Dieses Dokument wurde in Zusammenarbeit von IBM Deutschland GmbH, esatus AG, Bundesdruckerei GmbH im Rahmen des Projektes "Ökosystem Digitaler Identitäten" erstellt.

eID Integration im EESDI Ökosystem

Konzept zur Anwendung von eID für die höchste Vertrauensebene

Inhaltsverzeichnis

1. Zweck des Dokuments

Nach Beschluss des Lenkungskreises vom 01.12.2021 und der kommunizierten Entscheidung zur Koexistenz einer Basis-ID und (Smart-)eID beschreibt dieses Dokument die entsprechenden Details der daraus resultierenden Referenz-Lösungsarchitektur.
Dabei basiert die beschriebene Lösung auf einer Integration der SSI Architektur (Hyperleder Indy und Aries) für nicht hoheitliche (bzw. use cases ohne Anforderung eines „hohen“ Vertrauensniveaus) und der (Smart-)eID für hoheitliche Identitäten (bzw. Anwendungen mit Anforderung eines „hohen“ Vertrauensniveaus).

1.1. Abgrenzung

DLT-Technologie: Dieses Dokument beschreibt keine Evaluierung um die zugrundeliegende DLT-Technologie und deren kryptographischen Algorithmen durch klassische PKI-Lösungen zu ersetzen – sondern beschreibt wie diese in Koexistenz zu einer PKI basierten eID-Lösung funktionieren kann. Dabei ist jedoch nicht auszuschließen, dass andere Architekturkonzepte sich mit solch einer Thematik beschäftigen können/sollen.

1.2. Ziel Publikum

- Lösungsarchitekten
- Technische Projekt Stakeholders
- IT-Sicherheitsexperten (z.B. BSI)

2. Problemstellung

Das Ziel des ID-Ökosystems EESDI ist es eine uniforme Lösungsarchitektur anzubieten welche privatwirtschaftliche als auch öffentliche Anwendungsfälle unterstützen kann. Dabei gehen die Anforderungen hinsichtlich Sicherheit und einfacherer Verwendbarkeit (und damit auch die Skalierbarkeit des Ökosystems) stark auseinander.

2.1. Sicherheit

Hoheitliche Anwendungsfälle, wie der Personalausweis als auch eventuell der digitale Führerschein, repräsentieren sensible Daten, welche von einer staatlichen Behörde ausgestellt wurden und somit auch einen entsprechenden hoheitlichen Charakter besitzen. Anwendungsfälle, welche mit solch hoheitlichen Daten arbeiten wollen/müssen, erfordern eine entsprechende Sorgsamkeit, um sicherzustellen, dass die Daten nicht missbräuchlich verwendet werden.

Entsprechende Sicherheitsmaßnahmen wurden in die eID eingebaut. Diese hat es jedoch in den letzten 10 Jahren nicht geschafft, eine breite Akzeptanz zu erreichen. Nur 7% der deutschen Bevölkerung haben ausgesagt, die eID-Funktion ihres elektronischen Personalausweises (ePA) zumindest einmal genutzt zu haben.

2.2. Einfache Verwendbarkeit

Vergleichbar gibt es einige Anwendungsfälle, welche ohne hoheitliche Daten arbeiten wollen/müssen. Entsprechend sind die Sicherheitsanforderungen deutlich geringer und die einfache Nutzbarkeit und Skalierung des Anwendungsfalls stehen im Vordergrund. Einschränkungen, die aus der eID-Funktion resultieren (Berechtigungszertifikat, etc.), sind nicht akzeptabel.

Mit Verifiable Credentials und Verifiable Presentations nach den 10 SSI-Prinzipien[i], entsteht gerade eine neue Technologie, welche die Nutzbarkeit unter Berücksichtigung von Privatsphäre in den Vordergrund stellt. Dabei kommen neue Zero-Knowledge-Signaturverfahren zum Einsatz, welche erst seit wenigen Jahren existieren und noch nicht ausreichend in der Praxis getestet wurden

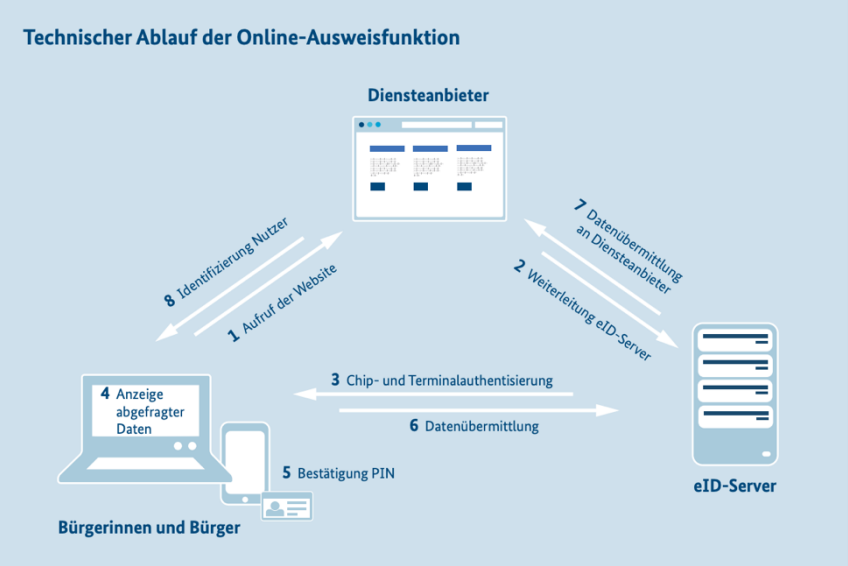
Beide Blickwinkel sind aus den unterschiedlichen Anwendungsfällen valide und müssen entsprechend adressiert werden.

3. Architekturüberblick eID und SSI

Das folgende Kapitel beschreibt einen Überblick der bestehenden Lösungsarchitekturen zur eID und SSI.

3.1. eID Architektur

Das folgende Diagramm[1] beschreiben einen typische Ablauf des Identitätsnachweises nach eID.



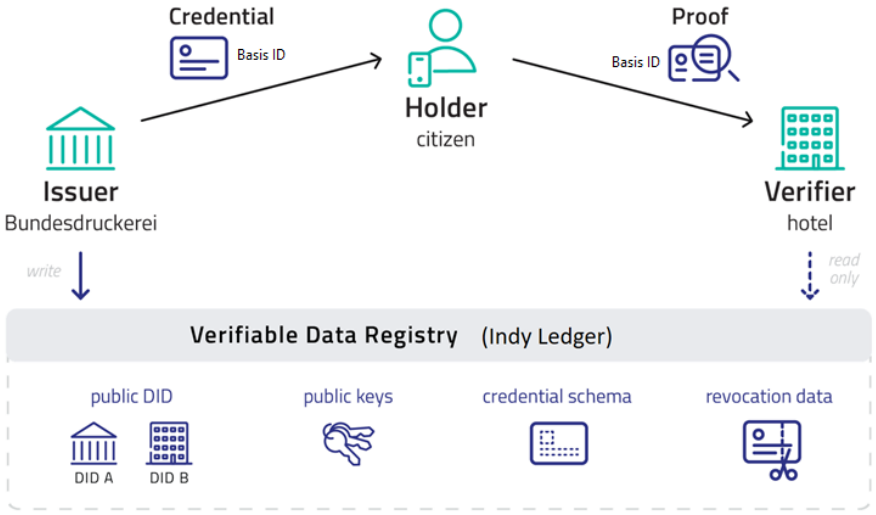
- Der **Holder** (Nutzer) ruft den Online-Dienst des Dienstanbieters auf, der eine Online-Identifizierung benötigt (1).
- Der **Verifier** (Online-Dienst Dienstanbieter) leitet eine Identifizierungsanfrage an den eID Server weiter (2).
- Zwischen dem eID-Server sowie dem Identifizierungsclient (bzw. NFC Lesegerät im Smartphone und dem Chip des eID Karte) wird ein sicherer TLS Kanal aufgebaut (3).
- Die Authentizität des Dienstanbieters sowie die Authentizität und Integrität des Ausweises werden geprüft. Zugleich erfolgt ein Abgleich des Online-Ausweises mit der Liste der gesperrten Ausweise im eID Server (**Sperrliste**).
- Wenn gewünscht, kann sich der Nutzer das Berechtigungszertifikat des Dienstanbieters und die angefragte Ausweisdatenkategorien im Identifizierungsclient anzeigen lassen (4).
- Durch Eingabe der PIN stimmt der Ausweisinhaber der Übermittlung der angefragten Ausweisdaten zu (5).
- Die Ausweisdaten werden an den eID-Server entsprechend dem PAOS flow übermittelt (6).

[1] Bundesministerium des Innern und für Heimat. *Anwenderhandbuch für Wirtschaft und Verwaltung: „Digitale Identifizierung mit dem deutschen Online-Ausweis - Informationen für Unternehmen und Behörden“* https://www.bmi.bund.de/SharedDocs/downloads/Webs/PA/DE/informationsmaterial/informationen-formulare-diensteanbieter/anwenderhandbuch.pdf?__blob=publicationFile&v=3

- Der eID-Server sendet eine Authentifizierungsantwort und die Ausweisdaten an den Diensteanbieter (7).
- Die Authentifizierungsantwort und die Ausweisdaten werden ausgelesen. Der Diensteanbieter prüft die Ergebnisse und entscheidet ob die Identifizierung erfolgreich war. Abschließend erfolgt eine Ergebnisantwort an den Nutzer (8).

