



Koordinierungsstelle
für IT-Standards

XTA-Konformitätsvorgaben zur Umsetzung von XTA 2 Version 3

© 2021 Koordinierungsstelle für IT-Standards

Dokumentenversion 1.0

veröffentlicht unter

https://www.xoev.de/sixcms/media.php/13/XTA_Konformitaetsvorgaben_Produkt_XTA2V3.pdf

Änderungshistorie

Version	Datum	Autor	Kapitel	Änderungen
1.0	12.05.2021	KoSIT	Alle	Erstellung

Inhaltsverzeichnis

1	Einleitung	4
2	Umfang, Bewertungsgegenstand und Erklärung der Konformität	4
	2.1. Gegenstand der Konformitätsvorgaben.....	4
	2.2. Gegenstand der Konformitätsbewertung	4
	2.3. Inhalt der Konformitätserklärung und der Umgang damit.....	5
3	Konformitätsvorgaben.....	5
	3.1. Grundlegende Methoden.....	5
	3.1.1. Methodenvorgaben für die Rolle Autor	6
	3.1.2. Methodenvorgaben für die Rolle Sender	6
	3.1.3. Methodenvorgaben für die Rolle Empfänger	6
	3.1.4. Methodenvorgaben für die Rolle Leser.....	7
	3.2. Ein- und Ausgabeparametern der Methoden	7
	3.3. Parameterinhalt	8
4	Nachweis der Konformität.....	13
5	Ausblick	13

1 Einleitung

Der Transportadapter XTA unterstützt über den grundlegenden Versand und Empfang von Nachrichten hinaus auch weitergehende Methoden, wie die Vorgabe von Servicequalitäten mittels Service-Profilen, den parallelen Abruf von Nachrichten, Notifikationen oder den zeitgesteuerten Versand. Diese Methoden decken spezifische Bedarfe ab und werden dementsprechend nur dann umgesetzt, wenn sie benötigt werden. Diese bedarfsorientierte Umsetzung des Transportadapters XTA führt jedoch zu Produkten, die über einen unterschiedlichen Funktionsumfang verfügen und folglich nicht interoperabel sind. Interoperable Umsetzungen sind jedoch genau das Ziel der XTA-Spezifikation. Die vorliegenden Konformitätsvorgaben unterstützen die Interoperabilität der XTA-Umsetzungen, in dem die grundlegenden Methoden mit ihren Parametern erfasst und als minimale Umsetzung der XTA-Spezifikation vorgegeben werden.

2 Umfang, Bewertungsgegenstand und Erklärung der Konformität

2.1. Gegenstand der Konformitätsvorgaben

Die hier formulierten Konformitätsvorgaben sind Vorgaben für die Umsetzung der

„Spezifikation XTA 2 Version 3 31. Januar 2017 / final“

im Folgenden kurz als „XTA-Spezifikation“ bezeichnet und verfügbar unter:

https://www.xoev.de/sixcms/media.php/13/XTA_2_Version_3_Spezifikation_31012017.pdf.

Das vorliegende Dokument legt den Umfang fest, in dem die XTA-Spezifikation mindestens implementiert werden muss, um konform zur XTA-Spezifikation zu sein. Das Ziel der Konformitätsvorgaben ist es, die interoperable Implementierung des generischen Transportadapters zu fördern. Gegenstand sind die Methoden des Transportadapters, die für die Nutzung von Transportverfahren zum Versand und Empfang von Nachrichten durch Fachverfahren grundlegend erforderlich sind. Dies schließt die Parameter der Methoden ein.

Gegenstand der Konformitätsvorgaben ist ausschließlich der in diesem Dokument festgelegte Umfang von XTA. Über den hier definierten Umfang hinaus können weitere Teile der XTA-Spezifikation umgesetzt oder eigene Entwicklungen hinzugefügt werden, sofern dadurch das Verhalten des generischen Transportadapters ausschließlich erweitert und nicht verändert wird.

Die Konformitätsvorgaben beziehen sich ausschließlich auf die Implementierung von XTA, in diesem Dokument werden keine Vorgaben für den ordnungsgemäßen und sicheren Betrieb von XTA getroffen.

