



Koordinierungsstelle  
für IT-Standards

---

# XTA-Konformitätsvorgaben zur Umsetzung von XTA 2 Version 4

---

© 2020 Koordinierungsstelle für IT-Standards

**Dokumentenversion 1.0**

veröffentlicht unter

[https://www.xoev.de/sixcms/media.php/13/XTA\\_Konformitaetsvorgaben\\_Produkt\\_XTA2V4.pdf](https://www.xoev.de/sixcms/media.php/13/XTA_Konformitaetsvorgaben_Produkt_XTA2V4.pdf)

## Änderungshistorie

Version	Datum	Autor	Kapitel	Änderungen
1.0	10.12.2020	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
	3.2. Ein- und Ausgabeparametern der Methoden.....	7
	3.3. Parameterinhalt .....	8
4	Nachweis der Konformität.....	14
5	Ausblick .....	14

## 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 4 28. Juli 2020 / Final“

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

[https://www.xoev.de/sixcms/media.php/13/XTA\\_2\\_Version\\_4\\_Spezifikation\\_20200728.pdf](https://www.xoev.de/sixcms/media.php/13/XTA_2_Version_4_Spezifikation_20200728.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 4“, 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 folgenden vier Tabellen 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.

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

Für die XTA-Rolle...	ist im PortType...	die Methode ... zu implementieren	bei der XTA-Rolle...	die Methode ... aufrufen können
Anwendungsebene				
Autor (asynchron)	authorPortType	-	Sender	checkAccountActive
				lookupService
				createMessageID
				sendMessage
				getTransportReport
Autor (synchron)	authorPortType	-	Sender	checkAccountActive
				lookupService
				createMessageID
				sendMessageSync
				getTransportReport

Legende: „-“ = keine Vorgabe

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

Für die XTA-Rolle...	ist im PortType...	die Methode ... zu implementieren	bei der XTA-Rolle...	die Methode ... aufrufen können
Anwendungsebene				
Leser (asynchron)	readerPortType	-	Empfänger	checkAccountActive
				getStatusList
				getMessage
				close
				getTransportReport
Leser (synchron)	readerPortType	deliverMessage	Empfänger	checkAccountActive
				getTransportReport

Legende: „-“ = keine Vorgabe

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

Für die XTA-Rolle...	ist im PortType...	die Methode ... zu implementieren	bei der XTA-Rolle...	die Methode ... aufrufen können
Transportebene				
Sender (asynchron)	transportPortType	checkAccountActive	Autor	-
		lookupService		
		getTransportReport		
	senderPortType	createMessageID		
		sendMessage		
Sender (synchron)	transportPortType	checkAccountActive	Autor	-
		lookupService		
		getTransportReport		
	senderPortType	createMessageID		
		sendMessageSync		

Legende: „-“ = keine Vorgabe

Tabelle 4 Vorgabe der zu implementierenden und aufrufbaren Methoden für die XTA-Rolle Empfänger (asynchron) und Empfänger (synchron)

Für die XTA-Rolle...	ist im PortType...	die Methode ... zu implementieren	bei der XTA-Rolle...	die Methode ... aufrufen können
Transportebene				
Empfänger (asynchron)	transportPortType	checkAccountActive	Leser	-
		getTransportReport		
	recipientPortType	getStatusList		
		getMessage		
		close		
Empfänger (synchron)	transportPortType	checkAccountActive	Leser	deliverMessage
		getTransportReport		

