# IT-Systemkonzept

SSI-basierter Führerscheinnachweis

Bundesdruckerei GmbH | Version für initialen Go-Live
18.08.2021

*Arbeitsstand vom August 2021*

# Inhaltsverzeichnis

# 1 Zweck des Dokuments

Zweck dieses Dokuments ist es die Funktionsweise und den Lebenszyklus des SSI-basierten Führerscheinnachweises für Dritte nachvollziehbar zu machen bzw. zu dokumentieren. Hierbei werden neben den funktionalen Aspekten der Lösung vor allem IT-Sicherheitstechnische Aspekte in den Vordergrund gestellt.

Da der vorliegende Führerscheinnachweis auf dem Hyperledger-basierten Wallet-Ökosystem beruht, wird eine Einordnung bzw. Abgrenzung der Lösung im Kontext des Ökosystems gegeben.

## 1.1 Vorgehen

Zunächst werden in Kapitel 2 die wichtigsten Definitionen gegeben. Kapitel 3 führt auf abstrakter Ebene in das System ein und dient zum groben Verständnis der Zusammenhänge und einer Einordnung des beschriebenen Anwendungsfalles in den größeren Kontext. Danach wird in Kapitel 4 eine technische Beschreibung des ID Lifecycles gegeben. Diese ermöglicht eine Nachvollziehbarkeit der Abläufe im System. Anschließend werden in Kapitel 5 die Sicherheitseigenschaften betrachtet. Dazu werden die Anforderungen der verschiedenen Bereiche einzeln betrachtet und ihre Umsetzung beschrieben.

# 2 Definitionen

**Aries Agents/Agents**: Software-Agenten, die SSI-Funktionalitäten zum Austausch mit anderen Agenten und zur Kommunikation für das Verifiable Data Registry (Hyperledger Indy) zur Verfügung stellen und mit der Wallet-App kommunizieren zu können

**Controller**: ist ein Überbegriff für Issuer und/oder Verifier Service.

**Claim:** Ein “Claim” ist eine Aussage über ein Subjekt. (z.B. Person XY ist ein Absolvent der Universität YZ)

**Holder** (im Dokument auch Halter[1] oder Nutzer): Die Nutzer erhalten die Credentials von dem Aussteller, speichern diese in ihrer Wallet und können diese gegenüber Verifiern präsentieren

**Identität**: ist eine Menge von Attributen, die sich auf eine Entität beziehen [ISO24].

**ID Wallet**: App zum Speichern, Freigeben und Übertragen von Credentials

[1] Aus Gründen der Lesbarkeit wurde im Text nun das generische maskulin verwendet, weibliche Formen sind aber stets mitgemeint.

**Indy Node/Node**: Der Hyperledger Indy Knoten, welcher eine Kopie des Verifiable Data Registries hält und am Konsens teilnimmt.

**Issuer** (im Dokument auch Aussteller genannt): Der Issuer/der Aussteller stellt Identitäten in Form von Credentials an den Holder aus.

**Konsens**: Beschreibt den Zustand des Netzwerks, in welchen man sich über den Zustand von Schreibaktionen geeinigt hat.

**Mediator-Service**: Der Mediator-Service agiert als Notifizierungsdienst und Inbox für eingehende SSI-bezogene DIDComm-Nachrichten für mobile Endgeräte.

**Originator**: Eine generische Bezeichnung des Systems, welches den Verifikationsprozess initiiert.

**Subjekt**: Ein Subjekt ist etwas, über das eine Aussage getroffen werden kann (z.B. eine Person).

**Verifiable Credential**: Ein digitaler Nachweis, welcher von einem Issuer herausgegeben wird und welcher vom Verifier auf seine Gültigkeit und Unverändertheit über den Abgleich mit dem öffentlichen Schlüsselmaterial im Verifiable Data Registry geprüft werden kann. Ein Credential beinhaltet einen oder mehrere Claims über ein Subjekt.

**Verifiable Data Registry**: Das Register in welchem unter anderem öffentliches Schlüsselmaterial abgespeichert werden. Dabei kann die Speicherung zentral oder dezentral vorgenommen werden. In diesem Fall wird ein öffentliches, genehmigungspflichtiges Blockchain-Netzwerk (Hyperledger Indy bestehend aus den Indy Nodes) verwendet. Diese dezentrale Prüfinfrastruktur wird von authentisierten und autorisierten Knotenbetreibern betrieben.

**Verifier**: Der Verifier ist der Empfänger von Credentials und den darin enthaltenen Identitätsattributen (“Claims") oder logischen Aussagen aus diesen und stößt den Verifizierungsprozess über das Netzwerk an, um die Integrität, Authentizität (u.a. Herkunft) und Konsistenz der Claims zu prüfen.

# 3 Einführung

Die Bundesregierung arbeitet am Aufbau eines Ökosystems für digitale Identitätsnachweise. Die Grundprämisse hinter diesem neuen Ökosystem ist, dass die Kontrolle digitaler Identitätsnachweise im Sinne eines Self-Sovereign-Identity-Ansatzes nach dem Ausstellen bei den Nutzern selbst liegt. Vergleichbar zu haptischen Nachweisen werden Identitätsnachweise als gültig anerkannt, wenn sie nachweislich von einer vertrauenswürdigen Stelle ausgestellt wurden. Die Glaubhaftigkeit von haptischen

Nachweisen wird über schwer zu kopierende Materialien und Wasserzeichen sowie Signaturen, Stempel oder Siegel sichergestellt. In der digitalen Welt können digitale Signaturen verwendet werden, um die Authentizität eines Dokuments nachzuweisen. Diese kryptographisch überprüfbaren Zertifikate oder Identitätsnachweise, wie etwa ein digitaler Führerscheinnachweis, ein digitales Zeugnis oder ein digitales Ticket, werden im SSI-Kontext als Verifiable Credentials (VC) bezeichnet. Sie werden lokal in einer Wallet-App (vergleichbar mit einem Portemonnaie) der Nutzer gespeichert und können von dort aus auf Nachfrage vorgezeigt und geprüft werden.

Dieses Dokument beschreibt den digitalen Führerscheinnachweis in dem die Potenziale von SSI für sicheres und effizientes Identitätsmanagement demonstriert werden. Als erste Anwendung zum Prüfen soll dabei die Verwendung im Rahmen des „Flottenmanagement“ zügig in der Praxis erprobt werden und die Umsetzbarkeit und die Vorteile dezentraler, digitaler Identitätsnachweise sichtbar machen.

Mit Beginn des produktiven Betriebs ist geplant, dass jede(r), dessen/deren Führerscheindaten im Zentralen Fahrerlaubnisregister (ZFER) des Kraftfahrt-Bundesamtes (KBA) gespeichert sind, sich den digitalen Führerscheinnachweis in der vorhandene ID Wallet als verifiable SSI Credential gemäß dem SSI Standard laden kann. Dieser steht somit für den initialen Flottenmanagement Betrieb, aber gleichzeitig auch für künftige Nutzungen in anderen Anwendungsfällen, wie Carsharing oder Autovermietung, die einmalig oder turnusmäßig die Überprüfung des Führerscheins verlangen, zur Verfügung.

Im Flottenmanagement-Betrieb wird Mitarbeitern und Flottennutzern von ausgewählten Unternehmen / Flottenbetreibern die Möglichkeit eingeräumt, die turnusmäßig Überprüfung des Führerscheins mittels des digitalen Führerscheinnachweises durchzuführen.

Das Ergebnis dieses Projektes erweitert die Nutzung von SSI in einem perspektivisch europaweit geplanten Ökosystem, welche parallel in weiteren sektoralen Arbeitsgruppen erarbeitet und umgesetzt werden.

Bei der Umsetzung der Funktionalität wird die DLT-/Blockchain-Technologie verwendet, um die Integrität der gespeicherten Daten zu sichern. Der gewählte dezentrale und offene Ansatz verhindert zudem, dass eine einzelne Organisation Kontrolle über das Ökosystem oder die dort verarbeiteten Daten erlangen kann und verhindert einen Vendor-Lock-in. Das Distributed Ledger Netzwerk wird derzeit von fünf Knotenbetreibern (Bundesdruckerei GmbH, IBM Deutschland GmbH, esatus AG, Deutsche Bahn AG und BWI GmbH) unter übergreifender Orchestrierung von IBM im Auftrag des Bundeskanzleramts bereitgestellt und in den kommenden Monaten um weitere Knotenbetreiber erweitert.

Eine Netzwerk-Governance ist hierzu in entsprechenden Steward Agreements definiert, welche sich in finaler Abstimmung zwischen dem Bundeskanzleramt und den Knotenbetreibern befinden.

Die Architektur erlaubt es Nachweise von dem Halter an den Prüfer weiterzugeben, ohne dass der Aussteller hiervon Kenntnis erlangt.

Es wurde bewusst eine über Blockchain Technologie unterstützte, verteilte Architektur gewählt, um den Single Point of Failure einer zentralen Certificate Authority zu vermeiden und die Abhängigkeit von einer zentralen Instanz auszuschließen.

Die Nutzer beantragen und empfangen in einem ersten Schritt die erforderliche VC (digitaler Führerscheinnachweis) und speichern diese in der Wallet-App. Mit diesem VC können sie gegenüber einem der teilnehmenden Unternehmen die Fahrberechtigung online nachweisen. Hierdurch entfällt die manuelle Sichtung des Führerscheindokumentes.

Eine erste einfache Darstellung des Systems ist in Abbildung 1 für den initialen Betrieb im Bereich “Flottenmanagement” gegeben.

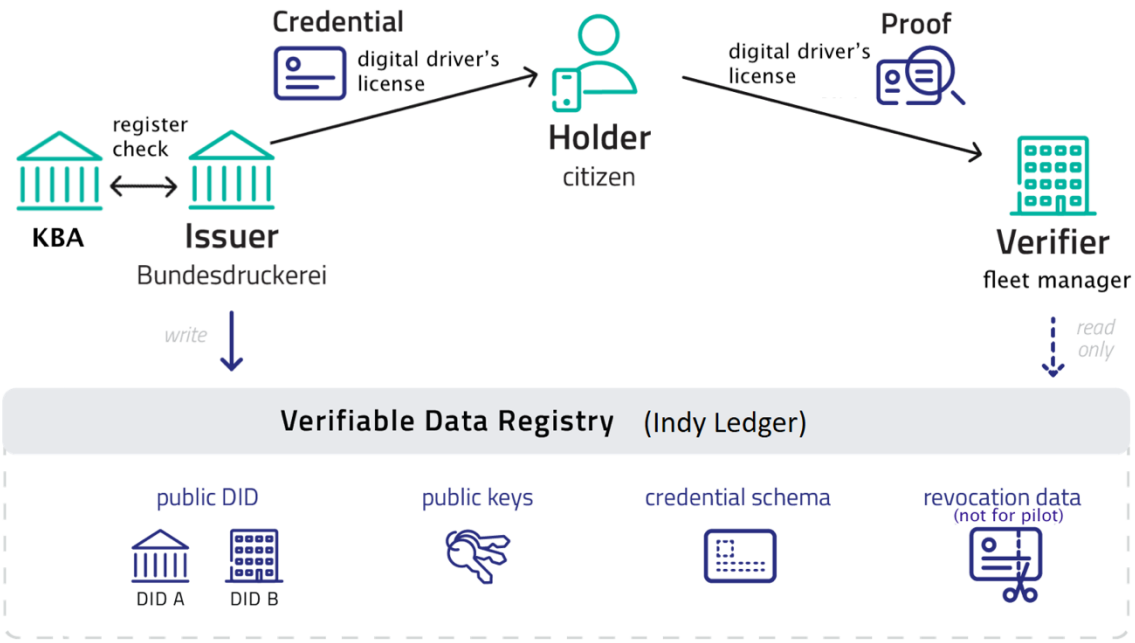


*Abbildung 1: Prozessüberblick.*

## 3.1 Systemübersicht

Das System besteht aus den folgenden Entitäten:

1. **Verifiable Data Registry:** Ein öffentliches, genehmigungspflichtiges Blockchain-Netzwerk (Hyperledger Indy bestehend aus den Indy Nodes) zur Speicherung von öffentlichem Schlüsselmaterial und dezentrale Prüfinfrastruktur, betrieben von authentisierten und autorisierten Knotenbetreibern
2. **Issuer (im Dokument auch Aussteller genannt)**: Der Issuer/der Aussteller stellt Identitäten in Form von Credentials an den Holder aus. Betreiber des Issuers ist die Bundesdruckerei (für den digitalen Führerscheinnachweis im Auftrag des KBAs, welches die Herausgeberschaft mit dem Start des Betriebs im rechtlichen Sinn übernimmt).

**Holder (im Dokument auch Halter):** Die Nutzer erhalten die Credentials von dem Aussteller, speichern diese in ihrer Wallet und können diese gegenüber Verifiern nachweisen.

**Verifier**: Der Verifier ist der Empfänger von Credentials und den darin enthaltenen Identitätsattributen (“Claims") oder logischen Aussagen aus diesen und stößt den Verifizierungsprozess über das Netzwerk an, um die Integrität, Authentizität der Daten zu prüfen. Auf den Servern der initial geplanten Flottenbetreiber erfolgt dabei der Prozess vom Erstellen des QR-Codes bis zur Prüfung gegen das Blockchain Netzwerk. Alternativ wird nach Bedarf für weitere Teilnehmer ein zentraler Verifikationsservice angeboten, welcher den Überprüfern der Fahrerlaubnis (wie Flottenbetreibern oder Mietwagen- und Carsharing Firmen) einen zentralen Service anbietet, welcher Verifizierungsanfragen empfängt, die Verifizierung gleichermaßen wie die hier beschriebenen Lösungen durchführt und das Ergebnis über eine Schnittstelle an den Überprüfer übermittelt. In dem Fall wird der zentrale Verifizierungsservice durch IBM im Auftrag des Bundeskanzleramtes zur Verfügung gestellt und betrieben.

An dem aktuellen Anwendungsfall “Flottenmanagement” sind die folgenden Entitäten beteiligt:

1. Die Bundesdruckerei als Betreiber des Issuers (KBA tritt als Herausgeber auf)
2. Das KBA als Einsprungpunkt in den Ausgabeprozess sowie als Quelle der Attribute des digitalen Führerschein-Credentials
3. Nutzer mit ihrer Wallet-App auf einem Smartphone
4. BWI GmbH, Deutsche Bahn AG, Bundesdruckerei GmbH, IBM Deutschland GmbH, esatus AG als Knotenbetreiber, die das Blockchain-Netzwerk zur Verfügung stellen
5. Volkswagen AG, BMW AG (sowie Bosch, noch zu verifizieren) als Unternehmen “Flottenbetreiber”, welche verifizierte Führerscheindaten nach Anforderung vom Halter und dessen Freigabe erhalten; perspektivisch werden die derzeit geplanten Flottenbetreiber durch weitere Unternehmen mit Bedarf zur Führerscheinverifizierung erweitert (bspw weitere Flottenbetreiber, Mietwagen und Car-Sharing Unternehmen).

Weitere beteiligte technische Komponenten im System beinhalten:

1. **Wallet-App:** zum Speichern, Freigeben und Übertragen von Credentials, herausgegeben von der Digital Enabling GmbH, eine Schwestergesellschaft der esatus AG.
2. **Aries Agents**, Software-Agenten, die SSI-Funktionalitäten zum Austausch mit anderen Agenten und zur Kommunikation für das Verifiable Data Registry (Hyperledger Indy) zur Verfügung stellen und mit der Wallet-App kommunizieren zu können, betrieben von allen Beteiligten.
3. **Mediator-Service**, der Mediator-Service agiert als Notifizierungsdienst und Inbox für eingehende SSI-bezogene DIDComm-Nachrichten für mobile Endgeräte (ausgehende Kommunikation erfolgt direkt mit den entsprechenden Endpunkten). Dort werden zusätzlich verpackte Nachrichten in der jeweils individuellen Inbox abgelegt. Nach Erhalt einer solchen Nachricht sendet der Mediator-Service zudem über einen Azure Notification Hub eine Nachricht an die Endgeräte. Diese Nachricht informiert lediglich über den Erhalt einer neuen Nachricht und gibt der Wallet Bescheid über eine eingehende Kommunikation ohne Verweis auf den Inhalt oder den Inhalt selbst. Der Mediator-Service selbst ist nicht in der Lage die Nachrichten zu lesen, die Entschlüsselung ist nur mit den auf den Endgeräten liegenden Schlüsseln für die jeweilige Verbindung, von welcher die Nachricht versandt wurde, möglich. Dieser wird von esatus betrieben.

Zusätzlich integriert das System mit den folgenden Systemen, welche sich außerhalb des hier beschriebenen System Kontextes befinden.

