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	1.2.1 (auf Basis der eIDAS-Verordnung)
Datum	15.03.2018

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

Inhaltsverzeichnis

Abbildungsverzeichnis

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 der 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 dient, die TR-ESOR-Middleware in die bestehende IT- und Infrastrukturlandschaft einzubetten;
- dem „ArchiSafe-Modul“ (**[TR-ESOR-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“ (**[TR-ESOR-M.2]**) nebst den zugehörigen Schnittstellen S.1 und S.3, das alle erforderlichen Funktionen zur Berechnung von Hashwerten, Prüfung elektronischer Signaturen bzw. Siegel bzw. Zeitstempel, zur Nachprüfung elektronischer Zertifikate und zum Einholen qualifizierter Zeitstempel sowie (optional) elektronischer Signaturen bzw. Siegel 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, das 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 Abbildung 1 dargestellte IT-Referenzarchitektur orientiert sich an der ArchiSafe[1] Referenzarchitektur 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

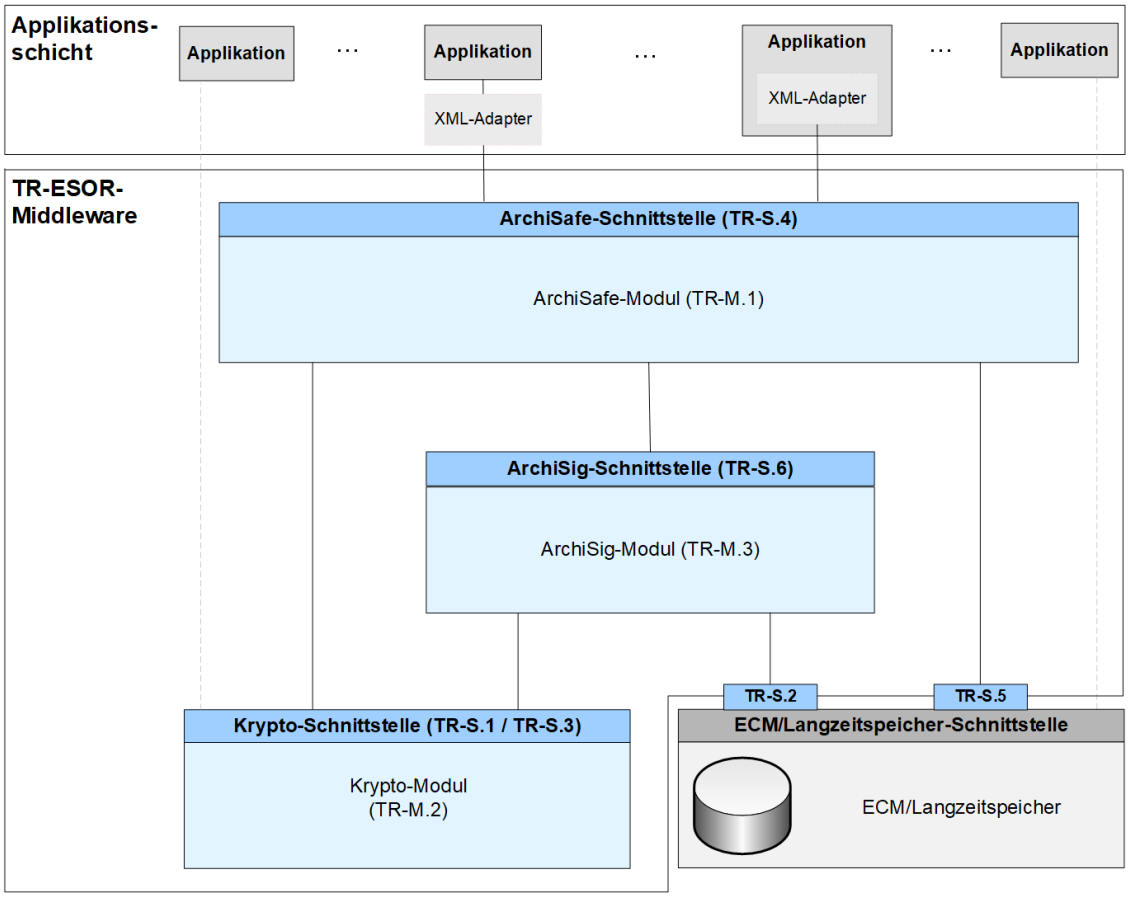


Abbildung 1: Schematische Darstellung der IT-Referenzarchitektur

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 dreistufiges 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 die folgenden im vorliegenden Dokument näher aufgeführten Funktionen mit den hier beschriebenen Parameterkonstellationen unterstützt werden:

- ArchiveSubmissionRequest und ArchiveSubmissionResponse (siehe Abschnitt 3.1)
- ArchiveRetrievalRequest und ArchiveRetrievalResponse (siehe Abschnitt 3.3)
- ArchiveEvidenceRequest und ArchiveEvidenceResponse (siehe Abschnitt 3.4)
- ArchiveDeletionRequest und ArchiveDeletionResponse (siehe Abschnitt 3.5)

Darüber hinaus sollen die folgenden im vorliegenden Dokument näher aufgeführten Funktionen mit den hier beschriebenen Parameterkonstellationen unterstützt werden:

- ArchiveUpdateRequest und ArchiveUpdateResponse (siehe Abschnitt 3.2)
- ArchiveDataRequest und ArchiveDataResponse (siehe Abschnitt 3.6)
- VerifyRequest und VerifyResponse (siehe Abschnitt 3.7)

(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 **[TR-ESOR-F]** als XML-Datenformat verwendet werden. Abweichungen im verwendeten XML-Datenformat sind zulässig, allerdings muss dann erläutert werden, wie eine gleichwertige Funktionalität realisiert wird. Insbesondere ist zu erläutern, wie eine Transformation in das XAIP Format aus Anhang F erfolgen kann.

(A2.0-4) 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 das Basis-ERS-Profil aus Anhang **[TR-ESOR-ERS]** als Evidence-Record-Format verwendet werden.

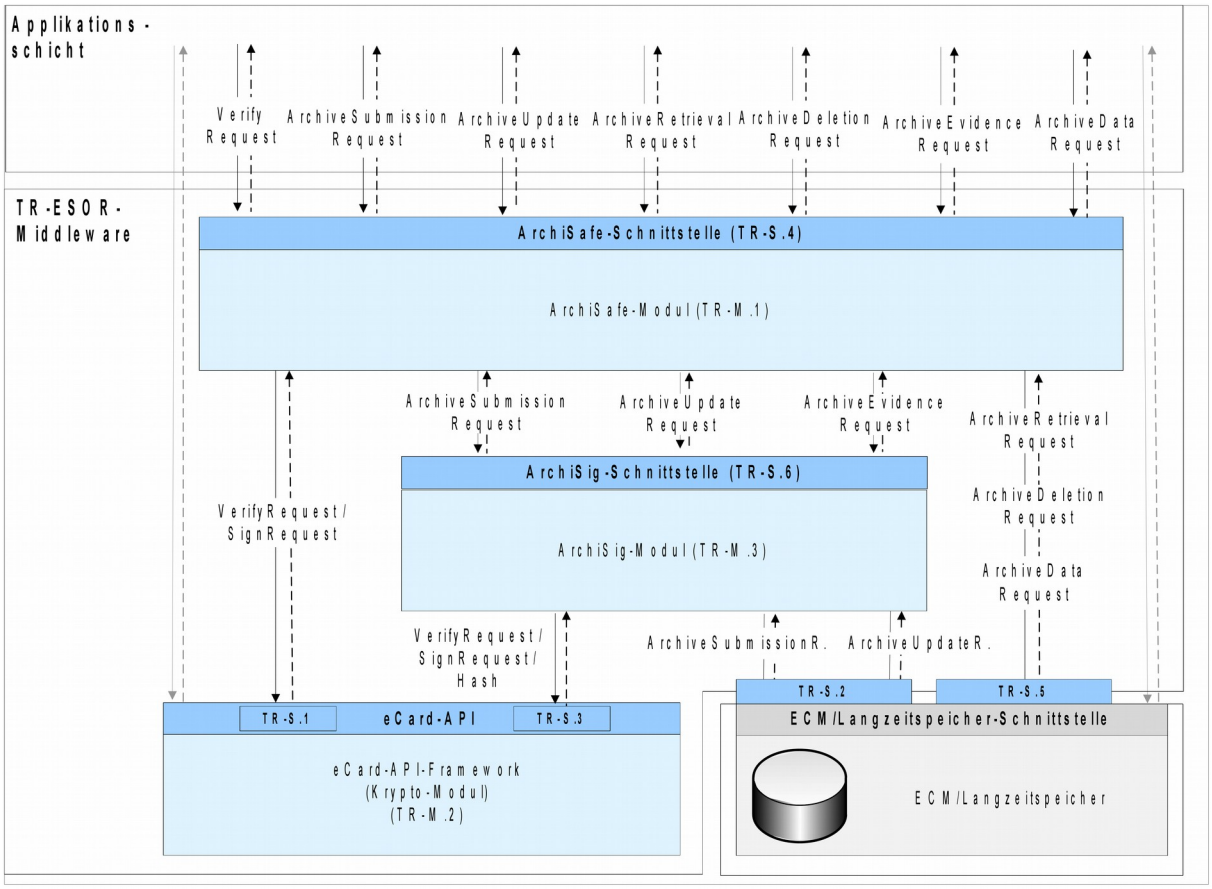


Abbildung 2: Umsetzung der IT-Referenzarchitektur auf Basis des eCard-API-Frameworks

Wie in Abbildung Abbildung 1angedeutet, 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-Modul 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/1.2, welches um entsprechende Bezeichner ergänzt wird. Dieser Namensraum ist in den visualisierten XML-Strukturen am Kürzel „`tr`“ erkennbar.

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 6.1) und den darauf aufbauenden Webservice-Schnittstellen (vgl. Abschnitt 6.2) darauf verzichtet werden, da die Version der Struktur implizit durch den oben genannten Namensraum 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, kann dies unter Verwendung der hierfür vorgesehenen Mechanismen aus **[OASIS-Async]** realisiert werden.
In den folgenden Abschnitten findet sich eine XML-basierte Spezifikation der verschiedenen in **[TR-ESOR-S]** eingeführten Funktionen zur Beweiswerterhaltung kryptographisch signierter Dokumente.

Hierbei werden die Funktionen der ArchiSafe-Schnittstelle (TR-S.4) in Abschnitt 3 spezifiziert. In Abschnitt 4 findet sich eine Beschreibung der internen Schnittstellen der TR-ESOR-Middleware, die auf die vorherige Spezifikation der Funktionen in Abschnitt 3 Bezug nimmt. In Abschnitt 5 sind die verwendeten Fehlercodes zusammengefasst und näher erläutert und in Abschnitt 6 finden sich schließlich die normativen XML-Schema- und WSDL-Spezifikationen für die in Abschnitt 3 spezifizierte ArchiSafe-Schnittstelle (TR-S.4).

