

PersoSim – Ein Simulator für den elektronischen Personalausweis

Holger Funke¹ und Tobias Senger²

Kurzfassung:

Dieser Artikel beschreibt die Vorteile einer Open-Source-Lösung zur Simulation der Chipkartenfunktionen des elektronischen Personalausweises. Es wird aufgezeigt, auf welcher Architektur der Simulator basiert. Anschließend wird eine Einordnung des Simulators in die deutsche eID-Landschaft vorgenommen. Darüber hinaus wird im Artikel das Prinzip der virtuellen Kartenleser dargestellt, ebenso wie eine Migration auf ein Android-Smartphone mit NFC-Unterstützung.

Stichworte: Personalausweis, Simulator, TR-03110, Musterkarte, eID-Client, eID-Server, Open Source, nPA

1. Einführung

Mit dem elektronischen Personalausweis bekommt der Bürger die Möglichkeit, bisherige Authentisierungsverfahren, wie beispielsweise die Kombination aus Benutzername und Passwort, durch ein Verfahren mit einer höheren Authentisierungsstärke zu ersetzen. Möglich wird dies durch den Einsatz eines integrierten Chips im Personalausweis, der damit das Potenzial erhält, eine größere Verbreitung sowohl im Bereich eGovernment als auch im Bereich eBusiness zu erhalten. Anwender können ihren Personalausweis nutzen, um sich für Dienste anzumelden, und Dienstanbieter können sicherstellen, dass es sich bei dem Anwender auch tatsächlich um genau diesen Anwender handelt. Dieses Prinzip der gegenseitigen Authentifizierung ist das zentrale Element des elektronischen Personalausweises.

Die Einführung des neuen Personalausweises im Jahr 2010 wurde vorab begleitet von einem umfangreichen Anwendungstest, in dessen Rahmen den Teilnehmern neben dem Zugang zu unterschiedlichen eID-Servern, die für den Betrieb notwendig sind, auch verschiedene Musterkarten zur Verfügung gestellt wurden. Mit diesen Musterausweisen konnten interessierte Anwender ihre Dienste bereits vorab prüfen, um dann ihre Dienste mit der Ausgabe der ersten Personalausweise online frei zu geben.

Im Rahmen des Anwendungstests waren die Dienstanbieter bisher jedoch darauf angewiesen, Musterkarten zu benutzen, deren Verwendung sich in der Praxis als zu unflexibel herausstellte. Diese Lücke kann nun das Open-Source-Projekt „PersoSim“ zumindest teilweise schließen, indem ein Simulator für die Funktionen der Chipkarte auf dem neuen Personalausweis bereitgestellt wird. Testkarten sind zwar weiterhin eine wichtige Komponente für die Evaluierung und Tests neuer eID-Dienste, eine reine Software-Simulation wie PersoSim hat demgegenüber jedoch den Vorteil, jederzeit kostenfrei zur Verfügung zu stehen und schnell an neue Bedürfnisse

¹ HJP Consulting GmbH, Borcheln (Paderborn)

² Bundesamt für Sicherheit in der Informationstechnik (BSI), Bonn

angepasst werden zu können. PersoSim simuliert dabei sämtliche Mechanismen und kryptographische Protokolle, die die technische Richtlinie TR-03110 [1] des Bundesamts für Sicherheit in der Informationstechnik (BSI) beschreibt. Darüber hinaus werden zehn unterschiedliche Personalisierungen mitgeliefert, die unterschiedliche Daten für verschiedene Personen darstellen. Auf diese Weise können bspw. auch Personalausweise simuliert werden, die beim Bundesverwaltungsamt als gesperrt vermerkt sind. Eine Liste mit allen zehn Personalisierungen und den dazugehörigen Daten steht auf der Website des Projekts bereit³.

Der Simulator steht allen Interessierten, die mit dem Einsatz von PersoSim eine Alternative zu bisherigen Musterkarten suchen, online zur Verfügung. Dies können Dienstleister sein, die ihre Implementierungen mit dem Simulator verifizieren möchten. Aber auch Entwickler von eID-Clients, die den Simulator nutzen können, um ihre Interpretation der technischen Richtlinien zu testen, profitieren von PersoSim. Daneben ist der Simulator aber auch für jeden interessant, der die Protokolle des Personalausweises näher untersuchen möchte. Für den Simulator wird als Open-Source-Projekt auch der Quellcode zur Verfügung gestellt, damit Interessierte anhand der Implementierung nachvollziehen können, wie die einzelnen kryptographischen Protokolle des Personalausweises funktionieren.

