite 27) abgeleitet. Um dies zu veranschaulichen, werden hier die Aufgaben und Abläufe zwischen Autor und Sender bzw. Empfänger und Leser (mit Verweisen auf die Sätze des Rollenmodells in Abschnitt 2.2.2, „Die Rollen“) beispielhaft beschrieben und der Bezug zu den Methoden des XTA-WS hergestellt.

Ergänzt wird diese Dokumentation durch Beispielcode in Abschnitt B.1, „Beispielcode“ für den asynchronen und synchronen Versand und Empfang und für den Rückruf von Nachrichten.

Die gegenseitige Authentifizierung der Kommunikationspartner ist eine sich wiederholende Aufgabe, die über Übermittlung und Überprüfung von Client-Zertifikaten erfolgt. Dieser Arbeitsschritt wird in den beispielhaften Darstellungen jeweils vorausgesetzt und nicht explizit erwähnt.

5.3.1 Aufgaben des Autors

Die Aufgaben eines Autors bestehen im Versenden von Fachnachrichten und der Überwachung des Nachrichtentransports (siehe Rollenmodell A 1.1, A 5.2, A 8.1, A 8.2):

- Asynchroner Versand einer Nachricht (siehe Abschnitt 5.3.1.1 auf Seite 107)
- Synchroner Versand einer Nachricht (siehe Abschnitt 5.3.1.2 auf Seite 107)
- Rückruf einer Nachricht (siehe Abschnitt 5.3.1.3 auf Seite 108)

Vor der ersten Übertragung sind technische und organisatorische Rahmenbedingungen zu schaffen (vgl. Rollenmodell A 10.1, A 9.2, A 8.1), die hier vorausgesetzt werden.

5.3.1.1 Asynchroner Versand einer Nachricht

Bei einem asynchronen Versand beauftragt der Autor den Sender mit der Übertragung einer Fachnachricht. Zu einem späteren Zeitpunkt prüft der Autor den Status der Übertragung, bis diese eindeutig (durch Erfolg oder Misserfolg) beendet wurde.

1. Sendebereitschaft und -berechtigung

   Der erfolgreiche Versand setzt voraus, dass der Autor in der Lage ist, zu senden, und auch, dass der Leser in der Lage und berechtigt ist, zu empfangen (vgl. Rollenmodell A 2.2).

   XTA 2 Funktionalitäten:
   - Verbindung Autor - Sender (siehe Abschnitt 5.4.1.1 auf Seite 111)
   - Erreichbarkeit Autor - Leser (siehe Abschnitt 5.4.1.2 auf Seite 112)
2. Erstellung Nachricht

   Der Autor erstellt die zu übertragende Nachricht (vgl. Rollenmodell A 1.3, A 2.1, A 2.3, A 3.1, A 4.1).
3. Erstellung Transportauftrag

   Der Autor legt die Metadaten fest, die den Transportauftrag beschreiben, z. B. fachliche Adressierung, Service Qualitäten und den eindeutigen Identifikator (MessageID) des Transportauftrags (vgl. Rollenmodell A 5.2, A 6.1, A 7.1, A7.2)

   XTA 2 Funktionalitäten:
   - Erzeugung eines eindeutigen Identifikators (siehe Abschnitt 5.4.1.5 auf Seite 116)
4. Asynchroner Versand

   Der Autor übergibt die Nachricht zusammen mit dem Transportauftrag für den Versand an den Sender (vgl. Rollenmodell A 1.2, A 5.1).

   XTA 2 Funktionalitäten:
   - Asynchroner Versand (siehe Abschnitt 5.4.2.1 auf Seite 119)
5. Überwachung des Versands

   Der Autor überprüft, ob der Versand der Nachricht erfolgreich durchgeführt werden konnte (vgl. Rollenmodell A 8.1), z. B. ob die verwendeten Zertifikate gültig waren. Im Falle eines Misserfolgs muss er ggf. Eskalationsmaßnahmen ergreifen.

   XTA 2 Funktionalitäten:
   - Abruf eines Transportprotokolls (siehe Abschnitt 5.4.1.3 auf Seite 114)

5.3.1.2 Synchroner Versand einer Nachricht

1. Sendebereitschaft und -berechtigung

Für den Versand einer Nachricht ist es wichtig, dass nicht nur der Autor in der Lage ist, zu senden, sondern auch der Leser in der Lage und berechtigt ist, zu empfangen (vgl. Rollenmodell A 2.2).

XTA 2 Funktionalitäten:

- Verbindung Autor → Sender (siehe Abschnitt 5.4.1.1 auf Seite 111)
- Erreichbarkeit Autor → Leser (siehe Abschnitt 5.4.1.2 auf Seite 112)

2. Erstellung Nachricht

   Der Autor erstellt die Nachricht, die übertragen werden soll (vgl. Rollenmodell A 1.3, A 2.1, A2.3, A 3.1, A 4.1).

3. Erstellung Transportauftrag

   Der Autor legt die Metadaten fest, die den Transportauftrag beschreiben, z. B. fachliche Adressierung, Service Qualitäten und den eindeutigen Identifikator des Transportauftrags (vgl. Rollenmodell A 5.2, A 6.1, A 7.1, A7.2)

   XTA 2 Funktionalitäten:

   - MessageID erzeugen (siehe Abschnitt 5.4.1.5 auf Seite 116)

4. Synchroner Versand

   Der Autor übergibt die Fachnachricht mit dem Transportauftrag für die Kommunikation mit dem Leser an den Sender (vgl. Rollenmodell A 1.2, A 5.1).

   - Synchroner Versand (siehe Abschnitt 5.4.2.2 auf Seite 120)

5. Überprüfung der Kommunikation

   Der Autor überprüft, ob der Versand der Nachricht erfolgreich durchgeführt werden konnte (vgl. Rollenmodell A 8.1), z. B. ob die verwendeten Zertifikate gültig waren. Bei positivem Ergebnis kann er die Rückantwort des Lesers verarbeiten. Im Falle eines Misserfolgs muss er ggf. Eskalationsmaßnahmen ergreifen.

   XTA 2 Funktionalitäten:

   - Abruf eines Transportprotokolls (siehe Abschnitt 5.4.1.3 auf Seite 114)

5.3.1.3 Rückruf einer Nachricht

Wenn der Autor eine Nachricht für den asynchronen Versand unter Angabe einer Sendeverzögerung (Schalter NotBefore) an den Sender übermittelt hat, kann er den Versandauftrag zurückziehen. Dies ist nur möglich, wenn die Nachricht vom Sender noch nicht an den Empfänger übergeben wurde, also der Verzögerungstermin noch nicht erreicht ist.

1. Rückruf eines Versandauftrags

   Der Autor übergibt dem Sender den eindeutigen Identifikator (MessageID) des Transportauftrags der Nachricht, die nach Erteilung eines asynchronen Versandauftrags dennoch nicht verschickt werden soll (vgl. Rollenmodell A 8.2).

   - Rückruf einer Nachricht (siehe Abschnitt 5.4.1.4 auf Seite 114)

5.3.2 Aufgaben des Lesers

Die Aufgaben eines Lesers bestehen im Empfangen von Fachnachrichten und in der Überwachung des Nachrichtentransports (vgl. Rollenmodell D1.1, D 5.1, D 5.2):

- Asynchroner Empfang einer Nachricht (siehe Abschnitt B.1.2.1 auf Seite 192)
- Asynchroner Empfang von Metadaten (siehe Abschnitt B.1.2.3 auf Seite 194)
- Synchroner Empfang einer Nachricht (siehe Abschnitt B.1.2.4 auf Seite 195)

Vor der ersten Übertragung sind technische und organisatorische Rahmenbedingungen zu schaffen (vgl. Rollenmodell D10.1, D 9.2, D 9.1, D 8.3).

Bei jedem Empfang einer Nachricht hat der Leser eine Reihe von Aufgaben zu erfüllen:

- Der Leser muss seinen Empfänger authentifizieren (vgl. Rollenmodell D8.2).
- Der Leser muss eingehende Nachrichten syntaktisch und semantisch prüfen (vgl. Rollenmodell D1.2, D1.3, D2.1, D2.2, D2.3, D3.1, D3.2).
- Der Leser muss, wenn notwendig, Teile der Nachricht fachlich entschlüsseln (vgl. Rollenmodell D4.1).
- Der Leser muss gemäß den geforderten Service Qualitäten reagieren (vgl. Rollenmodell D6.1).
- Der Leser muss Informationen zum Nachrichtentransport bewerten (vgl. Rollenmodell D7.1, D8.1).

5.3.2.1 Asynchroner Empfang von Nachrichten

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

Der Leser holt die Liste der MessageIDs der Transportaufträge der abzuholenden Nachrichten. Er entscheidet, ob und wann er die zugehörigen Nachrichten abholt.

1. Liste der Nachrichten ermitteln

   Zuerst holt der Leser die Liste der MessageIDs. Er kann Selektionskriterien angeben: Möchte er nur ungelesene Nachrichten? In welchem Zeitraum sollen diese Nachrichten eingegangen sein (vgl. Rollenmodell D5.1)?

   XTA 2 Funktionalitäten:

   a. Liste der MessageIDs und Metadaten holen (siehe Abschnitt 5.4.3.1 auf Seite 133)
2. Nachrichten an Hand der MessageID des zugehörigen Transportauftrages abholen

   Der Leser verarbeitet die vom Empfänger abgerufenen MessageIDs. Hierfür erfolgen für jede MessageID drei Arbeitsschritte:

   - Der Leser kann sich zu jeder abzuholenden Nachricht den Report vom Empfänger mit allen Transportinformationen holen. Hierfür verwendet er die MessageID des zugehörigen Transportauftrages (vgl. Rollenmodell D7.1). An Hand der Informationen aus dem Report bewertet der Leser, ob die Nachricht fachlich verarbeitet werden darf (vgl. Rollenmodell D8.1).
   - Ist alles regelgemäß gelaufen, ruft er die entsprechende Nachricht ab.
   - Wenn er diese korrekt empfangen konnte, muss er noch den Empfang quittieren. Das kann er sofort tun, oder erst nach einer angemessenen Bearbeitung der Nachricht, z. B. wenn er sie lokal persistieren konnte (vgl. Rollenmodell D5.1).

   XTA 2 Funktionalitäten:

   - Abruf eines Transportprotokolls (siehe Abschnitt 5.4.1.3 auf Seite 114)
   - Abholen einer Nachricht für eine MessageID (siehe Abschnitt 5.4.3.2 auf Seite 135)
   - Quittieren der Abholung (siehe Abschnitt 5.4.3.3 auf Seite 136)

5.3.2.2 Sukzessiver Empfang von Metadaten

Der in Abschnitt 5.3.2.1 auf Seite 109 beschriebene Empfang von Nachrichten setzt voraus, dass nur *ein* Leser-Client auf ein Postfach (auf das Postfach einer Fachbehörde) zugreift (die Fachbehörde hat in dem Szenario ihr Postfach und greift mit einem einzigen Client darauf zu). Für den parallelen Zugriff durch mehrere Leser-Clients auf *ein* Postfach wird in Abschnitt 5.4.3.5.1 auf Seite 139 eine zweite Variante beschrieben. Diese gehört in der vorliegenden Version zum optionalen Teil der Methoden des XTA-Webservice.

Nicht in jedem Fall sollen direkt alle Nachrichten abgeholt werden, häufig werden nur die Metadaten benötigt. Diese können ausgewertet werden, um z. B. Nachrichten eines bestimmten Absenders abzuholen.

Dieses Abrufen von Metadaten kann parallel von mehreren Lesern erfolgen. Jeder Leser bekommt hierfür einen Zugriff auf die Listen der MessageIDs und Metadaten der abzuholenden Nachrichten. Jeder Leser bekommt beim Abholen disjunkte Mengen von MessageIDs und Metadaten. Für eine parallele Abholung durch mehrere Leser reserviert ein Leser in Schritt 1 einen Ressourcenhandle. Diesen reicht er an andere Leser weiter, die die folgenden Arbeitsschritte 2 und 3 parallel mit ihm durchführen.

1. Teilliste der Metadaten ermitteln

   Der Leser holt einen Teil der MessageIDs und Metadaten als Liste (Anhang A Rollenmodell D5.1). Hierbei kann er Selektrionskriterien (Stati: gelesen, alle; Eingrenzung des Empfangszeitraum). Anders als bei „Liste der Nachrichten ermitteln“ bekommt er ggf. nur eine Teilliste. Das ist von Vorteil, wenn viele Nachrichten zur Abholung bereitliegen, und nicht die Liste aller MessageIDs und Metadaten auf einmal übertragen werden sollen. Außerdem wird hierdurch erreicht, dass die Abholung mit mehreren Lesern parallel erfolgen kann. Liefert der Empfänger eine leere Liste von MessageIDs und Metadaten zurück, wurden die Daten zu allen Nachrichten abgeholt.

   XTA 2 Funktionalitäten:

   • Teilliste der MessageIDs und Metadaten holen (siehe Abschnitt 5.4.3.1 auf Seite 133)
2. Überprüfung der Kommunikation

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

   XTA 2 Funktionalitäten:

   • Abruf eines Transportprotokolls (siehe Abschnitt 5.4.1.3 auf Seite 114)
3. Nächste Teilliste der Metadaten abholen

   Der Leser hat für das Abholen der Teillisten von MessageIDs und Metadaten eine Ressourcenkennung erhalten. Unter Angabe dieser Kennung kann er nun die nächste Teilliste von MessageIDs und Metadaten abholen (Anhang A Rollenmodell D5.1). Sollten keine weiteren auf die Selektionskriterien entsprechenden Einträge vorliegen, liefert der Empfänger eine leere Liste zurück.

   XTA 2 Funktionalität:

   • Nächste Teilliste der MessageIDs holen (siehe Abschnitt 5.4.3.5.1.1.2 auf Seite 142)
4. Beenden der Abholung von Teillisten von Metadaten

   Hat der Leser alle Teillisten abgeholt, soll er das dem Empfänger mitteilen. Wie beim Beenden der Abholung von Nachrichten werden hierdurch Ressourcen beim Empfänger freigegeben, die sonst für den Leser reserviert blieben.

   XTA 2 Funktionalität:

   • Quittieren der Abholung (siehe Abschnitt 5.4.3.3 auf Seite 136)

5.3.2.3 Synchroner Empfang von Nachrichten

Bei einem synchronen Empfang leitet der Empfänger die eingehenden Informationen sofort an den Leser weiter. Dies ist nur möglich, wenn der Leser die vorgegebene Schnittstelle implementiert. Wird der Leser über die Schnittstelle vom Empfänger aufgerufen, dann kann er die Nachricht verarbeiten und das Ergebnis an den Empfänger zurückgeben. Der Empfänger leitet es an den Sender und der an den Autor weiter.

1. Empfangsbereitschaft des Leser

Der Leser muss die WS-Methode sendMessageSync gemäß der Spezifikation implementieren (vgl. Rollenmodell D5.2, D 8.3). Den Zugriff auf diese Methode muss der Leser dem Empfänger gewähren. Hierzu hat er diesem die notwendigen Informationen mitzuteilen, z. B. die URI.

2. Empfang von Nachrichten

   Der Leser wartet auf eingehende Nachrichten und reagiert hierauf unverzüglich (vgl. Rollenmodell D5.2). Nach einer Prüfung der Korrektheit des Transports liefert er als Rückgabe an den Empfänger das Ergebnis seiner Verarbeitung.

   XTA 2-Funktionalität:

   Synchroner Versand einer Nachricht (siehe Abschnitt 5.4.2.2 auf Seite 120)

5.4 Methoden

Der XTA-WS wird durch die nachfolgend dokumentierten und spezifizierten Methoden beschrieben, die in vier Schnittstellentypen zusammengefasst sind (vgl. die Dateien XTA.wsdl und XTA-synchron.wsdl):

- Im Schnittstellentyp **managementPort** werden Methoden zusammengefasst, die Serviceleistungen im Umgang mit Nachrichten anbieten (vgl. Abschnitt 5.4.1 auf Seite 111).
- Die Schnittstellentypen **sendPort** und **sendSynchronPort** stellen Methoden dar, die direkt die Erteilung eines Transportauftrages betreffen (vgl. Abschnitt 5.4.2 auf Seite 119 bzw. Abschnitt 5.4.4 auf Seite 144).
- Im Schnittstellentyp **msgBoxPort** sind Methoden zusammengefasst, die „Postkasten-Funktionen“ wahrnehmen (vgl. Abschnitt 5.4.3 auf Seite 133): Es handelt sich hier um Aufgaben, die beim Entgegennehmen und Abholen einer oder mehrerer Nachricht erledigt werden müssen.

Die Beispiele, die der Dokumentation der Methoden beigefügt sind, nutzen folgende Werte:

- Das Meldewesen benutzt als Präfix das Kürzel „ags“.
- Der Autor ist die Meldebehörde Stadt Testhausen mit dem amtlichen Gemeindeschlüssel 87654321.
- Der Leser ist die Meldebehörde Dorf Testdorf mit dem amtlichen Gemeindeschlüssel 76859403.
- Der fachliche Dienst hat die Bezeichnung „http://www.osci.de/xmeld18/xmeld18Fortschreibung.wsdl“

5.4.1 Schnittstellentyp managementPort

In diesem Schnittstellentyp sind alle Management-Methoden des XTA-WS zusammengefasst, die als Service-Leistungen zur Verfügung gestellt werden. Dies sind:

- checkAccountActive (siehe Abschnitt 5.4.1.1 auf Seite 111)
- lookupService (siehe Abschnitt 5.4.1.2 auf Seite 112)
- createMessageId (siehe Abschnitt 5.4.1.5 auf Seite 116)
- cancelMessage (siehe Abschnitt 5.4.1.4 auf Seite 114)
- getTransportReport (siehe Abschnitt 5.4.1.3 auf Seite 114)

5.4.1.1 Methode checkAccountActive (Verbindung Autor–Sender bzw. Leser-Empfänger)

Mit der Methode checkAccountActive kann der Autor prüfen, ob seine Verbindung zum Sender funktioniert. Der Autor fragt beim Sender an und gibt dabei Informationen über seine Identität mit, die diese nachweist. Die Angabe der Identität erfolgt nach den Vorgaben des jeweiligen fachlichen Kontextes.

Die Methode checkAccoutActive prüft also, ob der Webservice verfügbar ist und ein Account beim XTA-Betreiber eingerichtet ist.

Typischerweise erfolgt der Aufruf nach einer Änderung der Konfiguration oder für den Fall, dass technische Probleme auftreten.

5.4.1.1.1 Ergebnisse

- Kehrt die Methode ohne Fehler zurück, ist der XTA-WS erreichbar und der Account aktiv.
- 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.

5.4.1.1.2 Operation checkAccountActive

Input.

Soap Part	Name	Type
Header	AuthorIdentifier	oscimeta:PartyType (vgl. Seite 117)

Wesentliche Parameter:

- „oscimeta:PartyType (vgl. Seite 106)“: AuthorIdentifier ist die fachliche Identität des Autors. Dieser Parameter ist optional, wenn das Zertifikat genau einen Autor/Leser identifiziert, sonst ist er mandatorisch zu übergeben.

Output. Keine Rückgabewerte.

5.4.1.1.3 Beispielcode (Aufruf der Methode)

```
checkAccountActive(oscimeta:PartyType)
```

5.4.1.2 Methode lookupService (Erreichbarkeit Autor – Leser)

Der Autor nutzt die Methode lookupService um festzustellen, ob ein Leser prinzipiell für einen Dienst elektronisch erreichbar ist: Über die Methode werden Verzeichnisdienste wie das DVDV angefragt, um die Informationen abzurufen.

Mit der Antwort erhält der Autor Parameter, die er für die technische Adressierung bzw. Aufbereitung der Nachricht benötigt: Dies kann z. B. das Inhaltsdatenverschlüsselungszertifikat des Lesers für die Ende-zu-Ende-Verschlüsselung sein.

Um die Prüfung durchführen zu können, also die Methode aufzurufen, benötigt der Autor die fachliche Identität des Lesers und die Bezeichnung des fachlichen Dienstes.

Die Methode kann zur fachlichen Steuerung der Prozesse im Fachverfahren verwendet werden.

5.4.1.2.1 Ergebnisse

- Folgende Ergebnisse sind möglich:
  - <ServiceIsAvailable> = "true": Der Leser bietet den Dienst an.
  - <ServiceIsAvailable > = "false": Der Leser bietet den Dienst nicht an.
  - <ServiceIsAvailableUnknown> = "true": Der benötigte Verzeichnisdienst ist nicht erreichbar bzw. innerhalb eines vom Sender festgelegten Timeouts nicht erreichbar .

- Wenn vorhanden bzw. notwendige, werden weitere Parameter für die technische Adressierung des Lesers geliefert.
- Der technische Fehler (SoapFault) <PermissionDeniedException> entsteht, wenn ein nicht autorisierter Zugriff auf den XTA-WS stattfindet.
- Der technische Fehler (SoapFault) <ParameterIsNotValidException> wird zurückgegeben, wenn ein Pflicht-Übergabeparameter fehlt oder ein Übergabeparameter fehlerhaft ist. Der Übergabeparameter ist fehlerhaft,
  - wenn der Parameter <ServiceType> des LookupServiceType keine gültige Dienstebezeichnung (z.B. aus dem DVDV) repräsentiert oder
  - wenn der Parameter <ReaderIdentifier> des LookupServiceType keinen gültigen Wert enthält.
- Der technische Fehler (SoapFault) <XTAWSTechnicalProblemException> entsteht, wenn ein technischer Fehler im XTA-WS aufgetreten ist.

5.4.1.2.2 Operation lookupService

Input.

Soap Part	Name	Type
Header	AuthorIdentifier	oscimeta:PartyType (vgl. Seite 117)
Body	LookupServiceRequest	xta:LookupServiceRequest (vgl. Seite 155)

Wesentliche Parameter:

- „oscimeta:PartyType (vgl. Seite 106)“: AuthorIdentifier ist die fachliche Identität des Autors. Dieser Parameter ist optional, wenn das Zertifikat genau einen Autor/Leser identifiziert, sonst ist er mandatorisch zu übergeben.
- „xta:lookupServiceRequest“: Liste der zu prüfenden Empfänger
  - „oscimeta:Reader“: die fachliche Identität des Empfängers z.B. oscimeta:Identifier = „ags:76859403”
  - „xta:ServiceType“: Dienstebezeichnungen des fachlichen Dienstes, z.B. „http://www.osci.de/xmeld18/xmeld18Fortschreibung.wsdl"

Output.

Soap Part	Name	Type
Body	LookupServiceResponse	xta:LookupServiceResponse (vgl. Seite 155)

Rückgabewerte:

- „xta:LookupServiceResponse“: Zitat des xta:LookupServiceRequest (auf den dies hier die Reaktion ist) mit den Ergebnissen:
  - „oscimeta:Reader“: die fachliche Identität des Empfängers
  - „xta:ServiceType“: Dienstebezeichnungen des fachlichen Dienstes
  - „xta:IsServiceAvailableValue“: Ergebnis der Erreichbarkeitsprüfung
  - „xta:ServiceParameterType“: ggf. Rückgabe von durch den Autor im Fachszenario benötigten Erreichbarkeitsparamtern des Lesers (z. B. ein öffentlicher Zertifikat).

5.4.1.2.3 Beispielcode (Aufruf der Methode)

```
lookupService(oscimeta:PartyType, LookupServiceRequest)
```

5.4.1.3 Methode getTransportReport (Abruf eines Transportprotokolls)

Autor und Leser haben die Verantwortung für den korrekten Nachrichtentransport. Für die notwendige Überwachung stellen Sender und Empfänger eine Funktion zum Abruf des Transportprotokolls zur Verfügung. Dieses Protokoll (<TransportReport>), enthält Angaben zum konkreten Transportauftrag und zu den Ereignissen, die während des Transports protokolliert worden sind.

Mit der Methode getTransportReport können also der Autor vom Sender - und der Leser vom Empfänger - das Transportprotokoll abholen. Es kann also erfragt werden, ob zu einer bestimmten Nachricht ein Transportprotokoll vorliegt, z.B., um zu ermitteln, mit welchem Ergebnis die Zertifikatsprüfungen durchgeführt wurden und wie die Nachricht weitergeleitet wurde.

5.4.1.3.1 Ergebnisse

- Es wird das Transportprotokoll in einem <TransportReport> Objekt zurückgegeben.
- Der technische Fehler (SoapFault) <PermissionDeniedException> entsteht, wenn der Account gesperrt oder nicht vorhanden ist,
- Der technische Fehler (SoapFault) <XTAWSTechnicalProblemException> entsteht, wenn ein technischer Fehler im XTA-WS aufgetreten ist,
- Der technische Fehler (SoapFault) <InvalidMessageIDException> entsteht, wenn zu der übergebenen MessageID kein Transportprotokoll für den Account bekannt ist.

5.4.1.3.2 Operation getTransportReport

Input.

Soap Part	Name	Type
Header	AuthorIdentifier	oscimeta:PartyType (vgl. Seite 117)
Body	MessageID	wsa:MessageID (vgl. Seite 118)

Wesentliche Parameter:

- "oscimeta:PartyType"(vgl. Seite 106): AuthorIdentifier ist die fachliche Identität des Autoren.
- „wsa:MessageID“: eindeutiger Identifikator des Transportauftrags, z.B. „urn:de:xta:messageid:clearingstelleXY_xta_01:000ca2fe-f4e1-45c2-8233-3a0eb760bd16“

Output.

Soap Part	Name	Type
Body	GetTransportReportResponse	xta:TransportReport (vgl. Seite 156)

Rückgabewerte:

- „xta:TransportReport“: Es wird der Report mit den Informationen über den Transport geliefert.

5.4.1.3.3 Beispielcode (Aufruf der Methode)

```
getTransportReport(oscimeta:PartyType, wsa:MessageID)
```

5.4.1.4 Methode cancelMessage (Rückruf einer Nachricht)

Diese Methode soll einem Autor, der zu einem Transportauftrag **NotBefore** gesetzt hat, die Möglichkeit geben, den Transportauftrag vor Absenden zurückzuziehen. Die Methode ist also nur einsetzbar

im Zusammenhang mit dem Zeitstempel **NotBefore** (siehe **DeliveryAttributes** des Headers MessageMetaData). Die Methode ist nicht vorgesehen für den Fall ungeplanter Verzögerungen des Absendens einer Nachricht durch den beauftragten Sender.

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

- Der Transportauftrag zu der entsprechenden Fachnachricht wird über die MessageID eindeutig bezeichnet.
- Die zugehörige Fachnachricht gehört dem Autor.
- Die Fachnachricht wurde zuvor vom Autor zum asynchronen Versand unter Angabe des Schalters **NotBefore** übergeben.
- Der im Schalter **NotBefore** angegebene Termin darf noch nicht erreicht bzw. der Transportauftrag noch nicht bearbeitet worden sein.

Der Aufruf ist erfolgreich, wenn kein Fehler (Exception) zurückgegeben wird.

Wenn die Methode erfolgreich ausgeführt worden ist, geht der Ampelstatus im **TransportReport** auf rot. Begründung: Der Transportauftrag wurde nicht ausgeführt.

Der **TransportReport** wird zum Nachweis des Rückrufs der Nachricht vorgehalten, der Payload wird gelöscht (wie das i.d.R. auch nach erfolgreichem Versand geschieht).

5.4.1.4.1 Ergebnisse

- Kehrt die Methode ohne Fehler zurück, so ist der XTA-WS erreichbar und der Transportauftrag wurde zurückgezogen.
- Der technische Fehler (SoapFault) <PermissionDeniedException> entsteht, wenn ein nicht autorisierter Zugriff auf den XTA-WS stattfindet.
- Der technische Fehler (SoapFault) <ParameterIsNotValidException> wird zurückgegeben, wenn ein Pflicht-Übergabeparameter fehlt oder ein Übergabeparameter fehlerhaft ist.
- Der technische Fehler (SoapFault) <InvalidMessageIDException> entsteht, wenn die MessageID, also der angeforderte Transportauftrag, dem Account nicht bekannt ist.
- Der technische Fehler (SoapFault) <XTAWSTechnicalProblemException> entsteht, wenn ein technischer Fehler im XTA-WS aufgetreten ist.
- Der technische Fehler (SoapFault) <CancelDeniedException> entsteht, wenn der referenzierte Transportauftrag nicht zurückgezogen werden kann.

5.4.1.4.2 Operation cancelMessage

Input.