1. **ZFER**: Das zentrale Fahrererlaubnisregister, welches das KBA betreibt.
2. **Originiator**: Beschreibt ein beliebiges Dritt-System (z.B. ein Flottenmanagement System), welches den Verifikationsprozess initiiert.

## 3.2 Unterschiede zur Basis-ID

Die Basis-ID wird im Dokument Systemkonzept [SYS21] beschrieben. Der digitale Führerscheinnachweis weist viele Gemeinsamkeiten mit der Basis-ID auf. Es gibt jedoch auch einige Unterschiede. Diese sind der Inhalt des Credentials an sich und die Quelle der Attribute dieser Inhalte. Grundsätzlich entsprechen die im ausgestellten Führerschein Credential enthaltenen Attribute denen des physischen Führerscheins.

Bei der Basis-ID sind dagegen Daten aus dem Personalausweis als Attribute vorhanden. Es gibt hier eine Überschneidung von Name, Vorname, Geburtsort und Geburtsdatum. Die übrigen Attribute unterscheiden sich. Eine vollständige Liste der der Attribute befindet sich in Anhang B dieses Dokuments.

Für die Basis-ID werden die Attribute direkt aus dem Personalausweis über NFC ausgelesen und in der Basis-ID gespeichert. Für den digitalen Führerscheinnachweis gibt es hier eine Indirektion in Form einer Anfrage an das Zentrale Fahrerlaubnisregister (ZFER) betrieben vom KBA. Die über die AusweisApp2 aus dem Personalausweis ausgelesenen Daten Name, Geburtsname, Vorname, Geburtsort und Geburtsdatum werden an das KBA übertragen, um den Inhaber des zugehörigen Führerscheindatensatzes zu identifizieren. Bei eindeutiger Übereinstimmung wird der zugehörige Datensatz des KBAs als

Registerauskunft an den Issuer der Bundesdruckerei übertragen, welcher mittels dieser Daten das Credential ausstellt. Der Halter hat vor Ausstellung des Credentials die Möglichkeit, die Ausstellung abzulehnen. Des Weiteren hat der Halter die Möglichkeit, das ausgestellte Credential in seiner Wallet zu löschen.

Die Gültigkeit des digitalen Führerscheinnachweises in der Wallet ist prinzipiell nicht begrenzt. Die im Flottenmanagement initial teilnehmenden Unternehmen prüfen jedoch das Credential auf Aktualität und akzeptieren im Verifizierungsvorgang nur ein tagesaktuell ausgestelltes Führerschein-Credential. Der Abgleich erfolgt zwischen Datum der Ausstellung und Datum des Verifizierungsvorgangs, welche beide am selben Tag stattfinden müssen. Es sind jedoch andere Anwendungsfälle denkbar, bei denen auf die Notwendigkeit der Tagesaktualität verzichtet werden kann.

Es wird bereits in der ersten Ausbaustufe dieselbe Gerätebindung umgesetzt wie sie für die Basis-ID implementiert wurde.

Die Revozierungsmöglichkeit ist in einer späteren Ausbaustufe geplant. Mit Umsetzung der Revozierung entfällt die in der Verifizierung geplante Prüfung auf tagesaktuelle Ausstellung des Führerscheinnachweises.

# 4 Funktionale Beschreibung des ID Lifecycles

Die sichere Umsetzung elektronischer Identitäten erfordert verschiedene Basisfunktionalitäten. Die Dokumentation wie elektronische Identifizierungssysteme diese Basisfunktionalitäten umsetzen ermöglicht es, diese Identifizierungssysteme hinsichtlich der verschiedenen Standards, Normen und Richtlinien gem. der Informationssicherheit und verschiedener Vertrauensniveaus zu evaluieren.

Im Folgenden wird die Basisfunktionalität des Führscheinnachweises allgemein in Form eines eID Life-Cycles beschrieben. Die einzelnen Phasen des ID Lifecycles orientieren sich hierbei an etablierten Standards [ISO24] [EIA12] [WBG16] [BTR07].

## 4.1 Enrolment

Der Enrolment Prozess hat zum Ziel das SSI-Credential für den Führerscheinnachweis im SSI-Wallet des Holders zu hinterlegen. Als Besonderheit an diesen Prozessschritt ist die Anforderung an die Identitätsprüfung des Holders zu betrachten. Da an diese hohe gesetzliche Anforderung gestellt werden, wird dazu auf Daten aus der eID Funktionen des Personalausweises zurückgegriffen. Dies ermöglicht es, ein hohes Vertrauensniveau zu erreichen.

### 4.1.1 Identitätsprüfung

Die Anforderung der Nutzung der eID Funktion des Personalausweises kommt durch das KBA. Die Ausstellung des digitalen Führerscheinnachweises muss an eine Identitätsprüfung nach dem eIDAS Level high erfolgen. Diese Anforderung ergibt sich aus dem Paragraphen 58 des StVG.

### 4.1.2 Ausgabe des Authentisierungsmittels

Der funktionale Ablauf des Enrolment-Prozesses ist in Abbildung 2 dargestellt. Die involvierten Entitäten werden im Folgenden gelistet:

- Der Holder/Nutzer mit seiner eID-Karte (Personalausweis)
- Das mobile Endgerät des Nutzers, auf dem die SSI-Wallet-Applikation mit dem eID-Client (AusweisApp2 SDK) vorhanden sind, in Verbindung mit dem Mediator Service
- Der Issuer (Bundesdruckerei im Auftrag des KBA) in Verbindung mit einem Aries Agent
- Der Ausweisident-Dienst (betrieben durch D-Trust im Sinne eines Identifizierungsdiensteanbieters) der Bundesdruckerei bestehend aus dem eID-Server (und einem eService)
- Eine vom KBA gehostete Website als Einstiegspunkt für Nutzer in den Prozess
- Das KBA-Backend für Registeranfragen für die Daten des Führerscheinnachweises

Zwischen diesen Entitäten kommt es beim Enrolment für das Führerschein-Credential zu einem festen Protokollablauf. Dieser lässt sich in mehrere voneinander abhängige und teils ineinander geschachtelte Teilprozesse untergliedern:

1. Einsprung in den Prozess
2. SSI-Credential (Führerscheinnachweis) Issuance/Ausstellung
3. eID basierte Identifizierung und Datenfreigabe mit dem Personalausweis
4. Registerabfrage durch Übermittlung der eID-Daten an das KBA

Die Teilprozesse “eID basierte Authentifizierung und Datenfreigabe mit dem Personalausweis“ und “Registerabfrage durch Übermittlung der eID-Daten an das KBA” ist dabei innerhalb des Prozesses “SSI-Credential Issuance” eingebettet. Die Teilprozesse sind in Abbildung 3 dargestellt und werden im Folgenden beschrieben. Dabei wird die Reihenfolge der tatsächlichen Ausführung berücksichtigt und somit der “SSI-Credential Issuance” durch die beiden folgenden unterbrochen.

#### *4.1.2.1 Einsprung in den Prozess*

Der Einsprung in den Ausstellungsprozess soll den Nutzer vorbereiten, den Ausstellungsprozess zu durchlaufen und den technischen Prozess direkt anstoßen. Dazu geht der Nutzer auf die KBA-Webseite, wo er alle notwendigen Informationen zur Ausstellung eines digitalen Führerscheinnachweises erhält. Hier kann er sich über technische Voraussetzungen, rechtliche Rahmenbedingungen und praktische Anwendungsmöglichkeiten informieren.

Zur Erstellung des digitalen Führerscheinnachweises kann entweder ein QR-Code oder ein direkter Link verwendet werden. Die im QR-Code enthaltene Information ist der direkte Link, bzw. Intent/Deeplink. Den QR-Code scannt der Nutzer mit seinem Smartphone, das den Link interpretiert und mit der entsprechend zugeordneten Anwendung, der Wallet-App, öffnet.

Den direkten Link zur Erstellung des digitalen Führerscheinnachweises muss der Nutzer direkt in seinem Browser anklicken.

Ein Beispiel für einen Deeplink:

didcomm://eid-integrated?eid_uc=de.kba.fs-nachweis

Durch Öffnen des Deeplinks erhält der Nutzer in der Wallet eine Anzeige, dass die Ausstellung des digitalen Führerscheinnachweises durch Einholen einer Auskunft aus dem KBA-Register gestartet werden kann. Nach Bestätigung wird der Nutzer zur hierfür notwendigen Identifizierung der Dialog der Identifizierungsanfrage mittels Personalausweis angezeigt. Hierbei wird eine Anfrage durch den AusweisIdent-Service der Bundesdruckerei gestellt, welcher die zu übermittelnden Daten angezeigt.

Der Deeplink löst einen festen Prozess aus. Die Endpunkte für den Einsprung in den eID Prozess sind in der ID Wallet fest hinterlegt.

Der von der Bundesdruckerei als Issuer des Credentials herausgegebene API Key (128 Bit UUID, Walletherausgeber- und versionsspezifisch) wird über einen out-of-band Kanal von der Bundesdruckerei GmbH an den Walletherausgeber einmalig für jedes Versionsudpate der ID Wallet übermittelt. Der Prozess ist folgendermaßen definiert:

1. UUID wird in von der Bundesdruckerei in ihrer Deployment Umgebung generiert

2. UUID wird mittels sicherem Transportmedium (verschlüsselte und signierte eMail) an den Walletherausgeber geschickt.

3. Der Walletherausgeber bindet den erhaltenen API Key als Konstante (X-API-KEY) in den Quellcode der ID Wallet App ein. Um ein unbefugtes Extrahieren zu erschweren, wird der API Key obfuskiert. Die eigentliche Authentisierung ggü. den Endpunkten erfolgt via X-API-Key Feld im HTTP-Header.

Beim Ausstellen des digitalen Führerscheins wird die Authentizität des Issuers (Bundesdruckerei) durch Public Key Pinning des Bundesdruckerei-TLS-Zertifikats in der Wallet App sichergestellt. Das Bundes-

druckerei-TLS-Zertifikat ist in der Wallet-App enthalten. Der Wallet-App-Herausgeber erhält dieses Zertifikat von der Bundesdruckerei über sicheren E-Mail-Versand.

Der vom Nutzer angestoßene Ablauf der Ausstellung des Credentials ist an das Vertrauen des Nutzers in die Quelle des Deeplink bzw. QR-Codes gebunden. Der Prozess kann mit einem passenden Deeplink oder QR-Code aus jeder Quelle initiiert werden. Bei einer anderen Quelle würde der Nutzer aber den folgenden Ablauf nicht erwarten und würde ggf. bei der oben genannten Anzeige zur Ausstellung des Credentials den Prozess abbrechen.

#### *4.1.2.2 SSI-Credential Issuance*

Die Ausstellung des digitalen Führerscheinnachweises als SSI-Credential ist zunächst der gleiche Prozess wie im Systemkonzept der Basis-ID beschrieben. Im Unterschied zur Basis-ID kommen die Daten für den Inhalt des zu erstellenden Credentials vom KBA. Das KBA gibt auf Basis der Daten aus dem eID-Workflow (Name, Vorname, Geburtstag, Geburtsort, Geburtsname) eine Registerauskunft an die Bundesdruckerei, welche das SSI-Credential, den digitalen Führerscheinnachweis damit ausstellt. Die Zuordnung der Anfrage und zugehöriger Ausstellungen des Credentials erfolgt über den Session-Key, welcher mit dem Aufruf der Ausstellung erstellt wird.

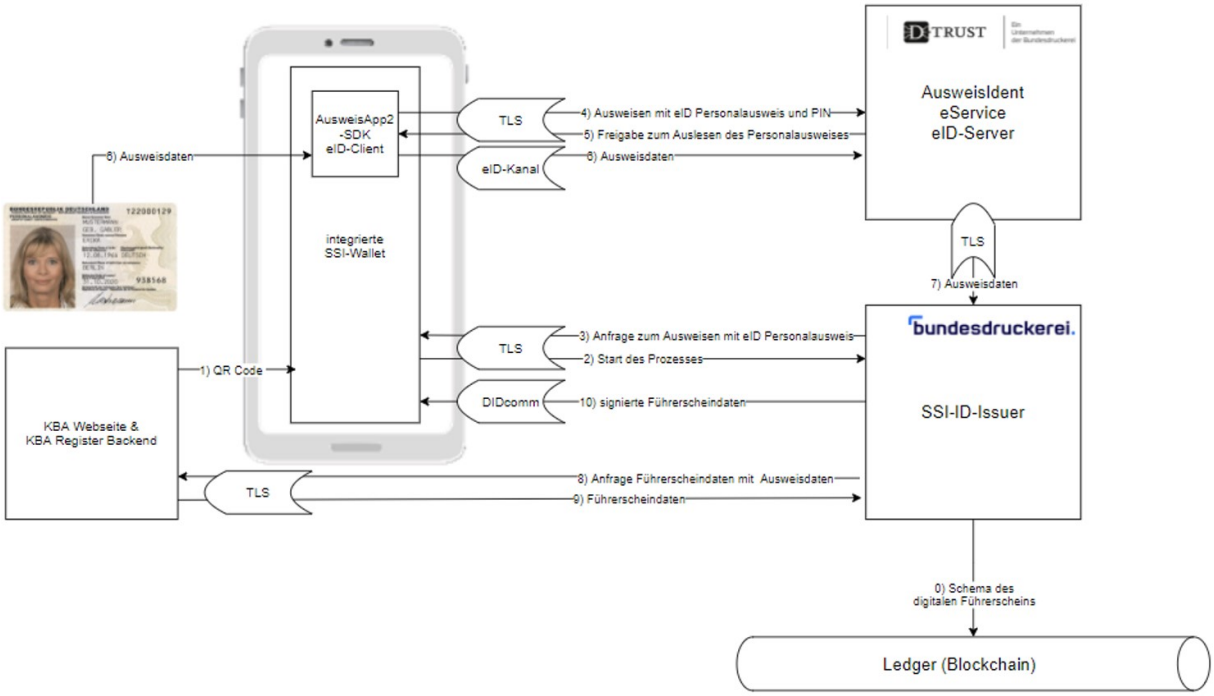


*Abbildung 2: Enrolment Sequenz-Diagramm.*

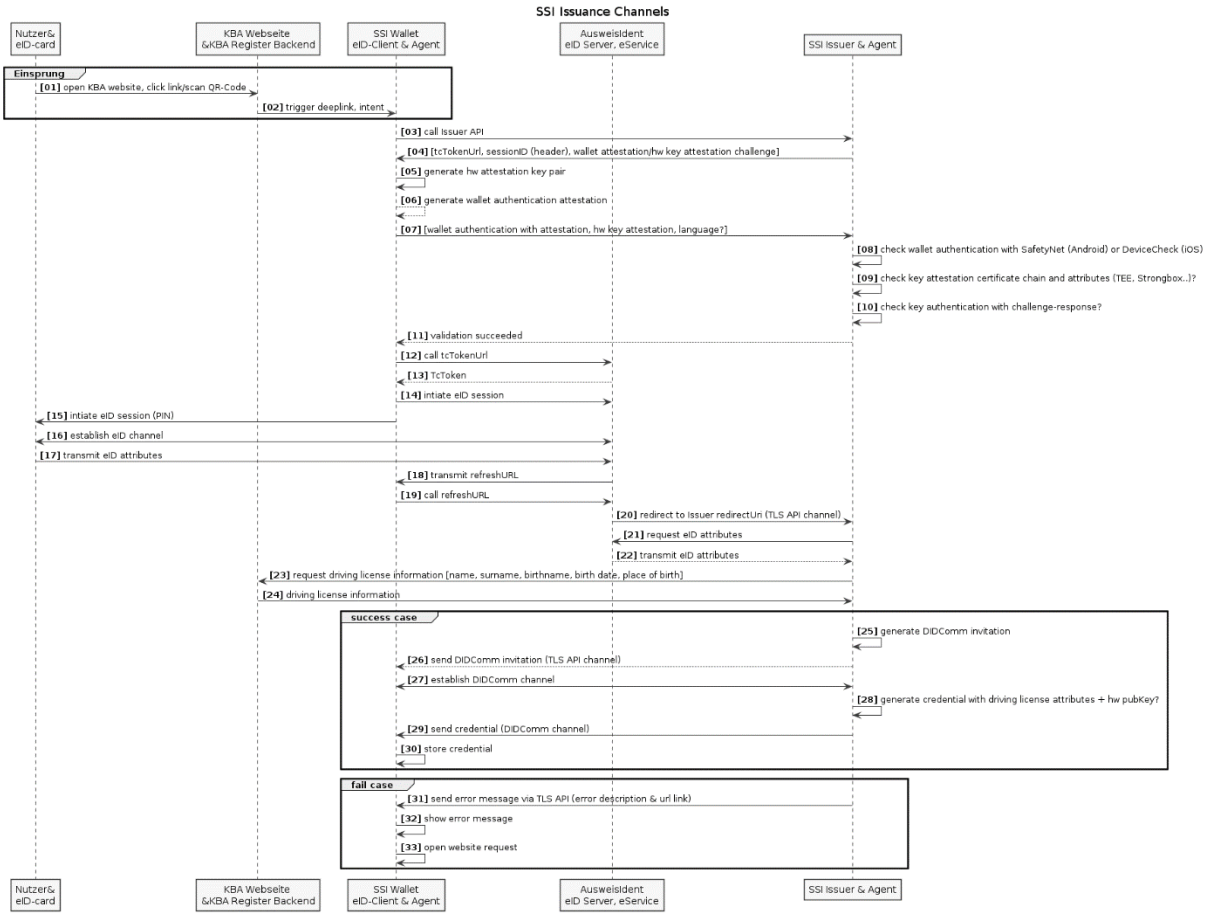


*Abbildung 3: Credential Issuance Sequenzdiagramm.*

Der Ausstellungsprozess wird im integrierten Flow durch den Nutzer in der Wallet-App initiiert. Dieser startet den Prozess mit dem autorisierten Aufruf des Bundesdruckerei-API-Endpunkts (über einen TLS-gesicherten Kanal). Die Wallet authentifiziert sich mittels eines API-Keys (Wallet- und App-versionsspezifisch). Der Nutzer stimmt den Nutzungsbedingungen und der Datenschutzerklärung des Ausstellers zu.

