D06-QM Architekturkonzept der Open-Source-Core

Editor:	Jens Kubieziel und Christian Kahlo (AGETO)
Prüfung:	Sven Wohlgemuth (TU Darmstadt/CASED)
Typ:	[TECHNISCHER BERICHT]
Projekt:	„PersoApp“
Version:	1.0
Datum:	28. Juni 2013
Status:	[FREIGABE]
Klasse:	[ÖFFENTLICHKEIT]
Datei:	D06-QM Architekturkonzept der Open-Source-Core.doc

Zusammenfassung

Das vorliegende Dokument beschreibt das Architekturkonzept der Open-Source-Core von „PersoApp“. Die Module und Schnittstellen werden beschrieben. Weiterhin ist eine Sicherheitsanalyse sowie Betrachtungen zur Benutzbarkeit und zum Schutz der Privatsphäre enthalten.

Konsortialleitung:
Prof. Dr. Ahmad-Reza Sadeghi und Dr. Sven Wohlgemuth
System Security Lab, TU Darmstadt/CASED, Mornewegstr. 32, 64293 Darmstadt
Tel.: +49-6151-16-75561 Fax: +49-6151-16-72135
E-Mail: persoapp@trust.cased.de Web: https://www.persoapp.de

Nutzungslizenz

Mitglieder des Konsortiums

1. **AGETO Service GmbH**, Deutschland
2. **Center for Advanced Security Research Darmstadt (CASED)**, Deutschland
3. **Fraunhofer Institut für Sichere Informationstechnologie (SIT)**, Deutschland
4. **Technische Universität (TU) Darmstadt**, Deutschland

Versionen

Version	*Datum*	*Beschreibung (Editor)*
0.1	2013-06-17	Initiale Version (AGETO)
0.2	2013-06-17	Kapitel von TU Darmstadt/CASED (AGETO)
0.3	2013-06-18	Architekturkonzept von AGETO (AGETO)
0.4	2013-06-23	Überarbeitung und Erweiterung des Architekturkonzepts (AGETO)
0.5	2013-06-25	Datenfluss-Diagramm überarbeitet (AGETO)
1.0	2013-06-28	Version für Freigabe (TU Darmstadt/CASED)

[1] http://creativecommons.org/licenses/by-sa/3.0/

Autoren

Autoren	*Beiträge*
Jens Kubieziel, Christian Kahlo (AGETO)	Initiale Version des Dokumentes erstellt, Architekturkonzept, Überarbeitung
Sven Wohlgemuth (TU Darmstadt/CASED)	Benutzbarkeit und Schutz der Privatsphäre
Siniša Đukanović (Fraunhofer SIT)	Sicherheitsanforderungen und Qualitätskriterien
Philipp Holzinger (Fraunhofer SIT)	Sicherheitsanforderungen und Qualitätskriterien
Stefan Triller (Fraunhofer SIT)	Sicherheitsanforderungen und Qualitätskriterien

Inhaltsverzeichnis

1 Zweck und Ziel des Dokumentes

Das Dokument „D06-QM Architekturkonzept der Open Source PersoApp“ beschreibt die Architektur der Software mit den Modulen und Schnittstellen. Weiterhin finden hier Betrachtungen zu den Sicherheitsanforderungen sowie zur Benutzbarkeit und dem Schutz der Privatsphäre statt. Das Architekturkonzept bildet die Grundlage für die weitere Entwicklung der Software.

2 Anwendungsbereich

Das Dokument „D06-QM Architekturkonzept der Open Source PersoApp“ ist für die Software-Entwicklung und das Release Management bestimmt. Es soll aktuellen und künftigen Entwicklern Aufschluss über den Aufbau und die Struktur der Software geben. Dies ermöglicht es, einen schnellen Einstieg in die Entwicklung zu erhalten.

3 Abkürzungen und Begriffsdefinitionen

3.1 Abkürzungen

Abkürzung	*Erläuterung*
API	Application Programming Interface
BSI	Bundesamt für Sicherheit in der Informationstechnik
CASED	Center for Advanced Security Research Darmstadt
CC	Creative Commons
CRM	Customer Relationship Management
eID	Elektronische IDentität
GUI	Graphical User Interface
HAL	Hardware Abstraction Layer
HCI	Human Computing Interface
HTTP	Hypertext Transfer Protocol
IFD	Interface Device
iOS	Apple iOS (iPhone OS)
JDK	Java Development Kit
JRE	Java Runtime Environment
nPA	neuer Personalausweis
OS	Operating System
P3P	Platform for Privacy Preferences Project
PACE	Password Authenticated Connection Establishment (Protokoll)

PAOS	Reversed HTTP Binding for SOAP
PC/SC	Personal Computer/Smart Card
PIN	Persönliche Identifikations-Nummer
PKI	Public-Key-Infrastruktur
QM	Qualitätsmanagement
SAL	Service Access Layer
SIT	Sichere Informationstechnologie (s. Fraunhofer SIT)
SOAP	Simple Object Access Protocol
TLS	Transport Layer Security
TR	Technische Richtlinie
TU	Technische Universität
WSDL	WebServices Description Language
XSD	XML Schema Definition
XML	eXtensible Markup Language

3.2 Begriffsdefinitionen

Begriff	Definition
Open-Source-Core	Open-Source-Software-Bibliothek mit den Modulen zur Implementierung der Online-Ausweisfunktion des neuen Personalausweises nach den Technischen Richtlinien des BSI

4 Zuständigkeiten und Verantwortlichkeiten

Das Dokument wurde in gemeinsamer Verantwortung der Projektpartner AGETO Service GmbH, Fraunhofer-Institut für Sichere Informationstechnologie (SIT) und Technische Universität Darmstadt/CASED (TU Darmstadt/CASED) entwickelt.

Hinweis: Wesentliche Teile der Architektur bauen auf den Technischen Richtlinien des BSI, insbesondere auf BSI TR-03112 [21], auf. Es zeichnet sich ab, dass Teile dieser Richtlinie in der nächsten Zukunft auf die TR 03116-4 und TR 03124-1 aufgeteilt werden. Durch diese Änderungen können daher Anpassungen an der hier beschriebenen Architektur notwendig werden.

5 Architekturkonzept

5.1 Implementierungssprache

Die „PersoApp“-Software wird vollständig in der Programmiersprache Java implementiert. Alle Entwicklungen sollen, unabhängig von der verwendeten Java-Version, kompatibel zur Version 1.6 (J2SE6) des Sun/Oracle bzw. OpenJDK sein. Das Java Development Kit (JDK) bezeichnet dabei die Java-Entwicklungsumgebung, und das

Java Runtime Environment (JRE) ist die zugehörige Java-Laufzeitumgebung inklusive der Java Virtual Machine (JVM).

Durch die Festlegung auf die Sprache Java erhält man eine plattformunabhängige Applikation, die ohne Anpassungen auf einer Vielzahl von Endgeräten und Betriebssystemen lauffähig ist. Für PC-Systeme ergibt sich damit eine Plattform-Bandbreite von Windows 2000 32-Bit bis Windows 8 64-Bit, sowie die meisten 32- und 64-Bit Linux-Systeme, Mac OS X ab Version 10.6 sowie einige BSD-Varianten.

Unterschiede sind beispielsweise die Hardware Schnittstellen der verschiedenen Plattformen, etwa PC-Systeme und Android-Systeme („javax.smartcardio.*“ bzw. „android.nfc.*“, USB CCID über „android.hardware.usb.*“ und Open Mobile API).

Einzig Apple unterstützt für deren Mobilgeräte (iPhone, iPad, etc.) kein Java. Mit J2ObjC[2] existiert jedoch eine Software, die Java-Code in Objective C umwandeln kann. Die Programmiersprache Objective C wird für Programme unter dem Betriebssystem iOS verwendet. Damit lässt sich gegebenenfalls ohne vollständige Neuentwicklung eine Unterstützung für die mobilen Apple-Produkte erzeugen. Zur Zeit existieren weder eine NFC-Schnittstelle noch die Option USB-Lesegeräte aus iOS heraus anzusteuern. Mit Unterstützung durch Apple wäre letzteres ggf. über das Apple Camera Connection Kit möglich.

5.2 Modulübersicht

Die Architektur der „PersoApp“-Open-Source-Software-Bibliothek gliedert sich in eine modulare Grundstruktur. Die drei Hauptkomponenten sind dabei der protokolltechnische Kern, die Benutzerschnittstelle (GUI) sowie der Hardware Abstraction Layer (HAL) für die Ansteuerung des Lesegerätes bzw. der NFC-Schnittstelle. Erst hinter dem HAL befindet sich die Smartcard in Form des neuen Personalausweises als ein logisches Protokoll-Endstück.

Auf diese Weise kann der Kern unabhängig vom Rest des Systems wiederverwendet werden. Für eine Portierung von PC-System auf Android-System ist z.B. die Android-GUI als Gerüst sowie der HAL zu re-implementieren. So können Änderungen in den technischen Details der Schnittstellen zügig portiert werden.

Als einfacher Überblick über den Kern soll die folgende Grafik dienen. Die Pfeile zeigen grob Beziehungen zwischen den Komponenten auf während je Kästchen größere funktionale Einheiten gruppiert sind.

