

# INTEGRAČNÁ DOKUMENTÁCIA

Mobilné SDK pre eID 2.0 s duálnym rozhraním

Názov projektu: Posilnenie zabezpečenia eID a eDoPP dokladov podľa nariadenia EÚ č. 2019/1157 (eID 2.0)

Realizátor projektu: Ministerstvo vnútra Slovenskej republiky

## Verzia a história dokumentu:

ID	Verzia	Popis	Autor
1.	0.1	Prvotná verzia dokumentácie	Dodávateľ
2.	0.3	Aktualizovaná verzia v rámci release 1. inkrementu SDK	Dodávateľ
3.	0.4	Aktualizovaná verzia knižnice sk.eid:eid-sdk v Mavene	Dodávateľ
4.	0.5	Aktualizované konfigurácia knižnice pre Android. Overenie ID tokenu.	Dodávateľ
5.	0.6	Doplnenie eIDEnvironment konfigurácie pre iOS	Dodávateľ
6.	0.7	Úprava API pre certifikáty	Dodávateľ
7.	0.8	API pre tutoriál	Dodávateľ
8.	0.9	API - dešifrovanie	Dodávateľ
9.	1.0	Doplnená špecifikácia ID token	Dodávateľ
10.	1.0.1	Update po bezpečnostných a UI/UX testoch: Tutoriál API, zabezpečenie SDK iOS pinning, Jazyky	Dodávateľ
11.	1.0.2	Update chybových kódov iOS (nový kód sessionTimeout), update iOS pinned domains.	Dodávateľ
12.	1.0.3	Update SDK a API po úpravách backendu. Update chybových kódov. Doplnená nová dependency pre iOS. Doplnené odporúčania z UX testov a bezpečnostných testov	Dodávateľ
13.	1.0.4	Pridané 2 nové chybové kódy v iOS mSDK	Dodávateľ
14.	1.0.5	Úprava kapitoly 5 a 5.3	Dodávateľ
15.	1.0.6	Úprava kapitoly 1	Dodávateľ
16.	1.0.7	Úprava kapitoly 4	Dodávateľ

**Účel dokumentu:**

V tomto dokumente je popísaná integrácia eID SDK pre mobilné zariadenia s operačnými systémami iOS a Android. SDK zabezpečuje komunikáciu s úradným autentifikátorom (eID 2.0 a eDoPP 2.0) s duálnym rozhraním na mobilných zariadeniach s NFC.

V Bratislave, dňa: **28.2.2023**

.....  
Zástupca dodávateľa

.....  
Podpis

.....  
Zástupca zadávateľa

.....  
Podpis

## Obsah

<b>1. Úvod .....</b>	<b>5</b>
<b>2. eID SDK – Prehľad funkcií .....</b>	<b>6</b>
2.1 Android .....	6
2.2 iOS .....	6
<b>3. Registrácia, inštalácia a konfigurácia SDK .....</b>	<b>7</b>
3.1 Registrácia.....	7
3.2 Inštalácia a konfigurácia .....	7
3.2.1 Android.....	7
3.2.2 iOS .....	8
3.3 Prostredia.....	11
3.4 Implementácia .....	11
3.4.1 Android.....	11
3.4.2 iOS .....	12
<b>4. Autentifikácia pomocou eID SDK .....</b>	<b>17</b>
4.1 Scenár App2SDK .....	17
4.1.1 Podporované typy autentifikácie .....	17
4.1.2 Príklad volania funkcie a odchytenia ID tokenu / Auth code.....	19
4.1.3 Získanie ID tokenu (Authorization code flow) .....	22
4.1.4 Výstupy (Príklad ID Tokenu) .....	23
4.1.5 Overenie ID Tokenu .....	25
4.2 Scenár Web2App .....	26
4.2.1 Android.....	26
4.2.2 iOS .....	26
4.3 Scenár Desktop2Mobile.....	27
4.3.1 Príklad volania funkcie.....	27
<b>5. Vyhodenie Kvalifikovaného elektronického podpisu .....</b>	<b>29</b>
5.1 Scenár App2SDK .....	29
5.1.1 Príklad volania funkcie a odchytenie výstupu .....	30
5.2 Scenár Web2App .....	39
5.2.1 Príklad volania funkcie.....	39
5.3 Scenár Desktop2Mobile.....	40
5.3.1 Variant A.....	40
5.3.2 Príklad volania funkcie.....	40
5.3.3 Variant B.....	41
5.3.4 Príklad volania funkcie.....	42
<b>6. Dešifrovanie pomocou encryption certifikátu .....</b>	<b>43</b>
6.1 Príklad volania funkcie .....	43

6.1.1	Android.....	43
6.1.2	iOS .....	44
<b>7.</b>	<b>Zobrazenie certifikátov z občianskeho preukazu .....</b>	<b>45</b>
<b>7.1</b>	<b>Príklad volania funkcie .....</b>	<b>45</b>
7.1.1	Android.....	45
7.1.2	iOS .....	46
<b>8.</b>	<b>PIN manažment.....</b>	<b>47</b>
<b>8.1</b>	<b>Príklad volania funkcie .....</b>	<b>47</b>
8.1.1	Android.....	47
8.1.2	iOS .....	48
<b>9.</b>	<b>Zobrazenie tutoriálu .....</b>	<b>49</b>
<b>9.1</b>	<b>Príklad volania funkcie .....</b>	<b>49</b>
9.1.1	Android.....	49
9.1.2	iOS .....	49
<b>10.</b>	<b>Deeplinky a QR kódy .....</b>	<b>50</b>
<b>10.1</b>	<b>Autentifikácia .....</b>	<b>50</b>
<b>10.2</b>	<b>Autorizácia - kvalifikovaný elektronický podpis .....</b>	<b>51</b>
<b>11.</b>	<b>Odporúčania pre integrátora eID mSDK .....</b>	<b>52</b>
<b>11.1</b>	<b>Odporúčania z UX/UI testovania .....</b>	<b>52</b>
<b>11.2</b>	<b>Odporúčania z bezpečnostného testovania .....</b>	<b>53</b>

## 1. Úvod

V tomto dokumente je popísaná integrácia **eID SDK** pre mobilné zariadenia s operačnými systémami **iOS** a **Android**. SDK zabezpečuje komunikáciu s úradným autentifikátorom (eID 2.0 a eDoPP 2.0) s duálnym rozhraním na mobilných zariadeniach s NFC.

