

DZAB Access Service

Archive Service SecDocs

Archivierung von Dokumenten gemäß Richtlinie TR-ESOR

Benutzerhandbuch

Stand der Beschreibung

DZAB Access Service 22.00.01.00

Archive Service SecDocs V3.1B

Ausgabe November 2021

1 Der DZAB Access Service

Zur rechtskonformen und revisionssicheren Langzeitarchivierung digitaler Objekte (Dateien, Akten, Vorgänge, Dokumente, ...) stellt das Bundesarchiv den Ministerien, Behörden und Gerichten des Bundes das Digitale Zwischenarchiv des Bundes (DZAB) bereit. Das zugrundeliegende eArchiv wird von der Bundesagentur für Arbeit (BA) im Rahmen der IT-Konsolidierung des Bundes betrieben. Es beinhaltet den Access Service als Frontend und die Fujitsu Software SecDocs als Archive Service.

Dieser Band enthält das Benutzerhandbuch der Fujitsu Software SecDocs für die Archivierung von Dokumenten im Format XAIP gemäß der Richtlinie TR-ESOR.

FUJITSU Software

SecDocs

Archivierung von Dokumenten gemäß Richtlinie TR-ESOR

Benutzerhandbuch

Stand der Beschreibung
V3.1B

Ausgabe November 2021

Kritik... Anregungen... Korrekturen...

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Sie können uns Ihre Kommentare per E-Mail an bs2000services@ts.fujitsu.com senden.

Zertifizierte Dokumentation nach DIN EN ISO 9001:2015

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001:2015 erfüllt.

Copyright und Handelsmarken

Copyright © 2021 Fujitsu Technology Solutions GmbH.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

Alle verwendeten Hard- und Softwareramen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller.

Inhaltsverzeichnis

SecDocs Benutzerhandbuch (TR-ESOR)	5
1 Einführung	6
1.1 Konzept und Zielgruppe des Handbuchs	7
1.2 Darstellungsmittel	8
2 Konzepte und Funktionen	9
2.1 Architektur	10
2.1.1 Komponenten von SecDocs	11
2.1.2 Software-Umgebung	14
2.1.3 Ablagestruktur im Archiv	15
2.1.4 Storage-Systeme	16
2.2 Sicherer Betrieb	17
2.2.1 Anforderungen für den sicheren Betrieb	18
2.2.2 Sichere Kommunikation	19
2.2.3 Logging	20
3 Administration für den Web-Service S4	21
3.1 Mandanten für den Web-Service S4 erzeugen	22
3.2 Zertifikate einbringen	23
3.3 Operationen für die Administration	24
3.3.1 Operation createMandantXAIP	25
3.3.2 Operation modifyXAIP	35
3.3.3 Operation setCredentials - Zugang zum Web-Service S4	40
3.3.4 Ausgabeinformationen für den Web-Service S4	43
3.3.4.1 Operationen getMandants, getMandantProperties, getArchiveInfo	44
3.3.4.2 Operation getSDOTypes	48
4 Web-Service S4 für die Client-Anwendung	49
4.1 Zugang (Endpoint-URL) und Zugangsprüfung	51
4.2 Format eines Archivdatenobjekts	52
4.2.1 SecDocs-spezifische Erweiterungen des Formats XAIP	55
4.3 Aufbau der SOAP-Nachricht beim SecDocs Web-Service S4	57
4.3.1 Request	58
4.3.2 Response	59
4.3.3 Status- und Fehlerinformation	61
4.4 Operationen des Web-Service S4	67
4.4.1 Operation ArchiveSubmission	68
4.4.2 Operation ArchiveRetrieval	76
4.4.3 Operation ArchiveEvidence	81
4.4.4 Operation ArchiveDeletion	87

4.4.5 Änderung der Aufbewahrungszeit	92
5 Fachwörter	101
6 Abkürzungen	106
7 Literatur	108

1 Einführung

Immer mehr Geschäftsprozesse werden elektronisch abgebildet, weil so die Kosten gesenkt und die Verwaltungsvorgänge nachhaltig beschleunigt werden können. Entsprechend verdrängen immer mehr elektronische Dokumente die in der Handhabung teuren Papierbelege.

Elektronische Dokumente müssen über die gleiche dauerhafte Beweiskraft verfügen und ebenso vertrauenswürdig sein wie die Papierdokumente, um Geschäftsprozesse rechtlich abzusichern. Außerdem muss der Nachweis über ihre Integrität und Authentizität jederzeit – teilweise über 100 Jahre – erbracht werden können.

SecDocs ist eine Archiv-Middleware für elektronische Dokumente und bietet:

- Langzeitarchivierung
- Konzepte der BSI Technischen Richtlinie TR-VELS / TR-ESOR
- Erhaltung der Beweiswerte elektronisch signierter Dokumente, d.h. SecDocs übernimmt die Aufgaben der Übersignatur autonom
- Hohe gerichtsverwertbare Beweiskraft bei dauerhaft niedrigen Betriebskosten
- Einfache Integration in eine Vielzahl von IT-Umgebungen und Fachverfahren
- Nachweis der Integrität des Dokuments mindestens seit dem Zeitpunkt der Archivierung
- Unterstützung unterschiedlicher Betriebsmodelle (stand-alone, public cloud, community cloud, private cloud)

SecDocs basiert auf offenen Standards und verwendet zertifizierte Sicherheitskomponenten. Die SecDocs-Security-Komponenten sind vom Bundesamt für Sicherheit in der Informationstechnik (BSI) nach Common Criteria zertifiziert.

SecDocs ist eine Service Oriented Architecture-Lösung (SOA-Lösung) und befreit die öffentliche Verwaltung und Unternehmen mit hohen Compliance-Vorgaben vom Umgang mit komplexen elektronischen Signaturen zur Beweissicherung.

Mit SecDocs V3.1B wird eine Archivierung von Dokumenten im Format XAIP gemäß der Richtlinie TR-ESOR V1.2.1 angeboten.

Zu diesem Zweck stellt SecDocs den Web-Service s4 mit den folgenden Funktionen bereit:

- `ArchiveSubmission`
Ablegen von XAIP-Datenobjekten im Langzeitspeicher
- `ArchiveRetrieval`
Abrufen eines Archivdatenobjekts aus dem Langzeitspeicher
- `ArchiveEvidence`
Abrufen von technischen Beweisdaten zu einem Archivdatenobjekt
- `ArchiveDeletion`
Löschen eines Archivdatenobjekts

XAIP steht für XML formatted Archival Information Package und bezeichnet ein selbstbeschreibendes und wohlgeformtes XML-Dokument, das gegen ein gültiges und autorisiertes XML-Schema geprüft werden kann. Ein solches Archivdatenobjekt enthält sämtliche Inhaltsdaten (Primärinformationen) und Metainformationen, die für eine zuverlässige und vollständige Rekonstruktion von Geschäfts- oder Verwaltungsvorgängen bis zum Ablauf der gesetzlich vorgeschriebenen Aufbewahrungsfristen erforderlich sind.

1.1 Konzept und Zielgruppe des Handbuchs

Dieses Handbuch ist eine Ergänzung des Handbuchs "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "[Literatur](#)"). Es wendet sich an Programmierer und Systemverwalter, die

- das Anwendungskonzept für SecDocs entwickeln,
- die Fachanwendung an SecDocs anbinden und die XAIP-Datenobjekte erstellen,
- das SecDocs-Archiv und die Mandanten administrieren,
- SecDocs betreiben und das System administrieren.

Für das Verständnis des Handbuchs werden folgende Kenntnisse vorausgesetzt:

- Linux-Betriebssystem und WildFly Application Server
- allgemeine XML-Kenntnisse
- Administration des jeweiligen Speichersystems
- benötigte Datenbank-Software
- Grundlagen der Langzeitarchivierung

Einen kurzen Einstieg in die Grundlagen der elektronischen Langzeitarchivierung und in die Funktionen von SecDocs, die dieses Konzept realisieren, finden Sie im Handbuch "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "[Literatur](#)").

Das [Kapitel „Konzepte und Funktionen“](#) gibt einen Überblick über die Anforderungen an den sicheren Betrieb und die Architektur von SecDocs.

Das [Kapitel „Administration für den Web-Service S4“](#) beschreibt notwendige Administrationsoperationen, um die Bedienung des Web-Service S4 zu ermöglichen.

Das [Kapitel „Web-Service S4 für die Client-Anwendung“](#) beschreibt die Arbeit mit dem Web-Service S4. Dieser Web-Service stellt die Operationen für die Archivierung von Dokumenten im XAIP-Format zur Verfügung.

Eine ausführliche Beschreibung der Konfiguration und Inbetriebnahme von SecDocs sowie der verschiedenen Logging-Möglichkeiten, die SecDocs zur Verfügung stellt, finden Sie im Handbuch "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "[Literatur](#)").

1.2 Darstellungsmittel

In diesem Handbuch wird folgende formale Darstellung verwendet:

Schreibmaschinenschrift

 feste Angaben, die genau in dieser Form ein- oder ausgegeben werden, wie z.B. Schlüsselwörter, URLs oder Dateinamen.

Kursive Schrift

 variable Teile, für die Sie konkrete Angaben einsetzen müssen.

alternative1 | *alternative2*

 Alternativen; die senkrechten Striche dürfen nicht angegeben werden.



Verweis auf detaillierte Informationen zum jeweiligen Thema.



Hinweistexte.



Warnhinweise.

2 Konzepte und Funktionen

Eine ausführliche Beschreibung der grundlegenden Konzepte, auf denen SecDocs basiert, finden Sie im Kapitel „Konzepte und Funktionen“ des Handbuchs ["SecDocs Administration und Bedienung"](#) ([SD1] im Abschnitt ["Literatur"](#)). Ausgenommen ist der Abschnitt „Architektur“ in diesem Kapitel, der nicht für die Archivierung von XAIP-Datenobjekten gilt. Diese Architektur wird im Folgenden beschrieben.

2.1 Architektur

Die Middleware SecDocs ist zentraler Baustein in der IT-Architektur für eine beweiswerterhaltende Archivierung. Mit SecDocs werden die unabdingbaren Anforderungen der technischen Richtlinie TR-03125 (V1.2) des BSI zuverlässig umgesetzt.

2.1.1 Komponenten von SecDocs

Die Architektur von SecDocs folgt der empfohlenen IT-Referenzarchitektur, die in der technischen Richtlinie TR-03125 beschrieben ist (siehe Abschnitt „Empfohlene IT-Referenzarchitektur“ im Dokument "BSI TR-03125" ([W1] im Abschnitt "Literatur")).

Die systemlogischen Komponenten sind zu einer Ablaufeinheit zusammengefasst.

Die Archivierungsfunktionen werden über die ArchiSafe-Schnittstelle (Schnittstelle S.4) angesprochen. Die Funktionen der Schnittstellen S.1 und S.6 sind durch interne Funktionsaufrufe realisiert und sind von außen (d.h. außerhalb der Middleware SecDocs) nicht ansprechbar. Diese Realisierung folgt dem Sicherheitsprinzip, nur Funktionen bereitzustellen, die von der Anwendung auch tatsächlich genutzt werden.

SecDocs besteht aus folgenden Komponenten und Schnittstellen::

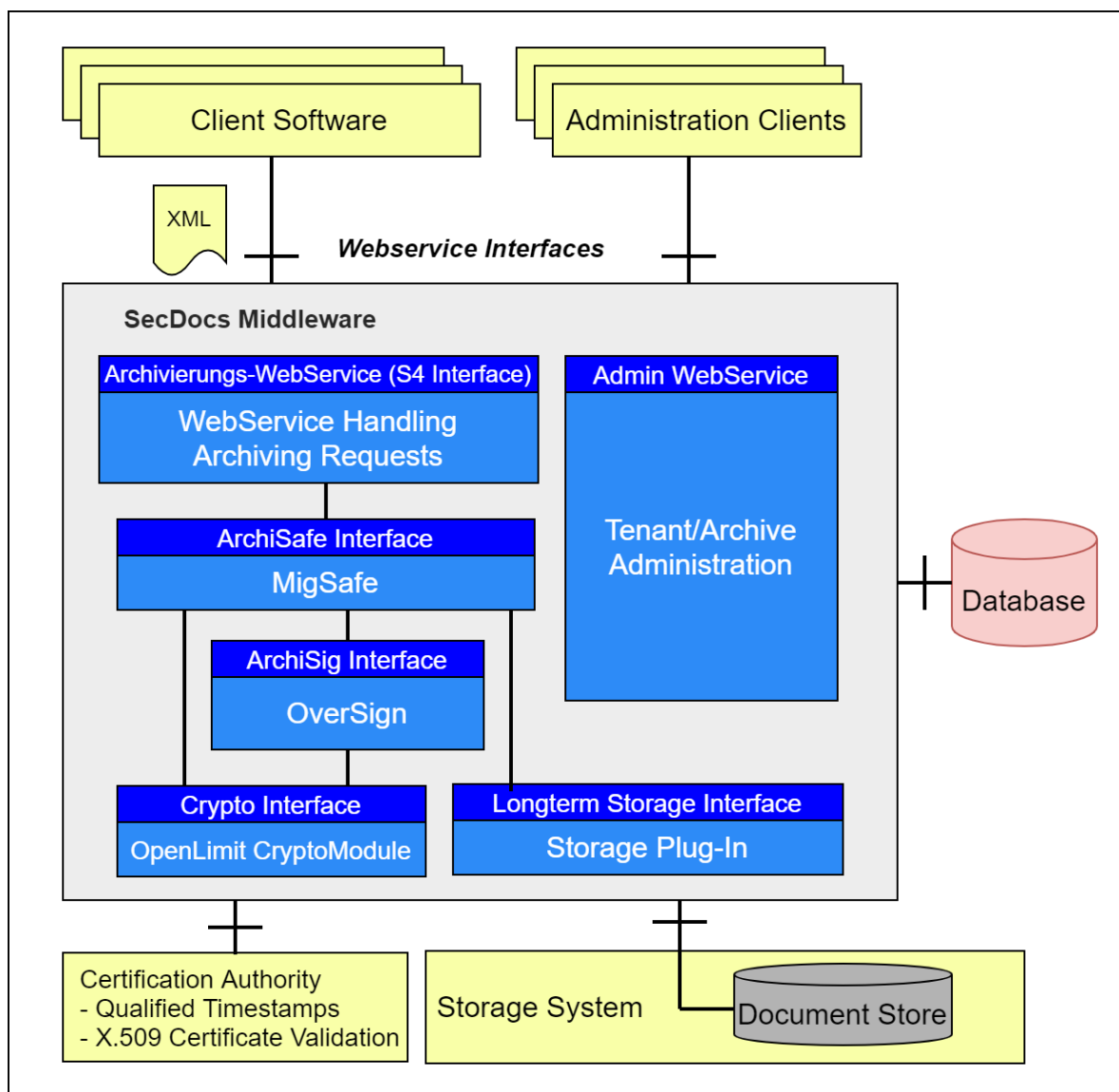


Bild 1: Komponenten von SecDocs (blau dargestellt) und Einbettung in die kundenspezifische Umgebung

WebService Handling, Archiving Requests

Web-Service als Schnittstelle für die Anbindung der Fachanwendungen (auch Client-Anwendungen genannt) wie DMS-, ERP- oder BPM-Systeme sowie fach- oder kundenspezifische Lösungen. Dieser Web-Service bietet Basisfunktionen, die laut TR-03125 gefordert sind.



Die ausführliche Beschreibung dieser Schnittstelle finden Sie im [Kapitel „Web-Service S4 für die Client-Anwendung“](#).

MigSafe, OverSign, OpenLimit CryptoModule

Krypto-Komponenten zum Nachweis der Integrität und Authentizität des archivierten Schriftguts.

- Überprüfen elektronischer Signaturen, die ggf. im Schriftgut enthalten sind und Erstellen der zugehörigen Prüfprotokolle,
- Bilden und Prüfen der Hash-Werte,
- Zusammenfassen der Hash-Werte von mehreren Archivobjekten zu einem gemeinsamen Hash-Wert (ArchiSig),
- Einholen von Zeitstempeln von ausgewählten Zeitstempelanbietern,
- Erzeugen von Evidence Records als Integritätsnachweis für die einzelnen Archivobjekte,
- Prüfen von Evidence Records und Erstellen eines Prüfberichts,
- Erneuern von Hash-Werten und Zeitstempeln sowie Erzeugen der zugehörigen erweiterten Evidence Records,
- Anbindung an entsprechende Zertifizierungsdiensteanbieter, die qualifizierte Zertifikate und Zeitstempel ausstellen.

Die Evaluierung der Komponenten MigSafe/OverSign erfolgt konform zum Schutzprofil BSI-PP-0049.

Die Komponenten MigSafe und OverSign sind von SecDocs unabhängige Teile, die jedoch fest in SecDocs eingebunden sind. Von außerhalb sind diese Komponenten nicht direkt ansprechbar, sondern nur implizit über die Schnittstellen der SecDocs-Web-Services `ArchivingService` und `S4`.

Admin Web Services

Web-Services als Schnittstelle für die Anbindung von Administrations-Clients.

Die Aufgaben der Administration sind das Verwalten des Gesamtarchivs (Web-Service `ArchiveAdminService`) und das Verwalten der einzelnen mandantenspezifischen Archivbereiche (Web-Service `MandantAdminService`).



Die ausführliche Beschreibung dieser Schnittstelle finden Sie im Kapitel „Archiv- und Mandantenadministration“ des Handbuchs "[SecDocs Administration und Bedienung](#)" ([SD1] im [Abschnitt "Literatur"](#)), zusätzliche Operationen und Hinweise zur Administration von S4 sind im [Kapitel „Administration für den Web-Service S4“](#) beschrieben.

Langzeitspeicher-Schnittstelle

Anbindung unterschiedlicher Storage-Systeme.

Die aktuelle Version von SecDocs wird mit einem Plug-In für ein NAS-Filesystem (Network Attached Storage) ausgeliefert.

Der Zugriff auf den Langzeitspeicher erfolgt ausschließlich über die ArchiSafe- bzw. die ArchiSig-Komponenten.

2.1.2 Software-Umgebung

Eine Beschreibung der Bestandteile des Lieferumfangs von SecDocs und der Voraussetzungen bzgl. der Software-Umgebung finden Sie im Abschnitt „Software-Umgebung“ des Handbuchs "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "Literatur").

2.1.3 Ablagestruktur im Archiv

Eine ausführliche Beschreibung der Ablagestruktur im Archiv finden Sie im Abschnitt „Ablagestruktur im Archiv“ des Handbuchs "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "[Literatur](#)").

2.1.4 Storage-Systeme

Die physische Speicherung der archivierten Dokumente ist keine Komponente von SecDocs. Für diese Funktion werden am Markt etablierte Storage-Systeme eingesetzt.

SecDocs enthält Storage Plug-Ins, die für die jeweiligen Storage-Systeme entwickelt und zur Verfügung gestellt werden.

i Zu Testzwecken kann für die Ablage auch das lokale Filesystem verwendet werden.

2.2 Sicherer Betrieb

Dieses Kapitel enthält folgende Abschnitte:

- Anforderungen für den sicheren Betrieb
- Sichere Kommunikation
- Logging

2.2.1 Anforderungen für den sicheren Betrieb

Die Ablaufumgebung muss vor Angriffen von außen (zum Beispiel durch Schadsoftware, nicht zugelassene Netzwerkverbindungen oder Viren) geschützt sein. Dazu muss ein leistungsfähiger Virens Scanner und eine sicher eingestellte Firewall eingesetzt werden. Der Virens Scanner muss regelmäßig aktualisiert werden. Für das Betriebssystem müssen regelmäßig die verfügbaren Sicherheitsupdates eingespielt werden.

SecDocs sowie die dem Produkt unterliegende IT-Plattform müssen Sie in einer sicheren Umgebung installieren. Auf den Servern, auf denen SecDocs abläuft bzw. auf denen sich Daten befinden, auf die SecDocs zugreifen soll, darf keine weitere Software installiert sein, die nicht für den Betrieb des Integrationsproduktes benötigt wird.

Die Server müssen gegen unautorisierten, physischen und logischen Zugriff sowie gegen Veränderung geschützt sein.

Alle TCP/IP-Verbindungen, die von SecDocs genutzt werden, müssen gegen unautorisierte Zugriffe und Abhörmöglichkeiten physisch (z.B. durch Isolierung des verwendeten lokalen Netzes) oder logisch (z.B. mit Verschlüsselungssoftware und Firewalls) geschützt sein.

Das als Basis für SecDocs dienende Betriebssystem muss vom Administrator des Produkts stets aktuell gehalten werden, d.h. die vom Hersteller des Betriebssystems veröffentlichten, sicherheitskritischen Updates sind einzuspielen. Ebenso muss der Administrator ein adäquates Virenschutzprogramm mit aktuellen Virensignaturen verwenden. Die Systemzeit der Server muss mit einer vertrauensvollen Zeitquelle synchronisiert sein (z.B. durch eine Funkuhr oder durch Verbindung mit einem NTP-Server).

Der von der SecDocs Security-Komponente "OpenLimit [CryptoModule](#)" angebotene Web-Service darf beim Betrieb des Produkts ausschließlich lokal angeboten werden, er muss also bei der Installation des Produkts auf das sogenannte "localhost"-Interface gebunden werden.

Die IT-Infrastruktur muss durch Komponenten (von Drittherstellern) vor Viren und Schadsoftware geschützt sein. Weiterhin muss die IT-Infrastruktur vor Netzwerk-basierten Angriffen geschützt sein. Potenzielle Angriffe über das Internet, ein angeschlossenes Intranet, einen manuellen Zugriff Unbefugter oder Datenaustausch per Datenträger müssen durch die bestehenden Sicherheitsvorkehrungen in der Einsatzumgebung mit hoher Sicherheit abgewehrt werden.

Die SecDocs-Komponenten und die von ihnen genutzten Daten müssen durch geeignete Schutzmechanismen davor geschützt werden, dass sie nichtautorisiert verändert werden.

Der Administrator muss vertrauenswürdig sein und die im Handbuch beschriebenen Installations- und Bedienungsanweisungen sorgfältig befolgen. Weiterhin hat er alle ihm zugänglichen Informationen zum Produkt vertraulich zu behandeln. Die benutzten Passwörter sind von ihm sicher zu verwahren. Insbesondere soll der Archivadministrator das Initialpasswort nach der Erstinstallation umgehend ändern.

Für den den nach TR-ESOR zertifizierten Betrieb muss SecDocs so konfiguriert werden, dass die Web-Services für die Administration nur über eine HTTPS-Verbindung erreichbar sind. Der Archivadministrator darf nur solche Mandanten einrichten, die den Web-Service `s4` nutzen können, so dass für die Archivierung von Dokumenten ausschließlich der Web-Service `s4` zur Verfügung steht. Darüber hinaus müssen die Properties `checkTspProductionTime` und `tresor.certified` im Betrieb beide auf `true` gesetzt sein.

