



Bundesamt
für Sicherheit in der
Informationstechnik

Deutschland
Digital•Sicher•BSI•

BSI Technische Richtlinie 03125

Beweiswerterhaltung kryptographisch signierter Dokumente

Anlage TR-ESOR-E: Konkretisierung der Schnittstellen auf
Basis des eCard-API-Frameworks und ETSI TS 119 512

Bezeichnung	Konkretisierung der Schnittstellen auf Basis des eCard-API-Frameworks
Kürzel	BSI TR-ESOR-E
Version	1.3 (auf Basis der eIDAS-Verordnung und der ETSI Preservation Standards mit einem neuen Zertifizierungsschema)
Datum	31.03.2022



Änderungshistorie

<i>Version</i>	<i>Datum</i>	<i>Name</i>	<i>Beschreibung</i>
1.3	31.03.2022	BSI	TR-ESOR-E

Tabelle 1: Änderungshistorie

Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63
53133 Bonn
Tel.: +49 22899 9582-0
E-Mail: tresor@bsi.bund.de
Internet: <https://www.bsi.bund.de>
© Bundesamt für Sicherheit in der Informationstechnik 2022

Inhalt

1. Einführung.....	6
2. Überblick.....	9
3. Funktionen der ArchiSafe-Schnittstelle (TR-S.4).....	14
3.1 Funktion: ArchiveSubmission.....	14
3.1.1 Eingabeparameter: ArchiveSubmissionRequest.....	15
3.1.2 Ausgabeparameter: ArchiveSubmissionResponse.....	18
3.2 Funktion: ArchiveUpdate.....	20
3.2.1 Eingabeparameter: ArchiveUpdateRequest.....	20
3.2.2 Ausgabeparameter: ArchiveUpdateResponse.....	21
3.3 Funktion: ArchiveRetrieval.....	22
3.3.1 Eingabeparameter: ArchiveRetrievalRequest.....	23
3.3.2 Ausgabeparameter: ArchiveRetrievalResponse.....	25
3.4 Funktion: ArchiveEvidence.....	27
3.4.1 Eingabeparameter: ArchiveEvidenceRequest.....	27
3.4.2 Ausgabeparameter: ArchiveEvidenceResponse.....	29
3.5 Funktion: ArchiveDeletion.....	31
3.5.1 Eingabeparameter: ArchiveDeletionRequest.....	31
3.5.2 Ausgabeparameter: ArchiveDeletionResponse.....	32
3.6 Funktion: ArchiveData.....	32
3.6.1 Eingabeparameter: ArchiveDataRequest.....	33
3.6.2 Ausgabeparameter: ArchiveDataResponse.....	35
3.7 Funktion: Verify.....	36
3.7.1 Eingabeparameter: VerifyRequest.....	37
3.7.2 Ausgabeparameter: VerifyResponse.....	40
3.8 Funktion: RetrieveInfo.....	41
3.8.1 Eingabeparameter: RetrieveInfoRequest.....	41
3.8.2 Ausgabeparameter: RetrieveInfoResponse.....	42
3.9 Funktion: ArchiveTrace.....	43
3.9.1 Eingabeparameter: ArchiveTraceRequest.....	43
3.9.2 Ausgabeparameter: ArchiveTraceResponse.....	44
4. Funktionen der Preservation-API gemäß ETSI TS 119 512 in der Profilierung [TR-ESOR-TRANS].....	46
4.1 Vergleich der TR-S.512- mit der TR-S.4-Schnittstelle.....	46
5. Funktionen der internen Schnittstellen.....	48
5.1 TR-S.1 (ArchiSafe-Modul – Krypto-Modul).....	48
5.1.1 Prüfung von digitalen Signaturen, beweisrelevanten Daten, Beweisdaten und Archivdatenobjekten.....	48

5.1.2	Anforderung einer digitalen Signatur	48
5.2	TR-S.2 (ArchiSig-Modul – ECM-/Langzeitspeichersystem)	50
5.2.1	Speichern eines Archivdatenobjektes.....	50
5.2.2	Ergänzen einer neuen Version eines Archivdatenobjektes	50
5.2.3	Auslesen von Archivdatenobjekten.....	50
5.3	TR-S.3 (ArchiSig-Modul – Krypto-Modul).....	51
5.3.1	Anfordern eines (qualifizierten) Zeitstempels.....	51
5.3.2	Prüfen eines (qualifizierten) Zeitstempels	52
5.3.3	Berechnung eines Hashwertes.....	54
5.4	TR-S.5 (ArchiSafe-Modul / Krypto-Modul – ECM-Langzeitspeichersystem).....	56
5.4.1	Abfrage beweiswerterhaltend archivierter Daten	56
5.4.2	Löschen von Archivdatenobjekten.....	57
5.4.3	Abfrage diskreter Datenobjekte	57
5.5	TR-ESOR-S.6 (ArchiSafe-Modul – ArchiSig-Modul)	57
5.5.1	Beweiswerterhaltende Archivierung elektronischer Daten.....	57
5.5.2	Ergänzen einer neuen Version eines Archivdatenobjektes	57
5.5.3	Rückgabe technischer Beweisdaten	57
6.	Upload/Download-Schnittstelle	58
6.1	Upload-Funktion.....	58
6.1.1	Upload-Anfrage	58
6.1.2	Upload-Antwort	58
6.2	Download-Funktion.....	59
6.2.1	Download-Anfrage.....	60
6.2.2	Download-Antwort.....	60
7.	Fehlercodes.....	62
8.	Spezifikation einer Webservice-basierten Schnittstelle	65
8.1	Spezifikation der Aufruf- und Rückgabeparameter als XML-Schema.....	65
8.2	WSDL-Spezifikation der Schnittstelle TR-S.4	70

Abbildungen

Abbildung 1: Schematische Darstellung der IT-Referenzarchitektur mit TR-S.4.	7
Abbildung 2: Schematische Darstellung der IT-Referenzarchitektur mit TR-S.512.	7
Abbildung 3: Umsetzung der IT-Referenzarchitektur auf Basis des eCard-API-Frameworks.....	10
Abbildung 4: Umsetzung der IT-Referenzarchitektur auf Basis [ETSI TS 119 512].	10

Tabellen

Tabelle 1: Änderungshistorie.....	2
Tabelle 2: Bewahrungstechniken.....	12

Tabelle 3: Vergleich ETSI TS 119 512 (prof. [TR-ESOR-TRANS]) Preservation-API mit TR-ESOR-S.4-Schnittstelle.....	47
Tabelle 4: Zusätzliche Fehlercodes.	64

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).¹

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 Schnittstellen, Funktionen und logischen Einheiten:

- der TR-S.4 oder TS119512-Eingangs-Schnittstelle TR-S.512 in der Profilierung **[TR-ESOR-TRANS]** der TR-ESOR-Middleware, die dazu dient, die TR-ESOR-Middleware in die bestehende IT- und Infrastrukturlandschaft einzubetten;
- dem „ArchiSafe-Modul“ (vgl. **[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“ (vgl. **[TR-ESOR-M.2]**) nebst den zugehörigen Schnittstellen TR-S.1 und TR-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“ (vgl. **[TR-ESOR-M.3]**) mit der Schnittstelle TR-S.6, das die erforderlichen Funktionen für die Beweiswerterhaltung der digital signierten Unterlagen bereitstellt;
- einem ECM-/Langzeitspeicher mit den Schnittstellen TR-S.2 und TR-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 empfohlene IT-Referenzarchitektur ist in Abbildung 1 und Abbildung 2 dargestellt und besteht im Wesentlichen aus den in **[TR-ESOR]**, Kap. 7 grob beschriebenen logischen Komponenten und Schnittstellen. Die Grafik zeigt zudem die externen Komponenten und Systeme an, die das Bild vervollständigen. Grundsätzlich wird als obere Schnittstelle der TR-ESOR-Middleware entweder die TR-S.4-Schnittstelle gemäß **[TR-ESOR-E]**, die in Abbildung 1 dargestellt ist, oder die TR-S.512-Schnittstelle gemäß **[ETSI TS 119 512]** in der Profilierung **[TR-ESOR-TRANS]**, die in Abbildung 2 gezeigt wird, unterstützt.

¹Siehe Hinweis 1

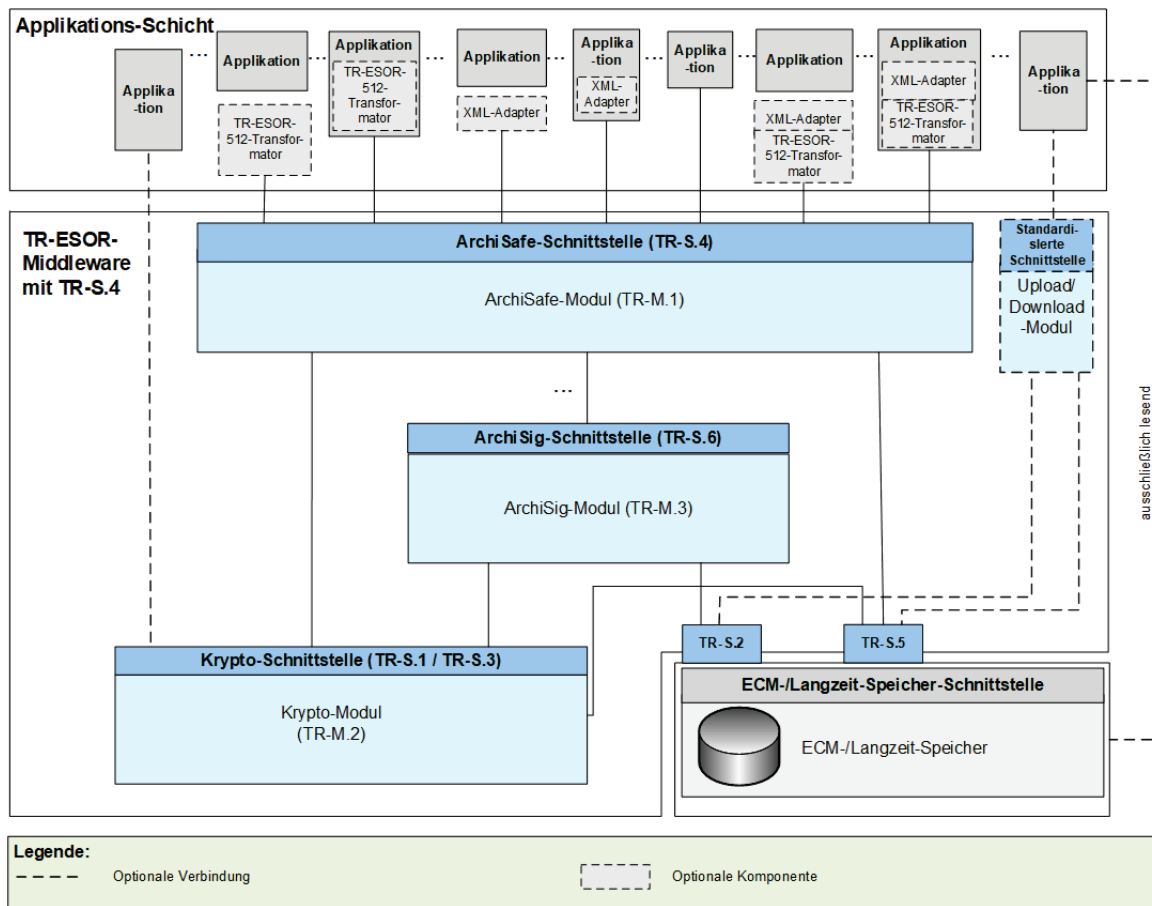


Abbildung 1: Schematische Darstellung der IT-Referenzarchitektur mit TR-S.4.

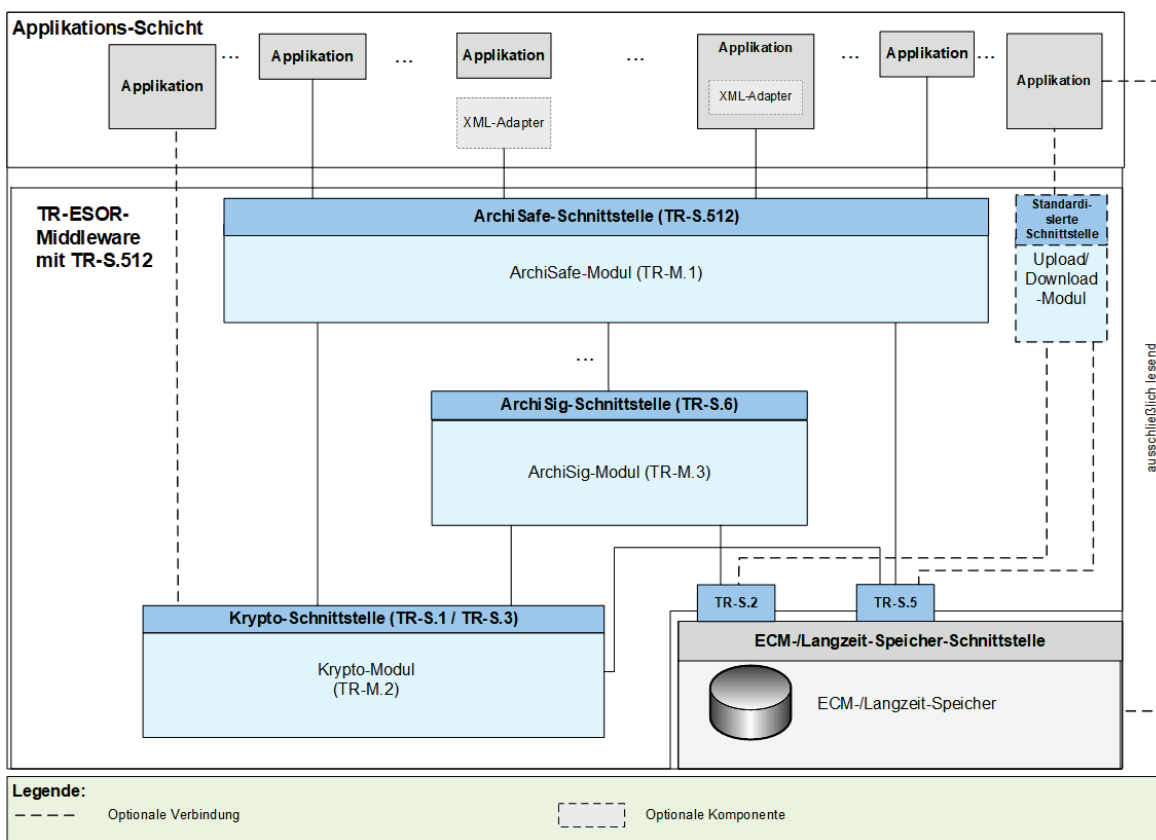


Abbildung 2: Schematische Darstellung der IT-Referenzarchitektur mit TR-S.512.

Die in Abbildung 1 bzw. Abbildung 2 dargestellte IT-Referenzarchitektur orientiert sich an der ArchiSafe Referenzarchitektur und soll die logische (funktionale) Interoperabilität künftiger Produkte mit den Zielen und Anforderungen der Technischen Richtlinie ermöglichen und unterstützen.

Sofern der optionale XML-Adapter und/oder der optionale TR-ESOR-512-Transformator² vorhanden sind, können beide in folgenden Ausprägungen vorliegen:

- Jeweils eigenständige Komponente mit Schnittstellen zur Applikation sowie zum ArchiSafe-Modul
- Jeweils eigenständige Komponente, jedoch Teil der Applikation mit Schnittstelle zum ArchiSafe-Modul
- XML-Adapter und TR-ESOR-512-Transformator als eine gemeinsame Komponente, die beide Teile enthält mit Schnittstellen zur Applikation sowie zum ArchiSafe-Modul
- XML-Adapter und TR-ESOR-512-Transformator als eine gemeinsame Komponente, die beide Teile enthält und Teil der Applikation ist, mit Schnittstelle zum ArchiSafe-Modul.

Der "ETSI TS119512 TR-ESOR Transformator" ermöglicht Bewahrungsdiensten gemäß [eIDAS-VO], empfangene ETSI TS119512 (V1.1.2) Nachrichten³ in TR-S4 Nachrichten zu transformieren. Diese Nachrichten können dann an ein angeschlossenen TR-ESOR-System⁴ geschickt werden, ohne irgendwelche Änderungen dieses TR-ESOR-Systems.

Der Einsatz des TR-ESOR-512-Transformators wird EMPFOHLEN, sofern das TR-ESOR-Produkt mit einer TR-S.4-Schnittstelle in Europa zum Einsatz kommt und Interoperabilität mit europäischen (qualifizierten) Bewahrungsdiensten und Bewahrungsprodukten hergestellt werden soll.

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 TR-S.4 auf Basis des in der [TR-03112] spezifizierten eCard-API-Frameworks sowie die TR-S.512-Schnittstelle auf Basis von [ETSI TS 119 512] in der Profilierung [TESOR-TRANS].

² Siehe [ETSI TS 119512 TR-ESOR Transformator unter einer Open Source Lizenz](#).

³ In der Profilierung von [TR-ESOR-TRANS]

⁴ Siehe <https://www.bsi.bund.de/EN/tr-esor> oder <https://www.bsi.bund.de/DE/tr-esor>.

2. Überblick

(A2.0–1) Als ArchiSafe-Schnittstelle muss entweder die nachfolgend spezifizierte TR-S.4-Schnittstelle implementiert sein oder die TR-S.512-Schnittstelle gemäß [ETSI TS 119 512] in der Profilierung [TR-ESOR-TRANS].

(A2.0–2) Falls die TR-S.4-Schnittstelle unterstützt wird, dann müssen die im Folgenden näher aufgeführten Funktionen mit den hier beschriebenen Parameterkonstellationen unterstützt werden:

- ArchiveSubmission (siehe Abs. 3.1)
- ArchiveUpdate (siehe Abs. 3.2)
- ArchiveRetrieval (siehe Abs. 3.3)
- ArchiveEvidence (siehe Abs. 3.4)
- ArchiveDeletion (siehe Abs. 3.5)
- Verify (siehe Abs. 3.7)
- RetrieveInfo (siehe Abs. 3.8).

