AusweisApp Erweiterte Dokumentation für Administratoren und Entwickler

Release 2.3.0

Governikus GmbH & Co. KG

Deutsch

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"␣
↪SYSTEMSETTINGS=false DESKTOPSHORTCUT=false PROXYSERVICE=false␣
↪AUTOSTART=false TRAYICON=true AUTOHIDE=false REMINDTOCLOSE=false␣
↪ASSISTANT=false TRANSPORTPINREMINDER=false CUSTOMPROXYTYPE="HTTP"␣
↪CUSTOMPROXYHOST="proxy.example.org" CUSTOMPROXYPORT=1337 UPDATECHECK=false␣
↪ONSCREENKEYBOARD=true SHUFFLESCREENKEYBOARD=true SECURESCREENKEYBOARD=true␣
↪ENABLECANALLOWED=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.

[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

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 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

Wird die Benutzeroberfläche der AusweisApp geöffnet, wird eine Überprüfung auf eine neue Version der AusweisApp gestartet, falls seit der letzten Überprüfung mindestens 24 Stunden vergangen sind. Liegt eine neue Version vor, wird der Benutzer darüber in einem Dialog informiert. 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 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>
```

(Fortsetzung auf der nächsten Seite)

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

(Fortsetzung der vorherigen Seite)

```
  <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.

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 notwending sein, ein erneutes Laden der vom Betriebssystem gecachten Daten zu erzwingen: `killall -u $USER cfprefsd`

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

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 Tab. 1.1 werden alle von der AusweisApp genutzten Ports aufgelistet. Eine schematische Darstellung der einzelnen Verbindungen, die von der AusweisApp genutzt werden, ist in Abb. 1.1 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.

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.