2.2. Gegenstand der Konformitätsbewertung

Gegenstand der Konformitätsbewertung ist ein Software-Produkt, welches den Transportadapter für die Anwendungsebene oder für die Transportebene implementiert. Eine Implementierung für die Anwendungsebene ist genau dann „konform zur Spezifikation XTA 2 Version 3“, wenn alle allgemeinen Anforderungen an die Anwendungsebene und zusätzlich alle Anforderungen an die Rollen erfüllt werden, die durch die Implementierung abgebildet werden (nur Autor, nur Leser oder Autor und Leser). In gleicher Weise muss eine Implementierung für die Transportebene alle allgemeinen Anforderungen an die Transportebene und zusätzlich alle Anforderungen an die umgesetzten Rollen der Transportebene abbilden (nur Sender, nur Empfänger oder Sender und Empfänger). Für jede Rolle ist anzugeben, welche Kommunikationsart (asynchron oder synchron) unterstützt wird, z.B. „Leser (asynchron)“ für die Implementierung der XTA-Rolle Leser für

asynchrone Kommunikation entsprechend dieser Vorgaben oder „Leser (asynchron), Leser (synchron)“ für eine Implementierung der Rolle Leser, die sowohl die Vorgaben für die asynchrone als auch synchrone Kommunikation umsetzt.

2.3. Inhalt der Konformitätserklärung und der Umgang damit

Die Konformitätserklärung ist eine freiwillige Selbsterklärung der Herstellerin eines Produktes, das einen Transportadapter XTA umsetzt. Die Herstellerin erklärt, dass die Umsetzung von XTA in diesem Produkt alle Vorgaben dieses Dokumentes erfüllt. Die Selbsterklärung muss folgende Informationen in einem eigenständigen Dokument enthalten:

- Eindeutige Kennzeichnung als Selbsterklärung zur XTA-Konformität
- Eindeutiger Verweis auf die XTA-Spezifikation mit Angabe von Version und Datum, die umgesetzt wurde (<XTA-Spezifikation>)
 - Verweis auf die Quelle der Spezifikation (Link zur KoSIT-Seite)
 - Es kann stets nur auf die XTA-Spezifikation verwiesen werden, nicht auf die Hilfsmittel, die zusammen mit der XTA-Spezifikation veröffentlicht werden (z.B. einzelne Schema- oder WSDL-Dateien).
- Eindeutiger Verweis auf das Dokument „XTA-Konformitätsvorgaben“ einschließlich Versions und Datumsangabe (<XTA-Konformitätsvorgaben>)
 - Verweis auf die Quelle des Dokumentes (Link zur KoSIT-Seite)
- Eindeutige Identifikation des Produktes, insbesondere die Produktbezeichnung einschließlich Versionsangabe (<Produkt>)
 - Angabe, welche Rollen umgesetzt worden sind
- Eindeutige Angabe der Herstellerin einschließlich gültiger Kontaktinformationen (<Herstellerin>)
- Eine Erklärung nach folgendem Muster:
 - „Selbsterklärung zur XTA-Konformität
Das Produkt <Produkt> der Herstellerin <Herstellerin> setzt die <XTA-Spezifikation> entsprechend der <XTA-Konformitätsvorgaben> für die XTA-Rollen {Autor (asynchron), Autor (synchron) | Leser (asynchron), Leser (synchron) | Sender (asynchron), Sender (synchron) | Empfänger (asynchron), Empfänger (synchron)} um.“
- Datum, Ort und Unterschrift einer vertretenden Person des Herstellers bzw. der Herstellerin

Die Korrektheit der freiwilligen Selbsterklärung wird nicht geprüft. Die KoSIT bietet an, eine eingescannte PDF-Version der Erklärung auf den Internetseiten der KoSIT mit Einwilligung der Herstellerin zu veröffentlichen. Die KoSIT behält sich dabei das Recht vor, veröffentlichte Erklärungen wieder von ihrer Internetseite zu entfernen, wenn sie ihrer eigenen Einschätzung nach ausreichende Kenntnis darüber erhält, dass eine Selbsterklärung unrichtig ist.

3 Konformitätsvorgaben

Alle folgenden Vorgaben MÜSSEN umgesetzt werden, es gibt keine SOLL- oder KANN-Vorgaben.