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 5 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 5 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	XTAIdentifier : Identifier	EmptyBody	-	EmptyBody
lookupService	XTAIdentifier : Identifier	LookupServiceRequest : LookupServiceRequest	-	LookupServiceResponse : LookupServiceResponse
createMessageID	XTAIdentifier : Identifier	EmptyBody	-	MessageID : MessageID
sendMessage	XTAIdentifier : Identifier, MessageMetaData : MessageMetaData	GenericContainer : GenericContentContainer	-	EmptyBody
sendMessageSync	XTAIdentifier : Identifier, MessageMetaData : MessageMetaData	GenericContainer : GenericContentContainer	MessageMetaDataResponse : MessageMetaData	MessageResponseBody : MessageResponse
getTransportReport	XTAIdentifier : Identifier	MessageID : MessageID	-	GetTransportReportResponse : TransportReport
getStatusList	XTAIdentifier : Identifier	FetchRequest : MsgBoxStatusListRequest	FetchResponseHeader : MsgBoxResponse	FetchResponse : MsgStatusList
getMessage	XTAIdentifier : Identifier	FetchRequest: MsgBoxFetchRequest	FetchResponseHeader : MsgBoxResponse, MessageMetaData : MessageMetaData	GenericContainer : GenericContentContainer
deliverMessage	XTAIdentifier : Identifier, MessageMetaData : MessageMetaData	GenericContainer : GenericContentContainer	MessageMetaData : MessageMetaData	MessageResponseContentContainer: MessageResponse
close	XTAIdentifier : Identifier	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 5 vorgegeben, die nachfolgende Tabelle 6 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 6 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	2.2.1.11
xta:Code.RecordType	Frei wählbare Codeliste, Empfehlung in Spezifikation	2.2.1.2
xta:Code.ReportType	Frei wählbare Codeliste, Empfehlung in Spezifikation	2.2.1.5
<i>SecurityToken</i>		
<b>oscimeta:SecurityTokenEnum</b>		2.2.1.52
+ wsse:BinarySecurityToken		B.1
+ wsse:SecurityTokenReference		B.1
+ wsse:UsernameToken		B.1
<b>SecurityToken</b>		2.2.1.52
+ payloadRef	xs:IDREF	
+ usage	oscimeta:SecurityTokenEnum	2.2.1.52
<i>Identifikation</i>		
<b>oscimeta:PartyIdentifierType</b>	erweitert NonEmptyStringType	2.2.1.40
+ category	oscimeta:NonEmptyStringType	2.2.1.40
+ name	oscimeta:NonEmptyStringType	2.2.1.40
+ type	oscimeta:NonEmptyStringType	2.2.1.40
<b>oscimeta:PartyType</b>		2.2.1.44
+ Identifier	oscimeta:PartyIdentifierType	2.2.1.45
+ oscimeta:SecurityToken	SecurityToken	2.2.1.52
oscimeta:Identifier	oscimeta:PartyType	2.2.1.44
oscimeta:Reader	oscimeta:PartyType	2.2.1.50
XTAIdentifier	oscimeta:Identifier	2.2.1.34
MessageID	wsa:MessageID	B.2
<b>OriginatorsType</b>		2.2.1.42
+ Author	oscimeta:PartyType	2.2.1.44
+ Sender	oscimeta:PartyType	2.2.1.44
oscimeta:DestinationsType	DestinationsType	2.2.1.33
<b>DestinationsType</b>		2.2.1.33

Parameterinhalt (Minimum)	Typ oder kurze Beschreibung	Typ / weitere Erläuterung steht in der XTA-Spezifikation im Abschnitt...
+ oscimeta:Reader		2.2.1.50
oscimeta:MsgIdentificationType	MsgIdentificationType	2.2.1.39
<b>MsgIdentificationType</b>		2.2.1.39
+ wsa:MessageID		B.2
<i>Nachrichten, Dienste, Parameter</i>		
TypeOfBusinessScenario	TypeOfBusinessScenarioType	2.2.1.30
<b>TypeOfBusinessScenarioType</b>	erweitert xs:anyURI	
+ IsReferenceParameter	xs:boolean	
osci:TypeOfBusinessScenario	TypeOfBusinessScenario	2.2.1.29
<b>xta:IsServiceAvailableValueType</b>		2.2.1.10
+ ServiceIsAvailable	xs:boolean	
+ ServiceIsAvailableUnknown	xs:boolean	
<b>xta:LookupServiceType</b>		2.2.1.8
+ oscimeta:Reader		2.2.1.50
+ ServiceType	xs:anyURI	
<b>xta:LookupServiceResultType</b>	erweitert xta:LookupServiceType	2.2.1.9
+ oscimeta:Reader	aus LookupServiceType	2.2.1.50
+ ServiceType	aus LookupServiceType	
+ IsServiceAvailableValue	xta:IsServiceAvailableValueType	2.2.1.10
+ ServiceParameter		
++ ParameterType	xta:Code.ServiceParameterType	2.2.1.11
++ Resource	xs:base64Binary	
oscimeta:MessageType	MessageType	2.2.1.29
<b>MessageType</b>	erweitert xs:string	2.2.1.29
+ payloadSchema	oscimeta:NonEmptyURIType	2.2.1.41
<i>Selektion</i>		
<b>MsgSelectorType</b>		2.2.1.24
+ newEntry	xs:boolean	
+ wsa:MessageID		B.2
+ wsa:RelatesTo		B.2
+ MsgBoxEntryTimeFrom	xs:dateTime	
+ MsgBoxEntryTimeTo	xs:dateTime	
+ Extension	xs:anyType	
MsgSelector	MsgSelectorType	2.2.1.24
osci:MsgSelector	MsgSelector	2.2.1.23
<i>Payloadattribute und Status</i>		
<b>MsgAttributeListType</b>		2.2.1.13