[2] https://code.google.com/p/j2objc/

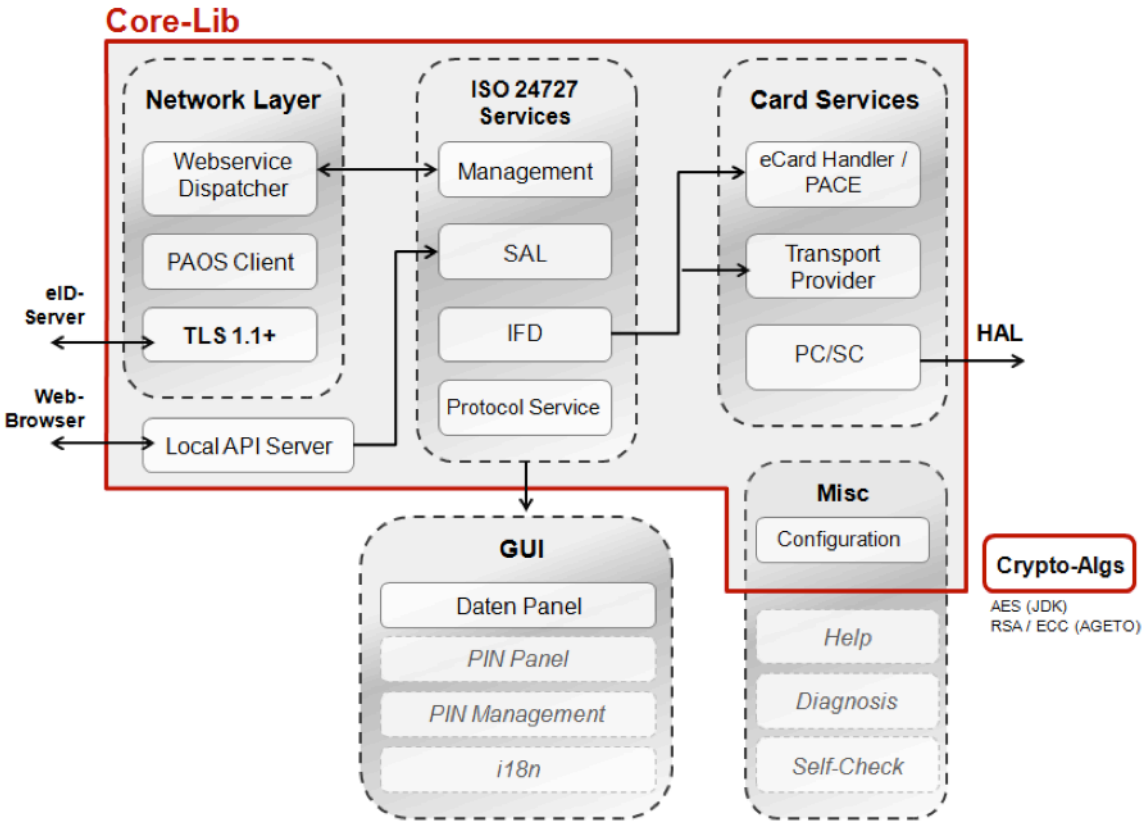

Abbildung 1: Überblick über die Architektur des Kerns der PersoApp

Der Kern der „PersoApp“-Open-Source-Software-Bibliothek besteht somit aus der Netzwerkschicht, den Schnittstellen (Java Interfaces) sowie den Implementierungen von Web-Services nach ISO-24727-3 für die Umsetzung der in BSI-TR 03112 geforderten Schnittstellen, den grundlegenden Funktionen für die Ansteuerung des Ausweises, Hilfsfunktionen und einer Anbindung von kryptographischen Funktionen.

Das BSI stellt auf seiner Webseite eine Zip-Datei mit Dateien in den XML-Beschreibungssprachen WSDL und XSD bereit mit deren Hilfe sich sowohl die Interfaces und die Gerüste für die Implementierung der Services als auch die Datenklassen generieren lassen. Die eigentliche Logik benutzt ausschließlich diese Klassen zur Kommunikation über die Web-Services. Dieser Prozess orientiert sich an den Vorgaben der modellgetriebenen Software-Entwicklung.

Die Netzwerkschicht regelt einheitlich die Kommunikation der PersoApp nach außen, d.h. zum Dienstanbieter und zum eID-Server. Entsprechend der BSI TR ist hier die Verwendung von mindestens TLS1.1 vorgeschrieben und weiterhin müssen jeweils die Serverzertifikate während der Kommunikation erfasst werden, um gegen das Berechtigungszertifikat des Dienstanbieters geprüft werden zu können. Für die Kommunikation mit dem eID-Server wird weiterhin eine als „PAOS-Client“ zusammengefasste Funktionalität eingesetzt in der HTTP und SOAP/PAOS umgesetzt werden. Da sich die „PersoApp“-Open-Source-Software-Bibliothek als netzwerkseitiger Client wie der logische (WebService-)Server verhält und der eID-Server entsprechend umgedreht als ein Client, ist eine Übersetzung notwendig, so dass die ISO-24727 Services dem eID-Server zur Ansteuerung bereit gestellt werden können. Kontrolliert wird der Prozess über einen „Worker“, der Abbruch, Timeout, Fehlersituation oder Erfolg des

Vorgangs erfasst und den anderen Komponenten mitteilt. Somit kann der Nutzer entsprechend informiert werden, die Smartcard zurückgesetzt und das Lesegerät freigegeben werden.

Speziell für PC-Systeme existiert zusätzlich ein „Local API Server“. Dabei handelt es sich um einen eingebetteten Web-Server aus der Java-Runtime, welcher über die IP-Adresse 127.0.0.1 (localhost) und den TCP Port 24727 einen HTTP-Dienst bereitstellt. Mit diesem integrierten, nur lokal verfügbaren, Server wird die Brücke zum Web-Browser geschlagen.

Unter Android und eingebetteten Systemen wird diese Funktionalität über „Intents“ oder andere, spezifischere Verfahren abgebildet.

Für die kryptographischen Funktionen wird für das Pre-Release von „PersoApp“ noch eine AGETO-eigene Implementierung verwendet während für die Open-Source-Software-Bibliothek von „PersoApp“ auf eine andere noch auszuwählende Open-Source-Bibliothek zurückgegriffen wird.

5.3 Modulbeschreibungen

In diesem und dem nachfolgenden Abschnitt sollen die einzelnen Komponenten und deren Interaktion sowie die Schnittstellen ausführlicher beleuchtet werden. Die folgenden Entitäten agieren dabei mit den Modulen der „PersoApp“ und sind von außen gegeben, d.h. aus dem Betrachtungswinkel der Module nicht veränderbar.

Diensteanbieter	Der Diensteanbieter ist der “Konsument” der eID-Daten. Beispiele für Diensteanbieter sind Online-Shops, Banken, E-Mail-Anbieter, Instanzen der kommunalen und staatlichen Verwaltung, Versicherer, etc.
Web-Browser	Ein auf dem System des Anwenders installierter Web-Browser, wie z.B. Chrome, Safari, Internet Explorer, Firefox und Opera. Im jetzigen Szenario gibt es keine Plugin- und Plattform-Abhängigkeiten mehr.
Anwender	Im Falle einer nationalen elektronischen Identitätsfunktion ist der Anwender immer der Bürger und damit auch der Inhaber der ID-Karte / des Personalausweises.

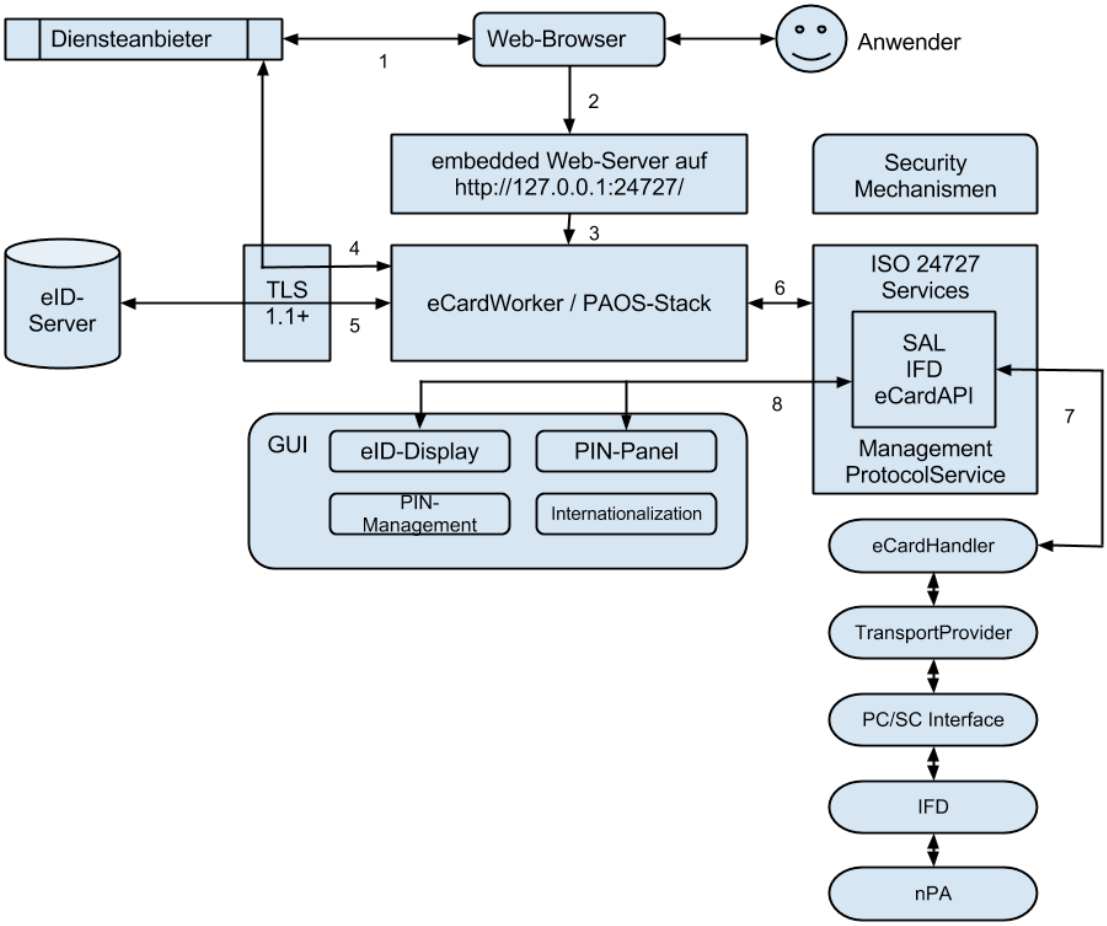

Abbildung 2: Module im Kern der PersoApp und Außenkomponenten

Die in Abbildung 2 dargestellten Module sollen in ihrer Funktionalität und soweit verfügbar in ihrer Normierungsgrundlage wie folgt verstanden werden:.

Eingebetter Web-Server

Es handelt sich hierbei um einen minimalistischen HTTP/1.1-fähigen Web-Server innerhalb des eID-Clients. Aktuell kommt der Sun „HttpServer“ zum Einsatz. Eine ausführlichere Beschreibung von Funktion und Aufbau findet sich u.a. in einem Oracle Mitarbeiter-Blog unter https://blogs.oracle.com/michaelmcm/entry/http_server_api_in_java.
Andere – zum Beispiel eigene – Varianten sind aber durchaus ebenso möglich. Unter Android wird die Funktion des eingebetteten Web-Servers durch den Intent-Mechanismus abgedeckt. Das Verhalten ist in [21] Kapitel 3.2 genormt.

