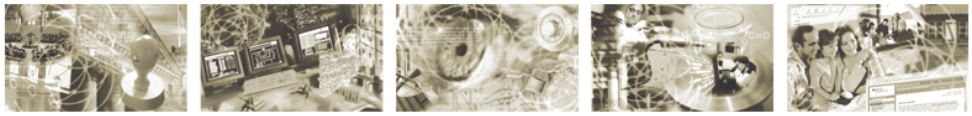
Bundesamt
für Sicherheit in der
Informationstechnik



BSI Technische Richtlinie 03125
Beweiswerterhaltung kryptographisch signierter Dokumente

Anlage TR-ESOR-E:
Konkretisierung der Schnittstellen auf Basis des eCard-API-Frameworks

Bezeichnung	Konkretisierung der Schnittstellen auf Basis des eCard-API-Frameworks
Kürzel	BSI TR-ESOR-E
Version	1.1
Datum	18.02.2011

Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63
53133 Bonn
Tel.: +49 228 99 9582-0
E-Mail: digsig@bsi.bund.de
Internet: https://www.bsi.bund.de

Inhaltsverzeichnis

1. Einführung

Ziel der Technischen Richtlinie „Beweiswerterhaltung kryptographisch signierter Dokumente“ ist die Spezifikation sicherheitstechnischer Anforderungen für den langfristigen Beweiswerterhalt von kryptographisch signierten elektronischen Dokumenten und Daten nebst zugehörigen elektronischen Verwaltungsdaten (Metadaten).

Eine für diese Zwecke definierte Middleware (TR-ESOR-Middleware) im Sinn dieser Richtlinie umfasst alle diejenigen Module (**M**) und Schnittstellen (**S)**, die zur Sicherung und zum Erhalt der Authentizität und zum Nachweis Integrität der aufbewahrten Dokumente und Daten eingesetzt werden.

Die im Hauptdokument dieser Technischen Richtlinie vorgestellte Referenzarchitektur besteht aus den nachfolgend beschriebenen funktionalen und logischen Einheiten:

- der Eingangs-Schnittstelle S.4 der TR-ESOR-Middleware, die dazu dienen, die TR-ESOR-Middleware in die bestehende IT- und Infrastrukturlandschaft einzubetten;
- dem zentralen Middlewaremodul M.1, welches den Informationsfluss in der Middleware regelt, die Sicherheitsanforderungen an die Schnittstellen zu den IT-Anwendungen umsetzt und für eine Entkopplung von Anwendungssystemen und ECM/Langzeitspeicher sorgt;
- dem „Krypto“-Modul M.2 nebst den zugehörigen Schnittstellen S.1 und S.3, das alle erforderlichen Funktionen zur Erstellung (optional) und Prüfung elektronischer Signaturen, zur Nachprüfung elektronischer Zertifikate und zum Einholen qualifizierter Zeitstempel für die Middleware zur Verfügung stellt. Darüber hinaus kann es Funktionen zur Ver- und Entschlüsselung von Daten und Dokumenten zur Verfügung stellen;
- dem „ArchiSig-Modul“ (TR-ESOR-M.3) mit der Schnittstelle S.6, dass die erforderlichen Funktionen für die Beweiswerterhaltung der digital signierten Unterlagen bereitstellt;
- einem ECM/Langzeitspeicher mit den Schnittstellen S.2 und S.5, der die physische Archivierung/Aufbewahrung und auch das Speichern der beweiswerterhaltenden Zusatzdaten übernimmt.
  Dieser ECM/Langzeitspeicher ist nicht mehr direkt Teil der Technischen Richtlinie, gleichwohl werden über die beiden Schnittstellen, die noch Teil der TR-ESOR-Middleware sind, Anforderungen daran gestellt.
  Ebenso wenig ist die Applikationsschicht, die auch einen XML-Adapter enthalten kann, direkter Teil der Technischen Richtlinie, auch wenn dieser XML-Adapter als Teil einer Middleware implementiert werden kann.

Die in Abbildung 1 dargestellte IT-Referenzarchitektur orientiert sich an der ArchiSafe[1] Referenz-architektur und soll die logische (funktionale) Interoperabilität künftiger Produkte mit den Zielen und Anforderungen der Technischen Richtlinie ermöglichen und unterstützen.

[1] Siehe dazu http://www.archisafe.de

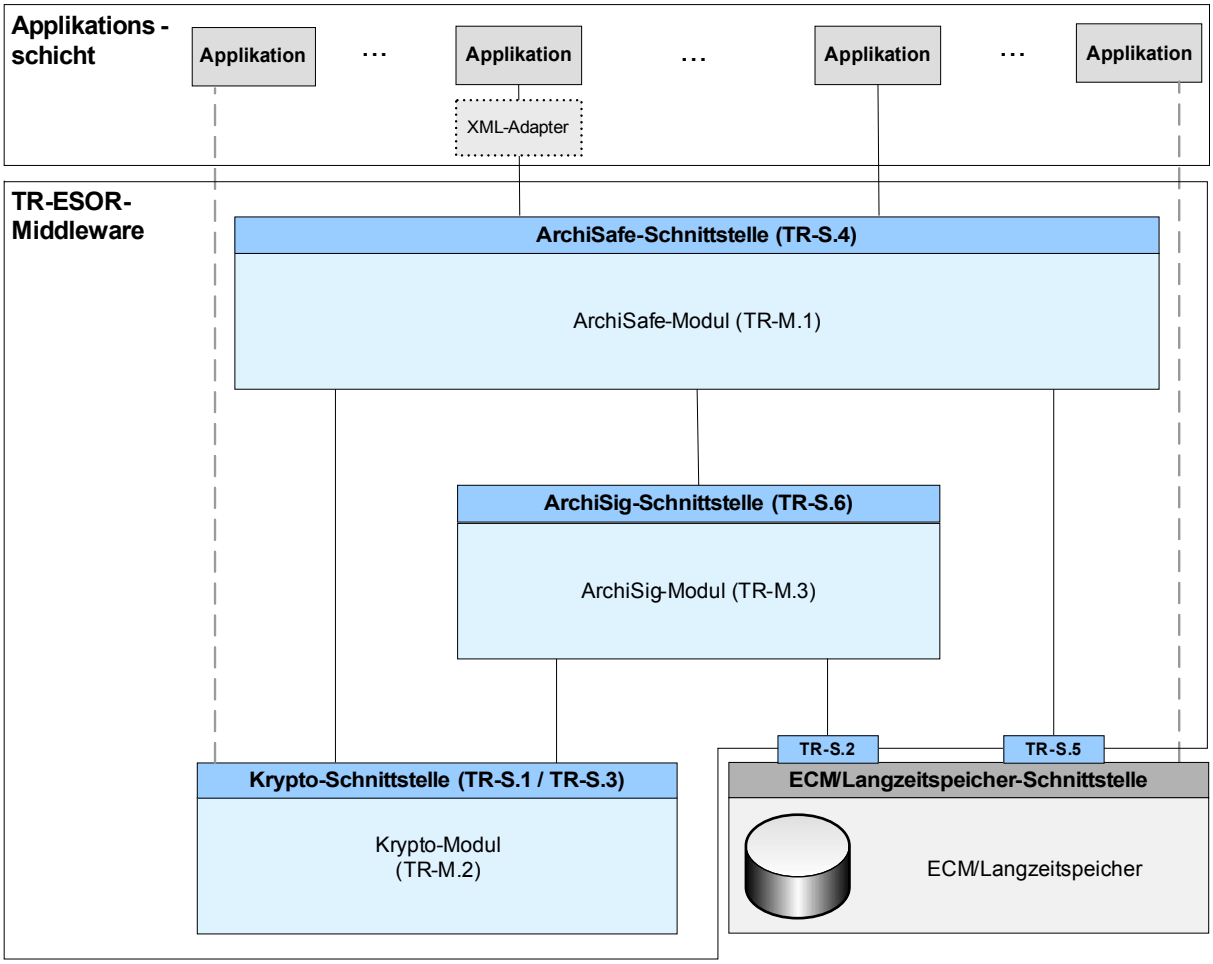


Abbildung 1: Schematische Darstellung der IT-Referenzarchitektur

Diese Technische Richtlinie ist modular aufgebaut und spezifiziert in einzelnen Anlagen zum Hauptdokument die funktionalen und sicherheitstechnischen Anforderungen an die erforderlichen IT-Komponenten und Schnittstellen der TR-ESOR-Middleware. Die Spezifikationen sind strikt plattform-, produkt-, und herstellerunabhängig.

Das vorliegende Dokument trägt die Bezeichnung „Anlage TR-ESOR-E“ und konkretisiert die in [TR-ESOR-S] eingeführten Schnittstellen auf Basis des in der BSI TR 03112 spezifizierten eCard-API-Frameworks.

2. Überblick

Wie in Abschnitt 9 des Hauptdokumentes näher erläutert, ist für den Nachweis der Konformität zur vorliegenden technischen Richtlinie ein zweistufiges Verfahren vorgesehen.

(A2.0-1) Demnach muss in *Konformitätsstufe 1* lediglich die funktionale und logische Konformität eines aus mindestens einem in dieser Richtlinie spezifizierten Modul bestehenden Produktes oder Systems mit den Anforderungen der Richtlinie nachgewiesen werden. Die Unterstützung der einzelnen in [TR-ESOR-S] und hier beschriebenen Schnittstellen ist somit optional.

(A2.0-2) Sofern bei der Realisierung eines aus mindestens einem in dieser Richtlinie spezifizierten Modul bestehenden Produktes oder Systems die technische Konformität und Interoperabilität der *Konformitätsstufe 2* nachgewiesen werden soll, muss diese auf Basis der in diesem Dokument beschriebenen Profilierung des eCard-API-Frameworks umgesetzt werden. Hierbei müssen zumindest die im vorliegenden Dokument näher aufgeführten Funktionen und Parameterkonstellationen unterstützt werden.

(A2.0-3) Sofern bei der Realisierung eines aus mindestens einem in dieser Richtlinie spezifizierten Modul bestehenden Produktes oder Systems die technische Konformität und Interoperabilität der *Konformitätsstufe 2* nachgewiesen werden soll, soll XAIP aus Anhang F als XML-Datenformat verwendet werden. Abweichungen im verwendeten XML-Datenformat sind zulässig, allerdings muss dann erläutert werden, dass gleichwertige Funktionalität unterstützt wird. Insbesondere ist zu erläutern, wie eine Transformation in das XAIP Format aus Anhang F erfolgen kann.

(A2.0-4) Für ein Krypto-Modul wird die vollständige Unterstützung des eCard-API-Frameworks gemäß BSI TR 03112 empfohlen.

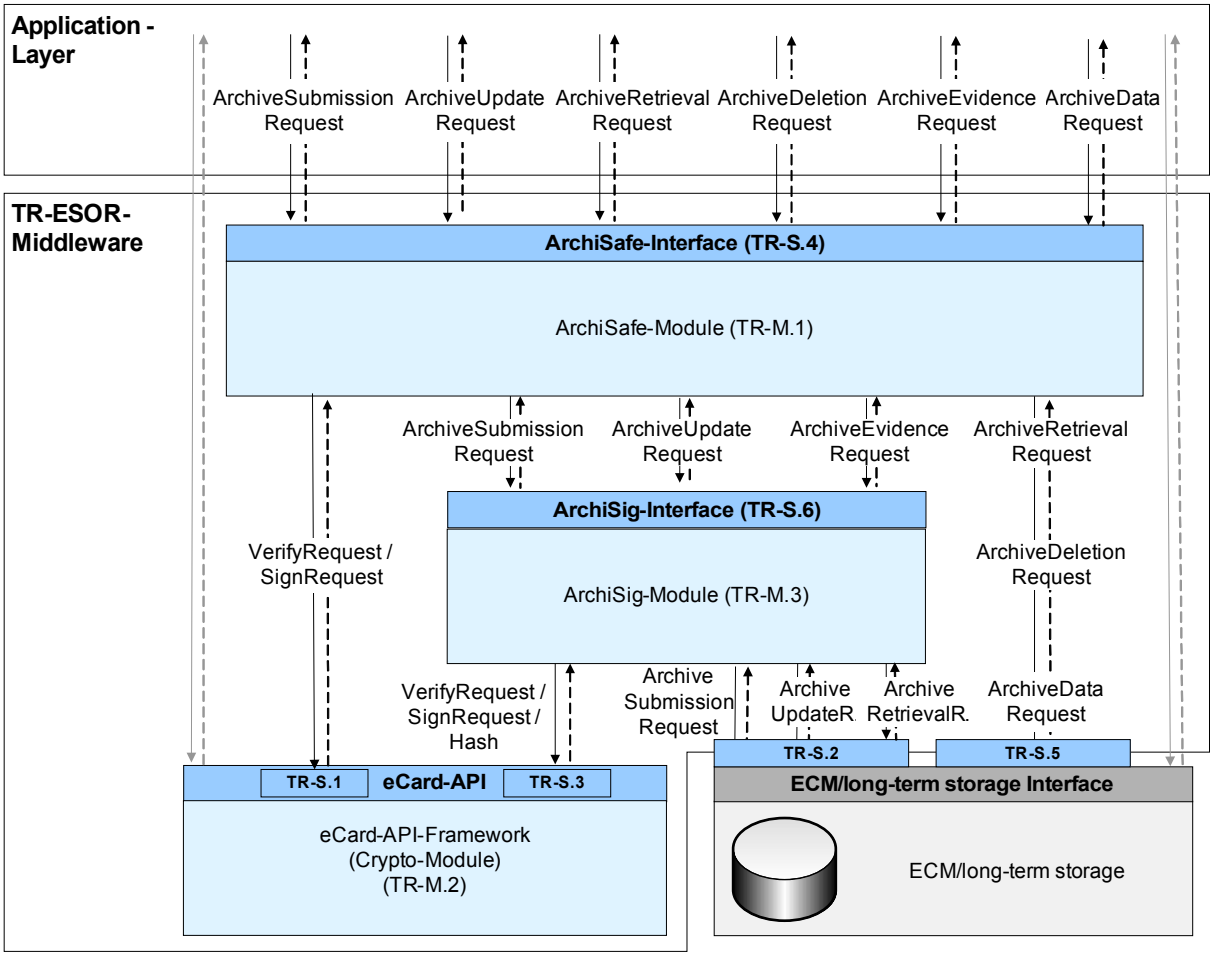


Abbildung 2: Umsetzung der IT-Referenzarchitektur auf Basis des eCard-API-Frameworks

Wie in Abbildung 2 angedeutet, werden bei der vollständigen Umsetzung der IT-Referenzarchitektur auf Basis des eCard-API-Frameworks

1. die Schnittstellen des Krypto-Moduls gemäß des eCard-API-Frameworks (Technische Richtlinie des BSI TR 03112) realisiert und
2. auch die Schnittstellen des ArchiSafe-, ArchiSig- und ECM/Langzeitspeichers nutzen die gleichen grundlegenden Schnittstellentypen (`dss:RequestBaseType` und `dss:ResponseBaseType`) aus [OASIS-DSS], die auch bei den Signatur- und Verschlüsselungsfunktionen aus [eCard-2]genutzt werden.

Die URI-Fehlercodes in den Rückgaben der nicht bereits in der Technischen Richtlinie des BSI TR 03112 definierten Funktionen haben das Präfix http://www.bsi.bund.de/tr-esor/api/1.1, welches um entsprechende Bezeichner ergänzt wird. Dieser Namensraum ist in den visualisierten XML-Strukturen am Kürzel „`tr`“ erkennbar.

In den folgenden Abschnitten findet sich zunächst eine XML-basierte Spezifikation der verschiedenen in [TR-ESOR-S] eingeführten Funktionen zur Beweiswerterhaltung kryptographisch signierter Dokumente (vgl. Abschnitt 3), bevor auf die Nutzung dieser Funktionen in den verschiedenen Schnittstellen eingegangen wird (vgl. Abschnitt 4). In Abschnitt 5 finden sich schließlich die normativen XML-Schema- und WSDL-Spezifikationen.

3. Funktionen der TR-ESOR-Middleware

In diesem Abschnitt findet sich eine XML-basierte Spezifikation der in [TR-ESOR-S] eingeführten Funktionen der TR-ESOR-Middleware:

- ArchiveSubmissionRequest und ArchiveSubmissionResponse (siehe Abschnitt 3.1)
- ArchiveUpdateRequest und ArchiveUpdateResponse (siehe Abschnitt 3.2)
- ArchiveRetrievalRequest und ArchiveRetrievalResponse (siehe Abschnitt 3.3)
- ArchiveEvidenceRequest und ArchiveEvidenceResponse (siehe Abschnitt 3.4)
- ArchiveDeletionRequest und ArchiveDeletionResponse (siehe Abschnitt 3.5)
- ArchiveDataRequest und ArchiveDataResponse (siehe Abschnitt 3.6)

Darüber hinaus finden sich in Abschnitt 3.7 grundlegende Informationen über die hier genutzten Funktionen des eCard-API-Frameworks.

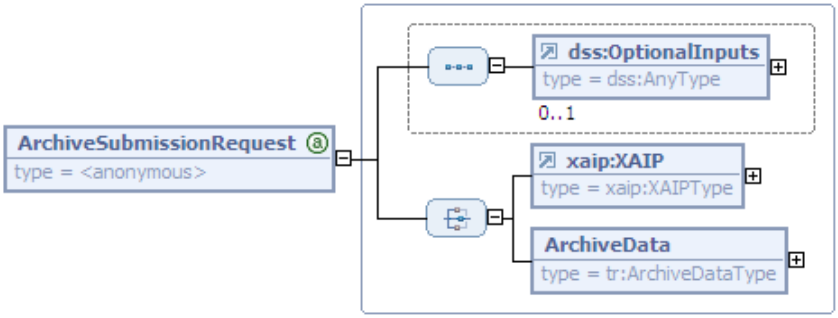
Die graphische Darstellung der Schnittstellen in diesem Kapitel wurde - analog zur Spezifikation des eCard-API-Frameworks (siehe z.B. [eCard-2]) - mit einem XML-Viewer erstellt und dient lediglich der Veranschaulichung der XML-Strukturen. Die normative Spezifikation der Schnittstellen ist durch das XML-Schema bzw. die darauf aufbauende WSDL-Spezifikation (siehe Abschnitt 5) gegeben.