Falls die TR-S.4-Schnittstelle unterstützt wird, dann sollen in der Schnittstelle TR-S.4 die folgenden im vorliegenden Dokument näher aufgeführten Funktionen mit den hier beschriebenen Parameterkonstellationen unterstützt werden:

- ArchiveData (siehe Abs. 3.6)
- ArchiveTrace (siehe Abs. 3.9)

(A2.0–3) Falls die TR-S.512-Schnittstelle unterstützt wird, dann müssen die im Folgenden näher aufgeführten Funktionen mit den in [ETSI TS 119 512] in der Profilierung [TR-ESOR-TRANS] beschriebenen Parameterkonstellationen unterstützt werden:

- PreservePO (siehe [TR-ESOR-TRANS], Abs. 3.2)
- UpdatePOC (siehe [TR-ESOR-TRANS], Abs. 3.3)
- RetrievePO (siehe [TR-ESOR-TRANS], Abs. 3.4)
- DeletePO (siehe [TR-ESOR-TRANS], Abs. 3.5)
- ValidateEvidence (siehe [TR-ESOR-TRANS], Abs. 3.6)
- RetrieveInfo (siehe [TR-ESOR-TRANS], Abs. 3.1).

Falls die TR-S.512-Schnittstelle unterstützt wird, dann sollen in der Schnittstelle TR-S.512 die folgenden im vorliegenden Dokument näher aufgeführten Funktionen mit den dort beschriebenen Parameterkonstellationen unterstützt werden:

- Search (siehe [TR-ESOR-TRANS], Abs. 3.7)
- RetrieveTrace (siehe [ETSI TS 119 512] Abs. 5.3.7).

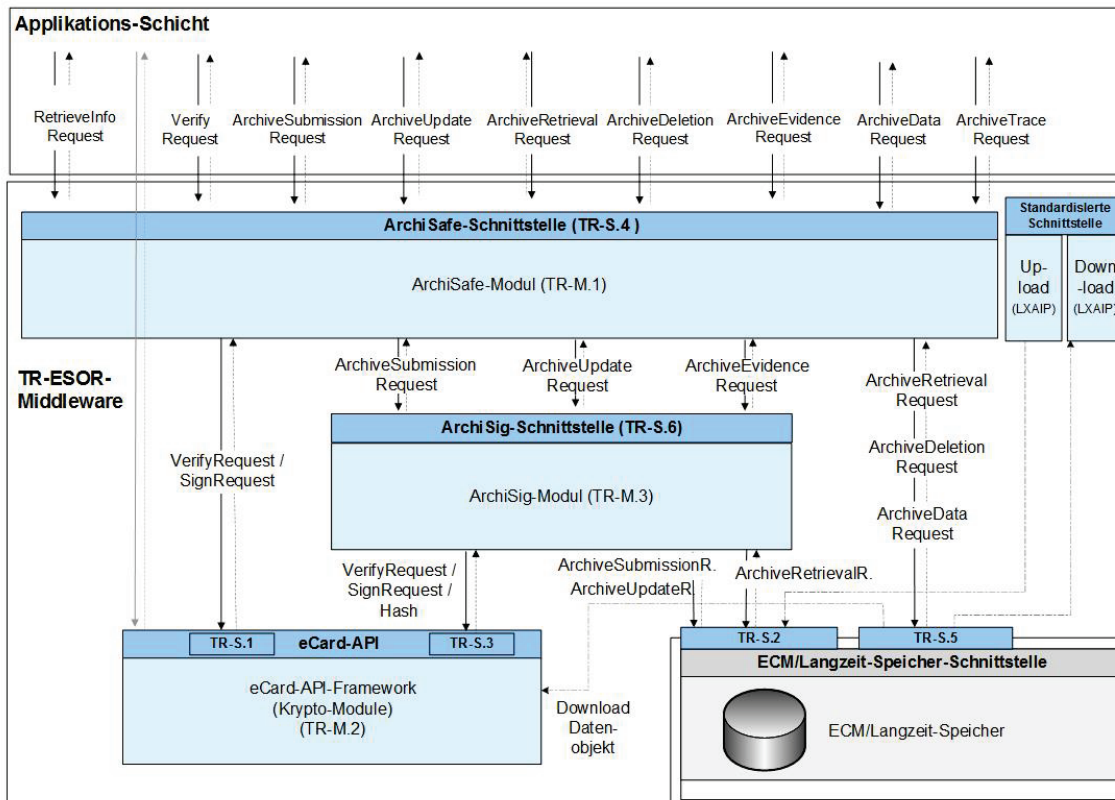


Abbildung 3: Umsetzung der IT-Referenzarchitektur auf Basis des eCard-API-Frameworks.

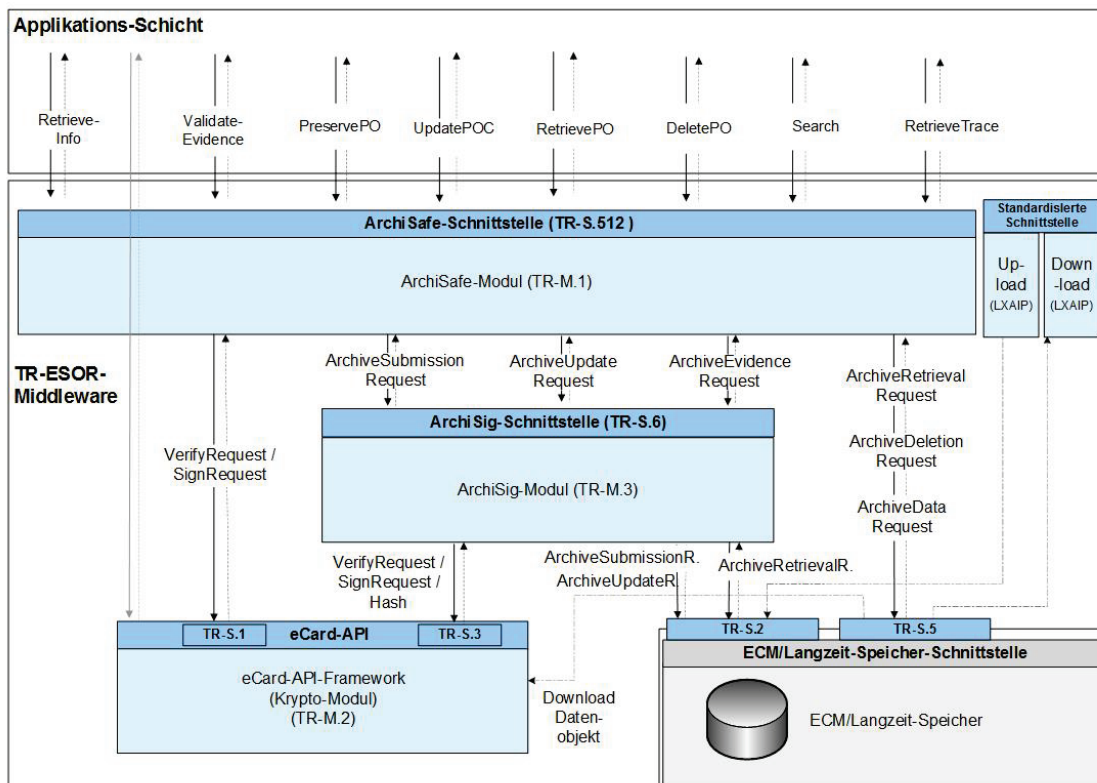


Abbildung 4: Umsetzung der IT-Referenzarchitektur auf Basis [ETSI TS 119 512].

Wie in Abbildung 1 und Abbildung 3 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 [TR-03112]) realisiert und 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 [TR-03112] definierten Funktionen haben das Präfix <http://www.bsi.bund.de/tr-esor/api/1.3>, welches um entsprechende Bezeichner ergänzt wird. Dieser Namensraum ist in den visualisierten XML-Strukturen am Kürzel „tr“ erkennbar.

Im Fall der Abbildung 2 und Abbildung 4 wird die obere Schnittstelle gemäß [ETSI TS 119 512], Kap. 5 auf Basis von [OASIS DSS-X], Core 2.0 realisiert.

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 und der TR-S.512-Schnittstelle in Abschnitt 4 spezifiziert. 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 1

In der vorliegenden TR-ESOR-Version 1.3 werden die drei Begriffe „(beweiswerterhaltende) Langzeitspeicherung“, „(beweiswerterhaltende) Bewahrung“ und „(beweiswerterhaltende) Archivierung“ synonym verwendet. Ebenso werden die drei Begriffe „Archivinformationspaket (AIP)“, „Archivinformationscontainer“ und „Archivdatenobjekt“ sowie die Begriffe „aufbewahren“ und „archivieren“ synonym verwendet.

HINWEIS 2

TR-ESOR spezifiziert ein Bewahrungsprodukt (engl. Preservation Product) gemäß [ETSI SR 019 510], [ETSI TS 119511] und [ETSI TS 119512] und [eIDAS-VO].

*Die TR 03125 TR-ESOR ist in [ETSI SR 019510] in dem Kapitel 4.7.3 und 5.2 und B3.2 beschrieben. Die in TR-ESOR erforderlichen grundlegenden Bewahrungstechniken, z. B. das Bewahrungsprotokoll, das Beweisdaten-Format Evidence Record, die Archivdatenobjekt-Format (L)XAIP und ASiC-AIP sind in der ETSI-Publikation [ETSI TS 119512] als **normative Elemente** enthalten.*

HINWEIS 3

*Die obere TR-ESOR-Eingangs-Schnittstelle **TR-S.4** oder die TS119512-Eingangsschnittstelle **TR-S.512** gemäß der „Preservation-API“ in [ETSI TS 119 512] in der Profilierung von [TR-ESOR-TRANS], die logisch äquivalent zur Eingangsschnittstelle S.4 gemäß [TR-ESOR-E] ist wie in der Tabelle 2 in [TR-ESOR-E], Kapitel 4.1 dargestellt, muss benutzt werden. Eine andere Eingangs-Schnittstelle anstelle von **TR-S.4** bzw. **TR-S.512** ist nicht erlaubt (vgl. A7.1-1 in [TR-ESOR]).*

HINWEIS 4

In der vorliegenden TR-ESOR-Version 1.3 umfasst der Begriff „Archivinformationscontainer“ (AIP) in allen TR-ESOR-Anhängen:

- a) das Archivdatenobjekt „XAIP“ gem. [TR-ESOR-F], Kap. 3.1 als auch
- b) das logische XAIP „LXAIP“ gem. [TR-ESOR V1.3], Kap. 3.2 und
- c) das „ASiC-AIP“ gem. [TR-ESOR-F], Kap. 3.3 auf Basis von [ETSI EN 319162-1].

In TR-ESOR Version V1.3 wird zwischen XAIP, LXAIP und ASiC-AIP differenziert.

Mit (L)XAIP wird XAIP oder LXAIP bezeichnet.

HINWEIS 5

In dieser TR-ESOR Version 1.3 ist "BIN" beschränkt auf die folgenden Bewahrungsobjekt-Formate (engl. *preservation object formats*):

- CAdES gemäß [ETSI TS 119 512], Annex A.1.1 (<http://uri.etsi.org/ades/CAdES>). Sofern kein MIME Type gesetzt ist, wird als Default application/cms verwendet;
- XAdES gemäß [ETSI TS 119 512], Annex A.1.2 (<http://uri.etsi.org/ades/XAdES>). Sofern kein MIME Type gesetzt ist, wird als Default application/xml verwendet;
- PAdES gemäß [ETSI TS 119 512], Annex A.1.3 (<http://uri.etsi.org/ades/PAdES>). Sofern kein MIME Type gesetzt ist, wird als Default application/pdf verwendet;
- ASiC-E gemäß [ETSI TS 119 512], Annex A.1.4 (<http://uri.etsi.org/ades/ASiC/type/ASiC-E>). Sofern kein MIME Type gesetzt ist, wird als Default [application/vnd.etsi.asic-e+zip](http://uri.etsi.org/ades/ASiC/type/ASiC-E) verwendet;
- ASiC-S gemäß [ETSI EN 319 162] (<http://uri.etsi.org/ades/ASiC/type/ASiC-S>). Sofern kein MIME Type gesetzt ist, wird als Default application/vnd.etsi.asic-s+zip verwendet.
- DigestList gemäß [ETSI TS 119 512], Annex A.1.6 (<http://uri.etsi.org/19512/format/DigestList>). Sofern kein MIME Type gesetzt ist, wird als Default application/xml verwendet;
- ASiC-ERS (in TR-ESOR v1.3 mit ASiC-AIP bezeichnet) gemäß [TR-ESOR-F], Kapitel 3.3 und gemäß [ETSI TS 119 512], Annex A.3.1 (<http://uri.etsi.org/ades/ASiC/type/ASiC-ERS>).

Im Falle Upload/Download-Funktion ist zusätzlich nachfolgendes Format erlaubt:

- Binärdaten (BIN) als "Octet Stream", die ausschließlich in den ECM-/Langzeitspeicher mit "Upload-Request" gespeichert werden, – aber nur sofern:
 - a) verbunden mit einem korrespondierenden LXAIP und dort referenziert gem. [TR-ESOR-F], Kap. 3.2,
 - b) ggf. mit "Download-Request" ausgelesen werden, – verbunden mit einem korrespondierenden LXAIP, das mit der „ArchiveRetrieval“-Funktion ausgelesen wurde, – oder eingebettet in einem XAIP und ausgelesen mit der „ArchivRetrieval“-Funktion.
 - c) Der Upload von XAIP oder LXAIP oder ASiC-AIP ist nicht zugelassen.

HINWEIS 6

TR-ESOR spezifiziert ein Bewahrungsprodukt (engl. *Preservation Product*) gemäß [ETSI SR 019 510], [ETSI TS 119 511] und [ETSI TS 119 512] und [eIDAS-VO].

Die TR 03125 TR-ESOR ist in [ETSI SR 019 510] in dem Kapitel 4.7.3 und 5.2 und B3.2 beschrieben. Die in TR-ESOR erforderlichen grundlegenden Bewahrungstechniken, z.B. das Bewahrungsprotokoll, das Beweisdaten-Format Evidence Record, die Archivdatenobjekt-Format (L)XAIP und ASiC-AIP sind in der ETSI-Publikation [ETSI TS 119 512] als normative Elemente enthalten, (siehe Tabelle darunter):

Tabelle 2: Bewahrungstechniken

Bewahrungstechnik	ETSI TS 119 512	Verbindlichkeitsgrad <i>N=normativ O=optional C=conditional</i>	TR-ESOR Dokument	Verbindlichkeitsgrad <i>N=normativ O=optional C=conditional</i>
Bewahrungsprotokoll („Preservation Protocol“): TR-S.512	Kapitel 5.3	N	[TR-ESOR-E], Kap. 4	C Auswahl: TR-S.512
Beweisdaten-Format („Preservation Evidence Format“): Evidence Record	A.2.2 bzw. A2.3	N	[TR-ESOR-F], Kap. 5.5, [TR-ESOR-ERS]	N
Archivdatenobjekt-Format „...Data Object Format“ XAIP	A.1.5 und A.3.2	N	[TR-ESOR-F], Kap. 3.1 und 3.2	N

Bewahrungstechnik	ETSI TS 119 512	Verbindlichkeitsgrad <i>N=normativ O=optional C=conditional</i>	TR-ESOR Dokument	Verbindlichkeitsgrad <i>N=normativ O=optional C=conditional</i>
Archivdatenobjekt-Format „...Data Object Format“ LXAIP	A.1.5 und A.3.2	N	[TR-ESOR-F], Kap. 3.1 und 3.2	C
Archivdatenobjekt-Format „...Data Object Format“ ASiC-E/ASiC-ERS	A.1.4 und A.3.1	N	[TR-ESOR-F], Kap. 3.3	C
Versionierung von Archivinformationspaketen („Preservation Object Container“)	E	C	[TR-ESOR-E], Kap. 3.2 [TR-ESOR-F], Kap. 3.1.6 und 3.2.2	N

HINWEIS 7.

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-S.4)

In diesem Abschnitt findet sich eine XML-basierte Spezifikation der Funktionen und deren Eingabe- und Ausgabeparameter der TR-ESOR-Middleware an der ArchiSafe-Schnittstelle **TR-S.4**:

- Funktion `ArchiveSubmission` mit den Parametern `ArchiveSubmissionRequest` und `ArchiveSubmissionResponse` (siehe Abs. 3.1)
- Funktion `ArchiveUpdate` mit den Parametern `ArchiveUpdateRequest` und `ArchiveUpdateResponse` (siehe Abs. 3.2)
- Funktion `ArchiveRetrieval` mit den Parametern `ArchiveRetrievalRequest` und `ArchiveRetrievalResponse` (siehe Abs. 3.3)
- Funktion `ArchiveEvidence` mit den Parametern `ArchiveEvidenceRequest` und `ArchiveEvidenceResponse` (siehe Abs. 3.4)
- Funktion `ArchiveDelete` mit den Parametern `ArchiveDeletionRequest` und `ArchiveDeletionResponse` (siehe Abs. 3.5)
- Funktion `ArchiveData` mit den Parametern `ArchiveDataRequest` und `ArchiveDataResponse` (siehe Abs. 3.6)
- Funktion `Verify` mit den Parametern `VerifyRequest` und `VerifyResponse` (siehe Abs. 3.7)
- Funktion `RetrieveInfo` mit den Parametern `RetrieveInfoRequest` und `RetrieveInfoResponse` (siehe Abs. 3.8)
- Funktion `ArchiveTrace` mit den Parametern `ArchiveTraceRequest` und `ArchiveTraceResponse` (siehe Abs. 3.9)

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 Abs. 7) gegeben.