Es findet keine technische Einschränkung bezüglich der Erstellung sowie Nutzung des digitalen Führerscheinnachweises statt, so dass alle Personen, die über einen Personalausweis mit Online Ausweisfunktion verfügen sowie im KBA Register stehen sich den digitalen Führerscheinnachweis ausstellen lassen und in der ID Wallet speichern können.

#### 4.1.2.3 Identität verifizieren/ eID basierte Identifizierung und Datenfreigabe mit dem Personalausweis

Die Identifizierung des Nutzers erfolgt mittels der eID-Funktion des Personalausweises. Die Issuer-API liefert dafür den Einstiegpunkt (tcTokenUrl) für den eID-Prozess mit. Damit baut das in der Wallet integrierte AusweisApp2-SDK eine Verbindung zum eID-Server auf, prüft die Ausleseberechtigung und holt die Zustimmung des Nutzers ein. Dem Nutzer wird dabei die Bundesdruckerei GmbH als anfragender Anbieter im Kontext der Identifizierungsvorgangs angezeigt und zusätzlich die angefragten Daten (Name, Vorname, Geburtstag, Geburtsort, Geburtsname). Dieser autorisiert den Auslesevorgang mit Eingabe seiner eID-PIN in der Wallet-App. Im eID-Kanal werden die angeforderten Identitätsdaten von der eID-Karte (Personalausweis) an den eID-Server übertragen. Der eID-Server antwortet dem integrierten eID-Client nach dem erfolgreichen Auslesen mit der redirectUrl. Diese beinhaltet lediglich den Identifier für die eID-Session, der integrierte eID-Client übergibt die redirectUrl and die Wallet und diese ruft den API-Endpunkt aus der redirectUrl auf. Der Ausstellungsdienst fragt die Identitätsdaten beim eID-Server an (über einen TLS-gesicherten Kanal) und der eID-Server gibt die ausgelesenen Identitätsdaten an den Issuer-Service frei. Der Ausstellungsdienst hat jetzt die verifizierten Identitätsdaten des Nutzers.

#### 4.1.2.4 Registerabfrage durch Übermittlung der eID-Daten an das KBA

Die Identitätsdaten werden in einer Anfrage an das KBA-Backend (Zentrales Fahrerlaubnisregister – ZFER) verwendet. Sofern das KBA anhand dieser Daten einen eindeutigen Führerscheindatensatz im ZFER identifizieren kann, liefert es diesen Datensatz an den Aussteller als Attribute für das Credential zurück. Liegt keine Fahrberechtigung im Register vor, oder kann zum angefragten Datensatz kein eindeutig zuordenbarer Eintrag gefunden werden, wird eine entsprechende Fehlermeldung als Fehlercode übermittelt.

Mögliche Fehlermeldungen und Fehlercodes:

| | |
|---|---|
| 001 | Zu den übermittelten Personendaten konnte kein eindeutiger Treffer im Zentralen Fahrerlaubnisregister (ZFER) ermittelt werden. Eine Auskunft darf nur bei einer sicheren Identifizierung erfolgen. |
| 002 | Es wurde kein Treffer im Zentralen Fahrerlaubnisregister (ZFER) ermittelt - Sie sind vermutlich nicht im Besitz eines deutschen Kartenführerscheins. |
| 003 | Zu Ihrer Person sind im Fahreignungsregister (FAER) aktuell negative Eintragungen (z.B. Fahrverbot) zu Ihrer Fahrerlaubnis vermerkt. |
| 004 | Aufgrund einer Entziehung der Fahrerlaubnis konnte keine gültige deutsche Fahrerlaubnis ermittelt werden. |
| 005 | Aufgrund eines technischen Fehlers kann keine Angabe zu einer aktuell gültigen Fahrerlaubnis gemacht werden. |

Die Absicherung erfolgt zweiteilig über eine Transportverschlüsselung mit TLS 1.2 inkl. Client Authentication und einer zusätzlichen Inhaltsdatenverschlüsselung über ws-security 1.2 mit der Algorithm-Suite sp:basic256sha256. Das initiale Vertrauen zwischen den Parteien wurde über dediziert benannte Ansprechpartner beider Seiten hergestellt. Der Austausch der Schlüssel erfolgt gemäß Absprache über verschlüsselte und signierte E-Mails.

Hierfür sind im System vier voneinander verschiedene Zertifikatsketten und private Schlüssel mit Sealed Secrets - Tool zur sicheren Übertragung und Speicherung von Schlüsselmaterial aus der Entwicklung in die Openshift-Plattform - (https://github.com/bitnami-labs/sealed-secrets) hinterlegt. Der Schlüsseltyp ist RSA (2048 und 4096 Bit Schlüssellänge). Der Schlüsseltyp ist RSA (2048 und 4096 Bit Schlüssellänge). Der Private Key liegt dabei ausschließlich auf dem Controller. Der Public Key liegt in der bdr-internen Dokumentation zur Nutzung von Sealed Secrets bei und kann ausschließlich zur Verschlüsselung verwendet werden, jedoch nicht zur Entschlüsselung. Die Verwendung von Sealed Secrets hat keinen Einfluss auf die eigentlichen Keys.

Ausschließlich Openshift Administratoren können den Sealed Secret private Key lesen und verändern, der verwendet wird, um die Schlüssel, Credentials, eines Projektes auf Openshift zu entschlüsseln.

Die Kommunikation ist somit vom vertrauenswürdigen KbaRegisterAdapterService bis zur DMZ des KBA zweifach gesichert.

In der DMZ des KBAs terminiert der Load-Balancer die Transportverschlüsselung und leitet die inhaltsverschlüsselte Nachricht an den DigitalerFuehrerscheinService im vertrauenswürdigen Backend weiter. Dort findet die Entschlüsselung und Überprüfung der Signatur statt.

Der Rückweg erfolgt analog.

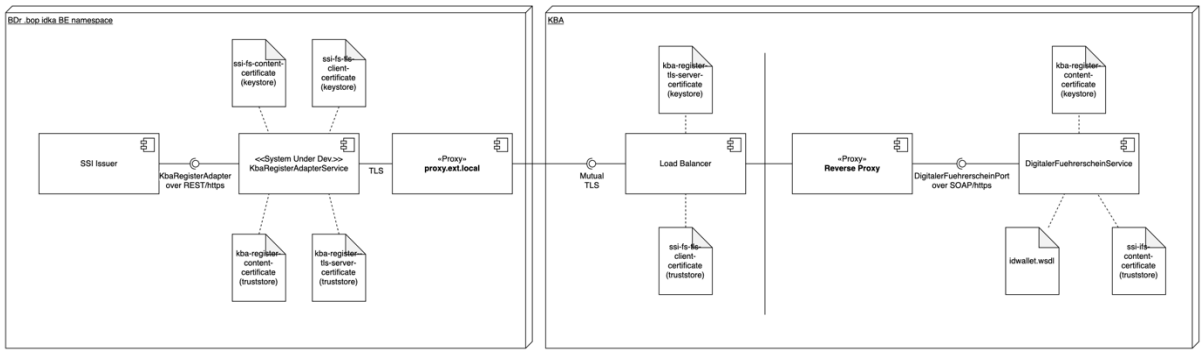


*Abbildung 4: Registerabfrage beim KBA.*

Die Eingangsdaten in den KbaRegisterAdapterService sind vom SSI Issuer (über die Online-Identifizierung) verifiziert worden und können demzufolge als vertrauenswürdig angenommen werden, so dass an dieser Schnittstelle nur einfache syntaktische Eingangsvalidierungen notwendig sind.

Der KbaRegisterAdapterService ist nur für den SSI Issuer im Netzwerk erreichbar. Das wird über Firewall-Regeln und TLS Client Authentication sichergestellt.

Der LoadBalancer akzeptiert nur Anfragen aus dem IP-Bereich des proxy.ext.local der BDr. Dies wird über Firewall-Regeln sichergestellt.

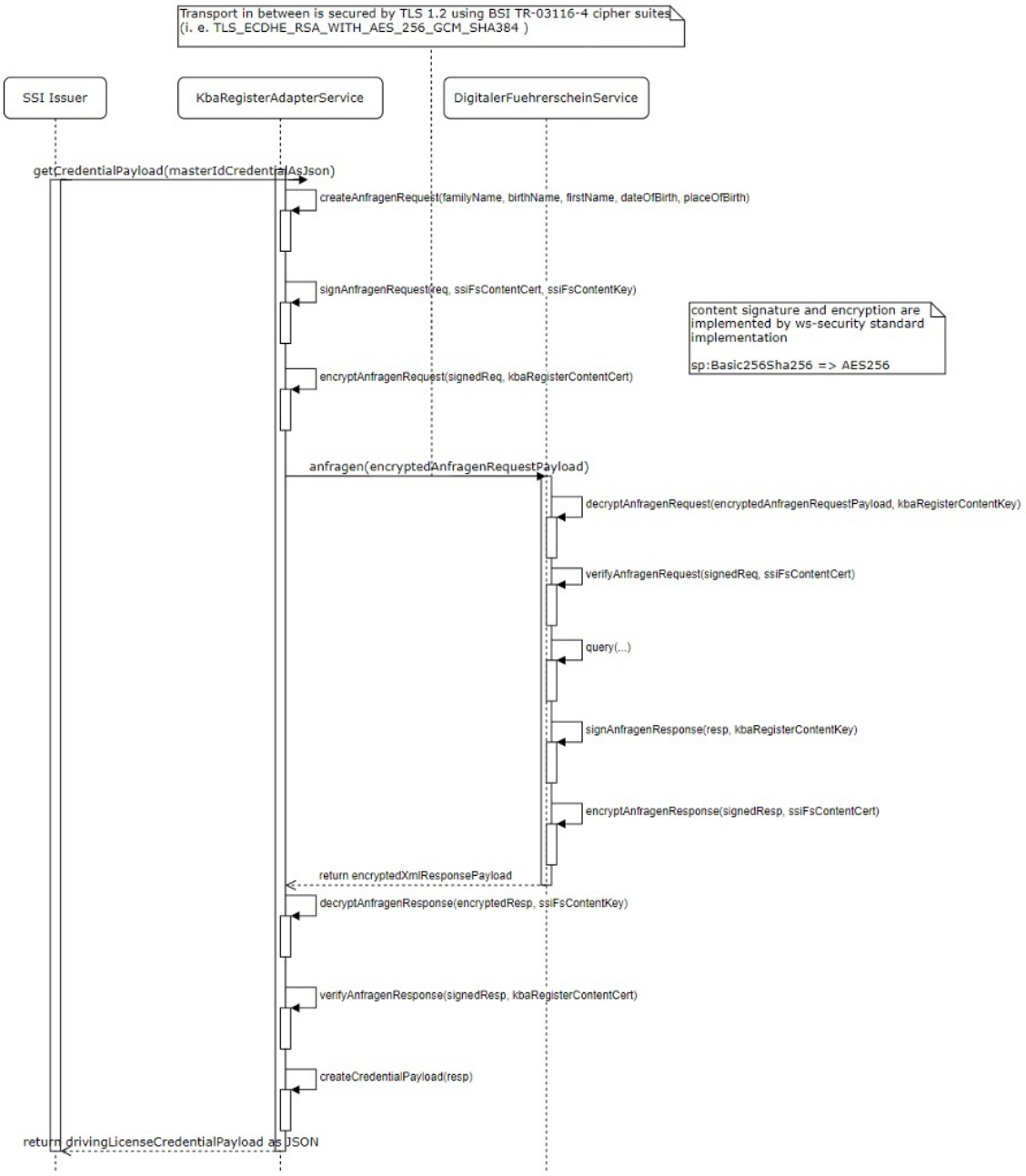


*Abbildung 5: Funktioneller Ablauf Enrolment.*

#### *4.1.2.5 Identität speichern*

Der Ausstellungsdienst (SSI-ID-Issuer) speichert keine Nutzerdaten, die verifizierten Nutzerattribute werden nur bis zur Ausstellung des Credentials temporär in einer In-Memory Datenbank (Redis) gehalten und danach gelöscht.

#### 4.1.2.6 Ausstellen / Ausliefern

Der Ausstellungsdienst antwortet an die Wallet nach dem erfolgreichen Durchlauf des eID-Prozesses mit einer DIDComm-Invitation. Diese Einladung beinhaltet die DID, kryptografische Schlüssel und Parameter für den Aufbau der Verbindung (DIDComm). Die Wallet empfängt diese und initiiert die DIDComm-Verbindung zum Aussteller. Der Aussteller kann das Credential mit den verifizierten Nutzerattributen erstellen und kryptografisch signieren und über die erfolgreich aufgebaute Verbindung an die Wallet schicken. Der Nutzer nimmt die Anfrage an und die Wallet speichert das Credential ab. Die Kanalbindung beim Ausstellen des digitalen Führerscheinnachweises wird durch folgende Kanäle erreicht (Abbildung 2: Enrollment des digitalen Führerscheinnachweises Enrolment Sequenz-Diagramm.): TLS zwischen Ausstellungsdienst (AusweisIdent/eService) und Wallet, und anschließendem Aufbau des eID-Kanals zwischen eID-Dokument und AusweisIdent/eID-Server unter Vermittlung von AusweisApp2/eID-Client. Das Ausstellen des digitalen Führerscheinnachweises als Verifiable Credential erfolgt im DIDComm-Kanal. Die enge Bindung zwischen Onlineausweisfunktion und Deployment der Credentials wird durch die Integration des AusweisApp2-SDKs in die ID Wallet App auf Softwareebene gewährleistet.

#### 4.1.2.7 Aktivieren

Sobald der Nutzer das Credential annimmt, ist dieses für die Nutzung bereit.

## 4.2 Nutzung

Der Vorgang der Nutzung unterscheidet sich technisch nicht von dem der Basis-ID. Die folgende Beschreibung ist somit analog zu [SYS21]. Lediglich die Rollen der einzelnen Entitäten werden beim Anwendungsfall Flottenmanagement von für diesen Anwendungsfall relevanten Organisationen übernommen. Durch den Verzicht auf Revocation/Sperrung sind auch die Schritte bzw. Daten, die explizit den Sperrstatus betreffen, ausgelassen worden. Dafür prüft der Verifier zusätzlich die Tagesaktualität des Credentials.