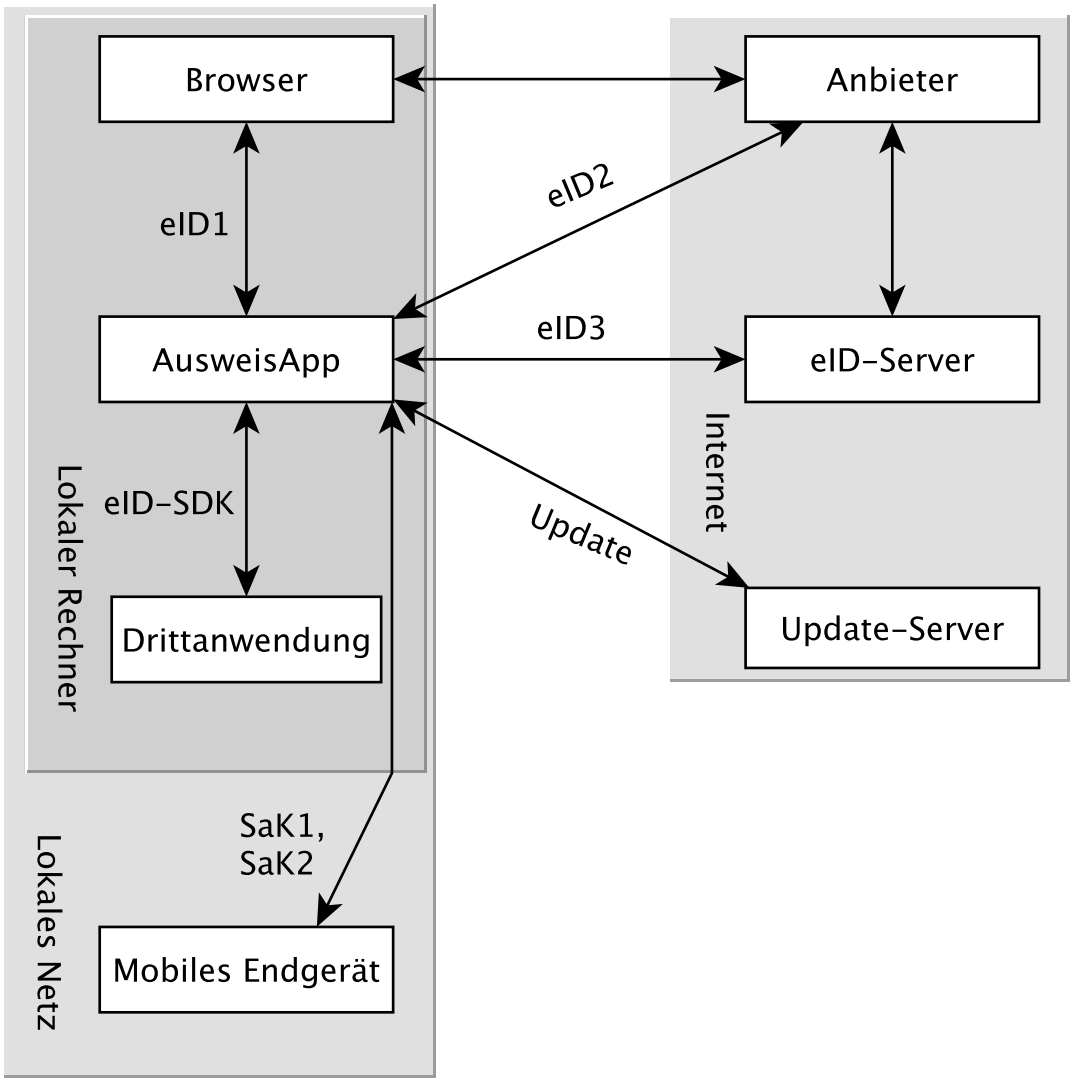

Abb. 1.1: Kommunikationsmodell der AusweisApp

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 Tab. 1.2 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.

Tab. 1.1: Netzwerkverbindungen der AusweisApp

Referenz	Protokoll	Port	Richtung	Optional	Zweck	Anmerkungen
eID1	TCP	24727[5]	eingehend	Nein	Online-Ausweisvorgang, eID-Aktivierung[6]	Nur erreichbar von localhost[6]
eID2	TCP	443[7]	ausgehend	Nein	Online-Ausweisvorgang, Verbindung zum Anbieter, TLS-1-2-Kanal[6]	TLS-Zertifikate verschränkt mit Berechtigungs-Zertifikat[6]
eID3	TCP	443[7]	ausgehend	Nein	Online-Ausweisvorgang, Verbindung zum eID-Server, TLS-2-Kanal[6]	TLS-Zertifikate verschränkt mit Berechtigungs-Zertifikat[6]
eID-SDK	TCP	24727[5]	eingehend	Nein	Verwendung der SDK-Schnittstelle	Nur erreichbar von localhost[6]
SaK1	UDP	24727[5]	eingehend	Ja	Smartphone als Kartenleser, Erkennung[8]	Broadcasts
SaK2	TCP		ausgehend	Ja	Smartphone als Kartenleser, Verwendung[8]	Verbindung im lokalen Subnetz
Update	TCP	443	ausgehend	Ja	Updates[9] zu Anbietern und Kartenlesern sowie Informationen zu neuen AusweisApp-Versionen[10] .	Die Zertifikate der TLS-Verbindung werden mit in der AusweisApp mitgelieferten CA-Zertifikaten validiert. Im Betriebssystem hinterlegte CA-Zertifikate werden ignoriert.

[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

Tab. 1.2: Firewallregeln der AusweisApp

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

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“.

Außerdem kann in den mobilen Versionen der AusweisApp der Testmodus (Test-PKI) für die Selbstauskunft durch zehn Klicks auf die Lupe im Bereich „Meine Daten einsehen“ aktiviert und deaktiviert werden.

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.

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)*).

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 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.

2.2.5 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.

Tab. 3.1: Integrationsmöglichkeiten auf den verschiedenen Plattformen

	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/.

English

4 Installation

4.1 Windows

Start the installer of AusweisApp using the command line to configure the installation process and preset system-wide default settings. The return value of msiexec indicates the result of the installation[1]. In addition to the usual arguments[2], the following command contains all supported arguments, which are explained below.

```
msiexec /i AusweisApp-X.YY.Z.msi /quiet INSTALLDIR="C:\AusweisApp"␣
↪SYSTEMSETTINGS=false DESKTOPSHORTCUT=false PROXYSERVICE=false␣
↪AUTOSTART=false TRAYICON=true AUTOHIDE=false REMINDTOCLOSE=false␣
↪ASSISTANT=false TRANSPORTPINREMINDER=false CUSTOMPROXYTYPE="HTTP"␣
↪CUSTOMPROXYHOST="proxy.example.org" CUSTOMPROXYPORT=1337 UPDATECHECK=false␣
↪ONSCREENKEYBOARD=true SHUFFLESCREENKEYBOARD=true SECURESCREENKEYBOARD=true␣
↪ENABLECANALLOWED=true SKIPRIGHTSONCANALLOWED=true LAUNCH=true
```

INSTALLDIR
States the installation directory. If not specified, the folder „C:\Program Files\AusweisApp“ is used.