3.2. SSI Architektur

Das folgende Diagramm zeigt eine vereinfachte Darstellung der SSI Architektur und deren Akteuren.



- Der **Issuer** (Bundesdruckerei) stellt ein signiertes, gerätegebundenes Credential (Basis-ID, Digitaler Führerschein) auf Grundlage der eID Attribute des Personalausweises aus.

- Der **Holder** (Nutzer) erhält nach Durchführung der Online-Ausweisfunktion die Basis-ID in seiner ID-Wallet App. Dies ermöglicht eine spätere Nutzung der Basis-ID mit dem beim Ausstellungsprozess verwendeten mobilen Endgerät.

- Der **Verifier** (z.B. Hotel) prüft die vom Nutzer vorgezeigten Credentials (Basis-ID und z.B. Arbeitgeberbescheinigung) während eines Authentifizierungsvorgangs. Dabei kann er die Authentizität der Daten mit Hilfe der Prüfinfrastruktur und die Gerätebindung mit Hilfe der ID-Wallet App verifizieren.

- Das **Verifiable Data Registry** ist ein öffentliches, genehmigungspflichtiges Blockchain-Netzwerk (Hyperledger Indy) zur Speicherung von öffentlichem Schlüsselmaterial und dezentrale Prüfinfrastruktur, betrieben von authentisierten und autorisierten Knotenbetreibern. Eine Blockchain kommt daher zum Einsatz, um die Verfügbarkeit der öffentlichen Schlüsselmaterialien als auch Schema und Revozierungsdaten zu erhöhen. Zusätzlich eliminiert die Blockchain die Notwendigkeit einer zentralen

Organisation, welche das öffentliche Schlüsselmaterial, Schema und Revorzierungsdaten verwaltet.

4. Anforderungen

Die folgenden Anforderungen wurden aus den diversen Arbeitsgruppen mit den externen Projektpartnern, Community (CCC, bitkom) und dem BSI erarbeitet. Diese Anforderungen motivieren die hier beschriebene Lösungsarchitektur.

- **AF001**: Die eID Funktionalität **muss** parallel zu den X509 und "Anoncreds" Credentials verwendbar sein.
- **AF002**: SSI und eID Funktionen **müssen** in einer integrierten Wallet App zur Verfügung gestellt werden.
- **AF003**: Die Implementierung der SSI Prozesse **müssen** das eIDAS LoA (Vertrauens-Niveau) "Substantiell" ohne eID Funktion ermöglichen.
- **AF004**: Der Implementierungs- und Wartungsaufwand für Verifier **muss** so gering wie möglich gehalten werden.
- **AF005**: Die Lösung **muss** auf offenen Standards basieren die es ein jedem externen (Verifier) ermöglichen zumindest auf dem Vertrauens-Niveau "Niedrig" entsprechend dem Trusted Verifier Konzept teilzunehmen.
- **AF006**: Das Ausstellen von Credentials (X509 oder Anon Cred) **muss** autorisiert sein.
- **AF007**: Die Lösung **muss** kompatibel mit der eIDAS / EU Lösung[2] sein.
- **AF008**: Alle Entwicklungen des Ökosystems **sollen** open source gestellt werden.
- **AF009**: Die Lösung **muss** auf dem Aries Interoperabilität Protokoll 1.0 basieren.
- **AF010**: Die Lösung **muss** ausschließlich auf open source Komponenten und open Standards entwickelt werden.

4.1. Prüfanforderungen aus [TR-03128-2]

Die folgenden Anforderungen ergeben sich aus der TR-03128-2[3] um einen eID Service (Identifizierungsdiensteanbieter) zu betreiben.

- **PAF001**: Es ist ein Sicherheitskonzept vorhanden.
- **PAF002**: Das Sicherheitskonzept berücksichtigt alle Prozesse und Komponenten der eID-Infrastruktur.
- **PAF003**: Die im Sicherheitskonzept beschriebenen Maßnahmen zum sicheren Betrieb des eID-Servers berücksichtigen die Mindestanforderungen aus [TR-03130] Teil 2 und die Vorgaben aus [CP CVCA-eID].

[2] https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TR03159/TR-03159-1.pdf;jsessionid=49547EA47459AF3E65B9155406C23AC8.internet472?__blob=publicationFile&v=1

[3] https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03128/BSI_TR-03128-2_Checkliste.pdf?__blob=publicationFile&v=1

- **PAF004**: Das Sicherheitskonzept ist Bestandteil eines ISMS.
- **PAF005**: Das ISMS umfasst alle Organisationseinheiten, die operativ am Betrieb der eID-Infrastruktur beteiligt sind. Hinweis: Die Anforderung gilt gleichermaßen für alle Komponenten deren operativer Betrieb (ganz oder teilweise) an Dritte ausgelagert ist.
- **PAF006**: Das ISMS ist nach [ISO27001] oder ISO 27001 auf Basis IT-Grundschutz [IT-GS]. zertifiziert. Hinweis: Die Anforderung der Zertifizierung gilt für jedes ISMS das gemäß [TR-03128-2] gefordert ist.
- **PAF007**: Für personenbezogene oder personenbeziehbare Daten, die über öffentliche Netze übermittelt werden, ist die Vertraulichkeit und Integrität gemäß den Anforderungen aus [TR-03116] geschützt.
- **PAF008**: Der Identifizierungsdiensteanbieter identifiziert und registriert den Auftraggeber (Endverwender der Daten) mit einem „hohen“ Vertrauensniveau gemäß [TR-03107] Teil 1, bevor er Daten an den Auftraggeber übermittelt.
- **PAF009**: Der Identifizierungsdiensteanbieter gibt dem Auftraggeber die Möglichkeit, die abgefragten Daten auf das notwendige Maß für die Anwendung zu beschränken.
- **PAF010**: Der Identifizierungsdiensteanbieter stellt technisch sicher, dass nur die angefragten Daten an den Auftraggeber übermittelt werden.
- **PAF011**: Die Mechanismen für die Kommunikation zwischen Identifizierungsdiensteanbieter und Auftraggebern erfüllen in jedem Fall Sicherheitsniveau „hoch“ gemäß [TR-03107] Teil 1. Falls hierbei Verfahren eingesetzt werden, die in [TR-03116] Teil 4 beschrieben sind, so sind die dort beschriebenen Vorgaben verpflichtend umgesetzt.
- **PAF012**: Eine Protokollierung personenbezogener oder personenbeziehbarer Daten erfolgt ausschließlich dann, wenn dies für den Zweck der Identifizierung notwendig ist.
- **PAF013**: Personenbezogene oder personenbeziehbare Daten aus der Online-Ausweisfunktion werden nur insoweit und nur solange wie technisch notwendig mit Protokolldaten verknüpft.
- **PAF014**: Personenbezogene Daten aus der Online-Ausweisfunktion werden gelöscht, sobald die Identifizierung abgeschlossen und gegebenenfalls das elektronische Formular sowie die auf Grund gesetzlicher Aufzeichnungspflichten aufgezeichneten. Daten an den Auftraggeber übermittelt wurden.