Soap Part	Name	Type
Header	AuthorIdentifier	oscimeta:PartyType (vgl. Seite 117)
Body	MessageID (Angabe der ID des Transportauftrags der zurückzuziehenden Fachnachricht)	wsa: MessageID (vgl. Seite 118)

Wesentliche Parameter:

- „oscimeta:PartyType (vgl. Seite 106)“: AuthorIdentifier ist die fachliche Identität des Autors. Dieser Parameter ist optional, wenn das Zertifikat genau einen Autor/Leser identifiziert, sonst ist er mandatorisch zu übergeben.

- „wsa:MessageID“: Übergabe der ID des Transportauftrags der zurückzuziehenden Fachnachricht, z.B. “urn:de:xta:messageid:clearingstelleXY_xta_01:000ca2fe-f4e1-45c2-8233-3a0eb760bd16"

Output. Keine Rückgabewerte.

5.4.1.4.3 Beispielcode (Aufruf der Methode)

```
cancelMessage(oscimeta:PartyType, wsa:MessageID)
```

5.4.1.5 Methode createMessageId (Erzeugung eines eindeutigen Identifikators)

Jeder Transportauftrag benötigt einen (räumlich und zeitlich) eindeutigen Identifikator. Diese MessageID kann über die gesamte Transportkette hinweg zur eindeutigen Identifikation des Transportauftrages oder zur Abfrage von Protokollinformationen verwendet werden.

Wie eine MessageID aufgebaut ist, ist Abschnitt 5.4.1.6.2 auf Seite 118 zu entnehmen.

Mit der Methode createMessageId veranlasst der Autor den Sender, eine Transportauftragsnummer zu generieren und zu liefern. Der Aufruf der Methode gehört zur Vorbereitung eines Transportauftrages, der über diese Nummer (MessageID) identifiziert werden soll.

5.4.1.5.1 Ergebnisse

- Es wird eine neu erzeugte MessageID zurückgeliefert.
- 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.

5.4.1.5.2 Operation createMessageID

Input.

Soap Part	Name	Type
Header	AuthorIdentifier	oscimeta:PartyType (vgl. Seite 117)

Wesentliche Parameter:

- „oscimeta:PartyType (vgl. Seite 106)“: AuthorIdentifier ist die fachliche Identität des Autors. Dieser Parameter ist optional, wenn das Zertifikat genau einen Autor/Leser identifiziert, sonst ist er mandatorisch zu übergeben.

Output.

Soap Part	Name	Type
Body	MessageID	wsa:MessageID (vgl. Seite 118)

Rückgabewerte:

- „wsa:MessageID“: Die vom Sender berechnete eindeutige MessageID für den Transportauftrag.

5.4.1.5.3 Beispielcode (Aufruf der Methode)

```
createMessageId(oscimeta:PartyType)
```

5.4.1.6 Wichtige Objekte der managementPort-Schnittstelle

5.4.1.6.1 PartyIdentifierType und PartyType

Die (generische) Adressierung erfolgt über die Kommunikationsendpunkte. Diese werden durch den Typ **PartyIdentifierType** modelliert: **PartyIdentifierType** ist die Typdefinition für die Instanzen der Source- und Target-Endpunkte **Originators** (Autor, auch Sender) und **Destinations** (Empfänger).

Der Identifier ist ein **xs:normalizedString**, attributiert durch **@type**, um das Interpretationsschema zum Eintrag in **PartyIdentifier** zu liefern (es kann sich z.B. um einen Identifier mit DVDV- oder mit S.A.F.E.-Benennungsschema handeln). Aus dem jeweils zugeordneten Verzeichnisdienst können die Verbindungsparameter entnommen werden, wie sie OSCI für WS-Addressing benötigt.

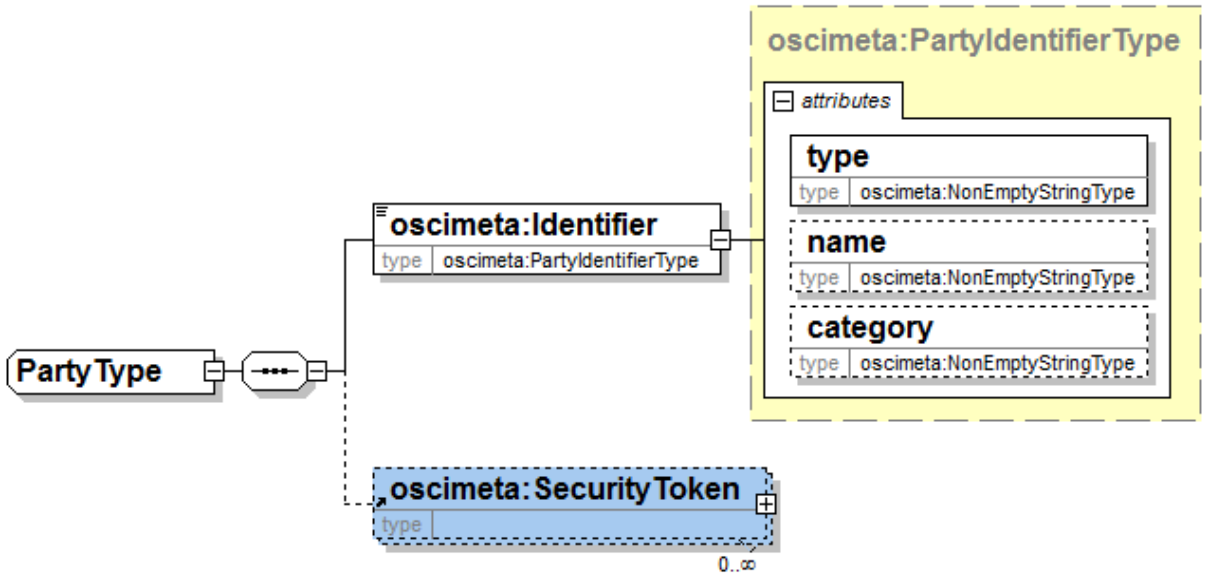
Wenn sich bei Entgegennahme des Transportauftrags durch den Sender im Element *Identifier* kein korrekter Eintrag befindet oder kein Eintrag, der sich zuordnen lässt, dann wird der Sender den Transportauftrag zurückweisen.

Um bei der Auswahl möglicher Inhalte für **@type** zu unterstützen, wird von der KoSIT die **Codeliste Type of Party Identifier** herausgegeben, welche einschlägige Einträge zur Verwendung in XTA 2 enthält. 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) unter Nennung ihrer Codelisten-URI *urn:de:xta:codeliste:type.of.party.identifier* auffindbar und kann dort im XML-Format OASIS Genericode in der aktuellen Version abgerufen werden.

Ergänzend wird **PartyIdentifierType** qualifiziert durch das Attribut **@category**, das optional die Angabe einer Behördenkategorie zum Eintrag in **PartyIdentifier** zu nennen gestatttet. Um die Semantik von **@category** zu spezifizieren, wird von der KoSIT die **Codeliste XOEV-Category of Party** herausgegeben, welche einschlägige Einträge für das Attribut zur Verwendung in XTA 2 enthält. Diese Codeliste ist im XRepository (www.xrepository.de) unter Nennung ihrer Codelisten-URI *urn:de:xta:codeliste:category.of.party* auffindbar und kann dort im XML-Format OASIS Genericode in der aktuellen Version abgerufen werden.

Für die Auflösung der generischen Adressen werden in OSCI 2 die WS-Addressing-Parameter beim Sender vom XTA WebService gesetzt.

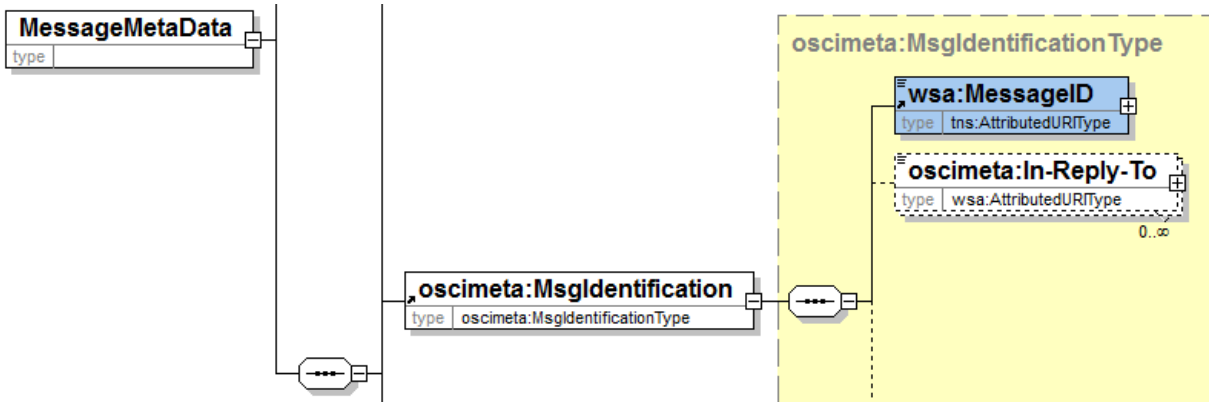
Die Authentisierungsinformationen (X509- oder SAML-Token) zu den Kommunikationsendpunkten werden durch den Typ **PartyType** übermittelt, der den **PartyIdentifierType** durch ein optionales Element **BinarySecurityToken** gemäß WS-Security ergänzt:



Die dargestellten komplexen Typen zur Aufnahme generischer Adressierungsinformationen werden in den Elementen instanziiert, die in dem Header **MessageMetaData** enthalten sind.

5.4.1.6.2 MessageID

Eine MessageID stellt eine eindeutige Identifizierung eines bestimmten Transportauftrags dar (Transportauftragsnummer). Damit entspricht ihre Semantik nicht derjenigen anderer MessageIDs, wie z. B. der MessageID aus OSCI-Transport 1.2 oder der ID der XÖV-Nachricht.



Die MessageID, welche in XTA 2 zu verwenden ist, hat die Syntax einer URN, deren spezifischer Teil aus zwei Komponenten besteht: einem den Aussteller eindeutig identifizierenden Präfix und einer UUID. Beispiel: „urn:de:xta:messageid:clearingstelleXY_xta_01:000ca2fe-f4e1-45c2-8233-3a0eb760bd16“.

Die MessageID ist eine zeitlich und räumlich unbegrenzt gültige eindeutige Identifikationsnummer (z.B. entsprechend WS-Adressing). Sie gewährleistet im Kontext des XTA-WS die eindeutige Identifikation eines Transportauftrags und wird normalerweise vom Fachverfahren (Autor/Leser) an das Transportverfahren (Sender/Empfänger) übergeben.

Für jeden Transportauftrag wird eine neue MessageID erzeugt, auch wenn dieselbe Fachnachricht 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. Selbst wenn ein Transportauftrag durch eine XTA-Exception abgewiesen worden ist, darf die MessageID nicht wiederverwendet werden.

Wenn im durch den Sender entgegengenommenen Transportauftrag keine korrekte MessageID enthalten ist, wird der er den Auftrag abweisen (Exception).

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 Fachnachricht (über deren Transportauftrag) über die gesamte Transportstrecke hinweg erleichtert.

Diese MessageID gilt sowohl für den Transportauftrag (einschließlich der zugehörigen Nachricht), wie auch für das dazugehörige Transportprotokoll. Die MessageID hat das Format einer URN. Sie sollte folgende Anforderungen erfüllen:

- Die MessageID muss beim Sender eindeutig sein. Sonst kann das Fachverfahren nicht den zugehörigen TransportReport abholen.
- Die MessageID muss beim Empfänger eindeutig sein. Sonst kann das Fachverfahren nicht den zugehörigen TransportReport abholen.
- Die MessageID soll erkennen lassen, wer die MessageID erstellt hat. So kann in Problemfällen der Ersteller der Nachricht leichter ermittelt werden.
- Die MessageID soll beim Empfänger dieselbe sein wie beim Sender. Nur so kann der Transportauftrag (einschließlich der zugehörigen Fachnachricht) über den gesamten Transwortweg eindeutig identifiziert werden.

Für die Erfüllung der ersten beiden Anforderungen reicht es aus, eine UUID als Identifikator zu verwenden. Wegen der dritten Anforderung wird ein Präfix hinzugefügt, welches einen Hinweis auf die ausstellende Softwareinstanz gibt, die die MessageID erstellt hat. 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_xta_01:000ca2fe-f4e1-45c2-8233-3a0eb760bd16"

5.4.2 Schnittstellentyp sendPort

In diesem Schnittstellentyp sind die Methoden des XTA-WS zusammengefasst, die für die Erteilung eines Transportauftrags angeboten werden. Dies sind:

- sendMessage (siehe Abschnitt 5.4.2.1 auf Seite 119)
- sendMessageSync (siehe Abschnitt 5.4.2.2 auf Seite 120)

5.4.2.1 Methode sendMessage (Asynchroner Versand einer Nachricht)

Für den asynchronen Versand erteilt der Autor dem Sender einen Transportauftrag. Dabei muss der Autor, der für den Transport verantwortlich ist, die folgenden Informationen mitgeben:

- die Fachnachricht,
- die Beschreibung des Transportauftrags (Metadaten) mit der MessageID,
- die Liste der zu prüfenden Zertifikate.

Diese Daten werden in einem Aufruf an den XTA-WS übergeben. Mit diesem Aufruf ist die Erteilung des Transportauftrags erfolgt.

Die Mitteilungspflicht des Fachverfahrens ist mit Aufruf dieser Methode durch den Webservice-Client erfüllt, wenn kein technischer Fehler (Exception) ausgelöst wurde. Dies enthebt den Autor allerdings nicht von der Pflicht, die Zustellung zu überwachen, also z.B. Transportprotokolle auszuwerten.

5.4.2.1.1 Ergebnisse

- Der technische Fehler (SoapFault) <PermissionDeniedException> entsteht, wenn ein nicht autorisierter Zugriff auf den XTA-WS stattfindet.
- Der technische Fehler (SoapFault) <ParameterIsNotValidException> entsteht, wenn ein Pflichtübergabeparameter fehlt oder ein Übergabeparameter fehlerhaft ist. Dies tritt in folgenden Fällen auf:
  - Der Parameter <MessageID> innerhalb des XTA 2 ist nicht eindeutig.
  - Der Parameter <ServiceType> repräsentiert keine gültige Dienstebezeichnung oder der Dienst wird vom Empfänger nicht angeboten.
  - Der Parameter <AuthorIdentifier> ist nicht gemäß der jeweiligen fachlichen Spezifikation gefüllt.
- Der technische Fehler (SoapFault) <XTAWSTechnicalProblemException> entsteht, wenn

  - ein technischer Fehler im XTA-WS aufgetreten ist,
  - der Empfänger nicht erreichbar ist,
  - der Empfänger nicht innerhalb eines vom Sender festgelegten Time-Outs antwortet.
- 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.
- Der technische Fehler (SoapFault) <SyncAsyncException> entsteht, wenn eine Fachnachricht übergeben wurde, die nur für den synchronen Versand gültig ist.

5.4.2.1.2 Operation sendMessage

Input.

Soap Part	Name	Type
Header	MessageMetaData	oscimeta:MessageMetaData (vgl. Seite 122)
Header	X509TokenContainer	osci:X509TokenContainer
Body	GenericContentContainer	xta:GenericContentContainer (vgl. Seite 154)

Wesentliche Parameter:

- „oscimeta:MessageMetaData“: In dieser Struktur werden die Metadaten des Transportauftrags definiert. Das Objekt ist als mandatorischer Parameter zu verwenden. Da die Daten dann als SOAP-Header zur Verfügung stehen, muss für diverse Zwecke nur dieser Header, aber nicht eine eingebettete Fachnachricht gelesen werden. Die Metadaten beinhalten Zeitstempel, Quittungsanforderungen, Service Profil, Angaben über den Autoren und den Leser, Informationen zur Identifikation der Fachnachricht und weitere Informationen.
- „osci:X509TokenContainer“: In diesem optionalen SOAP-Header können zu prüfende Zertifikate eingestellt werden. (Die Prüfung der Zertifikate kann auf dem Transportweg durchgeführt werden und ist eine optionale Serviceleistung der Transportinfrastruktur.) Beispiel für ein eingestelltes Zertifikat ist ein Signaturzertifikat für eine fachliche Signatur der zu übertragenden Fachnachricht im Kontext XhD.
- „xta:GenericContentContainer“: 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.

Output. Keine Rückgabewerte.

5.4.2.1.3 Beispielcode (Aufruf der Methode)

```
sendMessage(oscimeta:MessageMetaData, osci:X509TokenContainer,
 xta:GenericContentContainer)
```

5.4.2.2 Methode sendMessageSync - Sender (Synchroner Versand einer Nachricht)

Mit der Methode sendMessageSync kann der Autor über den Sender mit einem Leser kommunizieren: Der Autor schickt synchron eine Nachricht und bekommt (synchron) direkt eine Nachricht als Ergebnis

zurück. Die Methode sendMessageSync(-Sender) kann also nur in einem auf allen Teilstrecken synchronen Kommunikationsszenario genutzt werden.

Der Autor muss dabei alle notwendigen Informationen mitgeben, denn er ist für den Transport verantwortlich. Folgende Informationsblöcke müssen mitgegeben werden:

- die Fachnachricht,
- die Beschreibung des Transportauftrags (Metadaten) mit der MessageID
- die Liste der zu prüfenden Zertifikate.

Diese Informationen werden in einem Aufruf an den XTA-WS übergeben. Der Autor wartet, bis der Sender die Antwort des Lesers an ihn übergibt.

Mit der Methode sendMessageSync wird also eine Fachnachricht an den XTA-WS für einen synchronen Transport übergeben. Durch den Aufruf der Methode ist der Auftrag zum Transport erteilt.

5.4.2.2.1 Ergebnisse

- Im Erfolgsfall wird als Rückgabewert das Element <GenericContentContainer> zurückgegeben, das die Antwortnachricht des Lesers enthält.
- Der technische Fehler (SoapFault) <PermissionDeniedException> entsteht, wenn ein nicht autorisierter Zugriff auf den XTA-WS stattfindet.
- Der technische Fehler (SoapFault) <ParameterIsNotValidException> entsteht, wenn ein Pflichtübergabeparameter fehlt oder ein Übergabeparameter fehlerhaft ist. Dies tritt in folgenden Fällen auf:
  - Der Parameter <MessageID> ist innerhalb des XTA 2 nicht eindeutig.
  - Der Parameter <ServiceType> repräsentiert keine gültige Dienstebezeichnung oder der Dienst wird vom Empfänger nicht angeboten.
  - Der Parameter <AuthorIdentifier> ist nicht gemäß der jeweiligen fachlichen Spezifikation gefüllt.
- Der technische Fehler (SoapFault) <XTAWSTechnicalProblemException> entsteht, wenn
  - ein technischer Fehler im XTA-WS aufgetreten ist,
  - der Empfänger nicht erreichbar ist,
  - der Empfänger nicht innerhalb eines vom Sender festgelegten Time-Outs antwortet.
- 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 Fachnachricht 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.
- Der technische Fehler (SoapFault) <SyncAsyncException> entsteht, wenn eine Fachnachricht übergeben wurde, die nur für den asynchronen Versand gültig ist.

5.4.2.2.2 Operation sendMessageSync

Input.

Soap Part	Name	Type
Header	MessageMetaData	oscimeta:MessageMetaData (vgl. Seite 122)
Header	X509TokenContainer	osci:X509TokenContainer
Body	GenericContentContainer	xta:GenericContentContainer (vgl. Seite 154)

Wesentliche Parameter:

- „oscimeta:MessageMetaData“: In dieser Struktur werden die Metadaten des Transportauftrags definiert. Das Objekt ist als mandatorischer Parameter zu verwenden. Da die Daten dann als SOAP-Header zur Verfügung stehen, muss für diverse Zwecke nur dieser Header und nicht die ggf. eingebettete Fachnachricht gelesen werden. Die Metadaten beinhalten Zeitstempel, Quittungsanforderungen, das Service Profil, Angaben über den Autoren und den Leser, Informationen zur Identifikation der Fachnachricht und weitere Informationen.
- „osci:X509TokenContainer“: In diesem optionalen SOAP-Header können zu prüfende Zertifikate eingestellt werden. Die Prüfung kann auf dem Transportweg durchgeführt werden und ist eine optionale Serviceleistung der Transportinfrastruktur.
- „xta:GenericContentContainer“: 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.

Output.

Soap Part	Name	Type
Header	MessageMetaData	oscimeta:MessageMetaData (vgl. Seite 122)
Header	X509TokenContainer	osci:X509TokenContainer
Body	GenericContentContainer	xta: GenericContentContainer (vgl. Seite 154)

Rückgabewerte:

- „osci:X509TokenContainer“: In diesem optionalen SOAP-Header kann der Leser zu prüfende Zertifikate einstellen. Die Prüfung kann auf dem Transportweg durchgeführt werden und ist eine optionale Serviceleistung der Transportinfrastruktur.
- "oscimeta:MessageMetaData“: In dieser Struktur werden die Metadaten des Transportauftrags zu der Antwort des Lesers definiert. Das Objekt ist als mandatorischer Parameter zu übergeben. Da diese Daten dann als SOAP-Header mitgeführt werden, muss für diverse Zwecke nur dieser Header und nicht die eingebettete Fachnachricht gelesen werden. Die Metadaten beinhalten Zeitstempel, Quittungsanforderungen, das Service Profil, Angaben über den Autoren und den Leser, Informationen zur Identifikation der Fachnachricht und weitere Informationen.
- „xta:GenericContentContainer“: Dieses Objekt beinhaltet die Antwort des Lesers. Sie besteht aus einer Fachnachricht und einer beliebigen Anzahl von Anhängen (Attachments). Die Fachnachricht kann in einem verschlüsselten Container hinterlegt werden. Zu der Fachnachricht kann ein Betreff (Subject) angegeben werden.

5.4.2.2.3 Beispielcode (Aufruf der Methode)

```
sendMessageSync(ref oscimeta:MessageMetaData, ref osci:X509TokenContainer,
 ref xta:GenericContentContainer)
```

5.4.2.3 Wichtige Objekte der sendPort-Schnittstelle

5.4.2.3.1 Der Transportauftrag: Header-Block *MessageMetaData*

Der Header-Block *MessageMetaData* nimmt die Metadaten auf, die benötigt werden um den Nachrichtentransport zu beauftragen und durchzuführen. Die hervorzuhebenden Bestandteile dieser *Daten des Transportauftrags* lassen sich zwei Gruppen zuordnen. Dies sind ...

... die Transportdaten im engeren Sinne:

- von wem wurde die Fachnachricht erstellt: Angaben zum Autor

- an wen ist die Fachnachricht gerichtet: Angaben zum Leser
- Steuerungsinformationen für die Transportstationen, Service Qualitäten und auszustellende Quittungen

... sowie für Routing und Vorverarbeitung hilfreiche Metainformationen zum Payload:

- Identifikation und Prozesszusammenhang des Payload
- fachlicher Kontext, Service-Kontext und ggf. Nachrichtentyp des Payload

Zum Prozess von **Erstellung und Übergabe des Header-Blocks MessageMetaData** sind die folgenden Hinweise hervorzuheben:

- *Autor-Sender*: Der Header wird bei Erteilung des Transportauftrags durch den *Autor* erstellt und im Kontext des entsprechenden Methodenaufrufs an den *Sender* übergeben. Der *Sender* macht seine weiteren Aktivitäten wesentlich vom Inhalt dieses Headers abhängig. Der *Sender* wird, falls das Objekt technische Fehler haben sollte, den Transportauftrag nicht annehmen, sondern mit der passenden Exception zurückweisen.
- *Sender-Empfänger*: Die enthaltenen Informationen werden, je nach vom Sender angesprochener Messaging-Technologie, auf dem einen oder anderen Weg vom Sender zum Empfänger gelangen. Die Maßnahmen des Empfängers werden gesteuert durch diese Informationen. Was hat der Empfänger zu tun, falls ihm nur ein technisch defekter MessageMetaData-Header zur Verfügung gestellt wird? Dieser Fall ist nicht wesentlich unterschieden von einem Szenario, in dem ein MessageMeta-Data-Header fehlt. In diesem seltenen Fehlerfall wird der Empfänger sich anders behelfen, um den Prozess konform mit den rechtlichen Anforderungen und sonstigen Regeln weiterzuführen.

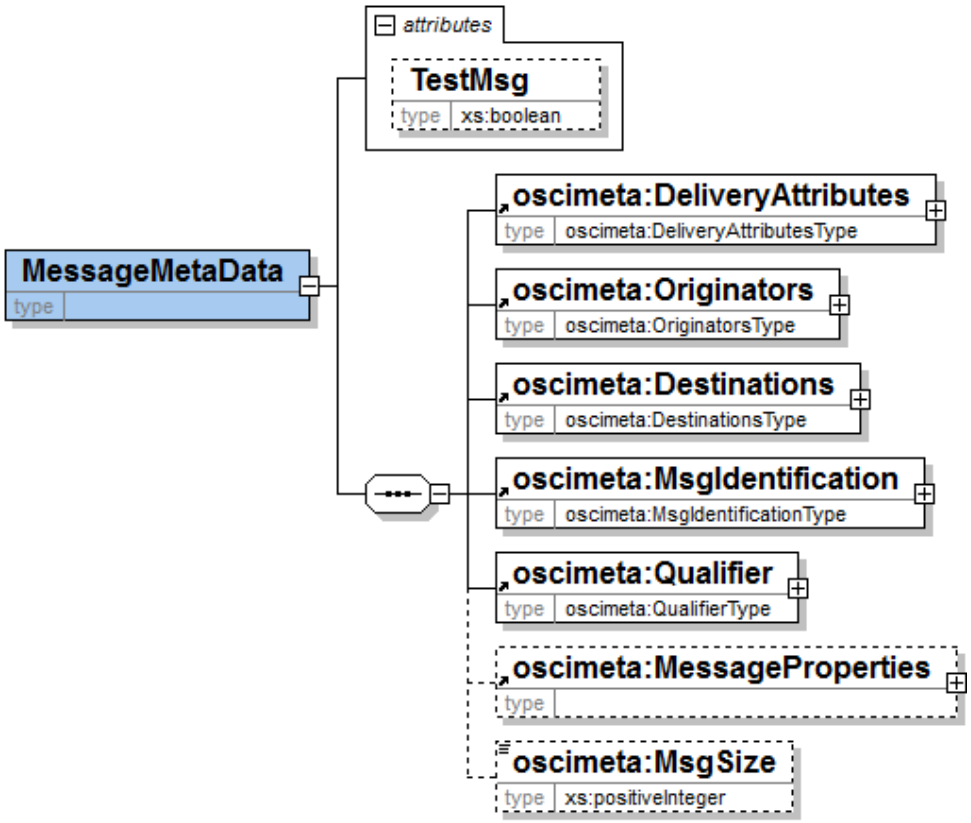
Protokollierung und Fortschreibung des MessageMetaData-Headers

Zu unterscheiden sind verschiedene Kontexte, in denen der MessageMetaData-Header protokolliert oder fortgeschrieben wird.

- *Protokollierung durch den Sender*: Der Sender führt den MessageMetaData-Header im TransportReport. Er legt ihn nach Entgegennahme des Transportauftrags (ist ihm durch den Autor übergeben worden) im TransportReport zu diesem Auftrag ab. Das Objekt wird im Anschluss fortgeschrieben gemäß Abarbeitung des Auftrags über die Knoten der Messaging-Infrastruktur. Hierfür wertet der Sender die Quittungen aus, die er im Verlauf dieser Abarbeitung von den anderen Knoten erhält.
- *Fortschreibung: durch die Messaging-Knoten*: Im Verlauf der Abarbeitung des Transportauftrags werden transportbezogene Daten von Knoten zu Knoten weitergereicht (Sender an Empfänger / Sender an Relay / Empfänger an Leser). Je nach verwendeter Messaging-Technologie wird hier der MessageMetaData-Header weitergeleitet werden. Dabei werden oft Einträge zu aktualisieren sein.
- *Protokollierung durch den Empfänger*: Auch der Empfänger führt einen TransportReport zum Transportauftrag, an dessen Abarbeitung er mitwirkt, um gegenüber dem Leser entsprechend Rechenschaft ablegen zu können. Er legt den MessageMetaData-Header, den er von Sender oder Relay übergeben bekommt bzw. den er sich aus erhaltenen Daten zusammengestellt hat, dort ab.

In den folgenden Abschnitten soll auf die Komponenten des Headers im Einzelnen eingegangen werden. Jeweils wird erläutert, welche Elemente im Angebot aus OSCI 2 enthalten sind und wie diese Elemente in XTA 2 verwendet werden. Die folgende Abbildung zeigt einen Überblick, im Anschluss ist dann den gezeigten Komponenten jeweils ein eigener Abschnitt gewidmet.

Der Header ist dem Standard OSCI 2 entnommen, vgl. Abschnitt 2.4.2 auf Seite 24.