3.1. Grundlegende Methoden

Für einen Versand bzw. Empfang von Nachrichten über den Transportadapter XTA muss ein grundlegender Funktionsumfang verfügbar sein. Der Funktionsumfang einer XTA-Implementierung wird über die darin umgesetzten Methoden bestimmt. Die nachfolgenden vier Unterkapitel listen für alle XTA-Rollen auf, in welchem PortType welche Methoden implementiert und welche Methoden bei anderen Rollen aufgerufen können werden müssen,

um den grundlegenden Funktionsumfang des Transportadapters anzubieten. Durch die rollenbezogene Auflistung werden Vorgaben gegebenenfalls mehrfach aufgeführt, PortTypes und Methoden müssen jedoch nur genau einmal implementiert werden.

3.1.1. Methodenvorgaben für die Rolle Autor

Für die Rolle Autor sind für die Kommunikationsszenarien synchron und asynchron keine Methoden zu implementieren. In der Rolle Autor müssen die bereitgestellten Methoden der Rolle Sender aus dem Kapitel 3.1.2 gemäß dem zu unterstützendem Kommunikationsszenario aufgerufen werden können.

3.1.2. Methodenvorgaben für die Rolle Sender

Für die Rolle Sender sind für die Kommunikationsszenarien synchron und asynchron die in der Tabelle 1 aufgelisteten Methoden zu implementieren. In der Rolle Sender müssen keine Methoden der anderen Rollen aufgerufen werden können.

Tabelle 1 Vorgabe der zu implementierenden Methoden für die XTA-Rolle Sender (asynchron) und Sender (synchron)

Für die XTA-Rolle...	ist im PortType...	die Methode ... zu implementieren
Transportebene		
Sender (asynchron)	managementPortType	checkAccountActive
		lookupService
		getTransportReport
		createMessageld
	sendPortType	sendMessage
Sender (synchron)	managementPortType	checkAccountActive
		lookupService
		getTransportReport
		createMessageld
	sendPortType	sendMessageSync

3.1.3. Methodenvorgaben für die Rolle Empfänger

Für die Rolle Empfänger sind für die Kommunikationsszenarien synchron und asynchron die in der Tabelle 2 aufgelisteten Methoden zu implementieren. In der Rolle Empfänger müssen die bereitgestellten Methoden der Rolle Leser aus dem Kapitel 3.1.4 gemäß dem zu unterstützendem Kommunikationsszenario aufgerufen werden können.

Tabelle 2 Vorgabe der zu implementierenden Methoden für die XTA-Rolle Empfänger (asynchron) und Empfänger (synchron)

Für die XTA-Rolle...	ist im PortType...	die Methode ... zu implementieren
Transportebene		
Empfänger (asynchron)	managementPortType	checkAccountActive
		getTransportReport

	msgBoxPortType	getStatusList
		getMessage
		close
Empfänger (synchron)	managementPortType	checkAccountActive
		getTransportReport

3.1.4. Methodenvorgaben für die Rolle Leser

Für die Rolle Leser sind für die Kommunikationsszenarien synchron und asynchron die in der Tabelle 3 aufgelisteten Methoden zu implementieren. In der Rolle Leser müssen die bereitgestellten Methoden der Rolle Empfänger aus dem Kapitel 3.1.3 gemäß dem zu unterstützendem Kommunikationsszenario aufgerufen werden können.

Tabelle 3 Vorgabe der zu implementierenden Methoden für die XTA-Rolle Leser (asynchron) und Leser (synchron)

Für die XTA-Rolle...	ist im PortType...	die Methode ... zu implementieren
Anwendungsebene		
Leser (asynchron)	-	-
Leser (synchron)	sendSynchronPortType	sendMessageSync

Legende: „-“ = keine Vorgabe

3.2. Ein- und Ausgabeparametern der Methoden

Das Verhalten einer Methode wird über ihre Parameter gesteuert. Dabei wird zwischen Parametern zur Steuerung des Ablaufs (Steuerungsdaten, Header-Parameter) und Inhaltsdaten (Prozessdaten, Body-Parameter) unterschieden. In der nachfolgende

Tabelle 4 wird für jede Methode festgelegt, welche Ein- sowie Ausgabeparameter mindestens unterstützt werden müssen und ob es sich um Steuerungsdaten oder Inhaltsdaten handelt. Diese Parameter müssen wie spezifiziert umgesetzt und mit der zugehörigen Funktionalität hinterlegt werden. Der genaue Inhalt der Parameter wird im später folgenden Abschnitt Parameterinhalt vorgegeben.