3.1 ArchiveSubmissionRequest und ArchiveSubmissionResponse

Mit der Funktion `ArchiveSubmissionRequest` kann dem aufgerufenen Modul ein Archivdatenobjekt (`xaip:XAIP`) zur Ablage übergeben werden und das aufrufende Modul erhält im Erfolgsfall in der `ArchiveSubmissionResponse` eine `AOID` zurück, mit der später wieder auf das archivierte Objekt oder die zugehörigen technischen Beweisdaten zugegriffen werden kann.

Wie in Abbildung 2 ersichtlich, wird diese Funktion in den Schnittstellen S.2 (vgl. Abschnitt 4.3), S.4 (vgl. Abschnitt 4.5) und S.6 (vgl. Abschnitt 4.7) genutzt.

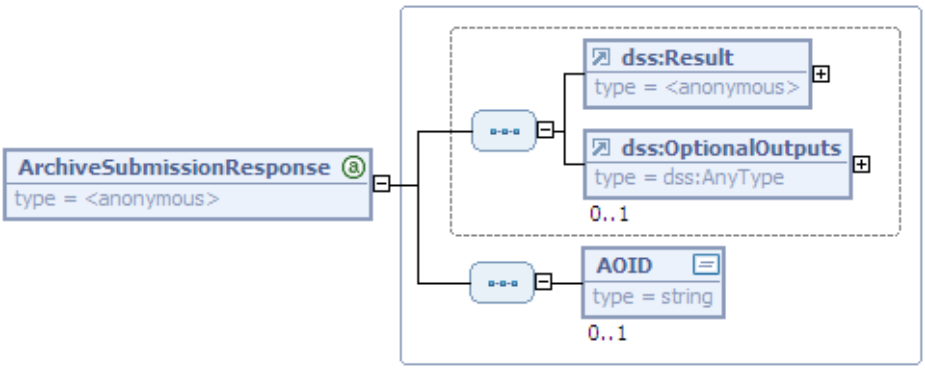
3.1.1 ArchiveSubmissionRequest

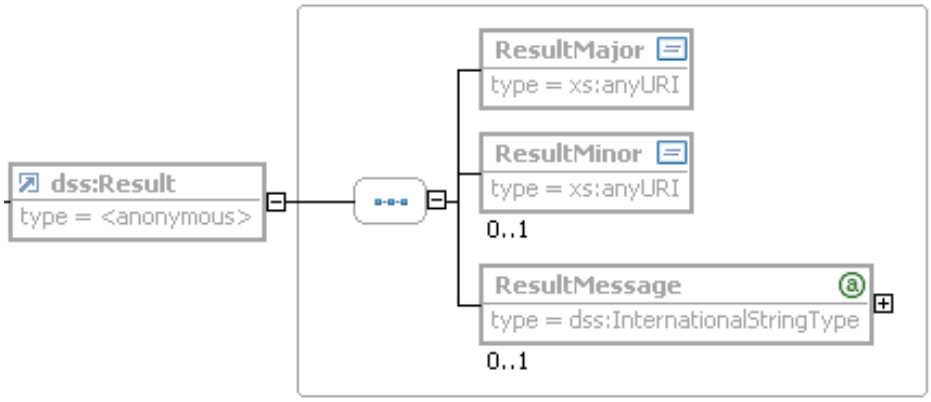
<table>
<tr><td>Name</td><td colspan="2"><code>ArchiveSubmissionRequest</code></td></tr>
<tr><td>Beschreibung</td><td colspan="2">Mit der Funktion <code>ArchiveSubmissionRequest</code> wird dem aufgerufenen Modul ein Archivdatenobjekt übergeben.</td></tr>
<tr><td rowspan="3">Aufruf</td><td colspan="2"> 

Aufruf der <code>ArchiveSubmissionRequest</code>-Funktion</td></tr>
<tr><td>Name</td><td>Beschreibung</td></tr>
</table>

	`dss:OptionalInputs`	Ist für optionale Eingabeelemente vorgesehen und kann beispielsweise Steuerelemente (`requestControls`) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen. Außerdem kann beispielsweise auch eine `AOID` übergeben werden, sofern diese vom aufrufenden Modul eindeutig erzeugt und verwaltet werden kann. Sofern hier keine AOID übergeben wird, muss diese vom aufgerufenen Modul bereitgestellt werden.
	`xaip:XAIP`	Enthält ein XML-basiertes Archivdatenobjekt gemäß [TR-ESOR-F], das durch den Aufruf der beweiswerterhaltenden Archivierung zugeführt werden soll.
	`ArchiveData`	Enthält ein in einem beliebigen anderen Format vorliegendes Archivdatenobjekt. Der hierfür genutzte `ArchiveDataType` ist als `anyType` mit einem optionalen `Type`-Attribut definiert. Details eines solchen Archivdatenobjektes können im Rahmen einer Profilierung der vorliegenden Spezifikation spezifiziert werden.

3.1.2 ArchiveSubmissionResponse

Name	**`ArchiveSubmissionResponse`**	
Beschreibung	Als Antwort auf einen `ArchiveSubmissionRequest` wird ein entsprechendes `ArchiveSubmissionResponse`-Element zurückgeliefert, das im Erfolgsfall einen eindeutigen Identifikator des Archivdatenobjektes, die `AOID`, enthält.	
Rückgabe	 `ArchiveSubmissionResponse` ist die Antwort zum `ArchiveSubmissionRequest`-Aufruf	
	Name	**Beschreibung**
	`dss:Result`	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in [eCard-1] und unten näher beschrieben.

<table>
<tr><td>Name</td><td colspan="2"><code>ArchiveSubmissionResponse</code></td></tr>
<tr><td rowspan="4"></td><td><code>dss:OptionalOutputs</code></td><td>Ist für optionale Ausgabeelemente vorgesehen und kann beispielsweise entsprechende Steuerelemente (<code>responseControls</code>) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen.</td></tr>
<tr><td><code>AOID</code></td><td>Muss, sofern die <code>AOID</code> vom aufgerufenen Modul erzeugt oder ergänzt wurde, vorhanden sein und für zukünftige Zugriffe auf das Archivdatenobjekt genutzt werden.</td></tr>
<tr><td colspan="2"> 

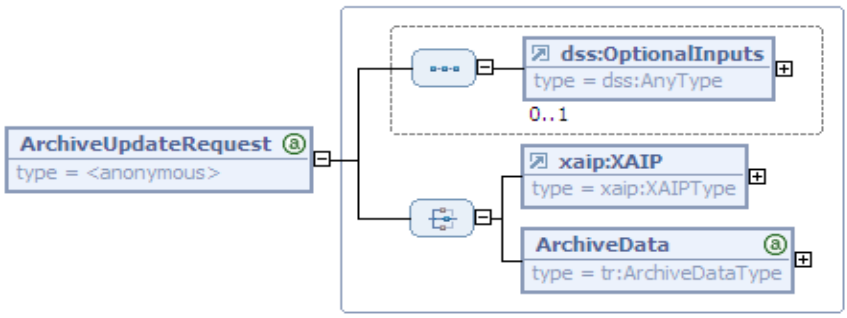
Statusinformationen und Fehler bei <code>ArchiveSubmissionResponse</code> (vgl. [eCard-1] Abschnitt 4.1 und 4.2).</td></tr>
<tr><td colspan="2"><table>
<tr><th>Name</th><th>Fehlercode</th></tr>
<tr><td><code>ResultMajor</code></td><td>• /resultmajor#ok
• /resultmajor#error
• /resultmajor#warning</td></tr>
<tr><td><code>ResultMinor</code></td><td>• /resultminor/al/common#noPermission
• /resultminor/al/common#internalError
• /resultminor/al/common#parameterError
• /resultminor/arl/lowSpaceWarning
• /resultminor/arl/noSpaceError
• /resultminor/arl/existingAOID
• /resultminor/arl/unknownOptionalInfo</td></tr>
</table></td></tr>
</table>

3.2 ArchiveUpdateRequest und ArchiveUpdateResponse

Mit der Funktion `ArchiveUpdateRequest` kann eine neue Version für ein bereits abgelegtes Archivdatenobjekt erzeugt werden.

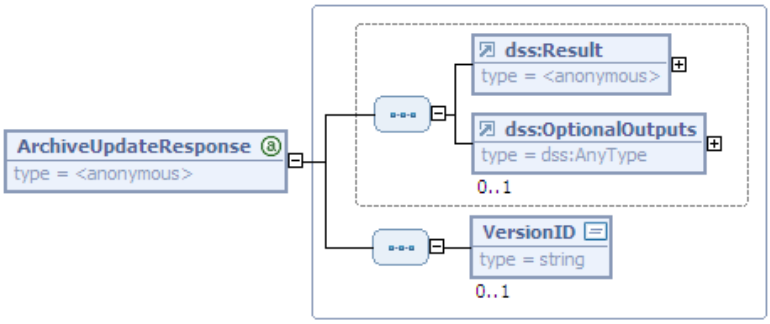
Wie in Abbildung 2 ersichtlich, wird diese Funktion in den Schnittstellen S.2 (vgl. Abschnitt 4.3), S.4 (vgl. Abschnitt 4.5) und S.6 (vgl. Abschnitt 4.7) genutzt.

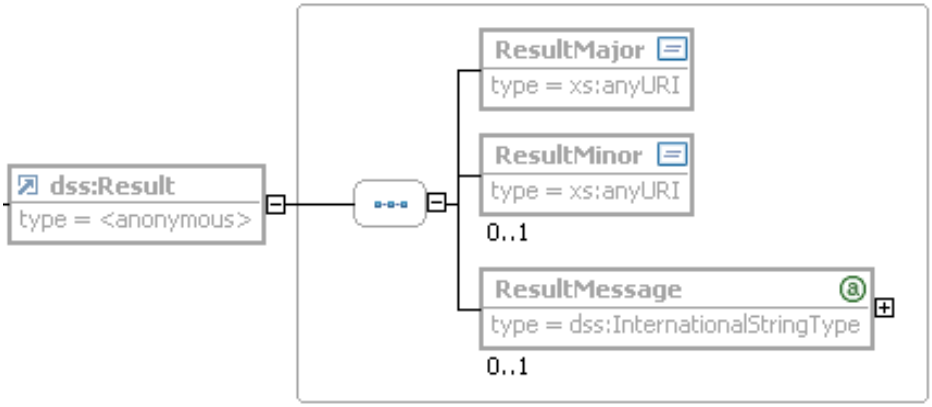
3.2.1 ArchiveUpdateRequest

<table>
<tr><td>Name</td><td colspan="2"><code>ArchiveUpdateRequest</code></td></tr>
<tr><td>Beschreibung</td><td colspan="2">Mit der Funktion <code>ArchiveUpdateRequest</code> wird eine neue Version für ein bereits abgelegtes Archivdatenobjekt erzeugt (vgl. [TR-ESOR-M.1]).</td></tr>
<tr><td>Aufruf</td><td colspan="2"> 

Aufruf der <code>ArchiveUpdateRequest</code>-Funktion</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
<tr><td></td><td><code>dss:OptionalInputs</code></td><td>Ist für optionale Eingabeelemente vorgesehen und kann beispielsweise Steuerelemente (<code>requestControls</code>) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen.
Sofern die <code>AOID</code> des zu ergänzenden Archivdatenobjektes nicht bereits im <code>xaip:XAIP</code>-Element enthalten ist, muss diese hier übergeben werden.</td></tr>
<tr><td></td><td><code>xaip:XAIP</code></td><td>Enthält ein XML-basiertes Archivdatenobjekt gemäß [TR-ESOR-F], das die zu ergänzenden Elemente enthält, die in einer neuen Version eines bereits abgelegten Archivdatenobjektes ergänzt werden sollen.</td></tr>
<tr><td></td><td><code>ArchiveData</code></td><td>Enthält ein in einem beliebigen anderen Format vorliegendes Archivdatenobjekt. Der hierfür genutzte <code>ArchiveDataType</code> ist als <code>anyType</code> mit einem optionalen <code>Type</code>-Attribut definiert. Details eines solchen Archivdatenobjektes können im Rahmen einer Profilierung der vorliegenden Spezifikation spezifiziert werden.</td></tr>
</table>

3.2.2 ArchiveUpdateResponse

<table>
<tr><td>Name</td><td><code>ArchiveUpdateResponse</code></td></tr>
<tr><td>Beschreibung</td><td>Als Antwort auf einen <code>ArchiveUpdateRequest</code> wird ein entsprechendes <code>ArchiveUpdateResponse</code>-Element zurückgeliefert, das im Erfolgsfall einen im Kontext einer <code>AOID</code> eindeutigen Identifikator der neuen Version des Archivdatenobjektes, die <code>VersionID</code>, enthält.</td></tr>
<tr><td>Rückgabe</td><td>


<code>ArchiveUpdateResponse</code> ist die Antwort zum <code>ArchiveUpdateRequest</code>-Aufruf
<table>
<tr><th>Name</th><th>Beschreibung</th></tr>
<tr><td><code>dss:Result</code></td><td>Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in [eCard-1] und unten näher beschrieben.</td></tr>
<tr><td><code>dss:OptionalOutputs</code></td><td>Ist für optionale Ausgabeelemente vorgesehen und kann beispielsweise entsprechende Steuerelemente (<code>responseControls</code>) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen.</td></tr>
<tr><td><code>VersionID</code></td><td>Ist im Erfolgsfall vorhanden und enthält den bezüglich des über die <code>AOID</code> identifizierten Archivdatenobjektes eindeutigen Versions-Identifikator.</td></tr>
</table>


Statusinformationen und Fehler bei <code>ArchiveUpdateResponse</code> (vgl. [eCard-1] Abschnitt 4.1 und 4.2).
<table>
<tr><th>Name</th><th>Fehlercode</th></tr>
</table>
</td></tr>
</table>

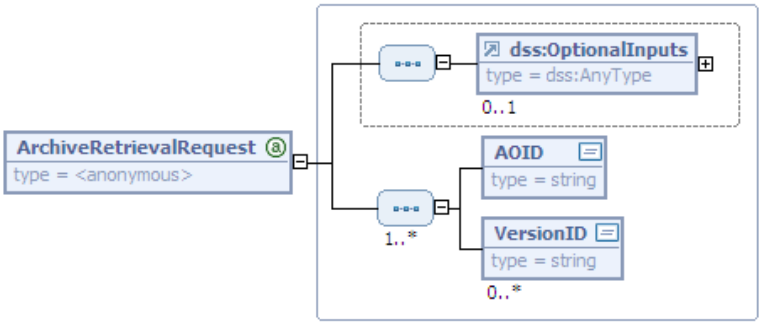
Name	**ArchiveUpdateResponse**	
	ResultMajor	• /resultmajor#ok • /resultmajor#error • /resultmajor#warning
	ResultMinor	• /resultminor/al/common#noPermission • /resultminor/al/common#internalError • /resultminor/al/common#parameterError • /resultminor/arl/lowSpaceWarning • /resultminor/arl/noSpaceError • /resultminor/arl/existingAOID • /resultminor/arl/unknownOptionalInfo

3.3 ArchiveRetrievalRequest und ArchiveRetrievalResponse

Mit der Funktion `ArchiveRetrievalRequest` kann das zu einer übergebenen `AOID` gehörende Archivdatenobjekt im `XAIP`-Format gemäß [TR-ESOR-F] über die TR-ESOR-Middleware aus dem ECM-/Langzeitspeichersystem ausgelesen werden.

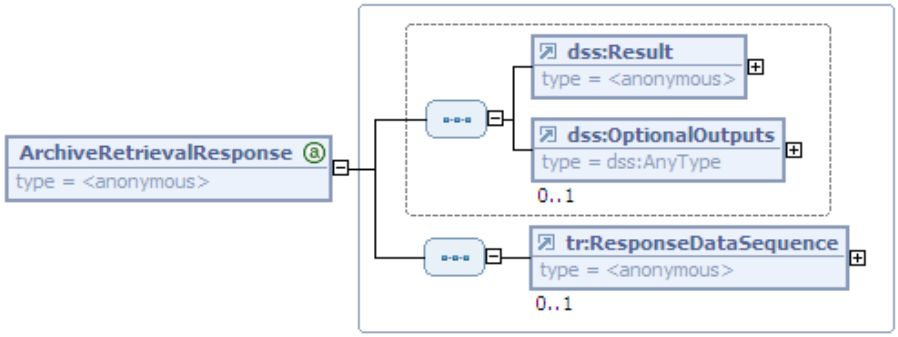
Wie in Abbildung 2 ersichtlich, wird diese Funktion in den Schnittstellen S.2 (vgl. Abschnitt 4.3), S.4 (vgl. Abschnitt 4.5) und S.5 (vgl. Abschnitt 4.6) genutzt.