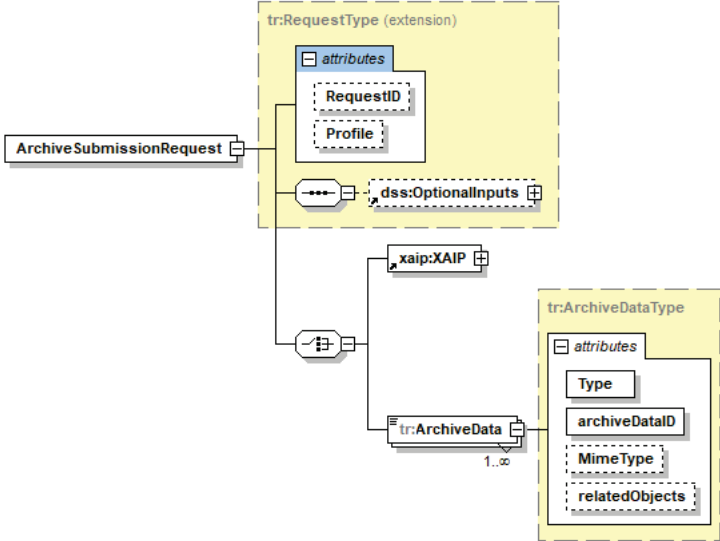
3.1 Funktion: ArchiveSubmission

Mit dem Funktionsparameter `ArchiveSubmissionRequest` wird dem aufgerufenen Modul ein Archivdatenobjekt zur Ablage übergeben und das aufrufende Modul erhält im Erfolgsfall in dem Ausgabeparameter `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 Abs. 3.1 in [TR-ESOR-F]) oder ein logisches XAIP (LXAIP) (siehe Abs. 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`-codierter⁵ ASiC-AIP-Container gemäß Abs. 3.3 in [TR-ESOR-F] mit einem `Type=http://uri.etsi.org/ades/ASiC/type/ASiC-ERS` Attribut übergeben werden.

Wie in Abbildung 3 oder Abbildung 4 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in den Schnittstellen TR-S.2 (vgl. Abs. 5.2) und TR-S.6 (vgl. Abs. 5.5) genutzt.

⁵Siehe <https://www.w3.org/TR/xmlschema-2/#base64Binary>.

3.1.1 Eingabeparameter: ArchiveSubmissionRequest

<i>Name</i>	<i>ArchiveSubmissionRequest</i>
Beschreibung	<p>Durch den Aufruf der Funktion <code>ArchiveSubmission</code> mit dem Eingabeparameter <code>ArchiveSubmissionRequest</code> wird dem aufgerufenen Modul ein oder mehrere <code>ArchiveData</code>-Element(e) oder ein Archivinformationspaket (XAIP, LXAIP, ASiC-AIP) übergeben.</p> <p>Hierbei kann für eine effiziente Übertragung von großen Binärdaten der optimierte Nachrichtenübertragungsmechanismus „SOAP Message Transmission Optimization Mechanism (MTOM)“⁶ genutzt werden.</p>
Details	<p>Der Eingabeparameter <code>ArchiveSubmissionRequest</code> weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</p>  <p>The diagram illustrates the structure of the <code>ArchiveSubmissionRequest</code> parameter. It is represented as a box connected to a larger container labeled <code>tr:RequestType (extension)</code>. This container has an <code>attributes</code> section with <code>RequestID</code> and <code>Profile</code> elements. Below the attributes is a dashed box labeled <code>dss:OptionalInputs</code>. A line connects the <code>ArchiveSubmissionRequest</code> box to a choice box (a circle with a vertical line). This choice box has two branches: one leading to <code>xaip:XAIP</code> and another leading to <code>tr:ArchiveData</code>. The <code>tr:ArchiveData</code> element is marked with a multiplicity of <code>1..∞</code>. To the right, the <code>tr:ArchiveDataType</code> is detailed with an <code>attributes</code> section containing <code>Type</code>, <code>archiveDataID</code>, <code>MimeType</code>, and <code>relatedObjects</code>.</p>
Name	Beschreibung

⁶Siehe <https://www.w3.org/TR/soap12-mtom/>.

Name	ArchiveSubmissionRequest
	<p>dss:OptionalInputs</p> <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"> • xaip:AOID, • vr:ReturnVerificationReport, • ImportEvidence. <p>Dabei gilt:</p> <ul style="list-style-type: none"> • xaip:AOID <div data-bbox="1075 546 1161 584" data-label="Diagram"> </div> <p>Durch die Übergabe eines xaip: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.</p> <ul style="list-style-type: none"> • vr:ReturnVerificationReport <div data-bbox="794 891 1310 1084" data-label="Diagram"> </div> <p>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. Abs. 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 gem. [TR-ESOR-VR] zurückgeliefert.</p>

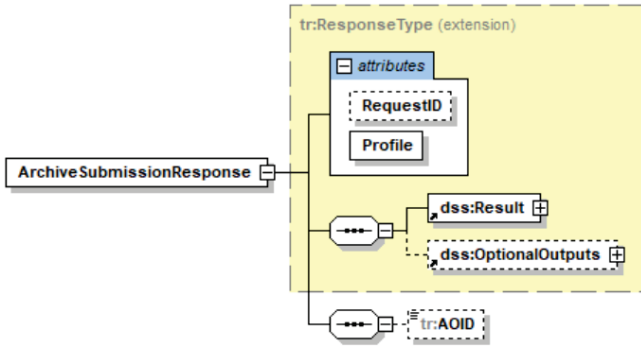
Name	ArchiveSubmissionRequest
	<ul style="list-style-type: none"> tr:ImportEvidence <div data-bbox="842 257 1329 448"> </div> <p>Mit der Übergabe des dargestellten ImportEvidence-Elementes <u>kann</u> der Import von einem oder mehreren zu einer bestimmten XAIP- bzw. LXAIP-Version bzw. zu den übergebenen Binärdaten gehörenden Evidence Records gemäß [RFC4998] oder [RFC6283]⁷ 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. 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 Prüfungsschritte⁸, wobei die jeweiligen Zertifikate der Zeitstempel vollständig bis hin zu einer vertrauenswürdigen Wurzel oder Vertrauensanker gemäß der vom [TR-ESOR-PEPT] abgeleiteten und veröffentlichten Preservation Policy (PEP) des TR-ESOR-Produktes bzw. Bewahrungsdienstes 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 <u>soll</u>.</p> <p>Hierbei kann es sich entweder ein XAIP (siehe Abs. 3.1 in [TR-ESOR-F]) oder ein LXAIP (siehe Abs. 3.2 in [TR-ESOR-F]) handeln.</p>
ArchiveData	<p>Enthält ein in einem beliebigen anderen Format vorliegendes Archivdatenobjekt. Der hierfür</p>

⁷[RFC4998] muss, [RFC6283] und [TR-ESOR-ERS] können unterstützt werden.

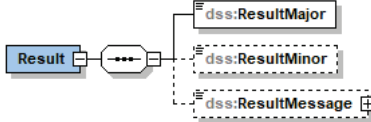
⁸Siehe Abschnitt 3.3 in [RFC4998] und Abschnitt 2.3 in [RFC6283] sowie [TR-ESOR-ERS].

<i>Name</i>	<i>ArchiveSubmissionRequest</i>
	<p>genutzte <code>ArchiveDataType</code> ist als <code>anyType</code> mit einem optionalen <code>Type</code>-Attribut definiert.</p> <p>Durch das <code>Type</code>-Attribut <code>http://uri.etsi.org/ades/ASiC/type/ASiC-ERS</code> wird klargestellt, dass es sich um einen <code>base64Binary</code>-codierten⁹ <code>ASiC-AIP-Container</code> gemäß Abs. 3.3 in [TR-ESOR-F] handelt.</p> <p>Darüber hinaus zugelassene binäre Datentypen mit dem zugehörigen Wert für das <code>Type</code>-Attribut sind dem HINWEIS 5 zu entnehmen.</p> <p>Weitere Übergabetypen <u>können</u> im Rahmen einer Profilierung der vorliegenden Spezifikation spezifiziert werden.</p>

3.1.2 Ausgabeparameter: ArchiveSubmissionResponse

<i>Name</i>	<i>ArchiveSubmissionResponse</i>				
Beschreibung	Als Antwort auf einen <code>ArchiveSubmissionRequest</code> wird ein entsprechendes <code>ArchiveSubmissionResponse</code> -Element zurückgeliefert, das im Erfolgsfall einen eindeutigen Identifikator des Archivdatenobjektes, die <code>AOID</code> , enthält.				
Details	<p>Der Ausgabeparameter <code>ArchiveSubmissionResponse</code> ist die Antwort zum Eingabeparameter <code>ArchiveSubmissionRequest</code> und weist folgenden Aufbau</p>  <pre> classDiagram class ArchiveSubmissionResponse class trResponseType["tr:ResponseType (extension)"] class RequestID class Profile class dssResult["dss:Result"] class dssOptionalOutputs["dss:OptionalOutputs"] class trAOID["tr:AOID"] ArchiveSubmissionResponse -- > trResponseType trResponseType --> RequestID trResponseType --> Profile trResponseType --> dssResult trResponseType --> dssOptionalOutputs trResponseType --> trAOID </pre> <table border="1"> <thead> <tr> <th>Name</th><th>Beschreibung</th></tr> </thead> <tbody> <tr> <td><code>dss:Result</code></td><td>Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in [eCard-1] und unten näher beschrieben.</td></tr> </tbody> </table>	Name	Beschreibung	<code>dss:Result</code>	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in [eCard-1] und unten näher beschrieben.
Name	Beschreibung				
<code>dss:Result</code>	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in [eCard-1] und unten näher beschrieben.				

⁹Siehe <https://www.w3.org/TR/xmlschema-2/#base64Binary>.

<i>Name</i>	<i>ArchiveSubmissionResponse</i>	
	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 AOID¹⁰ vom aufgerufenen Modul erzeugt oder ergänzt wurde, vorhanden sein und für zukünftige Zugriffe auf das Archivdatenobjekt genutzt werden.</p>
 <p>Statusinformationen und Fehler bei ArchiveSubmissionResponse (vgl. [eCard-1] Abschnitt 4.1 und 4.2).</p>		
	Name	Fehlercode
	ResultMajor	/resultmajor#ok /resultmajor#error /resultmajor#warning
	ResultMinor	/resultminor/al/common#noPermission /resultminor/al/common#internalError /resultminor/al/common#parameterError /resultminor/arl/lowSpaceWarning /resultminor/arl/noSpaceError /resultminor/arl/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 /resultminor/sal#invalidSignature

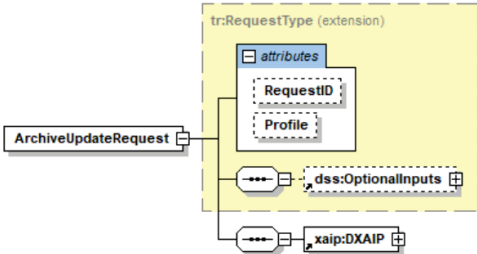
¹⁰Die AOID (Archive Object Identifier) im vorliegenden Dokument entspricht dem POID (Preservation Object Identifier) aus [ETSI TS 119 512].

3.2 Funktion: ArchiveUpdate

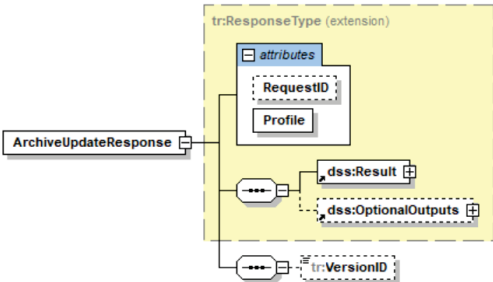
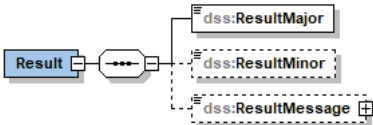
Mit dem Funktionseingabeparameter `ArchiveUpdateRequest` wird eine neue Version für ein bereits abgelegtes Archivdatenobjekt erzeugt. Hierbei werden die bereits abgelegten Daten nicht verändert, sondern es wird lediglich zusätzlich eine neue Version hinzugefügt.

Wie in Abbildung 3 und Abbildung 4 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in TR-S.2 (vgl. Abs. 5.2) und TR-S.6 (vgl. Abs. 5.5) genutzt.

3.2.1 Eingabeparameter: ArchiveUpdateRequest

Name	ArchiveUpdateRequest	
Beschreibung	Durch den Aufruf der Funktion <code>ArchiveUpdate</code> wird eine neue Version für ein bereits abgelegtes Archivdatenobjekt erzeugt (vgl. [TR-ESOR-M.1]). Die Beschreibung der neuen Version wird dabei mit Hilfe des Eingabeparameters <code>ArchiveUpdateRequest</code> vorgegeben.	
Details	<p>Der Eingabeparameter <code>ArchiveUpdateRequest</code> weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</p> 	
	Name	Beschreibung
	<code>dss:OptionalInputs</code>	<p>Ist für optionale Eingabeelemente vorgesehen.</p> <p>(A3.2.1-1) : Gemäß der vorliegenden Spezifikation <u>sollen</u> hier die unter (A3.1.1-1) spezifizierten optionalen Eingabeelemente <code>AOID</code>, <code>ReturnVerificationReport</code> und <code>ImportEvidence</code> unterstützt werden.</p>
	<code>xaip:DXAIP</code>	<p>Enthält ein ergänzendes XML-basiertes Archivdatenobjekt (Delta-XAIP) gemäß ([TR-ESOR-F], Abs. 3.1.6) bzw. (Delta-LXAIP) gemäß ([TR-ESOR-F], Abs. 3.2.2) das ein neues <code>versionManifest</code>, die Vorgängerversion, Verweise auf unverändert aus dieser übernommene Objekte und die zu ergänzenden Elemente enthält, die in einer neuen Version eines bereits abgelegten Archivdatenobjektes ergänzt werden sollen.</p>

3.2.2 Ausgabeparameter: 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.	
Details	Der Ausgabeparameter ArchiveUpdateResponse weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.	
		
	Name	Beschreibung
	dss:Result	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in [eCard-1] und unten näher beschrieben.
	dss:OptionalOutputs	Ist für optionale Ausgabeelemente vorgesehen. (A3.2.2-1) : Gemäß der vorliegenden Spezifikation <u>kann</u> das folgende Element auftreten: <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.
VersionID	Ist im Erfolgsfall vorhanden und enthält den bezüglich des über die AOID identifizierten Archivdatenobjektes eindeutigen Versions-Identifikator. Die VersionID <u>soll</u> in der Form v1, v2, ... vx gebildet werden.	
		
Statusinformationen und Fehler bei ArchiveUpdateResponse (vgl. [eCard-1] Abs. 4.1 und Abs. 4.2).		

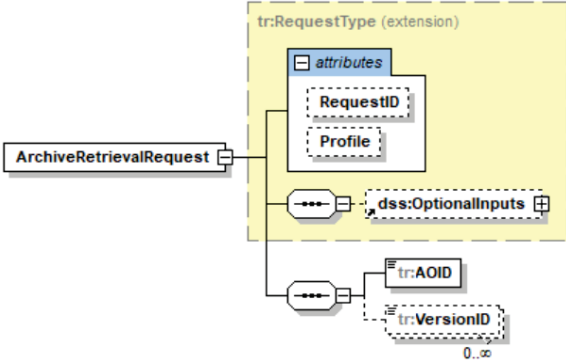
<i>Name</i>	<i>ArchiveUpdateResponse</i>	
	Name	Fehlercode
	ResultMajor	<ul style="list-style-type: none"> • /resultmajor#ok • /resultmajor#error • /resultmajor#warning
	ResultMinor	<ul style="list-style-type: none"> • /resultminor/al/common#noPermission • /resultminor/al/common#internalError • /resultminor/al/common#parameterError • /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 SUBMTIME • /resultminor/arl/DXAIP NOK SIG • /resultminor/arl/XAIP NOK ER • /resultminor/arl/DXAIP NOK ID • /resultminor/arl/DXAIP NOK Version • /resultminor/sal#invalidSignature

3.3 Funktion: ArchiveRetrieval

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

Wie in Abbildung 3 und Abbildung 4 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 in ähnlicher Weise auch in den Schnittstellen TR-S.2 (vgl. Abs. 5.2) und TR-S.5 (vgl. Abs. 5.4) genutzt.

3.3.1 Eingabeparameter: ArchiveRetrievalRequest

<i>Name</i>	<i>ArchiveRetrievalRequest</i>		
Beschreibung	<p>Durch den Aufruf der Funktion <code>ArchiveRetrieval</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)“¹¹ genutzt werden.</p>		
Details	<p>Der Eingabeparameter <code>ArchiveRetrievalRequest</code> weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</p>  <table border="1" data-bbox="432 969 1380 1010"> <thead> <tr> <th data-bbox="432 969 758 1010">Name</th><th data-bbox="758 969 1380 1010">Beschreibung</th></tr> </thead> </table>	Name	Beschreibung
Name	Beschreibung		

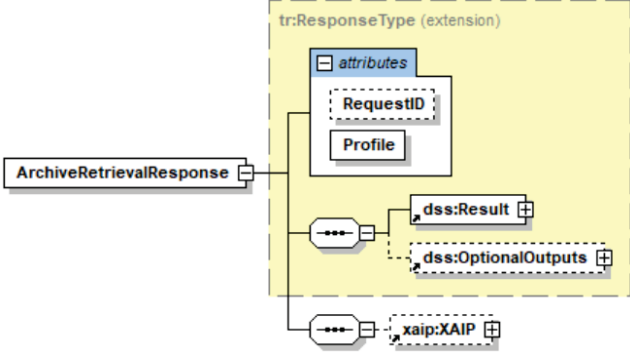
¹¹Siehe <https://www.w3.org/TR/soap12-mtom/>.

