Bundesamt
für Sicherheit in der
Informationstechnik

Technische Richtlinie eID-Server

Kürzel: **BSI TR-03130**

Version: 1.5

Datum: 06.12.2011

Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63
53133 Bonn
Tel.: +49 22899 9582-100
E- Mail: epa@bsi.bund.de
Internet: https://www.bsi.bund.de

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Am 1. November 2010 wurde der neue elektronische Personalausweis (nPA) mit dem elektronischen Identitätsnachweis (eID-Funktion) eingeführt und auch der elektronische Aufenthaltstitel (eAT) wurde, zur Einführung am 1. September 2011, mit dieser Funktion ausgestattet. Im Folgenden werden hoheitliche Dokumente mit eID-Funktion unter dem Begriff eID-Dokument zusammengefasst. Um die Nutzung der eID-Funktion in Web-Anwendungen zu vereinfachen, soll ein eID-Server realisiert werden. Der eID-Server stellt eine funktionale eID-Schnittstelle für Web-Anwendungen bereit, um die Komplexität der eID-Funktion zu kapseln.

Diese Richtlinie spezifiziert die Schnittstelle, die durch Web-Anwendungen genutzt wird und die entsprechenden Datenformate für den Austausch der Informationen. Des Weiteren werden Schnittstellen und Eigenschaften des eID-Servers beschrieben, die durch die funktionale eID-Schnittstelle beeinflusst werden.

Dieses Dokument gliedert sich in drei Teile. Im ersten Teil wird der eID-Server allgemein beschrieben und die Komponenten, mit denen der eID-Server kommuniziert, werden dargestellt. Außerdem wird der Betrieb eines eID-Servers im Kontext eines eID-Service herausgearbeitet. Im zweiten Abschnitt werden die getroffenen Entwurfsentscheidungen aufgeführt und explizit dargestellt. Im dritten Teil werden die eID-Schnittstelle und die darin verwendeten Datentypen beschrieben. Die eID-Schnittstelle stellt die Daten und Funktionen eines eID-Dokuments für eine Web-Anwendung in Form eines Web Services bereit.

An die Technische Richtlinie eID-Server schließen sich insgesamt drei Anhänge an. Der erste Anhang (*Anhang A: Verwendung mit SAML*) beschreibt eine SAML-Schnittstelle, die optional vom eID-Service bedient werden kann. Diese Schnittstelle ist insbesondere für den internationalen Zusammenhang und im Kontext eines Identity Providers vorgesehen. Beim *Anhang B: Schemadateien* handelt es sich lediglich um die formale Bezeichnung für die zur Technischen Richtlinie gehörenden Schemadateien. Anforderungen an den Betrieb von eID-Servern werden im *Anhang C: Anforderungen an den Betrieb von eID-Servern* beschrieben und sollen insbesondere eine Hilfestellung für Diensteanbieter zur Erstellung eines Sicherheitskonzeptes sein.

1.1 Historie

Um denjenigen, die bereits mit einer älteren Version dieser Technischen Richtlinie gearbeitet haben, den Einstieg in die hier vorliegende Version zu erleichtern, werden an dieser Stelle stichwortartig die vorgenommenen Änderungen in einer Historie zusammengefasst.

Version	***Datum***	***Änderungen***
1.0 RC1	19.05.2009	Initiale Version zur Kommentierung
1.0 RC2	13.07.2009	Überarbeitung der *Kapitel 2.4*, *2.5* und *2.6* zum eID-Service und Identity Provider; Harmonisierung mit weiteren Technischen Richtlinien; Kleinere Korrekturen und Präzisierungen; Überarbeitung der Funktionsaufrufe an der eID-Schnittstelle zur Reduzierung von Redundanzen und möglichen Fehlerquellen

Version	*Datum*	*Änderungen*
1.0	14.07.2009	Entspricht inhaltlich der Version 1.0 RC2
1.0	21.10.2009	Rechtschreibkorrekturen und Anpassungen zur Barrierefreiheit
1.0	09.01.2010	Fehlerkorrektur der Schemadatei zu Version 1.0.1
1.1	08.02.2010	Harmonisierung mit weiteren Technischen Richtlinien (direktes Auslesen der Wohnort-ID erlaubt; viele kleinere Korrekturen; Einarbeitung des Datentyp `GeneralPlaceType`
1.2	01.04.2010	Direktes Auslesen der Wohnort-ID entfernt; Postleitzahl in Datentyp `PlaceType` eingearbeitet
1.3	10.06.2010	Präzisierung der Kommunikation bei Verwendung von SAML; Möglichkeit für die Web-Anwendung den `PSK` bereits beim Aufruf der eID-Schnittstelle zu übergeben
1.4	14.09.2010	Berücksichtigung der Rückmeldungen aus dem Anwendungstest; Feinspezifikation der SAML-Schnittstelle in *Anhang A*; Integration *Anhang C*
1.4	14.09.2010	Element `DateOfExpiry` wurde kurzfristig aus Beispielnachrichten im *Kapitel 4.6* entfernt, da es unzulässig ist
1.4.1	08.10.2010	Ergänzende Verweise auf [SAML Security] in *Anhang A*; Kleinere Korrekturen; Änderungen in *Anhang C*
1.5 RC	07.10.2011	Kleinere Korrekturen, Einführung des Elements `<OneTimeUse>` und Attributs `RequiredAttribute` an der SAML-Schnittstelle, Entfernung der `Nationality`, Hinweise zu Sonderfällen (z.B. Hinauszögern der `getResultResponse`) ergänzt, Berücksichtigung des eAT, Entfernung des `TransportBinding` aus der WSDL, Ergänzung eines Fehlercodes für Schemaverletzungen
1.5 RC2	11.11.2011	Korrektur der *Abbildung 29: Protokollablauf*, Konkretisierung der Spezifikation zur Einbindung von Referenzen in der WSDL durch das Element `<sp:RequireIssuerSerialReference />`, Entfernung überflüssiger Teile der WSDL
1.5	06.12.2011	Entspricht inhaltlich der Version 1.5 RC2

Tabelle 1: Historie

1.2 Abgrenzung der Technischen Richtlinie

Diese Technische Richtlinie beschreibt keine organisatorischen Abläufe im Betrieb des eID-Servers und keine Anforderungen an die allgemeine Administration. Die Funktionen des Änderungsdienstes werden in dieser Richtlinie ebenfalls nicht betrachtet, d.h. es wird nur lesend auf die Daten und

Vergleichsfunktionen der eID-Funktion zugegriffen. Vorgaben zur Integration des eID-Servers in ein Identity Management sind in dieser Technischen Richtlinie ausschließlich im SAML-Kontext und nicht abschließend beschrieben. Beispielhafte Integrationsmöglichkeiten werden jedoch in *Kapitel 2.4* beschrieben. Des Weiteren werden die Komponenten, die zur Realisierung des eID-Servers verwendet werden, in separaten Richtlinien beschrieben:

- Die Kommunikation des eID-Servers mit dem eID-Dokument wird durch das eCard-API-Framework realisiert, welches in [eCard-API] spezifiziert ist.
- Die Kommunikation mit den Komponenten der PKI wird in [EAC-PKI Protocol] spezifiziert, sowie organisatorisch in [EAC-PKI'n ePA] beschrieben.

2 Funktionsumfang

In diesem Abschnitt werden die Dienste beschrieben, die der eID-Server anbietet und die Anforderungen, die an seine Einsatzumgebung gestellt werden. Außerdem beschreibt dieser Abschnitt die Komponenten, die im Kontext des eID-Servers relevant sind.

2.1 Beschreibung

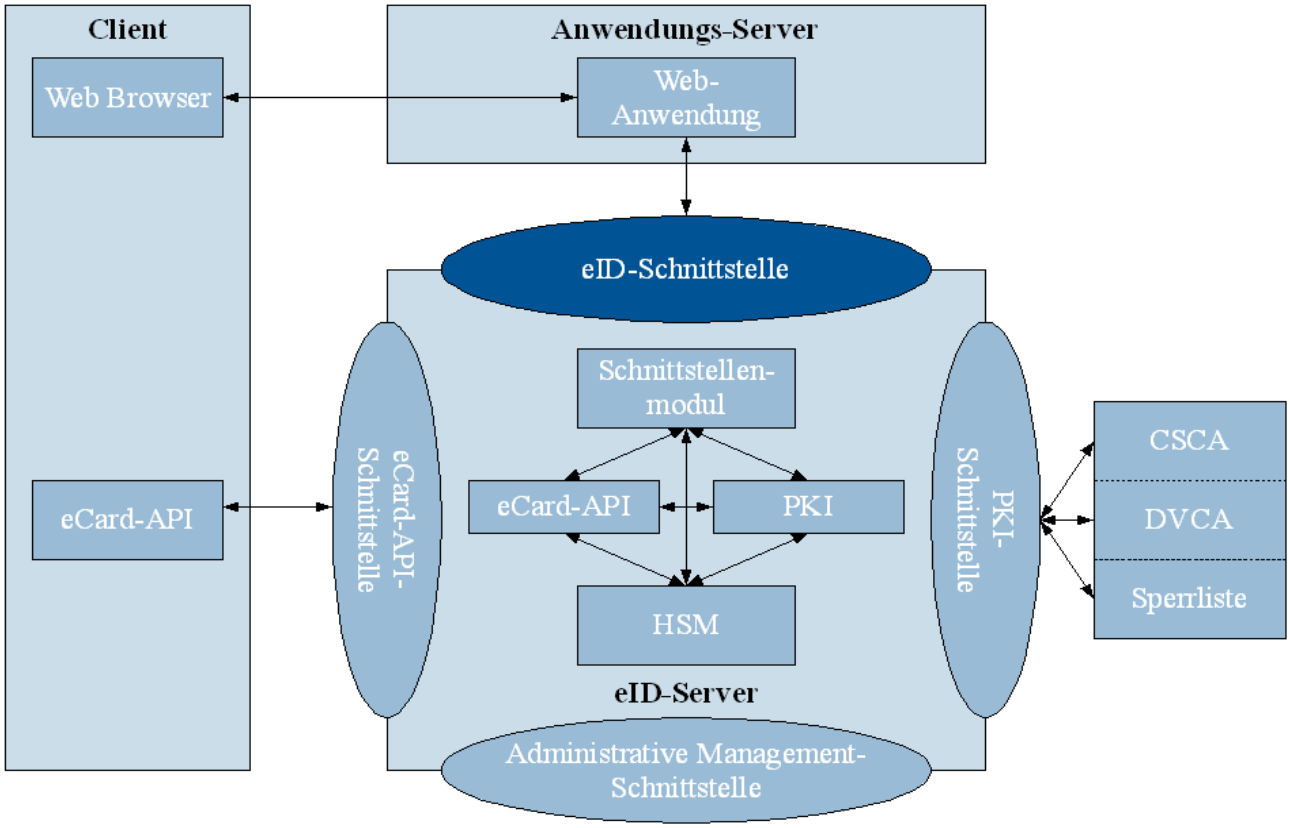

Abbildung 1: Kontext und Schnittstellen des eID-Servers

Der eID-Server als Hardware- und Softwarekomponente stellt die Kommunikation zur clientseitigen eCard-API her und übernimmt die Kommunikation zum Abruf von Terminal-Berechtigungszertifikaten (DVCA-Zertifikate), Sperrlisten und CSCA-Zertifikaten (siehe *Abbildung 1*).

Der eID-Server wird als logisch eigenständiger Server realisiert, so dass er von mehreren Web-Anwendungen (Mandanten) genutzt und auch entfernt, z.B. bei einem Dritten betrieben werden kann. Zur Wahrung der Vertraulichkeit und Integrität der verarbeiteten Daten werden die Daten bei der Übertragung zwischen eID-Server und Anwendungsserver verschlüsselt und signiert, wenn sie über ein offenes Netz übertragen werden.

Diese Technische Richtlinie spezifiziert die Außenkommunikation des eID-Servers mit der Web-Anwendung und beschreibt funktional die Schnittstelle des eID-Servers zur Administration dieser Kommunikation (siehe *Kapitel 2.2.3*). Die vorrangige Aufgabe der eID-Schnittstelle ist die

gegenseitige Authentisierung und die Bereitstellung von Daten aus dem eID-Dokument des Anwenders für die Web-Anwendung.

2.2 Schnittstellen

Der eID-Server kommuniziert über die drei Schnittstellen „eCard-API-Schnittstelle“, „PKI-Schnittstelle“ und „Administrative Management-Schnittstelle“, um die Leistung bereitzustellen, die er an der eID-Schnittstelle anbietet. Die direkten Schnittstellen des eID-Servers basieren auf Web Services und können daher grundsätzlich auch über ein offenes Netz angesprochen werden.

2.2.1 eCard-API

Die eCard-API besteht aus zwei Software-Komponenten, dem Client und dem Server. Die eCard-API wird in [eCard-API] beschrieben. Der Client, zum Beispiel die AusweisApp, wird auf dem PC des Anwenders ausgeführt, verwaltet die angeschlossenen Kartenlesegeräte und bietet dem Benutzer während der eID-Nutzung eine grafische Schnittstelle an. Durch diese grafische Schnittstelle kann der Benutzer auswählen, welche Datengruppen aus seinem eID-Dokument ausgelesen und auf welchen Datengruppen Operationen (z.B. Altersverifikation) ausgeführt werden dürfen. Die Auswahl des Benutzers wird der Web-Anwendung durch den eID-Server übergeben.

Der eCard-API Client reagiert auf Anfragen, die er durch den Browser des Benutzers erhält, und verbindet sich daraufhin mit dem eCard-API Server. Dazu baut der Client eine Verbindung zur eCard-API Schnittstelle des eID-Servers auf. Diese Verbindung nutzt der Server, um die Daten aus dem eID-Dokument auszulesen. Der eID-Server verwendet den eCard-API Server um mit dem eID-Dokument zu kommunizieren und die benötigten Funktionen aufzurufen.

2.2.2 PKI

Bevor Daten aus dem eID-Dokument ausgelesen werden können, muss die lesende Anwendung nachweisen, dass sie berechtigt ist, die Daten auszulesen. Diese Berechtigungen werden durch eine Public Key Infrastructure (PKI) abgebildet, die in [EAC 2] beschrieben wird. In dieser PKI betreibt jede Nation eine eigene Country Verifying CA (CVCA). Unterhalb der CVCA wird eine oder mehrere Document Verifier CA (DVCA) betrieben, die Terminal-Berechtigungszertifikate ausstellt. Eine DVCA wird aus diesem Grund auch als Berechtigungs-CA bezeichnet. Mit Hilfe des Terminal-Berechtigungszertifikats kann sich ein Diensteanbieter (eine Anwendung) gegenüber einem eID-Dokument authentisieren und nachweisen, dass er berechtigt ist, Daten auszulesen. Ein Terminal-Berechtigungszertifikat enthält dabei genaue Informationen darüber, welche Informationen aus der Karte ausgelesen werden dürfen. Der eID-Server verwaltet die Terminal-Berechtigungszertifikate und bestellt automatisiert entsprechend [EAC-PKI Protocol] und [EAC-PKI'n ePA] neue Terminal-Berechtigungszertifikate bei der Berechtigungs-CA. Die in diesem Zusammenhang genutzten Terminal-Berechtigungszertifikate erfordern die Erfüllung aller Anforderungen der Certificate Policy [CP_CVCA-eID]. Der automatisierte Abruf der Terminal-Berechtigungszertifikate kann mit Hilfe von Authentisierungszertifikaten (siehe *Kapitel 3.5*) abgesichert werden.

Um die Echtheit eines eID-Dokuments elektronisch zu prüfen, sind die einzelne Daten durch ein Zertifikat der Document Signer CA (DSCA) signiert. Diese DSCA wird analog zur DVCA/CVCA unterhalb einer Country Signing CA (CSCA) betrieben. Der eID-Server verwendet die Zertifikate

der PKI um die Echtheit eines eID-Dokuments durch die passive Authentisierung zu prüfen, während er die Daten für die Web-Anwendung ausliest.

Um abhanden gekommene eID-Dokumente zu deaktivieren, wird ein Sperrlistendienst betrieben, der diese als ungültig markiert. Der eID-Server verwendet den Sperrlistendienst, um diese eID-Dokumente zu erkennen und gibt im Falle eines kompromittierten eID-Dokuments eine Fehlermeldung (siehe *Kapitel 4.5.1*).

2.2.3 Administrative Management-Schnittstelle

Damit der eID-Server die spezifizierten Funktionen bereitstellen kann, benötigt er initiale Einstellungen und Schlüssel. Diese Parameter werden über die Management-Schnittstelle konfiguriert, welche in diesem Abschnitt funktional beschrieben wird. Wie diese Schnittstelle technisch organisatorisch umgesetzt ist, wird in dieser Technischen Richtlinie nicht spezifiziert.

Die Parameter werden in zwei Gruppen unterteilt: Globale und Mandanten spezifische Parameter. Diese Parameterliste ist nicht als abschließend zu betrachten, da eine eID-Server Implementierung weitere Konfigurationsparameter an dieser Schnittstelle anbieten kann.

Globale Parameter

Die folgenden Parameter können global definiert werden, wenn sie von allen Mandanten (Diensteanbietern) identisch verwendet werden.

Parameter	***Beschreibung***
eCard-API Server Adresse und Port	Die IP-Adresse/URL und der TCP-Port unter der die eCard-API des eID-Servers erreichbar ist. Alternativ kann anstelle dieser Konfiguration auch das Element `eCardServerAddress` der Funktion `useID` genutzt werden, um diese Adresse dynamisch zu wählen.
Sperrlistendienst Adresse und Port	Die IP-Adresse/URL und der TCP-Port unter der der Sperrlistendienst erreichbar ist.
DVCA Adresse und Port	Die IP-Adresse/URL und der TCP-Port unter der die DVCA erreichbar ist.
CSCA Adresse und Port	Die IP-Adresse/URL und der TCP-Port unter der die CSCA erreichbar ist.
Intervall der Sperrlistenaktualisierung	Die Zeitspanne, nach der der eID-Server die Sperrliste erneut abruft.

Tabelle 2: Globale Parameter

Mandanten spezifische Parameter

Für Parameter, die je nach Mandant unterschiedlich sind, muss der eID-Service Anbieter es dem Mandanten (Diensteanbieter) ermöglichen diese zu konfigurieren bzw. die Konfiguration in seinem Auftrag vornehmen.

Parameter	*Beschreibung*
Authentisierungszertifikat	Die Zugangsdaten oder das Zertifikat, mit dem der eID-Server im Namen des Diensteanbieters Terminal-Berechtigungszertifikate bei der DVCA abholt.

Tabelle 3: Mandanten spezifische Parameter

2.3 Anforderungen an die Einsatzumgebung

In diesem Abschnitt werden die Anforderungen des eID-Servers und der Web-Anwendung bezüglich ihrer Einsatzumgebung beschrieben. Dabei werden die technischen Voraussetzungen dargestellt, die zum Betrieb eines eID-Servers gegeben sein müssen. Der *Anhang C* dieser Technischen Richtlinie beschäftigt sich mit den sicherheitstechnischen Anforderungen, die an den Betrieb von eID-Servern gestellt werden.

2.3.1 eID-Server

Der eID-Server kommuniziert mit einer DVCA und dem Sperrlistendienst. Die verwendeten TCP-Ports sind abhängig von der Implementierung der DVCA und müssen für den eID-Server erreichbar sein.

Die serverseitige eCard-API muss von der clientseitigen eCard-API erreichbar sein. Der TCP-Port, auf dem der eCard-API Server auf eingehende Verbindungen des eCard-API Client wartet, ist von der Implementierung und Konfiguration der eCard-API abhängig.

Der eID-Server speichert und verwaltet die Authentisierungszertifikate im Auftrag der Diensteanbieter. Diese Authentisierungszertifikate müssen vor missbräuchlichem Zugriff geschützt werden.

2.3.2 Web-Anwendung

Die Web-Anwendung realisiert die Anwendungslogik der im Web-Browser dargestellten Anwendung. Die Web-Anwendung kann beliebige Funktionen realisieren. Für den Kontext dieser Richtlinie ist lediglich entscheidend, dass die Web-Anwendung die eID-Schnittstelle, um den elektronischen Identitätsnachweis des eID-Dokuments zu nutzen.

Daten, die durch die Web-Anwendung ausgelesen werden, sollen dem Benutzer angezeigt werden. Bei der Übertragung vom Anwendungs-Server zum Web-Browser muss die Vertraulichkeit der ausgelesenen Daten sichergestellt werden.

Die Web-Anwendung erhält für jeden Authentisierungsvorgang einen Pre-Shared Key (PSK) zur Bindung der Kommunikation der eCard-API und leitet diesen an die auf dem Client ausgeführte

Instanz der eCard-API weiter. Die Web-Anwendung muss die Vertraulichkeit und Integrität des PSK bei der Übertragung sicherstellen.

Alternativ kann der PSK auch von der Web-Anwendung generiert werden und dem eID-Server sowie der auf dem Client ausgeführten Instanz der eCard-API übermittelt werden.

Die Verbindung der Web-Anwendung mit dem Browser des Benutzers muss so gestaltet sein (z.B. durch Verwendung von TLS/SSL), dass sich die Authentizität des Anwendungs-Servers mit Hilfe der im Terminal-Berechtigungszertifikat gespeicherten Hashes überprüfen lässt. Beide Maßnahmen zur Kanalbindung (PSK und Hashes) werden technisch ausführlich in [eCard-API] *Teil 7 Abschnitt 2.3.2 und 3.3.10* beschrieben.

2.4 Beispielhafte Integrationsmöglichkeiten

Es gibt verschiedene Möglichkeiten einen eID-Server in eine Einsatzumgebung zu integrieren. Im Folgenden werden mehrere typische Szenarien dargestellt. Grundsätzlich bestehen die beiden Möglichkeiten den eID-Server lokal beim Diensteanbieter und entfernt bei einem eID-Service zu betreiben. Um zu gewährleisten, dass eID-Server und Web-Anwendung in einem geschlossenen Sicherheitskontext liegen, dürfen beim entfernten Betrieb des eID-Servers Daten nur verschlüsselt und signiert über offene Netze übertragen werden. Der Pre-Shared Key wird in allen Szenarien genutzt, um den Kommunikationskanal zwischen Browser und Webserver an den Kanal zwischen eCard-API Client und eID-Server zu binden. Es ist daher zwingend erforderlich, dass der PSK, unabhängig davon ob er beim eID-Server oder beim Webserver generiert wurde, über den Kanal zwischen Webserver und Browser an die clientseitige eCard-API übertragen wird.

2.4.1 Lokaler eID-Server beim Diensteanbieter

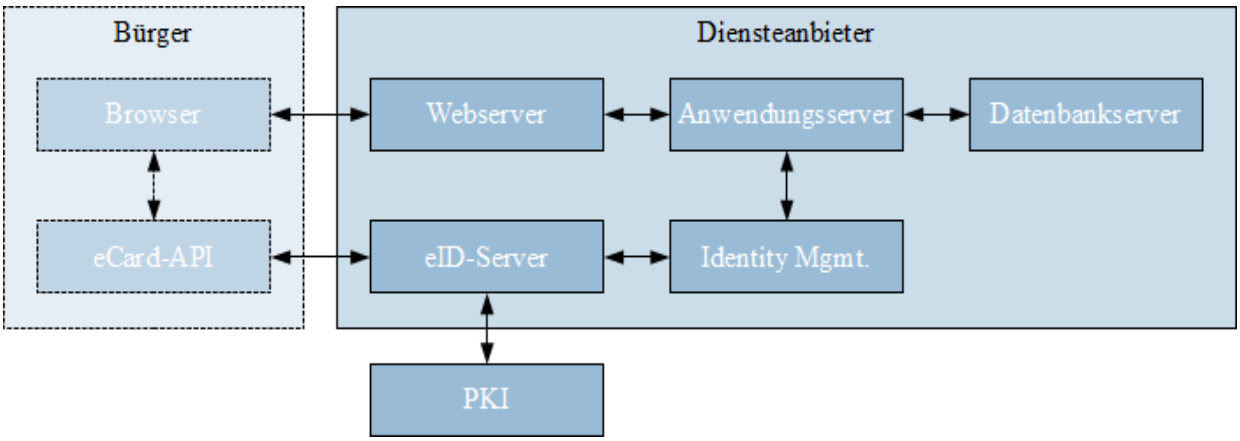

Abbildung 2: Integration in bestehendes Identity Management

In *Abbildung 2* ist eine Web-Anwendung dargestellt, die aus Webserver, Anwendungsserver, Datenbankserver und Identity Management besteht. Der eID-Server wird vom Identity Management als weitere Authentisierungsmethode verwendet. Der eID-Server wird von dem eCard-API Client über das Internet angesprochen, daher wird der eID-Server in diesem Szenario in einer Zone betrieben, die vor dem Identity Management liegt.

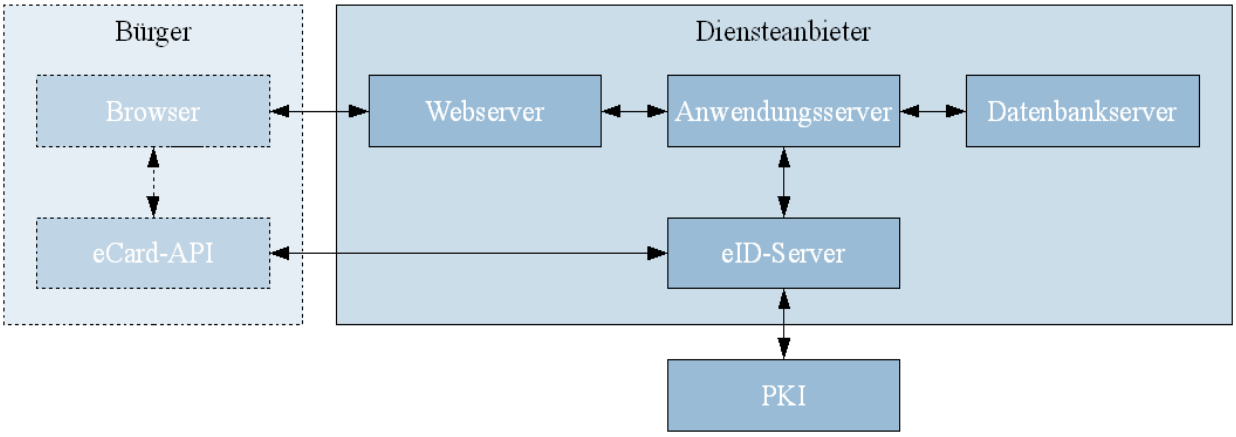

Abbildung 3: Integration ohne Identity Management

Der eID-Server kann auch ohne ein bestehendes Identity Management in eine Web-Anwendung integriert werden, dieses Szenario ist in *Abbildung 3* dargestellt. In diesem Szenario nutzt die Web-Anwendung direkt die Funktionen des eID-Dokuments durch die vom eID-Server angebotene Schnittstelle.

2.4.2 Entfernter eID-Server bei einem eID-Service Anbieter

Es ist auch möglich, dass ein Dienstleister das Auslesen von Daten aus dem eID-Dokument für einen oder mehrere Diensteanbieter übernimmt. Im Folgenden wird diese Dienstleistung „eID-Service" genannt. Hierzu kann der eID-Service die verwendeten Protokolle und Komponenten direkt ansprechen und die Kommunikation zwischen den verschiedenen Komponenten selbst koordinieren (siehe *Abbildung 4*) oder den eID-Server als ein Hilfssystem neben anderen benutzen. Beide Möglichkeiten werden im Folgenden skizziert.