2.2.2 Sichere Kommunikation

Die TR-ESOR-Spezifikation des BSI fordert einen sicheren Kommunikationskanal ("Trusted Channel") mit beidseitiger zertifikatsbasierter Authentisierung vor jeglicher Kommunikation zwischen der Client-Anwendung und der TR-ESOR-Middleware (hier SecDocs). Damit soll der Schutz der Integrität und Vertraulichkeit bei der Übertragung sowie eine Authentifizierung der Anfragen und Antworten sichergestellt werden.

Zu diesem Zweck wird in SecDocs zwischen der Client-Anwendung und dem SecDocs-Web-Service `s4` ausschließlich über HTTPS-Request/Reply unter Verwendung von Client- und Server-Zertifikaten kommuniziert. Dadurch kann die Client-Anwendung sicher sein, wirklich mit dem gewünschten Server verbunden zu sein, und der SecDocs-Server kann sicher sein, dass ein eintreffender Request von einem ihm bekannten Client stammt. Für die HTTPS-Kommunikation verwendet SecDocs das Protokoll TLS Version 1.2.

Der Zugriff auf den Web-Service `s4` ist erst nach einer erfolgreichen gegenseitigen Authentifizierung zwischen der Client-Anwendung und dem Web-Service `s4` möglich. SecDocs führt die gegenseitige Authentifizierung bei jedem Auftrag der Client-Anwendung an den Web-Service `s4` durch; es wird also kein Tunnel aufgebaut, der dann permanent aufrecht erhalten wird.

Nähere Informationen, z.B. Hinweise zum Einrichten einer HTTPS-Connector-Konfiguration und zum Einbringen von Zertifikaten, finden Sie im Handbuch "[SecDocs Installationsanleitung](#)" ([SD3] im Abschnitt "[Literatur](#)").

2.2.3 Logging

SecDocs protokolliert den Zugriff auf die TR-ESOR-Middleware bzw. auf den Langzeitspeicher zu Zwecken der Ablage, des Änderns, des Abrufs der Daten oder des Abrufs von Beweisdaten oder auch des Löschens abgelegter Dokumente und Daten.

Dazu bietet SecDocs die folgenden Möglichkeiten der Protokollierung an:

- Das Audit-Logging zeichnet jeden Aufruf einer Webservice-Operation auf und ermöglicht es, jede Aktion einem Verantwortlichen zuzuordnen.
- Das SecDocs-Logging protokolliert die Aktionen in SecDocs und im WildFly Application Server.

Zusätzlich sollten Sie das Umfeld von SecDocs überwachen.

Zugriff auf die Log-Dateien

Der Zugriff auf die Audit-Log Dateien eines Mandanten ist über den Webservice `MandantAdmin` möglich. Ein direkter Zugriff auf das SecDocs-Logging über die SecDocs-Web-Services ist dagegen nicht möglich. Nur der Systemadministrator des Produkts kann mit den Mitteln des Dateiverwaltungssystems direkt auf die Log-Dateien zugreifen.

Der Systemadministrator muss daher vertrauenswürdig sein und die Log-Dateien müssen durch geeignete Schutzmechanismen vor unautorisierter Veränderung geschützt werden.

Für Zugriffe auf die Log-Dateien können Sie z.B. eine eigene Benutzerkennung einrichten und diese Kennung der Gruppe `secdocs` hinzufügen.

Das Logging in SecDocs ist ausführlich im Abschnitt „Logging und Fehlerbehandlung“ des Handbuchs "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "[Literatur](#)") beschrieben.

Im Handbuch "[SecDocs Installations- und Einsatz-Anleitung für SecDocs und V3 Server](#)" ([SD3] im Abschnitt "[Literatur](#)") finden Sie Hinweise, wie sie die Logging-Konfiguration anpassen können.

3 Administration für den Web-Service S4

SecDocs bietet für die Administration ein zweistufiges Konzept. Die Administrationsschnittstelle unterscheidet zwischen der Administration des gesamten Archivs und der Mandantenadministration.

- Administration des gesamten Archivs:

Die Operationen für die Administration des Archivs in seiner Gesamtheit werden mit Hilfe des Web-Service `ArchiveAdminService` bereitgestellt. Dazu gehören unter anderem Operationen zum Einrichten der verwendbaren Zeitstempelanbieter sowie das Einrichten und Verwalten von Mandanten. Über die Archivadministration ist kein Einblick in die fachlichen und organisatorischen Gegebenheiten der einzelnen Mandanten möglich.

Eine ausführliche Beschreibung dieser Schnittstelle finden Sie im Abschnitt „Web-Service `ArchiveAdminService`“ des Handbuchs "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "[Literatur](#)").

- Mandantenadministration:

Die Operationen, die sich auf den jeweiligen Archivbereich eines Mandanten beziehen, werden mit Hilfe des Web-Service `MandantAdminService` bereitgestellt. Dazu gehören unter anderem Operationen zum Registrieren von Dokumententypen sowie zum Einrichten von spezifischen Zugängen zum Archiv für die Client-Software.

Eine ausführliche Beschreibung dieser Schnittstelle finden Sie im Abschnitt „Web-Service `MandantAdminService`“ des Handbuchs "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "[Literatur](#)").

Für die Archivierung stehen die Web-Services `S4` und `ArchivingService` zur Verfügung. Diese beiden Web-Services unterscheiden sich hinsichtlich der Struktur der Archivdatenobjekte:

- Beim Web-Service `S4` werden Datenobjekte im Format XAIP (XML formatted Archive Information Package) archiviert. Dieses Format ist im Anhang F der Richtlinie TR-ESOR V1.2 (siehe Dokument "[BSI TR-03125 Anlage TR-ESOR-F](#)" ([W5] im Abschnitt "[Literatur](#)")) festgelegt und steht mit der Installation von SecDocs zur Verfügung. Beim Web-Service `ArchivingService` werden XML-Container archiviert, die Submission Data Objects (SDOs), deren Dokumententyp (SDO-Typ) vom Mandantenadministrator eingetragen wird.

Ein Mandant kann in SecDocs immer nur mit einem dieser beiden Web-Services arbeiten, d.h. der Mandant kann entweder nur die Operationen des Web-Service `ArchivingService` oder nur diejenigen des Web-Service `S4` nutzen. Diese Festlegung müssen Sie bereits beim Anlegen eines Mandanten vornehmen und Sie können sie nachträglich auch nicht mehr ändern. Aus diesem Grund können Sie zum Archivieren von Dokumenten mit dem Web-Service `S4` auch keine Mandanten aus einer SecDocs Version kleiner als V3.0 verwenden, da diese ausschließlich für das Archivieren von SDOs eingerichtet sind.

3.1 Mandanten für den Web-Service S4 erzeugen

Für den SecDocs Web-Service S4 müssen Sie eigene Mandanten einrichten. Sie können dafür keine Mandanten nutzen, die Sie mit `createMandant` für den Web-Service `ArchivingService` eingerichtet haben.

Dazu gehen Sie folgendermaßen vor:

1. Erzeugen Sie einen Mandanten mit der Operation `createMandantXAIP` des Web-Service `ArchiveAdminService` (siehe [Abschnitt „Operation createMandantXAIP“](#)).
2. Erzeugen Sie für diesen Mandanten eine Organisation mit der Operation `createOrganisation` des Web-Service `MandantAdminService` (siehe Abschnitt „Operation createOrganisation“ im Handbuch ["SecDocs Administration und Bedienung" \(\[SD1\] im Abschnitt "Literatur"\)](#)). Secdocs erzeugt zu der neuen Organisation auch bereits die organisationsspezifische Rolle `Archivar`.
3. Machen Sie SecDocs das Client-Zertifikat bekannt, das Sie für die Kombination aus Mandant, Organisation und der Rolle (standardmäßig Rolle `Archivar`) verwenden wollen. Dazu steht Ihnen die Operation `setCredentials` des Web-Service `MandantAdmin-Service` zur Verfügung (siehe Abschnitt „Operation setCredentials“ im Handbuch ["SecDocs Administration und Bedienung" \(\[SD1\] im Abschnitt "Literatur"\)](#)).

Beispiel für den Request-Body eines setCredentials-Requests:

```
<soap:Body>
  <Credentials xmlns="http://ts.fujitsu.com/secdocs/v3_0/secdocs"
    xmlns:ns2="http://ts.fujitsu.com/secdocs/v3_0/adminData"
    xmlns:ns3="http://ts.fujitsu.com/secdocs/v3_0/adminUpdateData">
    <Type>Certificate</Type>
    <Credits>certificate as base64Binary</Credits>
    <Role>Archivar</Role>
    <Mandant>tenant name</Mandant>
    <OrgID>organisation name</OrgID>
  </Credentials>
</soap:Body>
```

Danach kann eine Client-Anwendung das organisationsspezifische Zertifikat verwenden und sich mit `https://servername:8444/...` an den Web-Service S4 wenden.

Beachten Sie, dass Sie für jede Organisation des Mandanten ein eigenes Zertifikat benötigen.

Nähere Hinweise finden Sie im Handbuch ["SecDocs Installationsanleitung" \(\[SD3\] im Abschnitt "Literatur"\)](#).

3.2 Zertifikate einbringen

Der Zugriff auf den Web-Service *s4* ist erst nach einer erfolgreichen gegenseitigen Authentifizierung zwischen der Client-Anwendung und dem Web-Service *s4* möglich.

Server-Zertifikat

Ein Server-Zertifikat ist einem Rechner zugeordnet. Für die Kommunikation über HTTPS wird für jeden Rechner, auf dem SecDocs läuft, ein Zertifikat benötigt.

Das Zertifikat der Zertifizierungsstelle, die die Server-Zertifikate ausstellt, muss in der Client-Anwendung als vertrauenswürdig bekannt sein.

Client-Zertifikat

Ein Client-Zertifikat ist einer Person oder einer bestimmten Client-Software zugeordnet. In SecDocs gilt ein Client-Zertifikat für eine bestimmte Kombination aus Mandant, Organisation und Rolle.

Alle in einer Client-Anwendung verwendeten Client-Zertifikate sollten von einer gemeinsamen Zertifizierungsstelle ausgestellt (signiert) sein. Das Zertifikat dieser Zertifizierungsstelle müssen Sie dem SecDocs zugrunde liegenden Applikationsserver WildFly als vertrauenswürdig bekannt machen.

Eine Anleitung, wie Sie die einzelnen Zertifikate in die SecDocs-Installation einbringen, finden Sie im Handbuch "[SecDocs Installationsanleitung](#)" ([SD3] im Abschnitt "[Literatur](#)").

Mandantentrennung

In SecDocs muss für jede Kombination aus Mandant, Organisation und Rolle ein eigenes Client-Zertifikat erzeugt werden. Außerdem besitzt jeder Mandant einen eigenen Teilbereich im Archiv (siehe Abschnitt „Ablagestruktur im Archiv“ im Handbuch "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "[Literatur](#)").

Damit ist SecDocs in der Lage, getrennte Mandanten zu verwalten. Eine aufrufende Client-Software kann nur auf diejenigen Archivdatenobjekte zugreifen, für die sie eine Zugriffsberechtigung besitzt.

3.3 Operationen für die Administration

Beim Web-Service *s4* werden Datenobjekte im Format XAIP (XML formatted Archive Information Package) archiviert. Dieses Format steht mit der Installation von SecDocs zur Verfügung.

Einen Mandanten, der Datenobjekte im XAIP-Format archivieren soll, richten Sie mit der Operation `createMandantXAIP` ein.

Beachten Sie: Diese Festlegung kann für den Mandanten nicht mehr geändert werden.

Näheres zur Operation `createMandantXAIP` finden Sie im [Abschnitt „Operation createMandantXAIP“](#).

Um festzustellen, ob ein Mandant für die Archivierung mit dem Web-Service *s4* eingerichtet ist, verwenden Sie die Operation `getMandants` oder `getMandantProperties` (siehe [Abschnitt „Operationen getMandants, getMandantProperties, getArchiveInfo“](#)).

Das Format XAIP ist im Anhang F der Richtlinie TR-ESOR V1.2 beschrieben (siehe Dokument "[BSI TR-03125 Anlage TR-ESOR-F](#)" ([W5] im [Abschnitt "Literatur"](#))). Die prinzipielle Struktur eines Datenobjekts, das mit *s4* archiviert werden soll, ist damit festgelegt.

Das XAIP-Schema sieht jedoch *Extension*-Elemente für benutzerspezifische Erweiterungen vor. Damit können Sie das XAIP-Format an Ihre eigenen Bedürfnisse anpassen.

Die zusätzlichen Erweiterungen müssen durch eigene Schemadateien beschrieben werden, um die Validierung eines Datenobjekts bei der Archivierung zu ermöglichen.

SecDocs bietet zur Unterstützung einer solchen benutzerspezifischen Erweiterung des XAIP-Schemas die Operation `modifyXAIP` an. Damit können Sie für jeden Mandanten das vorhandene XAIP-Format erweitern. Näheres finden Sie im [Abschnitt „Operation modifyXAIP“](#).

Um festzustellen, ob und ggf. welche Erweiterungen des XAIP-Schemas für einen bestimmten Mandanten definiert wurden, verwenden Sie die Operation `getSDOTypes` (siehe [Abschnitt „Operation getSDOTypes“](#)).

3.3.1 Operation createMandantXAIP

Die Operation `createMandantXAIP` ist eine Operation des Web-Service `ArchiveAdminService`.

`createMandantXAIP` legt einen neuen Mandanten für die Archivierung von Dokumenten mit dem Web-Service `S4` an. Dieser Mandant kann dann alle Operationen des Web-Service `S4` nutzen und Datenobjekte im Format XAIP archivieren.

Eine Archivierung von Dokumenten mit dem Web-Service `ArchivingService` ist für einen mit `createMandantXAIP` eingerichteten Mandanten nicht möglich. Umgekehrt kann ein mit `createMandant` eingerichteter Mandant die Operationen des Web-Service `S4` nicht nutzen.

Beachten Sie:

Die Festlegung des zur Verfügung stehenden Web-Service kann für einen Mandanten nicht mehr nachträglich geändert werden.

Der Zugang zum Web-Service `S4` erfordert für einen mit `createMandantXAIP` eingerichteten Mandanten die Einrichtung eines Zertifikats. Näheres hierzu finden Sie im [Abschnitt „Zertifikate einbringen“](#).

`createMandantXAIP` führt folgende Aktionen aus:

- Definition des Mandanten: Name und Kontaktinformation
- Definition des zugehörigen Archivbereichs: Dateipfad direkt unterhalb der Archiv-Wurzel, unter der alle Dateien und Verzeichnisse für diesen Mandanten auf dem Speichersystem abgelegt werden (mandantenspezifischer Archivbereich). Die Archiv-Wurzel wird mit dem Parameter `archiveRoot` in der Datei `secdocs.properties` konfiguriert. Die Beschreibung der Konfigurationsdatei `secdocs.properties` finden Sie im Handbuch ["SecDocs Administration und Bedienung" \(\[SD1\] im Abschnitt "Literatur"\)](#).
- Spezifikation der für die Versiegelung zu nutzenden TSPs
- Festlegen des Zugangs (Credentials) für die Rolle `MandantAdmin`
- Erzeugen der Rolle `SecDocs_MandantAuditor`
- Festlegen von Einstellungen für die Ablage der Archivdatenobjekte (`SDOPath`, `SoftWorm`) und Signaturprüfung (`SignatureVerification`, `SignatureQualityLevel`, `SignatureEmbedded`, `SignatureDetached`)
- Einstellen der Konfigurationsparameter `TreeSize` und `TreeAge`
- Einstellen von mandantenspezifischen Konfigurationsparametern

Voraussetzungen

- Die angegebenen TSPs müssen in SecDocs angelegt sein (siehe Abschnitt „Operation createTSP“ im Handbuch ["SecDocs Administration und Bedienung" \(\[SD1\] im Abschnitt "Literatur"\)](#)).
- Der Pfad `archiveRoot/Path` muss entweder existieren und mit Schreibrecht (für den Linux-Benutzer `secdocs`) zugreifbar sein (eventuell auch als Mount in Absprache mit dem Systemadministrator), oder er kann durch `createMandantXAIP` angelegt werden, falls dies im Parameter `createMandantDirLocal` in der Datei `secdocs.properties` eingestellt ist.

Beachten Sie:

- Die Operation `createMandantXAIP` wird abgewiesen, wenn für einen mandantenspezifischen Konfigurationsparameter ein ungültiger Wert angegeben wird, z.B. ein kleinerer Wert als der festgelegte Minimalwert bzw. ein größerer Wert als der festgelegte Maximalwert.

Request

Element CreateMandantXAIP *vom Datentyp* CreateMandantXAIPType

```
<xsd:complexType name="CreateMandantXAIPType">
  <xsd:sequence>
    <xsd:element name="Mandant" type="tns:MandantType" maxOccurs="1"
      minOccurs="1">
    </xsd:element>
    <xsd:element name="Credentials" type="tns:CredentialType"
      maxOccurs="1"
      minOccurs="1">
    </xsd:element>
    <xsd:element name="XAIPSubmission" type="tns:XAIPSubmissionType">
    </xsd:element>
    <xsd:element name="Policy" maxOccurs="1" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:base64Binary">
          <xsd:minLength value="1"></xsd:minLength>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="TimestampPolicy" maxOccurs="1"
      minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:base64Binary">
          <xsd:minLength value="1"></xsd:minLength>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

Mandant

Basisdaten für den Mandanten (Name, Kontaktinformation, Archiv, Konfigurationsparameter, Liste der TSPs).

Datentyp MandantType.

Die Beschreibung des Datentyps MandantType finden Sie im Handbuch "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "Literatur").

Credentials

Zugangsdaten für die Rolle MandantAdmin.

Für alle anderen Rollen müssen Sie die Zugangsdaten mit der Operation setCredentials festlegen (siehe [Abschnitt „Operation setCredentials - Zugang zum Web-Service S4“](#)).

Datentyp CredentialType.

XAIPSubmission

Konfigurationsparameter für die Operation ArchiveSubmission, z.B. Ablageort der Datenobjekte im Archiv, Umgang mit Signaturen, usw.

Datentyp XAIPSubmissionType.

Beachten Sie:

- Die für XAIPSubmission vorgenommenen Einstellungen können - mit Ausnahme des Ablageorts für die Datenobjekte im Archiv (Operand SDOPath) - nicht mehr geändert werden.
- Zum Ändern des Ablageorts verwenden Sie die Operation `updateMandant` des Web-Service `MandantAdminService`. Setzen Sie dort beim Operanden `Properties` die Property mit dem Namen `$SDOPath` auf den gewünschten Wert.
- Die für XAIPSubmission vorgenommenen Einstellungen können Sie mit einer der Operationen `getMandants` (Web-Service `ArchiveAdminService`) oder `getMandantProperties` bzw. `getArchiveInfo` (Web-Service `MandantAdminService`) ansehen. Näheres finden Sie im [Abschnitt „Operationen getMandants, getMandantProperties, getArchiveInfo“](#)

Datentyp `CredentialType`

```
<xsd:complexType name="CredentialType">
  <xsd:sequence>
    <xsd:element name="Type">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Password"></xsd:enumeration>
          <xsd:enumeration value="Certificate"></xsd:enumeration>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:choice>
      <xsd:element name="Credits">
        <xsd:simpleType>
          <xsd:restriction base="xsd:base64Binary">
            <xsd:minLength value="8"></xsd:minLength>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="Password" type="tns:PasswordSimpleType">
      </xsd:element>
    </xsd:choice>
    <xsd:element name="Role" type="xsd:string" maxOccurs="1"
                  minOccurs="0">
    </xsd:element>
    <xsd:element name="Mandant" type="xsd:string" minOccurs="0">
    </xsd:element>
    <xsd:element name="OrgID" type="xsd:string" maxOccurs="1"
                  minOccurs="0">
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

Type string:
 Typ des Credential.
 Mögliche Werte: Password | Certificate
 Für die Rolle MandantAdmin wird nur der Typ Password unterstützt.

Credits	base64Binary: Möglichkeit der Autorisierung Wird für die Rolle <code>MandantAdmin</code> nicht unterstützt.
Password	string: Initiales Passwort für die Rolle <code>MandantAdmin</code> . Mindestlänge: 8
Role	string <code>MandantAdmin</code> : Rolle für den Zugang zum Web-Service <code>MandantAdminService</code> . Eine andere Rolle können Sie hier nicht angeben.
Mandant	string: Dieses Element darf nicht angegeben werden.
OrgID	string: Dieses Element darf nicht angegeben werden.