Das BSI hat den Open Source Simulator durch die Firma HJP Consulting beauftragt. PersoSim richtet sich dabei sowohl an Entwickler aus dem Open Source Bereich, die sich aktiv an der Fortentwicklung des Simulators beteiligen wollen, als auch an Anwender, welche eine sofort lauffähige (Musterkarten-)Simulation benötigen.

Das BSI selbst verfolgt mit dem Simulator zusätzlich das Ziel, PersoSim sowohl zur Weiterentwicklung als auch zur prototypischen Umsetzung neuer Sicherheitsprotokolle einzusetzen. Die Weiterentwicklung bestehender sowie die Evaluierung neuer und stärkerer Sicherheitsprotokolle ist bei der Verwendung von physikalischen Chipkarten mit hohem zeitlichen und finanziellen Aufwand verbunden, da hier einige externe Parteien einbezogen werden müssen, wie Chip-Hersteller, COS-Hersteller, Anwendungsentwickler und Kartenproduzenten. Mit dem Einsatz eines software-basierten Simulators entfallen diese Einschränkungen, da direkt auf das zu testende Protokoll zugegriffen werden kann und dieses direkt nach der Änderung durch eine einfache Kompilierung lauffähig ist.

Die beiden Autoren haben das Projekt PersoSim im Rahmen der Veranstaltung Open Identity Summit im September 2013 erstmals öffentlich vorgestellt [2]. Darüber hinaus haben die beiden Autoren das Projekt in der Zeitschrift Datenschutz und Datensicherheit (DuD) beschrieben [3].

1.1. Einordnung des Simulators in die eID-Landschaft

Die folgende Abbildung 1 zeigt die eID-Landschaft des deutschen Personalausweises mit den verwendeten Systemen und deren Interaktionen.

³ Website: <http://www.persosim.de>

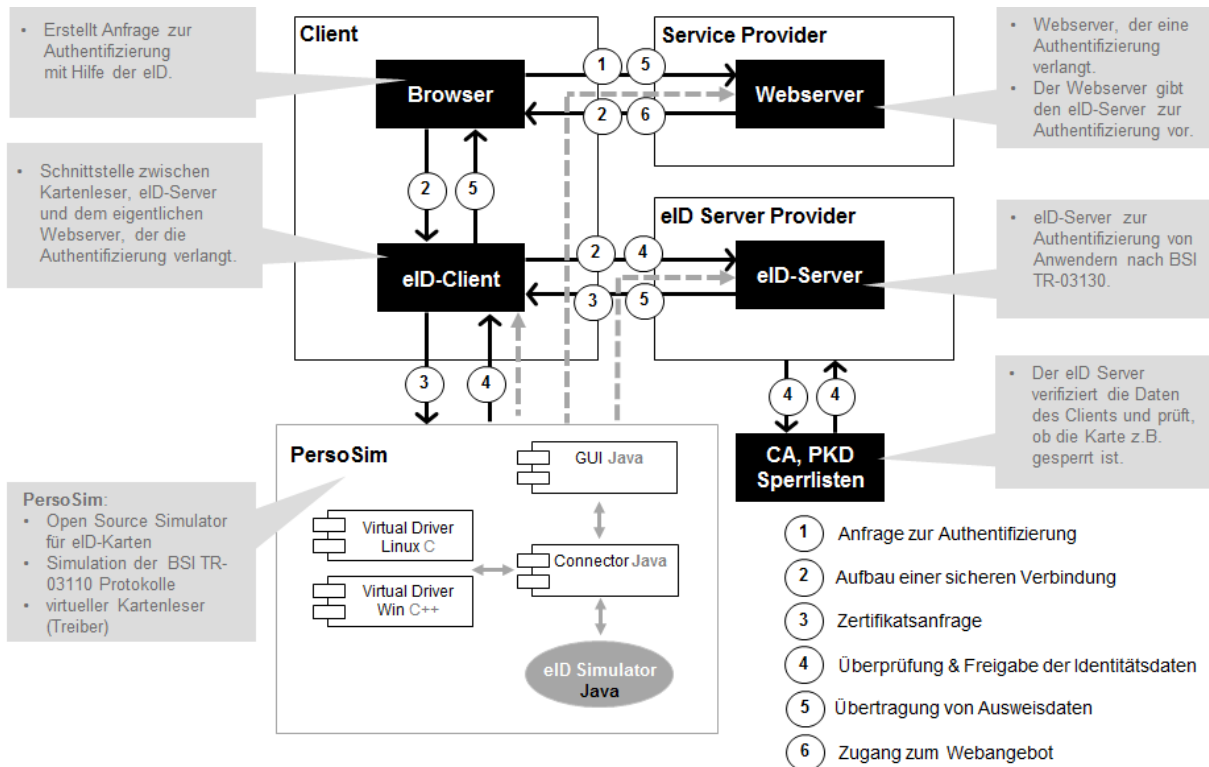


Abbildung 1: Die eID-Landschaft in Deutschland

PersoSim ersetzt dabei den Personalausweis und interagiert direkt mit dem eID-Client des Anwenders.

