AusweisApp2

- Change -

„SDK stationär“

Change-Nummer: 26

Version: 1.0

Status: Final

Stand: 12.04.2018

Änderungshistorie

Version	Datum	Änderung	Bearbeiter
0.1	13.02.2018	Initialer Entwurf	[redacted]
0.2	09.03.2018	Überarbeitung	[redacted]
0.3	28.03.2018	Ergänzung	[redacted]
0.4	11.04.2018	Feinschliff	[redacted]
0.5	11.04.2018	Ergänzungen	[redacted]
0.6	12.04.2018	Korrekturen	[redacted]
1.0	12.04.2018	Finale Version	[redacted]

Inhalt

1 Anforderungen

Welche Anforderungen haben dazu geführt, diesen Change zu formulieren und seine Realisierung anzubieten?

Nach dem Vorbild der Realisierung des mobilen SDK der AusweisApp2 (Changes #15 und #17) soll ein SDK für Integration der Online-Ausweisfunktion in stationäre Anwendungen über die stationäre Version der AusweisApp2 erfolgen.

Mit diesem Change wird ein teilintegriertes SDK zum Einsatz auf den unterstützten stationären Betriebssystemen (OS X bzw. MacOS und Windows) realisiert. Teilintegriert bedeutet in diesem Zusammenhang, dass zur Nutzung einer in Geschäftsprozesse integrierten Online-Ausweisfunktion immer eine AusweisApp2 auf dem System des Benutzers installiert sein muss. Für diesen Change muss eine Anpassung der Software erfolgen sowie eine Entwicklerdokumentation für Integratoren angefertigt werden.

Die Schnittstelle soll hierbei HTTP/Websocket über den vorhandenen localhost-Webserver der AusweisApp2 verwenden (vgl. [4]). Der Aufruf der SDK-Funktionalität bzw. der Start der Websocket-Verbindung soll über die ClientURL ws://127.0.0.1:24272/eID-Kernel erfolgen.

Die für die Kommunikation verwendeten Kommandos sollen sich möglichst vollständig an der mobilen SDK-Schnittstelle orientieren (vgl. [5]). Es wird eine Schnittstellenspezifikation erstellt bzw. die vorhandene Spezifikation entsprechend erweitert.

Wie bei der Umsetzung des mobilen SDK verbleibt die AusweisApp2 bei Aufruf im Hintergrund und die Steuerung und PIN-Eingabe erfolgt über das Frontend des aufrufenden Programms.

Eine bereits gestartete Instanz der AusweisApp2 wird auch für SDK-Zugriffe genutzt. Läuft noch keine Instanz der AusweisApp2, so kann ermöglicht werden, dass diese vorab durch das aufrufende Programm automatisiert gestartet wird.

Mehrfache Aufrufe der AusweisApp2 zur Authentisierung oder PIN-Verwaltung müssen abgefangen werden. Für einen eID-Client Aufruf aus dem Browser heraus wird das bisherige Verhalten (Fehlerwebseite durch AusweisApp2) verwendet. Die Benutzeroberfläche der AusweisApp2 wird darüber hinaus mit einem „Sperrbildschirm“ überlagert, während ein SDK-Zugriff erfolgt. An dieser Stelle soll auch der Name der aufrufenden App aus deren User-Agent angezeigt werden. In der Dokumentation sind die Integratoren darauf hinzuweisen, dass sie den User-Agent entsprechend korrekt und sinnvoll setzen müssen.

2 Realisierung

Mit Realisierung ist das zu erstellende Werk gemeint, das kann ein Software-Modul oder auch ein Konzept sein. Im Abschnitt 3.1 erfolgt eine Leistungsbeschreibung, es folgen die Aufwandsabschätzung (3.2) und der Zeitplan (3.3).

2.1 Beschreibung der Realisierung

Ziel dieses Changes ist die Bereitstellung eines stationären SDK für Integratoren zum Einbinden der Online-Ausweisfunktion über die AusweisApp2 in eigene Prozesse. Die nachfolgende Beschreibung setzt voraus, dass auf dem Rechner des Benutzers eine installierte Version der AusweisApp2 vorhanden ist. Für die Realisierung dieses stationären SDK kann die Kern-Funktionalität des mobilen SDKs übernommen werden.

Der bereits von der AusweisApp2 unter 127.0.0.1:24727 betriebene Webserver wird um den Pfad „/eID-Kernel“ erweitert.

Den auf diesem Pfad anfragenden Clients wird mit einem Upgrade auf das Websocket-Protokoll geantwortet. Die nachfolgende SDK-Kommunikation wird anschließend somit über das Websocket-

Protokoll erfolgen. Die Entscheidung, an dieser Stelle auf das Websocket-Protokoll zu setzen, ist gefallen, da es verbreitete und gut getestete Implementierungen sowohl in Webbrowsern als auch in Bibliotheken wie Qt gibt. Diese gestalten den Nachrichtenaustausch für die nutzenden Anwendungen simpel und robust, sind daher gut zu warten und weniger anfällig für Fehler.

Für die SDK-Kommunikation kann das Protokoll der mobilen SDK-Version übernommen werden. Dies reduziert die Menge des in der AusweisApp2 zu pflegenden und zu testenden Codes drastisch und vereinfacht die Einarbeitung für Integratoren, die beide SDK-Varianten unterstützen wollen.

Eine Mehrfachauthentisierung, beispielsweise durch gleichzeitiges Nutzen einer SDK-basierten Online-Ausweisfunktion und einem Standardaufruf über das Internet, wird verhindert. Es wird verhindert, dass sich Workflows – also Authentisierungen oder PIN-Änderungen – die über die UI der AusweisApp2 oder den lokalen Webserver gestartet werden mit denen über die SDK-Schnittstelle gestarteten Workflows überlagern.