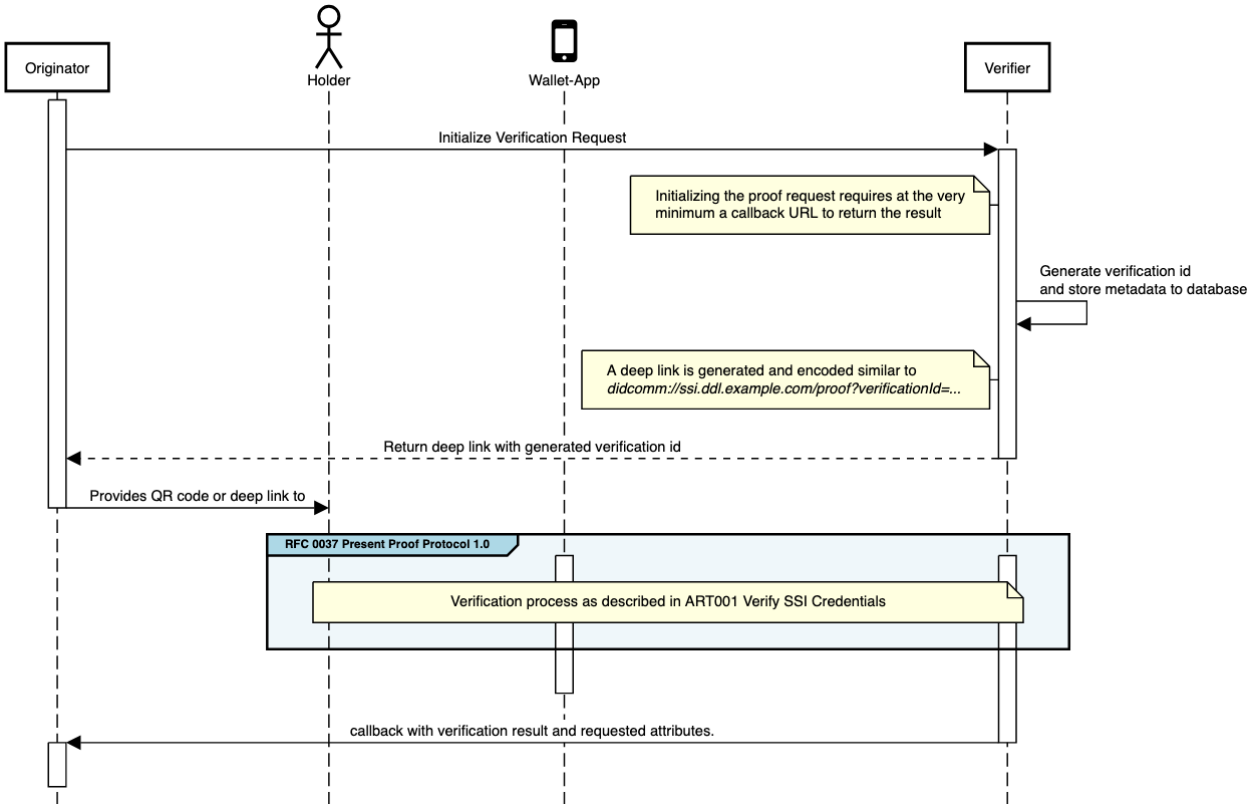


*Abbildung 6: Nutzung: genereller Verification Prozess.*

Nutzer als Mitarbeiter eines Unternehmens, das den Führerschein überprüfen muss, öffnen in einem Browser eine Website des Unternehmens und starten den Prozess der Überprüfung. Dabei wird dem Nutzer ein QR-Code bzw. deep link zugeschickt. Das kann über unterschiedliche Kanäle passieren, wie zum Beispiel per E-Mail. Der QR-Code kann auf einem beliebigen Gerät angezeigt werden und dient dazu, vom mobilen Gerät des Nutzer eingescannt zu werden. Der Link kann auf dem Mobilgerät des Nutzers direkt geöffnet werden. Das Sequenz-Diagramm Abbildung 6 beschreibt den generellen Ablauf der Integration eines "Proof Request Originators" (Originator). Sobald der QR code von dem Nutzer geöffnet wurde, fängt der generelle "RFC 00037 Present Proof Protocol 1.0" Ablauf an.

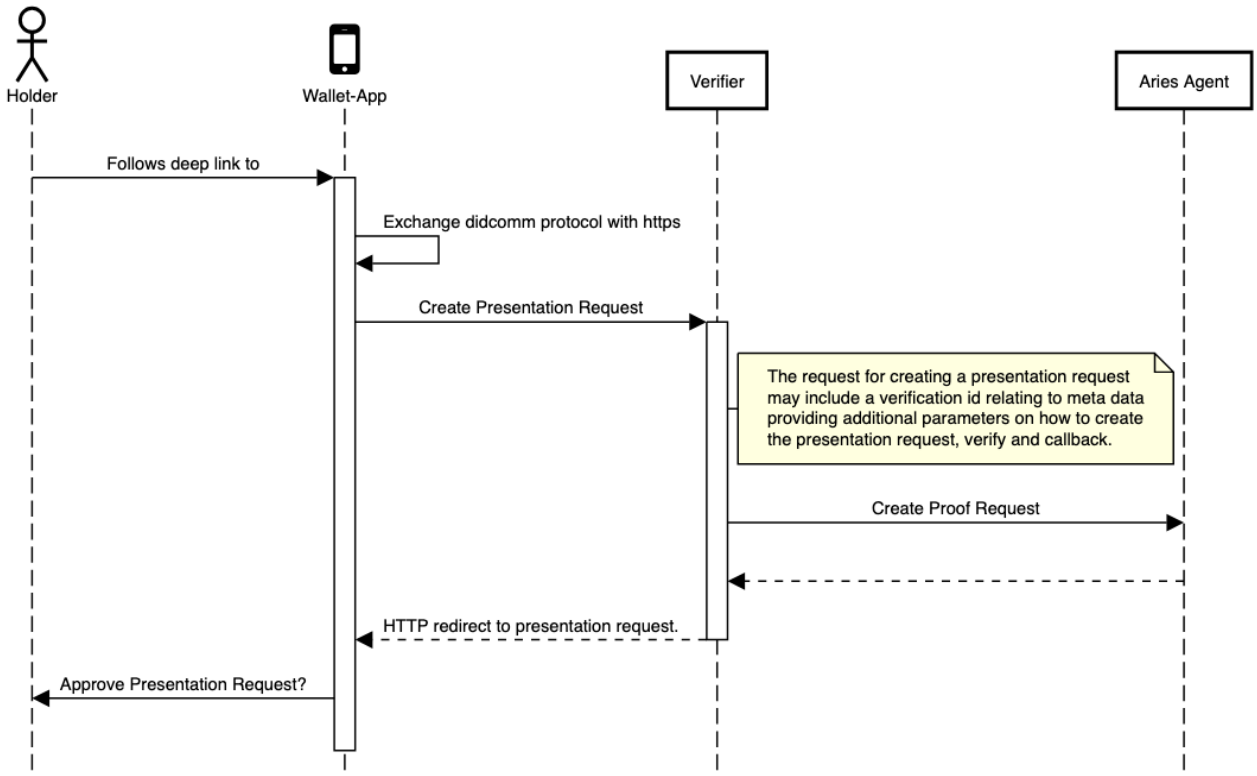


*Abbildung 7: RFC 00037 Present Proof Protocol 1.0.*

Es öffnet sich die ID Wallet-App und schickt (über einen TLS-gesicherten Kanal) eine Anfrage an den "Unternehmens Controller" (ein API Server, welcher sich im Backend des Unternehmens befindet). Das Unternehmen legt eine Referenz zu dem Vorgang in der MongoDB (innerhalb der Trust-Domain des Unternehmens) an und erstellt mit Hilfe des zugeordneten Unternehmens-Agents ("Aries Agent" / aca-py) einen sogenannten "Proof Request". Dieser "Proof Request" wird in einen "Connection-less Proof Request" transformiert und beinhaltet den Endpoint sowie Public Key des Cloud Agents des Unternehmens (für die spätere Kommunikation von Wallet-App zu Agent), eine Liste der nachzuweisenden Attribute aus dem digitalen Führerscheinnachweis. Die Authentizität des Public Key wird durch den gesicherten und authentischen HTTPS-Nachrichtenkanal sichergestellt. Hierbei erhält die ID Wallet App den HTTPS Endpoint (TLS >= 1.2) des Unternehmens (Teil der Email Invitation) und das zugehörigen Zertifikat. Nur Zertifikate aus vertrauten CAs des mobilen Betriebssystems (siehe verfügbare Liste von iOS und Android) sind zugelassen bzw. im System als vertrauenswürdige Zertifikate markiert. Ebenfalls wird auch die Einschränkung, dass der digitale Führerscheinnachweis einer bestimmten Credential Definition entsprechen muss (also von der Bundesdruckerei mit ihrem PublicKey auf dem Ledger signiert ist), mit aufgenommen.

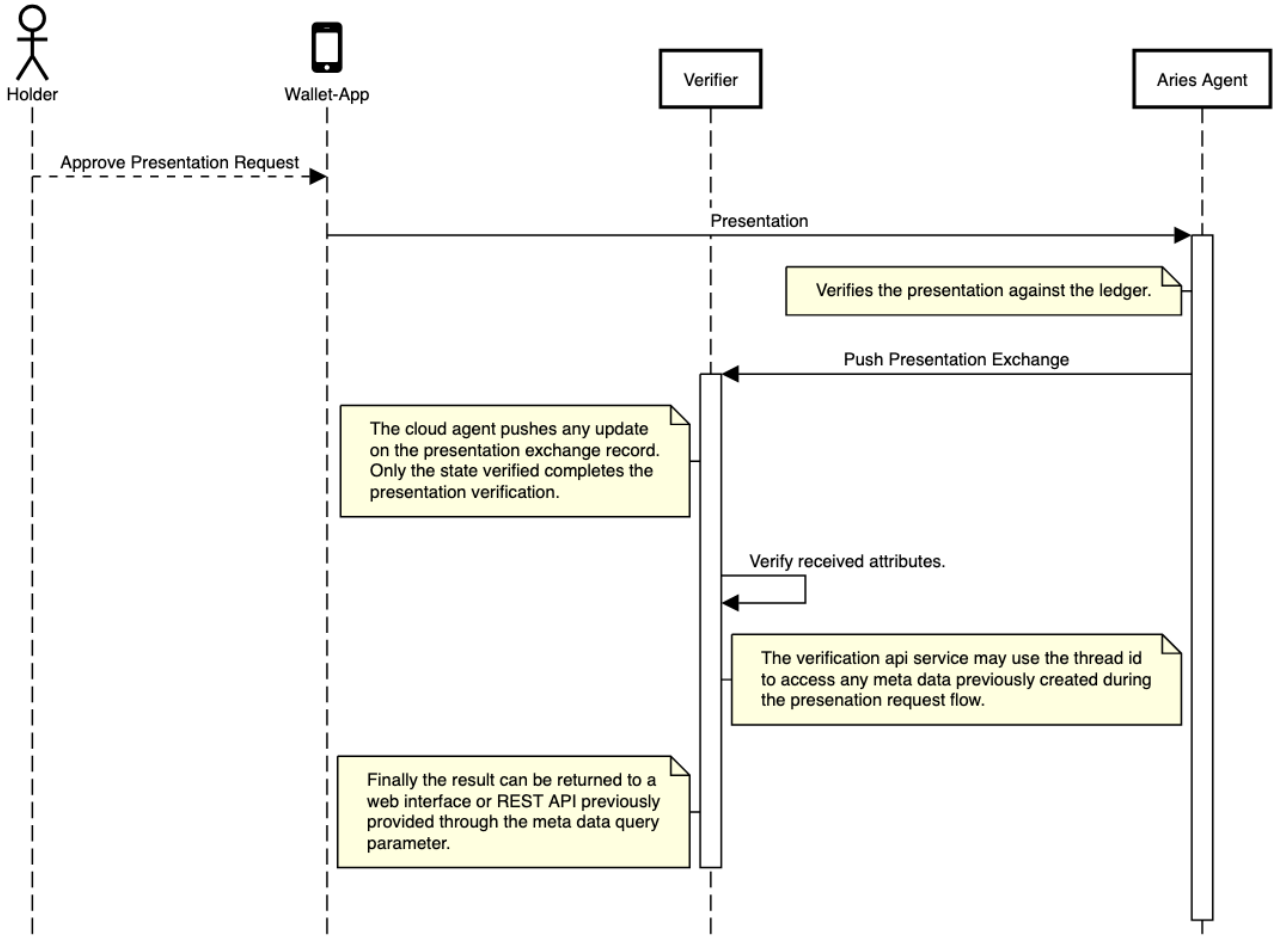


*Abbildung 8: Fortsetzung RFC 00037 Present Proof Protocol 1.0*

Über einen Redirect wird der "Connection-less Proof Request" als didcomm URI konvertiert zurück an die ID Wallet-App gegeben, wo die ID Wallet-App den "Connection-less Proof Request" verarbeitet. Die ID Wallet-App zeigt dem Nutzer den Hostname und Endpunktnamen des Verifiers an, fragt den Nutzer nach Zustimmung und erstellt dabei eine sogenannte "Verifiable Presentation" als Antwort auf den Proof Request. Hier kann der Nutzer entscheiden, ob er dem angezeigten Hostnamen und einem durch den Verifizierungs-Service definierten Namen vertrauen möchte oder nicht. Sofern er vertraut, wird lediglich eine überprüfbare Präsentation der angefragten Claims der Credentials übermittelt.

Diese Antwort enthält die geforderten Attribute aus dem digitalen Führerscheinnachweis, einen Beweis, dass diese jeweils tatsächlich aus einem VC, das die Bundesdruckerei signiert hat, entstammen. Im Gegensatz zu Zertifikaten (wie bspw. JSON-Web oder X.509) muss dabei das VC nicht vorgezeigt werden, um die Herkunft der angegebenen Werte mit Hilfe der Signatur zu beweisen (dieses Verfahren zum Nachweis, dass die Signatur vorliegt, wird als Zero-Knowledge Proof bezeichnet und ist auf Basis des Camenisch-Lysyanskaya[3]-Signaturverfahrens implementiert). Dadurch können nicht benötigte Attribute,

korrelierende Informationen wie der Wert der Signatur auf dem VC geheim bleiben und verlassen die Wallet des Nutzers nicht. Zudem verlässt das VC nicht die Wallet und kann vom Verifizierer nicht als solches verwendet werden.

1. Nutzer öffnet die App, gibt seine PIN ein und die Wallet-Datenbank wird entschlüsselt
2. Nutzer scannt mit entsperrtem Wallet den QR-Code und erhält damit eine redirectURL
3. Wallet ruft redirectURL auf und erhält eine DIDcomm-Message (Proof Request und Service End Point des Verifier)
    a. Der Proof Request enthält Anforderungen des Verifiers, welche Attribute des digitalen Führerscheinnachweises präsentiert werden sollen (inkl. **hardwareDid) und eine NONCE (80 Bit)** (nonce_indy)
    b. Zusätzlich erfolgt die Abfrage des self-attested attributes **hardwareDidProof** (dieses ist nicht Teil des Schemas/der Credential-Definition und kann vom Holder selbst befüllt werden)
4. Wallet selektiert alle zum Proof Request passenden Attribute
5. Wallet: Abfrage User Consent (mit User-Authentisierung mittels Inhaber-PIN)
    a. Der Empfänger ist über die Anzeige des Hostnamen und eines durch den Verifizierungs-Service definierten Namen zu erkennen.
6. Wallet: Abfrage Validierungsinformation aus dem Ledger über ZMQ-Verbindung (Schema-Information, Credential Definition Information)
    a. Schema des digitalen Führerscheinnachweises für die Informationen zu abrufbaren Attributnamen
    b. CredentialDefinition des digitalen Führerscheinnachweises für die Issuer-Publickeys und die ID der CredentialDefiniton zur Einschränkung des Proof Request
7. Wallet führt unter Nachweis der Inhaber-PIN das Challenge-Response-Verfahren mittels **hardwareDid** und **nonce_key_auth** durch, das Ergebnis wird im Attribut **hardwareDidProof** gespeichert,
    **a. nonce_key_auth** = SHA256(nonce_indy | 0x02)
    **b. hardwareDidProof** = SHA256withECDSA (**nonce_key_auth**)
8. Wallet erzeugt Zero Knowledge Proof (CL)
    a. Primary Proof über Attribute, Link Secret und NONCE
    b. Non Revocation Proof
9. Wallet: Versandt des Proofs an Service End Point des Verifier
10. Verifier: Abfrage Validierungsinformation aus dem Ledger über ZMQ-Verbindung (Schema-Information, Credential Definition Information)
    a. Schema des digitalen Führerscheinnachweises für die Attributnamen
    b. CredentialDefinition des digitalen Führerscheinnachweises für die Issuer Public Keys und die ID der CredentialDefiniton zur Einschränkung des Proof Request
11. Verifier prüft auf Richtigkeit
    a. Signaturen (inkl. NONCE)
    b. Validierungsinformationen
    c. Tagesaktualität
    d. Korrektheit des **hardwareDidProof**

Sobald die Antwort beim Agent (Verifier) von der Wallet-App (Sender) eingeht, prüft dieser den Beweis auf Korrektheit mittels eines NI-ZKP auf Basis des Camenisch-Lysyanskaya-Signatur-Verfahrens und benachrichtigt zudem über einen Webhook den Unternehmens-Controller. Dieser fragt anschließend die im Beweis enthaltenen Daten aus dem Unternehmens-Agent ab, verifiziert diesen auf Tagesaktualität und veranlasst anschließend die Löschung aller Informationen im Unternehmens-Agent, die im Zuge des Nachweisprozesses entstanden sind. Danach übermittelt der Unternehmens-Controller die aus dem digitalen Führerscheinnachweis abgefragten Daten an das Unternehmens-Backend, um die Führerscheinüberprüfung abzuschließen.