2. Der Simulator

Für die Simulation einer komplexen Smartcard wie dem nPA ist es notwendig, eine Vielzahl der Chipkarten-Kommandos nach ISO7816 sowie die Protokolle der BSI TR-03110 umzusetzen. HJP Consulting konnte durch die langjährige Erfahrung bei der Entwicklung von Simulatoren im Kontext hoheitlicher Dokumente einen schnellen Einstieg in das Projekt ermöglichen. So konnte bereits im Dezember 2013 eine erste lauffähige Version des Simulators auf der Website des Projekts veröffentlicht werden.

2.1. Simulation einer Java Card

Der Simulator besteht grundsätzlich aus zwei unterschiedlichen Bausteinen. Der eine Baustein beschreibt die eigentliche Kartenapplikation, der andere Baustein beschreibt die Schnittstellen zwischen der Kartenapplikation und der Außenwelt – sprich: hier werden die Kommandos vom Kartenleser entgegengenommen und an den Simulator weitergereicht.

Die Kartenapplikation stellt die eigentliche Simulation des elektronischen Personalausweises dar. In diesem Baustein werden alle kryptographischen Protokolle implementiert, die die Funktion des Personalausweises ausmachen und die in [1] beschrieben sind. Eine ausführliche Beschreibung der einzelnen Protokolle und Zugriffsmechanismen findet sich in [4]. An dieser Stelle sei nur

kurz erwähnt, um welche Protokolle der Extended Access Control (EAC) Protokollfamilie es sich hier handelt:

- Password Authenticated Connection Establishment (PACE),
- Terminal Authentication (TA),
- Chip Authentication (CA),
- Restricted Authentication (RI).

All diese Protokolle sind im Simulator umgesetzt, was eine vollständige Simulation der eID-Funktion erlaubt. Dazu gehören auch alle in der Technischen Richtlinie vorgesehenen kryptographischen Verfahren mit allen dort aufgeführten Schlüssellängen, angefangen bei Triple-DES bis hin zu AES mit 256 Bit-Schlüsseln. Die korrekte Funktionsweise des Simulators wurde im Januar 2015 durch eine offizielle Zertifizierung durch den TÜViT und das BSI gemäß TR-03105 bzw. TR-03110 nachgewiesen.

Neben den Protokollen spielen natürlich auch die eigentlichen Daten auf dem Personalausweis eine wichtige Rolle. Dabei wird unterschieden zwischen dokumentenspezifischen Daten, wie die Angabe zu kryptographischen Verfahren oder dazugehörigen Schlüssellängen, und inhaberspezifischen Daten, wie der Name oder der Wohnort des Dokumenteninhabers.

Der Simulator beinhaltet bereits zehn unterschiedliche Datensätze mit plausiblen Musterdaten („Erika Mustermann“). Auch die vorkonfigurierten kryptographischen Verfahren und Schlüssellängen stimmen mit den auf dem derzeitigen Personalausweis gewählten überein. Auf diese Weise kann PersoSim out-of-the-box mit sinnvollen Daten eingesetzt werden.

Für den Zugriff auf den realen Personalausweis benötigt ein Dienstanbieter ein entsprechendes Zertifikat, das er bei der Vergabestelle für Berechtigungszertifikate (VfB) beantragen muss. Erst mit diesen Zertifikaten ist der vollständige Zugriff auf die gespeicherten Daten möglich. Für den Einsatz des PersoSim sind analog Testzertifikate vorgesehen. Die Daten des Simulators enthalten dafür Testzertifikate, die aus der dazugehörigen Test-PKI des BSI stammen. Der PersoSim enthält also in der Standardkonfiguration bereits Zertifikate, die es dem Anwender ermöglichen, auf die Daten zuzugreifen.