Das Root-Element enthält ein optionales Attribut *@TestMsg*, das es bei Bedarf gestattet, den Kontext als Testnachricht auszuzeichnen.

Das Element *MsgSize* dient der Optimierung bei Empfangsknoten (Streaming) und wird von Sender oder Relay mit der Größe der Nachricht in Bytes gesetzt, falls kein Eintrag im Header enthalten war. Falls ein Eintrag enthalten ist, braucht er nicht geprüft und ggf. korrigiert zu werden.

5.4.2.3.2 MessageMetaData: Daten zur Steuerung des Transports

5.4.2.3.2.1 DeliveryAttributes - Zeitstempel

Die Zeitstempel unterhalb der Delivery Attributes setzen bzw. protokollieren den Zeitpunkt bestimmmter Ereignisse im Verlauf der Abarbeitung des Transportauftrags. Sie werden von den Knoten der Messaging-Infrastruktur entsprechend gesetzt (siehe auch das Objekt **MsgTimeStamps** aus OSCI 2, vgl. Abschnitt 2.4.2 auf Seite 24).

Die folgenden Attribute stehen zur Verfügung:

- *Origin*
- *ObsoleteAfter, NotBefore*
- *InitialSend*
- *Delivery*
- *InitialFetch*
- *Reception*

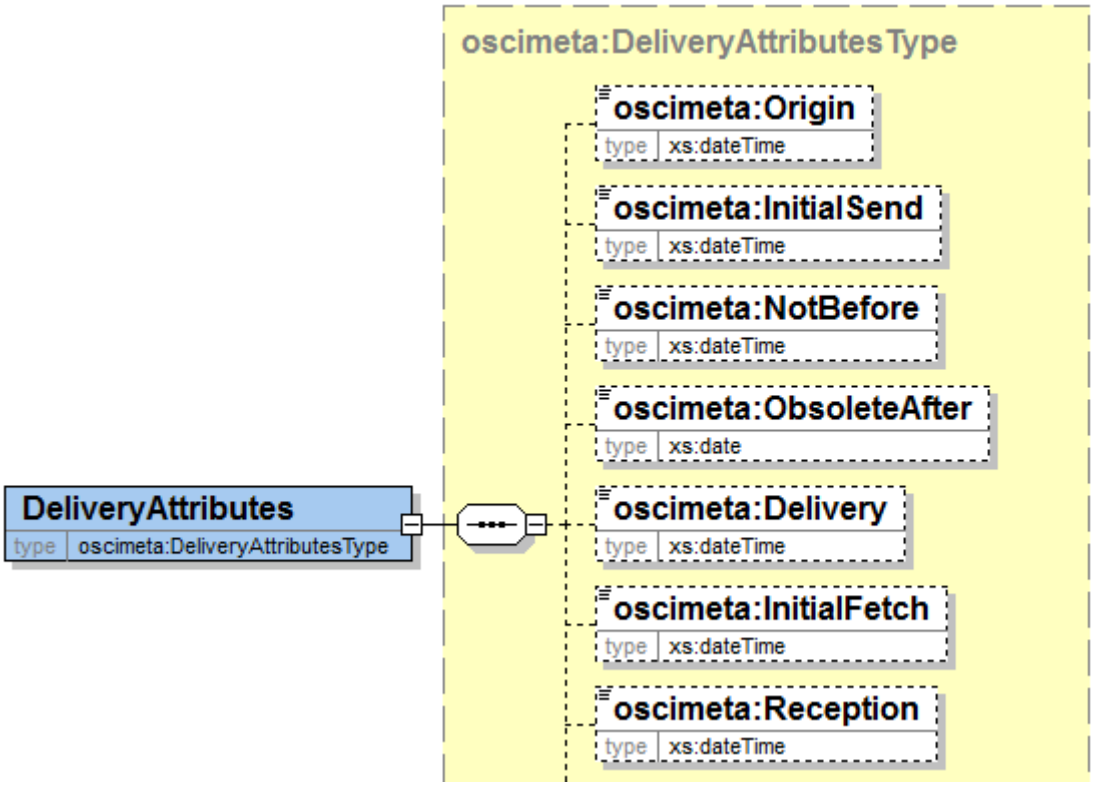


Tabelle 5.1, „Bearbeitung der Delivery Attributes in der 'Sender'-Infrastruktur“ listet die einzelnen Attribute, benennt die zu ihrer Befüllung (beim Erteilen des Transportauftrags) verantwortlichen Rollen und beschreibt die Auswirkung der Attribute auf den Status eines Transportauftrags (Eintragungen im TransportReport, der in der Sender-Infrastruktur auf Basis des Transportauftrags angelegt und fortgeschrieben wird).

In der Tabelle, Spalte 'Auswirkungen auf MessageStatus, besagt ein Eintrag 'Es kann Final State eintragen': Mit einem Eintrag in diesem Attribut kann der Transportauftrag - unter bestimmten Bedingungen (z.B. wenn keine weiteren Quittungen gefordert sind) - als erfolgreich abgeschlossen gewertet sein.

Tabelle 5.1. Bearbeitung der Delivery Attributes in der 'Sender'-Infrastruktur

Zeitstempel - Attribut	Befüllung: Autor oder Sender	Auswirkung auf MessageStatus (im TransportReport)	Bemerkung
Origin	Autor	Nein	Zu diesem Zeitpunkt wurde der Payload erstellt. Bei Fehlen: keine Korrektur oder Ergänzung durch Sender
InitialSend	Sender	Es kann Final State eintreten	Muss gesetzt werden. Eingetragen wird der Zeitpunkt, zu dem die Nachricht abgesendet wurde.
NotBefore	Autor	Nein	Funktionalität zum Eintragen einer Sendeverzögerung. Kann - nur bei asynchronem Versand - vom Autor gesetzt werden: Der Sender darf die Nachricht erst zu diesem Zeitpunkt versenden.

Zeitstempel - Attribut	Befüllung: Autor oder Sender	Auswirkung auf MessageStatus (im TransportReport)	Bemerkung
ObsoleteAfter	Autor	Ja, bei Fristüberschreitung negative Auswirkung	Funktionalität, durch die der Autor eine Frist setzen kann: Nach diesem Zeitpunkt darf die Nachricht nicht mehr versendet werden.
Delivery	Sender	Es kann Final State eintreten	Zeitpunkt des Eintreffens der Nachricht bei Empfänger bzw. MsgBox. Der Sender trägt ein auf der Basis einer entsprechenden Quittung, die ihm vom Empfänger zugestellt wurde.
InitialFetch	Sender	Es kann Final State eintreten	Zeitpunkt, zu dem die Nachricht erstmals aus der MsgBox abgerufen wurde. Sender trägt ein nach Eintreffen einer entsprechenden Quittung des Empfängers (bzw. der MsgBox des Empfängers)
Reception	/	/	Dieses Feld wird in einer XTA 2-Infrastruktur auf Sender-Seite nicht geführt.

Überschreibung oder Löschung der einmal gemachten Einträge im TransportReport ist nicht vorgesehen.

Im Bereich der Rollen *Leser* und *Empfänger* werden dieselben Attribute eingesetzt, aber in anderem Kontext. Es geht dabei um Eintragungen, die im *Empfänger*-TransportReport vorgenommen werden. Dieses Protokoll legt der Empfänger an, sobald er den Transportauftrag nach Eintreffen über die Messaging-Infrastruktur entgegengenommen hat. Er schreibt es fort auf der Basis der weiteren Schritte und Ereignisse der Abarbeitung des Transportauftrags.

Tabelle 5.2, „Bearbeitung der Delivery Attributes in der 'Empfänger'-Infrastruktur“ listet zu den einzelnen Attribute die an ihrer Befüllung (beim Erteilen des Transportauftrags) beteiligten Rollen und beschreibt die Auswirkung der Attribute auf den Status des vom Empfänger geführten Transportauftrags.

Tabelle 5.2. Bearbeitung der Delivery Attributes in der 'Empfänger'-Infrastruktur

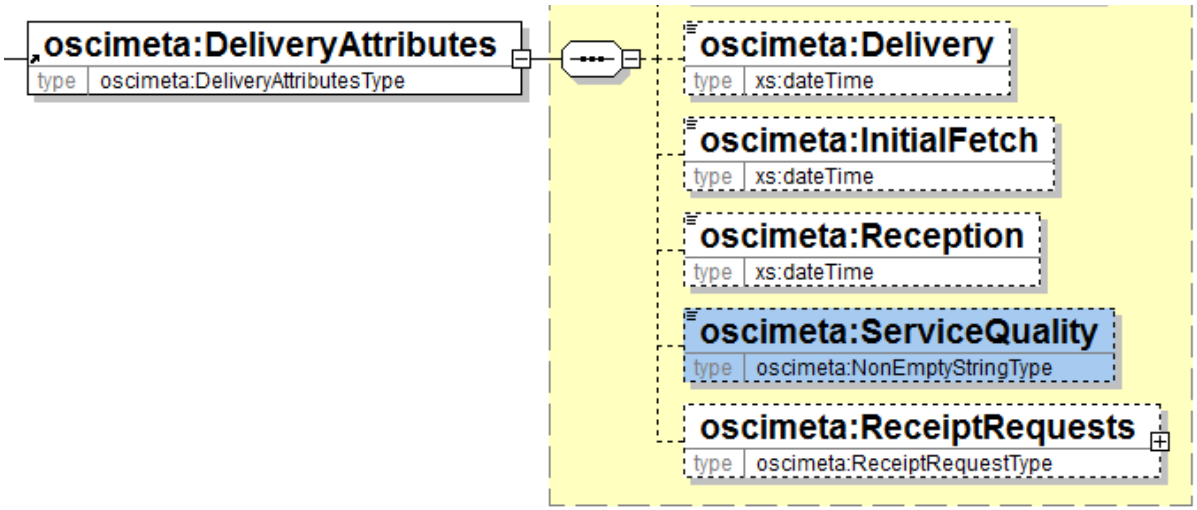
Zeitstempel - Attribut	Befüllung: Leser oder Empfänger	Auswirkung auf MessageStatus (im TransportReport)	Bemerkung
Origin	/		
InitialSend	/		
NotBefore	/		
ObsoleteAfter	/		
Delivery	Empfänger	Es kann Final State eintreten	Zeitpunkt Eingang beim Empfänger. Muss von der MsgBox (im asynchronen Szenario) bzw. Empfänger (im synchronen Szenario) gesetzt werden.
InitialFetch	Empfänger	Es kann Final State eintreten	Zeitpunkt des ersten Abrufs aus der MsgBox. Muss von MsgBox beim initialen Abholen einer Nachricht gesetzt werden.
Reception	Empfänger		Der Eintrag bedeutet: Leser hat Nachricht zu diesem Zeitpunkt abgeholt. Wird vom Empfänger gesetzt, sobald der Leser die

Zeitstempel - Attribut	Befüllung: Leser oder Empfänger	Auswirkung auf MessageStatus (im TransportReport)	Bemerkung
			Abholung der Nachricht mit Methode close() oder nextMessage() bestätigt hat. Muss vom Leser gesetzt bzw. getriggert werden.

Der in OSCI2.0.2 definierte Header **MsgTimeStamps** wird aus Kompatibilitätsgründen weiter in OSCI-Nachrichten mitgeführt. Die Einträge in **MsgTimeStamps** müssen von OSCI2.0.2-Gateways bzw. OSCI2.0.2-MsgBox-Services entsprechend der oben aufgeführten Elemente in **MessageMetaData** transparent gesetzt werden. Der damit redundante Header **MsgTimeStamps** wird möglicherweise in einer zukünftigen Version von OSCI Transport entfallen.

5.4.2.3.2.2 Delivery Attributes - Service Quality

Das Element *ServiceQuality* ist vorgesehen, um in den Daten des Transportauftrags die Service Qualitäten ansprechen zu können, die bei der Ausführung des Transportauftrags zu berücksichtigen sind.



In XTA 2 wird das Thema Service Qualitäten durch die Spezifikation der XTA Service Profile (vgl. Kapitel 4 auf Seite 49) behandelt. Die Verwendung des Elements ServiceQuality im XTA-Webservice ist daher eng an diesen Teil der Spezifikation gebunden.

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. Eine ServiceProfil-Instanz ist eine XML-Instanz auf der Basis des globalen Elements gemäß Spezifikation in Abschnitt 4.6.3.5 auf Seite 104.
- Die Vorgabe des Fachstandards (also das von ihm definierte Service Profil) muss bei der Ausführung des Transportauftrags (durch alle beteiligten Infrastrukturknoten) befolgt werden. Der Sender ist verpflichtet, zu prüfen, ob der Eintrag korrekt vorgenommen ist.

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

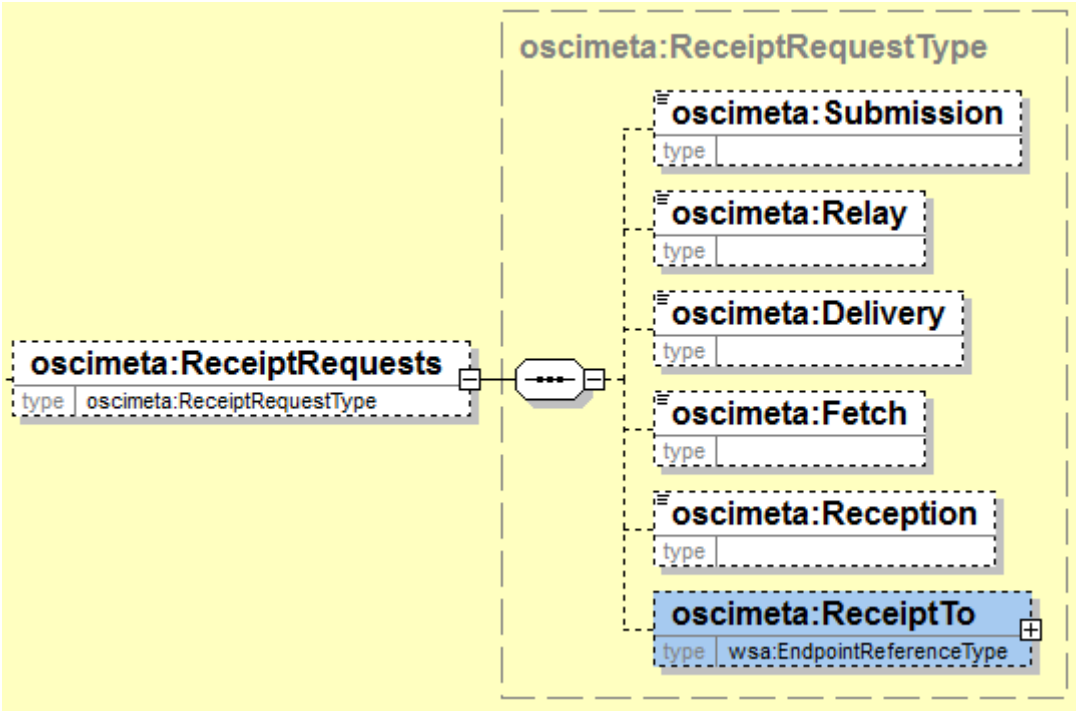
- Es wird dann eine lokal definierte ServiceProfil-Instanz referenziert.
- Alternativ wird das Element *nicht* verwendet (insbesondere dürfen keine informellen Inhalte als Freitext eingetragen werden).

Allgemein gilt:

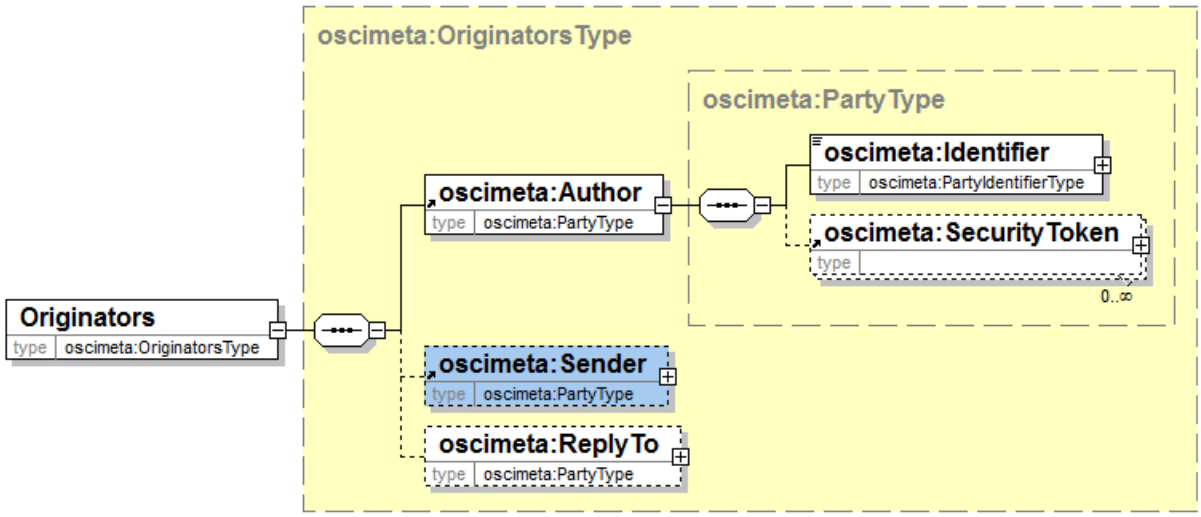
- In allen Fällen, in denen das Element *ServiceQuality* verwendet wird, ist eine Referenz auf eine (global oder lokal definierte) ServiceProfil-Instanz einzutragen.
- Das ist umzusetzen, indem der Inhalt des Elements *uriDerVersion* (vgl. Abschnitt 4.6.2.2, „Identifikation“) einer zu referenzierenden ServiceProfil-Instanz in das Element *ServiceQuality* des in Rede stehenden Transportauftrags eingetragen wird.

5.4.2.3.2.3 DeliveryAttributes - Receipt Requests

ReceiptRequests: Dieses Objekt gestattet es dem Initiator, im Rahmen des Transportauftrags Quittungen zu beauftragen. Grundsätzliches zum Thema Quittungen in XTA 2 ist Abschnitt 2.3 auf Seite 21 zu entnehmen. Dort ist auch geregelt, wie sich die im Service Profil festgelegten Quittungen (vgl. Element technischeQuittungen in Abschnitt 4.6.1.1.2, „Schutzkategorie“) zu den hier dokumentierten Einträgen im Objekt ReceiptRequests des Transportauftrags verhalten.



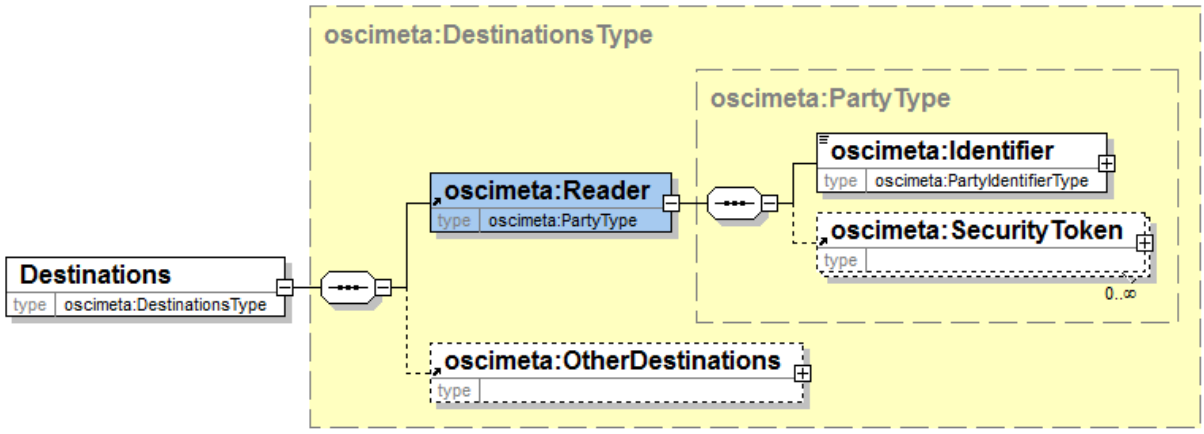
5.4.2.3.2.4 Originators



Die Elemente sind vom Typ *PartyType*. Neben dem generischen Identifier (*PartyIdentifierType*) können Authentisierungstoken (X509, auch SAML) aufgenommen werden (vgl. auch Abschnitt 5.4.1.6.1 auf Seite 117).

- **Author:** Obligatorisch. Entspricht Initiator, Source Application, WS-Addressing **From.**
- **Sender:** Sendender Knoten / OSCI Gateway, z.B. XTA-WS.
- **ReplyTo:** Zieladresse, an die Antwort gesendet werden soll; Default = **Author,** dies entspricht der Semantik von WS-Addressing.

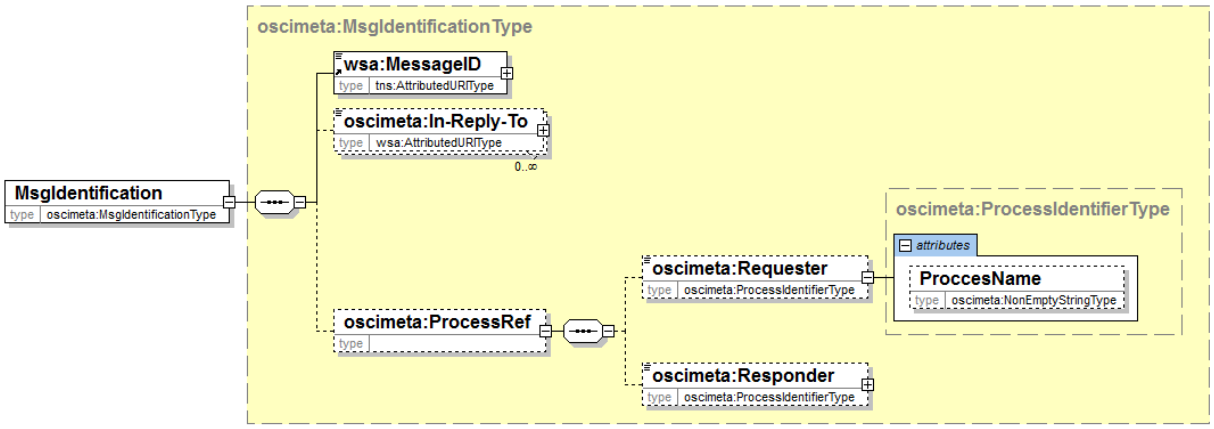
5.4.2.3.2.5 Destinations



- **Reader:** Obligatorisch; entspricht Target Application, Service Provider. Muss vom Autor gesetzt werden.
- **OtherDestinations:** Informatorisch, optionaler Eintrag: weitere Adressaten der Nachricht, unterschiedlich in **OtherReaders** und **CCReaders**. Es ist nicht vorgesehen, zum generischen „ **PartyIdentifier** “ auch optionale Authentisierungstoken aufzunehmen.

5.4.2.3.3 MessageMetaData: Identifikation und Metadaten zum Payload

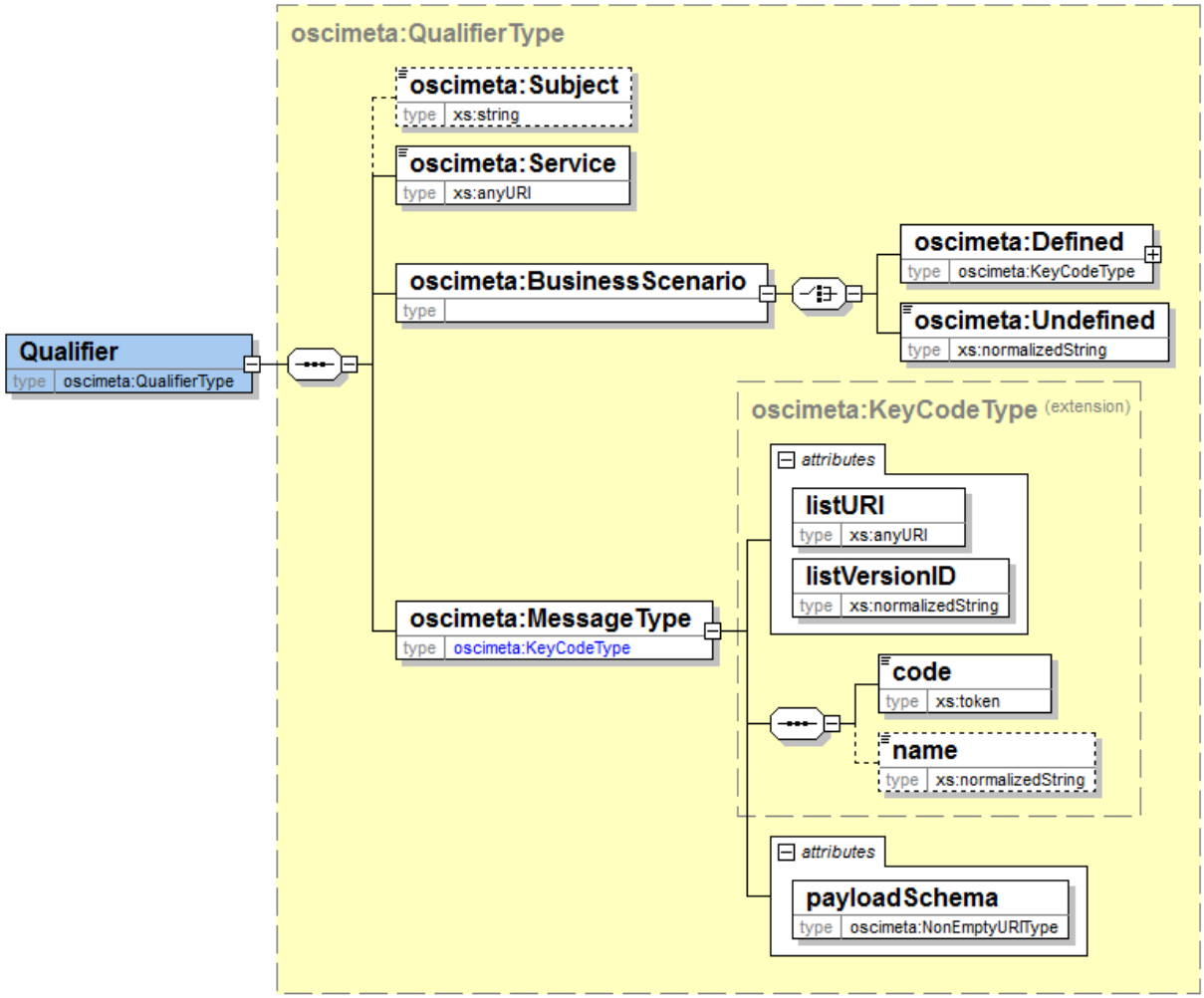
5.4.2.3.3.1 MsgIdentification



Dieses Objekt enthält die Identifikation des Transportauftrags und Bezüge auf Vorgänge und Fachnachrichten:

- **MessageID:** Obligatorischer eindeutiger Identifikator des Transportauftrags, vom Sender im Auftrag des Autors vergeben.
- **In-Reply-To:** bezogene Nachricht(en) (auf Applikations-/ Fachebene)
- **ProcessRef:** Bezug zu laufendem Vorgang (etwa durch Aktenzeichen). Hier kann die ID einer XÖV-Nachricht eingetragen werden (vgl. das Element *nachrichtenUUID* aus dem Typ *Identifikation.Nachricht* des Standards *XInneres*). Wie das Feld zu verwenden ist, wird durch das entsprechende Service Profil festgelegt.
  - Es ist eine Unterscheidung zwischen Vorgangsbezügen auf der Seite des Autors (Element *Requester*) und Lesers (Element *Responder*) möglich.
  - Die Vorgangsnummer kann mit einem Vorgangsnamen, Element *ProcessName*, attributiert werden.
  - **Zuordung von Fachnachricht und Transportauftrag**:
    - Die ID der Fachnachricht ist in Element */MsgIdenitification/ProcessRef/Requester* einzutragen. Damit ist die zu transportierende bzw. transportierte Fachnachricht ihrem Transportauftrag zugeordnet (sie kann darüberhinaus weiteren Transportaufträgen zugeordnet sein).
    - Damit ist diese ID auch Bestandteil des Protokolls (TransportReport). Zu den Protokollen haben Autor und Leser Zugang (Operation getTransportReport). So können sie für jede Fachnachricht-ID die zugeordneten Transportaufträge ermitteln.

5.4.2.3.3.2 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.

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 2 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) unter Nennung ihrer Codelisten-URI *urn:de:xta:codeliste:business.scenario* 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 Reportformate benennen) und an dieser Stelle einbinden. In die Attribute des vorliegenden Typs werden dann Codelisten-URI und Versionsnummer der selbstdefinierten Codeliste eingetragen.

