AusweisApp
Erweiterte Dokumentation für Administratoren und Entwickler

2.6.2

Governikus GmbH & Co. KG

Inhaltsverzeichnis

1 Installation **2**
1.1 Windows . . . 2
1.2 macOS . . . 4
1.3 Anforderungen an die Einsatzumgebung . . . 5
1.3.1 Rechte für Installation und Ausführung . . . 5
1.3.2 Verwendete Netzwerk-Ports . . . 5
1.3.3 TLS-Verbindungen . . . 7

2 Entwickleroptionen **9**
2.1 Aktivieren der Entwickleroptionen . . . 9
2.2 Erweiterte Einstellungen . . . 9
2.2.1 Testmodus für die Selbstauskunft (Test-PKI) . . . 9
2.2.2 Interner Kartensimulator . . . 9
2.2.3 Entwicklermodus (nur stationär) . . . 9
2.2.4 Benachrichtigungen in der App anzeigen . . . 10
2.2.5 CAN-Allowed Modus für Vor-Ort-Auslesen untertützen (nur mobil) . . . 10
2.2.6 Anzeige der Berechtigungen überspringen (nur mobil) . . . 10

3 Software Development Kit (SDK) **10**
3.1 Einsatzmöglichkeiten . . . 10
3.2 Integrationsmöglichkeiten . . . 10
3.3 Entwicklerdokumentation . . . 11
3.4 SDK Wrapper . . . 11

1 Installation

1.1 Windows

Der Installer der AusweisApp kann über die Kommandozeile gestartet werden, um den Installationsprozess zu konfigurieren und systemweite Standardeinstellungen vorzugeben. Der Rückgabewert von msiexec informiert über das Ergebnis der Installation[1]. Neben den üblichen Parametern[2] enthält das folgende Kommando alle unterstützten Parameter, die im Anschluss erläutert werden.

```
msiexec /i AusweisApp-X.YY.Z.msi /quiet INSTALLDIR="C:\AusweisApp" SYSTEMSETTING␣
→S=false DESKTOPSHORTCUT=false PROXYSERVICE=false AUTOSTART=false TRAYICON=true A␣
→UTOHIDE=false REMINDTOCLOSE=false ASSISTANT=false TRANSPORTPINREMINDER=false CUS␣
→TOMPROXYTYPE="HTTP" CUSTOMPROXYHOST="proxy.example.org" CUSTOMPROXYPORT=1337 UPD␣
→ATECHECK=false SHUFFLESCREENKEYBOARD=true SECURESCREENKEYBOARD=true ENABLECANALL␣
→OWED=true SKIPRIGHTSONCANALLOWED=true LAUNCH=true
```

INSTALLDIR Gibt das Installationsverzeichnis an. Ohne Angabe wird der Ordner "C:\Programme\AusweisApp" genutzt.

SYSTEMSETTINGS Betrifft die Erstellung von Firewall-Regeln der Windows Firewall. Ohne Angabe des Parameters werden die Firewall-Regeln erstellt (true). Durch Angabe von SYSTEMSETTINGS=false werden keine Firewall-Regeln erstellt.

DESKTOPSHORTCUT Durch Angabe von DESKTOPSHORTCUT=false kann die Erstellung einer Desktop-Verknüpfung vermieden werden. Ohne Angabe des Parameters wird eine Desktop-Verknüpfung für alle Benutzer erstellt (true).

PROXYSERVICE Um den parallelen Betrieb mehrer Instanzen der AusweisApp zu ermöglichen, ist der Proxy-Dienst notwendig. Der Proxy-Dienst übernimmt die Überwachung von Port 24727 (definiert in BSI TR-03124-1) und leitet Anfragen an die lokalen Instanzen der AusweisApp weiter. Eine Weiterleitung der Discovery-Nachrichten (Ergänzung zu BSI TR-03112-6 - IFD Service - Kapitel 3) erfolgt nicht, so dass SaK-Geräte in diesem Betriebsmodus nicht erkannt bzw. genutzt werden können. Ohne Angabe des Parameters wird der Proxy-Dienst automatisch eingerichtet, wenn Terminaldienste installiert sind und das System im Anwendungsservermodus ausgeführt wird.