Datentyp XAIPSubmissionType

```
<xsd:complexType name="XAIPSubmissionType">
  <xsd:sequence>
    <xsd:element name="SDOPath" type="tns:NonEmptyString"
      default="*$Year*/*$Month*/*$Day*"
      maxOccurs="1" minOccurs="0">

    </xsd:element>
    <xsd:element name="SoftWorm" type="tns:SoftWormType"
      default="NO"
      maxOccurs="1" minOccurs="0">

    </xsd:element>
    <xsd:element name="SignatureVerification"
      type="tns:SignatureVerificationType"
      default="INFORMATION"
      maxOccurs="1" minOccurs="0">

    </xsd:element>
    <xsd:element name="SignatureQualityLevel"
      type="tns:SignatureQualityLevelType"
      default="ADVANCED">
      maxOccurs="1" minOccurs="0">

    </xsd:element>
    <xsd:element name="SignatureEmbedded"
      type="tns:YesNoAutoType"
      default="AUTO"
      maxOccurs="1" minOccurs="0">

    </xsd:element>
    <xsd:element name="SignatureDetached"
      type="tns:YesNoAutoType"
      default="YES"
      maxOccurs="1" minOccurs="0">

    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

SDOPath

NonEmptyString:

Gibt den Ablageort des Archivdatenobjekts im Archiv an. Der Pfad ist relativ zu dem Pfad, der mit der Operation `createOrganisation` festgelegt wurde.

Mögliche Werte: *pfadname*

Der angegebene Pfadname darf nicht mit einem Schrägstrich (/) oder mit einem Unterstrich (_) beginnen. Er darf keinen der Strings „./“ oder „/_“ enthalten.

Standardwert: `*$Year*/*$Month*/*$Day*`

Weitere Informationen siehe Beschreibung des Systemschlüssels `$SDOPath` im Handbuch "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "Literatur").

Softworm

Gibt an, ob das Archivdatenobjekt bis zum Ende der Aufbewahrungsfrist als SoftWORM-Datum im Langzeitspeicher abgelegt wird.

Mögliche Werte: YES | NO

YES Die Daten werden als SoftWORM-Daten auf dem Storage abgelegt und auf read-only gesetzt. Wenn das Storage-System die SoftWORM-Eigenschaft nicht unterstützt, werden die Daten nur auf read-only gesetzt.

Abhängig von den Einstellungen auf dem Storage-System kann der Zeitpunkt, ab dem die Daten tatsächlich gelöscht werden können, nach dem ermittelten Freigabezeitpunkt liegen.

Standardwert: NO

Siehe hierzu auch die Beschreibung des Systemschlüssels `$SoftWorm` im Handbuch "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "[Literatur](#)").

SignatureVerification

Legt fest, welches Ergebnis die Auswertung der Signaturen in einem Datenobjekt mindestens erreichen muss, damit eine Archivierung durchgeführt wird. Das Datenobjekt wird nur archiviert, wenn das Ergebnis der Verifikation höher oder gleich dem Wert dieses Operanden ist.

Mögliche Werte: SUCCESS | INFORMATION | CERTIFICATE

SUCCESS

Alle Prüfungen müssen möglich und erfolgreich sein, d.h. die Verifikation muss ohne Fehler verlaufen, inklusive der Prüfung, ob ein Zertifikat gesperrt ist.

INFORMATION

Alle Prüfungen müssen möglich und erfolgreich sein. Für die Prüfung, ob ein Zertifikat gesperrt wurde, dürfen aber Sperrlisten herangezogen werden, deren Ausgabezeitpunkt vor dem Signatur-Erstellungszeitpunkt liegt. Der Wert ist also nur relevant, wenn zur Signaturprüfung CRLs und keine OCSP-Antworten verwendet werden.

CERTIFICATE

Die Zertifikatskette muss mindestens ein vertrauenswürdigen Zertifikat enthalten. Die Einstellung `CERTIFICATE` dient hauptsächlich zu Testzwecken, da eine Prüfung auf Sperrung des Zertifikats unterbleibt.

Standardwert: INFORMATION

Beachten Sie:

- Das Datenobjekt wird bei der Archivierung abgewiesen, wenn das Ergebnis der Verifikation kleiner als der Wert des Operanden ist.
- Die Archivierung wird mit Fehler abgebrochen, wenn eines der verwendeten Zertifikate gesperrt ist.

Siehe hierzu auch die Beschreibung des Systemschlüssels
\$SignatureVerification im Handbuch "[SecDocs Administration und Bedienung](#)" ([SD1] im
Abschnitt "[Literatur](#)").

SignatureQualityLevel

Legt die Anforderungen an die elektronische Signatur bzw. an den Zertifizierungsdiensteanbieter fest, die für eine Archivierung mit SecDocs notwendig sind.

Sollen elektronisch signierte Datenobjekte in SecDocs archiviert werden, werden die Signaturen geprüft (Operation ArchiveSubmission). Das Datenobjekt mitsamt dem dort eingefügten Prüfprotokoll wird nur bei genügend erfolgreicher Verifikation archiviert, andernfalls wird die Archivierung abgewiesen.

Mögliche Werte: ADVANCED | QUALIFIED | ACCREDITED

ADVANCED

Fortgeschrittene elektronische Signatur; diese Signatur ermöglicht eine Identifizierung des Unterzeichners.

QUALIFIED

Fortgeschrittene elektronische Signatur, die auf einem qualifizierten Zertifikat beruht. Jede QUALIFIED-Signatur ist auch ADVANCED.

ACCREDITED

Fortgeschrittene elektronische Signatur, die auf einem qualifizierten Zertifikat beruht. Das Zertifikat wurde von einer akkreditierten Stelle erstellt.
Jede ACCREDITED-Signatur ist auch QUALIFIED.

Standardwert: ADVANCED

Siehe hierzu auch die Beschreibung des Systemschlüssels
\$SignatureQualityLevel im Handbuch "[SecDocs Administration und Bedienung](#)" ([SD1] im
Abschnitt "[Literatur](#)").

SignatureEmbedded

Gibt an, ob das Datenobjekt eine eingebettete Signatur enthält.

Mögliche Werte: YES | NO | AUTO

YES Im Datenobjekt werden eine oder mehrere eingebettete Signaturen erwartet.

SecDocs führt eine Signaturprüfung für die abgesetzten Signaturen durch, vorausgesetzt, dass das Attribut mimeType des Elements `dataObjectsSection/dataObject/binaryData` einen der folgenden Werte hat (siehe Anmerkung 3 (in "[Format eines Archivdatenobjekts](#)")):

- application/pdf
- application/vnd.pdf
- application/vnd.cups-pdf
- application/x-pdf

-
- NO SecDocs soll keine Signaturprüfung für eine evtl. vorhandene eingebettete Signatur durchführen.
- AUTO Das Datenobjekt kann eine oder mehrere eingebettete Signaturen enthalten.
- Sind eine oder mehrere eingebettete Signaturen vorhanden, werden sie geprüft (sofern das Attribut `MimeType` einen entsprechenden Wert hat, siehe oben bei YES).

Standardwert: AUTO

Beachten Sie:

Ein Datenobjekt wird bei der Archivierung abgewiesen, wenn für `SignatureEmbedded` der Wert YES eingestellt ist, das Datenobjekt jedoch keine eingebettete Signatur enthält.

Siehe hierzu auch die Beschreibung des Systemschlüssels `$SignatureEmbedded` im Handbuch "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "Literatur")

`SignatureDetached`

Gibt an, ob SecDocs beim Archivieren eine abgesetzte Signatur in einem Datenobjekt erwartet.

Mögliche Werte: YES | NO | AUTO

- YES Im Datenobjekt werden eine oder mehrere abgesetzte Signaturen erwartet, d.h. das Element `credentialsSection/credential/SignatureObject/Base64Signature` muss in diesem Fall angegeben sein.
- SecDocs führt eine Signaturprüfung für die abgesetzten Signaturen durch.
- NO SecDocs erwartet keine abgesetzten Signaturen und führt auch keine Signaturprüfung für evtl. vorhandene abgesetzte Signaturen durch.
- AUTO SecDocs wertet das Element `credentialsSection/credential/SignatureObject/Base64Signature` aus, um die abgesetzten Signaturen zu einem Datenobjekt zu ermitteln. Enthält ein Datenobjekt eine oder mehrere abgesetzte Signaturen, dann werden diese geprüft. Andernfalls erfolgt eine Archivierung ohne Signaturprüfung.

Standardwert: YES

Beachten Sie:

- Ein Datenobjekt wird bei der Archivierung abgewiesen, wenn für `SignatureDetached` der Wert YES eingestellt ist, das Datenobjekt jedoch keine abgesetzte Signatur enthält.

Siehe hierzu auch die Beschreibung des Systemschlüssels `$SignatureDetached` im Handbuch "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "Literatur").

Policy

base64Binary:

Datei zur Steuerung der Bewertung des Ergebnisses der Prüfung von eingebetteten oder abgesetzten Signaturen bei ArchiveSubmission.

TimestampPolicy

base64Binary:

Datei zur Bewertung der Prüfung, wenn Zeitstempel als Signaturen mitgegeben werden



Das Erstellen einer Policy Datei ist eine Service-Leistung der Fujitsu GMBH und sollte mit ihrem technischen Betreuer abgestimmt werden.

Response

Leeres Element result

```
<result></result>
```

Beispiel:

Request-Body

```
<soap:Body>
  <CreateMandantXAIP xmlns="http://ts.fujitsu.com/secdocs/v3_0/adminData">
    <Mandant>
      <Name>MandantX01</Name>
      <DisplayName>XAIP mandant 01</DisplayName>
      <Contact>
        <FirstName>John</FirstName>
        <Surname>Doe</Surname>
        <Tel>12345</Tel>
        <Email>Doe@MyCompany.com</Email>
      </Contact>
      <Path>archive/MandantX01</Path>
      <TreeSize>100</TreeSize>
      <TreeAge>2</TreeAge>
      <TSP>SecDocsTestTSP</TSP>
      <State>productive</State>
    </Mandant>
    <Credentials>
      <Type>Password</Type>
      <Password>SecretPasswordOfMandantAdmin</Password>
      <Role>MandantAdmin</Role>
    </Credentials>
    <XAIPSubmission>
      <SDOPath>*$Year*$Month*$Day*</SDOPath>
    </XAIPSubmission>
  </CreateMandantXAIP>
</soap:Body>
```

Response-Body

```
<soap:Body>
  <result xmlns="http://ts.fujitsu.com/secdocs/v3_0/adminData"></result>
</soap:Body>
```

3.3.2 Operation modifyXAIP

Die Operation `modifyXAIP` ist eine Operation des Web-Service `MandantAdminService`.

`modifyXAIP` erweitert das im Anhang F der Richtlinie TR-ESOR V1.2 festgelegte Format XAIP (XML formatted Archive Information Package, siehe Dokument "[BSI TR-03125 Anlage TR-ESOR-F](#)" ([W5] im Abschnitt "Literatur")) um benutzerspezifische Definitionen. SecDocs validiert diese Benutzerdefinitionen, hinterlegt sie intern und zieht sie bei der Archivierung mit `ArchiveSubmission` zur Validierung des übergebenen XAIP-Datenobjekts heran. Darüberhinaus lassen sich auch die für den XAIP-Datentyp hinterlegten Policies ändern

Voraussetzungen

- Die benutzerspezifischen Erweiterungen sind als XML-Schema(s) (als .xsd-Datei(en)) festgelegt.
- Wenn Schemas, die in SecDocs registriert werden sollen, import-Anweisungen enthalten, müssen diese die Attribute `schemaLocation` und `namespace` enthalten.

Hinweise

- Die Operation `modifyXAIP` benötigen Sie nur, wenn für einen Mandanten benutzerspezifischen Erweiterungen des XAIP-Schemas erforderlich sind und diese bei der Archivierung mit `ArchiveSubmission` validiert werden sollen.
- Wenn Sie mit `modifyXAIP` das in TR-ESOR V1.2 festgelegte XAIP-Format für einen Mandanten mehrmals abändern, so erfolgt die Validierung eines übergebenen XAIP-Datenobjekts bei der Archivierung mit `ArchiveSubmission` immer gegen die neueste mit `modifyXAIP` festgelegte Definition.
- Das Erstellen einer Policy Datei ist eine Service-Leistung der Fujitsu GMBH und sollte mit ihrem technischen Betreuer abgestimmt werden.

Request

Element ModifyXAIP vom Datentyp ModifyXAIPType

```
<xsd:complexType name="ModifyXAIPType">
  <xsd:sequence>
    <xsd:element name="DependentSchema"
      type="tns:DependentSchemaType" maxOccurs="unbounded" minOccurs="0">
    </xsd:element>
    <xsd:element name="Policy" maxOccurs="1" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:base64Binary">
          <xsd:minLength value="1"></xsd:minLength>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="TimestampPolicy" maxOccurs="1"
      minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:base64Binary">
          <xsd:minLength value="1"></xsd:minLength>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

DependentSchema

Optional; beliebig viele Einträge vom Datentyp *DependentSchemaType*.

Ein Eintrag beschreibt ein Schema, das benutzerspezifische Erweiterungen des in TR-ESOR V1.2 festgelegten XAIP-Formats enthält.

Datentyp *DependentSchemaType*

```
<xsd:complexType name="DependentSchemaType">
  <xsd:sequence>
    <xsd:element name="Namespace" type="xsd:anyURI"
      maxOccurs="1" minOccurs="0">
    </xsd:element>
    <xsd:element name="SchemaLocation" type="xsd:anyURI"
      maxOccurs="1" minOccurs="1">
    </xsd:element>
    <xsd:choice>
      <xsd:element name="Schema" type="tns:SchemaSimpleType"
        maxOccurs="1" minOccurs="0">
      </xsd:element>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>
```

Namespace

anyURI:

optional; Target-Namespace für das Benutzerschema.

In dem bei der Archivierung übergebenen XAIP-Datenobjekt müssen Sie diesen Namespace im `schemaLocation`-Attribut jedes Elements angeben, das in einem Element `extension` anstelle des `any`-Elements steht.

SchemaLocation

anyURI:

Pfadangabe zur Schemadatei.

In dem bei der Archivierung übergebenen XAIP-Datenobjekt müssen Sie diese Pfadangabe im `schemaLocation`-Attribut jedes Elements angeben, das in einem Element `extension` anstelle des `any`-Elements steht.

Schema

base64Binary (Mindestlänge 1):

XML-Schema, das die Definition der benutzerspezifischen Erweiterungen enthält. Dieses Element muss angegeben werden.

Policy

base64Binary:

Datei zur Steuerung der Bewertung des Ergebnisses der Prüfung von eingebetteten oder abgesetzten Signaturen bei ArchiveSubmission.

TimestampPolicy

base64Binary:

Datei zur Bewertung der Prüfung, wenn Zeitstempel als Signaturen mitgegeben werden



Das Erstellen einer Policy Datei ist eine Service-Leistung der Fujitsu GMBH und sollte mit ihrem technischen Betreuer abgestimmt werden.

Response

Leeres Element `result`

```
<result></result>
```

Beispiel für ein Schema mit benutzerspezifischen Definitionen

Hinweis: In diesem Beispiel sind die korrespondierenden Einträge farbig markiert.

Das Schema enthält z.B. folgende Definition:

```

<xs:schema targetNamespace="http://ts.fujitsu.com/secdocs/myExtensionXAIP"
...
<xs:element name="specialInformation" type="tr:specialInformationType"/>
<xs:complexType name="specialInformationType">
    <xs:sequence>
        <xs:element name="Info1" type="xs:string"/>
        ...
    </xs:sequence>
</xs:complexType>
...

```

Dieses Schema machen Sie SecDocs mit `modifyXAIP` bekannt:

Request-Body

```

<soap:Body>
    <ModifyXAIP xmlns="http://ts.fujitsu.com/secdocs/v3_0/adminData">
        <DependentSchema>
            <Namespace>http://ts.fujitsu.com/secdocs/myExtensionXAIP
            </Namespace>
            <SchemaLocation>
                http://ts.fujitsu.com/secdocs/myExtensionXAIP/XAIPext01.xsd
            </SchemaLocation>
            <Schema>PHhzOn...Y2h1bWE+</Schema>
        </DependentSchema>
    </ModifyXAIP>
</soap:Body>

```

Response-Body

```

<soap:Body>
    <result xmlns="http://ts.fujitsu.com/secdocs/v3_0/adminData"></result>
</soap:Body>

```

XAIP-Datenobjekt, das bei ArchiveSubmission angegeben wird

```
...
<xaip:extension>
  <xaipext:specialInformation
    xmlns:xaipext="http://ts.fujitsu.com/secdocs/myExtensionXAIP"
    xsi:schemaLocation=
      "http://ts.fujitsu.com/secdocs/myExtensionXAIP
        http://ts.fujitsu.com/secdocs/myExtensionXAIP/XAIPext01.xsd">
    ...
  <xaipext:Inf1>some text</xaipext:Inf1>
  ...
</xaipext:specialInformation>
</xaip:extension>
...
```

3.3.3 Operation `setCredentials` - Zugang zum Web-Service S4

Mit `setCredentials` tragen Sie ein Zertifikat für die angegebene Rolle und Organisation ein (standardmäßig ist das die Rolle `Archivar`). Dieses Zertifikat dient als Berechtigungsnachweis für den Zugang zum Web-Service S4.

Eine allgemeine Beschreibung der Operation `setCredentials` des Web-Service `Mandant-AdminService` finden Sie im Abschnitt „Operation `setCredentials`“ für den Web-Service `MMandantAdminService` im Handbuch ["SecDocs Administration und Bedienung"](#) ([SD1] im Abschnitt "Literatur") .

Gültigkeit der Zertifikate in SecDocs

Ein Aufruf von `setCredentials` für die spezifizierte Rolle erlaubt den Zugang mit dem neuen Zertifikat, das "alte" Zertifikat, falls vorhanden, erlaubt aber weiterhin den Zugang. Durch einen weiteren `setCredentials`-Aufruf verliert das alte Zertifikat seine Gültigkeit in SecDocs, und das Zertifikat aus dem vorhergehenden `setCredentials`-Aufruf wird zum "alten" Zertifikat. Der Zugang ist also mit den beiden zuletzt definierten Zertifikaten möglich. Dies erlaubt es, die bei einem Zertifikatswechsel am Server und bei den Archivierungsclients notwendigen Aktionen zeitlich zu entkoppeln. Soll nur ein Zertifikat gelten, so muss zweimal nacheinander dasselbe Zertifikat angegeben werden.

Request-Body

Datentyp CredentialType

```
<xsd:complexType name="CredentialType">
  <xsd:sequence>
    <xsd:element name="Type">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Password">
          </xsd:enumeration>
          <xsd:enumeration value="Certificate">
          </xsd:enumeration>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:choice>
      <xsd:element name="Credits">
        <xsd:simpleType>
          <xsd:restriction base="xsd:base64Binary">
            <xsd:minLength value="8">
            </xsd:minLength>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="Password">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:minLength value="8">
            </xsd:minLength>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
    </xsd:choice>
    <xsd:element name="Role" type="xsd:string"
      minOccurs="1" maxOccurs="1">
    </xsd:element>
    <xsd:element name="Mandant" type="xsd:string" minOccurs="0">
    </xsd:element>
    <xsd:element name="OrgID" type="xsd:string"
      minOccurs="0" maxOccurs="1">
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

Type string:
Typ des Credential.
Mögliche Werte: Password | Certificate
Für den Zugang zum Web-Service S4 ist nur der Typ Certificate erlaubt.

Credits	base64Binary: Client-Zertifikat für die Kombination Mandant/Organisation/Rolle Dieses Element ist nur für den Zugang zum Web-Service S4 vorgesehen und daher nur zusammen mit <code>Type=Certificate</code> erlaubt.
Password	string: Dieses Element ist für den Zugang zum Web-Service S4 nicht erlaubt.
Role	string: Rolle für den Zugang zum Web-Service S4. Standardmäßig ist das die Rolle <code>Archivar</code> .
OrgID	string: Für einen mit <code>createMandantXAIP</code> eingerichteten Mandanten muss die Organisation immer angegeben werden.

Der Datentyp `CredentialType` ist im Abschnitt „Operation `createMandant`“ des Handbuchs "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "[Literatur](#)") vollständig beschrieben.

3.3.4 Ausgabeinformationen für den Web-Service S4

Dieser Abschnitt enthält folgende Themen:

- Operationen `getMandants`, `getMandantProperties`, `getArchiveInfo`
- Operation `getSDOTypes`

3.3.4.1 Operationen `getMandants`, `getMandantProperties`, `getArchiveInfo`

Die Operation `getMandants` des Web-Service `ArchiveAdminService` gibt eine Liste aller Mandanten im Archiv aus.

Die Operation `getMandantProperties` des Web-Service `MandantAdminService` gibt alle Eigenschaften des Mandanten aus.

Die Operation `getArchiveInfo` der Web-Services `ArchiveAdminService` und `MandantAdminService` liefert Informationen zum aktuellen Zustand der ereignisgesteuerten Aufträge.

Eine Beschreibung dieser Operationen finden Sie im Handbuch "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "[Literatur](#)").

Der Response-Body jeder dieser Operationen enthält ein oder mehrere Elemente `Mandant` des Datentyps `MandantType`. In diesem Element finden Sie für einen Mandanten, der mit der Operation `createMandantXAIP` erzeugt wurde, die folgenden Zusatzinformationen:

- Das Ausgabeelement `Type` gibt das Format der Dokumente an, die der Mandant archivieren kann. Dieses Format legt auch den Web-Service fest, den der Mandant zum Archivieren nutzen kann.

`Type`

string:

Format der Dokumente, die der Mandant archivieren kann.

Mögliche Werte:

XAIP	Der Mandant wurde mit <code>createMandantXAIP</code> eingerichtet und kann Datenobjekte im Format XAIP mit dem Web-Service <code>S4</code> archivieren.
SDO	Der Mandant wurde mit <code>createMandant</code> eingerichtet und kann Datenobjekte in Form eines XML-Containers als Submission Data Objects (SDOs) mit dem Web-Service <code>ArchivingService</code> archivieren.

- Das Ausgabeelement `Properties` enthält neben der Liste von mandantenspezifischen Konfigurationsparametern noch zusätzliche Property-Elemente. Jedes dieser Property-Elemente enthält Namen und Wert für einen der mit `createMandantXAIP` einstellbaren Konfigurationsparameter. Beachten Sie, dass den Parameternamen in der Ausgabe, im Gegensatz zur Eingabe, das Zeichen "\$" vorangestellt ist.

Folgende Parameter können ausgegeben werden:

`supportsXAIP`

Indikator, ob der Mandant für das Archivieren mit dem Web-Service S4, also mit der Operation `createMandantXAIP`, eingerichtet wurde.

Mögliche Werte: `false` | `true`

`$SDOPath`

Ablageort des Archivdatenobjekts im Archiv. Der Pfad ist relativ zu dem Pfad, der mit der Operation `createOrganisation` festgelegt wurde.

`$SoftWorm`

Gibt an, ob das Archivdatenobjekt bis zum Ende der Aufbewahrungsfrist als SoftWORM-Datum im Langzeitspeicher abgelegt wird.

Mögliche Werte: `YES` | `NO`

`$SignatureVerification`

Legt fest, welches Ergebnis die Auswertung der Signaturen in einem Datenobjekt mindestens erreichen muss, damit eine Archivierung durchgeführt wird. Ein Datenobjekt wird nur archiviert, wenn das Ergebnis der Verifikation höher oder gleich dem Wert dieses Operanden ist.

Mögliche Werte: `SUCCESS` | `INFORMATION` | `CERTIFICATE`

`$SignatureQualityLevel`

Legt die Anforderungen an die elektronische Signatur bzw. an den Zertifizierungsdiensteanbieter fest, die für eine Archivierung mit SecDocs notwendig sind.

Mögliche Werte: `ADVANCED` | `QUALIFIED` | `ACCREDITED`

`$SignatureEmbedded`

Gibt an, ob die Datenobjekte des Mandanten eine eingebettete Signatur enthalten.

Mögliche Werte: `YES` | `NO` | `AUTO`

`$SignatureDetached`

Gibt an, ob SecDocs beim Archivieren eine abgesetzte Signatur in einem Datenobjekt erwartet.

Mögliche Werte: `YES` | `NO` | `AUTO`

Die vollständige Beschreibung des Datentyps `MandantType` finden Sie im Abschnitt „Operation createMandant“ des Handbuchs "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "[Literatur](#)").

Beispiel:

Response