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"> • pres:POFormat • tr:IncludeERS. <p>pres:POFormat¹² – gibt das AIP-Format an, wobei folgende Formate definiert sind:</p> <div data-bbox="1002 568 1145 613" style="border: 1px solid black; padding: 2px; display: inline-block;">POFormat</div> <ul style="list-style-type: none"> • http://www.bsi.bund.de/tr-esor/xaip/1.3 – für ein XAIP gem. Abs. 3.1 in [TR-ESOR-F], • http://www.bsi.bund.de/tr-esor/lxaip/1.3 – für ein „logisches XAIP“ gem. Abs. 3.2 in [TR-ESOR-F], • http://uri.etsi.org/ades/ASiC/type/ASiC-ERS – für einen base64Binary-codierten ASiC-AIP-Container gem. Abs. 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>Sollte das Element POFormat ausgelassen werden, so ist http://www.bsi.bund.de/tr-esor/xaip/1.3 standardmäßig gesetzt.</p> <p>tr: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 28)</p> <div data-bbox="975 1451 1129 1496" style="border: 1px solid black; padding: 2px; display: inline-block;">ERSFormat</div> <p>enthalten <u>soll</u>.</p> <p>Dieser bzw. diese Evidence Record(s) wird bzw. werden bei XAIP bzw. LXAIP im dafür vorgesehenen</p> <p>xaip:credential/xaip:EvidenceRecord Element oder im Fall ASiC-AIP im ASiC-AIP-Container gem. Abs. 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>

¹²Das POFormat-Element ist in [ETSI TS 119 512] folgendermaßen definiert:
 <element name="POFormat" type="anyURI"/>.

Name	ArchiveRetrievalRequest	
		<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> <p>Umgekehrt <u>muss</u> auf die vom Evidence Record geschützten Datenobjekte im relatedObjects-Attribut des xaip:credential-Elementes verwiesen werden.</p>
	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 Ausgabeparameter: ArchiveRetrievalResponse

Name	ArchiveRetrievalResponse	
Beschreibung	<p>Als Antwort auf einen ArchiveRetrievalRequest wird ein entsprechendes ArchiveRetrievalResponse-Element zurückgeliefert, welches im Erfolgsfall das angeforderte Archivdatenobjekt (L)XAIP im xaip:XAIP-Format gem. [TR-ESOR-F] oder in dem po-element gem. [ETSI TS 119 512] (als ein base64Binary-codierter ASiC-E-Container gem. Abs. 3.3 in [TR-ESOR-F]) enthält.</p>	
Details	<p>Der Ausgabeparameter ArchiveRetrievalResponse weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</p> 	
	Name	Beschreibung

<i>Name</i>	<i>ArchiveRetrievalResponse</i>	
	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-codiertes ASiC-AIP gemäß Abs. 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>
	<div data-bbox="715 1131 1088 1254" data-label="Diagram"> <pre> graph LR Result[Result] --- dssResultMajor[dss:ResultMajor] Result --- dssResultMinor[dss:ResultMinor] Result --- dssResultMessage[dss:ResultMessage] </pre> </div> <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
	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 • /resultminor/arl/unknownVersionID¹³ • /resultminor/arl/unknownPOFormat

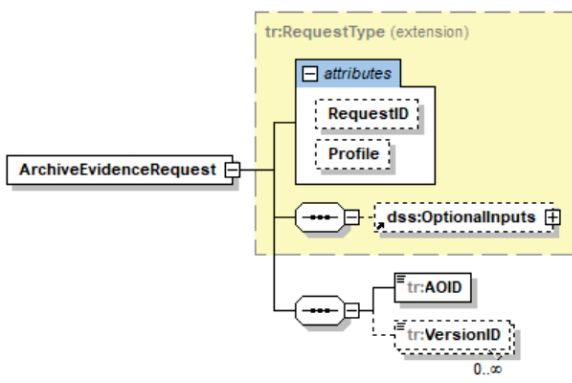
¹³ Im ResultMessage-Element soll die problematische VersionID zurückgeliefert werden.

3.4 Funktion: ArchiveEvidence

Mit dem Funktionseingabeparameter `ArchiveEvidenceRequest` werden die zugehörigen technischen Beweisdaten (Evidence Records gemäß [RFC4998] oder [RFC6283]¹⁴ oder mit der Profilierung aus [TR-ESOR-ERS]) für ein beweiswerterhaltend aufbewahrtes und über ein `AOID`-Element adressiertes Archivdatenobjekt zurückgeliefert.

Wie in Abbildung 3 und Abbildung 4 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in TR-S.6 (vgl. Abs. 5.5) genutzt.

3.4.1 Eingabeparameter: ArchiveEvidenceRequest

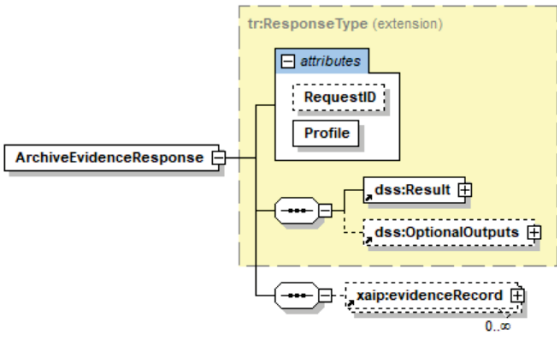
<i>Name</i>	<i>ArchiveEvidenceRequest</i>	
Beschreibung	Durch den Aufruf der Funktion <code>ArchiveEvidence</code> können für ein beweiswerterhaltend abgelegtes Archivdatenobjekt technische Beweisdaten in Form von Evidence Records gemäß [RFC4998] oder [RFC6283] ¹⁵ oder in der Profilierung gem. [TR-ESOR-ERS] angefordert werden.	
Details	<p>Der Eingabeparameter <code>ArchiveEvidenceRequest</code> weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</p> 	
	Name	Beschreibung

¹⁴[RFC4998] muss, [RFC6283] kann unterstützt werden.

¹⁵[RFC4998] muss, [RFC6283] kann unterstützt werden.

<i>Name</i>	<i>ArchiveEvidenceRequest</i>
dss:OptionalInputs	<p>Ist für optionale Eingabeelemente vorgesehen.</p> <p>(A3.4.1-1): Gemäß der vorliegenden Spezifikation <u>soll</u> das folgende Element unterstützt werden:</p> <div data-bbox="981 385 1141 430" style="border: 1px solid black; padding: 2px; display: inline-block;">ERSFormat</div> <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 gem. [RFC4998] oder • urn:ietf:rfc:6283 für XML-basierte Evidence Records gem. [RFC6283] oder • http://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03125/BSI_TR_03125_Anlage_ERS_V1_2.html#Basis-ERS-Profil gem. [TR-ESOR-ERS] oder • http://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03125/BSI_TR_03125_Anlage_ERS_V1_2.html#Basis-XERS-Profil gem. [TR-ESOR-ERS]. <p>Fehlt das <code>ERSFormat</code>-Element, so werden ASN.1-basierte Evidence Records gemäß [RFC4998] in der Profilierung Basis-ERS-Profile gem. [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 <code>AOID</code> 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 Ausgabeparameter: ArchiveEvidenceResponse

<i>Name</i>	<i>ArchiveEvidenceResponse</i>			
Beschreibung	Als Antwort auf einen ArchiveEvidenceRequest wird ein entsprechendes ArchiveEvidenceResponse-Element zurückgeliefert, das die angeforderten Beweisdaten enthält.			
Details	Der Ausgabeparameter ArchiveRetrievalResponse weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.			
				
	<table><tr><th>Name</th><th>Beschreibung</th></tr></table>	Name	Beschreibung	
	Name	Beschreibung		
	<table><tr><td>dss:Result</td><td>Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in Abs. 4.1.2 von [eCard-1] und unten näher beschrieben. Sofern nicht für alle mittels der übergebenen AOID adressierten Archivdatenobjekte entsprechende Beweisdaten (Evidence Records) zurückgeliefert werden konnten, wird dies durch die .../resultminor/arl/requestOnlyPartlySuccessfulWarning angezeigt.</td></tr></table>	dss:Result	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in Abs. 4.1.2 von [eCard-1] und unten näher beschrieben. Sofern nicht für alle mittels der übergebenen AOID adressierten Archivdatenobjekte entsprechende Beweisdaten (Evidence Records) zurückgeliefert werden konnten, wird dies durch die .../resultminor/arl/requestOnlyPartlySuccessfulWarning angezeigt.	
dss:Result	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in Abs. 4.1.2 von [eCard-1] und unten näher beschrieben. Sofern nicht für alle mittels der übergebenen AOID adressierten Archivdatenobjekte entsprechende Beweisdaten (Evidence Records) zurückgeliefert werden konnten, wird dies durch die .../resultminor/arl/requestOnlyPartlySuccessfulWarning angezeigt.			
<table><tr><td>dss:OptionalOutputs</td><td>Ist für optionale Ausgabeelemente vorgesehen und kann beispielsweise entsprechende Steuerelemente (responseControls) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen.</td></tr><tr><td>xaip:evidenceRecord</td><td>Sofern vom ArchiSig-Modul entsprechende Evidence Records¹⁶ gemäß [RFC4998] bzw. [RFC6283] oder [TR-ESOR-ERS] konstruiert werden können, werden diese zurückgeliefert. Die detaillierte Struktur dieses Elementes ist nachfolgend erläutert.</td></tr></table>	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 ¹⁶ gemäß [RFC4998] bzw. [RFC6283] oder [TR-ESOR-ERS] konstruiert werden können, werden diese zurückgeliefert. Die detaillierte Struktur dieses Elementes ist nachfolgend erläutert.
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 ¹⁶ gemäß [RFC4998] bzw. [RFC6283] oder [TR-ESOR-ERS] konstruiert werden können, werden diese zurückgeliefert. Die detaillierte Struktur dieses Elementes ist nachfolgend erläutert.			

¹⁶ Sofern die TR-ESOR-Middleware mehrere redundante Hashbäume pflegt, werden hier mehrere Evidence Records zurückgeliefert.

Name	ArchiveEvidenceResponse
	<div data-bbox="635 197 1189 488"> </div> <p>Das xaip:evidenceRecord-Element gemäß [TR-ESOR-F] ist vom Typ xaip:EvidenceRecordType, der entsprechend den Evidence Record beinhaltet 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].
	<div data-bbox="726 1093 1101 1227"> </div> <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
ResultMinor	<ul style="list-style-type: none"> • /resultminor/al/common#noPermission • /resultminor/al/common#internalError • /resultminor/al/common#parameterError • /resultminor/arl/notSupported¹⁷ • /resultminor/arl/unknownAOID • /resultminor/arl/unknownVersionID/ • resultminor/arl/requestOnlyPartlySuccessfulWarning

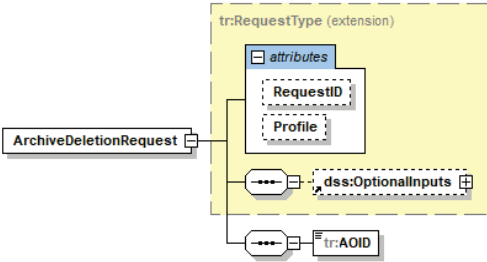
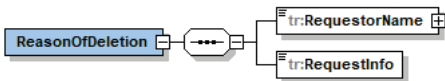
¹⁷Im ResultMessage-Element sollen nähere Informationen darüber zurückgeliefert werden, welche angeforderte Funktionalität nicht unterstützt wird.

3.5 Funktion: ArchiveDeletion

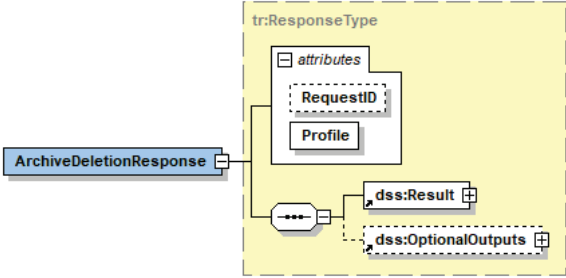
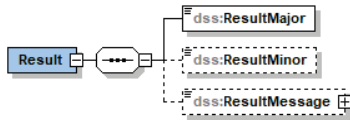
Mit dem Funktionseingabeparameter `ArchiveDeletionRequest` wird ein Archivdatenobjekt (inklusive aller zugehörigen Versionen und im Fall eines LXAIPs auch inklusive aller dort referenzierten Nutzdaten) über die TR-ESOR-Middleware aus dem ECM-/Langzeitspeichersystem gelöscht.

Wie in Abbildung 3 und Abbildung 4 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in der Schnittstelle TR-S.5 (vgl. Abs. 5.4) genutzt.

3.5.1 Eingabeparameter: ArchiveDeletionRequest

Name	ArchiveDeletionRequest	
Beschreibung	Durch den Aufruf der Funktion <code>ArchiveDeletion</code> kann ein im Langzeitspeicher abgelegtes Archivdatenobjekt (z.B. XAIP oder LXAIP oder ASiC-AIP oder die in [TR-ESOR-F], HINWEIS 5 aufgezählten Binärdaten), inklusive aller dazugehörigen Versionen und referenzierten Nutzdaten, gelöscht werden.	
Details	<p>Der Eingabeparameter <code>ArchiveDeletionRequest</code> weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</p> 	
	Name	Beschreibung
	<code>dss:OptionalInputs</code>	<p>Ist für optionale Eingabeelemente vorgesehen. Insbesondere bei einer vorzeitigen Löschung <u>muss</u> das folgende Element <code>ReasonOfDeletion</code> genutzt und unterstützt werden:</p> <p>(A3.5.1-1): Das <code>ReasonOfDeletion</code>-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 <code>RequestorName</code> <u>soll</u> mit den verwendeten Authentisierungsinformationen abgeglichen werden.</p> 
	<code>AOID</code>	Das <code>AOID</code> -Element gibt an, welches Archivdatenobjekt gelöscht werden soll.

3.5.2 Ausgabeparameter: 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.	
Details	<p>Der Ausgabeparameter ArchiveDeletionResponse weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</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	Ist für optionale Ausgabeelemente vorgesehen und kann beispielsweise entsprechende Steuerelemente (responseControls) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden
	 <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

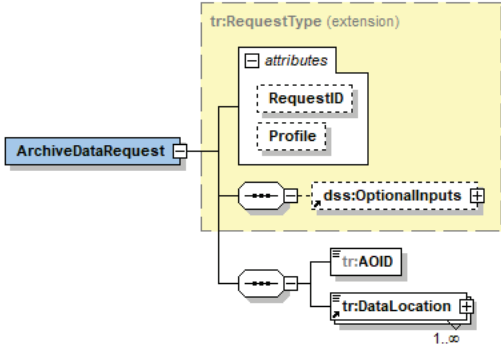
3.6 Funktion: ArchiveData

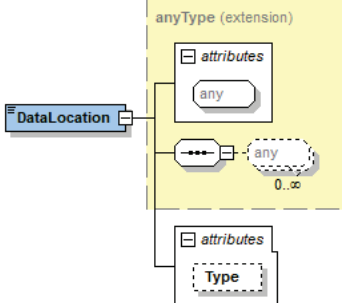

Mit dem Funktionseingabeparameter ArchiveDataRequest können diskrete Datenelemente aus einem bereits abgelegten Archivdatenobjekt (xaip:XAIP) ausgelesen werden.

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

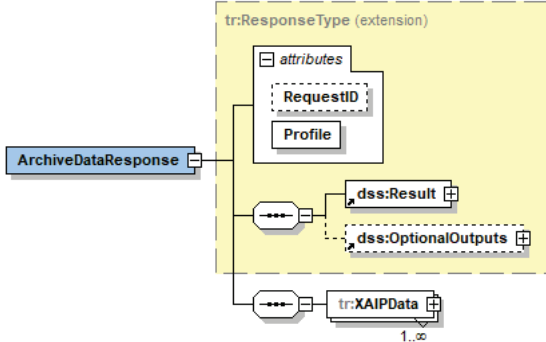
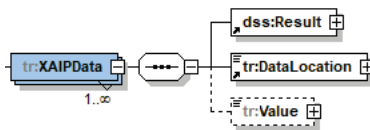
3.6.1 Eingabeparameter: ArchiveDataRequest

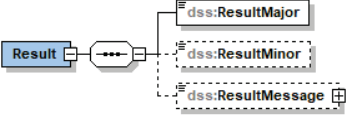
<i>Name</i>	<i>ArchiveDataRequest</i>	
Beschreibung	Mit dem Aufruf der Funktion <code>ArchiveData</code> können diskrete Datenelemente aus einem im zuvor abgelegten Archivinformationspaket (vgl. Abs. 3.1) ausgelesen werden. Die Archivdaten-Container müssen dabei als XAIP oder LXAIP gem. dieser Spezifikation vorliegen.	
Details	<p>Der Eingabeparameter <code>ArchiveDataRequest</code> weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</p> 	
	Name	Beschreibung
	<code>dss:OptionalInputs</code>	<p>Ist für optionale Eingabeelemente vorgesehen und <u>kann</u> beispielsweise Steuerelemente (<code>requestControls</code>) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden <u>sollen</u>.</p> <p>Die vorliegende Spezifikation definiert keine solchen optionalen Eingabeelemente.</p>
	<code>AOID</code>	Dieses Element enthält den Identifikator eines bestimmten Archivdatenobjektes.

Name	ArchiveDataRequest
tr:DataLocation	<p>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]¹⁸ vorliegenden Archivdatenobjektes.</p> <p>Die detaillierte Ausgestaltung der hier unterstützen Funktionalität bleibt dem Hersteller überlassen.</p> <p>(A3.6.1-1) : Sofern der ArchiveDataRequest unterstützt wird, <u>muss</u> dieser die Details der an der Schnittstelle angebotenen Funktionalität dokumentieren.</p>
	 <p>Das DataLocation-Element spezifiziert, welche Teile eines Archivobjektes zurückgeliefert werden sollen und ist folgendermaßen definiert:</p> <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. <p>Der zugehörige XPATH-Ausdruck ist in das XPathFilter-Element abzulegen und als Wert des DataLocation-Element zu übergeben.</p> 

¹⁸Im Falle eines XML-basierten Archivinformationspakets sind die folgenden diskreten Adressierung von XML Datenelementen möglich: XPath (siehe <http://www.w3.org/TR/2007/REC-xpath20-20070123/>).