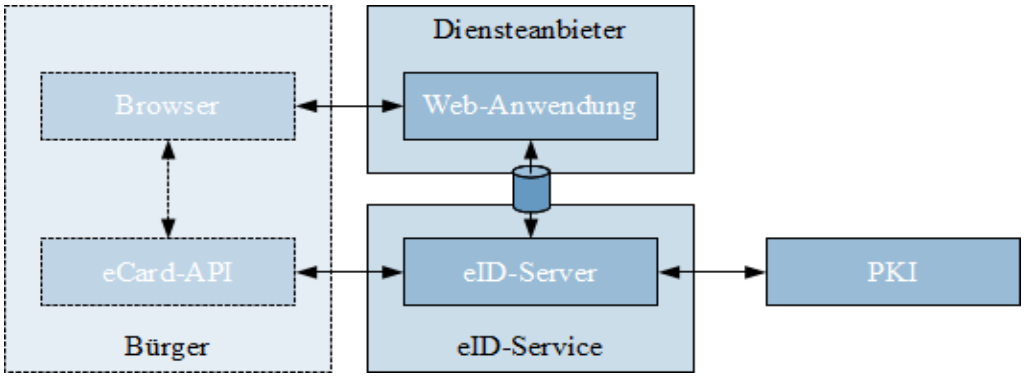

Abbildung 4: Online-Authentisierung bei entferntem Betrieb des eID-Servers

Im oben dargestellten Szenario greift der Diensteanbieter auf die in *Kapitel 4* dieser Technischen Richtlinie beschriebene funktionale Schnittstelle zu, um die eID-Funktion des eID-Dokuments zu nutzen. Neben der funktionalen Schnittstelle und der administrativen Management-Schnittstelle kann der eID-Service weitere Serviceleistungen beinhalten. Da der Zugriff hier über ein offenes Netz stattfindet, müssen alle Nachrichten verschlüsselt und signiert ausgetauscht werden. Dies wird in der Abbildung *4* durch den Kanal zwischen Web-Anwendung und eID-Server veranschaulicht und im *Anhang C* dieser Technischen Richtlinie erläutert.

Ein weiteres typisches Szenario mit eID-Service ist in *Abbildung 5* dargestellt. Hier wird der eID-Service als Teil der Serviceleistungen eines Identity Providers bereitgestellt. Der Bürger kann für die Web-Anwendung des Diensteanbieters verschiedene Authentisierungsarten benutzen. Der Diensteanbieter wird in diesem Szenario auch als „Relying Party" bezeichnet, da er für die Authentifizierung des Bürgers und ggf. weitere Attribute auf den eID-Service und dessen Identity Management zurückgreift. Hierzu stellt der Diensteanbieter dem Bürger die Möglichkeit bereit, eine oder mehrere Authentisierungsarten zu nutzen.

Nach Auswahl der Authentisierungsart durch den Bürger leitet der Diensteanbieter eine entsprechende Authentisierungsanfrage über den Web-Browser zusammen mit einem Token an den eID-Service weiter. Der eID-Service authentifiziert den Bürger je nach ausgewählter Authentisierungsart.

Wurde die eID-Funktion als Authentisierungsart ausgewählt, so führt der eID-Service die Authentisierung mit Hilfe des eID-Servers durch, kodiert die Authentifizierungsergebnisse bzw. einen Verweis darauf in einem Token und schickt dies über den Webbrowser des Bürgers (siehe gestrichelte Linie zwischen Identity Management und Browser in *Abbildung 5*) an den Diensteanbieter zurück. Der Diensteanbieter dekodiert das Token und die Authentifizierungsergebnisse bzw. holt diese über den dekodierten Verweis beim eID-Service ab.

Die aus dem eID-Dokument ausgelesenen Daten müssen dabei durch Signatur vor Manipulation und durch Verschlüsselung vor unberechtigtem Mitlesen geschützt werden. Diese sichere Kommunikation ist in der folgenden Abbildung durch einen Kanal zwischen Identity Management und Web-Anwendung dargestellt. Nach der Authentisierung müssen die aus dem eID-Dokument ausgelesenen Daten wieder aus den Systemen des eID-Service Anbieters gelöscht werden.

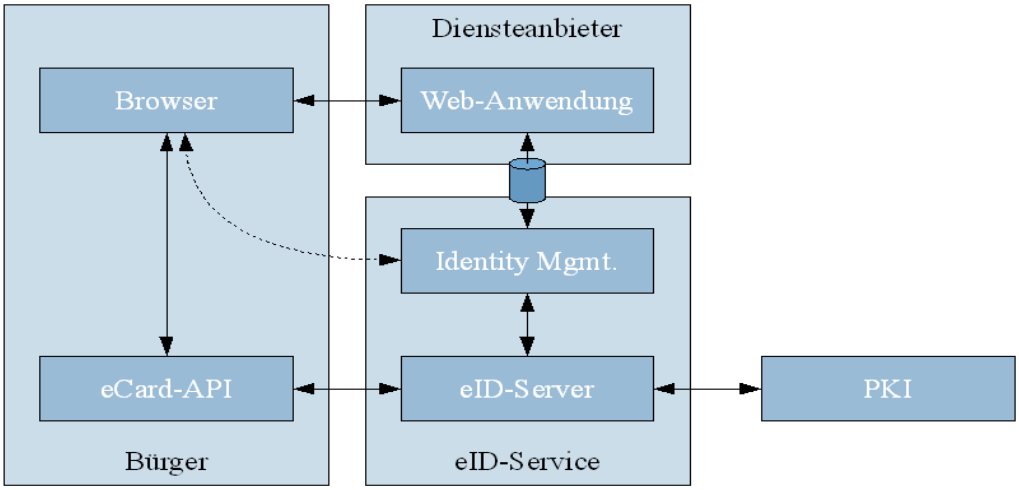

Abbildung 5: Betrieb des eID-Server mit Identity Management beim eID-Service Anbieter

Die skizzierte Architektur kann z.B. mit einem Security Token Service als Identity Provider und SAML- Token als Token realisiert werden (siehe *Anhang A: Verwendung mit SAML*). Je nach Einsatzzweck des Identity Providers können unterschiedliche Technologien zum Einsatz kommen, um die Übertragung der Informationen zwischen den drei Parteien zu realisieren.

Der Identity Provider muss für die Nutzung des elektronischen Identitätsnachweises das Authentisierungszertifikat des jeweiligen Diensteanbieters einsetzen. Die Mandantenfähigkeit des eID-Services ist daher gesondert zu betrachten.

2.5 Abgrenzung des eID-Servers

Der eID-Server liest die Identitätsdaten aus dem eID-Dokument aus und stellt sie der Geschäftslogik bereit. Der eID-Server hält keine Identitätsdaten vor und föderiert keine Bestätigungen über Authentisierungen oder Attribute an Dritte.

Der Bürger authentisiert sich formal nicht bei dem eID-Service, sondern bei dem Diensteanbieter. Die Identitätsdaten werden technisch vom eID-Dokument vorgehalten und bereitgestellt und nicht vom eID-Service.

Der eID-Service ist daher auch dann kein Identity Provider, wenn er entfernt und ggf. durch andere betrieben wird (siehe *Abbildung 4 und 5*).

3 Entwurfsentscheidungen

In diesem Abschnitt werden die Entscheidungen dokumentiert, die bei der Erstellung dieser Richtlinie getroffen wurden.

3.1 XML-Datenstruktur

Gegenstand:	Datenformat für Daten und Funktionen der eID-Funktion
Annahmen:	keine
Motivation:	XML-Strukturen entsprechen dem Stand der Praxis (sehr gut maschinell lesbar, in Maßen menschlich lesbar)
Entscheidung:	Für Anfragen, Antworten und übergebene Daten werden XML-Strukturen verwendet (siehe *Kapitel 4.3*). Die Funktionen und Datentypen der eID-Schnittstelle werden in XML spezifiziert und können unabhängig vom spezifizierten Kommunikationsprotokoll wiederverwendet werden.
Alternativen:	Binärformat; CSV-Format; ...
Betroffene Komponenten:	eID-Schnittstelle

3.2 XML-Signatur

Gegenstand:	Integrität und Authentizität (Mandantentrennung) von Anfragen und Antworten über die eID-Schnittstelle
Annahmen:	Es existieren eine oder mehrere Public-Key-Infrastrukturen auf Basis von X.509-Zertifikaten, die von Web-Anwendung und eID-Server zur Signatur von Daten genutzt werden können, so dass sie ihre Zertifikate gegenseitig validieren können.
Motivation:	XML-Signaturen schränken die Flexibilität von Web Services nicht ein und entsprechen dem Stand der Praxis.
Entscheidung:	Die Kommunikation über die eID-Schnittstelle wird auf der Ebene von XML-Nachrichten signiert.
Alternativen:	Integritäts- und Authentizitätsschutz auf Netzebene (z.B. mit TLS/SSL); ...
Betroffene Komponenten:	eID-Schnittstelle

3.3 Verschlüsselung der Netzkommunikation

Gegenstand:	Vertraulichkeit von Anfragen und Antworten über die eID-Schnittstelle
Annahmen:	Es existieren eine oder mehrere Public-Key-Infrastrukturen, die von Web-Anwendung und eID-Server zur Verschlüsselung der Netzkommunikation benutzt werden können.
Motivation:	Eine Verschlüsselung der Netzkommunikation ist Stand der Praxis und stellt z.B. bei Hardwarelösungen keinen Performanceengpass dar. Die gegenseitige Authentifizierung dient lediglich als Zugangskontrolle in der Netzkommunikation, so dass nur zugelassene Web-Anwendungen den eID-Server netztechnisch erreichen können.
Entscheidung:	Die Kommunikation über die eID-Schnittstelle erfolgt verschlüsselt, z.B. auf der Transportschicht.
Alternativen:	XML Verschlüsselung; ...
Betroffene Komponenten:	eID-Schnittstelle

3.4 Aufruf der eID-Schnittstelle

Gegenstand:	Informationsfluss vom eID-Server zur Web-Anwendung, nachdem die Anfrage an das eID-Dokument bearbeitet wurde.
Annahmen:	keine
Motivation:	Eine Anfrage der Web-Anwendung an die eID-Schnittstelle kann erst beantwortet werden, nachdem die eCard-API die Benutzerinteraktion durchgeführt hat und die entsprechenden Daten abgerufen wurden.
Entscheidung:	Die Web-Anwendung ruft wiederholt die eID-Schnittstelle auf, bis die Daten zur Beantwortung der Anfrage vorliegen.
Alternativen:	Sobald die Daten dem eID-Server vorliegen, ruft der eID-Server eine Schnittstelle in der Web-Anwendung auf und übermittelt die Daten. Nachteil: Diese Lösung bedingt eine ausgehende Kommunikation vom eID-Server und erhöhte Komplexität in der Web-Anwendung. Die Anfrage an die eID-Schnittstelle blockiert diese so lange, bis die Antwort vorliegt. Nachteil: Bei dieser Alternative schränken die maximal möglichen offenen Verbindungen die Leistung des eID-Servers stark ein. Dadurch kann die Verfügbarkeit des Systems nicht garantiert werden.
Betroffene Komponenten:	eID-Schnittstelle

3.5 Verwaltung der Authentisierungszertifikate

Gegenstand:	Schutz der Kommunikation des Diensteanbieters mit der Berechtigungs-CA (DVCA), z.B. für das erstmalige Abholen eines Terminal-Berechtigungszertifikats (card-verifiable Terminal Certificate).
Annahmen:	Es existieren eine oder mehrere Public-Key-Infrastrukturen auf Basis von X.509-Zertifikaten, die von der Berechtigungs-CA (DVCA) mit ihren Kunden (also den Diensteanbietern) benutzt werden kann, so dass sie ihre Zertifikate gegenseitig validieren können.
Motivation:	Die regelmäßige Erneuerung von Terminal-Berechtigungszertifikaten muss automatisiert und gemäß [EAC-PKI Protocol] erfolgen. Die sonstige Kommunikation zwischen Berechtigungs-CA und ihren Kunden kann durch die Berechtigungs-CA festgelegt werden.
Entscheidung:	Das Authentisierungszertifikat eines Diensteanbieters kann im eID-Server verwaltet werden und von diesem u.a. zur initialen Abholung von Terminal-Berechtigungszertifikaten benutzt werden.
Alternativen:	Der eID-Server stellt dem Diensteanbieter einen Teil der DVCA-Schnittstelle zur Verfügung. Darüber kann der Diensteanbieter bei der DVCA ein initiales Terminal-Berechtigungszertifikat abholen. Dieses Terminal-Berechtigungszertifikat übergibt er dann dem eID-Server an der Management-Schnittstelle.
Betroffene Komponenten:	Management-Schnittstelle

4 Funktionale eID-Schnittstelle

Der eID-Server bietet die eID-Schnittstelle an, durch die Web-Anwendungen die Funktionen der eID-Funktion des eID-Dokuments nutzen können. Die eID-Schnittstelle ist ein Web Service und wird in der Web Services Description Language (WSDL) formal beschrieben. Die WSDL-Datei ist dieser Technischen Richtlinie als Datei `TR-03130_TR-eID-Server.wsdl` beigefügt. Die Datentypen dieser Schnittstelle werden separat in der XSD-Datei `TR-03130_TR-eID-Server.xsd` beschrieben. Beide Dateien werden formal als *Anhang B: Schemadateien* zu dieser Technischen Richtlinie geführt.

Im Folgenden wird der Ablauf bei der Verwendung eines eID-Dokuments zur Authentisierung innerhalb einer Web-Anwendung beschrieben.

4.1 Übersicht

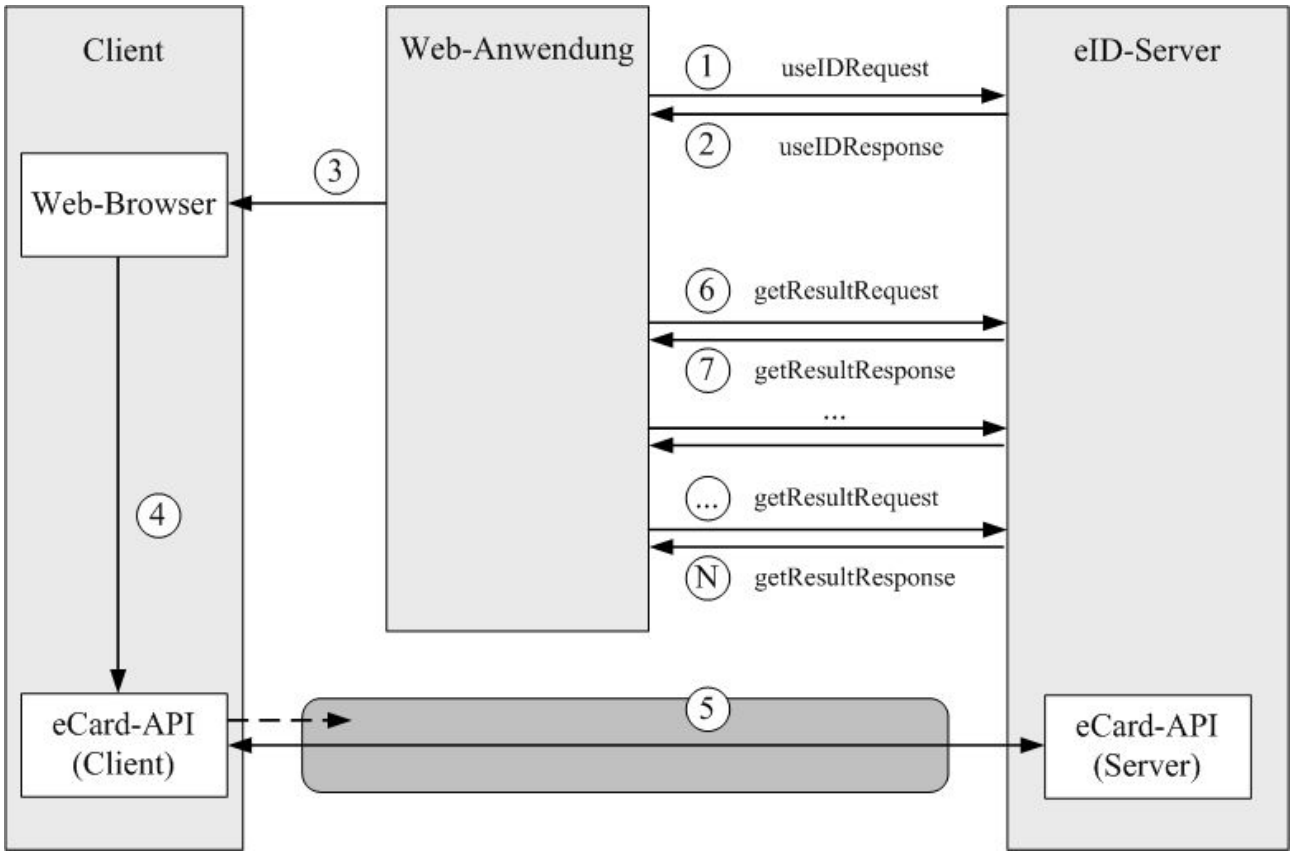

Abbildung 6: Ablauf der Kommunikation

In *Abbildung 6* sind die Systeme dargestellt, die bei der Nutzung der eID-Schnittstelle beteiligt sind. Im Folgenden werden die Schritte beschrieben, die eine Web-Anwendung durchführen muss, um die eID-Schnittstelle zu nutzen. Diese Schritte und die dabei übertragenen Nachrichten sind ebenfalls in der *Abbildung 7* verzeichnet.

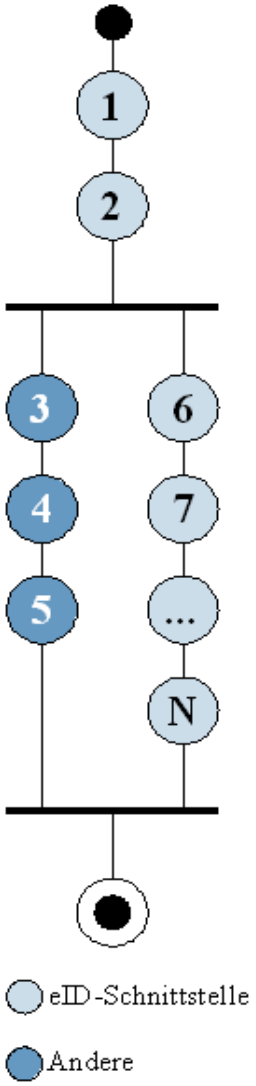

Abbildung 7: Ablaufdiagramm

1.) Aufruf der Funktion `useID`: Ein Benutzer der Web-Anwendung will den elektronischen Identitätsnachweis seines eID-Dokuments in der Web-Anwendung einsetzen. Die Web-Anwendung sendet daraufhin die Nachricht `useIDRequest` an die eID-Schnittstelle des eID-Servers. In dieser Nachricht werden die Funktionen und Daten des eID-Dokuments ausgewählt, die die Web-Anwendung nutzen bzw. auslesen will. Je nach ausgewählter Funktion werden noch weitere Informationen übergeben. Optional kann von der Web-Anwendung an dieser Stelle schon der Pre-Shared Key (PSK) mit dem die spätere eCard-API Kommunikation initialisiert wird festgelegt werden.

2.) Ergebnis der Funktion `useID`: Die in Schritt 1 ausgewählten Funktionen des eID-Dokuments müssen durch den Benutzer mit PIN-Eingabe freigegeben werden, daher werden die angeforderten Daten erst im Schritt N zurückgeliefert. Der Rückgabewert `useIDResponse` des Funktionsaufrufs enthält die Session-ID und den PSK, sowie optional die eCard-API Server Zieladresse.

3.) Weitergabe der Informationen an den Web-Browser: Die Web-Anwendung übergibt den PSK und die Verbindungsparameter für die eCard-API Kommunikation an den Web-Browser des Benutzers.

4.) Weitergabe der Verbindungsparameter an den eCard-API Client: Der Web-Browser leitet den PSK und die Verbindungsparameter an die clientseitige Instanz der eCard-API weiter. Die eCard-API bearbeitet daraufhin die Anfrage und führt die notwendige Benutzerinteraktion durch.

5.) Während der Benutzerinteraktion der eCard-API: Während der eCard-API Client die Benutzerinteraktion durchführt, ruft die Web-Anwendung wiederholt die Funktion `getResult` auf. Der Parameter `requestCounter` wird mit jedem Aufruf um 1 (siehe *Tabelle 4: Funktion useID Parameter*) erhöht.

6.) Aufruf der Funktion `getResult`: Die Web-Anwendung ruft das Ergebnis der in Schritt 1 gestellten Anfrage ab.

7.) Ergebnis der Funktion `getResult`: Wenn noch keine Informationen aus dem eID-Dokument zu der Anfrage vorliegen, beantwortet der eID-Server den Funktionsaufruf mit einer Fehlermeldung. Um die Anzahl der Aufrufe der `getResult`-Funktion zu reduzieren (ggf. sogar bis zu einem Aufruf) kann der eID-Server die `getResultResponse` herauszögern. Dies setzt jedoch eine enge Abstimmung mit dem Betreiber der Web-Anwendung (Diensteanbieter) voraus, da eventuell vorhandene Antwortzeiteinschränkungen (Timeouts) seitens der Web-Anwendung sonst zu einem Verbindungsabbruch führen können. Die Vorgehensweise kann darüber hinaus zu Lasten der Verfügbarkeit des eID-Servers (siehe *Kapitel 3.4 Aufruf der eID-Schnittstelle*) gehen. Die Erreichung der in *Anhang C: Anforderungen an den Betrieb von eID-Servern Kapitel 4.1 Grundwerte der Informationssicherheit* definierten Sicherheitsziele mit geeigneten Maßnahmen ist daher gesondert zu betrachten.

N.) Ergebnis der Funktion `useID` liegt vor: Nach Abschluss der Benutzerinteraktion und Verarbeitung der Daten durch die eCard-API beantwortet der eID-Server `getResultRequest`-Nachrichten der Web-Anwendung nicht länger mit einer Fehlermeldung, sondern liefert das Ergebnis der über die Funktion `useID` angefragten Daten und Funktionen.

Im Folgenden wird das einmalige Durchlaufen der Schritte 1-N als Anfrage an die eID-Schnittstelle bezeichnet.

4.2 Funktionen

In diesem Abschnitt werden die Funktionen der eID-Schnittstelle im Detail beschrieben.

4.2.1 Funktion `useID`

Durch die Funktion `useID` werden die Datenfelder und Funktionen der eID-Anwendung angegeben, die aus dem eID-Dokument ausgelesen werden sollen. Die wählbaren Funktionen sind die Altersverifikation und die Überprüfung des Wohnorts. In jeder Anfrage darf die Funktion `useID` nur einmal aufgerufen werden.

Parameter	*Beschreibung*
`UseOperations`	Legt die Datenfelder und Funktionen fest, die verpflichtend oder optional aus dem eID-Dokument ausgelesen werden sollen (unabhängig vom Terminal-Berechtigungszertifikat). Die folgenden Werte sind an dieser Stelle für die `AttributeSelectionType`-Elemente gültig: `REQUIRED` – Zur Kennzeichnung von Datenfeldern und Operationen, die für den Anwendungsfall verpflichtend benötigt werden. `ALLOWED` – Zur Kennzeichnung von für den Anwendungsfall optionalen Datenfeldern und Operationen. `PROHIBITED` – Zur Kennzeichnung von für den Anwendungsfall nicht erwünschten Datenfeldern.
`AgeVerificationRequest`	Wenn eine Altersverifikation durchgeführt werden soll, gibt dieser Parameter das Lebensjahr an, dass der Inhaber des eID-Dokuments vollendet haben soll. Wenn die Altersverifikation im Parameter `UseOperations` gewählt ist, muss dieser

	Parameter vorhanden sein.
`PlaceVerificationRequest`	Wenn eine Überprüfung des Wohnorts durchgeführt werden soll, gibt dieser Parameter die zu überprüfenden Wohnort-ID an. Wenn die Überprüfung des Wohnorts im Parameter `UseOperations` gewählt ist, muss dieser Parameter vorhanden sein.
`PSK`	In verschiedenen Szenarien stellt die initiale Übermittlung des PSK durch die Web-Anwendung eine Vereinfachung dar. Der PSK kann daher optional schon beim Aufruf der Funktion `useID` übermittelt werden.

Tabelle 4 : Funktion useID Parameter

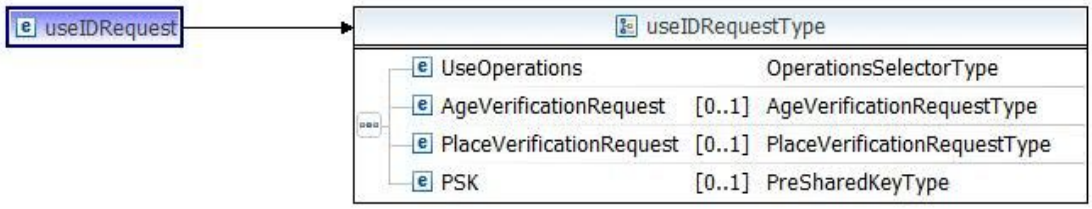

Abbildung 8: Funktion useID Parameter

Tabelle 4 beschreibt die in *Abbildung 8* dargestellten Parameter der Funktion useID. Unabhängig von den ausgewählten Funktionen führt der eID-Server die Gültigkeitsprüfung des eID-Dokuments durch, um die Echtheit des Dokuments sicherzustellen.

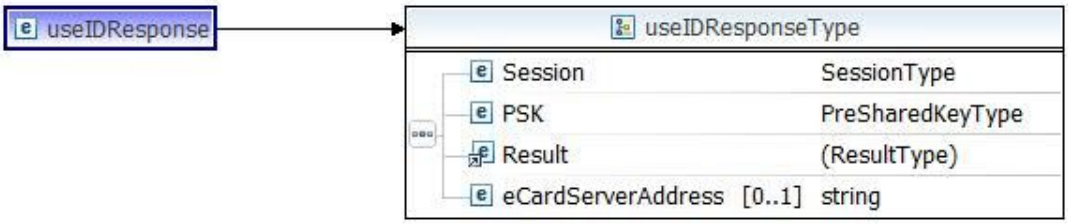

Abbildung 9: Funktion useID Rückgabewerte

In *Abbildung 9* werden die Rückgabewerte der Funktion `useID` dargestellt. Diese sind ein Element vom Typ `SessionType`, der Pre-Shared-Key (`PSK`), ein `Result`-Element, sowie optional ein Element `eCardServerAddress`.

Rückgabewert	*Beschreibung*
`Session`	Die `Session` verbindet den Funktionsaufruf `useID` mit dem Funktionsaufruf `getResult`. Nachdem das Ergebnis der Anfrage durch die Funktion `getResult` erfolgreich abgerufen wurde, oder die maximale Sitzungszeit überschritten wurde, wird die `Session` ungültig.
`PSK`	Der`PSK` ist der initiale Schlüssel für den in *Abbildung 6* dargestellten verschlüsselten Kanal zwischen eCard-API Client und Server und gehört zu den Parametern, die die Web-Anwendung an den eCard-API Client übergibt. Wurde der `PSK` bereits beim Aufruf der Funktion `useID` von der Web-Anwendung an den eID-Server übermittelt, so ist eben dieser `PSK` dennoch auch Bestandteil der Antwort des eID-Servers.
`Result`	Zeigt an, ob die Anfrage abgearbeitet werden konnte oder ob ein Fehler aufgetreten ist.
`eCardServerAddress`	Mit diesem Element kann der Web-Anwendung die Zieladresse, unter welcher die eCard-API Server-Komponente zu erreichen ist, mitgeteilt werden, wenn diese nicht fix ist. Die Zieladresse muss von der Web-Anwendung dann entsprechend [eCard-API] *Kapitel 2.3.1* als Parameter (`ServerAddress`) für das Browser-Plugin kodiert werden.

Tabelle 5: Funktion useID Rückgabewerte

Die eID-Schnittstelle erlaubt für jede Web-Anwendung ein Maximum an gleichzeitig aktiven Anfragen. Dieses Maximum wird so gewählt, dass eine hohe Auslastung einer einzelnen Web-Anwendung nicht zu einer eingeschränkten Verfügbarkeit anderer Web-Anwendungen führt. Wenn das Maximum an gleichzeitigen Anfragen für eine Web-Anwendung überschritten wird, sendet der eID-Server im `Result`-Element eine Fehlermeldung mit der `ResultMinor`-URI `.../useID#tooManyOpenSessions` (siehe *Kapitel 4.5.1*).

