Die Open eCard App für mehr Transparenz, Vertrauen und Benutzerfreundlichkeit beim elektronischen Identitätsnachweis

M. Horsch[1], D. Hühnlein[2], C. Breitenstrom[3], T. Wieland[5], A. Wiesmaier[4], B. Biallowons[2], D. Petrautzki[5], S. Potzernheim[5], J. Schmölz[2], A. Wesner[6], T. Wich[2]

Kurzfassung:
Mit der Einführung des neuen Personalausweises und der damit verbundenen Möglichkeit die starke Authentisierung und Signatur leicht in vielen Anwendungen nutzen zu können waren große Erwartungen verbunden. Eine breite Akzeptanz ist jedoch sowohl bei Diensteanbietern als auch bei Bürgerinnen und Bürgern bisher ausgeblieben. Zu den Ursachen hierfür scheinen die technischen Probleme sowie die mangelnde Funktionalität und Benutzerfreundlichkeit der AusweisApp zu zählen. Das Open eCard Projekt bietet mit der Open eCard App eine leichtgewichtige, vertrauenswürdige und benutzerfreundliche Alternative. Es werden bereits zahlreiche Plattformen unterstützt und hervorragende Erweiterungs- und Anpassungsmöglichkeiten geboten. Die Open eCard App setzt das eCard-API-Framework und die darin enthaltenen internationalen Standards um und integriert weitere innovative Ansätze auf dem Weg zum transparenten, vertrauenswürdigen und benutzerfreundlichen elektronischen Identitätsnachweis.

Stichworte: Open eCard App, neuer Personalausweis, eID-Funktion, eID-Applikation, eCard-API.

1. Einleitung

Der neue Personalausweis (nPA) setzt einen neuen Meilenstein bei hoheitlichen Dokumenten in Deutschland und Europa. Mit der eID- und eSign-Funktion sowie dem hohen Datenschutzstandard bietet er als Bestandteil der eCard-Strategie eine hervorragende Grundlage und großes Potenzial für vielseitige Anwendungen. Bei der Einführung kam es leider zu technischen und kommunikativen Problemen, die das Vertrauen beschädigt haben. Des Weiteren fällt die Anzahl der verfügbaren Anwendungen noch gering aus, sodass eine breite Nutzung bisher ausgeblieben ist.

Erforderlich sind daher Impulse und Anreize für Wirtschaft und Forschung, die neuen Technologien zu verwenden und weiter zu entwickeln. Das in diesem Dokument vorgestellte Open eCard Projekt möchte dazu einen Beitrag leisten.

Das vorliegende Dokument gliedert sich wie folgt: Kapitel 2 enthält eine kritische Würdigung der AusweisApp des Bundes und identifiziert Verbesserungspotenzial. Kapitel 3 stellt das Open eCard Projekt vor. Kapitel 4 beschreibt das Design und den Funktionsumfang der Open eCard App sowie den derzeitigen Entwicklungsstand. Eine Zusammenfassung und ein Ausblick finden sich abschließend in Kapitel 5.

[1] Technische Universität Darmstadt
[2] ecsec GmbH
[3] init AG
[4] AGT International
[5] Hochschule Coburg
[6] Wesner-IT-Service

2. Die AusweisApp des Bundes

Die AusweisApp des Bundes [1] wurde zusammen mit dem Start des nPA im November 2010 veröffentlicht und seitdem mehrfach aktualisiert. Inzwischen ist sie in der Version 1.9.0 (Stand Januar 2013) für die Plattformen Windows, Mac OS (ab 10.6) und verschiedene Linux-Derivate kostenfrei erhältlich. Sie zählt zu den am weitesten verbreiteten eID-Anwendungen für den nPA.

Trotz eines aufwändigen Marketingkonzepts ist es der Bundesregierung und den am nPA unmittelbar beteiligten Unternehmen und Institutionen bisher leider nicht gelungen, eine breite Akzeptanz des Ausweises zu schaffen. Weniger als jeder dritte Bürger, der eine neue Karte erhält, aktiviert auch die eID-Funktion. Gleichzeitig zeigt sich, dass es nur sehr wenige operative Diensteanbieter gibt. Laut „Kompetenzzentrum neuer Personalausweis“ (www.ccepa.de) sind es derzeit (Stand Januar 2013) weniger als 50 Anbieter, viele davon Behörden. Von einer breiten Integration der eID-Funktion durch die Wirtschaft kann daher nicht die Rede sein – zumal die großen Internetdienstanbieter wie eBay, Amazon, Facebook, Google etc. nicht darunter sind. Die Vorstellungen und Wünsche haben sich daher bisher leider nicht erfüllt.

Die aktuelle Situation hat daher deutliches Verbesserungs- und Optimierungspotenzial. Sie könnte zum einen durch neue Zugangsformen, zum anderen durch die Bereitstellung neuer Dienste signifikant verbessert werden. Dazu ist jedoch die AusweisApp in ihrer gegenwärtigen Form nicht in der Lage und verhindert sogar die Erschließung neuer Anwendungsfelder durch Privatleute und Unternehmen.