AUTOSTART Durch Angabe von AUTOSTART=true wird ein Autostart-Eintrag für alle Benutzer erstellt. Die Deaktivierung des Autostarts ist den Benutzern in der AusweisApp dadurch nicht möglich. Ohne Angabe wird der Autostart-Eintrag nicht erstellt (false). In diesem Fall ist es jedoch jedem Benutzer möglich, die Autostart-Funktion innerhalb der AusweisApp für sich zu aktivieren.

TRAYICON Aktiviert das Trayicon damit die AusweisApp dauerhaft im Hintergrund aktiv ist und jederzeit für eine Authentisierung zur Verfügung steht.

AUTOHIDE Betrifft die automatische Minimierung nach Abschluss einer erfolgreichen Authentisierung. Ohne Angabe ist diese aktiviert (true). Durch AUTOHIDE=false wird diese deaktiviert. Der Benutzer kann diese Einstellung anpassen.

REMINDTOCLOSE Wenn der Benutzer die AusweisApp per Klick auf das X schließt, wird er darauf hingewiesen, dass nur die Benutzeroberfläche geschlossen wird und die AusweisApp weiterhin im Infobereich zur Verfügung steht (falls das Trayicon aktiviert ist) bzw. dass die AusweisApp geschlossen wird und

[1] https://docs.microsoft.com/de-de/windows/desktop/msi/error-codes
[2] https://docs.microsoft.com/de-de/windows/desktop/msi/standard-installer-command-line-options

erneut geöffnet werden muss um sich gegenüber Diensteanbietern auszuweisen. Zu diesem Zeitpunkt ist es möglich, den Hinweis zukünftig zu unterdrücken. Durch REMINDTOCLOSE=false kann dieser Hinweis von vornherein deaktiviert werden. Ohne Angabe ist er aktiviert (true).

ASSISTANT Startet der Benutzer die AusweisApp zum ersten Mal, wird die Benutzeroberfläche geöffnet und ein Einrichtungsassistent angezeigt. Bei jedem weiteren Start wird die AusweisApp im Hintergrund gestartet und der Einrichtungsassistent erscheint nicht. Durch ASSISTANT=false wird die AusweisApp auch beim ersten Start im Hintergrund ohne Einrichtungsassistenten gestartet. Ohne Angabe ist der Einrichtungsassistent aktiviert (true).

TRANSPORTPINREMINDER Zu Beginn einer Selbstauskunft oder Authentisierung wird der Benutzer einmalig danach gefragt, ob er die Transport-PIN schon geändert hat. Durch TRANSPORTPINREMINDER=false kann diese Abfrage deaktiviert werden. Ohne Angabe ist die Abfrage aktiviert (true).

CUSTOMPROXYTYPE Teil der Konfiguration eines Proxys. Gültige Typen sind SOCKS5 und HTTP. Um einen Proxy zu nutzen müssen alle Parameter gesetzt sein (siehe CUSTOMPROXYHOST und CUSTOMPROXYPORT). Der Proxy kann nach der Installation über eine Checkbox in den Einstellungen deaktiviert werden.

CUSTOMPROXYHOST Teil der Konfiguration eines Proxys. Angabe des Hosts, unter dem der Proxy zu erreichen ist. Um einen Proxy zu nutzen müssen alle Parameter gesetzt sein (siehe CUSTOMPROXYTYPE und CUSTOMPROXYPORT). Der Proxy kann nach der Installation über eine Checkbox in den Einstellungen deaktiviert werden.

CUSTOMPROXYPORT Teil der Konfiguration eines Proxys. Angabe des Proxyports. Nur Werte von 1 bis 65536 sind gültig. Um einen Proxy zu nutzen müssen alle Parameter gesetzt sein (siehe CUSTOMPROXYTYPE und CUSTOMPROXYHOST). Der Proxy kann nach der Installation über eine Checkbox in den Einstellungen deaktiviert werden.

UPDATECHECK Auf Windows prüft die AusweisApp bei Start und danach in einem 24 Stunden Intervall ob eine neue Version verfügbar ist. Liegt eine neue Version vor wird der Benutzer darüber bei aktiviertem Trayicon mit einer Notification informiert. Beim nächsten Öffnen der AusweisApp wird zusätzlich ein Hinweis auf die neue Version angezeigt. Durch Setzen von UPDATECHECK auf false oder true kann diese Überprüfung deaktiviert bzw. aktiviert werden. Die Einstellung kann dann durch den Benutzer in der AusweisApp nicht geändert werden. Ohne Angabe ist die Überprüfung aktiviert, der Benutzer kann die Einstellung jedoch ändern. Der UPDATECHECK Parameter beeinflusst weder die Aktualisierung der Anbieterliste noch die Aktualisierung der Kartenleserinformationen.