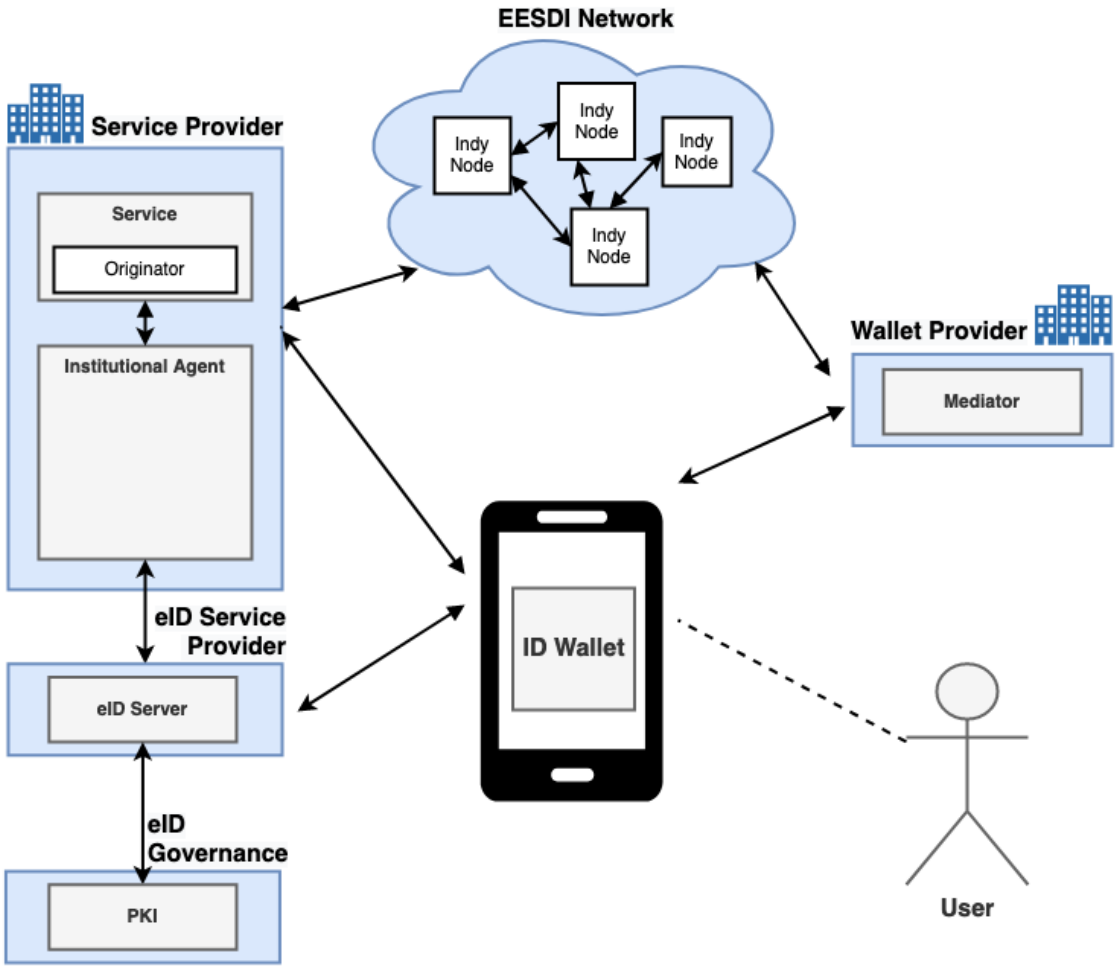
5. Referenzarchitektur

Das folgende Kapitel beschreibt eine Referenz-Lösungsarchitektur zur Integration von eID und SSI Funktionalitäten. Dabei werden hauptsächlich die folgenden Designziele verfolgt.

1. Vereinfachung für ein Unternehmen bzw. eine Entwickler:in eine SSI und eID integrierte Lösung durch skalierbare APIs umzusetzen.
2. Entkopplung wiederverwendbarerer Architekturbausteine um modular nutzbar zu sein.

5.1. System Kontext

Das folgende Diagramm illustriert den System Kontext und ihre Akteure einer eID und SSI integrierten Lösungsarchitektur. Die Nutzer:in benutzt dabei die ID Wallet auf ihrem Smartphone um sich bei einem Dienstanbieter zu identifizieren.



5.1.1. Service Provider

Der Service Provider repräsentiert eine Organisation, welche eine Integration mit dem EESDI Ökosystem umsetzt. Dabei implementiert und betreibt der Service Provider zumindest einen Web Service und einen Institutional Agent (siehe Kapitel 5.2). Über diese Komponenten kann der Service Provider dem Nutzer eine SSI (Basis-ID) oder eID basierte Identifikation anbieten.

5.1.2. eID Service Provider

Der eID Service Provider betreibt einen eID Server samt Berechtigungszertifikat und verantwortet die sichere Verarbeitung einer eID Identifizierung. Dabei kann diese Rolle vom Service Provider selbst oder einem Drittanbieter (z.B. D-Trust, skIDentity, etc.) ausgefüllt werden.

5.1.3. Wallet Provider

Der Wallet Provider stellt die ID Wallet zur Verfügung und betreibt den Mediator, der die Kommunikation zwischen der ID Wallet und dem EESDI Ökosystem unterstützt (siehe Kapitel 5.2).

5.1.4. EESDI Network

Das EESDI Network repräsentiert die Prüfinfrastruktur für die SSI basierten Nachweise und wird von unterschiedlichen Knotenbetreibern betrieben. Dabei werden öffentliche Schlüssel, Revozierungsdaten, Schema und Claim Definitions unveränderbar auf einer blockchain festgehalten.

5.1.5. eID Governance

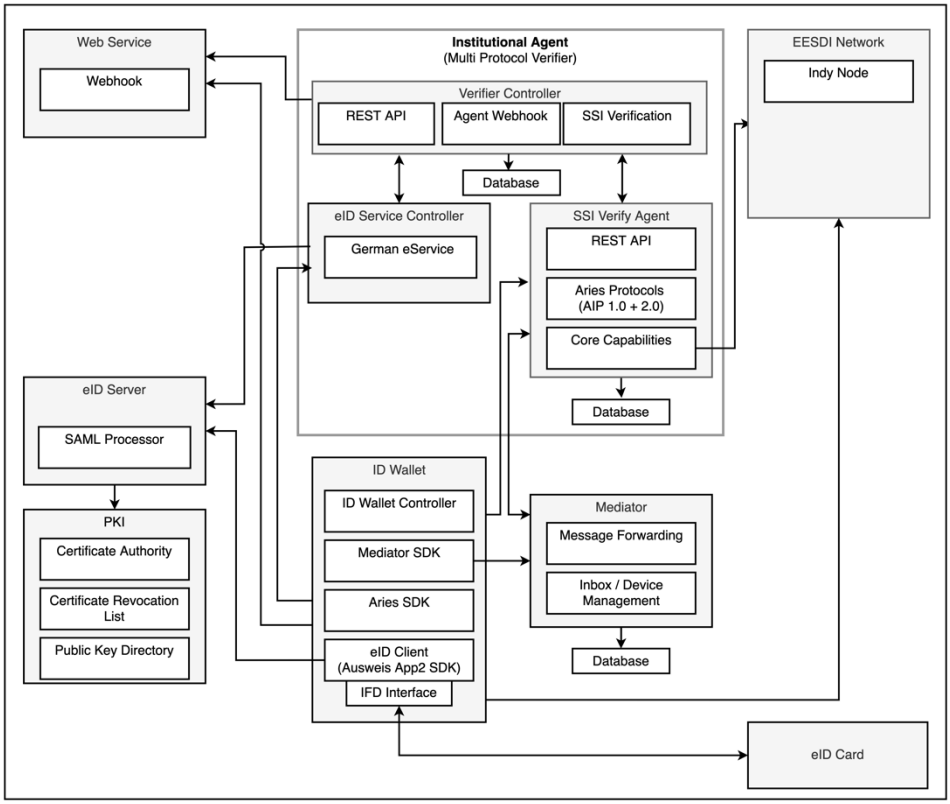
Die eID Governance verantwortet die Verwaltung der Berechtigungszertifikate, Sperrlisten und betreibt die Public Key Infrastruktur (PKI). An dieser PKI sind eine Reihe von Behörden und Institutionen beteiligt4:

- das Bundesamt für Sicherheit in der Informationstechnik (BSI) als Betreiber der Root-CA,
- das Bundesverwaltungsamt mit der Vergabestelle für Berechtigungszertifikate (VfB) als Betreiber der Registration Authority (RA) sowie
- die Zertifikateanbieter, die die technische Ausstellung der Berechtigungszertifikate übernehmen.

5.2. Komponenten Diagramm

Das untenstehende Diagramm zeigt die Komponenten, welcher zur Implementierung eines Multi Protocol Verifiers und zur Integration von einer SSI und eID Lösung verwendet werden kann.

[4] https://www.personalausweisportal.de/Webs/PA/DE/wirtschaft/technik/technik-node.html



5.2.1. Web Service

Die logische Komponente, welche die Business Logik verantwortet und mittels Webhook einen Rücksprung ermöglicht um eine Verifizierung oder Identifizierung zu verarbeiten.

5.2.2. Institutional Agent

Der Institutional Agent stellt eine einheitliche Schnittstelle zur Verfügung um Protokoll unabhängig (SSI, eID) die folgenden Aktionen umzusetzen.

1. **Identifikation**: Eine natürliche Person wird mittels einem eIDAS LoA “substantiell” oder “hoch” Identitätsverfahren rechtssicher identifiziert.
2. **Verifizierung**: Aussagen über eine Person von vertrauenswürdigen Austellern können verifizierbar abgefragt werden.

Die REST API kapselt dabei die zugrundeliegenden Protokolle auf eine Meta Protokoll Ebene. Das Web Service gibt mittels der Schnittstelle bekannt welche Aktion durchgeführt werden soll und der eID Service oder Verifier Controller übernimmt die entsprechenden Protokoll Schritte.

Die Verifizierung basiert auf SSI Credentials und wird mittels der REST API agnostisch zur Verfügung gestellt. Das Web Service muss sich nicht mit den Details der ACA-PY's API oder dem eID Protokoll auseinandersetzen.

Verifier Controller
Der Verifier Controller ist der Controller des SSI und eID Backends der Institution und übernimmt die folgenden Aufgaben:

- Verarbeitung von SSI bzw. eID Anfragen vom Web Service
- Koordinierung und Weiterleitung von eID Identifikationsanfragen
- Koordinierung und Weiterleitung von SSI Identifikation- und Verifizierungsanfragen (Proof Requests)

SSI Verify Agent
Der SSI Verify Agent stellt die Schnittstellen für die Kommunikation mit dem EESDI Netzwerk und anderen Agenten zur Verfügung. Darüber hinaus verantwortet der SSI Verify Agent die Verarbeitung des Aries Protokoll entsprechend dem Aries Interopability Protocols[5].

eID Service Controller
Der eID Service Controller stellt die Schnittstelle für die Anwendung des eID Verfahrens zur Identifikation dar und übernimmt die folgenden Aufgaben.