### 4.2.1 Deployment Erwägungen

Die Architektur des Verifikation Service, erlaubt es, diesen zentral, aber auch dezentral zu betreiben. Dabei sind die folgenden zwei Szenarien vorstellbar:

- Szenario 1 (dezentral): Ein Unternehmen betreibt den Verifikation Service inklusive aller benötigten Infrastruktur Komponenten (z.B Agent) für sich selbst.
- Szenario 2 (zentral): Ein Unternehmen verbindet sich zu einem zentralen Verifikation Service mittels einer REST Schnittstelle. Dieser zentrale Verifikations Service wird vom Bundeskanzleramt betrieben und den Verifizierern bereitgestellt.

### 4.2.2 Initiales Vertrauen

Um das Initiale Vertrauen zwischen "Originator” und "Verification Service” aufzubauen wird der folgender organisatorischer Prozess definiert.

1. Das KBA erlaubt einem “Originator” den zentral betriebenen “Verification Service" zu nutzen und teilt dem Unternehmen Onboarding Dokumentation (z.B. API Dokumentation, Test Netzwerk Addressen, etc.) mit.
2. Das KBA generiert einen unternehmensspezifischen API-Key, mit welchem API Aufrufe autorisiert werden und teilt diesen ebenfalls dem Unternehmen mit. Diese Key Übergabe passiert außerhalb des hier beschriebenen Systems über einen sicheren Kanal. (z.B. verschlüsselte email)

Dieses Vertrauensverhältnis ist nur dann aufzubauen, wenn die Verifikation als zentraler Service (Szenario 2) verwendet wird. Sollte sich ein Unternehmen entscheiden, den Verifikation Service selbst zu betreiben (Szenario 1) gibt es keine Authentifizierung. Das Unternehmen muss lediglich sicherstellen, dass es sich mit dem richtigen Netzwerk verbindet.

## 4.3 Verwaltung

### 4.3.1 Aktualisieren

Ein mehrmaliges paralleles Ableiten des digitalen Führerscheinnachweises ist möglich, eine Aktualisierung bzw. Übersicht des Nutzers über den Status seiner Ableitungen ist zu einem späteren Zeitpunkt angedacht.

Durch die Anforderung an Tagesaktualität des digitalen Führerscheinnachweises auf Seiten der Verfizierers muss es möglich sein, die Ausstellungen in kurzen Intervallen anzustoßen. Ein ggf. noch gültiges Credential wird bei einer erneuten Ausstellung in derselben ID Wallet automatisch gelöscht und durch das neu ausgestellte ersetzt.

### 4.3.2 Aussetzen / Sperren

Die Revocation/Sperrung wird in einer zukünftigen Ausbaustufe umgesetzt.

### 4.3.3 Reaktivieren

Ein Reaktivieren ist nicht vorgesehen. Stattdessen wird das Credential erneut ausgestellt.

### 4.3.4 Ersetzen

Jede erneute Ableitung des Credentials in dieselbe Wallet ersetzt zuvor abgeleitete Credentials. Da der Führerschein aufgrund der Anforderungen der Verifizierer, nur am Tage der Ausstellung nutzbar ist, ist miteiner erneuten Ableitung zu rechnen. Diese ersetzt programmatisch ein bereits vorhandenes Credential, indem das ältere gelöscht wird.

## 4.4 Widerruf

Ein Widerruf ist gegenwärtig nicht unterstützt. Der Nutzer kann lokal seine Credential löschen. Dabei werden alle nutzerbezogenen (pseudonymen) Daten gelöscht.

# 5 IT-Sicherheitsbetrachtung

Zur Dokumentation und zum Nachweis der IT-Sicherheitsmechanismen des digitalen Führerscheinnachweises werden sicherheitskritische Aspekte bzw. Phasen des Lebenszyklus näher betrachtet. Die Betrachtung der Aspekte bzw. Phasen erfolgt angelehnt an etablierte Standards zur Bewertung von digitalen Identitätslösungen [EIA12] [BTR07].

Bemerkung:

Die im Folgenden bewertete Identitätslösung weist zwei Besitzfaktoren auf. Zum einen das Link secret im Rahmen des Zero Knowlegde Proofs sowie das in den AnonCred als Attribut hinterlegte asymetrische, hardwaregebundene Schlüsselpaar hardwareDid. Im Rahmen des ZKP wird die Authentizität des

Credentials sichergestellt, und damit auch die Authentizität der hardwareDid. In einem zweiten Schritt (Challenge Response Verfahren) wird dann die Gerätebindung des Credentials mittels des asymetrischen, hardwaregebundenen Schlüsselpaars nachgewiesen. Soweit möglich werden beide Besitzfaktoren in der folgenden Bewertung berücksichtigt, jedoch ist die hardwareDid ausschlaggebend für die Einschätzung der Sicherheit der Identitätslösung.

## 5.1 Enrolment

### 5.1.1 Identitätsprüfung

**Anforderung „Identitätsprüfung“ (KBA):** Die Identitätsprüfung zur Ausgabe des digitalen Führerscheinnachweises muss eIDAS LoA high entsprechen.
**Umsetzung:** Die initiale Identitätsprüfung bei der Ausgabe des digitalen Führerscheinnachweises findet über eine Identifizierung mittels Onlineausweis-Funktion über das AusweisApp2 SDK in der Wallet-App statt. Die Onlineausweis-Funktion über das AusweisApp2 SDK erfüllt eIDAS LoA high.

### 5.1.2 Ausgabe des Authentisierungsmittels

**Anforderung „Ausgabe nur an berechtigte Inhaber“ ([BTR71], G2):** Es muss sichergestellt werden, dass Authentisierungsmittel nur an den berechtigten Inhaber ausgegeben werden.
**Umsetzung:** Die Kanalbündelung zwischen Onlineausweisfunktion und Ausstellen der Credentials wird durch die Integration des AusweisApp2 SDKs in die ID Wallet App auf Softwareebene gewährleistet (siehe Kapitel 7.1.3 [SYS21]). Über die Onlineausweis-Funktion des AusweisApp2 SDK werden innerhalb der eID Session die Anfrageattribute Familiename, Vorname, Geburtstag, Geburtsort, Geburtsname eID-Karte über den TLS API Kanal an den SSI Issuer vertrauenswürdig übermittelt.

Mit diesen Anfrageattributen führt der SSI Issuer im KBA-Register eine inhaltsverschlüsselte und –signierte, synchrone Anfrage über eine gegenseitig authentisierte TLS Verbindung aus.

Das KBA Register kann die übermittelten Anfrageattribute aufgrund der zweifach gesicherten Übertragung als vertrauenswürdig annehmen. Das KBA Register identifiziert anhand der übermittelten Anfrageattribute den zugrunde liegenden Kartenführerschein. Dafür wendet es einen entsprechenden Matching-Algorithmus an, der nur bei einer hinreichend hohen Punktzahl das Ergebnis als eindeutig bewertet und den Anfragesteller als berechtigt annimmt. Das KBA Register liefert nur eindeutige Ergebnisse an den berechtigten Aufrufer zurück.

Der SSI Issuer erhält das Anfrageergebnis vom KBA Register über den synchronen Aufruf und die zweifach gesicherte Verbindung zurück und kann daher die Vertrauenswürdigkeit des Anfrageergebnisses annehmen.

Der SSI Issuer stellt anhand des vertrauenswürdigen Anfrageergebnisses das Credential für den als berechtigt bewerteten Anfragesteller aus und übermittelt es über eine dedizierte, vertrauenswürdige didcom-Verbindung an die SSI Wallet, so dass es dann als Authentisierungsmittel genutzt werden kann.

Zusammenfassend wird die Kanalbündelung erreicht, indem die Wallet zwei TLS-Kanäle mit dem Issuer aufbaut (verbunden über eine Session ID) und über diese TLS-Kanäle sowohl der eID-Kanal als auch der DIDComm-Kanal initiiert und von der integrierten Wallet verwaltet werden. Weiterhin wird so auch der der TLS-Kanal vom Issuer zum KBA-Register initiiert.

### 5.1.3 Informationen für den Inhaber

**Anforderung „Geschäftsbedingungen und Verhaltensregeln“ ([BTR71], G4):** Dem Inhaber der Authentisierungsmittel müssen in geeigneter Weise die Geschäftsbedingungen sowie notwendige Verhaltensregeln zum Umgang mit den Authentisierungsmitteln übermittelt werden.
**Umsetzung:** Die Zustimmung zu den Nutzungsbedingungen erfolgt durch Aktivieren eines Feldes in der Wallet im Rahmen der Beantragung. Die zugehörigen Dokumente auf der Website des KBAs werden als Link dargestellt.

**Anforderung „Änderungen der Geschäftsbedingungen“ ([BTR71], G5)**: Dem Inhaber der Authentisierungsmittel müssen in geeigneter Weise die Geschäftsbedingungen sowie notwendige Verhaltensregeln zum Umgang mit den Authentisierungsmitteln übermittelt werden. Wenn sich die Geschäftsbedingungen oder notwendigen Verhaltensregeln ändern, müssen alle betroffenen Stellen und insbesondere der Authentisierungsmittelinhaber geeignet über die Änderungen informiert werden.
**Umsetzung:** Der Nutzer wird bei Ausstellen des Credentials über die dann aktuellen Nutzungsbedingungen über einen Link auf eine dafür erstellte Webseite des KBA informiert. Eine Änderung der Bedingungen ist aktuell nicht vorgesehen. Sollte dies doch erfolgen und wesentliche Neuerungen mit sich bringen, kann eine neue Credential Definition erzeugt werden, welche ab diesem Punkt durch die Verifizierer zu prüfen ist. Dies bedingt eine erneute Ausstellung des Credentials durch den Nutzer und somit die erneute Zustimmung zu den geänderten Geschäftsbedingungen..

## 5.2 Authentisierungsmittel und -protokoll

### 5.2.1 Authentisierungsmittel

**Anforderung „ Ausgabe des Tokens " ([BTR71], B1):** *Die Ausgabe eines auf Besitz basierenden Sicherungsmittels muss so erfolgen, dass der berechtigte Inhaber nach Erhalt erkennen kann, ob das Sicherungsmittel unberechtigt benutzt wurde [...]*

**Umsetzung:** Das Verifiable Credential für den digitalen Führerscheinnachweis wird bei der Ausstellung in der Wallet-App des Holders gespeichert, nachdem es vom Issuer erzeugt wurde. Der Issuer ist vertrauenswürdig und löscht seinerseits das Credential nach der Übertragung. Die Übertragung erfolgt vertraulich mit authentifizierten Endpunkten. Eine unberechtigte Nutzung des Credentials bei der Ausgabe wird somit technisch unterbunden.

**Anforderung „Anforderungen an den Token" ([BTR71], B2):** *Anforderungen aus Tabelle 3 in* [BTR07]*: Eigenschaften von Authentisierungsfaktoren*

**Umsetzung:**

***F:*** *Wie wird die Einmaligkeit und Nicht-Kopierbarkeit des Besitzes sichergestellt?*

**A:** Es ist technisch derzeit nicht möglich die Einmaligkeit des Besitzes sicherzustellen**.** Die Nicht-Kopierbarkeit des Besitzes wird über den jeweiligen sicheren Schlüsselspeicher der Plattformen (Android / iOS) sichergestellt. Beim initialen Generieren des Besitzes (hardwaregebundenes asymmetrisches Schlüsselpaar) wird mittels eines Parameters die Extraktion bzw. Kopierbarkeit deaktiviert (siehe dazu [GBK21]).

***F:*** *Wie wird der Inhaber darauf hingewiesen, dass er den Besitz nicht weitergeben darf?*

**A:** Aktuell gibt es keinen Hinweis an den Inhaber.

***F:*** *Wie wird verifiziert, ob der Besitz unter physischer Kontrolle des Inhabers ist?*

**A:** Der Besitz (hardwaregebundenes asymmetrisches Schlüsselpaar) kann nur nach erfolgreicher Eingabe des Wissen-Faktors (Inhaber-PIN) benutzt werden. Da die Inhaber-PIN nur dem Inhaber bekannt ist, wird lokal in der Wallet-App verifiziert, dass der Besitz unter der physischen Kontrolle des Inhabers ist (siehe dazu auch [GBK21]).

***F:*** *Wie wird der Inhaber darauf hingewiesen, dass der Besitz nur zur Authentisierung genutzt werden darf?*

**A:** Der Besitz (hardwaregebundenes asymmetrisches Schlüsselpaar) kann nur innerhalb der Wallet-App

für das Erstellen von Signaturen innerhalb eines Identitätsnachweises genutzt werden. Alternative Nutzungen sind technisch ausgeschlossen.

F: Wie wird ein Verlust des Besitzes erkannt?
A: Durch Gerätebindung kann Verlust des Besitzes durch den Nutzer direkt erkannt werden.

***F:*** *Ist eine Missbrauchserkennung realisiert? Falls ja: wie?*
**A:** Durch einen im softwareumgesetzten Fehlbedienungszähler des PINs (nach fünf falschen Eingaben werden die für die Entschlüsselung des Wallet benötigten Werte aus dem Xamarin-Secure-Storage entfernt und die Wallet kann nicht weiter verwendet werden) kann ein Missbrauchsversuch lokal erkannt werden. Es sind keine weiteren Maßnahmen zur Missbrauchserkennung realisiert.

**F**: Ist eine Sperrung über ein eindeutiges Merkmal des Besitzes möglich? Falls ja: wie?
**A**: Eine Sperrung ist beim initialen Start nicht vorgesehen.

**F**: Ist die Ausstellung eines neuen Besitztokens als Ersatz für ein gesperrtes Authentisierungsmittel möglich?
**A**: Um ein neues Besitztoken zu erhalten, muss eine erneute Registrierung durchlaufen werden.

#### *5.2.1.1 Besitz*

**Anforderung „Schlüsselspeicherung“ ([BTR71], SW1):** *Es gelten die Anforderungen aus Abschnitt 3.7 [„Kryptographie“]. Die privaten kryptographischen Schlüssel dürfen nicht außerhalb des Tokens vorliegen (kein Key-Backup oder Key-Escrow).*
**Umsetzung**: Das hardwaregebundene asymmetrische Schlüsselpaar (Besitzfaktor) wird im sicheren Schlüsselspeicher der jeweiligen Plattform generiert und verlässt diesen nicht. Key-Backup sowie Key-Escrow wird mittels Parameter bei der Generierung des Schlüsselmaterials verboten und kann vom Issuer mittels Key & ID Attestation (auf der Android Platform) überprüft werden (siehe dazu [GBK21]).

**Anforderung „Erzeugung und Löschung der Schlüssel“ ([BTR71], SW2):** *Sofern Schlüssel außerhalb des Tokens erzeugt werden, so muss dies in einer sicheren Umgebung erfolgen und die außerhalb des Tokens vorliegenden privaten Schlüssel [müssen] vor Auslieferung des Tokens gelöscht werden.*

**Umsetzung:** Das hardwaregebundene asymmetrische Schlüsselpaar (Besitzfaktor) wird im sicheren Schlüsselspeicher der jeweiligen Plattform generiert.

Zusätzlich befindet sich innerhalb des SSI-Wallet (SQLite Datenbank) ein Link Secret. Das Link Secret ist ein sogenanntes “hidden attribute” und wird initial von der Wallet generiert und von dem Aussteller blind

signiert (durch Kombination mit einem sogenannten “blinding factor”). Jeder Aussteller signiert das Link Secret blind. Das Link Secret ist also für jedes VC eines Nutzers gleich. Es ist also ein Attribut jedes Credentials, das der Aussteller selbst aber nicht zu sehen bekommt. Bei jedem Proof beweist der Nutzer die Kenntnis des Link Secrets und bei Verwendung mehrerer Credentials die Übereinstimmung des Link Secret. Somit wird die Zugehörigkeit der Credentials zu einem legitimen Nutzer nachgewiesen. Zusätzlich zum Schutz der Schlüssel durch den sicheren Schlüsselspeicher der Plattform wird durch die Gerätebindung [GBK21] der Faktor “Besitz” gestützt, da ein Kopieren des Verifiable Credentials und somit auch der zugehörigen Schlüssel auf ein anderes mobiles Endgerät bestmöglich unterbunden wird, sodass die Nutzung nur auf einem Smartphone möglich ist. Auch die Gerätebindung nutzt den sicheren Schlüsselspeicher der Plattform als Sicherheitsanker. Zusätzlich wird der sichere Zustand des mobilen Endgerätes beim Issuing per remote Attestation überprüft.