Der bisherige Simulator war Teil der Produktfamilie GlobalTester⁴. Dabei handelt es sich ebenfalls in der Grundversion um ein Open-Source-Projekt zum Testen von Chipkarten. Der GlobalTester wird bereits seit Jahren erfolgreich eingesetzt und die Erfahrungen, die mit ihm gesammelt werden konnten, fließen nun auch in PersoSim ein. Der Simulator aus der GlobalTester-Produktfamilie wurde unter anderem auch eingesetzt, um ein Beispiel für den Protokollablauf im Personalausweis abzubilden [5]. Mit dem Simulator können beide Seiten der

⁴ Website: <http://www.globaltester.org>

Kommunikation (Terminal und Chip) dargestellt werden, so dass auch die Informationen und Schlüssel, die der Chip intern berechnen würde und die im Allgemeinen nicht zugänglich sind, mit der Simulation nach außen bereitgestellt werden können.

Der Simulator des Projekts PersoSim ist in der Programmiersprache Java implementiert. Konkret handelt es sich dabei um eine reduzierte Java-Variante, die es erlaubt, Java Card Applets auszuführen. Die Wahl der Java Card Plattform ermöglicht eine chipkartennahe Implementierung und durch den hohen Bekanntheitsgrad von Java wird Entwicklern ein schneller und plattformübergreifender Einstieg in das Projekt PersoSim ermöglicht.

Der eigentliche Simulator innerhalb PersoSims basiert auf einer Architektur, wie sie in der folgenden Abbildung 2 beschrieben ist. Der Simulator besteht aus einer Plattform, die sich um einzelne Kommandos, Secure Messaging und das Dateimanagement kümmert. Basierend auf dieser Plattform können weitere Protokolle wie PACE hinzugefügt werden, die der Simulator anschließend ausführen kann.

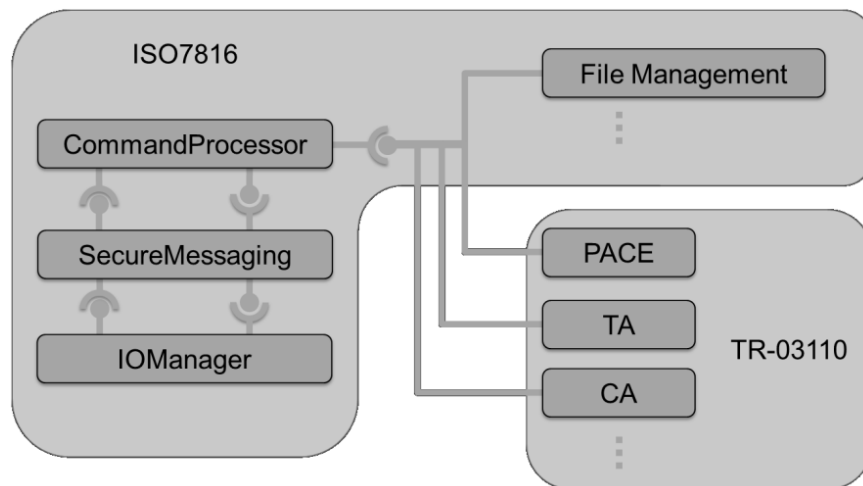


Abbildung 2: Architektur des Simulators

Die Anbindung des Simulators geschah bisher über spezielle Hardware, die die Kommunikation über die Funkschnittstelle für kontaktlose Chipkarten nach ISO/IEC 14443 [8] realisierte. Eine derartige dedizierte Hardware ist für den typischen PersoSim-Anwender nicht oder nur schwierig finanzierbar. Aus diesem Grund wurde für das Projekt PersoSim ein alternativer Weg für diese Kommunikationsschnittstelle entwickelt: ein virtueller Kartenleser.

2.2. Anbindung über virtuelle Kartenleser

Wie bereits dargestellt, ist die Nutzung von zusätzlicher Hardware für den direkten Zugriff auf den Simulator nicht vorgesehen. Aus diesem Grund wird für die Kommunikation mit dem Simulator ein virtueller Kartenleser bereitgestellt, den der Anwender auf seinem System installieren kann und der dem

jeweiligen Betriebssystem einen tatsächlichen Kartenleser vortäuscht. Bei diesem „Kartenleser“ handelt es sich lediglich um einen Treiber, der sich im Betriebssystem als neuer Kartenleser registriert.

Bei einem typischen Hardware-Kartenleser durchlaufen die Daten alle Schichten des ISO-OSI-Schichtenmodells, angefangen bei der Applikation in der oberen Schicht bis zur Bitübertragung in der untersten Schicht. Bei dem hier verwendeten virtuellen Kartenleser handelt es sich um eine reine Softwarelösung, so dass die unteren physikalischen Schichten ignoriert werden können.