„eCardWorker“ und PAOS-Stack

Der „eCardWorker“ ist keine genormte Instanz und – obwohl im proprietären AGETO eID-Client und somit im Pre-Release präsent – eher als theoretisches Element zu verstehen. Die Aufgabe des „eCardWorkers“ ist es über den eingebetteten Web-Server herein kommende eID-Anfragen zu bearbeiten, d.h.

den “alternativen Aufruf” aus [21] Kapitel 3 (ohne Kapitel 3.5) durchzuführen und anschließend den PAOS-Stack zur Kommunikation mit dem Ziel-eID-Server aufzufordern. Dabei wartet der eCardWorker auf Abbruch, Fehler, Timeout oder Erfolg des Vorgangs und meldet dies an die GUI und mittels dem eingebetteten Web-Server auch an den Browser weiter. Im Android-Fall wird der Web-Browser ebenso über einen Intent angesteuert. Der Ablauf entspricht dabei [21] Kapitel 3.4, Bild 5.

Der PAOS-Stack besteht faktisch aus mehreren Komponenten und funktioniert dabei sowohl als SOAP- und HTTP-Client gegenüber dem eID-Server als auch als PAOS-Server und vermittelt die Nachrichten zwischen eID-Server und den integrierten ISO24727 WebServices. Zum Vergleich dient [21] Kapitel 2.3, Bild 2, „PAOS Binding“ im Besonderen der Bildabschnitt „Application Logic“.

TLS 1.1+

Diese Komponente bezeichnet einen TLS-Protokollstack, welcher für die verschlüsselte Kommunikation zwischen eID-Client und Diensteanbieter-System sowie dem eID-Server verantwortlich ist. Bei letzterem ist zu beachten, dass TLS-RSA-PSK mit mind. AES-128 unterstützt werden muss. Damit scheiden die Standard-Bibliotheken im Regelfall aus.

eID-Server

Der eID-Server ist eine nach [9] definierte Komponente der eID-Infrastruktur, die entweder direkt vom Diensteanbieter betrieben werden kann oder aber auch als zukaufbarer Service bereitgestellt werden kann. Der eID-Server verwaltet das Berechtigungszertifikat des Diensteanbieters und steht in Verbindung mit der DVCA (Document Verifier Certification Authority) zur regelmäßigen Aktualisierung des Berechtigungszertifikats und der Sperrlisten.

ISO 24727 Services

In BSI-TR 03112-3, -4 und -6 werden generisch die von der „PersoApp“-Open-Source-Software-Bibliothek unterstützten Dienste eines eID-Clients definiert, dazu gehören das „Management Interface“ (-3), der „Service Access Layer“ (SAL, -4 und ISO 24727-3), das „InterFace Device“ (IFD, -6) Interface, der „Protocol Service“ (-4 und ISO24727-3) und spezifisch für den nPA bei Unterstützung der QES-Funktion der „eCard-API Service“ (BSI TR-03112-2). Da keine QES-Unterstützung für die „PersoApp“-Open-Source-Software-Bibliothek vorgesehen ist, ist die Nennung des „eCard-API Service“ nur der Vollständigkeit halber anzusehen.

eCardHandler

Der eCardHandler stellt eine abstrakte Schnittstelle zu den Funktionen der eID-Karte dar. U.a. wird beim Einsatz eines Basislesers das PACE-Protokoll (Password Authenticated Connection Establishment) aber auch weite Teile der Terminal- und Chip-Authentication hier direkt abgebildet. Der eCardHandler ersetzt aus Effizienzgründen den „Generic Card Access Layer“ aus BSI TR-03112-4 Kapitel 4 und ISO24727-2, da zwar sogenannte „CardInfo Files“ für den neuen Personalausweis existieren, diese sich aber auf frühe Versionen beziehen und fürs Erste lediglich eine einzige Karte, der deutsche neue

Personalausweis in der Form ab 01.11.2010, unterstützt werden soll. Bei Bedarf für Erweiterungen auf andere, z.B. neuere, Kartenmodelle ist der eCard-Handler eine Zwischenstufe vor dem Generic Card Access Layer, welcher abstrakte Operationen auf konkrete Kommandos und Parameter für die Karte übersetzt.

TransportProvider

Da nicht nur die PC/SC-Schnittstelle eine Möglichkeit darstellt auf Lesegeräte (IFDs) zuzugreifen wurde der „TransportProvider“ eingeführt. Darüber werden die Fähigkeiten der Lesegeräte abstrakt abgebildet. So könnte ein Akkubetriebener Komfortleser mit einem passenden „TransportProvider“ auch per Bluetooth oder W-LAN angesteuert werden. Die Android-Version des proprietären AGETO eID-Clients kennt für den Transportprovider Implementierungen auf PC/SC, USB CCID, experimentell OpenMobile API sowie NFC und wählt unter diesen automatisch aus. Für die „PersoApp“-Open-Source-Software-Bibliothek soll in Zukunft eine ähnliche Möglichkeit bestehen. Aus Sicht eines hypothetischen „Generic Card Access Layer“ bildet der „TransportProvider“ den Übergang in Hardware-nahe Schichten.

PC/SC-Interface

Im klassischen Umfeld der PC-Systeme wird das von der PC/SC-Workgroup standardisierte Personal-Computer/Smartcard Interface der JVM unter „javax.smartcardio.*“ angesteuert. Diese Komponente ist Bestandteil der jeweiligen JVM (Sun/Oracle, OpenJDK) und ist betriebssystemspezifisch. Die JVM stellt dabei selbst eine Java Native Interface (JNI, in der Regel „j2pcsc“) Bibliothek passend zur jeweiligen Betriebssystemarchitektur bereit worüber dann der PC/SC-Smart Card Daemon (Unixoide) bzw. WinScard/SCardSvr (Windows) angesteuert wird. Hierbei ist zu berücksichtigen, dass nicht alle PC-Betriebssysteme die Kommunikation zwischen Bibliotheken unterschiedlicher Architektur (32-Bit vs. 64-Bit) erlauben. So dass ein 64-Bit System mit 64-Bit PC/SC Daemon durchaus Probleme bereiten kann wenn die JRE des eID-Client 32-bittig ausgeführt ist. Im Allgemeinen stellen diese Situationen durch Angleichung der Distributionen zukünftig jedoch mehr und mehr Ausnahmen dar.

IFD

Der Begriff „Interface Device“ bezeichnet den – ggf. kontaktlosen – Smartcard-Leser für die Kommunikation mit der eID-Karte. Mögliche Varianten sind Basisleser, Standardleser und Komfortleser. Dabei ist der Basisleser nur die kontaktlose Kommunikationseinheit, während der Standardleser über eigenes Display und PIN-Pad für die sichere Anzeige des Inhabers des Berechtigungszertifikates sowie PIN-Eingabe verfügt. Der Komfortleser besitzt zusätzlich ein Security-Modul um die Verwendung der qualifizierten elektronischen Signatur abzusichern. Zum Vergleich zulässiger Geräte und deren Eigenschaften sei an dieser Stelle auf BSI TR-03119 verwiesen.

nPA

Die eID-Karte besteht in Form des neuen Personalausweises [10]. Die Eigenschaften, Strukturen und Kommandos der Smartcard sind in BSI-TR 03110 Teil 1 bis 3 festgeschrieben.

GUI

Das „Graphical User Interface“ (GUI) stellt die sichtbare Komponente im eID-Client dar. Hier erfolgt die Darstellung der Daten des Dienstanbieters im Kommunikationsprozess sowie die Auswahlmöglichkeiten des Anwenders für optionale Datenfelder (eID-Display) und im Falle eines Basislesers die Abfrage der PIN in einem PIN-Panel. Steht dem Anwender ein Standard- oder Komfortleser zur Verfügung blendet die GUI das PIN-Panel aus und fordert den Anwender auf den Anweisungen auf dem Display des Lesegeräts zu folgen.

Zusatzfunktionen der GUI sind das PIN-Management und die Internationalisierung. Für das PIN-Management ist keine Internet-Verbindung notwendig und der Prozess ist gegenüber der Online-Authentisierung gedreht, so dass die GUI über Events auf Funktionen des Kerns zurückgreift, um zu ermitteln ob z.B. ein Basis- oder Komfortleser vorhanden ist, ein Ausweis aufgelegt ist, wie viele PIN-Versuche verbleiben und schlussendlich eine PIN-Änderung, Aktivierung, PIN-Entsperrung zu veranlassen.

Die Internationalisierung ist im Wesentlichen eine Sache der GUI für die technisch definierten Felder entsprechend zur Spracheinstellung des Betriebssystems des Anwenders Regionsspezifische Entsprechungen darzustellen. Für Anwendernachrichten direkt aus dem Kern gibt es einen ähnlichen Mechanismus mit dem – sofern verfügbar – Detailnachrichten in unterschiedlichen Sprachen ausgegeben werden können.

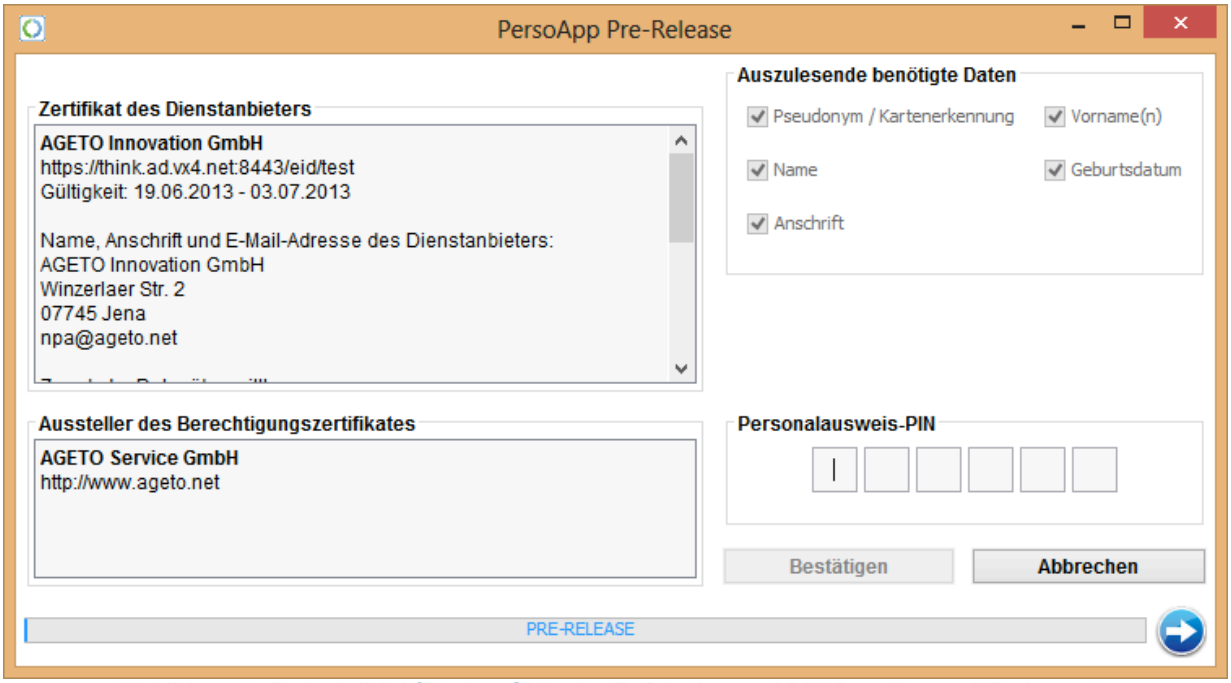

Abbildung 3: Beispielbild für eine GUI mit eID-Display und PIN-Panel unter Windows 8

„Security Mechanismen“

Unter „Security Mechanismen“ werden sämtliche Grundfunktionen für Ver- und Entschlüsselung, Schlüsselaushandlung und Authentifizierung verstanden. Für die PersoApp von großer Rolle sind die Verfahren AES, SHA256, ECDH, ECDSA, RSA und für die Kompatibilität auf Netzwerkebene zum Dienstanbieter derzeit noch 3-DES-CBC. Die Verfahren kommen an vielen

Stellen innerhalb der PersoApp zum Einsatz. Bereitgestellt werden die Algorithmen entweder durch die JRE, schlanke Eigenentwicklungen oder bereits existierende anderen Open-Source Software-Bibliotheken.