```

<soap:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:sdsh="http://ts.fujitsu.com/secdocs/v3_0/secdocs"
  xsi:schemaLocation=
    "http://schemas.xmlsoap.org/soap/envelope/
    http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <sdsh:soapHeaderData>
      <sdsh:security>
        <sdsh:principal>
          <sdsh:role>MandantAdmin</sdsh:role>
          <sdsh:mandant>MandantX1</sdsh:mandant>
        </sdsh:principal>
        <sdsh:password>*****</sdsh:password>
      </sdsh:security>
      <sdsh:operation>getMandantProperties</sdsh:operation>
    </sdsh:soapHeaderData>
  </soap:Header>
  <soap:Body>
    <Mandant xmlns="http://ts.fujitsu.com/secdocs/v3_0/adminUpdateData">
      <Name>MandantX1</Name>
      <Contact>
        <FirstName>Jane</FirstName>
        <Surname>Doe</Surname>
        <Street>MyStreet 1</Street>
        <City>MyCity</City>
        <Zip/>
        <Tel>011 11111111</Tel>
        <Email>Doe@MyCompany.com</Email>
      </Contact>
      <Path>archive/MandantX1</Path>
      <TreeSize>100</TreeSize>
      <TreeAge>2</TreeAge>
      <TSP>TSP-DFN</TSP>
      <State>productive</State>
      <Type>XAIP</Type>
      <Properties>
        <Property xmlns="http://ts.fujitsu.com/secdocs/v3_0/adminData">
          <Name>$SignatureQualityLevel</Name>
          <Value>ADVANCED</Value>
        </Property>
        <Property xmlns="http://ts.fujitsu.com/secdocs/v3_0/adminData">
          <Name>$SDOPath</Name>
          <Value>*$Year*$Month*$Day*</Value>
        </Property>
        <Property xmlns="http://ts.fujitsu.com/secdocs/v3_0/adminData">
          <Name>$SignatureVerification</Name>
          <Value>SUCCESS</Value>
        </Property>
        <Property xmlns="http://ts.fujitsu.com/secdocs/v3_0/adminData">
          <Name>supportsXAIP</Name>
          <Value>true</Value>
        </Property>
      </Properties>
    </Mandant>
  </soap:Body>
</soap:Envelope>

```


3.3.4.2 Operation getSDOTypes

Die Operation `getSDOTypes` des Web-Service `MandantAdminService` gibt für einen Mandanten, der mit der Operation `createMandantXAIP` erzeugt wurde, Informationen über das für diesen Mandanten festgelegte XAIP-Format aus. Anhand der ausgegebenen Schemata können Sie erkennen, ob und wenn ja, welche Erweiterungen im XAIP-Format der Richtlinie TR-ESOR für diesen Mandanten vorgenommen wurden.

Eine Beschreibung, wie Sie die Operation `getSDOTypes` aufrufen, finden Sie im Abschnitt „Operation `getSDOTypes`“ des Handbuchs ["SecDocs Administration und Bedienung"](#) ([SD1] im Abschnitt "Literatur").

Der ausgegebene Response-Body enthält genau ein Element `SDOType` des Datentyps `SDOType`. Als Name des "SDO-Typs" ist immer `SECDOCS_XAIP_1_2` angegeben. Das Sub-Element `Filter` ist immer leer.

Die vollständige Beschreibung des Datentyps `SDOType` finden Sie im Abschnitt „Operation `createSDOType`“ des Handbuchs ["SecDocs Administration und Bedienung"](#) ([SD1] im Abschnitt "Literatur").

Beispiel:

Response-Body

```
<soap:Body>
  <GetSDOTypes xmlns="http://ts.fujitsu.com/secdocs/v3_0/adminData">
    <SDOType xmlns="http://ts.fujitsu.com/secdocs/v3_0/adminData">
      <Name>SECDOCS_XAIP_1_2</Name>
      <isActive>true</isActive>
      <Schema>PD94bWwgdmVy...0KPC9zY2h1bWE+</Schema>
      <Filter />
      <DependentSchema>
        <Namespace>urn:oasis:names:tc:dss:1.0:core:schema</Namespace>
        <SchemaLocation>./deps/oasis-dss-core-schema-v1.0-os.xsd
        </SchemaLocation>
        <Schema>PD94bWwgdmVy...M6c2NoZWlhPgo=</Schema>
      </DependentSchema>
      <DependentSchema>
        <SchemaLocation>ISOCommon.xsd</SchemaLocation>
        <Schema>PD94bWwgdmVy...o8L3NjaGVtYT4K</Schema>
      </DependentSchema>
      <DependentSchema>
        <Namespace>http://www.bsi.bund.de/ecard/api/1.1</Namespace>
        ...
      </DependentSchema>
      <DependentSchema>
        <SchemaLocation>ISOIFD.xsd</SchemaLocation>
        ...
      </DependentSchema>
      <DependentSchema>
        <Namespace>urn:oasis:names:tc:dss:1.0:core:schema</Namespace>
        ...
      </DependentSchema>
      ...
      <isReplaceable>false</isReplaceable>
    </SDOType>
  </GetSDOTypes>
</soap:Body>
```

4 Web-Service S4 für die Client-Anwendung

Der SecDocs Web-Service S4 stellt die Archivierungsfunktionen für den Beweiswerterhalt kryptographisch signierter Dokumente zur Verfügung. Folgende Operationen werden unterstützt:

- **ArchiveSubmission**
Archivierung unsignierter und signierter Daten, ggf. inklusive bereits vorhandener zugehöriger beweisrelevanter Daten und technischer Beweisdaten (engl.: Evidence Records) im Langzeitspeicher,
- **ArchiveRetrieval**
Abrufen eines Archivdatenobjekts aus dem Langzeitspeicher,
- **ArchiveEvidence**
Abrufen von technischen Beweisdaten zu einem Archivdatenobjekt zum Nachweis der Authentizität und Integrität dieses Objekts,
- **ArchiveDeletion**
Löschen eines Archivdatenobjekts im Langzeitspeicher.

Schnittstellendefinition

Unter https://www.bsi.bund.de/DE/Publikationen/TechnischeRichtlinien/tr03125/index_htm.html werden die folgenden Schnittstellen- und Schemadateien zur Verfügung gestellt:

- `tr-esor-S-4-v1.2.wsdl`
Target-Namespace: `http://www.bsi.bund.de/tr-esor/api/1.2`
Enthält u.a. die Definition der Operationen und Operanden des SecDocs Web-Service S4.
Eine SecDocs-eigene .wsdl-Datei wird nicht ausgeliefert.
- `tr-esor-interfaces-v1.2.xsd`
Target-Namespace: `http://www.bsi.bund.de/tr-esor/api/1.2`
Enthält Datentypen für den Web-Service S4.
- `tr-esor-xaip-v1_2.xsd`
Target-Namespace: `http://www.bsi.bund.de/tr-esor/xaip/1.2`
Schemadatei für das XAIP-Format
- `tr-esor-schema-standalone-v1.2.zip`
ZIP-Archiv mit weiteren Schemadateien, u. a.:
 - `oasis-dss-core-schema-v1.0-os.xsd`
Target-Namespace: `urn:oasis:names:tc:dss:1.0:core:schema`
Enthält Datentypen für den Web-Service S4.
 - `saml-schema-assertion-2.0.xsd`
Target-Namespace: `urn:oasis:names:tc:SAML:2.0:assertion`
Enthält weitere Definitionen.

Diese Dateien werden auch mit SecDocs ausgeliefert. Das ZIP-Archiv wird jedoch nicht als Datei, sondern in entpackter Form mitausgeliefert.

SecDocs-spezifische Erweiterungen der vom BSI definierten Datentypen sind in der mitausgelieferten Schemadatei `XAIPExtensions.xsd` definiert. Näheres zu diesen Erweiterungen finden Sie in den Abschnitten „[SecDocs-spezifische Erweiterungen des Formats XAIP](#)“ und „[Status- und Fehlerinformation](#)“.

Unterstützte Formate

Beim Archivieren müssen Sie das zu archivierende Datenobjekt in dem im Anhang F der Richtlinie TR-ESOR beschriebenen Format XAIP (XML formatted Archive Information Package) übergeben. Entsprechend erhalten Sie beim Abrufen eines Archivdatenobjekts dieses im XAIP-Format zurück.

Beim Abrufen von technischen Beweisdaten liefert SecDocs jeden Evidence Record in ASN.1 Syntax gemäß der Spezifikation im Standard IETF RFC 4998.

Näheres hierzu finden Sie im [Abschnitt „Format eines Archivdatenobjekts“](#).

Zugang und Zugangsprüfung

Siehe [Abschnitt „Zugang \(Endpoint-URL\) und Zugangsprüfung“](#).

Logging

Jeder Zugriff auf den Langzeitspeicher zu Zwecken der Ablage, des Abrufs der Daten oder des Abrufs von Beweisdaten oder auch des Löschens abgelegter Dokumente und Daten wird in eine Audit-Log-Datei oder in eine SecDocs-Logdatei protokolliert. Siehe hierzu Abschnitt „Logging und Fehlerbehandlung“ im Handbuch "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "Literatur").

Beachten Sie:

- Das Ändern eines bereits bestehenden Archivdatenobjekts im Langzeitspeicher (Funktion `ArchiveUpdate`) wird derzeit nicht unterstützt.
Die Angabe eines Versions-Identifikators (`versionID`) in einem `ArchiveRetrievalRequest` oder `ArchiveEvidenceRequest` oder auch im Archivdatenobjekt selbst wird von SecDocs nicht ausgewertet.
- Das Abrufen einzelner Datenelemente aus einem Archivdatenobjekt (Funktion `ArchiveData`) wird derzeit nicht unterstützt.
- Operationen des Web-Service `ArchivingService` können nicht auf Archivdatenobjekte angewendet werden, die mit dem Web-Service `S4` archiviert wurden, und umgekehrt.
- Die Möglichkeit einer Recherche über Metadaten mit einem sogenannten TripleStore und die Funktionalität der externen Datenobjekte wird im Web-Service `S4` nicht unterstützt.

4.1 Zugang (Endpoint-URL) und Zugangsprüfung

Der SecDocs Web-Service S4 ist ausschließlich unter folgender Endpoint-URL zu erreichen:

`https://secdocsHost:8444/archiver/ws/3.1/xaip/1.2`

Für den Zugang zum Web-Service S4 benötigen Sie ein mandanten- und organisationsspezifisches Zertifikat. Der Mandanten-Administrator muss das Zertifikat entsprechend eingebracht haben (siehe [Abschnitt „Zertifikate einbringen“](#)) und Handbuch "SecDocs Installationsanleitung" ([SD3] im Abschnitt "Literatur").

4.2 Format eines Archivdatenobjekts

Der SecDocs Web-Service S4 unterstützt ausschließlich das **Format XAIP** (XML formatted Archive Information Package), das im Anhang F der Richtlinie TR-ESOR beschrieben ist (siehe "[BSI TR-03125 Anlage TR-ESOR-F](#)" ([W5] im Abschnitt "Literatur")).

Die Definition dieses Formats finden Sie in der Schemadatei [tr-esor-xaip-v1_2.xsd](#) (in Abschnitt "[Web-Service S4 für die Client-Anwendung](#)").

Beim Archivieren müssen Sie daher das zu archivierende Datenobjekt im XAIP-Format übergeben; beim Abrufen des Archivdatenobjekts gibt SecDocs dieses Objekt wieder im XAIP-Format zurück.

Wollen Sie mit dem SecDocs Web-Service S4 ein Dokument im binären Format archivieren, so müssen Sie dieses Dokument als Base64-codierten Bestandteil in ein "umgebendes" XAIP-Datenobjekt einbetten.

Technische Beweisdaten (Evidence Records) werden von SecDocs in ASN.1 Syntax gemäß der Spezifikation im Standard IETF RFC 4998 in das Archivdatenobjekt eingetragen und bei Abrufen dieser Beweisdaten auch in diesem Format geliefert.

SecDocs wertet die folgenden Elemente eines Datenobjekts im XAIP-Format aus:

Element	Sub-Element	Sub-Element	Sub-Element / Attribut	siehe
packageHeader	AOID			
packageHeader	CanonicalizationMethod			1)
packageHeader	versionManifest	preservationInfo	retentionPeriod	
dataObjectsSection	dataObject	binaryData		2)
dataObjectsSection	dataObject	binaryData	MimeType	3)
credentialsSection	credential	SignatureObject	Base64Signature	4)
credentialsSection	credential	evidenceRecord	asn1EvidenceRecord	5)
credentialsSection	credential	other		6)

Anmerkungen

1. Das Element `CanonicalizationMethod` spezifiziert die anzuwendende Kanonisierungsmethode. SecDocs unterstützt die folgenden Werte:

- `http://www.w3.org/TR/2001/REC-xml-c14n-20010315`
- `http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments`
- `http://www.w3.org/2001/10/xml-exc-c14n#`
- `http://www.w3.org/2001/10/xml-exc-c14n#WithComments`
- `http://www.w3.org/2006/12/xml-c14n11`
- `http://www.w3.org/2006/12/xml-c14n11#WithComments`

Der Standardwert ist <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>.

2. `dataObject` (Datentyp `dataObjectType`) enthält die Nutzdaten des Archivdatenobjekts. SecDocs unterstützt nur das Sub-Element `binaryData`, d.h. die Codierung der Nutzdaten als Binärdaten. Die Nutzdaten müssen gemäß dem IETF Standard RFC4648 Base64-codiert sein. Den Typ der Nutzdaten können Sie im Attribut `MimeType` angeben.
3. Das Attribut `MimeType` (Datentyp `string`) enthält den Typ der im Element `binaryData` angegebenen Nutzdaten. Es wird empfohlen, einen Typ gemäß der von der IANA (Internet Assigned Numbers Authority) gepflegten Liste der registrierten Media-Types anzugeben (siehe <http://www.iana.org/assignments/media-types/media-types.xhtml>).

Beachten Sie:

In ein Datenobjekt eingebettete Signaturen können nur dann durch SecDocs geprüft werden, wenn SecDocs das Datenobjekt als PDF-Dokument interpretiert. Dies ist nur dann der Fall, wenn Sie beim Attribut `MimeType` einen der folgenden Media-Types angeben:

- `application/pdf`
- `application/vnd.pdf`
- `application/vnd.cups-pdf`
- `application/x-pdf`

(Näheres zu den Randbedingungen bei der Signaturprüfung finden Sie im [Abschnitt „Operation ArchiveSubmission“](#) .

4. Im Element `SignatureObject` können Sie bei Bedarf abgesetzte Signaturen für Nutz- oder Metadatenobjekte hinterlegen. SecDocs unterstützt jedoch nur das Sub-Element `Base64Signature`, d.h. nur an dieser Stelle können Sie eine abgesetzte Signatur übergeben.
5. Im Element `evidenceRecord` können Sie bei Bedarf Beweisdaten in Form von Evidence Records hinterlegen. SecDocs unterstützt jedoch nur das Sub-Element `asn1EvidenceRecord`, d.h. Sie können nur Evidence Records übergeben, deren Format dem Standard IETF RFC 4998 entspricht.
6. Im Element `other` legt SecDocs bei Bedarf Signaturverifikationsdaten sowie Beweisdaten (Evidence Records) ab. Näheres hierzu siehe [Abschnitt „SecDocs-spezifische Erweiterungen des Formats XAIP“](#). SecDocs wertet jedoch bei der Archivierung das Element `other` nicht aus.

Alle anderen Elemente des XAIP-Formats werden von SecDocs nicht ausgewertet.

Insbesondere gilt dies für:

- Versionsangaben im Element `packageHeader/versionManifest`, genauer: in `versionInfo` (Element vom Datentyp `string`; enthält Information zur Version des Archivdatenobjekts in Textformat) und `versionID` (Attribut vom Datentyp `ID`; enthält einen eindeutigen Identifikator der Version des Archivdatenobjekts),
- die durch die Menge der Elemente `packageHeader/versionManifest/packageInfoUnit/protectedObjectPointer` angegebenen Teile des Datenobjekts (diese Teile werden nur bei der Versiegelung verwendet, wo sie in die Hashwertbildung einfließen),
- Einträge im Element `credentialsSection/credential/other`.

Beachten Sie:

Die Archivierung eines Datenobjekts im XAIP-Format wird in folgenden Fällen abgewiesen:

-
- Das Element `credentialsSection/credential` enthält weder das Sub-Element `SignatureObject` noch das Sub-Element `evidenceRecord`
 - Das Element `credentialsSection/credential/SignatureObject` enthält kein Sub-Element `Base64Signature`
 - Das Element `credentialsSection/credential/evidenceRecord` enthält kein Sub-Element `asn1EvidenceRecord`
 - Eines der Elemente `credentialsSection/credential/SignatureObject/Base64Signature` oder `credentialsSection/credential/evidenceRecord/asn1EvidenceRecord` bezieht sich nicht auf ein Element `dataObject`

4.2.1 SecDocs-spezifische Erweiterungen des Formats XAIP

SecDocs-spezifische Erweiterungen der vom BSI definierten Datentypen sind in der mitausgelieferten Schemadatei `XAIPExtensions.xsd` definiert.

Der Target-Namespace ist `http://ts.fujitsu.com/secdocs/v3_1/xaiparchiving`.

Die SecDocs-spezifischen Erweiterungen des Formats XAIP sind in der obengenannten Schemadatei im Abschnitt „credentialType - SecDocs Specific Definitions“ definiert.

Das folgende Element wird erweitert:

Element `credentialsSection/credential` (*Datentyp* `credentialType`)

Die Signaturverifikationsdaten sowie die technischen Beweisdaten (Evidence Records) werden von SecDocs nicht in den Elementen `VerificationReport` bzw. `evidenceRecord`, sondern im Element `other` abgespeichert. Zusätzlich wird von SecDocs der Name des Zeitstempelanbieters (Time Stamp Provider, TSP) hinterlegt.

SecDocs generiert für `credentialsSection/credential/other` die Sub-Elemente

- `vrInfo` für die Signaturverifikationsinformation

```
<element name="vrInfo" type="base64Binary">
  <annotation>
    <documentation>SecDocs signature verification information
                      provided by OpenLimit SignCubes GmbH
    </documentation>
  </annotation>
</element>
```

Derzeit wird nur **eine** Signaturverifikationsinformation für das gesamte XAIP-Datenobjekt erzeugt. Diese Signaturverifikationsinformation enthält für jede Signatur im XAIP-Datenobjekt alle Verifikationsprotokolle.

- `evidenceRecord` vom Datentyp `TEvidenceRecord` für einen Evidence Record

```
<element name="evidenceRecord" type="tr:TEvidenceRecord">
  <annotation>
    <documentation>SecDocs created evidence records
    </documentation>
  </annotation>
</element>
<complexType name="TEvidenceRecord">
  <simpleContent>
    <extension base="base64Binary">
      <attribute name="tsp" use="required" type="string"/>
    </extension>
  </simpleContent>
</complexType>
```

Beispiel:

```
...
<xaip:credentialsSection>
  <xaip:credential credentialID="..." relatedObjects="...">
    <xaip:other>
      <sd:vrInfo
        xmlns:sd="http://ts.fujitsu.com/secdocs/v3_0/xaiparchiving">
          PD94bWwgdmVyc2lvcj0iMS4...ZmljYXRpb25JbmZvMT4K
        </sd:vrInfo>
      </xaip:other>
    </xaip:credential>
    <xaip:credential credentialID="..." relatedObjects="...">
      <xaip:other>
        <sd:evidenceRecord tsp="TSP-DFN"
          xmlns:sd="http://ts.fujitsu.com/secdocs/v3_0/xaiparchiving">
            MIIIGgIBATALMAkGBSsOAw...FwFKA4eMiYDuPltkiWsOZ4=
          </sd:evidenceRecord>
        </xaip:other>
      </xaip:credential>
    </xaip:credentialsSection>
  ...
```

4.3 Aufbau der SOAP-Nachricht beim SecDocs Web-Service S4

Eine Client-Anwendung kommuniziert mit einem Web-Service über SOAP-Nachrichten (SOAP-Request und SOAP-Response) nach SOAP 1.1. Jeder SOAP-Request und jede SOAP-Response ist eine XML-Nachricht mit einem SOAP-Envelope als Wurzelement.

Beim SecDocs Web-Service S4 enthält der SOAP-Envelope keinen SOAP-Header, sondern als Sub-Element ausschließlich einen SOAP-Body. Die SOAP-Nachrichten enthalten keine Attachments. Als Nachrichtenformat wird "Document literal" verwendet.

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    . . .
  </s:Body>
</s:Envelope>
```

Der SecDocs Web-Service S4 verwendet außer in den nachfolgend genannten Ausnahmefällen den HTTP Server Response Code 200 (erfolgreiche Bearbeitung). Die Client-Anwendung erhält als Antwort auf einen SOAP-Request also immer eine operationsspezifische SOAP-Response. SecDocs gibt im Fehlerfall keine SOAP-Fault-Message zurück und erzeugt keine Exception.

Die SOAP-Response enthält immer eine Status-Information, außerdem wird im Fehlerfall noch zusätzliche Fehlerinformation zurückgegeben. Näheres finden Sie im [Abschnitt „Status- und Fehlerinformation“](#).

Ausnahmefälle

Die Client-Anwendung muss in folgenden Fällen mit einem HTTP Server Response Fehlercode als Reaktion rechnen:

- die Zugangsprüfung fällt negativ aus (z.B. ungültiges Zertifikat),
- der SOAP-Request enthält einen vom Web-Service S4 nicht unterstützten Operationsnamen.

Der Aufbau von SOAP-Request und -Response ist operationsspezifisch und ist in den Abschnitten [„Request“](#) und [„Response“](#) sowie bei den einzelnen Operationen beschrieben.

4.3.1 Request

Bei jeder Operation des SecDocs Web-Service S4 ist der SOAP-Request eine Erweiterung des in [tr-esor-interfaces-v1.2.xsd](#) (in "Web-Service S4 für die Client-Anwendung") und [oasis-dss-core-schema-v1.0-os.xsd](#) (in "Web-Service S4 für die Client-Anwendung") definierten allgemeinen Datentyps RequestType.

Datentyp RequestType

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

OptionalInputs

Datentyp AnyType

optional; Dieses Element ist für optionale Eingabeelemente vorgesehen, mit denen Sie zusätzliche Informationen übergeben können.

Diese Zusatzinformationen sind bei den einzelnen Operationen näher beschrieben.

Datentyp AnyType

```
<xs:complexType name="AnyType">
  <xs:sequence>
    <xs:any processContents="lax" minOccurs="0"
            maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

4.3.2 Response

Bei jeder Operation des SecDocs Web-Service S4 ist die SOAP-Response eine Erweiterung des in [tr-esor-interfaces-v1.2.xsd](#) (in "Web-Service S4 für die Client-Anwendung") und [oasis-dss-core-schema-v1.0-os.xsd](#) (in "Web-Service S4 für die Client-Anwendung") definierten allgemeinen Datentyps ResponseType.

Datentyp ResponseType

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

Result

Dieses Element enthält eine Statusinformation über das Ergebnis der Operation.

OptionalOutputs

Datentyp AnyType, siehe [Abschnitt „Request“](#).

optional; Dieses Element ist für optionale Ausgabeelemente vorgesehen, mit denen SecDocs zusätzliche Informationen zurückgibt.

Diese Zusatzinformationen sind im [Abschnitt „Status- und Fehlerinformation“](#) näher beschrieben.

Element Result

```
<xs:element name="Result">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ResultMajor" type="xs:anyURI" />
      <xs:element name="ResultMinor" type="xs:anyURI"
        minOccurs="0" />
      <xs:element name="ResultMessage"
        type="dss:InternationalStringType"
        minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

ResultMajor

anyURI:

Indikator, ob die ausgeführte Operation erfolgreich war.