Umsetzungshinweise:

- XInneres-Nachrichten: Wenn sich im Payload eine XInneres-Nachricht befindet, ist im vorliegenden Element nicht „XInneres“, sondern der Fachkontext einzutragen, innerhalb dessen sie gesendet wird. Im Kontext der XInneres-Weiterleitung einer XPersonenstand-Nachricht wäre hier beispielsweise der Code für den Fachkontext XPersonenstand einzutragen.
- Test-Nachrichten: Eine Kennzeichnung als Test-Nachricht geschieht über das Attribut @TestMsg (Element *MessageMetaData*)

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 Geschäftsszenarien (z.B. innerhalb des XÖV-Standards) definiert und sind im XRepository (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:

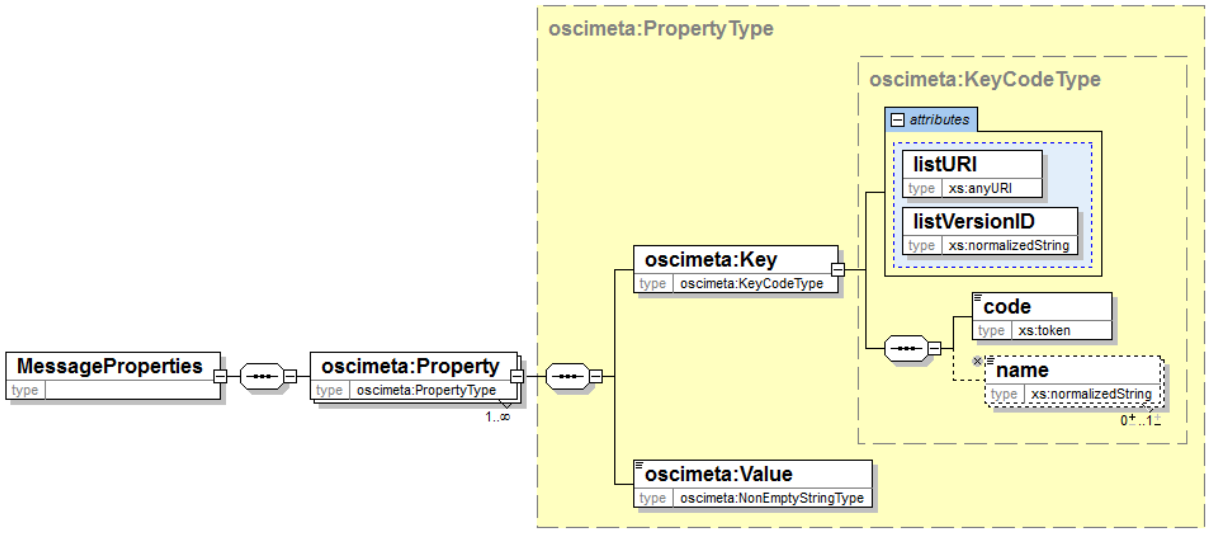
- OSCI-XMeld (ab Version 2.0): *urn:de:xmeld:schluesseltabelle:xmeld.nachrichten*
- XAusländer: *urn:de:xauslaender:codelist:nachrichtentyp*
- XPersonenstand: *urn:xpersonenstand:schluesseltabelle:nachrichtentyp*

Für das Geschäftsszenario XhD wurde durch XTA 2 eine Codeliste definiert und veröffentlicht, welche mittelfristig von einer durch XhD herausgegebene Codeliste abgelöst werden soll:

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

5.4.2.3.3.3 MessageProperties

Dieses Element nimmt bei Bedarf Metainformation auf, die bezogen auf ein spezielles Szenario benötigt wird. Semantik und Verarbeitung dieser Informationen erfolgen fachbezogen.



MessageProperties

- Dieser Container nimmt **Property** -Elemente auf, die innerhalb eines definierten Geschäftsszenarios festzulegen sind (Name, Werte, Semantik).
- Eine Property ist ein Key/Value-Pärchen innerhalb eines Namensraums. Für Kommunikationsszenarien innerhalb eines definierten Geschäftsszenarios werden bei Bedarf entsprechende Codelisten festgelegt, um eine identische semantische Interpretation des Elementes *Key* sicherzustellen. Im Element *Value* steht dann der zu übertragende Wert.
- Dieses Objekt des MessageMetaData-Header kann ggf. durch Messaging-Knoten bei Bedarf fortgeschrieben werden.

5.4.3 Schnittstellentyp msgBoxPort

Für den asynchronen Nachrichtenaustausch holt der Leser, der in der Regel meist nicht online erreichbar ist, die an ihn adressierten Nachrichten aus einem „Postkasten“ (auch „Message-Box“ genannt) ab. Dort sind die Nachrichten zwischengespeichert. (Der Leser nutzt hierfür einen Pull-Mechanismus.)

Im Rahmen des Abholens der Nachrichten werden dem Leser mehrere Funktionen angeboten: Der Leser ruft direkt alle Nachrichten oder aber einzelne Nachrichten aus dem „Postkasten“ ab, oder er läßt sich Statuslisten geben, aus denen ablesbar ist, wieviele Nachrichten noch nicht abgeholt wurden.

Diese Angebote werden durch einen Service von OSCI 2 spezifiziert, der in XTA-WS genutzt wird: Im Schnittstellentyp msgBoxPort werden alle „Postkasten-Funktionen“ für den XTA-WS zusammengefasst:

- getStatusList (siehe Abschnitt 5.4.3.1 auf Seite 133)
- getMessage (siehe Abschnitt 5.4.3.2 auf Seite 135)
- close (siehe Abschnitt 5.4.3.3 auf Seite 136)

5.4.3.1 Methode getStatusList (Abruf einer Liste bzw. Teilliste von Metadaten und MessageIDs)

Mit der Methode getStatusList kann der Leser vom Empfänger Informationen (MessageID und Metadaten des Transportauftrags) über die eingegangenen Nachrichten abrufen bzw. prüfen, ob Nachrichten bereitstehen.

Um die Ergebnisliste einzuschränken, kann der Leser dem Methodenaufruf Selektionskriterien (Status, Zeitraum) mitgeben. Die Rückgabeliste enthält pro Nachricht, die den Auswahlkriterien entspricht, Metainformationen, z.B. Autor, Leser, Subjekt, MessageID des zugehörigen Transportauftrags.

Mithilfe der Ergebnisliste entscheidet der Leser, welche Nachrichten er tatsächlich abholen möchte. Für die Abholung wird die Methode getMessage (siehe Abschnitt 5.4.3.2 auf Seite 135) verwendet.

5.4.3.1.1 Ergebnisse getStatusList

- 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.
- 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.

5.4.3.1.2 Operation getStatusList

Input.

Soap Part	Name	Type
Header	AuthorIdentifier (vgl. Seite 117	oscimeta:PartyType (vgl. Seite 117)
Body	MsgBoxStatusListRequest	osci:MsgBoxStatusListRequestType

Wesentliche Parameter:

- "oscimeta:PartyType"(vgl. Seite 106): AuthorIdentifier ist die fachliche Identität des Autoren.

  „osci:MsgBoxStatusListRequestType“ (vgl. Seite 137): Im Attribut ListForm (enumeration) ist im XTA-Kontext immer die Option "MessageMetaData" auszuwählen.
- „osci:MsgBoxEntryTimeFrom“: Wenn ein Zeitpunkt angegeben ist, werden nur die MessageIDs und Metadaten von Nachrichten geliefert, die nach diesem Zeitpunkt empfangen wurden.
- „osci:MsgBoxEntryTimeTo“: Wenn ein Zeitpunkt angegeben ist, werden nur die MessageIDs und Metadaten von Nachrichten geliefert, die bis zu diesem Zeitpunkt empfangen wurden.
- „osci:newEntry“: Festlegung, ob alle Nachrichten oder nur neue Nachricht berücksichtigt werden sollen.
- „osci:maxListItems“: Maximale Anzahl von Einträgen, die je Zugriff - also auch bei den nachfolgenden Aufrufen der Methode getNextStatusList() - zurückgegeben werden soll.
- „wsa:MessageID“: Angabe der MessageID des Transportauftrags der Nachricht, über die Informationen abgerufen werden soll bzw. für die zu prüfen ist, ob Nachrichten bereitstehen.
- „wsa:RelatesTo“: Eine Referenz auf eine Liste andere Nachrichten, die mit dieser Nachricht in einem Zusammenhang stehen.

Output.

Soap Part	Name	Type
Header	MsgBoxResponse	osci:MsgBoxResponseType
Body	MsgStatusList	osci:MsgStatusListType

Rückgabewerte:

- "osci:MsgBoxRequestID“: Die Ressourcenkennung für die weiteren Zugriffe auf den Iterator.
- „osci:NoMessageAvailable“: Angabe des Grundes, wenn zu den angegebenen Suchargumenten keine Daten gefunden wurden.
- „osci:ItemsPending“: Anzahl der gefundenen Nachrichten.

- "osci:MsgStatusListType“: Hier muss das im Input durch das Attribut ListForm angeforderte Element "oscimeta:MessageMetaData" geliefert werden.

5.4.3.1.3 Beispielcode (Aufruf der Methode)

```
getStatusList(„2013-03-01T02:00:00“, „2013-03-01T12:00:00“, true, 5)
```

(Angabe des zu berücksichtigenden Zeitraums, Berücksichtigung aller (- also nicht nur neuer) Nachrichten, Abholung von max. 5 Nachrichten)

5.4.3.2 Methode getMessage (Abholen einer Nachricht)

Selektionskriterium MessageID:

Mit der Methode getMessage holt der Leser eine Nachricht vom Empfänger ab. In der aktuellen Version des XTA-WS wird ausschließlich das Selektionskriterium MessageID unterstützt.

Die Identifikation der Nachricht erfolgt also durch die MessageID des zugehörigen Transportauftrages, die als Ergebnis der Methode <getStatusList> zurückgegeben wurde.

Wenn der Leser gezielt eine MessageID benennt, erhält er die entsprechende Nachricht. Bei dieser gezielten Abholung ist nicht relevant, ob die Nachricht bereits früher abgeholt wurde.

Der Empfang abgeholter Nachrichten muss vom Leser gegenüber dem Empfänger (mit der Methode close und mindestens der MessageID aus dem MessageMetaData-Container) quittiert werden.

Nutzung weiterer Selektionskriterien:

Perspektivisch hat der Leser neben der MessageID weitere Selektionskriterien zur Auswahl. Für die Methode getMessage sind ähnliche Parameter vorgesehen wie sie für die Methode getStatusList beschrieben sind, siehe außerdem Asynchroner Empfang, Variante II, die in Abschnitt 5.4.3.5.1, „Sukzessiver Abruf von Nachrichten aus dem Postfach“ dokumentiert wird und nicht im vorliegenden XTA-WS umgesetzt ist: Der Leser kann milthilfe dieser Variante dann Nachrichten vom Empfänger unter Angabe von diesen Kriterien abholen. Hierfür kann der Leser einen **Iterator** verwenden. Mit diesem verwaltet der Empfänger für die Dauer der Abholung die Liste der noch abzuholenden Nachrichten. Damit muss diese Arbeit nicht vom Leser übernommen werden. Das ist vor allem vorteilhaft, wenn mehrere Leser gleichzeitig Nachrichten abholen wollen: Der Status der Nachricht wird beim Empfänger verwaltet. Der Empfänger liefert beim Erzeugen des Iterators die Ressourcenkennung des Iterators zusammen mit der ersten Nachricht, die den Selektionskriterien entspricht, zurück.

5.4.3.2.1 Ergebnisse

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 <close> bestätigt worden ist.

Im Fehlerfall wird anstelle der Nachricht eine der folgenden Fehlermeldungen zurückgeliefert:

- 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.
- Der technische Fehler (SoapFault) <InvalidMessageIDException> entsteht, wenn die angeforderte Nachricht dem Account nicht bekannt ist.

5.4.3.2.2 Operation getMessage

Input.

Soap Part	Name	Type
Header	AuthorIdentifier	oscimeta:PartyType (vgl. Seite 117)
Body	MsgBoxFetchRequest	osci:MsgBoxRequestType

Wesentliche Parameter:

- „oscimeta:PartyType“ (vgl. Seite 106): Information zur Authentisierung des Lesers
- „osci:MsgBoxRequestType“(vgl. Seite 137): MsgBoxFetchRequest enthält Selektionskriterien mit denen eingegrenzt werden kann, welche bisher nicht abgeholten Nachrichten mit dieser Anfrage abgeholt werden sollen.
- „wsa:MessageID“: Angabe der MessageID des Transportauftrags der abzuholenden Nachricht

Output.

Soap Part	Name	Type
Header	MessageMetaData	osci:MessageMetaData(vgl. Seite 122)
Header	MsgBoxResponse	osci:MsgBoxResponseType
Body	GenericContentContainer	xta:GenericContentContainer (vgl. Seite 154)

- „oscimeta:MessageMetaData“: In dieser Struktur werden die Metadaten des Transportauftrags definiert. Das Objekt ist als mandatorischer Parameter zu verwenden. Da die Daten dann als SOAP-Header zur Verfügung stehen, muss für diverse Zwecke nur dieser Header, aber nicht eine eingebettete Fachnachricht gelesen werden. Die Metadaten beinhalten Zeitstempel, Quittungsanforderungen, Service Profil, Angaben über den Autoren und den Leser, Informationen zur Identifikation der Fachnachricht und weitere Informationen.
- „osci:MsgBoxResponse“: Metainformationen zur Anfrage (Request)
- „xta:GenericContentContainer“: Die zurückzuliefernde Nachricht

5.4.3.2.3 Beispielcode (Aufruf der Methode)

Abholung einer Nachricht durch Angabe der MessageID des zugehörigen Transportauftrages:

```
getMessage(“urn:de:xta:messageid:clearingstelleXY_xta_01:000ca2fe-
f4e1-45c2-8233-3a0eb760bd16“)
```

5.4.3.3 Methode close (Quittierung der Abholung)

Mithilfe der Methode close soll sichergestellt werden, dass Nachrichten oder Listen nicht mehrfach verarbeitet werden: Der Leser bestätigt dem Empfänger durch eine entsprechende Quittierung, dass Nachrichten und (Teil-)listen erfolgreich abgeholt werden konnten. Diese Empfangsquittierung soll möglichst zeitnah erfolgen, so dass gewährleistet wird, dass jede vom Leser verarbeitete Nachricht beim Empfänger als gelesen markiert wurde. Der Verlust von Nachrichten und Listen kann erkannt werden, wenn die erwarteten Quittierungen fehlen.

Mit der Methode close wird damit eine Ressource bei Sender oder Empfänger wieder freigegeben. Dies kann ein Iterator sein, der beim Abruf von Teillisten benötigt wurde. Sie beendet also die Transaktion nach Erhalt einer Ergebnisliste, die durch die Methode getStatusList erzeugt wurde und sie bestätigt die Abholung von Nachrichten.

5.4.3.3.1 Ergebnisse

Im Erfolgsfall wird kein Ergebniswert zurückgeliefert, dadurch wird signalisiert, dass die ID des Requests wieder freigegeben wird und die übergebenen Nachrichten zugestellt wurden.

Im Fehlerfall wird anstelle der Nachricht eine der folgenden Fehlermeldungen zurückgeliefert:

- Der technische Fehler (SoapFault) <PermissionDeniedException> entsteht, wenn ein nicht autorisierter Zugriff auf den XTA-WS stattfindet.
- Der technische Fehler (SoapFault) <InvalidMessageIDException> entsteht, wenn die angeforderte Nachricht dem Account nicht bekannt ist.
- Der technische Fehler (SoapFault) <XTAWSTechnicalProblemException> entsteht, wenn ein technischer Fehler im XTA-WS aufgetreten ist.

5.4.3.3.2 Operation close

Input.

Soap Part	Name	Type
Header	AuthorIdentifier	oscimeta:PartyType (vgl. Seite 117)
Body	MsgBoxCloseRequest	osci:MsgBoxCloseRequestType

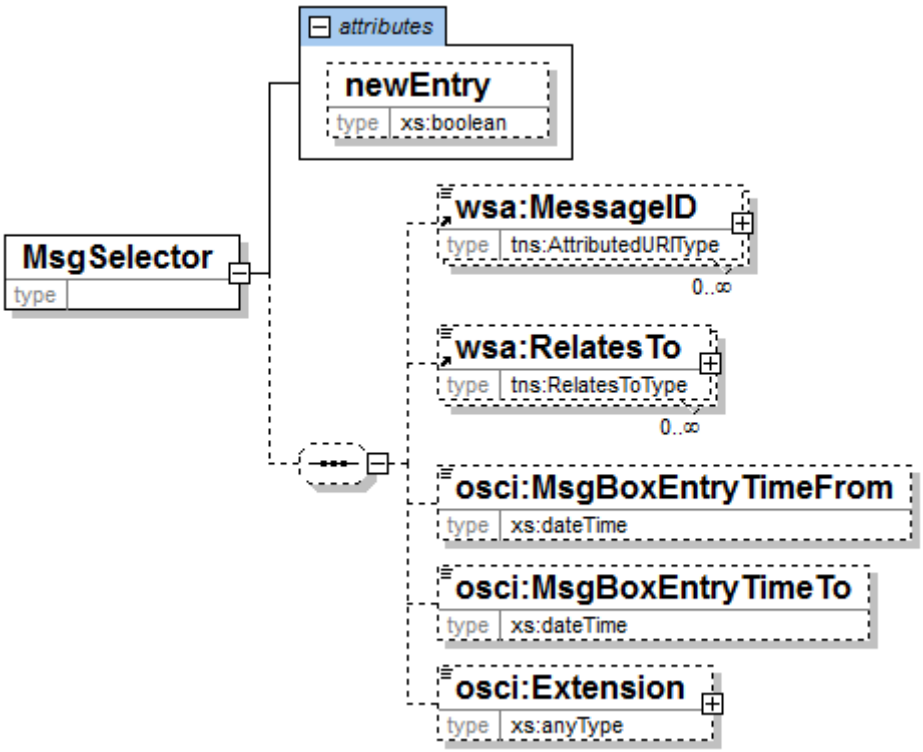
Wesentliche Parameter:

- „oscimeta:PartyType“ (vgl. Seite 106): Information zur Authentisierung des Lesers
- „osci:MsgBoxCloseRequestType“: Angabe der Ressourcenkennung, die durch diesen Methodenauruf freigegeben werden soll. Zusätzlich wird eine Liste der zur Bestätigung ausstehenden Nachrichten angehängt.

Output. Keine Rückgabewerte.

5.4.3.4 Wichtige Objekte der OSCI-MsgBox-Schnittstelle

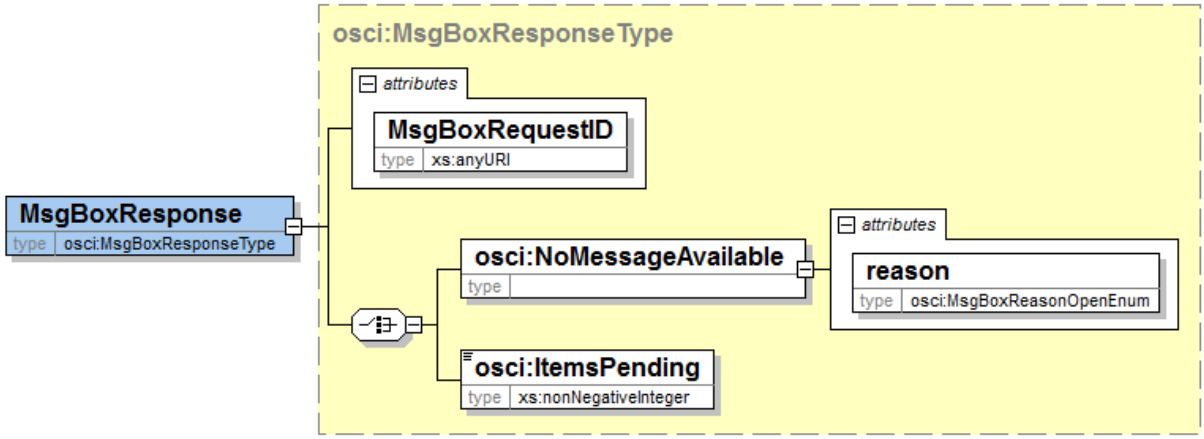
5.4.3.4.1 MsgSelector



MsgSelector enthält die Selektionsmöglichkeiten der Pull-Mechanismen auf die OSCI Message-Box.

- „osci:newEntry“: Festlegung, ob alle Nachrichten oder nur neue Nachricht berücksichtigt werden sollen.
- „wsa:MessageID“: Angabe der MessageID des Transportauftrags der abzuholenden Nachricht.
- wsa:RelatesTo: Referenzen auf Objekte, z. B. Vorgängernachrichten.
- „osci:MsgBoxEntryTimeFrom“: Wenn ein Zeitpunkt angegeben ist, werden nur die MessageIDs und Metadaten von Nachrichten geliefert, die nach diesem Zeitpunkt empfangen wurden.
- „osci:MsgBoxEntryTimeTo“: Wenn ein Zeitpunkt angegeben ist, werden nur die MessageIDs und Metadaten von Nachrichten geliefert, die bis zu diesem Zeitpunkt empfangen wurden.
- osci:Extension: Dieser Parameter wird nicht genutzt, da er sich in OSCI noch in der Entwicklung befindet.

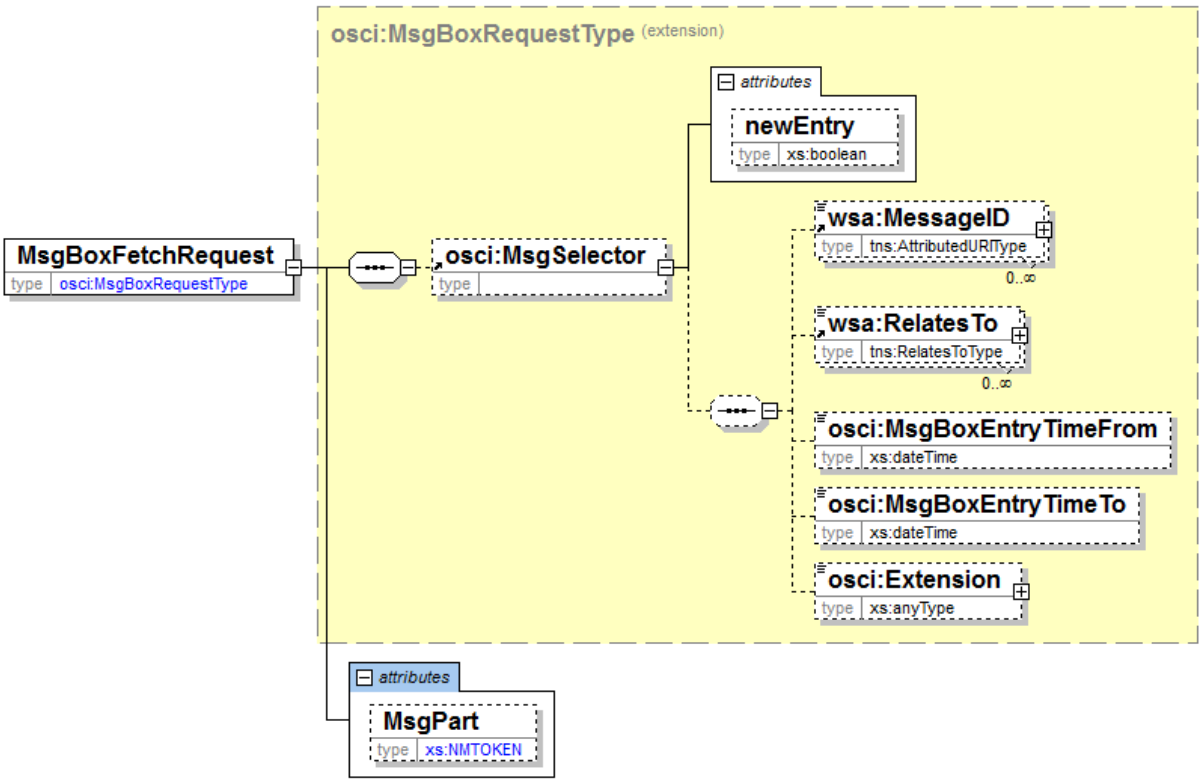
5.4.3.4.2 MsgBoxResponse



MsgBoxResponse enthält die Metainformationen zur Anfrage (Request).

- „MsgBoxRequestID“: Angabe der Ressourcenkennung
- „osci:ItemsPending“: Anzahl der abrufbaren Nachrichten. Dieser Parameter kann nicht verwendet werden, wenn „osci:NoMessageAvailable“ verwendet wird.
- „osci:NoMessageAvailable: Information und Grund über Nichtverfügbarkeit von Nachrichten. Dieser Parameter kann nicht verwendet werden, wenn „osci:ItemsPending“ verwendet wird.

5.4.3.4.3 MsgBoxFetchRequest



MsgBoxFetchRequest wird verwendet, um eine Nachricht anzufordern und enthält die erforderlichen Selektionskriterien in MsgSelector sowie das Attribut **@MsgPart**. MsgSelector wurde bereits zuvor beschrieben, mit dem optionalen Attribut @MsgPart vom Typ **xs:NMTOKEN** kann angegeben werden, welcher Teil einer Nachricht zurückgegeben werden soll:

- **Envelope:** Stellt alle Informationen (Header- und Body-Blöcke der selektieren Nachrichten) als Kindelement der MsgBoxResponse bereit.
- **Body:** Stellt nur den originalen SOAP-Body der selektierten Nachricht in unveränderter Form als Kindelement der MsgBoxResponse bereit. Der Empfehlung von OSCI folgend, dass ein SOAP Body nur ein Kindelement enthält, sollte dieser Wert nur verwendet werden, wenn genau eine Nachricht selektiert wird.
- **Header:** Stellt nur die Header-Blöcke der selektieren Nachrichten im Body der MsgBoxResponse bereit.

Wird das optionale Attribut @MsgPart nicht angegeben, ist „Body“ der Default-Wert.

5.4.3.5 Optionaler Teil des Schnittstellentyps msgBoxPort

5.4.3.5.1 Sukzessiver Abruf von Nachrichten aus dem Postfach

Dieser Abschnitt ist ein optionaler Teil der XTA-Spezifikation und muss nicht verpflichtend umgesetzt werden.

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 5.4.3.2 auf Seite 135)
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 5.4.1.3 auf Seite 114)
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 5.4.3.5.1.1.1 auf Seite 140)
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 5.4.3.3 auf Seite 136)

5.4.3.5.1.1 Methoden

Zur Umsetzung des parallelen Zugriffs auf ein Postfach werden zusätzlich folgende Methoden benötigt:

- getNextMessage
- getNextStatusList

5.4.3.5.1.1.1 Methode getNextMessage (Abholen einer nächsten Nachricht)

Hat der Leser zuvor eine Abfrage mittels der Methode <getMessage> durchgeführt und noch nicht alle Nachrichten abgeholt, kann der Leser die noch nicht abgeholten Nachrichten nachfolgend mit der Metho-

de <getNextMessage> abfragen. Die Abfrage mittels der Methode <getNextMessage> muss an denselben XTA-Server gerichtet werden, an die auch die vorausgegangene Abfrage mittels der Methode <getMessage> gerichtet wurde.

5.4.3.5.1.1.1.1 Ergebnisse

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.

Im Fehlerfall wird eine der folgenden Fehlernachrichten zurückgeliefert:

- 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.
- Der technische Fehler (SoapFault) <InvalidMessageIDException> entsteht, wenn die anhand der ID angeforderte Nachricht dem Account nicht bekannt ist.

5.4.3.5.1.1.1.2 Operation getNextMessage

Input.

Soap Part	Name	Type
Header	AuthorIdentifier	oscimeta:PartyType (vgl. Seite 117)
Body	MsgBoxGetNextRequest	osci:MsgBoxGetNextRequestType

Wesentliche Parameter:

- „oscimeta:PartyType (vgl. Seite 106)“: AuthorIdentifier ist die fachliche Identität des Autors. Dieser Parameter ist optional, wenn das Zertifikat genau einen Autor/Leser identifiziert, sonst ist er mandatorisch zu übergeben.
- „osci:MsgBoxGetNextRequestType“ (vgl. Seite 137): MsgBoxGetNextRequest enthält einen Verweis auf die initiale Anfrage, deren Selektionskriterien verwendet werden, um bisher nicht abgeholte Nachrichten mit dieser Anfrage abzurufen.
- Das Attribut **@MsgBoxRequestID** vom Typ xs:anyURI enthält die Ressourcenkennung für einen „Iterator“. Der Leser bekommt die Ressourcenkennung mit der ersten Nachricht und benötigt sie für die weitere Abholung von Nachrichten.

Output.