HINWEIS: *Im folgenden Text umfasst der Begriff* ***„Digitale Signatur“*** *„fortgeschrittene elektronische Signaturen“ gemäß* ***[eIDAS-VO, Artikel 3 Nr. 11],*** *„qualifizierte elektronische Signaturen“ gemäß* ***[eIDAS-VO, Artikel 3 Nr. 12]****, „fortgeschrittenen elektronische Siegel“ gemäß* ***[eIDAS-VO, Artikel 3 Nr. 26]*** *und „qualifizierte elektronische Siegel“ gemäß* ***[eiDAS-VO, Artikel 3 Nr. 27]****. Insofern umfasst der Begriff „digital signierte Dokumente“ sowohl solche, die fortgeschrittene elektronische Signaturen oder Siegel bzw. qualifizierte elektronische Signaturen oder Siegel tragen.*

Mit dem Begriff der ***„kryptographisch signierten Dokumente“*** *sind in dieser TR neben den gemäß* ***[eIDAS-VO, Artikel 3 Nr. 12]*** *qualifiziert signierten, den gemäß* ***[eIDAS-VO, Artikel 3 Nr. 27]*** *qualifiziert gesiegelten oder den gemäß* ***[eIDAS-VO, Artikel 3 Nr. 34]*** *qualifiziert zeitgestempelten Dokumenten (im Sinne der eIDAS-Verordnung)) auch Dokumente mit einer fortgeschrittenen Signatur gemäß* ***[eIDAS-VO, Artikel 3 Nr. 11]*** *oder mit einem fortgeschrittenen Siegel gemäß* ***[eIDAS-VO, Artikel 3 Nr. 26]*** *oder mit einem elektronischen Zeitstempel gemäß* ***[eIDAS-VO, Artikel 3 Nr. 33]*** *erfasst, wie sie oft in der internen Kommunikation von Behörden entstehen. Nicht gemeint sind hier Dokumente mit einfachen Signaturen oder Siegeln basierend auf anderen (z. B. nicht-kryptographischen) Verfahren.*

3. Funktionen der ArchiSafe-Schnittstelle (TR-ESOR-S.4)

In diesem Abschnitt findet sich eine XML-basierte Spezifikation der in **[TR-ESOR-S]** eingeführten Funktionen der TR-ESOR-Middleware an der ArchiSafe-Schnittstelle **TR-ESOR-S.4 (TR-S.4)**:

- `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)
- `VerifyRequest` und `VerifyResponse` (siehe Abschnitt 3.7)

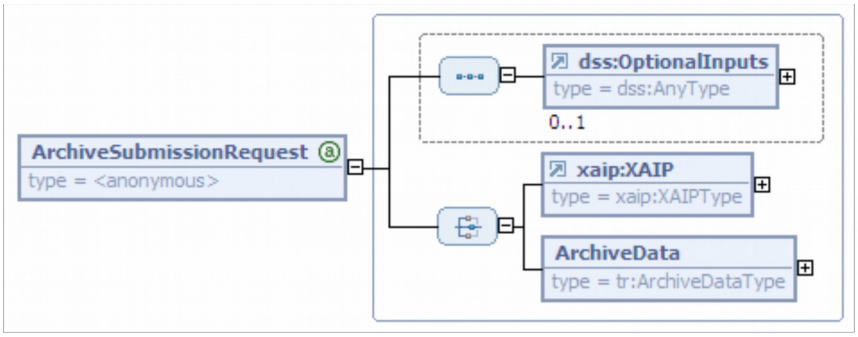
Die graphische Darstellung der Schnittstellen in diesem Kapitel wurde - analog zur Spezifikation des eCard-API-Frameworks (siehe z.B. **[eCard-2]**) - mit einem XML-Viewer erstellt und dient lediglich der Veranschaulichung der XML-Strukturen. Die normative Spezifikation der Schnittstellen ist durch das XML-Schema bzw. die darauf aufbauende WSDL-Spezifikation (siehe Abschnitt 6) gegeben.

3.1 ArchiveSubmissionRequest und ArchiveSubmissionResponse

Mit der Funktion `ArchiveSubmissionRequest` wird dem aufgerufenen Modul ein Archivdatenobjekt (`xaip:XAIP`) zur Ablage übergeben 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.

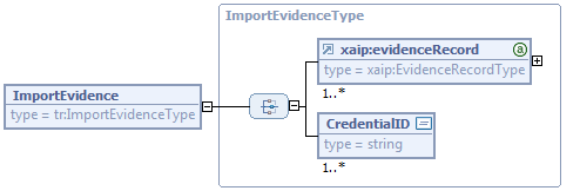
Wie in Abbildung Abbildung 2 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in den Schnittstellen TR-S.2 (vgl. Abschnitt 4.2) und TR-S.6 (vgl. Abschnitt 4.5) genutzt.

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>

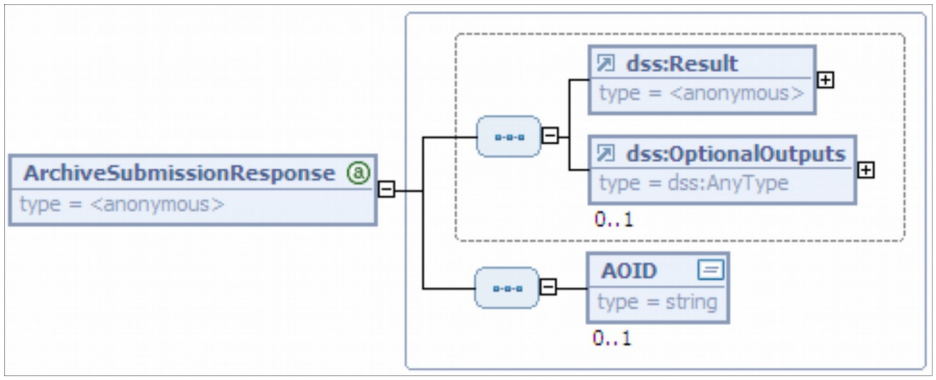
	`dss:OptionalInputs`	Ist für optionale Eingabeelemente vorgesehen. **(A3.1.1-1):** Gemäß der vorliegenden Spezifikation sollen folgende Elemente unterstützt werden: • `AOID`, • `ReturnVerificationReport`, • `ImportEvidence`. Dabei gilt: • `AOID` Durch die Übergabe eines `AOID`-Elementes kann die AOID von der aufrufenden Anwendung vergeben werden. Im Regelfall fehlt dieses Element und die AOID wird vom aufgerufenen Modul bereitgestellt.  • `ReturnVerificationReport` Durch die Übergabe eines `ReturnVerificationReport`-Elementes gemäß **[OASIS VR]** bzw. **[eCard-2]** kann ein ausführlicher Prüfbericht in Form eines `VerificationReport`-Elementes für die im `XAIP`-Element oder im unten genannten `ImportEvidence`-Element enthaltenen Signatur- bzw. Siegel- bzw. Zeitstempelobjekte oder Beweisdaten angefordert werden. Bei einem übergebenen `xaip:XAIP`-Element wird im `Details`-Element des `IndividualReport`-Elementes des zurückgelieferten Prüfberichts (vgl. Abschnitt 3.3 in **[OASIS VR]**) ein `XAIPReport`-Element gemäß **[TR-ESOR-VR]** zurückgeliefert. Sofern kein `xaip:XAIP` sondern ein `ArchiveData`-Element und im `ImportEvidence`-Element (siehe unten) ein Evidence Record übergeben wird, wird für jeden übergebenen Evidence Record ein `EvidenceRecordReport` gemäß **[TR-ESOR-VR]** zurückgeliefert.

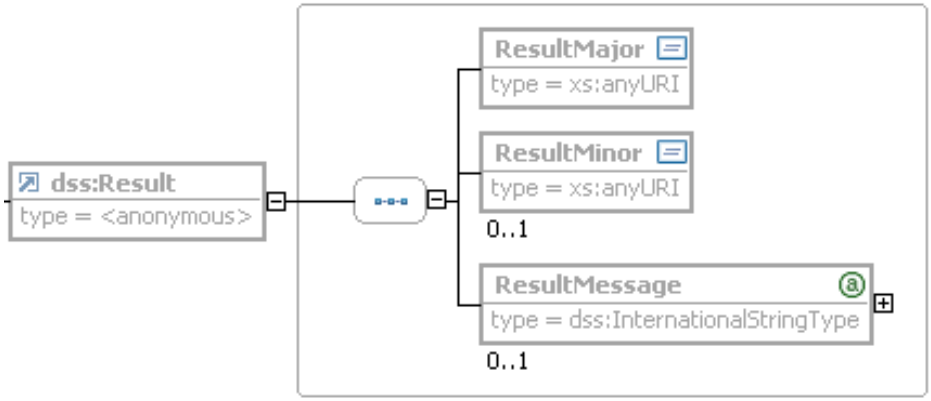
		• `ImportEvidence` Mit der Übergabe des nachfolgend dargestellten `ImportEvidence`-Elementes kann der Import von einem oder mehreren zu einer bestimmten XAIP-Version bzw. zu den übergebenen Binärdaten gehörenden Evidence Records gemäß **[RFC4998]** oder **[RFC6283]**[2] angestoßen werden. Die Struktur des `xaip:evidenceRecord`-Elementes ist in **[TR-ESOR-F]** erläutert. Um Evidence Records für mehrere Versionen eines XAIPs importieren, zu können, kann dieses Element mehrmals auftreten. Das `xaip:evidenceRecord`-Element muss hier die Attribute `AOID` und `VersionID` enthalten. Sofern die zu importierenden Evidence Records bereits im XAIP enthalten sind, wird statt des Evidence Records hier die entsprechende `CredentialID` übergeben.  **(A3.1.1-2):** Im Zuge des Imports von Evidence Records müssen diese von der TR-ESOR-Middleware vollständig geprüft werden. Dies umfasst die im entsprechenden ERS-Standard vorgesehenen Prüfungsschritte[3], wobei die jeweiligen Zertifikate der Zeitstempel vollständig bis hin zu einer vertrauenswürdigen Wurzel geprüft werden müssen.
	`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.

[2] **[RFC4998]** muss, **[RFC6283]** kann unterstützt werden.
[3] Siehe Abschnitt 3.3 in **[RFC4998]** und Abschnitt 2.3 in **[RFC6283]**.

		Durch das `Type`-Attribut http://www.bsi.bund.de/tr-esor/api/1.2/type/binaryData wird klargestellt, dass im `ArchiveData`-Element ein Kindelement `binaryData` übergeben wird, das Base 64 codierte Nutzdaten und ein `MimeType`-Attribut enthält, die beim entsprechenden XAIP in ein `dataObject`-Element eingebettet werden. Weitere Übergabetypen 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.
	`dss:OptionalOutputs`	Ist für optionale Ausgabeelemente vorgesehen. **(A3.1.2-1):** Gemäß der vorliegenden Spezifikation kann das folgende Element auftreten: • `VerificationReport` gemäß **[OASIS VR]** bzw. **[eCard-2] und [TR-ESOR-VR]**, der zurückgeliefert werden muss, sofern er explizit angefordert wurde oder bei der Prüfung der übergebenen Daten ein Fehler oder eine Warnung aufgetreten ist und deshalb als `ResultMajor` ein Fehlercode .../resultmajor#error oder .../resultmajor#warning zurückgeliefert wird.

<table>
<tr><td>Name</td><td colspan="2">ArchiveSubmissionResponse</td></tr>
<tr><td rowspan="4"></td><td>AOID</td><td>Muss, sofern die AOID vom aufgerufenen Modul erzeugt oder ergänzt wurde, vorhanden sein und für zukünftige Zugriffe auf das Archivdatenobjekt genutzt werden.</td></tr>
<tr><td colspan="2"> 

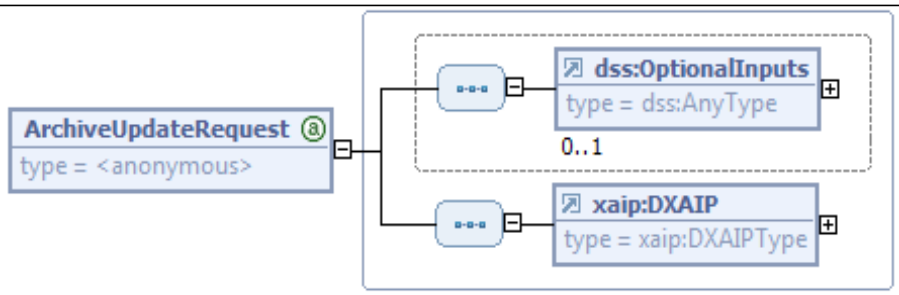
Statusinformationen und Fehler bei ArchiveSubmissionResponse (vgl. [eCard-1] Abschnitt 4.1 und 4.2).</td></tr>
<tr><td>Name</td><td>Fehlercode</td></tr>
<tr><td>ResultMajor</td><td>• /resultmajor#ok
• /resultmajor#error
• /resultmajor#warning</td></tr>
<tr><td>ResultMinor</td><td>• /resultminor/al/common#noPermission
• /resultminor/al/common#internalError
• /resultminor/al/common#parameterError
• /resultminor/arl/lowSpaceWarning
• /resultminor/arl/noSpaceError
• /resultminor/arl/existingAOID
• /resultminor/arl/notSupported
• /resultminor/arl/unknownArchiveDataType
• /resultminor/arl/XAIP_NOK
• /resultminor/arl/XAIP_NOK_EXPIRED
• /resultminor/arl/XAIP_NOK_SUBMTIME
• /resultminor/arl/XAIP_NOK_SIG
• /resultminor/arl/XAIP_NOK_ER</td></tr>
</table>

3.2 ArchiveUpdateRequest und ArchiveUpdateResponse

Mit der Funktion ArchiveUpdateRequest kann eine neue Version für ein bereits abgelegtes Archivdatenobjekt erzeugt werden. Hierbei werden die bereits abgelegten Daten nicht verändert, sondern es wird lediglich zusätzlich eine neue Version hinzugefügt.