4.2.2 Funktion `getResult`

Durch die Funktion `getResult` wird das Ergebnis einer Anfrage abgerufen. Nachdem die Funktion ohne Fehler ausgeführt wurde, wird die Session ungültig und der Server muss die abgefragten Daten löschen.

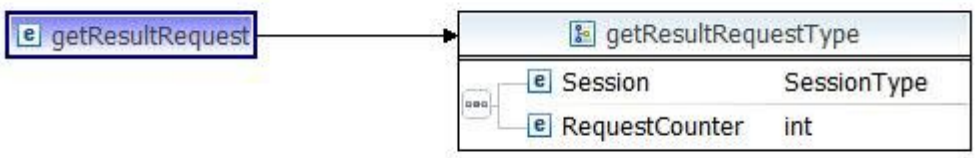

Abbildung 10: Funktion getResult Parameter

In *Abbildung 10* und *Tabelle 6* werden die Parameter der Funktion `getResult` beschrieben.

Parameter	***Beschreibung***
`Session`	Gibt die `Session` der Anfrage an zu der dieser Funktionsaufruf gehört.
`RequestCounter`	Ein Zähler, der die Funktionsaufrufe innerhalb einer Anfrage eindeutig identifizierbar machen muss. Für jeden erneuten Aufruf der Funktion `getResult` innerhalb einer Anfrage muss dieser Wert inkrementiert werden.

Tabelle 6: Funktion getResult Parameter

Innerhalb einer Anfrage, deren Ablauf in *Abbildung 6* dargestellt ist, wird die Funktion `getResult` potenziell mehrfach aufgerufen. Die Web-Anwendung ruft die Funktion innerhalb einer Anfrage so oft auf, bis sie nicht mehr den Fehler `.../getResult#noResultYet` erhält, siehe *Kapitel 4.5.1*. In jedem Aufruf der Funktion wird der Parameter `requestCounter` erhöht. Der eID-Server antwortet auf Anfragen, die einen gleichen oder kleineren `requestCounter`-Wert enthalten, als in einem vorherigen Aufruf der Funktion, mit einer Fehlermeldung. Durch dieses Verfahren werden Replay-Angriffe auf die `getResult`-Funktion verhindert.

Rückgabewert	***Beschreibung***
`PersonalData`	Enthält die aus dem eID-Dokument ausgelesenen Datenfelder.
`Result`	Zeigt an, ob die Anfrage abgearbeitet werden konnte oder ob ein Fehler aufgetreten ist. Insbesondere kann hier der Fehler .../getResult#noResultYet auftreten (siehe *Kapitel 4.5.1*).
`OperationsAllowedByUser`	Gibt die Funktionen und Datenfelder an, die effektiv aus dem eID-Dokument ausgelesen

	werden konnten, nachdem das Terminal-Berechtigungszertifikat und die einschränkende Auswahl des Benutzers angewandt wurden. Die folgenden Werte sind an dieser Stelle für die `AttributeSelectionType`-Elemente gültig: `ALLOWED` – Zur Kennzeichnung der vom Benutzer freigegebenen Datenfelder und Operationen. `PROHIBITED` – Zur Kennzeichnung der vom Benutzer nicht freigegebenen Datenfelder und Operationen. Der Wert `REQUIRED` ist in diesem Kontext nicht zu verwenden.
`FulfilsAgeVerification`	Wenn die Altersverifikation im Parameter `UseOperations` aktiviert wurde und dieser Funktionsaufruf erfolgreich war, enthält dieser Rückgabewert das Ergebnis der Altersverifikation.
`FulfilsPlaceVerification`	Wenn die Überprüfung des Wohnorts im Parameter `UseOperations` aktiviert wurde und dieser Funktionsaufruf erfolgreich war, enthält dieser Rückgabewert das Ergebnis der Überprüfung.

Tabelle 7: Funktion getResult Rückgabewerte

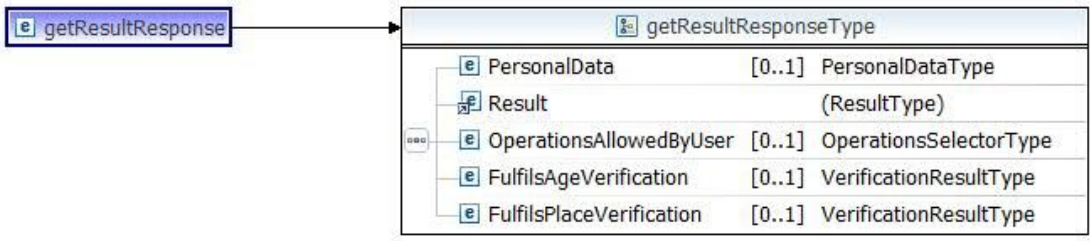

Abbildung 11: Funktion getResult Rückgabewerte

Als Ergebnis der Funktion `getResult` werden die in *Tabelle 7* beschriebenen und in *Abbildung 11* dargestellten Elemente zurückgeliefert.

4.2.3 Funktion `getServerInfo`

Die Funktion `getServerInfo` stellt der Web-Anwendung Informationen über den eID-Server zur Verfügung, anhand derer die Web-Anwendung die Konfiguration des eID-Servers überprüfen kann. Der Parameter der Funktion ist ein Element vom Typ `nullType` . Dieser Typ enthält keine Informationen und dient dazu, dass die Funktion korrekt aufgerufen werden kann.

Rückgabewert	*Beschreibung*
`ServerVersion`	Gibt die Version der eID-Schnittstelle an, die der eID-Server unterstützt.
`DocumentVerificationRights`	Gibt die Funktionen an, die die Web-Anwendung mit dem aktuell konfigurierten Terminal-Berechtigungszertifikat nutzen kann. Die folgenden Werte sind an dieser Stelle für die `AttributeSelectionType`-Elemente gültig: `ALLOWED` – Zur Kennzeichnung der Datenfelder für die das Terminal-Berechtigungszertifikat Leserechte hat. `PROHIBITED` – Zur Kennzeichnung der Datenfelder für die das Terminal-Berechtigungszertifikat keine Leserechte hat. Der Wert `REQUIRED` ist in diesem Kontext nicht zu verwenden.

Tabelle 8: Funktion getServerInfo Rückgabewerte

In *Abbildung 12* wird der Rückgabetyp der Funktion `getServerInfo` dargestellt und in *Tabelle 8* werden seine Elemente beschrieben.

Abbildung 12: Funktion getServerInfo Rückgabewerte

4.3 Datentypen

In diesem Abschnitt werden die Datentypen beschrieben, die an der eID-Schnittstelle verwendet werden. Bei den Datentypen handelt es sich größtenteils um komplexe Datentypen, die in einem XML-Schema, siehe [XML-Type], definiert werden. Dieses XML-Schema ist in der XSD-Datei `TR-03130_TR-eID-Server.xsd` enthalten, die dieser Richtlinie gemäß *Anhang B* beigefügt ist.

4.3.1 Datentyp `Session Type`

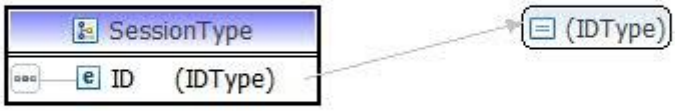

Abbildung 13: Datentyp SessionType

Der Typ `SessionType` wird verwendet, um eine Anfrage an die eID-Schnittstelle zu identifizieren. Zur eindeutigen Identifizierung von verschiedenen Anfragen vergibt in der Regel der eID-Server Werte für das `ID`-Element, siehe *Abbildung 13*. Ein Wert für das `ID`-Element muss zufällig gewählt werden und in hexadezimaler Darstellung mindestens 32 Zeichen lang sein, damit dieser nicht leicht zu erraten ist.

Um eine Funktion der eID-Schnittstelle aufzurufen, muss die Web-Anwendung die `Session` der Anfrage übergeben. Nach dem einmaligen erfolgreichen Abrufen der Daten einer Anfrage wird die dafür verwendete `Session` ungültig. Für jede Anfrage an die eID-Schnittstelle muss daher eine neue `Session` verwendet werden. Eine `Session` kann auch ablaufen und somit ungültig werden, wenn die maximale Sitzungszeit überschritten wurde.

4.3.2 Datentyp `RestrictedIDType`

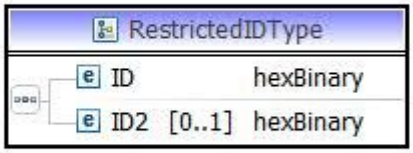

Abbildung 14: Datentyp RestrictedIDType

Der in *Abbildung 14* dargestellte Datentyp `RestrictedIDType` wird auch als (aus Sicht des Diensteanbieters) kartenspezifisches Kennzeichen bezeichnet und in [EAC 2] *Abschnitt 4.5* beschrieben. Die `ID` ist für jeden Benutzer einer Web-Anwendung eindeutig und kann daher zur Identifizierung eines Benutzers in der Web-Anwendung verwendet werden. Wenn das Terminal-Berechtigungszertifikat des Diensteanbieters einen zweiten Terminal-Sektor enthält, so wird eine zweite `ID` auf dem Chip des eID-Dokuments gebildet und in dem optionalen Element `ID2` übertragen.

4.3.3 Datentyp `PersonalDataType`

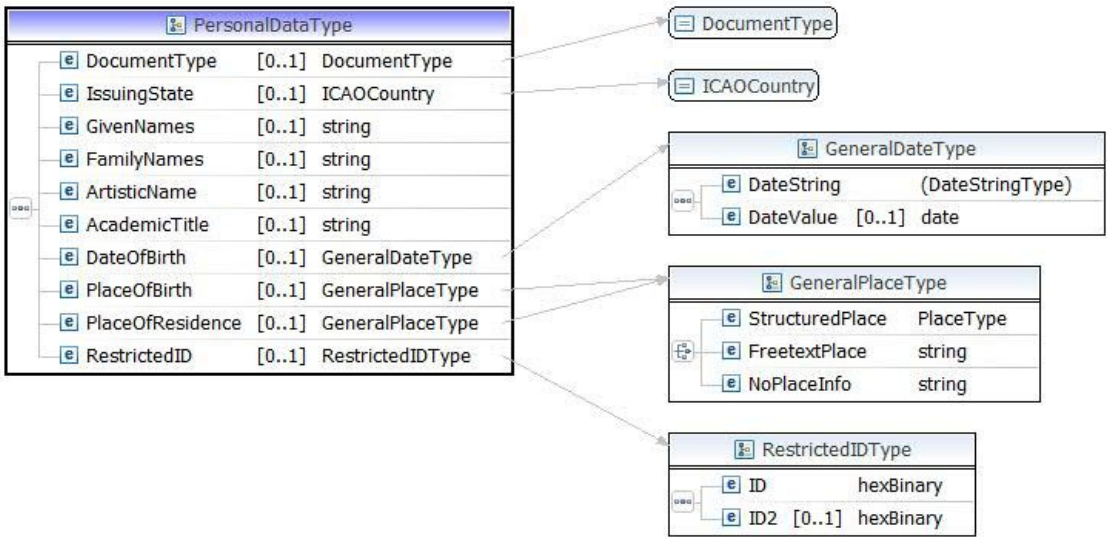

Abbildung 15: Datentyp PersonalDataType

Der in *Abbildung 15* dargestellte Datentyp `PersonalDataType` beschreibt die Daten, die aus der eID-Anwendung des eID-Dokuments ausgelesen werden können. Die Elemente des Datentyps entsprechen den in [EAC 2] *Anhang E.2* beschriebenen Datenfeldern der eID-Anwendung. Die in *Abbildung 15* dargestellten Typen `DocumentType` und `ICAOCountry` sind vom Typ `string` abgeleitet und beschränken die erlaubten Zeichen gemäß der in [EAC 2] *Anhang E.2* definierten Grammatik. Auch die neu definierten Datentypen `GeneralDateType` und `GeneralPlaceType` sind XML-Übersetzungen der in [EAC 2] beschriebenen ASN.1 Spezifikation.

4.3.4 Datentyp `GeneralPlaceType`

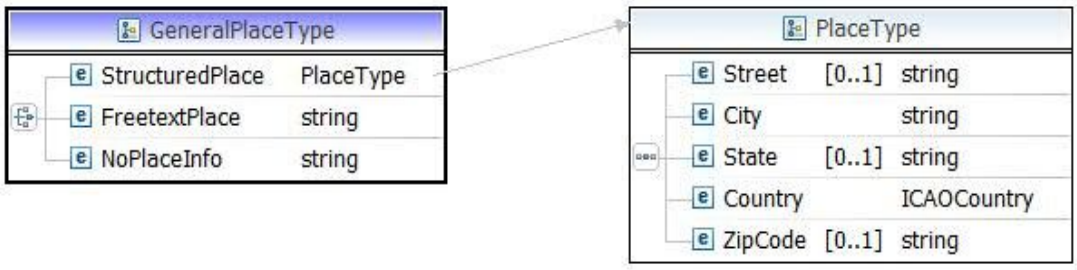

Abbildung 16: Datentyp GeneralPlaceType

Der in *Abbildung 16* dargestellte Datentyp `GeneralPlaceType` entspricht dem in [EAC 2] *Anhang E.2* beschriebenen Feld `General Place` der eID-Anwendung. Dieser Datentyp beschreibt verschiedene Formate in denen Adressen vorliegen können. Im Zusammenhang mit der eID-Funktion ist hier insbesondere der von dem Element `StructuredPlace` verwendete Datentyp `PlaceType` (siehe *Kapitel 4.3.5*) von Bedeutung.

4.3.5 Datentyp `PlaceType`

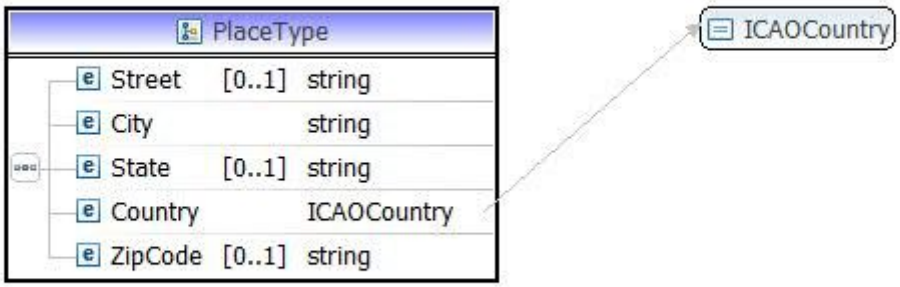

Abbildung 17: Datentyp PlaceType

Der in *Abbildung 17* dargestellte Datentyp `Place Type` entspricht dem in [EAC 2] *Anhang E.2* beschriebenen Feld `Place` der eID-Anwendung. Dieser Datentyp beschreibt strukturiert den Wohnort einer Person.

4.3.6 Datentyp `OperationsSelectorType`

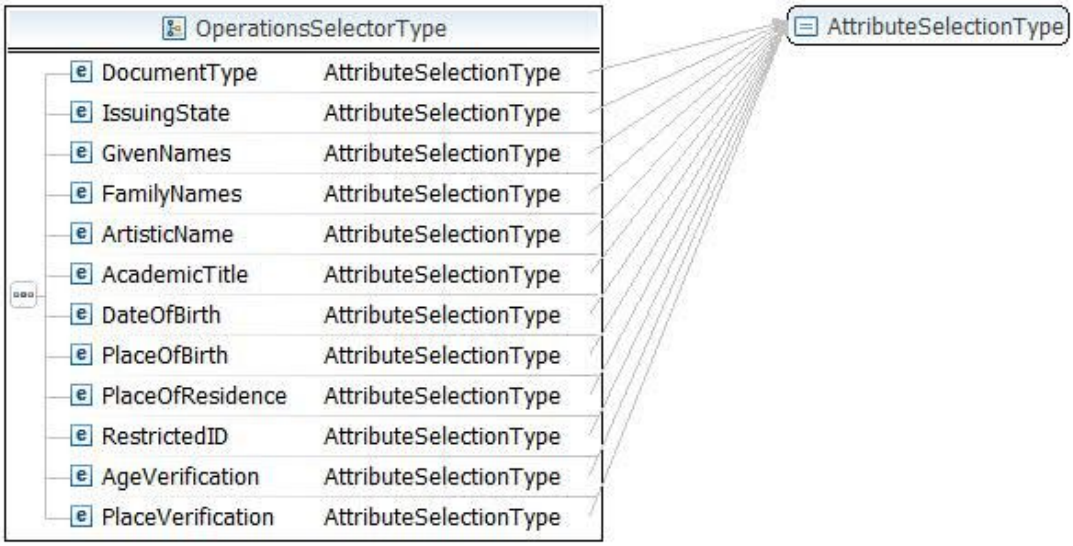

Abbildung 18: Datentyp OperationsSelectorType

In *Abbildung 18* ist der Datentyp `OperationsSelectorType` dargestellt, der die Funktionen und Datenfelder der eID-Anwendung des eID-Dokuments identifiziert. Diese können mit Hilfe des Datentyps `AttributeSelectionType` (siehe *Kapitel 4.3.13*) ausgewählt werden. Die Felder entsprechen den in [EAC 2] *Anhang E.2* beschriebenen Feldern der eID-Anwendung.

4.3.7 Datentyp `AgeVerificationRequestType`

Abbildung 19: Datentyp AgeVerificationRequestType

In *Abbildung 19* ist der Datentyp `AgeVerificationRequestType` dargestellt. Dieser Datentyp ist der Parameter für die Altersverifikation, die über die Funktion `useID` durchgeführt wird. Das Element `Age` gibt das Lebensjahr an, das der Inhaber des eID-Dokuments vollendet haben soll.

Um zu prüfen, ob eine Person ein bestimmtes Alter noch nicht erreicht hat, wird im Element `Age` das zu prüfende Lebensjahr angegeben. Wenn der Ergebnistyp `VerificationResultType` (siehe *Kapitel 4.3.10*) angibt, dass die Verifikation nicht erfüllt wird, bedeutet dies, dass die Person das Lebensjahr `Age` nicht vollendet hat, also jünger ist.

4.3.8 Datentyp `PlaceVerificationRequestType`

Abbildung 20: Datentyp PlaceVerificationRequestType

In *Abbildung 20* ist der Datentyp `PlaceVerificationRequestType` dargestellt. Dieser Datentyp ist der Parameter für die Überprüfung des Wohnorts, die über die Funktion `useID` durchgeführt wird. Das Element `CommunityID` entspricht der in [ePA Architektur] beschriebenen Wohnort-ID und wird mit der aus dem eID-Dokument ausgelesenen Wohnort-ID verglichen.

4.3.9 Datentyp `VersionType`

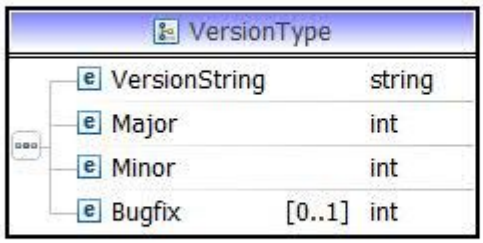

Abbildung 21: Datentyp VersionType

In *Abbildung 21* ist der Datentyp `VersionType` dargestellt. Dieser beschreibt die Version der eID-Schnittstelle in einer lesbaren Variante im Feld `VersionString`. Zur Identifizierbarkeit von verschiedenen Versionen bei der maschinellen Verarbeitung sind die Elemente `Major` für die Hauptversion und `Minor` für die Unterversion im Datentyp enthalten. Optional kann hier auch ein Kennzeichen `Bugfix` übertragen werden, welches Schemakorrekturen kennzeichnet, die bereits am eID-Server umgesetzt wurden.

4.3.10 Datentyp `VerificationResultType`

Abbildung 22: Datentyp VerificationResultType

In *Abbildung 22* ist der Datentyp `VerificationResultType` dargestellt. Dieser repräsentiert das Ergebnis einer Überprüfungsanfrage an das eID-Dokument. Das Element `FulfilsRequest` gibt an, ob die Anfrage durch den Besitzer des eID-Dokuments erfüllt wird oder nicht.

4.3.11 Datentyp `PreSharedKeyType`

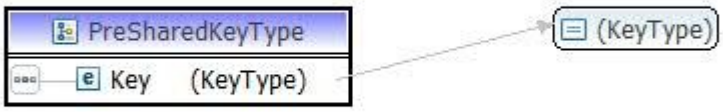

Abbildung 23: Datentyp PreSharedKeyType

In *Abbildung 23* ist der Datentyp `PreSharedKeyType` dargestellt. Dieser enthält in dem Element `Key` den Pre-Shared Key (PSK), welcher für die Initialisierung der eCard-API Kommunikation benötigt wird.

4.3.12 Datentyp `GeneralDateType`

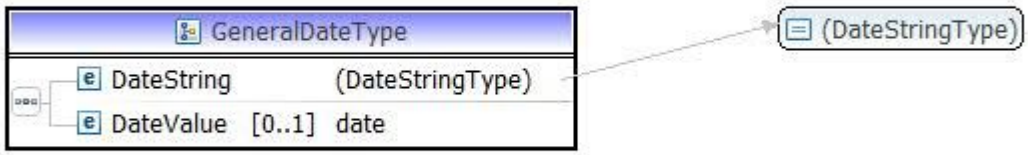

Abbildung 24: Datentyp GeneralDateType

In *Abbildung 24* ist der Datentyp `GeneralDateType` dargestellt. Dieser enthält immer die direkt aus dem Chip ausgelesene Repräsentation des Datums im Element `DateString`. Dabei handelt es sich um einen 8-stelligen `string`, der neben Zahlwerten (0-9) auch Leerzeichen enthalten kann und im Format **JJJJMMTT** (4 Stellen für das **J**ahr, 2 Stellen für den **M**onat und 2 Stellen für den **T**ag) übermittelt wird. Das optionale Element `DateValue` von Datentyp `date` wird nur verwendet, wenn ein vollständiges Datum vorliegt.

4.3.13 Datentyp `AttributeSelectionType`

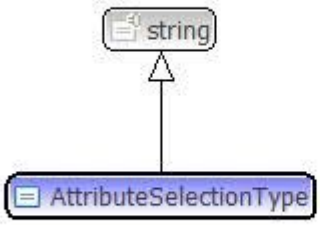

Abbildung 25: Datentyp AttributeSelectionType

In *Abbildung 25* ist der Datentyp `AttributeSelectionType` dargestellt. Dieser simple Datentyp wird vom Datentyp `string` abgeleitet und erlaubt die Werte `REQUIRED`, `ALLOWED` und `PROHIBITED`. Der Datentyp wird im Kontext mehrerer Funktionen verwendet, um die Selektion von einzelnen Attributen mit Hilfe des Datentyps `OperationsSelectorType` (siehe *Kapitel 4.3.6*) zu repräsentieren.

Die für den jeweiligen Kontext gültigen Werte sind in der folgenden Tabelle zusammengefasst:

Kontext	***REQUIRED***	***ALLOWED***	***PROHIBITED***
Element `UseOperations` in der Funktion `useIDRequest` (siehe Kapitel 4.2.1)			
Element `OperationsAllowedByUser` in der Funktion `getResultResponse` (siehe Kapitel 4.2.2)			
Element `DocumentVerificationRights` in der Funktion `getServerInfoResponse` (siehe Kapitel 4.2.3)			

Tabelle 9: Kontextabhängig erlaubte Werte des Datentyps AttributeSelectionType

4.4 Signatur und Verschlüsselung

Um die Vertraulichkeit, Integrität und Authentizität der übertragenen Daten und Funktionsaufrufe zu gewährleisten, wird eine Verschlüsselung auf der Transportschicht und eine Signatur der XML-Nachrichten eingesetzt. In der WSDL-Spezifikation `TR-03130_TR-eID-Server.wsdl`, die dieser Richtlinie beiliegt, spezifiziert das Element `Policy` die eingesetzten Verfahren und Schlüssel, die zur Benutzung der Schnittstelle verwendet werden müssen. Im Folgenden werden die Token erläutert, die in der Policy zur Beschreibung der Schlüssel verwendet werden.

Hinweis: Die Verschlüsselung auf Transportebene ist seit der Version 1.5 dieser Technischen Richtlinie nicht mehr Bestandteil der WSDL-Spezifikation. Eine entsprechende Anpassung war notwendig, da die Verwendung von zwei Bindings (Transport und Asymmetric) zu Problemen beim Einsatz einzelner Frameworks geführt hat. Trotz dieser Anpassung ist die Verschlüsselung auf der Transportebene weiterhin verpflichtend (siehe *Anhang C: Anforderungen an den Betrieb von eID-Servern 4.3.1 Betrachtung der Verbindung 2: eID-Server <=> Webserver*) umzusetzen. Dies muss nunmehr durch entsprechende Konfiguration sichergestellt werden.

4.4.1 InitiatorToken

```
<sp:InitiatorToken>
      <wsp:Policy>
            <sp:X509Token sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07
              /securitypolicy/IncludeToken/Never">
                <wsp:Policy>
                      <sp:RequireIssuerSerialReference/>
                      <sp:WssX509V3Token10 />
                </wsp:Policy>
            </sp:X509Token>
      </wsp:Policy>
</sp:InitiatorToken>
```

Das `InitiatorToken` ist ein X.509 Zertifikat, das eingesetzt wird, um die Anfragen an den eID-Server zu signieren. Der eID-Server überprüft die Gültigkeit der Signatur und unterscheidet

verschiedene Mandanten anhand ihrer X.509 Zertifikate. Die signierte Anfrage bearbeitet der eID-Server dann im Auftrag des Mandanten und mit dessen Berechtigungszertifikat.

4.4.2 RecipientToken

```
<sp:RecipientToken>
      <wsp:Policy>
            <sp:X509Token sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07
              /securitypolicy/IncludeToken/Never">
                <wsp:Policy>
                      <sp:RequireIssuerSerialReference/>
                      <sp:WssX509V3Token10 />
                </wsp:Policy>
            </sp:X509Token>
      </wsp:Policy>
</sp:RecipientToken>
```

Das `RecipientToken` ist das X.509 Zertifikat, dass die eID-Schnittstelle verwendet, um Antworten zu signieren. Die Web-Anwendung prüft die Signatur, um die Integrität und Authentizität der Antwort zu überprüfen.

4.5 Fehlerbehandlung

Abbildung 26: Das Element Result

Fehlerzustände, die bei der Bearbeitung einer Anfrage an der eID-Schnittstelle auftreten, werden an die Web-Anwendung gemeldet. Zur genaueren Beschreibung des Fehlers wird das in [eCard-API] *Teil 1 Abschnitt 4.1.2* beschriebene Element `Result` verwendet, siehe *Abbildung 26*.

Zusätzlich zu den in [eCard-API] *Teil 1 Abschnitt 4.2* beschriebenen Codes für Fehler, verwendet die eID-Schnittstelle den Präfix

`http://www.bsi.bund.de/eid/server/1.4/resultminor/`

in der `ResultMinor` -URI, um Fehler, deren Ursprung im eID-Server zu suchen ist, anzuzeigen.