SHUFFLESCREENKEYBOARD Ist die Bildschirmtastatur aktiviert, können die Zifferntasten zufällig angeordnet werden. Durch Setzen von SHUFFLESCREENKEYBOARD auf false oder true kann die zufällige Anordnung deaktiviert bzw. aktiviert werden. Der Benutzer kann diese Einstellung anpassen.

SECURESCREENKEYBOARD Ist die Bildschirmtastatur aktiviert, kann die Animation der Zifferntasten deaktiviert werden. Durch Setzen von SECURESCREENKEYBOARD auf false oder true kann die Animation aktiviert bzw. deaktiviert werden. Der Benutzer kann diese Einstellung anpassen.

ENABLECANALLOWED Aktiviert die Unterstützung für den CAN-Allowed-Modus (Vor-Ort-Auslesen). Wenn ein entsprechendes Berechtigungszertifikat vorliegt, muss zum Auslesen die CAN anstelle der PIN eingegeben werden.

SKIPRIGHTSONCANALLOWED Überspringt die Anzeige des Berechtigungszertifikat im CAN-Allowed-Modus und wechselt direkt zur CAN-Eingabe.

LAUNCH Startet die AusweisApp nach dem Ende der Installation.

Alternativ kann mit Orca[3] eine MST-Datei erzeugt werden, die die oben genannten Parameter definiert. Die

[3] https://docs.microsoft.com/de-de/windows/desktop/Msi/orca-exe

Parameter sind in den Tabellen "Directory" und "Property" verfügbar. Übergeben lässt sich die MST-Datei mit dem folgenden Kommando:

```
msiexec /i AusweisApp-X.YY.Z.msi /quiet TRANSFORMS=file.mst
```

Um den Start der AusweisApp auf Systemen mit fehlender Grafikbeschleunigung zu optimieren, kann die Systemvariable "QT_QUICK_BACKEND" auf den Wert "software" gesetzt werden. In diesem Fall verzichtet die AusweisApp auf den Versuch die Grafikbeschleunigung zu nutzen und startet direkt mit dem alternativen Softwarerenderer.

1.2 macOS

Unter macOS ist keine Installation per Kommandozeile vorgesehen. Jedoch können einige der oben genannten Einstellung durch eine plist-Datei im Verzeichnis /Library/Preferences systemweit vorgegeben werden. Diese plist-Datei muss dabei manuell durch den Administrator des Systems hinterlegt werden und wird von allen (zukünftigen) Installationen der AusweisApp verwendet. Alle nicht genannten Einstellungen werden auf macOS nicht unterstützt. Der Name der Datei muss "com.governikus.AusweisApp2.plist" lauten. Der Inhalt wird im folgenden dargestellt:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/␣
→PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>trayIcon</key>
  <false/>
  <key>autoCloseWindow</key>
  <false/>
  <key>remindToClose</key>
  <false/>
  <key>showOnboarding</key>
  <false/>
  <key>transportPinReminder</key>
  <false/>
  <key>customProxyType</key>
  <string>HTTP</string>
  <key>customProxyHost</key>
  <string>proxy.example.org</string>
  <key>customProxyPort</key>
  <integer>1337</integer>
  <key>shuffleScreenKeyboard</key>
  <true/>
  <key>visualPrivacy</key>
  <true/>
  <key>enableCanAllowed</key>
  <true/>
```

