

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	1.2.2 (auf Basis der eIDAS-Verordnung und der ETSI Preservation Standards)
Datum	02.05.2019

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>

© Bundesamt für Sicherheit in der Informationstechnik 2019

Inhaltsverzeichnis

1. Einführung.....	5
2. Überblick.....	7
3. Funktionen der ArchiSafe-Schnittstelle (TR-ESOR-S.4).....	9
3.1 ArchiveSubmissionRequest und ArchiveSubmissionResponse.....	9
3.1.1 ArchiveSubmissionRequest.....	10
3.1.2 ArchiveSubmissionResponse.....	13
3.2 ArchiveUpdateRequest und ArchiveUpdateResponse.....	15
3.2.1 ArchiveUpdateRequest.....	15
3.2.2 ArchiveUpdateResponse.....	16
3.3 ArchiveRetrievalRequest und ArchiveRetrievalResponse.....	18
3.3.1 ArchiveRetrievalRequest.....	18
3.3.2 ArchiveRetrievalResponse.....	20
3.4 ArchiveEvidenceRequest und ArchiveEvidenceResponse.....	22
3.4.1 ArchiveEvidenceRequest.....	22
3.4.2 ArchiveEvidenceResponse.....	24
3.5 ArchiveDeletionRequest und ArchiveDeletionResponse.....	26
3.5.1 ArchiveDeletionRequest.....	26
3.5.2 ArchiveDeletionResponse.....	27
3.6 ArchiveDataRequest und ArchiveDataResponse.....	28
3.6.1 ArchiveDataRequest.....	29
3.6.2 ArchiveDataResponse.....	30
3.7 VerifyRequest und VerifyResponse.....	32
3.7.1 VerifyRequest.....	32
3.7.2 VerifyResponse.....	36
4. Funktionen der Preservation-API gemäß ETSI TS 119 512.....	38
4.1 Vergleich der ETSI TS 119 512 Preservation-API mit der TR-ESOR-S.4-Schnittstelle.....	38
5. Funktionen der internen Schnittstellen.....	39
5.1 TR-ESOR-S.1 (ArchiSafe-Modul – Krypto-Modul).....	39
5.1.1 Prüfung von digitalen Signaturen, beweisrelevanten Daten, Beweisdaten und Archivdatenobjekten.....	39
5.1.2 Anforderung einer digitalen Signatur.....	39
5.2 TR-ESOR-S.2 (ArchiSig-Modul – ECM-/Langzeitspeichersystem).....	41
5.2.1 Speichern eines Archivdatenobjektes.....	41
5.2.2 Ergänzen einer neuen Version eines Archivdatenobjektes.....	41
5.2.3 Auslesen von Archivdatenobjekten.....	41
5.3 TR-ESOR-S.3 (ArchiSig-Modul – Krypto-Modul).....	42
5.3.1 Anfordern eines (qualifizierten) Zeitstempels.....	42
5.3.2 Prüfen eines (qualifizierten) Zeitstempels.....	43
5.3.3 Berechnung eines Hashwertes.....	45
5.4 TR-ESOR-S.5 (ArchiSafe-Modul – ECM-Langzeitspeichersystem).....	47
5.4.1 Abfrage beweiswerterhaltend archivierter Daten.....	47
5.4.2 Löschen von Archivdatenobjekten.....	48
5.4.3 Abfrage diskreter Datenobjekte.....	48
5.5 TR-ESOR-S.6 (ArchiSafe-Modul – ArchiSig-Modul).....	48

5.5.1 Beweiswerterhaltende Archivierung elektronischer Daten.....	48
5.5.2 Ergänzen einer neuen Version eines Archivdatenobjektes.....	48
5.5.3 Rückgabe technischer Beweisdaten.....	48
6. Fehlercodes.....	49
7. Spezifikation einer Webservice-basierten Schnittstelle.....	51
7.1 Spezifikation der Aufruf- und Rückgabeparameter als XML-Schema.....	51
7.2 WSDL-Spezifikation der Schnittstelle TR-ESOR-S.4.....	57

Abbildungsverzeichnis

Abbildung 1: Schematische Darstellung der IT-Referenzarchitektur.....	6
Abbildung 2: Umsetzung der IT-Referenzarchitektur auf Basis des eCard-API-Frameworks.....	8

Tabellenverzeichnis

Tabelle 1: Vergleich ETSI TS 119 512 Preservation-API mit TR-ESOR-S.4-Schnittstelle.....	39
--	----

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 1 dargestellte IT-Referenzarchitektur orientiert sich an der ArchiSafe¹ 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>

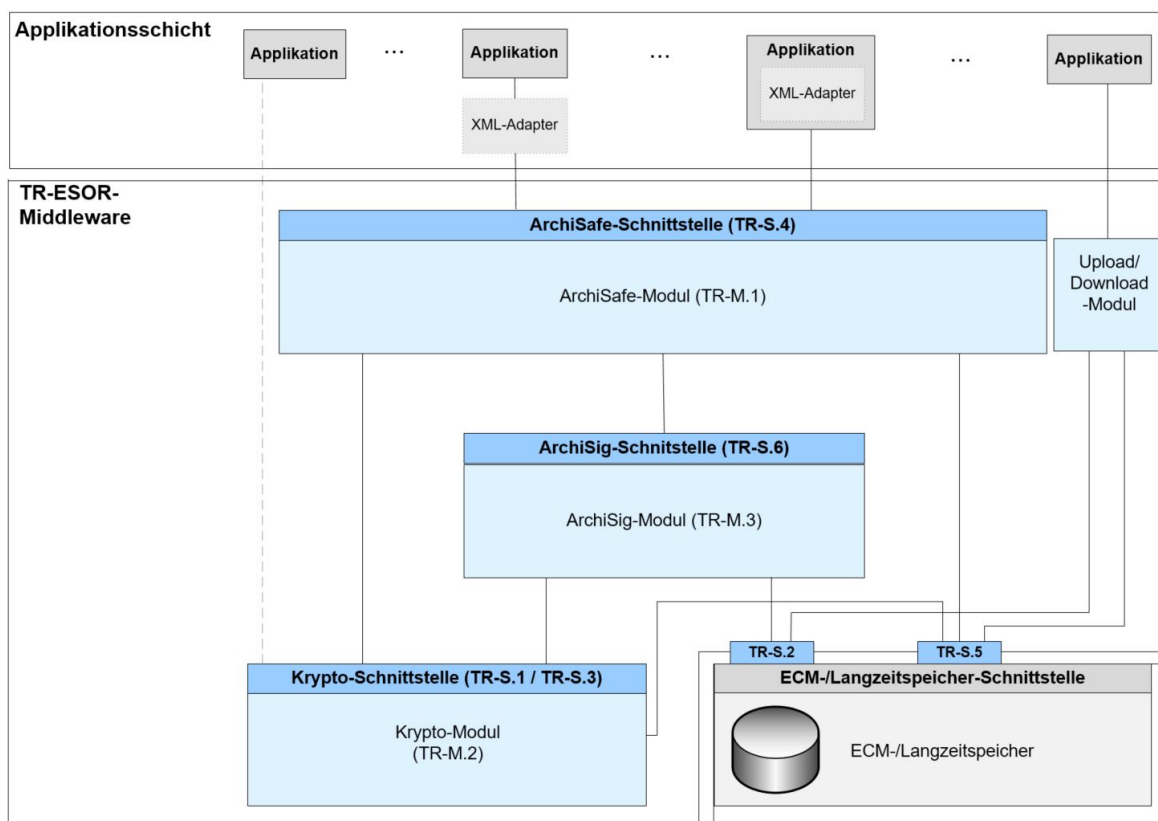


Abbildung 1: Schematische Darstellung der IT-Referenzarchitektur

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 TR-ESOR-spezifischen Schnittstellen auf Basis des in der BSI TR 03112 spezifizierten eCard-API-Frameworks.

2. Überblick

In der Schnittstelle TR-S.4 müssen die im Folgenden näher aufgeführten Funktionen mit den hier beschriebenen Parameterkonstellationen unterstützt werden:

- 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)

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

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

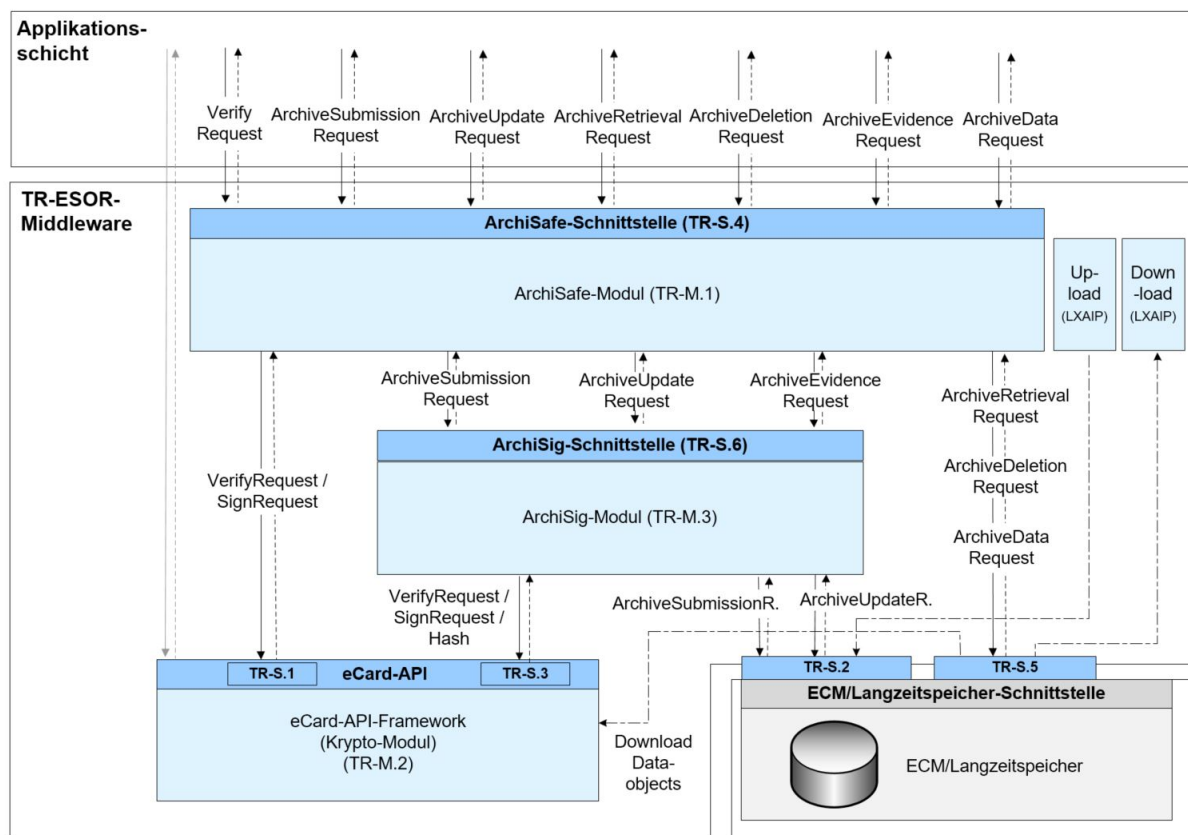


Abbildung 2: Umsetzung der IT-Referenzarchitektur auf Basis des eCard-API-Frameworks

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

1. die Schnittstellen des Krypto-Moduls gemäß des eCard-API-Frameworks (Technische Richtlinie des BSI TR 03112) realisiert und
2. auch die Schnittstellen des ArchiSafe-, ArchiSig-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.

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 Funktionen zur Beweiswerterhaltung kryptographisch signierter Dokumente. Hierbei werden die Funktionen der ArchiSafe-Schnittstelle (TR-S.4) in Abschnitt 3 spezifiziert. In Abschnitt 4 wird das Preservation-API von [TR 119 512] beschrieben und mit der TR-ESOR-S.4-Schnittstelle verglichen. In Abschnitt 5 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 6 sind die verwendeten Fehlercodes zusammengefasst und näher erläutert und in Abschnitt 7 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 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 7) gegeben.

3.1 ArchiveSubmissionRequest und ArchiveSubmissionResponse

Mit der Funktion `ArchiveSubmissionRequest` wird dem aufgerufenen Modul ein Archivdatenobjekt 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. Hierbei kann im `xaip:XAIP`-Element entweder ein physisches XAIP (siehe Abschnitt 3.1 in [TR-ESOR-F]) oder ein logisches XAIP (LXAIP) (siehe Abschnitt 3.2 in [TR-ESOR-F]) übergeben werden. Alternativ können im `ArchiveData`-Element binäre Nutzdaten übergeben werden. Hierbei wird der Typ des übergebenen Datenobjektes durch das `Type`-Attribut näher bestimmt. Dabei kann insbesondere ein base64Binary-codierter² ASiC-AIP-Container gemäß Abschnitt 3.3 in [TR-ESOR-F] mit einem `Type=http://uri.etsi.org/ades/ASiC/type/ASiC-ERS` Attribut übergeben werden.

