



Koordinierungsstelle
für IT-Standards

XTA-WS-Differenzen

von XTA 1.1.1 und XTA 2 Version 4

© 2021 Koordinierungsstelle für IT-Standards

Dokumentenversion 1.0

veröffentlicht unter

https://www.xoev.de/sixcms/media.php/13/XTA-Differenzen_XTA111_XTA2V4.pdf

Änderungshistorie

Version	Datum	Autor	Kapitel	Änderungen
1.0	26.02.2021	KoSIT	Alle	Erstellung

Inhaltsverzeichnis

1	Einleitung.....	4
2	Versionsdifferenzen.....	5
	2.1. Grundlegende Funktionen.....	5
	2.2. Ein- und Ausgabeparametern der Methoden	7
	2.3. Parameterinhalt.....	9
3	Ausblick	16

1 Einleitung

Um den Umstieg von der XTA 1.1.1 (weiter als „alte Version“) auf XTA 2 Version 4 (weiter als „neue Version“) zu unterstützen, werden nachfolgend dessen Unterschiede im XTA-Webservice dargestellt. Diese Darstellung fokussiert den Teil, der zur Erfüllung der Konformitätsvorgaben für die neue Version umgesetzt werden muss. Sie umfasst somit die grundlegend zu implementierenden Methoden, ihre Ein- und Ausgabeparameter sowie den Parameterinhalt in kurzer, tabellengestützter Form.

Die Spezifikation der „alten Version“ ist verfügbar unter:

<http://www.xoevta.de/download/XTA-WS-Unterlagen-111final.zip>

und der „neuen Version“ unter:

https://www.xoev.de/sixcms/media.php/13/XTA_2_Version_4_Spezifikation_20200728.pdf

Die Konformitätsvorgaben für die XTA-Implementierung der neuen Version sind unter folgendem Verweis abrufbar:

https://www.xoev.de/sixcms/media.php/13/XTA_Konformitaetsvorgaben_Produkt_XTA2V4.pdf

Als weitere Hilfestellung werden über die hellgrau (■) markierten Tabelleninhalte die Änderungen in den Spezifikationen und somit auch ein expliziter Änderungsbedarf der XTA-Implementierung hervorgehoben. Der Umfang der Funktionen und Parameterinhalte basiert auf den Konformitätsvorgaben und deckt somit nicht alles ab. Für die ausführlichen Details sind die oben genannten Dokumente zu berücksichtigen.

2 Versionsdifferenzen

2.1. Grundlegende Funktionen

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 Tabellen listen für alle XTA-Rollen auf, in welchem PortType welche Methoden implementiert werden müssen, um die grundlegenden Funktionen des Transportadapters anzubieten. Durch die rollenbezogene Auflistung werden Vorgaben gegebenenfalls mehrfach aufgeführt, PortTypes und Methoden müssen jedoch nur genau einmal implementiert werden. Die Tabellen beinhalten die Verweise „(z.z..)“ auf die Kapitel der Spezifikation von der neuen Version worin die Funktionen detailliert beschrieben werden.

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

Alte Version	Neue Version		
Methode	für die XTA-Rolle...	ist im PortType...	die Methode ... zu implementieren
	<i>Anwendungsebene</i>		
-	Autor (asynchron)	authorPortType	-
-	Autor (synchron)	authorPortType	-

Legende: „-“ = keine Vorgabe

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

Alte Version	Neue Version		
Methode	für die XTA-Rolle...	ist im PortType...	die Methode ... zu implementieren
	<i>Anwendungsebene</i>		
-	Leser (asynchron)	readerPortType	-
-	Leser (synchron)	readerPortType	deliverMessage (2.1.2.2.1)