Die immer stärkere Bedeutung von mobilen Geräten und mobilen Anwendungen erfordert eine mobile Auslegung der eCard-Strategie. Insbesondere weil deren Ziele, wie beispielsweise eine sichere und vertrauenswürdige Authentisierung, auch im mobilen Umfeld von hohem Interesse sind. Die NFC-Technologie ermöglicht dabei die kontaktlose Kommunikation mit Chipkarten im mobilen Umfeld. Auch wenn derzeit noch Probleme in Verbindung mit dem nPA auftreten[7] und die eGK wegen der kontaktbehafteten Schnittstelle nicht direkt genutzt werden kann, lassen sich bereits Kartenleser mit USB- oder Bluetooth-Schnittstelle mit mobilen Geräten verbinden. Die Unterstützung mobiler Plattformen gewinnt daher immer weiter an Bedeutung, insbesondere weil sich neben Smartphones auch Tablet-Computer immer stärker verbreiten.

Auch im stationären Bereich besteht durch die schleppende Unterstützung von Apples Mac OS Handlungsbedarf. Gerade dieser Missstand hält eine Reihe namhafter Internet-Unternehmen vom Einsatz der eID-Funktion ab. Hinzu kommen technische Mängel wie beispielsweise die schleppende Umsetzung aktualisierter Spezifikationen oder die fehlende Unterstützung von TLS 1.1. Dies führt neben Sicherheitsrisiken durch bekannte Angriffe auf TLS 1.0, auch zu einem „zweiten Ökosystem“. Das heißt, die weiteren Teilnehmer der eID-Infrastruktur müssen Anpassungen vornehmen, damit die Interaktion der Komponenten funktioniert.

[7] Siehe z.B. https://www.openecard.org/de/framework/extendedlength

Weitere Anwendungsbereiche ließen sich eröffnen, wenn auch die eSign-Funktion des nPA oder anderer Chipkarten der eCard-Strategie (eGK, HBA, Bank- und Signaturkarten) nutzbar wäre. Zwar gibt es dafür bereits separate Anwendungen, doch würde es sich für die Nutzer deutlich leichter gestalten, wenn eID- und eSign-Funktion durch eine einzige Anwendung bereitgestellt werden würden. Dies gilt auch für die weiteren Funktionen der eGK, die derzeit von den Versicherten mangels freier Software-Lösungen – und natürlich mangels einer PIN von ihrer Krankenkasse – überhaupt nicht zur Verfügung stehen. Insbesondere mit Hinblick auf die Akzeptanz der Nutzer ist ein reibungsloser und flächendeckender Nutzungsweg wichtig.

Schließlich ist zu beachten, dass es sich bei der AusweisApp des Bundes zwar um eine kostenlose, aber keineswegs – im Gegensatz zu früheren Ankündigungen – quelloffene Software handelt. Wie bei allen Anwendungen im Bereich der IT-Sicherheit ist eine vertrauenswürdige Sicherheit erst dann gegeben, wenn der Quellcode von unabhängigen Experten überprüft werden kann. Zudem bieten, neben einer stärkeren Vertrauensbasis und einer besseren Modifizierbarkeit, Open-Source-Projekte einen hohen Anreiz und Impulse für Innovation und Forschung. Das große Potenzial der eCard-Strategie könnte mit offenen Standards und offener Software besser unterstützt werden.

3. Das Open eCard Projekt

3.1. Überblick

Im Open eCard Projekt haben sich industrielle und akademische Experten zusammengefunden, um eine quelloffene und plattformunabhängige Implementierung des eCard-API-Frameworks [2] bereitzustellen, durch die beliebige Anwendungen für Zwecke der Authentisierung und Signatur leicht auf beliebige Chipkarten zugreifen können.

In einer ersten Projektphase soll dieses Rahmenwerk für die Realisierung einer leichtgewichtigen, vertrauenswürdigen und gleichsam gut bedienbaren Alternative zur AusweisApp des Bundes – der in diesem Beitrag vorgestellten Open eCard App (siehe auch [3]) – genutzt werden.

3.2. Möglichkeiten zur Mitwirkung

Das Open eCard Team ist eine **offene Gemeinschaft**, die alle interessierten Bürgerinnen und Bürger sowie entsprechende Institutionen und Verbände zur Mitwirkung aufruft. Wer die Entwicklung der Open eCard App aktiv unterstützen möchte, kann sich unter http://openecard.org/join registrieren.

Wer lediglich über Neuigkeiten aus der Open eCard Community informiert werden möchte, kann unter http://openecard.org/mail entsprechende Mailinglisten bzw. unter http://openecard.org/news entsprechende Newsfeeds bestellen. Sonstige Anregungen nimmt das Open eCard Team gerne unter feedback@openecard.org entgegen.