Soap Part	Name	Type
Header	MessageMetaData	oscimeta:MessageMetaData (vgl. Seite 122)
Header	MsgBoxResponse	osci:MsgBoxResponseType
Body	GenericContentContainer	xta:GenericContentContainer

Rückgabewerte:

- „oscimeta:MessageMetaData“: In dieser Struktur werden die Metadaten des Transportauftrags definiert. Das Objekt ist als mandatorischer Parameter zu verwenden. Da die Daten dann als SOAP-Header zur Verfügung stehen, muss für diverse Zwecke nur dieser Header, aber nicht eine eingebet-

tete Fachnachricht gelesen werden. Die Metadaten beinhalten Zeitstempel, Quittungsanforderungen, Service Profil, Angaben über den Autoren und den Leser, Informationen zur Identifikation der Fachnachricht und weitere Informationen.

- „osci:MsgBoxResponseType“: Metainformationen zur Anfrage (Request)
- „xta:GenericContentContainer“: Dieses Objekt enthält die nächste Nachricht. Sie besteht aus der eigentlichen Nachricht und einer beliebigen Anzahl von Anhängen (Attachments). Die Nachricht selber kann in einem verschlüsselten Container hinterlegt werden. Zu der Nachricht kann ein Betreff (Subject) angegeben werden.

5.4.3.5.1.1.2 Methode getNextStatusList (Nächste Teilliste von MessageIDs und Metadaten holen)

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 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 5.4.3.1 auf Seite 133) erhalten hat:

5.4.3.5.1.1.2.1 Ergebnisse

Im Erfolgsfall liefert die Methode <getNextStatusList> eine Nachricht zurück, welche die Metainformationen zu den noch ausstehenden Nachrichten einer vorausgegangenen Anfrage mittels der Methode <getStatusList> enthält.

Im Fehlerfall wird eine der folgenden Fehlernachrichten zurückgeliefert:

- 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.
- Der technische Fehler (SoapFault) <InvalidMessageIDException> entsteht, wenn die anhand der ID angeforderte Nachricht dem Account nicht bekannt ist.

5.4.3.5.1.1.2.2 Operation getNextStatusList

Input.

Soap Part	Name	Type
Header	AuthorIdentifier	oscimeta:PartyType (vgl. Seite 117)
Body	MsgBoxGetNextRequest	osci:MsgBoxGetNextRequestType

Wesentliche Parameter:

- „oscimeta:PartyType (vgl. Seite 106)“: AuthorIdentifier ist die fachliche Identität des Autors. Dieser Parameter ist optional, wenn das Zertifikat genau einen Autor/Leser identifiziert, sonst ist er mandatorisch zu übergeben.
- „osci:MsgBoxGetNextRequestType“ (vgl. Seite 137): MsgBoxGetNextRequest enthält einen Verweis auf die initiale Anfrage, deren Selektionskriterien verwendet werden, um bisher nicht abgeholte Nachrichten mit dieser Anfrage abzurufen.

- Das Attribut **@MsgBoxRequestID** vom Typ xs:anyURI enthält die Ressourcenkennung für einen „Iterator“. Der Leser bekommt die Ressourcenkennung mit demersten Teil einer Nachrichtenstatusliste und benötigt sie für die Abholung weiterer Teile.

Output.

Soap Part	Name	Type
Header	MsgBoxResponse	osci:MsgBoxResponseType
Body	MsgStatusList	osci:MsgStatusListType

Rückgabewerte:

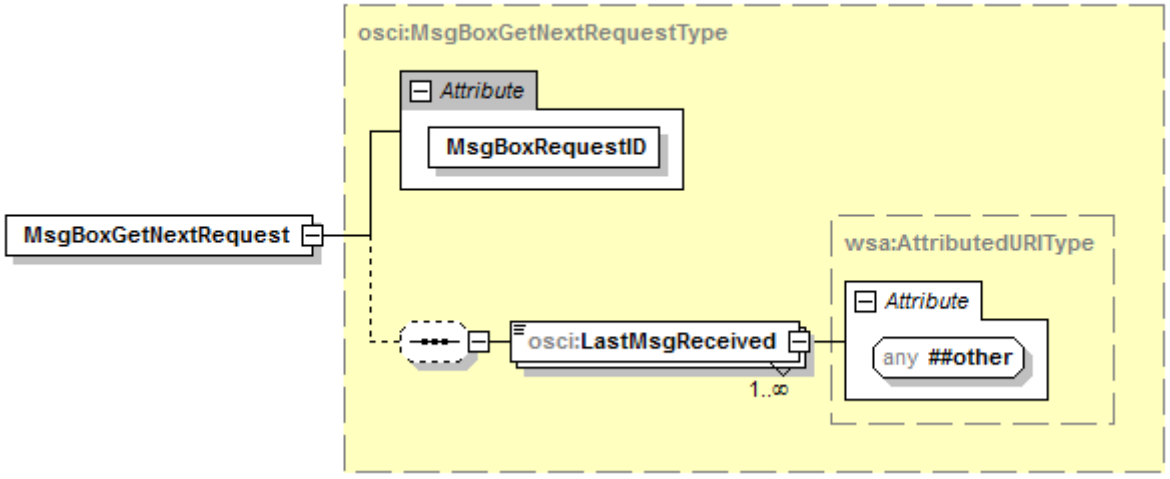
- „osci:MsgBoxResponseType“: Metainformationen zur Anfrage (Request)
- „osci:NoMessageAvailable“: Wurden zu den angegebenen Suchargumenten keine Daten gefunden, wird die Ursache angegeben.
- „osci:ItemsPending“: Anzahl der Nachrichten, auf die die Anfrage zutrifft.
- „osci:MsgStatusListType“: Hier muss die im Input der vorangegangenen Anfrage mittels <getStatusList> im Attribut ListForm angeforderte Liste von MessageMetaData-Objekten geliefert werden.

5.4.3.5.1.1.2.3 Beispielcode (Aufruf der Methode)

```
getNextStatusList(„osci:8dfrg8e7o485zfiuz84r7349“)
```

5.4.3.5.1.1.3 Wichtige Objekte der OSCI-MsgBox-Schnittstelle

5.4.3.5.1.1.3.1 MsgBoxGetNextRequest



MsgBoxGetNextRequest wird verwendet, um noch nicht abgeholte Nachrichten mittels <getNextMessage> oder <getNextStatusList> abzuholen. Es kann nur verwendet werden, wenn zuvor eine Anfrage mittels <getMessage> oder <getStatusList> erfolgte. Die Anfrage muss an denselben XTA-Server gerichtet werden, an die auch die vorangegangenen Anfragen gerichtet wurden.

„osci:LastMsgReceived“: Falls die vorangegangene Anfrage mittels <getMessage> durchgeführt wurde, können diese optionalen Elemente angegeben werden. Werden diese Elemente angegeben, sollten sie den Wert bzw. die Werte der `wsa:MessageID` der letzten, erhaltenen Nachricht enthalten und so den erfolgreichen Empfang dieser Nachricht anzeigen.

Das Attribut @MsgBoxRequestID vom Typ xs:anyURI enthält die Ressourcenkennung für einen „Iterator“. Der Leser bekommt die Ressourcenkennung mit der ersten Nachricht und benötigt sie für die weitere Abholung von Nachrichten. Sie muss dazu verwendet werden, die aktuelle Anfrage mit der vorangegangenen Anfrage in Verbindung zu bringen. Die Ressourcenkennung kann an andere Leser weitergereicht werden, falls mehrere Leser auf die Nachrichten zugreifen wollen.

Hinweis: In der Antwortnachricht muss das Header-Element osci:ItemsPending die Anzahl der noch nicht abgeholten Nachrichten enthalten. Wird die letzte, noch nicht abgeholte Nachricht oder der letzte, noch nicht abgeholte Teil einer Nachrichtenstatusliste zurückgesendet, muss dieser Wert gleich Null sein.

5.4.4 Schnittstellentyp sendSynchronPort - Leser (Synchroner Versand einer Nachricht)

Die Methode sendMessageSync - Leser ist naturgemäß bezüglich der mitzugebenden Informationsblöcke, Struktur, Fehler und Beispielcode der Methode sendMessageSync - Sender sehr ähnlich und potentiell auch identisch zu ihr. Dies ist aber nicht zwangsläufig so, denn die Methode sendMessageSync-Leser kann auch als synchrone Teilstrecke in einem insgesamt asynchronen Kommunikationsszenario genutzt werden, daher wird die Methode sendMessageSync - Leser in diesem Abschnitt separat dokumentiert.

Mit der Methode sendMessageSync (- Leser) kann der Empfänger den Transportauftrag synchron an den Leser weiterleiten. Diese Methode wird also als Service vom Leser angeboten. Die Antwort des Lesers wird innerhalb derselben Transaktion an den Empfänger weitergereicht.

Damit der Leser die Methode implementieren kann, ist sie im Auslieferungsumfang XTA 2 in einer separaten WSDL enthalten (Datei XTA-synchron.wsdl).

Mit der Methode sendMessageSync kann der Autor über den Sender mit einem Leser kommunizieren: Der Autor schickt synchron eine Nachricht und bekommt (synchron) direkt eine Nachricht als Ergebnis zurück.

Dabei muss der Empfänger alle notwendigen Informationen mitgeben, denn er ist für den Transport verantwortlich. Folgende Informationsblöcke müssen mitgegeben werden:

- die Fachnachricht,
- die Beschreibung des Transportauftrags (Metadaten) mit der MessageID
- die Liste der zu prüfenden Zertifikate.

Diese Informationen werden in einem Aufruf an den XTA-WS übergeben. Der Empfänger wartet, bis der Leser die Antwort an ihn übergibt.

Mit der Methode sendMessageSync wird also eine Fachnachricht an den XTA-WS für einen synchronen Transport übergeben. Durch den Aufruf der Methode ist der Auftrag zum Transport erteilt.

5.4.4.1 Ergebnisse

Im Erfolgsfall wird eine Nachricht zusammen mit zugehörigen Metainformationen zurückgeliefert.

Im Fehlerfall wird anstelle der Nachricht eine der folgenden Fehlermeldungen zurückgeliefert:

- Der technische Fehler (SoapFault) <PermissionDeniedException> entsteht, wenn ein nicht autorisierter Zugriff auf den XTA-WS stattfindet.
- Der technische Fehler (SoapFault) <ParameterIsNotValidException> entsteht, wenn ein Pflichtübergabeparameter fehlt oder ein Übergabeparameter fehlerhaft ist. Dies tritt in folgenden Fällen auf:
  - Der Parameter <MessageID> ist innerhalb des XTA 2 nicht eindeutig.
  - Der Parameter <ServiceType> repräsentiert keine gültige Dienstebezeichnung oder der Dienst wird vom Empfänger nicht angeboten.

  - Der Parameter <AuthorIdentifier> ist nicht gemäß der jeweiligen fachlichen Spezifikation gefüllt.
- Der technische Fehler (SoapFault) <XTAWSTechnicalProblemException> entsteht, wenn
  - ein technischer Fehler im XTA-WS aufgetreten ist,
  - der Empfänger nicht erreichbar ist,
  - der Empfänger nicht innerhalb eines vom Sender festgelegten Time-Outs antwortet.
- 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 Fachnachricht 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.
- Der technische Fehler (SoapFault) <SyncAsyncException> entsteht, wenn eine Fachnachricht übergeben wurde, die nur für den asynchronen Versand gültig ist.

5.4.4.2 Operation sendMessageSync

Input.

Soap Part	Name	Type
Header	MessageMetaData	oscimeta:MessageMetaData (vgl. Seite 122)
Header	X509TokenContainer	osci:X509TokenContainer
Body	GenericContentContainer	xta:GenericContentContainer (vgl. Seite 154)

Wesentliche Parameter:

- „oscimeta:MessageMetaData“: In dieser Struktur werden die Metadaten des Transportauftrags definiert. Das Objekt ist als mandatorischer Parameter zu verwenden. Da die Daten dann als SOAP-Header zur Verfügung stehen, muss für diverse Zwecke nur dieser Header und nicht die ggf. eingebettete Fachnachricht gelesen werden. Die Metadaten beinhalten Zeitstempel, Quittungsanforderungen, das Service Profil, Angaben über den Autoren und den Leser, Informationen zur Identifikation der Fachnachricht und weitere Informationen.
- „osci:X509TokenContainer“: In diesem optionalen SOAP-Header können zu prüfende Zertifikate eingestellt werden. Die Prüfung kann auf dem Transportweg durchgeführt werden und ist eine optionale Serviceleistung der Transportinfrastruktur.
- „xta:GenericContentContainer“: 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.

Output.

Soap Part	Name	Type
Header	MessageMetaData	oscimeta:MessageMetaData (vgl. Seite 122)
Header	X509TokenContainer	osci:X509TokenContainer
Body	GenericContentContainer	xta: GenericContentContainer (vgl. Seite 154)

Rückgabewerte:

- „osci:X509TokenContainer“: In diesem optionalen SOAP-Header kann der Leser zu prüfende Zertifikate einstellen. Die Prüfung kann auf dem Transportweg durchgeführt werden und ist eine optionale Serviceleistung der Transportinfrastruktur.

- "oscimeta:MessageMetaData“: In dieser Struktur werden die Metadaten des Transportauftrags zu der Antwort des Lesers definiert. Das Objekt ist als mandatorischer Parameter zu übergeben. Da diese Daten dann als SOAP-Header mitgeführt werden, muss für diverse Zwecke nur dieser Header und nicht die eingebettete Fachnachricht gelesen werden. Die Metadaten beinhalten Zeitstempel, Quittungsanforderungen, das Service Profil, Angaben über den Autoren und den Leser, Informationen zur Identifikation der Fachnachricht und weitere Informationen.
- „xta:GenericContentContainer“: Dieses Objekt beinhaltet die Antwort des Lesers. Sie besteht aus einer Fachnachricht und einer beliebigen Anzahl von Anhängen (Attachments). Die Fachnachricht kann in einem verschlüsselten Container hinterlegt werden. Zu der Fachnachricht kann ein Betreff (Subject) angegeben werden.

5.4.4.3 Beispielcode (Aufruf der Methode)

```
sendMessageSync(ref oscimeta:MessageMetaData, ref osci:X509TokenContainer,
 ref xta:GenericContentContainer)
```

5.5 Das XTA-WS-Informationsmodell

In diesem Abschnitt sind die Informationsobjekte dokumentiert, die im Zusammenhang des XTA-Webservice definiert sind.

Weitere Informationsobjekte werden aus externen Standards eingebunden (vgl. Abschnitt 2.4 auf Seite 23). Sie sind nicht Bestandteil des XTA-Informationsmodells, werden also nicht im vorliegenden Abschnitt dokumentiert.

5.5.1 Datentypen der Informationsobjekte des XTA-Webservice

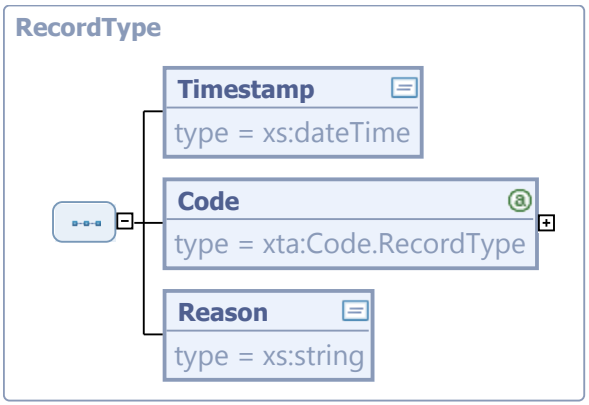
Hier werden die Bausteine beschrieben, aus denen sich die Informationsobjekte der Methodenaufrufe des XTA-WS zusammensetzen.

5.5.1.1 RecordType

Typ: *RecordType*

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

Abbildung 5.1. RecordType



Kindelemente von *RecordType*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
Timestamp	*xs:dateTime*	**1**		
Zeitstempel für den Zeitpunkt der Aufzeichnung der Meldung.				
Code	*xta:Code.RecordType*	**1**	5.5.1.2	147
Schlüssel, der die Bedeutung der Meldung kodiert. Dieser Schlüssel muss aus einer eingebundenen Codeliste stammen.				
Reason	*xs:string*	**1**		
Hier wird zur weiteren Erläuterung der Grund der Meldung als Freitext eingetragen.				

5.5.1.2 Code.RecordType

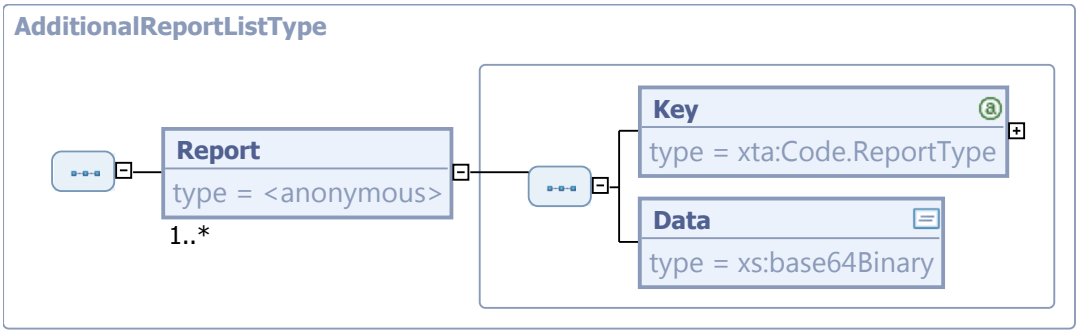
Code	**Code.RecordType**
Beschreibung	In diesen Typ ist eine auszuwählende bzw. selbst zu definierende Codeliste einzubinden, die Arten von Meldungen benennt, welche in das Protokoll zur Abarbeitung eines Transportauftrags (TransportReport) eingetragen werden. Dort können die Meldungen als Fehler-, Warn- 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 hat die Absicht, für den Standard XTA eine passende Codeliste zu definieren und als einheitliches Angebot zur Einbindung für diesen Typ bereitzustellen. Diese Codeliste ist, wenn die Bereitstellung erfolgt ist, im XRepository (www.xrepository.de) unter der Codelisten-URI *urn:de:xta:codeliste:record.type* auffindbar und kann von dort im XML-Format OASIS Genericode abgerufen werden.
Codelisten-Nutzung	Typ: 4, siehe Beschreibung
Codelisten-URI	unbestimmt
Codelisten-Version	unbestimmt

5.5.1.3 AdditionalReportListType

Typ: *AdditionalReportListType*

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

Abbildung 5.2. AdditionalReportListType



Kindelement von *AdditionalReportListType*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
Report		**1..n**		
In diesem Element ist ein zusätzlicher Report abgelegt, der das XTA-Protokoll (*TransportReport*) ergänzt. Die Art des Reports (z. B. „OSCI Process Card“) und der Inhalt des Reports werden bzw. sind in separaten Bereichen dieses Containers eingetragen.				
Key	*xta:Code.ReportType*	**1**	5.5.1.4	148
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.				
Data	*xs:base64Binary*	**1**		
Hier wird der zusätzliche Report in einem technisch neutralen Format eingetragen.				

5.5.1.4 Code.ReportType

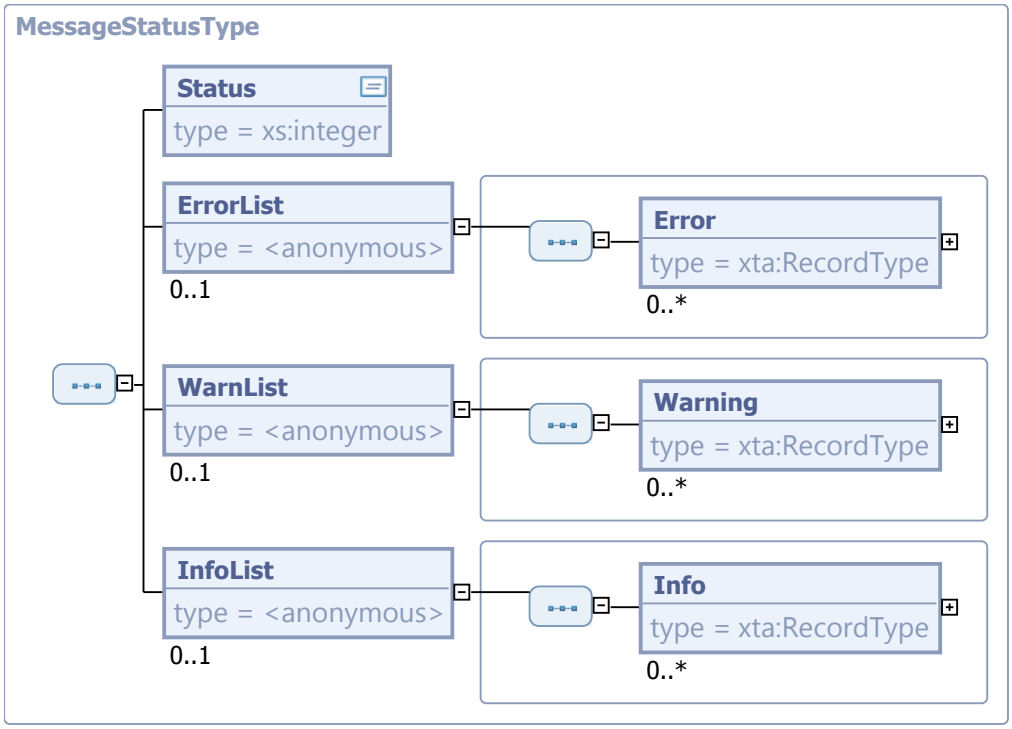
Code	Code.ReportType
Beschreibung	Dieser Typ gestattet die Kennzeichnung der Art eines zusätzlichen Reports. Es wird eine zu wählende Codeliste eingebunden, die mögliche Arten von Reports nennt (spezielles Format, innerhalb oder außerhalb von XTA definiert), die in das XTA-Protokoll (*TransportReport*) 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) unter Nennung ihrer Codelisten-URI *urn:de:xta:codeliste:report.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 Reportformate benennen) und an dieser Stelle einbinden. In die Attribute des vorliegenden Typs werden dann Codelisten-URI und Versionsnummer der selbstdefinierten Codeliste eingetragen.
Codelisten-Nutzung	Typ: 4, siehe Beschreibung
Codelisten-URI	unbestimmt
Codelisten-Version	unbestimmt

5.5.1.5 MessageStatusType

Typ: *MessageStatusType*

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

Abbildung 5.3. MessageStatusType



Kindelemente von *MessageStatusType*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
Status	*xs:integer*	**1**		
Wird durch Sender bzw. Empfänger fortgeschrieben. Wird der TransportReport noch fortgeschrieben, wird er hier mit 0=offen markiert. Nach Abschluss des TransportReports wird nach dem Max-Prinzip der höchste Ampelstatus aus den Elementen ErrorList, WarnList, InfoList hier numerisch dargestellt. • 0=offen: Die Nachricht befindet sich noch in der Verarbeitung. • 1=grün: Es sind keine Fehler oder Warnungen aufgetreten. • 2=gelb: Es sind Warnungen, aber keine kritischen Fehler aufgetreten. • 3=rot: Es sind kritische Fehler aufgetreten.				
ErrorList		**0..1**		
Liste der Fehlermeldungen.				
Error	*xta:RecordType*	**0..n**	5.5.1.1	146
Hier wird die Fehlermeldung mit ihren Parametern eingetragen.				
WarnList		**0..1**		
Liste der Warnungen.				
Warning	*xta:RecordType*	**0..n**	5.5.1.1	146
Hier wird die Warnung mit ihren Paramtern eingetragen.				
InfoList		**0..1**		
Liste der Informationsmeldungen.				
Info	*xta:RecordType*	**0..n**	5.5.1.1	146

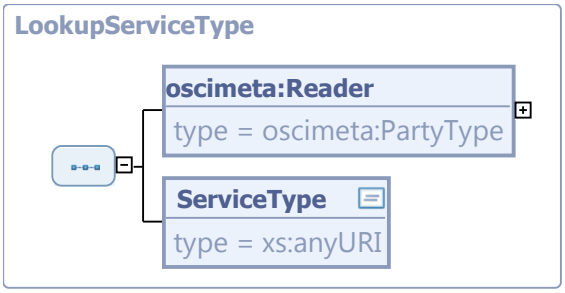
Kindelemente von *MessageStatusType*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
Hier wird die Informationsmeldung mit ihren Parametern eingetragen.				

5.5.1.6 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.

Abbildung 5.4. LookupServiceType



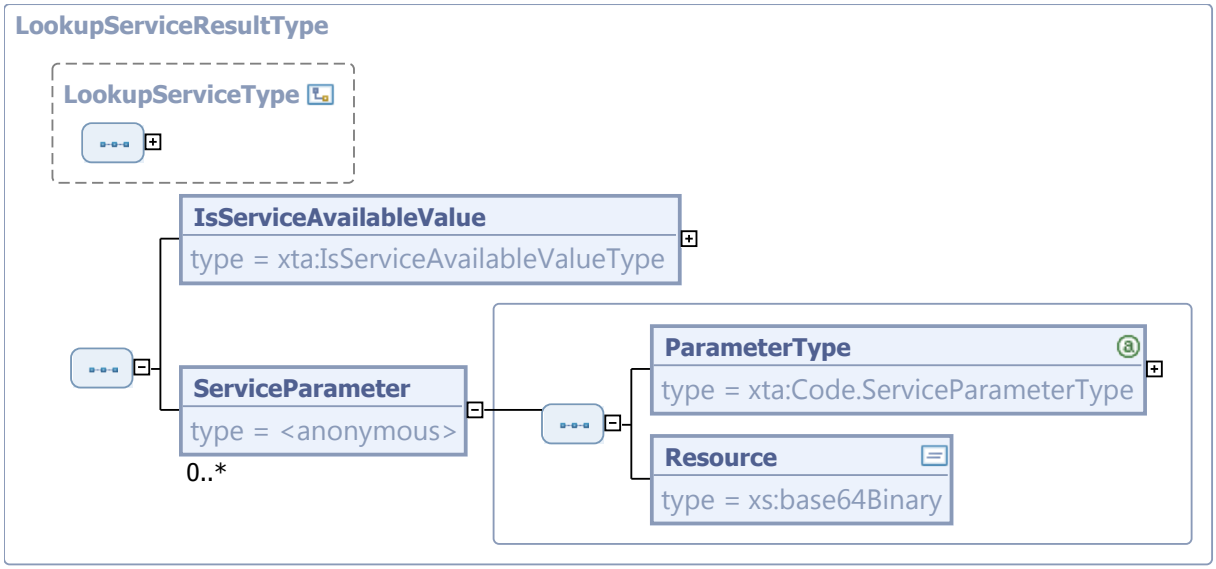
Kindelemente von *LookupServiceType*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
oscimeta:Reader	*globales Element*	**1**	2.4.2	24
Dies ist die fachliche Identifizierung des Lesers. Der Wert entspricht z.B. dem DVDV-Behördenschlüssel.				
ServiceType	*xs:anyURI*	**1**		
Dies ist die Bezeichnung des anzufordernden Dienstes. Sie wird im Format einer URL übergeben, was den Vorteil hat, dass damit auch eine Versionsnummer eingeschlossen ist. Beispiel für Dienstebezeichnungen, wie sie im DVDV verwendet werden: http://www.osci.de/xmeld181/xmeld181Rueckmeldung.wsdl Abgrenzung: "Dienst" ist das, was gemäß Diensteeinteilung der Fachdomäne im Verzeichnisdienst als Service (im Sinne eines Web Service) eingetragen ist. Dadurch ist die Dienstebezeichnung weniger differenziert als der Nachrichtentyp. Typischerweise sind im Verzeichnisdienst mehrere Nachrichtentypen in einer Service-WSDL zusammengefasst.				

5.5.1.7 LookupServiceResultType

Typ: *LookupServiceResultType*

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

Abbildung 5.5. LookupServiceResultType



Dieser Typ ist eine Erweiterung des Basistyps *LookupServiceType* (siehe Abschnitt 5.5.1.6 auf Seite 150).

Kindelemente von *LookupServiceResultType*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
IsServiceAvailableValue	*xta:IsServiceAvailableValueType*	**1**	5.5.1.8	151
Enthält das Ergebnis der Dienstanfrage: ob der Dienst angeboten wird oder nicht oder ob diese Information generell nicht bekannt ist.				
ServiceParameter		**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.				
ParameterType	*xta:Code.ServiceParameterType*	**1**	5.5.1.9	152
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 *listURI* und *listVersionID* referenziert ist.				
Resource	*xs:base64Binary*	**1**		
Hier ist der Parameter enthalten bzw. einnzutragen in technisch neutraler Darstellung.				

5.5.1.8 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.