Legende: „-“ = keine Vorgabe

XTA-WS-Differenzen von XTA 1.1.1 und XTA 2 Version 4

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

Alte Version	Neue Version		
Methode	für die XTA-Rolle...	ist im PortType...	die Methode ... zu implementieren
	<i>Transportebene</i>		
CheckAccountActive	Sender (asynchron)	transportPortType	checkAccountActive (2.1.1.1.1)
IsServiceAvailable			lookupService (2.1.1.1.4)
GetTrasportReport			getTransportReport (2.1.1.1.3)
- (Umweg über SendMessage)		senderPortType	createMessageID (2.1.1.2.2)
SendMessage			sendMessage (2.1.1.2.3)
CheckAccountActive	Sender (synchron)	transportPortType	checkAccountActive
IsServiceAvailable			lookupService
GetTrasportReport			getTransportReport
- (Umweg über SendMessageSync)		senderPortType	createMessageID
SendMessageSync			sendMessageSync (2.1.1.2.4)

Legende: „-“ = keine Vorgabe

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

Alte Version	Neue Version		
Methode	für die XTA-Rolle...	ist im PortType...	die Methode ... zu implementieren
	<i>Transportebene</i>		
CheckAccountActive	Empfänger (asynchron)	transportPortType	checkAccountActive
GetTrasportReport			getTransportReport
GetMessageldList		recipientPortType	getStatusList (2.1.1.3.5)
GetMessage			getMessage (2.1.1.3.2)
-			close (2.1.1.3.1)
CheckAccountActive	Empfänger (synchron)	transportPortType	checkAccountActive
GetTrasportReport			getTransportReport

Legende: „-“ = keine Vorgabe

2.2. 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 nachfolgenden 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 Gegenüberstellung der Methoden und deren Eingabewerte/Rückgabewerte

Methode	Input / Output	Parameter in der alten Version	Parameter in der neuen Version	Hinweise
checkAccountActive (alt: CheckAccountActive)	Input (Header)	XtaAccountIdentifikation : CheckAccountActive- XtaAccountIdentifikation	XTAIdentifier : Identifier	
	Input (Body)	parameters : CheckAccountActive	EmptyBody	
	Output (Header)	-	-	
	Output (Body)	parameters : CheckAccountActiveResponse	EmptyBody	
lookupService (alt: IsServiceAvailable)	Input (Header)	XtaAccountIdentifikation : IsServiceAvailable- XtaAccountIdentifikation	XTAIdentifier : Identifier	
	Input (Body)	parameters : IsServiceAvailable	LookupServiceRequest : LookupServiceRequest	
	Output (Header)	-	-	
	Output (Body)	parameters : IsServiceAvailable- Response	LookupServiceResponse : LookupServiceResponse	
createMessageID (alt: Umweg über SendMessage/-Sync)	Input (Header)	-	XTAIdentifier : Identifier	
	Input (Body)	-	GenericContainer : GenericContentContainer	
	Output (Header)	-	-	
	Output (Body)	-	EmptyBody	
sendMessage (alt: SendMessage)	Input (Header)	XtaAccountIdentifikation : SendMessage- XtaAccountIdentifikation	XTAIdentifier : Identifier, MessageMetaData : MessageMetaData	
	Input (Body)	parameters : SendMessage	GenericContainer : GenericContentContainer	
	Output (Header)	-	-	
	Output (Body)	parameters : SendMessageResponse	EmptyBody	