3.3. Qualitätsmanagement

3.3.1. Besondere Problematik

Die Open eCard App wird für verschiedene Plattformen ausgeliefert und in unterschiedlichsten Kontexten durch Menschen mit unterschiedlichster IT-Prägung verwendet. Die Qualität elektronischer Identitätsdokumente ist nicht auf Sicherheit im Sinne von IT-Sicherheit zu reduzieren: Die Qualität erfordert die Betrachtung der Ebenen Semantik/Usability, Code-Sicherheit und kryptografische Verfahren.

In der Literatur ist hinlänglich auf die subjektive Sicherheit als Akzeptanzfaktor hingewiesen worden. Die Akzeptanz des nPA ist neben den klassischen Kriterien wie „gefühlter Vorteil gegenüber der nächstschlechteren Lösung“, „geringe Komplexität“, „Ausprobierbarkeit“, „Sichtbarkeit“ etc., vgl. [4] essentiell von der subjektiven Sicherheit der Verwendung abhängig. Jeder in den Medien dokumentierte Fall eines Missbrauchs der eID-Funktion oder – noch schlimmer – einer Signatur bedeutet schwindende Akzeptanz.

Die Verfügbarkeit des Quellcodes ermöglicht es, die Integrität der kompilierten Software zu prüfen. Damit wird ein bisher unerreichtes Vertrauensniveau erreicht. Zugleich kann der Quellcode aber auch missbraucht werden, um eine täuschend echte eID Applikation zu konstruieren, die im Hintergrund ungewollte Aktionen auslöst.

Viele weitere Aspekte führen zu dem Schluss, dass Qualitätssicherung auf mehreren Ebenen erfolgen muss. Im Folgenden sind die Methoden und die Organisationsstruktur des Open eCard App Projektes dargelegt, die den Bedrohungen entgegenwirken.

3.3.2. Methodik

Um auf der Ebene der Semantik/Usability erfolgreich zu sein, muss eine qualitative Analyse mit unterschiedlichen Nutzergruppen bei vorhandenen Dienstanbietern erfolgen. Die Kommunikationsmuster, die „Ansprache“ des Nutzers, und der Informationsaustausch sind hier entscheidend. Folglich ist es Aufgabe eines Usability-Expertenteams im Projekt, einen repräsentativen Querschnitt an Nutzern in unterschiedlichen Anwendungsfällen zur Mitarbeit bei der Bewertung der Releases zu gewinnen. Ein öffentliches Zielkundenprofil, ein Bewertungsprotokoll einschließlich der Tests und Ergebnisse sind Resultate dieser Aktivitäten.

Die Sicherheit des verwendeten Codes wird durch bestehende Best-Practices in der Open Source Entwicklung erreicht. Dazu zählen

- Programmierkonventionen einschließlich wirksamer Mechanismen,
- Trennung der Verantwortlichkeiten (Entwickler erstellen Code, den besonders erfahrene Entwickler (Committer) prüfen und in das Repository übernehmen),
- Regelmäßige statische Quellcode-Analyse und daran anschließende Bugfixes, welche das objektive und kontinuierliche Messen der Reife ermöglichen,
- Unit-Tests und Mittel, die Testabdeckung zu prüfen sowie
- ein für die kontinuierliche Integration geeigneter Build-Prozess.

Die Summe dieser Maßnahmen erreicht Werte von 0,1 Fehlern pro 1000 Zeilen Quellcode, was vergleichbar bzw. besser ist, als kommerzielle Software (vgl. [5]). Der Schlüssel liegt hier im kontinuierlichen Verbesserungsprozess.

Die korrekte Anwendung kryptografischer Protokolle und Mechanismen wird durch Peer-Review erreicht. Dies erfordert, wie auch bei der Entwicklung, hochspezialisierte Informatiker, deren kritisches Hinterfragen der Implementierung durch nichts zu ersetzen ist. Der Nutzen des Peer-Reviews ist umso größer, je ernsthafter der Reviewer versucht, Lücken zu finden. Dies wird durch sogenannte „Hostile-Reviews“ erreicht.

3.3.3. Organisation

Im Open eCard Projekt wird ein Security-Team etabliert, das folgende Aufgaben wahrnimmt:

- Entgegennehmen, Verarbeiten und Schließen von gemeldeten Sicherheitslücken,
- Hilfestellung für die Maintainer von Modulen bei der Beseitigung von Sicherheitslücken,
- Erstellung und Pflege der Programmierkonventionen und
- Erstellen von Dokumentation zu sicheren Integrationsmöglichkeiten.

Das Open eCard Security-Team besteht aus anerkannten industriellen und akademischen Experten.