Parameterinhalt (Minimum)	Typ oder kurze Beschreibung	Typ / weitere Erläuterung steht in der XTA-Spezifikation im Abschnitt...
+ wsa:MessageID		B.2
+ wsa:RelatesTo		B.2
+ wsa:From		B.2
+ osci:TypeOfBusinessScenario		siehe oben
+ MsgSize	xs:positiveInteger	
+ ObsoleteAfterDate	xs:dateTime	
+ DeliveryTime	xs:dateTime	
+ InitialFetchedTime	xs:dateTme	
osci:MsgAttributeListType	MsgAttributeListType	siehe oben
<b>MsgStatuslistType</b>		
+ MsgAttributes	osci:MsgAttributeListType	siehe oben
+ osci:MessageMetaData		siehe oben
MsgStatusList	MsgStatuslistType	2.2.1.28
osci:MsgStatusList	MsgStatusList	2.2.1.27
<i>Payloadstruktur</i>		
<b>xta:ContentType</b>	erweitert xs:base64Binary	2.2.1.12
+ contentDescription	oscimeta:NonEmptyStringType	2.2.1.40
+ contentType	oscimeta:NonEmptyStringType	2.2.1.40
+ encoding	oscimeta:NonEmptyStringType	2.2.1.40
+ filename	oscimeta:NonEmptyStringType	2.2.1.40
+ id	xs:ID	
+ lang	xs:language	
+ size	xs:positiveInteger	
<b>xta:GenericContentContainer</b>		2.2.2.1
+ xenc:EncryptedData		B.3
+ ContentContainer		
++ Message	xta:ContentType	siehe oben
++ Attachment	xta:ContentType	siehe oben
GenericContainer	xta:GenericContentContainer	siehe oben
<i>Reporting</i>		
<b>xta:RecordType</b>		2.2.1.1
+ Timestamp	xs:dateTime	
+ Code	xta:Code.RecordType	2.2.1.2
+ Reason	xs:string	
<b>xta:MessageStatusType</b>		2.2.1.7
+ Status	xs:integer	
+ ErrorList		

Parameterinhalt (Minimum)	Typ oder kurze Beschreibung	Typ / weitere Erläuterung steht in der XTA-Spezifikation im Abschnitt...
++ Error	xta:RecordType	siehe oben
+ WarnList		
++ Warning	xta:RecordType	siehe oben
+ InfoList		
++ Info	xta:RecordType	siehe oben
<b>xta:AdditionalReportListType</b>		2.2.1.4
+ Report		
++ Key	xta:Code.ReportType	2.2.1.5
++ Data	xs:base64binary	
<b>xta:TransportReport</b>		2.2.2.4
+ ReportTime	xs:dateTime	
+ XTAServerIdentity	xs:token	
+ MessageStatus	xta:MessageStatusType	2.2.1.7
+ oscimeta:MessageMetaData		2.2.1.36
AdditionalReports	xta:AdditionalReportListType	2.2.1.4
ds:Signature		B.4
<i>Transportauftrag</i>		
oscimeta:DeliveryAttributesType	DeliveryAttributesType	2.2.1.31
<b>DeliveryAttributesType</b>		2.2.1.31
+ Origin	xs:dateTime	
+ InitialSend	xs:dateTime	
+ Delivery	xs:dateTime	
+ InitialFetch	xs:dateTime	
+ Reception	xs:dateTime	
oscimeta:OriginatorsType	OriginatorsType	2.2.1.42
oscimeta:QualifierType		2.2.1.48
<b>QualifierType</b>		2.2.1.48
+ Service	xs:anyURI	
+ BusinessScenario		
++ Defined	oscimeta:KeyCodeType	2.2.1.35
++ Undefined	xs:normalizedString	
+ MessageType	oscimeta:MessageType	2.2.1.29
oscimeta:MessageMetaData	MessageMetaData	2.2.1.36
<b>MessageMetaData</b>		2.2.1.36
+ TestMsg	xs:boolean	
+ DeliveryAttributes (-)	oscimeta:DeliveryAttributesType	2.2.1.31
+ Originators (-)	oscimeta:OriginatorsType	2.2.1.42