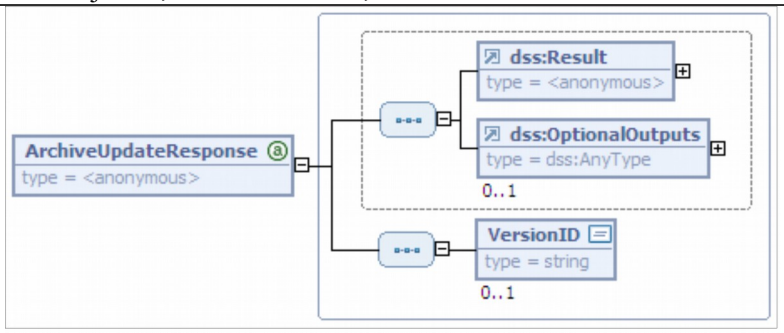
Wie in Abbildung Abbildung 2 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in TR-S.2 (vgl. Abschnitt 4.2) und TR-S.6 (vgl. Abschnitt 4.5) genutzt.

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></td><td colspan="2"> 

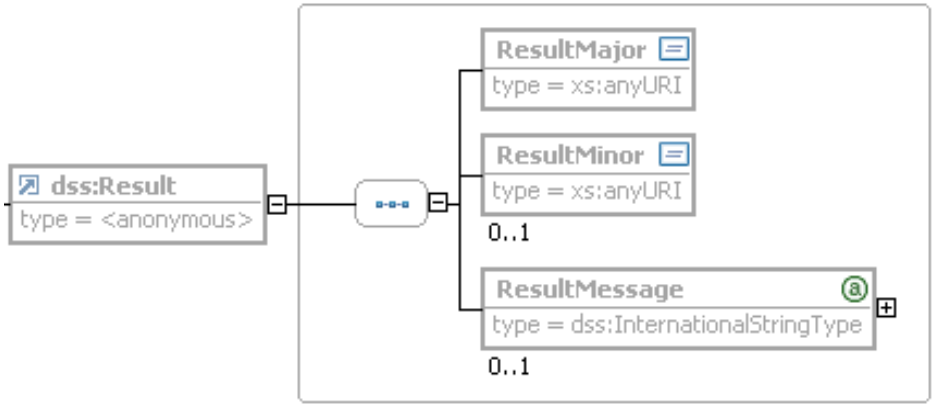
Aufruf der <code>ArchiveUpdateRequest</code>-Funktion</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
<tr><td></td><td><code>dss:OptionalInputs</code></td><td>Ist für optionale Eingabeelemente vorgesehen.
(A3.2.1-1): Gemäß der vorliegenden Spezifikation sollen hier die auf Seite 11 spezifizierten optionalen Eingabeelemente <code>AOID</code>, <code>ReturnVerificationReport</code> und <code>ImportEvidence</code> unterstützt werden.</td></tr>
<tr><td></td><td><code>xaip:DXAIP</code></td><td>Enthält ein ergänzendes XML-basiertes Archivdatenobjekt (Delta-XAIP) gemäß [TR-ESOR-F], das ein neues <code>versionManifest</code>, die Vorgängerversion, Verweise auf unverändert aus dieser übernommene Objekte und die zu ergänzenden Elemente enthält, die in einer neuen Version eines bereits abgelegten Archivdatenobjektes ergänzt werden sollen.</td></tr>
</table>

3.2.2 ArchiveUpdateResponse

<table>
<tr><td>Name</td><td colspan="2"><code>ArchiveUpdateResponse</code></td></tr>
<tr><td>Beschreibung</td><td colspan="2">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 colspan="2"> 

<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.
(A3.2.2-1): Gemäß der vorliegenden Spezifikation kann das folgende Element auftreten:
<code>VerificationReport</code> gemäß [OASIS VR] bzw. [eCard-2] und [TR-ESOR-VR], der zurückgeliefert werden muss, sofern er explizit angefordert wurde oder bei der Prüfung der übergebenen Daten ein Fehler oder eine Warnung aufgetreten ist und deshalb als <code>ResultMajor</code> ein Fehlercode .../resultmajor#error oder .../resultmajor#warning zurückgeliefert wird.</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. Die VersionID soll in der Form v1, v2, … vx gebildet werden.</td></tr>
</table></td></tr>
</table>

Name	**`ArchiveUpdateResponse`**



Statusinformationen und Fehler bei `ArchiveUpdateResponse` (vgl. **[eCard-1]** Abschnitt 4.1 und 4.2).

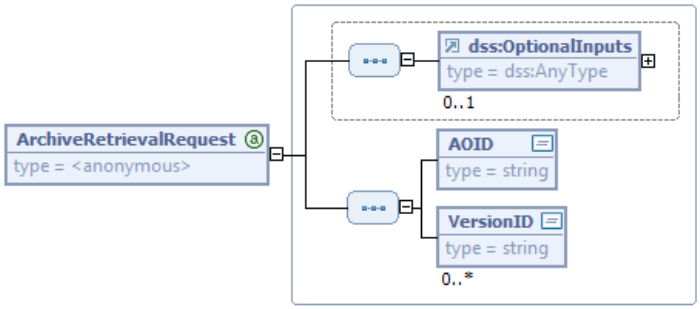
Name	**Fehlercode**
`ResultMajor`	• /resultmajor#ok • /resultmajor#error • /resultmajor#warning
`ResultMinor`	• /resultminor/al/common#noPermission • /resultminor/al/common#internalError • /resultminor/al/common#parameterError • /resultminor/arl/lowSpaceWarning • /resultminor/arl/noSpaceError • /resultminor/arl/existingPackage InfoWarning • /resultminor/arl/notSupported • /resultminor/arl/DXAIP_NOK • /resultminor/arl/DXAIP_NOK_AOID • /resultminor/arl/DXAIP_NOK_EXPIRED • /resultminor/arl/DXAIP_NOK_SUBMTIME • /resultminor/arl/DXAIP_NOK_SIG • /resultminor/arl/XAIP_NOK_ER • /resultminor/arl/DXAIP_NOK_ID • /resultminor/arl/DXAIP_NOK_Version

3.3 ArchiveRetrievalRequest und ArchiveRetrievalResponse

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

Wie in Abbildung Abbildung 2 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in den Schnittstellen S.2 (vgl. Abschnitt 4.2) und S.5 (vgl. Abschnitt 4.4) genutzt.

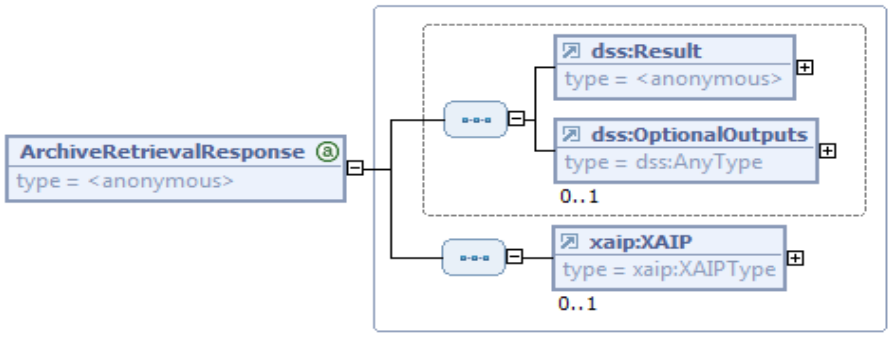
3.3.1 ArchiveRetrievalRequest

<table>
<tr><td>Name</td><td colspan="2"><code>ArchiveRetrievalRequest</code></td></tr>
<tr><td>Beschreibung</td><td colspan="2">Mit der Funktion <code>ArchiveRetrievalRequest</code> wird ein im Langzeitspeicher abgelegtes Archivdatenobjekt (<code>xaip:XAIP</code>) ausgelesen und zurückgeliefert.</td></tr>
<tr><td>Beschreibung</td><td colspan="2"> 

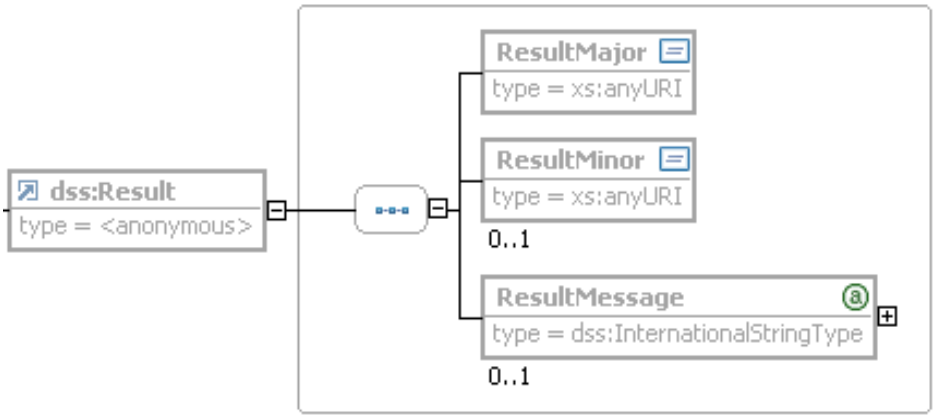
Aufruf der <code>ArchiveRetrievalRequest</code>-Funktion</td></tr>
<tr><td></td><td>Name</td><td>Beschreibung</td></tr>
</table>

Name	**`ArchiveRetrievalRequest`**	
	`dss:OptionalInputs`	Ist für optionale Eingabeelemente vorgesehen. **(A3.3.1-1):** Gemäß der vorliegenden Spezifikation soll das folgende optionale Eingabeelement unterstützt werden: IncludeERS. `IncludeERS` – gibt an, dass das zurückgelieferte XAIP den bzw. die entsprechenden Evidence Record im angegebenen Format (vgl. `ERSFormat`, Seite 22) enthalten sollen.  Dieser bzw. diese Evidence Record(s) wird bzw. werden im dafür vorgesehenen `xaip:credential/ xaip:EvidenceRecord` Element zurückgeliefert. **(A3.3.1-2):** Das `VersionID`-Attribut des `xaip:EvidenceRecord` Elementes muss auf die entsprechende Version verweisen. Sofern das `versionManifest` nicht kryptographisch geschützt ist, muss mit einem `unprotectedObjectPointer` Element im entsprechenden `versionManifest` auf die `credentialID` des `xaip:credential`-Elementes verwiesen werden. Umgekehrt muss auf die vom Evidence Record geschützten Datenobjekte im `relatedObjects`-Attribut des `xaip:credential`-Elementes verwiesen werden.
	`AOID`	Enthält den eindeutigen Identifikator des angeforderten Archivdatenobjektes.
	`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. Durch die Angabe von `all` werden alle existierenden Versionen eines Archivdatenobjektes zurückgeliefert.