Derzeit unterstützt PersoSim die folgenden Betriebssysteme:

- Microsoft Windows 7 (32 und 64 Bit),
- Microsoft Windows 8.1 (32 und 64 Bit),
- Ubuntu Linux 12.04 LTS (32 und 64 Bit).

Der Quellcode der jeweiligen Treiber wird ebenfalls bereitgestellt, so dass der geneigte Anwender auch Treiber für mögliche andere Linux-Derivate eigenständig kompilieren kann.

Mit dem virtuellen Kartenleser kann der Simulator in jeder Applikation genutzt werden, die eine Anbindung an einen Kartenleser verwendet. Im Kontext des elektronischen Personalausweises sind dies in erster Linie die eID-Clients, die lokal beim Anwender installiert sind und die die Kommunikation zwischen dem Chip und dem eID-Server realisieren. Aktuell handelt es sich dabei um die folgenden eID-Clients:

- AusweisApp1 (Bund, BSI),
- AusweisApp2 (Governikus, Bund, BSI),
- Open eCard App (Open eCard Team, ecsec),
- PersoApp (Fraunhofer SIT, TU Darmstadt, ageto).

Mit dem virtuellen Kartenleser ist der Anwender in der Lage, alle drei Lesertypen, die der Personalausweis vorsieht – nämlich Basisleser, Standardleser und Komfortleser – zu simulieren. Zu diesem Zweck greifen alle Treiber auf ein zusätzliches Java-Paket zu, das die Funktionen der Kartenleser unterstützt. Diese Vorgehensweise erlaubt es, dass alle Treiber plattformunabhängig auf eine Bibliothek zugreifen können, die weitere Funktionen bietet. Der Treiber für einen Basisleser kann die Kommandos einfach an den Simulator weitergeben. Standard- und Komfortleser dagegen bieten zusätzliche Funktionen an, die über das bloße Durchreichen der Kommandos hinausgehen. Die technische Richtlinie TR-03119 [6] beschreibt alle Funktionen, die die Kartenleser im Kontext des Personalausweises unterstützen müssen, wie z. B. die Funktionen eines PIN-Pads zur Eingabe von Passwörtern. Weiterhin soll diese Erweiterung auch die Statusmeldungen des Treibers ausgeben können wie in [6] gefordert. In diesem Paket lassen sich auch die notwendigen Kommandos wie

EstablishPACEChannel() unterbringen. Diese Vorgehensweise hat den großen Vorteil, dass die Logik der zusätzlichen Funktionen in einem einzigen Paket (Connector) gekapselt werden kann und auf beiden Betriebssystemen zur Verfügung steht. Somit muss der virtuelle Treiber nur noch die betriebssystem-spezifischen Informationen bereithalten und die Logik kann an nur einer einzigen Stelle entwickelt und gepflegt werden. Die folgende Abbildung 3 stellt den Aufbau der virtuellen Kartenleser dar:

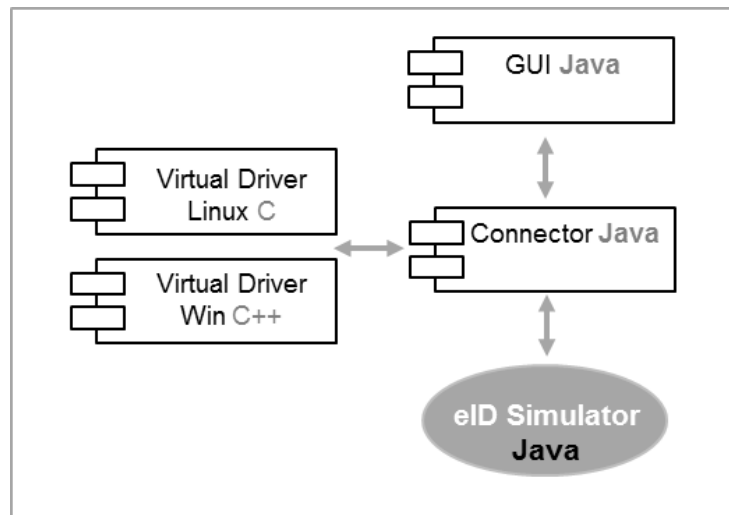


Abbildung 3: Architektur der virtuellen Lesegeräte

Die beiden Treiber für Windows und Linux integrieren sich in das jeweilige Betriebssystem wie ein normaler Kartenleser. Beide virtuelle Treiber sind möglichst schlank gehalten und greifen auf Funktionen des Connectors zurück, der wiederum für die Kommunikation mit dem Simulator verantwortlich ist. Die Oberfläche ist als Rich Client konzipiert. Hier kann der Anwender über den Connector mit dem Simulator interagieren.