SDK poskytuje nasledujúce okruhy **funkcionalít**:

1. **Autentifikácia** osoby na najvyššej úrovni zabezpečenia („Vysoká“) podľa eIDAS
2. **Kryptografické funkcie** s privátnymi kľúčmi na eID
  - a. pre vytvorenie kvalifikovaného elektronického podpisu kľúčom, na ktorý bol vydaný kvalifikovaný certifikát
  - b. pre vytvorenie elektronického podpisu kľúčom, na ktorý bol vydaný podpisový certifikát
  - c. dešifrovanie dát kľúčom, na ktorý bol vydaný šifrovací certifikát
3. **Zobrazenie certifikátov** z občianskeho preukazu
4. **Manažment znalostných faktorov** (BOK, KEP PIN, PUK)

SDK zabezpečuje použitie funkcionalít v troch hlavných **scenároch**:

1. **Desktop2Mobile** – proces je iniciovaný z webovej aplikácie na desktope alebo desktopovej aplikácie a mobilná aplikácia integrujúca eID SDK zrealizuje vykonanie danej funkcionality
2. **Web2APP** – proces je iniciovaný z webovej aplikácie v mobilnom prehliadači a mobilná aplikácia integrujúca eID SDK zrealizuje vykonanie danej funkcionality
3. **App2SDK** – proces je iniciovaný v natívnej aplikácii integrujúcej eID SDK, pričom SDK zrealizuje vykonanie danej funkcionality

V jednotlivých scenároch danej funkcionality SDK **poskytuje**:

- **obrazovky**:
  - s pokynmi pre používateľa
  - zobrazujúce výsledok operácie / dáta
  - pre zadanie vstupu od používateľa (BOK, KEP PIN, CAN, PUK)
- komunikáciu s občianskym preukazom cez **NFC**
- komunikáciu so serverom cez **REST API**
- spracovanie **chybových stavov** a ich komunikovanie používateľovi na UI
- **dynamické správanie scenárov**, na základe stavu znalostných faktorov (napr. pri suspendovanom BOK-u je nutné najskôr zadať CAN, vykonať od-suspendovanie BOK-u, následne zadať BOK a pokračovať v procese)

## 2. eID SDK – Prehľad funkcií

eID SDK poskytuje niekoľko **public funkcií**, za pomoci ktorých je možné spustiť vyššie spomínané funkcionality. V nasledujúcich kapitolách bude názorne zobrazené a vysvetlené použitie funkcií.

### 2.1 Android

- **initialize** – inicializácia knižnice
- **handleQRCode** – spracovanie dát z QR kódu a automatické spustenie príslušného scenáru
- **startAuth** – spustenie procesu autentifikácie
- **getCertificates** – načítanie podpisových certifikátov
- **signData** – podpísanie dát na občianskom preukaze
- **decryptData** – dešifrovanie dát na občianskom preukaze
- **startCertificates** – zobrazenie certifikátov z občianskeho preukazu
- **startPinManagement** – zobrazenie stavu znalostných faktorov (BOK, KEP PIN, PUK) a ich manažment
- **showTutorial** – zobrazenie tutoriálu s návodom na používanie eID s NFC

### 2.2 iOS

- **handleQRCode** – spracovanie dát z QR kódu a automatické spustenie príslušného scenáru
- **handleDeeplink** - spracovanie URL a v prípade podporovanej URL automatické spustenie príslušného scenáru
- **startAuth** – spustenie procesu autentifikácie
- **getCertificates** – načítanie podpisových certifikátov
- **signData** – podpísanie dát na občianskom preukaze
- **decryptData** – dešifrovanie dát na občianskom preukaze
- **startCertificates** – zobrazenie certifikátov z občianskeho preukazu
- **startPinManagement** – zobrazenie stavu znalostných faktorov (BOK, KEP PIN, PUK) a ich manažment
- **showTutorial** – zobrazenie tutoriálu s návodom na používanie eID s NFC

## 3. Registrácia, inštalácia a konfigurácia SDK

### 3.1 Registrácia

Pre každého klienta bude na serveri zaregistrované **Client ID** a vygenerovaný **Client secret**. Pre jednoduchosť pri testovaní sme založili public Client ID **eid\_mobile**, ktoré je možné použiť pre testovacie účely.

### 3.2 Inštalácia a konfigurácia

#### 3.2.1 Android

Mobilné SDK pre operačný systém **Android** je dodané ako local Maven repository, ktorý je potrebný si stiahnuť:

- Do repositories v **settings.gradle** súbore aplikácie je potrebné doplniť:

```
maven {  
    url = "https://maven.pkg.github.com/eIDmSDK/eID-SDK-Android/"  
    credentials {  
        username = "eIDmSDK"  
        password = "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"  
    }  
}
```

- Do dependencies v **build.gradle** súbore aplikácie je potrebné doplniť:

**implementation "sk.eid:eid-sdk:X.X.X"**

Aktuálna verzia uvedená v Maven repository a Github README:

<https://github.com/eIDmSDK/eID-mSDK-Android#readme>

- Následne klik na "Sync project with gradle files"

eID SDK je potrebné pred zavolaním akejkoľvek funkcie **inicializovať**. Inicializáciu je nutné vykonať len raz počas životného cyklu aplikácie (inicializáciu odporúčame vykonať hneď po spustení aplikácie v Application class projektu, kvôli scenárom odchyťavajúcim **deeplinky** a **intenty** z externých aplikácií), pomocou volania:

```
EIDHandler.initialize(this, eIDEnvironment)
```

**this** – application context

**eIDEnvironment** – enum EIDEnvironment, prostredia voči ktorým eID mSDK komunikuje (PLAUT\_DEV, PLAUT\_TEST, MINV\_TEST, MINV\_PROD)

### 3.2.2 iOS

Mobilné SDK pre operačný systém **iOS** je dodané v iOS Framework formáte (eID.framework), ktorý je potrebný importovať do projektu:

- Otvor Xcode
- Drag'n'Drop eID.framework súbor do projektu
- Dopln eID.framework do target dependencies a nastav flag „Embed & Sign“
- V sekcii „Signing & Capabilities“ pridaj „Near Field Communication Tag Reading“, následne bude vytvorený .entitlements súbor, ktorý by mal obsahovať

```
<dict>
  <key>com.apple.developer.nfc.readersession.formats</key>
  <array>
    <string>TAG</string>
  </array>
</dict>
```

- Do Info.plist súboru pridaj NFCReaderUsageDescription s textom popisujúcim účel použitia NFC
- Do Info.plist súboru dopln zoznam podporovaných AIDs na kartách (presný zoznam dodáme podľa podporovaných EID)

```
<key>com.apple.developer.nfc.readersession.iso7816.select-identifiers</key>
<array>
  <string>A00000000770108700A1000FE00000400</string>
  <string>E80704007F00070302</string>
</array>
```

- Pre účely deeplinkovania treba zaregistrovať **eid** schému, je to opäť možné v Xcode grafickom rozhraní v sekcii Target -> Info -> URL Types, alebo manuálne v súbore **Info.plist**

```
<key>CFBundleURLTypes</key>
<array>
  <dict>
    <key>CFBundleTypeRole</key>
    <string>Editor</string>
    <key>CFBundleURLName</key>
    <string>eid</string>
    <key>CFBundleURLSchemes</key>
    <array>
      <string>eid</string>
    </array>
  </dict>
</array>
```

- Pre zabráneniu MITM útokov treba nastaviť CA pinning manuálne v súbore **Info.plist**, pridaním nasledovného snippetu (\*snippet bude aktualizovaný po nasadení TEST a PROD prostredí na Ministerstve Vnútra):

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSPinnedDomains</key>
  <dict>
```