3.6.2 Ausgabeparameter: ArchiveDataResponse

Name	ArchiveDataResponse								
Beschreibung	Als Antwort auf einen ArchiveDataRequest wird ein entsprechendes ArchiveDataResponse-Element zurückgeliefert, das die gewünschten Informationen enthält.								
Details									
	Der Ausgabeparameter ArchiveDataResponse weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.								
	<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 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><tr><td>dss:OptionalOutputs</td><td>Ist für optionale Ausgabeelemente vorgesehen und kann beispielsweise entsprechende Steuerelemente (responseControls) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen.</td></tr><tr><td>XAIPData</td><td>Enthält im Erfolgsfall die gewünschten Daten und die „Lokation“, aus der diese aus der im ECM-/Langzeitspeichersystem zumindest logisch existierenden XAIP- bzw. LXAIP-Struktur ausgelesen wurden. Die detaillierte Struktur dieses Elementes ist nachfolgend dargestellt und erläutert.</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. Sofern nur ein Teil der angefragten diskreten Datenobjekte zurückgeliefert werden konnte, wird dies durch den Fehlercode .../resultminor/arl/requestOnlyPartlySuccessfulWarning angezeigt.	dss:OptionalOutputs	Ist für optionale Ausgabeelemente vorgesehen und kann beispielsweise entsprechende Steuerelemente (responseControls) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen.	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.
	Name	Beschreibung							
	dss:Result	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in [eCard-1] und unten näher beschrieben. Sofern nur ein Teil der angefragten diskreten Datenobjekte zurückgeliefert werden konnte, wird dies durch den Fehlercode .../resultminor/arl/requestOnlyPartlySuccessfulWarning angezeigt.							
dss:OptionalOutputs	Ist für optionale Ausgabeelemente vorgesehen und kann beispielsweise entsprechende Steuerelemente (responseControls) enthalten, die im Rahmen einer Profilierung der vorliegenden Spezifikation definiert werden sollen.								
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.								
									
Das XAIPData-Element enthält im Erfolgsfall die gewünschten Daten.									
<table><tr><th>Name</th><th>Beschreibung</th></tr></table>	Name	Beschreibung	<table><tr><th>Name</th><th>Beschreibung</th></tr></table>	Name	Beschreibung				
Name	Beschreibung								
Name	Beschreibung								

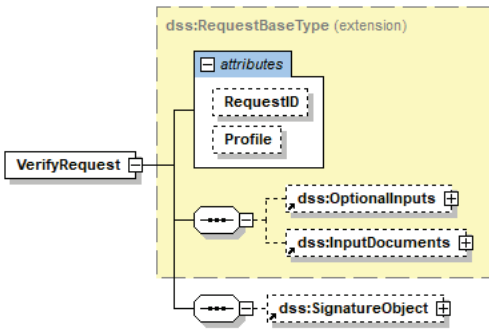
Name	ArchiveDataResponse	
	dss:Result	<p>Gibt an, ob die Anfrage erfolgreich durchgeführt werden konnte oder nicht.</p> <p>Als ResultMajor sind die beiden folgenden Werte möglich:</p> <ul style="list-style-type: none"> • ../resultmajor#ok • ../resultmajor#error <p>Als ResultMinor sind die folgenden Werte möglich:</p> <ul style="list-style-type: none"> • ../resultminor/ar/unknownLocation • ../resultminor/al/common#parameterError • ../resultminor/al/common#internalError
	tr:DataLocation	<p>Das DataLocation-Element spezifiziert, welche Teile eines Archivobjektes zurückgeliefert werden. Die detaillierte Ausgestaltung dieses Parameters ist der Seite 33 zu entnehmen.</p>
	Value	<p>Enthält im Erfolgsfall die gewünschten Daten.</p>
 <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/ar/unknownAOID • /resultminor/ar/unsupported • /resultminor/ar/requestOnlyPartlySuccessfulWarning

3.7 Funktion: Verify

Mit dem Funktionseingabeparameter `VerifyRequest` werden XML-basierte Archivinformationspakete (XAIP), logische XAIP (LXAIP) oder ASiC-AIP-basierte Datencontainer oder binäre Daten gemäß HINWEIS 5 oder optional XML-basiertes Delta- Archivinformationspakete (Delta-(L)XAIP) und Beweisdaten (Evidence Records) sowie den darin enthaltenen oder zusätzlich übergebenen beweisrelevanten Daten (Signaturen, Siegel, Zeitstempel, Zertifikate, Sperrlisten, OCSP-Responses etc.) geprüft.

Wie in Abbildung 3 und Abbildung 4 ersichtlich, wird diese Funktion neben der hier betrachteten Schnittstelle TR-S.4 auch in TR-S.1 (vgl. Abs. 5.1) genutzt.

3.7.1 Eingabeparameter: VerifyRequest

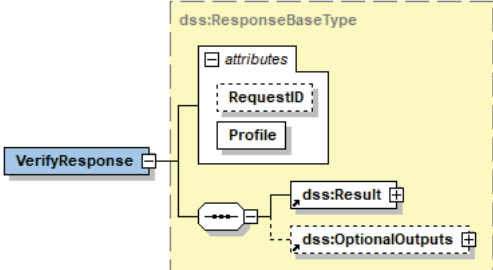
Name	VerifyRequest
Beschreibung	Mit der Funktion <code>VerifyRequest</code> (vgl. Abs. 3.2.2 von [eCard-2]) werden XML-basierte Archivinformationspakete (XAIP), logische XAIP oder ASiC-AIP-basierte Datencontainer oder optional XML-basiertes Archivinformationspakete Delta-L(XAIP), mit den darin enthaltenen beweisrelevanten Daten (Signaturen, Siegel, Zeitstempel, Zertifikate, Sperrlisten, OCSP-Responses etc.), und ebenfalls darin enthalten oder zusätzlich übergebenen Beweisdaten (Evidence Records), oder zusätzlich übergebenen beweisrelevanten Daten (Signaturen, Siegel, Zeitstempel, Zertifikate, Sperrlisten, OCSP-Responses etc.) geprüft.
Details	<p>Der Eingabeparameter <code>VerifyRequest</code> weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</p> 
Name	Beschreibung
<code>dss:OptionalInputs</code>	<p>Das <code>OptionalInputs</code>-Element <u>kann</u> zusätzliche Eingabeelemente enthalten.</p> <p>(A3.7.1-1) : Hierbei werden insbesondere die in [eCard-2] definierten Elemente und Aufrufoptionen unterstützt.</p> <p>Dies umfasst insbesondere die folgenden Elemente:</p> <ul style="list-style-type: none"> • <code>VerifyUnderSignaturePolicy</code> <u>soll</u> unterstützt werden, • <code>ReturnVerificationReport</code> <u>muss</u> unterstützt werden. <p>Es gilt im Einzelnen:</p> <ul style="list-style-type: none"> • <code>VerifyUnderSignaturePolicy</code> Sofern in einem <code>dss:Document/InlineXML</code>-Kindelement von <code>dss:InputDocuments</code> ein XAIP-Element in Form eines gewöhnlichen XAIP oder eines logischen XAIP gemäß [TR-ESOR-F] enthalten ist, kann mit dem Element <code>VerifyUnderSignaturePolicy</code> und der im <code>DefaultPolicy/SignaturePolicyIdentifier</code>-Element angegebenen Signature-Policy: <ul style="list-style-type: none"> ◦ http://www.bsi.bund.de/DE/tr-esor/sigpolicy/verify-xaip oder

Name	VerifyRequest
	<ul style="list-style-type: none"> ◦ http://www.bsi.bund.de/DE/tr-esor/sigpolicy/verify-xaip/shell oder ◦ http://www.bsi.bund.de/DE/tr-esor/sigpolicy/verify-xaip/chain <p>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 oder Vertrauensanker gemäß der vom [TR-ESOR-PEPT] abgeleiteten und veröffentlichten Preservation Policy (PEP) des TR-ESOR-Produktes bzw. Bewahrungsdienstes geprüft werden.</p> <p>Die hierbei ermittelten Prüfinformationen (Zertifikate, Sperrlisten, OCSP-Responses) <u>müssen</u> nach Möglichkeit als unsignierte Attribute bzw. Properties in den entsprechenden digitalen Signaturen bzw. in den Kind-Elementen certificateValues bzw. revocationValues des credential-Elementes abgelegt werden.</p> <p>Wenn sowohl die Signature-Policy:</p> <ul style="list-style-type: none"> ◦ http://www.bsi.bund.de/DE/tr-esor/sigpolicy/verify-xaip oder ◦ http://www.bsi.bund.de/DE/tr-esor/sigpolicy/verify-xaip/shell oder ◦ http://www.bsi.bund.de/DE/tr-esor/sigpolicy/verify-xaip/chain <p>als auch das Element ReturnVerificationReport übergeben werden, 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 credentialsSection des übergebenen XAIP-, LXAIP- oder Delta-(L)XAIP oder ASiC-AIP-Containers ein oder mehrere xaip:EvidenceRecord-Elemente gemäß [TR-ESOR-F] enthalten sind, <u>müssen</u> diese entsprechend geprüft werden.</p>

Name	VerifyRequest	
		<p>Die hierbei ermittelten Prüfinformationen (Zertifikate, Sperrlisten, OCSP-Responses) <u>müssen</u> nach Möglichkeit als unsignierte Attribute bzw. Properties in den entsprechenden digitalen Signaturen bzw. in den Kindelementen certificateValues bzw. revocationValues des credential-Elementes mit Bezug auf den entsprechenden Evidence Record abgelegt werden.</p> <ul style="list-style-type: none"> ReturnVerificationReport Durch die Übergabe eines ReturnVerificationReport-Elementes gemäß [OASIS VR] bzw. [eCard-2] und [TR-ESOR-VR] <u>kann</u> ein ausführlicher Prüfbericht in Form eines VerificationReport-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 VerifyResponse-Elements nur das erzeugte VerificationReport-Element zurück zu geben.
	dss:InputDocuments	<p>Das dss:InputDocuments-Element enthält die zur Prüfung benötigten Dokumente, sofern diese nicht bereits im unten erläuterten SignatureObject-Element enthalten sind.</p> <p>Außerdem <u>kann</u> in einem dss:Document/InlineXML-Kindelement ein XAIP-Element mit einem XAIP gemäß [TR-ESOR-F] (Abs. 3.1) oder einem LXAIP-Element gemäß [TR-ESOR-F] (Abs. 3.2) bzw. in einem dss:Document/dss:Base64Data-Kindelement ein ASiC-AIP gemäß [TR-ESOR-F] (Abs. 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>
	dss:SignatureObject	<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- <u>muss</u> als Kindelement von dss:InputDocuments/dss:Document/dss:InlineXML und im Falle von ASiC-AIP-Element <u>muss</u> als Kindelement von</p>

Name	VerifyRequest	
		<p>dss:InputDocuments/dss:Document /dss:Base64Data 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 Ausgabeparameter: VerifyResponse

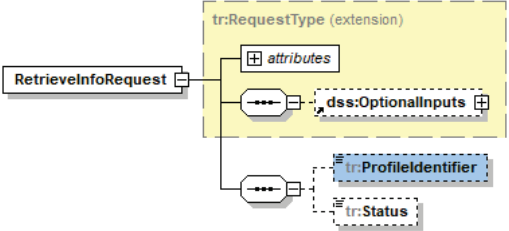
Name	VerifyResponse	
Beschreibung	Als Antwort auf einen VerifyRequest wird ein entsprechendes VerifyResponse-Element gemäß Abs. 3.2.2 von [eCard-2] zurückgeliefert.	
Details	<p>Der Ausgabeparameter ArchiveDataResponse weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</p> 	
	Name	Beschreibung
	dss:Result	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements und die möglichen Fehlercodes sind in Abs. 4.1.2 von [eCard-1] und Abs. 3.2.2 von [eCard-2] beschrieben.
	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 XAIP- bzw. LXAIP-Container finden sich in [TR-ESOR-F].</p>

3.8 Funktion: RetrieveInfo

Mit dem Aufruf der Funktion `RetrieveInfo` ist es möglich, die in Form eines Profils verfasste Beschreibung der Fähigkeiten des Bewahrungsproduktes bzw. -dienstes zu erfragen. Da es im Laufe der Zeit dazu kommen kann, dass ein Bewahrungsdienst bzw. -produkt mehrere solcher Profile unterstützt, ist es mit Hilfe entsprechender Parametrisierung der Funktionseingabe (`RetrieveInfoRequest`) möglich, nach gewünschten Profilen zu filtern. Die durch das Bewahrungsprodukt bzw. den Bewahrungsdienst ermittelten Ergebnisse werden mit Hilfe des Ausgabeparameters `RetrieveInfoResponse` zurückgeliefert.

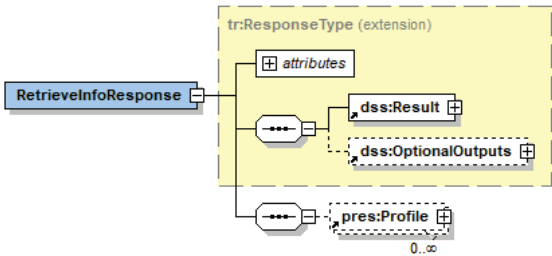
Wie in Abbildung 3 und Abbildung 4 ersichtlich, wird diese Funktion ausschließlich an der hier betrachteten Schnittstelle TR-S.4 angeboten.

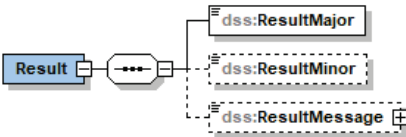
3.8.1 Eingabeparameter: RetrieveInfoRequest

Name	RetrieveInfoRequest	
Beschreibung	Mit dem Eingabeparameter <code>RetrieveInfoRequest</code> wird beim Aufruf der Funktion <code>RetrieveInfo</code> vorgegeben, nach welchen Profilen eines Bewahrungsprodukts bzw. -dienstes gesucht wird.	
Details	<p>Der Eingabeparameter <code>RetrieveInfoRequest</code> weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</p> 	
	Name	Beschreibung
	<code>dss:OptionalInputs</code>	Es werden standardmäßig <u>keine</u> optionalen Eingabeelemente unterstützt.
	<code>tr:ProfileIdentifier</code>	<p>Mit Hilfe dieses Parameters <u>kann</u> ein durch die Eingabe einer URI (gem. [RFC3986]) spezifizierte Profile gezielt angefragt werden.</p> <p>Gegenwärtig werden durch diese TR-ESOR-Spezifikation V1.3 folgende URIs unterstützt:</p> <ul style="list-style-type: none"> http://www.bsi.bund.de/tr-esor/V1.3.0/profile/S.4/V1.0 <p>oder</p> <ul style="list-style-type: none"> http://www.bsi.bund.de/tr-esor/V1.3.0/profile/preservation-api/v1.1.2 - Verweis auf die aktuelle TR-ESOR-S.512-Schnittstelle

<i>Name</i>	<i>RetrieveInfoRequest</i>	
	tr:Status	<p>Mit Hilfe dieses Parameters <u>kann</u> zwischen:</p> <ul style="list-style-type: none"> • aktiven (Wert: <code>active</code>), • nicht aktiven (Wert: <code>inactive</code>) oder • beiden (Wert: <code>all</code>) <p>Profilen bei der Suche unterschieden werden (vgl. Kap. 5.4.8 [ETSI TS 119 512]).</p> <p>Sollte dieser Parameter ungesetzt bleiben, so gilt die Standardbelegung: <code>active</code>.</p>

3.8.2 Ausgabeparameter: RetrieveInfoResponse

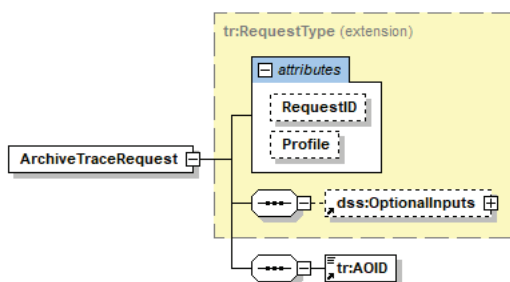
<i>Name</i>	<i>RetriveInfoResponse</i>	
Beschreibung	Als Antwort auf <code>RetrieveInfoRequest</code> wird ein <code>RetrieveInfoResponse</code> -Element zurückgeliefert, das die ermittelten Profile des Bewahrungsproduktes bzw. -dienstes beinhaltet.	
Details	<p>Der Ausgabeparameter <code>RetrieveInfoResponse</code> weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</p> 	
	Name	Beschreibung
	dss:Result	Das Element enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in [eCard-1] und unten näher beschrieben.
	dss:OptionalOutputs	Es werden standardmäßig <u>keine</u> optionalen Ausgabeelemente unterstützt.
	pres:Profile	<p>Eine Liste der ermittelten Profile, die entsprechend der Parametrisierung der Eingabe ermittelt wurden. Die Liste kann auch u.U. leer sein.</p> <p>Die Inhalte der Profile entsprechend der Spezifikation des Elements <code>pres:Profile</code> in [ETSI TS 119 512], Kap. 5.4.7.</p>

Name	RetriveInfoResponse	
	 <p>Statusinformationen und Fehler beim Aufruf der Funktion <code>RetrieveInfo</code> (vgl. [eCard-1] Abs. 4.1 und Abs. 4.2).</p>	
	Name	Fehlercode
	dss:ResultMajor	<ul style="list-style-type: none"> • /resultmajor#ok • /resultmajor#error
	dss:ResultMinor	<ul style="list-style-type: none"> • /resultminor/al/common#noPermission • /resultminor/al/common#internalError • /resultminor/al/common#parameterError • /resultminor/arl/notSupported¹⁹

3.9 Funktion: ArchiveTrace

Die Funktion `ArchiveTrace` erlaubt es, eine Dokumentation der bei der Verarbeitung eines Archivdatenobjekts innerhalb des Bewahrungsproduktes bzw. -dienstes ausgeführten Schritte abzurufen. Diese Dokumentation kann beispielsweise im Zuge eines Audits verwendet werden.