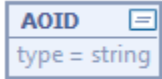
Wie in Abbildung 2 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in den Schnittstellen TR-S.2 (vgl. Abschnitt 5.2) und TR-S.6 (vgl. Abschnitt 5.5) genutzt.

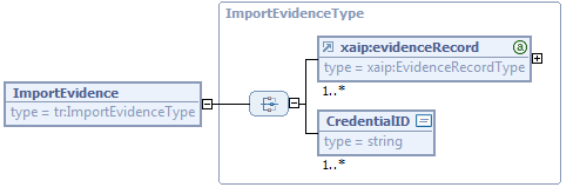
² Siehe <https://www.w3.org/TR/xmlschema-2/#base64Binary>.

3.1.1 ArchiveSubmissionRequest

Name	ArchiveSubmissionRequest		
Beschreibung	<p>Mit der Funktion <code>ArchiveSubmissionRequest</code> wird dem aufgerufenen Modul ein Archivdatenobjekt übergeben.</p> <p>Hierbei kann für eine effiziente Übertragung von großen Binärdaten der optimierte Nachrichtenübertragungsmechanismus „SOAP Message Transmission Optimization Mechanism (MTOM)“³ genutzt werden.</p>		
Aufruf	<p>Aufruf der <code>ArchiveSubmissionRequest</code>-Funktion</p> <table border="1"> <thead> <tr> <th>Name</th><th>Beschreibung</th></tr> </thead> </table>	Name	Beschreibung
Name	Beschreibung		

³ Siehe <https://www.w3.org/TR/soap12-mtom/>.

	dss:OptionalInputs	<p>Ist für optionale Eingabeelemente vorgesehen.</p> <p>(A3.1.1-1): Gemäß der vorliegenden Spezifikation <u>sollen</u> folgende Elemente unterstützt werden:</p> <ul style="list-style-type: none"> • AOID, • ReturnVerificationReport, • ImportEvidence. <p>Dabei gilt:</p> <ul style="list-style-type: none"> • AOID Durch die Übergabe eines AOID-Elementes <u>kann</u> die AOID von der aufrufenden Anwendung vergeben werden. Im Regelfall fehlt dieses Element und die AOID wird vom aufgerufenen Modul bereitgestellt.  <ul style="list-style-type: none"> • ReturnVerificationReport Durch die Übergabe eines ReturnVerificationReport-Elementes gemäß [OASIS VR] bzw. [eCard-2] <u>kann</u> 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.
--	--------------------	--

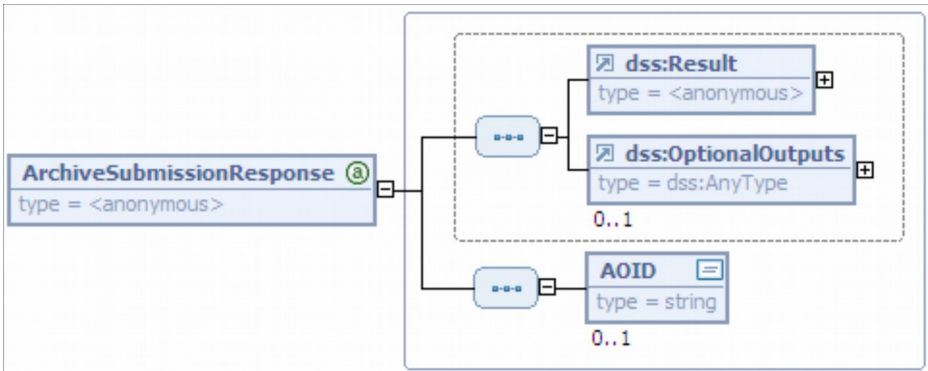
		<ul style="list-style-type: none"> • ImportEvidence <p>Mit der Übergabe des nachfolgend dargestellten ImportEvidence-Elementes <u>kann</u> der Import von einem oder mehreren zu einer bestimmten XAIP- bzw. LXAIP-Version bzw. zu den übergebenen Binärdaten gehörenden Evidence Records gemäß [RFC4998] oder [RFC6283]⁴ oder [TR-ESOR-ERS] angestoßen werden. Die Struktur des xaip:evidenceRecord-Elementes ist in [TR-ESOR-F] erläutert. Um Evidence Records für mehrere Versionen eines XAIPs oder LXAIPs importieren zu können, <u>kann</u> dieses Element mehrmals auftreten. Das xaip:evidenceRecord-Element <u>muss</u> hier die Attribute AOID und VersionID enthalten.</p> <p>Sofern die zu importierenden Evidence Records bereits im XAIP bzw. LXAIP enthalten sind, wird statt des Evidence Records hier die entsprechende CredentialID übergeben.</p>  <p>(A3.1.1-2): Im Zuge des Imports von Evidence Records <u>müssen</u> diese von der TR-ESOR-Middleware vollständig geprüft werden. Dies umfasst die im entsprechenden ERS-Standard vorgesehenen Prüfungsschritte⁵, wobei die jeweiligen Zertifikate der Zeitstempel vollständig bis hin zu einer vertrauenswürdigen Wurzel geprüft werden <u>müssen</u>.</p>
	xaip:XAIP	<p>Enthält ein XML-basiertes Archivdatenobjekt gemäß [TR-ESOR-F], das durch den Aufruf der beweiserhaltenden Archivierung zugeführt werden soll.</p> <p>Hierbei kann es sich entweder ein physisches XAIP (siehe Abschnitt 3.1 in [TR-ESOR-F]) oder ein logisches XAIP (LXAIP) (siehe Abschnitt 3.2 in [TR-ESOR-F]) handeln.</p>

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

⁵ Siehe Abschnitt 3.3 in [RFC4998] und Abschnitt 2.3 in [RFC6283] sowie [TR-ESOR-ERS].

	ArchiveData	<p>Enthält ein in einem beliebigen anderen Format vorliegendes Archivdatenobjekt. Der hierfür genutzte ArchiveDataType ist als anyType mit einem optionalen Type-Attribut definiert.</p> <p>Durch das Type-Attribut <code>http://uri.etsi.org/ades/ASiC/type/ASiC-ERS</code> wird klargestellt, dass es sich um einen base64Binary-codierten⁶ ASiC-AIP-Container gemäß Abschnitt 3.3 in [TR-ESOR-F] handelt.</p> <p>Durch das Type-Attribut <code>http://www.bsi.bund.de/tr-esor/api/1.2/type/binaryData</code> 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.</p> <p>Weitere Übergabetypen <u>können</u> im Rahmen einer Profilierung der vorliegenden Spezifikation spezifiziert werden.</p>
--	-------------	---

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	<div></div> <p>ArchiveSubmissionResponse ist die Antwort zum ArchiveSubmissionRequest-Aufruf</p> <table><tr><td>Name</td><td>Beschreibung</td></tr><tr><td>dss:Result</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></table>		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.
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.					

⁶ Siehe <https://www.w3.org/TR/xmlschema-2/#base64Binary>.

Name	ArchiveSubmissionResponse	
	dss:OptionalOutputs	<p>Ist für optionale Ausgabeelemente vorgesehen.</p> <p>(A3.1.2-1): Gemäß der vorliegenden Spezifikation <u>kann</u> das folgende Element auftreten:</p> <ul style="list-style-type: none"> VerificationReport gemäß [OASIS VR] bzw. [eCard-2] und [TR-ESOR-VR], der zurückgeliefert werden <u>muss</u>, 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.
	AOID	<p><u>Muss</u>, sofern die AOID⁷ vom aufgerufenen Modul erzeugt oder ergänzt wurde, vorhanden sein und für zukünftige Zugriffe auf das Archivdatenobjekt genutzt werden.</p>
	<div data-bbox="496 943 1425 1335"> <pre> xsd:sequence base="xs" name="dss:Result" type="anonymous"> xs:element base="dss" name="ResultMajor" type="anyURI" minOccurs="0" maxOccurs="1"/> xs:element base="dss" name="ResultMinor" type="anyURI" minOccurs="0" maxOccurs="1"/> xs:element base="dss" name="ResultMessage" type="InternationalStringType" minOccurs="0" maxOccurs="1"/> </pre> </div> <p>Statusinformationen und Fehler bei ArchiveSubmissionResponse (vgl. [eCard-1] Abschnitt 4.1 und 4.2).</p>	
	Name	Fehlercode
	ResultMajor	<ul style="list-style-type: none"> /resultmajor#ok /resultmajor#error /resultmajor#warning

⁷ Die AOID (Archive Object Identifier) im vorliegenden Dokument entspricht dem POID (Preservation Object Identifier) aus [ETSI TS 119 512].

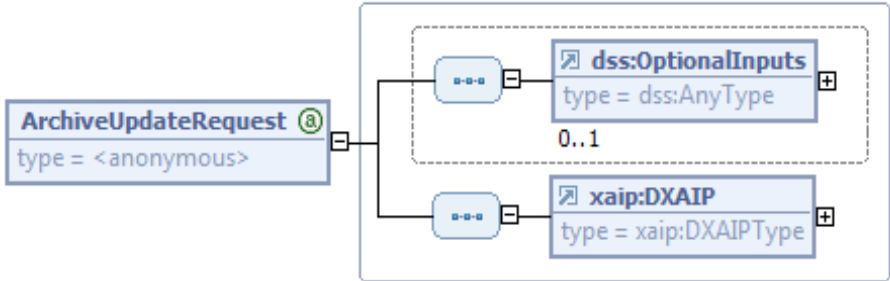
Name	ArchiveSubmissionResponse	
	ResultMinor	<ul style="list-style-type: none"> • /resultminor/al/common#noPermission • /resultminor/al/common#internalError • /resultminor/al/common#parameterError • /resultminor/arl/lowSpaceWarning • /resultminor/arl/noSpaceError • /resultminor/arl/existingAOID • /resultminor/arl/notSupported • /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

3.2 ArchiveUpdateRequest und ArchiveUpdateResponse

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

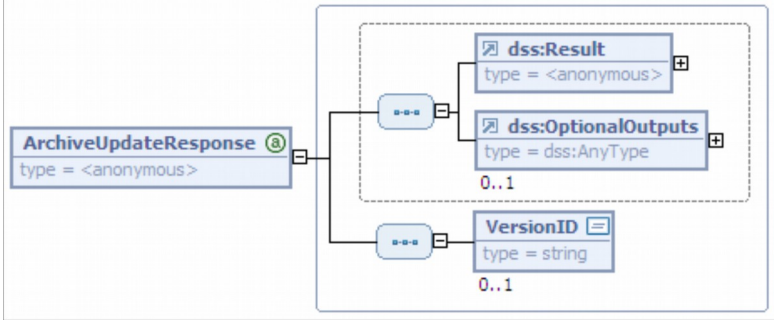
Wie in Abbildung 2 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in TR-S.2 (vgl. Abschnitt 5.2) und TR-S.6 (vgl. Abschnitt 5.5) genutzt.