4.5.1 Fehlercodes

Fehlercode	*Fehlerbeschreibung*
.../common#schemaViolation	**Schemaverletzung** Die Anfrage der Web-Anwendung entspricht nicht dem vom eID-Server verwendeten Schema.
.../common#internalError	**Interner Fehler.** Ein Fehler ist aufgetreten, der nicht mit den Fehlercodes abgebildet werden kann.
.../useID#invalidPSK	**Der initiale PSK ist ungültig.** Der von der Web-Anwendung an den eID-Server übertragene PSK ist ungültig und kann vom eID-Server nicht verarbeitet werden.
.../useID#tooManyOpenSessions	**Wegen zu hoher Auslastung des eID-Servers konnte die Anfrage nicht bearbeitet werden.**
.../useID#missingArgument	**Notwendige Parameter des Funktionsaufrufs fehlen.**

	Die Funktion `AgeVerification` oder `PlaceVerification` wurde ausgewählt, die entsprechenden `Request`-Elemente fehlen in der Anfrage.
.../useID#missingTerminalRights	**Notwendige Rechte fehlen.** Die für die Beantwortung der Anfrage notwendigen Rechte fehlen im Terminal-Berechtigungszertifikat.
.../getResult#noResultYet	**Die Anfrage ist noch nicht abgeschlossen.** Der eID-Server kann diese Anfrage noch nicht beantworten. Zu einem späteren Zeitpunkt kann der Funktionsaufruf erfolgreich sein.
.../getResult#invalidSession	**Die verwendete Session-ID ist ungültig.** Die Session ist abgelaufen oder wurde beantwortet und somit gelöscht.
.../getResult#invalidCounter	**Der `requestCounter` wurde nicht korrekt inkrementiert.** Die Anfrage ist ungültig und wird nicht bearbeitet, da der `requestCounter` kleiner oder gleich dem zuletzt bearbeitetem Wert ist.
.../getResult#invalidDocument	**Das verwendete eID-Dokument ist ungültig.** Durch die passive Authentisierung des eID-Dokuments oder die Überprüfung der Sperrliste wurde festgestellt, dass das verwendete eID-Dokument ungültig ist.

Tabelle 10: Liste der Fehlercodes

4.6 Beispielhafter Aufruf der eID-Schnittstelle

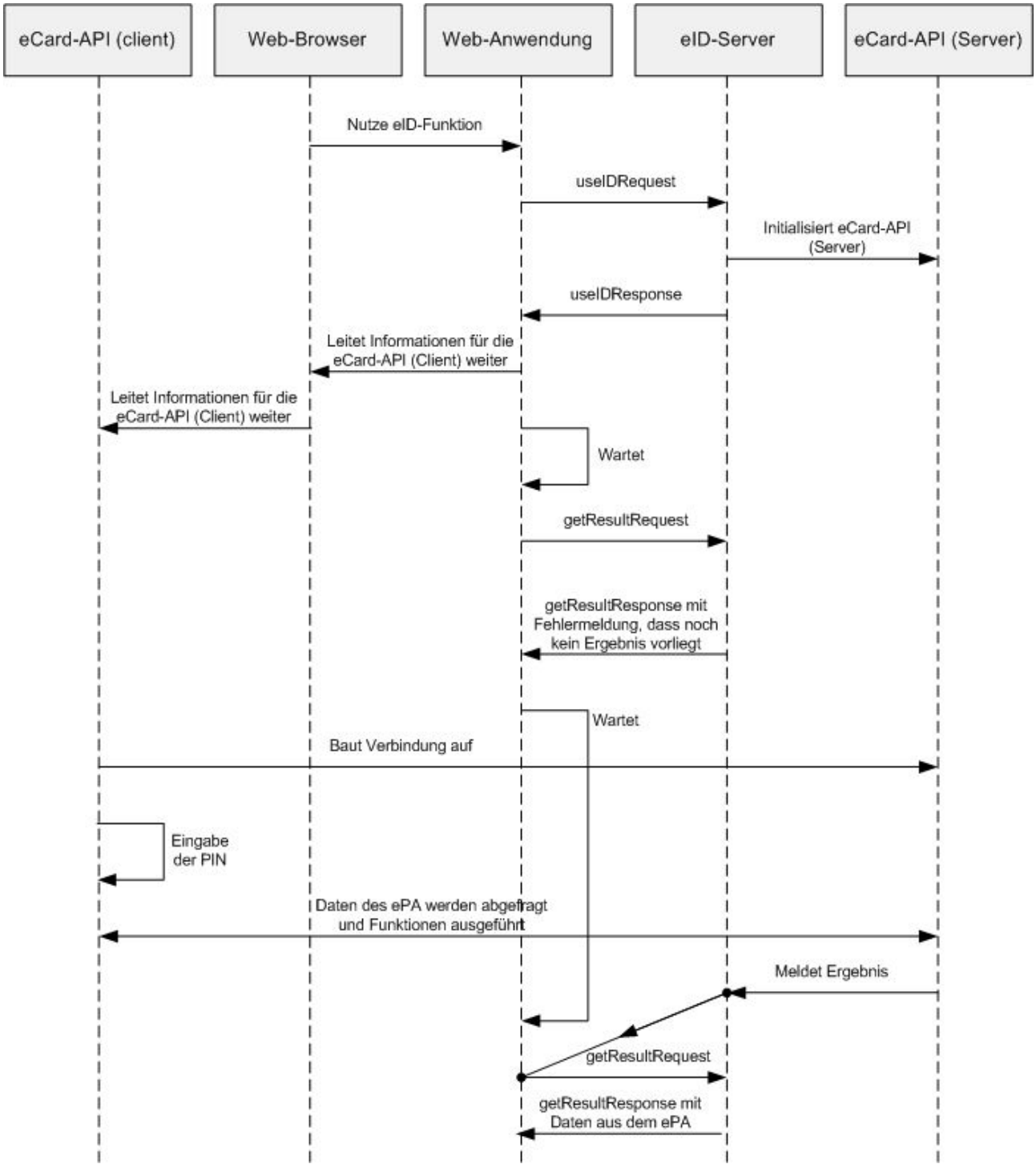

Abbildung 27: Sequenzdiagramm: Funktionaler Ablauf einer Anfrage

In *Abbildung 27* wird das Zusammenspiel der Funktionen der eID-Schnittstelle bei einer Anfrage in Form eines Sequenzdiagramms dargestellt. In diesem Beispiel führt die Web-Anwendung zwei Zyklen „Wartet“ durch. Im Allgemeinen führt die Web-Anwendung so lange Warte-Zyklen durch, bis der eID-Server von der eCard-API (Server) die Ergebnisse der Anfrage erhält.

Im Folgenden werden Beispiele für SOAP-Nachrichten dargestellt, mit denen die Funktionen der eID-Schnittstelle aufgerufen werden können.

4.6.1 Beispielhafter Aufruf der Funktion `useID`

Das folgende Codebeispiel zeigt einen exemplarischen Aufruf der Funktion `useID`. In diesem Beispiel erfragt die Web-Anwendung des Diensteanbieters beim Aufruf der Funktion den Vor- und Nachnamen. Der akademische Titel wird als optionales Datenfeld ebenfalls angefordert. Wie erwartet enthält die Rückmeldung des eID-Servers die Elemente `Session` und `PSK`. Das Element `Result` enthält in diesem Fall keine Fehlermeldung.

`useIDRequest`

```
<soapenv:Body>
      <eid:useIDRequest>
            <eid:UseOperations>
                  <eid:DocumentType>PROHIBITED</eid:DocumentType>
                  <eid:IssuingState>PROHIBITED</eid:IssuingState>
                  <eid:GivenNames>REQUIRED</eid:GivenNames>
                  <eid:FamilyNames>REQUIRED</eid:FamilyNames>
                  <eid:ArtisticName>PROHIBITED</eid:ArtisticName>
                  <eid:AcademicTitle>ALLOWED</eid:AcademicTitle>
                  <eid:DateOfBirth>PROHIBITED</eid:DateOfBirth>
                  <eid:PlaceOfBirth>PROHIBITED</eid:PlaceOfBirth>
                  <eid:PlaceOfResidence>PROHIBITED</eid:PlaceOfResidence>
                  <eid:RestrictedID>PROHIBITED</eid:RestrictedID>
                  <eid:AgeVerification>PROHIBITED</eid:AgeVerification>
                  <eid:PlaceVerification>PROHIBITED</eid:PlaceVerification>
            </eid:UseOperations>
      </eid:useIDRequest>
</soapenv:Body>
```

`useIDResponse`

```
<soapenv:Body>
      <eid:useIDResponse>
            <eid:Session>
                  <eid:ID>1234567890abcdef1234567890abcdef</eid:ID>
            </eid:Session>
            <eid:PSK>
                  <eid:Key>fedcba0987654321fedcba0987654321</eid:Key>
            </eid:PSK>
            <dss:Result>
                  <ResultMajor>
                        http://www.bsi.bund.de/ecard/api/1.1/resultmajor#ok
                  </ResultMajor>
            </dss:Result>
      </eid:useIDResponse>
</soapenv:Body>
```

4.6.2 Beispielhafter Aufruf der Funktion `getResult`

Das folgende Codebeispiel zeigt exemplarisch den fünften Aufruf der Funktion `getResult` durch die Web-Anwendung und setzt somit das Beispiel aus *Kapitel 4.6.1* fort. Zwischen diesem Aufruf der Funktion `getResult` und dem ursprünglichen Aufruf der Funktion `useID` liegen vier ergebnislose Aufrufe der Funktion `getResult`. Da nun die Ergebnisse beim eID-Server vorliegen, liefert der fünfte Aufruf die entsprechenden Rückgabewerte. Die Übertragung des akademischen Titels wurde abgewählt (siehe `OperationsAllowedByUser`). Es konnten daher nur der Vorname „Erika“ und der Nachname „Mustermann“ ausgelesen und übertragen werden. Wie die Web-Anwendung mit dieser Einschränkung umgeht, liegt im Ermessen des Diensteanbieters.

`getResultRequest`

```
<soapenv:Body>
      <eid:getResultRequest>
            <eid:Session>
                  <eid:ID>1234567890abcdef1234567890abcdef</eid:ID>
            </eid:Session>
            <eid:RequestCounter>5</eid:RequestCounter>
      </eid:getResultRequest>
</soapenv:Body>
```

`getResultResponse`

```
<soapenv:Body>
      <eid:getResultResponse>
            <eid:PersonalData>
                  <eid:FamilyNames>Mustermann</eid:FamilyNames>
                  <eid:GivenNames>Erika</eid:GivenNames>
            </eid:PersonalData>
            <dss:Result>
                  <ResultMajor>
                        http://www.bsi.bund.de/ecard/api/1.1/resultmajor#ok
                  </ResultMajor>
            </dss:Result>
            <eid:OperationsAllowedByUser>
                  <eid:DocumentType>PROHIBITED</eid:DocumentType>
                  <eid:IssuingState>PROHIBITED</eid:IssuingState>
                  <eid:GivenNames>ALLOWED</eid:GivenNames>
                  <eid:FamilyNames>ALLOWED</eid:FamilyNames>
                  <eid:ArtisticName>PROHIBITED</eid:ArtisticName>
                  <eid:AcademicTitle>PROHIBITED</eid:AcademicTitle>
                  <eid:DateOfBirth>PROHIBITED</eid:DateOfBirth>
                  <eid:PlaceOfBirth>PROHIBITED</eid:PlaceOfBirth>
                  <eid:PlaceOfResidence>PROHIBITED</eid:PlaceOfResidence>
                  <eid:RestrictedID>PROHIBITED</eid:RestrictedID>
                  <eid:AgeVerification>PROHIBITED</eid:AgeVerification>
                  <eid:PlaceVerification>PROHIBITED</eid:PlaceVerification>
            </eid:OperationsAllowedByUser>
      </eid:getResultResponse>
</soapenv:Body>
```

4.6.3 Beispielhafter Aufruf der Funktion `getServerInfo`

Das folgende Codebeispiel zeigt einen exemplarischen Aufruf der Funktion `getServerInfo`. Als Rückgabewerte werden neben der Version des eID-Servers (`ServerVersion`) auch die Berechtigungen des momentan auf dem eID-Server hinterlegten Terminal-Berechtigungszertifikates (`DocumentVerificationRights`) übermittelt.

`getServerInfoRequest`

```
<soapenv:Body>
      <getServerInfoRequest/>
</soapenv:Body>
```

`getServerInfoResponse`

```
<soapenv:Body>
      <eid:getServerInfoResponse>
            <eid:ServerVersion>
                  <eid:VersionString>Version 1.5 2011-12-06</eid:VersionString>
                  <eid:Major>1</eid:Major>
                  <eid:Minor>5</eid:Minor>
                  <eid:Bugfix>0</eid:Bugfix>
            </eid:ServerVersion>
            <eid:DocumentVerificationRights>
                  <eid:DocumentType>PROHIBITED</eid:DocumentType>
                  <eid:IssuingState>PROHIBITED</eid:IssuingState>
                  <eid:GivenNames>ALLOWED</eid:GivenNames>
                  <eid:FamilyNames>ALLOWED</eid:FamilyNames>
                  <eid:ArtisticName>PROHIBITED</eid:ArtisticName>
                  <eid:AcademicTitle>ALLOWED</eid:AcademicTitle>
                  <eid:DateOfBirth>PROHIBITED</eid:DateOfBirth>
                  <eid:PlaceOfBirth>PROHIBITED</eid:PlaceOfBirth>
                  <eid:PlaceOfResidence>PROHIBITED</eid:PlaceOfResidence>
                  <eid:RestrictedID>PROHIBITED</eid:RestrictedID>
                  <eid:AgeVerification>ALLOWED</eid:AgeVerification>
                  <eid:PlaceVerification>PROHIBITED</eid:PlaceVerification>
            </eid:DocumentVerificationRights>
      </eid:getServerInfoResponse>
</soapenv:Body>
```

5 Glossar

eID-Dokument

Das eID-Dokument ist als hoheitliches Dokument der physikalische Träger des elektronischen Identitätsnachweises.

eID-Server

Der eID-Server ist ein System, das aus Hard- und Softwarekomponenten besteht und die, in dieser Technischen Richtlinie spezifizierte, eID-Schnittstelle anbietet.

eID-Service

Der eID-Service ist eine Dienstleistung, die ein Diensteanbieter von einem Betreiber eines eID-Servers in Anspruch nehmen kann. Diese Dienstleistung kann der Diensteanbieter sowohl von einem Dritten als auch von einer eigenen Organisationseinheit nutzen.

AusweisApp

Die auf dem Computer des Bürgers eingesetzte clientseitige Instanz der eCard-API wird in diesem Dokument auch als AusweisApp bezeichnet. Die AusweisApp ist eine Implementierung der eCard-API, welche allen Bürgern kostenlos vom Bund zur Verfügung gestellt wird. Grundsätzlich kann auf dem Computer des Bürgers auch eine andere Implementierung des eCard-API-Frameworks zum Einsatz kommen.

Diensteanbieter

Der Diensteanbieter ist der Betreiber der Web-Anwendung und hat in der Regel ein vertragliches Verhältnis mit dem Betreiber des eID-Servers über die Nutzung der Dienste des eID-Servers. In dieser Technischen Richtlinie wird der Begriff Diensteanbieter entsprechend der Begriffsbestimmung unter §2 (3) des Personalausweisgesetzes verwendet.

Authentisierungszertifikat

Die Vergabestelle für Berechtigungszertifikate erstellt Bescheide, die einen Diensteanbieter berechtigen, die Dienste der DVCA (Berechtigungs-CA) zu nutzen. Diese Bescheide bilden die rechtliche Grundlage für die Vergabe von Authentisierungszertifikaten, mit denen Diensteanbieter ihre Terminal-Berechtigungszertifikate bei der DVCA abholen können.

Terminal-Berechtigungszertifikat

Die DVCA (Berechtigungs-CA) stellt kurzlebige Zertifikate aus, die Terminal-Berechtigungszertifikate genannt werden. Mit diesen Zertifikaten können die Funktionen und Daten des eID-Dokuments durch den Diensteanbieter genutzt werden.

Identity Provider

Ein Identity Provider ist ein vertrauenswürdiges System, das von Entitäten verwendet wird, um ihre Identität nachzuweisen. Eine Entität weist ihre Identität gegenüber dem Identity Provider nach. Der Identity Provider stellt diese Identitätsdaten über verschiedene Protokolle (z.B. LDAP, SAML, …) anderen Systemen zur Verfügung.

Literaturverzeichnis

BDSG: Bundesdatenschutzgesetz

BSI-100-2: BSI-Standard 100-2, IT-Grundschutz-Vorgehensweise

CP_CVCA-eID: BSI: Certificate Policy für die eID-Anwendung des ePA, Version 1.27

EAC 2: BSI TR-03110, Advanced Security Mechanisms for Machine Readable Travel Documents, Version 2.05

EAC-PKI Protocol: BSI TR-03129, PKIs for Machine Readable Travel Documents, Version 1.10

EAC-PKI'n ePA: BSI TR-03128, EAC-PKI'n für den elektronischen Personalausweis, Version 1.1

eCard des Bundes: BSI TR-03116-2, eCard-Projekte der Bundesregierung, Teil 2 - Hoheitliche Ausweisdokumente

eCard-API: BSI TR-03112, eCard-API-Framework, Version 1.1.1

ePA Architektur: BSI TR-03127, Architektur Elektronischer Personalausweis, Version 1.14

GSK: IT-Grundschutz-Kataloge, 11. Ergänzungslieferung

Krypto TR: BSI TR-02102, Kryptographische Verfahren: Empfehlungen und Schlüssellängen, in der jeweils aktuellen auf der BSI-Internetseite veröffentlichten Version

SAML Bindings: OASIS: Bindings for the OASIS Security Assertion Markup Language (SAML), Version 2.0

SAML Core: OASIS:Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML), Version 2.0

SAML Profiles: OASIS: Profiles for the OASIS Security Assertion Markup Language (SAML), Version 2.0

SAML Security: OASIS: Security and Privacy Considerations forthe OASIS Security Assertion MarkupLanguage (SAML), Version 2.0

XML-Type: W3C: XML Schema Part 2: Datatypes Second Edition

Anhang A: Verwendung mit SAML

1 Grundlagen

Im Kontext eines Identity Providers und auch im internationalen Zusammenhang ist SAML ein anerkannter Standard. An der funktionalen eID-Schnittstelle findet SAML jedoch keine Anwendung, da SAML zwei grundsätzlichen Prinzipien der eID-Funktion widerspricht.

Die Authentisierungsprotokolle der eID-Dokumente gehen immer von exakt zwei miteinander kommunizierenden Entitäten aus. Diese Entitäten sind einerseits der Bürger (mit seinem eID-Dokument) und andererseits der Diensteanbieter (mit seinem Terminal-Berechtigungszertifikat). Ein weiterer Grundsatz der Authentisierungsprotokolle ist, dass die Sicherheit der Authentisierung aus dem eID-Dokument und den im eID-Dokument implementierten Protokollen resultiert und nicht aus den umgebenden Infrastrukturkomponenten.

Die Protokolle von SAML gehen grundsätzlich von einem 3-Entitäten-Modell aus und basieren auf der Vertrauensbeziehung zwischen zwei Entitäten (Identity Provider und Service Provider).

Dennoch ist es möglich SAML so zu verwenden, dass die Grundsätze der Authentisierungsprotokolle gewahrt bleiben und eine sichere und gesetzeskonforme Authentisierung durchgeführt werden kann. Grundlage dafür ist, dass der Diensteanbieter einen Dritten mit der Authentisierung in seinem Namen beauftragen kann. Dieses Szenario wird unter anderem auch in *Kapitel 2.4.2* dieser Technischen Richtlinie skizziert.

In diesem Anhang zur TR eID-Server wird das Vorgehen zur Nutzung von SAML im Umfeld von eID-Dokumenten spezifiziert. Wie eingangs erwähnt, richtet sich dieser Teil der Technischen Richtlinie insbesondere an Implementierungen im Kontext eines Identity Providers, der neben der Authentisierung mit der eID-Funktion noch weitere Authentisierungsmethoden anbietet, und an Implementierungen im internationalen Zusammenhang. Dies erklärt auch, warum das hier beschriebene SAML-Profil deutlich modularer als die funktionale eID-Schnittstelle aufgebaut ist.

Im Folgenden wird erläutert, wie Single Sign-On (SSO) im Zusammenspiel mit dem eID-Server abläuft, welche Parteien daran beteiligt sind und welche Anforderungen durch ein SAML-Profil im Umfeld des eID-Servers erfüllt werden müssen.

1.1 Single Sign-On Szenario

In der untenstehenden *Abbildung 28* ist das grundlegende Szenario beim Single Sign-On im Kontext von SAML aufgezeigt. Die Abbildung zeigt dazu die drei beteiligten Entitäten *User*, *ServiceProvider* und *IdentityProvider*. Zwischen dem Identity Provider und dem Service Provider besteht eine *Vertrauensbeziehung (T)* die in der Abbildung entsprechend gekennzeichnet ist. Zwischen dem Service Provider und dem User findet im Allgemeinen zunächst eine *Identitätsselektion (IS)* statt, welche in diesem Szenario die eID-Funktion des Benutzers ist. Die einzelnen Rollen sind im Kontext von eID-Dokumenten wie folgt definiert:

- **Benutzer (User)**

  Der Benutzer möchte im Allgemeinen mittels eines Browsers (*UserAgent*) einen Dienst nutzen. Dazu muss er sich dem Diensteanbieter gegenüber authentisieren und eventuell weitere persönliche Daten zur Verfügung stellen. Hierfür verwendet er die eID-Funktion des eID-Dokuments und die notwendigen Komponenten (z.B. Kartenleser, AusweisApp)

- **Diensteanbieter (Service Provider)**

  Der Anbieter erlaubt nur authentisierten Benutzern den Zugriff auf seine Dienstleistungen. Zur Überprüfung der Benutzer ist er auf die Hilfe eines Identity Providers angewiesen, dem er vertraut und bei dem er die eID (Authentizität und persönliche Daten) des Benutzers anfragen kann.

- **eID-Service (Identity Provider)**

  Der eID-Service beantwortet als Identity Provider die Anfragen der Diensteanbieter. Er führt die gewünschte Authentisierung im Auftrag eines oder mehrerer Diensteanbieter durch und beschafft die angeforderten Daten des Benutzers. Hierzu verwaltet er auch die Terminal-Berechtigungszertifikate der jeweiligen Diensteanbieter.

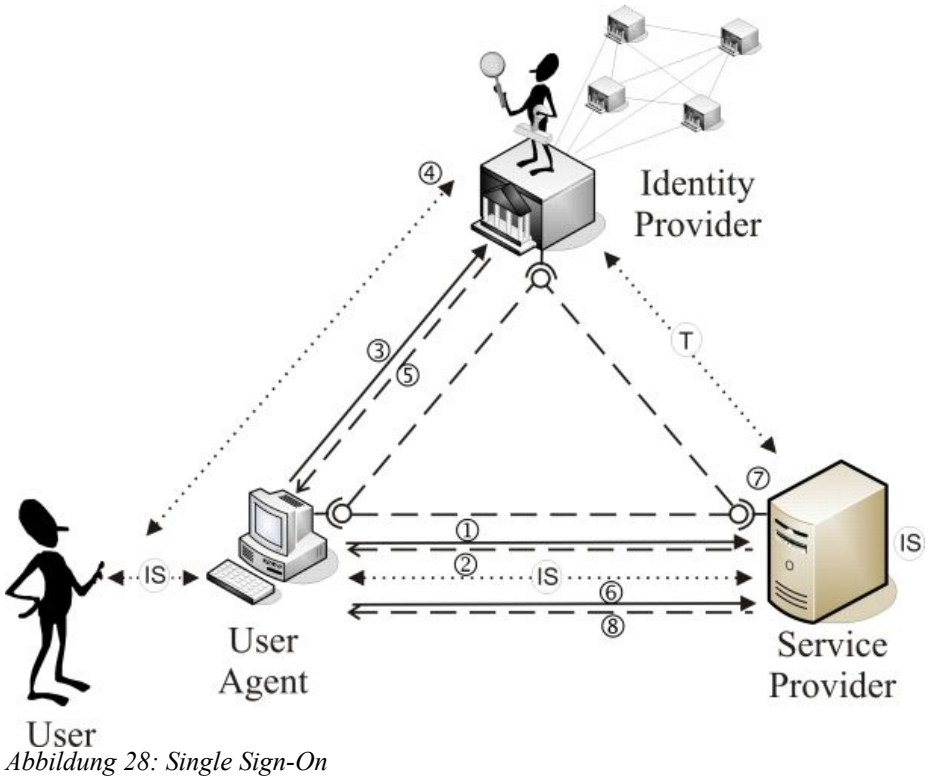

Abbildung 28: Single Sign-On

Die *Abbildung 28* zeigt neben den beteiligten Entitäten auch die einzelnen Schritte, welche im Rahmen eines Single Sign-On durchgeführt werden.

Der Protokollablauf besteht dementsprechend aus acht Schritten:

1. Der Benutzer *(User)* ruft mit seinem Browser *(User Agent)* den Dienst eines Anbieters *(Service Provider)* auf.
2. Der Diensteanbieter *(Service Provider)* fordert beim eID-Service *(Identity Provider)* eine Authentisierung des Benutzers und dessen benötigte persönliche Daten an. Diese Anforderung wird zunächst an den Benutzer geschickt.
3. Der Benutzer leitet die Anforderung des Diensteanbieters an den eID-Service weiter. Vom eID-Service erhält der Benutzer mit seinem Browser die Verbindungsdaten, welche an die clientseitige eCard-API weitergegeben werden und den eCard-API-Verbindungsaufbau (siehe *Schritt 4*) zum eID-Server einleiten.

4. Der eID-Server authentisiert den Benutzer unter Verwendung des in [eCard-API]*Teil 7 Abschnitt 3.3* definierten Protokollablaufs und liest die Benutzerdaten aus.
5. Der eID-Service bereitet seine Ergebnisse für den Diensteanbieter auf und sendet sie zunächst an den Benutzer.
6. Der Benutzer leitet die Antwort des eID-Service an den Diensteanbieter weiter.
7. Der Diensteanbieter prüft die Antwort und entscheidet anhand dessen über das weitere Vorgehen.
8. Der Diensteanbieter teilt dem Benutzer seine Entscheidung durch Erfüllung des Dienstes oder eine Fehlermeldung mit.

Aus dem oben beschriebenen Szenario ergeben sich einige grundsätzliche Anforderungen, die für das hier beschriebene SAML-Profil von Relevanz sind. Diese grundsätzlichen Anforderungen sind:

- Das Profil muss die Kommunikation der oben genannten drei Parteien beschreiben.
- Die Abfrage von eID-Authentisierung und eID-Daten muss als SAML-Nachricht formuliert werden (SAML Request). Hierbei muss es insbesondere möglich sein, einzelne eID-Datenfelder und Operationen auswählen und als „für den Geschäftsvorgang notwendig“ markieren zu können.
- Die Beantwortung einer Anforderung von eID-Authentisierung und eID-Daten muss als SAML-Nachricht formuliert werden (SAML Response).
- Es muss immer ein Nachrichtenpaar aus SAML Request und SAML Response geben. Eine unaufgeforderte SAML Assertion ist nicht gültig.
- Zwischen Diensteanbieter und eID-Server muss eine Ende-zu-Ende-Absicherung der Kommunikation implementiert werden. Integrität, Authentizität und Vertraulichkeit aller SAML-Nachrichten, sowohl Anfragen als auch Antworten, müssen sichergestellt werden.
- Die einzelnen Kommunikationskanäle müssen so miteinander verbunden werden, dass zu jeder Zeit eine Kommunikation nach dem oben beschriebenen Szenario gewährleistet ist.
- Der eID-Service-Provider und der Diensteanbieter sind sich der in [SAML Security] beschriebenen Gefährdungen bewusst und ergreifen Maßnahmen die den dort genannten Gefährdungen entgegenwirken.

Ausgehend von den in diesem Kapitel beschriebenen Grundlagen und ausgehend von dem Single Sign-On Szenario wird im Folgenden ein konkretes Profil definiert.

2 Profilübersicht

In diesem Kapitel wird ein SAML-Profil beschrieben, dass die Grundlagen aus dem vorherigen Kapitel umsetzt. Wie in den Grundlagen bereits beschrieben, basiert dieses Profil auf dem „Web Browser SSO Profile“ gemäß [SAML Profiles]. Die in dem Profil interagierenden Rollen entsprechen den bereits in *Kapitel 1.1* definierten Rollen.