```
  <key>skipRightsOnCanAllowed</key>
  <true/>
</dict>
</plist>
```

Für die einzelnen Werte gelten die gleichen Beschreibungen wie für die Windows-Version wobei die Bennennung der Attribute der folgenden Tabelle zu entnehmen ist.

Tabelle 1:

macOS	Windows
trayIcon	TRAYICON
autoCloseWindow	AUTOHIDE
remindToClose[4]	REMINDTOCLOSE
showOnboarding	ASSISTANT
transportPinReminder	TRANSPORTPINREMINDER
customProxyType	CUSTOMPROXYTYPE
customProxyPort	CUSTOMPROXYPORT
customProxyHost	CUSTOMPROXYHOST
shuffleScreenKeyboard	SHUFFLESCREENKEYBOARD
visualPrivacy	SECURESCREENKEYBOARD
enableCanAllowed	ENABLECANALLOWED
skipRightsOnCanAllowed	SKIPRIGHTSONCANALLOWED

Nach Änderung der Datei kann es notwendig sein, ein erneutes Laden der vom Betriebssystem gecachten Daten zu erzwingen: `killall -u $USER cfprefsd`

1.3 Anforderungen an die Einsatzumgebung

1.3.1 Rechte für Installation und Ausführung

Für die Installation der AusweisApp sind Administratorrechte erforderlich.

Die Ausführung der AusweisApp erfordert keine Administratorrechte.

1.3.2 Verwendete Netzwerk-Ports

In Netzwerkverbindungen der AusweisApp (Seite 8) werden alle von der AusweisApp genutzten Ports aufgelistet. Eine schematische Darstellung der einzelnen Verbindungen, die von der AusweisApp genutzt werden, ist in Kommunikationsmodell der AusweisApp (Seite 6) dargestellt.

Die AusweisApp startet einen HTTP-Server, der über Port 24727 erreichbar ist. Der Server empfängt nur auf der localhost Netzwerkschnittstelle. Die Erreichbarkeit dieses lokalen Servers ist für die Onlineausweisfunktion notwendig, da Anbieter mit einem HTTP-Redirect auf den lokalen Server umleiten um den Ausweisvorgang in der AusweisApp fortzuführen (eID1). Außerdem wird über den Server die Verwendung der AusweisApp von anderen Anwendungen über eine Websocket-Schnittstelle angeboten (SDK-Funktion, eID-SDK). Daher müssen eingehende lokale Netzwerkverbindungen auf dem TCP Port 24727 ermöglicht werden.

[4] Unter macOS wird die AusweisApp in die Menüleiste minimiert.

Bei aktiviertem Proxy-Dienst übernimmt der AusweisApp-Proxy die Serverfunktionen der AusweisApp auf Port 24727. Die Instanzen der AusweisApp erkennen den Proxy und benutzen in diesem Fall einen zufälligen freien Port auf den der Proxy die Anfragen weiterleitet.

Für die Verwendung von der "Smartphone als Kartenleser"-Funktion über WLAN müssen außerdem Broadcasts auf UDP Port 24727 im lokalen Subnetz empfangen werden können. Hierzu muss eventuell die AP Isolation im Router deaktiviert werden.

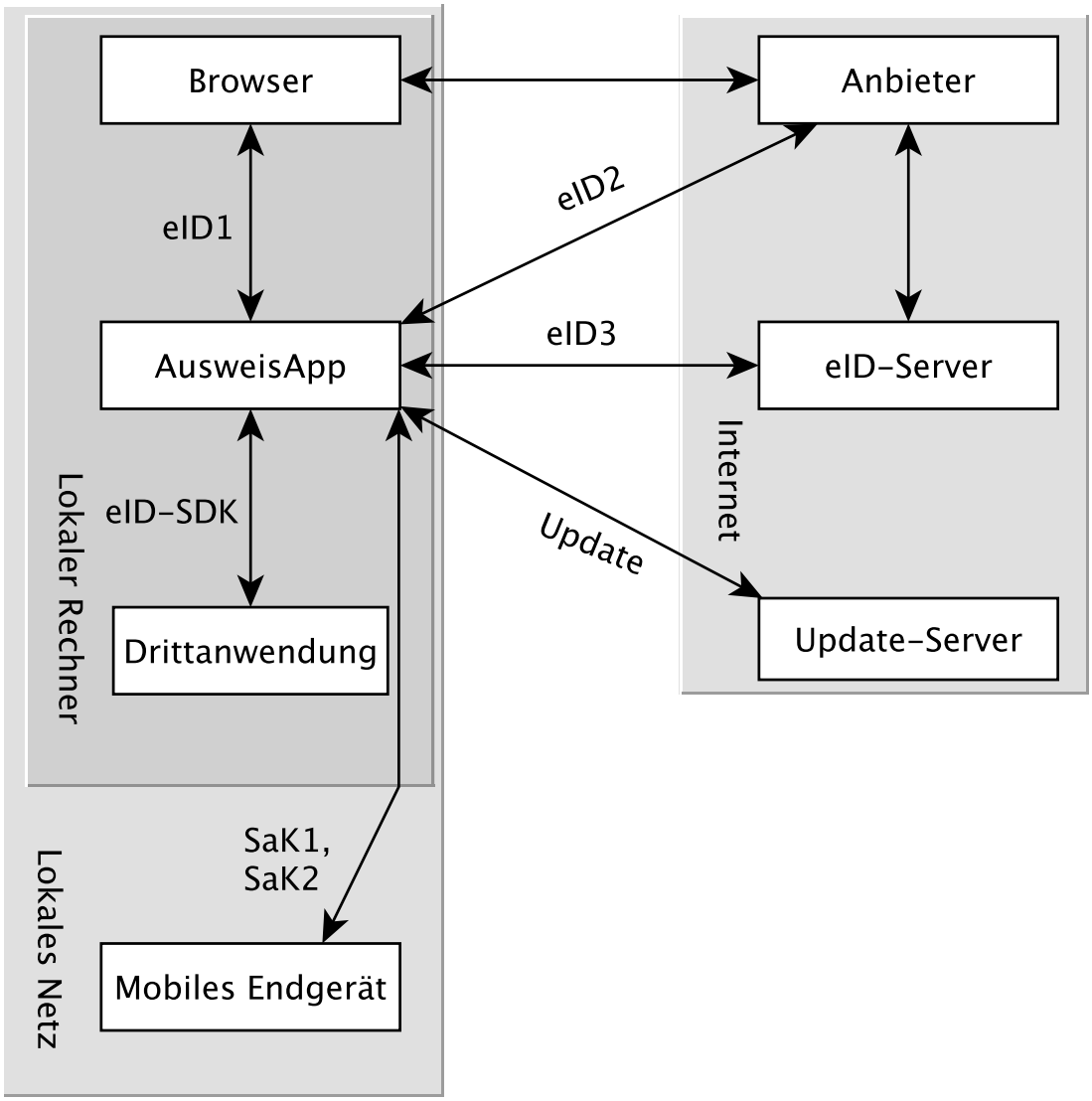


Abbildung 1: Kommunikationsmodell der AusweisApp

Der Installer der AusweisApp bietet die Option, für alle angebotenen Funktionen der AusweisApp die erforderlichen Firewall-Regeln in der Windows-Firewall zu registrieren. Erfolgt die Registrierung der Firewall-Regeln nicht, wird der Benutzer bei einem Verbindungsaufbau der AusweisApp mit einem Dialog der Windows-Firewall aufgefordert, die ausgehenden Datenverbindungen zuzulassen. Durch Registrierung der Firewall-Regeln während