SYSTEMSETTINGS
Concerns the settings of firewall rules of the Windows Firewall. When not specifying the argument, firewall rules are created (true). By indicating SYSTEMSETTINGS=false, no firewall rules are created.

DESKTOPSHORTCUT
By specifying DESKTOPSHORTCUT=false, no desktop shortcut is created. Without specifying the argument, the desktop shortcut is created for all users (true).

PROXYSERVICE
The proxy service is required to enable the parallel operation of several entities of AusweisApp. The proxy service monitors port 24727 (defined in BSI TR-03124-1) and forwards requests to the local AusweisApp instances. The Discovery messages (amendment to BSI TR-03112-6 - IFD Service - Chapter 3) are not forwarded, so that SaC devices cannot be recognized or used in this operating mode. Not specified, the proxy service will be installed automatically if Terminal Services is installed and the system is running in application server mode.

AUTOSTART
By setting AUTOSTART=true, an autostart entry is created for all users. Users are unable to deactivate the autostart function in the AusweisApp. Not specified, no autostart entry is created (false). In that case, users are able to activate the autostart function in the AusweisApp.

TRAYICON
Activates the tray icon to keep the AusweisApp active in the background to always be available for an authentication.

AUTOHIDE

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

Concerns the automatic minimization after a successful authentication. Not specified, it is activated (true). Setting AUTOHIDE=false, it is deactivated. Users can adjust this setting to their preferences.

REMINDTOCLOSE

Closing the AusweisApp by clicking on the X, the user is notified that only the user interface is closed and that the AusweisApp is still available in the info tray (if the tray icon is enabled) or that the AusweisApp will be shut down and the user needs to restart it to identify towards providers. At this point, it is possible to prevent future notifications. Setting REMINDTOCLOSE=false deactivates this notification from the outset. Not specified, it is activated (true).

ASSISTANT

Starting the AusweisApp for the first time, the user interface is displayed and the installation wizard is shown. With each subsequent start, the AusweisApp is started in the background, without the installation wizard being shown. By indicating ASSISTANT=false, the AusweisApp is started in the background without the installation wizard from the outset. Not specified, the installation wizard is activated (true).

TRANSPORTPINREMINDER

Prior to the first authentication, the user is asked once whether they have changed their Transport PIN. Setting TRANSPORTPINREMINDER=false deactivates this reminder. Not specified, the reminder is activated (true).

CUSTOMPROXYTYPE

Part of a proxy configuration. Valid values are SOCKS5 and HTTP. All proxy parameters have to be set to use the proxy (see CUSTOMPROXYHOST and CUSTOMPROXYPORT). You can disable the proxy after installation with a checkbox in the settings.

CUSTOMPROXYHOST

Part of a proxy configuration. Sets the Host of the proxy. All proxy parameters have to be set to use the proxy (see CUSTOMPROXYTYPE and CUSTOMPROXYPORT). You can disable the proxy after installation with a checkbox in the settings.

CUSTOMPROXYPORT

Part of a proxy configuration. Sets the port of the proxy. Only values between 1 and 65536 are valid. All proxy parameters have to be set to use the proxy (see CUSTOMPROXYTYPE and CUSTOMPROXYHOST). You can disable the proxy after installation with a checkbox in the settings.

UPDATECHECK

Upon opening the user interface of the AusweisApp, an update check is started, provided that at least 24 hours have elapsed since the last update check. If a newer version is available, the user is notified accordingly. Setting UPDATECHECK to false or true deactivates or activates the update check respectively. Users are unable to change this setting in the AusweisApp. Not specified, the update check is activated, but users can adjust the settings. The UPDATECHECK parameter affects neither updates of the service provider list nor updates of card reader information.

SHUFFLESCREENKEYBOARD

If the on-screen keyboard is activated, the number keys can be arranged at random. By setting SHUFFLESCREENKEYBOARD to false or true, the random arrangement can be deactivated or activated. Users are able to adjust the setting.

SECURESCREENKEYBOARD

If the on-screen keyboard is activated, the animation of the number keys can be disabled. By setting SECURESCREENKEYBOARD to false or true, the animation can be activated or deactivated. Users are able to adjust the setting.