2.1 Protokollablauf

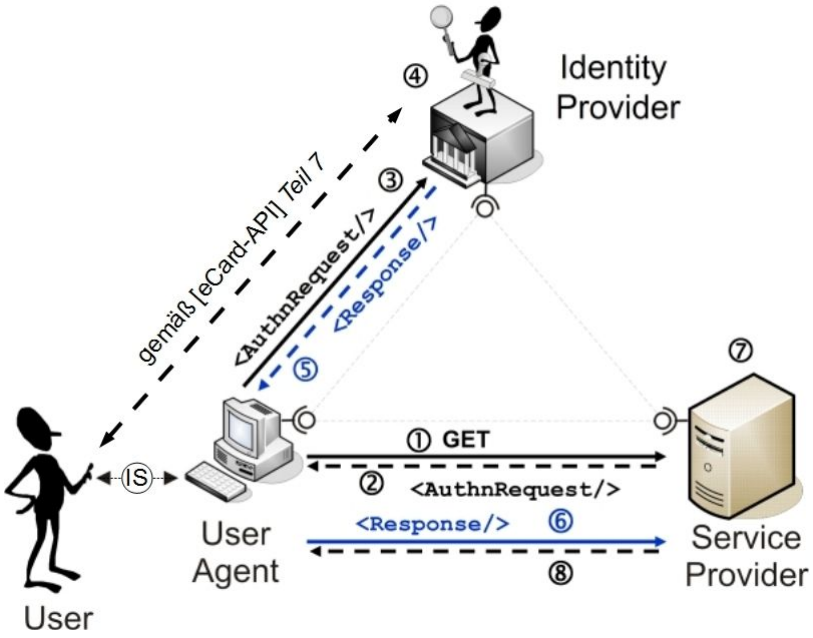

Abbildung 29: Protokollablauf

Mittels SAML wird das in *Kapitel 1.1* beschriebene Single Sign-On abgebildet. Wie in *Abbildung 29* zu erkennen ist, besteht auch hier der Protokollablauf aus acht Schritten:

1. Der Benutzer *(User)* ruft mit seinem Browser *(User Agent)* den Dienst eines Anbieters (Service Provider) auf. Dies geschieht üblicherweise per `HTTP-GET`.
2. Der Diensteanbieter *(Service Provider)* formuliert eine Authentisierungsanforderung als `<AuthnRequest>` in einer SAML-Nachricht an den eID-Service *(Identity Provider)*, die an den Browser des Benutzers geschickt wird.
3. Der Browser des Benutzers leitet die SAML-Nachricht des Diensteanbieters gemäß des in *Kapitel 2.2* geforderten SAML-Bindings an den eID-Service weiter. Der eID-Service wiederum generiert nun in seiner Antwort an den Browser das gemäß [eCard-API] *Teil 7 Abschnitt 2.3.1* gefordert MIME-Typ-Objekt zum Aufruf der clientseitigen eCard-API durch das Browser-Plugin. Neben den Verbindungsdaten für die eCard-API-Kommunikation (siehe *Schritt 4*) enthält das Objekt auch die Rücksprungadresse, welche der Browser nach der Authentisierung aufruft.

4. Die durch das Browser-Plugin respektive das MIME-Typ-Objekt aufgerufene clientseitige eCard-API stellt nun die Verbindung zum eID-Server her. Dieser authentisiert den Benutzer und liest die von ihm freigegebenen Daten aus. Um die ursprüngliche Verbindung des Diensteanbieters zum Bürger an die Verbindung der clientseitigen eCard-API zum eID-Server zu binden wird der Pre-Shared Key zum Verbindungsaufbau genutzt. Der an dieser Stelle stattfindende eCard-API-Protokollablauf ist in [eCard-API]*Teil 7 Abschnitt 3.3* spezifiziert.
5. Der eID-Service formuliert seine Ergebnisse als SAML-Assertions innerhalb einer `<Response>` und sendet diese SAML-Nachricht zunächst an den Benutzer.
6. Der Benutzer leitet die SAML-Nachricht des eID-Service gemäß des in *Kapitel 2.2* geforderten SAML-Bindings an den Diensteanbieter weiter.
7. Der Diensteanbieter prüft die SAML-Response und entscheidet anhand der enthaltenen SAML-Assertions über das weitere Vorgehen.
8. Im Erfolgsfall gewährt der Diensteanbieter Zugriff auf den ursprünglich angeforderten Dienst.

2.2 Binding

Die Abbildung von SAML-Nachrichten auf andere Kommunikationsmechanismen und Protokolle wird anhand von SAML-Bindings festgelegt.

In diesem Profil wird für die Übermittlung von SAML-Request und SAML-Response ausschließlich das POST-Binding gemäß [SAML Bindings]*Kapitel 3.5* verwendet, um eventuellen Beschränkungen anderer Bindings vorzubeugen. Insbesondere das Redirect-Binding darf nicht verwendet werden, da durch Verschlüsselung und Signatur der Nachrichten deren Größe zu Problemen beim Transport führen kann.

3 Profildetails

3.1 Attribute

In der folgenden Tabelle sind die Namensbezeichner der SAML-Attribute und die Zuordnung zu den Datengruppen und Funktionen der eID-Anwendung des eID-Dokuments gegeben (siehe *Anhang E „eID- Application"* in [EAC 2]). Eb enfalls angegeben ist der XML-Datentyp des jeweiligen Attributwertes, wenn dieser innerhalb einer Authentisierungsantwort bzw. innerhalb einer Authentisierungsanfrage übertragen wird. Der Datentyp ist entweder ein build -in XML Schema Datent yp [XML-Type] , i m Folgenden gekennzeichnet durch den Präfix „ `xs :` " oder ein Datentyp aus der zu dieser Technischen Richtlinie gehörenden XSD-Schemadatei TR-03130_TR-eID-Server.xsd, dargestellt mit dem Präfix „ `eid:` ".
Die Tabelle enthält alle für eID-Dokumente relevanten Datengruppen. Diese Liste ist als nicht abschließend zu betrachten, d.h. weitere Attribute können durch individuelle Vereinbarung der Kommunikationspartner Verwendung finden.

Attributname	***Datengruppe oder Funktion***	***Inhalt***	***Typ des Attributwertes in Authentisie-rungsantwort***	***Typ des Attributwertes in Authentisierung sanfrage***
DocumentType	DG1	Dokumententyp, z.B. „ID" für nPA	eid:DocumentTyp e	-
IssuingState	DG2	Ausgebender Staat, „D" für Deutschland	eid:ICAOCountry	-
GivenNames	DG4	Vornamen	xs:string	-
FamilyNames	DG5	Familiennamen	xs:string	-
ArtisticName	DG6	Ordensname/ Künstlername	xs:string	-
AcademicTitle	DG7	Doktorgrad	xs:string	-
DateOfBirth	DG8	Geburtsdatum	eid:GeneralDateT ype	-
PlaceOfBirth	DG9	Geburtsort	eid:GeneralPlace Type	-
PlaceOfResidence	DG17	Adresse	eid:GeneralPlace Type	-
RestrictedId	Restricted Identification	Sektorspezifische Kennung (Pseudonym)	eid:RestrictedIDT ype	-

Attributname	*Datengruppe oder Funktion*	*Inhalt*	*Typ des Attributwertes in Authentisie-rungsantwort*	*Typ des Attributwertes in Authentisierung sanfrage*
CommunityIdVeri fication	Community ID Verification	Ergebnis bzw. Anfragewert der Vergleichsfunktion Wohnortabfrage	eid: CommunityIdVeri ficationResultTyp e	xs:string 1)
AgeVerification	Age Verification	Ergebnis bzw. Anfragewert der Vergleichsfunktion Altersüberprüfung	eid: AgeVerificationR esultType	xs:unsignedShort
DocumentValidity	Gültigkeits-prüfung 2)	Ergebnis der Gültigkeitsprüfung des Dokumentes	eid:DocumentVali dityResultType	3)

Tabelle 11: Liste der SAML-Attribute

Legende:

- Für dieses Attribut kann innerhalb einer Authentisierungsanfrage kein Attributvergleichswert angegeben werden (nur möglich bei Vergleichsfunktionen und Pre-Shared Key).

1) Die Wohnort-ID besteht aus 14 Ziffern entsprechend [ePA Architektur], jedoch alphanumerisch als String codiert (und nicht packed BCD).

2) Die Gültigkeitsprüfung des Dokumentes wird immer durchgeführt, unabhängig von den Rechten im Berechtigungszertifikat bzw. den angeforderten Attributen. Alle anderen Attribute setzen das jeweilige Recht im Berechtigungszertifikat voraus.

3) Diese Attribute dürfen nicht in einer Authentisierungsanfrage angegeben werden.

3.2 Erweiterte Datentypen

Im Folgenden werden die anwendungsspezifischen Datentypen beschrieben, die in der zu dieser Technischen Richtlinie gehörenden XSD-Datei `TR-03130_TR-eID-Server.xsd` definiert sind.

3.2.1 Datentyp `CommunityIdVerificationResultType`

Abbildung 30: Datentyp CommunityIdVerificationResultType

Dieser Datentyp stellt das Ergebnis der Vergleichsfunktion Wohnortabfrage dar. Das Element `Request` enthält den angefragten Vergleichswert (Wohnort-ID) und das Element `Result` das Ergebnis der Vergleichsfunktion.

3.2.2 Datentyp `AgeVerificationResultType`

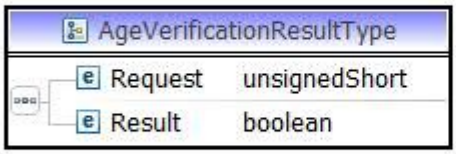

Abbildung 31: Datentyp AgeVerificationResultType

Dieser Datentyp stellt das Ergebnis der Vergleichsfunktion Altersüberprüfung dar. Das Element `Request` enthält den angefragten Vergleichswert (Alter) und das Element `Result` das Ergebnis der Vergleichsfunktion.

3.2.3 Datentyp `DocumentValidityResultType`

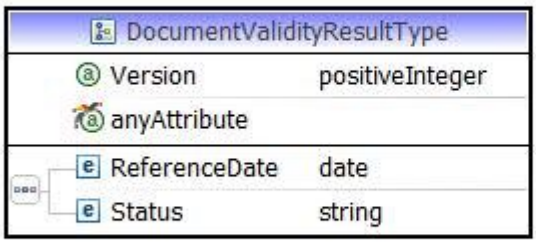

Abbildung 32: Datentyp DocumentValidityResultType

Dieser Datentyp enthält allgemeine Informationen über die gelieferten Daten und das Ergebnis der Dokumentenprüfung in den folgenden Feldern:

Attribut/Element	***Beschreibung***
`Version`	Gibt die Version der Datendarstellung der Authentisierungsantwort und somit der Schnittstellenversion auf Seite des Identity Providers an. Für gewöhnlich ist hier die Version 1 zu verwenden. Diese Versionierung bezieht sich lediglich auf den Inhalt und die Semantik der Daten der Authentisierungsantwort, eine Versionierung des Schemas selbst wird über andere Mechanismen durchgeführt (z.B. Schema-Namensraum).
`ReferenceDate`	Das Datum mit dem die Gültigkeitsüberprüfung (Ablaufdatum und Sperrstatus) des Dokumentes durchgeführt wurde. Grundsätzlich entspricht dies dem aktuellen Tagesdatum, ist jedoch von Relevanz in der zeitlichen Nähe des Tageswechsel. Entsprechend *Abschnitt D.2.1.3* der [EAC 2] beziehen sich alle Datumsangaben auf die Zeitzone GMT.
`Status`	Der Ergebnisstatus der Dokumentengültigkeitsprüfung mit den folgenden verpflichtend möglichen Werten: `valid:` Dokument ist gültig `failed:` Prüfung fehlgeschlagen Optional können auch die folgenden Werte vom Identity Provider implementiert werden: `expired:` Dokument ist abgelaufen `revoked:` Dokument ist gesperrt `notAuthentic:` Dokument ist nicht authentisch Ist das Dokument nicht gültig (der `Status` ist ungleich `valid`), werden keine der aus dem eID-Dokument auszulesenden Attribute übermittelt.

Tabelle 12: Felder des Datentyps DocumentValidityResultType

3.2.4 Datentyp `RequestedAttributesType`

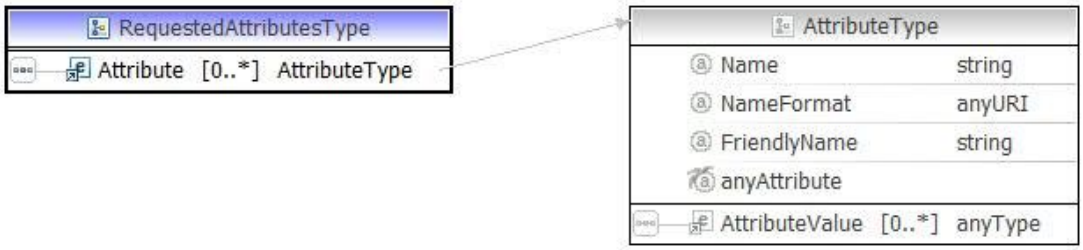

Abbildung 33: Datentyp RequestedAttributesType

Dieser Datentyp beinhaltet die angeforderten Attribute und die Anfragewerte für Vergleichsfunktionen. Er kann mehrere Elemente `Attribute` vom in [SAML Core] spezifizierten Datentyp `AttributeType` enthalten. Die Verwendung der Bezeichner aus *Kapitel 3.1* für das Attribut `Name` ist verpflichtend. Darüber hinaus können jedoch auch weitere Attribute vom Identity Provider definiert werden. Optional kann jedes `Attribute` Element auch ein Attribut `Required Attribute` enthalten, um Pflichtfelder zu kennzeichnen. Dieses Attribut ist durch das Schema definiert (siehe *Kapitel 3.4.1*). Fehlt das Attribut, so ist von einem Pflichtfeld auszugehen.

3.2.5 Datentyp `AuthnRequestExtensionType`

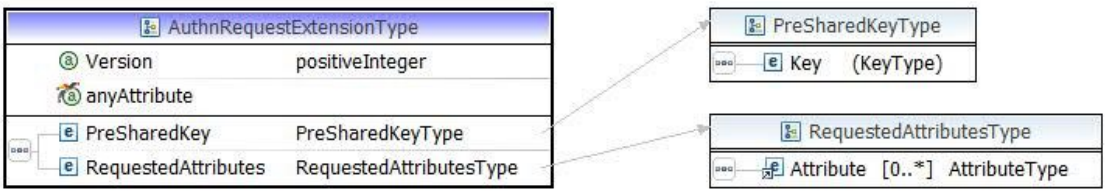

Abbildung 34: Datentyp AuthnRequestExtensionType

Dieser Datentyp bildet die Grundlage für die Authentisierungsanfrage und enthält neben der `Version` auch Elemente für den `Pre-Shared Key` und die auszulesenden Attribute.

Attribut/Element	*Beschreibung*
`Version`	Gibt die `Version` der Datendarstellung der Authentisierungsanfrage und somit der Schnittstellenversion auf Seite des Diensteanbieters an. Für gewöhnlich ist hier die Version 1 zu verwenden. Diese Versionierung bezieht sich lediglich auf den Inhalt und die Semantik der Daten der Authentisierungsanfrage. Eine Versionierung des Schemas selbst wird über andere Mechanismen durchgeführt (z.B. Schema-Namensraum).
`PreSharedKey`	Enthält den `Pre-Shared Key`, welcher vom Identity Provider für die Kanalbindung der eCard-API-Kommunikation an die Kommunikation zwischen Web-Anwendung und Browser genutzt werden muss.
`Requested Attributes`	Die Liste der Attribute, die angefordert werden bzw. der Attribute, für die weitere Informationen innerhalb der Authentisierungsanfrage übertragen werden.

Tabelle 13: Felder des Datentyps AuthnRequestExtensionType

3.3 Zusätzlich definierte Elemente

Die folgenden Elemente werden als SAML-Komponenten verwendet (siehe *Kapitel 3.5*) und sind ebenfalls in der XSD-Datei `TR-03130_TR-eID-Server.xsd` definiert worden.

3.3.1 Element AuthnRequestExtension

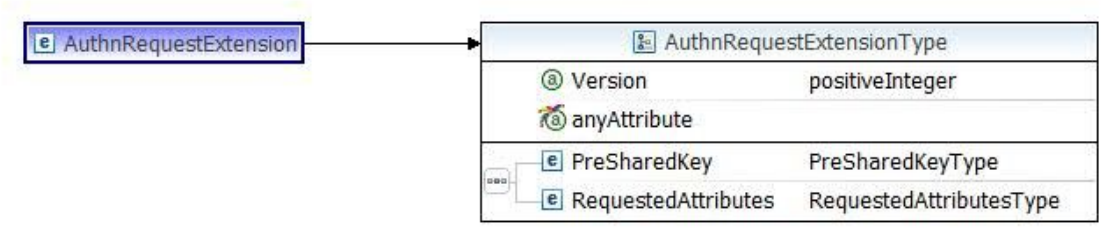

Abbildung 35: Element AuthnRequestExtension

Das Element `AuthnRequestExtension` ist von dem, in diesem Schema definierten, Datentyp `AuthnRequestExtensionType` (siehe *Kapitel 3.2.5*).

3.3.2 Element EncryptedAuthnRequestExtension

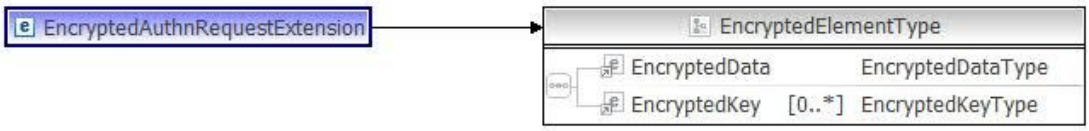

Abbildung 36: Element EncryptedAuthnRequestExtension

Das Element `EncryptedAuthnRequestExtension` ist von dem, in [SAML Core] definierten, Datentyp `EncryptedElementType` und beinhaltet die zuvor definierte `AuthnRequestExtension` (siehe *Kapitel 3.3.1*) in verschlüsselter Form.

3.4 Zusätzlich definierte Attribute

Die folgenden Attribute können von SAML-Komponenten verwendet werden und sind ebenfalls in der XSD-Datei `TR-03130_TR-eID-Server.xsd` definiert worden.

3.4.1 Attribut `RequiredAttribute`

Das Attribut `RequiredAttribute` ist vom Datentyp `boolean` und dient dazu Attribute als verpflichtend (`true`) bzw. optional (`false`) zu kennzeichnen. Der Standardwert für dieses Attribut ist verpflichtend (`true`).

3.5 SAML-Komponenten

Die folgenden Kapitel definieren die einzelnen SAML-Komponenten. Die Grundlage für diese Komponenten bilden das in *Kapitel 2* beschriebene Szenario und die in *Kapitel 3.1-3.3* definierten Datentypen und Elemente.

3.5.1 AuthnRequest

Der SAML AuthnRequest muss folgende XML-Elemente (`<element>`) und XML-Attribute (*`attribut`*) enthalten. Soweit es nicht anders angegeben wird, sind somit alle hier dargestellten Angaben Pflichtfelder. Zudem müssen die im [SAML Core] definierten Prüfungen auf Seiten des Empfängers durchgeführt werden.

<AuthnRequest>	
Version	„2.0“ gemäß [SAML Core].
ID	zufällige ID gemäß [SAML Core].
IssueInstant	Zeitstempel gemäß [SAML Core].
Destination	URL des eID-Servers, wohin der AuthnRequest versandt wird.
ProtocolBinding	„urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST“. Das Attribut ist optional und muss bei Verwendung den oben festgelegten Wert haben.
AssertionConsumerServiceURL	URL des Diensteanbieters, zu dem die Response versandt wird. Das Attribut ist optional, wenn dem eID-Server bereits im Vorfeld eine Zieladresse für die Response bekannt ist. Wenn der Diensteanbieter von dieser Vorgabe abweichen möchte, so wird dieses Attribut mit der gewünschten URL gefüllt.
<Issuer>	URL des Diensteanbieters, anhand derer er vom eID-Server identifiziert werden kann.
<Signature>	Gemäß *Kapitel 3.6.2*.
<Extensions>	
<EncryptedAuthnRequestExtension>	Dieses Element (siehe *Kapitel 3.3.2*) enthält die verschlüsselte Extension <AuthnRequestExtension> mit der Datenanforderung und dem PSK des Diensteanbieters. Das Element ist vom Typ EncryptedElementType, welcher in [SAML Core] spezifiziert ist.

Tabelle 14: Elemente und Attribute des AuthnRequest

3.5.2 AuthnRequestExtension

Die verschlüsselte Extension innerhalb des SAML Request muss folgende XML-Elemente (<element>) und XML-Attribute (*attribut*) enthalten. Soweit es nicht anders angegeben wird, sind somit alle hier dargestellten Angaben Pflichtfelder.

<AuthnRequestExtension>	Dieses Element (siehe *Kapitel 3.3.1*) ist vom Datentyp AuthnRequestExtensionType (siehe *Kapitel 3.2.5*).
Version	„1“ als Versionsnummer dieser Extension.
<PreSharedKey>	
<Key>	Dieses Element enthält den PSK, welcher zur Kanalbindung (siehe *Kapitel 3.6.3*) für die eCard-API-Kommunikation dient.
<RequestedAttributes>	Dieses Element umschließt eine Menge von <Attribute>-Elementen gemäß [SAML Core]. Es beinhaltet Informationen über die vom Diensteanbieter neben der Authentisierung zusätzlich angeforderten persönlichen Daten des Benutzers.
<Attribute>	Benötigt die Abfrage der persönlichen Information eine Eingabe, z.B. bei der Altersverifikation, so wird diese in einem <AttributeValue>-Unterelement notiert.
Name	Bezeichner der angeforderten Information gemäß [SAML Core]. Für die Attribute der eID-Funktion sind die Bezeichner der in *Kapitel 3.1* definierten Liste von Attributen zu verwenden.
Required	Optionale Angabe, ob das Attribut zwingend für den jeweiligen Geschäftsprozess benötigt wird. Standardmäßig wird von dem Wert „true“ ausgegangen. (In [SAML Core] ist die Verwendung anwendungs-spezifischer XML-Attribute vorgesehen.)

Tabelle 15: Elemente und Attribute der AuthnRequestExtension

3.5.3 Response

Die SAML Response muss folgende XML-Elemente (<element>) und XML-Attribute (*attribut*) enthalten. Soweit es nicht anders angegeben wird, sind somit alle hier dargestellten Angaben Pflichtfelder. Zudem müssen die im [SAML Core] definierten Prüfungen auf Seiten des Empfängers durchgeführt werden.

<Response>	
Version	„2.0“ gemäß [SAML Core].
ID	zufällige ID gemäß [SAML Core].
InResponseTo	ID des beantworteten Requests gemäß [SAML Core].
IssueInstant	Zeitstempel gemäß [SAML Core].
Destination	URL des Diensteanbieters, wohin die Response versandt wird. Hier ist das Attribut *AssertionConsumerServiceURL* aus dem Request zu berücksichtigen.
<Issuer>	URL des eID-Service, anhand derer er vom Diensteanbieter identifiziert werden kann.
<Signature>	gemäß *Kapitel 3.6.2*.
<Status>	
<StatusCode>	
Value	Primärer StatusCode, gemäß [SAML Core]. Bei Erfolg wird „urn:oasis:names:tc:SAML:2.0:status:Success“ eingetragen, alle anderen Werte kennzeichnen einen Fehlerfall.
<StatusCode>	
Value	Untergeordneter StatusCode, gemäß [SAML Core]. Dieses Element ist optional und darf nur im Fehlerfall verwendet werden. Die konkreten Werte sind anwendungsspezifisch und sollen dem Diensteanbieter ermöglichen, eine Klartextmeldung an den Benutzer zu generieren.
<StatusMessage>	Dieses Element ist optional und darf nur im Fehlerfall verwendet werden. Es beinhaltet einen unstrukturierten Informationstext für die Protokollierung beim Diensteanbieter und dient daher nicht zur Anzeige gegenüber dem Benutzer.

<EncryptedAssertion>	Dieses Element enthält die vom eID-Service erstellte, verschlüsselte Assertion.

Tabelle 16: Elemente und Attribute der Response

3.5.4 Assertion

Die verschlüsselte Assertion innerhalb der SAML Response muss folgende XML-Elemente (<element>) und XML-Attribute (*attribut*) enthalten. Soweit es nicht anders angegeben wird, sind somit alle hier dargestellten Angaben Pflichtfelder. Zudem müssen die im [SAML Core] definierten Prüfungen auf Seiten des Empfängers durchgeführt werden.

<Assertion>	
Version	„2.0“ gemäß [SAML Core].
ID	zufällige ID gemäß [SAML Core].
IssueInstant	Zeitstempel gemäß [SAML Core].
<Issuer>	URL des eID-Service, anhand derer er vom Diensteanbieter identifiziert werden kann.
<Subject>	
<NameID>	Das Element beinhaltet eine zufällige ID gemäß [SAML Core].
Format	"urn:oasis:names:tc:SAML:2.0:nameid-format:transient"
<SubjectConfirmation>	
Method	"urn:oasis:names:tc:SAML:2.0:cm:bearer"
<SubjectConfirmationData>	
Address	IP-Adresse des Benutzers.
InResponseTo	ID des beantworteten Requests gemäß [SAML Core].
NotOnOrAfter	Zeitpunkt fünf Minuten nach der Erstellung.
Recipient	URL des Diensteanbieters, wohin die Response versandt wird.

<Conditions>	
<AudienceRestriction>	
<Audience>	Das Element beinhaltet die URL des Diensteanbieters, anhand derer er vom eID-Server identifiziert werden kann.
<OneTimeUse>	Dieses Element ist verpflichtend zu verwenden, da die Authentisierung mit der eID-Funktion ausschließlich zum Zeitpunkt der Authentisierung gültig ist.
<AuthnStatement>	
<AuthnContext>	
<AuthnContextDeclRef>	Der Inhalt dieses Elements ist auf einen konstanten Wert festgelegt: „urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI“
<AttributeStatement>	Das Element beinhaltet eine Menge von <Attribute>-Elementen mit <AttributeValue>-Unterelementen gemäß den angeforderten persönlichen Daten des Benutzers. Auch hier ist für die Attribute der eID-Funktion die Verwendung der in *Kapitel 3.1* definierten Bezeichner verpflichtend.

Tabelle 17: Elemente und Attribute der Assertion

3.6 Sicherheitsmaßnahmen

In diesem Kapitel werden Sicherheitsmaßnahmen beschrieben, die im SAML-Kontext von den Kommunikationspartnern ergriffen werden müssen. Diese Maßnahmen resultieren insbesondere aus der Vertrauensbeziehung zwischen dem Identity Provider (eID-Service) und dem Service Provider (Diensteanbieter). Aufgrund dessen werden SAML-Request und SAML-Response verschlüsselt und signiert, um Authentizität und Vertraulichkeit der Nachrichten zu gewährleisten.

Zusätzlich wird der Transport der SAML-Nachrichten gemäß den Empfehlungen aus [SAML Core] durch SSL/TLS-Verbindungen zwischen Diensteanbieter und Benutzer sowie eID-Server und Benutzer abgesichert. Eine Authentisierung findet hierbei jedoch lediglich anhand der Serverzertifikate statt.

Die hier beschriebenen Sicherheitsmaßnahmen sind nicht der abschließende Nachweis über die Einhaltung der CertificatePolicy für die Terminal-Berechtigungszertifikate. Für diesen Nachweis

wird vom Diensteanbieter ein separates Sicherheitskonzept gefordert, bei dessen Erstellung der *Anhang C* dieser Technischen Richtlinie als Grundlage dient.

Den in [SAML Security] genannten Gefährdungen ist mit angemessenen und ebenfalls in diesem Dokument genannten Maßnahmen entgegen zuwirken. Auf Grund des in diesem Anhang spezifizierten Profils sind die *Kapitel 6.4 "HTTP Redirect/POST Binding" und 7.1.1 "SSO Profile"* von besonderer Bedeutung.