5.4 Schnittstellen der Module

Die Schnittstellen der Module werden in Softwareschnittstellen (Language-Binding), Netzwerkschnittstellen (HTTP) und Hardwareschnittstellen unterschieden.

Die folgenden Softwareschnittstellen existieren innerhalb des Pre-Release und spiegeln die Basis für die Weiterentwicklung der PersoApp wieder.

Klasse	Typ	Beschreibung
Main	Runnable	Initialisierung des Kerns, d.h. der ISO 24727 Services, Netzwerk- und Hardwareschnittstellen sowie grundlegender Datenstrukturen. Von der Umgebung muss eine Implementierung von „IMainView“ übergeben werden.
IMainView	Interface / Object	Wird von der GUI implementiert um dem Kern einen einheitlichen abgeschotteten Kommunikationsweg zur GUI zu ermöglichen. Der Kern registriert an der Implementierung von „IMainView“ einen „EventListener“ um auf Rückmeldungen der GUI (Schließen der Applikation, Abbruch von Vorgängen, PIN-Management, etc.) reagieren zu können.
IMainView. EventListener	Interface / Object	Der Kern stellt einen „IMainView.EventListener“ bereit mit dem Ereignisse von der Anwenderseite mitgeteilt werden können.
IEAC_Info	Interface / Object	Eine Implementierung von „IEAC_Info“ enthält bei der Online-Authentisierung alle für den Nutzer darzustellenden Informationen über den eID-Vorgang.
SecureHolder	Object	Bei einer PIN-Eingabe durch das PIN-Panel wird über eine vom Kern bereitgestellte Instanz „SecureHolder“ die PIN von der GUI zum Kern übertragen. Auch Abbruch oder Fortsetzung des Vorgangs werden hier signalisiert.

Die Kommunikation auf Netzwerkseite übernehmen die nachstehenden Netzwerkschnittstellen.

Name	Art	Beschreibung
embedded	In-bound	Der Start der Online-Authentisierung nach [21] Kapitel

Web-Server / Intent		3.4 erfolgt entweder über einen GET-Request auf den integrierten Web-Server auf 127.0.0.1:24727 oder über einen „Intent“. Dabei wird der Parameter „tcTokenURL“ an den eID-Client übertragen. Die Antwort ist in der Regel ein HTTP Redirect auf eine entsprechende Seite des Dienstanbietersystems. In seltenen Fehlerfällen (Fehlkonfigurationen, Systemausfälle bei eID-Servern, u.ä.) erfolgt eine Text-Rückmeldung welche im Browser behandelt wird.
eCardWorker / TCToken-Handler	Out-bound	Die „tcTokenURL“ wird durch einen „TCTokenHandler“ (Bestandteil des „ECApiHttpHandlers“ oder „Intent-Handlers“) beim Dienstanbietersystem abgeholt dabei kommt der interne TLS1.1-Stack zum Einsatz und das Zertifikat der Gegenstelle wird zwecks späterer Prüfung erfasst und hinterlegt.
PAOS-Stack	Out-bound	Bestehend aus integriertem HTTP-Client, SOAP-Client, PAOS-Initiator und -Dispatcher wird hier ebenfalls über den TLS1.1-Stack eine TLS-RSA-PSK Verbindung zum eID-Server aufgebaut. Analog zum Dienstanbietersystem wird auch das TLS-Zertifikat des eID-Servers zur späteren Prüfung erfasst und hinterlegt. Nach dem initialen StartPAOS-Request werden die als „Antworten“ zurückgelieferten Requests an die Web Services verteilt und deren Antworten wiederum als Requests verpackt an den eID-Server geliefert.

Die Hardwareschnittstellen sind nicht in allen Implementierungen vollständig notwendig. Für PC-Systeme ist lediglich die javax.smartcardio.* PC/SC Schnittstelle vorhanden.

Name	Typ	Beschreibung
JSCIO-Transport	PC/SC	Abstraktion der „javax.smartcardio.*“-Schnittstelle der JRE auf einem PC-System. Es wird generisch das Suchen eines bestimmten Kartentyps auf allen angeschlossenen Lesegeräten unterstützt. Das Monitoring in Form eines regelmäßigen Scans aller Leser wird z.Zt. absichtlich nicht eingesetzt, um Probleme durch abgestürzte Lesertreiber, Kollision mit anderer Software auf dem Anwender-PC und ggf. blockierende Threads zu vermeiden.
Nfc-Transport	Android NFC	Stellt basierend auf „android.nfc.*“ den Zugriff auf den integrierten ISO14443-fähigen NFC-Controller her. Vo-

		raussetzung ist die Unterstützung von „extendend length APDUs“ (Application Protocol Data Units) nach ISO7816-4:2005.
UsbCCID-Transport	Android USB	Beginnend mit Android 3.1/3.2 besteht die Möglichkeit des Zugriffs auf USB-Geräte im USB Host-Mode. Der UsbCCID Transport bindet eine Reihe CCID-konformer Lesegeräte an ohne dabei Plattformspezifischen Code zu benötigen. D.h. es ist eine Unabhängigkeit zur CPU-Architektur (ARMv7/v9 und ATOM/x86) gewährleistet.
ISOSM-Transport	Logik	Logischer Transport zur transparenten Unterstützung von „ISO7816-4 Secure Messaging“ für die sichere Kommunikation mit der Smartcard nach Aufbau des PACE-Tunnels.
Transport-Provider	Interface	Abstraktes, generisches Interface über alle Transport-Varianten zur Vereinheitlichung der Kommunikation zu oberen Schichten. Die Eigenschaften der tatsächlichen Hardware können abgefragt werden, z.B. CCID-konform das Vorhandensein eines Komfortlesers an einer PC/SC oder USB-CCID Schnittstelle.

Für die verständliche Erläuterung der Schnittstellen der „PersoApp“-Open-Source-Software-Bibliothek soll der Ablauf einer Online-Authentisierung aus Abbildung 2 dargestellt werden. Aus Prozesssicht sei auf [21] Kapitel 3.4, Bild 5 verwiesen.

Schritt	Beschreibung
1	Der Anwender besucht die Webseite eines Dienstanbieters und wählt die Authentisierung mittels Online-Funktion aus. Dabei wird im Browser ein Button oder Link bereitgestellt welcher auf das Ziel http://127.0.0.1:24727/eID-Client?tcTokenURL=https://<diensteanbieter-adresse..> zeigt. Die IP-Adresse 127.0.0.1 bezeichnet dabei immer die eigene Maschine des Anwenders, es wird immer HTTP verwendet und es wird immer der Port 24727 angesprochen. Auch der Pfad “eID-Client” und der Parameter “tcTokenURL” und die Pflicht einer https-URL als Wert sind in [21] festgeschrieben.
2	Klickt der Anwender auf diesen Link stellt der Web-Browser eine Anfrage an den lokalen eID-Client. Dabei wird der Request als “GET /eID-Client?tcTokenURL=https%3A%2F%2F HTTP/1.1” abgesetzt. Auf diesem Weg erhält der eID-Client die sog. „tcTokenURL“. Zum Vergleich [21] Kapitel 3.2.
3	Der „eCardWorker“ prüft die „tcTokenURL“ auf Korrektheit und belässt den Web-Browser in einer Warteschleife.

4	Der „eCardWorker“ baut eine TLS 1.1+-Verbindung zum Diensteanbieter-System auf und versucht das „TCToken“ zu erhalten. Siehe auch [21] 3.3. Dabei wird zusätzlich das Server-Zertifikat des Diensteanbieter-Systems erfasst.
5	Das „TCToken“ enthält die Adresse und die Kommunikationsparameter für den eID-Server. Der PAOS-Stack baut die Verbindung zum eID-Server auf und vermittelt in einer Schleife die herein kommenden Anfragen des eID-Servers an die ISO 24727 Services.
6	Über einen internen, von außen nicht erreichbaren, Web-Service Container (WSContainer) werden die ISO 24727 Service im JAX-WS 2.0 Stil gebunden und initialisiert. Der PAOS-Stack vermittelt über den „WSContainer“ zwischen eID-Server und den an Hand der „WSDLs“ und „XSDs“ aus BSI TR-03112 definierten Services.
7	„SAL“ und „IFD“ greifen für die Erkennung von Lesegeräten und das Auffinden einer eID-Karte auf den „eCardHandler“ zu. In der „PersoApp“-Open-Source-Software-Bibliothek gibt es nur einen „eCard-Handler“ für den nPA. Im Desktop-Umfeld kennt der „eCardHandler“ nur einen TransportProvider (JSCIOTransport) welcher „javax.smartcardio.*“ ansteuert, um die betriebssystemspezifische PC/SC-Implementierung zu erreichen. Der PC/SC-Stack des Systems verfügt über die notwendigen Treiber um ein angeschlossenes „IFD“ ansteuern zu können und liefert über „PC/SC IOCTLs (Input Output Controls)“ die Eigenschaften des „IFD“ zurück und leitet Kommandos für den nPA zum „IFD“. Das „IFD“ stellt in Form des Lesegeräte die nächstgelegene externe Einheit mit Interaktion zum Benutzer dar. Über das „IFD“ wird die Verbindung zur eID-Karte/zum nPA aufgebaut. Evtl. notwendige low-level Protokollaushandlungen werden meist hier durchgeführt. In der Regel wird über eine LED eine erkannte Karte signalisiert. Der nPA (eID-Karte) stellt das „Target“ dieses Kommunikationskanals dar und wird während der Online-Authentisierung vom eID-Server über die Kanäle 5, 6 und 7 mit Kommandos versorgt. Ausgelesene Daten gehen über 7, 6 und 5 an den eID-Server zurück.
8	Zur Signalisierung einer aktiven eID-Anfrage sendet der „SAL“ ein Event über ein zentrales, einheitliches aber nicht genormtes Interface ***IMainView***. Daraufhin wird z.B. in einer Task-Tray „Sprechblase“ der Nutzer über einen Verbindungsversuch in Kenntnis gesetzt. Ist der Versuch auch erfolgreich erfolgt die Übermittelung der Berechtigungszertifikatsdaten in einer Struktur ***EAC_Info*** woraus die GUI alle notwendigen Angaben über den eID-Vorgang entnehmen und darstellen kann. Der „SAL“ erkennt automatisch ob ein nPA vorhanden ist und signalisiert ggf. die Aufforderung zum Auflegen des nPA an die GUI woraufhin ein entsprechender Dialog erscheint. Wird der Dialog bestätigt und es kann ein nPA gefunden werden, wird der GUI signalisiert, ob Sie mit oder ohne PIN-Feld erscheinen soll. Nur für den Basisleser