3.3.1 ArchiveRetrievalRequest

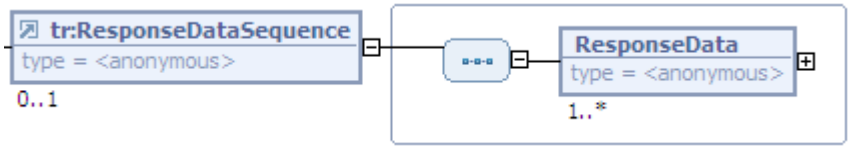
Name	**ArchiveRetrievalRequest**	
Beschreibung	Mit der Funktion `ArchiveRetrievalRequest` kann eine Folge von im Langzeitspeicher abgelegten Archivdatenobjekten (`xaip:XAIP`) ausgelesen und zurückgeliefert werden.	
Beschreibung	 Aufruf der `ArchiveRetrievalRequest`-Funktion	
	Name	**Beschreibung**
	`dss:OptionalInputs`	Ist für optionale Eingabeelemente vorgesehen und kann beispielsweise Steuerelemente (`requestControls`) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen.
	`AOID`	Enthält den eindeutigen Identifikator des angeforderten Archivdatenobjektes.

Name	**ArchiveRetrievalRequest**	
	VersionID	Kann eine Folge von Versions-Identifikatoren enthalten, durch die angegeben wird welche Versionen des Archivdatenobjektes genau zurückgeliefert werden sollen. Sofern das VersionID-Element nicht angegeben ist, werden die zur letzten Version gehörigen Datenobjekte und Verwaltungsinformationen zurückgeliefert.

3.3.2 ArchiveRetrievalResponse

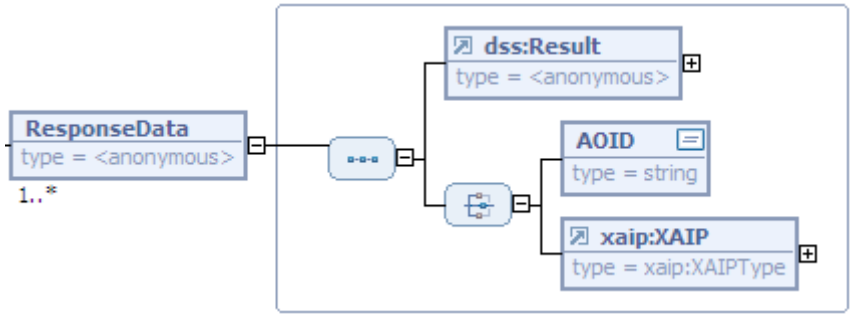
Name	**ArchiveRetrievalResponse**	
Beschreibung	Als Antwort auf einen ArchiveRetrievalRequest wird ein entsprechendes ArchiveRetrievalResponse-Element zurückgeliefert, das die angeforderte Folge von Archivdatenobjekten im xaip:XAIP-Format gemäß [TR-ESOR-F] enthält.	
Rückgabe	 ArchiveRetrievalResponse ist die Antwort zum ArchiveRetrievalRequest-Aufruf	
	Name	**Beschreibung**
	dss:Result	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in [eCard-1] und weiter unten näher beschrieben. Sofern nur ein Teil der angeforderten Archivdatenobjekte zurückgeliefert werden konnte, wird dies durch den Fehlercode /resultminor/arl/requestOnlyPartlySuccessfulWarning angezeigt.
	dss:OptionalOutputs	Ist für optionale Ausgabeelemente vorgesehen und kann beispielsweise entsprechende Steuerelemente (responseControls) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen.

Name	`ArchiveRetrievalResponse`	
	`tr:ResponseData Sequence`	Sofern kein Fehler aufgetreten ist, wird das `ResponseDataSequence`-Element zurückgeliefert, das die angeforderten Archivdatenobjekte oder entsprechende `ArchiveToken`-spezifische Fehlermeldungen enthält. Die detaillierte Struktur dieses Elementes ist nachfolgend visualisiert.



Das `ResponseDataSequence`-Element kann im `ArchiveRetrievalResponse` zurückgeliefert werden.

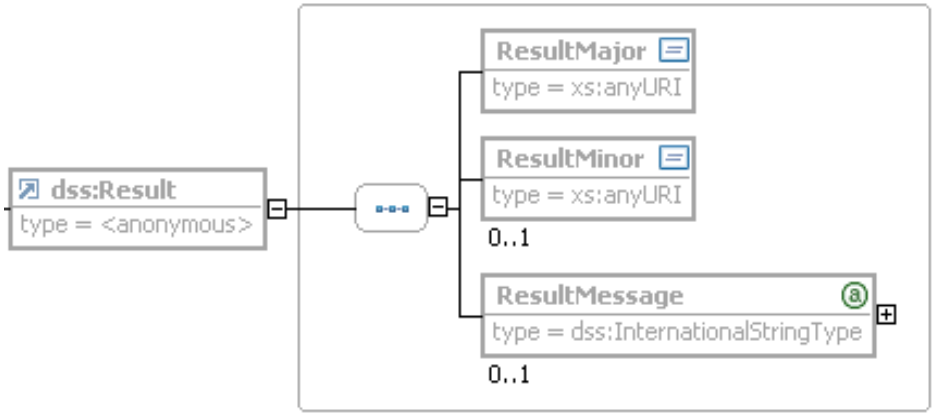
Name	Beschreibung
`ResponseData`	Dieses Element ist für jedes angeforderte Archivdatenobjekt jeweils einmal vorhanden und enthält neben einem Statuscode in Form eines `dss:Result`-Elementes entweder im Erfolgsfall das gewünschte Archivdatenobjekt im `xaip:XAIP`- Format gemäß [TR-ESOR-F]oder bei einem Fehler die AOID des problembehafteten Archivdatenobjektes.



Das `ResponseData`-Element ist für jedes angeforderte Archivdatenobjekt einmal vorhanden.

Name	Beschreibung

<table>
<tr><td>Name</td><td colspan="2"><code>ArchiveRetrievalResponse</code></td></tr>
<tr><td rowspan="4"></td><td><code>dss:Result</code></td><td>Gibt an, ob das angeforderte Archivdatenobjekt zurückgeliefert werden konnte oder nicht.
Als <code>ResultMajor</code> sind die beiden folgenden Werte möglich:
• /resultmajor#ok
• /resultmajor#error

Als <code>ResultMinor</code> sind die beiden folgenden Werte möglich:
• /resultminor/al/common#noPermission
• /resultminor/arl/unknownAOID</td></tr>
<tr><td><code>AOID</code></td><td>Dieses Element ist nur im Fehlerfall vorhanden und enthält den eindeutigen Identifikator des Archivdatenobjekts.</td></tr>
<tr><td><code>xaip:XAIP</code></td><td>Ist im Erfolgsfall vorhanden und enthält das XML-basierte Archivdatenobjekt (<code>xaip:XAIP</code>). Die detaillierte Struktur dieses Elements ist in [TR-ESOR-F] beschrieben.</td></tr>
<tr><td colspan="2"> 

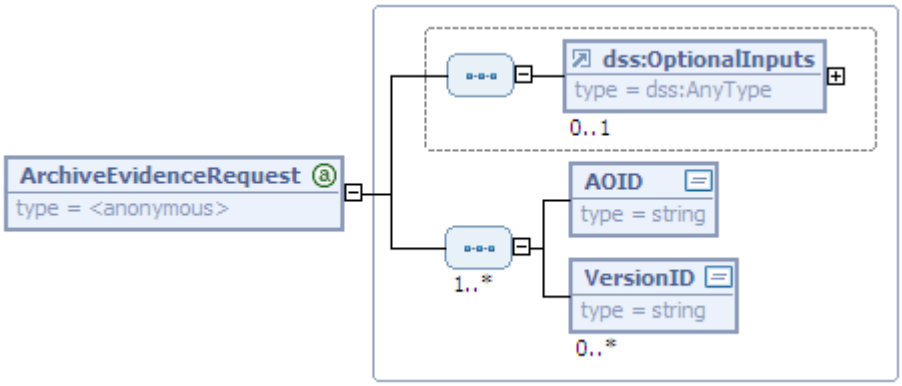
Statusinformationen und Fehler bei <code>ArchiveRetrievalResponse</code> (vgl. [eCard-1]).
<table>
<tr><td>Name</td><td>Fehlercode</td></tr>
<tr><td><code>ResultMajor</code></td><td>• /resultmajor#ok
• /resultmajor#error
• /resultmajor#warning</td></tr>
<tr><td><code>ResultMinor</code></td><td>• /resultminor/al/common#noPermission
• /resultminor/al/common#internalError
• /resultminor/al/common#parameterError
• /resultminor/arl/unknownAOID
• /resultminor/arl/unknownOptionalInfo
• /resultminor/arl/requestOnlyPartlySuccessfulWarning</td></tr>
</table></td></tr>
</table>

3.4 ArchiveEvidenceRequest und ArchiveEvidenceResponse

Mit der Funktion `ArchiveEvidenceRequest` können die zugehörigen technischen Beweisdaten (Evidence Records gemäß [RFC 4998] oder [ERSXML]) für beweiswerterhaltend aufbewahrte und über `AOID`-Elemente adressierte Archivdatenobjekte (`xaip:XAIP`) zurückgeliefert werden.

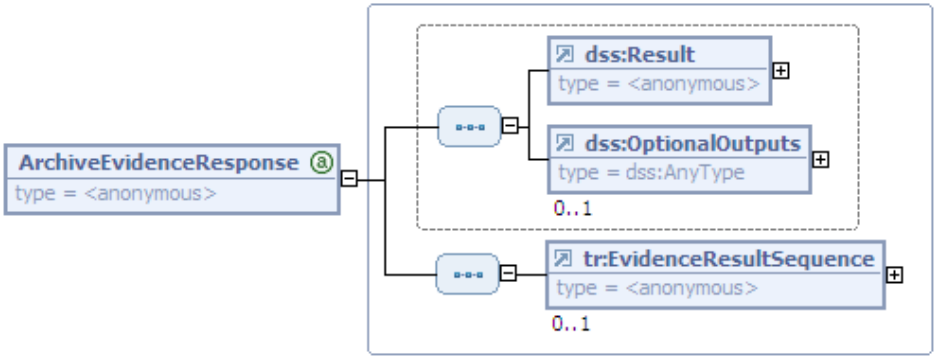
Wie in Abbildung 2 ersichtlich, wird diese Funktion in den Schnittstellen S.4 (vgl. Abschnitt 4.5) und S.6 (vgl. Abschnitt 4.7) genutzt.

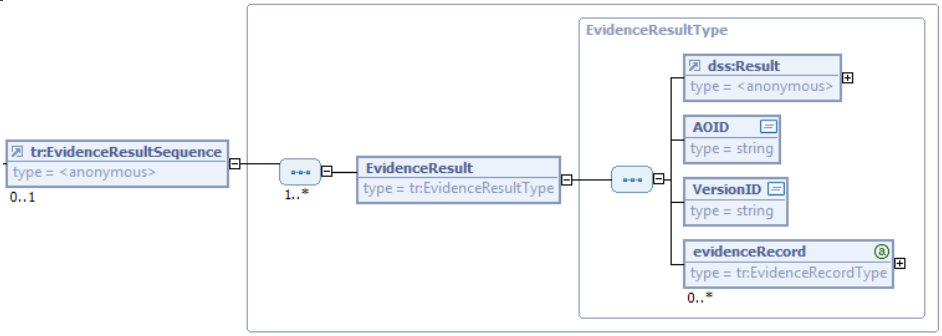
3.4.1 ArchiveEvidenceRequest

<table>
<tr><td>Name</td><td colspan="2"><code>ArchiveEvidenceRequest</code></td></tr>
<tr><td>Beschreibung</td><td colspan="2">Mit der Funktion <code>ArchiveEvidenceRequest</code> können für beweiswerterhaltend abgelegte Archivdatenobjekte technische Beweisdaten in Form von Evidence Records gemäß [RFC4998] oder [ERSXML] angefordert werden.</td></tr>
<tr><td>Beschreibung</td><td colspan="2">Aufruf der <code>ArchiveEvidenceRequest</code>-Funktion
 
</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
<tr><td></td><td><code>dss:OptionalInputs</code></td><td>Ist für optionale Ausgabeelemente vorgesehen und kann beispielsweise entsprechende Steuerelemente (<code>responseControls</code>) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen.
Beispielsweise ist hier das folgende Element vorgesehen:</td></tr>
</table>

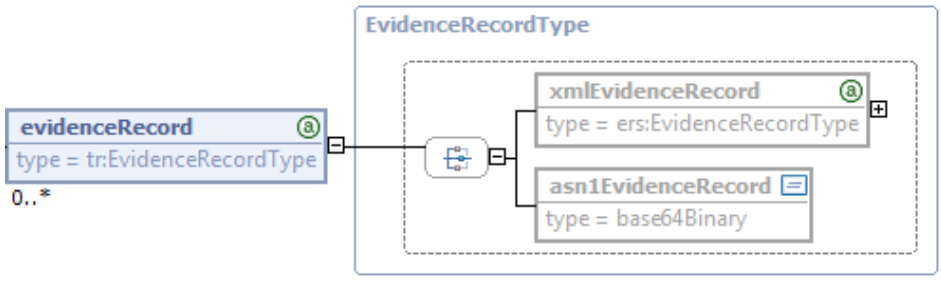
Name	**ArchiveEvidenceRequest**	
		 Mit dem Element `tr:ERSFormat` vom Typ `anyURI` kann das gewünschte Format der zurückgelieferten Evidence Records angegeben werden, wobei folgende URIs vorgesehen sind: • urn:ietf:rfc:4998 für ASN.1-basierte Evidence Records gemäß [RFC4998] • http://tools.ietf.org/html/draft-ietf-ltans-xmlers-10 für XML-basierte Evidence Records gemäß [ERSXML]
	`AOID`	Ist der eindeutige Identifikator des angeforderten Archivdatenobjektes.
	`VersionID`	Kann mehrfach auftreten und angeben für welche Versionen eines über die AOID identifizierten Archivdatenobjektes eindeutigen Versions-Identifikator zurückgeliefert werden sollen. Sofern das `VersionID`-Element nicht angegeben ist, wird der Beweisdatensatz für die aktuelle Version des XAIP zurückgeliefert.

3.4.2 ArchiveEvidenceResponse

Name	**ArchiveEvidenceResponse**	
Beschreibung	Als Antwort auf einen `ArchiveEvidenceRequest` wird ein entsprechendes `ArchiveEvidenceResponse`-Element zurückgeliefert, das die angeforderten Beweisdaten enthält.	
Rückgabe	 `ArchiveEvidenceResponse` ist die Antwort zum `ArchiveEvidenceRequest`-Aufruf	
	Name	**Beschreibung**

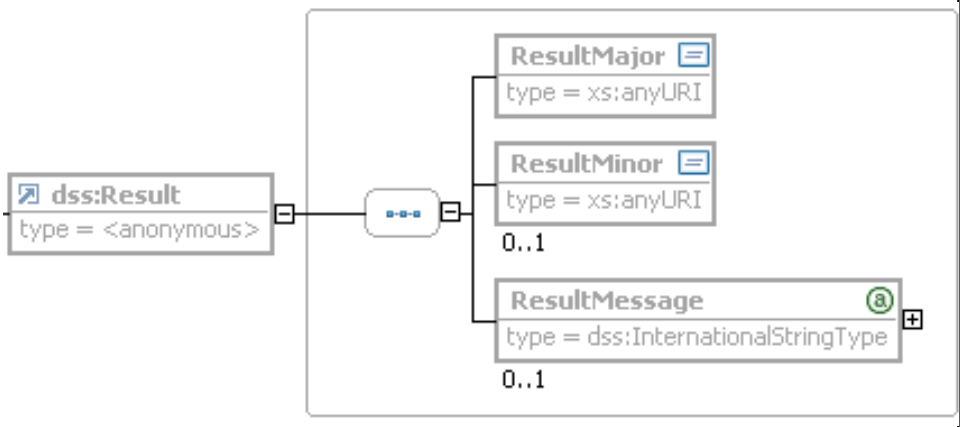
<table>
<tr><td>Name</td><td colspan="2"><code>ArchiveEvidenceResponse</code></td></tr>
<tr><td rowspan="4"></td><td><code>dss:Result</code></td><td>Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in Abschnitt 4.1.2 von [eCard-1] und unten näher beschrieben.
Sofern nicht für alle mittels der AOID im <code>ArchiveToken</code> adressierten Archivdatenobjekte entsprechende Beweisdaten (Evidence Records) zurückgeliefert werden konnten, wird dies durch die /resultminor/arl/requestOnlyPartlySuccessfulWarning angezeigt.</td></tr>
<tr><td><code>dss:OptionalOutputs</code></td><td>Ist für optionale Ausgabeelemente vorgesehen und kann beispielsweise entsprechende Steuerelemente (<code>responseControls</code>) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen.</td></tr>
<tr><td><code>tr:EvidenceResultSequence</code></td><td>Sofern kein Fehler aufgetreten ist, wird das <code>EvidenceResultSequence</code>-Element zurückgeliefert, das die angeforderten Evidence Records gemäß [RFC4998] oder [ERSXML] enthält. Die detaillierte Struktur dieses Elementes ist nachfolgend dargestellt.</td></tr>
<tr><td colspan="2"> 

Das <code>EvidenceResultSequence</code>-Element kann im <code>ArchiveEvidenceResponse</code> zurückgeliefert werden. Es besteht aus einer Folge von <code>EvidenceResult</code>-Elementen, die wiederum folgende Kindelemente besitzen:
<table>
<tr><td>Name</td><td>Beschreibung</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Name</th><th colspan="2">ArchiveEvidenceResponse</th></tr>
<tr><td rowspan="5"></td><td>dss:Result</td><td>Anders als beim oben erläuterten dss:Result-Element, das als Kind-Element von ArchiveEvidenceResponse Auskunft über das Gesamtergebnis des Aufrufs gibt, enthält das dss:Result-Element hier die Information, ob für das angeforderte Archivdatenobjekt entsprechende Evidence Records konstruiert werden konnten oder nicht.
Als ResultMajor sind die beiden folgenden Werte möglich:
• .../resultmajor#ok
• .../resultmajor#error
Als ResultMinor sind die beiden folgenden Werte möglich:
• .../resultminor/al/common#internalError
• .../resultminor/arl/unknownAOID</td></tr>
<tr><td>AOID</td><td>Enthält den Identifikator des Archivdatenobjektes.</td></tr>
<tr><td>VersionID</td><td>Enthält die Version des Archivdatenobjektes auf das sich die Beweisdaten beziehen.</td></tr>
<tr><td>evidenceRecord</td><td>Sofern die übergebene AOID bekannt ist und deshalb vom ArchiSig-Modul entsprechende Evidence Records[2] gemäß [RFC4998] bzw. [ERSXML] konstruiert werden können, werden diese hier zurückgeliefert. Die detaillierte Struktur dieses Elementes ist nachfolgend erläutert.</td></tr>
<tr><td colspan="2"> 

Das evidenceRecord-Element ist vom Typ tr:EvidenceRecordType, der als Erweiterung des ec:EvidenceRecordType aus [eCard-2] definiert ist und zusätzlich die beiden String-basierten Attribute credentialID und relatedObjects enthält, die in [TR-ESOR-F] näher erläutert sind.</td></tr>
<tr><td colspan="2">Name</td><td>Beschreibung</td></tr>
</table>

[2] Sofern die TR-ESOR-Middleware mehrere redundante Hashbäume pflegt, werden hier mehrere Evidence Records zurückgeliefert.