3.6.1 Verschlüsselung

Bei der Erstellung des Vertrauensverhältnisses zwischen Diensteanbieter und eID-Service tauschen beide Parteien auf sicherem Wege die notwendigen Schlüssel für die Verschlüsselung der SAML-Nachrichten aus. Jede Partei erstellt ein individuelles Verschlüsselungsschlüsselpaar, welches sich vom Signaturschlüsselpaar unterscheiden muss. Hierbei sind insbesondere die Vorgaben aus [Krypto TR] zu beachten. Für den Austausch wird die Verwendung von jeweils selbstsignierten X.509-Zertifikaten als Transportbehälter empfohlen.

Beim SAML-Request werden die Extensions, welche die Angabe der angeforderten eID-Daten und den PSK beinhalten, verschlüsselt. Das jeweilige XML-Element wird als Ganzes verschlüsselt und innerhalb eines `<EncryptedData>`-Elements in das Element `EncryptedAuthnRequestExtension` (siehe *Kapitel 3.3.2* bzw. *Kapitel 3.5.1*) notiert.

Beim SAML-Response wird für die Verschlüsselung der Mechanismus „`EncryptedAssertion`“ aus [SAML Core] *Abschnitt 2.3.4* verwendet.

Bei beiden Nachrichten werden die ursprünglichen Verschlüsselungsschlüssel, welche bei der Herstellung des Vertrauensverhältnisses ausgehandelt wurden, nicht übermittelt. Falls es die verwendete Verschlüsselungstechnik erfordert, z.B. bei hybrider Verschlüsselung, werden flüchtige bzw. temporäre Schlüssel entsprechend gesichert übermittelt.

Die Verwendung statischer symmetrischer Schlüssel oder rein symmetrischer Verschlüsselungsverfahren ist nicht möglich, da die verschlüsselten Nachrichten und die darin enthaltenen Elemente hochgradig strukturiert sind. Die Verwendung hybrider Verschlüsselungsverfahren mit zufälligen Schlüsseln wird empfohlen.

Übergeordnete Anforderung ist das Erreichen des in *Anhang C Kapitel 4.1* geforderten Sicherheitsniveaus der Kommunikationsschnittstelle.

3.6.2 Signatur

Bei der Erstellung des Vertrauensverhältnisses zwischen Diensteanbieter und eID-Server tauschen beide Parteien auf sicherem Wege die notwendigen Schlüssel für die Signatur der SAML-Nachrichten aus. Jede Partei erstellt ein individuelles Signaturschlüsselpaar, welches sich vom Verschlüsselungsschlüsselpaar unterscheiden muss. Hierbei sind insbesondere die Vorgaben aus [Krypto TR] zu beachten. Für den Austausch wird die Verwendung von jeweils selbstsignierten X.509-Zertifikaten als Transportbehälter empfohlen.

SAML sieht bereits standardmäßig XML-Elemente für die Signatur einer SAML-Nachricht vor, welche auch innerhalb dieses Profils genutzt werden. Hierbei wird gemäß [SAML Core] die SAML-Nachricht als Ganzes signiert, um insbesondere Attacken vorzubeugen, die die besondere Struktur von XML-Dokumenten ausnutzen.

Bei beiden Nachrichten werden die ursprünglichen Signaturschlüssel, welche bei der Herstellung des Vertrauensverhältnisses ausgehandelt wurden, nicht übermittelt. Die Signaturen der Nachrichten müssen immer vom jeweiligen Empfänger überprüft werden.

Übergeordnete Anforderung ist das Erreichen des in *Anhang C Kapitel 4.1*geforderten Sicherheitsniveaus der Kommunikationsschnittstelle.

3.6.3 Kanalbindung

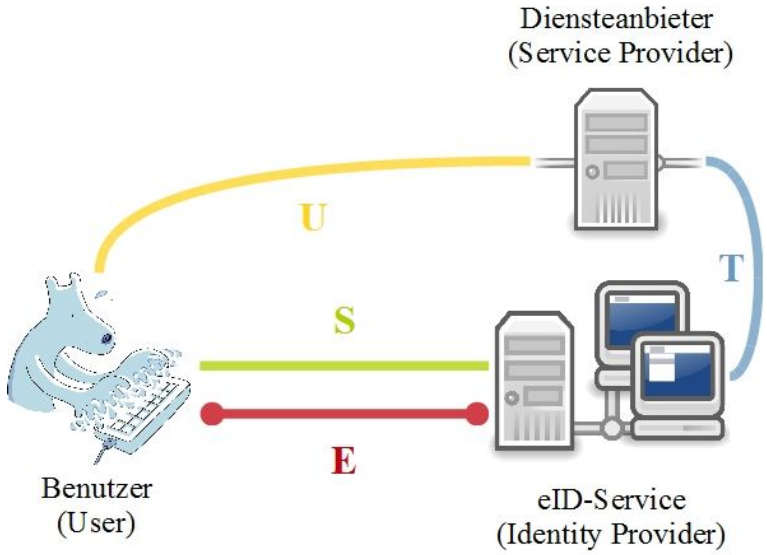

Abbildung 37: Kommunikationskanäle

Um die sichere Nutzung der eID-Funktion mit SAML zu gewährleisten, ist es erforderlich, dass die Rollen, die in diesem Szenario beschrieben werden (siehe *Kapitel 1.1* und *Abbildung 37*) über den gesamten Protokollablauf eingehalten werden. Auf Grund der bestehenden Vertrauensbeziehung zwischen dem eID-Service und dem Diensteanbieter ist die Authentizität dieses Kommunikationskanals (siehe *Kanal T* in *Abbildung 37* (Diensteanbieter <=> eID-Service)) zu jeder Zeit sichergestellt.

Die *Kanäle U* (Benutzer <=> Diensteanbieter) und *S* (Benutzer <=> eID-Service) werden mit Hilfe der im Terminal-Berechtigungszertifikat enthaltenen Hashes als authentisch erkannt. Die Korrektheit der Rollen ist damit jedoch noch nicht gewährleistet, da die Kanäle kryptographisch mit Hilfe der Hashes nur in eine Richtung verbunden sind. Es ist daher notwendig, dass die *Kanäle U, S* und *E* mit einem Pre-Shared Key verknüpft werden. Der in den *Abbildungen 37*und*38* rot dargestellte *Kanal E* repräsentiert die Kommunikation gemäß [eCard-API] *Teil 7 Abschnitt 3.3* respektive [EAC 2] und ist somit der Kern der Authentisierung. Beide Maßnahmen (Hashes und Pre-Shared Key) werden technisch ausführlich in [eCard-API] *Teil 7 Abschnitt 2.3.2 und 3.3.10* beschrieben.

Die in der folgenden *Abbildung 38* dargestellten Tunnel (*U*, *S* und *E*) beziehen sich auf die pro Authentisierung dynamisch aufgebauten Kanäle. Konkrete Flüsse von Informationen werden durch Linien dargestellt, welche die einzelnen Entitäten miteinander verbinden. Diese Verbindungen sind mit Verweisen auf die Spezifikationen versehen, gemäß derer Sie implementiert werden müssen.

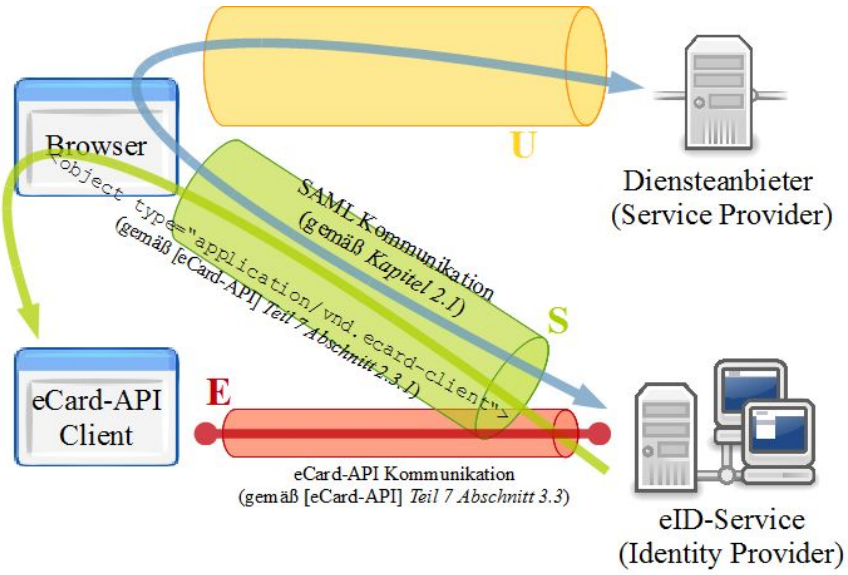

Abbildung 38: Kanalbindung im SAML-Kontext

Wie in *Abbildung 38* dargestellt sendet der Diensteanbieter den für eine spezifische Sitzung generierten Pre-Shared Key mit dem `AuthnRequest` (gemäß *Kapitel 2.1*) an den eID-Service. Der eID-Service generiert nun das MIME-Type-Objekt (gemäß [eCard-API] *Teil 7 Abschnitt 2.3.1*) und initiiert über den Browser des Bürgers den eCard-API Client. Der eCard-API Client kann nur dann einen gültigen Kommunikationskanal (*E*) zum eID-Service aufbauen, wenn beide eCard-API Komponenten (Client und Server) den gleichen PSK verwenden. Da der PSK über den vertraulichen *Kanal T* vom Diensteanbieter an den eID-Service gesendet wurde sind die *Kanäle U* und *E,* sowie der zusätzlich genutzte *Kanal S*, kryptographisch miteinander verbunden.

Da der PSK für den Aufbau der sicheren eCard-API-Kommunikation ausschließlich der Kanalbindung dient (siehe [eCard-API] *Abschnitt 2.3.6.1*), sind die Anforderungen an den PSK gering. Für Datenformat und Länge gelten dabei die Anforderungen aus *Kapitel 4.3.11*. Außerdem muss der PSK schwer vorhersehbar generiert werden. Es wird empfohlen den PSK als Zufallszahl entsprechend den Vorgaben der [Krypto TR] zu bilden.

3.7 Beispielhafte SAML-Nachrichten

Die folgenden Codebeispiele zeigen den exemplarischen Ablauf einer Authentisierung unter Benutzung der in diesem Anhang spezifizierten SAML-Komponenten. Der besseren Lesbarkeit halber enthalten diese Codebeispiele Zeilenumbrüche und Einrückungen, die so nicht in tatsächlichen Nachrichten auftauchen müssen.

3.7.1 AuthnRequest

Dieses Codebeispiel zeigt exemplarisch einen `AuthnRequest` gemäß *Kapitel 3.5.1*.

```
<?xml version="1.0" encoding="UTF-8"?>
<samlp:AuthnRequest
AssertionConsumerServiceURL="https://www.service-provider.de/processResponse"
Destination="https://www.identity-provider.de/processRequest"
ID="Request_1234567890"
```

```
IssueInstant="2010-12-31T12:00:00Z" Version="2.0"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:eid="http://bsi.bund.de/eID/">
      <saml2:Issuer>http://www.service-provider.de</saml2:Issuer>
      <ds:Signature>
            <ds:SignedInfo>
                  <ds:CanonicalizationMethod
            Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
                  <ds:SignatureMethod
            Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
                  <ds:Reference URI="#Request_1234567890">
                        <ds:Transforms>
                              <ds:Transform
            Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
                              <ds:Transform
            Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                                    <ec:InclusiveNamespaces
            xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
            PrefixList="ds samlp saml2 eid xenc" />
                              </ds:Transform>
                        </ds:Transforms>
                        <ds:DigestMethod
            Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
                              <ds:DigestValue>
                                    p7pVaPes3fATexPMKSlXFgQHsD8=
                              </ds:DigestValue>
                  </ds:Reference>
            </ds:SignedInfo>
            <ds:SignatureValue>
iCKmlQxH5KOzwsmkZnN1MuiKydataEQfg957MUZcnJV4Ld6iPYWRtmgZVMIBTq4EFfcqkcOqMJM5
WeTwiE49kHGP6ROGdEiLqQiXo05bLKp2M3URphCqjlyJI8/kPdJ+HpaSB4BBxq8/Tt3Fd60L58yWYNU
loFSyjc6GP2KKxwQ=
            </ds:SignatureValue>
      </ds:Signature>
      <samlp:Extensions>
            <eid:EncryptedAuthnRequestExtension>
                  <xenc:EncryptedData
            Type="http://www.w3.org/2001/04/xmlenc#Element">
                        <xenc:EncryptionMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
                        <ds:KeyInfo>
                              <xenc:EncryptedKey>
                                    <xenc:EncryptionMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5" />
                                    <xenc:CipherData>
                                          <xenc:CipherValue>
RqsJKl9XTbZkxrj2d0o4TCgwAJhOTqZfzywOPO9Tj4Gwz4zPcnq15n+viEjKSp0l4MpwnKtm8OGx
3fP/RQyJkOCeGelcleB3iJJadDongQe5p0PEUiGrzP8sqm/SSqWnmYw7hroG1Xm61jGhE0ynVgdO
ac9kdw98qXdm8VtfzZM=
                                          </xenc:CipherValue>
                                    </xenc:CipherData>
                              </xenc:EncryptedKey>
                        </ds:KeyInfo>
                        <xenc:CipherData
```

```
            xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
                              <xenc:CipherValue>
Pyh3WvmaP3YKS+LfROIMuPKqZBjWcHn22JiH+uoPwKcw7n+4ySZlBKuK8sV84Nf1TR8ktllxJw26
W8E6T9W+3iDu4wH7Ai70lW2BLAAUcl4FYgnlDirkYbk+Wb84RQBhBw6yBcDi/0lnpPJjGMtMEPAf
wcqMJ0PQLQ+YgEN3wEbe+j/5rpJYL9ur1EDXpbySz9xFR9Tr+tqJVUe7kjyz5Xzya8h1Fl2J9FpC
eQY+zMfGKHiWCQ5yQR8zPB+SJcXzM66Z/cuozqgLoxgV09fqhq0l2gSnQJkK5lwDkc96JdNrhB+z
oHkOWdEq4ag1GN5j/M7qJ7vB0oWaOaTUUDldloQTqIuC2+SuvV9fnF/RuaK6L3LNJFWxXxR2kHpi
3AuKF1DmBS1dAUuwifn7wsAh8QZh+rs=
                              </xenc:CipherValue>
                        </xenc:CipherData>
                  </xenc:EncryptedData>
            </eid:EncryptedAuthnRequestExtension>
      </samlp:Extensions>
</samlp:AuthnRequest>
```

3.7.2 AuthnRequestExtension

Dieses Codebeispiel zeigt die `AuthnRequestExtension`, welche in den zuvor beschriebenen `AuthnRequest` verschlüsselt eingebettet wird. Die `AuthnRequestExtension` enthält gemäß *Kapitel 3.5.2* die vom Diensteanbieter abgefragten Datenfelder und Operationen. In diesem Beispiel sind alle Datenfelder und Operationen vom Diensteanbieter als Pflichtfelder markiert worden.

```
<?xml version="1.0" encoding="UTF-8"?>
<eid:AuthnRequestExtension Version="1"
xmlns:eid="http://bsi.bund.de/eID/"
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <eid:PreSharedKey>
            <eid:Key>0123456789ABCDEF0123456789ABCDEF</eid:Key>
      </eid:PreSharedKey>
      <eid:RequestedAttributes>
            <saml2:Attribute Name="DocumentType" eid:RequiredAttribute="true" />
            <saml2:Attribute Name="IssuingState" eid:RequiredAttribute="true" />
            <saml2:Attribute Name="GivenNames" eid:RequiredAttribute="true" />
            <saml2:Attribute Name="FamilyNames" eid:RequiredAttribute="true" />
            <saml2:Attribute Name="ArtisticName" eid:RequiredAttribute="true" />
            <saml2:Attribute Name="AcademicTitle"
                  eid:RequiredAttribute="true" />
            <saml2:Attribute Name="DateOfBirth" eid:RequiredAttribute="true" />
            <saml2:Attribute Name="PlaceOfBirth" eid:RequiredAttribute="true" />
            <saml2:Attribute Name="PlaceOfResidence"
                  eid:RequiredAttribute="true" />
            <saml2:Attribute Name="RestrictedId" eid:RequiredAttribute="true" />
            <saml2:Attribute Name="CommunityIdVerification"
                  eid:RequiredAttribute="true">
                  <saml2:AttributeValue xsi:type="xs:string">
                        05001234
                  </saml2:AttributeValue>
            </saml2:Attribute>
            <saml2:Attribute Name="AgeVerification"
                  eid:RequiredAttribute="true">
                  <saml2:AttributeValue xsi:type="xs:unsignedShort">
                        18
                  </saml2:AttributeValue>
            </saml2:Attribute>
```

```
      </eid:RequestedAttributes>
</eid:AuthnRequestExtension>
```

3.7.3 Response

Dieses Codebeispiel zeigt exemplarisch eine `Response` gemäß *Kapitel 3.5.3*.

```
<?xml version="1.0" encoding="UTF-8"?>
<samlp:Response
Destination="https://www.service-provider.de/processResponse"
ID="Response_1234567890" InResponseTo="Request_1234567890"
IssueInstant="2010-12-31T12:00:00" Version="2.0"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <saml2:Issuer>http://www.identity-provider.de</saml2:Issuer>
      <ds:Signature>
            <ds:SignedInfo>
                  <ds:CanonicalizationMethod
            Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
                  <ds:SignatureMethod
            Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
                  <ds:Reference URI="#Response_1234567890">
                        <ds:Transforms>
                              <ds:Transform
            Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
                              <ds:Transform
            Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                                    <ec:InclusiveNamespaces
            xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
            PrefixList="ds samlp saml2 eid xenc" />
                              </ds:Transform>
                        </ds:Transforms>
                        <ds:DigestMethod
            Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
                        <ds:DigestValue>
                              p7pVaPes3fATexPMKSlXFgQHsD8=
                        </ds:DigestValue>
                  </ds:Reference>
            </ds:SignedInfo>
            <ds:SignatureValue>
iCKmlQxH5KOzwsmkZnN1MuiKydataEQfg957MUZcnJV4Ld6iPYWRtmgZVMIBTq4EFfcqkcOqMJM5
WeTwiE49kHGP6ROGdEiLqQiXo05bLKp2M3URphCqjlyJI8/kPdJ+HpaSB4BBxq8/Tt3Fd60L58yW
YNUloFSyjc6GP2KKxwQ=
            </ds:SignatureValue>
      </ds:Signature>
      <samlp:Status>
      <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success" />
      </samlp:Status>
      <saml2:EncryptedAssertion>
            <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element">
                  <xenc:EncryptionMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
                  <ds:KeyInfo>
```

```
                        <xenc:EncryptedKey>
                              <xenc:EncryptionMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5" />
                              <xenc:CipherData>
                                    <xenc:CipherValue>
RqsJKl9XTbZkxrj2d0o4TCgwAJhOTqZfzywOPO9Tj4Gwz4zPcnq15n+viEjKSp0l4MpwnKtm8OGx
3fP/RQyJkOCeGelcleB3iJJadDongQe5p0PEUiGrzP8sqm/SSqWnmYw7hroG1Xm61jGhE0ynVgdO
ac9kdw98qXdm8VtfzZM=
                                    </xenc:CipherValue>
                              </xenc:CipherData>
                        </xenc:EncryptedKey>
                  </ds:KeyInfo>
                  <xenc:CipherData
                  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
                        <xenc:CipherValue>
Pyh3WvmaP3YKS+LfROIMuPKqZBjWcHn22JiH+uoPwKcw7n+4ySZlBKuK8sV84Nf1TR8ktllxJw26
W8E6T9W+3iDu4wH7Ai70lW2BLAAUcl4FYgnlDirkYbk+Wb84RQBhBw6yBcDi/0lnpPJjGMtMEPAf
wcqMJ0PQLQ+YgEN3wEbe+j/5rpJYL9ur1EDXpbySz9xFR9Tr+tqJVUe7kjyz5Xzya8h1Fl2J9FpC
eQY+zMfGKHiWCQ5yQR8zPB+SJcXzM66Z/cuozqgLoxgV09fqhq0l2gSnQJkK5lwDkc96JdNrhB+z
oHkOWdEq4ag1GN5j/M7qJ7vB0oWaOaTUUDldloQTqIuC2+SuvV9fnF/RuaK6L3LNJFWxXxR2kHpi
3AuKF1DmBS1dAUuwifn7wsAh8QZh+rs=
                        </xenc:CipherValue>
                  </xenc:CipherData>
            </xenc:EncryptedData>
      </saml2:EncryptedAssertion>
</samlp:Response>
```

3.7.4 Assertion

Dieses Codebeispiel zeigt die `Assertion`, welche in die zuvor beschriebene `Response` verschlüsselt eingebettet wird. Sie enthält die Rückgabewerte der erfolgreich durchgeführten Authentisierung. Die Assertion wurde entsprechend der Anforderungen aus *Kapitel 3.5.4* gebildet.

```
<?xml version="1.0" encoding="UTF-8"?>
<saml2:Assertion
ID="Assertion_0123456789"
IssueInstant="2010-12-31T12:00:00Z"
Version="2.0"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:eid="http://bsi.bund.de/eID/"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <saml2:Issuer>http://www.identity-provider.de</saml2:Issuer>
      <saml2:Subject>
            <saml2:NameID
            Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">
                  _6b69710c2d804b48356209c9788a661f
            </saml2:NameID>
            <saml2:SubjectConfirmation
            Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
                  <saml2:SubjectConfirmationData
                  Address="127.0.0.1"
                  InResponseTo="Request_1234567890"
```

```
                NotOnOrAfter="2010-12-31T12:05:00Z"
                Recipient="https://www.service-provider.de/processResponse" />
          </saml2:SubjectConfirmation>
    </saml2:Subject>

    <saml2:Conditions>
          <saml2:AudienceRestriction>
                <saml2:Audience>
                      http://www.service-provider.de
                </saml2:Audience>
          </saml2:AudienceRestriction>
          <saml2:OneTimeUse />
    </saml2:Conditions>

    <saml2:AuthnStatement AuthnInstant="2010-12-31T12:00:00Z">
          <saml2:AuthnContext>
                <saml2:AuthnContextDeclRef>
                      urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI
                </saml2:AuthnContextDeclRef>
          </saml2:AuthnContext>
    </saml2:AuthnStatement>

    <saml2:AttributeStatement>

          <saml2:Attribute Name="DocumentType">
                <saml2:AttributeValue xsi:type="xs:string">
                      ID
                </saml2:AttributeValue>
          </saml2:Attribute>

          <saml2:Attribute Name="IssuingState">
                <saml2:AttributeValue xsi:type="xs:string">
                      D
                </saml2:AttributeValue>
          </saml2:Attribute>

          <saml2:Attribute Name="GivenNames">
                <saml2:AttributeValue xsi:type="xs:string">
                      Erika
                </saml2:AttributeValue>
          </saml2:Attribute>

          <saml2:Attribute Name="FamilyNames">
                <saml2:AttributeValue xsi:type="xs:string">
                      Mustermann
                </saml2:AttributeValue>
          </saml2:Attribute>

          <saml2:Attribute Name="ArtisticName">
                <saml2:AttributeValue xsi:type="xs:string">
                </saml2:AttributeValue>
          </saml2:Attribute>

          <saml2:Attribute Name="AcademicTitle">
                <saml2:AttributeValue xsi:type="xs:string">
                      Dr.
                </saml2:AttributeValue>
          </saml2:Attribute>

          <saml2:Attribute Name="DateOfBirth">
                <saml2:AttributeValue xsi:type="eid:GeneralDateType">
                      <eid:DateString>19740101</eid:DateString>
```

```
                  <eid:DateValue>1974-01-01</eid:DateValue>
            </saml2:AttributeValue>
      </saml2:Attribute>

      <saml2:Attribute Name="PlaceOfBirth">
            <saml2:AttributeValue xsi:type="eid:GeneralPlaceType">
                  <eid:StructuredPlace>
                        <eid:City>Berlin</eid:City>
                        <eid:Country>D</eid:Country>
                  </eid:StructuredPlace>
            </saml2:AttributeValue>
      </saml2:Attribute>

      <saml2:Attribute Name="PlaceOfResidence">
            <saml2:AttributeValue xsi:type="eid:GeneralPlaceType">
                  <eid:StructuredPlace>
                        <eid:Street>Heidestrasse 17</eid:Street>
                        <eid:City>Köln</eid:City>
                        <eid:Country>D</eid:Country>
                        <eid:ZipCode>51147</eid:ZipCode>
                  </eid:StructuredPlace>
            </saml2:AttributeValue>
      </saml2:Attribute>

      <saml2:Attribute Name="RestrictedId">
            <saml2:AttributeValue xsi:type="eid:RestrictedIDType">
                  <eid:ID>
01A4FB509CEBC6595151A4FB5F9C75C6FE01A4FB59CB655A4FB5F9C75C6FEE
                  </eid:ID>
                  <eid:ID2>
5C6FE01A4FB59CB655A4FB5F9C75C6FEE01A4FB509CEBC6595151A4FB5F9C7
                  </eid:ID2>
            </saml2:AttributeValue>
      </saml2:Attribute>

      <saml2:Attribute Name="CommunityIdVerification">
            <saml2:AttributeValue
            xsi:type="eid:CommunityIdVerificationResultType">
                  <eid:Request>05001234</eid:Request>
                  <eid:Result>false</eid:Result>
            </saml2:AttributeValue>
      </saml2:Attribute>

      <saml2:Attribute Name="AgeVerification">
            <saml2:AttributeValue
            xsi:type="eid:AgeVerificationResultType">
                  <eid:Request>18</eid:Request>
                  <eid:Result>true</eid:Result>
            </saml2:AttributeValue>
      </saml2:Attribute>

      <saml2:Attribute Name="DocumentValidity">
            <saml2:AttributeValue
            xsi:type="eid:DocumentValidityResultType" Version="1">
                  <eid:ReferenceDate>2010-12-31</eid:ReferenceDate>
                  <eid:Status>valid</eid:Status>
            </saml2:AttributeValue>
      </saml2:Attribute>

   </saml2:AttributeStatement>
</saml2:Assertion>
```

Anhang B: Schemadateien

Die folgenden Schemadateien sind Teile dieses Dokuments:

- `TR-03130_TR-eID-Server.wsdl` enthält die WSDL-Spezifikation der eID-Schnittstelle.
- `TR-03130_TR-eID-Server.xsd` enthält die XSD-Spezifikation der eID-Schnittstelle und der in Anhang A beschriebenen SAML-Erweiterung. Neben den Datentypen, die an der eID-Schnittstelle verwendet werden, sind in diesem Schema auch die Datentypen und Elemente für die Verwendung mit SAML definiert.

Anhang C: Anforderungen an den Betrieb von eID-Servern

1 Problemstellung

In diesem Anhang werden die Anforderungen an den Betrieb von eID-Servern durch Diensteanbieter bzw. durch eID-Service-Provider aus Sicht der Informationssicherheit definiert. Bereits in der *Certificate Policy für die eID-Anwendung des ePA* [CP_CVCA-eID] und in der *Technischen Richtlinie 'EAC-PKI'n für den elektronischen Personalausweis'*[EAC-PKI'n ePA], die für alle PKI Teilnehmer gelten, werden Verfahren und Sicherheitsanforderungen, zu denen auch die Erstellung eines Sicherheitskonzeptes gehört, beschrieben. Davon ausgehend wird für den Betrieb von eID-Servern ein angemessenes Sicherheitsniveau spezifiziert.

Ein Diensteanbieter kann den technischen Betrieb des eID-Servers selbst vornehmen oder auf einen Dienstleister, sei es als ausgelagertes Service-Zentrum seines eigenen Hauses oder Unternehmensverbundes oder im Sinne eines spezifischen Dienstleister (eID-Service-Provider) übertragen.

In jedem Fall ist der Diensteanbieter der Inhaber der Berechtigung, die die Vergabestelle für Berechtigungszertifikate (VfB) ausstellt, und damit für die Einhaltung sämtlicher Vorgaben und Anforderungen im eigenen Haus sowie im Falle der Aufgabenübertragung an einen Dritten verantwortlich.

Nachfolgend werden diese grundlegenden Konstellationen eines eID-Server-Betriebs charakterisiert und sicherheitstechnisch analysiert.