#### *5.2.1.2 Wissen*

**Anforderung „Ausgabe des Wissens“ ([BTR71], W1):** *Die Ausgabe von wissensbasierten Sicherungsmitteln muss so erfolgen, dass der Inhaber unberechtigte Kenntnisnahme erkennen kann (Unversehrtheit des „PIN-Briefes“).*

**Umsetzung:** Das wissensbasierte Sicherungsmittel (hier PIN) wird von Nutzer beim initialen Wallet Start festgelegt und wird weder an Dritte übertragen oder von diesem einsehbar gemacht (siehe dazu Kapitel 7.1.2.1 in [SYS21]).

**Anforderung „Anforderungen an das Wissen“ ([BTR71], W2):** *Anforderungen aus Tabelle 3 in [TR-03107-1]: Eigenschaften von Authentisierungsfaktoren*

**Umsetzung:**

***F:*** *Wie wird sichergestellt, dass das Wissen nur dem Inhaber und ggf. der verifizierenden Entität bekannt ist?*

**A:** Das Wissen (hier PIN) ist nur dem Inhaber bekannt.

***F:*** *Wird das Wissen zu irgendeinem Zeitpunkt ungesichert übertragen oder gespeichert?*

**A:** Nein, der PIN wird zu keinem Zeitpunkt übertragen und liegt verschlüsselt und gehasht im sicheren Speicher (Xamarin Secure Storage) der Wallet-App.

***F:*** *Können die Mitarbeiter oder Dienstleister des Verfahrensbetreibers einen Zugriff auf ungesichertes Wissen erhalten?*

**A:** Nein, ein Zugriff durch Dritte ist nicht möglich.

***F:*** *Wie wird der Inhaber darauf hingewiesen, dass das Wissen nicht weitergegeben werden darf?*
**A:** Aktuell sind keine Maßnahmen zu diesen Punkt implementiert.

***F:*** *Wie wird der Inhaber darauf hingewiesen, dass das Wissen nur zur Authentisierung genutzt werden darf?*
**A:** Aktuell sind keine Maßnahmen zu diesem Punkt implementiert.

***F:*** *Wie wird der Inhaber darauf hingewiesen, dass der gleiche Wissens-Token nicht noch für andere Dienste verwendet werden darf?*
**A:** Aktuell sind keine Maßnahmen zu diesem Punkt implementiert.

***F:*** *Wird der Inhaber darauf hingewiesen, dass er den Wissens-Token nicht aufschreiben oder in der Cloud speichern soll? Gibt es besondere Ausnahmen, die erlauben, das Wissen aufzuschreiben?*
**A:** Aktuell sind keine Maßnahmen zu diesem Punkt implementiert.

***F:*** *Ist eine Missbrauchserkennung realisiert? Falls ja, wie?*
**A:** Ja, eine Missbrauchserkennung ist in Form eines Fehlbedienungszählers realisiert, welcher nach 5-maliger Falscheingabe die Nutzung des Besitzfaktors unmöglich macht (siehe dazu **Anforderung “Passwortentropie”**)

***F:*** *Ist Sperren des zugehörigen Accounts (bei entfernter Verifikation durch Server) bzw. Besitzes (bei lokaler Verifikation durch Besitztoken) bei Missbrauchsverdacht möglich?*

**A:** Eine Sperrung ist beim initialen Start des Betriebs nicht vorgesehen.

***F:*** *Ist Setzen eines neuen Passworts / einer neuen PIN als Ersatz für gesperrtes Passwort / PIN möglich?*
**A:** Nein, bei einem gesperrten PIN muss die Identität erneut ausgegeben werden.

**Hinweis**: Analog zur BasisID [SYS21] sind im aktuellen Entwicklungsstand keine Mechanismen zur Prüfung der Sicherheit der in Mobilgeräten verbauten biometrischen Verifikationsmechanismen, sowie deren Metriken (z.B. FAR, Überwindungssicherheit) implementiert. Für die Testphase des digitalen Führerscheinnachweises werden daher die biometrischen Verifikationsmechanismen deaktiviert.

**Anforderung „Passwortentropie" ([BTR71], W4):** *Bei Verwendung eines Fehlbedienungszählers, der maximal drei Versuche, eine PIN zu raten zu-lässt, sollte eine PIN mindestens 4 (Vertrauensniveau normal), 5 (Vertrauensniveau substantiell) bzw. 6 (Vertrauensniveau hoch) dezimale Stellen haben (vgl. [AIS 20/31]).*

**Umsetzung:** Es wird eine 6-stellige PIN eingesetzt. Nach 5-maliger Falscheingabe wird das Authentisierungsmittel durch einen Fehlbedienungszähler gesperrt, sodass das Authentisierungsmittel danach nicht mehr genutzt werden kann. Der Fehlbedienungszähler ist in Software umgesetzt (siehe dazu auch Kapitel 7.1.2.1 und 7.1.2.2 in [SYS21]).

#### *5.2.1.3 Zwei-Faktor Authentisierung*

**Anforderung „Multi-Faktor" ([BTR71], G24):** *Zur Erreichung des Vertrauensniveaus [..] ist grundsätzlich die Nutzung von zwei Faktoren zur Absicherung der Authentisierungsmittel notwendig, die die alleinige Kontrolle des Nutzers über seine Authentisierungsmittel sicherstellen. Dabei müssen die beiden Faktoren unterschiedlichen Kategorien angehören.*

**Umsetzung:** Die betrachtete Lösung bittet eine Zwei-Faktor Authentisierung, bestehend aus dem Faktor "Wissen" in Form einer PIN, sowie aus dem Faktor "Besitz" in Form eines anonymous credentials, sowie einem dort referenzierten, gerätegebundenen asymmetrischen Schlüsselpaars (privater Schlüssel der hardwareDid) [GBK21]).

**Anforderung „Widerstandsfähigkeit des Authentisierungsmittels" ([BTR71], G25):** *Die Authentisierungsmittel müssen so gestaltet werden, dass der berechtigte Inhaber sie gegen Missbrauch durch Dritte mit Angriffspotential gemäß Abschnitt 3.1 [BTR07] schützen kann.*

**Umsetzung:** Zur Widerstandsfähigkeit des Authentisierungsmittels liegt keine dedizierte Analyse zur Resistenz gegen ein bestimmtes Angriffspotential vor. Wichtige Aspekte zur Widerstandsfähigkeit wurden jedoch bereits in Kapitel 7 in [SYS21], sowie in [GBK21] aufgeführt und analysiert.

**Anforderung „Verknüpfung Sicherungsfaktoren" ([BTR71], MF1):** *Bei Nutzung von Wissen in Kombination mit Besitz müssen beide Sicherungsfaktoren miteinander verknüpft sein, zum Beispiel die Benutzung einer PIN zur Freischaltung einer Chipkarte.*

**Umsetzung:** Die Nutzung des Besitzfaktor (privater Schlüssel der hardwareDid) wird von der Wallet verwaltet und nur nach erfolgreicher Validierung der Inhaber-PIN (Wissen-Faktor) für einen Signaturvorgang freigegeben (siehe dazu [GBK21]).

Die Bindung beider Sicherungsfaktoren wird hierbei von der Integrität der Wallet-App gewährleistet. Die Integrität der Wallet-App wiederum wird beim Aufbringen der Identität von den jeweiligen Attestierungsmechanismen der Plattformen (Android SafetyNet Attestation, Apple iOS AppAttest) und zur Laufzeit von den Sicherheitsmechanismen der Plattformen gewährleistet (siehe dazu [GBK21], sowie Kapitel 7.1.1. in [SYS21]).

**Anforderung „Fehlschlagen eines Faktors“ ([BTR71], MF2):** *[...] darf ein Angreifer das Fehlschlagen eines Authentisierungsversuches nicht einem einzelnen Authentisierungsfaktor zuordnen können.*
**Umsetzung**: Der Nutzer erhält einen Hinweis, wenn der PIN falsch eingegeben wurde. Dadurch könnte ein Angreifer das Fehlschlagen des Authentisierungsversuchs einem einzelnen Authentisierungsfaktor zuordnen.

**Anforderung „Resistenz beider Faktoren“ ([BTR71], MF3):** *[...] [...] dürfen nicht beide Faktoren gemeinsam durch einen einzelnen Angriff auf die Nutzerumgebung angreifbar sein.*
**Umsetzung:** Ein Angreifer kann nicht beide Faktoren gemeinsam durch einen einzelnen Angriff auf die Nutzerumgebung angreifen, da der Besitzfaktor (privater Schlüssel der hardwareDid) getrennt vom Wissen-Faktor (Inhaber-PIN) im sicheren Schlüsselspeicher der jeweiligen Plattform gespeichert ist. Wird die Nutzerumgebung (Wallet App) angegriffen, könnte unter Umständen der Wissen-Faktor erlangt werden, der Zugriff auf den Besitzfaktor wird dann aber vom jeweiligen Betriebssystem unterbunden. Wird die Systemplattform angegriffen (Android Framework, Linux / Darwin Kernel, …) kann unter Umständen der Besitzfaktor missbraucht werden. Eine Extraktion des Besitzfaktors ist jedoch nicht möglich [GBK21].

**Anforderung „Ausgabe über getrennte Übermittlungswege“ ([BTR71], MF4):** Die Ausgabe für die Vertrauensniveaus substantiell/hoch muss so erfolgen, dass die beiden Sicherungsfaktoren [...] auf verschiedenen Übermittlungswegen ausgegeben werden. Diese Anforderung kann auch dadurch erfüllt werden, in dem die beiden Faktoren zeitlich getrennt auf gleichem Wege übermittelt werden, sofern sichergestellt ist, dass der erste Faktor den Inhaber erreicht hat, bevor der zweite übermittelt wird.
**Umsetzung:** Der Faktor “Wissen” ist die Inhaber-PIN. Der Faktor “Besitz” ist das anonymous credential. Die Inhaber-PIN wird von dem Nutzer selbst festgelegt. Der Besitz des anonymous credentials erfolgt über den Ausgabeprozess an dessen Ende das Credential in der Wallet-App gespeichert wird. Die Inhaber-PIN wird dabei nicht übermittelt.

### 5.2.2 Authentisierungsprotokoll

**Anforderung „Sicherheit des Authentisierungsprotokolls“ ([BTR71], G6):** Das Authentisierungsprotokoll muss gegen Angreifer mit Angriffspotential gemäß Abschnitt 3.1 [BTR07] sicher sein.
**Umsetzung:**

Das Authentisierungsprotokolls verwendet kryptographische Primitive, um Schutz gegen verschiedene Angriffe zu bieten:

**Schutz gegen das Raten von Authentisierungsdaten bei der Verification:**

ZKP: Das Raten von Authentisierungsdaten bedeutet bei der Verification, dass ein gültiger Zero-Knowledge Proof erraten wird. Das Angriffspotential kann aufgrund der Größe des Schlüsselraumes (2048 Bit) als gering angesehen werden. Camenisch-Lysanskaya-Signaturen sind beweisbar sicher unter der starken RSA Annahme.

hardwareDidProof: Es kann nicht erraten werden, da wir etablierte Protokolle und Standards verwenden (SHA256withECDSA).

**Schutz gegen das Duplizieren von Authentisierungsdaten bei der Verification**:

ZKP: Zum Schutz gegen Reply-Angriffe wird ein Nonce-Value (80 Bit) verwendet, der von der Verifier App als Teil des Zero-Knowledge-Proofs und vom Verifier überprüft wird.

hardwareDidProof: Zum Schutz gegen Reply-Angriffe wird ein Nonce-Value (80 Bit) verwendet, der von der Verifier-App als Teil der Signatur des Proofers geprüft wird.

Durch die Gerätebindung wird einem Duplizieren des Credentials entgegengewirkt.

**Schutz gegen Abfangen/Verfälschen von Informationen bei der Verification:**

ZKP: Durch die Verwendung von Signaturen sind die übertragenen Daten Integritäts-geschützt. Angreifer würden durch ein Verändern der Daten die Signatur invalidieren. Zur Absicherung vor einem Abfangen der Daten wird ein sicherer DIDcomm-Kanal zwischen den Endpunkten (Proofer und Verifier) verwendet. Diese Kommunikation ist über die im Connectionless Proof Request übermittelten Information des Verifiers zusätzlich Ende-zu-Ende verschlüsselt.

hardwareDidProof: Durch die Verwendung von Signaturen sind die übertragenen Daten Integritäts-geschützt. Angreifer würden durch ein Verändern der Daten die Signatur invalidieren. Zur Absicherung vor einem Abfangen der Daten wird ein sicherer DIDcomm-Kanal zwischen den Endpunkten (Proofer und Verifier) verwendet.

**Anforderung „Forward Secrecy“ ([BTR71], G7):** Sofern für einen Mechanismus die Vertraulichkeit ein Sicherheitsziel ist, so sollen kryptographische Verfahren eingesetzt werden, die Vorwärtssicherheit (Forward Secrecy) bieten.
**Umsetzung:** Das für den Identitätsnachweis verwendete Kommunikationsprotokoll (DIDComm) bietet eine “weak perfect forward secrecy” [DIC21]. Aufgrund des in ECDH-1PU verwendeten dreifachen

Schlüsselableitungsalgorithmus verfügen alle über DIDComm gesendeten Nachrichten über weak perfect forward secrecy. Dies wird durch die Verschlüsselung des Inhaltsverschlüsselungsschlüssels mit der Ausgabe des Hashes von Ze (ECDH des ephemeren Schlüssels und des statischen Empfängerschlüssels) und Zs (ECDH des statischen Absenderschlüssels und des statischen Empfängerschlüssels) erreicht. Da Ze das geänderte abgeleitete Geheimnis in jeder Nachricht und Zs die widerlegbare Authentizität jeder Nachricht mitbringt, trägt das resultierende Z (Hash von Ze und Zs) die Eigenschaften des schwachen perfekten Vorwärtsgeheimnisses und der widerlegbaren Authentizität für jede Nachricht.

Bei der Verifikation wird auf Transportschicht-Ebene HTTPS (mindestens TLSv1.2) verwendet. Sofern in einem Protokollschritt Vertraulichkeit ein Sicherheitsziel ist (beispielsweise bei der Übertragung der personenbezogenen Daten von Cloud Agent an den Controller) werden Cipher-Suiten nach BSI-TR-02102 zur "Verwendung von TLS" mit perfect forward secrecy verwendet.

**Anforderung „Dynamisches Authentisierungsprotokoll" ([BTR71], G8):** [..] das Authentisierungsprotokoll [muss] dynamisch sein, d.h. das Verfahren muss dazu geeignet sein nachzuweisen, dass sich die Authentisierungsmittel im Augenblick der Authentisierung unter Kontrolle des Inhabers befinden. Dieser Nachweis muss für jede Authentisierung neu erzeugt werden.
**Umsetzung:**

hardwareDidProof: Der Verifier sendet im Rahmen des DIDcomm Proof Request eine Nonce die der Proofer mit seinem privaten Schlüssel signiert und dann im Rahmen des Proof als signierte Nonce an den Verifier sendet.

**Anforderung „Eindeutige Inhaberidentifizierung"([BTR71], G18):** Authentisierungsverfahren müssen den Inhaber der Authentisierungsmittel gegenüber der Gegenstelle eindeutig identifizieren (üblicherweise durch die Registrierung einer eindeutigen Kennung des Authentisierungsmittels bei der Gegenstelle).
**Umsetzung:** Die Inhaberidentifizierung erfolgt über die eindeutige Hardware DID (Public Key des Smartphones) und über das Anwendungsfall-spezifische Set von Attributen im Rahmen des ZKPs. Zusätzlich wird bei jedem ZKP durch den Beweis des Besitzes des Link Secrets eine Verknüpfung von mehreren Identity Proofs ermöglicht.

**Anforderung „Geheimhaltung der Nutzerkennung" ([BTR71], G19):** Das Authentisierungsverfahren muss einerseits diese Kennung der Gegenstelle gegenüber entsprechend der Anforderungen des

jeweiligen Vertrauensniveaus nachweisen, Dritten darf die Kennung aus Datenschutzgründen aber nicht bekannt werden.

**Umsetzung:** Die Hardware DID bleibt für jeden Nutzer konstant. Die Geheimhaltung der Hardware DID wird sichergestellt, indem der Public Key unmittelbar nach Durchführung des Proofs vom Verifier gelöscht wird.