ENABLECANALLOWED

Enables support for the CAN allowed mode. If the provider got issued a corresponding authorization certificate the ID card can be read by entering the CAN instead of the PIN.

SKIPRIGHTSONCANALLOWED
Skips the page with the authorization certificate in the CAN allowed mode and asks directly for the CAN.

LAUNCH
Starts the AusweisApp after the installation has finished.

Alternatively, Orca[3] can be used to create an MST file that defines the above arguments. The arguments are available in the „Directory“ and „Property“ tables. The MST file can be transferred with the following command:

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

In order to optimize the start of the AusweisApp on systems with no graphics acceleration, the system variable „QT_QUICK_BACKEND“ can be set to the value „software“. In this case, the AusweisApp does not attempt to use graphics acceleration and starts directly with the alternative software renderer.

4.2 macOS

MacOS does not provide a command line installation. However, some of the above settings can be specified system-wide by a plist file in the /Library/Preferences directory. This plist file must be manually stored by the administrator of the system and will be used by all (future) installations of AusweisApp. All not mentioned settings are not supported on macOS. The name of the file must be „com.governikus.AusweisApp2.plist“. The content is shown below:

```
<?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>
```

(Fortsetzung auf der nächsten Seite)

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

(Fortsetzung der vorherigen Seite)

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

The description for each value is applicable for both Windows and macOS, although the naming of the attributes differs, as shown in the following table:

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

It might be necessary to force a reload of the data cached by the operating system: `killall -u $USER cfprefsd`

4.3 Operational Environment Requirements

4.3.1 Required authorization for installation and execution

Administrator privileges are required to install the AusweisApp.

The execution of the AusweisApp does not require administrator privileges.

4.3.2 Used network ports

All network ports used by the AusweisApp are listed in Tab. 4.1. Abb. 4.1 shows a schematic representation of the individual connections made by the AusweisApp.

The AusweisApp starts a HTTP-Server on port 24727. The server binds only to the localhost network interface. The availability of the local server is necessary for the online eID function, because providers will redirect the user with a HTTP redirect to the local server to continue the authentication process in

[4] On macOS the AusweisApp is minimized to the menu bar.

the AusweisApp (eID1). The server is also used to offer other local applications to use the AusweisApp via a websocket interface (SDK function, eID-SDK). Therefore local incoming network connections to TCP Port 24727 must be permitted.

If the proxy service is activated, the AusweisApp proxy takes over the server functions of AusweisApp on port 24727. The entities of AusweisApp recognize the proxy and use a free random port in this case to which the proxy forwards the requests.

Broadcast on UDP port 24727 in the local subnet have to be receivable by the AusweisApp to use the „Smartphone as Card Reader“ functionality. It may be necessary to deactivate AP isolation on your router.