4. Die Open eCard App

4.1. Grundlegendes Design

Vor dem Hintergrund der existierenden Spezifikation des eCard-API-Framework [2], der in [3] näher erläuterten Anforderungen und den im Zuge der verschiedenen Implementierungen [1], [6], [7], [8], [9], [10] und [11] gesammelten Erfahrungen wurde der in Abbildung 1 dargestellte Architektur-Entwurf für die Open eCard App entwickelt. Durch den hochgradig modularen Ansatz und die plattformunabhängige, Java-basierte Realisierung der Kernmodule kann die Open eCard App leicht erweitert und auf unterschiedlichen Plattformen (z.B. Windows, Linux, Mac OS, Android etc.) eingesetzt werden.

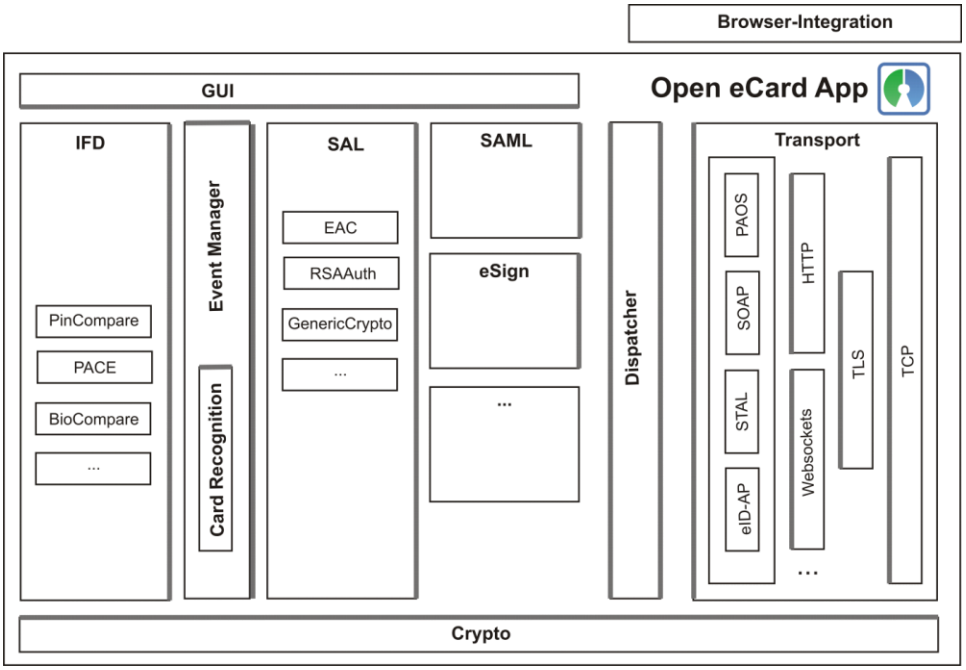

Abbildung 1: Die Architektur der Open eCard App

Die wesentlichen Module der Open eCard App sind im Folgenden näher erläutert:

- **Interface Device (IFD)**
  Diese Komponente implementiert das IFD-Interface, das in TR-03112-6 [2] und ISO/IEC 24727-4 [12] spezifiziert ist. Weiterhin enthält es zusätzliche Schnittstellen für Passwort-basierte Protokolle (z.B. PACE) und stellt eine einheitliche Schnittstelle für die Kommunikation mit unterschiedlichen Kartenlesern und Smartcards zur Verfügung.

- **Event Manager**
  Der Event Manager überwacht eintreffende Events (z.B. das Hinzufügen oder Entfernen von Lesegeräten oder Karten) und führt die Typerkennung von neu hinzugefügten Karten durch. Da für die Erkennung des Chipkartentyps die in CEN 15480 [13] bzw. ISO/IEC 24727 [12] spezifizierten CardInfo-Dateien genutzt werden, kann die Open eCard App sehr leicht zusätzliche Chipkarten unterstützen. Die derzeit unterstützten Chipkarten sind unter http://openecard.org/ecards dargestellt und entsprechende Demonstratoren finden sich unter http://openecard.org/demo, http://demo.skidentity.de.

- **Service Access Layer (SAL)**
  Dieses Modul implementiert den Service Access Layer wie er in TR-03112-4 [14] und ISO/IEC 24727-3 [12] spezifiziert ist. Ein wichtiger Aspekt dieser Komponente ist sein Erweiterungsmechanismus, welcher das Hinzufügen neuer Authentisierungsprotokolle ermöglicht, ohne andere Teile der Implementierung ändern zu müssen.

- **Crypto**
  Das Crypto-Modul vereinheitlicht den Zugriff auf kryptographische Funktionen für andere Module. Durch die Nutzung der BouncyCastle Crypto Bibliothek [15] ist es möglich die Open eCard App leicht auf Plattformen ohne vollständige Implementierung der Java Cryptographic Architecture (JCA) [16] zu portieren.