Mögliche Werte:

SUCCESS Die Operation wurde erfolgreich durchgeführt

FAILURE Die Operation war nicht erfolgreich

ResultMessage

InternationalStringType:

Gibt das Ergebnis der ausgeführten Operation zurück.

Mögliche Werte:

siehe Datentyp TResultMessage.

Datentyp TResultMessage

```
<simpleType name="TResultMessage">
  <restriction base="string">
    <enumeration value="XAIP document submitted successfully" />
    <enumeration value="XAIP document already submitted" />
    <enumeration value="XAIP document already submitted and sealed" />
    <enumeration value="XAIP document retrieved successfully" />
    <enumeration value="evidence records retrieved successfully"/>
    <enumeration value="no evidence records exist" />
    <enumeration value="all versions of XAIP document deleted
      successfully" />
    <enumeration value="submission failed" />
    <enumeration value="processing not successful" />
  </restriction>
</simpleType>
```

Das Element ResultMinor wird von SecDocs nicht ausgegeben.

4.3.3 Status- und Fehlerinformation

Der SecDocs Web-Service *S4* verwendet – außer in den nachfolgend genannten Ausnahmefällen – den HTTP Server Response Code 200 (erfolgreiche Bearbeitung). Die Client-Anwendung erhält als Antwort auf einen SOAP-Request also immer eine operationsspezifische SOAP-Response. SecDocs gibt im Fehlerfall keine SOAP-Fault-Message zurück.

Ausnahmefälle

Die Client-Anwendung muss in folgenden Fällen mit einem HTTP Server Response Fehlercode als Reaktion rechnen:

- die Zugangsprüfung fällt negativ aus (z.B. ungültiges Zertifikat),
- der SOAP-Request enthält einen vom Web-Service *S4* nicht unterstützten Operationsnamen.

Die SOAP-Response enthält immer eine Status-Information. Im Fehlerfall wird zusätzlich Fehlerinformation zurückgegeben.

Für diese Status- und Fehlerinformationen sind in SecDocs eigene Datentypen definiert, um die Möglichkeit einer detaillierten Rückgabeinformation zu bieten.

SecDocs gibt die Status- und Fehlerinformation im Element `OptionalOutputs` der SOAP-Response zurück (siehe [Abschnitt „Response“](#)). Die Definitionen für diese Informationen sind in der mitausgelieferten Schemadatei `XAIPExtensions.xsd` im Abschnitt "OptionalOutputs - SecDocs specific Definitions" abgelegt.

(Target-Namespace dieser Schemadatei:

`http://ts.fujitsu.com/secdocs/v3_0/xaiparchiving`).

Statusinformation

Element status vom Datentyp `ResponseStatus`

```
<complexType name="ResponseStatus">
  <sequence>
    <element name="statusCode" type="tr:TStatusCode"
      minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

`statusCode`

Datentyp `TStatusCode`:

Gibt das Ergebnis der ausgeführten Operation zurück.

Datentyp

TStatusCode

```
<simpleType name="TStatusCode">
  <restriction base="string">
    <enumeration value="Success: XAIP document submitted successfully" />
    <enumeration value="Success: XAIP document already submitted" />
    <enumeration value=
      "Success: XAIP document already submitted and sealed" />
    <enumeration value="Success: XAIP document retrieved successfully" />
    <enumeration value=
      "Success: evidence records retrieved successfully"/>
    <enumeration value="Success: no evidence records exist" />
    <enumeration value=
      "Success: all versions of XAIP document deleted successfully" />
    <enumeration value="Failure: submission failed" />
    <enumeration value="Failure: processing not successful" />
  </restriction>
</simpleType>
```

Success: ...

Die Operation wurde erfolgreich bearbeitet.

Failure: ...

Bei Bearbeitung der Operation trat ein Fehler auf.

Genauere Information finden Sie im Element `faultDetails` (siehe „[Fehlerinformation](#)“).

Fehlerinformation

Element `faultDetails` vom Datentyp `TFaultDetails`

```
<complexType name="TFaultDetails">
  <sequence>
    <element name="nodeName" type="tr:TNonEmptyString"
      minOccurs="0" maxOccurs="1"/>
    <element name="requestNumber" type="long"
      minOccurs="1" maxOccurs="1"/>
    <element name="operation" type="string"
      minOccurs="0" maxOccurs="1"/>
    <element name="errorMessage" type="string"
      minOccurs="1" maxOccurs="1"/>
    <element name="errorCode" type="integer"
      minOccurs="1" maxOccurs="1"/>
    <element name="errorDetail" type="string"
      minOccurs="0" maxOccurs="1"/>
    <element name="migSafeFaultDetails" type="tr:TMigSafeFaultDetails"
      minOccurs="0" maxOccurs="1"/>
  </sequence>
</complexType>
```

nodeName

TNonEmptyString:

optional; Name des Knotens, auf dem der Fehler aufgetreten ist. Dieses Element wird nur ausgegeben, wenn Sie den Konfigurationsparameter

nodeNameInFaultMessage in der Datei `secdocs.properties` auf "true" gesetzt haben (siehe hierzu Abschnitt „Konfigurationsdatei `secdocs.properties`“ im Handbuch "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "[Literatur](#)").

requestNumber

long:

Nummer des Request.

SecDocs ordnet jedem Request aufsteigend eine Nummer zu, die im Fehlerfall auf der Server-Seite protokolliert wird. Mit Hilfe dieser Nummer kann die Verbindung zwischen Fehlermeldung auf der Client-Seite und Fehlermeldung auf der Server-Seite hergestellt werden.

operation

string:

optional; Name der aufgerufenen Operation.

Ist ein Fehler aufgetreten, bevor die Operation ermittelt werden konnte, ist dieses Element leer.

errorMessage

string:

Text der Fehlermeldung.

Eine Liste der Fehlermeldungen finden Sie im Handbuch "[SecDocs-Rückgabewerte](#)" ([SD4] im Abschnitt "[Literatur](#)"). Bezieht sich `errorMessage` auf Rückgabewerte der ArchiSig- oder Krypto-Schnittstelle, so finden Sie dafür detaillierte Informationen im Handbuch "[MigSafe / OverSign - Rückgabewerte](#)" ([SD6 im Abschnitt "[Literatur](#)") .

errorCode

integer:

Error-Code der Fehlermeldung

errorDetail

string:

optional; Zusatzinformation zur Fehlermeldung

migSafeFaultDetails

Datentyp `TMigSafeFaultDetails`:

optional; Information zum Grund einer Ablehnung der Operation `ArchiveSubmission`.

Dieses Element wird nur ausgegeben, wenn die Verifikation einer oder mehrerer Signaturen fehlgeschlagen ist. In diesem Fall erhalten Sie für jeden Signatur-Container, der mindestens eine negativ geprüfte Signatur enthält, genau ein Element `migSafeFaultDetail`.

Datentyp

TMigSafeFaultDetails

```
<complexType name="TMigSafeFaultDetails">
  <sequence>
    <element name="migSafeFaultDetail" type="tr:TMigSafeFaultDetail"
      minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

migSafeFaultDetail

Datentyp TMigSafeFaultDetail

Datentyp TMigSafeFaultDetail

```
<complexType name="TMigSafeFaultDetail">
  <sequence>
    <!-- MigSafe error info string -->
    <element name="info" type="string"
      minOccurs="1" maxOccurs="1"/>
    <!-- MigSafe ECODE -->
    <element name="ecode" type="string"
      minOccurs="1" maxOccurs="1"/>
    <!-- OpenLimit V3 verification protocol -->
    <element name="xmlVerificationProtocol" type="base64Binary"
      minOccurs="0" maxOccurs="1"/>
    <!-- HTML version of the OpenLimit V3 verification protocol -->
    <element name="htmlVerificationProtocol" type="base64Binary"
      minOccurs="0" maxOccurs="1"/>
  </sequence>
</complexType>
```

info

string:
Text der Fehlermeldung.

ecode

string:
Error-Code der Signaturprüfung.

xmlVerificationProtocol

base64Binary:
optional; Verifikationsprotokoll des CryptoModule mit allen Detail-Angaben für die Signaturprüfung.

htmlVerificationProtocol

base64Binary:

optional; als XHTML-Datei aufbereitetes Verifikationsprotokoll.

Die Generierung der XHTML-Datei ist standardmäßig ausgeschaltet. Wenn Sie diese Protokolle erzeugen wollen, müssen Sie den Konfigurationsparameter

doHTMLProtocolsOnRetrieval in der Datei secdocs.properties auf "true" setzen (siehe hierzu Abschnitt „Konfigurationsdatei secdocs.properties“ im Handbuch "[SecDocs Administration und Bedienung](#)" ([SD1] im Abschnitt "Literatur")).

Beispiel:

Grün: SecDocs-Fehlermeldung

Blau: Fehlermeldung der ArchiSig-/Krypto-Komponenten

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header></soapenv:Header>
  <soapenv:Body>
    <ns2:ArchiveSubmissionResponse
      xmlns:ns10="http://ts.fujitsu.com/secdocs/v3_0/xaiparchiving"
      xmlns:ns9="urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:schema#"
      xmlns:ns8="http://uri.etsi.org/01903/v1.3.2#"
      xmlns:ns7="http://www.bsi.bund.de/tr-esor/xaip/1.2"
      xmlns:ns6="http://www.setcce.org/schemas/ers"
      xmlns:ns5="urn:iso:std:iso-iec:24727:tech:schema"
      xmlns:ns4="http://www.bsi.bund.de/ecard/api/1.1"
      xmlns:ns3="http://www.w3.org/2000/09/xmldsig#"
      xmlns:ns2="http://www.bsi.bund.de/tr-esor/api/1.2"
      xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
      Profile="http://ts.fujitsu.com/secdocs/SecDocs 3.0A SOAP Service">
      <Result>
        <ResultMajor>FAILURE</ResultMajor>
        <ResultMessage xml:lang="en">submission failed</ResultMessage>
      </Result>
      <OptionalOutputs>
        <ns10:status>
          <ns10:statusCode>Failure: submission failed</ns10:statusCode>
        </ns10:status>
        <ns10:faultDetails>
          <ns10:nodeName>MyNode2</ns10:nodeName>
          <ns10:requestNumber>2</ns10:requestNumber>
          <ns10:operation>ArchiveSubmission</ns10:operation>
          <ns10:errorMessage>XRQ0004 : Error in OpenLimit function migSafe.
ol_submitSDO:
ERROR_VERIFICATION_WRONG: Verification failed! (ECODE: c021c018)OpenLimit detail
info: Sig-
Path:/XAIP/credentialSection/credential/ [@credentialID='Signature_01']/text()
```

MSG:ERROR_SUBMIT_SDO_INVALID_INPUT_DATA: Invalid input data! (ECODE: c021c056)

```
</ns10:errorMessage>
<ns10:errorCode>1204</ns10:errorCode>
<ns10:migSafeFaultDetails>
  <ns10:migSafeFaultDetail>
    <ns10:info>
      error information of the ArchiSig- or Crypto-Modules
    </ns10:info>
    <ns10:ecode> error code of the ArchiSig- or Crypto-Modules
    </ns10:ecode>
  </ns10:migSafeFaultDetail>
</ns10:migSafeFaultDetails>
</ns10:faultDetails>
</OptionalOutputs>
</ns2:ArchiveSubmissionResponse>
</soapenv:Body>
</soapenv:Envelope>
```

4.4 Operationen des Web-Service S4

Dieser Abschnitt enthält folgende Themen:

- Operation ArchiveSubmission
- Operation ArchiveRetrieval
- Operation ArchiveEvidence
- Operation ArchiveDeletion
- Änderung der Aufbewahrungszeit

4.4.1 Operation ArchiveSubmission

ArchiveSubmission archiviert ein Datenobjekt konform zur technischen Richtlinie TR-ESOR unter einer archivweit eindeutigen Identifikation, der AOID.

Als Übergabeobjekt wird ein Datenobjekt im Format XAIP (XML formatted Archive Information Package) erwartet. Dieses Format ist im Anhang F der Richtlinie TR-ESOR beschrieben (siehe [BSI TR-03125 Anlage TR-ESOR-F \(\[W5\] im Abschnitt "Literatur"\)](#)).

Die Client-Anwendung kann eine eigene AOID vergeben, andernfalls wird die AOID von SecDocs generiert.

SecDocs führt im Rahmen einer ArchiveSubmission-Operation folgende Aktionen durch:

- Das Archivdatenobjekt wird auf syntaktische Richtigkeit geprüft.
Es erfolgt immer eine Syntaxprüfung gegen das in der Datei `tr-esor-xaip-v1_2.xsd` (in "Web-Service S4 für die Client-Anwendung") bereitgestellte Schema. Diese Datei wird vom BSI unter https://www.bsi.bund.de/DE/Publikationen/TechnischeRichtlinien/tr03125/index_hm.html zur Verfügung gestellt und auch mit SecDocs ausgeliefert.
Wenn für den Mandanten benutzerspezifische Erweiterungen des XAIP-Schemas registriert wurden, erfolgt zusätzlich eine Syntaxprüfung gegen diese Schemata. Benutzerspezifische Erweiterungen im Archivdatenobjekt, die nicht mit `modifyXAIP` registriert worden sind, werden ohne Syntaxprüfung akzeptiert.
- Enthält das Archivdatenobjekt Signaturen oder Beweisdaten (Evidence Records), werden diese beweisrelevanten Daten bzw. Beweisdaten auf ihre Gültigkeit geprüft (Randbedingungen bei eingebetteten Signaturen siehe unten).
Schlägt die Verifikation der beweisrelevanten Daten bzw. Beweisdaten fehl, wird die Operation ArchiveSubmission abgewiesen. Das übergebene Datenobjekt wird nicht archiviert.

Beachten Sie:

Enthält das Datenobjekt eingebettete Signaturen, so prüft SecDocs diese Signaturen nur dann, wenn die folgenden Voraussetzungen erfüllt sind:

- SecDocs interpretiert das Datenobjekt als PDF-Dokument; dies ist nur dann der Fall, wenn Sie beim Attribut `MimeType` des Elements `dataObjectsSection/dataObject/binaryData` einen der folgenden Media-Types angeben (siehe Anmerkung 3 (in "Format eines Archivdatenobjekts")):
 - `application/pdf`
 - `application/vnd.pdf`
 - `application/vnd.cups-pdf`
 - `application/x-pdf`

und

- Für den Mandanten ist der Wert `SignatureEmbedded` auf "YES" oder "AUTO" eingestellt (siehe "Operation [createMandantXAIP](#)").

- Die bei der Gültigkeitsprüfung von beweisrelevanten Daten bzw. Beweisdaten ermittelten Prüfergebnisse (Zertifikate, Sperrlisten, OCSP-Responses) werden in standardisierter Form in das Archivdatenobjekt eingetragen.
SecDocs erweitert dazu das Element `credentialsSection/credential` um das Element `other` mit dem Sub-Element `vrInfo` (siehe hierzu auch [Abschnitt „SecDocs-spezifische Erweiterungen des Formats XAIP“](#)).
Derzeit wird nur **eine** Signaturverifikationsinformation für das gesamte XAIP-Datenobjekt erzeugt. Diese Signaturverifikationsinformation enthält für jede Signatur im XAIP-Datenobjekt alle Verifikationsprotokolle.
- Enthält das übergebene Archivdatenobjekt keine von der Client-Anwendung vergebene AOID, erzeugt SecDocs eine eigene AOID und fügt diese ins Archivdatenobjekt ein (Element `packageHeader/AOID`).
- Werden im XAIP-Datenobjekt Namespaces verwendet, die zwar im `ArchiveSubmission-Request` deklariert sind, jedoch nicht im XAIP-Datenobjekt selbst, so fügt SecDocs die Deklarationen dieser Namespaces in das XAIP-Datenobjekt ein, (genauer: in das Element `XAIP`, siehe unten).
- SecDocs legt das übergebene Archivdatenobjekt inklusive der evtl. eingefügten Daten (Prüfergebnisse, AOID) im Langzeitspeicher ab.
SecDocs führt keine XML-Kanonisierung des Archivdatenobjekts durch.

Nach der Versiegelung des Archivdatenobjekts werden die erzeugten Evidence Records in standardisierter Form in das Archivdatenobjekt eingetragen.

SecDocs erweitert dazu das Element `credentialsSection/credential` um das Element `other` mit dem Sub-Element `evidenceRecord` (siehe hierzu auch [Abschnitt „SecDocs-spezifische Erweiterungen des Formats XAIP“](#)).

Voraussetzungen

- Das zu archivierende Datenobjekt liegt im Format XAIP vor. Dieses Format ist im Anhang F der Richtlinie TR-ESOR beschrieben (siehe [BSI TR-03125 Anlage TR-ESOR-F \(\[W5\] im Abschnitt "Literatur"\)](#)).
- Wenn für den Mandanten benutzerspezifische Erweiterungen des XAIP-Schemas definiert sind und diese Erweiterungen bei der Archivierung syntaktisch geprüft werden sollen, müssen die dazugehörigen Schemata bei SecDocs registriert sein (siehe hierzu [Abschnitt „Operation modifyXAIP“](#)).

Beachten Sie:

- Die Operation `ArchiveSubmission` wird abgewiesen, wenn die Verifikation einer oder mehrerer Signaturen fehlschlägt. In diesem Fall wird im Element `faultDetails` zusätzlich zu den SecDocs-spezifischen Fehlerinformationen die Fehlerinformation der ArchiSig-/Krypto-Komponenten ausgegeben (Element `misSafeFaultDetails`), die u.a. die Verifikationsprotokolle aller geprüften Signaturen enthält. Es wird empfohlen, diese Verifikationsprotokolle für eine nachfolgende Analyse als Dateien abzuspeichern.
- SecDocs wertet nicht alle Elemente des übergebenen XAIP-Datenobjekts aus (siehe ["Format eines Archivdatenobjekts"](#))
Insbesondere werden Versionsangaben im XAIP-Datenobjekt von SecDocs nicht unterstützt: vorhandene Versionsangaben im Element `packageHeader/versionManifest` (Element `versionInfo` und Attribut `versionID`) werden bei der Archivierung nicht ausgewertet.
- Als `retentionPeriod` ist nur ein Datum nach dem 1.1.1970 00:01 Uhr erlaubt.
SecDocs generiert bei der Archivierung keine Versionsnummer.

- Die Anzahl der Archivdatenobjekte, die im Archiv unter einem Knoten archiviert werden können, ist durch die Anzahl der möglichen Unterverzeichnisse unter einem Knoten limitiert. Diese Grenze ist abhängig vom verwendeten Storage- bzw. Dateisystem. Wählen Sie als Unterstruktur z.B. /\$Year/\$Month/\$Day, so kann jeden Tag diese Zahl an Dokumenten gespeichert werden.
Unterstrukturen definieren Sie mit dem Operanden `SDOPath` bei der Operation `createMandantXAIP` (siehe [Abschnitt „Operation createMandantXAIP“](#)).
- Die maximale Größe eines SOAP-Requests, die SecDocs verarbeiten kann, können Sie mit dem Parameter `maxSoapRequestSize` in der Konfigurationsdatei `secdocs.properties` einstellen. Die Beschreibung hierzu finden Sie im Abschnitt „Konfigurationsdatei `secdocs.properties`“ im Handbuch „[SecDocs Administration und Bedienung \(\[SD1\] im Abschnitt "Literatur"\)](#)“.

Request

Element ArchiveSubmissionRequest

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

Die Beschreibung des Datentyps `RequestType` finden Sie im [Abschnitt „Request“](#).

ArchiveData Datentyp `ArchiveDataRequest`:

SecDocs wertet dieses Element nicht aus.

Sie müssen immer das unten beschriebene Element `XAIP` angeben.

XAIP Datentyp `XAIPTyp`:

Zu archivierendes Datenobjekt im XAIP-Format.

Das Format XAIP ist in der Schemadatei `tr-esor-xaip-v1_2.xsd` (in "Web-Service S4 für die [Client-Anwendung](#)") definiert und im Anhang F der Richtlinie TR-ESOR beschrieben (siehe "BSI [TR-03125 Anlage TR-ESOR-F](#)" ([W5] im Abschnitt "Literatur")). Die Schemadatei wird vom BSI unter "BSI Technische Richtlinie 03125" (siehe [W1] im Abschnitt "Literatur") zur Verfügung gestellt und auch mit SecDocs ausgeliefert.

Der Target-Namespace dieser Datei ist

`http://www.bsi.bund.de/tr-esor/xaip/1.2.`

Hinweise zu einzelnen Elementen:

SecDocs wertet nur bestimmte Elemente des übergebenen XAIP-Datenobjekts aus (siehe "[Format eines Archivdatenobjekts](#)")

CanonicalizationMethod

Element vom Datentyp `CanonicalizationMethodType` in der Struktur `packageHeader`, das die anzuwendende Kanonisierungsmethode angibt. SecDocs unterstützt die folgenden Werte:

- <http://www.w3.org/TR/2001/REC-xml-c14n-20010315> (Defaultwert)
- <http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments>
- <http://www.w3.org/2001/10/xml-exc-c14n#>
- <http://www.w3.org/2001/10/xml-exc-c14n#WithComments>
- <http://www.w3.org/2006/12/xml-c14n11>
- <http://www.w3.org/2006/12/xml-c14n11#WithComments>

Der Standardwert ist

<http://www.w3.org/TR/2001/REC-xml-c14n-20010315>.

Falls Sie die Kanonisierungsmethode nicht angegeben haben, trägt SecDocs diesen Standardwert in das Element `packageHeader/CanonicalizationMethod` ein.

dataObject

Element vom Datentyp `dataObjectType` in der Struktur `dataObjectsSectionType`, das die Nutzdaten des Archivdatenobjekts enthält.

Hier müssen Sie das Element `binaryData` angeben, d.h. Sie müssen die Nutzdaten in dem zu archivierenden Datenobjekt als Binärdaten übergeben. Die Nutzdaten müssen gemäß dem IETF Standard RFC4648 Base64-kodiert sein.

MimeType

Attribut des Elements `binaryData` (Datentyp `string`), das den Typ der in `binaryData` angegebenen Nutzdaten enthält. Es wird empfohlen, einen Typ gemäß der von der IANA (Internet Assigned Numbers Authority) gepflegten Liste der registrierten Media-Types anzugeben (siehe <http://www.iana.org/assignments/media-types/media-types.xhtml>).

Beachten Sie, dass SecDocs das Datenobjekt nur dann als PDF-Dokument interpretiert und somit eine Verifikation evtl. eingebetteter Signaturen vornimmt, wenn Sie bei `MimeType` einen der folgenden Media-Types angeben:

- `application/pdf`
- `application/vnd.pdf`
- `application/vnd.cups-pdf`
- `application/x-pdf`

SignatureObject

Bei Bedarf können Sie hier abgesetzte Signaturen für Nutz- oder Metadatenobjekte hinterlegen. Sie müssen dazu das Element `Base64Signature` angeben.

evidenceRecord

Bei Bedarf können Sie hier Beweisdaten in Form von Evidence Records hinterlegen. Sie müssen dazu das Element `asn1EvidenceRecord` angeben, d.h. Sie können nur Evidence Records übergeben, deren Format dem Standard IETF RFC 4998 entspricht.

Vergabe einer client-spezifischen Identifikation des Archivdatenobjekts (AOID)

Wenn Sie für ein zu archivierendes Datenobjekt eine eigene AOID vergeben wollen, müssen Sie im Element `packageHeader` des XAIP-Datenobjekts das Element `AOID` angeben.