2 Struktur des eID-Servers

Zu den Aufgaben des eID-Servers gehören, neben der Kommunikation mit der clientseitigen eCard-API, verbunden mit der Durchführung der notwendigen kryptographischen Protokolle, auch die Anbindung an die Hintergrundsysteme (Berechtigungs-PKI, Dokumenten-PKI, Ausweissperrliste), sowie die Übermittlung des Ergebnisses der Online-Authentisierung an die weiteren Systeme des Diensteanbieters.

Da die Einbindung von eID-Servern in sehr individuellen Anwendungsumgebungen auf verschiedene Weise erfolgen kann, werden nachfolgend die Grundtypen eines dedizierten und eines mandantenfähigen eID-Servers beschrieben, welche die Basis für die weitere Betrachtung sind. Ausführlichere Informationen zu den Integrationsmöglichkeiten sowie zu den Diensten und Komponenten des eID-Servers finden sich in *Kapitel 2.4*.

2.1 Dedizierter Server

2.1.1 Lokaler eID-Server beim Diensteanbieter

Der eID-Server steht im Rechenzentrum des Diensteanbieters und ist direkt an den Webserver bzw. die (Web-)Anwendung angebunden. Inwieweit er einem Identitätsmanagement zugeordnet wird, kann an dieser Stelle unberücksichtigt bleiben, da die vergleichbaren Kommunikations-verbindungen zum eID-Server in gleichem Maße abgesichert werden müssen. Damit ergibt sich die in *Abbildung 39* dargestellte Struktur.

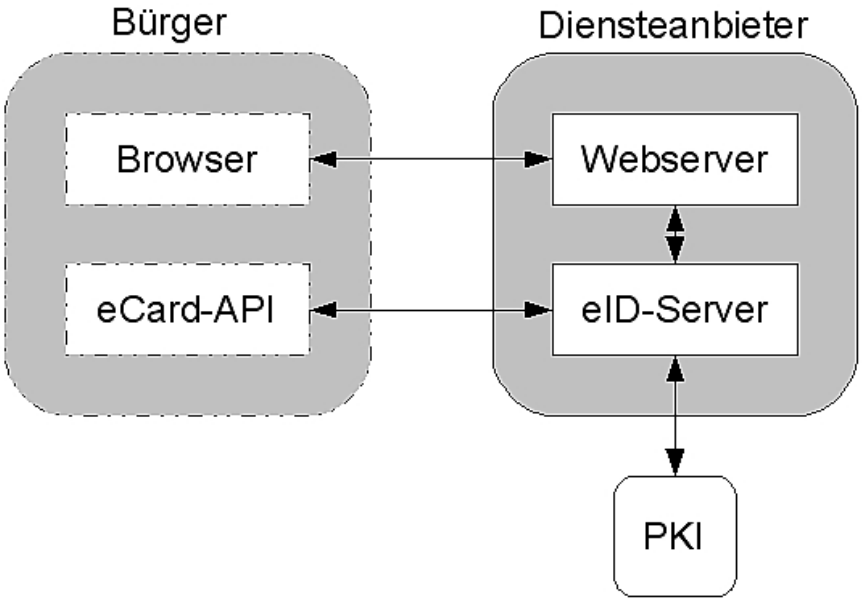

Abbildung 39: Lokaler eID-Server beim Diensteanbieter

2.1.2 Ausgelagerter eID-Server

Bei dieser Variante betreibt der Diensteanbieter den eID-Server örtlich soweit getrennt vom Webserver bzw. der (Web-)Anwendung, dass die Kommunikation zwischen den Komponenten über offene Netze laufen muss. Ansonsten gelten die Ausführungen des *Kapitels 2.1.1*.

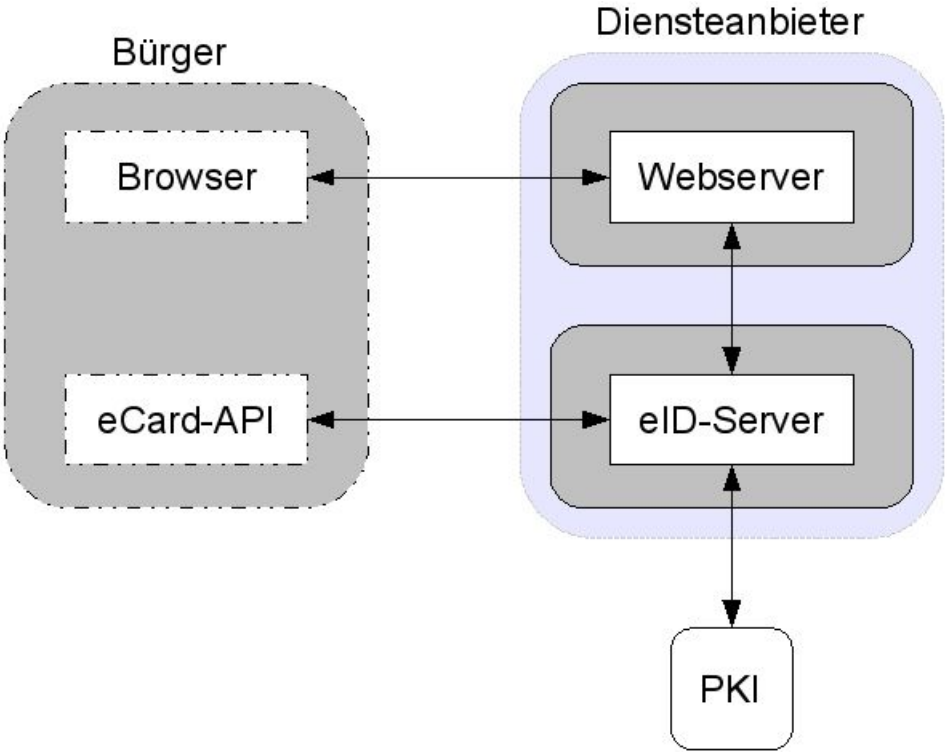

Abbildung 40: Ausgelagerter eID-Server

2.1.3 Bei einem eID-Service-Provider ausgelagerter eID-Server

Übernimmt ein Dienstleister die Aufgaben des elektronischen Identitätsnachweises, wird diese Dienstleistung „eID-Service" , der Dienstleister „ eID-Service-Provider " genannt. Dabei handelt es sich dann um eine sogenannte Auftragsdatenverarbeitung nach §11 Bundesdatenschutzgesetz.

Diese in *Abbildung 41* verdeutlichte Struktur entspricht hinsichtlich ihrer Beziehungen der des *Kapitels 2.1.2*. Allerdings ergeben sich aus der Trennung voneinander unabhängiger juristischer Betreiber besondere Gefährdungen (siehe *Kapitel 4.2*).

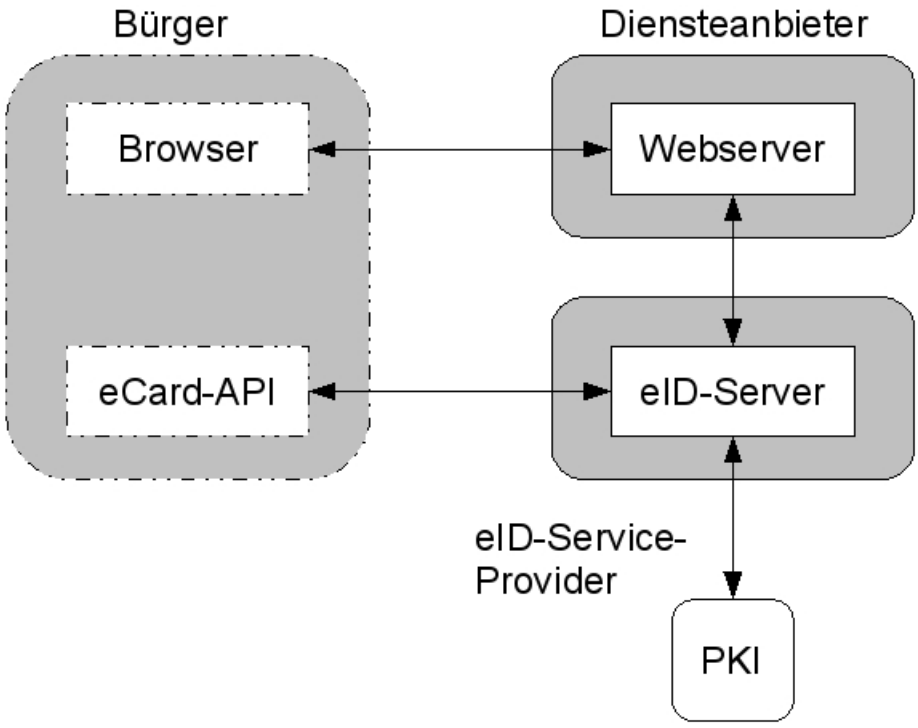

Abbildung 41: Bei einem eID-Service-Provider ausgelagerter eID-Server

2.2 Mandantenfähiger eID-Server

Die Varianten ausgelagerter eID-Server und bei einem eID-Service Provider ausgelagerter eID-Server, wie in den *Kapiteln 2.1.2* und *2.1.3* beschrieben, können auch in einer mandantenfähigen Ausprägung auftreten. Die Begrifflichkeit eID-Service-Provider trifft für beide Realisierungen zu.

Abbildung 42 zeigt den strukturellen Zusammenhang und verdeutlicht, dass die grundlegenden Beziehungen auch in diesem Fall identisch sind. Abweichungen zeigen sich gegenüber den Darstellungen in *Kapitel 2.1.3* gerade in der mandantenfähigen Auslegung des technischen und organisatorischen Betriebsablaufs mit seinen spezifischen (Sicherheits-)Anforderungen.

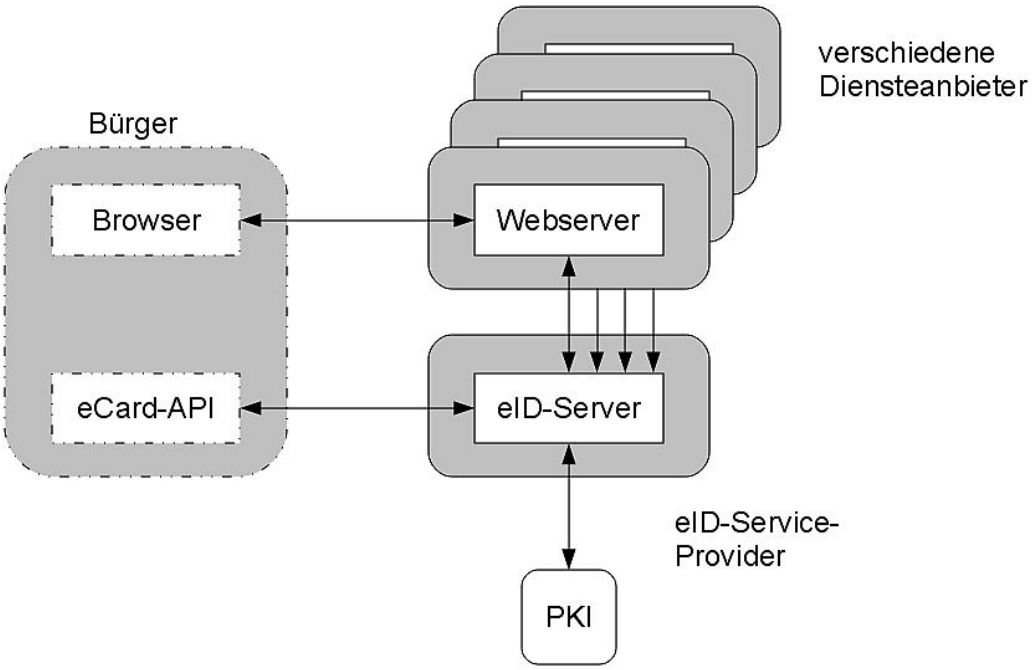

Abbildung 42: Mandantenfähiger eID-Server

2.3 eID -Server

Um später das Gefährdungspotential vollständig identifizieren und abgrenzen zu können, muss auch der eID-Server selbst in die Untersuchung einbezogen werden.

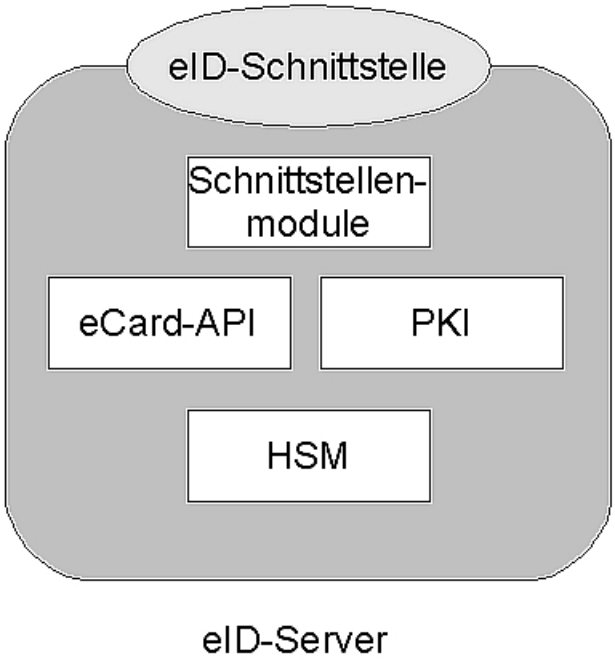

Abbildung 43: Aufbau eines eID-Servers

Der eID-Server stellt über die eID-Schnittstelle einen einfachen Web-Service zur Nutzung der eID-Funktion bereit, der die Komplexität der verwendeten Protokolle und Komponenten kapselt. So werden die Identitätsdaten aus dem eID-Dokument ausgelesen und der Geschäftslogik bereit

gestellt. Die entsprechenden Authentisierungszertifikate werden vom eID-Server gespeichert und verwaltet.

Um dieser Aufgabenstellung gerecht werden zu können, beinhaltet ein eID-Server mehrere Funktionseinheiten, die modulartig integriert sind (siehe *Abbildung 43*):

- Die **Schnittstellenmodule** stellen den Datenaustausch zwischen den internen Modulen des eID-Servers sicher. Für die externe Kommunikation stehen jeweils aufgabenspezifische Schnittstellen zur Verfügung.
- Über die **eCard-API** erfolgt die Kommunikation des eID-Servers mit dem eCard-API Client (z.B. AusweisApp).
- Hinter dem Bereich **PKI** verbirgt sich die Verwaltung und Bereitstellung aller für die Abwicklung des EAC-Protokolls notwendigen Zertifikate, Sperr- und DefectListen sowie die diensteanbieterspezifischen (Dokumenten-)Sperrlisten.
- Das **HSM** (Hardware Sicherheitsmodul, HighSecureModule) ist die zentrale Signaturerstellungseinheit und Schlüsselspeicher des privaten Terminalschlüssels.

Detaillierte Informationen zu den Schnittstellen und Modulen können *Kapitel 2* entnommen werden.

3 Informationsfluss bei der Nutzung des eID-Servers

Die auf ihre Grundstruktur zurückgeführten Implementierungsmöglichkeiten von eID-Servern werden nachfolgend auf ihre Kommunikationsvielfalt hin untersucht und deren Relevanz für die weitere Betrachtung festgehalten.

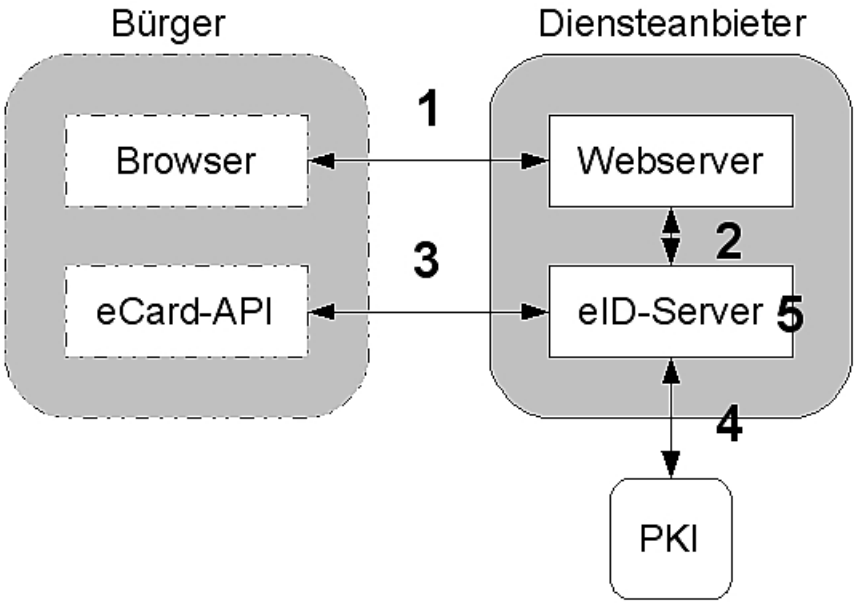

Abbildung 44: Informationsflüsse bei einem dedizierten eID-Server

Hierzu lassen sich ausgehend von der Nummerierung in *Abbildung 44* die einzelnen Vorgänge schrittweise skizzieren. Siehe hierzu *Tabelle 18*.

Nr.	*Beschreibung*	*Sender*	*Empfänger*
1	Der vom Bürger initiierte Webaufruf mit der Option, die eID-Funktion des eID-Dokuments einzusetzen,	AusweisApp (Browser)	Web-Anwendung
	Seitens DA individuell ausgestaltete Kommunikationsstrecke, empfohlener Weise als SSL/TLS-Verbindung.		
2	führt zur Auswahl der Funktionen und Daten des eID-Dokuments, die erst zu einem späteren Zeitpunkt ('x') durch die PIN-Eingabe des Bürgers bestätigt wird.	Web-Anwendung auf Webserver	eID-Server
	Austausch von Sitzungsparametern, Initialisierung der sicheren Verbindung zwischen Web-Anwendung und eID-Server.		
2	Der eID-Server vermittelt der Web-Anwendung für die eindeutige Identifizierung des Vorgangs eine Sitzungsnummer und Informationen für die eCard-API des Bürgers.	eID-Server	Web-Anwendung
1	Die Web-Anwendung übermittelt diese Daten an die AusweisApp (Browser),	Web-Anwendung	AusweisApp (Browser)
	der diese seinerseits an seine eCard-API (client-seitige Instanz der eCard-API) weiterreicht.	[Browser]	[eCard-API]
3	Die eCard-API wickelt mit diesen Angaben die Interaktion mit dem eID-Server ab.	AusweisApp (eCard-API)	eID-Server
	Dies beinhaltet insbesondere die Abwicklung der General AuthenticationProcedure mit PACE, Terminal-Authentication, PassiveAuthentication und Chip-Authentication.		
	Aufbau eines verschlüsselten und integritätsgesicherten Kanals.		
5	Nach erfolgreicher Chip-Authentication startet der eID-Server auf Basis des ausgelesenen 'Sperrmerkmals' eine Sperrlistenabfrage.	eID-Server	
	Der eID-Server selbst ist Bindeglied der verschiedenen Kommunikationsstrecken und nimmt damit die zentrale Rolle des sicheren 'Datenhändlers' ein.		
3	Durch PIN-Eingabe bestätigtes Auslesen von Daten respektive Ausüben der speziellen Rechte.	AusweisApp (eCard-API)	eID-Server
3	Nach Abwicklung der Interaktion zwischen AusweisApp (eCard-API) und eID-Server löst die AusweisApp (eCard-API) die Kommunikationsverbindung.	AusweisApp (eCard-API)	

2	Während der bestehenden Interaktion versucht die Web-Anwendung beim eID-Server regelmäßig, die Ergebnisse ihrer Auswahlanfrage abzurufen, bis diese zum Zeitpunkt 'x' vorliegen.	Web-Anwendung	eID-Server
2	Bereitstellung und Übergabe der Ergebnisse an die Web-Anwendung.	eID-Server	Web-Anwendung
5	Der eID-Server löscht diese Ergebnisdaten unverzüglich nach Übergabe.	eID-Server	
4	Die Beziehung über die PKI-Schnittstelle kann als Hintergrundprozess bezeichnet werden, der der regelmäßigen Versorgung mit Zertifikaten, Sperrlisten, DefectListen und diensteanbieterspezifischen Sperrlisten dient.	eID-Server	PKI
	Definierte und gesicherte Kommunikationsverbindung nach *[TR-03129]*.		

Tabelle 18: Kommunikationsbeziehungen zum eID-Server

(Für detailliertere Informationen zu den einzelnen Abläufen vgl. *Kapitel 4* der vorliegenden TR sowie *Kapitel 4.5* [ePA Architektur].)

Das Ablaufschema verdeutlicht, dass die Verbindung des eID-Servers zur Web-Anwendung auf die einwandfreie und gesicherte Zuarbeit dieser Web-Anwendung angewiesen ist und selbst keinen direkten Einfluss auf die Kommunikationsverbindung **1** zur AusweisApp (Browser) hat.

Die Kommunikationsverbindung 1 ist jedoch insofern zu betrachten, da die Web-Anwendung auf dem Webserver die Anwendungslogik der im Web-Browser dargestellten Anwendung realisiert. Über diese Verbindung ist es möglich, den elektronische Identitätsnachweis des eID-Dokuments zu nutzen. Vgl. *Kapitel 2.3.2 Web-Anwendung* der vorliegenden TR. Via Webserver des Diensteanbieters werden dazu die für den Aufbau der Verbindung notwendigen Parameter übermittelt. Der sichere Kanal zum eID-Server wird im folgenden über den lokalen Bürgerclient - die AusweisApp - aufgebaut.

Auch die Kommunikationsverbindung **2** mit direkter, gegenseitiger Beauftragung und Versorgung mit personenbezogenen Daten muss in die Sicherheitsbetrachtung mit einbezogen werden. Ihre Problematik vervielfacht sich durch die Möglichkeit des Grundtyps mandantenfähiger, ausgelagerter eID-Server, der eine Vielzahl solcher Kommunikationsprozesse unterhält.

Mit der direkten Verbindung zwischen der AusweisApp (eCard-API) und dem eID-Server als Basis des berechtigten Zugriffs auf die im Chip des Ausweises gespeicherten Daten, ist die Kommunikationsverbindung **3** als sehr kritisch einzustufen und ebenfalls in die Betrachtung mit einzubeziehen.

Über die Kommunikationsverbindung **4** erhält der eID-Server die für die Prüfung der Ausweise und der Zugriffsberechtigung notwendigen Zertifikatsketten, Sperr- und DefectListen sowie die DA-spezifischen Sperrlisten. Auch wenn es sich hier um eine eher als Hintergrundprozess etablierte Kommunikation handelt, bestehen hohe Integritätsanforderungen an die Inhalte der übertragenen Daten.

Zuletzt zeigt sich auch der eID-Server selbst als mit laufender Nummer **5** gekennzeichnete Einheit als Betrachtungsgegenstand. Er ist nicht nur die Gegenstelle der Kommunikationsverbindungen, sondern insbesondere auch Verarbeitungs- und Speichereinheit personenbezogener Daten, Träger der Berechtigungszertifikate mit zugehörigem geheimen Schlüssel und Speichereinheit der Zertifikatsketten, Sperrlisten (CRLs), DefectListen nebst DA-spezifischen Sperrlisten und ist damit Dreh- und Angelpunkt aller Vorgänge zur Nutzung der elektronischen Identitätsfunktion der hoheitlichen Ausweise.

Der Webserver wird aufgrund der Tatsache, dass dort die entsprechenden Web-Anwendungen als Informations- und Anwendungsangebote im Internet bereitgestellt werden, betrachtet. Er geht im folgenden jedoch nicht in die Schutzbedarfsbetrachtungen mit ein.

4 Schutzbedarf

Bevor die identifizierten Komponenten auf abzusichernde Gefährdungen hin beleuchtet werden, ist es notwendig festzustellen, wie hoch ihr Schutzniveau anzusetzen ist. Hierzu kann auf das „*Sicherheitsrahmenkonzept für das Gesamtsystem des elektronischen Personalausweises (ePA)“* zurückgegriffen werden.

4.1 Grundwerte der Informationssicherheit

Bei der Sicherheitsbetrachtung nach der *IT-Grundschutz Vorgehensweise [BSI-100-2]* wurden die Grundwerte der Informationssicherheit Vertraulichkeit *(Geheimhaltung von Daten)*, Integrität *(Nachweis, dass Daten nicht manipuliert werden)* und Authentifizierung *(Nachweis der Identität der Person/der Anwendung)* sowie Verfügbarkeit *(Funktionsweise des eID-Servers)* in nachstehender Weise berücksichtigt.

Vertraulichkeit: Die Informationen des elektronischen Personalausweises, im Bereich der eID-Funktionen die personenbezogenen Daten, sind vertraulich und dürfen nicht unberechtigt zur Kenntnis genommen oder weitergegeben werden. Dies gilt im Bereich des eID-Servers im besonderen für die Kommunikation zwischen eCard und Gegenstelle und für das Auslesen der freigegebenen Daten. Da dem Lesevorgang gemäß [BDSG] eine Zweckbindung unterliegt, dürfen die ausgelesenen Daten im eID-Server nicht gespeichert, sondern nur zur Abarbeitung des konkreten Authentisierungsvorgangs verwendet werden.

Integrität: Die Korrektheit (Unversehrtheit) der aus dem elektronischen Personalausweis ausgelesenen Informationen und der mit ihm verbundenen Verfahren, Anwendungen und Systeme, in diesem Fall der eID-Server, muss sichergestellt werden.

Verfügbarkeit: Der eID-Server steht zur Prüfung der Berechtigungen den entsprechenden Diensteanbietern zur Verfügung.

Authentizität : Die Authentizität der Daten im Personalausweis muss verifizierbar sein. Gleichzeitig muss die Authentizität von Personen/Stellen und technischen Komponenten gewährleistet werden, die auf Daten des eID-Dokuments zugreifen wollen oder in die Prozesse involviert sind.

In der nachfolgenden Tabelle wird der Schutzbedarf für die in *Kapitel 3* festgehaltenen Verbindungen bzw. Komponenten festgelegt und erklärt.

Nr.	*Verbindung/Komponente*		*Schutzziel*
	Grundwert	*Schutzniveau*	
1	Verbindung 1: *Browser - Webserver*		
	Vertraulichkeit	normal	Die transportierten Daten dürfen Unberechtigten nicht zur Kenntnis gelangen.
	Integrität	hoch	Die Unversehrtheit der transportierten Daten muss gewährleistet sein.
	Authentizität	normal	Die Verbindung darf nur dann initialisiert werden, wenn die Sender- und Empfänger-Identitäten eindeutig bewiesen werden konnten.
	Verfügbarkeit	normal	Die Funktionalitäten/Betriebsbereitschaft der beteiligten Komponenten sind der Aufgabenerledigung entsprechend auszulegen.
2	Verbindung 2: *eID-Server - Webserver*		
	Vertraulichkeit	hoch	Die ausgelesenen und transportierten Daten dürfen Unberechtigten nicht zur Kenntnis gelangen.
	Integrität	hoch	Die Unversehrtheit der ausgelesenen und transportierten Daten muss gewährleistet sein.
	Authentizität	hoch	Die Verbindung darf nur dann erfolgen, wenn die Sender- und Empfänger-Identitäten eindeutig bewiesen werden konnten.
	Verfügbarkeit	normal	Die Funktionalitäten/Betriebsbereitschaft der beteiligten Komponenten sind der Aufgabenerledigung entsprechend auszulegen.
3	Verbindung 3: *eCard-API - eID-Server*		
	Vertraulichkeit	hoch	Diese Verbindung ist die direkte Verbindung zwischen dem eCard-API Client und dem eID-Server, d.h. es wird auf die im Chip des Ausweises gespeicherten Daten zugegriffen. Dieser Datenzugriff muss so erfolgen, dass keine unbefugte Einsichtnahme möglich ist und nur berechtigte Zugriffe erfolgen können.
	Integrität	hoch	Die Unversehrtheit der ausgelesenen und transportierten Daten muss gewährleistet sein.
	Authentizität	normal/hoch	Die Authentizität der Sender-/Empfänger-Identitäten muss eindeutig bewiesen werden können. Dies gilt hier insbesondere für den eID-Server und den eCard-API Client sowie die eindeutige Zuordnung der einzelnen Ablaufschritte zum gemeinsamen Vorgang.