Abbildung 5.6. IsServiceAvailableValueType

Kindelemente von *IsServiceAvailableValueType (Choice)*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
ServiceIsAvailable	*xs:boolean*	**1**		
Der Dienst wird angeboten (true) oder nicht angeboten (false).				
ServiceIsAvailableUnknown	*xs:boolean*	**1**		
Es ist nicht bekannt, ob der Dienst angeboten wird oder nicht.				

5.5.1.9 Code.ServiceParameterType

Code	**Code.ServiceParameterType**
Beschreibung	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) 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-Nutzung	Typ: 4, siehe Beschreibung
Codelisten-URI	unbestimmt
Codelisten-Version	unbestimmt

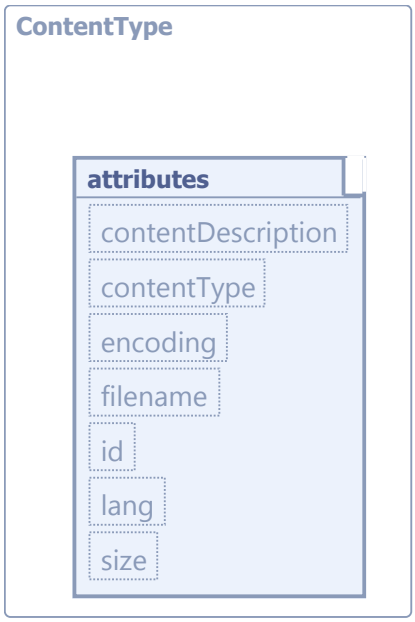
5.5.1.10 ContentType

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.

Abbildung 5.7. ContentType



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

Kindelemente von *ContentType*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
contentDescription	*oscimeta:NonEmptyStringType*	**0..1**	2.4.2	24
Beschreibung des fachlichen Inhalts, z.B. 'Angebot' oder 'Rechnung'.				
contentType	*oscimeta:NonEmptyStringType*	**1**	2.4.2	24
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).				
encoding	*oscimeta:NonEmptyStringType*	**0..1**	2.4.2	24
Der Zeichensatz, der der Kodierung des Inhalts zugrunde gelegen hat.				
filename	*oscimeta:NonEmptyStringType*	**0..1**	2.4.2	24
Der Dateiname der Datenquelle, falls der Inhalt einer Datei entnommen worden ist. Bsp.: Für die Übermittlung von xdomea-Nachrichten ist dieses Attribut Pflicht.				
id	*xs:ID*	**0..1**		
Bietet die Möglichkeit, den Inhalt über z.B. eine laufende Nummer zu referenzieren.				
lang	*xs:language*	**0..1**		
Sprache, in der der Inhalt formuliert ist.				
size	*xs:positiveInteger*	**0..1**		
Die Größe des Inhalts in Bytes.				

5.5.2 Globale Elemente der Informationsobjekte des XTA-Webservice

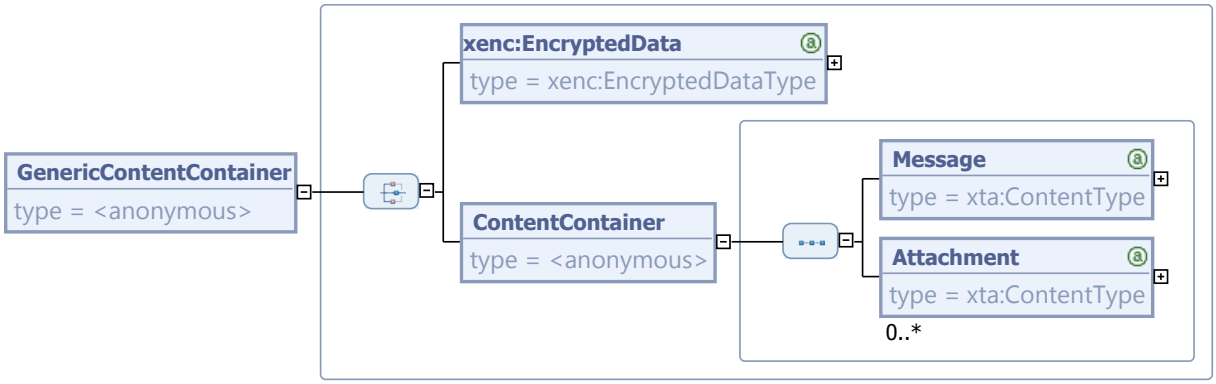
Die in diesem Abschnitt beschriebenen Objekte sind in den Methodenaufrufen des XTA-WS eingebunden. Sie sind zusammengesetzt teils aus in XTA definierten Bestandteilen wie beschrieben in Abschnitt 5.5.1 auf Seite 146, teils aus Typen der externen Standards, die aufgeführt sind in Abschnitt 2.4 auf Seite 23)

5.5.2.1 GenericContentContainer

Nachricht: *GenericContentContainer*

Der GenericContentContainer nimmt den zu transportierenden oder abzuliefernden Inhalt auf, z.B. eine XÖV-Nachricht mit ihren Anlagen. Diese Inhalte können unverschlüsselt (Element ContentContainer) oder auch verschlüsselt (Element xenc:EncryptedData) hinterlegt werden. Die Verschlüsselung an dieser Stelle eignet sich für Ende-zu-Ende-Verschlüsselung durch den Autor, wenn dieses Objekt durch den Autor erstellt wird.

Abbildung 5.8. GenericContentContainer



Kindelemente von *GenericContentContainer (Choice)*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
xenc:EncryptedData	*globales Element*	**1**	2.4.5	24
Dieses Objekt ist dafür vorgesehen, den Container-Inhalt verschlüsselt zu hinterlegen. Im entschlüsselten Zustand müssen die Daten dem Schwester-Element ContentContainer entsprechen.				
ContentContainer		**1**		
Der ContentContainer enthält genau eine Nachricht (Element *Message*) und null bis beliebig viele Anlagen, die alle in technisch neutraler Darstellung (base64-kodiert) eingefügt werden (Element *Attachment*). Die Gesamtgröße des Containers darf 40 MB nicht überschreiten.				
Message	*xta:ContentType*	**1**	5.5.1.10	152
Enthält den base64-kodierten Inhalt, der zwischen WebService-Client und XTA-Server transportiert wird. Die Attribute sind der MIME-Spezifikation (RFC 2183) entnommen. Die zu übermittelnde Nachricht als primärer Inhalt dieses Containers ist optional durch Anhänge (Element Attachment) zu ergänzen. In die Attribute wird je nach Kontext Metainformation zur Nachricht eingetragen.				
Attachment	*xta:ContentType*	**0..n**	5.5.1.10	152
Hier können optional ergänzende Anhänge zur übermittelnden Nachricht eingefügt werden.				

Kindelemente von *GenericContentContainer (Choice)*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
Die Attribute transportieren je nach Kontext Metainformation zum enthaltenen Anhang.				

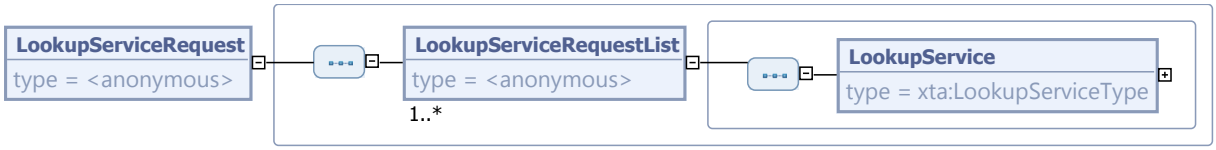
5.5.2.2 LookupServiceRequest

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.

Abbildung 5.9. LookupServiceRequest



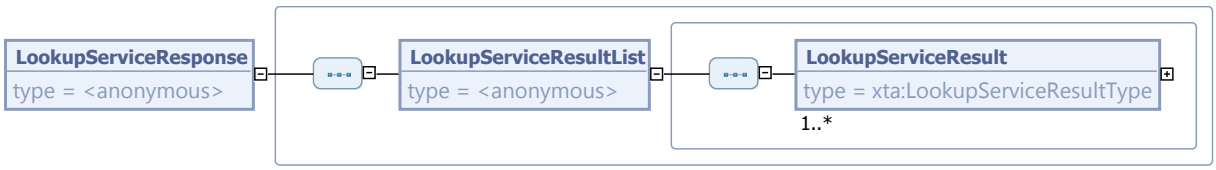
Kindelement von *LookupServiceRequest*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
LookupServiceRequestList		**1..n**		
Dies ist die Struktur für eine Liste von Dienstanfragen.				
LookupService	*xta:LookupServiceType*	**1**	5.5.1.6	150
Dies ist eine Service-Anfrage. Sie enthält Daten zum potentiellen 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.				

5.5.2.3 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.

Abbildung 5.10. LookupServiceResponse



Kindelement von *LookupServiceResponse*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
LookupServiceResultList		**1**		
Die Struktur einer Liste von Dienstanfrageergebnissen.				
LookupServiceResult	*xta:LookupServiceResultType*	**1..n**	5.5.1.7	150

Kindelement von *LookupServiceResponse*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
	Dies ist die Struktur der Liste von Ergebnissen zur Liste von Diensteanfragen.			

5.5.2.4 TransportReport

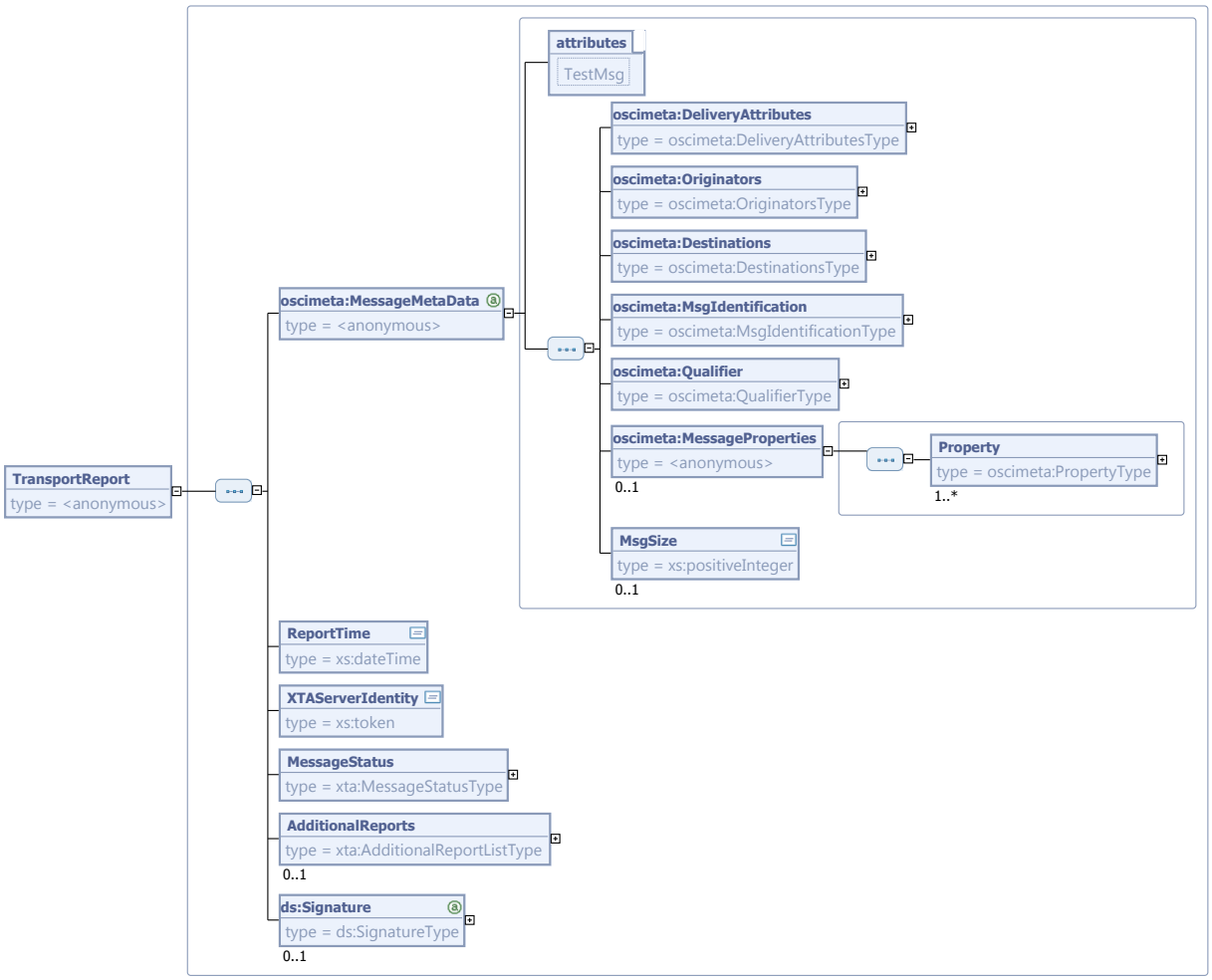
Nachricht: *TransportReport*

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.

Abbildung 5.11. TransportReport



Kindelemente von *TransportReport*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
oscimeta:MessageMetaData	*globales Element*	**1**	2.4.2	24
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 Service Qualität betreffen. Weitere Informationen zu diesem Objekt sind in Abschnitt 5.4.2.3.1 auf Seite 122 zu finden.				
TestMsg	*xs:boolean*	**0..1**		
Hier ist "true" eingestellt, falls die vorliegende Instanz zu Testzwecken versendet wurde bzw. dafür vorgesehen ist. Default ist "false".				
oscimeta:DeliveryAttributes	*globales Element*	**1**	2.4.2	24
Hier geht es um spezielle Merkmale wie Zeitstempel, indiividuell angeforderte Quittungen und Service Qualität.				
oscimeta:Originators	*globales Element*	**1**	2.4.2	24
Hier wird der Ersteller der Nachricht (Autor) eingetragen, sowie die gewünschte Antwort-Adressierung.				
oscimeta:Destinations	*globales Element*	**1**	2.4.2	24
Hier wird der Endkonsument der Nachricht (Leser) eingetragen.				
oscimeta:MsgIdentification	*globales Element*	**1**	2.4.2	24
Hier steht die Identifikation des Transportauftrags (MessageID), sowie Zusammenhänge der Nachricht zu Fachprozessen.				
oscimeta:Qualifier	*globales Element*	**1**	2.4.2	24
Generischer Container für Payload-Eigenschaften, die in jedem Szenario anwendbar sind.				
oscimeta:MessageProperties	*globales Element*	**0..1**	2.4.2	24
Generischer Container für szenarienspezifische Payload-Eigenschaften, die im Anwendungsszenario zu konkretisieren sind.				
Property	*oscimeta:PropertyType*	**1..n**	2.4.2	24
Pro Element wird eine solche Payload-Eigenschaft eingetragen.				
MsgSize	*xs:positiveInteger*	**0..1**		
Hier ist die Nachrichtengröße in Bytes zu erfassen.				
ReportTime	*xs:dateTime*	**1**		
Zeitpunkt der letzten Aktualisierung des Protokolls. Ist bei Fortschreibung des Protokolls zu überschreiben.				
XTAServerIdentity	*xs:token*	**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.				
MessageStatus	*xta:MessageStatusType*	**1**	5.5.1.5	148
Enthält Information über den Veraluf 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.				
AdditionalReports	*xta:AdditionalReportListType*	**0..1**	5.5.1.3	147
Hier sind weitere Prüfberichte abgelegt bzw. abzulegen, welche das XTA-Protokoll (*TransportReport*) ergänzen sollen.				
ds:Signature	*globales Element*	**0..1**	2.4.6	25
Falls der TransportReport signiert ist, findet sich hier die Signatur.				

5.6 XTA-WS SOAP Exceptions

Fehler des XTA-WS werden als SOAP 1.2 Exceptions geworfen. Es sind eine Reihe solcher Exceptions für den XTA-Webservice definiert. Sie werden in Abschnitt 5.6.1 auf Seite 158 aufgezählt und dokumentiert. Sie sind alle von derselben Basisklasse abgeleitet, die in Abschnitt 5.6.2 auf Seite 161 erläutert wird.

5.6.1 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 5.6.3 auf Seite 161).

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.

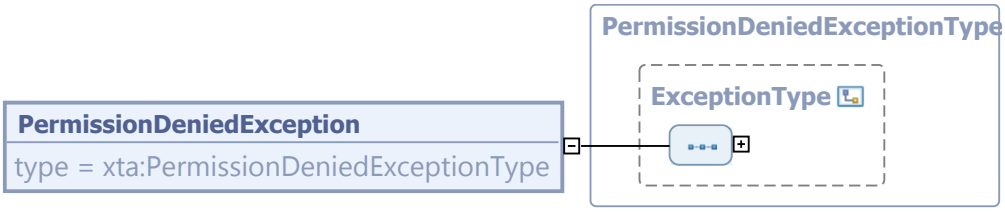
Umsetzungshinweis für die Implementierung von Webservice-Clients: Bei fast allen Exceptions macht ein automatisierter erneuter Aufruf der entsprechenden Methode keinen Sinn. Es ist erst eine Fehlerklärung erforderlich. Eine Ausnahme bildet nur die Exception XTAWSTechnicalProblemException (vgl. Abschnitt 5.6.1.8 auf Seite 160). Tritt diese Exception auf, ist typischerweise ein erneuter, automatisierter Aufruf der entsprechenden Methode (im Fall eines Nachrichten-Versands mit neuer MessageID) sinnvoll.

5.6.1.1 PermissionDeniedException

Nachricht: *PermissionDeniedException*

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

Abbildung 5.12. PermissionDeniedException

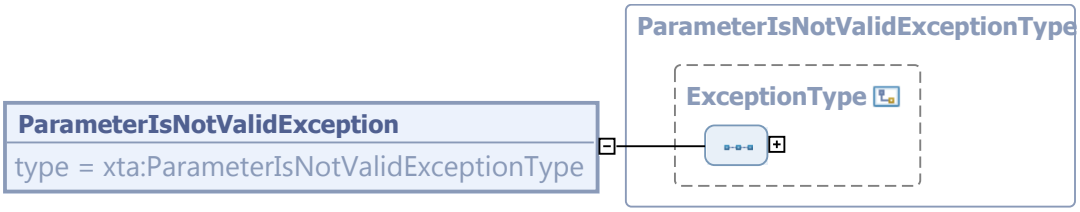


5.6.1.2 ParameterIsNotValidException

Nachricht: *ParameterIsNotValidException*

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

Abbildung 5.13. ParameterIsNotValidException

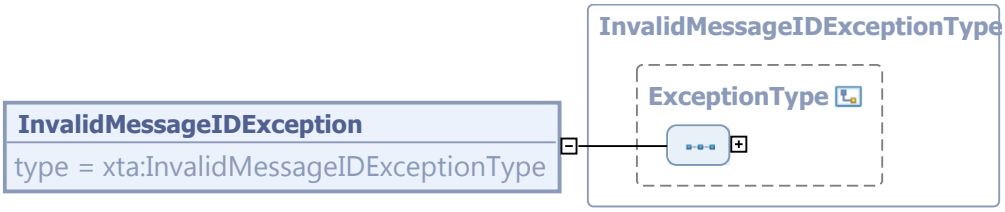


5.6.1.3 InvalidMessageIDException

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.

Abbildung 5.14. InvalidMessageIDException

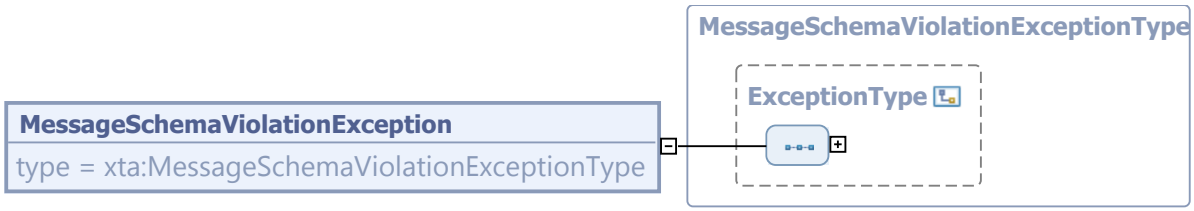


5.6.1.4 MessageSchemaViolationException

Nachricht: *MessageSchemaViolationException*

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

Abbildung 5.15. MessageSchemaViolationException



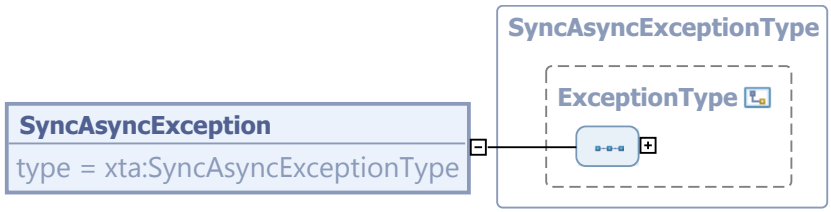
5.6.1.5 SyncAsyncException

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.

Abbildung 5.16. SyncAsyncException

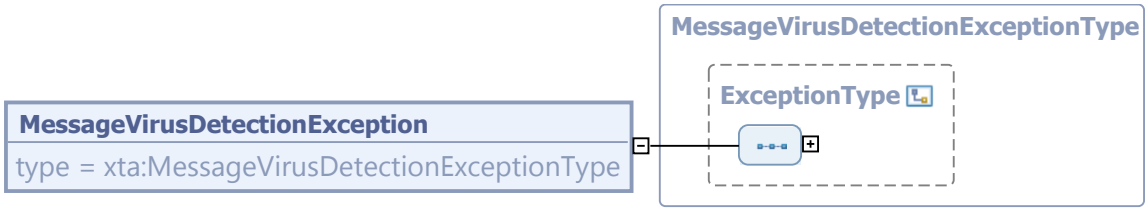


5.6.1.6 MessageVirusDetectionException

Nachricht: *MessageVirusDetectionException*

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

Abbildung 5.17. MessageVirusDetectionException



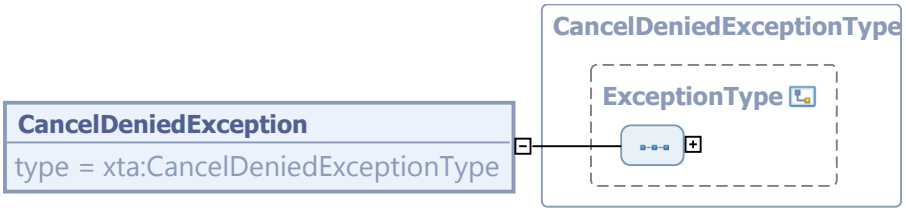
5.6.1.7 CancelDeniedException

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.

Abbildung 5.18. CancelDeniedException

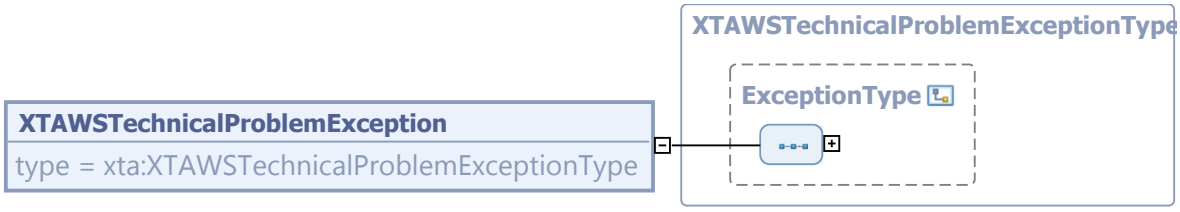


5.6.1.8 XTAWSTechnicalProblemException

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.

Abbildung 5.19. XTAWSTechnicalProblemException



5.6.2 Struktur von Exception und Fehlernummer

Die in Abschnitt 5.6.1 auf Seite 158 dargestellten Exceptions sind alle von einer gemeinsamen Basisklasse abgeleitet. Sie kapselt die Fehlernummer, die zur näheren Beschreibung einer Fehlersituation verwendet wird.

5.6.2.1 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.

Abbildung 5.20. ExceptionType

Kindelement von *ExceptionType*				
Kindelement	**Typ**	**Anz.**	**Ref.**	**Seite**
errorCode	*xta:Code.Fehlernummer*	**1**	5.6.2.2	161
In diesem Element werden Fehlernummer und Fehlertext übermittelt, die einen Fehler näher beschreiben (gemäß verlinkter Codeliste). In das Unterelement *code* ist die Fehlernummer einzutragen, ins Unterelement *name* die entsprechende textuelle Repräsentation.				

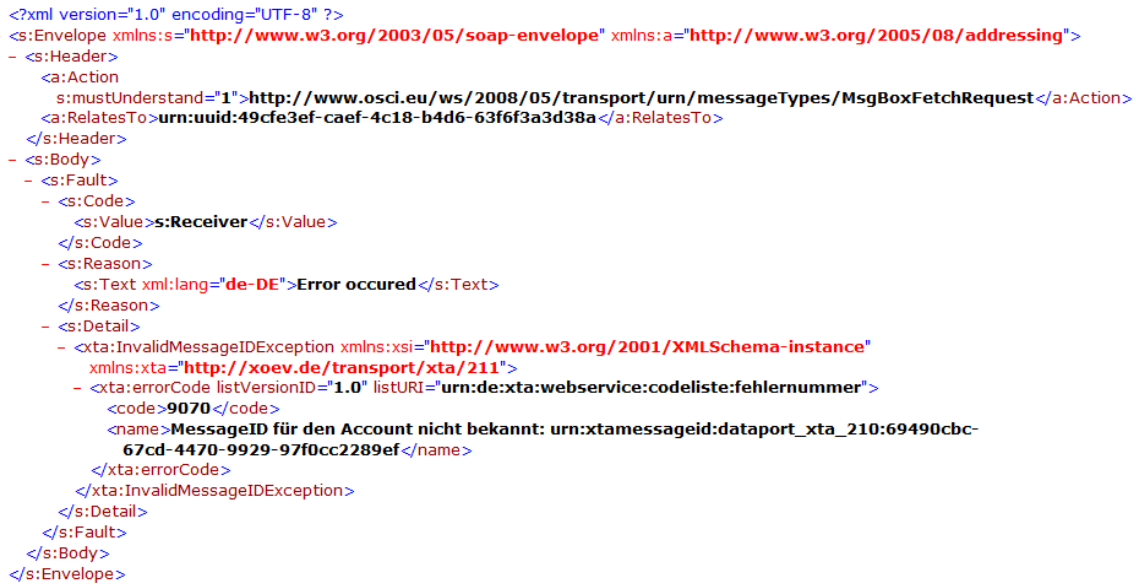
5.6.2.2 Code.Fehlernummer

Code	**Code.Fehlernummer**
Beschreibung	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.
Codelisten-Nutzung	Typ: 2, Inhalte der Codeliste siehe Seite 183
Codelisten-URI	urn:de:xta:webservice:codeliste:fehlernummer
Codelisten-Version	1.0

5.6.3 Exceptions als XML-Instanzen

Ein Beispiel (Abbildung 5.21, „Exception als SOAP-Text“) soll verdeutlichen, wie eine in XTA-WS definiertes Exception-Objekt im Rahmen einer transportierten SOAP-Exception eingesetzt wird.