<table>
<tr><td>Name</td><td colspan="2"><code>ArchiveEvidenceResponse</code></td></tr>
<tr><td rowspan="4"></td><td><code>xmlEvidenceRecord</code></td><td>Enthält einen XML-basierten Evidence Record gemäß [ERSXML].</td></tr>
<tr><td><code>asn1EvidenceRecord</code></td><td>Enthält einen ASN.1-basierten Evidence Record gemäß [RFC 4998].</td></tr>
<tr><td colspan="2"> 

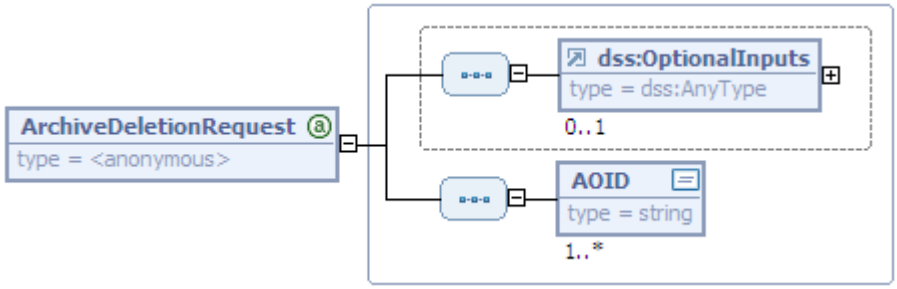
Statusinformationen und Fehler bei <code>ArchiveRetrievalResponse</code> (vgl. [eCard-1]).</td></tr>
<tr><td colspan="2">Name | Fehlercode
<code>ResultMajor</code> | • /resultmajor#ok • /resultmajor#error • /resultmajor#warning
<code>ResultMinor</code> | • /resultminor/al/common#noPermission • /resultminor/al/common#internalError • /resultminor/al/common#parameterError • /resultminor/arl/unknownAOID • /resultminor/arl/unknownOptionalInfo • /resultminor/arl/requestOnlyPartlySuccessfulWarning</td></tr>
</table>

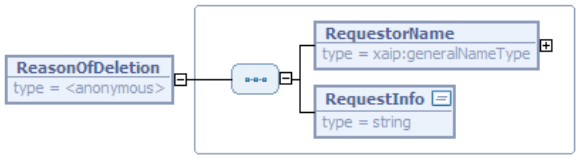
3.5 ArchiveDeletionRequest und ArchiveDeletionResponse

Mit der Funktion `ArchiveDeletionRequest` kann eine Folge von Archivdatenobjekten über die TR-ESOR-Middleware aus dem ECM-/Langzeitspeichersystem gelöscht werden.

Wie in Abbildung 2 ersichtlich, wird diese Funktion in den Schnittstellen S.4 (vgl. Abschnitt 4.5) und S.5 (vgl. Abschnitt 4.6) genutzt.

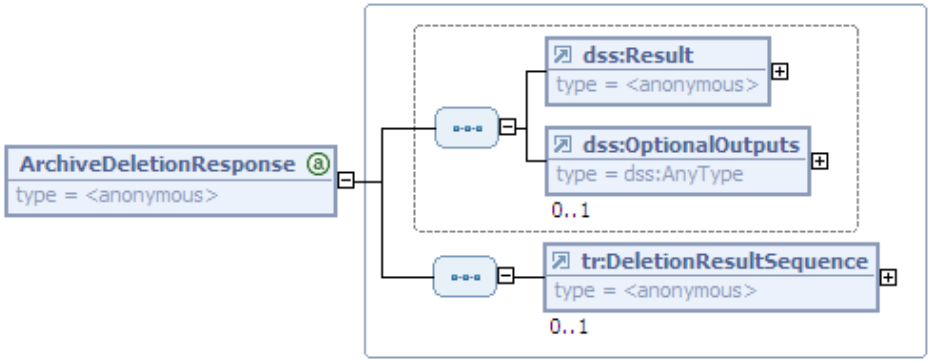
3.5.1 ArchiveDeletionRequest

<table>
<tr><td>Name</td><td colspan="2"><code>ArchiveDeletionRequest</code></td></tr>
<tr><td>Beschreibung</td><td colspan="2">Mit der Funktion <code>ArchiveDeletionRequest</code> kann eine Folge von im Langzeitspeicher abgelegten Archivdatenobjekten (<code>xaip:XAIP</code>) gelöscht werden.</td></tr>
<tr><td>Beschreibung</td><td colspan="2"> 

Aufruf der <code>ArchiveDeletionRequest</code>-Funktion</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
<tr><td></td><td><code>dss:OptionalInputs</code></td><td>Ist für optionale Eingabeelemente vorgesehen und kann beispielsweise Steuerelemente (<code>requestControls</code>) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen.
Insbesondere muss bei einer vorzeitigen Löschung das folgende Element genutzt werden:
 

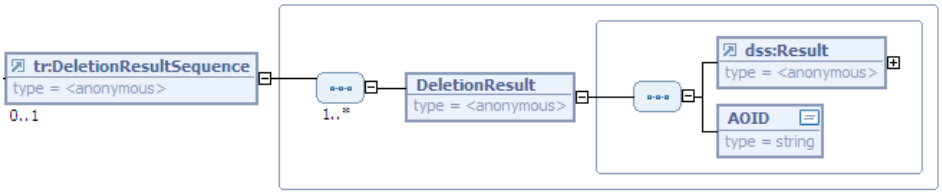
Das <code>ReasonOfDeletion</code>-Element muss vorhanden sein, sofern die Aufbewahrungsdauer noch nicht abgelaufen ist und enthält neben dem Namen der aufrufenden Instanz auch eine Begründung für die Löschung.
Die gesamte Aktion einschließlich der Begründung muss protokolliert werden.</td></tr>
<tr><td></td><td><code>AOID</code></td><td>Kann mehrmals vorhanden sein und gibt an, welche Archivdatenobjekte gelöscht werden sollen.</td></tr>
</table>

3.5.2 ArchiveDeletionResponse

<table>
<tr><td>Name</td><td colspan="2"><code>ArchiveDeletionResponse</code></td></tr>
<tr><td>Beschreibung</td><td colspan="2">Als Antwort auf einen <code>ArchiveDeletionRequest</code> wird ein entsprechendes <code>ArchiveDeletionResponse</code>-Element zurückgeliefert, das Informationen über den Erfolg oder Misserfolg der Anfrage enthält.</td></tr>
<tr><td>Rückgabe</td><td colspan="2"> 

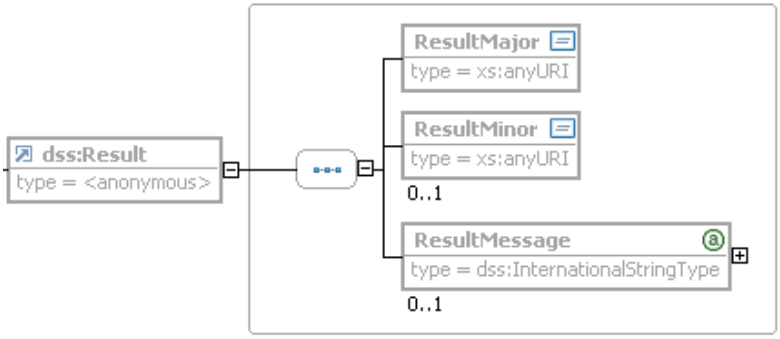
<code>ArchiveDeletionResponse</code> ist die Antwort zum <code>ArchiveDeletionRequest</code>-Aufruf
<table>
<tr><th>Name</th><th>Beschreibung</th></tr>
<tr><td><code>dss:Result</code></td><td>Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in [eCard-1] und unten näher beschrieben. Sofern nur ein Teil der für die Löschung vorgesehenen Archivdatenobjekte gelöscht werden konnte, wird dies durch den Fehlercode .../resultminor/arl/request-OnlyPartlySuccessfulWarning angezeigt.</td></tr>
<tr><td><code>dss:OptionalOutputs</code></td><td>Ist für optionale Ausgabeelemente vorgesehen und kann beispielsweise entsprechende Steuerelemente (<code>responseControls</code>) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen.</td></tr>
<tr><td><code>tr:DeletionResult Sequence</code></td><td>Sofern kein schwerwiegender Fehler aufgetreten ist, wird das <code>tr:DeletionResultSequence</code>-Element zurückgeliefert, das Informationen über den Erfolg oder Misserfolg der individuellen Löschungen enthält.
Die detaillierte Struktur dieses Elementes ist nachfolgend visualisiert.</td></tr>
</table></td></tr>
</table>

Name	`ArchiveDeletionResponse`



Das `DeletionResultSequence`-Element kann im `ArchiveDeletionResponse` zurückgeliefert werden. Es enthält eine Folge von `DeletionResult`-Elementen, die wiederum die folgenden beiden Kindelemente besitzen.

Name	Beschreibung
`dss:Result`	Gibt an, ob das die angeforderte Löschung des Archivdatenobjektes durchgeführt werden konnte oder nicht. Als `ResultMajor` sind die beiden folgenden Werte möglich: • .../resultmajor#ok • .../resultmajor#error Als `ResultMinor` sind die beiden folgenden Werte möglich: • .../resultminor/al/common#noPermission • .../resultminor/arl/unknownAOID • .../resultminor/arl/missingReasonOfDeletion
`AOID`	Gibt an, auf welches Archivdatenobjekt sich das zurückgelieferte Ergebnis bezieht.



Statusinformationen und Fehler bei `ArchiveDeletionResponse` (vgl. [eCard-1]).

Name	Fehlercode

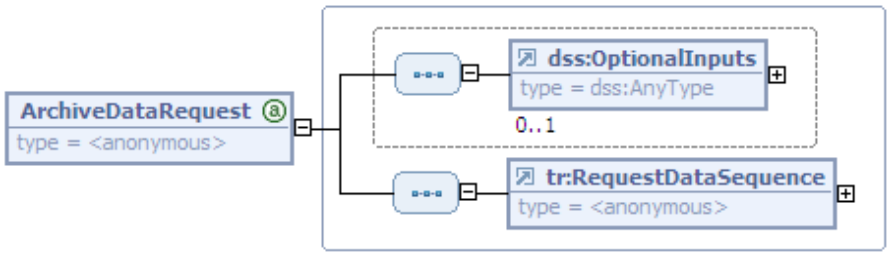
Name	ArchiveDeletionResponse	
	ResultMajor	• /resultmajor#ok • /resultmajor#error • /resultmajor#warning
	ResultMinor	• /resultminor/al/common#noPermission • /resultminor/al/common#internalError • /resultminor/al/common#parameterError • /resultminor/arl/unknownAOID • /resultminor/arl/unknownOptionalInfo • /resultminor/arl/requestOnlyPartlySuccessfulWarning

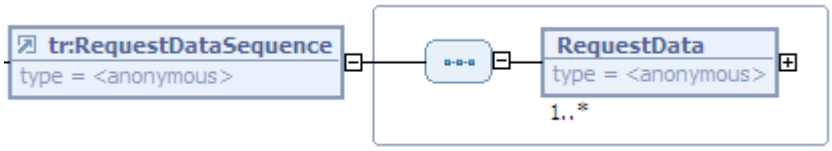
3.6 ArchiveDataRequest und ArchiveDataResponse

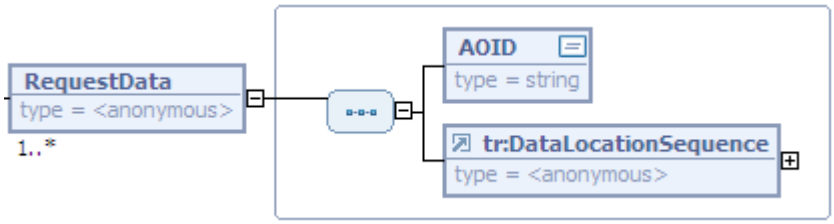
Mit der Funktion `ArchiveDataRequest` können diskrete Datenelemente aus einem bereits abgelegten Archivdatenobjekt (`xaip:XAIP`) ausgelesen werden.

Wie in Abbildung 2 ersichtlich, wird diese Funktion in den Schnittstellen S.4 (vgl. Abschnitt 4.5) und S.5 (vgl. Abschnitt 4.6) genutzt.

3.6.1 ArchiveDataRequest

Name	ArchiveDataRequest	
Beschreibung	Mit der Funktion `ArchiveDataRequest` können diskrete Datenelemente aus einem im ECM-/Langzeitspeichersystem abgelegten, zumindest logisch im `xaip:XAIP`-Format gemäß [TR-ESOR-F] vorliegenden, Archivdatenobjekt ausgelesen werden.	
Beschreibung	 Aufruf der `ArchiveDataRequest`-Funktion	
	Name	**Beschreibung**
	`dss:OptionalInputs`	Ist für optionale Eingabeelemente vorgesehen und kann beispielsweise Steuerelemente (`requestControls`) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen.

<table>
<tr><td>Name</td><td colspan="2">ArchiveDataRequest</td></tr>
<tr><td rowspan="8"></td><td>tr:RequestData Sequence</td><td>Enthält eine Folge von RequestData-Elementen, deren Struktur nachfolgend beschrieben ist.</td></tr>
<tr><td colspan="2"> 

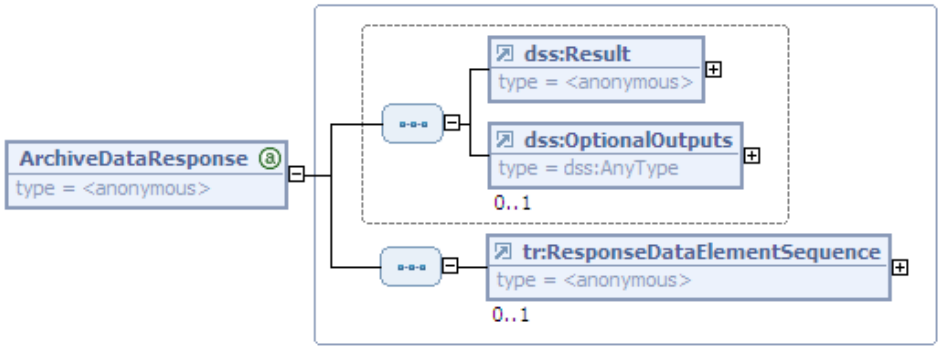
Die tr:RequestDataSequence wird im ArchiveDataRequest übergeben.</td></tr>
<tr><td>Name</td><td>Beschreibung</td></tr>
<tr><td>RequestData</td><td>Enthält die Information welche Teile eines bestimmten Archivdatenobjekts zurückgeliefert werden sollen.</td></tr>
<tr><td colspan="2"> 

Im RequestData-Element wird spezifiziert, welche Teile eines bestimmten Archivdatenobjektes zurückgeliefert werden sollen.</td></tr>
<tr><td>Name</td><td>Beschreibung</td></tr>
<tr><td>AOID</td><td>Enthält den Identifikator des adressierten Archivdatenobjektes.</td></tr>
<tr><td>tr:DataLocation Sequence</td><td>Enthält eine Folge von DataLocation-Elementen, durch die die „Lokation“ der auszulesenden diskreten Datenelemente - bezüglich eines zumindest logisch im xaip:XAIP-Format gemäß [TR-ESOR-F] vorliegenden Archivdatenobjektes - spezifiziert sind.[3]</td></tr>
<tr><td colspan="2">Das DataLocation-Element spezifiziert, welche Teile eines Archivobjektes zurückgeliefert werden sollen und ist folgendermaßen definiert:
<pre>
<element name="DataLocation">
  <complexType>
      <simpleContent>
            <extension base="string">
                  <attribute name="Type" type="anyURI"/>
            </extension>
      </simpleContent>
  </complexType>
</element>
</pre>
Im Type-Attribut wird angegeben, welche Transformation für den Zugriff auf</td></tr>
</table>

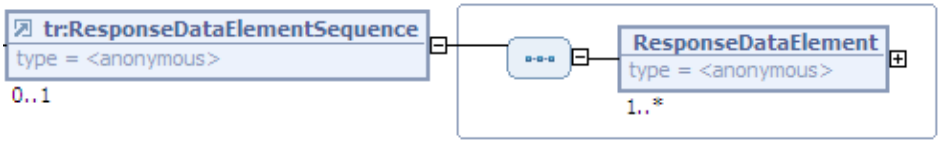
[3] Ausgehend von einem XML-basierten Archivdatenobjekt bieten sich für die diskrete Adressierung von XML-Datenelementen hier XPath (siehe unter: http://www.w3.org/TR/2007/REC-xpath20-20070123/), XQuery (siehe unter: http://www.w3.org/TR/2007/REC-xquery-20070123/) oder die XML Pointer Language XPointer (siehe unter: http://www.w3.org/TR/2003/REC-xptr-framework-20030325/) an.