```
<xs:element name="AOID" type="xs:string"
            maxOccurs="1" minOccurs="0"></xs:element>
```

Diese AOID kann ein beliebiger nichtleerer String sein. Sie darf jedoch weder Steuerzeichen noch die Zeichen "?" oder "*" enthalten und ist auf maximal 256 Zeichen beschränkt. Die Angabe ist case-sensitiv, d.h. sie wird von SecDocs genau so verwendet, wie sie definiert ist.

Response

Element `ArchiveSubmissionResponse`

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

Die Beschreibung des Datentyps `ResponseType` finden Sie im [Abschnitt „Response“](#).

AOID

string:

Enthält im Fall einer erfolgreichen Speicherung die archivweit eindeutige Identifikation des Archivdatenobjekts.

Wenn Sie im XAIP-Datenobjekt des `ArchiveSubmissionRequest` eine eigene AOID angegeben haben, so wird genau diese AOID im Element `AOID` zurückgeliefert.

Haben Sie keine eigene AOID angegeben, wird hier die von SecDocs vergebene AOID (Universally Unique Identifier gemäß Standard IETF RFC 4122) zurückgegeben. Diese von SecDocs vergebene AOID wird zudem auch in das Archivdatenobjekt (Element `AOID` im Element `packageHeader`) eingefügt.

Ausgaben im Element Result:

ResultMajor

Im Erfolgsfall:

SUCCESS

Im Fehlerfall:

FAILURE

ResultMessage

Im Erfolgsfall sind folgende Werte möglich:

```
XAIP document submitted successfully
XAIP document already submitted
XAIP document already submitted and sealed
```

Im Fehlerfall:

submission failed

Ausgaben im Element OptionalOutputs:

statusCode im Element status

Im Erfolgsfall sind folgende Werte möglich:

```
Success: XAIP document submitted successfully
Success: XAIP document already submitted
Success: XAIP document already submitted and sealed
```

Im Fehlerfall:

Failure: submission failed

faultDetails

Datentyp `TFaultDetails`, siehe [Abschnitt „Status- und Fehlerinformation“](#).

Dieses Element wird nur im Fehlerfall ausgegeben.

Beispiel

SOAP-Body des ArchiveSubmissionRequest

```
<S:Body>
  <ns3:ArchiveSubmissionRequest xmlns:ns8="http://www.bsi.bund.de/tr-esor/xaip/1.2"
                                xmlns:ns3="http://www.bsi.bund.de/tr-esor/api/1.2" >
    <ns8:XAIP xmlns:ns8="http://www.bsi.bund.de/tr-esor/xaip/1.2"
              xmlns:ns3="http://www.bsi.bund.de/tr-esor/api/1.2"
              xmlns:ns2="urn:oasis:names:tc:dss:1.0:core:schema" >
      <ns8:packageHeader packageID="QAPackageId">
        <ns8:versionManifest VersionID="QAVersionID">
          <ns8:preservationInfo>
            <ns8:retentionPeriod>2021-12-12</ns8:retentionPeriod>
          </ns8:preservationInfo>
          <ns8:packageInfoUnit packageUnitID="Package0123">
            <ns8:protectedObjectPointer>Signature_01</ns8:protectedObjectPointer>
            <ns8:unprotectedObjectPointer>Data_01</ns8:unprotectedObjectPointer>
          </ns8:packageInfoUnit>
        </ns8:versionManifest>
      </ns8:packageHeader>
      <ns8:dataObjectsSection>
        <ns8:dataObject dataObjectID="Data_01">
          <ns8:binaryData MimeType="">SukqAB5...AAAA==</ns8:binaryData>
        </ns8:dataObject>
      </ns8:dataObjectsSection>
      <ns8:credentialsSection>
        <ns8:credential relatedObjects="Data_01" credentialID="Signature_01">
          <ns2:SignatureObject>
            <ns2:Base64Signature>MIIRZ...MzD+Asu8=</ns2:Base64Signature>
          </ns2:SignatureObject>
        </ns8:credential>
      </ns8:credentialsSection>
    </ns8:XAIP>
  </ns3:ArchiveSubmissionRequest>
</S:Body>
```

ArchiveSubmissionResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header></soapenv:Header>
  <soapenv:Body>
    <ns2:ArchiveSubmissionResponse
      xmlns:ns10="http://ts.fujitsu.com/secdocs/v3_0/xaiparchiving"
      xmlns:ns9="urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:
schema#"
      xmlns:ns8="http://uri.etsi.org/01903/v1.3.2#"
      xmlns:ns7="http://www.bsi.bund.de/tr-esor/xaip/1.2"
      xmlns:ns6="http://www.setcce.org/schemas/ers"
      xmlns:ns5="urn:iso:std:iso-iec:24727:tech:schema"
      xmlns:ns4="http://www.bsi.bund.de/ecard/api/1.1"
      xmlns:ns3="http://www.w3.org/2000/09/xmldsig#"
      xmlns:ns2="http://www.bsi.bund.de/tr-esor/api/1.2"
      xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
      Profile="http://ts.fujitsu.com/secdocs/SecDocs 3.0A SOAP Service">
      <Result>
        <ResultMajor>SUCCESS</ResultMajor>
        <ResultMessage xml:lang="en">XAIP document submitted successfully
        </ResultMessage>
      </Result>
      <OptionalOutputs>
        <ns10:status>
          <ns10:statusCode>Success: XAIP document submitted successfully
          </ns10:statusCode>
        </ns10:status>
      </OptionalOutputs>
      <ns2:AOID>539c7326-05f5-4ba6-bd14-bf8dea7b116b</ns2:AOID>
    </ns2:ArchiveSubmissionResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

4.4.2 Operation ArchiveRetrieval

ArchiveRetrieval ruft ein Archivdatenobjekt konform zur technischen Richtlinie TR-ESOR aus dem Langzeitspeicher ab. Das abzurufende Archivdatenobjekt wird durch die Angabe einer AOID identifiziert.

Das mit der angegebenen AOID im Langzeitspeicher verknüpfte Archivdatenobjekt wird von SecDocs im Format XAIP (XML formatted Archive Information Package) zurückgeliefert. Dieses Format ist im Anhang F der Richtlinie TR-ESOR beschrieben (siehe "[BSI TR-03125 Anlage TR-ESOR-F](#)" ([W5] im Abschnitt "Literatur")).

Hinweise

- Die Operation ArchiveRetrieval wird abgewiesen, wenn keine gültige, d.h. syntaktisch korrekte und tatsächlich vergebene AOID angegeben wurde.
- Versionsangaben im ArchiveRetrievalRequest werden von SecDocs nicht ausgewertet.
- Die Möglichkeit der expliziten Anforderung, dass das zurückgelieferte XAIP-Datenobjekt den bzw. die entsprechenden Evidence Record(s) enthalten soll, wird derzeit nicht unterstützt.
Ist das abzurufende Archivdatenobjekt bereits versiegelt, enthält es den/die entsprechenden Evidence Record(s) ohnehin.

Request

Element ArchiveRetrievalRequest

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

Die Beschreibung des Datentyps RequestType finden Sie im [Abschnitt „Request“](#).

AOID

string:

Identifikation des Archivdatenobjekts, das abgerufen werden soll.

VersionID

string:

Dieses Element ist für zukünftige Erweiterungen vorgesehen und sollte nicht angegeben werden.

Eingaben im Element OptionalInputs:

SecDocs wertet Angaben im Element `OptionalInputs` nicht aus.

Response

Element `ArchiveRetrievalResponse`

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

Die Beschreibung des Datentyps `ResponseType` finden Sie im [Abschnitt „Response“](#).

XAIP Datentyp `XAIPTyp`:

Abgerufenes Archivdatenobjekt im XAIP-Format.

Dieses Format XAIP ist in der Schemadatei `tr-esor-xaip-v1_2.xsd` (in "Web-Service S4 für die Client-Anwendung") definiert und im Anhang F der Richtlinie TR-ESOR beschrieben (siehe "BSI TR-03125 Anlage TR-ESOR-F" ([W5] im Abschnitt "Literatur")). Die Schemadatei wird vom BSI unter https://www.bsi.bund.de/DE/Publikationen/TechnischeRichtlinien/tr03125/index_hm.html zur Verfügung gestellt und auch mit SecDocs ausgeliefert.

Der Target-Namespace dieser Datei ist

`http://www.bsi.bund.de/tr-esor/xaip/1.2.`

Beachten Sie:

Die Signaturverifikationsdaten sowie die technischen Beweisdaten (Evidence Records) werden von SecDocs nicht in den Elementen `VerificationReport` bzw. `evidenceRecord`, sondern im Element `other` abgespeichert. Zusätzlich wird von SecDocs der Name des Zeitstempelanbieters (Time Stamp Provider, TSP) hinterlegt. Siehe hierzu auch [Abschnitt „SecDocs-spezifische Erweiterungen des Formats XAIP“](#).

Ausgaben im Element `Result`:

`ResultMajor`

Im Erfolgsfall:

`SUCCESS`

Im Fehlerfall:

`FAILURE`

`ResultMessage`

Im Erfolgsfall:

XAIP document retrieved successfully

Im Fehlerfall:

processing not successful

Ausgaben im Element OptionalOutputs:

statusCode im Element status

Im Erfolgsfall:

Success: XAIP document retrieved successfully

Im Fehlerfall:

Failure: processing not successful

faultDetails

Datentyp TFaultDetails, siehe [Abschnitt „Status- und Fehlerinformation“](#).

Dieses Element wird nur im Fehlerfall ausgegeben.

Beispiel

SOAP-Body des ArchiveRetrievalRequest

```
<S:Body>
  <ns3:ArchiveRetrievalRequest xmlns:ns3="http://www.bsi.bund.de/tr-esor/api/1.2" >
    <ns3:AOID>131c038e-8608-45c5-83fb-84a8e155dc87</ns3:AOID>
  </ns3:ArchiveRetrievalRequest>
</S:Body>
```

ArchiveRetrievalResponse

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header></soapenv:Header>
  <soapenv:Body>
    <tr:ArchiveRetrievalResponse
      Profile="http://ts.fujitsu.com/secdocs/SecDocs 3.0A SOAP Service"
      xmlns:tr="http://www.bsi.bund.de/tr-esor/api/1.2">
      <Result xmlns="urn:oasis:names:tc:dss:1.0:core:schema">
        <ResultMajor>SUCCESS</ResultMajor>
        <ResultMessage xml:lang="en">XAIP document retrieved successfully
        </ResultMessage>
      </Result>
      <ns2:OptionalOutputs
        xmlns:ns10="http://ts.fujitsu.com/secdocs/v3_0/xaiparchiving"
        xmlns:ns9="urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:schema#"
        xmlns:ns8="http://www.setcce.org/schemas/ers"
        xmlns:ns7="http://uri.etsi.org/01903/v1.3.2#"
        xmlns:ns6="http://www.bsi.bund.de/tr-esor/xaip/1.2"
        xmlns:ns5="urn:iso:std:iso-iec:24727:tech:schema"
        xmlns:ns4="http://www.bsi.bund.de/ecard/api/1.1"
        xmlns:ns3="http://www.w3.org/2000/09/xmldsig#"
```



```

        xmlns:ns2="urn:oasis:names:tc:dss:1.0:core:schema">
    <ns10:status>
        <ns10:statusCode>Success: XAIP document retrieved successfully
    </ns10:statusCode>
    </ns10:status>
</ns2:OptionalOutputs>
<ns8:XAIP xmlns:ns2="urn:oasis:names:tc:dss:1.0:core:schema"
    xmlns:ns3="http://www.bsi.bund.de/tr-esor/api/1.2"
    xmlns:ns8="http://www.bsi.bund.de/tr-esor/xaip/1.2">
    <ns8:packageHeader packageID="Package0123">
        <ns8:AOID>131c038e-8608-45c5-83fb-84a8e155dc87</ns8:AOID>
        <ns8:versionManifest VersionID="Version04">
            <ns8:preservationInfo>
                <ns8:retentionPeriod>2025-12-31</ns8:retentionPeriod>
            </ns8:preservationInfo>
            <ns8:packageInfoUnit packageUnitID="PackagePart01">
                <ns8:protectedObjectPointer>Signature_01</ns8:protectedObjectPointer>
                <ns8:unprotectedObjectPointer>Data_01</ns8:unprotectedObjectPointer>
                <ns8:protectedObjectPointer>cid_93886ebe-e040-453a-b29d-5c8a6f2e28eb
            </ns8:protectedObjectPointer>
            </ns8:packageInfoUnit>
        </ns8:versionManifest>
    </ns8:packageHeader>
    <ns8:dataObjectsSection>
        <ns8:dataObject dataObjectID="Data_01">
            <ns8:binaryData MimeType="">SUKqAB5...AAAAA==</ns8:binaryData>
        </ns8:dataObject>
    </ns8:dataObjectsSection>
    <ns8:credentialsSection>
        <ns8:credential credentialID="Signature_01" relatedObjects="Data_01">
            <ns2:SignatureObject>
                <ns2:Base64Signature>MIIRZ...MzD+Asu8=</ns2:Base64Signature>
            </ns2:SignatureObject>
        </ns8:credential>
        <ns8:credential credentialID="cid_93886ebe-e040-453a-b29d-5c8a6f2e28eb"
            relatedObjects="Package0123">
            <ns8:other>
                <sd:vrInfo xmlns:sd="http://ts.fujitsu.com/secdocs/v3_0/xaiparchiving"
>
                    PD94bWw...ZvMT4K
                </sd:vrInfo>
            </ns8:other>
        </ns8:credential>
        <ns8:credential
            credentialID="cid_279b69f1-d85d-4616-bdb9-4cb2667c2f97"
            relatedObjects="Signature_01 cid_93886ebe-e040-453a-b29d-5c8a6f2e28eb"
>
            <ns8:other>
                <sd:evidenceRecord
                    xmlns:sd="http://ts.fujitsu.com/secdocs/v3_0
/xaiparchiving"
                        tsp="SecDocsTestTSP">
                            MIIV9AIBATA...Sr0+nWfa+1Q=
                        </sd:evidenceRecord>
            </ns8:other>
        </ns8:credential>
    </ns8:credentialsSection>
</ns8:XAIP>
</tr:ArchiveRetrievalResponse>

```

```
</soapenv:Body>  
</soapenv:Envelope>
```

4.4.3 Operation ArchiveEvidence

ArchiveEvidence ruft konform zur technischen Richtlinie TR-ESOR technische Beweisdaten (Evidence Records) zu einem Archivdatenobjekt aus dem Langzeitspeicher ab. Damit ist ein lückenloser Nachweis der Authentizität und Integrität seit dem Zeitpunkt der Archivierung des Archivdatenobjektes möglich.

Das Archivdatenobjekt, dessen Beweisdaten abgerufen werden sollen, wird durch die Angabe einer AOID identifiziert.

Die Beweisdaten des durch die AOID angegebenen Archivdatenobjektes werden von SecDocs generell in ASN.1 Syntax gemäß der Spezifikation im Standard IETF RFC 4998 zurückgeliefert. Sie enthalten sämtliche Informationen, die zur Verifikation der Authentizität und Integrität der gespeicherten Daten, deren Signaturen, Zertifikaten und der Signaturerneuerungen benötigt werden.

Hinweise

- Ist das Archivdatenobjekt noch nicht versiegelt, kann die Operation ArchiveEvidence keine Beweisdaten zurückliefern.
- Die Operation ArchiveEvidence wird abgewiesen, wenn keine gültige, d.h. syntaktisch korrekte und tatsächlich vergebene AOID angegeben wurde.
- Versionsangaben im ArchiveEvidenceRequest werden von SecDocs nicht ausgewertet.
- Formatangaben im ArchiveEvidenceRequest für das zurückzuliefernde Format der Evidence Records werden von SecDocs nicht ausgewertet.

Request

Element ArchiveEvidenceRequest

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

Die Beschreibung des Datentyps RequestType finden Sie im [Abschnitt „Request“](#).

AOID

string:

Identifikation des Archivdatenobjektes, dessen Beweisdaten abgerufen werden sollen.

VersionID

string:

Diese Angabe wird von SecDocs derzeit nicht ausgewertet.

VersionID wird evtl. in einer Folgeversion von SecDocs ausgewertet und sollte daher nicht angegeben werden.

Eingaben im Element `OptionalInputs`:

SecDocs wertet Angaben im Element `OptionalInputs` nicht aus.

Response

Element `ArchiveEvidenceResponse`

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

Die Beschreibung des Datentyps `ResponseType` finden Sie im [Abschnitt „Response“](#).

`evidenceRecord`

Datentyp `EvidenceRecordType`:

Abgerufener Evidence Record in ASN.1 Syntax gemäß der Spezifikation im Standard IETF RFC 4998.

Der Datentyp `EvidenceRecordType` ist in den folgenden vom BSI bereitgestellten Schemadateien definiert:

[tr-esor-xaip-v1_2.xsd](#) (in "Web-Service S4 für die Client-Anwendung")

(Target-Namespace `http://www.bsi.bund.de/tr-esor/xaip/1.2`)

und

`eCard.xsd`

(Target-Namespace `http://www.bsi.bund.de/ecard/api/1.1`)

Diese Datei ist in der Datei [tr-esor-schema-standalone-v1.2.zip](#) (in "Web-Service S4 für die Client-Anwendung") enthalten.

Datentyp xaip:EvidenceRecordType

```
<xs:complexType name="EvidenceRecordType" >
  <xs:complexContent>
    <xs:extension base="ec:EvidenceRecordType">
      <xs:attribute name="AOID" type="xs:string" />
      <xs:attribute name="VersionID" type="xs:string" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

AOID

string:
Dieses Attribut wird von SecDocs nicht ausgegeben.

VersionID

string:
Dieses Attribut wird von SecDocs nicht ausgegeben.

Datentyp ec:EvidenceRecordType

```
<complexType name="EvidenceRecordType">
  <choice>
    <element name="xmlEvidenceRecord" type="ers:EvidenceRecordType" />
    <element name="asn1EvidenceRecord" type="base64Binary" />
  </choice>
</complexType>
```

xmlEvidenceRecord

Datentyp ers:EvidenceRecordType:
Dieses Element wird von SecDocs nicht ausgegeben.

asn1EvidenceRecord

base64Binary:
Evidence Record in ASN.1 Syntax (gemäß Standard IETF RFC 4998).

Ausgaben im Element Result:

ResultMajor

Im Erfolgsfall:

SUCCESS

Im Fehlerfall:

FAILURE

ResultMessage

Im Erfolgsfall sind folgende Werte möglich:

evidence records retrieved successfully

no evidence records exist

Im Fehlerfall:

processing not successful

Ausgaben im Element OptionalOutputs:

statusCode im Element status

Im Erfolgsfall sind folgende Werte möglich:

Success: evidence records retrieved successfully

Success: no evidence records exist

Im Fehlerfall:

Failure: processing not successful

faultDetails

Datentyp TFaultDetails, siehe Kapitel [Abschnitt „Status- und Fehlerinformation“](#).

Dieses Element wird nur im Fehlerfall ausgegeben.

Beispiel

SOAP-Body des ArchiveEvidenceRequest