Wird eine SDK-Verbindung aufgebaut, so zeigt die UI der AusweisApp2, sofern sie vom Benutzer geöffnet wird, die Fehlermeldung „Die Anwendung <Anwendungs-Name> verwendet bereits die SDK-Schnittstelle.“ an und die Verwendung der AusweisApp2 ist nicht möglich. Die Funktionen der AusweisApp2 sind dabei ausgegraut (vgl. Abbildung 1). Der Anwendungs-Name kann hierbei dem in SDK-Verbindung verwendeten „User Agent“-String entnommen werden. Wird in diesem Zustand ein Workflow über den lokalen Webserver http://127.0.0.1:24727/eID-Client gestartet, so wird diese Anfrage mit einem Fehler beantwortet, analog zu dem bisherigen Verhalten, wenn bereits ein Workflow bearbeitet wird. Wird die SDK-Verbindung gelöst, so wird die UI wieder freigegeben und es werden wieder Anforderungen auf dem lokalen Webserver angenommen.

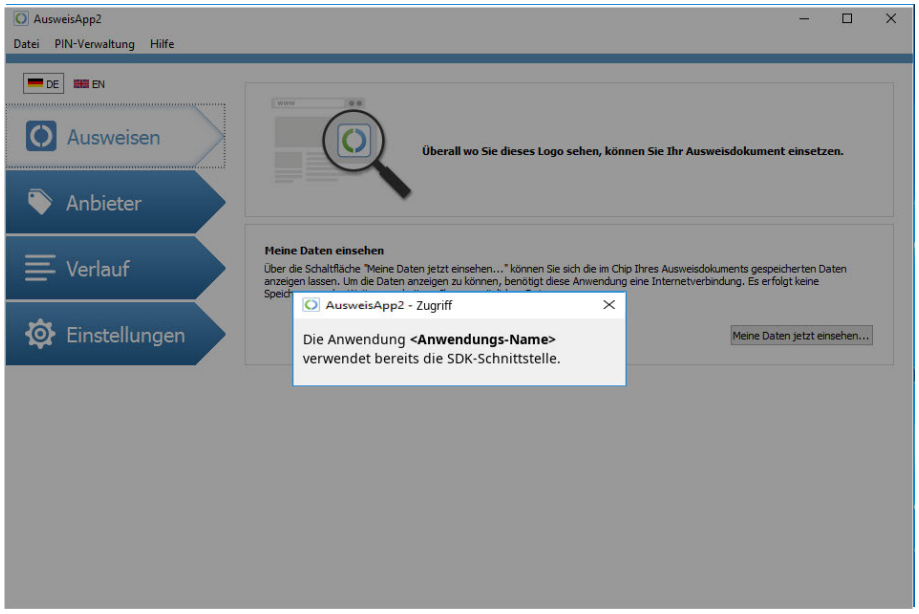

Abbildung 1: Zugriff über SDK-Schnittstelle

Wird ein Workflow über die UI der AusweisApp2 oder über den lokalen Webserver gestartet, so werden SDK-Verbindungen mit dem Websocket „Closed“-Status abgewiesen.

Für diese Funktionalität ist die Dokumentation des mobilen SDK zu erweitern und umzubenennen. Aufgrund der Nähe zum mobilen SDK ist es nicht notwendig, zwei unterschiedliche Dokumente zu erstellen und künftig zu pflegen.

2.2 Bereitstellungsumfang

Mit der Bereitstellung zur Abnahme erhält der Auftraggeber:

- Eine Vorabversion der AusweisApp2 welche den erweiterten Websocket-Pfad enthält für macOS und für Windows
- Eine überarbeitete Version der Entwicklerdokumentation (Zusammenführung mobiles und stationäres SDK)

2.3 Aufwand der Realisierung

Position	Aufwand (PT)
Projektmanagement	[redacted]
Implementierung	[redacted]
Dokumentation	[redacted]
Testaufwand	[redacted]
Summe:	[redacted]

2.4 Zeitplan

2.4.1 Machbarkeit der Realisierung

Die beschriebene Umsetzung kann bei einer Beauftragung bis zum 31. Mai 2018 bis zum 31. August 2018 realisiert werden.

2.4.2 Bereitstellung zur Abnahme

Die Bereitstellung zur Abnahme erfolgt am 31. August 2018.

2.4.3 Verfügbarkeit für Benutzer der AusweisApp2

Die mit diesem Change erfolgte Erweiterung kann mit dem auf die Abnahme durch den Auftraggeber folgenden Release im Jahr 2018 den Benutzern zur Verfügung stehen.

3 Abgrenzung zur Pflege

Die mit diesem Change beschriebenen Anpassungen können nicht im Rahmen der Pflege durchgeführt werden.

Der Bedarf an dieser Funktionalität beruht auf Anforderungen von Benutzern, die nach dem Erstellungsprojekt liegen. Im Rahmen der Pflege kann diese funktionale Erweiterung daher nicht abgebildet werden.

4 Risiken

Welche Risiken sind mit der Umsetzung bzw. der Nicht-Umsetzung dieses Changes verbunden?

Sollte dieser Change nicht umgesetzt werden, so besteht die Gefahr, dass wir den Bedürfnissen von Integratoren der AusweisApp2 nicht gerecht werden. Die Integration der Online-Ausweisfunktion in eigene Geschäftsprozesse gewinnt aufgrund der „neuen“ Möglichkeit des Vor-Ort-Auslesens an Bedeutung.