- **Graphische Benutzerschnittstelle (GUI)**
  Die GUI wird über eine abstrakte Schnittstelle angesprochen und ist jederzeit leicht gegen andere Implementierungen austauschbar. Dies ermöglicht Plattform-spezifische Implementierungen ohne eine Änderung anderer Module.

- **Security Assertion Markup Language (SAML)**
  Diese Komponente bietet Unterstützung für erweiterte SAML-Profile [17], und Bindings [18], die zu einem effizienteren und besser geschützten Authentisierungsablauf führen.

- **Elektronische Signaturen (eSign)**
  Diese Komponente ermöglicht die Erzeugung fortgeschrittener elektronischen Signaturen nach [19], [20], und [21]. Die Komponente implementiert eine an TR-03112-2 [2] angelehnte und auf OASIS DSS [22] basierende Schnittstelle.

- **Dispatcher**
  Der Dispatcher stellt eine zentrale Komponente dar, die alle ein- und ausgehenden Nachrichten an die entsprechenden Module weiterleitet. Diese Zentralisierung ermöglicht eine deutliche Reduzierung der Gesamtkomplexität, zugleich ist es aber auch möglich zusätzliche Logik, die den Datenfluss steuert, einzubringen. Dadurch sind u. A. Filter für externe Nachrichten realisierbar.

- **Transport**
  Diese Komponente kapselt die individuellen Transportprotokolle auf den unterschiedlichen Schichten. Durch diese Architektur ist es leicht, verschiedene Protokolle zu nutzen und weitere hinzuzufügen. Um die momentan vorhandenen eID-Server bedienen zu können, unterstützt die Open eCard App den Austausch von PAOS-basierten Nachrichten [23] per HTTP und die gesicherte Übertragung mit TLS. Das Design ist aber bewusst so entworfen, dass auch beliebige andere Bindings (z.B. SOAP [24]) und alternative Protokolle wie der österreichische Security Token Abstraction Layer (STAL) [25] oder das belgische eID-Applet Protokoll [26] unterstützt werden können.

- **Browser Integration**
  Damit die Open eCard App nicht nur in Verbindung mit fest installierten Fachanwendungen, sondern auch in Browser-gestützten Webanwendungen genutzt werden kann, wird der in TR-03110-7 [2] spezifizierte, HTTP-basierte eID-Aktivierungsmechanismus unterstützt. Längerfristig ist die tiefere Integration der Open eCard App in populäre Browser über die von diesen unterstützten kryptografischen Schnittstellen (z.B. PKCS#11 [27] oder CSP [28]) anvisiert.

4.2. Eigenschaften und Stärken

4.2.1. Plattformunabhängigkeit

Die Open eCard App wird in Java entwickelt und lässt sich neben den stationären Betriebssystemen wie Windows, Mac OS und Linux auch auf mobilen Endgeräten, auf Basis der Android Plattform, einsetzen. Damit wird bereits ein breites Feld an Plattformen und Geräten von der Open eCard App unterstützt. Des Weiteren verzichtet die Open eCard App auf browser-spezifische Plug-ins und setzt ganz auf die alternative eID-Aktivierung per Localhost gemäß TR-03112-7 [2]. Damit werden alle gängigen stationären und mobilen Browser unterstützt.

Die breite Unterstützung von verschiedenen Plattformen und die Unabhängigkeit von Browser-Plug-ins bringen viele Vorteile für Anwender und Entwickler. Anwender können die Anwendung auf der Mehrzahl ihrer Geräte einsetzen und das unabhängig von ihrer bevorzugten Web-Browser Anwendung. Die hohe Wiederverwendbarkeit des Quellcodes reduziert die Fehleranfälligkeit und ermöglicht eine schnelle Erweiterung und Integration von neuen Funktionen.

4.2.2. Client Control Interface

Die Aktivierung von eID-Applikationen über Browser-Plug-ins ist mit erheblichen Nachteilen verbunden. Neben der Entwicklung und Wartung der Plug-ins ist auch eine breite Unterstützung aller Browser nur schwer umzusetzen. Die Open eCard App setzt bereits auf die alternative eID-Aktivierung per Localhost gemäß TR-03112-7. Zusätzlich stellt sie mit dem sogenannten Client Control Interface (CCI) eine Schnittstelle zum Zugriff auf Informationen und Steuerungsmechanismen bereit.

Dabei geht es insbesondere um die essentielle Frage, ob eine eID-Anwendung installiert ist und darüber hinaus welche Funktion und Karten erkannt und unterstützt werden. Zusätzlich können aktuelle Statusinformationen bzw. -änderungen abgefragt werden, wie beispielsweise die erkannten Chipkarten. Das Ziel des CCI ist es, den Benutzer bestmöglich bei der Auswahl des Authentisierungsmechanismus und der Nutzung der Anwendung zu unterstützen. Das heißt beispielsweise, dass ein Service die verfügbaren Informationen über Karten und Funktionen abrufen kann und dem Benutzer dann die beste Authentisierungsmöglichkeit anbieten kann.