	Verfügbarkeit	normal	Die Funktionalitäten/Betriebsbereitschaft der beteiligten Komponenten beim Bürger und beim Diensteanbieter sind der Aufgabenerledigung entsprechend auszulegen. Der eID-Server muss für den Ablauf der General AuthenticationProcedure stets alle gültigen Zertifikate für den/die Diensteanbieter bereithalten.
4	Verbindung 4: *PKI - eID-Server*		
	Vertraulichkeit	hoch	Besondere Anforderungen an die Vertraulichkeit lassen sich nur für die DA-spezifischen (Ausweis-)Sperrlisten ableiten. Deren unberechtigte Kenntnisnahme muss verhindert werden.
	Integrität	hoch	Bei der Versorgung mit allen aktuellen für den Ablauf der General AuthenticationProcedure (GAP) notwendigen Zertifikaten, Sperr-(CRL, DA-spezifische Ausweis-Sperrliste) und DefectListen muss sichergestellt werden, dass daran keine unberechtigten Modifikationen durchgeführt werden können.
	Authentizität	hoch	Diensteanbieter dürfen nur dann ein neues Zertifikat und die spezifischen (Ausweis-)Sperrlisten erhalten, wenn sie ihre Identität eindeutig beweisen können.
	Verfügbarkeit	hoch	Die Funktionalitäten/Betriebsbereitschaft des eID-Servers ist der Aufgabenerledigung entsprechend auszulegen. Der eID-Server muss zudem stets alle aktuellen für den Ablauf der GAP notwendigen Zertifikate, Sperr-(CRLs und DA-spezifische Ausweis-Sperrliste) und DefectListen bereithalten.
5	eID-Server		
	Vertraulichkeit	hoch	Die vom eID-Server – berechtigt – ausgelesenen Daten dürfen Unbefugten nicht zur Kenntnis gelangen können und dürfen aufgrund ihrer Zweckbindung von ihm auch nicht gespeichert werden. Die Mandantenfähigkeit impliziert entsprechend hoch einzustufende Mechanismen zur gegenseitigen Abgrenzung der Vorgänge der einzelnen Diensteanbieter.
	Integrität	hoch	Alle internen Prozesse im eID-Server sowie alle hier gespeicherten Daten müssen vor Veränderung geschützt sein.
	Authentizität	hoch	Die Authentizität der internen Sender-/Empfänger-Identitäten (Prozesse) muss eindeutig bewiesen werden können. Mandantenfähige eID-Server müssen

			durchgehend eine Trennung der Abläufe für die unterschiedlichen Diensteanbieter einhalten. Dies gilt auch für alle auf dem eID-Server gespeicherten und verwalteten Zertifikate, Sperr-und DefectListen, die für den Ablauf der GAP notwendig sind.
	Verfügbarkeit	hoch	Die Funktionalitäten/Betriebsbereitschaft des eID-Servers ist der Aufgabenerledigung entsprechend auszulegen. Der eID-Server muss zudem stets alle aktuellen für den Ablauf der GAP notwendigen Zertifikate, Sperr-(CRLs u. DA-spezifische Ausweis-Sperrliste) und DefectListen bereithalten. Bei mandantenfähig ausgelegtem Betrieb müssen erweiterte Verfügbarkeitsanforderungen berücksichtigt werden.

Tabelle 19: Schutzbedarf des eID-Servers

Der Schutzbedarf für die Grundwerte Vertraulichkeit, Integrität und Authentizität der entsprechenden Komponenten ist also durchweg als hoch eingestuft. Lediglich die Verfügbarkeit wird in einigen Prozessen als normal angegeben. Für die kritischen Verbindungen und Komponenten des eID-Server müssen aufgrund dessen angemessene Sicherheitsanforderungen abgeleitet werden. Zusätzlich sind die Sicherheitsanforderungen aus den Bestimmungen der Technischen Richtlinien des BSI und weiteren Grundsatzdokumenten einschließlich rechtlicher Regelungen zu beachten.

4.2 Gefährdungen

Bevor der Betrieb von eID-Servern abgesichert werden kann, sind die Gefährdungen, denen die Maßnahmen entgegen wirken sollen, zu identifizieren.

Im vorliegenden Dokument werden nur die den Betrieb von eID-Servern wirklich charakterisierenden Gefährdungen benannt. Auf die umfassende Behandlung aller grundsätzlichen Gefährdungen der IT-Grundschutz -Kataloge [GSK] (z.B. „Übergreifende Aspekte“ der Schicht 1) wird verzichtet, da diese bereits in dem Sicherheitskonzept des/der Betreiber berücksichtigt werden.

Nachfolgende Tabelle beschreibt übersichtlich die Gefährdungen der einzelnen Kommunikationsverbindungen und Komponenten. Hierbei werden den identifizierten, spezifischen Gefährdungen für einen eID-Server informativ vergleichbare aus den IT-Grundschutz-Katalogen zur Seite gestellt.

Der Webserver als eigenständige Komponente, aber auch mit seiner Kommunikation zum Browser, ist den im Grundschutz-Baustein *B 5.4 Webserver* beschriebenen Gefährdungen ausgesetzt. Diesen Gefährdungen muss mit den entsprechenden Maßnahmen aus dem IT-Grundschutz-Baustein entgegengewirkt werden.

Komponente	*Spezifische Gefährdungen*	*angelehnt an IT-Grundschutz*
Verbindung 2: eID-Server - Webserver	2.1 Abhören von Informationen beispielsweise bei der Bereitstellung und Übergabe der Ergebnisse der Abfrage an die Webanwendung. Hinweis mandantenfähiges System: besonders kritisch!	G 2.87 Verwendung unsicherer Protokolle in öffentlichen Netzen G 5.7 Abhören von Leitungen G 5.71 Vertraulichkeitsverlust schützenswerter Informationen G 5.85 Integritätsverlust schützenswerter Informationen G 5.104 Ausspähen von Informationen
	2.2 Eingriff in die Initialisierung der sicheren Verbindung zwischen Webserver und eID-Server. Hinweis mandantenfähiges System: besonders kritisch!	G 5.143 Man-in-the-Middle-Angriff G 5.89 Hijacking von Netz-Verbindungen G 5.71 Vertraulichkeitsverlust schützenswerter Informationen G 5.85 Integritätsverlust schützenswerter Informationen
	2.3 Eingriff in den Austausch von Sitzungsparametern. Hinweis mandantenfähiges System: besonders kritisch!	G 5.71 Vertraulichkeitsverlust schützenswerter Informationen G 4.33 Schlechte oder fehlende Authentikation
	2.4 Verbindung mit nicht authentisierten Identitäten. Hinweis mandantenfähiges System: besonders kritisch!	G 4.33 Schlechte oder fehlende Authentikation G 5.84 Gefälschte Zertifikate G 5.83 Kompromittierung kryptographischer Schlüssel
	2.5 Verfügbarkeitsstörungen der Komponenten, insbesondere die Netzwerkkommunikation.	G 4.31 Ausfall oder Störung von Netzkomponenten G 2.45 Konzeptionelle Schwächen des Netzes
Verbindung 3: eCard-API - eID-Server	3.1 Unberechtigte Verwendung von Berechtigungszertifikaten. Hinweis mandantenfähiges System: besonders kritisch!	G 5.71 Vertraulichkeitsverlust schützenswerter Informationen G 5.82 Manipulation eines Kryptomoduls G 4.33 Schlechte oder fehlende Authentikation G 2.105 Verstoß gegen gesetzliche Regelungen und vertragliche Vereinbarungen

	3.2 Manipulation der Daten	G 5.143 Man-in-the-Middle-Angriff G 5.89 Hijacking von Netz-Verbindungen G 5.85 Integritätsverlust schützenswerter Informationen
	3.3 auch 2.5 Verfügbarkeitsstörungen der Komponenten, insbesondere die Netzwerkkommunikation.	G 4.31 Ausfall oder Störung von Netzkomponenten G 2.45 Konzeptionelle Schwächen des Netzes
Verbindung 4: PKI - eID-Server	4.1 Manipulationen der übertragenen Daten.	G 5.84 Gefälschte Zertifikate (scheinbar sichere Verbindung, die aber beispielsweise keinen Tunnel aufbaut und angreifbar ist) G 5.143 Man-in-the-Middle-Angriff G 5.89 Hijacking von Netz-Verbindungen
	4.2 Unberechtigtes Erlangen von (DA-)Berechtigungszertifikaten.	G 4.33 Schlechte oder fehlende Authentikation
	4.3 Unberechtigte Kenntnisnahme der eID-Sperrliste.	G 5.71 Vertraulichkeitsverlust schützenswerter Informationen
	4.4 wie 2.5 Verfügbarkeitsstörungen der Komponenten, insbesondere die Netzwerkkommunikation.	G 4.31 Ausfall oder Störung von Netzkomponenten G 2.45 Konzeptionelle Schwächen des Netzes
Komponente 5: eID-Server	5.1 Unberechtigte Verwendung des DA-Schlüsselmaterials. Siehe auch 2.3 Eingriff in den Austausch von Sitzungsparametern.	G 4.33 Schlechte oder fehlende Authentikation
	5.2 Manipulation von Zertifikaten und Sperrlisten.	G 5.84 Gefälschte Zertifikate
	5.3 und 3.4 Fehlende Zuordnung der einzelnen Ablaufschritte zu einer Sitzung.	z.B. geregelte Verfahrensabläufe, Gefährdungen aus dem organisatorischen Bereich
	5.4 Unzureichendes Löschen der Ergebnisdaten.	Beachtung des Bausteins B 1.15 Löschen und Vernichten von Daten: G 2.54 Vertraulichkeitsverlust durch Restinformationen G 5.71 Vertraulichkeitsverlust schützenswerter Informationen G 2.105 Verstoß gegen gesetzliche

		Regelungen und vertragliche Vereinbarungen
	5.5 Unbefugte Weiterverwendung der Ergebnisdaten	G 5.71 Vertraulichkeitsverlust schützenswerter Informationen G 2.105 Verstoß gegen gesetzliche Regelungen und vertragliche Vereinbarungen
	5.6 wie 2.5 Verfügbarkeitsstörungen der Komponenten	G 4.9 Ausfall der internen Stromversorgung G 4.31 Ausfall oder Störung von Netzkomponenten G 4.34 Ausfall eines Kryptomoduls

Tabelle 20: Gefährdungen

4.3 Maßnahmen

Im Rahmen der Entwicklung der eID-Funktion des elektronischen Personalausweises wurden bereits weitreichende elementare Sicherheitsmaßnahmen umgesetzt und verbindliche Standards und Richtlinien definiert, um die notwendige Sicherheit der Kommunikationsverbindungen bei der Nutzung des eID-Servers zu gewährleisten. So werden beispielsweise in der [CP_CVCA-eID] Sicherheitsmaßnahmen für die CVCA-eID PKI Teilnehmer definiert, die auch im Bereich des eID-Servers Anwendung finden.

Die Beschreibung der Sicherheitsanforderungen orientiert sich nachfolgend an den Gefährdungen, die die Kommunikationsbeziehungen des eID-Servers betreffen.

Zur Absicherung des Webservers sowie der Verbindung Webserver <=> Browser sollte der Baustein *B 5.4 Webserver* aus den IT-Grundschutz-Katalogen [GSK] mit den spezifischen Gefährdungen und Maßnahmen beachtet werden. Insbesondere die Maßnahme *M 5.66 Verwendung von TLS/SSL* ist dabei zu berücksichtigen, da mit Hilfe der dabei verwendeten Server-Zertifikate die Verbindung Webserver <=> Browser an die Verbindung eCard-API <=> eID-Server gebunden wird.

4.3.1 Betrachtung der Verbindung 2: eID-Server <=> Webserver

2.1 Abhören von Informationen bei der Bereitstellung und Übergabe der Ergebnisse der Abfrage an die Webanwendung.		
Maßnahme	***Beschreibung***	***Kommentar***
1	Aufbau einer autorisierten Kommunikationsverbindung	(siehe *Kapitel 3.3*)
2	Verschlüsselung der Kommunikationsverbindung	1. Dedizierter eID-Server: - innerhalb Netzstruktur eindeutige Identifizierung/Authentisierung von Sender und Empfänger - Verschlüsselung auf Transportebene (XML) 2. Mandantenfähiger eID-Server: - auf TLS/SSL abgestellte Sender-Empfänger-Authentisierung mit verschlüsseltem Kanal - Verschlüsselung auf Transportebene (XML)
3	Angelehnte Maßnahmen aus den IT-Grundschutz-Katalogen: *M 2.42 Festlegung der möglichen Kommunikationspartner* *M 2.164 Auswahl eines geeigneten kryptographischen Verfahrens* *M 2.204 Verhinderung ungesicherter Netzzugänge*	

Tabelle 21: Abhören von Informationen bei der Bereitstellung und Übergabe der Ergebnisse der Abfrage an die Webanwendung

2.2 Eingriff in die Initialisierung der sicheren Verbindung		
Maßnahme	***Beschreibung***	***Kommentar***
4	Siehe Maßnahme 1 und 2	
5	Angelehnte Maßnahmen aus den IT-Grundschutz-Katalogen: *M 2.164 Auswahl eines geeigneten kryptographischen Verfahrens*	

Tabelle 22: Eingriff in die Initialisierung der sicheren Verbindung

2.3 Eingriff in den Austausch von Sitzungsparametern		
Maßnahme	***Beschreibung***	***Kommentar***
6	Siehe Maßnahme 1 und 2	
7	Angelehnte Maßnahmen aus den IT-Grundschutz-Katalogen: *M 2.164 Auswahl eines geeigneten kryptographischen Verfahrens* *M 2.165 Auswahl eines geeigneten kryptographischen Produktes*	

Tabelle 23: Eingriff in den Austausch von Sitzungsparametern

2.4 Verbindung mit nicht authentisierten Identitäten		
Maßnahme	***Beschreibung***	***Kommentar***
8	Siehe Maßnahme 1 und 2	
9	Angelehnte Maßnahmen aus den IT-Grundschutz-Katalogen: *M 2.46 Geeignetes Schlüsselmanagement* *M 2.164 Auswahl eines geeigneten kryptographischen Verfahrens* *M 2.204 Verhinderung ungesicherter Netzzugänge*	

Tabelle 24: Verbindung mit nicht authentisierten Identitäten

2.5 Verfügbarkeitsstörungen der Komponenten, insbesondere die Netzwerkkommunikation		
Maßnahme	***Beschreibung***	***Kommentar***
10	Lastabhängige Ausgestaltung der Netze und Komponenten	Im Einzelfall vom DA zu eruieren bzw. zwischen DA und Betreiber des eID-Servers abzustimmen und mit gängigen Maßnahmen aufzufangen (z.B. Redundanzen von Netzkomponenten, Load-Balancer etc.).
11	Angelehnte Maßnahmen aus den IT-Grundschutz-Katalogen: *M 2.314 Verwendung von hochverfügbaren Architekturen für Server* *M 4.183 Sicherstellen der Verfügbarkeit und Performance des IIS* *M 6.53 Redundante Auslegung der Netzkomponenten*	

Tabelle 25: Verfügbarkeitsstörungen der Komponenten, insbesondere die Netzwerkkommunikation

4.3.2 Betrachtung der Verbindung 3: eCard-API <=> eID-Server

3.1 Unberechtigte Verwendung von Berechtigungszertifikaten		
Maßnahme	***Beschreibung***	***Kommentar***
12	Sicherstellung der eineindeutigen Zuordnung gültiger Berechtigungszertifikate zu DA's.	(siehe [EAC 2], [EAC-PKI'n ePA], [CP_CVCA-eID]) Inhalt und Ablauf der berechtigten Zertifikatsanträge, -ausstellung und -verteilung. Eindeutige Verknüpfung von Berechtigungszertifikaten zu DA's mittels Funktionalitäten des eID-Servers.
13	Revisionssichere Protokollierung der Verwendung der Berechtigungszertifikate.	Gestaltung eines jederzeit nachvollziehbaren eID-Server-Betriebs und der Verwendung der Berechtigungszertifikate.
14	Einsatz zertifizierter Produkte	Konformitätsnachweis des eCard-API Client und des eCard-API Servers. (vgl. *Maßnahme 29*).
15	Angelehnte Maßnahmen aus den IT-Grundschutz-Katalogen: *M 2.164 Auswahl eines geeigneten kryptographischen Verfahrens* *M 2.204 Verhinderung ungesicherter Netzzugänge* *M 2.340 Beachtung rechtlicher Rahmenbedingungen*	

Tabelle 26: Unberechtigte Verwendung von Berechtigungszertifikaten

3.2 Manipulation der Daten während der Übertragung		
Maßnahme	***Beschreibung***	***Kommentar***
16	Aufbau einer gesicherten Verbindung.	(siehe [EAC 2], [eCard-API], [eCard des Bundes]) Mit der seitens des eCard-API Clients eingeleiteten Benutzerinteraktion zum eID-Server ist eine verschlüsselte TLS-Verbindung mit Server-Zertifikat aufzubauen, die den Tunnel für die weiteren Abläufe und den über die General AuthenticationProcedure (GAP) zusätzlich einzurichtenden sicheren Kanal zwischen dem eID-Dokument und eCard-API Server darstellt

Tabelle 27: Manipulation der Daten während der Übertragung

3.3 Verfügbarkeitsstörungen der Komponenten, insbesondere der Netzwerkkommunikation		
Maßnahme	***Beschreibung***	***Kommentar***
17	Siehe Maßnahme 10.	Im Einzelfall vom DA bzw. vom Betreiber des eID-Servers zu eruieren und mit gängigen Maßnahmen aufzufangen (z.B. Redundanzen von Netzkomponenten, Load-Balancer etc.).
18	Angelehnte Maßnahmen aus den IT-Grundschutz-Katalogen: *M 5.123 Absicherung der Netzkommunikation unter Windows* *M 6.53 Redundante Auslegung der Netzkomponenten*	

Tabelle 28: Verfügbarkeitsstörungen der Komponenten, insbesondere der Netzwerkkommunikation

4.3.3 Betrachtung der Verbindung 4: PKI <=> eID-Server

4.1 Manipulation der übertragenen Daten		
Maßnahme	***Beschreibung***	***Kommentar***
19	Aufbau einer gegenseitig authentisierten und gesicherten Verbindung.	(siehe [EAC-PKI'n ePA], [EAC-PKI Protocol]) Einrichtung einer TLS-verschlüsselten Verbindung mit gegenseitiger Authentisierung auf Basis eines Registrierungsverfahrens der DA's und der eID-Server-Betreiber bei der VfB und bei der BerCA.
20	Verifizieren der von der Internetseite der Root CVCA-eID heruntergeladenen Zertifikate, Sperr- und DefectListen.	Prüfung der signierten Downloads anhand von Fingerprints und/oder Zertifikaten.
21	Angelehnte Maßnahmen aus den IT-Grundschutz-Katalogen: *M 2.205 Übertragung und Abruf personenbezogener Daten*	

Tabelle 29: Manipulation der übertragenen Daten

4.2 Unberechtigtes Erlangen von (DA-)Berechtigungszertifikaten		
Maßnahme	***Beschreibung***	***Kommentar***
22	Siehe Maßnahme 19.	
23	Durchlaufen eines Registrierungsverfahrens.	(siehe [EAC-PKI'n ePA]) Ausstellen von DA-Berechtigungszertifikaten nur nach vollständigem Durchlaufen einer Registrierung der DA's und eID-Service-Betreiber bei der VfB und bei der BerCA.
24	Angelehnte Maßnahmen aus den IT-Grundschutz-Katalogen: *M 2.164 Auswahl eines geeigneten kryptographischen Verfahrens* *M 2.165 Auswahl eines geeigneten kryptographischen Produktes*	

Tabelle 30: Unberechtigtes Erlangen von (DA-)Berechtigungszertifikaten

4.3 Unberechtigte Kenntnisnahme der eID-Sperrliste		
Maßnahme	***Beschreibung***	***Kommentar***
25	Siehe *Maßnahme 19*.	
26	Angelehnte Maßnahmen aus den IT-Grundschutz-Katalogen: *M 2.164 Auswahl eines geeigneten kryptographischen Verfahrens* *M 2.165 Auswahl eines geeigneten kryptographischen Produktes*	

Tabelle 31: Unberechtigte Kenntnisnahme der eID-Sperrliste

4.4 Verfügbarkeitsstörungen der Komponenten, insbesondere die Netzwerkkommunikation		
Maßnahme	***Beschreibung***	***Kommentar***
27	Siehe *Maßnahme 10*.	Im Einzelfall vom DA bzw. vom Betreiber des eID-Servers zu eruieren und auf die interne Organisation abzustimmen.
28	Angelehnte Maßnahmen aus den IT-Grundschutz-Katalogen: *M 2.314 Verwendung von hochverfügbaren Architekturen für Server* *M 4.183 Sicherstellen der Verfügbarkeit und Performance des IIS* *M 6.53 Redundante Auslegung der Netzkomponenten*	

Tabelle 32: Verfügbarkeitsstörungen der Komponenten, insbesondere die Netzwerkkommunikation

4.3.4 Betrachtung der Komponente 5: eID-Server

5.1 Unberechtigte Verwendung des DA-Schlüsselmaterials		
Maßnahme	***Beschreibung***	***Kommentar***
29	Einsatz sicherer Kryptographiemodule: physikalisch gesicherte Speicherung, rollenbasierte Zugriffsmechanismen, Kontrolle der logischen Zugriffe auf das Schlüsselmaterial.	(siehe [CP_CVCA-eID]) High Security Module (HSM): PP-CryptographicModules "moderate“ [PP-CM-m]. -Chipkarten: PP-SecureSignature-Creation Device [PP-SSCD]. N.B: Ausnahmen von dieser Regelung können auf Antrag im Einzelfall vom BSI zugestimmt werden.
30	Sichere Betriebsumgebung.	(siehe [CP_CVCA-eID], [EAC-PKI'n ePA], [BDSG]) Erstellung eines Sicherheitskonzeptes gemäß den BSI-Standards 100-2, 100-3 und 100-4. Eine Grundschutz-Zertifizierung nach ISO 27001 wird empfohlen.
31	Einsatz zertifizierter Software-Module.	(siehe PauswG/PAuswV) Konformitätsprüfung der eCard-API nach BSI-TestSuite (in Entwicklung).
32	Angelehnte Maßnahmen aus den IT-Grundschutz-Katalogen: *M 2.164 Auswahl eines geeigneten kryptographischen Verfahrens* *M 2.165 Auswahl eines geeigneten kryptographischen Produktes*	

Tabelle 33: Unberechtigte Verwendung des DA-Schlüsselmaterials

5.2 Manipulation von Zertifikaten und Sperrlisten		
Maßnahme	***Beschreibung***	***Kommentar***
33	Verifikation der zum Einsatz kommenden Zertifikate und Sperrlisten.	(siehe [CP_CVCA-eID], [EAC-PKI'n ePA]) Die eingerichteten PublicKey Infrastrukturen gewährleisten den geforderten Integritätsschutz.
34	Angelehnte Maßnahmen aus den IT-Grundschutz-Katalogen: *M 2.164 Auswahl eines geeigneten kryptographischen Verfahrens* *M 2.165 Auswahl eines geeigneten kryptographischen Produktes*	

Tabelle 34: Manipulation von Zertifikaten und Sperrlisten

5.3 Fehlende Zuordnung der einzelnen Ablaufschritte zu einer Sitzung		
Maßnahme	***Beschreibung***	***Kommentar***
35	Einbringen zuverlässiger Identifizierungsmechanismen und integere Weiterleitung an und Verwendung in den (Teil-)Prozessen.	Identifizierungsmechanismen der zusammengehörenden Prozesse (Kanalbindung) zwischen Web-Anwendung, Browser, eCard-API Client, eCard-API Server und erneut Web-Anwendung (z.B. Pre-SharedKey)
36	Freigabeverfahren für Hard- und Software	Qualifizierte Testverfahren zur Sicherstellung des Leistungsumfangs.

Tabelle 35: Fehlende Zuordnung der einzelnen Ablaufschritte zu einer Sitzung

5.4 Unzureichendes Löschen der Ergebnisdaten		
Maßnahme	***Beschreibung***	***Kommentar***
37	Übertragung und Abruf personenbezogener Daten	Erheben, Speichern, Weitergeben und Löschen personenbezogener Daten erfolgt stets nur im Rahmen der erlaubten Verwendung (Zweckbindung).
38	Siehe Maßnahme 36.	
	Angelehnte Maßnahmen aus den IT-Grundschutz-Katalogen: Baustein *B 1.15 Löschen und Vernichten von Daten*	

Tabelle 36: Unzureichendes Löschen der Ergebnisdaten

5.5 Unbefugte Weiterverwendung von Ergebnisdaten		
Maßnahme	***Beschreibung***	***Kommentar***
39	**Datenschutz** Allgemeine Datenschutzbestimmung, insbesondere aber das Bundesdatenschutzgesetz (BDSG) sind zu beachten, da im Zuge der Online-Authentisierung personenbezogene Daten erhoben (aus dem Ausweis ausgelesen) und verarbeitet werden. Die Verpflichtungen umfassen u.a. - Einhalten der Zweckbindung der ausgelesenen Daten - Recht auf Auskunft über gespeicherte Daten bzw. Löschen gespeicherter Daten - Verpflichtung zur Absicherung der Verarbeitung personenbezogener Daten (§ 9 BDSG) - Verbot der Weitergabe der Daten ohne Einwilligung Bei Nutzung eines eID-Service-Provider ist insbesondere §11 BDSG sowohl vom Diensteanbieter als auch vom eID-Service-Provider zu beachten. Insbesondere ist der Diensteanbieter weiterhin für die Einhaltung der Datenschutzvorschriften verantwortlich.	
40	Angelehnte Maßnahmen aus den IT-Grundschutz-Katalogen: Baustein *B 1.5 Datenschutz.*	

Tabelle 37: Unbefugte Weiterverwendung von Ergebnisdaten

5.6 Verfügbarkeitsstörungen der Komponenten, insbesondere die Netzwerkkommunikation		
Maßnahme	***Beschreibung***	***Kommentar***
41	Siehe *Maßnahme 10*.	1. Dedizierter eID-Server: - Im Einzelfall vom DA bzw. vom Betreiber des eID-Servers zu eruieren und auf die interne Organisation abzustimmen. 2. Mandantenfähiger eID-Server: - Im Einzelfall vom Betreiber des eID-Servers zu eruieren, auf die interne Organisation sowie mit den externen Kommunikationspartnern abzustimmen.
42	Angelehnte Maßnahmen aus den IT-Grundschutz-Katalogen: *M 2.314 Verwendung von hochverfügbaren Architekturen für Server* *M 4.183 Sicherstellen der Verfügbarkeit und Performance des IIS* *M 6.53 Redundante Auslegung der Netzkomponenten*	

Tabelle 38: Verfügbarkeitsstörungen der Komponenten, insbesondere die Netzwerkkommunikation

5 Rechtlich verankerte Sicherheitsanforderungen

Gesetz über Personalausweise und den elektronischen Identitätsnachweis sowie zur Änderung weiterer Vorschriften

Verordnung über Personalausweise und den elektronischen Identitätsnachweis

Bundesdatenschutzgesetz [BDSG]

Certificate Policy für die eID-Anwendung des ePA [CP_CVCA-eID]

Alle Technischen Richtlinien, insbesondere

BSI TR-03110: Advanced Security Mechanisms for Machine Readable Travel Documents – Extended Access Control (EAC), Password Authenticated Connection Establishment (PACE) and Restricted Identification (RI) [EAC 2]

BSI TR-03112: eCard-API-Framework [eCard-API]

BSI TR-03117: eCards mit kontaktloser Schnittstelle als sichere Signaturerstellungseinheit

BSI TR-03128: EAC-PKI'n für den elektronischen Personalausweis [EAC-PKI'n ePA]