3.3.2 ArchiveRetrievalResponse

<table>
<tr><td>Name</td><td><code>ArchiveRetrievalResponse</code></td></tr>
<tr><td>Beschreibung</td><td>Als Antwort auf einen <code>ArchiveRetrievalRequest</code> wird ein entsprechendes <code>ArchiveRetrievalResponse</code>-Element zurückgeliefert, welches im Erfolgsfall das angeforderte Archivdatenobjekt im <code>xaip:XAIP-Format</code> gemäß [TR-ESOR-F] enthält.</td></tr>
<tr><td>Rückgabe</td><td>


<code>ArchiveRetrievalResponse</code> ist die Antwort zum <code>ArchiveRetrievalRequest</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 weiter unten näher beschrieben.
Sofern nur ein Teil der angeforderten Versionen des Archivdatenobjektes zurückgeliefert werden konnte, wird dies durch den Fehlercode /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>xaip:XAIP</code></td><td>Sofern kein Fehler aufgetreten ist, wird das angeforderte XML-basierte Archivdatenobjekt (XAIP) gemäß [TR-ESOR-F] zurückgeliefert.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td colspan="2"><code>ArchiveRetrievalResponse</code></td></tr>
<tr><td></td><td colspan="2"> 

Statusinformationen und Fehler bei <code>ArchiveRetrievalResponse</code> (vgl. [eCard-1]).
<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/unknownAOID
• /resultminor/arl/notSupported
• /resultminor/arl/requestOnlyPartlySuccessfulWarning
• /resultminor/arl/unknownVersionID</td></tr>
<tr><td><code>ResultMessage</code></td><td>Beim Auftreten der Fehlermeldung .../unknownVersionID soll die problematische <code>VersionID</code> hier zurückgeliefert werden.</td></tr>
</table></td></tr>
</table>

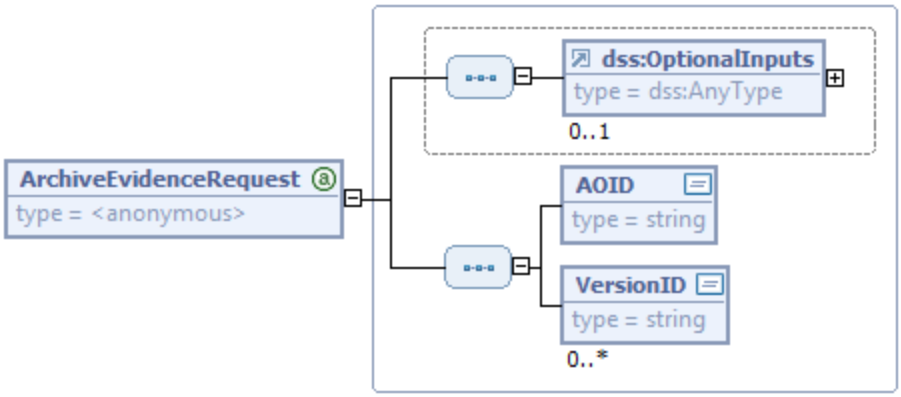
3.4 ArchiveEvidenceRequest und ArchiveEvidenceResponse

Mit der Funktion `ArchiveEvidenceRequest` können die zugehörigen technischen Beweisdaten (Evidence Records gemäß **[RFC4998]** oder **[RFC6283]**[4]) für beweiswerterhaltend aufbewahrte und über `AOID`-Elemente adressierte Archivdatenobjekte (`xaip:XAIP`) zurückgeliefert werden.

Wie in Abbildung Abbildung 2 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in TR-S.6 (vgl. Abschnitt 4.5) genutzt.

[4] **[RFC4998]** muss, **[RFC6283]** kann unterstützt werden.

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 [RFC6283][5] angefordert werden.</td></tr>
<tr><td>Beschreibung</td><td colspan="2"> 

Aufruf der <code>ArchiveEvidenceRequest</code>-Funktion
<table>
<tr><th>Name</th><th>Beschreibung</th></tr>
<tr><td><code>dss:OptionalInputs</code></td><td>Ist für optionale Eingabeelemente vorgesehen.
(A3.4.1-1): Gemäß der vorliegenden Spezifikation soll das folgende Element unterstützt werden:

Mit dem Element <code>tr:ERSFormat</code> vom Typ <code>anyURI</code> kann das gewünschte Format der zurückgelieferten Evidence Records angegeben werden, wobei folgende URIs vorgesehen sind:

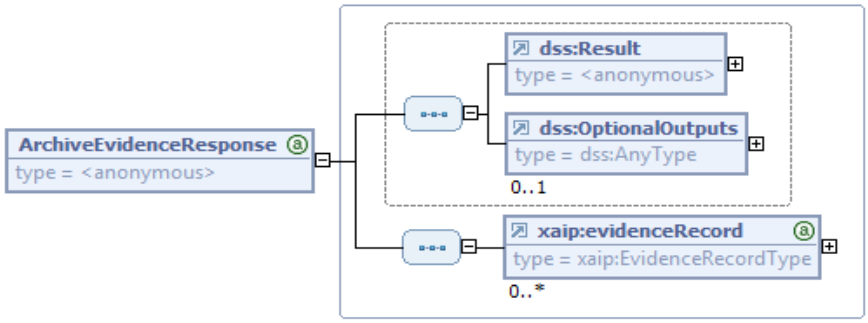
• urn:ietf:rfc:4998 für ASN.1-basierte Evidence Records gemäß [RFC4998] oder
• urn:ietf:rfc:6283 für XML-basierte Evidence Records gemäß [RFC6283].

Fehlt das <code>ERSFormat</code>-Element, so werden ASN.1-basierte Evidence Records gemäß [RFC4998] zurückgeliefert.</td></tr>
<tr><td><code>AOID</code></td><td>Ist der eindeutige Identifikator des angeforderten Archivdatenobjektes.</td></tr>
</table></td></tr>
</table>

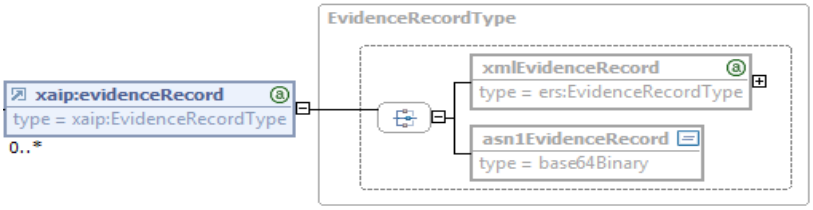
[5] **[RFC4998]** muss, **[RFC6283]** kann unterstützt werden.

Name	**ArchiveEvidenceRequest**	
	VersionID	Kann mehrfach auftreten und angeben für welche Versionen eines über die AOID identifizierten Archivdatenobjektes Evidence Records zurückgeliefert werden sollen. Sofern das VersionID-Element nicht angegeben ist, wird der Beweisdatensatz für die aktuelle Version des XAIP zurückgeliefert. Durch die Angabe von all werden Evidence Records für alle existierenden Versionen eines Archivdatenobjektes 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**
	dss:Result	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 übergebenen AOID adressierten Archivdatenobjekte entsprechende Beweisdaten (Evidence Records) zurückgeliefert werden konnten, wird dies durch die /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.

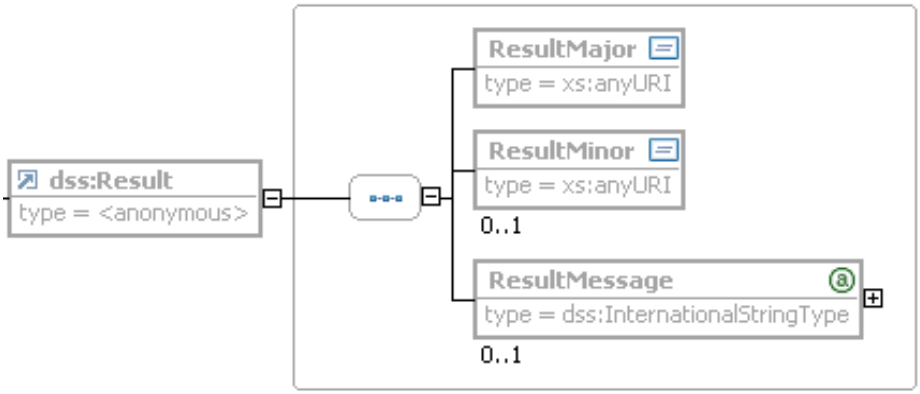
Name	**ArchiveEvidenceResponse**	
	`xaip:evidenceRecord`	Sofern vom ArchiSig-Modul entsprechende Evidence Records[6] gemäß **[RFC4998]** bzw. **[RFC6283]** konstruiert werden können, werden diese hier zurückgeliefert. Die detaillierte Struktur dieses Elementes ist nachfolgend erläutert.



Das `xaip:evidenceRecord`-Element gemäß **[TR-ESOR-F]** ist vom Typ **`xaip:EvidenceRecordType`**, der als Erweiterung des **`ec:EvidenceRecordType`** aus **[eCard-2]** definiert ist und zusätzlich die Attribute `AOID` und `VersionID`, enthält, die in **[TR-ESOR-F]** näher erläutert sind.

(A3.4.2-1): Bei der hier beschriebenen Verwendung von `xaip:evidenceRecord` müssen die Attribute `AOID` und `VersionID` gesetzt

Name	Beschreibung
`xmlEvidenceRecord`	Enthält einen XML-basierten Evidence Record gemäß **[RFC6283]**.
`asn1EvidenceRecord`	Enthält einen ASN.1-basierten Evidence Record gemäß **[RFC4998]**.



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

Name	Fehlercode

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

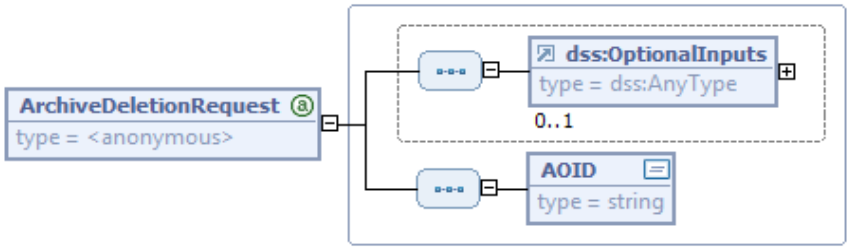
Name	**ArchiveEvidenceResponse**	
	ResultMajor	• /resultmajor#ok • /resultmajor#error • /resultmajor#warning
	ResultMinor	• /resultminor/al/common#noPermission • /resultminor/al/common#internalError • /resultminor/al/common#parameterError • /resultminor/arl/notSupported[7] • /resultminor/arl/unknownAOID • /resultminor/arl/unknownVersionID/ • resultminor/arl/requestOnlyPartly SuccessfulWarning

3.5 ArchiveDeletionRequest und ArchiveDeletionResponse

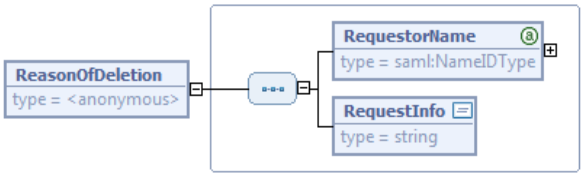
Mit der Funktion `ArchiveDeletionRequest` kann ein Archivdatenobjekt über die TR-ESOR-Middleware aus dem ECM-/Langzeitspeichersystem gelöscht werden.

Wie in Abbildung Abbildung 2 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in der Schnittstelle TR-S.5 (vgl. Abschnitt 4.4) genutzt.

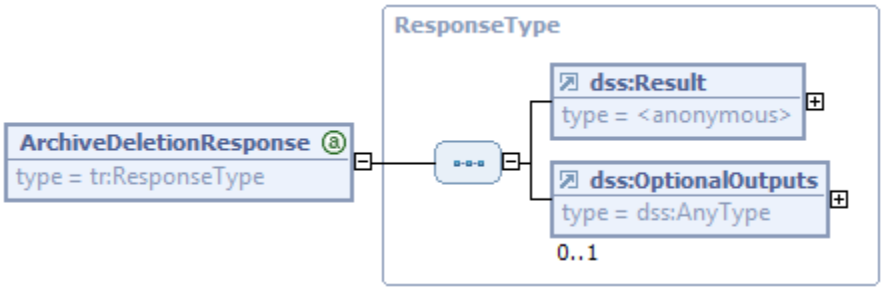
3.5.1 ArchiveDeletionRequest

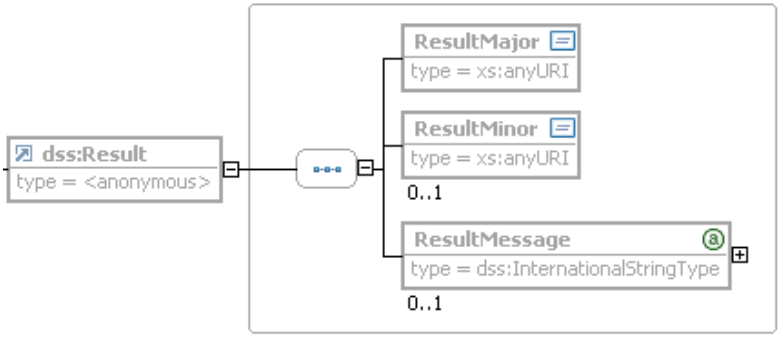
Name	**ArchiveDeletionRequest**	
Beschreibung	Mit der Funktion `ArchiveDeletionRequest` kann ein im Langzeitspeicher abgelegtes Archivdatenobjekt (`xaip:XAIP`) gelöscht werden.	
Beschreibung	 Aufruf der `ArchiveDeletionRequest`-Funktion	
	Name	**Beschreibung**

[7] Im `ResultMessage`-Element sollen nähere Informationen darüber zurückgeliefert werden, welche angeforderte Funktionalität nicht unterstützt wird.