Technisch wird dies durch JavaScript in Verbindung mit dem Cross-Origin Ressource Sharing (CORS) [29] Mechanismus eines Web-Browsers umgesetzt.

4.2.3. Benutzerinteraktion

Die Benutzerfreundlichkeit einer Anwendung ist ein wichtiger Bestandteil für die Akzeptanz. Die subjektive Wahrnehmung lässt sie sich aber nur schwer messen und beziffern. Die Open eCard App gestattet durch die definierte GUI Schnittstelle (vgl. Kapitel 4.1) die einfache Entwicklung beliebiger grafischer Oberflächen, ohne Anpassungen an den Sicherheits- oder Kommunikationsprotokollen vorzunehmen. Durch die modulare Architektur und die Offenlegung des Quellcodes kann die GUI leicht an spezifische Bedürfnisse angepasst werden.

Die Open eCard App bietet auf Basis der CardInfo-Files (CIF) eine robuste und sehr einfach zu erweiternde Kartenerkennung. Bei der stationären Variante wird die Erkennung durch graphische Elemente unterstützt und dem Benutzer erkannte Kartenleser und Karten inklusive entsprechender Abbildungen angezeigt (siehe Abbildung 2).

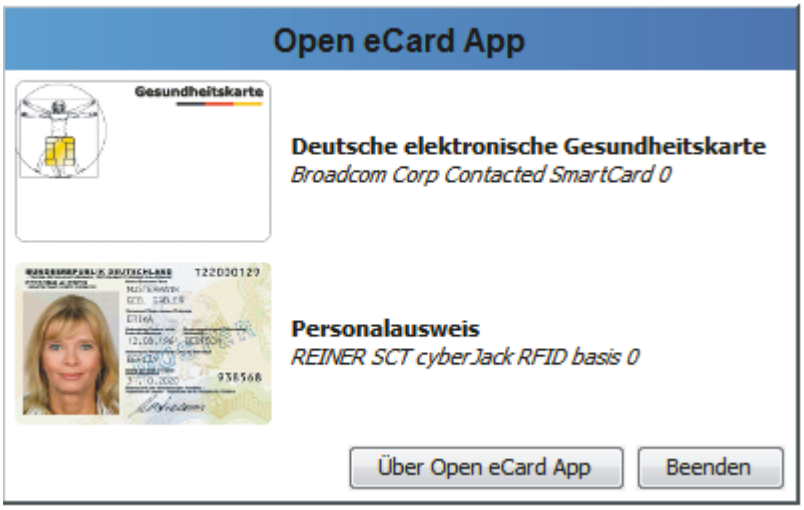

Abbildung 2: Grafische Rückmeldung der Kartenerkennung

4.2.4. Universelle Authentisierungsmechanismen

Neben einer breiten Unterstützung verschiedener Plattformen ist auch die einfache Erweiterung, wie beispielsweise die Integration neuer Authentisierungs-mechanismen, von großer Bedeutung.

Die Open eCard App unterstützt bereits CardInfo-Files (CIF) gemäß CEN 15480-3 [13] und ISO/IEC 24727-3 [12]. Ein CIF beschreibt, in einer fest definierten Struktur, die Eigenschaften und die Funktionen einer Chipkarte bzw. eines Credentials. Die Unterstützung einer neuen Karte durch die Open eCard App lässt sich bequem durch das Hinzufügen des entsprechenden CIF realisieren. Die CIFs dienen ebenfalls als Grundlage für die Kartenerkennung.

Durch die Ausrichtung der Architektur der Open eCard App an das eCard-API-Framework bestehen bereits die nötigen Datenstrukturen und Interfaces zur Integration

weiterer Authentisierungsprotokolle. Zum jetzigen Entwicklungsstand sind auf SAL-Ebene die in TR-03112-7 [30] aufgeführten Protokolle EAC, Generic Cryptography und PIN Compare implementiert. Die Selektion erfolgt dabei über die in den eCard-API Nachrichten angegebene Protocol-URI.

Für die weitere Entwicklung wird eine Unterstützung der Modellierung und Integration in einer formalen Sprache definierten Authentisierungsprotokolle angestrebt.

4.3. Stand der Entwicklung

Die Open eCard App basiert auf dem in Kapitel 4.1 vorgestellten Design. Die App unterstützt bereits eine Kartenerkennung auf Basis von CardInfo-Dateien, eine Authentisierung via eID-Funktion mit dem nPA und eine prototypische zertifikatsbasierte TLS-Client-Authentisierung, die beispielsweise mit der eGK oder den durch CardInfo-Dateien beschriebenen Signaturkarten genutzt werden kann. Die Open eCard App wird dabei in folgenden Ausführungen bereitgestellt:

4.3.1. Browser Applet

Das Java Applet ist als Demonstrator unter https://www.openecard.org/de/demo/applet verfügbar.