## 5.3 Rückruf/Sperrung

Der Rückruf bzw. die Sperrung wurden zum Zeitpunkt der Evaluation noch nicht umgesetzt. Durch einen relativ kurzen Gültigkeitszeitpunkt des betrachteten Führerscheinnachweises (maximal 24 Stunden) wurden jedoch Maßnahmen getroffen, um die fehlende Umsetzung des Rückrufs/ der Sperrung zu kompensieren. Es ist möglich das Credential in der Wallet zu löschen. Zusätzlich wird bei Neuausstellung eines Credentials das vorherige gelöscht

### 5.3.1 Sperrung

**Anforderung „Sperrung“ ([BTR71], G9):** Im Falle der Kompromittierung von Authentisierungsmitteln muss es dem Inhaber möglich sein, die Authentisierungsmittel zu sperren.

**Umsetzung: Aktuell Out of Scope, in einer weiteren Ausbaustufe geplant**

**Anforderung „Übermittlung der Sperrmeldung“ ([BTR71], G10):** Die Möglichkeit zur Übermittlung der Sperrmeldung muss über öffentliche Kommunikationswege verfügbar sein und den Inhabern des Authentisierungsmittels in geeigneter Weise bekannt gemacht werden.

**Umsetzung: Aktuell Out of Scope, in einer weiteren Ausbaustufe geplant**

**Anforderung „Attributsänderung“ ([BTR71], G11):** Ein Rückruf von Authentisierungsmitteln ist auch dann notwendig, wenn die authentisierten Identitätsattribute nicht mehr gültig sind (z.B. Namensänderung) oder der Inhaber nicht mehr zum Besitz berechtigt ist.
Sowohl für externe als auch interne Attribute muss festgelegt sein, inwiefern deren Gültigkeit erlischt, wenn sich die zugrunde liegende nachgewiesene Identität der Entität ändert.

**Umsetzung: Aktuell Out of Scope, in einer weiteren Ausbaustufe geplant**

### 5.3.2 Reaktivierung

**Anforderung „<Titel>“ (Quelle): …**

**Umsetzung: …**

## 5.4 Vertrauenswürdigkeit von Stellen

**Anforderung „Dienstbindung an den Sitzungskontext“ ([BTR71], G20):** Als Bestandteil des Aufbaus eines Sitzungskontextes muss sichergestellt werden, dass die Identifizierungen des Dienstes an diese Sitzung gebunden werden. Dies umfasst, dass die Identität eindeutig einer bestimmten Session und nicht lediglich einem bestimmten Kommunikationsendpunkt zugeordnet werden muss und auch nur dort gültig sein darf. Für die Vertrauensniveaus substantiell/hoch muss diese Bindung über geeignete technische/kryptographische Mechanismen erfolgen, etwa kryptographisch sichere Session-Identifier/-Cookies.

**Umsetzung:** Nach Aufruf des im QR-Code enthaltenen Endpunktes wird auf Seiten des Verifiers die Erzeugung eines Proof Request angestoßen. Bei der Erzeugung dieses Proof Request (Request Presentation) wird dieser um eine Thread ID ergänzt. Diese Thread ID wird in der Antwort des Proof Request (Proof Presentation) hinzugefügt.

**Anforderung „Nutzerumgebung“ ([BTR71], G17):** Es muss sichergestellt werden, dass die Mechanismen mit entsprechend den Empfehlungen [des BSI für Bürger zur Absicherung des lokalen Endgeräts] konfigurierten Rechnern verwendbar sind, Mechanismen dürfen keine Anforderungen stellen, die den Empfehlungen des BSI [für Bürger zur Absicherung des lokalen Rechners] widersprechen.

**Umsetzung:**

Das Verfahren stellt keine Anforderungen, die den Empfehlungen des BSI für Bürger zur Absicherung des lokalen Rechners widersprechen

**Anforderung „Stellen“ ([BTR71], S1):** *Bei den meisten Mechanismen übernehmen – neben dem Inhaber der Authentisierungsmittel und der vertrauenden Entität – weitere Stellen für die Sicherheit des Mechanismus relevante Aufgaben, zum Beispiel Enrolment, Identitätsprüfung und Ausgabe der Authentisierungsmittel (Abschnitt 3.2), Sicherung von Kommunikationsbeziehungen (Abschnitt 3.6) oder Speicherung von Daten. Auch Identitätsprovider sind Stellen in diesem Sinne. Sämtliche Stellen müssen*

- *Behörden oder juristische Personen sein und rechtlich befugt sein, die jeweilige Aufgabe wahrzunehmen;*
- *für ihre jeweiligen wahrgenommen Aufgaben ein Regelwerk aufstellen und dieses einhalten;*
- *organisatorisch und technisch in der Lage sein, die Aufgaben auf Basis des Regelwerks wahrzunehmen;*

- *genügend Ressourcen für die Erfüllung der Aufgaben und ggf. die Übernahme der sich aus den Aufgabenergebende Haftung haben;*

*und ein Informationssicherheitsmanagementsystem auf Basis etablierter Standards (z.B. IT-Grundschutz [BSI100-2]oder [ISO27001]) nutzen.*

**Umsetzung: Die Bundesdruckerei ist ISO 27001 zertifiziert.**

- Enrolment (siehe unten)
    - Identitätsprüfung - BDR im Auftrag des KBA
        - Ausgabe der Authentisierungsmittel - BDR
- Sicherung von Kommunikationsbeziehungen – alle Issuing und Verification:
  Issuing Service, KBA Backend, Wallet, Mediation Agent, alle Komponenten auf Verifizierer Seite, Nodes
- Speicherung von Daten – alle obigen
- Andere relevanten Stellen

1) Issuer (BDR)

| | |
|---|---|
| Name der Stelle | Bundesdruckerei GmbH |
| Die Stelle ist eine Behörde oder juristische Person. | Ja |
| Die Stelle ist rechtlich befugt, ihre Aufgabe wahrzunehmen. | Ja |
| Die Stelle stellt für ihre Aufgabe ein Regelwerk auf und hält dieses ein. Bitte benennen Sie das Dokument und die aktuell gültige Versionsnummer. Besteht seitens der Stelle die Möglichkeit, dem Prüfer Einsicht in das Dokument zu gewähren (ggf. in gekürzter Form)? | In Erstellung |
| Die Stelle ist organisatorisch und technisch in der Lage, ihre Aufgaben auf Basis des Regelwerks wahrzunehmen. | Ja |
| Die Stelle hat genügend Ressourcen für die Erfüllung ihrer Aufgaben und ggf. für die Übernahme der sich aus den Aufgaben ergebender Haftung. | Ja |
| Die Stelle ist gemäß IT-Grundschutz [BSI100-2] oder [ISO27001] zertifiziert. Die Zertifizierung erstreckt sich über sämtliche für die Tätigkeit in Ihrem Authentisierungsverfahren relevanten Systeme und Komponenten<br>Falls die Stelle nicht selbst eine Behörde ist: ist sie behördlich für die wahrgenommene Aufgabe nach geltenden Gesetzen / Prozessen anerkannt? Durch welche Behörde? | ISO 27001 |

2) Wallet Herausgeber (Digital Enabling GmbH)

| | |
|---|---|
| Name der Stelle | Digital Enabling GmbH |
| Die Stelle ist eine Behörde oder juristische Person. | Ja |
| Die Stelle ist rechtlich befugt, ihre Aufgabe wahrzunehmen. | Ja |

| | |
|---|---|
| Die Stelle stellt für ihre Aufgabe ein Regelwerk auf und hält dieses ein. Bitte benennen Sie das Dokument und die aktuell gültige Versionsnummer. Besteht seitens der Stelle die Möglichkeit, dem Prüfer Einsicht in das Dokument zu gewähren (ggf. in gekürzter Form)? | In Erstellung |
| Die Stelle ist organisatorisch und technisch in der Lage, ihre Aufgaben auf Basis des Regelwerks wahrzunehmen. | Ja |
| Die Stelle hat genügend Ressourcen für die Erfüllung ihrer Aufgaben und ggf. für die Übernahme der sich aus den Aufgaben ergebender Haftung. | Ja |
| Die Stelle ist gemäß IT-Grundschutz [BSI100-2] oder [ISO27001] zertifiziert. Die Zertifizierung erstreckt sich über sämtliche für die Tätigkeit in Ihrem Authentisierungsverfahren relevanten Systeme und Komponenten<br>Falls die Stelle nicht selbst eine Behörde ist: ist sie behördlich für die wahrgenommene Aufgabe nach geltenden Gesetzen / Prozessen anerkannt? Durch welche Behörde? | In Planung (Start 2021) |

3) Verifier. Details zu spezifischen Verifier Organisationen können bei Bedarf zu einem späteren Zeitpunkt ergänzt werden, wenn diese Organisationen in konkreter Vorbereitung zum Deployment sind. Anforderungen des BSI werden an diese Organisationen kommuniziert.

4) Knotenbetreiber

| | |
|---|---|
| Name der Stelle | Bundesdruckerei, esatus, IBM, BWI, Deutsche Bahn |
| Die Stelle ist eine Behörde oder juristische Person. | Alle: Juristische Person. |
| Die Stelle ist rechtlich befugt, ihre Aufgabe wahrzunehmen. | Alle: Ja. |
| Die Stelle stellt für ihre Aufgabe ein Regelwerk auf und hält dieses ein. Bitte benennen Sie das Dokument und die aktuell gültige Versionsnummer. Besteht seitens der Stelle die Möglichkeit, dem Prüfer Einsicht in das Dokument zu gewähren (ggf. in gekürzter Form)? | Ein gemeinsames Regelwerk, welches gleichermaßen für alle Knotenbetreiber gilt, ist in Abstimmung und kann zur Einsicht zur Verfügung gestellt werden. |
| Die Stelle ist organisatorisch und technisch in der Lage, ihre Aufgaben auf Basis des Regelwerks wahrzunehmen. | Alle: Ja. |
| Die Stelle hat genügend Ressourcen für die Erfüllung ihrer Aufgaben und ggf. für die Übernahme der sich aus den Aufgaben ergebender Haftung. | Alle: Ja. |
| Die Stelle ist gemäß IT-Grundschutz [BSI100-2] oder [ISO27001] zertifiziert. Die Zertifizierung erstreckt sich über sämtliche für die Tätigkeit in Ihrem Authentisierungsverfahren relevanten Systeme und Komponenten<br>Falls die Stelle nicht selbst eine Behörde ist: ist sie behördlich für die wahrgenommene Aufgabe nach geltenden Gesetzen / Prozessen anerkannt? Durch welche Behörde? | Bundesdruckerei: ISO 27001<br>IBM: ISO 27001<br>esatus: In Planung (Start 2021)<br>BWI: in Klärung<br>Deutsche Bahn: in Klärung |

5) Betreiber Mediator-Service

| | |
|---|---|
| Name der Stelle | esatus AG |
| Die Stelle ist eine Behörde oder juristische Person. | Ja |
| Die Stelle ist rechtlich befugt, ihre Aufgabe wahrzunehmen. | Ja |

| | |
|---|---|
| Die Stelle stellt für ihre Aufgabe ein Regelwerk auf und hält dieses ein. Bitte benennen Sie das Dokument und die aktuell gültige Versionsnummer. Besteht seitens der Stelle die Möglichkeit, dem Prüfer Einsicht in das Dokument zu gewähren (ggf. in gekürzter Form)? | In Erstellung |
| Die Stelle ist organisatorisch und technisch in der Lage, ihre Aufgaben auf Basis des Regelwerks wahrzunehmen. | ja |
| Die Stelle hat genügend Ressourcen für die Erfüllung ihrer Aufgaben und ggf. für die Übernahme der sich aus den Aufgaben ergebender Haftung. | Ja |
| Die Stelle ist gemäß IT-Grundschutz [BSI100-2] oder [ISO27001] zertifiziert. Die Zertifizierung erstreckt sich über sämtliche für die Tätigkeit in Ihrem Authentisierungsverfahren relevanten Systeme und Komponenten<br>Falls die Stelle nicht selbst eine Behörde ist: ist sie behördlich für die wahrgenommene Aufgabe nach geltenden Gesetzen / Prozessen anerkannt? Durch welche Behörde? | In Planung (Start 2021) |

## 5.5 Absicherung von Kommunikationsbeziehungen

**Anforderung „Absicherung von Kommunikationsbeziehungen“ ([BTR71], G12):**

**Umsetzung:**

| # | Source | Target | Richtung | Protokolle | Beschreibung |
|---|---|---|---|---|---|
| 1 | Issuer / Verifier | Aries Agent | Bi-direktional | HTTPS, mind. TLS1.2 | Der Issuer / Verifier kommuniziert mit dem Aries Agent via einer REST Schnittstelle. Asynchrone callbacks werden ebenfalls an den Issuer / Verifier via einer REST Schnittstelle aufgerufen. Die jeweiligen API Aufrufe werden neben HTTPS zusätzlich über einen API-Key im HTTP Header abgesichert. |
| 2 | ID Wallet | Aries Agent / Mediator -Service | Uni-direktional | DIDComm über HTTP/HTTPS | Die ID Wallet kommuniziert mit Agenten mittels DIDComm über HTTP/HTTPS. Jegliche DIDComm Kommunikation ist über PAYLOAD_ENCRYPTION verschlüsselt.<br>• Schlüsseleinigung: X25519 (ECDH)<br>• Verschlüsselung: XChaCha20<br>• Datenauthentisierung: Poly1305<br>Der Mediator-Service führt für die ID Wallet eine Inbox, in welcher sie eingehende Nachrichten über eine, bei der initialen Einrichtung der ID Wallet etablierten, DIDComm Verbindung abrufen kann. |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | Diese Nachrichten sind so verschlüsselt, dass der Mediator-Service diese Nachrichten nicht lesen kann. |
| 3 | ID Wallet | Verifier | Uni-direktional | HTTP/HTTPS | Die ID Wallet ruft über einen HTTP-GET Aufruf bei dem Verifier einen dort erzeugten Connectionless Proof Request ab. Dieser wird im Location Header des HTTP-Response übermittelt. |
| 4 | Originator | Verifier | Bi-direktional | HTTPS, TLS1.2 | Eine einfache REST Schnittstelle welche zusätzlich mit einem API-Key im HTTP Header abgesichert wird. Der Callback wird ebenfalls via HTTPS aufgerufen. Jedoch ist dieser Callback in der Verantwortung des Originators daher kann nicht garantiert werden, dass der callback via HTTPS zur Verfügung gestellt wird. |
| 5 | Issuer | ZFER | Uni-direktional | HTTPS, mutual TLS, TLS1.2 | Eine einfache REST Schnittstelle welche mittels TLS1.2 abgesichert wird. Zusätzlich wird Inhaltsdatenverschlüsselung und Inhaltsdatensignierung verwendet. |
| 6 | Aries AgentID Wallet | Indy Node | Uni-direktional | CurveZMQ | Kommunikation zwischen Netzwerkknoten und SSI-Agenten (Issuer, Holder oder Verifier) zum Abruf von Leder Objekten.<br>Schlüsseleinigung: X25519 (ECDH)<br>Verschlüsselung: XSalsa20<br>Datenauthentisierung Poly1305 |
| 7 | Indy Node | Indy Node | Bi-direktional | ZeroMQ, STP, RBFT | Das Indy Netzwerk baut sich mittels dem "Spannig Tree Protocol" (STP) auf. Es verwendet ZeroMQ als sicheres Transport Protokoll.* TCP-based<br>* CurveCP, libsodium<br>* Authenticated encryption, no digital signatures<br>** Authentication: Poly1305 MAC<br>** Symmetric key crypto: XSalsa20<br>** Public Key crypto: Curve25519<br><br>BLS multi-signature wird verwendet um merkle roots zu signieren.<br>Konsensus wird innerhalb des Netzwerks mittels einem Redundanten Byzantine Fault Tolerant (RBFT – Plenum Protokoll) Algorithmus gebildet.<br><br>* Schreibzugriffe:<br>* Müssen signiert sein (Ed25519) |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | * Signaturen werden gegen Public Keys welche auf dem “Ledger” gespeichert sind geprüft. (DID txn)<br>* Jeder Transaktions Author muss eine DID Transaktion auf dem Ledger haben.<br><br>* Lesezugriff:<br>* Keine Authentifizierung ist für das Lesen von der Blockchain notwendig. |
| 8 | Aries Agent | Mediator -Service | Uni-direktional | DIDComm über HTTP/HTTPS | Nachrichten eines Agenten an die ID Wallet müssen über einen Mediator-Service zugestellt werden. Diese Nachrichten werden mittels DIDComm über HTTP/HTTPS an den Mediator-Service übermittelt.<br>Jegliche DIDComm Kommunikation ist über PAYLOAD_ENCRYPTION verschlüsselt.<br><br>• Schlüsseleinigung: X25519 (ECDH)<br>• Verschlüsselung: XChaCha20<br>• Datenauthentisierung: Poly1305 |

Die initialen Vertrauensbeziehungen wurden mittels Austausch öffentlicher Schlüssel mit vertrauenswürdigem Zertifizierungspfad hergestellt.