```
<key>eid.plaut.sk</key>
<dict>
  <key>NSIncludesSubdomains</key>
  <true/>
  <key>NSPinnedCAIdentities</key>
  <array>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>r/mIkG3eEpVdm+u/ko/cwxzOMo1bk4TyHilByibiA5E=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>C5+lpZ7tcVwmwQIMcRtPbsQtWLABXhQzejna0wHFr8M=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>cCEWzNi/I+FkZvDg26DtaiOanBzWqPWmazmvNZUCA4U=</string>
    </dict>
  </array>
</dict>
<key>apigw.eid.plaut.sk</key>
<dict>
  <key>NSIncludesSubdomains</key>
  <true/>
  <key>NSPinnedCAIdentities</key>
  <array>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>r/mIkG3eEpVdm+u/ko/cwxzOMo1bk4TyHilByibiA5E=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>C5+lpZ7tcVwmwQIMcRtPbsQtWLABXhQzejna0wHFr8M=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>cCEWzNi/I+FkZvDg26DtaiOanBzWqPWmazmvNZUCA4U=</string>
    </dict>
  </array>
</dict>
<key>login.eid.plaut.sk </key>
<dict>
  <key>NSIncludesSubdomains</key>
  <true/>
  <key>NSPinnedCAIdentities</key>
  <array>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>r/mIkG3eEpVdm+u/ko/cwxzOMo1bk4TyHilByibiA5E=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>C5+lpZ7tcVwmwQIMcRtPbsQtWLABXhQzejna0wHFr8M=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
```

```
<string>cCEWzNi/I+FkZvDg26DtaiOanBzWqPWmazmvNZUCA4U=</string>
</dict>
</array>
</dict>
<key>identity.eid.plaut.sk </key>
<dict>
  <key>NSIncludesSubdomains</key>
  <true/>
  <key>NSPinnedCAIdentities</key>
  <array>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>r/mIkG3eEpVdm+u/ko/cwxzOMolbk4TyHilByibiA5E=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>C5+lpZ7tcVmmwQIMcRtPbsQtWLABXhQzejna0wHFr8M=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>cCEWzNi/I+FkZvDg26DtaiOanBzWqPWmazmvNZUCA4U=</string>
    </dict>
  </array>
</dict>
<key>eidas.minv.sk</key>
<dict>
  <key>NSIncludesSubdomains</key>
  <true/>
  <key>NSPinnedCAIdentities</key>
  <array>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>r/mIkG3eEpVdm+u/ko/cwxzOMolbk4TyHilByibiA5E=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>C5+lpZ7tcVmmwQIMcRtPbsQtWLABXhQzejna0wHFr8M=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>cCEWzNi/I+FkZvDg26DtaiOanBzWqPWmazmvNZUCA4U=</string>
    </dict>
  </array>
</dict>
<key>teidas.minv.sk</key>
<dict>
  <key>NSIncludesSubdomains</key>
  <true/>
  <key>NSPinnedCAIdentities</key>
  <array>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
      <string>r/mIkG3eEpVdm+u/ko/cwxzOMolbk4TyHilByibiA5E=</string>
    </dict>
    <dict>
      <key>SPKI-SHA256-BASE64</key>
```