Tabelle 4 Vorgaben zu Ein- und Ausgabeparametern

Die Methode...	...akzeptiert die Eingabewerte (Input)...		...liefert die Ausgabewerte (Output)...	
	als Steuerungsdaten (Header)	als Inhaltsdaten (Body)	als Steuerungsdaten (Header)	als Inhaltsdaten (Body)
checkAccountActive	AuthorIdentifier : Author	EmptyBody	-	EmptyBody
lookupService	AuthorIdentifier : Author	LookupServiceRequest : LookupServiceRequest	-	LookupServiceResponse : LookupServiceResponse
createMessageld	AuthorIdentifier : Author	EmptyBody	-	MessageID : MessageID
sendMessage	X509TokenContainer :	GenericContainer : GenericContentContainer	-	EmptyBody

	X509TokenContainer , MessageMetaData : MessageMetaData			
sendMessageSync	X509TokenContainer : X509TokenContainer , MessageMetaData : MessageMetaData	GenericContainer : GenericContentContainer	MessageMetaData : MessageMetaData, X509TokenContainer : X509TokenContainer	GenericContainer : GenericContentContainer
getTransportReport	AuthorIdentifier : Author	MessageID : MessageID	-	GetTransportReportResponse : TransportReport
getStatusList	AuthorIdentifier : Author	FetchRequest : MsgBoxStatusListRequest	FetchResponseHeader : MsgBoxResponse	FetchResponse : MsgStatusList
getMessage	AuthorIdentifier : Author	FetchRequest: MsgBoxFetchRequest	FetchResponseHeader : MsgBoxResponse, MessageMetaData : MessageMetaData	GenericContainer : GenericContentContainer
close	AuthorIdentifier : Author	FetchRequest: MsgBoxCloseRequest	-	EmptyBody

Legende: „-“ = keine Vorgabe; EmptyBody = nicht explizit definiertes, leeres Element, das von der Methode akzeptiert und verarbeitet werden muss

3.3. Parameterinhalt

Der Leistungsumfang einer Methode wird durch den konkreten Inhalt der Ein- und Ausgabeparameter bestimmt. Die Ein- und Ausgabeparameter wurden in

Tabelle 4 vorgegeben, die nachfolgende Tabelle 5 legt den konkreten, minimalen Inhalt dieser Parameter fest. Minimaler Inhalt bedeutet auch hier, dass für die aufgeführten Strukturen nur die vorgegebenen Elemente aufgelistet werden. Die vorgegebenen Strukturen können über die hier getroffenen Vorgaben weitere Elemente enthalten, die hier nicht aufgeführt und damit auch nicht vorgegeben sind. Namensräume der Parameter werden genannt, sofern dies zur eindeutigen Kennzeichnung des Parameters beiträgt.

Tabelle 5 Vorgaben zum minimalen Parameterinhalt

Parameterinhalt (Minimum)	Typ oder kurze Beschreibung	Typ / weitere Erläuterung steht in der XTA-Spezifikation im Abschnitt...
<i>Leeres Element</i>		
EmptyBody	nicht explizit definiertes, leeres Element, das von der Methode akzeptiert und verarbeitet werden muss	-
<i>Codelisten</i>		
code.ServiceParameterType	Frei wählbare Codeliste, Empfehlung in Spezifikation	5.5.1.9
xta:Code.RecordType	Frei wählbare Codeliste, Empfehlung in Spezifikation	5.5.1.1
xta:Code.ReportType	Frei wählbare Codeliste, Empfehlung in Spezifikation	5.5.1.4