der Installation werden diese Aufforderungen unterbunden.

Für die lokalen Verbindungen eID1 und eID-SDK müssen (unter den gängigen Standardeinstellungen der Windows-Firewall) keine Regeln in der Windows-Firewall eingetragen werden.

Die durch den Installer angelegten Regeln werden in Tabelle Firewallregeln der AusweisApp (Seite 8) aufgelistet.

1.3.3 TLS-Verbindungen

Es ist generell nicht möglich, die AusweisApp mit einem TLS-Termination-Proxy zu verwenden, da die übertragenen TLS-Zertifikate über eine Verschränkung mit dem Berechtigungszertifikat aus der Personalausweis-PKI validiert werden. CA-Zertifikate im Windows-Truststore werden daher ignoriert.

Tabelle 2: Netzwerkverbindungen der AusweisApp

Referenz	Protokoll	Port	Richtung	Optional	Zweck	Anmerkungen
eID1	TCP	24727 1[4]	eingehend	Nein	Online-Ausweisvorgang, eID-Aktivierung 2[4]	Nur erreichbar von localhost 2[4]
eID2	TCP	443 3[4]	ausgehend	Nein	Online-Ausweisvorgang, Verbindung zum Anbieter, TLS-1-2-Kanal 2[4]	TLS-Zertifikate verschränkt mit Berechtigungs-Zertifikat 2[4]
eID3	TCP	443 3[4]	ausgehend	Nein	Online-Ausweisvorgang, Verbindung zum eID-Server, TLS-2-Kanal 2[4]	TLS-Zertifikate verschränkt mit Berechtigungs-Zertifikat 2[4]
eID-SDK	TCP	24727 1[4]	eingehend	Nein	Verwendung der SDK-Schnittstelle	Nur erreichbar von localhost 2[4]
SaK1	UDP	24727 1[4]	eingehend	Ja	Smartphone als Kartenleser, Erkennung 4[4]	Broadcasts
SaK2	TCP		ausgehend	Ja	Smartphone als Kartenleser, Verwendung 4[4]	Verbindung im lokalen Subnetz
Update	TCP	443	ausgehend	Ja	Updates 5[4] zu Anbietern und Kartenlesern sowie Informationen zu neuen AusweisApp-Versionen 6[4] .	Die Zertifikate der TLS-Verbindung werden mit in der AusweisApp mitgelieferten CA-Zertifikaten validiert. Im Betriebssystem hinterlegte CA-Zertifikate werden ignoriert.

Tabelle 3: Firewallregeln der AusweisApp

Name	Protokoll	Port	Richtung	Umgesetzte Verbindung
AusweisApp-Outbound	TCP	*	ausgehend	eID2, eID3, SaK2, Update
AusweisApp-SaC	UDP	24727	eingehend	SaK1

[5] Oder ein zufälliger Port bei Verwendung des AusweisApp-Proxys.

[6] Siehe TR-03124 des BSI

[7] Port 443 wird für die initiale Kontaktaufnahme zum Anbieter bzw. eID-Server verwendet. Durch die Konfiguration des Dienstes durch den Diensteanbieter können durch Weiterleitungen beliebige andere Ports zum Einsatz kommen.

[8] Siehe TR-03112-6 des BSI

[9] Erreichbar unter dem URL https://updates.autentapp.de/

[10] Die Überprüfung auf neue AusweisApp-Versionen kann deaktiviert werden, siehe Kommandozeilenparameter UPDATECHECK

2 Entwickleroptionen

Die AusweisApp verfügt über sogenannte Entwickleroptionen. Diese bieten erweiterte Einstellmöglichkeiten und unterstützen die Integration eines eID-Dienstes. Die Entwickleroptionen werden standardmäßig ausgeblendet.