Die Simulation eines PIN-Pads ermöglicht eine Annäherung an das typische Verhalten eines Standard- oder Komfortkartenlesers. Da die hier beschriebenen Treiber in Kombination mit dem Simulator aber in erster Linie für Testzwecke eingesetzt werden sollen, erscheint die Eingabe eines Passworts wie PIN oder CAN über das virtuelle PIN-Pad allerdings ein wenig mühselig. Aus diesem Grund wird eine Konfigurationsmöglichkeit für die Treiber angeboten, mit der sich die Passwörter speichern lassen, so dass diese nicht bei jedem Aufruf des Simulators bzw. des Kartenlesers eingegeben werden müssen.

3. Simulation auf Android mit NFC

In einer nächsten Ausbaustufe des Simulators soll der gesamte Funktionsumfang von PersoSim auf mobile NFC-Geräte wie bspw. Smartphones übertragen werden. Statt des virtuellen Kartenlesers wird die im Gerät vorhandene NFC-Schnittstelle für die Kommunikation mit der Gegenstelle verwendet (siehe Abbildung 3). Durch die Nutzung von PersoSim auf Smartphones ergibt sich ein erweitertes Einsatzgebiet bspw.

bei der Verwendung auf stationären Terminals oder Automaten mit eID-Unterstützung, auf denen kein Zugriff für eine Installation des virtuellen Kartenlesers möglich ist.

Für diese Umsetzung bieten sich derzeit besonders NFC-fähige Smartphones mit Android-Betriebssystem an, da diese über eine gut dokumentierte API Zugriffe auf die NFC-Schnittstelle bieten. Einige Open-Source-Projekte wie bspw. das Projekt androsmex⁵ oder die mobilen Versionen der o.g. Open eCard App und PersoApp nutzen bereits die NFC-Schnittstelle unter Android für Kommunikation mit dem nPA. Diese Apps verwenden die NFC-Schnittstelle im Reader/Writer-Mode.

Für die bei PersoSim geplante Simulation einer Smartcard auf einem NFC-fähigen Smartphone wird jedoch der Card-Emulation-Mode verwendet. Seit der Android-Version 4.4 (KitKat) wurde die „Host Card Emulation“-API (HCE) eingeführt, welche es einer App erlaubt, auf einem Android-Gerät eine virtualisierte Smartcard zu erzeugen. HCE emuliert dabei eine auf ISO/IEC 7816 [7] basierende Smartcard, die das kontaktlose ISO/IEC 14443-4 [8] (ISO-DEP)-Protokoll für die Kommunikation über die NFC-Schnittstelle des Geräts verwendet.