Parameterinhalt (Minimum)	Typ oder kurze Beschreibung	Typ / weitere Erläuterung steht in der XTA-Spezifikation im Abschnitt...
<i>Identifikation</i>		
oscimeta:PartyIdentifierType	erweitert xs:normalizedString	5.4.1.6.1
+ category	oscimeta:NonEmptyStringType	
+ type	oscimeta:NonEmptyStringType	
oscimeta:PartyType		5.4.1.6.1
+ Identifier	oscimeta:PartyIdentifierType	
oscimeta:Author	oscimeta:PartyType	
oscimeta:Reader	oscimeta:PartyType	
AuthorIdentifier	oscimeta:Author	5.4.1.6.1
MessageID	wsa:MessageID	5.4.1.6.2
OriginatorsType		5.4.2.3.1
+ Author	oscimeta:PartyType	
+ Sender	oscimeta:PartyType	
oscimeta:DestinationsType	DestinationsType	
DestinationsType		5.4.2.3.1
+ oscimeta:Reader		
oscimeta:MsgIdentificationType	MsgIdentificationType	
MsgIdentificationType		5.4.2.3.1
+ wsa:MessageID		
<i>Nachrichten, Dienste, Parameter</i>		
TypeOfBusinessScenario	TypeOfBusinessScenarioType	
TypeOfBusinessScenarioType	erweitert xs:anyURI	
+ IsReferenceParameter	xs:boolean	
osci:TypeOfBusinessScenario	TypeOfBusinessScenario	
xta:IsServiceAvailableValueType		5.5.1.8
+ ServiceIsAvailable	xs:boolean	
+ ServiceIsAvailableUnknown	xs:boolean	
xta:LookupServiceType		5.5.1.6
+ oscimeta:Reader		2.4.2
+ ServiceType	xs:anyURI	
xta:LookupServiceResultType	erweitert xta:LookupServiceType	5.5.1.7
+ oscimeta:Reader	aus LookupServiceType	2.4.2
+ ServiceType	aus LookupServiceType	
+ IsServiceAvailableValue	xta:IsServiceAvailableValueType	5.5.1.8
+ ServiceParameter		
++ ParameterType	xta:Code.ServiceParameterType	5.5.1.9
++ Resource	xs:base64Binary	

Parameterinhalt (Minimum)	Typ oder kurze Beschreibung	Typ / weitere Erläuterung steht in der XTA-Spezifikation im Abschnitt...
oscimeta:MessageType	MessageType	5.4.2.3.3.2
MessageType	erweitert oscimeta:KeyCodeType	5.4.2.3.3.2
+ payloadSchema	oscimeta:NonEmptyURIType	
<i>Selektion</i>		
osci:MsgSelector	MsgSelector	
MsgSelector		5.4.3.4.1
+ newEntry	xs:boolean	
+ wsa:MessageID		
+ MsgBoxEntryTimeFrom	xs:dateTime	
+ MsgBoxEntryTimeTo	xs:dateTime	
<i>Payloadattribute und Status</i>		
MsgStatuslistType		5.4.3.5.1.1.2.2
+ osci:MessageMetaData		siehe unten
MsgStatusList	MsgStatuslistType	siehe oben
osci:MsgStatusList	MsgStatusList	siehe oben
<i>Payloadstruktur</i>		
xta:ContentType	erweitert xs:base64Binary	5.5.1.10
+ contentType	oscimeta:NonEmptyStringType	2.4.2
+ encoding	oscimeta:NonEmptyStringType	2.4.2
+ filename	oscimeta:NonEmptyStringType	2.4.2
+ id	xs:ID	
+ size	xs:positiveInteger	
xta:GenericContentContainer		5.5.2.1
+ xenc:EncryptedData		
+ ContentContainer		
++ Message	xta:ContentType	siehe oben
++ Attachment	xta:ContentType	siehe oben
GenericContainer	xta:GenericContentContainer	siehe oben
<i>Reporting</i>		
xta:RecordType		5.5.1.1
+ Timestamp	xs:dateTime	
+ Code	xta:Code.RecordType	5.5.1.2
+ Reason	xs:string	
xta:MessageStatusType		5.5.1.5
+ Status	xs:integer	
+ ErrorList		
++ Error	xta:RecordType	siehe oben