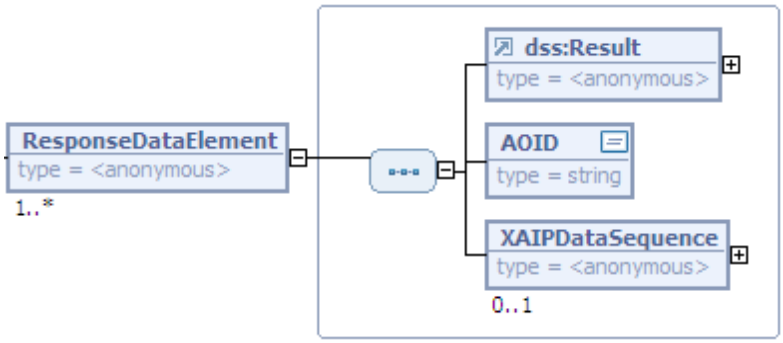
Name	`ArchiveDataRequest`
	die gewünschten Daten angewandt werden soll, wobei die folgenden URIs vorgesehen sind: • http://www.w3.org/TR/2007/REC-xpath20-20070123/ für XPath, • http://www.w3.org/TR/2007/REC-xquery-20070123/ für XQuery und • http://www.w3.org/TR/2003/REC-xptr-framework-20030325 für XPointer

3.6.2 ArchiveDataResponse

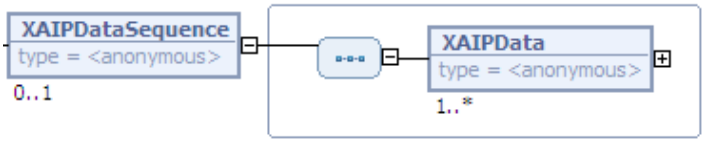
Name	`ArchiveDataResponse`
Beschreibung	Als Antwort auf einen `ArchiveDataRequest` wird ein entsprechendes `ArchiveDataResponse`-Element zurückgeliefert, das die gewünschten Informationen enthält.
Rückgabe	 `ArchiveDataResponse` ist die Antwort zum `ArchiveDataRequest`-Aufruf

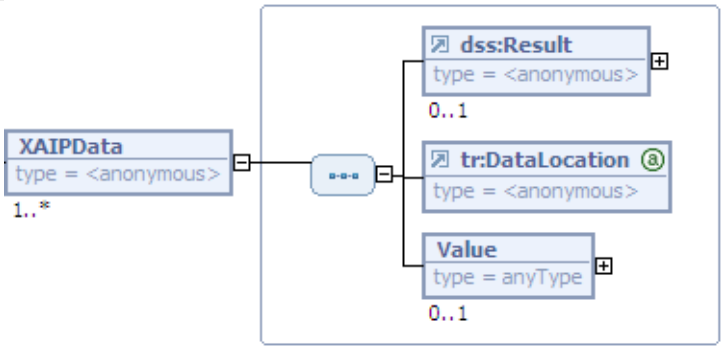
Name	Beschreibung
`dss:Result`	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in [eCard-1] und unten näher beschrieben. Sofern nur ein Teil der angefragten diskreten Datenobjekte zurückgeliefert werden konnte, wird dies durch den Fehlercode /resultminor/arl/requestOnlyPartlySuccessfulWarning angezeigt.
`dss:OptionalOutputs`	Ist für optionale Ausgabeelemente vorgesehen und kann beispielsweise entsprechende Steuerelemente (`responseControls`) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen.

<table>
<tr><td>Name</td><td colspan="2"><code>ArchiveDataResponse</code></td></tr>
<tr><td rowspan="4"></td><td><code>tr:ResponseDataElementSequence</code></td><td>Sofern kein Fehler aufgetreten ist, wird das <code>tr:ResponseDataElementSequence</code> Element zurückgeliefert, das die gewünschten Informationen enthält.
Die detaillierte Struktur dieses Elementes ist nachfolgend visualisiert.</td></tr>
<tr><td colspan="2"> 

Das <code>ResponseDataElementSequence</code>-Element kann im <code>ArchiveDataResponse</code> zurückgeliefert werden. Es enthält eine Folge von <code>ResponseDataElement</code>-Elementen, deren Struktur nachfolgend dargestellt und erläutert wird.</td></tr>
<tr><td colspan="2"> 

Das <code>ResponseDataElement</code> enthält das Ergebnis der Anfrage und im Erfolgsfall die gewünschten Daten.</td></tr>
<tr><td colspan="2"><table>
<tr><th>Name</th><th>Beschreibung</th></tr>
<tr><td><code>dss:Result</code></td><td>Gibt an, ob die Anfrage erfolgreich durchgeführt werden konnte oder nicht.
Als <code>ResultMajor</code> sind die beiden folgenden Werte möglich:
.../resultmajor#ok
.../resultmajor#error
Als <code>ResultMinor</code> sind die folgenden Werte möglich:
.../resultminor/al/common#noPermission
.../resultminor/arl/unknownAOID
.../resultminor/arl/unknownLocation
.../resultminor/al/common#parameterError</td></tr>
</table></td></tr>
</table>

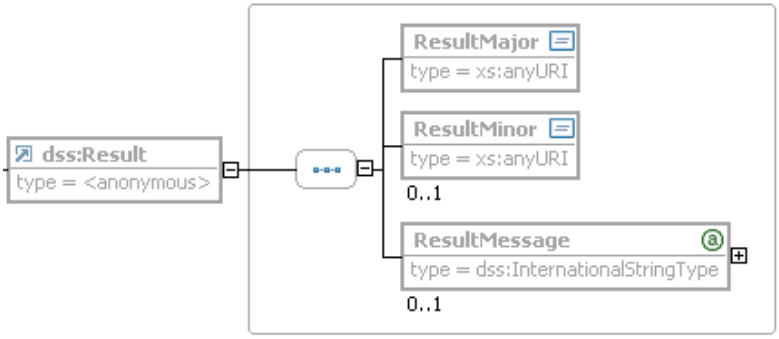
<table>
<tr><td>Name</td><td colspan="2">ArchiveDataResponse</td></tr>
<tr><td rowspan="6"></td><td>AOID</td><td>Ist der eindeutigen Identifikator des Archivdatenobjektes.</td></tr>
<tr><td>XAIPDataSequence</td><td>Im Erfolgsfall wird in diesem Element die gewünschte Folge der diskreten Datenelemente zurückgeliefert. Die Struktur der XAIPDataSequence ist nachfolgend erläutert.</td></tr>
<tr><td colspan="2"> 

Die XAIPDataSequence ist im Erfolgsfall Teil des ResponseDataElements und enthält eine Folge von XAIPData-Elementen.</td></tr>
<tr><td>Name</td><td>Beschreibung</td></tr>
<tr><td>XAIPData</td><td>Enthält im Erfolgsfall die gewünschten Daten und die „Lokation“ aus der diese aus der im ECM-/Langzeitspeichersystem zumindest logisch existierenden XAIP-Struktur ausgelesen wurden. Die detaillierte Struktur dieses Elementes ist nachfolgend dargestellt und erläutert.</td></tr>
<tr><td colspan="2"> 

Das XAIPData-Element enthält im Erfolgsfall die gewünschten Daten.</td></tr>
<tr><td>Name</td><td>Beschreibung</td></tr>
</table>

<table>
<tr><th>Name</th><td colspan="2">ArchiveDataResponse</td></tr>
<tr><td></td><td>dss:Result</td><td>Gibt an, ob die Anfrage erfolgreich durchgeführt werden konnte oder nicht.
Als ResultMajor sind die beiden folgenden Werte möglich:
• .../resultmajor#ok
• .../resultmajor#error

Als ResultMinor sind die folgenden Werte möglich:
• .../resultminor/arl/unknownLocation
• .../resultminor/al/common#parameterError

Im Erfolgsfall (ResultMajor = …#ok) kann auf das dss:Result-Element verzichtet werden.</td></tr>
<tr><td></td><td>tr:DataLocation</td><td>Das DataLocation-Element spezifiziert, welche Teile eines Archivobjektes zurückgeliefert werden sollen. Weitere Details zu diesem Element finden sich auf Seite 27.</td></tr>
<tr><td></td><td>Value</td><td>Enthält im Erfolgsfall die gewünschten Daten.</td></tr>
<tr><td></td><td colspan="2"> 

Statusinformationen und Fehler bei ArchiveDataResponse (vgl. [eCard-1]).</td></tr>
<tr><td></td><td>Name</td><td>Fehlercode</td></tr>
<tr><td></td><td>ResultMajor</td><td>• /resultmajor#ok
• /resultmajor#error
• /resultmajor#warning</td></tr>
<tr><td></td><td>ResultMinor</td><td>• /resultminor/al/common#noPermission
• /resultminor/al/common#internalError
• /resultminor/al/common#parameterError
• /resultminor/arl/unknownAOID
• /resultminor/arl/unknownOptionalInfo
• /resultminor/arl/unknownLocation
• /resultminor/arl/requestOnlyPartlySuccessful-Warning</td></tr>
</table>

3.7 Funktionen des eCard-API-Frameworks

Neben den sechs in dieser Richtlinie eingeführten Funktionen (vgl. Abschnitt 3.1 - 3.6) werden für die Zwecke des Beweiswerterhalts kryptographisch signierter Dokumenteauch die folgenden Funktionen des eCard-API-Frameworks [BSI-TR-03112] genutzt:

- `VerifyRequest / VerifyResponse` (vgl. [eCard-2], Abschnitt 3.2.1)
- `SignRequest / SignResponse` (vgl. [eCard-2], Abschnitt 3.2.2)
- `Hash / HashResponse` (vgl. [eCard-4], Abschnitt 3.5.4)

4. Nutzung der Funktionen in den verschiedenen Schnittstellen der TR-ESOR-Middleware

In diesem Abschnitt wird erläutert, wie die in Abbildung 1 und 2 dargestellte Referenzarchitektur und die in [TR-ESOR-S] eingeführten Schnittstellen auf Basis der in Abschnitt 3 vorgestellten Funktionen umgesetzt werden können.

Hierfür werden die verschiedenen Schnittstellen S.1-S.6 (vgl. Abbildung 2) betrachtet und deren Umsetzung als Webservice (vgl. Abschnitt 4.1) auf Basis der oben eingeführten Funktionen erläutert:

- TR-ESOR-S.1 (ArchiSafe-Modul – Krypto-Modul) (siehe Abschnitt 4.2)
- TR-ESOR-S.2 (ArchiSig-Modul – ECM-/Langzeitspeichersystem) (siehe Abschnitt 4.3)
- TR-ESOR-S.3 (ArchiSig-Modul – Krypto-Modul) (siehe Abschnitt 4.4)
- TR-ESOR-S.4 (Geschäftsanwendung– ArchiSafe-Modul) (siehe Abschnitt 4.5)
- TR-ESOR-S.5 (ArchiSafe-Modul –ECM-/Langzeitspeichersystem) (siehe Abschnitt 4.6)
- TR-ESOR-S.6 (ArchiSafe-Modul – ArchiSig-Modul) (siehe Abschnitt 4.7)

4.1 Grundsätzliches zu einer XML-basierten Realisierung der Schnittstellen

Während in den ASN.1-Strukturen [TR-ESOR-S] für die Angabe des verwendeten Protokolls bzw. der verwendeten API-Version stets ein explizites Element `Version` enthalten ist, kann bei den als XML-Schema definierten Strukturen (vgl. Abschnitt 5.1) und den darauf aufbauenden Webservice-Schnittstellen (vgl. Abschnitt 5.2) darauf verzichtet werden, da die Version der Struktur implizit durch den verwendeten Namensraum „`http://www.bsi.bund.de/tr-esor/api/1.1`“ spezifiziert ist.

Außerdem werden jeweils statt der generischen `Controls`-Elemente in der ASN.1-Struktur die entsprechenden `dss:OptionalInputs`- und `dss:OptionalOutputs`-Elemente aus [OASIS-DSS] genutzt, die bei Bedarf auch weitere Elemente enthalten können.

Falls die in diesem Dokument beschriebenen Schnittstellen und Funktionen asynchron genutzt werden sollen, soll dies unter Verwendung der hierfür vorgesehenen Mechanismen aus [OASIS-Async] realisiert werden.

4.2 TR-ESOR-S.1 (ArchiSafe-Modul – Krypto-Modul)

Dieser Abschnitt beschreibt, wie die in [TR-ESOR-S] skizzierte Schnittstelle S.1 auf Basis des eCard-API-Frameworks (BSI TR 03112) umgesetzt werden kann.

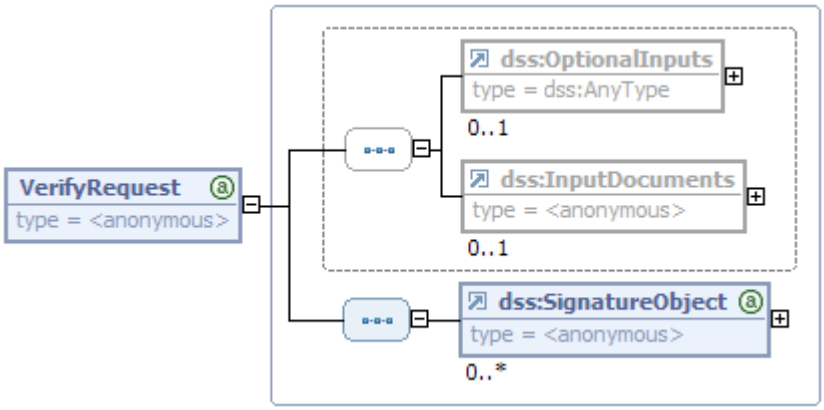
Die in [TR-ESOR-S] definierte Schnittstelle S.1 umfasst zwei wesentliche Funktionen:

- Signaturprüfung (`VerifyRequest / VerifyResponse`)
- Signaturerzeugung (optional) (`SignRequest / SignResponse`)

4.2.1 Signaturprüfung

Für die Signaturprüfung sind in [TR-ESOR-S] die Schnittstellensignaturen `VerifyRequest` und `VerifyResponse` als ASN.1-Strukturen definiert. Dies entspricht den gleichnamigen XML-Strukturen in [OASIS-DSS] und [eCard-2].

4.2.1.1 VerifyRequest

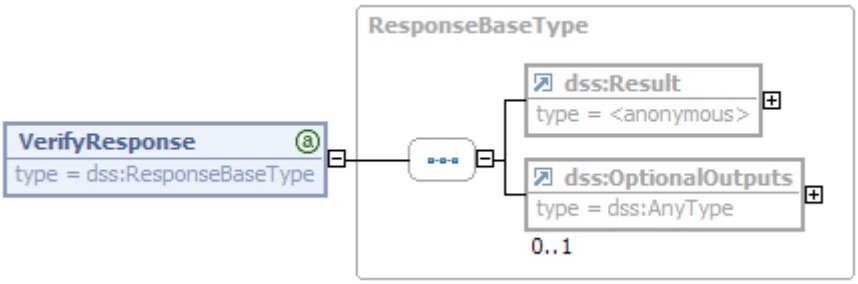
Name	`VerifyRequest`
Beschreibung	Mit der Funktion `VerifyRequest` (vgl. Abschnitt 3.2.2 von [eCard-2]) im Kontext der Schnittstelle S.1 werden die in einem Archivdatenobjekt (XAIP-Dokument) enthaltenen Signaturinformationen (Signaturen, Zeitstempel, Zertifikate, Sperrlisten, OCSP-Responses etc.) geprüft.
Aufrufparameter	 Aufruf der `VerifyRequest`-Funktion.

Name	Beschreibung

Name	**`VerifyRequest`**	
	`dss:OptionalInputs`	Entspricht dem `requestControls`-Element und kann zusätzliche Eingabeelemente enthalten. Hierbei sollen insbesondere die in [eCard-2] definierten Elemente und Aufrufoptionen unterstützt werden. Sofern in einem `dss:Document`-Kindelement von `dss:InputDocuments` ein `XAIP`-Element gemäß [TR-ESOR-F] enthalten ist, kann mit dem Element `VerifyUnderSignaturePolicy` und der im `DefaultPolicy/SignaturePolicyIdentifier`-Element angegebenen Signature-Policy `http://www.bsi.bund.de/tr-esor/sigpolicy/verify-xaip` die Prüfung und Ergänzung aller im übergebenen XAIP-Container enthaltenen Signaturen angefordert werden. Hierbei müssen alle Signaturinformationen (Signaturen, Zeitstempel, Zertifikate, Sperrlisten, OCSP-Responses etc.) bis hin zu einer vertrauenswürdigen Wurzel geprüft werden. Die hierbei ermittelten Prüfinformationen (Zertifikate, Sperrlisten, OCSP-Responses) werden nach Möglichkeit in den entsprechenden Signaturen bzw. im `VerificationReport`-Element gemäß [eCard-2] im XAIP abgelegt.

Name	`VerifyRequest`	
	`dss:InputDocuments`	Das `dss:InputDocuments`-Element kann die zur Prüfung benötigten Dokumente enthalten, sofern diese nicht bereits im unten erläuterten `SignatureObject`-Element enthalten sind. Außerdem kann in einem `dss:Document`-Kindelement ein `XAIP`-Element gemäß [TR-ESOR-F] übergeben werden, so dass alle darin enthaltenen Signaturen in Verbindung mit der oben angegebenen Signature-Policy geprüft und ergänzt werden.
	`dss:SignatureObject`	In `dss:SignatureObject`-Elementen können grundsätzlich eigenständige Signaturen (detached signatures) zur Prüfung übergeben werden. Wenn Signaturen bereits im `dss:InputDocuments` enthalten sind, können die optionalen `dss:SignatureObject`-Elemente entfallen.