## 5.6 Kryptographie

**Anforderung „Kryptographie“ ([BTR71], G13):** *Für verschiedene Mechanismen werden konkrete kryptographische Anforderungen in den verschiedenen Teilen der [TR-03116] festgelegt, die jeweils in den Beschreibungen der Mechanismen referenziert werden. Sofern die [TR-03116] für einen Mechanismus keine Vorgaben enthält, so sind die Anforderungen aus [TR-02102] einzuhalten.*

**Umsetzung:** Die eingesetzten kryptographischen Mechanismen und Algorithmen werden im Systemkonzept des Hotel Check-In Pilotvorhabens [SYS21] in Kapitel 8 ausführlich erläutert und decken sich mit den hier eingesetzten Mechanismen und Algorithmen. Die im Systemkonzept vorgenommene Evaluation zeigt, dass einige der verwendeten Algorithmen und Primitiven nicht im Rahmen der [TR-02102] für den Gebrauch freigegeben wurden. Für diese Verfahren werden jedoch in Kapitel 8.4 in [SYS21] weiterführende Sicherheitsaussagen getroffen.

### 5.6.1 Schlüsselspeicher

**Anforderung „Speicherung privater Schlüssel“ ([BTR71], G14):** Private kryptographische Schlüssel aller Entitäten eines Authentisierungssystems (einschließlich des Inhabers von Authentisierungsmitteln) müssen sicher, das heißt vertraulich, gespeichert werden. Dies setzt voraus, dass der private Schlüssel gegen Kopieren geschützt ist und die Verwendung des Schlüssels durch Unberechtigte verhindert wird.
**Umsetzung:**

**F:** Wie werden die privaten kryptographischen Schlüssel aller Entitäten gespeichert? Benennen Sie hierbei insbesondere auch die relevanten Zertifizierungen der verwendeten Komponenten.

Folgend wird bei der ID Wallet, sofern erforderlich, zwischen SSI-Wallet (SQLite Datenbank), Schlüssel für Gerätebindung und weiteren kryptographischen Schlüsseln differenziert.

**ID Wallet:**

SSI-Wallet:

Das Schlüsselmaterial für Connections, Credentials, Proof History, Link Secret und Basis-ID Sperrkennwort werden in einer verschlüsselten SQLite Datenbank auf App-internem Speicher abgelegt.

- Speicherort: SQLite Datenbank auf App-internem Speicher
- Verschlüsselt und authentisiert mit: key_enc_data
    - Base58(SHA256(Pin_validation_deriv | pre_key)
- Algorithmus: CHACHA20-POLY1305
- Erstellt von: Wallet App (Indy SDK)
- Erstellt wie: Link Secret: Indy SDK Funktion ProverCreateMasterSecret, Rest: Indy SDK Funktion AddRecord

Erstellt wann: Link Secret: bei Erzeugung des Wallet, Rest: bei Erhalt der jeweiligen Nachrichten / DatenSchlüssel für Gerätebindung:

Das Schlüsselmaterial welches für die Gerätebindung verwendet wird liegt unter Android entweder im TEE oder falls vorhanden in der StrongBox unter iOS über die Keychain in der Secure Enclave.

Weitere kryptographische Schlüssel:

Weitere private kryptographische Schlüssel werden nur verschlüsselt und authentisiert abgelegt. Dazu werden die jeweiligen plattformabhängig zur Verfügung stehenden Möglichkeiten genutzt. Diese sind

bei der Entwicklung durch Xamarin SecureStorage abstrahiert. Die privaten Schlüssel werden von der Wallt App über die Android KeyStore API oder die Apple iOS Keychain beim initialen Starten der Wallet App erstellt. Praktisch kommen folgende Technologien zum Einsatz:

Key Schema:

**Android**: AES-256 GCM

Geräte mit TEE/SE: Hardware backed key

Geräte ohne TEE/SE: Software backed key (wenige Android 6 Devices, App ist auf Android 7+ zum Start des Betriebs eingeschränkt)

**iOS**: AES-256 GCM (mittel ECC Schlüssel aus Secure Enclave verschlüsselt)

Speicherort: nativer, gerätespezifischer Schlüsselspeicher

**Issuer:**

Beim Issuer für den digitalen Führerscheinnachweis werden die gleichen kryptografischen Schlüssel wie bei der Basis-ID verwendet.

Für die Abfrage beim KBA-Register werden zwei weitere Schlüsselpaare verwendet. Für die Inhaltsdatenverschlüsselung wird RSA 4096 Standardprofil (Basic-Device-ID) genutzt und für die Transportverschlüsselung RSA 2048 Standardprofil (Basic-Device-ID). Die Speicherung der öffentlichen und privaten Schlüssel erfolgt im Sealed Secrets (https://github.com/bitnami-labs/sealed-secrets) und mit den Kubeseal Mechanismus wird sichergestellt, dass nur auf der Zielplattform die Entschlüsselung erfolgt.

**Verifier:**

Beim Verifier für den digitalen Führerscheinnachweis werden die gleichen kryptografischen Schlüssel wie bei dem Hotel Use Case verwendet. Das Schlüsselmaterial wird dem Aries Agent als Secret im Open Shift Cluster zur Verfügung gestellt.

**F:** Wie wird sichergestellt, dass Auslesen, Kopieren oder unberechtigtes Nutzen von privaten Schlüsseln nicht möglich ist?

ID Wallet:

Die Daten liegen im App-internen Speicher, andere Apps haben somit keinen Zugriff auf diese.

Der Schlüssel für die SQLite Datenbank wird durch eine Kombination aus Wissen (PIN) und Informationen aus dem Xamarin SecureStorage (pre_key und pin_salt) zur Laufzeit abgeleitet.

Schlüssel für Gerätebindung:

Die Schlüssel der Gerätebindung sind durch die Nutzung von unter Android TEE und StrongBox an die App gebunden, ein Backup ist hier für die gesamte App deaktiviert unter iOS liegen die Daten über die Keychain verwaltet in der Secure Enclave, diese Schlüssel sind an das Gerät gebunden und explizit aus dem systemweiten Backup ausgeschlossen.

Weitere kryptographische Schlüssel:

Weitere private kryptographische Schlüssel werden nur verschlüsselt und authentisiert abgelegt. Dazu werden die jeweiligen plattformabhängig zur Verfügung stehenden Möglichkeiten genutzt. Diese sind bei der Entwicklung durch Xamarin SecureStorage abstrahiert. Ein Zugriff außerhalb des App Kontextes ist nicht möglich.

Issuer:

Die Speicherung der öffentlichen und privaten Schlüssel erfolgt im Sealed Secrets (https://github.com/bitnami-labs/sealed-secrets) und mit den Kubeseal Mechanismus wird sichergestellt, dass nur auf der Zielplattform die Entschlüsselung erfolgt.

Verifier:

Zugriff auf die Produktionsumgebung ist lediglich einer kleineren Gruppe an DevOps und SRE Ressourcen erlaubt.

**F:** Werden einzelne, zur Aufbewahrung privater Schlüssel genutzte Komponenten in geschützten Umgebungen (entsprechend [ISO27001]) betrieben? Falls ja, geben Sie bitte an, welche Komponenten dies sind und in welcher Umgebung sich diese jeweils befinden.

ID Wallet:

Das Schlüsselmaterial der ID Wallet liegt auf den mobilen Endgeräten, hier ist keine Prüfung auf geschützte Umgebungen vorhanden.

Für die gerätebindungsspezifischen Daten findet unter Android eine Prüfung auf die Verwendung der TEE, bzw. der StrongBox, über eine Überprüfung der Zertifikatskette statt. Unter iOS wird zur Prüfung ein DeviceCheck durchgeführt.

Issuer:

Die Bundesdruckerei GmbH ist ISO 27001 zertifiziert.

Verifier:

Die durch IBM betriebene Umgebung ist ISO 27001 zertifiziert.

**Anforderung „Speicherung öffentlicher Schlüssel“ ([BTR71], G15):** [...] müssen öffentliche Schlüssel, die für die Authentifizierung genutzt werden, sicher, also gegen Manipulation geschützt, gespeichert werden.

**Umsetzung:**

ID Wallet:

Die öffentlichen Schlüssel für Connections werden beim Verbindungsaufbau ausgetauscht und sind jeweils nur für diese spezielle Verbindung (DIDcomm) gülitg.

Diese liegen in der verschlüsselten SQLite Datenbank im App-internen Speicher.

Issuer:

Die Speicherung der öffentlichen und privaten Schlüssel erfolgt im Sealed Secrets (https://github.com/bitnami-labs/sealed-secrets) und mit den Kubeseal Mechanismus wird sichergestellt, dass nur auf der Zielplattform die Entschlüsselung erfolgt.

Verifier:

Speichert keine öffentlichen Schlüssel, sondern ladet diese von einem Indy Node.

Nodes (Verifiable Data Registry): Das Indy Netzwerk speichert die öffentlichen Schlüssel auf einer “Public Permissioned Blockchain”. Mittels dem Konsensus Mechanismus (RBFT) und kryptografisch verlinkten Blöcken wird sichergestellt, dass die öffentlichen Schlüssel nicht verändert werden können.

### 5.6.2 Agilität

**Anforderung „Agilität“ ([BTR71], G16):** Die kryptographischen Verfahren müssen so gestaltet werden, dass sie neuen kryptographischen Erkenntnissen angepasst werden können.

**Umsetzung:** Zum Zeitpunkt der Evaluation lagen noch keine Kenntnisse bzgl. der kryptoagilen Gestaltung der kryptographischen Verfahren im System vor.

## 5.7 Identifizierung einer Person

**Anforderung „Nutzerbindung an den Sitzungskontext“ ([BTR71], G21):** Die übertragene Identität muss an den Sitzungskontext gebunden werden. Dies bedeutet unter anderem, dass die Identität einer Person eindeutig einer bestimmten Session und nicht lediglich einem bestimmten Kommunikationsendpunkt zugeordnet werden muss und auch nur dort gültig sein darf. Für die Vertrauensniveaus substantiell/hoch muss diese Bindung über geeignete technische/kryptographische Mechanismen erfolgen.

**Umsetzung:** Nach Aufruf des im QR-Code enthaltenen Endpunktes wird auf Seiten des Verifiers die Erzeugung eines Proof Request angestoßen. Bei der Erzeugung dieses Proof Request (Request Presentation) wird dieser um eine Thread ID ergänzt. Diese Thread ID wird in der Antwort des Proof Request (Proof Presentation) hinzugefügt. **Anforderung „Übermittlung der Identitätsattribute“ ([BTR71], G22):** Es muss sichergestellt sein, dass Identitätsattribute erst nach erfolgter Freigabe durch die Person übermittelt werden.

**Umsetzung: Im Rahmen des Identitätsnachweises erfolgt eine visuell erfassbare Anfrage der Wallet-App, die der Nutzer bestätigen muss.**

**Anforderung „Identifizierung des Dienstes“ ([BTR71], G23):** Eine vorhergehende Identifizierung des Dienstes (und damit verbunden der Aufbau einer sicheren Verbindung) ist Voraussetzung für die nachfolgenden Kriterien und muss daher mindestens mit dem angestrebten Vertrauensniveau der Identifizierung einer Person erfolgen;

Die Vertraulichkeit der Identitätsattribute einer Person setzt eine Identifizierung des empfangenden Dienstanbieters auf gleichem Vertrauensniveau wie die Identifizierung der Person voraus.

**Umsetzung: API Keys im HTTP Header siehe 5.5.**

# A. Quellenverzeichnis


**[ISO24]** ISO/IEC 24760-1:2019: IT Security and Privacy — A framework for identity management — Part 1: Terminology and concepts, International Organization for Standardization, 2019.

**[EID14]** Verordnung (EU) Nr. 910/2014 DES EUROPÄISCHEN PARLAMENTS UND DES RATES vom 23. Juli 2014über elektronische Identifizierung und Vertrauensdienste für elektronische Transaktionen im Binnenmarkt und zur Aufhebung der Richtlinie 1999/93/EG

**[EIA12]** Durchführungsverordnung (EU) 2015/1502 der Kommission vom 8. September 2015 zur Festlegung von Mindestanforderungen an technische Spezifikationen und Verfahren für Sicherheitsniveaus elektronischer Identifizierungsmittel [..], Europäische Kommission, 2015.

**[WBG16]** Digital Identity: Towards Shared Principles for Public and Private Sector, Cooperation A joint World Bank Group – GSMA – Secure Identity Alliance Discussion Paper, World Bank Group, 2016.

**[BTR07]** TR-03107-1 Elektronische Identitäten und Vertrauensdienste im E-Government Teil 1, Bundesamt für Sicherheit in der Informationstechnik, 2016.

**[BTR71]** Bewertung von Authentisierungslösungen gemäß TR-03107 in Version 1.1.1, Anwendungs- und Vorgehensbeschreibung, Version 1.05, Bundesamt für Sicherheit in der Informationstechnik, 2020.

**[BEI21]** BSI - Elektronische Identitäten, https://www.bsi.bund.de/DE/Themen/Oeffentliche-Verwaltung/Elektronische-Identitaeten/elektronische-identitaeten_node.html, abgerufen am 07.07.2021

**[SYS21]** Systemkonzept zum Projekt 2080069021, “Digitale Identitäten: Pilotvorhaben Hotel Check-In", Version vom 28.04.2021, IBM Deutschland GmbH

**[GBK21]** Technische Beschreibung der Gerätebindung, Version v6 vom 21.07.2021, IBM Deutschland GmbH

**[DIC21]** https://identity.foundation/didcomm-messaging/spec/#perfect-forward-secrecy, abgerufen am 30.07.2021

# B. Schema für das Verifiable Credential des digitalen Führerscheinnachweises

{

"schema_name": "Fuehrerschein Demo",

"schema_version": "0.2",

"attributes":

"firstName",

"familyName", //enthält familiennameBestandteil und familienname

"academicTitle",

"placeOfBirth",

"dateOfBirth", //als YYYYMMDD

"dateOfIssuance", //als YYYYMMDD, Tag der Credential-Ausstellung

"hardwareDid",

"issuingEntity",

"id",

"generalRestrictions",

"licenseCategoryA_DateOfIssuance", //als YYYYMMDD

"licenseCategoryA_Restrictions",

"licenseCategoryA1_DateOfIssuance",

"licenseCategoryA1_Restrictions",

"licenseCategoryA2_DateOfIssuance",

"licenseCategoryA2_Restrictions",

"licenseCategoryAM_DateOfIssuance",

"licenseCategoryAM_Restrictions",

"licenseCategoryB_DateOfIssuance",

"licenseCategoryB_Restrictions",

"licenseCategoryBE_DateOfIssuance",

"licenseCategoryBE_Restrictions",

"licenseCategoryB1_DateOfIssuance",

"licenseCategoryB1_Restrictions",

"licenseCategoryC_DateOfIssuance",

"licenseCategoryC_Restrictions",

"licenseCategoryC_DateOfExpiry", //als YYYYMMDD

"licenseCategoryC1_DateOfIssuance",

"licenseCategoryC1_Restrictions",

"licenseCategoryC1_DateOfExpiry",

"licenseCategoryCE_DateOfIssuance",

"licenseCategoryCE_Restrictions",

"licenseCategoryCE_DateOfExpiry",

"licenseCategoryC1E_DateOfIssuance",

"licenseCategoryC1E_Restrictions",

"licenseCategoryC1E_DateOfExpiry",

"licenseCategoryD_DateOfIssuance",

"licenseCategoryD_Restrictions",

"licenseCategoryD_DateOfExpiry",

"licenseCategoryD1_DateOfIssuance",

"licenseCategoryD1_Restrictions",

"licenseCategoryD1_DateOfExpiry",

"licenseCategoryDE_DateOfIssuance",

"licenseCategoryDE_Restrictions",

"licenseCategoryDE_DateOfExpiry",

"licenseCategoryD1E_DateOfIssuance",

"licenseCategoryD1E_Restrictions",

"licenseCategoryD1E_DateOfExpiry",

"licenseCategoryL_DateOfIssuance",

"licenseCategoryL_Restrictions",

"licenseCategoryT_DateOfIssuance",

"licenseCategoryT_Restrictions",

"licenseCategoryM_DateOfIssuance",

"licenseCategoryM_Restrictions".