2.1 Aktivieren der Entwickleroptionen

Um die Entwickleroptionen zu aktivieren, öffnen Sie im Menü "Hilfe" den Punkt "Information". Klicken Sie zehnmal auf die "Anwendungsversion". Versionsinformationen. Nach dem zehnten Klick erhalten Sie eine Benachrichtigung, dass die Entwickleroptionen aktiviert sind. Im Bereich Einstellungen befindet sich nun eine neue Kategorie "Entwickleroptionen". In den mobilen Versionen erscheinen zusätzlich Optionen zum "Vor-Ort-Auslesen".

2.2 Erweiterte Einstellungen

Die Entwickleroptionen bieten erweiterte Einstellungsmöglichkeiten, die nachfolgend erläutert werden.

2.2.1 Testmodus für die Selbstauskunft (Test-PKI)

Die Selbstauskunft ist ein fest integrierter Dienst der AusweisApp und kann nur mit Echtausweisen genutzt werden. Wird der Testmodus (Test-PKI) aktiviert, nutzt die AusweisApp einen Test-Dienst, der es ermöglicht, eine Selbstauskunft mit einem Testausweis durchzuführen. Der Testmodus (Test-PKI) für die Selbstauskunft kann auch ohne Aktivierung der Entwickleroptionen durch zehn Klicks auf die Lupe im Bereich "Meine Daten einsehen" aktiviert und deaktiviert werden.

2.2.2 Interner Kartensimulator

Der interne Kartensimulator ermöglicht die Durchführung einer Authentisierung in der Test-PKI ohne Ausweis oder Kartenleser. Beachten Sie, dass in den stationären Versionen kein anderer Kartenleser verwendet werden kann, während der Simulator aktiviert ist.

In der aktuellen Version ist ein einzelnes statisches Profil hinterlegt, das über die grafische Oberfläche nicht geändert werden kann. Lediglich im SDK ist es möglich die Daten über das Kommando SET_CARD zu beeinflussen. Weitere Informationen dazu finden Sie in der Dokumentation des AusweisApp SDK (siehe Software Development Kit (SDK) (Seite 10)).

2.2.3 Entwicklermodus (nur stationär)

Mit der Aktivierung des Entwicklermodus werden einige Sicherheitsabfragen während einer Authentisierung ignoriert. In Entwicklungsszenarien, in denen ohnehin mit Test-Diensten gearbeitet wird, führt das Ignorieren der Sicherheitsabfragen dazu, dass eine Authentisierung erfolgreich durchgeführt werden kann. Auf jede Sicherheitsverletzung wird in den internen Benachrichtigungen der AusweisApp bzw. des Betriebssystems hingewiesen.

Die folgenden Sicherheitsüberprüfungen sind im Entwicklermodus abgeschaltet:

- Die verwendeten TLS-Schlüssel und ephemeralen TLS-Schlüssel haben die notwendige Mindestlänge.
- Die URL der Beschreibung des TLS-Zertifikats des eID-Servers und die TcToken-URL müssen die Same-Origin-Policy erfüllen.
- Die verwendeten TLS-Zertifikate müssen mit dem Berechtigungszertifikat verschränkt sein.
- Die RefreshAddress-URL und etwaige Redirect-URL müssen das HTTPS-Schema erfüllen.

Der Entwicklermodus ist nur unter Windows und macOS verfügbar.

Wichtig: Der Entwicklermodus kann nur für Test-Dienste verwendet werden, eine Verwendung mit echten Berechtigungszertifikaten ist nicht möglich.