- Erstellung der TLS Verbindungen zum eID Client und eID Server.
- Erstellung des tcTokens.
- Verarbeitung des Identifikationsergebnis.

5.2.3. eID-Server

Der eID-Server stellt eine einfache Schnittstelle für den eService Controller und die ID Wallet bereit, um die Komplexität der eID Identitätsfunktion zu kapseln und übernimmt die folgenden Aufgaben.

- Erstellt eine sichere Kommunikation mit der ID Wallet und der eID Card und gibt ausgelesene Daten an den eService Controller weiter.
- Stellt die Authentizität und die Gültigkeit des Personalausweises fest, prüft, ob dieser von der Ausweisinhaberin oder dem Ausweisinhaber gesperrt wurde, und übermittelt die Ergebnisse der eID-Funktion an den eService Controller.
- Bezieht von der eID Governance (PKI) regelmäßig neue Berechtigungszertifikate sowie aktualisierte Sperrlisten.

Der eID-Server muss die Richtlinien der BSI TR-03130 "eID-Server"[6] folgen. Insbesondere muss der Betreiber des eID-Servers ein Berechtigungszertifikat beim Bundesverwaltungsamt erwerben und alle technischen und rechtlichen Vorgaben ausführlich folgen (siehe auch Kapitel 4.1 für eine Liste der Anforderungen eines eID Service Anbieters). Der Service Provider hat dabei die Option seinen eigene eID-Server samt eigenem Berechtigungszertifikat zu betreiben oder eine eID-Service eines Drittanbieters zu nutzen. Drittanbieter eID-Services, wie zum Beispiel Ausweisident von D-Trust, bieten APIs (z.B. OpenID Connect) und/oder eine SDK, die Zugriff auf denen eID-Server erlauben.

[5] https://github.com/hyperledger/aries-rfcs/blob/main/index.md
[6] https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr03130/TR-03130_node.html

SAML Processor
Der SAML Processor stellt die Schnittstellen für die Authentifizierung mittels SAML Nachrichten in Verbindung mit dem eID Verfahren zur Verfügung.

5.2.4. EESDI Network

Das EESDI Network repräsentiert das Verifiable Data Registry, in welchem die Credential Schemas, Public DIDs, Revocation Lists and Public Keys abgelegt sind und stellt dadurch eine dezentrale Prüfinfrastruktur dar. Das EESDI Network besteht aus mehrere Hyperledger Indy Knoten, die von verschiedenen Organisationen betrieben werden wodurch es keine zentrale Abhängigkeit gibt.

5.2.5. ID Wallet

Die ID Wallet ist eine App auf dem Smartphone zum Speichern, Freigeben und Übertragen von Credentials. Der eID Client ist als AusweisApp2 SDK in die ID Wallet App integriert. Dadurch wird die Nutzung der (Smart)-eID über die ID Wallet basierend auf der eID PIN-Eingabe ermöglicht.

ID Wallet Controller
Der ID Wallet Controller ist eine interne Komponente der ID Wallet App, sie dient zur Erkennung ob es sich bei eingehenden Anfragen um (Smart)-eID Anfragen oder Proof Requests nach dem Anoncreds Schema handelt.

Mediator SDK
Das Mediator SDK dient der ID Wallet zur Kommunikation mit dem Mediator Agent. Dort legt die ID Wallet unter Verwendung des Mediator SDK eine Inbox an und regelt den Abruf eingehender DIDComm Nachrichten.

Aries SDK
Das Aries SDK dient der ID Wallet als Grundlage zur Verwaltung des lokalen Aries Agent. Es beinhaltet die Umsetzung der eingesetzten Kryptographie und setzt somit alle nötigen Funktionen bezüglich des DIDComm Verbindungsaufbaus, der Ausstellung von Credentials und der Beantwortung von Proof Requests. Das Aries SDK implementiert hierfür die Funktionen des Hyperledger Indy-SDK.

eID Client
Der eID-Client stellt die verschlüsselte Kommunikation zwischen der eID Card, dem eID Server und dem eID Service Controller (eID Application) her und übernimmt die folgenden Aufgaben.

- Erstellung einer sicheren Kommunikation mit der eID Server und der eID Card und gibt ausgelesene eID Daten an den eID Server weiter.
- Prüfung der Authentizität und Gültigkeit des Berechtigungszertifikats des eID Servers (gegenseitige Authentifizierung).

Der eID-Client muss den Richtlinien von BSI TR-03124 "eID-Client"7folgen. In der Referenzarchitektur ist der eID-Client in der ID Wallet integriert, basiert auf der eID-SDK (AusweisApp2 SDK) und kommuniziert mit dem eID-Server über eine TLS-gesicherte Verbindung. Über das IFD Interface kommuniziert der eID-Client mit der eID-Card.

IFD Interface
Das IFD Interface stellt eine einheitliche API für jegliche eID Lesegeräte und eID Karten dar und ist in der BSI TR-03112-6 "eCard-API-Framework – IFD-Interface"8 spezifiziert. Das IFD Interface kann entweder „Local“ oder „Remote“ sein. Ein lokales IFD kommuniziert mit der eID Card durch ein angeschlossenes Kartenlesergerät oder eine integrierte NFC Schnittstelle vom Smartphone. Ein remote IFD kommuniziert über eine Netzwerkverbindung mit einem anderen Rechner oder Smartphone und dessen Kartenlesegerät/NFC Schnittstelle. Mit einem remote IFD ist es möglich, z.B. ein Smartphone als eID-Card Lesegerät an einem Computer zu nutzen.

5.2.6. Mediator
Der Mediator ist der Cloud Agent für den Edge Agent der ID Wallet und ermöglicht die direkte Kommunikation via eines statischen Endpunktes. Da der Mediator für alle ID Wallets verwendet wird, wird dadurch implizit eine “herd privacy” implementiert. Der Mediator kann dabei nicht den Inhalt der Nachrichten selbst einsehen und wird lediglich für das Routing zu der mobilen Anwendung (ID Wallet) genutzt. Ankommende DIDComm Nachrichten werden in der entsprechenden Inbox für die ID Wallet gespeichert. Mittels “Push Notifications” wird die ID Wallet über neue Nachrichten in ihrer Inbox informiert.

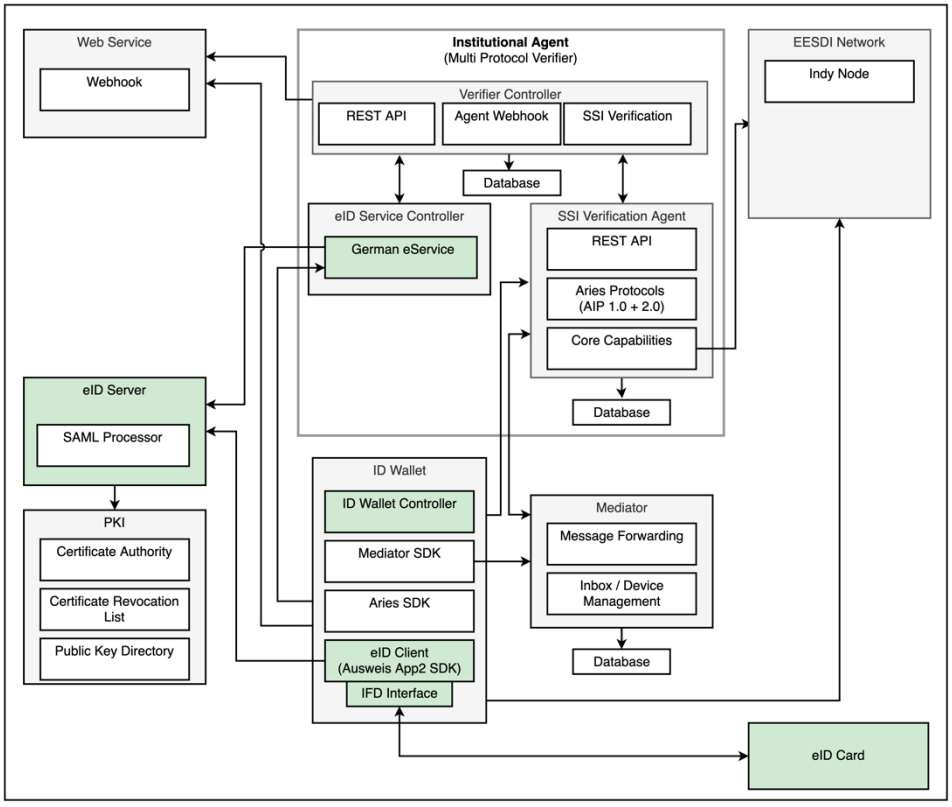
5.2.7. PKI
Die Public Key Infrastruktur repräsentiert die Prüfinfrastruktur für das eID Verfahren und beinhaltet eine Root Certificate Authority, eine Certificate Revocation List (Sperrliste) und ein Public Key Directory. Mittels dieser Prüfinfrastruktur lässt sich zentral die Gültigkeit eines Berechtigungszertifkats prüfen.