```
<soap:Body>
  <tr:ArchiveEvidenceRequest xmlns:tr="http://www.bsi.bund.de/tr-esor/api/1.2">
    <tr:AOID>131c038e-8608-45c5-83fb-84a8e155dc87</tr:AOID>
  </tr:ArchiveEvidenceRequest>
</soap:Body>
```

ArchiveEvidenceResponse

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header></soapenv:Header>
  <soapenv:Body>
    <tr:ArchiveEvidenceResponse
      Profile="http://ts.fujitsu.com/secdocs/SecDocs 3.0A SOAP Service"
      xmlns:tr="http://www.bsi.bund.de/tr-esor/api/1.2">
      <Result xmlns="urn:oasis:names:tc:dss:1.0:core:schema">
        <ResultMajor>SUCCESS</ResultMajor>
        <ResultMessage xml:lang="en">evidence records retrieved successfully
        </ResultMessage>
      </Result>
      <ns2:OptionalOutputs
        xmlns:ns10="http://ts.fujitsu.com/secdocs/v3_0/xaiparchiving"
        xmlns:ns9="urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:
schema#"
        xmlns:ns8="http://uri.etsi.org/01903/v1.3.2#"
        xmlns:ns7="http://www.bsi.bund.de/tr-esor/xaip/1.2"
        xmlns:ns6="http://www.setcce.org/schemas/ers"
        xmlns:ns5="urn:iso:std:iso-iec:24727:tech:schema"
        xmlns:ns4="http://www.bsi.bund.de/ecard/api/1.1"
        xmlns:ns3="http://www.w3.org/2000/09/xmldsig#"
        xmlns:ns2="urn:oasis:names:tc:dss:1.0:core:schema">
        <ns10:status>
          <ns10:statusCode>Success: evidence records retrieved successfully
          </ns10:statusCode>
        </ns10:status>
      </ns2:OptionalOutputs>
      <xaip:evidenceRecord xmlns:xaip="http://www.bsi.bund.de/tr-esor/xaip/1.2"
        xmlns:ec="http://www.bsi.bund.de/ecard/api/1.1">
        <ec:asn1EvidenceRecord>MIIV9AIBATA...TSzhiSr0+nWfa+lQ=</ec:asn1EvidenceRecord>
      </xaip:evidenceRecord>
    </tr:ArchiveEvidenceResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

ArchiveEvidenceResponse, falls keine Evidence Records vorhanden sind:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header></soapenv:Header>
  <soapenv:Body>
    <tr:ArchiveEvidenceResponse
      Profile="http://ts.fujitsu.com/secdocs/SecDocs 3.0A SOAP Service"
      xmlns:tr="http://www.bsi.bund.de/tr-esor/api/1.2">
      <Result xmlns="urn:oasis:names:tc:dss:1.0:core:schema">
        <ResultMajor>SUCCESS</ResultMajor>
        <ResultMessage xml:lang="en">no evidence records exist
        </ResultMessage>
      </Result>
      <ns2:OptionalOutputs
        xmlns:ns10="http://ts.fujitsu.com/secdocs/v3_0/xaiparchiving"
        xmlns:ns9="urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:
schema#"
        xmlns:ns8="http://www.setcce.org/schemas/ers"
        xmlns:ns7="http://uri.etsi.org/01903/v1.3.2#"
        xmlns:ns6="http://www.bsi.bund.de/tr-esor/xaip/1.2"
        xmlns:ns5="urn:iso:std:iso-iec:24727:tech:schema"
        xmlns:ns4="http://www.bsi.bund.de/ecard/api/1.1"
        xmlns:ns3="http://www.w3.org/2000/09/xmldsig#"
        xmlns:ns2="urn:oasis:names:tc:dss:1.0:core:schema">
        <ns10:status>
          <ns10:statusCode>Success: no evidence records exist</ns10:statusCode>
        </ns10:status>
      </ns2:OptionalOutputs>
    </tr:ArchiveEvidenceResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

4.4.4 Operation ArchiveDeletion

`ArchiveDeletion` löscht ein Archivdatenobjekt konform zur technischen Richtlinie TR-ESOR im Langzeitspeicher. Das zu löschende Archivdatenobjekt wird durch die Angabe einer AOID identifiziert.

Wenn Sie das Archivdatenobjekt vor dem Ablauf der eingestellten Aufbewahrungsfrist löschen wollen, müssen Sie neben dem Namen der aufrufenden Instanz auch einen Begründungstext für diese Aktion übermitteln.

SecDocs liefert eine Statusmeldung über das erfolgreiche Löschen zurück.

SecDocs führt im Rahmen einer `ArchiveDeletion`-Operation folgende Aktionen durch:

- SecDocs überprüft die eingestellte Aufbewahrungsfrist und prüft, ob ggf. eine Begründung für die Löschaktion im `ArchiveDeletionRequest` enthalten ist.
- Das angegebene Archivdatenobjekt wird im Langzeitspeicher gelöscht.
- Der Löschvorgang und die einzelnen Elemente der Begründung (falls vorhanden) werden in die mandantenspezifische Audit-Log-Datei protokolliert.

Voraussetzungen

- Die eingestellte Aufbewahrungsfrist ist abgelaufen oder der `ArchiveDeletionRequest` enthält eine Begründung für die Löschaktion.

Hinweise

- Die Operation `ArchiveDeletion` wird abgewiesen, wenn keine gültige, d.h. syntaktisch korrekte und tatsächlich vergebene AOID angegeben wurde.
- Die Operation `ArchiveDeletion` wird abgewiesen, wenn die eingestellte Aufbewahrungsfrist noch nicht abgelaufen ist und der `ArchiveDeletionRequest` keine Begründung (Element `ReasonOfDeletion`, siehe unten) enthält.

Request

Element `ArchiveDeletionRequest`

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

Die Beschreibung des Datentyps `RequestType` finden Sie im [Abschnitt „Request“](#).

AOID

string:

Identifikation des Archivdatenobjekts, das gelöscht werden soll.

Eingaben im Element `OptionalInputs`:

`ReasonOfDeletion`

Dieses Element müssen Sie angeben, wenn der eingestellte Aufbewahrungszeitraum noch nicht abgelaufen ist.

Die Sub-Elemente `RequestorName` und `RequestInfo` müssen beide angegeben sein.

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

`RequestorName`

Datentyp `NameIDType`:

Eindeutiges Identifikationsmerkmal der aufrufenden Client-Anwendung.

Der Datentyp `NameIDType` ist im BSI-Schema [saml-schema-assertion-2.0.xsd](#) (in "Web-Service S4 für die Client-Anwendung") (Target-Namespace: `urn:oasis:names:tc:SAML:2.0:assertion`) näher beschrieben.

`RequestInfo`

string:

Begründungstext für die Löschaktion.

Datentyp `NameIDType`

```
<complexType name="NameIDType">
  <simpleContent>
    <extension base="string">
      <attributeGroup ref="saml:IDNameQualifiers" />
      <attribute name="Format" type="anyURI" use="optional" />
      <attribute name="SPProvidedID" type="string" use="optional" />
    </extension>
  </simpleContent>
</complexType>
```

Attributgruppe IDNameQualifiers

```
<attributeGroup name="IDNameQualifiers">
  <attribute name="NameQualifier" type="string" use="optional" />
  <attribute name="SPNameQualifier" type="string" use="optional" />
</attributeGroup>
```

Response

Element ArchiveDeletionResponse

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

Die Beschreibung des Datentyps ResponseType finden Sie im [Abschnitt „Response“](#).

Ausgaben im Element Result:

ResultMajor

Im Erfolgsfall:

SUCCESS

Im Fehlerfall:

FAILURE

ResultMessage

Im Erfolgsfall:

all versions of XAIP document deleted successfully

Im Fehlerfall:

processing not successful

Ausgaben im Element OptionalOutputs:

statusCode im Element status

Im Erfolgsfall:

Success: all versions of XAIP document deleted successfully

Im Fehlerfall:

Failure: processing not successful

faultDetails

Datentyp TFaultDetails, siehe Kapitel [Abschnitt „Status- und Fehlerinformation“](#).

Dieses Element wird nur im Fehlerfall ausgegeben.

Beispiel

SOAP-Body des ArchiveDeletionRequest

```
<soap:Body>
  <tr:ArchiveDeletionRequest xmlns:tr="http://www.bsi.bund.de/tr-esor/api/1.2"
                             xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema">
    <dss:OptionalInputs>
      <tr:ReasonOfDeletion>
        <tr:RequestorName>AdministratorXY</tr:RequestorName>
        <tr:RequestInfo>some compelling reason</tr:RequestInfo>
      </tr:ReasonOfDeletion>
    </dss:OptionalInputs>
    <tr:AOID>f179d2a0-815a-4f5a-886d-813515cd548e</tr:AOID>
  </tr:ArchiveDeletionRequest>
</soap:Body>
```

ArchiveDeletionResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header></soapenv:Header>
  <soapenv:Body>
    <ns2:ArchiveDeletionResponse
      xmlns:ns10="http://ts.fujitsu.com/secdocs/v3_0/xaiparchiving"
      xmlns:ns9="urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:
schema#"
      xmlns:ns8="http://uri.etsi.org/01903/v1.3.2#"
      xmlns:ns7="http://www.bsi.bund.de/tr-esor/xaip/1.2"
      xmlns:ns6="http://www.setcce.org/schemas/ers"
      xmlns:ns5="urn:iso:std:iso-iec:24727:tech:schema"
      xmlns:ns4="http://www.bsi.bund.de/ecard/api/1.1"
      xmlns:ns3="http://www.w3.org/2000/09/xmldsig#"
      xmlns:ns2="http://www.bsi.bund.de/tr-esor/api/1.2"
      xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
      Profile="http://ts.fujitsu.com/secdocs/SecDocs 3.0A SOAP Service">
      <Result>
        <ResultMajor>SUCCESS</ResultMajor>
        <ResultMessage xml:lang="en">all versions of XAIP document deleted successfully
        </ResultMessage>
      </Result>
      <OptionalOutputs>
        <ns10:status>
          <ns10:statusCode>Success: all versions of XAIP document deleted
successfully
          </ns10:statusCode>
        </ns10:status>
      </OptionalOutputs>
    </ns2:ArchiveDeletionResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Protokolleinträge in der mandantenspezifischen Audit-Log-Datei (zum Aufbau der Einträge siehe Abschnitt „Die Audit-Log-Datei“ im Handbuch ["SecDocs Administration und Bedienung"](#) ([SD1] im Abschnitt "Literatur")):

```
<110>1 2015-12-11T16:15:33.778+01:00 172.17.32.100 SecDocs 25369 DELETE_SDO [SecDocs:
OpenLimit@231
requestNumber="28" SessionID="2419c7b3-3e70-4b51-belf-a97102feldfc" IdentToken="SecDocs"
Aoid="f179d2a0-815a-4f5a-886d-813515cd548e"] [SecDocs:audit@231]
<110>1 2015-12-11T16:15:33.821+01:00 172.17.32.100 SecDocs 25369 DELETE_SDO [SecDocs:
OpenLimit@231
requestNumber="28" SessionID="2419c7b3-3e70-4b51-belf-a97102feldfc" IdentToken="SecDocs"
Aoid="f179d2a0-815a-4f5a-886d-813515cd548e" MSG="Current date:Fri Dec 11 16:15:33 CET 2015,
ExpireDate: Wed Dec 31 01:00:00 CET 2025, Reason: AdministratorXY: some compelling reason
successfully
deleted!"] [SecDocs:audit@231]
```

4.4.5 Änderung der Aufbewahrungszeit

Die Änderung der Aufbewahrungszeit ist mit der in der TR-ESOR 1.2 Richtlinie beschriebene ArchiveUpdate-Request möglich. Allerdings ist nur die Änderung der Aufbewahrungszeit erlaubt, alle anderen in der Richtlinie erlaubten Änderungen werden abgelehnt.

! Mit dieser eingeschränkten Version des ArchiveUpdate ist keine Versionierung von XAIP-Dokumenten implementiert. Dem XAIP wird lediglich das versionManifest mit dem neuen Ablaufdatum hinzugefügt. Ein ArchiveRetrieval-Request liefert immer die jüngste Version des Dokuments.

! ArchiveUpdate ist erst nach erfolgreicher Versiegelung möglich

! Das Ablaufdatum kann nicht in die Vergangenheit verschoben werden.

Request

Element ArchiveUpdateRequest

```
<element name="ArchiveUpdateRequest">
  <complexType>
    <complexContent>
      <extension base="tr:RequestType">
        <sequence>
          <element ref="xaip:DXAIP"></element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
```

Element DXAIP

```
<element name="DXAIP" type="xaip:DXAIPType" />
<complexType name="DXAIPType">
  <complexContent>
    <extension base="xaip:XAIPTYPE">
      <sequence>
        <element name="updateSection" type="xaip:updateSectionType" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
</element>
```

DXAIP Datentyp DXAIPType:

Zu verändernde Teile im Delta-XAIP Format (DXAIP). Das DXAIP-Format ist ein XAIP-Format, erweitert um das Element `updateSection`, das heißt alle in TR-ESOR 2.1 Anlage F beschriebenen Elemente des XAIP-Formats können prinzipiell beim `ArchiveUpdate` angegeben werden. Die Formate XAIP und DXAIP sind in der Schemadatei [tr-esor-xaip-v1_2.xsd](#) definiert und im Anhang F der Richtlinie TR-ESOR beschrieben (siehe "BSI TR-03125 Anlage TR-ESOR-F" ([W5] in Abschnitt "Literatur")). Die Schemadatei wird vom BSI unter https://www.bsi.bund.de/DE/Publikationen/TechnischeRichtlinien/tr03125/index_hm.html zur Verfügung gestellt und auch mit SecDocs ausgeliefert.

Der Target-Namespace dieser Datei ist <http://www.bsi.bund.de/tr-esor/xaip/1.2>.

Hinweise zu einzelnen Elementen:

Ab SecDocs V3.0A04 ist nur das Ändern der Aufbewahrungszeit möglich, deshalb behandelt SecDocs bestimmte Elemente des übergebenen DXAIP-Datenobjekts anders als in der Richtlinie beschrieben. Im Folgenden werden die wesentlichen Elemente und ihre Bedeutung aufgelistet

packageHeaderElement

Das `packageHeader`-Element besteht aus der Abfolge von folgenden Elementen:

1. `AOID`
2. `packageInfo`
3. `versionManifest`

Die Elemente `CanonicalizationMethod` und `extension` sind laut TR-ESOR beim Update nicht erlaubt.

AOID:

string

Identifikation des Archivdatenobjekts, das abgerufen werden soll.

packageInfo

Das `packageInfo`-Element darf nicht angegeben werden

versionManifest

Im `versionManifest`-Element werden alle relevanten Informationen zu der neuen Version übergeben. Das hier angegebene Element wird in das `packageHeader`-Element des XAIP-Dokuments aufgenommen

Hinweise zu den Elementen

packageInfoUnit

Laut TR-ESOR kann man über die Elemente `protectedObjectPointer` und `unprotectedObjectPointer` steuern, ob Datenobjekte, Credentials oder Metadaten beibehalten oder gelöscht werden sollen. Objekte, die in keiner der beiden Listen aufgeführt sind, sind zu löschen. Da in der vorliegenden Implementierung keine Daten gelöscht werden können, müssten alle oben genannten Objekte in einer der beiden Listen enthalten sein.

Um aber ein Arbeiten mit der neuen Funktion zu erleichtern, ist folgende Vorgehensweise erlaubt: Das Element `packageInfoUnit` muss alle `protectedObjectPointer`-Elemente enthalten, wie sie bereits im Originaldokument beim Archivieren mit der Funktion `ArchiveSubmission` enthalten waren. Verweise auf eventuell von SecDocs während des Submits oder der Versiegelung hinzugefügte Objekte wie den Signaturprüfbericht werden von SecDocs ergänzt, wenn sie nicht angegeben sind.

Werden keine `unprotectedObjectPointer` angegeben, so erzeugt SecDocs `unprotectedObjectPointer`-Elemente für alle fehlenden Daten, fügt sie dem Element `packageInfoUnit` hinzu und übernimmt das veränderte `versionManifest`-Element in das XAIP-Dokument.

metaDataSection, dataObjectsSection, credentialsSection

Da nur die Aufbewahrungszeit geändert werden kann, dürfen diese Elemente nicht angegeben werden.

updateSection: Datentyp updateSectionType

```
<xs:complexType name="updateSectionType">
  <xs:sequence>
    <xs:element name="prevVersion" type="xs:string" />
    <xs:element name="placeholder" type="xaip:placeholderType" maxOccurs="unbounded"
      minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

prevVersion: string

Version, die geändert werden soll.

Die Vorgängerversion des Dokuments muss so angegeben werden muss, wie sie im `VersionID`-Attribut des `versionManifest`-Elements dieser Version steht. Erlaubt ist hier nur die Versions-ID der jüngsten Version, anders lautende Angaben werden mit Fehlermeldung abgelehnt.

placeholder: Datentyp placeholderType

```
<xs:complexType name="placeholderType">
  <xs:attribute name="objectID" type="xs:ID" use="required" />
</xs:complexType>
```

Werden im `versionManifest` `protectedObjectPointer` oder `unprotectedObjectPointer` angegeben, so muss das `updateSection`-Element für jedes Objekt, das dort angegeben ist ein `placeholder`-Element enthalten, das jeweils im Attribut `ObjectID` die ID des `metaDataObject`-, `dataObject`- und `credential`-Element enthält, auf das in den `protectedObjectPointer`- und `unprotectedObjectPointer`-Elementen verwiesen wird

Response

Element ArchiveUpdateResponse

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

VersionID

Ist im Erfolgsfall vorhanden und enthält den bezüglich des über die AOID identifizierten Archivdatenobjektes eindeutigen Versions-Identifikator, wie er im Request im versionManifest-Element angegeben wurde

Ausgaben im Element Result:

ResultMajor

Im Erfolgsfall:

SUCCESS

Im Fehlerfall:

FAILURE

ResultMessage

Im Erfolgsfall sind folgende Werte möglich:

Retention period updated successfully

Im Fehlerfall:

processing not successful

Ausgaben im Element OptionalOutputs:

statusCode im Element status

Im Erfolgsfall sind folgende Werte möglich:

Success: Retention period updated successfully

Im Fehlerfall:

Failure: processing not successful

faultDetails

Datentyp TFaultDetails, siehe Abschnitt "[Status- und Fehlerinformation](#)".
Dieses Element wird nur im Fehlerfall ausgegeben.

Beispiel

Dokument auf dem Speicher (nach Versiegelung)

```

<xaip:XAIP xmlns:ds=http://www.w3.org/2000/09/xmldsig#
  xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema"
  xmlns:ec=http://www.bsi.bund.de/ecard/api/1.1
  xmlns:tr=http://www.bsi.bund.de/tr-esor/api/1.2
  xmlns:xaip="http://www.bsi.bund.de/tr-esor/xaip/1.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema">
  <xaip:packageHeader packageID="test_packageID">
    <xaip:AOID>ykzmpyyh/xaip:AOID>
    <xaip:versionManifest VersionID="v1">
      <xaip:preservationInfo>
        <xaip:retentionPeriod>2018-09-04</xaip:retentionPeriod>
      </xaip:preservationInfo>
      <xaip:packageInfoUnit packageUnitID="test_packageUnitID">
        <xaip:protectedObjectPointer>Data1
        </xaip:protectedObjectPointer>
        <xaip:protectedObjectPointer>cid_031d27a8-1504-4d99-af29-40760130f840
        </xaip:protectedObjectPointer>
      </xaip:packageInfoUnit>
    </xaip:versionManifest>
    <CanonicalizationMethod xmlns=http://www.w3.org/2000/09/xmldsig#
      Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315">
    </CanonicalizationMethod>
  </xaip:packageHeader>
  <xaip:dataObjectsSection>
    <xaip:dataObject dataObjectID="Data1">
      <xaip:binaryData MimeType="application/pdf">JVBE...==
      </xaip:binaryData>
    </xaip:dataObject>
  </xaip:dataObjectsSection>
  <xaip:credentialsSection>
    <xaip:credential credentialID="Signatur1" relatedObjects="Data1">
      <dss:SignatureObject>
        <dss:Base64Signature Type="urn:ietf:rfc:3852">MIA...==
        </dss:Base64Signature>
      </dss:SignatureObject>
    </xaip:credential>
    <xaip:credential credentialID="cid_031d27a8-1504-4d99-af29-40760130f840"
      relatedObjects="test_packageID">
      <xaip:other>
        <sd:vrInfo xmlns:sd="http://ts.fujitsu.com/secdocs/v3_0/xaiparchiving">
          PD94...==
        </sd:vrInfo>
      </xaip:other>
    </xaip:credential>
    <xaip:credential credentialID="cid_556cb322-0f33-4097-ae7-ec1adb79b700"
      relatedObjects="Data1 cid_031d27a8-1504-4d99-af29-40760130f840">
      <xaip:other>
        <sd:evidenceRecord
          xmlns:sd="http://ts.fujitsu.com/secdocs/v3_0/xaiparchiving" tsp="REG_TESTS_TSP"
        >MIIV...==
        </sd:evidenceRecord>
      </xaip:other>
    </xaip:credential>
  </xaip:credentialsSection>
</xaip:XAIP>

```

SOAP-Body eines ArchiveUpdate-Requests:

```
<tr:ArchiveUpdateRequest xmlns:tr="http://www.bsi.bund.de/tr-esor/api/1.2">
  <xaip:DXAIP xmlns:tr="http://www.bsi.bund.de/tr-esor/api/1.2"
    xmlns:xaip="http://www.bsi.bund.de/tr-esor/xaip/1.2"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema"
    xmlns:ec="http://www.bsi.bund.de/ecard/api/1.1">
    <xaip:packageHeader packageID="BMFSFJ-Z2-231245023-42-Stellungnahme">
      <xaip:AOID>ykzmpyyh</xaip:AOID>
      <xaip:versionManifest VersionID="v2">
        <xaip:preservationInfo>
          <xaip:retentionPeriod>2019-09-05</xaip:retentionPeriod>
        </xaip:preservationInfo>
        <xaip:packageInfoUnit packageUnitID="afqiwyfgyr">
          <xaip:protectedObjectPointer>Data1 | (1)
        </xaip:protectedObjectPointer>
          <xaip:unprotectedObjectPointer>Signatur1 | (1)
        </xaip:unprotectedObjectPointer>
        </xaip:packageInfoUnit>
      </xaip:versionManifest>
    </xaip:packageHeader>
    <xaip:updateSection>
      <xaip:prevVersion>v1</xaip:prevVersion>
      <xaip:placeholder objectID="Data1" /> | (1)
      <xaip:placeholder objectID="Signatur1" /> | (1)
    </xaip:updateSection>
  </xaip:DXAIP>
</tr:ArchiveUpdateRequest>
```

(1) Jede ID, die im packageInfoUnit-Element verwendet wird (hier: Data1, Signatur1), muss mittels eines placeholder-Elements in der update-Section definiert werden

Soap-Body der ArchiveUpdate-Response

```

<ns2:ArchiveUpdateResponse
  xmlns:ns10="http://ts.fujitsu.com/secdocs/v3_0/xaiparchiving"
  xmlns:ns9="urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:schema#"
  xmlns:ns8="http://uri.etsi.org/01903/v1.3.2#" xmlns:ns7="http://www.bsi.bund.de/tr-esor/xaip/1.2"
  xmlns:ns6="http://www.setcce.org/schemas/ers" xmlns:ns5="urn:iso:std:iso-iec:24727:tech:schema"
  xmlns:ns4="http://www.bsi.bund.de/ecard/api/1.1" xmlns:ns3="http://www.w3.org/2000/09/xmldsig#"
  xmlns:ns2="http://www.bsi.bund.de/tr-esor/api/1.2" xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
  Profile="http://ts.fujitsu.com/secdocs/SecDocs 3.0A SOAP Service">
  <Result>
    <ResultMajor>SUCCESS</ResultMajor>
    <ResultMessage xml:lang="en">Retention period updated successfully</ResultMessage>
  </Result>
  <OptionalOutputs>
    <ns10:status>
      <ns10:statusCode>Success: Retention period updated successfully</ns10:statusCode>
    </ns10:status>
  </OptionalOutputs>
  <ns2:VersionID>v2</ns2:VersionID>
</ns2:ArchiveUpdateResponse>

```

XAIP-Dokument nach erfolgreichem Update

```

<xaip:XAIP xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema"
  xmlns:ec="http://www.bsi.bund.de/ecard/api/1.1"
  xmlns:tr="http://www.bsi.bund.de/tr-esor/api/1.2" xmlns:xaip="http://www.bsi.bund.de/tr-esor/xaip/1.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema">
  <xaip:packageHeader packageID="test_packageID">
    <xaip:AOID>ykzmpyyh</xaip:AOID>
    <xaip:versionManifest VersionID="v1">
      <xaip:preservationInfo>
        <xaip:retentionPeriod>2018-09-04</xaip:retentionPeriod>
      </xaip:preservationInfo>
      <xaip:packageInfoUnit
        packageUnitID="test_packageUnitID">
          <xaip:protectedObjectPointer>Data1</xaip:protectedObjectPointer>
          <xaip:protectedObjectPointer> cid_031d27a8-1504-4d99-af29-40760130f840</xaip:protectedObjectPointer>
        </xaip:packageInfoUnit>
      </xaip:versionManifest>
      <xaip:versionManifest VersionID="v2">
        <xaip:preservationInfo>
          <xaip:retentionPeriod>2019-09-05</xaip:retentionPeriod>
        </xaip:preservationInfo>
        <xaip:packageInfoUnit packageUnitID="afqiwyfgyr">
          <xaip:protectedObjectPointer>Data1</xaip:protectedObjectPointer>
        </xaip:packageInfoUnit>
      </xaip:versionManifest>
    </xaip:versionManifest>
  </xaip:packageHeader>

```

<xaip:protectedObjectPointer>cid_ba0a0f11-2ec2-4341-a8ad-f4d0860711a9		(1)
</xaip:protectedObjectPointer>		(2)
<xaip:unprotectedObjectPointer>Signatur1		
</xaip:unprotectedObjectPointer>		
<xaip:unprotectedObjectPointer>cid_50151c29-d488-4602-a0a8-d1c18209e180		
</xaip:unprotectedObjectPointer>		(2)
</xaip:packageInfoUnit>		
</xaip:versionManifest>		
<CanonicalizationMethod xmlns="http://www.w3.org/2000/09/xmldsig#" Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"></CanonicalizationMethod>		
</xaip:packageHeader>		
<xaip:dataObjectsSection>		
<xaip:dataObject dataObjectID="Data1">		
<xaip:binaryData MimeType="application/pdf">JVBE...==		
</xaip:binaryData>		
</xaip:dataObject>		
</xaip:dataObjectsSection>		
<xaip:credentialsSection>		
<xaip:credential credentialID="Signatur1" relatedObjects="Data1">		
<dss:SignatureObject>		
<dss:Base64Signature Type="urn:ietf:rfc:3852">MIAG...==		
</dss:Base64Signature>		
</dss:SignatureObject>		
</xaip:credential>		
<xaip:credential credentialID="cid_ba0a0f11-2ec2-4341-a8ad-f4d0860711a9"		
relatedObjects="test_packageID">		
<xaip:other>		
<sd:vrInfo xmlns:sd="http://ts.fujitsu.com/secdocs/v3_0/xaiparchiving">PD94...		
</sd:vrInfo>		
</xaip:other>		
</xaip:credential>		
<xaip:credential credentialID="cid_50151c29-d488-4602-a0a8-d1c18209e180"		
relatedObjects="Data1 cid_ba0a0f11-2ec2-4341-a8ad-f4d0860711a9">		
<xaip:other>		
<sd:evidenceRecord		
xmlns:sd=http://ts.fujitsu.com/secdocs/v3_0/xaiparchiving tsp="REG_TESTS_TSP">		
MIIVY...==		
</sd:evidenceRecord>		
</xaip:other>		
</xaip:credential>		
</xaip:credentialsSection>		
</xaip:XAIP>		

(1) An ArchiveUpdate übergebenes versionManifest

(2) Von ArchiveUpdate hinzugefügt

5 Fachwörter

Fachwörter, die an anderer Stelle erklärt werden, sind mit ->*kursiver* Schrift ausgezeichnet.