2.2.4 Benachrichtigungen in der App anzeigen

Mit dieser Option werden Benachrichtigungen in der App anstelle von Systembenachrichtigungen angezeigt. Wenn die Option aktiviert ist, erscheint oben rechts im Applikationsfenster ein Glockensymbol. Ein Klick auf das Glockensymbol zeigt die bisherigen Benachrichtigungen mit Zeitstempel. Diese Option wird automatisch aktiviert wenn der Entwicklermodus aktiviert ist, da dort die Sicherheitsverletzungen aufgeführt werden.

2.2.5 CAN-Allowed Modus für Vor-Ort-Auslesen untertützen (nur mobil)

Aktiviert die Unterstützung für den CAN-Allowed-Modus (Vor-Ort-Auslesen). Wenn ein entsprechendes Berechtigungszertifikat vorliegt, muss zum Auslesen die CAN anstelle der PIN eingegeben werden.

2.2.6 Anzeige der Berechtigungen überspringen (nur mobil)

Überspringt die Anzeige des Berechtigungszertifikat im CAN-Allowed-Modus und wechselt direkt zur CAN-Eingabe.

3 Software Development Kit (SDK)

3.1 Einsatzmöglichkeiten

Mit dem Software Development Kit (SDK) der AusweisApp ist es Ihnen möglich, die Online-Ausweisfunktion direkt in die eigene Anwendung bzw. App zu integrieren. Damit ermöglichen Sie Ihren Benutzern die medienbruchfreie Durchführung einer Authentisierung - z.B. für Registrierungen oder Logins.

Das SDK bietet Ihnen dabei den Vorteil, die Online-Authentisierung durchgehend im eigenen Markendesign durchzuführen - ohne dass die Benutzer die gewohnte Umgebung verlassen müssen.

Das AusweisApp SDK ermöglicht auch die Integration des Vor-Ort-Auslesens. Hierbei wird anstelle der PIN zur Freigabe der Datenübertragung die CAN übermittelt. Diese ist auf der Vorderseite des Ausweises aufgedruckt und wird zur Freigabe des Auslesevorgangs benötigt.

3.2 Integrationsmöglichkeiten

Bei der voll-integrierten Version des SDKs wird die AusweisApp als AAR Package bzw. Swift Package in Ihre eigene Anwendung eingebunden. Der Vorteil: Die AusweisApp wird direkt mit ausgeliefert, sodass Benutzer die AusweisApp nicht separat auf Ihrem Smartphone installiert haben müssen.

Bei der teil-integrierten Version des SDKs wird die AusweisApp im Hintergrund aufgerufen. Ggf. kann die App jedoch trotz Teil-Integration mit dem Installer ausgeliefert werden.

Tabelle 4:

	Teil-Integration	Voll-Integration
Windows / macOS	Ja	Nein
Android	Nein	Ja
iOS	Nein	Ja

3.3 Entwicklerdokumentation

Eine ausführliche Entwicklerdokumentation des SDKs und eine Auflistung der möglichen Fehlercodes finden Sie unter https://www.ausweisapp.bund.de/sdk/.

3.4 SDK Wrapper

Sie können den SDK Wrapper der AusweisApp zur Vereinfachung der Einbindung des SDKs in Ihre App verwenden. Der SDK Wrapper bietet Swift und Kotlin Bindings für iOS und Android an.

Informationen zur Integration des SDK Wrappers finden Sie in der Entwicklerdokumentation unter https://www.ausweisapp.bund.de/sdkwrapper/.