Name	**ArchiveDeletionRequest**	
	dss:OptionalInputs	Ist für optionale Eingabeelemente vorgesehen. Insbesondere bei einer vorzeitigen Löschung muss das folgende Element ReasonOfDeletion genutzt und unterstützt werden: **(A3.5.1-1):** Das ReasonOfDeletion-Element muss vorhanden sein, sofern die Aufbewahrungsdauer der letzten Version noch nicht abgelaufen ist, und enthält neben dem Namen der aufrufenden Instanz auch eine Begründung für die Löschung. **(A3.5.1-2):** Die gesamte Aktion einschließlich der Begründung muss protokolliert werden und der übergebene RequestorName soll mit den verwendeten Authentisierungsinformationen  abgeglichen werden.
	AOID	Das AOID-Element gibt an, welches Archivdatenobjekt gelöscht werden soll.

3.5.2 ArchiveDeletionResponse

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

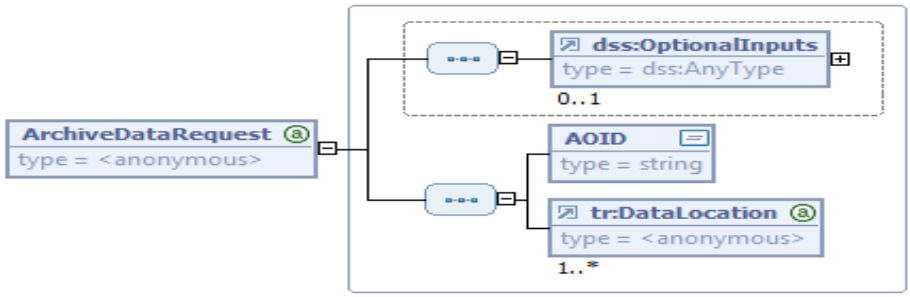
Statusinformationen und Fehler bei <code>ArchiveDeletionResponse</code> (vgl. [eCard-1]).</td></tr>
<tr><td>Name</td><td>Fehlercode</td></tr>
<tr><td><code>ResultMajor</code></td><td>• /resultmajor#ok
• /resultmajor#error</td></tr>
<tr><td><code>ResultMinor</code></td><td>• /resultminor/al/common#noPermission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

3.6 ArchiveDataRequest und ArchiveDataResponse

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

Wie in Abbildung Abbildung 2 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in TR-S.5 (vgl. Abschnitt 4.4) genutzt.

3.6.1 ArchiveDataRequest

<table>
<tr><td>Name</td><td><code>ArchiveDataRequest</code></td></tr>
<tr><td>Beschreibung</td><td>Mit der Funktion <code>ArchiveDataRequest</code> können diskrete Datenelemente aus einem im ECM-/Langzeitspeichersystem abgelegten, zumindest logisch im <code>xaip:XAIP-Format</code> gemäß [TR-ESOR-F] vorliegenden, Archivdatenobjekt ausgelesen werden.</td></tr>
<tr><td>Beschreibung</td><td> 

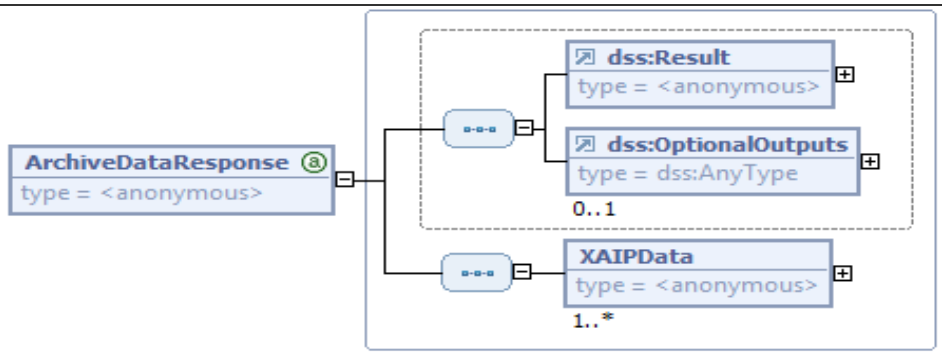
Aufruf der <code>ArchiveDataRequest</code>-Funktion
<table>
<tr><th>Name</th><th>Beschreibung</th></tr>
<tr><td><code>dss:OptionalInputs</code></td><td>Ist für optionale Eingabeelemente vorgesehen und kann beispielsweise Steuerelemente (<code>requestControls</code>) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen.
Die vorliegende Spezifikation definiert keine solchen optionalen Eingabelemente.</td></tr>
<tr><td><code>AOID</code></td><td>Dieses Element enthält den Identifikator eines bestimmten Archivdatenobjektes.</td></tr>
<tr><td><code>tr:DataLocation</code></td><td>Das <code>tr:DataLocation</code>-Element kann mehrmals auftreten und bestimmt die „Lokation“ der auszulesenden diskreten Datenelemente bezüglich eines zumindest logisch im <code>xaip:XAIP</code>-Format gemäß [TR-ESOR-F] vorliegenden Archivdatenobjektes.[8]</td></tr>
</table></td></tr>
</table>

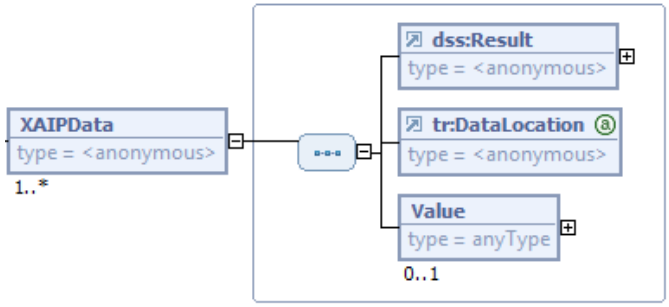
[8] Ausgehend von einem XML-basierten Archivdatenobjekt bieten sich für die diskrete Adressierung von XML-Datenelementen hier XPath (siehe unter: http://www.w3.org/TR/2007/REC-xpath20-20070123/), XQuery (siehe unter: http://www.w3.org/TR/2007/REC-xquery-20070123/) oder die XML Pointer Language XPointer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

Name	`ArchiveDataRequest`

	Das `DataLocation`-Element spezifiziert, welche Teile eines Archivobjektes zurückgeliefert werden sollen und ist folgendermaßen definiert:
	`<element name="DataLocation">`
	`  <complexType>`
	`  <complexContent>`
	`  <extension base="anyType">`
	`  <attribute name="Type" type="anyURI"/>`
	`  </extension>`
	`  </complexContent>`
	`  </complexType>`
	`</element>`
	Im `Type`-Attribut wird angegeben, welche Transformation für den Zugriff auf 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

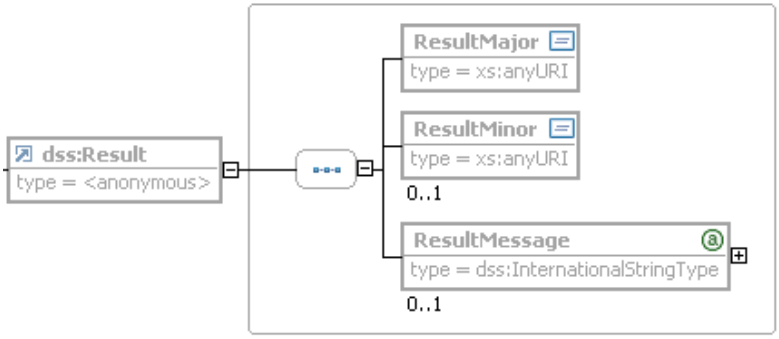
3.6.2 ArchiveDataResponse

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.

<table>
<tr><td>Name</td><td colspan="2"><code>ArchiveDataResponse</code></td></tr>
<tr><td rowspan="6"></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>XAIPData</code></td><td>Enthält im Erfolgsfall die gewünschten Daten und die „Lokation“, aus der diese aus der im ECM-/Langzeitspeichersystem zumindest logisch existierenden <code>XAIP</code>-Struktur ausgelesen wurden. Die detaillierte Struktur dieses Elementes ist nachfolgend dargestellt und erläutert.</td></tr>
<tr><td colspan="2"> 

Das <code>XAIPData</code>-Element enthält im Erfolgsfall die gewünschten Daten.</td></tr>
<tr><td>Name</td><td>Beschreibung</td></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/arl/unknownLocation
• .../resultminor/al/common#parameterError
• .../resultminor/al/common#internalError</td></tr>
<tr><td><code>tr:DataLocation</code></td><td>Das <code>DataLocation</code>-Element spezifiziert, welche Teile eines Archivobjektes zurückgeliefert werden. Weitere Details zu diesem Element finden sich auf Seite 29.</td></tr>
<tr><td><code>Value</code></td><td>Enthält im Erfolgsfall die gewünschten Daten.</td></tr>
</table>

<table>
<tr><td>Name</td><td><code>ArchiveDataResponse</code></td></tr>
<tr><td></td><td>


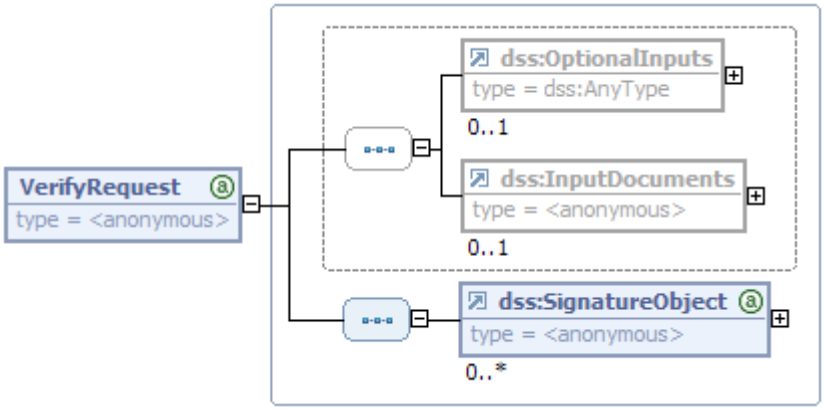
Statusinformationen und Fehler bei <code>ArchiveDataResponse</code> (vgl. [eCard-1]).
<table>
<tr><th>Name</th><th>Fehlercode</th></tr>
<tr><td><code>ResultMajor</code></td><td>• /resultmajor#ok
• /resultmajor#error
• /resultmajor#warning</td></tr>
<tr><td><code>ResultMinor</code></td><td>• /resultminor/al/common#noPermission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

3.7 VerifyRequest und VerifyResponse

3.7.1 VerifyRequest

Mit der Funktion `VerifyRequest` können XML-basierte Archivdatenobjekte (XAIP) oder ergänzende XML-basierte Archivdatenobjekte (Delta-XAIP)samt der darin enthaltenen oder zusätzlich übergebenen beweisrelevanten Daten (Signaturen, Siegel, Zeitstempel, Zertifikate, Sperrlisten, OCSP-Responses etc.) und Beweisdaten (Evidence Records) geprüft werden.

Wie in Abbildung Abbildung 2 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in TR-S.1 (vgl. Abschnitt 4.1) genutzt.

<table>
<tr><td>Name</td><td><code>VerifyRequest</code></td></tr>
<tr><td>Beschreibung</td><td>Mit der Funktion <code>VerifyRequest</code> (vgl. Abschnitt 3.2.2 von [eCard-2]) werden XML-basierte Archivdatenobjekte (XAIP) bzw. ergänzendes XML-basiertes Archivdatenobjekt (Delta-XAIP) samt der darin enthaltenen oder zusätzlich übergebenen beweisrelevanten Daten (Signaturen, Siegel, Zeitstempel, Zertifikate, Sperrlisten, OCSP-Responses etc.) und Beweisdaten (Evidence Records) geprüft.</td></tr>
<tr><td>Aufrufparameter</td><td>Aufruf der <code>VerifyRequest</code>-Funktion.
 