	erfolgt die PIN-Eingabe an der Tastatur des Desktops. Die PIN wird daraufhin sind 6 einzelnen Eingabefeldern Stelle für Stelle erfasst und in einer besonderen Struktur ***SecureHolder*** an den „SAL“ zurückgeliefert. Der „SAL“ veranlasst daraufhin über den „eCardHandler“ die Durchführung des PACE-Protokolls. Wobei die PIN nach dem ersten Schritt der Key-Derivation im RAM des Desktops vernichtet wird.

5.5 Sicherheitsanforderungen und Qualitätskriterien

Vorhergehend wurde das Architekturkonzept der „PersoApp“-Open-Source-Software-Bibliothek ausführlich dargestellt und erläutert. Dieses Unterkapitel widmet sich darauf aufbauend der Herleitung spezifischer Sicherheitsanforderungen und Qualitätskriterien, die sich unmittelbar durch die Systemkonzeption ergeben. Dies geschieht unter Berücksichtigung der Prozessbeschreibungen in „D03-QM Entwurfs- und Entwicklungsprozess von sicheren Open-Source-Softwaremodulen der ‚PersoApp‘“, sowie der „Checkliste zur Sicherheitsarchitektur von Softwarereleases“ in „D08-QM-2 Operative Planung und Durchführung von Reviews und Release-Updates“. Zukünftige Änderungen der Software können potentiell Anpassungen der Sicherheitsanforderungen und Qualitätskriterien bedürfen, eine wiederkehrende Überprüfung und Überarbeitung der Teilergebnisse ist daher notwendig.

An dieser Stelle ist jedoch zu beachten, dass sich zu dieser frühen Projektphase nur grundsätzliche Aussagen treffen lassen. Die aufgeführte Prozessbeschreibung und Checkliste kann daher nur zu Teilen bearbeitet werden, liefert jedoch für die zukünftigen Software-Releases eine fundamentale Grundlage. Im weiteren Projektverlauf gilt es die Vorlagen stetig weiter abzuarbeiten und zu überarbeiten, um zu einer vollständigen Sicherheitsarchitekturdokumentation zu kommen. Da die Sicherheitsarchitekturdokumentation, genauso wie der Quellcode selbst, einer internationalen Community zur Verfügung gestellt werden soll, wurde die „Checkliste zur Sicherheitsarchitektur von Softwarereleases“ aus „D08-QM-2 Operative Planung und Durchführung von Reviews und Release-Updates“ in englischer Sprache verfasst. Da die folgenden Abschnitte die Grundlage für die zu veröffentlichende Sicherheitsarchitekturdokumentation bilden, sind sie ebenfalls in englischer Sprache formuliert.

5.5.1 Allgemeine Eigenschaften der Software

Key properties	
Name	PersoApp
Version and Release	Unreleased
Application Domain	Desktop application with networking capabilities and hardware access
Purpose	Providing eID-functionality for end users in order to allow for secure authentication within web environments

Architecture overview	PersoApp-software serves as a connecting link between online service providers (such as online shops, administrative bodies, etc.), eID-service-providers, and end users

5.5.2 Datenfluss-Diagramm auf Kontextebene

Die nachfolgende Abbildung zeigt den Anwendungskontext in Form eines Datenfluss-Diagramms. Datenfluss-Diagramme (DFD) sind ein bewährtes Mittel um Datenflüsse von IT-Systemen abzubilden. DFDs werden zu vielerlei Zwecken eingesetzt, darunter auch die IT-Sicherheitsanalyse. Dabei können DFDs einen generellen Überblick über ein IT-System geben, aber auch dediziert zum Analysieren von Bedrohungen für IT-Systeme verwendet werden. So bilden DFDs, zum Beispiel, die Grundlage für die Bedrohungsmodellierung nach STRIDE.[3] Andere Verfahren zur Bedrohungsmodellierung basieren zwar nicht auf DFDs, können diese aber dennoch verwenden.

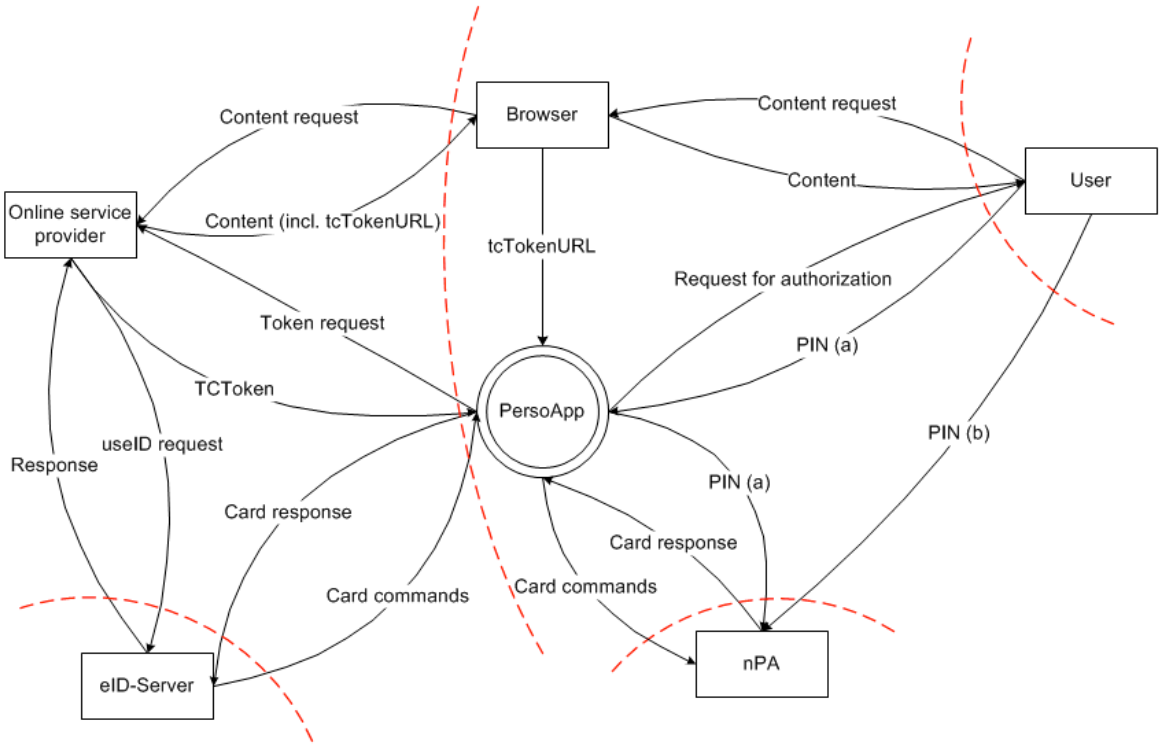

Abbildung 4 - Datenfluss-Diagramm auf Kontextebene

Ablaufbeschreibung:

1. Der Benutzer („User“) verwendet einen Browser um die Inhalte eines Dienstanbieters („Online service provider“) abzufragen.
2. Der Dienstanbieter übermittelt die angefragten Inhalte an den Browser, dieser stellt sie für den Benutzer dar. Dies umfasst auch eine sog. „tcTokenURL“.

[3] http://msdn.microsoft.com/en-us/library/ee823878%28v=cs.20%29.aspx

3. Die „tcTokenURL“ wird vom Browser an die „PersoApp“-Open-Source-Software-Bibliothek übermittelt. Diese ruft daraufhin das sog. „TCToken“ vom Dienstanbieter ab.
4. Unter Verwendung der Daten des „TCToken“ baut die „PersoApp“-Open-Sourcr-Software-Bibliothek eine Verbindung zum eID-Server auf.
5. Die GUI der „PersoApp“-Open-Sourcr-Software-Bibliothek zeigt dem Benutzer an, welche persönlichen Daten der Dienstanbieter über den eID-Server angefordert.
6. Der Benutzer gibt die Weitergabe seiner persönlichen Daten durch die Eingabe seiner geheimen PIN frei. Dies erfolgt, in Abhängigkeit der verwendeten Hardware, entweder direkt über ein PIN-Pad am Lesegerät (b), oder über die GUI der „PersoApp“-Open-Sourcr-Software-Bibliothek (a).
7. Die „PersoApp“-Open-Sourcr-Software-Bibliothek erhält die persönlichen Daten vom nPA und leitet sie an den eID-Server weiter.
8. Der eID-Server leitet die persönlichen Daten an den Dienstanbieter weiter.

5.5.3 Relevante, historische Sicherheitsvorfälle

Die „AusweisApp“ (ehemals Bürgerclient) der OpenLimit SignCubes AG ist ein proprietäres Softwareprodukt mit vergleichbarem Anwendungszweck wie die PersoApp-Software. Sie wird vom Bundesamt für Sicherheit in der Informationstechnik (BSI) unter https://www.ausweisapp.bund.de in Version 1.10.0 zur Verfügung gestellt. In der Vergangenheit ist eine ältere Version jedoch durch eine Sicherheitslücke auffällig geworden, durch die Schadcode eingeschleust werden könnte.