5.3. Abläufe

Das folgende Kapitel beschreibt die wichtigsten Abläufe zur (Smart-)eID und SSI (Basis ID) Identifikation und Verifikation basierend auf SSI credentials.

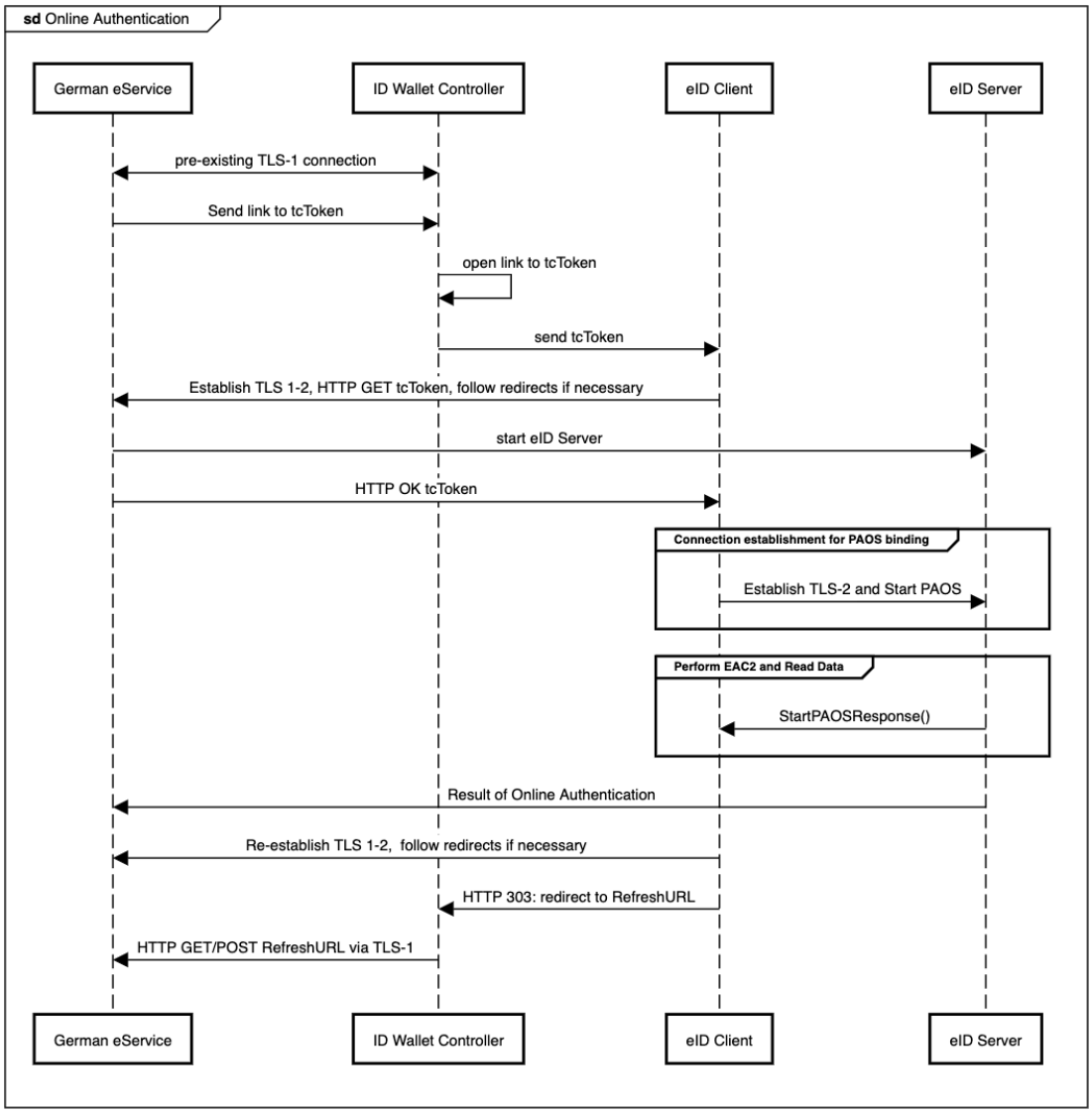
5.3.1. eID Identitätsnachweis
Dieser Ansatz beschreibt alleinigen Identitätsnachweis mittels eID ohne SSI Proof Request, mit welchem ein eIDAS LoA (Vertrauensebene) „hoch“ erreicht werden kann. Dieser Ansatz ermöglicht die ID Wallet App bei (non-SSI) eID basierte Systems benutzt zu werden.

[7] https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr03124/TR-03124_node.html
[8] https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03112/TR-03112-api_teil6.pdf?__blob=publicationFile&v=1



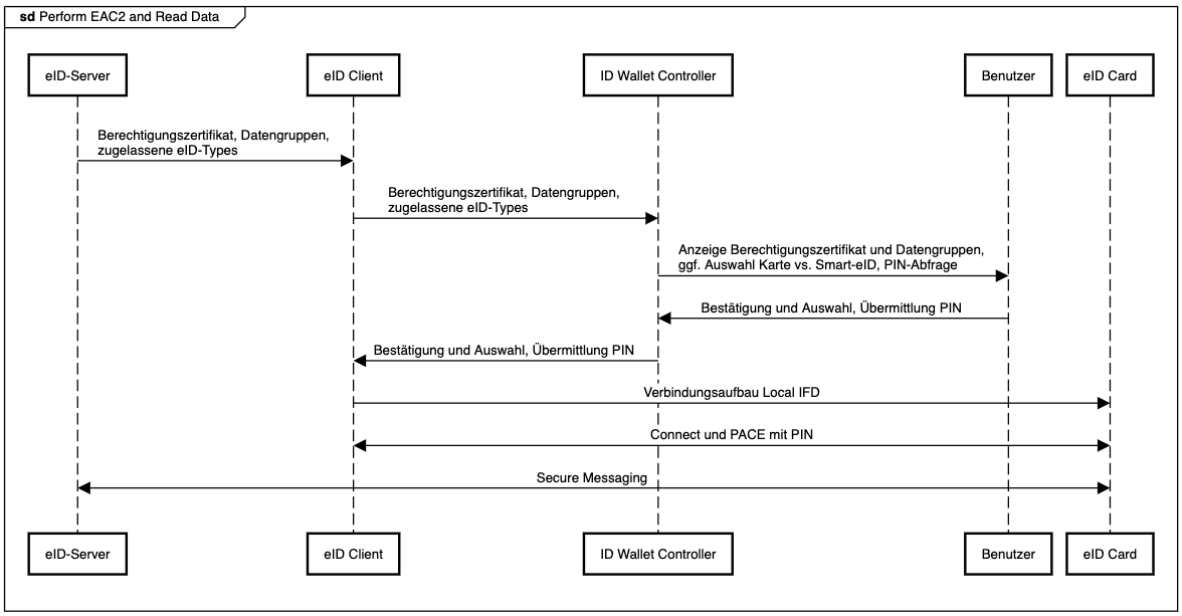
Das folgende Sequenzdiagram beschreibt ein vereinfachtes Sequenzdiagramm für eine online Authentisierung aus der TR-031249.

[9] https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr03124/TR-03124_node.html



1. Der ID-Wallet Controller kommuniziert mit der German e-Service durch eine bereits existierende Verbindung (TLS-1). Der eService Controller sendet den tcToken Link zum ID Wallet Controller.
2. Wenn der Nutzer den Link öffnet, sendet der ID Wallet Controller den Token zum eID Client weiter.
3. Der eID Client kommuniziert mit der eService Controller durch eine neue TLS Verbindung (TLS 1-2) und holt den tcToken ab.
4. Das German eService startet den eID Server und schickt ein OK zum eID Client.
5. eID Client verbindet sich mit dem eID Server (TLS-2) und startet die PAOS Bindung.
6. eID Server startet die PAOS Response für die EAC2 um die eID Data zu lesen. Dieses Verfahren wird in dem Diagramm unten beschreibt.
7. eID Server gibt dem German eService das Ergebnis der Authentifizierung weiter.
8. eID Client verbindet sich wieder über TLS 1-2 mit dem German eService und folgt den Redirects wenn nötig.
9. eID Client gibt den Redirect zum ID Wallet Controller weiter
10. ID Wallet Controller folgt den Redirect durch die TLS-1 Verbindung.