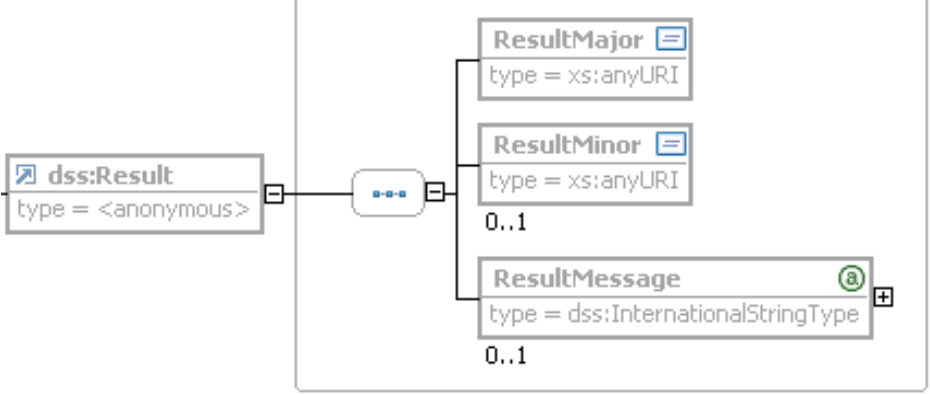
3.2.1 ArchiveUpdateRequest

Name	ArchiveUpdateRequest	
Beschreibung	Mit der Funktion <code>ArchiveUpdateRequest</code> wird eine neue Version für ein bereits abgelegtes Archivdatenobjekt erzeugt (vgl. [TR-ESOR-M.1]).	
	 <p>Aufruf der <code>ArchiveUpdateRequest</code>-Funktion</p>	
Name	Beschreibung	
dss:OptionalInputs	Ist für optionale Eingabeelemente vorgesehen. (A3.2.1-1): Gemäß der vorliegenden Spezifikation <u>sollen</u> hier die auf Seite 11 spezifizierten optionalen Eingabeelemente AOID, ReturnVerificationReport und ImportEvidence unterstützt werden.	

	xaip:DXAIP	Enthält ein ergänzendes XML-basiertes Archivdatenobjekt (Delta-XAIP) gemäß ([TR-ESOR-F], Kap.3.1.6) bzw. (Delta-LXAIP) gemäß ([TR-ESOR-F], Kap.3.2.2) das ein neues versionManifest, 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.
--	------------	--

3.2.2 ArchiveUpdateResponse

Name	ArchiveUpdateResponse	
Beschreibung	Als Antwort auf einen ArchiveUpdateRequest wird ein entsprechendes ArchiveUpdateResponse-Element zurückgeliefert, das im Erfolgsfall einen im Kontext einer AOID eindeutigen Identifikator der neuen Version des Archivdatenobjektes, die VersionID, enthält.	
Rückgabe	 <p>ArchiveUpdateResponse ist die Antwort zum ArchiveUpdateRequest-Aufruf</p>	
	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	<p>Ist für optionale Ausgabeelemente vorgesehen.</p> <p>(A3.2.2-1): Gemäß der vorliegenden Spezifikation <u>kann</u> das folgende Element auftreten:</p> <ul style="list-style-type: none"> VerificationReport gemäß [OASIS VR] bzw. [eCard-2] und [TR-ESOR-VR], der zurückgeliefert werden <u>muss</u>, 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.

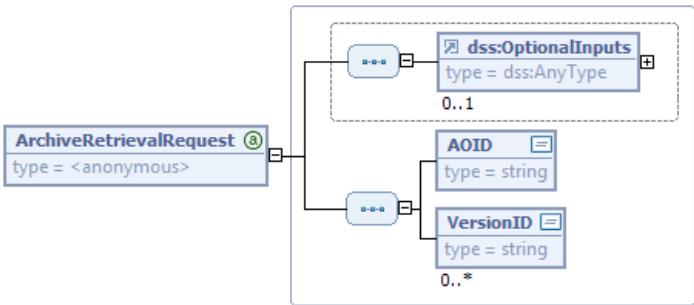
Name	ArchiveUpdateResponse	
	VersionID	Ist im Erfolgsfall vorhanden und enthält den bezüglich des über die AOID identifizierten Archivdatenobjektes eindeutigen Versions-Identifikator. Die VersionID soll in der Form v1, v2, ... vx gebildet werden.
	 <p>Statusinformationen und Fehler bei ArchiveUpdateResponse (vgl. [eCard-1] Abschnitt 4.1 und 4.2).</p>	
	Name	Fehlercode
	ResultMajor	<ul style="list-style-type: none"> • /resultmajor#ok • /resultmajor#error • /resultmajor#warning
	ResultMinor	<ul style="list-style-type: none"> • /resultminor/al/common#noPermission • /resultminor/al/common#internalError • /resultminor/al/common#parameterError • /resultminor/arl/lowSpaceWarning • /resultminor/arl/noSpaceError • /resultminor/arl/existingPackageInfoWarning • /resultminor/arl/notSupported • /resultminor/arl/DXAIP_NOK • /resultminor/arl/DXAIP_NOK_AOID • /resultminor/arl/DXAIP_NOK_EXPIRED • /resultminor/arl/DXAIP_NOK_SUBM-TIME • /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

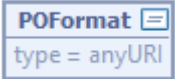
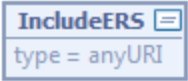
Mit der Funktion `ArchiveRetrievalRequest` wird das zu einer übergebenen `AOID` und `VersionID` gehörende physische XAIP-Archivdatenobjekt gemäß [TR-ESOR-F] (Abschnitt 3.1), das logische XAIP gemäß [TR-ESOR-F] (Abschnitt 3.2) oder das ASiC-AIP gemäß [TR-ESOR-F] (Abschnitt 3.3) über die TR-ESOR-Middleware aus dem ECM-/Langzeitspeichersystem ausgelesen.

Wie in Abbildung 2 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 in ähnlicher Weise auch in den Schnittstellen S.2 (vgl. Abschnitt 5.2) und S.5 (vgl. Abschnitt 5.4) genutzt.

3.3.1 ArchiveRetrievalRequest

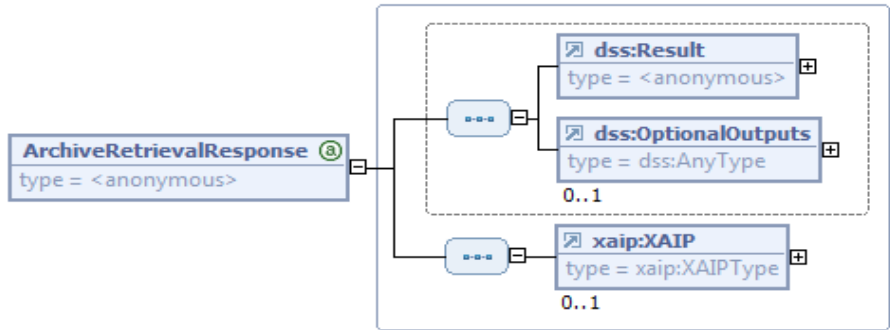
Name	ArchiveRetrievalRequest	
Beschreibung	<p>Mit der Funktion <code>ArchiveRetrievalRequest</code> wird ein im Langzeitspeicher abgelegtes Archivdatenobjekt ausgelesen und zurückgeliefert.</p> <p>Hierbei kann für eine effiziente Übertragung von großen Binärdaten der optimierte Nachrichtenübertragungsmechanismus „SOAP Message Transmission Optimization Mechanism (MTOM)“⁸ genutzt werden.</p>	
Beschreibung	 <p>Aufruf der <code>ArchiveRetrievalRequest</code>-Funktion</p>	
	Name	Beschreibung

⁸ Siehe <https://www.w3.org/TR/soap12-mtom/>.

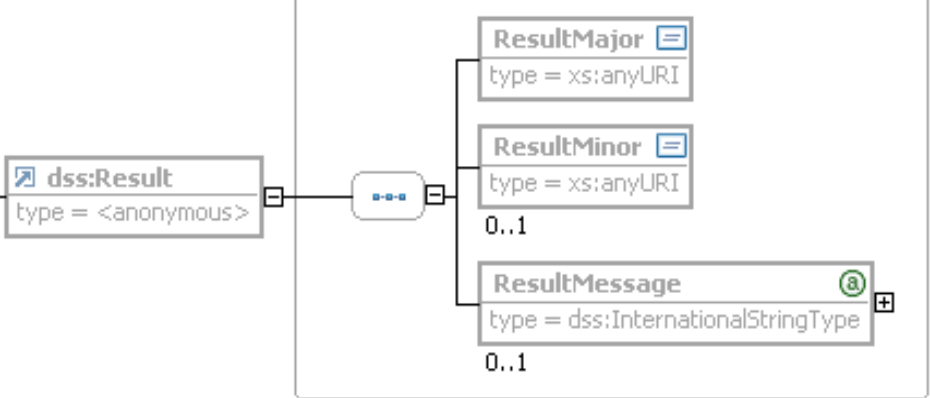
Name	ArchiveRetrievalRequest
	<p>dss:OptionalInputs</p> <p>Ist für optionale Eingabeelemente vorgesehen.</p> <p>(A3.3.1-1): Gemäß der vorliegenden Spezifikation <u>sollen</u> die folgenden optionalen Eingabeelemente unterstützt werden:</p> <ul style="list-style-type: none"> • POFormat • IncludeERS <p>POFormat⁹ – gibt das AIP-Format an, wobei folgende Formate definiert sind:</p>  <ul style="list-style-type: none"> • http://www.bsi.bund.de/tr-esor/xaip/1.2 – für ein XAIP gemäß Abschnitt 3.1 in [TR-ESOR-F], • http://www.bsi.bund.de/tr-esor/lxaip/1.3 – für ein „Logisches XAIP“ gemäß Abschnitt 3.2 in [TR-ESOR-F], • http://uri.etsi.org/ades/ASiC/type/ASiC-ERS für einen base64Binary-codierten ASiC-AIP-Container gemäß Abschnitt 3.3 in [TR-ESOR-F] in einem PO-Element gemäß [ETSI TS 119 512], das im dss:OptionalOutputs-Element des ArchiveRetrievalResponse zurückgeliefert wird. <p>Bei Nicht-Eingabe eines POFormats ist XAIP das Default-Format.</p> <p>IncludeERS – gibt an, dass das zurückgelieferte XAIP oder das logische XAIP (LXAIP) oder das ASiC-AIP den bzw. die entsprechenden Evidence Record(s) im angegebenen Format (vgl. ERSFormat, Seite 23) enthalten <u>soll</u>.</p>  <p>Dieser bzw. diese Evidence Record(s) wird bzw. werden bei XAIP bzw. LXAIP im dafür vorgesehenen xaip:credential/xaip:EvidenceRecord Element oder im Fall ASiC-AIP im ASiC-AIP-Container gemäß Abschnitt 3.3 in [TR-ESOR-F] zurückgeliefert.</p> <p>(A3.3.1-2): Das VersionID-Attribut des xaip:EvidenceRecord Elementes <u>muss</u> auf die entsprechende Version verweisen.</p> <p>Sofern das versionManifest nicht kryptographisch geschützt ist, <u>muss</u> mit einem unprotectedObjectPointer Element im entsprechenden versionManifest auf die credentialID des xaip:credential-Elementes verwiesen werden.</p>

Name	ArchiveRetrievalRequest	
		Umgekehrt <u>muss</u> 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	<p><u>Kann</u> eine Folge von Versions-Identifikatoren enthalten, durch die angegeben wird welche Versionen des Archivdatenobjektes XAIP bzw. LXAIP genau zurückgeliefert werden sollen.</p> <p>Sofern das VersionID-Element nicht angegeben ist, werden die zur letzten Version gehörigen Datenobjekte und Verwaltungsinformationen eines XAIPs bzw. LXAIPs zurückgeliefert.</p> <p>Durch die Angabe von all werden alle existierenden Versionen eines Archivdatenobjektes zurückgeliefert.</p>

3.3.2 ArchiveRetrievalResponse

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

⁹ Das POFormat-Element ist in [ETSI TS 119 512] folgendermaßen definiert:
 <element name="POFormat" type="anyURI" />

Name	ArchiveRetrievalResponse	
	dss:Result	<p>Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in [eCard-1] und weiter unten näher beschrieben.</p> <p>Sofern nur ein Teil der angeforderten Versionen des Archivdatenobjektes zurückgeliefert werden konnte, wird dies durch den Fehlercode .../resultminor/arl/requestOnlyPartlySuccessfulWarning angezeigt.</p>
	dss:OptionalOutputs	<p>Ist für optionale Ausgabeelemente vorgesehen, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden <u>sollen</u>.</p> <p>Insbesondere kann hier ein PO-Element gemäß [ETSI TS 119 512] enthalten sein, das ein base64Binary-codierten ASiC-AIP gemäß Abschnitt 3.3 in [TR-ESOR-F] enthält, sofern dieses angefordert wird.</p>
	xaip:XAIP	<p>Sofern kein Fehler aufgetreten ist, wird das angeforderte XML-basierte Archivdatenobjekt (XAIP oder LXAIP) gemäß [TR-ESOR-F] zurückgeliefert.</p>
	 <p>Statusinformationen und Fehler bei ArchiveRetrievalResponse (vgl. [eCard-1]).</p>	
Name	Fehlercode	
ResultMajor		<ul style="list-style-type: none"> • /resultmajor#ok • /resultmajor#error • /resultmajor#warning

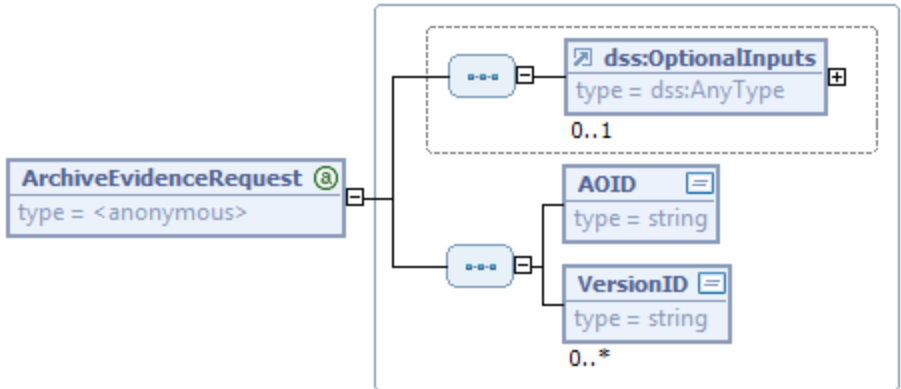
Name	ArchiveRetrievalResponse	
	ResultMinor	<ul style="list-style-type: none"> • /resultminor/al/common#noPermission • /resultminor/al/common#internalError • /resultminor/al/common#parameterError • /resultminor/ar/unknownAOID • /resultminor/ar/unsupported • /resultminor/ar/requestOnlyPartlySuccessfulWarning • /resultminor/ar/unknownVersionID • /resultminor/ar/unknownPOFormat
	ResultMessage	Beim Auftreten der Fehlermeldung .../unknown-VersionID soll die problematische VersionID hier zurückgeliefert werden.