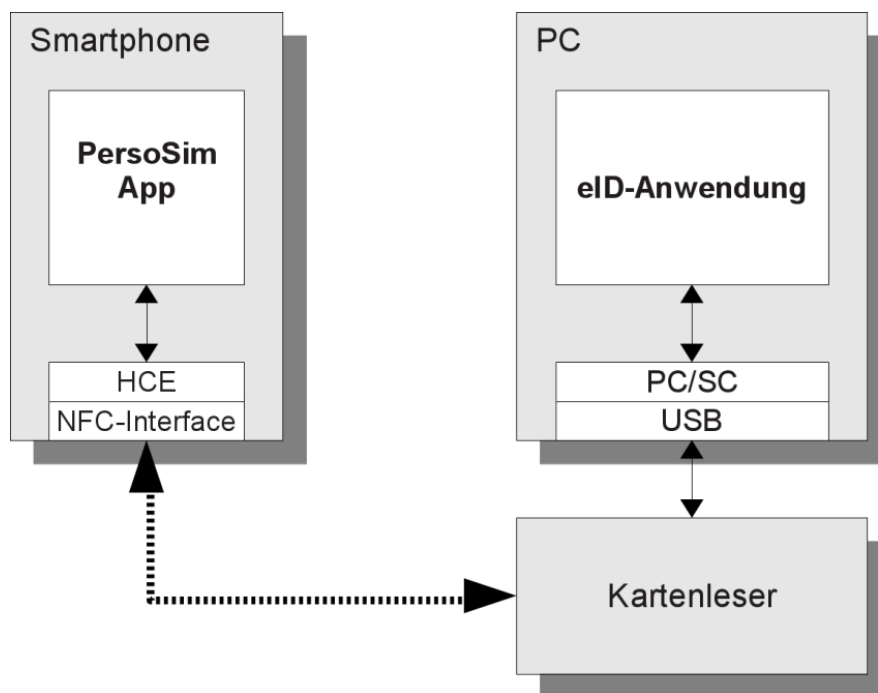


Abbildung 4: PersoSim auf einem NFC-Smartphone

Da PersoSim bereits in Java implementiert wird, ist eine Portierung des Programmcodes auf die Android-Plattform mit geringem Aufwand möglich. Anpassungen werden vor allem an der GUI sowie an der Anbindung zur NFC-Schnittstelle notwendig sein. Statt der Anbindung über einen virtuellen Kartenleser, wird dann unter Android direkt auf die HCE-API zugegriffen werden.

⁵ Website: <https://code.google.com/p/androsmex>

4. Die Open Source Community

Ein wichtiger Teil eines Open-Source-Projekts ist der Aufbau und die Pflege einer Community. Bei einer Community wird üblicherweise unterschieden in Anwender und Entwickler. Ein Open-Source-Projekt steht und fällt mit der Beteiligung weiterer Anwender und Entwickler. Erst durch die Zusammenarbeit unterschiedlicher Anwender und Entwickler mit durchaus unterschiedlichen Interessen wird ein Open-Source-Projekt lebendig.

Das Projekt PersoSim veröffentlicht den Quellcode unter der GNU General Public License v3.0 (GPLv3), so dass alle Erweiterungen, die auf der Originalversion basieren, wiederum der Community zur Verfügung gestellt werden. Die GPL ist die erste Copyleft-Lizenz für den allgemeinen Einsatz, die gewährleistet, dass Änderungen oder Ableitungen von GPL-lizenzierten Werken nur unter den gleichen Lizenzbedingungen veröffentlicht werden dürfen.

4.1. Vorteile von Open Source

Ein wichtiges Argument, das Anwender für den Einsatz von Open Source Software (OSS) ins Feld bringen, ist u.a. das Einsparen von Lizenzkosten und die Möglichkeit, die Software an die eigenen Anforderungen anzupassen. Darüber hinaus gibt es aber noch weitere wichtige Gründe, die für den Einsatz von OSS sprechen. Neben den offenen Standards, die gerade beim Einsatz eines Simulators wie PersoSim wichtig sind, ist auch die Herstellerunabhängigkeit ein wichtiges Argument. Mit PersoSim wird eine Simulator-Plattform bereitgestellt, die auch von anderen Firmen als Basis für ihre Entwicklungen im Chipkartenbereich eingesetzt werden kann. Auch die Argumente Leistungsfähigkeit und technische Qualität werden bei OSS gerne vorgebracht, denn auch hier gilt: Viele Augen sehen viel. Eine weltweite Entwicklergemeinschaft hat Einblick in den Quellcode und kann gegebenenfalls Probleme rasch erkennen und beheben.

Diese Gründe und die positiven Erfahrungen, die die HJP Consulting mit dem OSS-Testwerkzeug GlobalTester sammeln konnte, haben dazu beigetragen, das Projekt PersoSim wiederum als Open-Source-Projekt zur Verfügung zu stellen.

4.2. Die Community der Anwender

Für die PersoSim-Anwender wurde bereits eine spezielle Website aufgesetzt (www.persosim.de). Zusätzlich zu einer kurzen Einführung in das Projekt PersoSim, werden dem Anwender hier alle Informationen zur Verfügung gestellt, die für den Betrieb des Simulators notwendig sind. Auf der Seite findet der interessierte Nutzer die aktuelle Version des Simulators und der dazugehörigen virtuellen Treiber. Neben den üblichen Dokumentationen, die den Einsatz des Simulators beschreiben, findet der Anwender hier auch eine Liste mit häufig gestellten Fragen, sowie Kontaktmöglichkeiten zu den Personen hinter PersoSim. Das BSI und HJP Consulting informieren die Community auf der Website auch über Neuigkeiten und Updates. Die Website dient folglich als zentrale Einstiegsseite für das Projekt PersoSim und soll in erster Linie die Anwender adressieren. Auf der Website findet der Anwender eine Übersicht

über aktuelle eID-Clients mit zusätzlichen Informationen, aber auch Details zu den unterschiedlichen Varianten des Personalausweises, die bereits im Feld sind. Darüber hinaus werden an dieser Stelle auch weitere Open-Source-Projekte referenziert, die sich mit dem elektronischen Personalausweis beschäftigen.