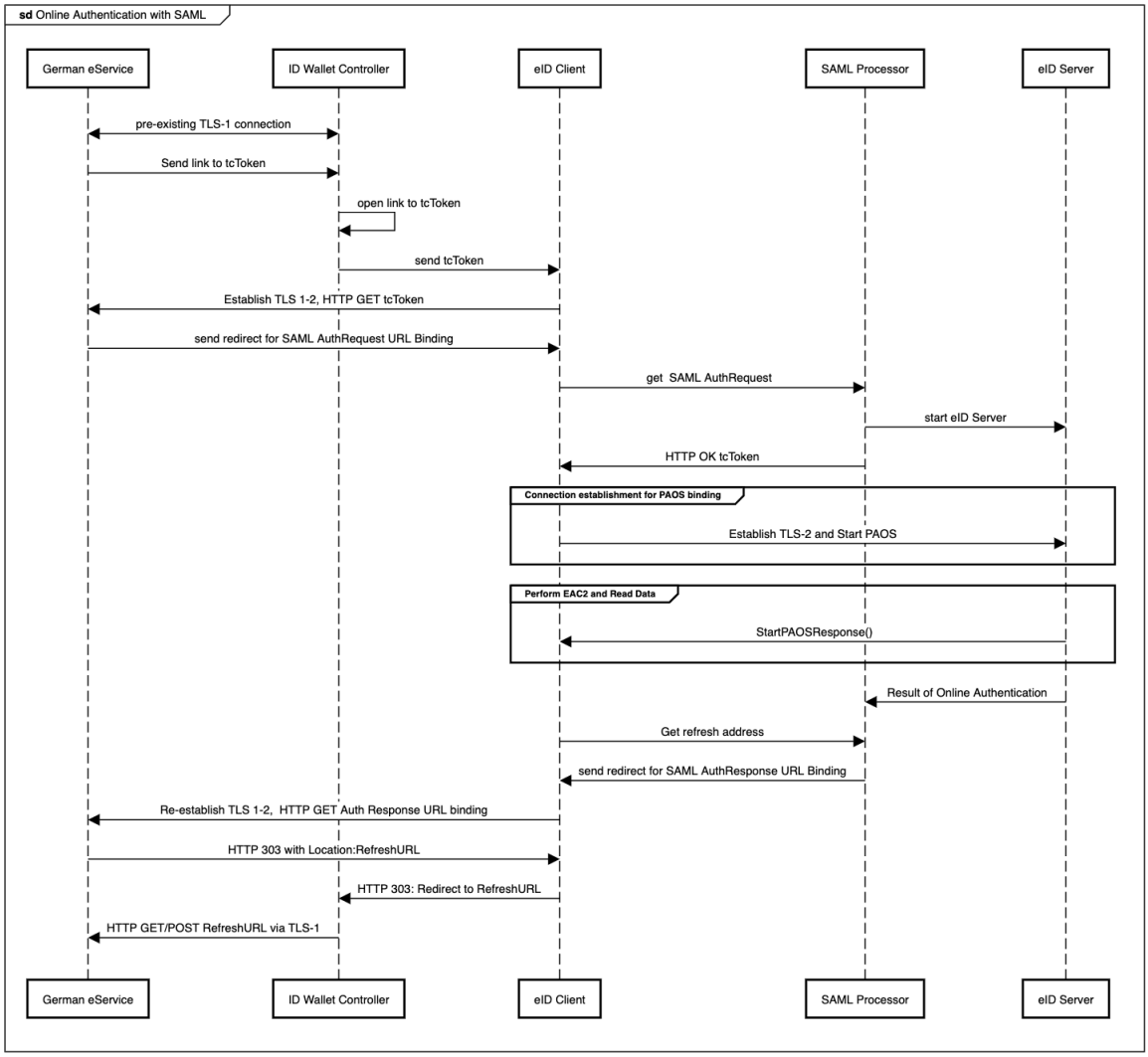
Innerhalb der Box “Perform EAC2 and Read Data” erfolgt dann die eigentliche eID-Funktion, hier genügt die folgende vereinfachte Darstellung:



- Der eID-Server sendet dem eID Client seine Berechtigungszertifikat, Datengruppen, und zugelassene eID-Types. Der eID Client gibt dem ID Wallet Controller die Daten vom eID Server weiter.
- Der ID Wallet Controller zeigt dem Benutzer das Berechtigungszertifikat und Datengruppen, gegeben falls die Auswahl zwischen eID Card und Smart-eID.
- Der Benutzer bestätigt und gibt den PIN ein.
- Der ID Wallet Controller gibt die Bestätigung und PIN zum eID Client weiter.
- Der eID Client verbindet sich mit der eID Card über eine lokale IFD.
- Der eID Client startet den Passwort authentisierten Verbindungsaufbau (PACE) mit dem RF Chip in der eID Card.
- Die Daten von der eID Card werden verschlüsselt zum eID Server übertragen.

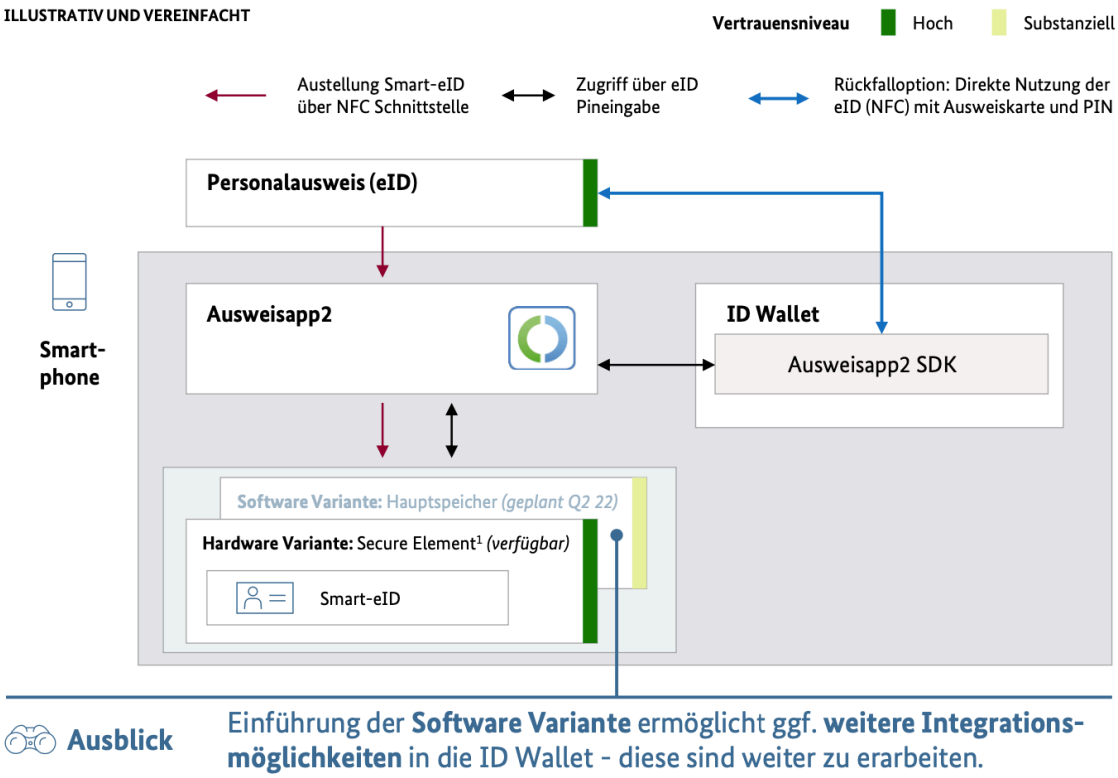
SAML

Eine eID Authentifizierung mit dem SAML Protokoll ist auch möglich. Das folgende Diagramm beschreibt den Ablauf:


sd Online Authentication with SAML
German eService
ID Wallet Controller
eID Client
SAML Processor
eID Server
pre-existing TLS-1 connection
Send link to tcToken
open link to tcToken
send tcToken
Establish TLS 1-2, HTTP GET tcToken
send redirect for SAML AuthRequest URL Binding
get SAML AuthRequest
start eID Server
HTTP OK tcToken
Connection establishment for PAOS binding
Establish TLS-2 and Start PAOS
Perform EAC2 and Read Data
StartPAOSResponse()
Result of Online Authentication
Get refresh address
send redirect for SAML AuthResponse URL Binding
Re-establish TLS 1-2, HTTP GET Auth Response URL binding
HTTP 303 with Location:RefreshURL
HTTP 303: Redirect to RefreshURL
HTTP GET/POST RefreshURL via TLS-1
German eService
ID Wallet Controller
eID Client
SAML Processor
eID Server

Smart-eID

In Kombination mit der Ausweisapp2 kann die Smart-eID über die ID Wallet genutzt werden



Derzeit wird die Smart-eID mit der Ausweisapp2 ausgestellt und in das Secure Element des Smartphones abgelegt (Hardware Variante) und erfüllt Vertrauensniveau Hoch.