3.4 ArchiveEvidenceRequest und ArchiveEvidenceResponse

Mit der Funktion `ArchiveEvidenceRequest` werden die zugehörigen technischen Beweisdaten (Evidence Records gemäß [RFC4998] oder [RFC6283]¹⁰ oder [RFC4998] mit der Profilierung aus [TR-ESOR-ERS]) für beweiswerterhaltend aufbewahrte und über AOID-Elemente adressierte Archivdatenobjekte (`xaip:XAIP`) zurückgeliefert.

Wie in Abbildung 2 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in TR-S.6 (vgl. Abschnitt 5.5) genutzt.

3.4.1 ArchiveEvidenceRequest

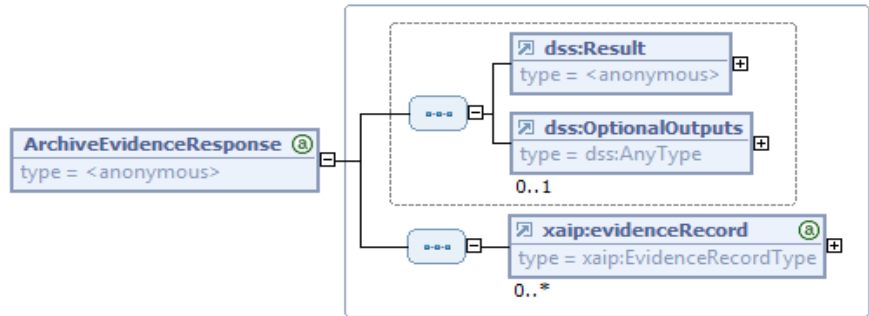
Name	ArchiveEvidenceRequest	
Beschreibung	Mit der Funktion <code>ArchiveEvidenceRequest</code> können für beweiswerterhaltend abgelegte Archivdatenobjekte technische Beweisdaten in Form von Evidence Records gemäß [RFC4998] oder [RFC6283] ¹¹ in der Profilierung gemäß [TR-ESOR-ERS] angefordert werden.	
Beschreibung	 <p>Aufruf der <code>ArchiveEvidenceRequest</code>-Funktion</p>	
	Name	Beschreibung

¹⁰ [RFC4998] muss, [RFC6283] kann unterstützt werden.

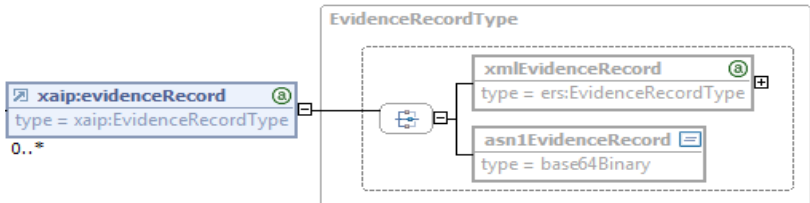
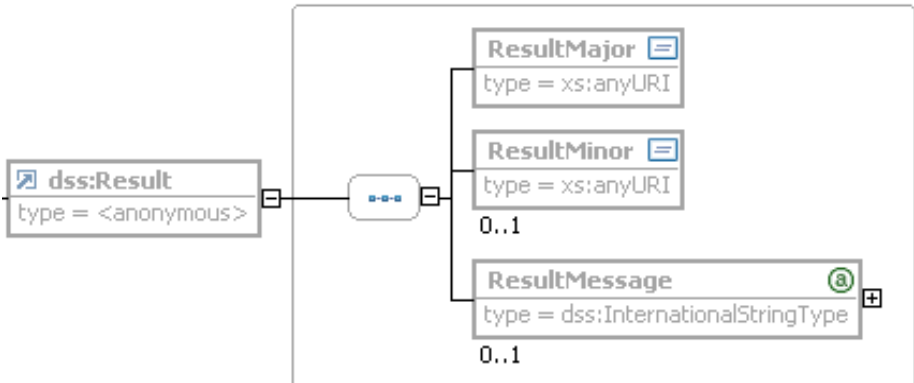
¹¹ [RFC4998] muss, [RFC6283] kann unterstützt werden.

Name	ArchiveEvidenceRequest	
	dss:OptionalInputs	<p>Ist für optionale Eingabeelemente vorgesehen. (A3.4.1-1): Gemäß der vorliegenden Spezifikation <u>soll</u> das folgende Element unterstützt werden:</p> <p>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:</p> <ul style="list-style-type: none"> • urn:ietf:rfc:4998 für ASN.1-basierte Evidence Records gemäß [RFC4998] oder • urn:ietf:rfc:6283 für XML-basierte Evidence Records gemäß [RFC6283]. <p>Fehlt das <code>ERSFormat</code>-Element, so werden ASN.1-basierte Evidence Records gemäß [RFC4998] in der Profilierung gemäß [TR-ESOR-ERS] zurückgeliefert.</p>
	AOID	Ist der eindeutige Identifikator des angeforderten Archivdatenobjektes.
	VersionID	<p><u>Kann</u> mehrfach auftreten und angeben für welche Versionen eines über die AOID identifizierten Archivdatenobjektes XAIP bzw. LXAIP Evidence Records zurückgeliefert werden sollen.</p> <p>Sofern das <code>VersionID</code>-Element nicht angegeben ist, wird der Beweisdatensatz für die aktuelle Version des XAIP bzw. des LXAIP zurückgeliefert.</p> <p>Durch die Angabe von <code>all</code> werden Evidence Records für alle existierenden Versionen eines Archivdatenobjektes zurückgeliefert.</p>

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/ar1/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.	
xaip:evidenceRecord	Sofern vom ArchiSig-Modul entsprechende Evidence Records ¹² gemäß [RFC4998] bzw. [RFC6283] konstruiert werden können, werden diese hier in der Profilierung gemäß [TR-ESOR-ERS] zurückgeliefert. Die detaillierte Struktur dieses Elementes ist nachfolgend erläutert.	

¹² Sofern die TR-ESOR-Middleware mehrere redundante Hashbäume pflegt, werden hier mehrere Evidence Records zurückgeliefert.

Name	ArchiveEvidenceResponse
	 <p>Das xaip:evidenceRecord-Element gemäß [TR-ESOR-F] ist vom Typ xaip:EvidenceRecordType, der als Erweiterung des ers:EvidenceRecordType aus [eCard-2] definiert ist und zusätzlich die Attribute AOID und VersionID, enthält, die in [TR-ESOR-F] näher erläutert sind.</p> <p>(A3.4.2-1): Bei der hier beschriebenen Verwendung von xaip:evidenceRecord <u>müssen</u> die Attribute AOID und VersionID gesetzt sein.</p>
Name	Beschreibung
xmlEvidenceRecord	Enthält einen XML-basierten Evidence Record gemäß [RFC6283].
asn1EvidenceRecord	Enthält einen ASN.1-basierten Evidence Record gemäß [RFC4998].
	 <p>Statusinformationen und Fehler bei ArchiveEvidenceResponse (vgl. [eCard-1]).</p>
Name	Fehlercode
ResultMajor	<ul style="list-style-type: none"> /resultmajor#ok /resultmajor#error /resultmajor#warning

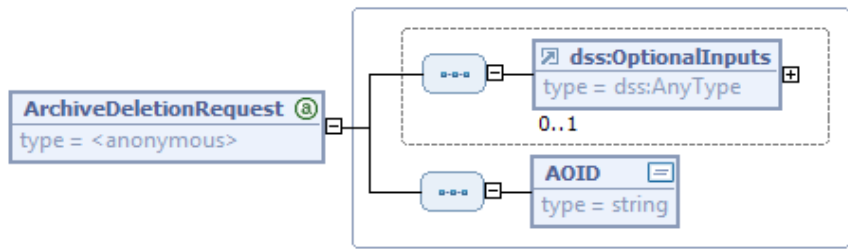
Name	ArchiveEvidenceResponse	
	ResultMinor	<ul style="list-style-type: none"> • /resultminor/al/common#noPermission • /resultminor/al/common#internalError • /resultminor/al/common#parameterError • /resultminor/arl/notSupported¹³ • /resultminor/arl/unknownAOID • /resultminor/arl/unknownVersionID/ • resultminor/arl/requestOnlyPartlySuccessfulWarning

3.5 ArchiveDeletionRequest und ArchiveDeletionResponse

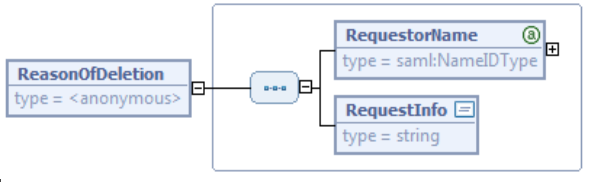
Mit der Funktion `ArchiveDeletionRequest` wird ein Archivdatenobjekt über die TR-ESOR-Middleware aus dem ECM-/Langzeitspeichersystem gelöscht.

Wie in Abbildung 2 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in der Schnittstelle TR-S.5 (vgl. Abschnitt 5.4) genutzt.

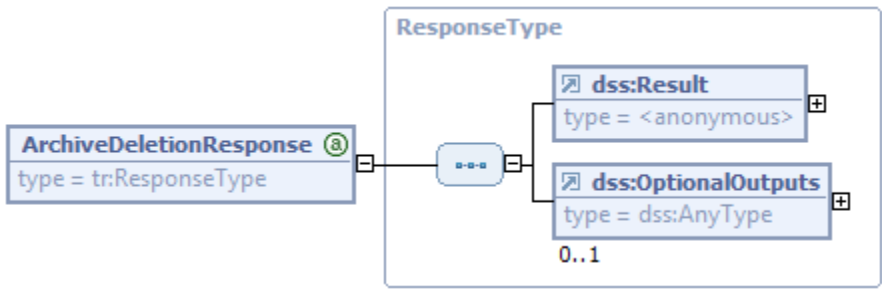
3.5.1 ArchiveDeletionRequest

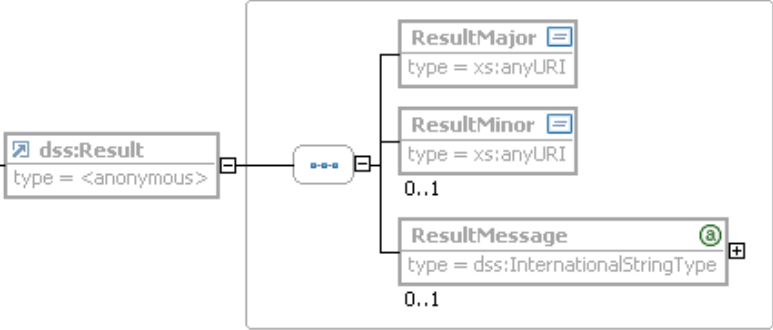
Name	ArchiveDeletionRequest	
Beschreibung	Mit der Funktion <code>ArchiveDeletionRequest</code> kann ein im Langzeitspeicher abgelegtes Archivdatenobjekt (XAIP oder LXAIP oder ASiC-AIP) gelöscht werden.	
Beschreibung	 <p>Aufruf der <code>ArchiveDeletionRequest</code>-Funktion</p>	
	Name	Beschreibung

¹³ Im `ResultMessage`-Element sollen nähere Informationen darüber zurückgeliefert werden, welche angeforderte Funktionalität nicht unterstützt wird.