XTA-WS-Differenzen von XTA 1.1.1 und XTA 2 Version 4

Methode	Input / Output	Parameter in der alten Version	Parameter in der neuen Version	Hinweise
sendMessageSync (alt: SendMessageSync)	Input (Header)	XtaAccountIdentifikation : SendMessageSync- XtaAccountIdentifikation	XTAIdentifier : Identifier, MessageMetaData : MessageMetaData	
	Input (Body)	parameters : SendMessageSync	GenericContainer : GenericContentContainer	
	Output (Header)	-	MessageMetaDataResponse : MessageMetaData	
	Output (Body)	parameters : SendMessageSyncResponse	MessageResponseBody : MessageResponse	
getTransportReport (alt: GetTrasportReport)	Input (Header)	XtaAccountIdentifikation : GetTransportReport- XtaAccountIdentifikation	XTAIdentifier : Identifier	
	Input (Body)	parameters : GetTransportReport	MessageID : MessageID	
	Output (Header)	-	-	
	Output (Body)	parameters : GetTransportReportResponse	GetTransportReportResponse : TransportReport	
getStatusList (alt: GetMessageldList)	Input (Header)	XtaAccountIdentifikation : GetMessageldList- XtaAccountIdentifikation	XTAIdentifier : Identifier	
	Input (Body)	parameters : GetMessageldList	FetchRequest : MsgBoxStatusListRequest	
	Output (Header)	-	FetchResponseHeader : MsgBoxResponse	
	Output (Body)	parameters : GetMessageldListResponse	FetchResponse : MsgStatusList	
getMessage (alt: GetMessage)	Input (Header)	XtaAccountIdentifikation : GetMessage- XtaAccountIdentifikation	XTAIdentifier : Identifier	
	Input (Body)	parameters : GetMessage	FetchRequest: MsgBoxFetchRequest	
	Output (Header)	-	FetchResponseHeader : MsgBoxResponse, MessageMetaData : MessageMetaData	
	Output (Body)	parameters : GetMessageResponse	GenericContainer : GenericContentContainer	
deliverMessage (alt: -)	Input (Header)		XTAIdentifier : Identifier, MessageMetaData : MessageMetaData	
	Input (Body)		GenericContainer : GenericContentContainer	
	Output (Header)		MessageMetaData : MessageMetaData	
	Output (Body)		MessageResponseContentContainer: MessageResponse	

Methode	Input / Output	Parameter in der alten Version	Parameter in der neuen Version	Hinweise
close (alt: -)	Input (Header)	-	XTAIdentifier : Identifier	
	Input (Body)	-	FetchRequest: MsgBoxCloseRequest	
	Output (Header)	-	-	
	Output (Body)	-	EmptyBody	

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

2.3. Parameterinhalt

Der Leistungsumfang einer Methode wird durch den konkreten Inhalt der Ein- und Ausgabeparameter bestimmt. Die Ein- und Ausgabeparameter werden in den folgenden Tabellen verglichen, sie legen 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 zu Ein- und Ausgabeparametern

Parameterinhalt in der alten Version	Typ oder kurze Beschreibung	Parameterinhalt in der neuen Version	Typ oder kurze Beschreibung	Hinweis
		<i>Leeres Element</i>		
-		EmptyBody	nicht explizit definiertes, leeres Element, das von der Methode akzeptiert und verarbeitet werden muss	
		<i>Codelisten</i>		
		code.ServiceParameter-Type	Frei wählbare Codeliste, Empfehlung in Spezifikation	
		xta:Code.RecordType	Frei wählbare Codeliste, Empfehlung in Spezifikation	
		xta:Code.ReportType	Frei wählbare Codeliste, Empfehlung in Spezifikation	
		<i>SecurityToken</i>		
		oscimeta:Security-TokenEnum		
		+ wsse:BinarySecurity-Token		