4.3.2. Rich Client

Der Java-basierte Rich Client kann unter https://openecard.org/de/download/pc als .jar-Datei heruntergeladen und gestartet werden. Darüber hinaus sind eine portable Installation[8] sowie Plattform-spezifische Installer für die Betriebssysteme Windows, Mac OS und Linux konzipiert vorgesehen.

4.3.3. Java Web Start

Der Java-basierte Rich Client kann auch über Java Web Start Mechanismen https://www.openecard.org/de/demo/webstart heruntergeladen und genutzt werden.

4.3.4. Android

Die Android-basierte Open eCard App befindet sich auf einem mit den stationären Lösungen vergleichbaren Entwicklungsstand. Dies beinhaltet insbesondere den automatischen Start der Anwendung via Link (Localhost und TCToken) aus dem Browser heraus, sowie die Durchführung einer Authentisierung mittels nPA und einem geeigneten, für die Unterstützung von Extended Length APDUs vorbereiteten, NFC-fähigen Smartphone oder einer zertifikatsbasierten TLS-Client-Authentisierung mit der eGK und einem extern angeschlossenen Kartenterminal.

[8] Gemäß der von http://portableapps.com bereitgestellten Infrastruktur.

4.4. Ausgewählte Anwendungsfälle

Die Vorteile einer Zwei-Faktor-Authentisierung, beispielsweise durch eine Chipkarte und der dazugehörigen PIN, sind bekannt. Neben einer reinen Fokussierung auf eine Online-Authentisierung durch den nPA, ist es denkbar, die Open eCard App stärker in das Betriebssystem zu intergieren und so beispielsweise auf das unter Linux genutzte PAM (Pluggable Authentication Modules) bzw. den Microsoft-spezifischen Login-Mechanismus für Windows zurückzugreifen. Das Ziel ist hierbei eine chipkartenbasierte Anmeldung am Betriebssystem mittels der Open eCard App.

Insbesondere die Integration und Interaktion mit existierenden Infrastrukturen und Projekten ist ein wichtiger Punkt, um die Verbreitung und Akzeptanz zu stärken. Dazu zählt beispielsweise die Interaktion mit GnuPG, um das Signieren und Verschlüsseln von Mails mittels der Open eCard App für beliebige Chipkarten zu ermöglichen. Denkbar ist auch die Interaktion mit TrueCrypt, um die Passwörter zur Datenverschlüsselung oder die „Keyfiles“ auf einer Chipkarte zu speichern.

Wünschenswert ist es ebenfalls, dass der Bund, wie beispielsweise beim Kraftfahrt-Bundesamt geschehen, die eID-Infrastruktur weiter in die eigenen Progresse intergiert. Beispielsweise könnte die Authentisierung, Signatur und Verschlüsselung bei der eVergabe, um die eID-Infrastruktur erweitert werden. Durch die Schaffung neuer Einsatzmöglichkeiten könnte der Bund die eigene eCard-Strategie stärken.

5. Zusammenfassung und Ausblick

Die innovativen Technologien des neuen Personalausweises haben leider noch keine breite Akzeptanz erfahren. Verunsicherte Anwender, geringe Einsetzbarkeit und technische Probleme sind nur einige Bestandteile der Problematik. Das Open eCard Projekt möchte mit der Open eCard App daher Impulse und Anreize für Wirtschaft und Forschung sowie neues Vertrauen bei Anwendern schaffen.

Die Open eCard App bietet durch ihr fundiertes und an etablierten Standards ausgerichtetes Design eine hervorragende und robuste Grundlage für die Entwicklung einer universellen, modularen und benutzerfreundlichen eID-Applikation. Weitere Protokolle können bequem auf den passenden Ebenen wie IFD, SAL oder Transport hinzugefügt werden und neue Chipkarten können leicht anhand der dazugehörigen CardInfo-Files integriert werden. Die Open eCard App unterstützt bereits Windows, Linux, Mac OS und Android und ist durch die eID-Aktivierung per Localhost mit beliebigen Browsern verwendbar.

Die weiteren Entwicklungen der Open eCard App betreffen insbesondere die qualitätsgesicherte Integration der TLS-basierten Authentisierung, einer erweiterten Internationalisierung sowie der Unterstützung der Signaturfunktion. Des Weiteren ist in der Open eCard App eine effiziente und sichere SAML-Unterstützung sowie längerfristig die Unterstützung Attribut-basierter Credentials geplant.

6. Literatur

[1] Bundesamt für Sicherheit in der Informationstechnik (BSI), *AusweisApp,* https://www.ausweisapp.bund.de.

[2] Bundesamt für Sicherheit in der Informationstechnik (BSI), *eCard-API-Framework,* Technical Guideline TR-03112, Part 1 - 7, Version 1.1.2, https://www.bsi.bund.de/ContentBSI/Publikationen/TechnischeRichtlinien/tr03112/index_htm.html.