Name	ArchiveDeletionRequest	
	dss:OptionalInputs	<p>Ist für optionale Eingabeelemente vorgesehen. Insbesondere bei einer vorzeitigen Löschung <u>muss</u> das folgende Element ReasonOfDeletion genutzt und unterstützt werden:</p> <p>(A3.5.1-1): Das ReasonOfDeletion-Element <u>muss</u> 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.</p> <p>(A3.5.1-2): Die gesamte Aktion einschließlich der Begründung <u>muss</u> protokolliert werden und der übergebene RequestorName <u>soll</u> mit den verwendeten Authentisierungsinformationen</p>  <p>abgeglichen werden.</p>
	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	 <p>ArchiveDeletionResponse ist die Antwort zum ArchiveDeletionRequest-Aufruf</p>	
	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.

Name	ArchiveDeletionResponse	
	dss:OptionalOutputs	Ist für optionale Ausgabeelemente vorgesehen und <u>kann</u> beispielsweise entsprechende Steuerelemente (responseControls) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden <u>sollen</u> .
	 <p>Statusinformationen und Fehler bei ArchiveDeletionResponse (vgl. [eCard-1]).</p>	
	Name	Fehlercode
	ResultMajor	<ul style="list-style-type: none"> • /resultmajor#ok • /resultmajor#error
	ResultMinor	<ul style="list-style-type: none"> • /resultminor/al/common#noPermission • /resultminor/al/common#internalError • /resultminor/al/common#parameterError • /resultminor/ar/unknownAOID • /resultminor/ar/notSupported • /resultminor/ar/missingReasonOfDeletion

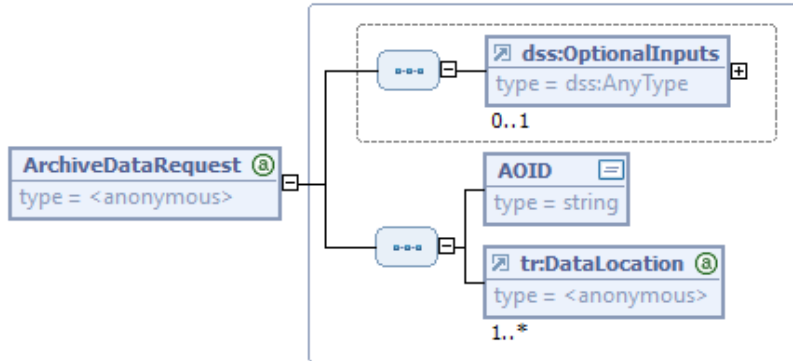
3.6 ArchiveDataRequest und ArchiveDataResponse

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

Die detaillierte Ausgestaltung dieser Funktion wird dem Hersteller überlassen. Der Hersteller ist zur Dokumentation der an der Schnittstelle unterstützten Funktionalität verpflichtet. Im Zuge der Zertifizierung wird geprüft, dass die in der Dokumentation beschriebene Funktionalität umgesetzt ist.

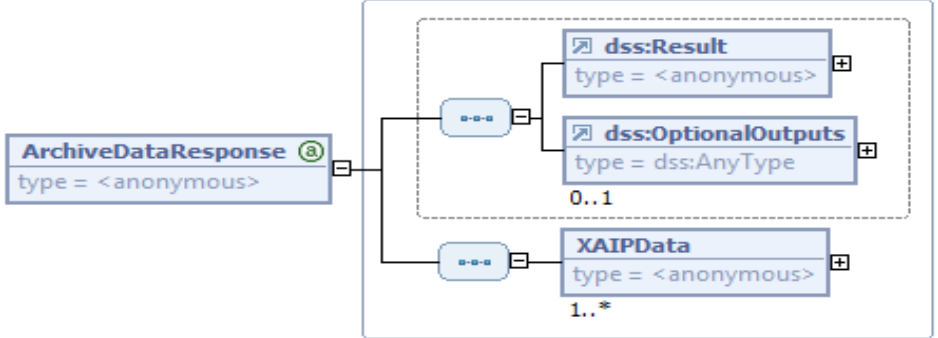
Wie in Abbildung 2 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in TR-S.5 (vgl. Abschnitt 5.4) genutzt.

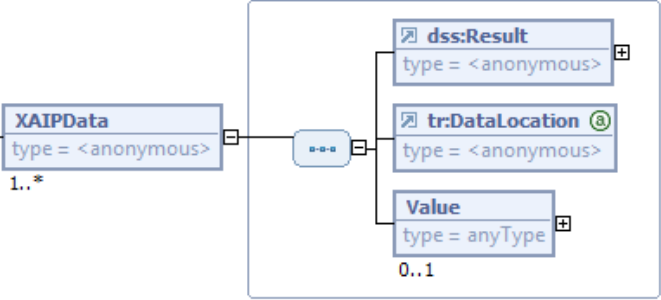
3.6.1 ArchiveDataRequest

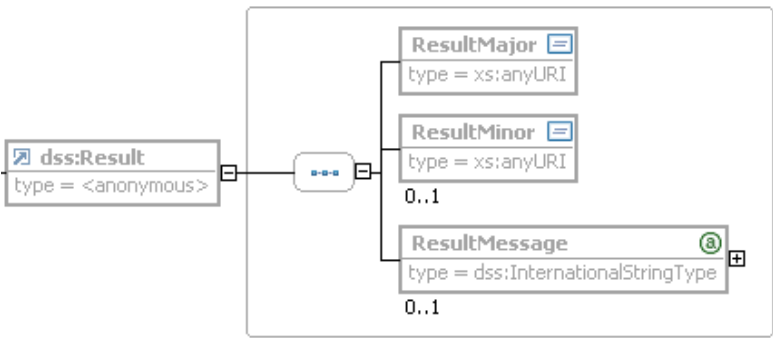
Name	ArchiveDataRequest									
Beschreibung	Mit der Funktion ArchiveDataRequest können diskrete Datenelemente aus einem im ECM-/Langzeitspeichersystem abgelegten, zumindest logisch im xaip:XAIP-Format gemäß [TR-ESOR-F] vorliegenden, Archivdatenobjekt ausgelesen werden.									
Beschreibung	<div></div> <p>Aufruf der ArchiveDataRequest-Funktion</p> <table><tr><th>Name</th><th>Beschreibung</th></tr><tr><td>dss:OptionalInputs</td><td>Ist für optionale Eingabeelemente vorgesehen und <u>kann</u> beispielsweise Steuerelemente (requestControls) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden <u>sollen</u>. Die vorliegende Spezifikation definiert keine solchen optionalen Eingabeelemente.</td></tr><tr><td>AOID</td><td>Dieses Element enthält den Identifikator eines bestimmten Archivdatenobjektes.</td></tr><tr><td>tr:DataLocation</td><td>Das tr:DataLocation-Element kann mehrmals auftreten und bestimmt die „Lokation“ der auszulesenden diskreten Datenelemente bezüglich eines zumindest logisch im xaip:XAIP-Format gemäß [TR-ESOR-F] vorliegenden Archivdatenobjektes. Die detaillierte Ausgestaltung der hier unterstützen Funktionalität bleibt dem Hersteller überlassen. Sofern der ArchiveDataRequest unterstützt wird, <u>muss</u> dieser die Details der an der Schnittstelle angebotenen Funktionalität dokumentieren.</td></tr></table>		Name	Beschreibung	dss:OptionalInputs	Ist für optionale Eingabeelemente vorgesehen und <u>kann</u> beispielsweise Steuerelemente (requestControls) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden <u>sollen</u> . Die vorliegende Spezifikation definiert keine solchen optionalen Eingabeelemente.	AOID	Dieses Element enthält den Identifikator eines bestimmten Archivdatenobjektes.	tr:DataLocation	Das tr:DataLocation-Element kann mehrmals auftreten und bestimmt die „Lokation“ der auszulesenden diskreten Datenelemente bezüglich eines zumindest logisch im xaip:XAIP-Format gemäß [TR-ESOR-F] vorliegenden Archivdatenobjektes. Die detaillierte Ausgestaltung der hier unterstützen Funktionalität bleibt dem Hersteller überlassen. Sofern der ArchiveDataRequest unterstützt wird, <u>muss</u> dieser die Details der an der Schnittstelle angebotenen Funktionalität dokumentieren.
Name	Beschreibung									
dss:OptionalInputs	Ist für optionale Eingabeelemente vorgesehen und <u>kann</u> beispielsweise Steuerelemente (requestControls) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden <u>sollen</u> . Die vorliegende Spezifikation definiert keine solchen optionalen Eingabeelemente.									
AOID	Dieses Element enthält den Identifikator eines bestimmten Archivdatenobjektes.									
tr:DataLocation	Das tr:DataLocation-Element kann mehrmals auftreten und bestimmt die „Lokation“ der auszulesenden diskreten Datenelemente bezüglich eines zumindest logisch im xaip:XAIP-Format gemäß [TR-ESOR-F] vorliegenden Archivdatenobjektes. Die detaillierte Ausgestaltung der hier unterstützen Funktionalität bleibt dem Hersteller überlassen. Sofern der ArchiveDataRequest unterstützt wird, <u>muss</u> dieser die Details der an der Schnittstelle angebotenen Funktionalität dokumentieren.									
	Name	Beschreibung								
	dss:OptionalInputs	Ist für optionale Eingabeelemente vorgesehen und <u>kann</u> beispielsweise Steuerelemente (requestControls) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden <u>sollen</u> . Die vorliegende Spezifikation definiert keine solchen optionalen Eingabeelemente.								
	AOID	Dieses Element enthält den Identifikator eines bestimmten Archivdatenobjektes.								
	tr:DataLocation	Das tr:DataLocation-Element kann mehrmals auftreten und bestimmt die „Lokation“ der auszulesenden diskreten Datenelemente bezüglich eines zumindest logisch im xaip:XAIP-Format gemäß [TR-ESOR-F] vorliegenden Archivdatenobjektes. Die detaillierte Ausgestaltung der hier unterstützen Funktionalität bleibt dem Hersteller überlassen. Sofern der ArchiveDataRequest unterstützt wird, <u>muss</u> dieser die Details der an der Schnittstelle angebotenen Funktionalität dokumentieren.								

Name	ArchiveDataRequest
	<p>Das DataLocation-Element spezifiziert, welche Teile eines Archivobjektes zurückgeliefert werden sollen und ist folgendermaßen definiert:</p> <pre> <element name="DataLocation"> <complexType> <complexContent> <extension base="anyType"> <attribute name="Type" type="anyURI"/> </extension> </complexContent> </complexType> </element> </pre> <p>Im Type-Attribut wird angegeben, welche Transformation für den Zugriff auf die gewünschten Daten angewandt werden soll, wobei die folgenden URIs vorgesehen sind:</p> <ul style="list-style-type: none"> • 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	 <p>ArchiveDataResponse ist die Antwort zum ArchiveDataRequest-Aufruf</p> <table border="1"> <thead> <tr> <th>Name</th><th>Beschreibung</th></tr> </thead> <tbody> <tr> <td>dss:Result</td><td>Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in [eCard-1] und unten näher beschrieben. Sofern nur ein Teil der angefragten diskreten Datenobjekte zurückgeliefert werden konnte, wird dies durch den Fehlercode .../resultminor/arl/requestOnlyPartlySuccessfulWarning angezeigt.</td></tr> </tbody> </table>	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.
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.				

Name	ArchiveDataResponse	
	dss:OptionalOutputs	Ist für optionale Ausgabeelemente vorgesehen und <u>kann</u> beispielsweise entsprechende Steuerelemente (responseControls) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden <u>sollen</u> .
	XAIPData	Enthält im Erfolgsfall die gewünschten Daten und die „Lokation“, aus der diese aus der im ECM-/Langzeitspeichersystem zumindest logisch existierenden XAIP- bzw. LXAIP-Struktur ausgelesen wurden. Die detaillierte Struktur dieses Elementes ist nachfolgend dargestellt und erläutert.
 <p>Das XAIPData-Element enthält im Erfolgsfall die gewünschten Daten.</p>		
Name		Beschreibung
	dss:Result	<p>Gibt an, ob die Anfrage erfolgreich durchgeführt werden konnte oder nicht.</p> <p>Als ResultMajor sind die beiden folgenden Werte möglich:</p> <ul style="list-style-type: none"> • .../resultmajor#ok • .../resultmajor#error <p>Als ResultMinor sind die folgenden Werte möglich:</p> <ul style="list-style-type: none"> • .../resultminor/ar/unknownLocation • .../resultminor/al/common#parameterError • .../resultminor/al/common#internalError
	tr:DataLocation	Das DataLocation-Element spezifiziert, welche Teile eines Archivobjektes zurückgeliefert werden. Die detaillierte Ausgestaltung dieses Parameters ist dem Hersteller überlassen. Siehe auch oben (Seite 30).
	Value	Enthält im Erfolgsfall die gewünschten Daten.