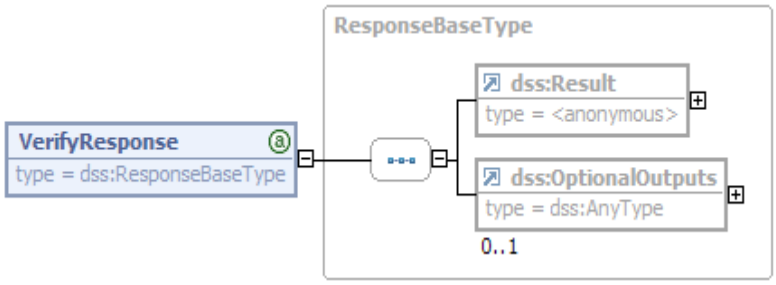
<table>
<tr><td>Name</td><td>Beschreibung</td></tr>
</table></td></tr>
</table>

Name	**`VerifyRequest`**	
	`dss:OptionalInputs`	Entspricht dem `requestControls`-Element und kann zusätzliche Eingabeelemente enthalten. **(A3.7.1-1):** Hierbei sollen insbesondere die in **[eCard-2]** definierten Elemente und Aufrufoptionen unterstützt werden. Dies umfasst insbesondere die folgenden Elemente: • `VerifyUnderSignaturePolicy`, • `ReturnVerificationReport` `Es gilt im Einzelnen:` • `VerifyUnderSignaturePolicy` Sofern in einem `dss:Document/InlineXML`-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/DE/tr-esor/sigpolicy/verify-timestamp` die Prüfung und Ergänzung aller im übergebenen XAIP-Container enthaltenen digitalen Signaturen angefordert werden. **(A3.7.1-2):** Hierbei müssen alle digitalen Signaturinformationen (Signaturen, Siegel, 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 als unsignierte Attribute bzw. Properties in den entsprechenden digitalen Signaturen bzw. in den Kind-Elementen `certificateValues` bzw. `revocationValues` des `credential`-Elementes abgelegt.

Name	**VerifyRequest**	
		(A3.7.1-3): Sofern in der `credentialSection` des übergebenen XAIP-Containers ein oder mehrere `xaip:EvidenceRecord`-Elemente gemäß **[TR-ESOR-F]** enthalten sind, müssen diese entsprechend geprüft werden. • `ReturnVerificationReport` Durch die Übergabe eines `ReturnVerificationReport`-Elementes gemäß **[OASIS VR]** bzw. **[eCard-2]** und **[TR-ESOR-VR]** kann ein ausführlicher Prüfbericht in Form eines `VerificationReport`-Elementes für die übergebenen Objekte (Signaturen, Siegel, Zeitstempel, Zertifikate, Sperrinformationen, Evidence Records, XAIP mit den vorgenannten Daten) angefordert werden.
	`dss:InputDocuments`	Das `dss:InputDocuments`-Element enthält die zur Prüfung benötigten Dokumente, sofern diese nicht bereits im unten erläuterten `SignatureObject`-Element enthalten sind. Außerdem kann in einem `dss:Document/InlineXML`-Kindelement ein `XAIP`-Element gemäß **[TR-ESOR-F]** übergeben werden, so dass alle darin enthaltenen digitalen Signaturen in Verbindung mit der oben angegebenen Signature-Policy geprüft und ergänzt werden oder die Prüfung der darin enthaltenen Evidence Records angestoßen wird.

Name	**VerifyRequest**	
	dss:SignatureObject	In dss:SignatureObject-Elementen können grundsätzlich eigenständige digitale Signaturen (detached digital signatures) zur Prüfung übergeben werden. Wenn digitale Signaturen bereits im dss:InputDocuments enthalten sind, können die optionalen dss:SignatureObject-Elemente entfallen. **(A3.7.1-4):** Als Kindelement von dss:SignatureObject/Other kann auch ein xaip:EvidenceRecord-Element übergeben werden, um die entsprechende Prüfung des Evidence Record anzustoßen. In diesem Fall müssen die Attribute AOID und VersionID vorhanden sein und das zugehörige XAIP-Element muss als Kindelement von dss:InputDocuments/dss:Document/InlineXML übergeben werden. Sofern das dss:SignatureObject-Element fehlt, muss genau ein dss:InputDocuments-Element vorhanden sein, das die zu prüfenden digitalen Signaturobjekte enthält.

3.7.2 VerifyResponse

Name	**VerifyResponse**	
Beschreibung	Als Antwort auf einen VerifyRequest wird ein entsprechendes VerifyResponse-Element gemäß Abschnitt 3.2.2 von **[eCard-2]** zurückgeliefert.	
Rückgabe		
	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.

Name	**`VerifyResponse`**	
	`dss:OptionalOutputs`	Sofern ein Fehler aufgetreten ist, enthält dieses Element entweder den Prüfbericht in Form eines `VerificationReport`-Elementes oder das um diese Prüfinformationen ergänzte Archivdatenobjekt in Form eines `xaip:XAIP`-Elements. Die grundsätzliche Struktur des Prüfberichtes ist in **[OASIS-VR]** näher beschrieben. In **[TR-ESOR-VR]** finden sich entsprechende Korrekturen für den `EvidenceRecordReport` sowie die Beschreibung des `XAIPReport`. Details zur Ablage dieser Prüfinformationen im XAIP-Container finden sich in **[TR-ESOR-F]**.

4. Funktionen der interne Schnittstellen

In diesem Abschnitt werden die internen Schnittstellen der Referenzarchitektur TR-S.1 bis TR-S.3 und TR-S.5 bis TR-S.6 (vgl. Abbildung Abbildung 2) erläutert:

- TR-S.1: TR-ESOR-S.1 (ArchiSafe-Modul – Krypto-Modul) (siehe Abschnitt 4.1)
- TR-S.2: TR-ESOR-S.2 (ArchiSig-Modul – ECM-/Langzeitspeichersystem) (siehe Abschnitt 4.2)
- TR-S.3: TR-ESOR-S.3 (ArchiSig-Modul – Krypto-Modul) (siehe Abschnitt 4.3)
- TR-S.5: TR-ESOR-S.5 (ArchiSafe-Modul –ECM-/Langzeitspeichersystem) (siehe Abschnitt 4.4)
- TR-S.6: TR-ESOR-S.6 (ArchiSafe-Modul – ArchiSig-Modul) (siehe Abschnitt 4.5)

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

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

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

- Prüfung von digitalen Signaturen, beweisrelevanten Daten, Beweisdaten und Archivdatenobjekten (`VerifyRequest / VerifyResponse`)
- Anforderung von digitalen Signaturen (optional) (`SignRequest / SignResponse`)

4.1.1 Prüfung von digitalen Signaturen, beweisrelevanten Daten, Beweisdaten und Archivdatenobjekten

Für die Prüfung von digitalen Signaturen, beweisrelevanten Daten (Zertifikaten, Zertifikatstatusinformationen, Zeitstempeln, etc.), Beweisdaten (Evidence Records) und Archivdatenobjekten (XAIPs) 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]** mit den entsprechenden Korrekturen und Ergänzungen aus **[TR-ESOR-VR]**, wie in Abschnitt 3.7 erläutert.

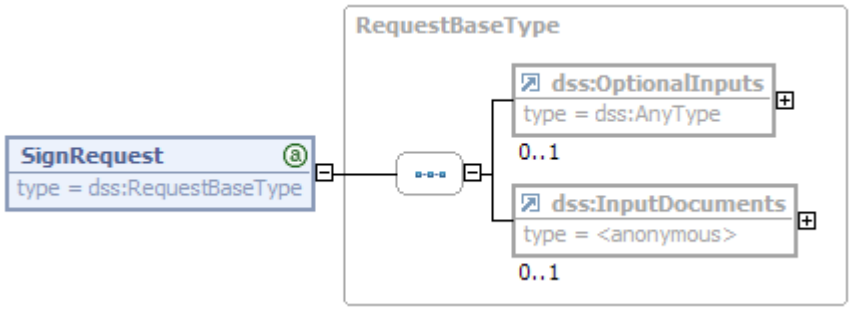
Die Durchführung der eigentlichen Prüffunktion von beweisrelevanten Daten sowie Beweisdaten muss im Krypto-Modul (siehe Anlage **[TR-ESOR-M.2]**) als Komponente der TR-ESOR-Middleware oder in einem, vom Krypto-Modul aufgerufen, (qualifizierten) Vertrauensdiensteanbieter erfolgen. Die für die Prüfung notwendigen Prüfinformationen müssen von den Vertrauensdiensteanbietern abgerufen werden.

4.1.2 Anforderung einer digitalen Signatur

Für die Anforderung einer digitalen Signatur 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]**.

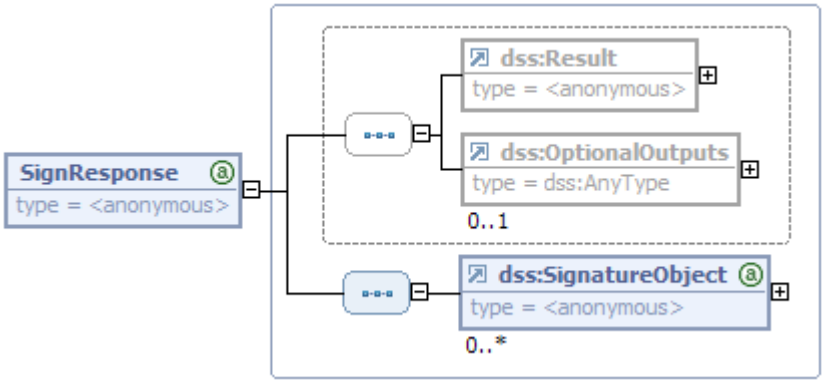
4.1.2.1 SignRequest (Anforderung einer digitalen Signatur)

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

Name	**`SignRequest`**
Beschreibung	Mit der Funktion `SignRequest` aus **[eCard-2]** kann für das übergebene Archivdatenobjekt eine (qualifizierte) digitale Signatur von einem (qualifizierten) Vertrauensdiensteanbieter gemäß **[eIDAS-VO, Artikel 3 Nr. 19 bzw. Nr. 20]** angefordert 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]**.

4.1.2.2 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.
Rückgabe	 `SignResponse` ist die Antwort zum `SignRequest`-Aufruf

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 3.2.1 von **[eCard-2]** beschrieben.

Name	SignResponse	
	dss:OptionalOutputs	Kann ein DocumentWithSignature-Element enthalten, in denen z.B. ein XAIP-Element mit der eingebetteten digitalen Signatur enthalten ist. Details finden sich in Abschnitt 3.2.1 von **[eCard-2]**.
	dss:SignatureObject	Kann eine erzeugte digitale Signatur in Form eines dss:SignatureObject-Elementes enthalten. Details finden sich in Abschnitt 3.2.1 von **[eCard-2]**. Sofern die erstellte digitale Signatur bereits im o.g. DocumentWithSignature-Element vorhanden ist, wird kein dss:SignatureObject-Element zurückgeliefert.

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

Dieser Abschnitt beschreibt, wie die in **[TR-ESOR-S]** skizzierte Schnittstelle TR-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)
- Ergänzen einer neuen Version eines Archivdatenobjektes (ArchiveUpdateRequest / ArchiveUpdateResponse)
- Auslesen eines Archivdatenobjektes (ArchiveRetrievalRequest / ArchiveRetrievalResponse)

4.2.1 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.

4.2.2 Ergänzen einer neuen Version eines Archivdatenobjektes

Für das Ergänzen einer neuen Version eines Archivdatenobjektes 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.

4.2.3 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.

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

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

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

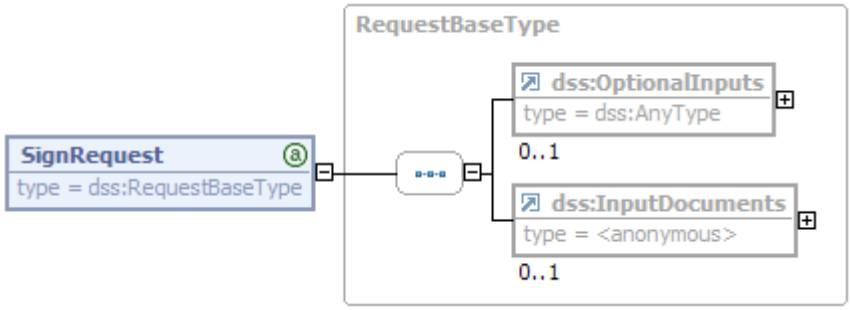
- Anfordern eines (qualifizierten) Zeitstempels (`TimestampRequest / TimeStampResponse`)
- Prüfen eines (qualifizierten) Zeitstempels (`VerifyRequest / VerifyResponse`)
- Berechnung eines Hashwertes (`Hash / HashResponse`)

4.3.1 Anfordern eines (qualifizierten) Zeitstempels

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

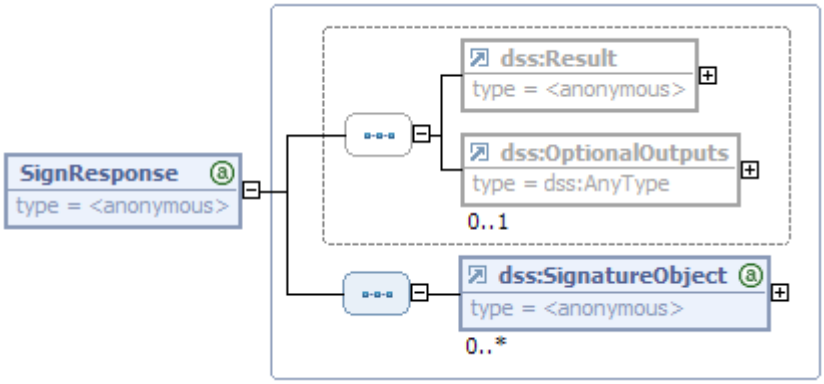
Der qualifizierte Zeitstempel muss von einem (vertrauenswürdigen) qualifizierten Vertrauensdienstanbieter gemäß **[eIDAS-VO, Artikel 3 Nr. 20]** durch das Krypto-Modul (siehe Anlage **[TR-ESOR-M.2]**) als eine Komponente der Middleware angefordert werden.