Version	*Source*	*Notes*
1.0.1	http://www.heise.de/newsticker/meldung/Neuer-Personalausweis-AusweisApp-mit-Luecken-2-Update-1133376.html	The initial release of AusweisApp allowed for abusing its update-functionality to infect end user PCs. This was possible due to • Improper check of update-server certificate • Automatic extraction of unsigned archives

5.5.4 Asset Stakeholder

ID	*Asset Stakeholder*	*Optional comment or description*
S1	End user	
S2	Online service provider	
S3	eID-service-provider	

5.5.5 Relevante Assets

ID	*Asset*	*Asset Stakeholder*
A1	PIN	S1
A2	Personal data (such as name, date of birth, address, etc.)	S1, S2, S3

5.5.6 Festlegung und Priorisierung von Sicherheitszielen

	ID	*Short Name*
Security Objective	O1	Confidentiality of PIN
Asset	A1	PIN
Stakeholder	S1	End user
Specification	Secure authentication is based on two conditions: • Only the end user owns a copy of the eID • Only the end user knows the PIN Thus, loss of confidentiality of the PIN is considered risky.	
Ranking of security objective **(conditions under which a violation of the security objective would be considered negligible, marginal, critical, or catastrophic)**		
Negligible		
Marginal		
Critical	Attacker obtains a PIN that can be correlated to an individual end user	
Catastrophic	Attacker obtains multiple PINs that can be correlated to individual users	

	ID	*Short Name*
Security Objective	O2	Confidentiality of personal data
Asset	A2	Personal data
Stakeholder	S1	End user
Specification	Online service providers might request personal data from end users who want to keep this information private from others.	
Ranking of security objective		

(conditions under which a violation of the security objective would be considered negligible, marginal, critical, or catastrophic)	
Negligible	Attacker obtains an encrypted portion of personal data
Marginal	Attacker obtains an entire data set that is encrypted
Critical	Attacker obtains personal data from one individual
Catastrophic	Attacker obtains personal data from a large amount of individuals

	ID	***Short Name***
Security Objective	O3	Integrity of personal data
Asset	A2	Personal data
Stakeholder	S1, S2	End user, Online service provider
Specification	End users need to be sure that no one else can spoof their identity. Online service providers need to be sure that their customers and users can’t pretend to be somebody else.	
Ranking of security objective **(conditions under which a violation of the security objective would be considered negligible, marginal, critical, or catastrophic)**		
Negligible	Attacker manipulates personal data but it becomes evident	
Marginal	Attacker restrictively manipulates personal data for an individual without becoming evident	
Critical	Attacker arbitrarily manipulates personal data for an individual without becoming evident	
Catastrophic	Attacker arbitrarily manipulates personal data for a large amount of individuals without becoming evident	

5.5.7 Auflistung der Systemnutzer

ID	***Actor***	***Description/Privileges***
AC1	End user	The end user permits eID service providers to receive personal data by entering a secret PIN.
AC2	Online service provider	The online service provider offers services to end users (such as web shops, online banking, etc.) and requests personal data from

		them for authentication purposes using an eID-server. This eID server may be provided by a third-party: the eID service provider.
AC3	eID-service-provider	The eID service provider offers eID server capabilities for online service providers, in order to allow them to request personal data from end users.

5.5.8 Externe Anforderungen

Wie in „D04-QM Programmierrichtlinien zur Erstellung von ‚PersoApp‘-Softwaremodulen“ bereits aufgeführt, stellen die folgenden technischen Richtlinien des Bundesamtes für Sicherheit in der Informationstechnik (BSI) die technischen Anforderungen an eine Implementierung der Online-Ausweisfunktion des neuen Personalausweises:

- BSI-TR-03112 “Das eCard-API-Framework”
- BSI-TR-03127 „Architektur elektronischer Personalausweis“
- BSI-TR-03128 „EAC-PKI’n für den elektronischen Personalausweis“
- BSI-TR-03130 „eID-Server“

5.6 Benutzbarkeit und Schutz der Privatsphäre

Die Sicherheit heutiger Systeme wird zu einem erheblichen Umfang von der Bereitschaft der Anwender bestimmt, die verfügbaren Sicherheitsmechanismen zu verwenden. Untersuchungen [1] zeigen, dass die Sicherheit durch eine scheinbare Verweigerungshaltung der Anwender gefährdet ist. Diese unterschätzen die Folgen der Verletzung von Sicherheit und „verweigern“ daher den Aufwand, die Sicherheitskonzepte zu nutzen. Die Sichtung des Standes der Wissenschaft in der Software-Ergonomie [2, 3] und der Human Computer Interaction (HCI) [4] zeigt, dass Sicherheit und Nutzbarkeit bisher unabhängig voneinander betrachtet werden. Untersuchungen zur Nutzbarkeit von Sicherheitswerkzeugen [1, 5] zeigen, dass heutige Nutzerschnittstellen systeminduziert sind [6]. Die Schlussfolgerung daraus ist, dass komplexe Sicherheitsmechanismen nicht vom Benutzer konfiguriert, sondern systemunterstützt und vor dem Nutzer verborgen durchgeführt werden müssen [7].

Die Definition und Zuweisung von Zugriffsrechten für einen Zugriff auf Daten eines nPA basieren bei der Online-Ausweisfunktion auf einer PKI [8]. Insbesondere das Berechtigungszertifikat eines Dienstanbieters spezifiziert seine Rechte auf Daten eines nPA zuzugreifen. Ein Ausweisinhaber kann diese Zugriffsrechte über seinen eID-Client weiter einschränkten [9]. Die Protokolle der Online-Ausweisfunktion sowie die Nutzung eines eID-Servers abstrahieren die Zertifikatsverwaltung und Durchsetzung der Zugriffsrechte, die durch Zertifikate repräsentiert sind, gegenüber dem Ausweisinhaber [9,10]. Gegenüber einem Ausweisinhaber wird ein Berechtigungszertifikat durch einen eID-Client mindestens wie folgt präsentiert [10]:

- Name des Dienstanbieters

- Erwünschte Zugriffsrechte

Weiterhin müssen auf Anforderungen des Ausweisinhabers die folgenden Daten eines Berechtigungszertifikates angezeigt werden [10]:

- Anschrift und E-Mail-Adresse des Dienstanbieters
- Zweck der Datenübermittlung
- Hinweis auf die für den Dienstanbieter zuständige Datenschutzbehörde
- Gültigkeitszeitraum des Zertifikates

Dies entspricht neben der PIN-Eingabe durch den Ausweisinhaber die rudimentäre GUI der Open-Source-Ausweis-Bibliothek.

Die erhobenen Daten nach Zugriff auf einen nPA – inklusive der Autorisierung durch den Ausweisinhaber durch Eingabe seiner PIN – dürfen von dem Dienstanbieter für Zwecke der Identitätsfeststellung nicht jedoch für geschäftsmäßige Zwecke und zur geschäftsmäßigen Übermittlung von Daten an Dritte verwendet werden [11]. Zahlreiche Anwendungsszenarien erfordern jedoch eine Verarbeitung und Weitergabe von persönlichen Daten an Dritte, wie z.B. CRM (vgl. [12]). Eine Möglichkeit für eine geschäftsmäßige Nutzung der Daten eines Ausweisinhabers besteht in einer zusätzlichen Eingabe der Daten durch den Ausweisinhaber und ihrer Verifizierung mit dem nPA und ihrer Bereitstellung über einen weiteren Dienst (s. „ID Safe“ [13]) und dem Beispielprozess „Anlage eines Kundenkontos und Abschluss eines Versicherungsantrags“ [14]. Im Folgenden werden mit dem Konzept der „Teil-Identität“ und der „Delegation von Rechten“ für eine Transparenz einer Datenweitergabe zwei Optionen für eine Erweiterung der GUI eines eID-Clients vorgeschlagen. Diese Erweiterungen sind nicht Bestandteil der rudimentären GUI, sondern als Vorschlag für einen zukünftigen eID-Client zu verstehen.

5.6.1 Die Teil-Identität

Die Aspekte IT-Sicherheit, Schutz der Privatsphäre und Benutzbarkeit adressieren das Konzept der Teil-Identität für Identitätsmanagement als eine Gestaltungsmöglichkeit der Benutzungsschnittstelle eines eID-Clients. Dazu bilden Teil-Identitäten ein natürliches Verhalten des Menschen auf IT-Systeme ab: Jeder Mensch tritt gegenüber anderen Menschen unterschiedlich auf. Teil-Identitäten unterstützen den Nutzer bei dieser Verhaltensweise, indem es ihm ein nachvollziehbares und individuelles Auftreten gegenüber seinen Kommunikationspartnern ermöglicht. So ist man beispielsweise beim Einkaufen annähernd anonym, beim Besuch bei Freunden jedoch sehr gut bekannt. Dieser (oft unbewusste) Wechsel der offenbarten Identität wird durch den Wechsel von situationsabhängigen Rollen modelliert, die *Teil-Identitäten* genannt werden. Eine Teil-Identität ist eine Menge von persönlichen Daten eines Benutzers, wobei jeder Benutzer über mehrere Teil-Identitäten verfügen kann. Ähnlich wie in der realen Welt wechselt der Benutzer in Rechnernetzen seine Teil-Identität, wodurch er sich – je nach Situation und Rolle – im Spektrum zwischen Anonymität und Identifikation bewegt. Auf diese Art können Nutzer ihre Privatsphäre bei einer bewussten Datenerhebung schützen und ermöglichen so zum anderen ei-

nen Reputationsaufbau gegenüber einem Kommunikationspartner unter der verwendeten Rolle.

5.6.2 Datentypen einer Teil-Identität

Eine Teil-Identität setzt sich aus einer Menge von Tupeln zusammen. Jedes Tupel beinhaltet die Beschreibung des persönlichen Datums (Bezeichner) und das persönliche Datum selbst:

- **Persönliches Datum (Pseudonym)**: In Abhängigkeit des gewünschten Auftretens im Spektrum zwischen Anonymität und Identifizierung, kann der Nutzer ein Pseudonym verwenden. Der Umfang einer Erhebung von persönlichen Daten des neuen Personalausweises bzw. ein Auftreten unter einem Pseudonym ist von dem Berechtigungszertifikat des Dienstanbieters und dessen Durchsetzung durch den Betreiber des verwendeten eID-Servers abhängig.
- **Schablone**: Die Schablone einer Teil-Identität ist die Menge der Bezeichner einer Teil-Identität, enthält aber keine persönlichen Daten. Schablonen können dem Nutzer daher als Muster für Teil-Identitäten zur Verfügung gestellt werden.
- **Bezeichner**: Jedes persönliche Datum sollte durch einen Bezeichner referenziert, z.B. „personname.given“.
- **Schlüssel-Bezeichner** verweisen auf Daten, die eindeutig sind wie bspw. ein privater kryptographischer Schlüssel.
- **Identifizierende Schlüssel-Bezeichner** verweisen auf Daten, mit denen eine Person eindeutig identifiziert werden kann wie bspw. die Personalausweisnummer.