XTA-WS-Differenzen von XTA 1.1.1 und XTA 2 Version 4

		+ wsse:SecurityToken-Reference		
		+ wsse:UsernameToken		
		SecurityToken		
		+ payloadRef	xs:IDREF	
		+ usage	oscimeta:Security-TokenEnum	
		<i>Identifikation</i>		
EmpfaengerKennung, AbsenderKennung	xs:string	oscimeta:Party-IdentifierType	erweitert NonEmptyStringType	
EmpfaengerPraefix, AbsenderPraefix	xs:string	+ category	oscimeta:NonEmpty-StringType	
		+ name	oscimeta:NonEmpty-StringType	
		+ type	oscimeta:NonEmpty-StringType	
		oscimeta:PartyType		
		+ Identifier	oscimeta:PartyIdentifierType	
		+ oscimeta:Security-Token	SecurityToken	
		oscimeta:Identifier	oscimeta:PartyType	
		oscimeta:Reader		
		XTAIdentifier	oscimeta:Identifier	
MessageID	wsa:AttributedURIType	MessageID	wsa:MessageID	
		OriginatorsType		
		+ Author	oscimeta:PartyType	
		+ Sender	oscimeta:PartyType	
		oscimeta:Destinations-Type	DestinationsType	
		DestinationsType		
		+ oscimeta:Reader		
		oscimeta:Msg-IdentificationType	MsgIdentificationType	
		MsgIdentificationType		
		+ wsa:MessageID		
		<i>Nachrichten, Dienste, Parameter</i>		
		TypeOfBusiness-Scenario	TypeOfBusiness-ScenarioType	
		TypeOfBusiness-ScenarioType	erweitert xs:anyURI	
		+ IsReferenceParameter	xs:boolean	

		osci:TypeOfBusiness-Scenario	TypeOfBusiness-Scenario	
		xta:IsServiceAvailable-ValueType		
		+ ServiceIsAvailable	xs:boolean	
		+ ServiceIsAvailable-Unknown	xs:boolean	
		xta:LookupServiceType		
		+ oscimeta:Reader		
		+ ServiceType	xs:anyURI	
		xta:LookupService-ResultType	erweitert xta:LookupServiceType	
		+ oscimeta:Reader	aus LookupServiceType	
		+ ServiceType	aus LookupServiceType	
		+ IsServiceAvailable-Value	xta:IsServiceAvailable-ValueType	
		+ ServiceParameter		
		++ ParameterType	xta:Code.Service-ParameterType	
		++ Resource	xs:base64Binary	
		oscimeta:MessageType	MessageType	
		MessageType	Erweitert xs:string	
		+ payloadSchema	oscimeta:NonEmpty-URIType	
		<i>Selektion</i>		
		MsgSelectorType		
		+ newEntry	xs:boolean	
		+ wsa:MessageID		
		+ wsa:RelatesTo		
		+ MsgBoxEntryTime-From	xs:dateTime	
		+ MsgBoxEntryTimeTo	xs:dateTime	
		+ Extension	xs:anyType	
		MsgSelector	MsgSelectorType	
		osci:MsgSelector	MsgSelector	
		<i>Payloadattribute und Status</i>		
		MsgAttributeListType		
		+ wsa:MessageID		
		+ wsa:RelatesTo		
		+ wsa:From		
		+ osci:TypeOfBusiness-Scenario		

XTA-WS-Differenzen von XTA 1.1.1 und XTA 2 Version 4

		+ MsgSize	xs:positiveInteger	
		+ ObsoleteAfterDate	xs:dateTime	
		+ DeliveryTime	xs:dateTime	
		+ InitialFetchTime	xs:dateTime	
		osci:MsgAttributeList-Type	MsgAttributeListType	
		MsgStatuslistType		
		+ MsgAttributes	osci:MsgAttributeList-Type	
		+ osci:MessageMeta-Data		
		MsgStatusList	MsgStatuslistType	
		osci:MsgStatusList	MsgStatusList	
		<i>Payloadstruktur</i>		
NachrichtType	Erweiterung von xs:base64Binary	xta:ContentType	erweitert xs:base64Binary	
		+ contentDescription	oscimeta:NonEmpty-StringType	
		+ contentType	oscimeta:NonEmpty-StringType	
		+ encoding	oscimeta:NonEmpty-StringType	
+ filename		+ filename	oscimeta:NonEmpty-StringType	
		+ id	xs:ID	
		+ lang	xs:language	
+ size	xs:string	+ size	xs:positiveInteger	
+ read-date	xs:string	-		
+ modification-date	xs:string	-		
+ creation-date	xs:string	-		
		xta:GenericContent-Container		
		+ xenc:EncryptedData		
		+ ContentContainer		
Nachricht	NachrichtType	++ Message	xta:ContentType	Mit GenericContent-Container ist nun auch der Versand von Attachments möglich
		++ Attachment	xta:ContentType	
		GenericContainer	xta:GenericContent-Container	
		<i>Reporting</i>		
-		xta:RecordType		