Parameterinhalt (Minimum)	Typ oder kurze Beschreibung	Typ / weitere Erläuterung steht in der XTA-Spezifikation im Abschnitt...
+ WarnList		
++ Warning	xta:RecordType	siehe oben
+ InfoList		
++ Info	xta:RecordType	siehe oben
xta:AdditionalReportListType		5.5.1.3
+ Report		
++ Key	xta:Code.ReportType	5.5.1.4
++ Data	xs:base64binary	
xta:TransportReport		5.5.2.4
+ ReportTime	xs:dateTime	
+ XTAServerIdentity	xs:token	
+ MessageStatus	xta:MessageStatusType	5.5.1.5
+ oscimeta:MessageMetaData		siehe unten
+ AdditionalReports	xta:AdditionalReportListType	5.5.1.3
+ ds:Signature		2.4.6
<i>Transportauftrag</i>		
oscimeta:DeliveryAttributesType	DeliveryAttributesType	5.4.2.3.2.1
DeliveryAttributesType		5.4.2.3.2.1
+ Origin	xs:dateTime	
+ InitialSend	xs:dateTime	
+ Delivery	xs:dateTime	
+ InitialFetch	xs:dateTime	
oscimeta:OriginatorsType	OriginatorsType	siehe oben
oscimeta:QualifierType		5.4.2.3.3.2
QualifierType		5.4.2.3.3.2
+ Service	xs:anyURI	
+ BusinessScenario		
++ Defined	oscimeta:KeyCodeType	
++ Undefined	xs:normalizedString	
+ MessageType	oscimeta:MessageType	
oscimeta:MessageMetaData	MessageMetaData	5.4.2.3.1
MessageMetaData		5.4.2.3.1
+ DeliveryAttributes (-)	oscimeta:DeliveryAttributesType	5.4.2.3.2.1
+ Originators (-)	oscimeta:OriginatorsType	siehe oben
+ Destinations (-)	oscimeta:DestinationsType	siehe oben
+ MsgIdentification (-)	oscimeta:MsgIdentificationType	siehe oben
+ Qualifier (-)	oscimeta:QualifierType	siehe oben

Parameterinhalt (Minimum)	Typ oder kurze Beschreibung	Typ / weitere Erläuterung steht in der XTA-Spezifikation im Abschnitt...
+ MsgSize	xs:positiveInteger	
<i>Request / Response</i>		
LookupServiceRequest	xta:LookupServiceRequest	5.5.2.2
xta:LookupServiceRequest		5.5.2.2
+ LookupServiceRequestList		
++ LookupService	xta:LookupServiceType	5.5.1.6
LookupServiceResponse	xta:LookupServiceResponse	5.5.2.3
xta:LookupServiceResponse		5.5.2.3
+ LookupServiceResultList		
++ LookupServiceResult	xta:LookupServiceResultType	5.5.1.7
MsgBoxRequestType		5.4.3.2.2
+ osci:MsgSelector		
MsgBoxResponseType		5.4.3.2.2
+MsgBoxRequestID	xs:anyURI	
+NoMessageAvailable	xs:anyURI	
+ItemsPending	xs:nonNegativeInteger	
MsgBoxResponse	MsgBoxResponseType	5.4.3.2.2
osci:MsgBoxResponse	MsgBoxResponse	5.4.3.2.2
FetchResponseHeader	osci:MsgBoxResponse	
MsgBoxFetchRequest	erweitert MsgBoxRequestType	5.4.3.4.3
+ osci:MsgSelector	Aus MsgBoxRequestType	siehe oben
+ MsgPart	xs:NMTOKEN	
osci:MsgBoxFetchRequest	MsgBoxFetchRequest	5.4.3.4.3
FetchRequest (in Methode getMessage)	osci:MsgBoxFetchRequest	5.4.3.4.3
MsgBoxStatusListRequestType	erweitert MsgBoxRequestType	5.4.3.1.2
+ osci:MsgSelector	aus MsgBoxRequestType	siehe oben
+ ListForm	osci:ListFormEnum	
+ maxListItems	xs:positiveInteger	
MsgBoxStatusListRequest	MsgBoxStatusListRequestType	5.4.3.1.2
osci:MsgBoxStatusListRequest	MsgBoxStatusListRequest	5.4.3.1.2
FetchRequest (in Methode getStatusList)	osci:MsgBoxStatusListRequest	siehe oben
FetchResponse	osci:MsgStatusList	siehe oben
MsgBoxCloseRequestType		
+ MsgBoxRequestID	xs:anyURI	
+ LastMsgReceived	wsa:AttributedURIType	
MsgBoxCloseRequest	MsgBoxCloseRequestType	5.4.3.3.2
osci:MsgBoxCloseRequest	MsgBoxCloseRequest	5.4.3.3.2