Abbildung 5.21. 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:xtamessageid: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-WS 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 5.6.1 auf Seite 158. Es muss vom XTA-WS-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 Schlüsseltabellen

A.1 Codelisten-Index

Name	# Einträge	Tabellendetails	Code-Datentyp
Abgabestation	3	Seite 164	Seite 100
Geltungsbereich Infrastruktur-Parameter	4	Seite 165	Seite 78
Geltungsbereich Schutzprofil-Parameter	5	Seite 166	Seite 74
Kanal	5	Seite 167	Seite 79
Kommunikation Typ	2	Seite 168	Seite 99
Nachweis Verlässlichkeit	2	Seite 169	Seite 75
Qualität Authentizität	2	Seite 170	Seite 74
Qualität Kryptographie	2	Seite 171	Seite 88
Qualität Löschen	2	Seite 172	Seite 74
Qualität Protokollierung	3	Seite 173	Seite 74
Qualität Unveränderbarkeit	3	Seite 174	Seite 75
Qualität Verfügbarkeit	4	Seite 175	Seite 99
Qualität Vertraulichkeit	3	Seite 176	Seite 75
Record Type	Identität der Codeliste nicht durch den Standard XTA festgelegt	Identität der Codeliste nicht durch den Standard XTA festgelegt	Seite 147
Report Type	Identität der Codeliste nicht durch den Standard XTA festgelegt	Identität der Codeliste nicht durch den Standard XTA festgelegt	Seite 148
Service Parameter Type	Identität der Codeliste nicht durch den Standard XTA festgelegt	Identität der Codeliste nicht durch den Standard XTA festgelegt	Seite 152
Technische Quittungen	5	Seite 177	Seite 75
Transportnachrichten Format	5	Seite 178	Seite 79
Transportprotokoll	6	Seite 179	Seite 79
Verzeichnis für Adressierung	4	Seite 180	Seite 78

Name	# Einträge	Tabellendetails	Code-Datentyp
Verzeichnis für Identifizierung	5	Seite 181	Seite 79
XTA-Rolle	4	Seite 182	Seite 89
XTA-WS Fehlernummer	23	Seite 183	Seite 161
Zertifikat Medium	2	Seite 184	Seite 75
Zertifikat Niveau	2	Seite 185	Seite 76
Zertifikat Quelle	5	Seite 186	Seite 99
Zustellfrist	3	Seite 187	Seite 100

A.2 Details

A.2.1 Schlüsseltabelle Abgabestation

Codeliste	**Abgabestation (urn:xoev-de:xta:serviceprofile:codeliste:abgabestation)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
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.**
Schlüssel	**Wert**
relay	Abgabestation ist der Knoten 'Relay des Empfängers (Postfach)'
empfänger	Abgabestation ist die Rolle 'Empfänger'
leser	Abgabestation ist die Rolle 'Leser'

A.2.2 Schlüsseltabelle Geltungsbereich Infrastruktur-Parameter

Codeliste	**Geltungsbereich Infrastruktur-Parameter (urn:xoev-de:xta:serviceprofile:codeliste:geltungsbereich.infrastrukturprofil-parameter)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste enthält die Schlüssel für den Geltungsbereich eines Infrastrukturprofils. Jeder Eintrag der Codeliste nennt einen (fachneutral bezeichneten) Ausschnitt des Nachrichtenaustauschs der öffentlichen Verwaltung.**
Schlüssel	**Wert**
bundesweit	Geltungsbereich ist der bundesweite Nachrichtenaustausch.
länderübergreifend	Geltungsbereich ist der länderübergreifende Nachrichtenaustausch.
landesintern	Geltungsbereich ist landesinterner Nachrichtenaustausch.
kein	Es ist kein Geltungsbereich definiert.

A.2.3 Schlüsseltabelle Geltungsbereich Schutzprofil-Parameter

Codeliste	**Geltungsbereich Schutzprofil-Parameter (urn:xoev-de:xta:serviceprofile:codeliste:geltungsbereich.schutzprofil-parameter)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste enthält die Schlüssel für die Kommunikationsstrecken, die in einer XTA-Infrastruktur betrachtet werden können bzw. die für die Schutzprofile von Interesse sind (z.B. die Strecke 'Autor-Sender' oder die Strecke 'Autor-Leser'). Sie werden verwendet, um den Geltungsbereich der Ausprägung einer Service Qualität der Nachrichtenkommunikation zu benennen.** **Beispielsweise kann in einem Schutzprofil die Service Qualität 'Vertraulichkeit hoch' für die Strecke (=den Geltungsbereich) 'Autor-Leser' gefordert werden.**
Schlüssel	**Wert**
autor-leser	Betrifft den Bereich der Kommunikation Autor - Leser
autor-sender	Betrifft den Bereich der Kommunikation Autor - Sender
autor-empfänger	Betrifft den Bereich der Kommunikation Autor - Empfänger
sender-empfänger	Betrifft den Bereich der Kommunikation Sender - Empfänger
empfänger-leser	Betrifft den Bereich der Kommunikation Empfänger - Leser

A.2.4 Schlüsseltabelle Kanal

Codeliste	**Kanal (urn:xoev-de:xta:serviceprofile:codeliste:kanal)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste enthält die Schlüssel für die Beschreibung des Kanals der Kommunikation von Sender und Empfänger, d. h. die Art der Verbindung oder das Netzsegment, über das sie kommunizieren.**
Schlüssel	**Wert**
internet	Es handelt sich um Übermittlung über das Internet.
verbindungsnetz	Es handelt sich um Übermittlung über das Verbindungsnetz.
notarnet	Es handelt sich um Übermittlung über das NotarNet.
rz-intern	Es handelt sich um rechenzentrumsinterne Übermittlung.
systemintern	Es handelt sich um systeminterne Übermittlung.

A.2.5 Schlüsseltabelle Kommunikation Typ

Codeliste	**Kommunikation Typ (urn:xoev-de:xta:serviceprofile:codeliste:kommunikationstyp)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste nennt die Arten, wie der Empfang einer Nachricht durch den Leser mit dem Absenden der Antwortnachricht durch ihn technisch gekoppelt sein kann.**
Schlüssel	**Wert**
asynchron	Die Kommunikation ist asynchron (Aktion und Reaktion snd technisch voneinander entkoppelt).
synchron	Die Kommunikation ist synchron (technisch Kopplung von Aktion und Reaktion).

A.2.6 Schlüsseltabelle Nachweis Verlässlichkeit

Codeliste	**Nachweis Verlässlichkeit (urn:xoev-de:xta:serviceprofile:codeliste:nachweis.verlaesslichkeit)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste beschreibt Arten, wie eine Organisation (Hersteller oder Betreiber einer Software) ihre Verlässlichkeit belegen kann. Kraft dieser Verlässlichkeit steht sie ein für die Unveränderbarkeit der durch die genannte Software unterstützten Prozesse (abgesehen von Kontexten, die der definierten Intervenierbarkeit dienen). Auch steht sie dafür ein, dass es ein effektives Changemanagement für die entsprechenden Prozesse gibt und diese Prozesse auch nicht umgangen werden können.**
Schlüssel	**Wert**
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.2.7 Schlüsseltabelle Qualität Authentizität

Codeliste	**Qualität Authentizität (urn:xoev-de:xta:serviceprofile:codeliste:authentizitaetqualitaet)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste enthält die Schlüssel für die zur Verfügung stehenden Niveaus, auf denen sich die Authentizität der Nachrichtenkommunikation auf einer entsprechenden Strecke abgesichert werden kann.**
Schlüssel	**Wert**
normal	Das geforderte Sicherheitsniveau der Daten ist 'normal'. Die Identität wird mit gewöhnlichen Mitteln geprüft. Beispiel für die Prüfung der Autorschaft einer Nachricht durch einen Leser (Fachbehörde): Die Behördenkategorie des Kommunikationspartners passt zum übertragenen Nachrichtentyp.
hoch	Das geforderte Sicherheitsniveau der Daten ist 'hoch'. Die Identität ist zusätzlich mit speziellen Mitteln zu prüfen. Beispiel für die Prüfung der Autorschaft einer Nachricht durch einen Leser (Fachbehörde): Die Behördenkategorie des Kommunikationspartners passt zum übertragenen Nachrichtentyp und zusätzlich muss der Identifikator des Kommunikationspartners gültig sein.

A.2.8 Schlüsseltabelle Qualität Kryptographie

Codeliste	**Qualität Kryptographie (urn:xoev-de:xta:serviceprofile:codeliste:kryptographie.qualitaet)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste enthält die Schlüssel für die Abstufungen 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 .**
Schlüssel	**Wert**
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.2.9 Schlüsseltabelle Qualität Löschen

Codeliste	**Qualität Löschen (urn:xoev-de:xta:serviceprofile:codeliste:loeschen.qualitaet)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste enthält die Schlüssel für die Ausprägungen der Service Qualität „Löschen von personen- oder organisationsbezogenen Daten“. Sie enthält mögliche Arten der Löschung dieser Daten zum vorgegebenen Zeitpunkt.**
Schlüssel	**Wert**
normal	Einfaches Löschen reicht aus.
hoch	Es ist eine sicherere Methode des Löschens vorgesehen, beispielsweise Löschen mit mehrfachem Überschreiben.

A.2.10 Schlüsseltabelle Qualität Protokollierung

Codeliste	**Qualität Protokollierung (urn:xoev-de:xta:serviceprofile:codeliste:protokollierung.qualitaet)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste enthält die Schlüssel für Ausprägungen der Protokollführung. Es wird festgelegt, ob ein Protokoll zu führen ist und unter welchem Absicherungsniveau dies ggf. zu geschehen hat.**
Schlüssel	**Wert**
niedrig	Protokollierung kann erfolgen, ist aber nicht zwingend erforderlich.
normal	Standardprotokoll. Protokollierung muss erfolgen. Sie kann, muss aber nicht revisionssicher sein.
hoch	Protokollierung muss erfolgen. Es müssen Schutzmaßnahmen entsprechend der für das Protokoll gemäß Abschnitt 2.1.3, „Begriffe zu Datenschutz und Datensicherheit“ zu berücksichtigenden Schutzziele getroffen sein. Zum Beispiel können Schutzmaßnahmen für das Schutzziel Integrität auf dem Niveau "hoch" sein: Revisionsfeste Protokollierung, ggf. kryptogesichert unter Verwendung eines zertifizierten Zeitstempels mit angemessener Auflösung, ggf. auch der Einsatz eines dedizierten Protokollservers.

A.2.11 Schlüsseltabelle Qualität Unveränderbarkeit

Codeliste	**Qualität Unveränderbarkeit (urn:xoev-de:xta:serviceprofile:codeliste:unveraenderbarkeit.qualitaet)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste enthält die Schlüssel für die verschiedenen Niveaus von abgesicherter Unveränderbarkeit, die auf entsprechenden Strecken gefordert sein können.**
Schlüssel	**Wert**
normal	Das Sicherheitsniveau der Daten - und damit der Bedarf an Absicherung ihrer Unveränderbarkeit - ist normal.
hoch	Das Sicherheitsniveau der Daten - und damit der Bedarf an Absicherung ihrer Unveränderbarkeit - ist hoch.
niedrig	Das Sicherheitsniveau der Daten - und damit der Bedarf an Absicherung ihrer Unveränderbarkeit - ist niedrig.

A.2.12 Schlüsseltabelle Qualität Verfügbarkeit

Codeliste	**Qualität Verfügbarkeit (urn:xoev-de:xta:serviceprofile:codeliste:verfuegbarkeit.qualitaet)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste beschreibt die für Transportaufträge vorgesehenen Verfügbarkeitsstufen. Die Verfügbarkeit ist dabei die Wahrscheinlichkeit, dass der Transportauftrag innerhalb des vereinbarten Zeitraums ausgeführt wird.**
Schlüssel	**Wert**
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.2.13 Schlüsseltabelle Qualität Vertraulichkeit

Codeliste	**Qualität Vertraulichkeit (urn:xoev-de:xta:serviceprofile:codeliste:vertraulichkeit.qualitaet)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste enthält die Schlüssel für die zur Verfügung stehenden Niveaus der Service Qualität der Vertraulichkeit der Nachrichtenkommunikation auf einer bestimmten Strecke der Messaging Infrastruktur.**
Schlüssel	**Wert**
öffentlich	Keine Vertraulichkeit notwendig. Das Schutzniveau ist niedrig, es handelt sich bei den Daten um öffentliche Daten.
normal	Es ist normale Vertraulichkeit vorgesehen. Das geforderte Sicherheitsniveau der Daten ist normal.
hoch	Es ist hohe Vertraulichkeit vorgesehen. Das Sicherheitsniveau der Daten ist hoch.

A.2.14 Schlüsseltabelle Technische Quittungen

Codeliste	**Technische Quittungen (urn:xoev-de:xta:serviceprofile:codeliste:technische.quittungen)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste enthält die Schlüssel für die Arten technischer Quittungen, welche in einer XTA-Infrastruktur vorgesehen sind (siehe Abschnitt 2.3, „Quittungen in XTA 2“).**
Schlüssel	**Wert**
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 *Relay* 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. fie Nachricht aus dem Postkorb abgeholt wurde.
keineQuittung	Es werden keine technischen Quittungen verlangt.

A.2.15 Schlüsseltabelle Transportnachrichten Format

Codeliste	**Transportnachrichten Format (urn:xoev-de:xta:serviceprofile:codeliste:transportnachrichten.format)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste nennt verfügbare Nachrichtenformate für Transportnachrichten (normalerweise basierend auf dem XML-Format SOAP).**
Schlüssel	**Wert**
osci12	OSCI 1.2
osci2	OSCI 2
xta11	XTA 1.1
xta2	XTA 2
systemintern	Es ist ein systeminternes Nachrichtenformat vorgesehen.

A.2.16 Schlüsseltabelle Transportprotokoll

Codeliste	**Transportprotokoll (urn:xoev-de:xta:serviceprofile:codeliste:transportprotokoll)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste nennt verfügbare Protokolle, dier die Kommunikaiton von Daten zwischen Partnern festlegen.**
Schlüssel	**Wert**
http	HTTP
https	HTTPS
ftp	FTP
sftp	SFTP
smtp	SMTP
systemintern	Es ist ein systemintern definiertes Protokoll vorgesehen.

A.2.17 Schlüsseltabelle Verzeichnis für Adressierung

Codeliste	**Verzeichnis für Adressierung (urn:xoev-de:xta:serviceprofile:codeliste:verzeichnis.adressierung)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste enthält die Schlüssel für Verzeichnislösungen zur Bereitstellung von Parametern für die technische Adressierung von Teilnehmern.**
Schlüssel	**Wert**
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.2.18 Schlüsseltabelle Verzeichnis für Identifizierung

Codeliste	**Verzeichnis für Identifizierung (urn:xoev-de:xta:serviceprofile:codeliste:verzeichnis.identifzierung)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste enthält die Schlüssel für Verzeichnislösungen zur Verwaltung von elektronischen Identitäten (Bezeichnungen und kryptographische Token für Identität und Identitätsnachweis).**
Schlüssel	**Wert**
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.2.19 Schlüsseltabelle XTA-Rolle

Codeliste	**XTA-Rolle (urn:xoev-de:xta:serviceprofile:codeliste:xta-rolle)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste benennt die Rollen, die in einer XTA-Kommunikationsinfrastruktur am Prozess der Nachrichtenübermittlung beteiligt sind.** **Ein Knoten Relay ist nicht in den Einträgen der Codeliste aufgeführt: Ein Relay ist keine eigene Rolle, sondern entweder der XTA-Rolle Sender oder der XTA-Rolle Empfänger zugeordnet.**
Schlüssel	**Wert**
autor	XTA-Rolle 'Autor'
sender	XTA-Rolle 'Sender'
empfänger	XTA-Rolle 'Empfänger'
leser	XTA-Rolle 'Leser'

A.2.20 Schlüsseltabelle XTA-WS Fehlernummer

Codeliste	**XTA-WS Fehlernummer (urn:de:xta:webservice:codeliste:fehlernummer)**	
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**	
Beschreibung	**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.**	
Schlüssel	**Wert**	**Beschreibung**
9000	Unspezifizierter Fehler, als Freitext beschrieben	
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	ParameterIsNotValidExeption
9021	Keine gültige URI	ParameterIsNotValidExeption
9022	Ungültige Parameterkombination	ParameterIsNotValidException
9023	Die Nachricht überschreitet die Größenbeschränkung.	ParameterIsNotValidException
9024	MessageID ist bereits vergeben.	ParameterIsNotValidException
9030	Interner Fehler beim XTA-Server bzw. XTA-Dienstleister	XTAWSTechnicalProblemException
9031	Fehler beim externen Verzeichnisdienst	XTAWSTechnicalProblemException
9032	Fehler bei der Zustellung	XTAWSTechnicalProblemException
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
9100	Der durch den Schalter NotBefore gesetzte Termin ist verstrichen.	CancelDeniedException
9101	Der Schalter NotBefore wurde nicht gesetzt.	CancelDeniedException

A.2.21 Schlüsseltabelle Zertifikat Medium

Codeliste	**Zertifikat Medium (urn:xoev-de:xta:serviceprofile:codeliste:zertifikat.medium)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste benennt die verschiedenen Medien, die ein Zertifikat tragen können.**
Schlüssel	**Wert**
hardwarebasiert	Es ist ein hardwarebasiertes Medium für das Zertifikat vorgesehen.
softwarebasiert	Es ist ein softwarebasiertes Medium für das Zertifikat vorgesehen.

A.2.22 Schlüsseltabelle Zertifikat Niveau

Codeliste	**Zertifikat Niveau (urn:xoev-de:xta:serviceprofile:codeliste:zertifikat.niveau)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste enthält die Schlüssel für die definierten Niveaus eines Zertifikats.**
Schlüssel	**Wert**
fortgeschritten	Vorgesehen sind Zertifikate für fortgeschrittene Signatur.
qualifiziert	Vorgesehen sind Zertifikate für qualifizierte Signatur.

A.2.23 Schlüsseltabelle Zertifikat Quelle

Codeliste	**Zertifikat Quelle (urn:xoev-de:xta:serviceprofile:codeliste:zertifikat.quelle)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste nennt Möglichkeiten für festzulegende Quellen (Herausgeber) von Zertifikaten für eine Public Key Infrastruktur.**
Schlüssel	**Wert**
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: http://www.xjustiz.de/system/pdf/2009-05-28-zertifikatsherausgeber.pdf
beliebig	Wer der Herausgeber ist, wird nicht näher eingeschränkt.

A.2.24 Schlüsseltabelle Zustellfrist

Codeliste	**Zustellfrist (urn:xoev-de:xta:serviceprofile:codeliste:zustellfrist)**
Herausgeber	**Koordinierungsstelle für IT-Standards (KoSIT)**
Beschreibung	**Diese Codeliste nennt Zeitintervalle, die als Zustellfrist in Frage kommen. Die vorgegebene Zustellfrist eines Transportauftrags ist das Zeitspanne, innerhalb derer die Nachrichtenzustellung erfolgt sein muss.**
Schlüssel	**Wert**
72h	Vorgesehen ist die Zustellung in den Verfügungsbereich des Lesers innerhalb von drei Tagen bzw. 72 Stunden.
24h	Vorgesehen ist die Zustellung in den Verfügungsbereich des Lesers innerhalb von einem Tag bzw. 24 Stunden.
unverzüglich	Vorgesehen ist unverzügliche Zustellung (so schnell wie möglich, ohne dass schuldhaft verzögert wird). Diese Qualität ist im Zusammenhang der weiteren festgelegten Service Qualitäten, wie z. B. der Art der Kommunikation, zu interpretieren.

B Anhang zum XTA Webservice

B.1 Beispielcode

Als zusätzliche Dokumentation werden die unterschiedlichen Versand- und Empfangsoptionen in Beispielcode (in einer an verbreitete Programmiersprachen angelehnte Pseudo-Sprache) dargestellt. Er soll den prinzipiellen Aufbau der Programme bzgl. der Nutzung der XTA-Funktionalitäten verdeutlichen.

B.1.1 Autor

Es werden folgende Variablen verwendet:

- $myAuthor: Die fachliche Kennung des Autors.
- $myMetadata: Die zu der zu versendenden Nachricht gehörende Metadaten
- $myMessage: Ein GenericContantContainer, der sowohl die zu sende Nachricht als auch die Antwort Nachricht enthält.
- $myMessageID: Die ID des zurückzurufenden Transportauftrags.
- $myLookupServiceRequest: Liste der zu prüfenden Empfänger
- $myLookupServiceResponse: Liste der geprüften Empfänger mit Prüfergebnis
- $myX509TokenContainer: Liste der zu Prüfenden Zertifikate

Ergänzend zu den XTA-WS-Methoden müssen für den Leser folgende Funktionen implementiert werden:

- extractState() : Aus dem TransportReport wird der Status extrahiert, der angibt, ob der Versand bereits durchgeführt wurde und ob er erfolgreich durchgeführt wurde.
- sendEscalationForMessage(): Hinweis auf ein Versand-Problem

B.1.1.1 Asynchroner Versand einer Nachricht

```
# Zuerst wird geprüft:
# 1. Gibt es einen aktiven Account beim Sender/Empfänger?
# 2. Funktioniert die Verbindung zum Sender/Empfänger?
if (not checkAccountActive($myAuthor)) {
   exit;
}

# Bietet der Empfänger den gewünschten Dienst an?
$myLookupServiceResponse =
```

```
lookupService($myAuthor, $myLookupServiceRequest);

# Erzeugen einer neuen MessageID
$aMessageID = ceateMessageID($myAuthor);

Try
{
  # Asynchroner Versand der Nachricht
  sendMessage($myMetadata, $myX509TokenContainer, $myMessage);

  # Wurde der Transport erfolgreich durchgeführt?
  # Hole Report bis Versandauftrag bearbeitet wurde. Eine
  # angemessene Wartezeit verhindert unangemessen viele
  # Anfragen beim Sender.

  repeat {
    # Eine Wartepause lässt dem Sender Zeit zum Senden.
    sleep(3600);
    # TransportReport holen
    $aReport = getTransportReport($aMessageID);
    # Den Status des Transportauftrags auslesen
    $aState = extractState($aReport);
  } until ($aState != „offen“);

  # Ist der Versand fehlgeschlagen, wird der
  # Verantwortliche informiert.
  if ($aState equal „rot“) {
    sendEscalationForMessage($aMessageID, $aReport);
  }
}
Catch
{
   # XTA Exception verarbeiten
   sendEscalationForMessage($myMessage);
}
```

B.1.1.2 Synchroner Versand einer Nachricht

```
# Zuerst wird geprüft:
# 1. Gibt es einen aktiven Account beim Sender/Empfänger?
# 2. Funktioniert die Verbindung zum Sender/Empfänger?
if (not checkAccountActive($myAuthor)) {
   exit;
}

# Bietet der Empfänger den gewünschten Dienst an?
$myLookupServiceResponse =
lookupService($myAuthor, $myLookupServiceRequest);

# Erzeugen einer neuen MessageID
$aMessageID = ceateMessageID($myAuthor);
```

```
Try
{
  # Synchroner Versand der Nachricht. Die Argumente werden
  # als „ref“ übergeben, d. h. sie können geändert werden.
  # Hier wird die Antwort zurückgegeben.
  sendMessageSync(ref $myMetadata, ref $myX509TokenContainer,
                  ref $myMessage);

  # TransportReport holen
  $aReport = getTransportReport($myAuthor, $aMessageID);
  # Den Status des Transportauftrags auslesen
  $aState = extractState($myAuthor, $aReport);

  # Ist der Versand fehlgeschlagen, wird der
  # Verantwortliche informiert.
  if ($aState equal „rot“) {
    sendEscalationForMessage($aMessageID, $aReport);
  }
}
Catch
{
   # XTA Exception verarbeiten
   sendEscalationForMessage($myMessage);
```

B.1.1.3 Rückruf einer Nachricht

```
# Zuerst wird geprüft:
# 1. Gibt es einen aktiven Account beim Sender/Empfänger?
# 2. Funktioniert die Verbindung zum Sender/Empfänger?
if (not checkAccountActive($myAuthor)) {
   exit;
}

# Status der Nachricht im TransportReport prüfen.
$aReport = getTransportReport($myAuthor, $myMessageID);
# Den Status des Transportauftrags auslesen
$aState = extractStatus($myAuthor, $aReport);

# Rückruf der Nachricht – falls Nachricht noch in der Warteschlange
# (Schalter NotBefore gesetzt).
If ($aState == „wird noch zurückgehalten“) {
   cancelMessage($myAuthor ,$myMessageID);
}
```

B.1.2 Leser

In den Beispielen werden folgende Variablen verwendet:

- $myFilter: Filter zur Beschreibung der Attribute der abzuholenden Nachrichten.
- $myFrom: Die abzuholenden Nachrichten sollen nach diesem Zeitpunkt eingegangen sein.
- $myTo: Die abzuholenden Nachrichten sollen vor diesem Zeitpunkt eingegangen sein.

Ergänzend zu den XTA-WS-Methoden müssen für den Leser folgende Funktionen implementiert werden:

- setPeriodOfTime: Mit dieser Funktion wird der Zeitraum gesetzt, in dem die abzuholenden Nachrichten angekommen sein müssen.
- setState(): Hiermit wird der Status gesetzt, den die abzuholenden Nachrichten haben sollen.
- setMaxCount(): Hiermit wird im Filter die maximale Anzahl der abzuholenden Einträge gesetzt.
- extractMessage(): Die Funktion liest aus der Rückgabe von getMessage() die enthaltene Nachricht aus.
- extractMessageIDs(): Diese Funktion liest aus der Rückgabe der WS-Methode getStatusList() die enthaltenen MessageIDs aus.
- extractMetadatas(): Diese Funktion liest aus der Rückgabe der WS-Methode getStatusList() die enthaltenen Metadaten aus.
- extractState() : Aus dem TransportReport wird der Status extrahiert der angibt, ob der Versand bereits durchgeführt wurde und wenn ja, ob er erfolgreich durchgeführt wurde.
- extractHandle(): Diese Funktion liest aus den Rückgaben der WS-Methoden getMessage() und getStatusList() die enthaltenen Ressourcenkennung (Handle) aus.
- sendEscalationForMessage(): Hiermit wird ein Problem mit dem Transport eskaliert.
- messageProcessing(): Mit dieser Funktion wird die empfangene Nachricht fachlich verarbeitet.
- metadataProcessing(): Mit dieser Funktion werden die Metadaten einer empfangenen Nachricht verarbeitet.
- popMetadata(): Die Funktion entfernt das erste Metadatenelement aus einer Liste und liefert es als Ergebnis.
- hasNext(): Die Funktion prüft ob noch weitere Einträge vorhanden sind.
- notEmpty(): Die Funktion prüft ob es sich nicht um eine leere Liste handelt.
- startWebService(): Hiermit wird der WebService des Lesers gestartet, der auf eintreffende Nachrichten wartet. Beim Eintreffen einer Nachricht beendet sich der WebService und liefert die empfangene Nachricht als Ergebnis.