XTA-WS-Differenzen von XTA 1.1.1 und XTA 2 Version 4

		+ Timestamp	xs:dateTime	
		+ Code	xta:Code.RecordType	
		+ Reason	xs:string	
MessageStatusType		xta:MessageStatusType		
+ Status	xs:integer	+ Status	xs:integer	
+ ErrorList	ErrorListType	+ ErrorList		
++ Error	RecordType	++ Error	xta:RecordType	
+ WarnList	WarningListType	+ WarnList		
++ Warning	RecordType	++ Warning	xta:RecordType	
+ InfoList	InfoListType	+ InfoList		
++ Info	RecordType	++ Info	xta:RecordType	
AdditionalReportList-Type		xta:AdditionalReportList-Type		
+ Report	ReportType	+ Report		
++ Typ	xs:string	++ Key	xta:Code.ReportType	siehe Anfang der Tabelle
++ Data	xs:base64Binary	++ Data	xs:base64binary	
TransportReport		xta:TransportReport		
-		+ ReportTime	xs:dateTime	
-		+ XTAServerIdentity	xs:token	
+ MessageStatus	MessageStatusType	+ MessageStatus	xta:MessageStatusType	
+ RelatesToMessage		+ oscimeta:Message-MetaData		
++ MessageId	wsa:AttributedURIType			siehe MsgIdentification in MessageMetaData
++ Action	wsa:AttributedURIType			wie o.g.
++ To	wsa:EndpointReferenceType			siehe Originators in MessageMetaData
++ From	wsa:EndpointReferenceType			siehe Destinations in MessageMetaData
++ ReceiveTimestamp	xs:dateTime			siehe DeliveryAttributes in MessageMetaData
++ DeliveryTimestamp	xs:dateTime			wie o.g.
+ AdditionalReports	AdditionalReportList-Type	+ AdditionalReports	xta:AdditionalReportList-Type	
-	-	+ ds:Signature		
		<i>Transportauftrag</i>		
		oscimeta:Delivery-AttributesType	DeliveryAttributesType	
		DeliveryAttributesType		

XTA-WS-Differenzen von XTA 1.1.1 und XTA 2 Version 4

		+ Origin	xs:dateTime	
		+ InitialSend	xs:dateTime	
		+ Delivery	xs:dateTime	
		+ InitialFetch	xs:dateTime	
		+ Reception	xs:dateTime	
		oscimeta:OriginatorsType	OriginatorsType	
		oscimeta:QualifierType		
		QualifierType		
DiensteBezeichnung	xs:anyURI	+ Service	xs:anyURI	
		+ BusinessScenario		
		++ Defined	oscimeta:KeyCodeType	
		++ Undefined	xs:normalizedString	
		+ MessageType	oscimeta:MessageType	
		oscimeta:MessageMetaData	MessageMetaData	
		MessageMetaData		
		+ TestMsg	xs:boolean	
		+ DeliveryAttributes (-)	oscimeta:DeliveryAttributesType	
„Absender“	wsa:EndpointReferenceType	+ Originators (-)	oscimeta:OriginatorsType	
„Empfänger“	wsa:EndpointReferenceType	+ Destinations (-)	oscimeta:DestinationsType	
		+ MsgIdentification (-)	oscimeta:MsgIdentificationType	
		+ Qualifier (-)	oscimeta:QualifierType	
		+ MsgSize	xs:positiveInteger	
		<i>Request / Response</i>		
IsServiceAvailable-SoapIn	xta:IsServiceAvailable	LookupServiceRequest	xta:LookupServiceRequest	
		xta:LookupServiceRequest		
		+ LookupServiceRequestList		
		++ LookupService	xta:LookupServiceType	
IsServiceAvailableSoap-Out	xta:IsServiceAvailable-Response	LookupServiceResponse	xta:LookupServiceResponse	
		xta:LookupServiceResponse		
		+ LookupServiceResultList		
		++ LookupServiceResult	xta:LookupServiceResultType	