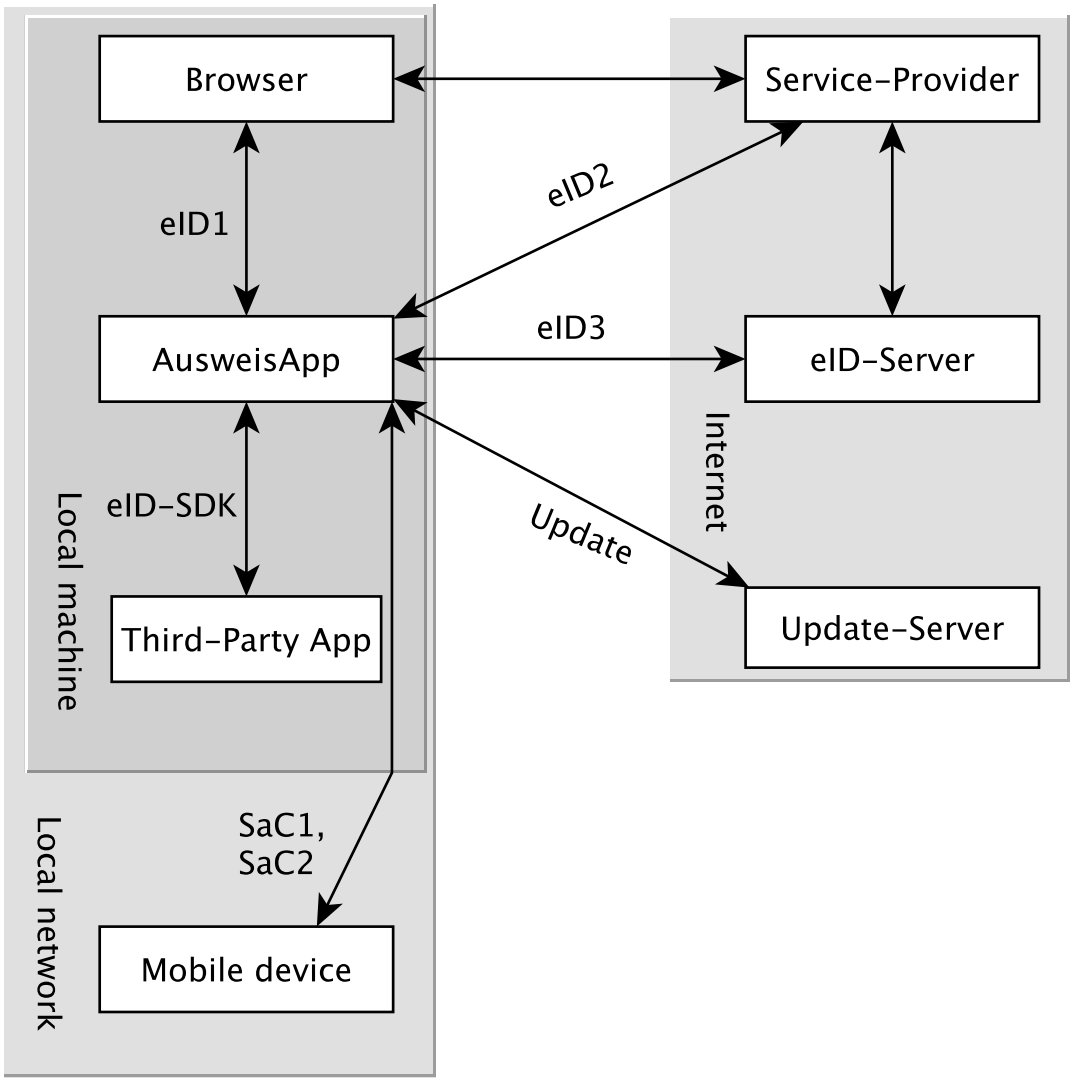

Abb. 4.1: Communication model of the AusweisApp

The installer of the AusweisApp provides an option to register all needed firewall rules in the Windows Firewall. If the rules are not registered, the user will be prompted by the Windows Firewall to allow the outgoing connections once the AusweisApp tries to connect to a server. These prompts are suppressed by registering the firewall rules during installation. No rules have to be added to the Windows Firewall for the local connections eID1 and eID-SDK (when using the standard settings).

In table Tab. 4.2 all firewall rules registered by the installer are listed.

4.3.3 TLS connections

Transmitted TLS certificates are solely validated via the interlacing with the authorization certificate issued by the german eID PKI. CA certificates in the Windows truststore are thus ignored. It is therefore generally not possible to use the AusweisApp behind a TLS termination proxy.

Tab. 4.1: Network connections of the AusweisApp

Reference	Protocol	Port	Direction	Optional	Purpose	Note
eID1	TCP	24727[5]	incoming	no	Online eID function, eID activation[6]	Only accessible from localhost[6]
eID2	TCP	443[7]	outgoing	no	Online eID function, connection to the provider, TLS-1-2 channel[6]	TLS certificates interlaced with authorization certificate[6]
eID3	TCP	443[7]	outgoing	no	Online eID function, connection to eID-Server, TLS-2 channel[6]	TLS certificates interlaced with authorization certificate[6]
eID-SDK	TCP	24727[5]	incoming	no	Usage of the SDK functionality	Only accessible from localhost[6]
SaC1	UDP	24727[5]	incoming	yes	Smartphone as Card Reader, detection[8]	Broadcasts
SaC2	TCP		outgoing	yes	Smartphone as Card Reader, usage[8]	Connection in local subnet
Update	TCP	443	outgoing	yes	Updates[9] of provider and card reader information as well as information on new AusweisApp versions[10] .	TLS certificates will be validated against CA certificates included in the AusweisApp. CA certificates provided by the OS are ignored.