B.1.2.1 Asynchroner Empfang von Nachrichten

```
# Definiere gewünschte Nachrichten über Angabe eines
# Zeitintervalls und ob neue, alte oder alle Nachrichten
# abgeholt werden sollen.
setPeriodOfTime($myFilter, $myFrom, $myTo);
setState($myFilter, „NeueNachrichten“);

# Liste der MessageIDs holen
$result = getStatusList($myFilter);
$listOfMessageIDs = extractMessageIDs($result);

# Für jede einzelne ID die Nachricht abholen
foreach $anId ($listOfMessageIDs) {

   # Report für die Nachricht holen und Status extrahieren
   $aReport = getTransportReport($anId);

   # Status extrahieren
   $aState = extractState($aReport);
```

```
   # War beim Transport alles ok, dann darf die Nachricht
   # verarbeitet werden.
   If ( $aState equal „grün“ ) {
      # getMessage() liefert die Nachricht und
      # Zusatzinformationen. Hiervon brauchen wir
      # die Ressourcenkennung (Neudeutsch: Handle)
      $result = getMessage($anId);
      $aMessage = extractMessage($result);
      $aGetMessageHandle = extractHandle($result);

      # Ressourcen freigeben
      close($aGetMessageHandle);

      # Alles ist ok. Die Nachricht darf verarbeitet
      # werden.
      messageProcessing($aMessage);
   } else {
      sendEscalationForMessage($anId, $aReport);
   }
}
```

B.1.2.2 Asynchroner Empfang von Nachrichten – (Zugriff mehrerer Leser)

```
# Definiere gewünschte Nachrichten über Angabe eines
# Zeitintervalls und ob neue, alte oder alle Nachrichten
# abgeholt werden sollen.
setPeriodOfTime($myFilter, $myFrom, $myTo);
setState($myFilter, „NeueNachrichten“);

# Ressourcenkennung für den Zugriff auf die Liste
# der gewünschten Nachrichten holen.
$result = getMessage($myFilter);
$aMessage = extractMessage($result);
$aGetMessageHandle = extractHandle($result);

# Solange Nachrichten abholen bis keine mehr da
# sind.
while ( hasNext($aGetMessageHandle) ){
   # Abholen der nächsten Nachricht.
   $result = getNextMessage($aGetMessageHandle);

   # Die eingebettete Nachricht auslesen.
   $aMessage = extractMessage($result);

   # Die MessageID holen
   $anId = extractMessageIDs($result);

   # Report für die Nachricht holen und Status extrahieren
   $aReport = getTransportReport($anId);

   # Status extrahieren
   $aState = extractState($aReport);
```

```
   # War beim Transport nicht alles ok, dann
   # wird die Nachricht eskaliert.
   If ( $aState equal „rot“ ) {
      sendEscalationForMessage($anId, $aReport);
   } else {
      # Alles ist ok. Die Nachricht darf verarbeitet
      # werden.
      messageProcessing($aMessage);
   }
}

# Ressourcen freigeben
close($aGetMessageHandle);
```

B.1.2.3 Asynchroner Empfang der Metadaten

```
# Definiere gewünschte Metadaten über Angabe eines
# Zeitintervalls und ob neue, alte oder alle Nachrichten
# abgeholt werden sollen. Es sollen jeweils maximal 100
# MessageIDs und Metadaten geholt werden.
setPeriodOfTime($myFilter, $myFrom, $myTo);
setState($myFilter, „NeueNachrichten“);
setMaxCount($myFilter, 100);

# Liste der MessageIDs und Metadaten holen
$result = getStatusList($myFilter);
$listOfMessageIDs = extractMessageIDs($result);
$listOfMetadatas = extractMetadatas($result);
$aGetStatusListHandle = extractHandle($result);

while ( notEmpty($listOfMessageIDs) ) {

   # Überprüfung des Transports und Verarbeitung der
   # einzelnen Metadaten.
   foreach $anId ($listOfMessageIDs) {

      # Report für die Nachricht holen und Status extrahieren
      $aReport = getTransportReport($anId);

      # Status extrahieren
      $aState = extractState($aReport);

      # War beim Transport nicht alles ok, dann
      # wird die Nachricht eskaliert.
      If ( $aState equal „rot“ ) {
         sendEscalationForMessage($anId, $aReport);
      } else {
         # Alles ist ok. Die Metadaten dürfen verarbeitet
         # werden.
         $aMetadata = popMetadata($listOfMetadatas);
         metadataProcessing($aMetadata);
```

```
      }
   }
   # Die nächsten MessageIDs holen
   $listOfMessageIDs = getNextStatusList($aGetStatusListHandle);
}
# Ressourcen freigeben
close($aGetStatusListHandle);
```

B.1.2.4 Synchroner Empfang von Nachrichten

```
# Der Empfang der Nachricht soll immer laufen.
while ( true ) {
   # Den Empfang für die Methode sendMessageSync starten.
   # Er wird beim Empfang einer Nachricht beendet. Die
   # Nachricht wird als Rückgabewert geliefert.
   $aMessage = startWebService(„sendMessageSync“);

   # Die MessageID holen
   $anId = extractId($aMessage);

   # Report für die Nachricht holen und Status extrahieren
   $aReport = getTransportReport($anId);

   # Status extrahieren
   $aState = extractState($aReport);

   # War beim Transport nicht alles ok, dann
   # wird die Nachricht eskaliert.
   if ( $aState equal „rot“ ) {
      sendEscalationForMessage($anId, $aReport);
   } else {
      # Alles ist ok. Die Nachricht darf verarbeitet werden.
      messageProcessing($aMessage);
   }
}
```

C Mitwirkende

Folgende Institutionen und Personen haben bei der Erstellung dieser Spezifikation mitgewirkt:

Institution	Name
KoSIT (Koordinierungsstelle für IT-Standards)	Diederich, Günther Schulte, Beate Steimke, Frank
AKDB (Anstalt für Kommunale Datenverarbeitung in Bayern)	Hack, Peter
ARD ZDF Deutschlandradio Beitragsservice	Collatz, Jürgen
BMI (Bundesministerium des Innern), Referat V II 2 (Meldewesen)	Prauser, Ulrike
Brandenburgischer IT-Dienstleister	Weber, Falk
BSI (Bundesamt für Sicherheit in der Informationstechnik)	Biere, Thomas Laude, Uwe
Bundesdruckerei	Landvogt,Walter
citeq	Helmer, Frank
citkomm (Citkomm services)	Kampmann, Michael
Dataport	Hauschild, Helge Schlüter, Dieter Sorgenfrei, Sören
Datenverarbeitungszentrum Mecklenburg-Vorpommern	Röhl, Mathias
Datenzentrale Baden-Württemberg	Merkel, Klaus Riedel, Martin
Der Landesbeauftragte für Datenschutz und Informationsfreiheit Mecklenburg-Vorpommern (vertritt auch: AK Technik der Konferenz der Datenschutzbeauftragten des Bundes und der Länder)	Weichelt, Rene
Der Sächsische Datenschutzbeauftragte (vertritt auch: AK Technik der Konferenz der Datenschutzbeauftragten des Bundes und der Länder)	Thalheim-Heinecke, Katja
Deutsche Rentenversicherung Bund	Genski, Carina Kannewischer, Sven

Institution	Name
	Meckelein, Werner
ekom21	Grubert, Frank Martin, Matthias Rammenzweig, Martin Steinbeck, Volker
Governikus	Apitzsch, Jörg Ganzer, Marco Lindemann, Ralf Schlegel, Stefanie
HSH	Mütze, Mario Reimann, Carsten Westphal, Andrea
infora	Gerber, Joachim
]init[	Rabenstein,Yorck
Innenministerium Mecklenburg-Vorpommern	Thede, Heiko
IT-Innovationszentrum des Saarlandes	Sokoll, Thorsten
KommWis	Sauer, Andreas
KDO (Kommunale Datenverarbeitung Oldenburg)	Behrens, Marc
Kommunale Informationsverarbeitung Reutlingen-Ulm	Neumann, Andreas
Landesamt für Bürger- und Ordnungsangelegenheiten (LABO), Berlin	Fröhlich, Peter
Landesbetrieb Daten und Information Rheinland Pfalz	Weck, Andreas
Landesbetrieb für Statistik und Kommunikationstechnologie Niedersachsen	Buchmann-Cattau, Astrid
Ministerium des Inneres, für Sport und Infrastruktur Rheinland-Pfalz	Fuhrmann, Martin
procilon	Albus, Hagen Nitzsche, Lars
Staatsbetrieb Sächsische Informatik Dienste	Söhnel, Andreas
Staatsministerium der Justiz und für Europa Sachsen	Popp, Ronald
Thüringer Landesrechenzentrum	Schwarz, Stefan
Unabhängiges Landeszentrum für Datenschutz Schleswig-Holstein (vertritt auch: AK Technik der Konferenz der Datenschutzbeauftragten des Bundes und der Länder)	Rost, Martin
Verlag für Standesamtswesen	Horn, Olaf Jancar, Stephan
Zweckverband Kommunale Datenverarbeitung Region Stuttgart	Rauser, Rainer

D Versionshistorie

D.1 Release XTA 2, Version 3 (31.01.2017)

CR-2013-05 Service Quality

XTA Service Profile: Der Einsatz der *OSCI 2*-Elemente */MessageMetaData/Qualifier/Service* und */MessageMetaData/DeliveryAttributes/ServiceQuality* in XTA wurde geklärt.

In allen Fällen, in denen das Element *ServiceQuality* verwendet wird, ist dadurch eine Referenz auf eine (global oder lokal definierte) ServiceProfil-Instanz vorzunehmen (die *uriDerVersion* der zu referenzierenden ServiceProfil-Instanz ist einzutragen (weitere Detailregelungen siehe Abschnitt 5.4.2.3.2.2 auf Seite 127.

Die Tatsache, dass eine durch einen Fachstandard definierte ServiceProfil-Instanz für mehrere Services dieses Fachstandards gelten kann (vgl. Typ *Servicekategorie*), wird jetzt auch in Abschnitt 4.3 auf Seite 52 und Abbildung 4.3, „Bereitstellung von Profilobjekten in der Zuständigkeit des Fachstandards“ korrekt abgebildet.

CR-2015-03 Servicekategorie und Qualifier

XTA Service Profile: Der Typ *ServiceKategorie* wurde geändert, so dass eine einzige ServiceProfil-Instanz einer Vielzahl von Services eines Fachstandards zugeordnet werden kann.

Die Kardinalität des Elements *service* wurde entsprechen auf 1…n geändert, die Dokumentation der Elemente *service*, *bezeichnung* u.a. angepasst an die neue Struktur und Semantik.

Im Abschnitt 5.4.2.3.3.2 auf Seite 131 wurde Dokumentation zur Verwendung des Elements *Service* nachgetragen.

CR-2015-07 Profil-Instanzen Benennung und Versionierung

XTA Service Profile: Es wurden (zur Integration von Angaben zu Identität und Gültigkeit, sowie weiterer Dokumentation) im Datenmodell der Service Profile drei neue Typen angelegt: *InstanzIdentifikation*, *InstanzGueltigkeit* und *Identifikation*. Auf dieser Basis wurden die Typen *ServiceProfil*, *SchutzProfil*, *InfrastrukturProfil*, *TechnischesStrukturprofil* und *KryptographieProfil* mit einem Datenobjekt *Profilkopf* ausgestattet. Außerdem wurden die auf Profilinstanzen der Profilarten *SchutzProfil* , *InfrastrukturProfil* , *TechnischesStrukturprofil* referenzierenden Elemente innerhalb des Typs *ServiceProfil* geändert: Ihr Name enthält jetzt den Präfix "ref", und sie verwenden zur Referenzierung auf Profil-Instanz den Typ *InstanzIdentifikation*.

CR-2015-09 Schutzkategorie Protokollierung

XTA Service Profile: Die Codeliste Abschnitt A.2.10 auf Seite 173 wurde angepasst. Der Eintrag 'keine' ist entfallen. Die Definition der übrigen Einträge wurde präzisiert.

CR-2015-11 Löschfristen für Protokolle und Daten

XTA Service Profile: Das Datenmodell der XTA Service Profile (Typ Servicekatergorie) wurde um (a) Löschfristen und Vorhaltedauern für Transportprotokolle und um (b) Löschfristen für Daten aus

Fachnachrichten ergänzt. Es lässt sich also jetzt im Service Profil eintragen, wie lange eine Fachnachricht durch einen Transportknoten maximal aufbewahrt werden darf (insofern sie im Zuge der Abarbeitung eines Transportauftrags vom diesem Knoten zwischengespeichert wurde).

CR-2015-12 Termin Zustellung

XTA Service Profile: Das Element terminZustellung wurde aus der Kommunikationskategorie entfernt, weil sich die entsprechende Service Qualität nicht nutzbringend definieren ließ.

CR-2015-13 contentType und Service Profil

In der Dokumentation zum Typ *ContentType* wurde ergänzt, dass die Belegung der Attribute des Typs für verschiedene Fachlichkeiten unterschiedlich ist und durch den jeweiligen Fachstandard festzulegen ist.

CR-2015-15 Sender und Briefkopf

Der zweite Unterpunkt aus Satz A 1.2 des Rollenmodells war veraltet und wurde entfernt (vgl. Abschnitt 2.2.2.1.1 auf Seite 14).

Dort hieß es: *Es ist nicht ausgeschlossen, dass einzelne Informationen aus dem Briefkopf auch für den Sender relevant sind. Dies kann der Fall sein, wenn der Sender die Informationen benötigt, um die technischen Adressdaten des Lesers / Empfängers zu ermitteln.*

Diese Passage wurde auch aus dem Abschnitt Payload erstellen gestrichen

Gemäß Abschnitt 2.1.2.4 auf Seite 12 (unverändert geblieben) wird genau dies von der Architektur gemäß XTA ausgeschlossen.

CR-2015-16 Reihenfolge Processing

XTA Service Profile: Es ist die Notwendigkeit erkannt worden, dass die Instanzen des Elements *processing* des Technischen Strukturprofils sequentiell nach vorgegebener Reihenfolge abgearbeitet werden. Um eine entsprechende Reihenfolge vorzugeben wurden Positionsattribute auf mehreren Ebenen (Ebene der Kindelemente eines *processingList*-Elements und Ebene der Kindelemente eines *processing*-Elements) ergänzt.

CR-2015-18 XTA-WS Soap Faults

In den WSDL-Dateien XTA.wsdl und XTA-synchron.wsdl wurden die SOAP-Fehlermeldungen (Exceptions) der einzelnen Methoden als XSD-Typen eingefügt. Dafür wurde ein Basistyp ExceptionType entworfen, von dem die benötigten Exceptions abgeleitet wurden. Die auf dieser Basis erstellten globalen Elemente wurden, wo erforderlich, in die WSDL-Objekte in den WSDL-Dateien eingetragen. Die Methode GetNextMessage kann eine MessageID entgegen nehmen, dementsprechend wurde die bis dahin fehlende Fehlernachricht InvalidMassageID dieser Methode in der zugehörigen Datei XTA.wsdl hinzugefügt. Die bisherige Schreibweise "InvalidMessageId" wurde der Schreibweise des zugehörigen Elementes MessageID (Großschreibung "ID") im Modell, den WSDL-Dateien und der Spezifikation angepasst. Die verschiedenen SOAP-Fehlercodes wurden als XTA-Codeliste in die Spezifikation integriert.

CR-2015-23 Im MsgBoxPort fehlt MMD

Es wurde der Bedarf erkannt, dem Header der Antwortnachrichten der Operationen getMessage() und getNextMessage() den Container oscimeta:MessageMetaData hinzuzufügen. Insbesondere wird die MessageID aus dem MessageMetaData-Container benötigt, um bei der Bestätigung der Nachricht in der Methode close() über die darin enthaltene MessageID zu verfügen. Der Header wurde entsprechend hinzugefügt und die Dokumentation angepasst. Da durch Hinzufügen des Headers künftig auch andere Selektionskriterien möglich sind, weist die Spezifikation nun darauf hin, dass in der aktuellen (vormals: ersten) Version von XTA ausschließlich das Selektionskriterium MessageID unterstützt wird.

CR-2016-01 MMD vs. Nachrichtenkopf Fachnachricht

Es wurde festgestellt, dass der Eintrag der Autor- bzw. Leser-Identifkation im Nachrichtenkopf der Fachnachricht und im MessageMetaData-Container (*PartyType*) beim XTA-WS-Aufruf konsistent sein müssen, hierfür jedoch keine Verantwortlichkeiten festgelegt wurden. Es wurde festgelegt, dass

die Verantwortung bei der Person liegt, die den Transportauftrag erstellt. Entsprechende Einträge wurden im Rollenmodell vorgenommen.

CR-2016-03 Optionale Teile der XTA-Spezifikation

In Anhang B.1 Asynchroner Abruf von Nachrichten aus dem Postfach wird für die darin beschriebene Variante festgelegt, dass sie „optional implementiert werden kann“. Die praktische Bedeutung wurde bisher nicht festgelegt.

Eine entsprechende Klarstellung wurde in Kapitel 1.2 eingefügt. Zudem werden optionale Teile künftig nicht mehr im Anhang sondern im Hauptteil der Spezifikation beschrieben. Dementsprechend ist die Beschreibung der optionalen Methoden vormalig in Anhang B „Anhang zu „XTA Webservice““ nicht mehr Bestandteil des Anhangs, sondern nunmehr als Abschnitt 5.4.3.5 Optionaler Teil des Schnittstellentyps msgBoxPort optionaler Teil der Methodenbeschreibungen der Spezifikation des XTA-Webservice.

CR-2016-04 XTA-WS Methode getStatusList

In der Dokumentation der Methode getStatusList wurden Fehler und unzureichende Beschreibungen gefunden. Die Bearbeitung führte zu weiterem Verbesserungsbedarf an den Abschnitten 5.4.3 Schnittstellentyp msgBoxPort, 5.4.4 Schnittstellentyp sendSynchronPort - Leser (Synchroner Versand einer Nachricht) und B.1.1.1 Methode getNextMessage sowie B.1.1.2 Methode getNextStatusList. Die Methoden und ihre Parameter wurden in Ihren Bezügen und Übernahmen aus dem Standard OSCI 2.0.1 überprüft und bei Bedarf korrigiert bzw. ergänzt. Der Abschnitt 5.4.4 Schnittstellentyp sendSynchronPort - Leser (Synchroner Versand einer Nachricht) ist nunmehr ein eigenständiger Abschnitt und verweist nicht mehr auf Abschnitt 5.4.2.2 Methode sendMessageSync - Sender (Synchroner Versand einer Nachricht).

CR-2016-05 MTOM für die WSDLs des XTA-WS

Es wurde der Bedarf festgestellt, MTOM zur Performance-Optimierung im Transport von Attachments einzusetzen. Dies wurde bisher nicht von XTA-WS unterstützt. Die Dateien XTA.wsdl und XTA-synchron.wsdl wurden jeweils um den erforderlichen Namespace und eine zugehörige Policy ergänzt, so dass MTOM nun durch XTA unterstützt wird.

CR-2017-02 ServiceProfil-Codelisten URNs

XTA Service Profile: Im Bereich XTA Service Profile wurde auf die URNs aller Codelisten die XÖV-konforme Syntax angewendet ("urn:xoev-de:xta:..." statt bisher "urn:de:xta:...").

D.2 Release XTA 2.1 (30.09.2015)

CR-2013-01 Vereinheitlichung Property Type (Codeliste Service Parameter)

Auf den Typ *LookupServiceResultType* wurde die KoSIT-Codelisten-Infrastruktur angewendet, d.h. die Codelisten-Einbindung auf der Basis des Typs Code (http://xoev.de/schemata/basisdatentypen/1_1) aus dem XÖV-Standardisierungsrahmenwerk vorgenommen. Das Element *Property* wurde umbenannt in *ServiceParameter*. Zusätzlich wurde das neue Element *resource* eingefügt. Die für diesen Kontext empfohlene Codeliste „XTA Service ParameterType“ wurde erstellt und im Repository veröffentlicht. Die Dokumentation zu *Code.ServiceParameterType* und den verbundenen Typen und Elementen wurde angepasst (vgl. Abschnitt 5.5 auf Seite 146). Die Dokumentation zur Methode lookupService (vgl. Abschnitt 5.4 auf Seite 111) wurde konsistent damit formuliert.

CR-2013-02 Cancel Message Fehler

Neudefinition von Semantik, Voraussetzungen, Konsequenzen und Fehlerfällen der Methode cancelMessage. Bearbeitung der Abschnitte (aus der Spezifikation XTA 2.0): *2.2.7.2.10 Transportauftrag zurückziehen*,*4.3.1.3 Rückruf einer Nachricht*, *4.4.1.4 Methode cancelMessage (Rückruf einer Nachricht)*, *4.4.2.3.1.2 DeliveryAttributes*, *4.6 Fehler* und *D.1.3 Rückruf einer Nachricht*.

CR-2013-03 MessageID

Verschiedene redaktionelle Überarbeitungen zu Zweckbestimmung der MesssageID und Klarstellung dazu, wer sie zu erstellen hat (Abschnitte *4.4.1.5 Methode createMessageId, 4.4.1.6.2 MessageID, Rollenmodell A 7.2*).

CR-2013-04 Qualifier

Zu den mandatorischen Kindelementen des Elements Qualifier des MessageMetaData-Containers wurden die nötigen Anleitungen ergänzt, die beim Erstellen des Transportauftrags und bei seiner Interpretation helfen sollen. Wo passend wurde auf entsprechende, durch XTA definierte und im XRepository veröffentlichte Codelisten verwiesen, auf deren Basis der Inhalt von Elementen zu kodieren bzw. zu interpretieren ist. (Abschnitt *4.4.2.3.3.2 Qualifier*).

CR-2013-06 Quittungen

Es wurde ein grundsätzlicher Passus zum Thema Quittungen in Kapitel 3 aufgenommen als Abschnitt *3.2 Quittungen in XTA*. Der bisherige Inhalt des Kapitels 3 wurde zum Abschnitt *3.1 XTA-Profilkonzept*, die Bezeichnung des Kapitels entsprechend generalisiert in "Allgemeines / konzeptionelle Festlegungen". *4.4.2.3.2.3 DeliveryAttributes - Receipt Requests* wurde entsprechend aktualisiert mit Bezug auf den neuen Passus.

CR-2013-09 Parameter Wesensprofile

Die bisher vorliegenden Ansätze des Profilkonzepts (vgl. in XTA 2.0 die Abschnitte *Kapitel 3 XTA-Profilkonzept* mit den Anhängen *B.1 Schutzprofile*, *B.2 Infrastrukturprofile* und *B.3 Wesensprofile*) wurden überarbeitet und ausgearbeitet. Das neu erstellte Kapitel 4, *XTA Service Profile (1.1)*, dokumentiert das Profilkonzept gemäß XTA 2.1. Die Bezeichnung "Service Profile" tritt an die Stelle der Bezeichnung "Wesensprofile". Das in Abschnitt 4.6, „Struktur der Profile“, dokumentierte Datenmodell der Profile wird zusätzlich in Form technischer Artefake durch anliegende XSD-Dateien dargestellt.

CR-2014-01 Umgang mit MessageMetaData

In den Abschnitt *4.4.2.3.1 Daten des Transportauftrags* (neu: *4.4.2.3.1 Der Transportauftrag: Header-Block MessageMetaData*) wurde ein Text aufgenommen, der die Rolle des MessageMetaData-Headers genauer beschreibt und ausführt, was zu tun ist für Sender bzw. Empfänger, falls dieser Header technische Defekte aufweisen sollte.

CR-2014-02 Codeliste Report

Die für diesen Kontext empfohlene Codeliste „Report Type“ wurde definiert und im XRepository veröffentlicht. Der Typ *Code.ReportType* wurde angepasst, so dass zur Laufzeit die Codeliste eingebunden werden kann (aber alternativ auch andere Codelisten eingebunden werden können). Die Dokumentation zu den Typen Code.ReportType und ReportType wurde entsprechend ergänzt. (Abschnitt *4.5 Das Informationsmodell*).

CR-2014-04 Codelisten für PartyType

Die für diesen Kontext empfohlenen Codelisten „XTA Type of PartyIdentifier“ und „XTA XOEV-Category of Party“ wurden definiert und im XRepository veröffentlicht. Die Dokumentation zu PartyIdentifierType wurde entsprechend durch Verweise darauf ergänzt. (Abschnitt *4.4.1.6.1 PartyIdentifier-Type*).

CR-2014-05 DeliveryAttributes

Abschnitt 5.4.2.3, „Wichtige Objekte der sendPort-Schnittstelle“, wurde neu gegliedert und darin ein Abschnitt *Der Transportauftrag: Header-Block MessageMetaData* aufgenommen, um die verschiedenen grundsätzlichen Punkte zum MessageMetaData-Header im Zusammenhang auszuführen. Enthalten ist hier auch eine Passage zu *Protokollierung und Fortschreibung des MessageMetaData-Headers*, in der das Prinzip der Fortschreibung der enthaltenen Daten erläutert wird. Für weitere Detailinformation wurden neue Unterabschnitte eingeführt (zu den TimeStamps der DeliveryAttributes ist dies Abschnitt 5.4.2.3.2.1, „DeliveryAttributes - Zeitstempel“) und dort weitere Vorschriften für die Verarbeitung aufgenommen.

CR-2014-06 XTA-Serverumgebung

Das neue Element *XTAServerIdentity* wurde im Container *TransportReport* angelegt, damit der XTA-Server seine Prozessidentität hinterlegen kann. Zweck ist, dass aus dem Betreiberprotokoll die nötige Kontextinformation hervorgeht (Support-Unterstützung).

CR-2014-07 neue MessageID

Klarstellung, dass eine MessageID nicht wiederverwendet werden darf, falls sie für einen Methodenaufruf zur Erteilung eines Transportauftrags verwendet worden ist, auch wenn dieser als Ergebnis nur zu einer Exception durch den Sender geführt hat (*4.4.1.6.2 MessageID*).

Klarstellungen bzw. redaktionelle Korrekturen, so dass klar ist, dass die MessageID einen Transportauftrag identifiziert. Der wiederum über ihre ID einer fachlichen Nachricht zugeordnet ist; nicht aber die MessageID direkt eine fachliche Nachricht identifiziert (besonders *4.4.2.3.1.5 MsgIdentification*; ansonsten diverse wie u.a. *4.4.1.4.2 Operation cancelMessage* und *4.4.3.1 Methode getStatusList*).

CR-2014-08 Exception Client-Reaktion

Im Abschnitt *4.6.2 Fehlernummern* Text eingefügt darüber, wann nach Exception ein neuer, automatisierter Sendeversuch durch den XTA-Client sinnvoll ist.

CR-2014-09 Ergebnisliste Löschfrist

Redaktionelle Überarbeitung Abschnitt *4.4.3.1 Methode getStatusList*. Satz zu Löschfrist gestrichen, der hier zu Unklarheiten geführt hat.

CR-2014-10 parallele Leser-Clients

Redaktionelle Überarbeitung: Unklarheiten beseitigt im Abschnitt *4.3.2.1 Asynchroner Empfang von Nachrichten* zum Thema paralleler Zugriff durch mehrere Leser-Clients auf ein Postfach.

CR-2014-11 ContentType

Dokumentation ergänzt zum Element *Message* und zum Attribut *contentType* (*4.5.1.4 ContentType* und *4.5.2.1 GenericContentContainer*).

CR-2014-12 Autorisierung

Der Text des letzten Absatzes aus Abschnitt 4.2.2 wurde korrigiert, d.h. inhaltlich auf die aktuellen Objekte aus XTA-WS bezogen, nicht wie bisher fälschlicherweise auf XTA-WS in der früheren Version 1.1.1. *Kapitel 4 Spezifikation des XTA-Webservice*.

CR-2014-13 diverse redaktionelle Korrekturen

Diverse offensichtliche redaktionelle Fehler korrigiert in *Kapitel 4 Spezifikation des XTAWebservice*.

CR-2014-14 Schemavalidierung

Im Rollenmodell wurde die Möglichkeit für Autor bzw. Leser eingetragen, die Durchführung der Schemavalidierung zu delegieren an Sender bzw. Empfänger: Jetzt ist das Rollenmodell in dieser Hinsicht konsistent mit den entsprechenden Aspekten der Darstellung in den Anwendungsfälle. Editiert wurden in Abschnitt 2.2.2, „Die Rollen“, die Abschnitte A 1.3 (so dass konsistent mit Abschnitt 2.2.4 UC Payload vorbereiten) und D 1.3 (so dass konsistent mit Abschnitt 2.2.5 UC Nachricht auswerten).

CR-2015-01 XTA-Dokumentation an OSCI 2.0.2 anpassen

Der Standard OSCI 2 wurde in der neuen Version 2.0.2 eingebunden, Namespaces und Präfixe aktualisiert. Durchgehend wurden die Abbildungen und alle Referenzen in der Dokumentation entsprechend aktualisiert. Betroffen sind besonders die Abschnitte *4.4.1.6 Wichtige Objekte der managementPort-Schnittstelle*, *4.4.2.3 Wichtige Objekte der sendPort-Schnittstelle*, *4.4.3.4 Wichtige Objekte der OSCI-MsgBox-Schnittstelle* und *F.1 OSCI-Transport-V2.0.2*

CR-2015-02 Management Summary aktualisieren

Die einleitenden Abschnitte *Überblick über das Dokument*, *Mitwirkung* und *Kapitel 1 Einleitung: Projektauftrag und Ergebnisse* wurden überarbeitet. Die neu gefassen Abschnitte sind Kapitel 1, *Einleitung*, und Anhang C, *Mitwirkende*.

CR-2015-06 Gliederung und Grundlegende Begriffe

Die Gliederung des Gesamtdokuments war veraltet und entsprach nicht mehr dem Stand der Qualitätsansprüche an die Dokumentation von XÖV-Standards. Sie wurde entsprechend umgestellt und konsolidiert. Die neuen Abschnitte wurden auf konsistente Weise integriert.

D.3 Release XTA-WS 2.0 (23.08.2013)

- Überarbeitung des WS mit dem Ziel der OSCI 2 Profilierung
- Authentifizierung und Autorisierung geändert

Einführung von PortTypes (managementPortType, sgBoxPortType, sendPortType)

- Methode IsServiceAvailable in lookupService umbenannt
- Methode getTransportreport überarbeitet
- Methode cancelMessaeg neu aufgenommen
- Methode createMessageId neu aufgenommen
- Methode sendMessage überarbeitet
- Methode sendMessageSync überarbeitet
- Methode getMessageIdList in getStatusList umbenannt
- Methode getNextMessage neu aufgenommen
- Methode getNextStatusList neu aufgenommen
- Methode close neu aufgenommen

Parameter und Returnwerte geändert

- umbenannt in ServiceType
- Präfix entfällt
- neu X509TokenContainer
- umbenannt in ReaderIdentifier
- Nachricht/NachrichtResponse zu GenericContantContainer geändert
- MessageID definition überarbeitet (Aufbau der uri korrigiert)
- Transportreport neue Struktur
- Input/Output Struktur von lookupService geändert
- MessageMetaData Container neu hinzugefügt
- neu MsgBoxStatusListRequestType
- neu MsgBoxRequestType
- neu MsgSelector
- neu MsgStatusList
- neu MsgBoxResponseType
- neu MsgBoxCloseRequestType
- Exceptions von SOAP 1.1 auf SOAP 1.2 umgestellt
- Technische Schnittstelle entfernt
- Exceptions geändert
- Adressierung angepasst (Partyidentifier)
- MessageMetaData notbefore / obsoleteafter neu hinzugefügt
- PropertyType geändert (Property Listen werden nach XÖV-Muster als Codelisten strukturiert)
- MessageMetaData in OSCI 2.01 ausgelagert und mit P23R abgestimmt

Synchrone Schnittstelle für den Leser neu aufgenommen (sendSynchronPortType)

- Methode sendMessageSync neu aufgenommen

D.4 Release XTA-WS 1.1 (18.09.2011)

Erstes finales Release der Schnittstellenbeschreibung XTA-WS.