4.2.1.2 VerifyResponse

Name	`VerifyResponse`	
Beschreibung	Als Antwort auf einen `VerifyRequest` wird vom Krypto-Modul ein entsprechendes `VerifyResponse`-Element gemäß Abschnitt 3.2.2 von [eCard-2] zurückgeliefert.	
Rückgabe	 Rückgabe der `VerifyRequest`-Funktion	
	Name	**Beschreibung**
	`dss:Result`	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements und die möglichen Fehlercodes sind in Abschnitt 4.1.2 von [eCard-1] und Abschnitt 3.2.2 von [eCard-2] beschrieben.
	`dss:OptionalOutputs`	Sofern nicht ein Fehler aufgetreten ist, enthält dieses Element entweder den Prüfbericht in Form eines `VerificationReport`-Elementes

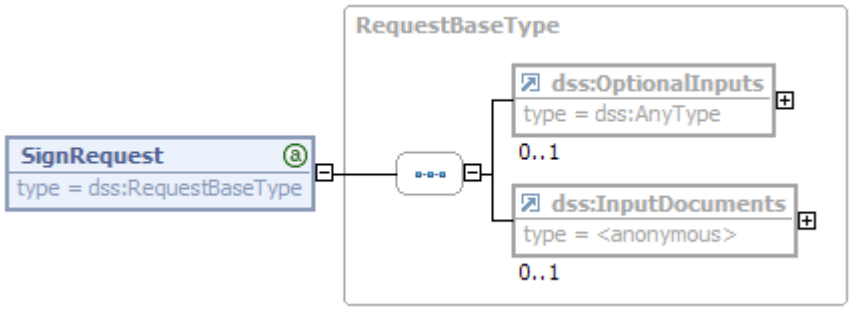
Name	`VerifyResponse`	
		oder das um diese Prüfinformationen ergänzte XAIP-Dokument in Form eines `xaip:XAIP`-Elements. Die Struktur des Prüfberichtes ist in [OASIS-VR] näher beschrieben. Details zur Ablage dieser Prüfinformationen im XAIP-Container finden sich in [TR-ESOR-F].

4.2.2 Signaturerstellung

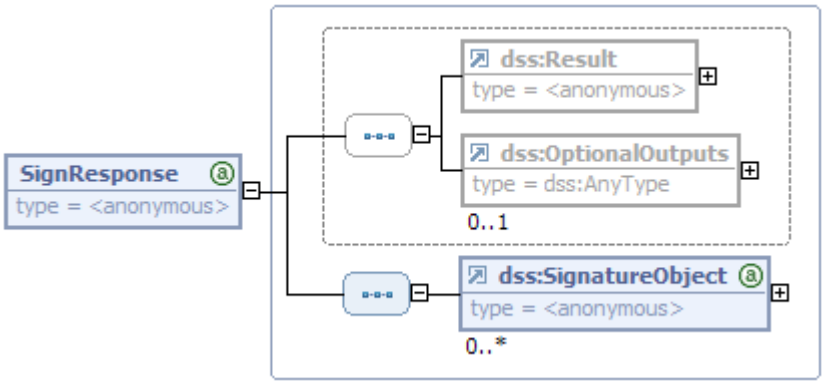
Für die Signaturerstellung sind in [TR-ESOR-S] die ASN.1-Strukturen `SignRequest` und `SignResponse` definiert. Dies entspricht den gleichnamigen XML-Strukturen in [OASIS-DSS] und [eCard-2].

4.2.2.1 SignRequest

Ein `SignRequest` im Kontext der Schnittstelle S.1 übergibt ein Archivdatenobjekt (XAIP-Dokument) an das Krypto-Modul zur Erzeugung einer elektronischen Signatur.

Name	`SignRequest`	
Beschreibung	Mit der Funktion `SignRequest` aus [eCard-2] kann das übergebene Archivdatenobjekt mit einer (qualifizierten) elektronischen Signatur versehen werden.	
Beschreibung	 Aufruf der `SignRequest`-Funktion	
	Name	**Beschreibung**
	`dss:OptionalInputs`	Entspricht dem `requestControls`-Element aus [TR-ESOR-S] und kann eines oder mehrere der in [eCard-2] definierten optionalen Eingabeelemente enthalten.
	`dss:InputDocuments`	Enthält die zu signierenden Dokumente oder Datenstrukturen. Weitere Informationen hierzu finden sich in [OASIS-DSS] und [eCard-2].

4.2.2.2 SignResponse

<table>
<tr><td>Name</td><td colspan="2">SignResponse</td></tr>
<tr><td>Beschreibung</td><td colspan="2">Als Antwort auf einen SignRequest wird vom Krypto-Modul ein entsprechendes SignResponse-Element gemäß Abschnitt 3.2.1 von [eCard-2] zurückgeliefert.</td></tr>
<tr><td>Rückgabe</td><td colspan="2"> 

SignResponse ist die Antwort zum SignRequest-Aufruf
<table>
<tr><th>Name</th><th>Beschreibung</th></tr>
<tr><td>dss:Result</td><td>Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements und die möglichen Fehlercodes sind in Abschnitt 4.1.2 von [eCard-1] und in Abschnitt 3.2.1 von [eCard-2] beschrieben.</td></tr>
<tr><td>dss:OptionalOutputs</td><td>Kann ein DocumentWithSignature-Element enthalten, in denen z.B. ein XAIP-Element mit der eingebetteten Signatur enthalten ist. Details finden sich in Abschnitt 3.2.1 von [eCard-2].</td></tr>
<tr><td>dss:SignatureObject</td><td>Kann eine erzeugte Signatur in Form eines dss:SignatureObject-Elementes enthalten. Details finden sich in Abschnitt 3.2.1 von [eCard-2]. Sofern die erstellte Signatur bereits im o.g. DocumentWithSignature-Element vorhanden ist, wird kein dss:SignatureObject-Element zurückgeliefert.</td></tr>
</table></td></tr>
</table>

4.3 TR-ESOR-S.2 (ArchiSig-Modul – ECM-/Langzeitspeichersystem)

Dieser Abschnitt beschreibt, wie die in [TR-ESOR-S] skizzierte Schnittstelle S.2 auf Basis der auch dem eCard-API-Frameworks (BSI TR 03112) zu Grunde liegenden Basistypen aus [OASIS-DSS] umgesetzt werden kann.

Diese Schnittstelle umfasst drei wesentliche Funktionen:

- Speichern eines Archivdatenobjektes (ArchiveSubmissionRequest / ArchiveSubmissionResponse)
- Ändern eines Archivdatenobjektes (ArchiveUpdateRequest / ArchiveUpdateResponse)

- Auslesen eines Archivdatenobjektes (`ArchiveRetrievalRequest / ArchiveRetrievalResponse`)

4.3.1 Speichern eines Archivdatenobjektes

Für das Speichern eines Archivdatenobjektes sind in [TR-ESOR-S] die Schnittstellensignaturen `ArchiveSubmissionRequest` und `ArchiveSubmissionResponse` als ASN.1-Strukturen definiert. Dies entspricht den gleichnamigen und in Abschnitt 3.1 näher beschriebenen XML-Strukturen, die analog zu [eCard-2] unter Verwendung der grundlegenden Schnittstellentypen (`dss:RequestBaseType` und `dss:ResponseBaseType`) aus [OASIS-DSS] spezifiziert wurden.

4.3.2 Ändern von Archivdatenobjekten

Für das Ändern von Archivdatenobjekten sind in [TR-ESOR-S] die Schnittstellensignaturen `ArchiveUpdateRequest` und `ArchiveUpdateResponse` als ASN.1-Struktur definiert. Dies entspricht den gleichnamigen und in Abschnitt 3.2 näher beschriebenen XML-Strukturen, die analog zu [eCard-2] unter Verwendung der grundlegenden Schnittstellentypen (`dss:RequestBaseType` und `dss:ResponseBaseType`) aus [OASIS-DSS] spezifiziert wurden.

4.3.3 Auslesen von Archivdatenobjekten

Für das Auslesen von Archivdatenobjekten sind in [TR-ESOR-S] die Schnittstellensignaturen `ArchiveRetrievalRequest` und `ArchiveRetrievalResponse` als ASN.1-Struktur definiert. Dies entspricht den gleichnamigen und in Abschnitt 3.3 näher beschriebenen XML-Strukturen, die analog zu [eCard-2] unter Verwendung der grundlegenden Schnittstellentypen (`dss:RequestBaseType` und `dss:ResponseBaseType`) aus [OASIS-DSS] spezifiziert wurden.

4.4 TR-ESOR-S.3 (ArchiSig-Modul – Krypto-Modul)

Dieser Abschnitt beschreibt, wie die in [TR-ESOR-S] skizzierte Schnittstelle S.3 auf Basis des eCard-API-Frameworks (BSI TR 03112) umgesetzt werden kann.

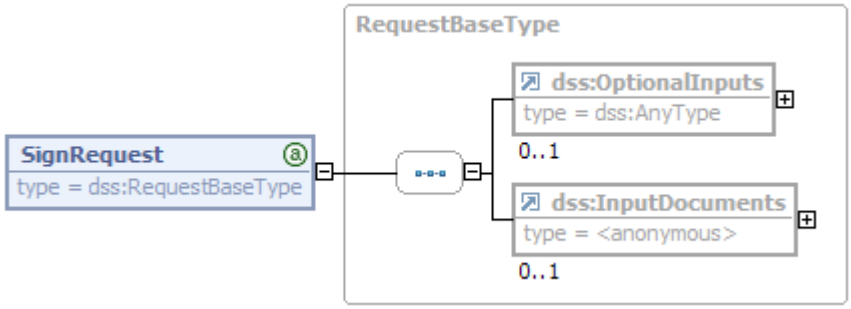
Die in [TR-ESOR-S] definierte Schnittstelle S.3 umfasst drei wesentliche Funktionen:

- Anfordern eines (qualifizierten) Zeitstempels (`TimestampRequest / TimeStampResponse`)
- Prüfen eines (qualifizierten) Zeitstempels (`VerifyRequest / VerifyResponse`)
- Berechnung eines Hashwertes (`Hash / HashResponse`)

4.4.1 Anfordern eines (qualifizierten) Zeitstempels

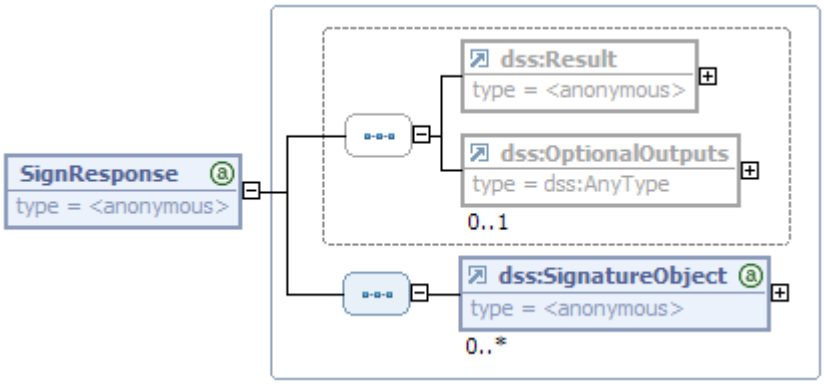
Zum Anfordern eines (qualifizierten) Zeitstempels sind in S.3 die auf [RFC3161] zurückgehenden ASN.1-Strukturen `TimestampRequest / TimestampResponse` vorgesehen. Dies entspricht der [OASIS-DSS]-basierten Funktion `SignRequest / SignResponse` aus [eCard-2].

4.4.1.1 TimestampRequest wird realisiert durch SignRequest

Name	`SignRequest`
Beschreibung	Ein `SignRequest` im Kontext der Schnittstelle S.3 übergibt einen Hashwert, zu dem ein (qualifizierter) Zeitstempel erstellt werden soll, an das Krypto-Modul.
Beschreibung	 Aufruf der `SignRequest`-Funktion

Name	Beschreibung
`dss:OptionalInputs`	Enthält genau ein Element `SignatureType` mit der URI urn:ietf:rfc:3161 durch die klargestellt wird, dass ein Zeitstempel gemäß [RFC3161] erzeugt werden soll.
`dss:InputDocuments`	Während das Element `dss:InputDocuments` in [OASIS-DSS] und [eCard-2] optional ist, muss es hier vorhanden sein und genau ein `dss:Document`-Element in der `DocumentHash`-Ausprägung enthalten. Dieses Element enthält den Hashwert, aus dem ein (qualifizierter) Zeitstempel erzeugt werden soll.

4.4.1.2 TimestampResponse wird realisiert durch SignResponse

Name	`SignResponse`
Beschreibung	Als Antwort auf einen `SignRequest` wird vom Krypto-Modul ein entsprechendes `SignResponse`-Element gemäß Abschnitt 3.2.1 von [eCard-2] zurückgeliefert. Im Kontext der Schnittstelle S.3 wird hier ein (qualifizierter) Zeitstempel zurückgeliefert.
Rückgabe	 `SignResponse` ist die Antwort zum `SignRequest`-Aufruf

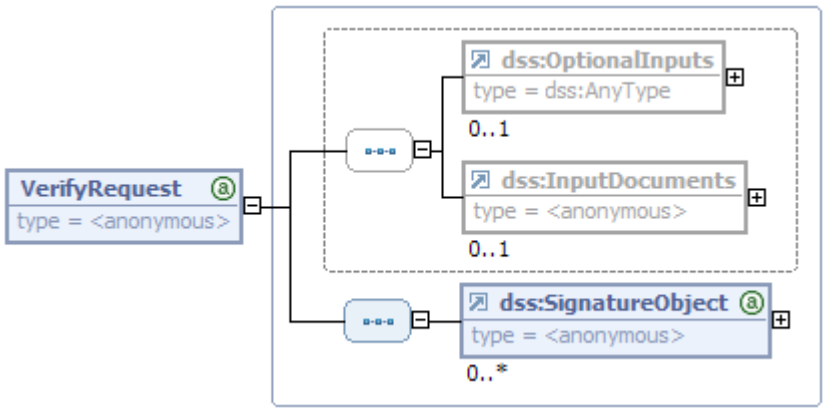
Name	Beschreibung

Name	`SignResponse`	
	`dss:Result`	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements und die möglichen Fehlercodes sind in Abschnitt 4.1.2 von [eCard-1] und in Abschnitt 3.2.1 von [eCard-2] beschrieben.
	`dss:OptionalOutputs`	Das optionale Element `dss:OptionalOutputs` ist nicht vorhanden.
	`dss:SignatureObject`	Enthält – sofern kein Fehler aufgetreten ist – genau ein `dss:SignatureObject`-Element, das ein `dss:Timestamp`-Element enthält, in dem der Zeitstempel in Form eines `RFC3161TimeStampToken`-Elementes enthalten ist.

4.4.2 Prüfen eines (qualifizierten) Zeitstempels

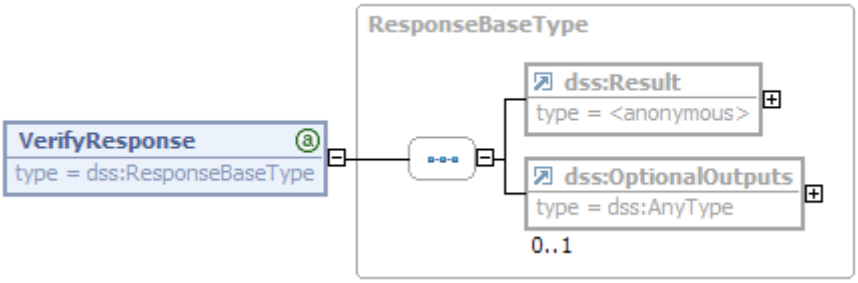
Zum Prüfen eines (qualifizierten) Zeitstempels sind in S.3 die ASN.1-Strukturen `VerifyRequest` / `VerifyResponse` vorgesehen. Dies entspricht den gleichnamigen Funktionen aus [OASIS-DSS] und [eCard-2].

4.4.2.1 VerifyRequest

Name	`VerifyRequest`	
Beschreibung	Ein `VerifyRequest` im Kontext der Schnittstelle S.3 übergibt einen (qualifizierten) Zeitstempel an das Krypto-Modul zur Verifikation der darin enthaltenen (qualifizierten) elektronischen Signatur. Außerdem werden die für die Prüfung genutzten Zertifikate und Sperrinformationen in den zurück gelieferten Zeitstempel eingefügt. Entsprechende Empfehlungen für die Ablage dieser Informationen finden sich in [TR-ESOR-F].	
Aufrufparameter	 Aufruf der `VerifyRequest`-Funktion.	
	Name	**Beschreibung**

Name	VerifyRequest	
	dss:OptionalInputs	Entspricht dem requestControls-Element aus [TR-ESOR-S] und enthält genau ein Element VerifyUnderSignaturePolicy mit dem DefaultPolicy/SignaturePolicyIdentifier http://www.bsi.bund.de/tr-esor/sigpolicy/verify-timestamp durch den spezifiziert wird, dass der Zeitstempel und alle Zertifikate bis hin zu einer vertrauenswürdigen Wurzel geprüft werden und die hierbei erhaltenen Prüfinformationen in der in [TR-ESOR-F] spezifizierten Weise in den Zeitstempel eingefügt werden müssen.
	dss:InputDocuments	Das optionale Element dss:InputDocuments soll nicht vorhanden sein und wird ignoriert.
	dss:SignatureObject	Es ist genau ein dss:SignatureObject-Element in der dss:TimeStamp/RFC3161TimeStampToken Ausprägung vorhanden, das den Zeitstempel enthält.