- Integration der Smart-eID Nutzung in die ID Wallet erfolgt über das Ausweisapp2 SDK
- Nutzung der Smart-eID über die ID Wallet basierend auf eID PIN-Eingabe
- Künftige Software Variante der Smart-eID hebt Secure Element Restriktionen auf, sodass breitere Abdeckung von Endgeräten gegeben, und erfüllt Vertrauensniveau Substanziell
- Die eID mittels Ausweiskarte und PIN stellt in jedem Fall eine Rückfalloption für Anwendungen mit erforderlichen Vertrauensniveau Hoch dar

5.3.2. Basis ID Identitätsnachweis

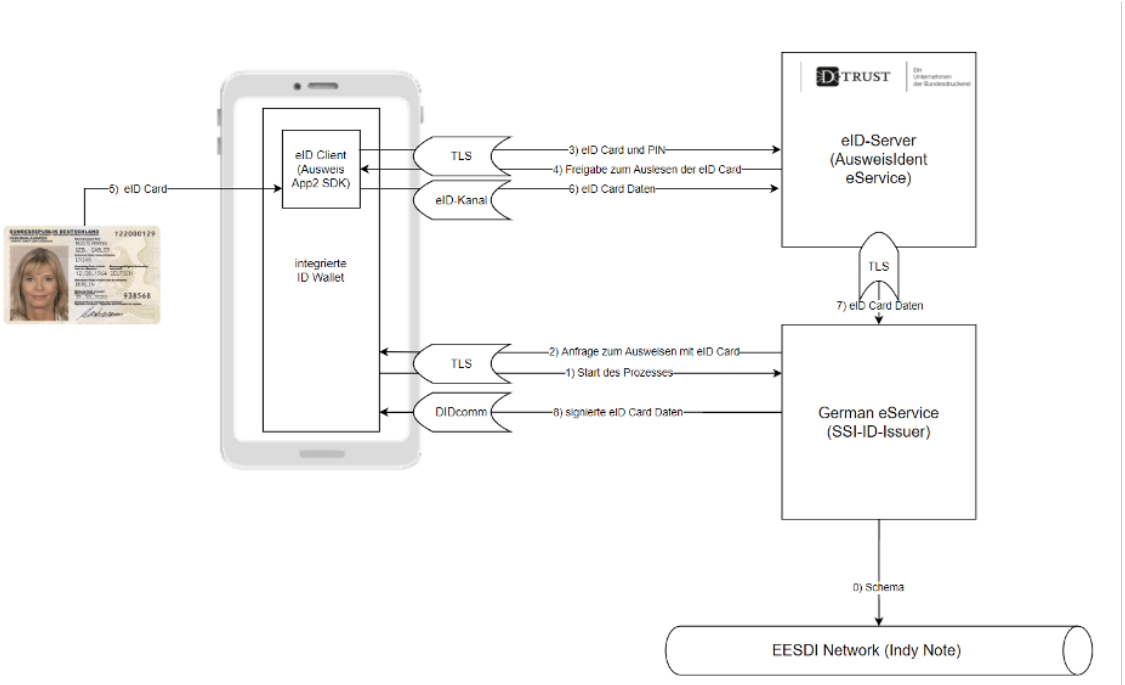
Die Basis-ID ist ein SSI-Nachweis, der auf Basis der Daten aus der Online-Ausweisfunktion des Personalausweises zur Verfügung gestellt wird und in der ID Wallet-App gespeichert wird. Die Basis-ID ist noch kein staatlich herausgegebener Identitätsnachweis.

Die initiale Identitätsprüfung bei der Ausgabe der Basis-ID findet über eine Identifizierung mittels Onlineausweisfunktion des Personalausweises des Inhabers über das AusweisApp2 SDK in der Wallet-App statt. Die Onlineausweisfunktion über das AusweisApp2 SDK erfüllt eIDAS LoA high. Die Ausgabe nur an berechtigte Personen wird durch die Kanalbündelung zwischen Onlineausweisfunktion des Personalausweises des Inhabers und Ausstellen des Credentials durch die Integration des AusweisApp2 SDKs in die ID Wallet App auf Softwareebene gewährleistet. Der Ausstellungsprozess der Basis-ID wird durch die Wallet initiiert. Sie baut einen TLS-Kanal zum API-Endpunkt des Ausstellers (SSI Issuer) auf, dieser

wird zusätzliche mit Zertifikatspinning (Public Key) geschützt. Zusammenfassend wird die Kanalbündelung erreicht, indem die Wallet zwei TLS-Kanäle mit dem Aussteller 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.

Die Integrität der Wallet-App 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 [Gerätebindungskonzept], sowie Kapitel 7.1.1. in [Systemkonzept Basis-ID]).

Die folgende Darstellung zeigt die Ausstellung der Basis-ID unter Nutzung der eID-Funktion des Personalausweises.



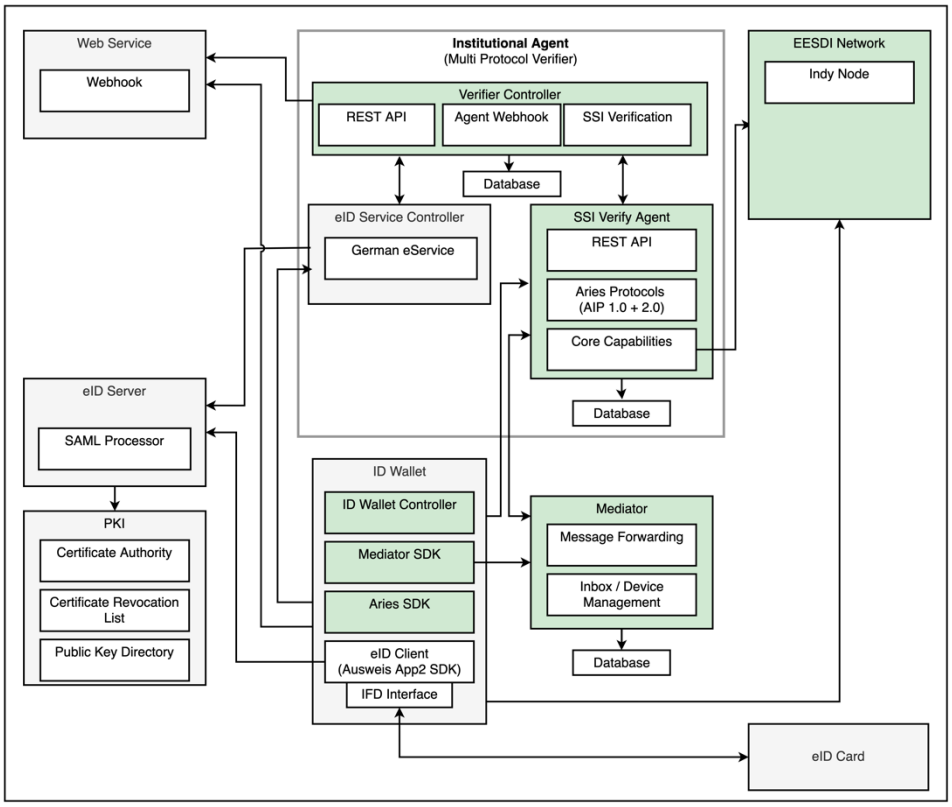
Die Verwendung der Basis-ID bietet eine Zwei-Faktor-Authentisierung. Der Faktor “Wissen” wird duch Nutzerbindung in Form der PIN und der zweite Faktor “Besitz” durch die Gerätebindung (siehe separates Dokument Referenzen) in Form eines hardwaregebundenen asymmetrischen Schlüsselpaars erreicht (privater Schlüssel der hardwareDid).

Die Inhaberidentifizierung erfolgt über die eindeutige hardwareDid (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.

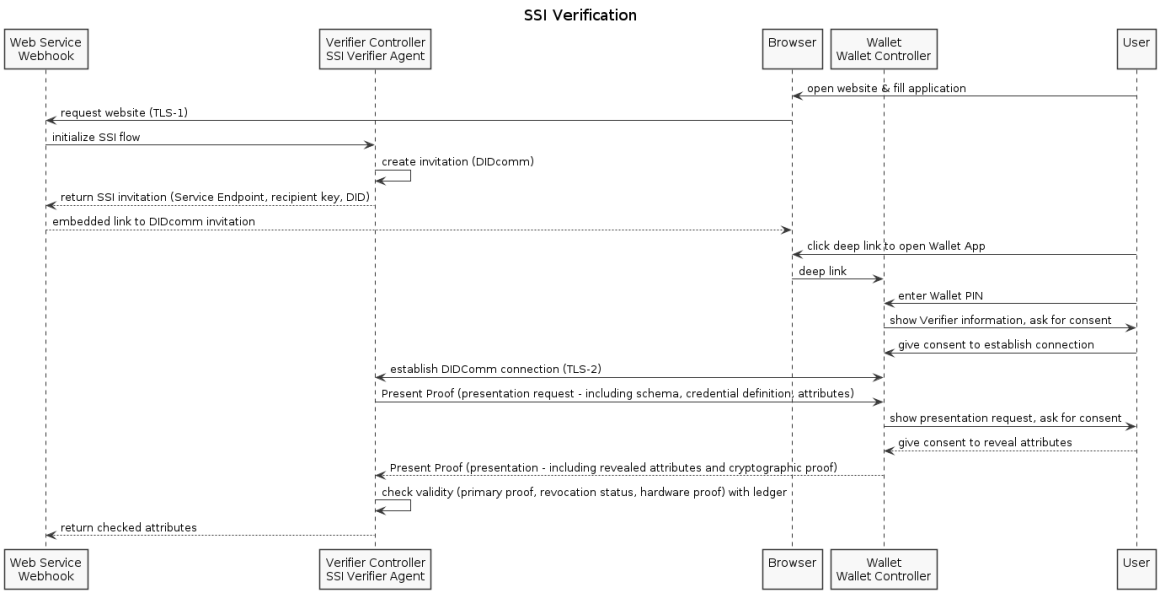
Der private Schlüssel des Challenge-Response-Verfahrens des Authentisierungsalgorithmus wird in einem hardwaregestützten TEE des mobilen Endgerätes des Nutzers generiert, gespeichert und verwendet, das mindestens gegen Angriffe mit moderatem Angriffspotenzial geschützt ist. Eine entsprechende Betrachtung der Widerstandsfähigkeit gegen moderates Angriffspotential befindet sich im Dokument Bedrohungsanalyse Authentisierung. Die Ermittlung hinsichtlich Sicherheitsniveau (substantiell) wird in einem getrennten Dokument der Prüfberichtsvorlage nach TR 03107-01 bearbeitet (siehe Referenzen).