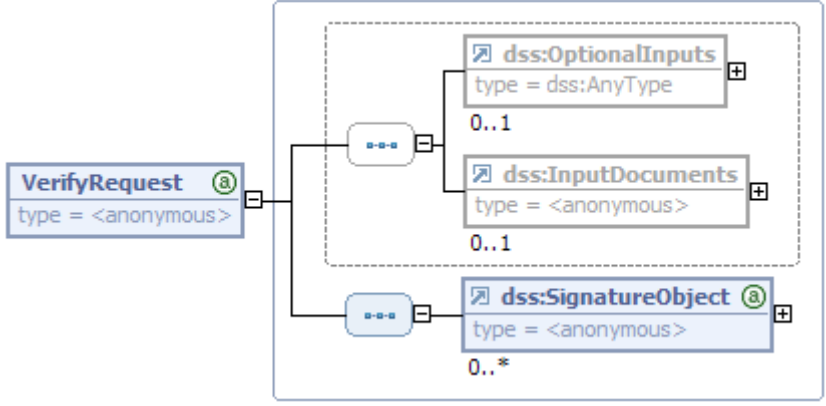
Name	ArchiveDataResponse	
	 <p>Statusinformationen und Fehler bei ArchiveDataResponse (vgl. [eCard-1]).</p>	
Name	Fehlercode	
ResultMajor	<ul style="list-style-type: none"> • /resultmajor#ok • /resultmajor#error • /resultmajor#warning 	
ResultMinor	<ul style="list-style-type: none"> • /resultminor/al/common#noPermission • /resultminor/al/common#internalError • /resultminor/al/common#parameterError • /resultminor/arl/unknownAOID • /resultminor/arl/notSupported • /resultminor/arl/requestOnlyPartlySuccessfulWarning 	

3.7 VerifyRequest und VerifyResponse

3.7.1 VerifyRequest

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

Wie in Abbildung 2 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in TR-S.1 (vgl. Abschnitt 5.1) genutzt.

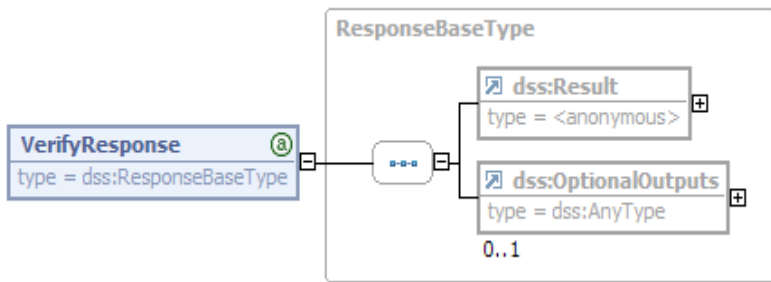
Name	VerifyRequest	
Beschreibung	Mit der Funktion <code>VerifyRequest</code> (vgl. Abschnitt 3.2.2 von [eCard-2]) werden XML-basierte Archivdatenobjekte (XAIP), logische XAIP oder ASiC-AIP-basierte Datencontainer samt der darin enthaltenen oder zusätzlich übergebenen beweisrelevanten Daten (Signaturen, Siegel, Zeitstempel, Zertifikate, Sperrlisten, OCSP-Responses etc.) und Beweisdaten (Evidence Records), geprüft.	
Aufrufparameter	 <p>Aufruf der <code>VerifyRequest</code>-Funktion.</p>	
	Name	Beschreibung

Name	VerifyRequest
	<p>dss:OptionalInputs</p> <p>Das OptionalInputs-Element <u>kann</u> zusätzliche Eingabeelemente enthalten.</p> <p>(A3.7.1-1): Hierbei <u>sollen</u> insbesondere die in [eCard-2] definierten Elemente und Aufrufoptionen unterstützt werden.</p> <p>Dies umfasst insbesondere die folgenden Elemente:</p> <ul style="list-style-type: none"> • VerifyUnderSignaturePolicy, • ReturnVerificationReport <p>Es gilt im Einzelnen:</p> <ul style="list-style-type: none"> • VerifyUnderSignaturePolicy <p>Sofern in einem dss:Document/InlineXML-Kindelement von dss:InputDocuments ein XAIP-Element in Form eines gewöhnlichen XAIP oder eines logischen XAIP gemäß [TR-ESOR-F] enthalten ist, kann mit dem Element VerifyUnderSignaturePolicy und der im DefaultPolicy/SignaturePolicyIdentifier-Element angegebenen Signature-Policy http://www.bsi.bund.de/tr-esor/sigpolicy/verify-xaip die Prüfung und Ergänzung aller im übergebenen XAIP- bzw. LXAIP-Container bzw. ASiC-AIP enthaltenen digitalen Signaturen angefordert werden.</p> <p>(A3.7.1-2): Hierbei <u>müssen</u> alle digitalen Signaturinformationen (Signaturen, Siegel, Zeitstempel, Zertifikate, Sperrlisten, OCSP-Responses etc.) bis hin zu einer vertrauenswürdigen Wurzel geprüft werden.</p> <p>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.</p> <p>Wenn sowohl die Signature-Policy</p>

Name	VerifyRequest
	<p>http://www.bsi.bund.de/tr-esor/sigpolicy/verify-xaip als auch das Element ReturnVerificationReport übergeben wird, dann <u>muss</u> der dann erzeugte Prüfbericht in das Kind-Element vr:VerificationReport des credential- Elements abgelegt werden.</p> <p>(A3.7.1-3): Sofern in der <code>credentialSection</code> des übergebenen XAIP-, LXAIP- oder ASiC-AIP-Containers ein oder mehrere <code>xaip:EvidenceRecord</code>-Elemente gemäß [TR-ESOR-F] enthalten sind, <u>müssen</u> diese entsprechend geprüft werden.</p> <ul style="list-style-type: none"> • ReturnVerificationReport Durch die Übergabe eines <code>ReturnVerificationReport</code>-Elementes gemäß [OASIS VR] bzw. [eCard-2] und [TR-ESOR-VR] <u>kann</u> ein ausführlicher Prüfbericht in Form eines <code>VerificationReport</code>-Elementes für die übergebenen Objekte (Signaturen, Siegel, Zeitstempel, Zertifikate, Sperrinformationen, Evidence Records, XAIP, LXAIP, ASiC-AIP mit den vorgenannten Daten) angefordert werden. Wenn nur das Element ReturnVerificationReport übergeben wird ohne Angabe der Signature-Policy, dann <u>ist</u> im Rahmen des <code>VerifyResponse</code> nur das erzeugte <code>VerificationReport</code>-Element zurück zu geben.
	<p><code>dss:InputDocuments</code> Das <code>dss:InputDocuments</code>-Element enthält die zur Prüfung benötigten Dokumente, sofern diese nicht bereits im unten erläuterten <code>SignatureObject</code>-Element enthalten sind.</p> <p>Außerdem <u>kann</u> in einem <code>dss:Document/InlineXML</code>-Kindelement ein XAIP-Element mit einem XAIP gemäß [TR-ESOR-F] (Abschnitt 3.1) oder einem LXAIP-Element gemäß [TR-ESOR-F] (Abschnitt 3.2) bzw. in einem <code>dss:Document/dss:Base64Data</code>-Kindelement ein ASiC-AIP gemäß [TR-ESOR-F] (Abschnitt 3.3) 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.</p>

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

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	 <p>Rückgabe der VerifyRequest-Funktion</p>	
	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	<p>Sofern ein VerificationReport angefordert wurde oder ein Fehler aufgetreten ist, enthält dieses Element den Prüfbericht in Form eines VerificationReport-Elementes oder das um diese Prüfinformationen ergänzte Archivdatenobjekt in Form eines xaip:XAIP-Elements.</p> <p>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.</p> <p>Details zur Ablage dieser Prüfinformationen im (L)XAIP-Container finden sich in [TR-ESOR-F].</p>

4. Funktionen der Preservation-API gemäß ETSI TS 119 512

Neben der in Abschnitt 3 spezifizierten TR-ESOR-S.4 Schnittstelle steht mit der „Preservation-API“ aus [ETSI TS 119 512] eine funktional weitgehend äquivalente, aber in Kürze international standardisierte Alternative zur Verfügung, die zusätzlich oder anstatt der TR-ESOR-S.4-Schnittstelle als Eingangsschnittstelle zur TR-ESOR-Middleware genutzt werden kann.

Für den Einsatz der „Preservation-API“ gemäß [ETSI TS 119 512] im Rahmen der vorliegenden Technischen Richtlinie werden im Rahmen von TR-ESOR die folgenden Mindestanforderungen definiert:

- RetrieveInfo gemäß Abschnitt 5.3.2 von [ETSI TS 119 512] muss unterstützt werden. Hierbei muss zumindest ein Bewahrungsprofil unterstützt werden, welches das Bewahrungsschema <http://uri.etsi.org/19512/scheme/pds+pgd+aug+wst+ers> gemäß Annex F.1 von [ETSI TS 119 512] umsetzt.
- PreservePO gemäß Abschnitt 5.3.3 von [ETSI TS 119 512] muss unterstützt werden, wobei zumindest eines der in [TR-ESOR-F] definierten Archivdatenobjekt-Formate (XAIP, LXAIP oder ASiC-AIP) unterstützt werden muss.
- RetrievePO gemäß Abschnitt 5.3.4 von [ETSI TS 119 512] muss unterstützt werden, wobei zumindest eines der in [TR-ESOR-F] definierten Archivdatenobjekt-Formate (XAIP, LXAIP oder ASiC-AIP) sowie Evidence Records gemäß [RFC4998] in der Profilierung gemäß [TR-ESOR-ERS] unterstützt werden müssen.
- DeletePO gemäß Abschnitt 5.3.5 von [ETSI TS 119 512] muss unterstützt werden.
- UpdatePOC gemäß Abschnitt 5.3.6 von [ETSI TS 119 512] muss unterstützt werden.
- RetrieveTrace gemäß Abschnitt 5.3.7 von [ETSI TS 119 512] kann unterstützt werden.
- ValidateEvidence gemäß Abschnitt 5.3.8 von [ETSI TS 119 512] soll unterstützt werden. Sofern diese Operation unterstützt wird, muss zumindest die Validierung von Evidence Records gemäß [RFC4998] in der Profilierung gemäß [TR-ESOR-ERS] **sowie** die Validierung der in [TR-ESOR-F] definierten Archivdatenobjekt-Formate (XAIP, LXAIP **oder** ASiC-AIP) unterstützt werden. Darüber hinaus kann die Validierung von Evidence Records gemäß [RFC6283] unterstützt werden.
- Search gemäß Abschnitt 5.3.7 von [ETSI TS 119 512] kann unterstützt werden.

4.1 Vergleich der ETSI TS 119 512 Preservation-API mit der TR-ESOR-S.4-Schnittstelle

Hierbei entspricht die Preservation-API gemäß [ETSI TS 119 512] der Eingangs-Schnittstelle S.4 zur TR-ESOR-Middleware [TR-ESOR-F], wie in der folgenden Tabelle dargestellt.

ETSI TS 119 512	Verbindlich -keitsgrad	TR-ESOR V1.2 ff	Verbindlich- keitsgrad
PreservePO	mandatory	ArchiveSubmissionRequest	mandatory
DeletePO	mandatory	ArchiveDeletionRequest	mandatory
RetrievePO	mandatory	ArchiveEvidenceRequest	Mandatory
RetrievePO	mandatory	ArchiveRetrievalRequest	mandatory
UpdatePOC (optional)	optional	ArchiveUpdateRequest	optional
Validate Evidence	optional	VerifyRequest	optional
RetrieveInfo	mandatory		
RetrieveTrace	optional		
Search	optional	ArchiveDataRequest	optional

Tabelle 1: Vergleich ETSI TS 119 512 Preservation-API mit TR-ESOR-S.4-Schnittstelle

In TR-ESOR V1.3 wird die Transformation von der ETSI TS 119 512 Preservation-API mit der TR-ESOR-S.4-Schnittstelle im Detail weiter ausgearbeitet.

5. Funktionen der internen 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 2) erläutert:

- TR-S.1: TR-ESOR-S.1 (ArchiSafe-Modul – Krypto-Modul) (siehe Abschnitt 5.1)
- TR-S.2: TR-ESOR-S.2 (ArchiSig-Modul – ECM-/Langzeitspeichersystem) (siehe Abschnitt 5.2)
- TR-S.3: TR-ESOR-S.3 (ArchiSig-Modul – Krypto-Modul) (siehe Abschnitt 5.3)
- TR-S.5: TR-ESOR-S.5 (ArchiSafe-Modul – ECM-/Langzeitspeichersystem) (siehe Abschnitt 5.4)
- TR-S.6: TR-ESOR-S.6 (ArchiSafe-Modul – ArchiSig-Modul) (siehe Abschnitt 5.5)

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

Dieser Abschnitt beschreibt, wie die Abbildung 2 dargestellte Schnittstelle TR-S.1 auf Basis des eCard-API-Frameworks ([BSI TR 03112]) umgesetzt werden kann.