4.3.1.1 TimestampRequest wird realisiert durch SignRequest

<table>
<tr><td>Name</td><td colspan="2"><code>SignRequest</code></td></tr>
<tr><td>Beschreibung</td><td colspan="2">Ein <code>SignRequest</code> im Kontext der Schnittstelle S.3 übergibt einen Hashwert, zu dem ein (qualifizierter) Zeitstempel erstellt werden soll, an das Krypto-Modul.</td></tr>
<tr><td>Beschreibung</td><td colspan="2"> 

Aufruf der <code>SignRequest</code>-Funktion
<table>
<tr><th>Name</th><th>Beschreibung</th></tr>
<tr><td><code>dss:OptionalInputs</code></td><td>Enthält genau ein Element <code>SignatureType</code> mit der URI urn:ietf:rfc:3161, durch die klargestellt wird, dass ein Zeitstempel gemäß [RFC3161] erzeugt werden soll.</td></tr>
<tr><td><code>dss:InputDocuments</code></td><td>(A4.3.1.1-1): Während das Element <code>dss:InputDocuments</code> in [OASIS-DSS] und [eCard-2] optional ist, muss es hier vorhanden sein und genau ein <code>dss:Document</code>-Element in der <code>DocumentHash</code>-Ausprägung enthalten. Dieses Element enthält den Hashwert, aus dem ein (qualifizierter) Zeitstempel erzeugt werden soll.</td></tr>
</table></td></tr>
</table>

4.3.1.2 TimestampResponse wird realisiert durch SignResponse

<table>
<tr><td>Name</td><td colspan="2"><code>SignResponse</code></td></tr>
<tr><td>Beschreibung</td><td colspan="2">Als Antwort auf einen <code>SignRequest</code> wird vom Krypto-Modul ein entsprechendes <code>SignResponse</code>-Element gemäß Abschnitt 3.2.1 von [eCard-2] zurückgeliefert. Im Kontext der Schnittstelle S.3 wird hier ein (qualifizierter) Zeitstempel zurückgeliefert.</td></tr>
<tr><td>Rückgabe</td><td colspan="2"> 

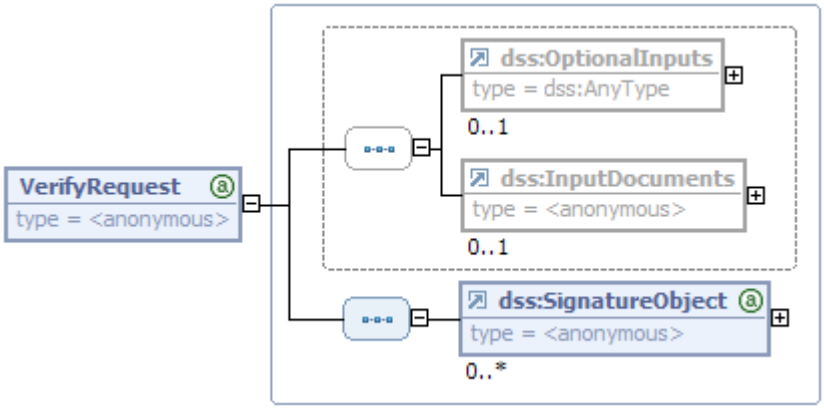
<code>SignResponse</code> ist die Antwort zum <code>SignRequest</code>-Aufruf
<table>
<tr><th>Name</th><th>Beschreibung</th></tr>
<tr><td><code>dss:Result</code></td><td>Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements und die möglichen Fehlercodes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

4.3.2 Prüfen eines (qualifizierten) Zeitstempels

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

Die Durchführung der eigentlichen Prüffunktion eines (qualifizierten Zeitstempels) muss im Krypto-Modul (siehe Anlage **[TR-ESOR-M.2]**) als Komponente der TR-ESOR-Middleware oder in einem, vom Krypto-Modul aufgerufen, (qualifizierten) Vertrauensdiensteanbieter erfolgen. Die für die Prüfung notwendigen Prüfinformationen müssen von den (qualifizierten) Vertrauensdiensteanbietern abgerufen werden.

4.3.2.1 VerifyRequest

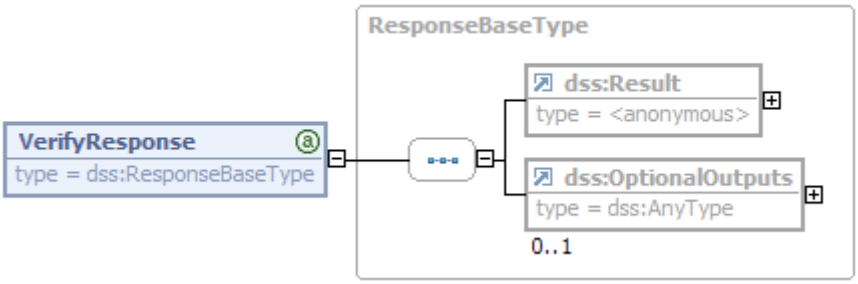
<table>
<tr><td>Name</td><td colspan="2"><code>VerifyRequest</code></td></tr>
<tr><td>Beschreibung</td><td colspan="2">Ein <code>VerifyRequest</code> im Kontext der Schnittstelle S.3 übergibt einen (qualifizierten) Zeitstempel an das Krypto-Modul zur Verifikation der darin enthaltenen (qualifizierten) digitalen 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].</td></tr>
<tr><td>Aufrufparameter</td><td colspan="2"> 

Aufruf der <code>VerifyRequest</code>-Funktion.
<table>
<tr><th>Name</th><th>Beschreibung</th></tr>
<tr><td><code>dss:OptionalInputs</code></td><td>Entspricht dem <code>requestControls</code>-Element aus [TR-ESOR-S] und kann optionale Eingabeelemente enthalten.
(A4.3.2.1-1): Gemäß der vorliegenden Spezifikation muss das optionale Eingabeelement <code>ReturnUpdatedSignature</code> aus Abschnitt 4.5.8 von [OASIS-DSS] unterstützt werden, bei dem mit dem <code>Type</code>-Attribut http://www.bsi.bund.de/tr-esor/api/1.2 klargestellt wird, dass alle bei der Prüfung verwendeten Zertifikate und Sperrinformationen wie in [TR-ESOR-F] spezifiziert in den Zeitstempel eingefügt werden müssen.

(A4.3.2.1-2): Darüber hinaus soll das optionale Eingabeelement <code>ReturnVerificationReport</code> unterstützt werden, so dass für den entsprechenden Zeitstempel ein Prüfbericht gemäß [OASIS-VR] zurückgeliefert werden kann.</td></tr>
<tr><td><code>dss:InputDocuments</code></td><td>Das optionale Element <code>dss:InputDocuments</code> soll nicht vorhanden sein und wird ignoriert.</td></tr>
</table></td></tr>
</table>

Name	`VerifyRequest`	
	`dss:SignatureObject`	Es ist genau ein `dss:SignatureObject`-Element in der `dss:TimeStamp/ RFC3161TimeStampToken` Ausprägung vorhanden, das den zu prüfenden Zeitstempel enthält.

4.3.2.2 VerifyResponse

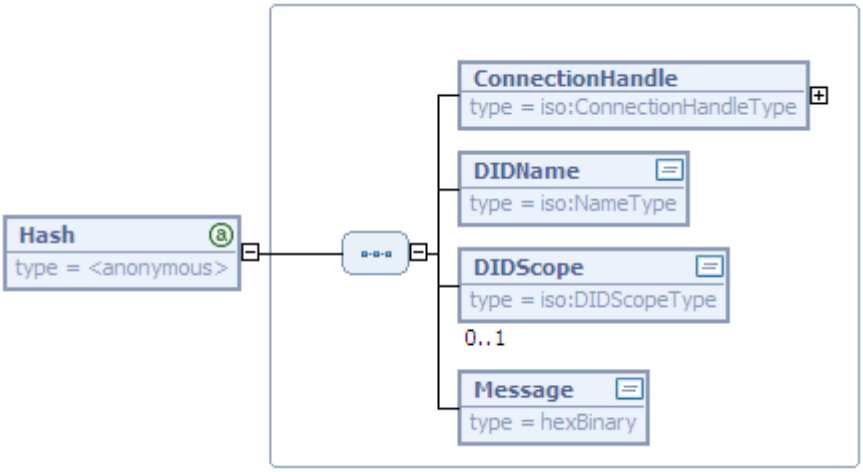
Name	`VerifyResponse`	
Beschreibung	Als Antwort auf einen `VerifyRequest` wird vom Krypto-Modul ein entsprechendes `VerifyResponse`-Element gemäß Abschnitt 3.2.2 von **[eCard-2]** zurückgeliefert.	
Rückgabe	 Rückgabe der `VerifyRequest`-Funktion	
	Name	**Beschreibung**
	`dss:Result`	Enthält die Statusinformationen 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, muss ein `UpdatedSignature`-Element vorhanden sein, das ein `dss:SignatureObject`-Element in der `dss:TimeStamp/ RFC3161TimeStampToken`-Ausprägung enthält, in dem sich der um die bei der Prüfung genutzten Zertifikate und Sperrinformationen ergänzte Zeitstempel befindet. Darüber hinaus kann ein `VerificationReport`-Element gemäß **[OASIS VR]** vorhanden sein, das im `IndividualReport/Details`-Element ein `IndividualTimeStampReport`-Element enthält.

4.3.3 Berechnung eines Hashwertes

Zur Berechnung eines Hashwertes sind in TR-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]**.

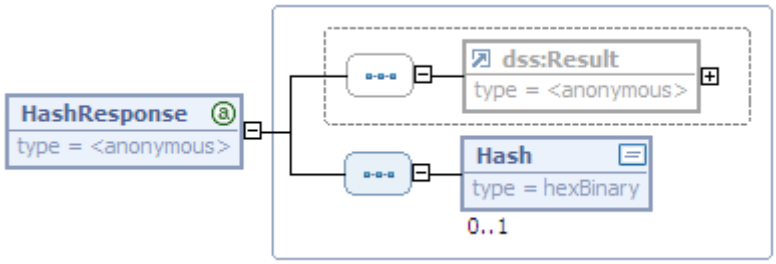
4.3.3.1 Hash

<table>
<tr><td>Name</td><td colspan="2"><code>Hash</code></td></tr>
<tr><td>Beschreibung</td><td colspan="2">Bei einem <code>Hash</code>-Aufruf im Kontext der Schnittstelle S.3 wird für die übergebenen Daten ein Hashwert berechnet.</td></tr>
<tr><td>Aufruf parameter</td><td colspan="2"> 

Aufruf der Funktion <code>Hash</code>.
<table>
<tr><th>Name</th><th>Beschreibung</th></tr>
<tr><td><code>ConnectionHandle</code></td><td>Das <code>ConnectionHandle</code>-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 <code>ConnectionHandle</code>-Element leer sein.</td></tr>
<tr><td><code>DIDName</code>[9]</td><td>Dieser Parameter spezifiziert den zu verwendenden Hashalgorithmus, wobei derzeit folgende Algorithmen unterstützt werden müssen:
SHA-256
SHA-384
SHA-512</td></tr>
<tr><td><code>DIDScope</code></td><td>Löst im ISO/IEC 24727-3 Standard Mehrdeutigkeiten zwischen lokalen und globalen DIDs mit gleichem Namen auf. Dieser Parameter wird hier nicht verwendet und sofern vorhanden ignoriert.</td></tr>
<tr><td><code>Message</code></td><td>Enthält die Nachricht (bzw. einen Teil derselben, siehe [eCard-7]), aus der ein Hashwert berechnet werden soll.</td></tr>
</table></td></tr>
</table>

[9] Eine in ISO/IEC 24727 näher beschriebene Differential Identity ermöglicht die Ausführung von kryptographischen Operationen. Der `DIDName` ist der logische Name, der für den Zugriff auf dieses kryptographische Objekt genutzt wird.