5.3.3. SSI Verification

Das folgende Kapitel beschreibt den detaillierten Prozess zur Verifikation eines SSI credentials.

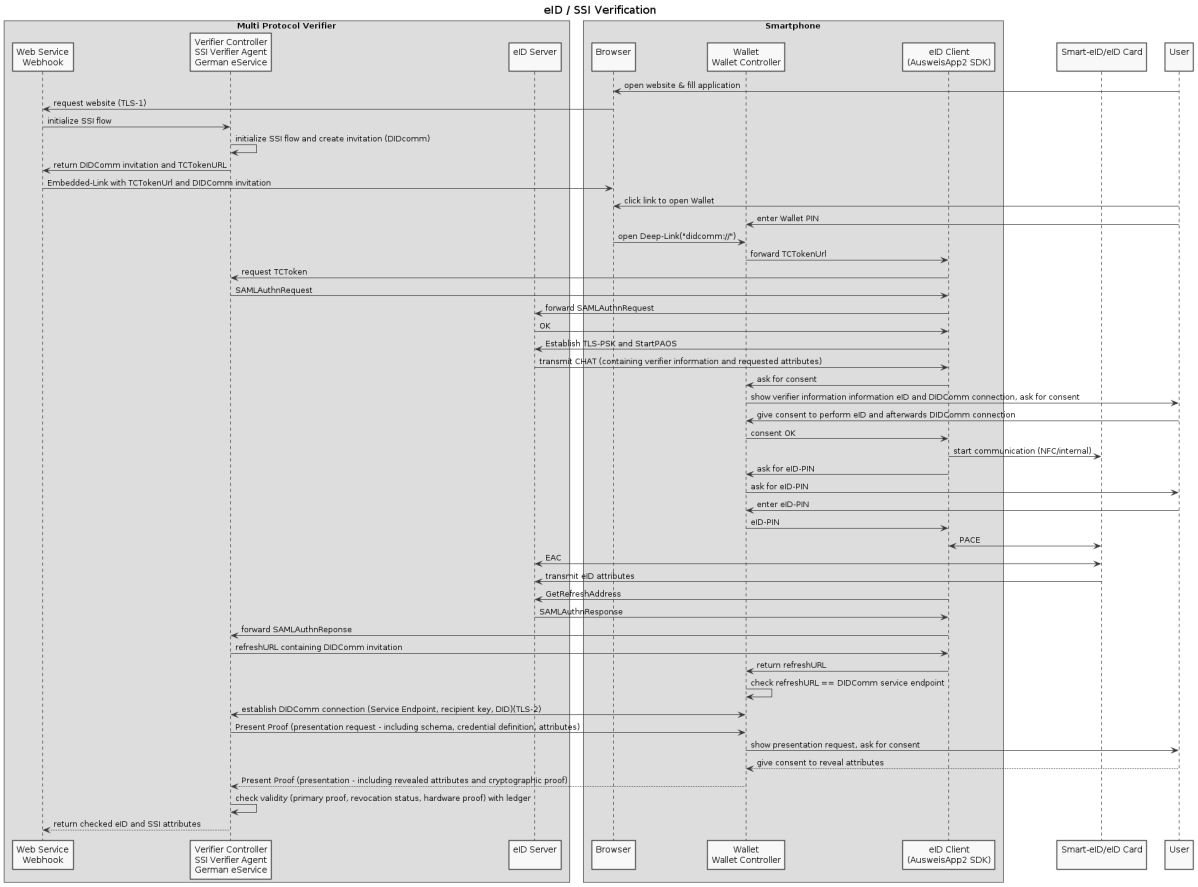


Das folgende Sequenzdiagramm beschreibt den vereinfachten Prozess zur Verifikation von SSI Credentials mit DIDComm Verbindungsaufbau, Nachweisanfrage, Präsentierung und Validierung. Die Prüfung der optionalen hardwareDid im Challenge-Response-Verfahren ist hier nicht dargestellt.



5.3.4. Gemeinsame Verifikation – Identifikation mit eID + Verifikation mit SSI

Das folgende Sequenzdiagramm zeigt beispielhaft anhand einer Bewerbung eines Nutzers bei einem Unternehmen die gemeinsame Verifikation von eID (Identität) und SSI-Credential (Zeugnis). Wir nutzen hier im dargestellten Sequenzdiagramm für den eID-Prozess beispielhaft den SAML Flow. Um die Nutzerinteraktion zu reduzieren, wurden hier bereits eID (TCtokenURL und SSI (DIDComm) beim Verbindungsaufbau und Verifier-Prüfung zusammengefasst.



Der beispielhafte Bewerbungsprozess wird von der Web Service des Verifizierers gestartet.

1. Der SSI Flow wird von dem Web Service gestartet.
2. Es erfolgen Vorbereitung eines kombinierten Verbindungsaufbaus (SSI-DIDComm und eID-TCtokenURL) und Erstellung eines Embedded Links.
3. Via QR-Code oder Deeplink wird die Wallet gestartet und der Aufbau eID-Kanal vorbereitet und die eID Attribute angefragt.
4. Der Verifizierer kommuniziert über den eService Controller und eID Server mit dem eID Client auf dem Smartphone des Nutzers.
5. Der Nutzer erhält kombinierte Informationen zu Verifizierer (eID und SSI) für Verbindungsaufbau zur Prüfung und wird um Zustimmung für den eID-Prozess und SSI-Verbindungsaufbau gebeten.
6. Nutzer gibt in einem Schritt gleichzeitig seine Zustimmung um den eID- Prozess und den SSI Verbindungaufbau auszuführen.
7. Über den eID Client kann die ID Wallet den Identifizierungsprozess über die (Smart-)eID starten.

8. Dazu wird die eID PIN des Nutzers zur Authentifizierung abgefragt
9. Der eID Client und der eID Server führen den EAC Prozess aus und die eID-Daten werden an den eService Controller übermittelt.
10. Abschließend wird über die mittels eID validierte TLS- Verbindung eine DIDComm Verbindung aufgebaut, um eine Verifkationsanfrage zu senden.

6. Offene Punkte

- Wie könnte die integrierte Nutzung der eId + SSI für die Nutzer ausgestaltet werden?
  - Design Team arbeitet am Design
- Wie könnte die integrierte Nutzung für den Entwickler ausschauen?
- Wie schaut eine Integration für Smart eId Funktionalität mit der Ausweis App aus?
  - Benötigt eine smart eID Integration einen Wechsel auf die AusweisApp2?

7. Referenzen

- **BSI**. Die Datenformate der auf dem Ausweis gespeicherten Daten (**BSI TR-03110**), https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr03110/TR-03110_node.html
- **BSI**. Vertrauensniveaubewertung von Verfahren zur Identitätsprüfung (**BSI TR-03147**), https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr03147/tr03147_node.html
- **BSI**. Vertrauensniveaubewertungen von elektronischen Identitäten und Vertrauensdiensten (**BSI TR-03107**). https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr03107/TR-03107_node.html
- **Systemkonzept zum Projekt 2080069021, "Digitale Identitäten: Pilotvorhaben Hotel Check-In"**, Version vom 28.04.2021, IBM Deutschland GmbH, Fraunhofer AISEC, Bundesdruckerei GmbH, esatus AG
- **Technische Beschreibung der Gerätebindung**, Version v6 vom 21.07.2021, IBM Deutschland GmbH Fraunhofer AISEC, Bundesdruckerei GmbH, esatus AG
- **Konzept Verifizierer Authentifizierung v1.1**: Dieses Dokument basiert auf den bereits beschriebenen Konzepten zur Erhöhung des Vertrauens in die anfragende Partei. Dabei gliedert sich die Smart-eID-Integration als entweder Stufe 2 oder sogar als neuerer Stufe 3 in die entsprechenden Ausbaustufen ein.
- **Ökosystem digitale Identitäten: Architekturzirkel Diskussionsunterlage** 01.12.2021 Beschluss zu Integrationsoptionen von Smart-eID und Basis-ID, Umsetzungskonzept von Nutzung Smart-eID und Basis-ID in einer Wallet
- **Bedrohungsanalyse eID-Ökosystem Authentisierung und** Entwurf der Prüfberichtsvorlage entsprechend TR-03107-01
- **Entscheidungsgrundlage zur Authentifizierung von Verifizier** https://app.mural.co/t/ssibk4158/m/ssibk4158/1638183384578/44405ab589eb3b20250af397ca2e9d806e85118a?sender=richardholzeis6419