Diese 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`)

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

Für die Prüfung von digitalen Signaturen, beweisrelevanten Daten (Zertifikaten, Zertifikatsstatusinformationen, Zeitstempeln, etc.), Beweisdaten (Evidence Records) und Archivdatenobjekten (XAIPs bzw. LXAIps bzw. ASiC-AIPs) ist in [OASIS-DSS] und [eCard-2] der Funktionsaufruf `VerifyRequest` und die zugehörige Antwort `VerifyResponse` definiert. Entsprechende Korrekturen und Ergänzungen sind darüber hinaus in [TR-ESOR-VR] bzw. 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 (z.B. OCSP-Antworten oder Sperrlisten) müssen von den Vertrauensdiensteanbietern abgerufen werden.

5.1.2 Anforderung einer digitalen Signatur

Für die Anforderung einer digitalen Signatur ist in [OASIS-DSS] und [eCard-2] die Funktion `SignRequest` und die zugehörige Antwort `SignResponse` definiert.

5.1.2.1 SignRequest (Anforderung einer digitalen Signatur)

Ein `SignRequest` im Kontext der Schnittstelle S.1 übergibt ein Archivdatenobjekt (XAIP- bzw. LXAIp- bzw. ASiC-AIP-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 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	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].	

5.1.2.2 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	<p>SignResponse ist die Antwort zum SignRequest-Aufruf</p> <table><tr><th>Name</th><th>Beschreibung</th></tr><tr><td>dss:Result</td><td>Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements und die möglichen Fehlercodes sind in Abschnitt 4.1.2 von [eCard-1] und in Abschnitt 3.2.1 von [eCard-2] beschrieben.</td></tr></table>		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	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	<u>Kann</u> 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	<u>Kann</u> 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 oben genannten DocumentWithSignature-Element vorhanden ist, wird kein dss:SignatureObject-Element zurückgeliefert.

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

Dieser Abschnitt beschreibt in den folgenden Unterkapiteln, wie die in Abbildung 2 dargestellte 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)

Neben der Umsetzung der Funktion „ArchivSubmission-Request/-Response“ zum Speichern eines Archivdatenobjektes“ (Upload) auf Basis der, auch dem eCard-API-Frameworks ([BSI TR 03112]) zu Grunde liegenden, Basistypen aus [OASIS-DSS] kann diese Funktion auch anders technisch umgesetzt werden, um den Upload von Datenobjekten im Rahmen eines logischen XAIP (LXAIP) gemäß ([TR-ESOR-F], Kap. 3.2) technisch performant zu ermöglichen. Dabei sind die Anforderungen gemäß ([TR-ESOR], Kap. 7.2 und 7.4.4) zu erfüllen.

Laut [ETSI TS 119 511] gilt: OVR-7.8-02 [WST] The preservation service shall be integrated in the IT environment implemented in such a way that all storage access by the preservation client changing the content of the storage shall only be done by the preservation service.

Daher ist es erforderlich, dass die eigentliche „Upload-Komponente“ ein (eigenständiges) Modul des TR-ESOR-Bewahrungsdienstes darstellt und logisch als Teil des TR-ESOR-Systems zu betrachten ist.

5.2.1 Speichern eines Archivdatenobjektes

Für das Speichern eines Archivdatenobjektes ist in Abbildung 2 der Funktionsaufruf ArchiveSubmissionRequest und die zugehörige Antwort ArchiveSubmissionResponse gemäß Abschnitt 3.1 vorgesehen.

5.2.2 Ergänzen einer neuen Version eines Archivdatenobjektes

Für das Ergänzen einer neuen Version eines Archivdatenobjektes ist in Abbildung 2 der Funktionsaufruf ArchiveUpdateRequest und die zugehörige Antwort ArchiveUpdateResponse gemäß Abschnitt 3.2 vorgesehen.

5.2.3 Auslesen von Archivdatenobjekten

Für das Auslesen von Archivdatenobjekten ist in Abbildung 2 der Funktionsaufruf `ArchiveRetrievalRequest` und `ArchiveRetrievalResponse` gemäß Abschnitt 3.3 vorgesehen.

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

Dieser Abschnitt beschreibt, wie die in Abbildung 2 dargestellte Schnittstelle TR-S.3 auf Basis des eCard-API-Frameworks (BSI TR 03112) umgesetzt werden kann.

Die Schnittstelle TR-S.3 umfasst drei wesentliche Funktionen:

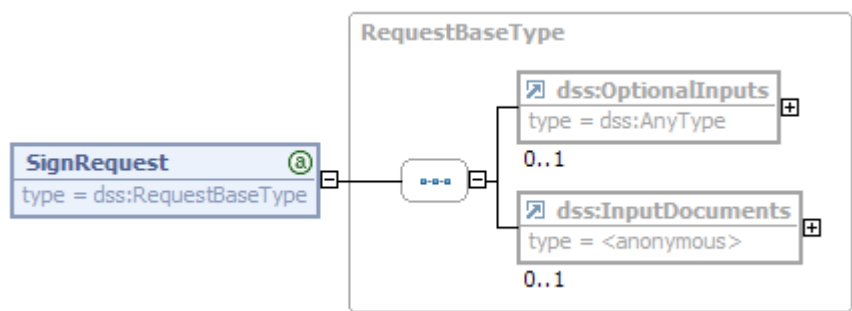
- Anfordern eines (qualifizierten) Zeitstempels (`TimestampRequest` / `TimeStampResponse`)
- Prüfen eines (qualifizierten) Zeitstempels (`VerifyRequest` / `VerifyResponse`)
- Berechnung eines Hashwertes (`Hash` / `HashResponse`)

5.3.1 Anfordern eines (qualifizierten) Zeitstempels

Zum Anfordern eines (qualifizierten) Zeitstempels kann ein geeignet profilierter Funktionsaufruf `SignRequest` mit entsprechender Antwort `SignResponse` gemäß [OASIS-DSS] bzw. [eCard-2] genutzt werden.

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

5.3.1.1 SignRequest für das Anfordern eines Zeitstempels

Name	SignRequest	
Beschreibung	Ein SignRequest im Kontext der Schnittstelle S.3 übergibt einen Hashwert, zu dem ein (qualifizierter) Zeitstempel erstellt werden soll, an das Krypto-Modul.	
Beschreibung		
	Aufruf der SignRequest-Funktion	
	Name	Beschreibung
	dss:OptionalInputs	Enthält genau ein Element SignatureType mit der URI urn:ietf:rfc:3161 , durch die klargestellt wird, dass ein Zeitstempel gemäß [RFC3161] erzeugt werden soll.
dss:InputDocuments	(A4.3.1.1-1): Während das Element dss:InputDocuments in [OASIS-DSS] und [eCard-2] optional ist, muss es hier vorhanden sein und genau ein dss:Document-Element in der DocumentHash-Ausprägung enthalten. Dieses Element enthält den Hashwert, aus dem ein (qualifizierter) Zeitstempel erzeugt werden soll.	

5.3.1.2 SignResponse mit Zeitstempel

Name	SignResponse	
Beschreibung	Als Antwort auf einen SignRequest wird vom Krypto-Modul ein entsprechendes SignResponse-Element gemäß Abschnitt 3.2.1 von [eCard-2] zurückgeliefert. Im Kontext der Schnittstelle S.3 wird hier ein (qualifizierter) Zeitstempel zurückgeliefert.	
Rückgabe	<p>SignResponse ist die Antwort zum SignRequest-Aufruf</p>	
	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.
	dss:OptionalOutputs	Das optionale Element dss:OptionalOutputs ist nicht vorhanden.
	dss:SignatureObject	Enthält – sofern kein Fehler aufgetreten ist – genau ein dss:SignatureObject-Element, das ein dss:Timestamp-Element enthält, in dem der Zeitstempel in Form eines RFC3161TimeStampToken-Elementes enthalten ist.

5.3.2 Prüfen eines (qualifizierten) Zeitstempels

Zum Prüfen eines (qualifizierten) Zeitstempels ist in TR-S.3 (vgl. Abbildung 2) der Funktionsaufruf `VerifyRequest` und die Antwort `VerifyResponse` gemäß [OASIS-DSS] und [eCard-2] vorgesehen.

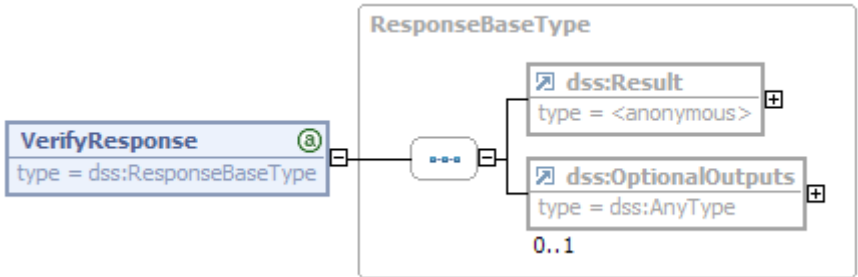
Die Durchführung der eigentlichen Prüffunktion eines (qualifizierten Zeitstempels) muss im Krypto-Modul (siehe Anlage [TR-ESOR-M.2]) als Komponente der TR-ESOR-Middleware oder in einem, vom Krypto-Modul aufgerufen, externen Validierungsdienst eines (qualifizierten) Vertrauensdiensteanbieters erfolgen.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

5.3.2.1 VerifyRequest

Name	VerifyRequest						
Beschreibung	Ein <code>VerifyRequest</code> im Kontext der Schnittstelle S.3 übergibt einen (qualifizierten) Zeitstempel an das Krypto-Modul zur Verifikation der darin enthaltenen 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].						
Aufrufparameter	<p>Aufruf der <code>VerifyRequest</code>-Funktion.</p> <table border="1"> <thead> <tr> <th>Name</th><th>Beschreibung</th></tr> </thead> <tbody> <tr> <td><code>dss:OptionalInputs</code></td><td> <p><u>Kann</u> optionale Eingabeelemente enthalten.</p> <p>(A4.3.2.1-1): Gemäß der vorliegenden Spezifikation <u>muss</u> 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 <u>müssen</u>.</p> <p>(A4.3.2.1-2): Darüber hinaus <u>soll</u> das optionale Eingabeelement <code>ReturnVerificationReport</code> unterstützt werden, so dass für den entsprechenden Zeitstempel ein Prüfbericht gemäß [OASIS-VR] zurückgeliefert werden kann.</p> </td></tr> <tr> <td><code>dss:InputDocuments</code></td><td>Das optionale Element <code>dss:InputDocuments</code> <u>soll nicht</u> vorhanden sein und wird ignoriert.</td></tr> </tbody> </table>	Name	Beschreibung	<code>dss:OptionalInputs</code>	<p><u>Kann</u> optionale Eingabeelemente enthalten.</p> <p>(A4.3.2.1-1): Gemäß der vorliegenden Spezifikation <u>muss</u> 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 <u>müssen</u>.</p> <p>(A4.3.2.1-2): Darüber hinaus <u>soll</u> das optionale Eingabeelement <code>ReturnVerificationReport</code> unterstützt werden, so dass für den entsprechenden Zeitstempel ein Prüfbericht gemäß [OASIS-VR] zurückgeliefert werden kann.</p>	<code>dss:InputDocuments</code>	Das optionale Element <code>dss:InputDocuments</code> <u>soll nicht</u> vorhanden sein und wird ignoriert.
Name	Beschreibung						
<code>dss:OptionalInputs</code>	<p><u>Kann</u> optionale Eingabeelemente enthalten.</p> <p>(A4.3.2.1-1): Gemäß der vorliegenden Spezifikation <u>muss</u> 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 <u>müssen</u>.</p> <p>(A4.3.2.1-2): Darüber hinaus <u>soll</u> das optionale Eingabeelement <code>ReturnVerificationReport</code> unterstützt werden, so dass für den entsprechenden Zeitstempel ein Prüfbericht gemäß [OASIS-VR] zurückgeliefert werden kann.</p>						
<code>dss:InputDocuments</code>	Das optionale Element <code>dss:InputDocuments</code> <u>soll nicht</u> vorhanden sein und wird ignoriert.						