		MsgBoxRequestType		
		+ osci:MsgSelector		
		MsgBoxResponseType		
		+MsgBoxRequestID	xs:anyURI	
		+NoMessageAvailable	xs:anyURI	
		+ItemsPending	xs:nonNegativeInteger	
GetMessageSoapOut	xta:GetMessage-Response	MsgBoxResponse	MsgBoxResponseType	
		osci:MsgBoxResponse	MsgBoxResponse	
		FetchResponseHeader	osci:MsgBoxResponse	
		MsgBoxFetchRequest-Type	erweitert MsgBoxRequestType	
		+ osci:MsgSelector	Aus MsgBoxRequestType	
		+ MsgPart	xs:NMTOKEN	
		MsgBoxFetchRequest	MsgBoxFetchRequest- Type	
		osci:MsgBoxFetch- Request	MsgBoxFetchRequest	
GetMessageSoapIn	xta:GetMessage	FetchRequest (in Methode getMessage)	osci:MsgBoxFetch- Request	
		xta:MessageResponse		
		+ xta:GenericContent- Container		
		MessageResponse- ContentContainer	xta:MessageResponse	
		MessageMetaData- Response	oscimeta:MessageMeta Data	
		MsgBoxStatusList- RequestType	erweitert MsgBoxRequestType	
		+ osci:MsgSelector	aus MsgBoxRequestType	
		+ ListForm	osci:ListFormEnum	
		+ maxListItems	xs:positiveInteger	
		MsgBoxStatusList- Request	MsgBoxStatusList- RequestType	
		osci:MsgBoxStatusList- Request	MsgBoxStatusList- Request	
GetMessageldList- SoapIn	xta:GetMessageldList	FetchRequest (in Methode getStatusList)	osci:MsgBoxStatusList- Request	
GetMessageldList- SoapOut	xta:GetMessageldList- Response	FetchResponse	osci:MsgStatusList	
		MsgBoxCloseRequest- Type		
		+ MsgBoxRequestID	xs:anyURI	

		+ LastMsgReceived	wsa:AttributedURIType	
		MsgBoxCloseRequest	MsgBoxCloseRequest- Type	
		osci:MsgBoxClose- Request	MsgBoxCloseRequest	
		FetchRequest (in Methode close)	osci:MsgBoxClose- Request	
GetTransportReport- SoapOut	xta:GetTransportReport Response	GetTransportReport- Response	xta:TransportReport	

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

3 Ausblick

Die hier beschriebenen Differenzen dienen als ein Hilfsmittel für einen Umstieg auf eine neuere XTA Version. Dadurch sollen die Aufwände für den Umstieg minimiert und die Interoperabilität zwischen verschiedenen XTA-Implementierungen gefördert werden. Im aktuellen Umfang wird ausschließlich ein Teilbereich die XTA-Webservices des Transportadapters adressiert. Weiterhin wird bei der Menge der Methoden und der Parameterinhalte auf die Konformitätsvorgaben als Grundlage gestützt. In beiden Punkten kann die vorliegende Differenzierung 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 Differenzdokumente erstellt werden. Diese Trennung berücksichtigt, dass die Service-Profile unabhängig von den Webservices verwendet werden können. Auch das Reporting kann auf anderen Bereich übertragen werden, um dieselben Metadaten und dieselbe Berichtsstruktur zu verwenden, jedoch mit selbstdefinierten Ereignissen. Spezifische Differenzdokumente würden die gezielte Pflege und Weiterentwicklung der Bereiche Webservices, Service-Profile und Reporting unterstützen.