4.3. Die Community der Entwickler

Neben der Community für die Anwender, wird auch eine Community für Entwickler aufgebaut (www.github.com/PersoSim). Interessierte Entwickler finden hier tiefere Informationen, die über den Betrieb des Simulators hinausgehen. An dieser Stelle kann ein Entwickler auch den Quellcode einsehen und herunterladen. Mit Informationen zum Kompilieren des Quellcodes soll der Entwickler in die Lage versetzt werden, den Code selbstständig zu übersetzen. Da der Simulator in Java implementiert ist, stellt das Kompilieren hier keine allzu große Herausforderung dar. Schwieriger ist hier schon das Übersetzen der Treiber für den virtuellen Kartenleser unter Microsoft Windows. Um unter Windows Treiber zu kompilieren, wird die kommerzielle Entwicklungsumgebung von Microsoft selbst benötigt. Testversionen der Entwicklungsumgebung sind online verfügbar. Im Projekt PersoSim werden für die Windows-Treiber alle benötigten Dateien zur Verfügung gestellt, so dass Entwickler diese auch selbst übersetzen können.

Um den Austausch unter den Entwicklern zu fördern, werden an dieser Stelle unterschiedliche Möglichkeiten geboten:

- Dokumentationen (Architektur des Simulators, Anweisungen zum Kompilieren usw.),
- Wiki (Kurze Beschreibungen für Funktionen usw.),
- Bug-Tracking (Erfassung und Verwaltung von Fehlern im Simulator bzw. in den Treibern),
- Coding-Guidelines (Förderung von einheitlichem Quellcode).

Der Entwickler-Community wird mit PersoSim also eine Infrastruktur zur Verfügung gestellt, auf der die Entwicklung des Simulators weiter vorangetrieben werden kann. Die Mitarbeiter der HJP Consulting übernehmen in der Community die Rolle des Moderators und steuern selbstverständlich auch den Quellcode für das Projekt bei.

5. Fazit und Ausblick

PersoSim bietet interessierten Anwendern und Entwicklern eine einfache und unbürokratische Möglichkeit, den elektronischen Personalausweis zu simulieren und damit eigene Implementierungen zu verifizieren. Darüber hinaus bietet der Simulator die Möglichkeit, die kryptographischen Protokolle, die auf dem integrierten Chip angewendet werden, zu verstehen und nachzuvollziehen.

Auch wenn der Simulator derzeit nur auf den Personalausweis beschränkt ist, wird er zukünftig problemlos auf andere Protokolle ausweitbar sein. So werden in naher Zukunft auch die Mechanismen, die bspw. in der eIDAS-Verordnung beschrieben sind, in PersoSim umgesetzt. Letztlich kann der Simulator alle Chipkartenprotokolle simulieren, die gängige Kommandos aus der ISO/IEC 7816 [7] verwenden. Auf diese Weise könnte eine universelle Open Source Plattform entstehen, mit der man zukünftig die verschiedensten Chipkartenapplikationen simulieren könnte. Eine derartige Plattform könnte die Einsatzmöglichkeit von Chipkarten noch weiter vergrößern und verbessern.

Literaturhinweise

- [1] BSI: Technical Guideline TR-03110: Advanced Security Mechanisms for Machine Readable Travel Documents, Part 1- 3
- [2] Tobias Senger, Holger Funke: An open source eID simulator. Proceedings of Open Identity Summit 2013, GI-Edition, 2013
- [3] Holger Funke, Tobias Senger: PersoSim. – Der Open Source Simulator für den elektronischen Personalausweis. DuD – Datenschutz und Datensicherheit 4/2014
- [4] Jens Bender, Dennis Kügler, Marian Margraf, Ingo Naumann: Sicherheitsmechanismen für kontaktlose Chips im deutschen elektronischen Personalausweis. DuD – Datenschutz und Datensicherheit 32(3), 2008
- [5] BSI: Worked Example for Extended Access Control, Version 1.02, 03.08.2011
- [6] BSI: Technical Guideline TR-03119: Requirements for Smart Card Readers supporting eID and eSign Based on Extended Access Control
- [7] ISO/IEC 7816 Identification cards - Integrated circuit cards
- [8] ISO/IEC 14443 Identification cards - Contactless integrated circuit cards - Proximity cards