Tab. 4.2: Firewall rules of the AusweisApp

Name	Protocol	Port	Direction	Connection reference
AusweisApp-Firewall-Rule	TCP	*	outgoing	eID2, eID3, SaC2, Update
AusweisApp-SaC	UDP	24727	incoming	SaC1

[5] Or a random port when using AusweisApp proxy.
[6] See TR-03124 specification from the BSI
[7] Port 443 is used for the initial contact with the provider or eID server. Due to configuration of the service on the service provider's behalf, any other port might be used by forwarding.
[8] See TR-03112-6 specifiaction from the BSI
[9] All updates are based on the URL https://updates.autentapp.de/
[10] Automatic checks for new AusweisApp versions can be deactivated, see commandline parameter UPDATECHECK.

5 Developer Options

AusweisApp features so-called developer options. They provide advanced settings and facilitate the integration of eID services. The developer options are hidden by default.

5.1 Activating the Developer Options

Developer options are activated by clicking the „Application Version“ accessible via „Help“ -> „Information“ 10 times. After the 10th time, you will receive a notification that the developer options are activated. Once activated, you will find a new category „developer options“ in the settings menu. In the mobile versions additional options for „on-site reading“ appear.

In the mobile versions of AusweisApp you can also activate and deactivate the test mode (Test PKI) for self-authentication by clicking the magnifying glass on the start screen 10 times.

5.2 Advanced Settings

The developer options offer advanced settings, which are explained below.

5.2.1 Test mode for self-authentication (Test PKI)

In general, the self-authentication is a built-in service of AusweisApp and can only be used with genuine ID cards. However, when in test mode, AusweisApp uses a test service allowing for self-authentication with a test ID card.

5.2.2 Internal card Simulator

The internal card simulator allows to run an authentication in the Test PKI without any ID card or card reader. Note that no other card reader can be used in the stationary versions while the simulator is activated.

A single static profile is stored in the current version, which cannot be changed via the graphical user interface. Only the SDK allows to change the profile’s data using the SET_CARD command. Further information can be found at the documentation of AusweisApp SDK (see *Software Development Kit (SDK)*).

5.2.3 Developer mode (stationary only)

When the developer mode is activated, some safety measures during an authentication process are ignored. Ignoring the safety measures with test services usually employed in test scenarios, yields a successful authentication. Each safety breach will be highlighted as an internal notification in AusweisApp or the operating system respectively.

The following safety tests are disabled in the developer mode:

- The used TLS keys and ephemeral TLS keys have the necessary minimum length.
- The URL of the TLS certificate description of the eID server and the TcToken URL must fulfill the same-origin policy.

- The used TLS certificates must be entwined with the authorization certificate.
- The RefreshAddress URL and possible redirect URLs must conform to the HTTPS scheme.

Please note: Developer mode can only be used for test services, usage with genuine provider certificates is not possible.

5.2.4 Support CAN Allowed mode for on-site reading (mobile only)

Enables support for the CAN allowed mode. If the provider got issued a corresponding authorization certificate the ID card can be read by entering the CAN instead of the PIN.

5.2.5 Skip rights page

Skips the page with the authorization certificate in the CAN allowed mode and asks directly for the CAN.

6 Software Development Kit (SDK)

6.1 Possible Uses

The software development kit (SDK) of AusweisApp enables you to integrate the eID function directly into your own application or app. This enables users to authenticate themselves without media discontinuity.

The SDK offers the advantage of being able to carry out an online authentication in your own brand design - without users having to leave the familiar environment.

The AusweisApp SDK also enables the integration of on-site reading. In this case, the CAN is transmitted instead of the PIN to enable data transmission. You find the CAN on the front of the ID card and you need it to enable the readout process.

6.2 Integration Options

With the fully integrated version of the SDK, AusweisApp is integrated into your own application as an AAR package or Swift package. The advantage: AusweisApp is delivered directly with the application so that users don't have to install AusweisApp separately on their smartphone.

With the partially integrated version of the SDK, AusweisApp is called in the background. Where applicable, however, the app can be delivered with the installer regardless of partial integration.

Tab. 6.1: Integration options for the different platforms

	partially integrated	fully integrated
Windows / macOS	Ja	Nein
Android	Nein	Ja
iOS	Nein	Ja

6.3 Developer documentation

You can find a detailed developer documentation of the SDK with a list of possible failure codes at https://www.ausweisapp.bund.de/sdk/.

6.4 SDK Wrapper

You can use the SDK Wrapper of the AusweisApp to simplify the integration of the SDK into your app. The SDK Wrapper offers Swift and Kotlin bindings for iOS and Android.

You can find information for integrating the SDK Wrapper in the developer documentation at https://www.ausweisapp.bund.de/sdkwrapper/.