4.4.2.2 VerifyResponse

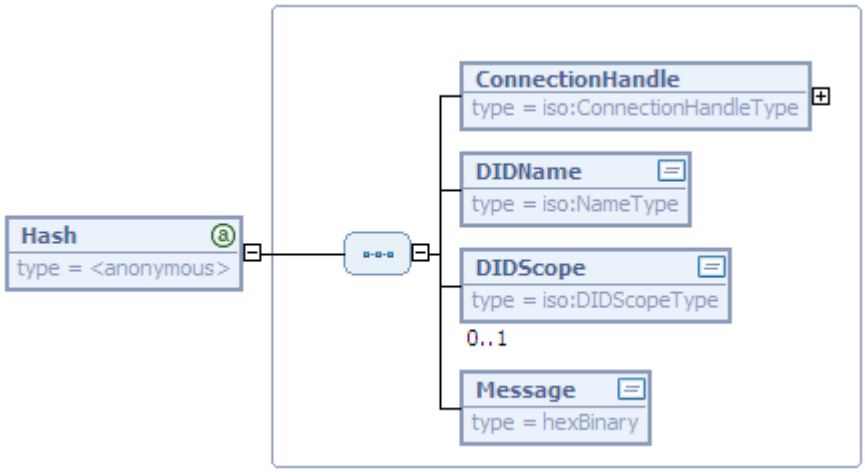
Name	VerifyResponse	
Beschreibung	Als Antwort auf einen VerifyRequest wird vom Krypto-Modul ein entsprechendes VerifyResponse-Element gemäß Abschnitt 3.2.2 von [eCard-2] zurückgeliefert.	
Rückgabe	 Rückgabe der VerifyRequest-Funktion	
	Name	**Beschreibung**
	dss:Result	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements und die möglichen Fehlercodes sind in Abschnitt 4.1.2 von [eCard-1] und in Abschnitt 3.2.2 von [eCard-2] beschrieben.
	dss:OptionalOutputs	Sofern nicht ein Fehler aufgetreten ist, ist genau ein UpdatedSignature-Element vorhanden, das ein dss:SignatureObject-

Name	`VerifyResponse`	
		Element in der `dss:TimeStamp/RFC3161TimeStampToken`-Ausprägung enthält, in dem sich der um die Sperrinformationen ergänzte Zeitstempel befindet.

4.4.3 Berechnung eines Hashwertes

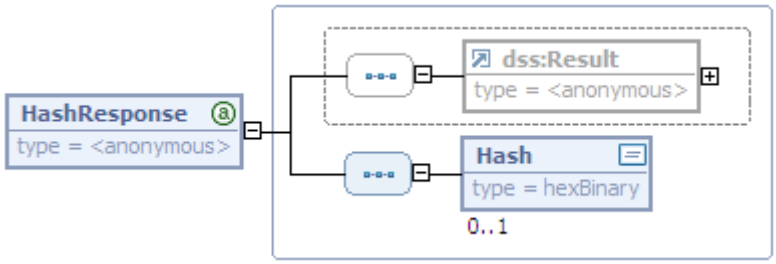
Zur Berechnung eines Hashwertes sind in S.3 die Schnittstellensignaturen `HashRequest` / `HashResponse` als ASN.1-Strukturen vorgesehen. Dies entspricht der Funktion `Hash` / `HashResponse` aus [eCard-4] in Verbindung mit dem Generic Cryptography-Protokoll aus [eCard-7].

4.4.3.1 Hash

Name	`Hash`	
Beschreibung	Bei einem `Hash`-Aufruf im Kontext der Schnittstelle S.3 wird für die übergebenen Daten ein Hashwert berechnet.	
Aufruf-parameter	 Aufruf der Funktion `Hash`.	
	Name	**Beschreibung**
	`ConnectionHandle`	Das `ConnectionHandle`-Element (vgl. [eCard-4], Abschnitt 3.1.3) gibt bei Bedarf an, auf welchem Hardwaremodul oder entfernten eCard-API-Framework die Berechnung des Hashwertes erfolgen soll. Sofern die Berechnung des Hashwertes durch das lokale Software-Modul erfolgen soll, soll das `ConnectionHandle`-Element leer sein.
	`DIDName`	Gibt bei Bedarf an, welche Differential-Identity (DID) für die Erzeugung des Hashwertes verwendet wird. Sofern ein leeres `ConnectionHandle` übergeben wird, soll der `DIDName` den Wert „`SWCrypto`“ besitzen und wird ignoriert.
	`DIDScope`	Löst ggf. Mehrdeutigkeiten zwischen lokalen und globalen DIDs mit gleichem Namen auf.
	`Message`	Enthält die Nachricht (bzw. einen Teil derselben, siehe

Name	Hash	
		[eCard-7]), aus der ein Hashwert berechnet werden soll.

4.4.3.2 HashResponse

Name	HashResponse	
Beschreibung	Als Antwort auf einen `Hash`-Aufruf wird vom Krypto-Modul ein entsprechendes `HashResponse`-Element gemäß Abschnitt 3.5.4 von [eCard-4] zurückgeliefert.	
Rückgabe	 Rückgabe der Funktion `Hash`.	
	Name	**Beschreibung**
	`dss:Result`	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements und die möglichen Fehlercodes sind in Abschnitt 4.1.2 von [eCard-1] und in Abschnitt 3.5.4 von [eCard-4] beschrieben.
	`Hash`	Enthält den Hashwert, sofern ein solcher berechnet werden konnte.

4.5 TR-ESOR-S.4 (Geschäftsanwendung – ArchiSafe-Modul)

Dieser Abschnitt beschreibt, wie die in [TR-ESOR-S] skizzierte Schnittstelle S.4 auf Basis der auch dem eCard-API-Framework (BSI TR 03112) zu Grunde liegenden Basistypen aus [OASIS-DSS] umgesetzt werden kann.

Die in [TR-ESOR-S] definierte Schnittstelle S.4 umfasst die folgenden Funktionen:

- Beweiswerterhaltende Archivierung elektronischer Daten (`ArchiveSubmissionRequest / ArchiveSubmissionResponse`)
- Ändern eines Archivdatenobjektes (`ArchiveUpdateRequest / ArchiveUpdateResponse`)
- Abfrage beweiswerterhaltend archivierter Daten (`ArchiveRetrievalRequest / ArchiveRetrievalResponse`)
- Rückgabe technischer Beweisdaten (`ArchiveEvidenceRequest / ArchiveEvidenceResponse`)
- Löschen von Archivdatenobjekten (`ArchiveDeletionRequest / ArchiveDeletionResponse`)
- Abfrage diskreter Datenobjekte (`ArchiveDataRequest / ArchiveDataResponse`)

4.5.1 Beweiswerterhaltende Archivierung elektronischer Daten

Für die beweiswerterhaltende Archivierung elektronischer Daten sind in [TR-ESOR-S] die Schnittstellensignaturen `ArchiveSubmissionRequest` und `ArchiveSubmissionResponse` als ASN.1-Struktur definiert. Dies entspricht den gleichnamigen und in Abschnitt 3.1 näher beschriebenen XML-Strukturen, die analog zu [eCard-2] unter Verwendung der grundlegenden Schnittstellentypen (`dss:RequestBaseType` und `dss:ResponseBaseType`) aus [OASIS-DSS] spezifiziert wurden.

4.5.2 Ändern von Archivdatenobjekten

Für das Ändern von Archivdatenobjekten sind in [TR-ESOR-S] die Schnittstellensignaturen `ArchiveUpdateRequest` und `ArchiveUpdateResponse` als ASN.1-Struktur definiert. Dies entspricht den gleichnamigen und in Abschnitt 3.2 näher beschriebenen XML-Strukturen, die analog zu [eCard-2] unter Verwendung der grundlegenden Schnittstellentypen (`dss:RequestBaseType` und `dss:ResponseBaseType`) aus [OASIS-DSS] spezifiziert wurden.

4.5.3 Abfrage beweiswerterhaltend archivierter Daten

Für die Abfrage beweiswerterhaltend archivierter Daten sind in [TR-ESOR-S] die Schnittstellensignaturen `ArchiveRetrievalRequest` und `ArchiveRetrievalResponse` als ASN.1-Struktur definiert. Dies entspricht den gleichnamigen und in Abschnitt 3.3 näher beschriebenen XML-Strukturen, die analog zu [eCard-2] unter Verwendung der grundlegenden Schnittstellentypen (`dss:RequestBaseType` und `dss:ResponseBaseType`) aus [OASIS-DSS] spezifiziert wurden.

4.5.4 Rückgabe technischer Beweisdaten

Für die Rückgabe technischer Beweisdaten sind in [TR-ESOR-S] die Schnittstellensignaturen `ArchiveEvidenceRequest` und `ArchiveEvidenceResponse` als ASN.1-Struktur definiert. Dies entspricht den gleichnamigen und in Abschnitt 3.4 näher beschriebenen XML-Strukturen, die analog zu [eCard-2] unter Verwendung der grundlegenden Schnittstellentypen (`dss:RequestBaseType` und `dss:ResponseBaseType`) aus [OASIS-DSS] spezifiziert wurden.

4.5.5 Löschen von Archivdatenobjekten

Für das Löschen von Archivdatenobjekten sind in [TR-ESOR-S] die Schnittstellensignaturen `ArchiveDeletionRequest` und `ArchiveDeletionResponse` als ASN.1-Struktur definiert. Dies entspricht den gleichnamigen und in Abschnitt 3.5 näher beschriebenen XML-Strukturen, die analog zu [eCard-2] unter Verwendung der grundlegenden Schnittstellentypen (`dss:RequestBaseType` und `dss:ResponseBaseType`) aus [OASIS-DSS] spezifiziert wurden.

4.5.6 Abfrage diskreter Datenobjekte

Für die Abfrage diskreter Datenobjekte sind in [TR-ESOR-S] die Schnittstellensignaturen `ArchiveDataRequest` und `ArchiveDataResponse` als ASN.1-Struktur definiert. Dies entspricht den gleichnamigen und in Abschnitt 3.6 näher beschriebenen XML-Strukturen, die analog zu [eCard-2] unter Verwendung der grundlegenden Schnittstellentypen (`dss:RequestBaseType` und `dss:ResponseBaseType`) aus [OASIS-DSS] spezifiziert wurden.

4.6 TR-ESOR-S.5 (ArchiSafe-Modul – ECM-Langzeitspeichersystem)

Dieser Abschnitt beschreibt, wie die in [TR-ESOR-S.5] skizzierte Schnittstelle auf Basis der auch dem eCard-API-Framework (BSI TR 03112) zu Grunde liegenden Basistypen aus [OASIS-DSS] umgesetzt werden kann.

Die in [TR-ESOR-S.5] definierte Schnittstelle umfasst die folgenden Funktionen:

- Abfrage beweiswerterhaltend archivierter Daten (`ArchiveRetrievalRequest / -Response`)
- Löschen von Archivdatenobjekten (`ArchiveDeletionRequest / -Response`)
- Abfrage diskreter Datenobjekte (`ArchiveDataRequest / -Response`)

4.6.1 Abfrage beweiswerterhaltend archivierter Daten

Für die Abfrage beweiswerterhaltend archivierter Daten sind in [TR-ESOR-S] die Schnittstellensignaturen `ArchiveRetrievalRequest` und `ArchiveRetrievalResponse` als ASN.1-Struktur definiert. Dies entspricht den gleichnamigen und in Abschnitt 3.3 näher beschriebenen XML-Strukturen, die analog zu [eCard-2] unter Verwendung der grundlegenden Schnittstellentypen (`dss:RequestBaseType` und `dss:ResponseBaseType`) aus [OASIS-DSS] spezifiziert wurden.

4.6.2 Löschen von Archivdatenobjekten

Für das Löschen von Archivdatenobjekten sind in [TR-ESOR-S] die Schnittstellensignaturen `ArchiveDeletionRequest` und `ArchiveDeletionResponse` als ASN.1-Struktur definiert. Dies entspricht den gleichnamigen und in Abschnitt 3.5 näher beschriebenen XML-Strukturen, die analog zu [eCard-2] unter Verwendung der grundlegenden Schnittstellentypen (`dss:RequestBaseType` und `dss:ResponseBaseType`) aus [OASIS-DSS] spezifiziert wurden.

4.6.3 Abfrage diskreter Datenobjekte

Für die Abfrage diskreter Datenobjekte sind in [TR-ESOR-S] die Schnittstellensignaturen `ArchiveDataRequest` und `ArchiveDataResponse` als ASN.1-Struktur definiert. Dies entspricht den gleichnamigen und in Abschnitt 3.6 näher beschriebenen XML-Strukturen, die analog zu [eCard-2] unter Verwendung der grundlegenden Schnittstellentypen (`dss:RequestBaseType` und `dss:ResponseBaseType`) aus [OASIS-DSS] spezifiziert wurden.

4.7 TR-ESOR-S.6 (ArchiSafe-Modul – ArchiSig-Modul)

Dieser Abschnitt beschreibt, wie die in [TR-ESOR-S] skizzierte Schnittstelle S.6 auf Basis der auch dem eCard-API-Frameworks (BSI TR 03112) zu Grunde liegenden Basistypen aus [OASIS-DSS] umgesetzt werden kann.

Die in [TR-ESOR-S] definierte Schnittstelle umfasst die folgenden Funktionen:

- Beweiswerterhaltende Archivierung elektronischer Daten (`ArchiveSubmissionRequest / ArchiveSubmissionResponse`)

- Ändern eines Archivdatenobjektes (`ArchiveUpdateRequest / ArchiveUpdateResponse`)
- Rückgabe technischer Beweisdaten (`ArchiveEvidenceRequest / ArchiveEvidenceResponse`)

4.7.1 Beweiswerterhaltende Archivierung elektronischer Daten

Für die beweiswerterhaltende Archivierung elektronischer Daten sind in [TR-ESOR-S] die Schnittstellensignaturen `ArchiveSubmissionRequest` und `ArchiveSubmissionResponse` als ASN.1-Struktur definiert. Dies entspricht den gleichnamigen und in Abschnitt 3.1 näher beschriebenen XML-Strukturen, die analog zu [eCard-2] unter Verwendung der grundlegenden Schnittstellentypen (`dss:RequestBaseType` und `dss:ResponseBaseType`) aus [OASIS-DSS] spezifiziert wurden.

4.7.2 Ändern von Archivdatenobjekten

Für das Ändern von Archivdatenobjekten sind in [TR-ESOR-S] die Schnittstellensignaturen `ArchiveUpdateRequest` und `ArchiveUpdateResponse` als ASN.1-Struktur definiert. Dies entspricht den gleichnamigen und in Abschnitt 3.2 näher beschriebenen XML-Strukturen, die analog zu [eCard-2] unter Verwendung der grundlegenden Schnittstellentypen (`dss:RequestBaseType` und `dss:ResponseBaseType`) aus [OASIS-DSS] spezifiziert wurden.

4.7.3 Rückgabe technischer Beweisdaten

Für die Rückgabe technischer Beweisdaten sind in [TR-ESOR-S] die Schnittstellensignaturen `ArchiveEvidenceRequest` und `ArchiveEvidenceResponse` als ASN.1-Struktur definiert. Dies entspricht den gleichnamigen und in Abschnitt 3.4 näher beschriebenen XML-Strukturen, die analog zu [eCard-2] unter Verwendung der grundlegenden Schnittstellentypen (`dss:RequestBaseType` und `dss:ResponseBaseType`) aus [OASIS-DSS] spezifiziert wurden.

5. Spezifikation einer Webservice-basierten Schnittstelle

Die Spezifikation der Webservice-basierten Schnittstelle besteht aus zwei Bestandteilen: Zunächst werden die Aufruf- und Rückgabeparameter als XML-Schema [XSD] spezifiziert (vgl. Abschnitt 5.1). Darauf aufbauend wird in einem zweiten Schritt eine Webservice-Spezifikation gemäß [WSDL] entwickelt.

Abschnitt 5.2 enthält die Webservice-Spezifikation der Schnittstelle S.4 (vgl. Abschnitt 4.5). Die internen Schnittstellen der TR-ESOR-Middleware können bei Bedarf leicht daraus abgeleitet werden, indem nur die benötigte Teilmenge der Funktionen genutzt wird.

Für den Nachweis der Konformitätsstufe 2 müssen die für das oder die Module relevanten Webservice-basierten Schnittstellen gemäß Abschnitt 5.2 unterstützt werden. Darüber hinaus können weitere Schnittstellen, wie z.B. eine sprachgebundene Java- oder C-Schnittstelle gemäß [eCard-1] unter Verwendung der in Abschnitt 5.1 spezifizierten XML-Strukturen unterstützt werden.