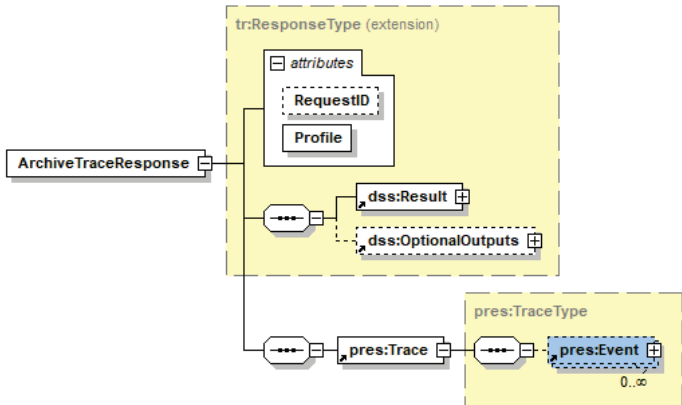
3.9.1 Eingabeparameter: ArchiveTraceRequest

Name	ArchiveTraceRequest	
Beschreibung	Mit dem Eingabeparameter <code>ArchiveTraceRequest</code> wird beim Aufruf der Funktion <code>ArchiveTrace</code> vorgegeben, nach welcher Verarbeitungsdokumentation (z.B. Logdateien) für welches Archivdatenobjekt eines Bewahrungsproduktes bzw. -dienstes gesucht wird.	
Details	<p>Der Eingabeparameter <code>ArchiveTraceRequest</code> weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</p> 	
	Name	Beschreibung

¹⁹ Im `ResultMessage`-Element sollen nähere Informationen darüber zurückgeliefert werden, welche angeforderte Funktionalität nicht unterstützt wird.

Name	ArchiveTraceRequest	
	dss:OptionalInputs	Es werden standardmäßig <u>keine</u> optionalen Eingabeelemente unterstützt.
	tr:AOID	Mit Hilfe dieses Parameters <u>muss</u> das Archivdatenobjekt, dessen Verarbeitungsdokumentation ermittelt werden soll, referenziert werden.

3.9.2 Ausgabeparameter: ArchiveTraceResponse

Name	ArchiveTraceResponse	
Beschreibung	Als Antwort auf den Eingabeparameter ArchiveTraceRequest wird ein ArchiveTraceReponse-Element zurückgeliefert, das die angeforderte Verarbeitungsdokumentation des gewünschten Archivdatenobjekts beinhaltet.	
Details	<p>Der Ausgabeparameter ArchiveTraceResponse weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</p> 	
	Name	Beschreibung
	dss:Result	Das Element enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements ist in [eCard-1] und unten näher beschrieben.
	dss:OptionalOutputs	Es werden standardmäßig <u>keine</u> optionalen Ausgabeelemente unterstützt.
	pres:Trace	<p>Die Rückgabe der Funktion <u>muss</u> das pres:Trace-Element enthalten.</p> <p>Im Erfolgsfall <u>muss</u> dieses Element zumindest ein Unterelement pres:Event beinhalten. Die genaue Ausgestaltung des pres:Event-Elements ist dem Kap. 5.4.10 in [ETSI TS 119 512] zu entnehmen.</p> <p>In einem Fehlerfall kann das pres:Trace-Element leer sein.</p>

Name	ArchiveTraceResponse	
	<div data-bbox="678 264 1086 405"> </div> <p data-bbox="399 416 1364 488">Statusinformationen und Fehler beim Aufruf der Funktion <code>ArchiveTrace</code> (vgl. [eCard-1] Abs. 4.1 und Abs. 4.2).</p>	
	Name	Fehlercode
	dss:ResultMajor	<ul style="list-style-type: none"> • /resultmajor#ok • /resultmajor#error
	dss:ResultMinor	<ul style="list-style-type: none"> • /resultminor/al/common#noPermission • /resultminor/al/common#internalError • /resultminor/al/common#parameterError • /resultminor/arl/notSupported²⁰ • /resultminor/arl/unknownAOID

²⁰Im ResultMessage-Element sollen nähere Informationen darüber zurückgeliefert werden, welche angeforderte Funktionalität nicht unterstützt wird.

4. Funktionen der Preservation-API gemäß ETSI TS 119 512 in der Profilierung [TR-ESOR-TRANS]

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

(A4.0–1) Für den Einsatz der „Preservation-API“ gemäß [ETSI TS 119 512] in der Profilierung [TR-ESOR-TRANS] im Rahmen der vorliegenden Technischen Richtlinie müssen die folgenden Mindestanforderungen unterstützt werden:

- `RetrieveInfo` gemäß Abs. 3.1 von [TR-ESOR-TRANS] 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 Abs. 3.2 von [TR-ESOR-TRANS] muss unterstützt werden, wobei zumindest eines der in [TR-ESOR-F] definierten Archivdatenobjekt-Formate (XAIP, LXAIP oder ASiC-AIP) unterstützt werden muss.
- `RetrievePO` gemäß Abs. 3.4 von [TR-ESOR-TRANS] muss unterstützt werden, wobei zumindest eines der in [TR-ESOR-F] definierten Archivdatenobjekt-Formate (XAIP, LXAIP oder ASiC-AIP) sowie Evidence Records gemäß [RFC4998] bzw. gemäß [RFC4998] in der Profilierung gemäß [TR-ESOR-ERS] unterstützt werden müssen.
- `DeletePO` gemäß Abs. 3.5 von [TR-ESOR-TRANS] muss unterstützt werden.
- `UpdatePOC` gemäß Abs. 3.3 von [TR-ESOR-TRANS] muss unterstützt werden.
- `RetrieveTrace` gemäß Abs. 5.3.7 von [ETSI TS 119 512] kann unterstützt werden.
- `ValidateEvidence` gemäß Abs. 3.6 von [TR-ESOR-TRANS] muss unterstützt werden. Sofern diese Operation unterstützt wird, muss zumindest die Validierung von Evidence Records gemäß [RFC4998] oder gemäß [RFC4998] in der Profilierung gemäß [TR-ESOR-ERS], Basic-ERS-Profile und 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 Abs. 3.7 von [TR-ESOR-TRANS] kann unterstützt werden.

(A4.0–2) Die Belegung der Eingabe- und Ausgabe-Parameter der unterstützten Funktionen im Rahmen des „Preservation-APIs“ muss gemäß dem TR-ESOR-Anlage [TR-ESOR-TRANS] erfolgen, der eine geeignet profilierte Ausprägung der Preservation-API gemäß [ETSI TS 119 512] spezifiziert, die auf die TR-S.4-Schnittstelle gemäß [TR-ESOR-E] abgebildet werden kann.

(A4.0–3) Für den Einsatz der „Preservation-API“ gemäß [ETSI TS 119 512] in der Profilierung [TR-ESOR-TRANS] im Rahmen der vorliegenden Technischen Richtlinie müssen die folgenden Basistypen für „Request“ und „Response“ unterstützt werden:

- Falls das `OptionalInputs` Element vorhanden ist, dann muss es eine Sub-Komponente, wie definiert in ([OASIS DSS-X], Kapitel 4.1.8), enthalten.
- Falls das `OptionalOutputs` Element vorhanden ist, dann muss es eine Sub-Komponente, wie definiert in ([OASIS DSS-X], Kapitel 4.1.9), enthalten.

4.1 Vergleich der TR-S.512- mit der TR-S.4-Schnittstelle

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

Tabelle 3: Vergleich ETSI TS 119 512 (prof. [TR-ESOR-TRANS]) Preservation-API mit TR-ESOR-S.4-Schnittstelle

ETSI TS 119 512 (prof. [TR-ESOR-TRANS])	Verbindlichkeitsgrad	TR-ESOR V1.3 ff	Verbindlichkeitsgrad
PreservePO	verpflichtend	ArchiveSubmission	verpflichtend
DeletePO	verpflichtend	ArchiveDeletion	verpflichtend
RetrievePO	verpflichtend	ArchiveEvidence	verpflichtend
RetrievePO	verpflichtend	ArchiveRetrieval	verpflichtend
UpdatePOC	verpflichtend	ArchiveUpdate	verpflichtend
ValidateEvidence	verpflichtend	Verify	verpflichtend
RetrieveInfo	verpflichtend	RetrieveInfo	verpflichtend
RetrieveTrace	optional	ArchiveTrace	optional
Search	optional	ArchiveData	optional

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 3 und Abbildung 4) erläutert:

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

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

Dieser Abschnitt beschreibt, wie die Abbildung 3 und Abbildung 4 dargestellte Schnittstelle TR-S.1 auf Basis des eCard-API-Frameworks (vgl. [TR-03112]) umgesetzt werden kann.

Diese Schnittstelle TR-S.1 umfasst zwei wesentliche Funktionen:

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

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-AIP) 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 Abs. 3.7 erläutert.

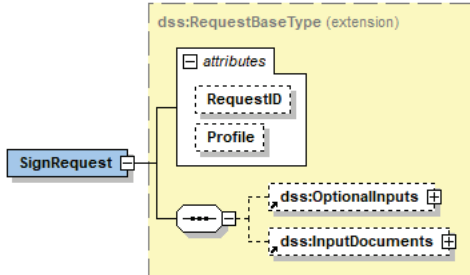
(A5.1.1–1) Die Durchführung der eigentlichen Prüffunktion von beweisrelevanten Daten sowie Beweisdaten muss im Krypto-Modul (siehe Anlage [TR-ESOR-M.2]) als Komponente der TR-ESOR-Middleware oder in einem vom Krypto-Modul aufgerufenen, (qualifizierten) Vertrauensdienst erfolgen. Die für die Prüfung notwendiger Prüfinformationen (z. B. OCSP-Antworten oder Sperrlisten) müssen von den Vertrauensdiensteanbietern abgerufen werden.

5.1.2 Anforderung einer digitalen Signatur

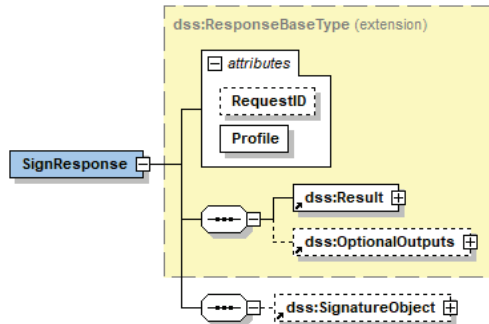
Für die Anforderung einer digitalen Signatur ist in [OASIS-DSS] und [eCard-2] die Anfrage `SignRequest` und die zugehörige Antwort `SignResponse` definiert.

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	(A5.1.2–1) Mit dem Funktionseingabeparameter <code>SignRequest</code> aus [eCard-2] <u>kann</u> für das übergebene Archivdatenobjekt eine digitale Signatur von einem (qualifizierten) Vertrauensdiensteanbieter gemäß [eIDAS-VO], Artikel 3 Nr. 19 bzw. Nr. 20 angefordert werden.	
Details	Der Eingabeparameter <code>SignRequest</code> weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.	
		
	Name	Beschreibung
	<code>dss:OptionalInputs</code>	<u>Kann</u> eines oder mehrere der in [eCard-2] definierten optionalen Eingabeelemente
<code>dss:InputDocuments</code>	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 Abs. 3.2.1 von [eCard-2] zurückgeliefert.	
Details	Der Ausgabeparameter SignResponse weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.	
		
	Name	Beschreibung
dss:Result	Dieses Element enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements und die möglichen Fehlercodes sind in Abs. 4.1.2 von [eCard-1] und in Abs. 3.2.1 von [eCard-2] beschrieben.	

<i>Name</i>	<i>SignResponse</i>	
	dss:OptionalOutputs	<u>Kann</u> ein <code>DocumentWithSignature-Element</code> enthalten, in denen z.B. ein <code>XAIP-Element</code> mit der eingebetteten digitalen Signatur enthalten ist. Details finden sich in Abs. 3.2.1 von [eCard-2].
	dss:SignatureObject	<u>Kann</u> eine erzeugte digitale Signatur in Form eines <code>dss:SignatureObject-Elementes</code> enthalten. Details finden sich in Abs. 3.2.1 von [eCard-2]. Sofern die erstellte digitale Signatur bereits im oben genannten <code>DocumentWithSignature-Element</code> vorhanden ist, wird kein <code>dss:SignatureObject-Element</code> zurückgeliefert.

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

Dieser Abschnitt beschreibt in den folgenden Unterkapiteln, wie die in Abbildung 3 und Abbildung 4 dargestellte Schnittstelle TR-S.2 auf Basis der auch dem eCard-API-Frameworks (vgl. [TR-03112] zu Grunde liegenden Basistypen aus [OASIS-DSS] umgesetzt werden kann.

Diese Schnittstelle umfasst drei wesentliche Funktionen:

- Speichern eines Archivdatenobjektes (Funktion `ArchiveSubmission`)
- Ergänzen einer neuen Version eines Archivdatenobjektes (Funktion `ArchiveUpdate`)

Auslesen eines Archivdatenobjektes (Funktion `ArchiveRetrieval`).

(A5.2–1) Neben der Umsetzung der Funktion „Speichern eines Archivdatenobjektes“ (`ArchiveSubmission`) auf Basis der, auch dem eCard-API-Frameworks [TR-03112] zu Grunde liegenden, Basistypen aus [OASIS-DSS] kann diese Funktion auch als „Upload-Request“ beliebig anders technisch umgesetzt werden, um den Upload von Datenobjekten im Rahmen eines logischen XAIP (LXAIP) gemäß [TR-ESOR-F], Abs. 3.2 technisch performant zu ermöglichen. Dabei müssen die Anforderungen gemäß [TR-ESOR], Abs. 7, insbesondere Abs. 7.2 und Abs. 7.4.4 erfüllt werden.

(A5.2–2) Laut [ETSI TS 119 511] muss die folgende Anforderung erfüllt sein: „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 der TR-ESOR-Middleware darstellen muss und logisch als Teil des TR-ESOR-Systems zu betrachten sein muss.

5.2.1 Speichern eines Archivdatenobjektes

Für das Speichern eines Archivdatenobjektes ist in Abbildung 3 und Abbildung 4 die Anfrage `ArchiveSubmissionRequest` und die zugehörige Antwort `ArchiveSubmissionResponse` gemäß Abs. 3.1 vorgesehen.

5.2.2 Ergänzen einer neuen Version eines Archivdatenobjektes

Für das Ergänzen einer neuen Version eines Archivdatenobjektes ist in Abbildung 3 und Abbildung 4 die Anfrage `ArchiveUpdateRequest` und die zugehörige Antwort `ArchiveUpdateResponse` gemäß Abs. 3.2 vorgesehen.

5.2.3 Auslesen von Archivdatenobjekten

Für das Auslesen von Archivdatenobjekten ist in Abbildung 3 und Abbildung 4 die Anfrage `ArchiveRetrievalRequest` und `ArchiveRetrievalResponse` gemäß Abs. 3.3 vorgesehen.

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

Dieser Abschnitt beschreibt, wie die in Abbildung 3 und Abbildung 4 dargestellte Schnittstelle TR-S.3 auf Basis des eCard-API-Frameworks [TR-03112] umgesetzt werden kann.

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

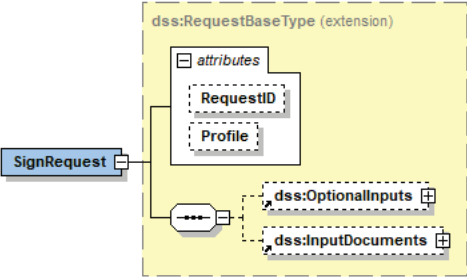
- Anfordern eines (qualifizierten) Zeitstempels (Funktion `SignRequest`)
- Prüfen eines (qualifizierten) Zeitstempels (Funktion `VerifyRequest`)
- Berechnung eines Hashwertes (Funktion `HashRequest`)

5.3.1 Anfordern eines (qualifizierten) Zeitstempels

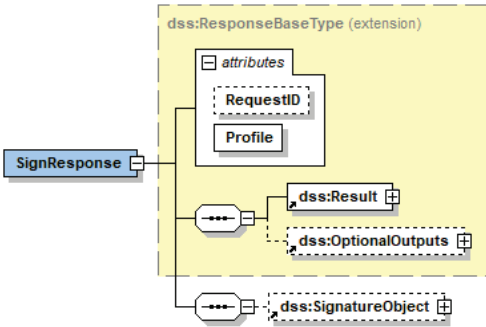
Zum Anfordern eines (qualifizierten) Zeitstempels kann eine geeignet profilierte Anfrage `SignRequest` mit entsprechender Antwort `SignResponse` gemäß [OASIS-DSS] bzw. [eCard-2] genutzt werden.

(A5.3.1-1) 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 <code>SignRequest</code> im Kontext der Schnittstelle S.3 übergibt einen Hashwert, zu dem ein (qualifizierter) Zeitstempel erstellt werden soll, an das Krypto-Modul.	
Details	<p>Der Eingabeparameter <code>SignRequest</code> weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</p> 	
	Name	Beschreibung
	<code>dss:OptionalInputs</code>	Enthält genau ein Element <code>SignatureType</code> mit der URI urn:ietf:rfc:3161 , durch die klargestellt wird, dass ein Zeitstempel gemäß [RFC3161] erzeugt werden soll.
	<code>dss:InputDocuments</code>	(A5.3.1-2) Während das Element <code>dss:InputDocuments</code> in [OASIS-DSS] und [eCard-2] optional ist, <u>muss</u> es hier vorhanden sein und genau ein <code>dss:Document</code> -Element in der <code>DocumentHash</code> -Ausprägung enthalten. Dieses Element enthält den Hashwert, aus dem ein (qualifizierter) Zeitstempel erzeugt werden soll.