Parameterinhalt (Minimum)	Typ oder kurze Beschreibung	Typ / weitere Erläuterung steht in der XTA-Spezifikation im Abschnitt...
FetchRequest (in Methode close)	osci:MsgBoxCloseRequest	siehe oben
GetTransportReportResponse	xta:TransportReport	5.4.1.3.1

Legende: „-“ = keine Vorgabe; „+“, „++“, „+++“, usw. = Einrückung zur Anzeige, dass es sich um ein Bestandteil der zuvor genannten Struktur handelt; „(-)“ = Nicht alle Teile dieser Struktur sind verpflichtend umzusetzen

4 Nachweis der Konformität

Die Selbsterklärung der Herstellerin muss auf einem Nachweis darüber basieren, dass die hier formulierten Konformitätsvorgaben im betreffenden Produkt umgesetzt wurden. Für diesen Nachweis gibt es keine Vorgaben, es wird jedoch folgender Umfang und Inhalt empfohlen:

- Festlegung umgesetzten XTA-Rolle bzw. –Rollen, zum Beispiel als kurze Beschreibung der Umsetzung im Produkt
- Dokumentation der diesem Dokument entnommenen Anforderungen für diese Rollen
 - Für jede dieser Anforderungen eine Auflistung aller Nachweise über die Umsetzung, zum Beispiel in Form einer Tabelle (Anforderung – Nachweise)
- Nachweis über die unterstützten Parameter, zum Beispiel in Form einer entsprechend dokumentierten Schemadatei
- Nachweis über die umgesetzten Methoden, zum Beispiel in Form einer entsprechend dokumentierten WSDL-Datei
- Nachweis über die umgesetzten Kommunikationsabläufe, zum Beispiel in Form entsprechend dokumentierter Durchlauftests

Der Nachweis wird nicht durch die KoSIT überprüft, er unterstützt jedoch im Bedarfsfall die Klärung der Frage, ob ein Produkt tatsächlich über die behauptete Eigenschaft „XTA-Konformität“ im Sinne dieser Vorgaben verfügt. Grundsätzlich trägt er auch zur Qualitätssicherung einer Umsetzung der XTA-Spezifikation bei.

5 Ausblick

Die hier beschriebenen Vorgaben sind ein erster Startpunkt zur Förderung der Interoperabilität zwischen verschiedenen XTA-Implementierungen. Im aktuellen Umfang adressieren sie ausschließlich einen Teilbereich des Transportadapters, die XTA-Webservices. Die Konformitätsvorgaben sind zwar die Grundlage für den Nachweis, dass die XTA-Spezifikation umgesetzt wurde, das Erbringen der Nachweise wird jedoch nicht unterstützt. In beiden Punkten können die vorliegenden Konformitätsvorgaben weiterentwickelt werden.

Mit dem Bericht TransportReport und den Service-Profilen werden zwei weitere Leistungsbereiche des Transportadapters spezifiziert. Sowohl für das Reporting als auch für die Service-Profile können eigene, spezifische Konformitätsvorgaben festgelegt werden. Diese Trennung berücksichtigt, dass die Service-Profile unabhängig von den Webservices verwendet werden können. Auch das Reporting kann auf andere Bereich übertragen werden, um dieselben Metadaten und dieselbe Berichtsstruktur zu verwenden, jedoch mit selbstdefinierten Ereignissen. Spezifische Konformitätsvorgaben würden die gezielte Pflege und Weiterentwicklung der Bereiche Webservices, Service-Profile und Reporting unterstützen.

Die Bereitstellung und Vorgabe von Testfällen, mit denen das Verhalten einer XTA-Implementation auf die Einhaltung der Konformitätsvorgaben geprüft und nachgewiesen

werden kann, würde die Interoperabilität der XTA-Implementierungen weiter fördern und wäre somit ebenfalls eine sinnvolle Ergänzung dieser Vorgaben.