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

5.3.2.2 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	 <p>Rückgabe der VerifyRequest-Funktion</p>	
	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, <u>muss</u> 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 <u>kann</u> ein VerificationReport-Element gemäß [OASIS VR] vorhanden sein, das im IndividualReport/Details-Element ein IndividualTimeStampReport-Element enthält.

5.3.3 Berechnung eines Hashwertes

Zur Berechnung eines Hashwertes ist in TR-S.3 (vgl. Abbildung 2) der Funktionsaufruf `Hash` und die Antwort `HashResponse` aus [eCard-4] in Verbindung mit dem Generic Cryptography-Protokoll aus [eCard-7] vorgesehen.

5.3.3.1 Hash

Name	Hash										
Beschreibung	Bei einem Hash-Aufruf im Kontext der Schnittstelle S.3 wird für die übergebenen Daten ein Hashwert berechnet.										
Aufrufparameter	<p>Aufruf der Funktion Hash.</p> <table> <thead> <tr> <th>Name</th><th>Beschreibung</th></tr> </thead> <tbody> <tr> <td>ConnectionHandle</td><td>Das ConnectionHandle-Element (vgl. [eCard-4], Abschnitt 3.1.3) gibt bei Bedarf an, auf welchem Hardwaremodul oder entfernten eCard-API-Framework die Berechnung des Hashwertes erfolgen soll. Sofern die Berechnung des Hashwertes durch das lokale Software-Modul erfolgen soll, <u>soll</u> das ConnectionHandle-Element leer sein.</td></tr> <tr> <td>DIDName¹⁴</td><td>Dieser Parameter spezifiziert den zu verwendenden Hashalgorithmus. Welche kryptographischen Algorithmen zu einem bestimmten Zeitpunkt als geeignet erachtet werden, ist Gegenstand von [ETSI TS 119 312] und [SOG-IS].</td></tr> <tr> <td>DIDScope</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>Message</td><td>Enthält die Nachricht (bzw. einen Teil derselben, siehe [eCard-7]), aus der ein Hashwert berechnet werden soll.</td></tr> </tbody> </table>	Name	Beschreibung	ConnectionHandle	Das ConnectionHandle-Element (vgl. [eCard-4], Abschnitt 3.1.3) gibt bei Bedarf an, auf welchem Hardwaremodul oder entfernten eCard-API-Framework die Berechnung des Hashwertes erfolgen soll. Sofern die Berechnung des Hashwertes durch das lokale Software-Modul erfolgen soll, <u>soll</u> das ConnectionHandle-Element leer sein.	DIDName ¹⁴	Dieser Parameter spezifiziert den zu verwendenden Hashalgorithmus. Welche kryptographischen Algorithmen zu einem bestimmten Zeitpunkt als geeignet erachtet werden, ist Gegenstand von [ETSI TS 119 312] und [SOG-IS].	DIDScope	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.	Message	Enthält die Nachricht (bzw. einen Teil derselben, siehe [eCard-7]), aus der ein Hashwert berechnet werden soll.
Name	Beschreibung										
ConnectionHandle	Das ConnectionHandle-Element (vgl. [eCard-4], Abschnitt 3.1.3) gibt bei Bedarf an, auf welchem Hardwaremodul oder entfernten eCard-API-Framework die Berechnung des Hashwertes erfolgen soll. Sofern die Berechnung des Hashwertes durch das lokale Software-Modul erfolgen soll, <u>soll</u> das ConnectionHandle-Element leer sein.										
DIDName ¹⁴	Dieser Parameter spezifiziert den zu verwendenden Hashalgorithmus. Welche kryptographischen Algorithmen zu einem bestimmten Zeitpunkt als geeignet erachtet werden, ist Gegenstand von [ETSI TS 119 312] und [SOG-IS].										
DIDScope	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.										
Message	Enthält die Nachricht (bzw. einen Teil derselben, siehe [eCard-7]), aus der ein Hashwert berechnet werden soll.										

¹⁴ 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.

5.3.3.2 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	<p>Rückgabe der Funktion Hash.</p>	
	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.

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

Dieser Abschnitt beschreibt in den folgenden Unterkapiteln, wie die in TR-S.5 (vgl. Abbildung 2) 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)

Neben der Umsetzung der Funktion „ArchivRetrieval-Request/-Request“ zum Auslesen eines Archivdatenobjektes“ (Download) auf Basis der, auch dem eCard-API-Frameworks ([BSI TR 03112]) zu Grunde liegenden, Basistypen aus [OASIS-DSS] kann diese Funktion auch anders technisch umgesetzt werden, um den Download von Datenobjekten im Rahmen eines logischen XAIP (LXAIP) gemäß ([TR-ESOR-F], Kap. 3.2) technisch performant zu ermöglichen. Dabei sind die Anforderungen gemäß des Hauptdokuments ([TR-ESOR], Kap. 7.2 und 7.45) zu erfüllen.

5.4.1 Abfrage beweiswerterhaltend archivierter Daten

Für die Abfrage beweiswerterhaltend archivierter Daten ist der Funktionsaufruf ArchiveRetrievalRequest und die Antwort ArchiveRetrievalResponse gemäß Abschnitt 3.3 vorgesehen.

5.4.2 Löschen von Archivdatenobjekten

Für das Löschen von Archivdatenobjekten ist der Funktionsaufruf `ArchiveDeletionRequest` und `ArchiveDeletionResponse` gemäß Abschnitt 3.5 vorgesehen.

5.4.3 Abfrage diskreter Datenobjekte

Für die Abfrage diskreter Datenobjekte ist der Funktionsaufruf `ArchiveDataRequest` und `ArchiveDataResponse` gemäß Abschnitt 3.5 vorgesehen.

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

Dieser Abschnitt beschreibt, wie die in Abbildung 2 dargestellte Schnittstelle TR-S.6 auf Basis der auch dem eCard-API-Framework (BSI TR-03112) zu Grunde liegenden Basistypen aus [OASIS-DSS] umgesetzt werden kann.

Die in Abbildung 2 dargestellte 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`)

5.5.1 Beweiswerterhaltende Archivierung elektronischer Daten

Für die beweiswerterhaltende Archivierung elektronischer Daten ist der Funktionsaufruf `ArchiveSubmissionRequest` und die Antwort `ArchiveSubmissionResponse` gemäß Abschnitt 3.1 vorgesehen.

5.5.2 Ergänzen einer neuen Version eines Archivdatenobjektes

Für das Ergänzen einer neuen Version eines Archivdatenobjektes ist der Funktionsaufruf `ArchiveUpdateRequest` und die Antwort `ArchiveUpdateResponse` gemäß Abschnitt 3.2 vorgesehen.

5.5.3 Rückgabe technischer Beweisdaten

Für die Rückgabe technischer Beweisdaten ist der Funktionsaufruf `ArchiveEvidenceRequest` und die Antwort `ArchiveEvidenceResponse` gemäß Abschnitt 3.4 vorgesehen.

6. 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/ar/DXAIP_NOK	Die Syntax des beim ArchiveUpdateRequest übergebenen Delta-XAIP-Elements ist nicht korrekt.
../resultminor/ar/DXAIP_NOK_AOID	Die AOID in dem beim ArchiveUpdateRequest übergebenen Delta-XAIP ist nicht bekannt.
../resultminor/ar/DXAIP_NOK_EXPIRED	Das beim ArchiveUpdateRequest übergebene Delta-XAIP-Element kann nicht abgelegt werden, da die Aufbewahrungsfrist abgelaufen ist.
../resultminor/ar/DXAIP_NOK_SUBMTIME	Die beim ArchiveUpdateRequest im übergebenen Delta-XAIP-Element angegebene submissionTime ist nicht korrekt, da sie in der Zukunft liegt.
../resultminor/ar/DXAIP_NOK_SIG	Das beim ArchiveUpdateRequest übergebene Delta-XAIP-Element enthält zumindest eine ungültige digitale Signatur.
../resultminor/ar/DXAIP_NOK_ER	Das beim ArchiveUpdateRequest übergebene Delta-XAIP-Element enthält zumindest einen ungültigen Evidence Record.
../resultminor/ar/DXAIP_NOK_ID	Die beim ArchiveUpdateRequest in einem placeHolder-Element übergebene XML-ID ist im bereits abgelegten XAIP-Element nicht vorhanden.
../resultminor/ar/DXAIP_NOK_Version	Die beim ArchiveUpdateRequest im prevVersion-Element übergebene Version ist nicht die aktuellste Version.
../resultminor/ar/existingAOID	Die im Rahmen des ArchiveSubmissionRequest übergebene AOID existiert bereits.
../resultminor/ar/existingPackageInfoWarning	Bei der ArchiveUpdateRequest-Funktion wird ein Delta-XAIP-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/ar1/lowSpaceWarning	Diese Warnung gibt an, dass der verfügbare Speicherplatz einen kritischen Wert unterschritten hat.
../resultminor/ar1/missingReasonOfDeletion	Da beim ArchiveDeletionRequest kein ReasonOfDeletion-Element übergeben wurde, muss der Löschvorgang abgewiesen werden.
../resultminor/ar1/noSpaceError	Diese Fehlermeldung gibt an, dass kein Speicherplatz verfügbar war und deshalb das Archivdatenobjekt nicht abgelegt werden konnte.
../resultminor/ar1/notSupported	Diese Fehlermeldung gibt an, dass eine angeforderte Funktion, ein angefordertes Format oder ein übergebener optionaler Eingabeparameter nicht unterstützt wird.
../resultminor/ar1/requestOnlyPartlySuccessfulWarning	Diese Warnung gibt an, dass nicht alle angeforderten Daten zurückgeliefert werden konnten.
../resultminor/ar1/unknownArchiveDataType	Es wird ein binäres Datenobjekt mit einem nicht unterstützten Datenformat übergeben.
../resultminor/ar1/unknownLocation	Die im ArchiveDataRequest angegebene DataLocation ist nicht vorhanden bzw. fehlerhaft.
../resultminor/ar1/unknownAOID	Die übergebene AOID existiert nicht.
../resultminor/ar1/unknownVersionID	Die übergebene VersionID ist im entsprechenden XAIP nicht bekannt.
../resultminor/ar1/XAIP_NOK	Die Syntax des übergebenen AIP-Containers (d.h. XAIP, LXAIP, ASiC-AIP) ist nicht korrekt.
../resultminor/ar1/XAIP_NOK_ER	Der übergebene AIP-Container (d.h. XAIP, LXAIP, ASiC-AIP) enthält zumindest einen ungültigen Evidence Record.
../resultminor/ar1/XAIP_NOK_EXPIRED	Der übergebene AIP-Container (d.h. XAIP, LXAIP, ASiC-AIP) kann nicht abgelegt werden, da die Aufbewahrungsfrist abgelaufen ist.
../resultminor/ar1/XAIP_NOK_SIG	Der übergebene AIP-Container (d.h. XAIP, LXAIP, ASiC-AIP) enthält zumindest eine ungültige Signatur.
../resultminor/ar1/XAIP_NOK_SUBMTIME	Die im übergebenen AIP-Container (d.h. XAIP, LXAIP, ASiC-AIP) angegebene submissionTime ist nicht korrekt, da sie in der Zukunft liegt.
../resultminor/ar1/noDataAccessWarning	Der Zugriff auf die in einem übergebenen LXAIP referenzierten Daten ist nicht möglich.
../resultminor/ar1/unknownPOFormat	Der angeforderte POFormat- Typ ist nicht bekannt.

7. 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 7.1). Darauf aufbauend wird in einem zweiten Schritt eine Webservice-Spezifikation gemäß [WSDL] entwickelt.

Abschnitt 7.2 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.

Die Unterstützung des optimierten Nachrichtenübertragungsmechanismus „SOAP Message Transmission Optimization Mechanism (MTOM)“¹⁵ kann durch den Import des geringfügig angepassten XAIP-Schema (tr-esor-xaip-v1.2+xmlmime.xsd) erfolgen.

7.1 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 (+xmlmime) vom 20.12.2018 -->
  <!-- ===== -->

  <import namespace="http://www.bsi.bund.de/tr-esor/xaip/1.2"
    schemaLocation="tr-esor-xaip-v1.2+xmlmime.xsd" />

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

  <import namespace="urn:ietf:params:xml:ns:ers"
    schemaLocation="./deps/xml-ers-rfc6283.xsd" />

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

  <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
    schemaLocation="./deps/saml-schema-assertion-2.0.xsd" />

  <!-- ===== -->
```

¹⁵ Siehe <https://www.w3.org/TR/soap12-mtom/>.

```
<!--      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>
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>

```

7.2 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 (+xmlmime) vom 20.12.2018 -->
<!--=====-->

<!-- ===== -->
<!-- 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="tr-esor-interfaces-
v1.2+xmlmime.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: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>
```