[3] D. Hühnlein, D. Petrautzki, J. Schmölz, T. Wich, M. Horsch, T. Wieland, J. Eichholz, A. Wiesmaier, J. Braun, F. Feldmann, S. Potzernheim, J. Schwenk, K. C., A. Kühne und H. Veit, *On the design and implementation of the Open eCard App,* GI SICHERHEIT 2012 Sicherheit - Schutz und Zuverlässigkeit, März 2012.

[4] E. M. Rogers und E. Rogers, *Diffusion of Innovations 5th Edition,* Free Press, 2003.

[5] Coverity, *Coverity Scan: 2011 Open Source Integrity Report,* http://www.coverity.com/library/pdf/coverity-scan-2011-open-source-integrity-report.pdf.

[6] Ageto Innovation GmbH, *Ageto AusweisApp,* http://www.ageto.de/egovernment/ageto-ausweis-app .

[7] bos GmbH & Co. KG, *Governikus Autent,* http://www.bos-bremen.de/de/governikus_autent/1854605.

[8] M. Frank, O. Dominik und M. Wolf, *OpenPACE Project - Crypto library for the PACE protocol,* http://openpace.sourceforge.net.

[9] P. Dirk, *Security of Authentication Procedures for Mobile Devices,* Master Thesis, Hochschule Coburg, 2011.

[10] M. Horsch, *MobilePACE - Password Authenticated Connection Establishment implementation on mobile devices,* Bachelor Thesis, TU Darmstadt, http://www.cdc.informatik.tu-darmstadt.de/mona/pubs/200909_BA_MobilePACE.pdf, 2009.

[11] M. Horsch, *Mobile Authentisierung mit dem neuen Personalausweis (MONA),* Master Thesis, TU Darmstadt, https://www-old.cdc.informatik.tu-darmstadt.de/reports/reports/Moritz_Horsch_MONA.master.pdf, 2011.

[12] ISO/IEC, *Identification cards - Integrated circuit card programming interfaces,* ISO/IEC 24727, Part 1 - 5.

[13] European Committee for Standardization (CEN), Identification card systems - European Citizen Card, Part 1 - 4, CEN/TS 15480, 2008.

[14] Bundesamt für Sicherheit in der Informationstechnik (BSI), *eCard-API-Framework - ISO 24727-3-Interface,* Technical Guideline TR-03112-4, Version 1.1.2, 2012.

[15] The Legion of the Bouncy Castle, „Bouncy Castle Crypto API," [Online]. Available: http://www.bouncycastle.org/java.html.

[16] Oracle, *Java Cryptography Architecture (JCA) Reference Guide,* http://download.oracle.com/javase/6/docs/technotes/guides/security/crypto/CryptoSpec.html .

[17] OASIS, *SAML V2.0 Enhanced Client or Proxy Profile Version 2.0,* https://www.oasis-open.org/committees/download.php/41209/sstc-saml-ecp-v2.0-wd02.pdf, 2011.

[18] OASIS, *SAML V2.0 Holder-of-Key Web Browser SSO Profile Version 1.0,* http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso.pdf, 2010.

[19] ETSI, *CMS Advanced Electronic Signatures (CAdES),* ETSI TS 101 733, Version 1.8.1, 2009.

[20] ETSI, *PDF Advanced Eleltronic Signature Profies,* TSI TS 102 778, Part 1 - 5, 2009.

[21] ETSI, *Technical Specification XML Advanced Electronic Signatures (XAdES),* ETSI TS 101 903, Version 1.4.1, 2009.

[22] OASIS, *Digital Signature Service Core Protocols, Elements, and Bindings,* Version 1.0, OASIS Standard, http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.pdf .

[23] Liberty Alliance Project, *Liberty Reverse HTTP Bindin for SOAP Specification,* Version 2.0, http://www.projectliberty.org/liberty/content/download/909/6303/file/liberty-paos-v2.0.pdf.

[24] B. Don, E. David, K. Gopal, L. Andrew und M. Noah, Simple Object Access Protocol (SOAP) 1.1, 2000.

[25] *MOCCA: Modular Open Citizen Card Architecture Project,* http://mocca.egovlabs.gv.at.

[26] C. Frank, *eID-Applet Project,* http://code.google.com/p/eid-applet/.

[27] RSA Laboratories, *PKCS #11 Base Functionality v2.30: Cryptoki,* 2009: http://www.cryptsoft.com/pkcs11doc/STANDARD/pkcs-11v2-30b-d5.doc.

[28] Microsoft, „Cryptographic Service Providers," [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/aa380245%28v=vs.85%29.aspx.

[29] W3C, *Cross-Origin Resource Sharing,* W3C Working Draft 3, http://www.w3.org/TR/cors, 2012.

[30] Bundesamt für Sicherheit in der Informationstechnik (BSI), *eCard-API-Framework - Protocols,* Technical Guideline TR-03112-7, Version 1.1.2, 2012.