Parameterinhalt (Minimum)	Typ oder kurze Beschreibung	Typ / weitere Erläuterung steht in der XTA-Spezifikation im Abschnitt...
+ Destinations (-)	oscimeta:DestinationsType	2.2.1.33
+ MsgIdentification (-)	oscimeta:MsgIdentificationType	2.2.1.39
+ Qualifier (-)	oscimeta:QualifierType	2.2.1.48
+ MsgSize	xs:positiveInteger	
<i>Request / Response</i>		
LookupServiceRequest	xta:LookupServiceRequest	2.2.2.2
<b>xta:LookupServiceRequest</b>		2.2.2.2
+ LookupServiceRequestList		
++ LookupService	xta:LookupServiceType	2.2.1.8
LookupServiceResponse	xta:LookupServiceResponse	2.2.2.3
<b>xta:LookupServiceResponse</b>		2.2.2.3
+ LookupServiceResultList		
++ LookupServiceResult	xta:LookupServiceResultType	2.2.1.9
<b>MsgBoxRequestType</b>		2.2.1.20
+ osci:MsgSelector		
<b>MsgBoxResponseType</b>		2.2.1.22
+MsgBoxRequestID	xs:anyURI	
+NoMessageAvailable	xs:anyURI	
+ItemsPending	xs:nonNegativeInteger	
MsgBoxResponse	MsgBoxResponseType	2.2.1.22
osci:MsgBoxResponse	MsgBoxResponse	2.2.1.21
FetchResponseHeader	osci:MsgBoxResponse	2.2.1.21
<b>MsgBoxFetchRequestType</b>	erweitert MsgBoxRequestType	2.2.1.20
+ osci:MsgSelector	Aus MsgBoxRequestType	2.2.1.20
+ MsgPart	xs:NMTOKEN	
MsgBoxFetchRequest	MsgBoxFetchRequestType	2.2.1.17
osci:MsgBoxFetchRequest	MsgBoxFetchRequest	2.2.1.16
FetchRequest (in Methode getMessage)	osci:MsgBoxFetchRequest	2.2.1.16
<b>xta:MessageResponse</b>		2.2.2.6
+ xta:GenericContentContainer		2.2.2.1
MessageResponseContentContainer	xta:MessageResponse	2.2.2.6
MessageMetaDataResponse	oscimeta:MessageMetaData	2.2.1.36
<b>MsgBoxStatusListRequestType</b>	erweitert MsgBoxRequestType	2.2.1.26
+ osci:MsgSelector	aus MsgBoxRequestType	siehe oben
+ ListForm	osci:ListFormEnum	
+ maxListItems	xs:positiveInteger	
MsgBoxStatusListRequest	MsgBoxStatusListRequestType	2.2.1.26

Parameterinhalt (Minimum)	Typ oder kurze Beschreibung	Typ / weitere Erläuterung steht in der XTA-Spezifikation im Abschnitt...
osci:MsgBoxStatusListRequest	MsgBoxStatusListRequest	2.2.1.25
FetchRequest (in Methode getStatusList)	osci:MsgBoxStatusListRequest	siehe oben
FetchResponse	osci:MsgStatusList	siehe oben
<b>MsgBoxCloseRequestType</b>		
+ MsgBoxRequestID	xs:anyURI	
+ LastMsgReceived	wsa:AttributedURIType	B.2
MsgBoxCloseRequest	MsgBoxCloseRequestType	2.2.1.15
osci:MsgBoxCloseRequest	MsgBoxCloseRequest	2.2.1.14
FetchRequest (in Methode close)	osci:MsgBoxCloseRequest	siehe oben
GetTransportReportResponse	xta:TransportReport	2.2.2.4

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 den Berichten TransportReport und ServiceReport sowie 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.