5.3.1.2 SignResponse mit Zeitstempel

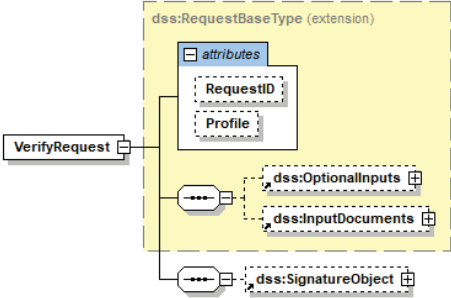
Name	SignResponse	
Beschreibung	Als Antwort auf einen SignRequest wird vom Krypto-Modul ein entsprechendes SignResponse-Element gemäß Abs.3.2.1 von [eCard-2] zurückgeliefert. Im Kontext der Schnittstelle S.3 wird hier ein (qualifizierter) Zeitstempel zurückgeliefert.	
Details	Der Ausgabeparameter SignResponse weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.	
		
	Name	Beschreibung
	dss:Result	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements und die möglichen Fehlercodes sind in Abs. 4.1.2 von [eCard-1] und in Abs. 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 3 und Abbildung 4) die Anfrage `VerifyRequest` und die Antwort `VerifyResponse` gemäß [OASIS-DSS] und [eCard-2] vorgesehen.

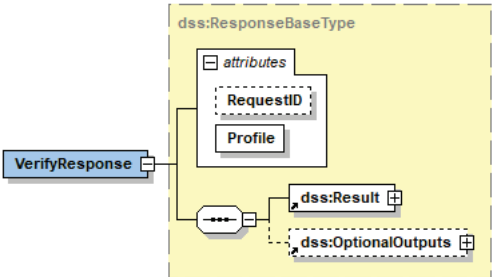
(A5.3.2–1) Die Durchführung der eigentlichen Prüffunktion eines (qualifizierten Zeitstempels) muss im Krypto-Modul (siehe Anlage [TR-ESOR-M.2]) als Komponente der TR-ESOR-Middleware oder in einem, vom Krypto-Modul aufgerufen, externen Validierungsdienst eines (qualifizierten) Vertrauensdiensteanbieters 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].
Details	<p>Der Eingabeparameter <code>VerifyRequest</code> weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</p> 
Name	Beschreibung
dss:OptionalInputs	<p><u>Kann</u> optionale Eingabeelemente enthalten.</p> <p>(A5.3.2-2) Gemäß der vorliegenden Spezifikation <u>muss</u> das optionale Eingabeelement <code>ReturnUpdatedSignature</code> aus Abs. 4.5.8 von [OASIS-DSS] unterstützt werden, bei dem mit dem Type-Attribut:</p> <ul style="list-style-type: none"> http://www.bsi.bund.de/DE/tr-esor/sigpolicy/verify-timestamp http://www.bsi.bund.de/DE/tr-esor/sigpolicy/verify-timestamp/shell oder http://www.bsi.bund.de/DE/tr-esor/sigpolicy/verify-timestamp/chain <p>klargestellt wird, dass:</p> <ol style="list-style-type: none"> 1) alle bei der Prüfung verwendeten Zertifikate und Sperrinformationen und Prüfinformationen, wie in [TR-ESOR-F] spezifiziert, in den Zeitstempel eingefügt werden <u>müssen</u>. 2) alle digitalen Signaturinformationen (Signaturen, Siegel, Zeitstempel, Zertifikate, Sperrlisten, OSCP-Responses etc.) bis hin zu einer vertrauenswürdigen Wurzel bzw. Vertrauensanker gemäß der vom [TR-ESOR-PEPT] abgeleiteten und veröffentlichten Preservation Policy (PEP) des TR-ESOR-Produktes bzw. Bewahrungsdienstes geprüft werden <u>müssen</u>. <p>(A5.3.2-3) Darüber hinaus <u>soll</u> das optionale Eingabeelement <code>ReturnVerificationReport</code> unterstützt werden, sodass für den entsprechenden Zeitstempel ein Prüfbericht gemäß [OASIS-VR] zurückgeliefert werden kann.</p>

<i>Name</i>	<i>VerifyRequest</i>	
	dss:InputDocuments	Das optionale Element dss:InputDocuments <u>soll nicht</u> vorhanden sein und wird ignoriert.
	dss:SignatureObject	Es ist genau ein dss:SignatureObject-Element in der dss:TimeStamp/RFC3161TimeStampToken-Ausprägung vorhanden, das den zu prüfenden Zeitstempel enthält.

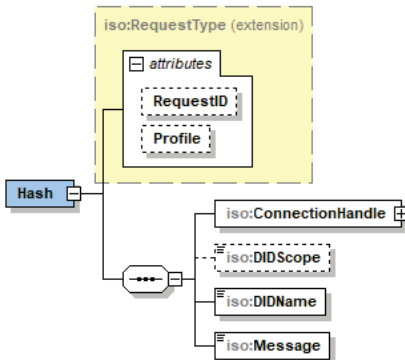
5.3.2.2 VerifyResponse

<i>Name</i>	<i>VerifyResponse</i>	
Beschreibung	Als Antwort auf einen VerifyRequest wird vom Krypto-Modul ein entsprechendes VerifyResponse-Element gemäß Abs. 3.2.2 von [eCard-2] zurückgeliefert.	
Details	<p>Der Ausgabeparameter ArchiveDataResponse weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</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 Abs. 4.1.2 von [eCard-1] und in Abs. 3.2.2 von [eCard-2] beschrieben.
	dss:OptionalOutputs	<p>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.</p> <p>Darüber hinaus <u>kann</u> ein VerificationReport-Element gemäß [OASIS VR] vorhanden sein, das im IndividualReport/Details-Element ein IndividualTimeStampReport-Element enthält.</p>

5.3.3 Berechnung eines Hashwertes

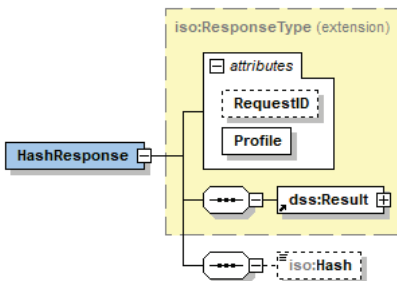
Zur Berechnung eines Hashwertes ist in TR-S.3 (vgl. Abbildung 3 und Abbildung 4) die Anfrage HashRequest und die Antwort HashResponse aus [eCard-4] in Verbindung mit dem Generic Cryptography-Protokoll aus [eCard-7] vorgesehen.

5.3.3.1 HashRequest

Name	HashRequest	
Beschreibung	Bei einem Hash-Aufruf im Kontext der Schnittstelle TR-S.3 wird für die übergebenen Daten ein Hashwert berechnet.	
Details	Der Eingabeparameter HashRequest weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.	
		
	Name	Beschreibung
	ConnectionHandle	Das ConnectionHandle-Element (vgl. [eCard-4], Abs. 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 ²¹	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 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. Wird als Inhalt von Message-Feld ein asic:DataObjectReference-Element gem. [TR-ESOR-F], Kap. 3.2.1 übergeben, so wird der Hashwert nicht über den Elementinhalt selbst, sondern über die aus dem Element referenzierten Daten berechnet. Die referenzierten Daten sind entsprechend anhand der mitgeführten Referenz zu ermitteln	

²¹Eine in [ISO 24727-3] 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 Abs. 3.5.4 von [eCard-4] zurückgeliefert.	
Details	<p>Der Ausgabeparameter HashResponse weist folgenden Aufbau auf und kann wie folgt parametrisiert werden.</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 Abs. 4.1.2 von [eCard-1] und in Abs. 3.5.4 von [eCard-4] beschrieben.
	Hash	Enthält den Hashwert, sofern ein solcher berechnet werden konnte.

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

Dieser Abschnitt beschreibt in den folgenden Unterkapiteln, wie die in TR-S.5 (vgl. Abbildung 3 und Abbildung 4) skizzierte Schnittstelle auf Basis der auch dem eCard-API-Framework [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 (Funktion ArchiveRetrieval)
- Löschen von Archivdatenobjekten (Funktion ArchiveDeletion)
- Abfrage diskreter Datenobjekte (Funktion ArchiveData)

(A5.4–1) Neben der Umsetzung der Funktion „Abfrage beweiswerterhaltend archivierter Daten (ArchiveRetrieval)“ auf Basis der, auch dem eCard-API-Frameworks [TR-03112] zu Grunde liegenden, Basistypen aus [OASIS-DSS] kann diese Funktion auch mittels eines „Download-Requests“ beliebig anders technisch umgesetzt werden, um den Download von Datenobjekten im Rahmen eines LXAIP gemäß [TR-ESOR-F], Abs. 3.2 technisch performant zu ermöglichen. In diesem Fall müssen die Anforderungen gemäß [TR-ESOR], Abs. 7 und insbesondere Abs. 7.2 und Abs. 7.4.5 erfüllt werden.

5.4.1 Abfrage beweiswerterhaltend archivierter Daten

Für die Abfrage beweiswerterhaltend archivierter Daten ist die Anfrage ArchiveRetrievalRequest und die Antwort ArchiveRetrievalResponse gemäß Abs. 3.3 vorgesehen.

5.4.2 Abfrage diskreter Datenobjekte

Für die Abfrage diskreter Datenobjekte ist die Anfrage `ArchiveDataRequest` und `ArchiveDataResponse` gemäß Abs. 3.6 vorgesehen.

5.4.3 Löschen von Archivdatenobjekten

Für das Löschen von Archivdatenobjekten ist die Anfrage `ArchiveDeletionRequest` und `ArchiveDeletionResponse` gemäß Abs. 3.5 vorgesehen.

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

Dieser Abschnitt beschreibt, wie die in Abbildung 3 und Abbildung 4 dargestellte Schnittstelle TR-S.6 auf Basis der auch dem eCard-API-Framework [TR-03112] zu Grunde liegenden Basistypen aus [OASIS-DSS] umgesetzt werden kann.

Die in Abbildung 3 und Abbildung 4 dargestellte Schnittstelle TR-S.6 umfasst die folgenden Funktionen:

- Beweiswerterhaltende Archivierung elektronischer Daten (Funktion `ArchiveSubmission`)
- Ergänzen einer neuen Version eines Archivdatenobjektes (Funktion `ArchiveUpdate`)
- Rückgabe technischer Beweisdaten (Funktion `ArchiveEvidence`)

5.5.1 Beweiswerterhaltende Archivierung elektronischer Daten

Für die beweiswerterhaltende Archivierung elektronischer Daten ist die Anfrage `ArchiveSubmissionRequest` und die Antwort `ArchiveSubmissionResponse` gemäß Abs. 3.1 vorgesehen.

5.5.2 Ergänzen einer neuen Version eines Archivdatenobjektes

Für das Ergänzen einer neuen Version eines Archivdatenobjektes ist die Anfrage `ArchiveUpdateRequest` und die Antwort `ArchiveUpdateResponse` gemäß Abs. 3.2 vorgesehen.

5.5.3 Rückgabe technischer Beweisdaten

Für die Rückgabe technischer Beweisdaten ist die Anfrage `ArchiveEvidenceRequest` und die Antwort `ArchiveEvidenceResponse` gemäß Abs. 3.4 vorgesehen.

6. Upload/Download-Schnittstelle

Die genaue Ausgestaltung der Upload/Download-Schnittstelle wird grundsätzlich dem einzelnen Hersteller überlassen. Die bereits genannten Anforderungen an die Upload/Download-Schnittstelle müssen aber stets eingehalten werden, darüber hinaus werden in den nachfolgenden Kapiteln einige wichtige Aspekte der Schnittstelle präzisiert, die bei der Umsetzung zwingend zu beachten sind.

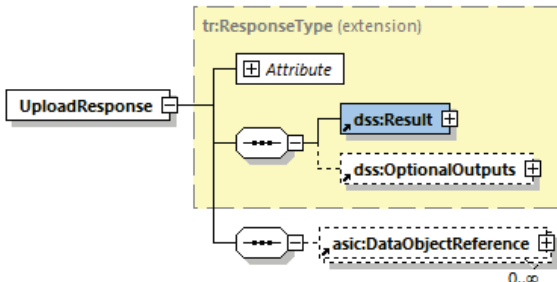
6.1 Upload-Funktion

Mit Hilfe der Upload-Funktion können binäre Daten (BIN gem. HINWEIS 5) an die Middleware initial übermittelt werden, die erst im Nachgang mit Hilfe der Funktionen `ArchiveSubmission` (vgl. Kap. 3.1) bzw. `ArchiveUpdate` (vgl. Kap. 3.2) unter Verwendung des korrespondierenden LXAIP bzw. DLXAIP in die beweiswerterhaltende Bewahrung aufgenommen werden (vgl. auch hierzu [TR-ESOR], Kap. 7.5.1 und Kap. 7.5.2). Ohne diesen Aufruf der Funktionen `ArchiveSubmission` (vgl. Kap. 3.1) bzw. `ArchiveUpdate` (vgl. Kap. 3.2) erfolgt keine beweiswerterhaltende Bewahrung der zuvor übermittelten Daten und diese **binären Daten werden nach Ablauf einer Frist unwiderruflich gelöscht**.

6.1.1 Upload-Anfrage

Name	Upload-Anfrage
Beschreibung	Upload-Anfrage kann binäre Daten in die Middleware übermitteln.
Details	<p>(A6.1-1) Es <u>dürfen</u> ausschließlich die im HINWEIS 5 als BIN definierten Datenformate angenommen werden. <u>Ausgeschlossen</u> sind explizit folgende Formate:</p> <ul style="list-style-type: none"> • XAIP gem. [TR-ESOR-F], Kap. 3.1 • LXAIP gem. [TR-ESOR-F], Kap. 3.2 • ASiC-AIP gem. [TR-ESOR-F], Kap. 3.3. <p>(A6.1-2) Die übermittelten Datenobjekte <u>müssen</u> in einer Relation mit einem LXAIP gem. [TR-ESOR-F] Kap. 3.2 stehen.</p>

6.1.2 Upload-Antwort

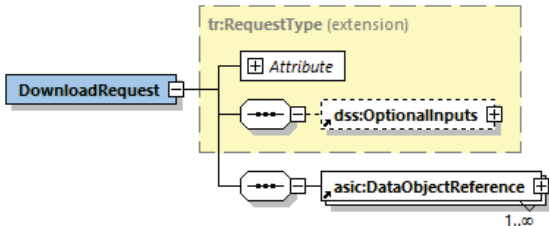
Name	Upload-Antwort
Beschreibung	Als Antwort auf eine Upload-Anfrage wird von der Upload/Download-Schnittstelle für je ein übermitteltes Datenobjekt eine <code>asic:DataObjectReference</code> zurückgeliefert. Im Falle eines Fehler <u>muss</u> der Zustand an die übermittelnde Instanz deutlich signalisiert werden.
Details	<p>Folgende Darstellung einer möglichen Upload-Antwort in Form des Elements <code>UploadResponse</code> stellt eine Empfehlung für die Umsetzung dar.</p> 

<i>Name</i>	<i>Upload-Antwort</i>	
	Name	Beschreibung
	dss:Result	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements und die möglichen Fehlercodes sind in Abs. 4.1.2 von [eCard-1] und in Abs. 3.2.1 von [eCard-2] beschrieben.
	dss:OptionalOutputs	Das optionale Element dss:OptionalOutputs ist nicht vorhanden.
	asic:DataObjectReference	Enthält – sofern kein Fehler aufgetreten ist – mindestens eine Instanz des Elements asic:DataObjectReference gem. [TR-ESOR-F], Kap. 3.2.1, die das übermittelte Datenobjekt eindeutig referenziert.
	<div data-bbox="651 779 1091 936" data-label="Diagram"> <pre> graph LR Result[Result] --- Container[...] Container --- dss:ResultMajor[dss:ResultMajor] Container --- dss:ResultMinor[dss:ResultMinor] Container --- dss:ResultMessage[dss:ResultMessage] </pre> </div> <p data-bbox="440 999 1398 1066">Statusinformationen und Fehler beim Aufruf der Funktion Upload (vgl. [eCard-1] Abs. 4.1 und Abs. 4.2).</p>	
	Name	Fehlercode
	ResultMajor	<ul style="list-style-type: none"> • /resultmajor#ok • /resultmajor#error
	ResultMinor	<ul style="list-style-type: none"> • /resultminor/ar1/uploadDataFormatNotSupported

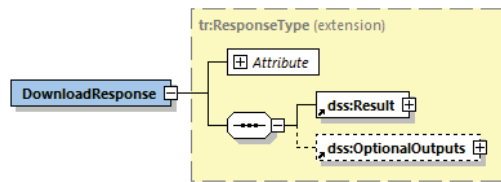
6.2 Download-Funktion

Mit Hilfe der Download-Funktion können in einem zuvor bewahrten LXAIP bzw. DLXAIP referenzierten Datenobjekte ausgelesen werden.

6.2.1 Download-Anfrage

Name	Download-Anfrage	
Beschreibung	Mit Hilfe der Download-Anfrage wird ein (oder mehrere) zuvor im Langzeitspeicher und in einem LXAIP referenziertes/-en Datenobjekt(e) ausgelesen werden.	
Details	<p>Folgende Darstellung einer möglichen Download-Anfrage in Form des Elements <code>DownloadRequest</code> stellt eine Empfehlung für die Umsetzung dar.</p> 	
	Name	Beschreibung
	<code>dss:OptionalInputs</code>	Das optionale Element <code>dss:OptionalInputs</code> ist nicht vorhanden.
	<code>asic:DataObjectReference</code>	Enthält mindestens eine Instanz des Elements <code>asic:DataObjectReference</code> gem. [TR-ESOR-F], Kap. 3.2.1, die das zuvor übermittelte und im zugehörigen LXAIP referenzierte Datenobjekt eindeutig beschreibt.