Mit der Abbildung der Teil-Identitäten in ein IT-System werden die personenbezogenen Daten und die Einstellungen zur Zurechenbarkeit des Nutzers in einem eID-Client gespeichert. Der Nutzer wird mehrere Teil-Identitäten auf seinem System speichern, die er mit einem Identitätsmanagement-System verwaltet.

5.6.3 Komponenten einer generischen Benutzungsschnittstelle mit Teil-Identitäten

Aufbauend auf grundlegenden Funktionen eines eID-Clients für eine Nutzung der Online-Ausweisfunktion des neuen Personalausweises bauen die Komponenten zur Identitätskonfiguration, zur Identitätsaushandlung und einer Transparenz der Datenverarbeitung für eine Benutzungsschnittstelle mit Teil-Identitäten zum Schutz der Privatsphäre auf.

Die „Benutzungsoberfläche“ sollte entsprechend dem Modell der Teil-Identität adaptiv an das Benutzerwissen anpassbar sein [15]. Ihre Gestaltung trägt zur Akzeptanz des Sicherheitswerkzeugs bei [16]. Die Benutzungsoberfläche sollte die Sicherheit eines eID-Clients in verständlicher Weise widerspiegeln, um das Verständnis von Sicherheitslaien für die Online-Ausweisfunktion des neuen Personalausweises und ihre Einschätzung in den Schutz der Privatsphäre unterstützen zu können.

Die „Identitätskonfiguration“ sollte es einem Nutzer ermöglichen, situationsgerecht eine Teil-Identität auszuwählen und zu erstellen, mit der er sich seinem Kommunikationspartner gegenüber zeigen möchte bzw. gezeigt hat, um den gewünschten Dienst nutzen zu können. Dieser Auswahlvorgang sollte größtenteils automatisierbar sein, da das System eine erneute Kommunikation mit einem schon bekannten Partner erkennen und die früher mit diesem Partner genutzte Teil-Identität auswählen soll. Das System sollte die Möglichkeit haben, Kontextinformationen wie Ortswechsel, Zeit oder Nutzeraktionen auszuwerten, um einen Situationswechsel zu erkennen. Wechselt der Benutzer nachträglich innerhalb dieser Situation die Teil-Identität, so muss das System prüfen, ob der Grad der Anonymität der neuen Teil-Identität noch erreichbar ist. Da man gegenüber einem Partner seinen Grad der Anonymität nicht nachträglich erhöhen kann (Monotonie der Anonymität [17]), ist die Teil-Identität innerhalb einer Situation nicht beliebig wechselbar.

Sie soll Funktionen zur Verwaltung und Auswahl von Teil-Identitäten, eine geschützte Datenbank und die Situationserkennung enthalten. Die gesicherte Datenbank speichert die Teil-Identitäten des Nutzers sowie die Regeln für den situationsabhängigen Umgang mit den Teil-Identitäten. Eine Situation kann u.a. durch den Kommunikationspartner, die aktuelle Anwendung und die vorliegende Teil-Identität bestimmt werden. Ein Filter kann bspw. den Datenstrom der Anwendung mit der integrierten Online-Ausweisfunktion (eID-Client) nach personenbezogenen Daten des Nutzers überprüfen, wobei hier nicht in die Protokolle der Online-Ausweisfunktion des nPA eingegriffen werden soll. So kann er beispielsweise Felder eines web-basierten Formulars erkennen, falls sie sich an den P3P-Standard richten, und die dort geforderten Daten einfügen, soweit sie in der aktuellen Teil-Identität freigegeben sind.

Eine „Identitätsaushandlung“ ist dann notwendig, wenn ein Teilnehmer über seinen Kommunikationspartner mehr wissen möchte, als dieser anfangs preiszugeben bereit ist oder ein Konflikt über den Grad der Verbindlichkeit dieser Kommunikation besteht. Deshalb sollte den Kommunikationspartnern die Möglichkeit gegeben werden, die Teil-Identitäten untereinander „auszuhandeln“. Dabei sollten drei verschiedenen Arten der Aushandlung berücksichtigt werden:

- **Mensch-Mensch:** Zwei Kommunikationspartner einigen sich über die auszutauschenden Teil-Identitäten. Es kann eine mehrstufige Aushandlung stattfinden.
- **Mensch-Maschine:** Diese Situation findet sich häufig im Bereich des E-Commerce. Der Nutzer muss dem Server persönliche Daten wie Name, Adresse oder Finanzdaten bekannt geben. Oft werden weitere Daten verlangt. Hier kann eine Aushandlung stattfinden, wobei beispielsweise eine Übertragung von Bezahlinformationen von weiteren Daten durch den Server abhängen kann.
- **Maschine-Maschine:** In vielen Situationen ist eine automatische Aushandlung ohne direkte Benutzerbeteiligung möglich. Durch eine Implementierung von P3P zur Darstellung der personenbezogenen Daten ist eine begrenzte Aushandlung möglich, wenn das System des Nutzers die Datenschutzrichtlinie des Nutzers mit der des Kommunikationspartners vergleicht. Im Fall eines

Konfliktes kann der eID-Client den Nutzer auf diesen Konflikt aufmerksam machen und ihm Lösungen zur Auflösung des Konfliktes vorschlagen.

5.6.4 Transparenz einer Datenverarbeitung

Eine Transparenz einer Datenverarbeitung zielt auf eine Prüfung der Einhaltung von Datenschutzrichtlinien, z.B. anhand eines Datenschutzaudits durch einen Datenschutzbeauftragten. Das Konzept der Nutzungskontrolle für einen kontrollierbaren Zugriff auf (persönliche) Daten erweitert Zugriffskontrolle durch Regeln (Obligationen), die nach einem erfolgten Zugriff eingehalten werden sollten [18]. Obligationen beschreiben akzeptierbare Zustände der Nutzung persönlicher Daten, ohne einen Zugriff auf diese Daten im Voraus einzuschränken. Zusammen mit den Regeln für den Erstzugriff auf persönliche Daten (Bedingungen) sind sie Teil einer Datenschutzpolitik. Die Durchsetzung von Obligationen bezieht sich auf ihre Eigenschaften Kontrollierbarkeit und Beobachtbarkeit [19]. Hinsichtlich ihrer Kontrollierbarkeit kann ein Datenanbieter durch die Ausweitung seines Zugriffskontrollbereichs auf den Datenkonsumenten, z.B. durch Mechanismen des „Digital Rights Management“, oder durch die Integration von Kontrollmechanismen in den Prozess mit anschließender Verifikation die Obligationen selbst durchsetzen. Jedoch ist dieses Vorgehen nicht immer praktikabel. So müsste der Datenkonsument die Kontrolle über sein IT-System (teilweise) an den Datenanbieter abtreten. Beobachtbarkeit hingegen bedeutet, dass der Datenanbieter ohne Ausweitung seines Zugriffkontrollbereiches prüfen kann, ob der Datenkonsument die Obligationen einhält bzw. eingehalten hat. Beobachtbare Obligationen können direkt von dem Referenzmonitor des Datenanbieters durchgesetzt werden, z.B. die Obligation „Erneuter Zugriff auf die Daten innerhalb der nächsten k Tage“. Nicht-beobachtbare Obligationen beziehen sich auf Zugriffsanfragen, die sich außerhalb des Zugriffkontrollbereiches des Datenanbieters beziehen, z.B. „diese Daten dürfen nicht an Dritte weitergegeben werden“ und „lösche diese Daten spätestens nach k Tagen“.

Der im Folgenden beschriebene Vorschlag für eine Transparenz einer Datenverarbeitung verwendet Attributzertifikate nach dem dezentralisiertem „Trust Management„ [20], da sie eine fallweise Autorisierung und Widerruf einer Kooperation ermöglichen. Das Zugriffskontrollmodel besteht aus zwei Zugriffskontrollbereichen – (a) der Bereich des Nutzers als Datensubjekt und (b) der Bereich des Dienstanbieters als Datenanbieter, wobei jeder Bereich über einen Referenzmonitor verfügt. Der Einzelne (in der Rolle des Datensubjektes) definiert die autorisierten Zugriffe durch Zugriffsrechte auf die Daten zusammen mit Obligationen für die Nutzung der delegierten Zugriffsrechte und delegiert sie an den Dienstanbieter, der in der Rolle des Datenanbieters handelt. Die Teilnehmer an einer Delegation von Rechten sind wie folgt:

- **Nutzer:** Ein Nutzer nimmt die Rolle eines Datenanbieters ein, indem er Internetdiensten Zugriff auf seine Daten gewährt. Entsprechend zu der vereinbarten Datenschutzpolitik mit Dienstanbieter handelt der Nutzer als Dateneigentümer und delegiert die Autorisierungen für den Zugriff auf seine herausgegebenen Daten an die mit ihm kooperierenden Dienstanbieter. Ein Zugriffsrecht innerhalb einer Autorisierung ist aus der Menge {read, write, delete}.

- **Dienstanbieter:** Ein Dienstanbieter nimmt die Rolle eines Datenkonsumenten ein, wenn er Daten eines Nutzers verarbeitet. Er nimmt die Rolle eines Datenanbieters ein, wenn er Daten eines Nutzers an Dritte weitergibt.
- **Datenschutzbeauftragter:** Ein Datenschutzbeauftragter überprüft, ob eine Regel einer Datenschutzpolitik verletzt wurde. Ist dies der Fall, so soll der Datenschutzbeauftragte den Verursacher der Verletzung identifizieren.
- **„CA“:** Eine „CA (Certification Authority)“ zertifiziert die Identitäten der Nutzer und Dienstanbieter; ebenso stellt sie die Attributzertifikate für die zu delegierenden Rechte aus.