4.3.3.2 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.

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

Dieser Abschnitt beschreibt, wie die in TR-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-S.5 definierte Schnittstelle umfasst die folgenden Funktionen:

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

4.4.1 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.

4.4.2 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.

4.4.3 Abfrage diskreterDatenobjekte

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

Dieser Abschnitt beschreibt, wie die in **[TR-ESOR-S]** skizzierte Schnittstelle TR-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 TR-S.6 umfasst die folgenden Funktionen:

- Beweiswerterhaltende Archivierung elektronischer Daten (`ArchiveSubmissionRequest / ArchiveSubmissionResponse`)
- Ergänzen einer neuen Version eines Archivdatenobjektes (`ArchiveUpdateRequest / ArchiveUpdateResponse`)
- Rückgabe technischer Beweisdaten (`ArchiveEvidenceRequest / ArchiveEvidenceResponse`)

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 Ergänzen einer neuen Version eines Archivdatenobjektes

Für das Ergänzen einer neuen Version eines Archivdatenobjektes 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 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. Fehlercodes

Die vorliegende Spezifikation nutzt die folgenden generellen Fehlercodes aus **[eCard-1]**:

- .../resultmajor#ok
- .../resultmajor#error
- .../resultmajor#warning
- .../resultminor/al/common#noPermission
- .../resultminor/al/common#internalError
- .../resultminor/al/common#parameterError

Darüber hinaus werden zusätzlich die folgenden Fehlercodes definiert:

Fehlercode	Beschreibung
.../resultminor/arl/DXAIP_NOK	Die Syntax des beim `ArchiveUpdateRequest` übergebenen `DXAIP`-Elements ist nicht korrekt.
.../resultminor/arl/DXAIP_NOK_AOID	Die `AOID` in dem beim `ArchiveUpdateRequest` übergebenen Delta-`XAIP` ist nicht bekannt.
.../resultminor/arl/DXAIP_NOK_EXPIRED	Das beim `ArchiveUpdateRequest` übergebene Delta-`XAIP`-Element kann nicht abgelegt werden, da die Aufbewahrungsfrist abgelaufen ist.
.../resultminor/arl/DXAIP_NOK_SUBMTIME	Die beim `ArchiveUpdateRequest` im übergebenen Delta-XAIP-Element angegebene `submissionTime` ist nicht korrekt, da sie in der Zukunft liegt.
.../resultminor/arl/DXAIP_NOK_SIG	Das beim `ArchiveUpdateRequest` übergebene Delta-XAIP-Element enthält zumindest eine ungültige digitale Signatur.
.../resultminor/arl/DXAIP_NOK_ER	Das beim `ArchiveUpdateRequest` übergebene Delta-`XAIP`-Element enthält zumindendest einen ungültigen Evidence Record.
.../resultminor/arl/DXAIP_NOK_ID	Die beim `ArchiveUpdateRequest` in einem `placeHolder`-Element übebergebene XML-`ID` ist im bereits abgelegten XAIP-Elemen nicht vorhanden.
.../resultminor/arl/DXAIP_NOK_Version	Die beim `ArchiveUpdateRequest` im `prevVersion`-Element übergebene Version ist nicht die aktuellste Version.
.../resultminor/arl/existingAOID	Die im Rahmen des `ArchiveSubmissionRequest` übergebene `AOID` existiert bereits.
.../resultminor/arl/existingPackageInfoWarning	Bei der `ArchiveUpdateRequest`-Funktion wird ein `DXAIP`-Element übergeben, das ein `packageInfo`-Element enthält. Da im vorher existierenden XAIP bereits das `packageInfo`-Element belegt war, wird das übergebene `packageInfo`-Element ignoriert und eine entsprechende Warnung zurückgeliefert.

Fehlercode	Beschreibung
.../resultminor/arl/lowSpaceWarning	Diese Warnung gibt an, dass der verfügbare Speicherplatz einen kritischen Wert unterschritten hat.
.../resultminor/arl/missingReasonOf Deletion	Da beim `ArchiveDeletionRequest` kein `ReasonOfDeletion`-Element übergeben wurde, muss der Löschvorgang abgewiesen werden.
.../resultminor/arl/noSpaceError	Diese Fehlermeldung gibt an, dass kein Speicherplatz verfügbar war und deshalb das Archivdatenobjekt nicht abgelegt werden konnte.
.../resultminor/arl/notSupported	Diese Fehlermeldung gibt an, dass eine angeforderte Funktion, ein angefordertes Format oder ein übergebener optionaler Eingabeparameter nicht unterstützt wird.
.../resultminor/arl/requestOnlyPartly SuccessfulWarning	Diese Warnung gibt an, dass nicht alle angeforderten Daten zurückgeliefert werden konnten.
.../resultminor/arl/unknownArchiveDataType	Es wird ein binäres Datenobjekt mit einem nicht unterstützten Datenformat übergeben.
.../resultminor/arl/unknownLocation	Die im `ArchiveDataRequest` angegebene `DataLocation` ist nicht vorhanden bzw. fehlerhaft.
.../resultminor/arl/unknownAOID	Die übergebene `AOID` existiert nicht.
.../resultminor/arl/unknownVersionID	Die übergebene `VersionID` ist im entsprechenden XAIP nicht bekannt.
.../resultminor/arl/XAIP_NOK	Die Syntax des übergebenen `XAIP`-Elements ist nicht korrekt.
.../resultminor/arl/XAIP_NOK_ER	Das übergebene `XAIP`-Element enthält zumindendest einen ungültigen Evidence Record.
.../resultminor/arl/XAIP_NOK_EXPIRED	Das übergebene `XAIP`-Element kann nicht abgelegt werden, da die Aufbewahrungsfrist abgelaufen ist.
.../resultminor/arl/XAIP_NOK_SIG	Das übergebene XAIP-Element enthält zumindest eine ungültige Signatur.
.../resultminor/arl/XAIP_NOK_SUBMTIME	Die im übergebenen XAIP-Element angegebene `submissionTime` ist nicht korrekt, da sie in der Zukunft liegt.

6. 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 6.1). Darauf aufbauend wird in einem zweiten Schritt eine Webservice-Spezifikation gemäß **[WSDL]** entwickelt.

Abschnitt 6.2 enthält die Webservice-Spezifikation der Schnittstelle TR-S.4 (vgl. Abschnitt 3). 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 6.2 unterstützt werden. Darüber hinaus können weitere Schnittstellen, wie z.B. eine sprachgebundene Java- oder C-Schnittstelle gemäß **[eCard-1]** unter Verwendung der in Abschnitt 6.1 spezifizierten XML-Strukturen unterstützt werden.

6.1 Spezifikation der Aufruf- und Rückgabeparameter als XML-Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
      xmlns:tr="http://www.bsi.bund.de/tr-esor/api/1.2"
      xmlns:xaip="http://www.bsi.bund.de/tr-esor/xaip/1.2"
      xmlns:ers="urn:ietf:params:xml:ns:ers"
      xmlns:ec="http://www.bsi.bund.de/ecard/api/1.1"
      xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema"
      xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
      targetNamespace="http://www.bsi.bund.de/tr-esor/api/1.2"
      elementFormDefault="qualified" attributeFormDefault="unqualified">

      <!-- ====================================== -->
      <!-- Version 1.2 vom 31.01.2015             -->
      <!-- ====================================== -->

      <import namespace="http://www.bsi.bund.de/tr-esor/xaip/1.2"
            schemaLocation="http://www.bsi.bund.de/SharedDocs/Download/tr-
esor/xaip/1_2/tr-esor-xaip-v1_2_xsd" />

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

      <import namespace="urn:ietf:params:xml:ns:ers"
            schemaLocation="http://ws.openecard.org/schema/xml-ers-
rfc6283.xsd" />

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

      <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
```

```
            schemaLocation="http://ws.openecard.org/schema/saml-schema-
assertion-2.0.xsd" />

      <!-- ================================== -->
      <!--     Uebergreifende 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>

      <element name="AOID" type="string"/>


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

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


      <element name="ImportEvidence" type="tr:ImportEvidenceType"/>
```

```
      <complexType name="ImportEvidenceType">
            <choice>
                  <element ref="xaip:evidenceRecord" maxOccurs="unbounded"
minOccurs="1" />
                  <element name="CredentialID" type="string"
maxOccurs="unbounded" minOccurs="1" />
            </choice>
      </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">
                              <sequence>
                                    <element ref="xaip:DXAIP"></element>
                              </sequence>
                        </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>
                                    <element name="AOID" type="string" />
                                    <element name="VersionID" type="string"
maxOccurs="unbounded" minOccurs="0"></element>
                              </sequence>
                        </extension>
                  </complexContent>
            </complexType>
      </element>

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

      <element name="ArchiveRetrievalResponse">
```

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



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


      <element name="ArchiveEvidenceRequest">
            <complexType>
                  <complexContent>
                        <extension base="tr:RequestType">
                              <sequence>
                                    <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="xaip:evidenceRecord"
maxOccurs="unbounded"
                              minOccurs="0">
                        </element>
                              </sequence>
                        </extension>
                  </complexContent>
            </complexType>
```

```
      </element>



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

      <element name="ArchiveDeletionRequest">
            <complexType>
                  <complexContent>
                        <extension base="tr:RequestType">
                              <sequence>
                                    <element name="AOID"
type="string"></element>
                              </sequence>
                        </extension>
                  </complexContent>
            </complexType>
      </element>

      <element name="ReasonOfDeletion">
            <complexType>
                  <sequence>
                        <element name="RequestorName"
type="saml:NameIDType" />
                        <element name="RequestInfo" type="string" />
                  </sequence>
            </complexType>
      </element>

      <element name="ArchiveDeletionResponse" type="tr:ResponseType"/>



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

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

```
                        </extension>
                  </complexContent>
            </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 name="XAIPData"
                                          maxOccurs="unbounded"
minOccurs="1">
                                          <complexType>
                                                <sequence>
                                                      <element
ref="dss:Result" maxOccurs="1" minOccurs="1" />
                                                      <element
ref="tr:DataLocation" />
                                                      <element name="Value"
type="anyType" maxOccurs="1" minOccurs="0" />
                                                </sequence>
                                          </complexType>
                                    </element>
                              </sequence>
                        </extension>
                  </complexContent>
            </complexType>
      </element>

</schema>
```

6.2 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/1.2"
      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
      xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
      xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema"
      xmlns:tr="http://www.bsi.bund.de/tr-esor/api/1.2"
      >

      <!--================================================================-->
      <!-- Version 1.2 of 31.01.2015
                 -->
      <!--================================================================-->


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

      <wsdl:types>
            <xsd:schema targetNamespace="http://www.bsi.bund.de/tr-
esor/api/1.2"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xaip="http://www.bsi.bund.de/tr-esor/xaip/1.2"
                  xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema"
                  elementFormDefault="qualified">
                  <xsd:include
schemaLocation="http://www.bsi.bund.de/SharedDocs/Download/tr-
esor/api/1_2/tr-esor-interfaces-v1_2_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>

    <!-- VerifyRequest -->

      <wsdl:message name="VerifyRequest">
            <wsdl:part name="parameters" element="dss:VerifyRequest" />
      </wsdl:message>
      <wsdl:message name="VerifyResponse">
            <wsdl:part name="parameters" element="dss:VerifyResponse"/>
      </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:operation name="Verify">
                  <wsdl:input message="tr:VerifyRequest" />
                  <wsdl:output message="tr:VerifyResponse" />
```

```
            </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:operation name="Verify">
                    <soap:operation
                        soapAction="http://www.bsi.bund.de/tr-esor/Verify"
/>
                    <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>
```