6.2.2 Download-Antwort

Name	Download-Antwort	
Beschreibung	<p>Als Antwort auf eine Download-Anfrage wird von der Upload/Download-Schnittstelle zu jedem mit einer Instanz des Elements <code>asic:DataObjectReference</code> angefragten Datenobjekt dieses auch ausgeliefert.</p> <p>Wenn die Anfrage nicht erfolgreich ausgeführt werden kann, muss eine Fehlermeldung zurückgegeben werden</p>	
Details	<p>Folgende Darstellung einer möglichen Download-Antwort in Form des Elements <code>DownloadResponse</code> stellt eine Empfehlung für die Umsetzung im Falle eines Fehlers dar.</p> 	

<i>Name</i>	<i>Download-Antwort</i>	
	Name	Beschreibung
	dss:Result	Enthält die Statusinformationen und die Fehler zu einer durchgeführten Aktion. Die Struktur dieses Elements und die möglichen Fehlercodes sind in Abs. 4.1.2 von [eCard-1] und in Abs. 3.2.1 von [eCard-2] beschrieben.
	dss:OptionalOutputs	Das optionale Element dss:OptionalOutputs ist nicht vorhanden.
	<div data-bbox="651 577 1091 725" data-label="Diagram"> <pre> graph LR Result[Result] --- Container[...] Container --- dss:ResultMajor[dss:ResultMajor] Container --- dss:ResultMinor[dss:ResultMinor] Container --- dss:ResultMessage[dss:ResultMessage] </pre> </div> <p>Statusinformationen und Fehler beim Aufruf der Funktion <code>ArchiveTrace</code> (vgl. [eCard-1] Abs. 4.1 und Abs. 4.2).</p>	
	Name	Fehlercode
	ResultMajor	<ul style="list-style-type: none"> • /resultmajor#ok • /resultmajor#error
	ResultMinor	<ul style="list-style-type: none"> • /resultminor/ar/unknownDataObjectReference

7. 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:

<i>Fehlercode</i>	<i>Beschreibung</i>
.../resultminor/arl/DXAIP NOK	Die Syntax des beim ArchiveUpdateRequest übergebenen Delta-XAIP-Elements ist nicht korrekt.
.../resultminor/arl/DXAIP NOK AOID	Die AOID in dem beim ArchiveUpdateRequest übergebenen Delta-XAIP ist nicht bekannt.
.../resultminor/arl/DXAIP NOK EXPIRED	Das beim ArchiveUpdateRequest übergebene Delta-XAIP-Element kann nicht abgelegt werden, da die Aufbewahrungsfrist abgelaufen ist.
.../resultminor/arl/DXAIP NOK SUBMTIME	Die beim ArchiveUpdateRequest im übergebenen Delta-XAIP-Element angegebene submissionTime ist nicht korrekt, da sie in der Zukunft liegt.
.../resultminor/arl/DXAIP NOK SIG	Das beim ArchiveUpdateRequest übergebene Delta-XAIP-Element enthält zumindest eine ungültige digitale Signatur.
.../resultminor/arl/DXAIP NOK ER	Das beim ArchiveUpdateRequest übergebene Delta-XAIP-Element enthält zumindest einen ungültigen Evidence Record.
.../resultminor/arl/DXAIP NOK ID	Die beim ArchiveUpdateRequest in einem placeHolder-Element übergebene XML-ID ist im bereits abgelegten XAIP-Element nicht vorhanden.
.../resultminor/arl/DXAIP NOK Version	Die beim ArchiveUpdateRequest im prevVersion-Element übergebene Version ist nicht die aktuellste Version.
.../resultminor/arl/existingAOID	Die im Rahmen des ArchiveSubmissionRequest übergebene AOID existiert bereits.
.../resultminor/arl/existingPackage InfoWarning	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.
.../resultminor/arl/lowSpaceWarning	Diese Warnung gibt an, dass der verfügbare Speicherplatz einen kritischen Wert unterschritten hat.

<i>Fehlercode</i>	<i>Beschreibung</i>
.../resultminor/arl/missingReasonOfDeletion	Da beim ArchiveDeletionRequest kein ReasonOfDeletion-Element übergeben wurde, muss der Löschvorgang abgewiesen werden.
.../resultminor/arl/noSpaceError	Diese Fehlermeldung gibt an, dass kein Speicherplatz verfügbar war und deshalb das Archivdatenobjekt nicht abgelegt werden konnte.
.../resultminor/arl/notSupported	Diese Fehlermeldung gibt an, dass eine angeforderte Funktion, ein angefordertes Format oder ein übergebener optionaler Eingabeparameter nicht unterstützt wird.
.../resultminor/arl/requestOnlyPartlySuccessfulWarning	Diese Warnung gibt an, dass nicht alle angeforderten Daten zurückgeliefert werden konnten.
.../resultminor/arl/unknownArchiveDataType	Es wird ein binäres Datenobjekt mit einem nicht unterstützten Datenformat übergeben.
.../resultminor/arl/unknownLocation	Die im ArchiveDataRequest angegebene DataLocation ist nicht vorhanden bzw. fehlerhaft.
.../resultminor/arl/unknownAOID	Die übergebene AOID existiert nicht.
.../resultminor/arl/unknownVersionID	Die übergebene VersionID ist im entsprechenden XAIP nicht bekannt.
.../resultminor/arl/XAIP_NOK	Die Syntax des übergebenen AIP-Containers (d. h. XAIP, LXAIP, ASiC-AIP) ist nicht korrekt.
.../resultminor/arl/XAIP_NOK_ER	Der übergebene AIP-Container (d. h. XAIP, LXAIP, ASiC-AIP) enthält zumindest einen ungültigen Evidence Record.
.../resultminor/arl/XAIP_NOK_EXPIRED	Der übergebene AIP-Container (d. h. XAIP, LXAIP, ASiC-AIP) kann nicht abgelegt werden, da die Aufbewahrungsfrist abgelaufen ist.
.../resultminor/arl/XAIP_NOK_SIG	Der übergebene AIP-Container (d. h. XAIP, LXAIP, ASiC-AIP) enthält zumindest eine ungültige Signatur.
.../resultminor/arl/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/arl/noDataAccessWarning	Der Zugriff auf die in einem übergebenen LXAIP referenzierten Daten ist nicht möglich.
.../resultminor/arl/unknownPOFormat	Der angeforderte POFormat-Typ ist nicht bekannt.
.../resultminor/arl/uploadDataFormatNotSupported	Das Datenformat ist für den Upload nicht zugelassen.
.../resultminor/arl/unknownDataObjectReference	Das durch eine Instanz von asic:DataObjectReference beschriebene Datenobjekt ist nicht bekannt.
.../resultminor/sal#invalidSignature	Die übergebene Signatur ist falsch.

Tabelle 4: Zusätzliche Fehlercodes.

8. 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. Abs. 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. Abs. 3). Die internen Schnittstellen der TR-ESOR-Middleware können bei Bedarf leicht daraus abgeleitet werden, indem nur die benötigte Teilmenge der Funktionen genutzt wird.

(A8.0-1) Die Unterstützung des optimierten Nachrichtenübertragungsmechanismus „SOAP Message Transmission Optimization Mechanism (MTOM)“²² kann durch den Import des geringfügig angepassten XAIP-Schema (tr-esor-xaip-v1.3.xsd) erfolgen.

8.1 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/1.3"
  xmlns:xaip="http://www.bsi.bund.de/tr-esor/xaip"
  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"
  xmlns:xmime="http://www.w3.org/2005/05/xmlmime"
  xmlns:pres="http://uri.etsi.org/19512/v1.1.2#"
  xmlns:asic="http://uri.etsi.org/02918/v1.2.1#"
  targetNamespace="http://www.bsi.bund.de/tr-esor/api/1.3"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  version="1.3.0">
  <!-- ===== -->
  <!-- Version 1.3.0 vom 14.03.2022 -->
  <!-- ===== -->
  <import namespace="http://www.bsi.bund.de/tr-esor/xaip"
    schemaLocation="tr-esor-xaip-v1.3.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"/>
  <import namespace="http://www.w3.org/2005/05/xmlmime"
    schemaLocation="./deps/xmlmime.xsd"/>
  <import namespace="http://uri.etsi.org/19512/v1.1.2#"
    schemaLocation="19512-Preservation-API V.1.1.2.xsd"/>
  <import namespace="http://uri.etsi.org/02918/v1.2.1#"
    schemaLocation="./deps/en_31916201v010101.xsd"/>
  <!-- ===== -->
  <!-- Uebergreifende Definitionen -->
  <!-- ===== -->
  <complexType name="RequestType">
    <complexContent>
      <restriction base="dss:RequestBaseType">
        <sequence>
          <element ref="dss:OptionalInputs" minOccurs="0"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
```

²²Siehe <https://www.w3.org/TR/soap12-mtom/>.

```

</complexContent>
</complexType>
<complexType name="ResponseType">
  <complexContent>
    <restriction base="dss:ResponseBaseType">
      <sequence>
        <element ref="dss:Result"/>
        <element ref="dss:OptionalOutputs" minOccurs="0"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
<element name="AOID" type="string"/>
<element name="VerifyUnderSignaturePolicy" type="anyURI"/>
<element name="XPathFilter" type="string"/>
<!-- ===== -->
<!-- RetrieveInfo -->
<!-- ===== -->
<element name="RetrieveInfoRequest">
  <complexType>
    <complexContent>
      <extension base="tr:RequestType">
        <sequence>
          <element name="ProfileIdentifier" type="anyURI" minOccurs="0"/>
          <element name="Status" type="pres:StatusType" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<element name="RetrieveInfoResponse">
  <complexType>
    <complexContent>
      <extension base="tr:ResponseType">
        <sequence>
          <element ref="pres:Profile" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<!-- ===== -->
<!-- ArchiveSubmission -->
<!-- ===== -->
<complexType name="ArchiveDataType" xmlns:xmime="http://www.w3.org/2003/xml/soap/encoding#">
  <simpleContent>
    <extension base="base64Binary">
      <attribute name="Type" type="anyURI" use="required"/>
      <attribute name="archiveDataID" type="ID" use="required"/>
      <attribute name="MimeType" type="string" use="optional"/>
      <attribute name="relatedObjects" type="IDREFS" use="optional"/>
    </extension>
  </simpleContent>
</complexType>
<element name="ImportEvidence" type="tr:ImportEvidenceType"/>
<complexType name="ImportEvidenceType">
  <choice>
    <element ref="xaip:evidenceRecord" maxOccurs="unbounded"/>
    <element name="CredentialID" type="string" maxOccurs="unbounded"/>
  </choice>
</complexType>
<element name="ArchiveSubmissionRequest">
  <complexType>
    <complexContent>
      <extension base="tr:RequestType">
        <choice>
          <element ref="xaip:XAIP"/>
          <element name="ArchiveData" type="tr:ArchiveDataType" maxOccurs="unbounded"/>
        </choice>
      </extension>
    </complexContent>
  </complexType>

```

```

</complexType>
</element>
<element name="ArchiveSubmissionResponse">
  <complexType>
    <complexContent>
      <extension base="tr:ResponseType">
        <sequence>
          <element name="AOID" type="string" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<!-- ===== -->
<!--      ArchiveUpdate      -->
<!-- ===== -->
<element name="ArchiveUpdateRequest">
  <complexType>
    <complexContent>
      <extension base="tr:RequestType">
        <sequence>
          <element ref="xaip:DXAIP"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<element name="ArchiveUpdateResponse">
  <complexType>
    <complexContent>
      <extension base="tr:ResponseType">
        <sequence>
          <element name="VersionID" type="string" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<!-- ===== -->
<!--      ArchiveRetrieval      -->
<!-- ===== -->
<element name="ArchiveRetrievalRequest">
  <complexType>
    <complexContent>
      <extension base="tr:RequestType">
        <sequence>
          <element name="AOID" type="string"/>
          <element name="VersionID" type="string" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<element name="IncludeERS" type="anyURI"/>
<element name="ArchiveRetrievalResponse">
  <complexType>
    <complexContent>
      <extension base="tr:ResponseType">
        <sequence>
          <element ref="xaip:XAIP" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<!-- ===== -->
<!--      ArchiveEvidence      -->
<!-- ===== -->
<element name="ArchiveEvidenceRequest">
  <complexType>
    <complexContent>

```

```

    <extension base="tr:RequestType">
      <sequence>
        <element name="AOID" type="string"/>
        <element name="VersionID" type="string" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
</element>
<element name="ERSFormat" type="anyURI"/>
<element name="ArchiveEvidenceResponse">
  <complexType>
    <complexContent>
      <extension base="tr:ResponseType">
        <sequence>
          <element ref="xaip:evidenceRecord" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<!-- ===== -->
<!--      ArchiveDeletion      -->
<!-- ===== -->
<element name="ArchiveDeletionRequest">
  <complexType>
    <complexContent>
      <extension base="tr:RequestType">
        <sequence>
          <element name="AOID" type="string"/>
        </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"/>
<!-- ===== -->
<!--      ArchiveData      -->
<!-- ===== -->
<element name="ArchiveDataRequest">
  <complexType>
    <complexContent>
      <extension base="tr:RequestType">
        <sequence>
          <element name="AOID" type="string"/>
          <element ref="tr:DataLocation" maxOccurs="unbounded"/>
        </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">
      <complexType>
        <sequence>
          <element ref="dss:Result"/>
          <element ref="tr:DataLocation"/>
          <element name="Value" type="anyType" minOccurs="0"/>
        </sequence>
      </complexType>
    </element>
  </sequence>
</extension>
</complexContent>
</complexType>
</element>
<!-- ===== -->
<!--      ArchiveTrace      -->
<!-- ===== -->
<element name="ArchiveTraceRequest">
  <complexType>
    <complexContent>
      <extension base="tr:RequestType">
        <sequence>
          <element ref="tr:AOID"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<element name="ArchiveTraceResponse">
  <complexType>
    <complexContent>
      <extension base="tr:ResponseType">
        <sequence>
          <element ref="pres:Trace"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<!-- ===== -->
<!--      Upload      -->
<!-- ===== -->
<element name="UploadResponse">
  <complexType>
    <complexContent>
      <extension base="tr:ResponseType">
        <sequence>
          <element ref="asic:DataObjectReference" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<!-- ===== -->
<!--      Download      -->
<!-- ===== -->
<element name="DownloadRequest">
  <complexType>
    <complexContent>
      <extension base="tr:RequestType">
        <sequence>
          <element ref="asic:DataObjectReference" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<element name="DownloadResponse">
  <complexType>

```

```

    <complexContent>
      <extension base="tr:ResponseType"/>
    </complexContent>
  </complexType>
</element>
</schema>

```

8.2 WSDL-Spezifikation der Schnittstelle TR-S.4

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions 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/1.3"
  targetNamespace="http://www.bsi.bund.de/tr-esor/api/1.3">
  <!-- ===== -->
  <!-- Version 1.3.0 of 14.03.2022 -->
  <!-- ===== -->
  <!-- ===== -->
  <!-- Definition of types -->
  <!-- (only include XSDs) -->
  <!-- ===== -->
  <wsdl:types>
    <xsd:schema targetNamespace="http://www.bsi.bund.de/tr-esor/api/1.3"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xaip="http://www.bsi.bund.de/tr-esor/xaip"
      xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema"
      elementFormDefault="qualified">
      <xsd:include schemaLocation="tr-esor-interfaces-v1.3.xsd"/>
    </xsd:schema>
  </wsdl:types>
  <!-- ===== -->
  <!-- Definition of messages -->
  <!-- ===== -->
  <!-- RetrieveInfo -->
  <wsdl:message name="RetrieveInfoRequest">
    <wsdl:part name="parameters" element="tr:RetrieveInfoRequest"/>
  </wsdl:message>
  <wsdl:message name="RetrieveInfoResponse">
    <wsdl:part name="parameters" element="tr:RetrieveInfoResponse"/>
  </wsdl:message>
  <!-- Archivesubmission -->
  <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>
  <!-- ArchiveUpdate -->
  <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>
  <!-- ArchiveRetrieval -->
  <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>
  <!-- ArchiveEvidence -->
  <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>

```

```

</wsdl:message>
<!-- ArchiveDeletion -->
<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>
<!-- ArchiveData -->
<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>
<!-- Verify -->
<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>
<!-- ArchiveTrace -->
<wsdl:message name="ArchiveTraceRequest">
  <wsdl:part name="parameters" element="tr:ArchiveTraceRequest"/>
</wsdl:message>
<wsdl:message name="ArchiveTraceResponse">
  <wsdl:part name="parameters" element="tr:ArchiveTraceResponse"/>
</wsdl:message>
<!-- ===== -->
<!-- Definition of portType -->
<!-- ===== -->
<wsdl:portType name="S4">
  <wsdl:operation name="RetrieveInfo">
    <wsdl:input message="tr:RetrieveInfoRequest"/>
    <wsdl:output message="tr:RetrieveInfoResponse"/>
  </wsdl:operation>
  <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:operation name="ArchiveTrace">
    <wsdl:input message="tr:ArchiveTraceRequest"/>
    <wsdl:output message="tr:ArchiveTraceResponse"/>
  </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="RetrieveInfo">
    <soap:operation soapAction="http://www.bsi.bund.de/tr-esor/RetrieveInfo"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ArchiveSubmission">
    <soap:operation soapAction="http://www.bsi.bund.de/tr-esor/ArchiveSubmission"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ArchiveUpdate">
    <soap:operation soapAction="http://www.bsi.bund.de/tr-esor/ArchiveUpdate"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ArchiveRetrieval">
    <soap:operation soapAction="http://www.bsi.bund.de/tr-esor/ArchiveRetrieval"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ArchiveEvidence">
    <soap:operation soapAction="http://www.bsi.bund.de/tr-esor/ArchiveEvidence"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ArchiveDeletion">
    <soap:operation soapAction="http://www.bsi.bund.de/tr-esor/ArchiveDeletion"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ArchiveData">
    <soap:operation soapAction="http://www.bsi.bund.de/tr-esor/ArchiveData"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="Verify">
    <soap:operation soapAction="http://www.bsi.bund.de/tr-esor/Verify"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
  </wsdl:operation>

```



```
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="ArchiveTrace">
  <soap:operation soapAction="http://www.bsi.bund.de/tr-esor/ArchiveTrace"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<!-- ===== -->
<!-- Definition of 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>
```