5.1 Spezifikation der Aufruf- und Rückgabeparameter als XML-Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.bsi.bund.de/tr-esor/api/1.1"
      xmlns="http://www.w3.org/2001/XMLSchema"
      xmlns:tr="http://www.bsi.bund.de/tr-esor/api/1.1"
      xmlns:xaip="http://bsi.bund.de/tr-esor/xaip/1.1"
      xmlns:ers="http://www.setcce.org/schemas/ers"
      xmlns:ec="http://www.bsi.bund.de/ecard/api/1.1"
      xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema"
      elementFormDefault="qualified" attributeFormDefault="unqualified">




      <!-- ======================================= -->
      <!-- Version 1.1 vom 21.01.2011              -->
      <!-- ======================================= -->


      <import namespace="http://bsi.bund.de/tr-esor/xaip/1.1"
            schemaLocation="tr-esor-xaip-v1.1.xsd" />

      <import namespace="http://bsi.bund.de/tr-esor/xsip/1.1"
            schemaLocation="tr-esor-xsip-experiment.xsd" />

      <import namespace="urn:oasis:names:tc:dss:1.0:core:schema"
            schemaLocation="http://docs.oasis-open.org/dss/v1.0/oasis-dss-
core-schema-v1.0-os.xsd" />

      <import namespace="http://www.setcce.org/schemas/ers"
            schemaLocation="../ecard/api/1.1/xmlers-schema-draft-v0.3.xsd"
/>
```

```
	<import namespace="http://www.bsi.bund.de/ecard/api/1.1"
		schemaLocation="../ecard/api/1.1/eCard.xsd" />

	<!-- ================================== -->
	<!--      Übergreifende Definitionen    -->
	<!-- ================================== -->



	<complexType name="RequestType">
		<complexContent>
			<restriction base="dss:RequestBaseType">
				<sequence>
					<element ref="dss:OptionalInputs"
maxOccurs="1"
						minOccurs="0" />
				</sequence>
			</restriction>
		</complexContent>
	</complexType>

	<complexType name="ResponseType">
		<complexContent>
			<restriction base="dss:ResponseBaseType">
				<sequence>
					<element ref="dss:Result" />
					<element ref="dss:OptionalOutputs"
maxOccurs="1"
						minOccurs="0" />
				</sequence>
			</restriction>
		</complexContent>
	</complexType>

	<!-- ================================ -->
	<!--    ArchiveSubmissionRequest      -->
	<!-- ================================ -->

	<complexType name="ArchiveDataType">
		<complexContent>
			<extension base="anyType">
				<attribute name="Type" type="anyURI" />
			</extension>
		</complexContent>
	</complexType>


	<element name="ArchiveSubmissionRequest">
		<complexType>
```

```
                <complexContent>
                      <extension base="tr:RequestType">
                            <choice>
                                  <element ref="xaip:XAIP"></element>
                                  <element name="ArchiveData"
type="tr:ArchiveDataType"></element>
                            </choice>
                      </extension>
                </complexContent>
          </complexType>
    </element>


    <element name="ArchiveSubmissionResponse">
          <complexType>
                <complexContent>
                      <extension base="tr:ResponseType">
                            <sequence>
                                  <element name="AOID" type="string"
maxOccurs="1"
                                        minOccurs="0">
                                  </element>
                            </sequence>
                      </extension>
                </complexContent>
          </complexType>
    </element>


    <!-- ========================== -->
    <!--    ArchiveUpdateRequest    -->
    <!-- ========================== -->

    <element name="ArchiveUpdateRequest">
          <complexType>
                <complexContent>
                      <extension base="tr:RequestType">
                            <choice>
                                  <element ref="xaip:XAIP"></element>
                                  <element name="ArchiveData"
type="tr:ArchiveDataType"></element>
                            </choice>
                      </extension>
                </complexContent>
          </complexType>
    </element>


    <element name="ArchiveUpdateResponse">
```

```
            <complexType>
                  <complexContent>
                        <extension base="tr:ResponseType">
                              <sequence>
                                    <element name="VersionID" type="string"
maxOccurs="1" minOccurs="0"></element>
                              </sequence>
                        </extension>
                  </complexContent>
            </complexType>
      </element>



      <!-- ================================ -->
      <!--    ArchiveRetrievalRequest       -->
      <!-- ================================ -->


      <element name="ArchiveRetrievalRequest">
            <complexType>
                  <complexContent>
                        <extension base="tr:RequestType">
                              <sequence maxOccurs="unbounded" minOccurs="1">
                                    <element name="AOID" type="string"
                                          maxOccurs="1" minOccurs="1">
                                    </element>
                                    <element name="VersionID" type="string"
maxOccurs="unbounded" minOccurs="0"></element>
                              </sequence>
                        </extension>
                  </complexContent>
            </complexType>
      </element>


      <element name="ArchiveRetrievalResponse">
            <complexType>
                  <complexContent>
                        <extension base="tr:ResponseType">
                              <sequence>
                                    <element ref="tr:ResponseDataSequence"
                                          maxOccurs="1" minOccurs="0" />
                              </sequence>
                        </extension>
                  </complexContent>
            </complexType>
      </element>
```

```
      <element name="ResponseDataSequence">
            <complexType>
                  <sequence>
                        <element name="ResponseData" maxOccurs="unbounded"
                              minOccurs="1">
                              <complexType>
                                    <sequence>
                                          <element ref="dss:Result" />
                                          <choice>
                                                <element name="AOID"
type="string">
                                                </element>
                                                <element ref="xaip:XAIP" />
                                          </choice>
                                    </sequence>
                              </complexType>
                        </element>
                  </sequence>
            </complexType>
      </element>



      <!-- ================================ -->
      <!--    ArchiveEvidenceRequest        -->
      <!-- ================================ -->


      <element name="ArchiveEvidenceRequest">
            <complexType>
                  <complexContent>
                        <extension base="tr:RequestType">
                              <sequence maxOccurs="unbounded" minOccurs="1">
                                    <element name="AOID"
type="string"></element>
                                    <element name="VersionID" type="string"
maxOccurs="unbounded" minOccurs="0"></element>
                              </sequence>
                        </extension>
                  </complexContent>
            </complexType>
      </element>

      <element name="ERSFormat" type="anyURI" />

      <element name="ArchiveEvidenceResponse">
            <complexType>
```

```
                <complexContent>
                      <extension base="tr:ResponseType">
                            <sequence>
                                  <element ref="tr:EvidenceResultSequence"
                                        maxOccurs="1" minOccurs="0" />
                            </sequence>
                      </extension>
                </complexContent>
          </complexType>
    </element>

    <element name="EvidenceResultSequence">
          <complexType>
                <sequence maxOccurs="unbounded" minOccurs="1">
                      <element name="EvidenceResult"
                            type="tr:EvidenceResultType" />

                </sequence>
          </complexType>
    </element>

    <complexType name="EvidenceRecordType" >
          <complexContent>
                <extension base="ec:EvidenceRecordType">
                      <attribute name="credentialID" type="string" />
                      <attribute name="relatedObjects" type="string"/>

                </extension>
          </complexContent>
    </complexType>


    <complexType name="EvidenceResultType">
          <sequence>
                <element ref="dss:Result" />
                <element name="AOID" type="string"></element>
                <element name="VersionID" type="string"></element>
                <element name="evidenceRecord" maxOccurs="unbounded"
minOccurs="0"
                      type="tr:EvidenceRecordType">
                </element>
          </sequence>
    </complexType>

    <!-- ================================ -->
    <!--    ArchiveDeletionRequest      -->
    <!-- ================================ -->

    <element name="ArchiveDeletionRequest">
```

```
            <complexType>
                  <complexContent>
                        <extension base="tr:RequestType">
                              <sequence>
                                    <element name="AOID" type="string"
                                          maxOccurs="unbounded"
minOccurs="1">
                                    </element>
                              </sequence>
                        </extension>
                  </complexContent>
            </complexType>
      </element>

      <element name="ReasonOfDeletion">
            <complexType>
                  <sequence>
                        <element name="RequestorName"
                              type="xaip:generalNameType" />
                        <element name="RequestInfo" type="string" />
                  </sequence>
            </complexType>
      </element>

      <element name="ArchiveDeletionResponse">
            <complexType>
                  <complexContent>
                        <extension base="tr:ResponseType">
                              <sequence>
                                    <element ref="tr:DeletionResultSequence"
                                          maxOccurs="1" minOccurs="0" />
                              </sequence>
                        </extension>
                  </complexContent>
            </complexType>
      </element>


      <element name="DeletionResultSequence">
            <complexType>
                  <sequence maxOccurs="unbounded" minOccurs="1">
                        <element name="DeletionResult">
                              <complexType>
                                    <sequence>
                                          <element ref="dss:Result" />
                                          <element name="AOID"
type="string"></element>
                                    </sequence>
                              </complexType>
```

```
                        </element>
                  </sequence>
            </complexType>
      </element>


      <!-- ========================== -->
      <!--    ArchiveDataRequest      -->
      <!-- ========================== -->

      <element name="ArchiveDataRequest">
            <complexType>
                  <complexContent>
                        <extension base="tr:RequestType">
                              <sequence>
                                    <element ref="tr:RequestDataSequence" />
                              </sequence>
                        </extension>
                  </complexContent>
            </complexType>
      </element>


      <element name="RequestDataSequence">
            <complexType>
                  <sequence>
                        <element name="RequestData" maxOccurs="unbounded"
                              minOccurs="1">
                              <complexType>
                                    <sequence>
                                          <element name="AOID"
type="string"></element>
                                          <element
ref="tr:DataLocationSequence" />
                                    </sequence>
                              </complexType>
                        </element>
                  </sequence>
            </complexType>
      </element>

      <element name="DataLocationSequence">
            <complexType>
                  <sequence>
                        <element ref="tr:DataLocation"
maxOccurs="unbounded"
                              minOccurs="1" />
                  </sequence>
            </complexType>
```

```
      </element>

      <element name="DataLocation">
            <complexType>
                  <complexContent>
                        <extension base="anyType">
                              <attribute name="Type" type="anyURI" />
                        </extension>
                  </complexContent>
            </complexType>
      </element>

      <element name="ArchiveDataResponse">
            <complexType>
                  <complexContent>
                        <extension base="tr:ResponseType">
                              <sequence>
                                    <element
ref="tr:ResponseDataElementSequence"
                                          maxOccurs="1" minOccurs="0" />
                              </sequence>
                        </extension>
                  </complexContent>
            </complexType>
      </element>


      <element name="ResponseDataElementSequence">
            <complexType>
                  <sequence>
                        <element name="ResponseDataElement"
                              maxOccurs="unbounded" minOccurs="1">
                              <complexType>
                                    <sequence>
                                          <element ref="dss:Result"/>
                                          <element name="AOID"
type="string"></element>
                                          <element name="XAIPDataSequence"
                                                maxOccurs="1" minOccurs="0">
                                                <complexType>
                                                      <sequence>
                                                            <element
name="XAIPData"

      maxOccurs="unbounded" minOccurs="1">

      <complexType>

      <sequence>
```

```
      <element
            ref="dss:Result" maxOccurs="1" minOccurs="0" />

      <element
            ref="tr:DataLocation" />

      <element
            name="Value" type="anyType" maxOccurs="1" minOccurs="0" />

      </sequence>
      </complexType>
                                                            </element>
                                                      </sequence>
                                                </complexType>
                                          </element>
                                    </sequence>
                              </complexType>
                        </element>
                  </sequence>
            </complexType>
      </element>

</schema>
```

5.2 WSDL-Spezifikation der Schnittstelle TR-ESOR-S.4

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://www.bsi.bund.de/tr-esor/api/1.1"
      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
      xmlns:tr="http://www.bsi.bund.de/tr-esor/api/1.1"
      >

      <!--================================================================-->
      <!-- Version 1.1 of 15.08.2010
                  -->
      <!--================================================================-->
```

```
      <!-- ================== -->
      <!-- Definition of types -->
      <!-- (only include XSDs) -->
      <!-- ================== -->

      <wsdl:types>
            <xsd:schema targetNamespace="http://www.bsi.bund.de/tr-
esor/api/1.1"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xaip="http://bsi.bund.de/tr-esor/xaip/1.1"
                  xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema"
                  elementFormDefault="qualified">
                  <xsd:include schemaLocation="tr-esor-interfaces-v1.1.xsd"
/>
            </xsd:schema>
      </wsdl:types>



      <!-- ===================== -->
      <!-- Definition of messages -->
      <!-- ===================== -->


    <!-- ArchiveSubmissionRequest -->

      <wsdl:message name="ArchiveSubmissionRequest">
            <wsdl:part name="parameters"
element="tr:ArchiveSubmissionRequest" />
      </wsdl:message>
      <wsdl:message name="ArchiveSubmissionResponse">
            <wsdl:part name="parameters"
element="tr:ArchiveSubmissionResponse"/>
      </wsdl:message>

    <!-- ArchiveUpdateRequest -->

      <wsdl:message name="ArchiveUpdateRequest">
            <wsdl:part name="parameters" element="tr:ArchiveUpdateRequest"
/>
      </wsdl:message>
      <wsdl:message name="ArchiveUpdateResponse">
            <wsdl:part name="parameters"
element="tr:ArchiveUpdateResponse"/>
      </wsdl:message>


    <!-- ArchiveRetrievalRequest -->
```

```
      <wsdl:message name="ArchiveRetrievalRequest">
            <wsdl:part name="parameters"
element="tr:ArchiveRetrievalRequest" />
      </wsdl:message>
      <wsdl:message name="ArchiveRetrievalResponse">
            <wsdl:part name="parameters"
element="tr:ArchiveRetrievalResponse" />
      </wsdl:message>


    <!-- ArchiveEvidenceRequest -->

      <wsdl:message name="ArchiveEvidenceRequest">
            <wsdl:part name="parameters"
element="tr:ArchiveEvidenceRequest" />
      </wsdl:message>
      <wsdl:message name="ArchiveEvidenceResponse">
            <wsdl:part name="parameters"
element="tr:ArchiveEvidenceResponse" />
      </wsdl:message>


    <!-- ArchiveDeletionRequest -->

      <wsdl:message name="ArchiveDeletionRequest">
            <wsdl:part name="parameters"
element="tr:ArchiveDeletionRequest" />
      </wsdl:message>
      <wsdl:message name="ArchiveDeletionResponse">
            <wsdl:part name="parameters"
element="tr:ArchiveDeletionResponse" />
      </wsdl:message>


    <!-- ArchiveDataRequest -->

      <wsdl:message name="ArchiveDataRequest">
            <wsdl:part name="parameters" element="tr:ArchiveDataRequest" />
      </wsdl:message>
      <wsdl:message name="ArchiveDataResponse">
            <wsdl:part name="parameters" element="tr:ArchiveDataResponse"
/>
      </wsdl:message>

      <!-- ====================== -->
      <!-- Definition of portType -->
      <!-- ====================== -->


      <wsdl:portType name="S4">
```

```
            <wsdl:operation name="ArchiveSubmission">
                  <wsdl:input message="tr:ArchiveSubmissionRequest" />
                  <wsdl:output message="tr:ArchiveSubmissionResponse" />
            </wsdl:operation>
            <wsdl:operation name="ArchiveUpdate">
                  <wsdl:input message="tr:ArchiveUpdateRequest" />
                  <wsdl:output message="tr:ArchiveUpdateResponse" />
            </wsdl:operation>
            <wsdl:operation name="ArchiveRetrieval">
                  <wsdl:input message="tr:ArchiveRetrievalRequest" />
                  <wsdl:output message="tr:ArchiveRetrievalResponse" />
            </wsdl:operation>
            <wsdl:operation name="ArchiveEvidence">
                  <wsdl:input message="tr:ArchiveEvidenceRequest" />
                  <wsdl:output message="tr:ArchiveEvidenceResponse" />
            </wsdl:operation>
            <wsdl:operation name="ArchiveDeletion">
                  <wsdl:input message="tr:ArchiveDeletionRequest" />
                  <wsdl:output message="tr:ArchiveDeletionResponse" />
            </wsdl:operation>
            <wsdl:operation name="ArchiveData">
                  <wsdl:input message="tr:ArchiveDataRequest" />
                  <wsdl:output message="tr:ArchiveDataResponse" />
            </wsdl:operation>
      </wsdl:portType>

      <!-- ==================== -->
      <!-- Definition of Binding -->
      <!-- ==================== -->

      <wsdl:binding name="S4" type="tr:S4">
            <soap:binding style="document"
                  transport="http://schemas.xmlsoap.org/soap/http" />
            <wsdl:operation name="ArchiveSubmission">
                  <soap:operation
                        soapAction="http://www.bsi.bund.de/tr-
esor/ArchiveSubmission" />
                  <wsdl:input>
                        <soap:body use="literal" />
                  </wsdl:input>
                  <wsdl:output>
                        <soap:body use="literal" />
                  </wsdl:output>
            </wsdl:operation>
            <wsdl:operation name="ArchiveUpdate">
                  <soap:operation
                        soapAction="http://www.bsi.bund.de/tr-
esor/ArchiveUpdate" />
                  <wsdl:input>
```

```
                        <soap:body use="literal" />
                    </wsdl:input>
                    <wsdl:output>
                        <soap:body use="literal" />
                    </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="ArchiveRetrieval">
                    <soap:operation
                        soapAction="http://www.bsi.bund.de/tr-
esor/ArchiveRetrieval" />
                    <wsdl:input>
                        <soap:body use="literal" />
                    </wsdl:input>
                    <wsdl:output>
                        <soap:body use="literal" />
                    </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="ArchiveEvidence">
                    <soap:operation
                        soapAction="http://www.bsi.bund.de/tr-
esor/ArchiveEvidence" />
                    <wsdl:input>
                        <soap:body use="literal" />
                    </wsdl:input>
                    <wsdl:output>
                        <soap:body use="literal" />
                    </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="ArchiveDeletion">
                    <soap:operation
                        soapAction="http://www.bsi.bund.de/tr-
esor/ArchiveDeletion" />
                    <wsdl:input>
                        <soap:body use="literal" />
                    </wsdl:input>
                    <wsdl:output>
                        <soap:body use="literal" />
                    </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="ArchiveData">
                    <soap:operation
                        soapAction="http://www.bsi.bund.de/tr-
esor/ArchiveData" />
                    <wsdl:input>
                        <soap:body use="literal" />
                    </wsdl:input>
                    <wsdl:output>
                        <soap:body use="literal" />
                    </wsdl:output>
```

```
            </wsdl:operation>
      </wsdl:binding>

      <!-- Definition of Support-Service -->

      <wsdl:service name="S4">
            <wsdl:port name="S4" binding="tr:S4">
                  <soap:address location="http://127.0.0.1:18080" />
            </wsdl:port>
      </wsdl:service>
</wsdl:definitions>
```