AOID (Archive Object Identifier)

Ein eindeutiger Identifikator für ein gespeichertes Submission Data Object (->*SDO*). Mit Hilfe der AOID kann die Anwendung Operationen bzgl. eines SDOs ausführen, z.B. lesen oder löschen.

ArchiSafe beschreibt den Mindestumfang der Archivierungs-Funktionen in der ->*TR-03125*.
<http://www.commoncriteriaportal.org/files/ppfiles/pp0049b.pdf>

ArchiSig Verbundprojekt, das vom Bundesministerium für Wirtschaft und Technologie (BMWi) im Rahmen des Programms „VERNET - Sichere und verlässliche Transaktionen in offenen Kommunikationsnetzen“ gefördert wird. ArchiSig befasst sich mit der beweiskräftigen und sicheren Langzeitspeicherung elektronisch signierter Dokumente. Im Rahmen von ArchiSig wurden aus den allgemeinen gesetzlichen Regelungen konkrete rechtliche Anforderungen abgeleitet. Diese Anforderungen an Systeme zur langfristigen Aufbewahrung elektronisch signierter Dokumente wurden prototypisch implementiert. Es konnte erstmalig gezeigt werden, dass die Langzeitaufbewahrung elektronisch signierter Dokumente gesetzeskonform, performant und akzeptabel umgesetzt werden kann.

Authentizität

kennzeichnet die Echtheit eines Unterzeichners. SecDocs verwendet Qualifizierte ->*Elektronische Signaturen* zum Nachweis der Authentizität eines elektronischen Dokumentes.

Client Anwendung, die die Dienste eines ->*Web-Service* anfordert und nutzt. Der Begriff ist synonym zum „Service Consumer“ beim ->*SOA-Konzept*.

COID (Client Object Identifier)

Ein vom Client vergebbarer Bezeichner zur Identifizierung eines Archivobjekts. Die COID kann bei den Operationen des *Archiving Webservice* statt einer ->*AOID* angegeben werden. Sie kann auch dazu verwendet werden, mehrere Versionen eines Dokuments über eine gemeinsamen COID zu verwalten.

Digitale Signatur

Klasse von kryptografischen (d.h. mathematischen) Verfahren.siehe auch ->*elektronische Signatur*

Elektronische Archivierung

unveränderbare, langzeitige Aufbewahrung elektronischer Information.

Elektronische Signatur

mit elektronischen Informationen verknüpfte Daten, mit denen der Unterzeichner (Signaturersteller) identifiziert und die Integrität der signierten elektronischen Informationen geprüft werden kann. In der Regel handelt es sich bei den elektronischen Informationen um elektronische Dokumente. Die elektronische Signatur erfüllt technisch gesehen den gleichen Zweck wie eine eigenhändige Unterschrift auf Papierdokumenten.

Die elektronische Signatur ist ein rechtlicher Begriff, im Gegensatz zur ->*Digitalen Signatur*, die eine Klasse von kryptografischen Verfahren bezeichnet.

Evidence Record

liefert den Beweis für die Integrität eines Dokuments, das in einem Langzeitarchiv gespeichert ist. Der Evidence Record muss eine nahtlose Kette von gültigen ->*Zeitstempeln* enthalten, rückwirkend bis zum Zeitpunkt der Übernahme des Dokuments in das Archivsystem (gemäß IETF RFC 4998).

Externes Datenobjekt

Zu archivierendes Datenobjekt, das unabhängig vom SOAP-Request auf den Server übertragen wird. Datenübertragung und submitSDO-/retrieveSDO-Request erfolgen zu verschiedenen Zeitpunkten: Bei der Archivierung erfolgt zuerst die Übertragung der externen Datenobjekte vom client-lokalen Rechner auf den SecDocs-Server und dann zu einem späteren Zeitpunkt der submitSDO- Request. Analog dazu wird beim Lesen von Dokumenten erst der retrieveSDO-Request abgesetzt und danach erst die Datenübertragung vom Server auf den lokalen Rechner des Client durchgeführt.

Externe Referenz-ID

Eindeutige Referenz auf ein externes Datenobjekt im SDO.
Wird ein Dokument als externes Datenobjekt archiviert, so steht im SDO nur noch eine Referenz auf dieses externe Dokument.

Freigabezeitpunkt

Datum und Uhrzeit, ab der ein ->SDO gelöscht werden kann.

Gegenseitige Authentifizierung

In einer Netzwerkumgebung müssen alle Kommunikationspartner ihre Identität nachweisen, bevor untereinander vertrauliche Daten ausgetauscht werden.
Auf diese Art können Client und Server sicher gehen, dass sie es mit legitimen Partnern zu tun haben.

Hash-Algorithmus

(auch Digest-Algorithmus oder Fingerprint-Algorithmus)
Identifikationsmerkmal für einen Datenbereich beliebiger Größe mit einer extrem geringen Wahrscheinlichkeit, dass zwei unterschiedliche Datenbereiche das gleiche Identifikationsmerkmal erhalten. Umgekehrt kann aus dem Identifikationsmerkmal nicht auf den Inhalt der Daten geschlossen werden. Es ist auch nahezu unmöglich, ein Dokument zu erzeugen oder zu ändern, das das gleiche Identifikationsmerkmal bekommt wie ein bestimmtes anderes Dokument.
Ein Hash-Algorithmus ist eine mathematische Einweg-Funktion, die zu einem großen Eingangswert (z.B. einer 700 GB = 700.000.000.000 Byte/Oktett großen Datei) einen kleinen Ausgangswert (16 Byte/Oktett) berechnet.
Im Umfeld der Qualifizierten Elektronischen Signatur gemäß SigV erstellt das BSI regelmäßig im Auftrag Bundesnetzagentur eine Bewertung zur zeitlichen Verwendung bestimmter Hash-Algorithmen für die Signaturerzeugung und Signaturprüfung.

Hash-Baum

Datenstruktur, die die nachprüfbare Zusammenfassung einer beliebigen Anzahl von ->Hash-Werten zu einem einzigen Hash-Wert (Wurzel-Hash-Wert) ermöglicht (1979, Ralph Merkle).

Hash-Wert

(auch Digest, Fingerprint oder Fingerabdruck)
Ergebnis (Ausgangswert) der Anwendung eines ->Hash-Algorithmus auf einen Eingangswert.

HDO

(Hash Data Object)
besteht aus der Konkatenation von ->SignatureVerificationInfo und ->SDO (in dieser Reihenfolge).

HTTPS (HyperText Transfer Protocol Secure)

Kommunikationsprotokoll im World Wide Web, um Vertraulichkeit und Integrität in der Kommunikation zwischen Webserver und Webbrowser herzustellen.
Dies wird u.a. durch Verschlüsselung und Authentifizierung erreicht.

Langzeitarchivierung

Erfassung, langfristige Aufbewahrung und Erhaltung der dauerhaften Verfügbarkeit von Informationen.

Mandantenfähigkeit

Technik, die auf demselben Server oder demselben Software-System mehrere Mandanten (Kunden, Auftraggeber oder Organisationseinheiten) verwalten kann. Das Schriftgut der verschiedenen Mandanten wird strikt voneinander getrennt gehalten. Ein Mandant kann nicht auf die Daten, Dokumente oder Parameter eines anderen Mandanten zugreifen.

Multi-Node-Betrieb, Multi-Node-Konfiguration

Parallelbetrieb mehrerer SecDocs-Instanzen für ein Archiv. Vgl. -> *Single-Node-Betrieb, Single-Node-Konfiguration*

Primärdokumente

Dokumente (Dateien), die BASE64-codiert im ->SDO eingebettet sind.

SDO

(Submission Data Object)

->XML-Container, in dem ein zu archivierendes Datenobjekt an SecDocs übergeben wird.

SDO-Typ

Ein SDO-Typ beschreibt das zu archivierende ->SDO. Er besteht aus

- den Informationen über die Struktur des zu archivierenden Schriftguts. Diese Struktur wird in einem ->XML-Schema festgelegt.
- den Informationen zur Adressierung für (Nutz-)Daten, Signaturen und Metadaten, die in der zugehörigen Filterdefinition abgelegt sind.

Signatur-Algorithmus

Verschlüsselungs-Algorithmus, der zum Erzeugen ->elektronischer Signaturen verwendet wird.

SFTP-Server

Zu archivierende externe Datenobjekte werden unabhängig vom SOAP-Request per Secure File-Transfer-Protokoll (SFTP) auf den SecDocs-Server übertragen. Zu diesem Zweck ist ein integrierter SFTP-Server als fester Bestandteil des SecDocs-Servers installiert.

Signatur-Container

Datencontainer in einem der Formate PKCS#7, CMS oder XMLDSig, der eine oder mehrere Signaturen enthalten kann. Ein Signatur-Container kann in einem Primärdokument enthalten sein, oder als abgesetzte Signatur zu einem Primärdokument vorliegen.
Ein Primärdokument kann höchstens einen Signatur-Container enthalten, jedoch können mehrere Signatur-Container als abgesetzte Signaturen zu einem Primärdokument vorliegen.

Signature Verification Information

Die zum Zeitpunkt der Archivierung eines SDOs verwendeten vollständigen Zertifikatsketten und Sperrinformationen (Sperrlisten und OCSP-Antworten) werden in der Signature Verification Information archiviert und zusammen mit dem SDO durch einen Evidence Record abgesichert. Mit diesen Informationen – vollständige Zertifikatsketten und Sperrinformationen – ist es möglich, die zum Zeitpunkt der Archivierung durchgeführte Verifikation zu einem beliebigen Zeitpunkt später noch einmal durchzuführen. Darüber hinaus enthält die Signature Verification Information das Protokoll der zum Archivierungszeitpunkt durchgeführten Verifikation. Dieses Verifikationsprotokoll zeigt, welche Signaturen geprüft wurden, mit welchen Zertifikaten und Sperrinformationen die Verifikation durchgeführt wurde und zu welchem Resultat das Verifikationsmodul gelangte. Detaillierte Informationen zum Prüfprotokoll können beim Hersteller OpenLimit angefordert werden.

Single-Node-Betrieb, Single-Node-Konfiguration

Eine einzige SecDocs-Instanz bedient alle Schnittstellen für ein Archiv.
Vgl. ->Multi-Node-Betrieb, Multi-Node-Konfiguration

SOA

(Service-Oriented Architecture).

Eine SOA ist ein Konzept für eine Systemarchitektur, in dem Funktionen in Form von wieder verwendbaren, technisch voneinander unabhängigen und fachlich lose gekoppelten *Services* implementiert werden. Services können unabhängig von zugrunde liegenden Implementierungen über Schnittstellen aufgerufen werden, deren Spezifikationen öffentlich und damit vertrauenswürdig sein können. Service-Interaktion findet über eine dafür vorgesehene Kommunikationsinfrastruktur statt.

SOAP

(Simple Object Access Protocol)

Protokoll, mit dessen Hilfe Daten zwischen Systemen ausgetauscht und Remote Procedure Calls durchgeführt werden können. SOAP stützt sich auf die Dienste anderer Standards, ->XML zur Repräsentation der Daten und Internet-Protokolle der Transport- und Anwendungsschicht zur Übertragung der Nachrichten.

Timestamp

siehe ->Zeitstempel.

TR-03125

(Technische Richtlinie „Vertrauenswürdige elektronische Langzeitspeicherung“)

Diese vom Bundesamt für Sicherheit in der Informationstechnik veröffentlichte Richtlinie beschreibt ein Architekturmodell zur vertrauenswürdigen elektronischen Langzeitspeicherung unter Verwendung von ArchiSig, ArchiSafe und Krypto-Modul. Die TR-03125 ist die Entwicklungsbasis von SecDocs.

TSP

(Timestamp Provider)

siehe ->Zeitstempelanbieter.

Web-Service

Anwendung, die auf einem Web-Server läuft und über eine standardisierte und programmatische Schnittstelle (öffentlich) verfügbar ist. Die Anwendung kann über das ->SOAP-Protokoll angesprochen werden. Die Schnittstelle eines Web-Service ist in ->WSDL beschrieben.

WSDL

(Web Services Description Language)

bietet ->XML-Sprachregeln für die Beschreibung von ->Web-Services. Ein Web-Service wird dabei durch eine Auswahl von Ports definiert.

XAIP (XML Formatted **A**rchival **I**nformation **P**ackage)
selbstbeschreibendes und wohlgeformtes XML-Dokument, das gegen ein gültiges und autorisiertes XML-Schema geprüft werden kann. Ein solches Archivdatenobjekt enthält sämtliche Inhaltsdaten (Primärinformationen) und Metainformationen, die für eine zuverlässige und vollständige Rekonstruktion von Geschäfts- oder Verwaltungsvorgängen bis zum Ablauf der gesetzlich vorgeschriebenen Aufbewahrungsfristen erforderlich sind.

XML (e**X**tensible **M**arkup **L**anguage)
definiert eine Sprache zur logischen Strukturierung von Dokumenten mit dem Ziel, diese einfach zwischen verschiedenen Anwendungen auszutauschen.

XML-Schema

definiert die zulässigen Elemente und Attribute einer ->XML-Beschreibung.
XML-Schemas können verschiedene Formate haben, z.B. DTD (**D**ocument **T**ype **D**efinition), XML Schema (W3C-Standard) oder XDR (**X**ML **D**ata **R**educed).

XPath (**X**ML **P**ath Language)
Abfragesprache, mit der Teile eines ->XML-Dokuments adressiert werden können. XPath wurde vom W3-Konsortium entwickelt und ist die Grundlage weiterer Standards wie XSLT, XPointer und XQuery.

Zeitstempel

Bestätigung der Vorlage eines Dokuments zu einem bestimmten Zeitpunkt durch den -> *Zeitstempelanbieter*. Der Zeitstempel bezieht sich auf die Existenz des Dokuments und nicht auf dessen Inhalt.

Zeitstempelanbieter

(auch Zeitstempel-Dienst)
nimmt als Server im Internet/Intranet signierte Dateien oder auch nur deren Signaturen entgegen und versieht diese mit einem ->*Zeitstempel*. Ein Zeitstempel-Dienst kann sowohl in einem internen Netz als auch im Internet angeboten und genutzt werden.

Es wird empfohlen, nur Zeitstempelanbieter zu verwenden, die offiziell durch Ämter akkreditiert sind. In Deutschland durch die Bundesnetzagentur. Jeder Zeitstempelanbieter ist auch ein ->*Zertifizierungsdiensteanbieter*.

Zertifikat Ein digitales Zertifikat ist ein digitaler Datensatz, der bestimmte Eigenschaften von Personen oder Objekten bestätigt und dessen Authentizität und Integrität durch kryptografische Verfahren geprüft werden kann. Das digitale Zertifikat enthält insbesondere die zu seiner Prüfung erforderlichen Daten.

Zertifizierungsdiensteanbieter

natürliche oder juristische Personen, die qualifizierte Zertifikate und Zeitstempel ausstellen. Zertifizierungsdiensteanbieter können sich freiwillig bei der Bundesnetzagentur gemäß §15 Abs. 1 SigG und §11 SigV akkreditieren lassen. Eine Akkreditierung dient als Nachweis für die technische und administrative Sicherheit von qualifizierten Signaturen und wird nach einer Überprüfung des Zertifizierungsdiensteanbieters durch die Bundesnetzagentur ausgesprochen. Akkreditierte Zertifizierungsdiensteanbieter müssen die von ihnen ausgestellten Zertifikate noch 30 Jahre nach Ablauf des Gültigkeitszeitraumes über einen Verzeichnisdienst abrufbar und nachprüfbar halten.

6 Abkürzungen

ACM_PP	Protection Profile for an ArchiSafe Compliant Middleware for Enabling the Long-Term Preservation of Electronic Documents
ADO	Archivdatenobjekt
AOID	Archive Object Identifier, Archivobjekt-ID
BMWi	Bundesministerium für Wirtschaft und Technologie
BPM	Business Process Management
BSI	Bundesamt für Sicherheit in der Informationstechnik
CRL	Certificate Revocation List
COID	Client Object Identifier
DMS	Dokumenten-Management-System
ERP	Enterprise Resource Planning
FTP	File Transfer Protocol
HDO	Hash Data Object
HTTPS	HyperText Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
OID	Object Identifier
OCSP	Online Certificate Status Protocol
PKI	Public Key Infrastructure
RFC	Request for Comments
PKCS	Public Key Cryptography Standards
SDO	Submission Data Object
SFTP	Secure File Transfer Protocol
SHA	Secure Hash Algorithm
SigG	Signaturgesetz
SigV	Signaturverordnung
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SSH	Secure Shell

SVI	Signature Verification Information
TCP/IP	Transmission Control Protocol / Internet Protocol
TLS	Transport Layer Security
TR	Technische Richtlinie
TSP	Zeitstempelanbieter, Timestamp Provider
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTC	Coordinated Universal Time
WSDL	Web Services Description Language
XAIP	XML Formatted Archive Information Package
XML	Extended Markup Language
XPath	XML Path Language
ZDA	Zertifizierungsdiensteanbieter

7 Literatur

Die Handbücher sind online unter <http://manuals.ts.fujitsu.com> und auf der mitgelieferten DVD zu finden. Eine Zuordnung der Handbuchtitel zu den entsprechenden Dateinamen entnehmen Sie bitte der Freigabemitteilung.

- [SD1] **SecDocs Administration und Bedienung**
Benutzerhandbuch
- [SD2] **SecDocs Archivierung von Dokumenten gemäß Richtlinie TR-ESOR**
Benutzerhandbuch
- [SD3] **SecDocs V3.1 Installationsanleitung**
Referenzhandbuch
- [SD4] **SecDocs-Rückgabewerte**
Referenzhandbuch
- [SD5] **CryptoModule Runtime Umgebung für Fujitsu SecDocs V3.1**
Benutzerhandbuch
- [SD6] **MigSafe / OverSign Rückgabewerte**
Benutzerhandbuch

Literatur zur elektronischen Langzeitarchivierung im Internet

- [W1] **BSI Technische Richtlinie 03125:**
Vertrauenswürdige elektronische Langzeitspeicherung
Bundesamt für Sicherheit in der Informationstechnologie (BSI)
https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr03125/TR-03125_node.html

Unter dieser Adresse finden Sie Verweise auf die nachfolgend genannten Dokumente der BSI TR-03125 Version 1.2:

- [W2] **BSI TR-03125**
Vertrauenswürdige elektronische Langzeitspeicherung Version 1.2.1
https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03125/BSI_TR_03125_V1_2_1.pdf?__blob=publicationFile&v=2

-
- [W3] BSI TR-03125 Anlage M.1
ArchiSafe Modul Version 1.2.1
https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03125/BSI_TR_03125_Anlage_M1_V1_2_1.pdf?__blob=publicationFile&v=1
- [W4] BSI TR-03125 Anlage TR-ESOR-S
Schnittstellenspezifikation Version 1.2.1
https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03125/BSI_TR_03125_Anlage_S_V1_2_1.pdf?__blob=publicationFile&v=1
- [W5] BSI TR-03125 Anlage TR-ESOR-F
Formate Version 1.2.1
https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03125/BSI_TR_03125_Anlage_F_V1_2_1.pdf?__blob=publicationFile&v=1
- [W6] **Common Criteria Protection Profile for an ArchiSafe Compliant Middleware for Enabling the Long-Term Preservation of Electronic Documents**
<http://www.commoncriteriaportal.org/files/ppfiles/pp0049a.pdf>

Verwendete Standards

- [S1] **Data elements and interchange formats – Information interchange – Representation of dates and times, ISO8601:**
http://de.wikipedia.org/wiki/ISO_8601
- [S2] **Date and Time on the Internet: Timestamps, RFC 3339**
<http://tools.ietf.org/html/rfc3339>
- [S3] **Evidence Record Syntax (ERS), RFC 4998**
<http://tools.ietf.org/html/rfc4998>
- [S4] **Internet X.509 Public Key Infrastructure , Time-Stamp Protocol (TSP), RFC 3161**
<http://tools.ietf.org/html/rfc3161>
- [S5] **Java™ Platform, Enterprise Edition (Java EE) Specification, v6**
<http://www.oracle.com/technetwork/java/javasee/tech/javasee6technologies-1955512.html>
- [S7] **Simple Object Access Protocol (SOAP) 1.1**
<http://www.w3.org/TR/soap/>
- [S9] **The Syslog Protocol, RFC 5424**
<http://tools.ietf.org/html/rfc5424>

[S10] **Web Services Description Language (WSDL) 1.1**

<http://www.w3.org/TR/wsdl.html>