```
<string>C5+lpZ7tcVmwWQIMcRtPbsQtWLABXhQzejna0wHFr8M=</string>
</dict>
<dict>
  <key>SPKI-SHA256-BASE64</key>
  <string>cCEWzNi/I+FkZvDg26DtaiOanBzWqPWmazmvNZUCA4U=</string>
</dict>
</array>
</dict>
</dict>
</dict>
```

- eID.framework vyžaduje 3 dependencies – iOS knižnicu OpenSSL, Lottie a JWTDecode ktoré môžu byť integrované cez Swift Package Manager, CocoaPods, Carthage alebo ako zbuildovaný framework. URL ku knižniciam a postup pre integráciu:  
<https://github.com/krzyzanowskim/OpenSSL>  
<https://github.com/airbnb/lottie-ios.git>  
<https://github.com/auth0/JWTDecode.swift>
- **Build** project

### 3.3 Prostredia

Na potreby vývoja a testovania poskytujeme viacero prostredí (**eIDEnvironment**):

- **plautDev** – vývojové prostredie dodávateľa
- **plautTest** – testovacie stabilné prostredie dodávateľa
- **minvTest** – testovacie prostredie Ministerstva vnútra
- **minvProd** – produkčné prostredie Ministerstva vnútra

Prostredia definujú konfiguráciu URL a serverov, voči ktorým aplikácia komunikuje.

\*Pozn.: Na vývoj odporúčame používať **plautTest** a **minvTest** a k nim prislúchajúce eID kartičky.

### 3.4 Implementácia

#### 3.4.1 Android

**EIDHandler** – class, public API eID mSDK

**EIDEnvironment** – enum, prostredia voči ktorým eID mSDK komunikuje (PLAUT\_DEV, PLAUT\_TEST, MINV\_TEST, MINV\_PROD)

**EIDCertificateType** – enum, typy certifikátov na karte (ALL, QES, ES, ENC)

## 3.4.2 iOS

### 3.4.2.1 Jazyk SDK

eID mSDK framework podporuje SK a EN jazyk, pričom vychádza zo systémového nastavenia jazyka mobilného zariadenia. Jazyk frameworku neodporúčame nastavovať explicitne (nastavením hodnoty `AppleLanguages`, `AppleLanguage` v `UserDefaults`) nakoľko NFC UI komponenty nerešpektujú toto nastavenie a ich texty sa budú naďalej zobrazovať v systémovom jazyku, pričom budú potom vznikať obrazovky so zmiešanou EN a SK lokalizáciou.

### 3.4.2.2 Public classes

**eIDHandler** – class, public API eID mSDK

**eIDError** – enum, zoznam všetkých chýb z eID mSDK

**eIDLogLevel** – enum, úroveň logovania v eID mSDK, každej inštancii eIDHandlera možno nastaviť úroveň logovania do konzoly (`.verbose`, `.debug`, `.info`, `.warning`, `.error`, `.none`)

**eIDEnvironment** – enum, prostredia voči ktorým eID mSDK komunikuje (`.plautDev`, `.plautTest`, `.minvTest`, `.minvProd`)

**eIDCertificateIndex** – enum, typy certifikátov na karte (`.QES`, `.ES`, `.Encryption`)

### 3.4.2.3 Chybové kódy mSDK a navrhované akcie

Nasledovná tabuľka uvádza zoznam všetkých chybových kódov, ktoré eID mSDK na iOS môže vrátiť, ich vysvetlenie a odporúčaný spôsob reakcie integrujúcej aplikácie.

Chyba	Popis chyby	Reakcia aplikácie integrujúcej eID mSDK na chybu			
		Ignorovanie chyby	Zobrazenie chybovej hlášky bez akcie	Zobrazenie chybovej hlášky s akciami "zopakovať" a "zrušiť"	Zobrazenie chybovej hlášky s akciami "Správa kódov" a "zrušiť"
<b>unknownTag</b>	Priložený NFC tag (karta/zariadenie) nie je podporované		x		
<b>unsupportedCardType</b>	eID karta nie je podporovaná		x		
<b>nfcNotSupported</b>	NFC nie je podporované, komunikácia s eID kartou nebude možná.		x		
<b>jailbreakDetected</b>	Jailbreaknuté zariadenie, eID mSDK by sa nemalo spustiť na takomto zariadení.		x		
<b>certificatesNotIssued</b>	Na eID karte neboli vydané certifikáty a je ich treba vydať na desktopovom eID klientovi.		x		
<b>qrNotSupported</b>	Nascanovaný QR kód nie je podporovaný.		x		
<b>usedTCTokenQRCode</b>	Nascanovaný QR kód už bol použitý (QR kód obsahuje jednorazový token), preto treba vygenerovať nový QR kód pre ďalšie prihlásenie.			x	
<b>deeplinkNotSupported</b>	Deeplink nie je podporovaný.		x		
<b>invalidClientIdOrSecret</b>	Nesprávna konfigurácia aplikácie integrujúcej eID mSDK	x			
<b>unsupportedSignatureScheme</b>	Nesprávna konfigurácia podpisovania aplikácie integrujúcej eID mSDK - zle zadaná podpisová schéma	x			
<b>invalidCertificateIndex</b>	Nesprávna konfigurácia podpisovania aplikácie integrujúcej	x			

	eID mSDK - nesprávny index certifikátu				
<b>unsupportedSigningCertificate</b>	Nepodporovaný certifikát na podpis.	x			
<b>unsupportedDecryptionCertificate</b>	Nepodporovaný certifikát na dekryptovanie	x			
<b>unsupportedSDKVersion</b>	Verzia mSDK nie je podporovaná. Odporúčame integráciu novej verzie.	x			
<b>tagConnectionLost</b>	Komunikácia s kartou bola prerušená (pohnutie eID karty, timeout...)			x	
<b>cancelledByUser</b>	Proces bol zrušený používateľom (kliknutím na tlačidlo Zrušiť)			x	
<b>sessionTimeout</b>	Timeout pri čakaní na priloženie eID karty k telefónu a začatie NFC komunikácie.			x	
<b>certificateReadFailed</b>	Načítanie certifikátov neprebehlo úspešne.			x	
<b>signingFailed</b>	Podpisovanie neprebehlo úspešne.			x	
<b>decryptionFailed</b>	Dekryptovanie neprebehlo úspešne.			x	
<b>authInitFailed</b>	Inicializácia procesu autentifikácie neprebehla úspešne.			x	
<b>authCompletionFailed</b>	Ukončenie procesu autentifikácie neprebehlo úspešne.			x	
<b>unableToReadCodeStates</b>	Nepodarilo sa načítať stavy kódov.			x	
<b>networkError(String)</b>	Chyba v sieťovej komunikácii (timeout / no internet connection / server error).			x	
<b>bokInvalid</b>	BOK bol zadáný nesprávne.			x	
<b>bokSuspended</b>	BOK je suspendovaný, odblokovať sa dá v PIN manažmente.				x
<b>bokBlocked</b>	BOK je blokový, odblokovať sa dá v PIN manažmente.				x