Ein Nutzer definiert die erlaubte Nutzung seiner Daten an einen Dienstanbieter als Datenkonsument als Attributzertifikate durch eine Datenschutzrichtlinie. Dabei richtet sich die Datenschutzrichtlinie an den Zweck der Delegation dieser Attribute. Entsprechend der Terminologie von Zugriffsrechten nach Subjekt, Objekt und Rechteart, ist ein Dienstanbieter als Datenkonsument das Subjekt, das Attributzertifikat mit den personenbezogenen Daten das Schutzobjekt und die Regeln zu deren Verwendung entsprechen den Rechtearten. Eine Datenschutzrichtlinie setzt sich wie folgt zusammen:

- **Datenkonsument:** Die personenbezogenen Daten des Nutzers dürfen ausschließlich und in Abhängigkeit der vereinbarten Datenschutzrichtlinie an den hier angegebenen Datenkonsumenten weitergegeben werden.
- **Dienstanbieter:** Ein Dienstanbieter als Datenanbieter darf die an ihn übertragenen Attribute nur gegenüber bestimmten Typen von Dienstanbietern weitergeben. Die Einschränkung wird durch den Nutzer mit der Auflistung dieser Dienstanbieter spezifiziert.
- **Dienstfunktion:** Ein Dienstanbieter kann mehrere Dienstleistungen anbieten, die jeweils wiederum mehrere Funktionen umfassen. Falls ein Attributzertifikat nicht auf eine bestimmte Funktion beschränkt ist, so kann ein Stellvertreter mit diesem Attributzertifikat Funktionen des Dienstanbieters aufrufen, die für den Zweck der Delegation nicht erforderlich sind. Um dies zu vermeiden, spezifiziert der Nutzer mit diesem Attribut die vereinbarte Datenverarbeitung.
- **Anzahl der erlaubten Verwendungen eines Attributzertifikates:** Überträgt der Nutzer seine Daten mit den obigen Einschränkungen an einen Dienstanbieter, so kann dieser diese Daten beliebig oft verwenden. Der Nutzer schränkt die Anzahl der Anwendungen durch eine obere Schranke für deren Verwendung ein.
- **Re-Delegation eines Attributzertifikates:** An dieser Stelle spezifiziert der Nutzer, ob ein Dienstanbieter die übertragenen Daten bzw. die Rechte für einen Zugriff auf sie als ein Attributzertifikat weitergeben darf.
- **Gültigkeitsdauer:** Ein Attributzertifikat mit den übertragenen Daten des Nutzers ist nur in einem bestimmten Zeitraum gültig. Dieser Zeitraum wird von dem Nutzer mit einem Start- und einem Enddatum festgelegt.

An eine solche Delegation von Rechten werden die folgenden Anforderungen entsprechend zur Transparenz einer Datenverarbeitung nach der jeweils vereinbarten Datenschutzrichtlinie gestellt:

- Ein Zugriff auf die Daten eines Nutzers soll nur dann erfolgen, wenn der anfragende Dienstanbieter das entsprechende Recht erhalten hat.
- Eine „CA“ stellt einem Dienstanbieter nur dann eine Autorisierung aus, wenn der betreffende Nutzer dazu explizit zugestimmt und damit die „CA“ hinsichtlich der Delegation eines Rechtes angefragt hat. Eine ausgestellte Autorisierung wird von dem Auditor verwendet, um den SOLL-Zustand der Datenweitergaben zu rekonstruieren und mit dem IST-Zustand abzugleichen.
- Nur die autorisierten Dienstanbieter sollen Zugriff auf die entsprechenden Daten des Nutzers erhalten. Zusätzliche Daten über den Nutzer und seinen Transaktionen, über die Dienstanbieter ihr Profil über den Nutzer erweitern können, dürfen nicht anfallen.
- Ein Internetdienst „S1“ darf Daten „d“ eines Nutzer „U1“ in den Datensatz eines anderen Nutzers „U2“ schreiben, falls
    - der Nutzer „U1“ an den Anbieter des Internetdienstes „S1“ das Schreibreicht die Daten „d“ in den Datensatz des Nutzers „U1“ delegiert hat oder die Daten „d“ anonymisiert sind UND
    - der Nutzer „U2“ an den Anbieter des Internetdienstes „S1“ das Recht delegiert hat die Daten „d“ in seinen Datensatz zu schreiben.
- Ein Internetdienst „S1“ darf die Daten „d“ des Nutzers „U1“ an den Internetdienst „S2“ weitergeben, falls der Nutzer „U1“ an den Anbieter des Internetdienstes „S2“ das Recht delegiert hat die Daten „d“ von dem Anbieter des Internetdienstes „S1“ zu lesen oder die Daten „d“ sind anonymisiert.
- Falls ein Dateneigentümer dem Anbieter des Internetdienstes „S1“ die delegierten Rechte auf die Daten „d“ zuzugreifen widerrufen hat, so muss der Internetdienst „S1“ die entsprechenden Daten „d“ löschen. Dies muss rekursiv geschehen.
- Damit ein Datenschutzbeauftragter den IST-Zustand der Datenweitergaben mit dem SOLL-Zustand abgleichen kann, soll die Historie der Weitergabe zu den betroffenen Daten dokumentiert werden.

6 Interne und externe Anforderungen

Interne Anforderungen haben Einfluss auf die Durchführbarkeit einzelner Aktivitäten, wie sie in diesem Dokument beschrieben worden sind. Daher ist es den Projektverantwortlichen freigestellt, situationsbedingt die Entscheidung zu treffen, einzelne Aktivitäten hinzuzufügen, zu modifizieren, oder gänzlich auszulassen. Mögliche Ursachen hierfür können in der Zeit- und Ressourcenplanung liegen, oder technischen und organisatorischen Gegebenheiten geschuldet sein, die sich im Projektverlauf

ergeben. In jedem Fall sind Abweichungen von dem im vorliegenden Dokument beschriebenen Prozess in nachvollziehbarer Weise zu dokumentieren.

Externe Anforderungen umfassen den Einsatz von Hardware, Software und ggf. Dokumentationen externer Parteien, die ausschließlich gemäß ihrer vorliegenden Lizenz verwendet werden dürfen. Weiterhin unterliegen die der Architektur zugrundeliegenden Änderungen. Daraus können sich künftig Anpassungen bei der Architektur ergeben.

7 Abläufe

Durch eine sorgfältige Prüfung der hier getroffenen Festlegungen wurde sichergestellt, dass das Architekturkonzept, die darauf aufbauenden Sicherheitsanforderungen sowie die Festlegungen zur Benutzbarkeit und zum Schutz der Privatsphäre korrekt und angemessen sind. Sollte sich dennoch im Projektverlauf die Notwendigkeit für eine Anpassung ergeben, dann ist dies einerseits an alle Projektteilnehmer explizit und verständlich zu kommunizieren, zum anderen gilt es die entsprechende Änderung detailliert festzuhalten und nachvollziehbar zu dokumentieren.

8 Prozesskennzahlen und Qualitätskriterien

Im Dokument „D02-QM Qualitätskriterien: Aufbau, Messgrößen und Bewertung“ wurden Qualitätskriterien für das Projekt „PersoApp“ definiert. Diese Qualitätskriterien finden ebenfalls auf das zugrundeliegende Dokument Anwendung.

9 Mitgeltende Dokumente

[1] Whitten, Alma und Tygar, J.D. *Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0*. In Proceedings of the 8th USENIX Security Symposium, August 1999.

[2] Nielsen, Jakob. *Usability Engineering*. Academic Press. 1993.

[3] Deutsches Institut für Normung. *DIN EN ISO 9241 Teil 10: Grundsätze der Dialoggestaltung*, 1996.

[4] Dix, Alan, Finlay, Janet, Abowd, Gregory und Beale, Russell. *Human-Computer Interaction, Volume 2*. Prentice Hall. 1998.

[5] Kröger, Veli-Pekka. *Security of User Interfaces – A Usability Evaluation of F-Secure SSH*. 1999. http://www.tml.tkk.fi/Opinnot/Tik-110.501/1999/papers/uisecurity/ui_security.html

[6] Waidner, Michael. *Open Issues in Secure Electronic Commerce*. Technical report, IBM Research Division, Zurich, October 1998.

[7] Gerd tom Markotten, Daniela und Jendricke, Uwe. *Usability meets Security - The Identity-Manager as your Personal Security Assistant for the Internet*. In Proceedings of the 16th Annual Computer Security Applications Conference, S. 344-353, Dezember 2000.

[8] BSI. *Technische Richtlinie TR-03128 EAC-PKI'n für den elektronischen Personalausweis. Version 1.1*. Bundesamt für Sicherheit in der Informationstechnik, 2010.

[9] BSI. Technische *Richtlinie TR-03130 Technische Richtlinie eID-Server. Version 2.0*. Bundesamt für Sicherheit in der Informationstechnik, 2012.

[10] BSI. Technische *Richtlinie TR-03127 Architektur elektronischer Personalausweis und elektronischer Aufenthaltstitel. Version 1.15*. Bundesamt für Sicherheit in der Informationstechnik, 2012.

[11] Bundestag. *Gesetz über Personalausweise und den elektronischen Identitätsnachweis sowie zur Änderung weiterer Vorschriften vom 18. Juni 2009*. Bundesgesetzblatt Jahrgang 2009 Teil I Nr. 33, ausgegeben zu Bonn am 24. Juni 2009.

[12] Wohlgemuth, Sven. *Privatsphäre durch die Delegation von Rechten*. Vieweg+Teubner Verlag. 2009.

[13] Bundesverwaltungsamt. *Vergabestelle für Berechigungszertifikate. Leitlinie für die Vergabe von Berechtigungen für Diensteanbieter nach $ 21 Abs. 2 Personalausweisgesetz. Version1.0*. http://www.personalausweisportal.de/SharedDocs/Downloads/DE/Material-Dienstleis-ter/Leitlinie_VfB_Vergabe_Berechtigungszertifikate.pdf?__blob=publicationFile . 2010.

[14] BMI. *Beispielprozess: Anlage eines Kundenkontos und Abschluss eines Versicherungsantrags*. http://www.personalausweisportal.de/SharedDocs/Downloads/DE/Material-Dienstleis-ter/Beispielprozess_Kundenmanagement_Versicherung.html;jsessionid=9E62F956418226E33269572A9B40BF33.2_cid297?nn=3043440, 2011.

[15] Gerd tom Markotten, Daniela und Kaiser, Johannes. *Benutzbare Sicherheit – Herausforderungen und Modell für E-Commerce-Systeme*. Wirtschaftsinformatik, 42(6):531-538, Dezember 2000.

[16] SINIUS-Institut Heidelberg im Auftrag des Deutschen Instituts für Vertrauen und Sicherheit im Internet (DIVSI). *DIVSI Milieu-Studie zu Vertrauen und Sicherheit im Internet*. DIVSI, 2012.

[17] Wolf, Gerrit und Pfitzmann, Andreas. *Properties of protection goals and their integration into a user interface*. Computer Networks, 32:685-699, 2000.

[18] Park, Jaehong und Sandhu, Ravi. *The $UCON_{ABC}$ usage control model*. 24th ACM Transactions on Information and System Security 7(1):128-174, 2004.

[19] Hilty, Manuel, Basin, David und Pretschner, Alexander. *On Obligations*. In: European Symp. on Research in Computer Security (ESORICS 2005), Milan, 2005.

[20] Blaze, Matt, Feigenbaum, Joan und Lacy, Hack. *Decentralized Trust Management*. In: Symposium on Security and Privacy, Oakland, 1996.

[21] BSI. Technische *Richtlinie TR-03112 Teil 7 eCard-API-Framework - Protocols. Version 1.1.2*. Bundesamt für Sicherheit in der Informationstechnik, 2012.