<b>bokNotActivated</b>	BOK nie je aktivovaný,		x		

	odblokovať sa dá na pracovisku polície.				
<b>canInvalid</b>	Nesprávny CAN kód.			x	
<b>mrzInvalid</b>	MRZ string je nesprávny.			x	
<b>kepPinInvalid</b>	Nesprávny Podpisový PIN.			x	
<b>kepPinSuspended</b>	Podpisový PIN je suspendovaný, odblokovať sa dá v PIN manažmente.				x
<b>kepPinBlocked</b>	Podpisový PIN je blokovaný, odblokovať sa dá v PIN manažmente.				x
<b>kepPinNotActivated</b>	Podpisový PIN nie je aktívny, aktivovať sa dá na desktopovom eID klientovi.		x		

Android exceptions:

### **IllegalStateException**

- Chýbajúce povinné údaje funkcie
- Nekorektne vyplnené vstupné údaje funkcie
- Nekorektný QR kód alebo Deeplink

### **DeviceRootedException**

- Používateľ má rootované zariadenie, v takomto prípade z bezpečnostných dôvodov nie je možné v procese pokračovať

### **ServerException / EacFailedException**

- Chyba v komunikácii so serverom:
  - o Autentifikácia – používateľ musí opätovne kliknúť na prihlásiť alebo pregenerovať QR kód
  - o Načítanie dát z OP
  - o Overenie platnosti certifikátu

### **CertificateNotFoundException**

- Na karte nie je vydaný zvolený certifikát

Piny:

- **PINNotActivatedException**
- **PINSuspendedException**
- **PINBlockedException**
- **CANInvalidException**

**UsedTokenException**

- Autentifikácia - naskenovaný QR kód / prijatý deeplink už bol použitý (token je jednorazový) a preto treba vygenerovať nový

**UnsupportedSDKVersionException**

- Verzia mSDK nie je podporovaná, odporúčame aktualizovať na novšiu verziu

Android handluje väčšinu chýb týkajúcich sa PINov, automaticky.  
V prípade zablokovania PINu je potrebné navigovať do sekcie “Správa PIN kódov”.  
V prípade neaktívneho PINu treba navigovať do sekcie “Správa PIN kódov” v desktop klientovi, prípadne na ktorékoľvek oddelenie dokladov PZ SR.



## 4. Autentifikácia pomocou eID SDK

V tejto kapitole je popísaná integrácia autentifikácie pomocou **eID SDK** prostredníctvom úradného autentifikátora (eID 2.0 a eDoPP 2.0) s duálnym rozhraním schopným komunikovať bezkontaktné s mobilnými zariadeniami prostredníctvom NFC rozhrania.

Podporované scenáre sú:

- **App2SDK** - autentifikácia v kombinácii natívna aplikácia < -- > eID SDK
- **Web2App** - autentifikácia v kombinácii mobile web < -- > natívna aplikácia
- **Desktop2Mobile** - autentifikácia v kombinácii desktop web/app < -- > mobilná aplikácia

### 4.1 Scenár App2SDK

Autentifikácia je iniciovaná z **natívnej aplikácie** integrujúcej eID SDK, autentifikácia prebehne v eID SDK a dokončenie prihlasovania prebehne opäť v natívnej aplikácii, z ktorej bol proces spustený.

#### 4.1.1 Podporované typy autentifikácie

OIDC/OAuth rozhranie eID AS podporuje **dva typy** autentifikačného flow-u:

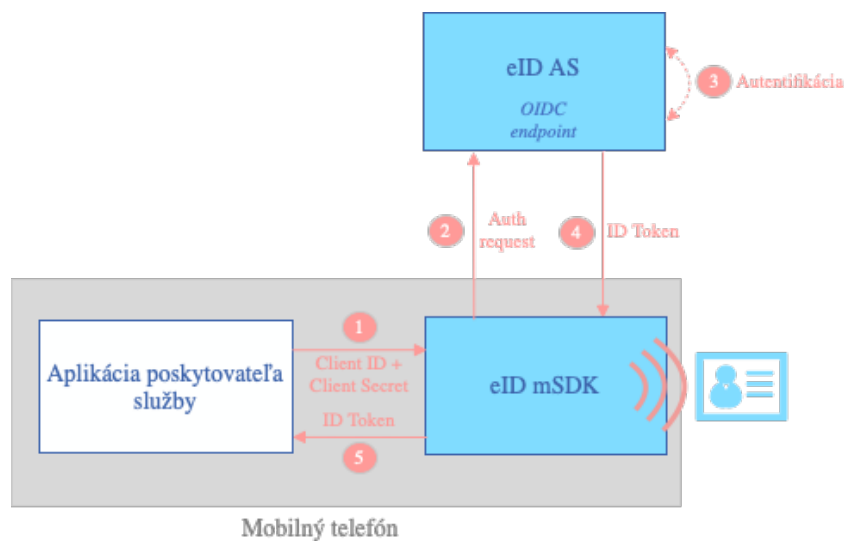
- Implicit Flow
- Authorization Code Flow.

##### 4.1.1.1 Implicit flow

Je určený pre standalone mobilné aplikácie, ktoré nemajú svoj backend na strane serverov. V takom prípade si prístupové údaje (client\_id a client\_secret) k rozhraniu OIDC/OAuth musí bezpečne strážiť samotná mobilná aplikácia.

#### Kroky:

1. Klient vytvorí Authentication request s požadovanými parametrami
2. Klient odošle request na autorizačný server
3. Autorizačný server autentifikuje používateľa
4. Autorizačný server vráti klientovi ID token
5. Klient overí ID token



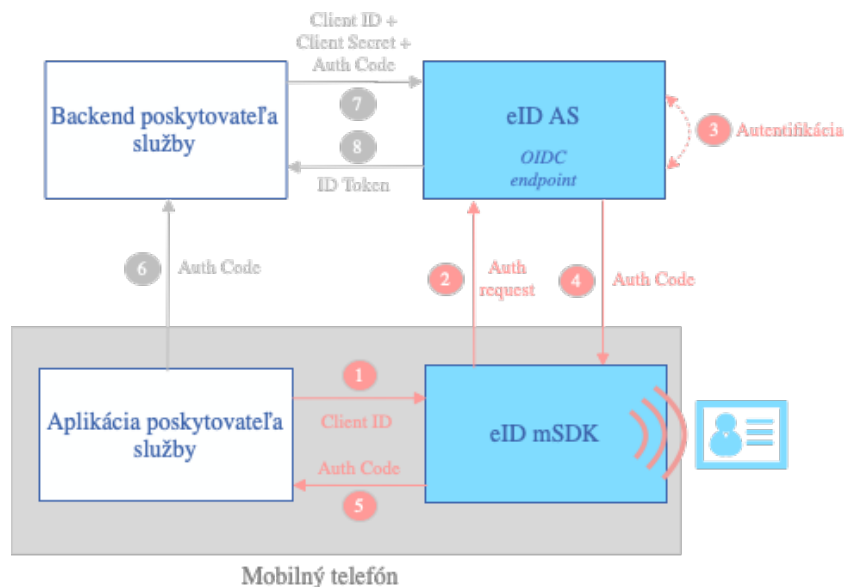
V tomto scenári musí aplikácia integrujúca eID SDK zavolať funkciu **startAuth** a odovzdať jej registrované **Client ID**, **Client secret**, **ApiKeyId** a **ApiKeyValue**. Výstupom procesu je **podpísaný ID token** obsahujúci údaje identity autentifikovaného používateľa.

#### 4.1.1.2 Authorization code flow

Je určený pre mobilné a webové aplikácie. V tomto prípade je dôvera nastavená medzi backendom mobilnej aplikácie a severom eID AS.

##### Kroky:

1. Klient vytvorí Authentication request s požadovanými parametrami
2. Klient odošle request na autorizačný server
3. Autorizačný server autentifikuje používateľa
4. Autorizačný server vráti klientovi autorizačný kód
5. Klient si vyžiada token od autorizačného servera pomocou autorizačného kódu
6. Autorizačný server vráti klientovi ID token
7. Klient overí ID token



V tomto scenári musí aplikácia integrujúca eID SDK zavolať funkciu **startAuth** a odovzdať jej registrované **Client ID**, **ApiKeyId** a **ApiKeyValue**. Výstupom procesu je **authorization code** za pomoci, ktorého si dokáže backend poskytovateľa služby vyžiadať **podpísaný ID token** obsahujúci údaje identity autentifikovaného používateľa.

Viac informácií o OIDC/OAuth, Implicit Flow a Authorization Code Flow v Open ID Connect dokumentácii:

**Open ID Connect:** [https://openid.net/specs/openid-connect-core-1\\_0.html](https://openid.net/specs/openid-connect-core-1_0.html)

**Implicit Flow:** [https://openid.net/specs/openid-connect-core-1\\_0.html#ImplicitFlowAuth](https://openid.net/specs/openid-connect-core-1_0.html#ImplicitFlowAuth)

**Authorization Code Flow:** [https://openid.net/specs/openid-connect-core-1\\_0.html#CodeFlowAuth](https://openid.net/specs/openid-connect-core-1_0.html#CodeFlowAuth)

## 4.1.2 Príklad volania funkcie a odchytenia ID tokenu / Auth code

### 4.1.2.1 Android

Funkcia, v ktorej si klient volí typ autentifikačného flow-u:

```

EIDHandler.startAuth(clientID: String,
    clientSecret: String?,
    apiKeyId: String?,
    apiKeyValue: String?,
    activity: Activity,
    activityLauncher: ActivityResultLauncher<Intent>,
    authenticationFlow: EIDAAuthenticationFlow,
    language: String?,
    nonce: String?)

```

Funkcia, určená pre Implicit flow:

```
EIDHandler.startAuth(clientID: String,  
    clientSecret: String,  
    apiKeyId: String?,  
    apiKeyValue: String?,  
    activity: Activity,  
    activityLauncher: ActivityResultLauncher<Intent>,  
    language: String?  
    nonce: String?)
```

Funkcia, určená pre Authorization code flow:

```
EIDHandler.startAuth(clientID: String,  
    apiKeyId: String?,  
    apiKeyValue: String?,  
    activity: Activity,  
    activityLauncher: ActivityResultLauncher<Intent>,  
    language: String?  
    nonce: String?)
```

Parameter	Hodnota	Povinný
<b>clientID</b>	Registrované Client ID	Áno
<b>clientSecret</b>	Registrovaný Client Secret	Áno – Pre Implicit flow Nie – Pre Authorization code flow
<b>apiKeyId</b>	Registrované ID API kľúča pre gateway	Nie
<b>apiKeyValue</b>	Registrovaná hodnota API kľúča pre gateway	Nie
<b>activity</b>	Activity, z ktorej je proces vyvolávaný	Áno
<b>activityLauncher</b>	Intent launcher pre možnosť získania ID Tokenu v success scenári alebo Exception v prípade chyby	Áno
<b>language</b>	String, jazyk v ktorom má byť scenár zobrazený (sk/en). V prípade, že nie je uvedený, je zvolený jazyk aplikácie/systémový.	Nie
<b>nonce</b>	String, ktorý je možné vygenerovať a zadať na vstupe a ktorý sa bude následne vyskytovať aj v ID tokene. Nonce zadaný na vstupe musí byť zhodný s tým v ID tokene (bezpečnostné overenie garantujúce, že ID token prislúcha k vyvolanému procesu autentifikácie). V prípade, že nie je zadaný ako vstupný parameter, eID mSDK si vygeneruje vlastný a overí si ho potom s hodnotou nonce v ID tokene automaticky. Možnosť zadať vlastný nonce rozširuje bezpečnostné možnosti	Nie

	integrátora ako daný nonce naviazať napr na timestamp/aplikáciu/usera.	
--	--	--

Výstupom je **ID token** alebo **Auth code** (podľa zvoleného typu autentifikčného flow-u). Výstup je možné získať cez **activityLauncher**, ako string s kľúčom **ID\_TOKEN/AUTH\_CODE**. V prípade chyby je možné Exception získať pomocou kľúča **EXCEPTION**.

#### Result code:

- **RESULT\_OK**
- **RESULT\_CANCELED**

#### Data:

- Parameter name – **ID\_TOKEN (String) / AUTH\_CODE (String)**
- Parameter name – **EXCEPTION (Throwable)**

Príklad získania ID tokenu/Auth code:

```
authenticationLauncher =
registerForActivityResult(ActivityResultContracts.StartActivityForResult()) {
result ->
    if (result.resultCode == Activity.RESULT_OK) {
        // Retrieve ID token from eID SDK
        val idToken = result.data?.getStringExtra("ID_TOKEN")
        // Process ID Token

        // Retrieve Auth code from eID SDK
        val authCode = result.data?.getStringExtra("AUTH_CODE")
        // Process Auth code
    } else if (result.resultCode == Activity.RESULT_CANCELED) {
        // Retrieve exception from eID SDK
    }
}
```

#### 4.1.2.2 iOS

Funkcia, určená pre Implicit OAuth flow:

```
eIDHandler().startAuth(from viewController: UIViewController,
                        environment: eIDEnvironment,
                        clientId: String,
                        clientSecret: String,
                        apiKeyId: String?,
                        apiKeyValue: String?,
                        nonce: String = UUID().uuidString,
                        completion: (Result<String, eIDError>) -> ())
```

Funkcia, určená pre Auth code flow:

```
eIDHandler().startAuth(from viewController: UIViewController,
                        environment: eIDEnvironment,
                        clientId: String,
                        apiKeyId: String?,
                        apiKeyValue: String?,
                        nonce: String = UUID().uuidString,
                        completion: (Result<String, eIDError>) -> ())
```

Parameter	Hodnota	Povinný
<b>viewController</b>	ViewController, z ktorého je proces vyvolávaný	Áno
<b>environment</b>	Prostredie, nad ktorým volanie prebehne - .plautDev, .plautTest, .minvTest, .minvProd. (Vid' <b>eIDEnvironment</b> )	Áno
<b>clientId</b>	Registrované Client ID	Áno
<b>clientSecret</b>	Registrovaný Client Secret <ul style="list-style-type: none"> <li>Iba pre implicit flow</li> </ul>	Áno
<b>apiKeyId</b>	Registrované ID API kľúča pre gateway	Nie
<b>apiKeyValue</b>	Registrovaná hodnota API kľúča pre gateway	Nie
<b>nonce</b>	String, ktorý je možné vygenerovať a zadať na vstupe a ktorý sa bude následne vyskytovať aj v ID tokene. Nonce zadaný na vstupe musí byť zhodný s tým v ID tokene (bezpečnostné overenie garantujúce, že ID token prislúcha k vyvolanému procesu autentifikácie). V prípade, že nie je zadaný ako vstupný parameter, eID mSDK si vygeneruje vlastný a overí si ho potom s hodnotou nonce v ID tokene automaticky. Možnosť zadať vlastný nonce rozširuje bezpečnostné možnosti integrátora ako daný nonce naviazať napr na timestamp/aplikáciu/usera.	Nie
<b>completion</b>	Closure (completion block) volaný po ukončení procesu, ktorý vracia Result<String, eIDError>, teda idToken base64 encoded data, prípadne chybu, ktorá nastala počas procesu	Áno

#### 4.1.3 Získanie ID tokenu (Authorization code flow)

Výsledkom autentifikačného procesu je **Authorization Code**. S týmto kódom je potrebné zavolať službu **POST oidc/token**, ktorá vráti **ID token**.

**Príklad volania služby pre získanie ID tokenu pomocou autorizačného kódu:**

##### Request:

```
POST https://eidas.minv.sk/idp/profile/oidc/token
Content-Type: application/x-www-form-urlencoded
Content-Length: 1130
grant_type=authorization_code&client_id=XYZ&client_secret=XYZ&code=XYZ&scope=openid
&redirect_uri=eid%3A%2F%2FauthResult%3Fsuccess%3Dtrue
```

## Response:

```
OK https://eidas.minv.sk/idp/profile/oidc/token
Date: Sat, 24 Feb 2024 21:01:48 GMT
Server: Apache
Cache-Control: no-store
Pragma: no-cache
Content-Type: application/json;charset=UTF-8
X-Powered-By: ARR/3.0
Content-Length: 2470
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
Keep-Alive: timeout=1800, max=95
Connection: Keep-Alive
{
  "access_token": "AAdz...dVK7",
  "id_token": "eyJr...078A",
  "token_type": "Bearer",
  "expires_in": 600
}
```

### 4.1.4 Výstupy (Príklad ID Tokenu)

#### Header:

```
{
  "kid": "defaultRSASign",
  "alg": "RS256"
}
```

#### Payload:

```
{
  "at_hash": "MIQ0vijobyASzGyUoy5sRA",
  "sub": "EBC01122",
  "birthdate": "1996-10-06",
  "gender": "F",
  "identification_number": "966006/1111",
  "iss": "https://identity.eid.plaut.sk/",
  "BIFO": "AAFF140322",
  "auth_time": 1712922328,
  "issuing_state": "SVK",
  "exp": 1712925931,
  "validuntil": "2025-03-01",
  "iat": 1712922331,
  "docnum": "EBC01122",
  "address": {
    "street_address": "Záhradná 458/C4",
    "country": "SVK",
    "formatted": "Záhradná 458/C4\nŽilina\nSlovakia\nSVK\n452 01\n",
    "locality": "Žilina",
    "region": "Slovakia",
  }
}
```

```
    "postal_code": "452 01"  
  },  
  "issuing_date": "2015-03-01",  
  "issuing_office": "Žilina",  
  "given_name": "Hana",  
  "nonce": "gAYEjvk_VQwTjD7FVSCM3bxXM6dMs4RxeWJFK1YV3W4=",  
  "doctype": "ID",  
  "aud": "https://www.plaut.sk/eDoc/sp-prodMVP1-plaut-t1",  
  "nationality": "SVK",  
  "birthplace": "\\nŽilina\\n\\nSVK\\n\\n",  
  "PCO": "1234567890ABCDEF",  
  "family_name": "Molnarova"  
}
```

**Poznámka:** výsledná množina OIDC atribútov obsiahnutých v ID tokene, ktorá bude pre daného klienta poskytovaná, bude závisieť od dohody a registrácie na eID AS.



### 4.1.5 Overenie ID Tokenu

Endpoint pre získanie RSA Public Key, potrebného na overenie JWT tokenu:

<https://dev.eid.plaut.sk/eDocIdP-eid32/profile/oidc/keyset>

Príklad responsu:

```
{
  "keys": [
    {
      "kty": "RSA",
      "e": "AQAB",
      "use": "sig",
      "kid": "defaultRSASign",
      "n": "yV4pytrqvXkSE0XE8BY5axl--kbib-jXyN1V426H_LKQa-
SuZpkwKUi9n_CUxWqZgMjqhBWqLWz0Q3Dp6nU8WhSD1t8AHqaTG1fHo7uuEz0jHnp_-
WhL_Go4hgs2M5U9hTe0Xh73fUDWjB3jV8vzkUmdC2SiMgzcUz3sMFT7wqpKfoH9Rjlp-hX-
NKXbPLFKD_NefwplglecjzobBTjplK0itKjvd4RuvuZuM66w06NkRxY7lPJz284tf7V86tPwAq8M75sdAkD
dApvtm49x8FfLZY0ZWyEk8Y2WS96WrAtoo3JMPP2GWYkrE4an4AvVbID70f3PuPdgmKJg4VHFQ"
    },
    {
      "kty": "EC",
      "use": "sig",
      "crv": "P-256",
      "kid": "defaultECSign",
      "x": "Il6a-DdXWg6r9ombnVrpzLDbur88jncFkA40w--2NEs",
      "y": "aNp3efo4_wPV00Ux5m2XpW2l3C8zBgKid7Uj0wZVLGE"
    },
    {
      "kty": "RSA",
      "e": "AQAB",
      "use": "enc",
      "kid": "defaultRSAEnc",
      "n": "gMxIbGFgIi6qRT4xWQdaP_BZ50JuV-
hbYTuZh15Q014yYMAeGQfZZ98wFzLUI8AB27mHK7u5vYxxDttt8Z_8a_mU41FroeyVJ0FEQSMl0ze09cfvm
ZmFnfddi4pCdQE05zopw-kc3WJfwr3Dd4igM-5-
DZQjaxue1WtHt4il7TcVIzqrQ6X9YlDz66TobKK4hiUm9cKiNC93vi1zwSovsFry-
ze92yBUn73vwvcQmHkmyxDR_d6nb9qkwC--
F7sbxvBZNZgn5piXSLHBAPhVmiHbg4KPIMW5kDE8by6YYu7lRD4p48l_zSLWDJBspznRR_hX21uqSuGoWb
2dGwgw"
    }
  ]
}
```

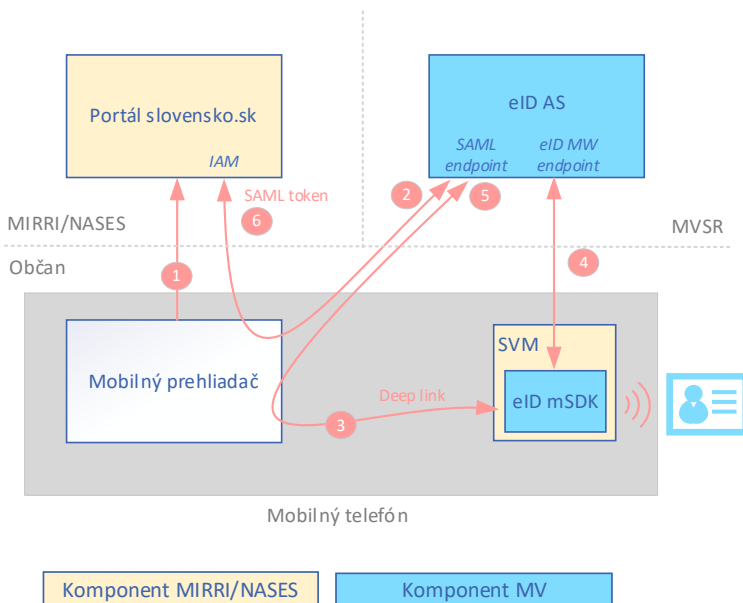
Pomocou Public key je možné vykonať overenie podpisu ID tokenu.

Užitočné linky:

<https://connect2id.com/blog/how-to-validate-an-openid-connect-id-token>

## 4.2 Scenár Web2App

Autentifikácia je iniciovaná z **webovej aplikácie** v **mobilnom** prehliadači, autentifikácia prebehne v aplikácii integrujúcej **eID SDK** a dokončenie prihlasovania opäť prebehne vo **webovej aplikácii** v **mobilnom** prehliadači.



1. Voľba „Prihlásenie“
2. Žiadosť o autentifikáciu
3. Štart SVM/eID
4. Autentifikácia
  - Priloženie eID k NFC
  - Zadanie BOK
  - Získanie identity cez EAC
5. Vytvorenie SAML tokenu
6. Odovzdanie SAML tokenu

### 4.2.1 Android

V tomto scenári, **nie je potrebné** na strane aplikácie integrujúcej eID SDK **robiť volanie funkcie** ani **registrovať filtre** pre odchytyvanie deeplinky. Dodané SDK zabezpečuje túto funkcionálnosť automaticky. Požadovaná je len inicializácia knižnice, ako je uvedené v časti [Inštalácia a konfigurácia](#). Po odchytení deeplinky v požadovanom formáte (viac v kapitole [Deeplinky a QR kódy](#)) je otvorený proces autentifikácie a po jej vykonaní je používateľ automaticky presmerovaný späť na web, kde sa dokončí proces prihlásenia.

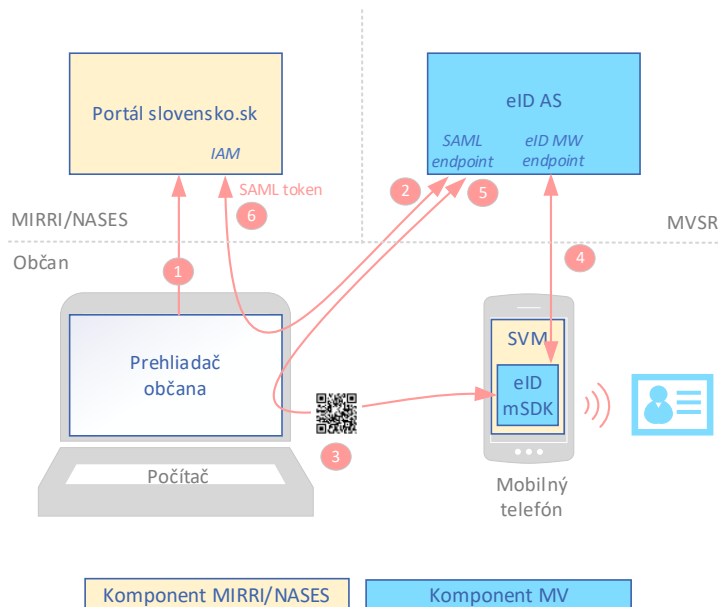
### 4.2.2 iOS

Pre potreby tohto scenáru treba mať správne nakonfigurovanú URL schému (sekcia [Inštalácia a konfigurácia](#)) a následne po odchytení deeplinky v **AppDelegate** prípadne **SceneDelegate** podať túto **url** na spracovanie EIDHandler-u, volaním `eIDHandler().handleDeeplink(url)`

Po odchytení deeplinky v požadovanom formáte (viac v kapitole [Deeplinky a QR kódy](#)) je otvorený proces autentifikácie a po jej vykonaní je používateľ automaticky presmerovaný späť na web, kde sa dokončí proces prihlásenia.

### 4.3 Scenár Desktop2Mobile

Autentifikácia je iniciovaná z **webovej aplikácie** na **desktope**, následne autentifikácia prebieha v aplikácii integrujúcej **eID SDK** a dokončenie prihlasovania opäť prebehne vo **webovej aplikácii** na **desktope**.



1. Voľba „Prihlásenie“
2. Žiadosť o autentifikáciu
3. Štart SVM/eID
4. Autentifikácia
  - Priloženie eID k NFC
  - Zadanie BOK
  - Získanie identity cez EAC
5. Vytvorenie SAML tokenu
6. Odovzdanie SAML tokenu

V tomto scenári musí aplikácia integrujúca eID SDK zabezpečiť **naskenovanie QR kódu** z webového portálu. Naskenovaný QR kód v nezmenenom formáte **odovzdá** eID SDK pomocou volania funkcie **handleQRCode** a SDK zabezpečí spracovanie dát z QR kódu a celý proces autentifikácie. Na konci procesu je vygenerovaný a zobrazený **4 miestny kód**, ktorý používateľ prepíše do webového portálu, kde sa dokončí proces prihlásenia.

#### 4.3.1 Príklad volania funkcie

##### 4.3.1.1 Android

```
EIDHandler.handleQRCode(apiKeyId: String,
    apiKeyValue: String,
    qrCodeData: String,
    activity: Activity,
    activityLauncher: ActivityResultLauncher<Intent>,
    language: String?,
    onError: ((Exception) -> Unit)?)
```

Parameter	Hodnota	Povinný
<b>apiKeyId</b>	Registrované ID API kľúča pre gateway	Áno
<b>apiKeyValue</b>	Registrovaná hodnota API kľúča pre gateway	Áno
<b>qrCodeData</b>	Naskenovaný string	Áno
<b>activity</b>	Activity, z ktorej je proces vyvolávaný	Áno
<b>activityLauncher</b>	Intent launcher pre možnosť získania Activity Result v success scenári alebo Exception v prípade chyby	Áno

<b>language</b>	String, jazyk v ktorom má byť scenár zobrazený (sk/en). V prípade, že nie je uvedený, je zvolený jazyk aplikácie/systémový.	Nie
<b>onError</b>	Handler exceptions odchytených na úrovni EIDHandler-a (pred spustením activity)	Nie

V tomto scenári nie je ID token vracaný aplikácii, nakoľko prihlásenie pokračuje na desktape. Výstupom procesu je **result code** (RESULT\_CANCELED – back button click) a je možné ho odchytiť cez **activityLauncher**,

**Result code** - Activity.RESULT\_CANCELED

#### 4.3.1.2 iOS

```
eIDHandler().handleQRCode(from viewController: UIViewController,
                           qrCodeData: String,
                           apiKeyId: String?,
                           apiKeyValue: String?,
                           completion: (eIDError?) -> ())
```

Parameter	Hodnota	Povinný
<b>viewController</b>	ViewController, z ktorého je proces vyvolávaný	Áno
<b>qrCodeData</b>	Naskenovaný string	Áno
<b>apiKeyId</b>	Registrované ID API kľúča pre gateway	Nie
<b>apiKeyValue</b>	Registrovaná hodnota API kľúča pre gateway	Nie
<b>completion</b>	Closure (completion block) volaný po ukončení procesu, ktorý vracia chybu ak proces neprebehol úspešne alebo nil	Áno

V tomto scenári nie je ID token vracaný aplikácii, nakoľko prihlásenie pokračuje na desktape. Výstupom procesu je prípadná chyba.

## 5. Vyhodenie Kvalifikovaného elektronického podpisu

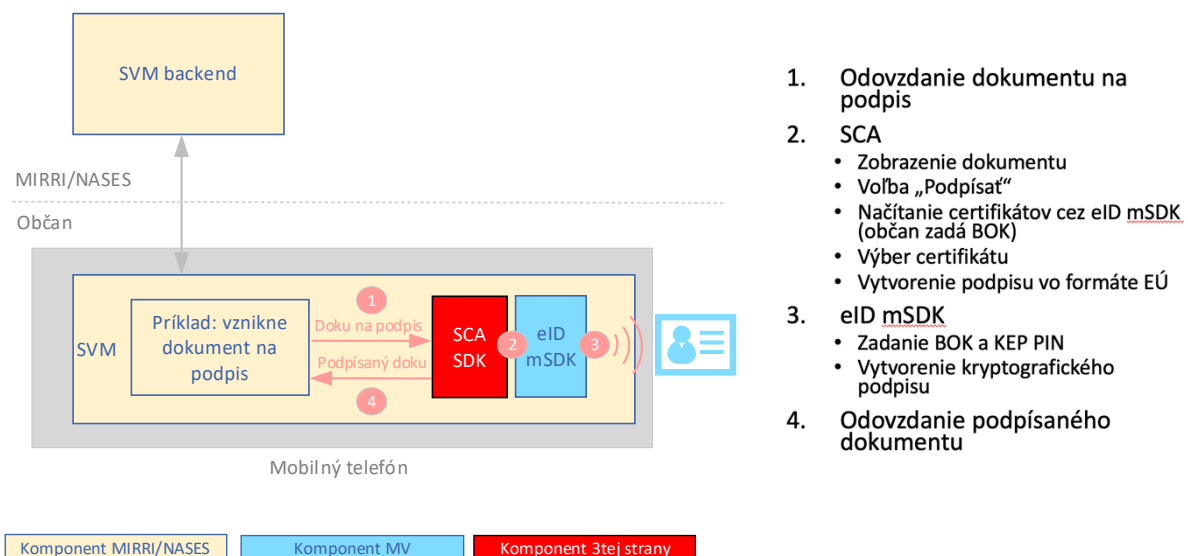
V tejto kapitole sú popísané funkcie eID SDK pre integráciu funkcionality vyhotovenia kvalifikovaného elektronického podpisu pomocou **občianskeho preukazu s bezkontaktným čipom**.

eID SDK poskytuje potrebné funkcie pre mobilnú SCA (Signature Creation Application) ako **načítanie certifikátov** z občianskeho preukazu, **overenie certifikátu** a **podpísanie hash-u dokumentu** s použitím privátneho kľúča bezpečne uloženého v čipe občianskeho preukazu.

Funkcionalita podpisovej aplikácie SCA nie je súčasťou eID SDK.

### 5.1 Scenár App2SDK

Autorizácia je iniciovaná z **natívnej aplikácie** integrujúcej eID SDK, pričom eID SDK poskytuje funkcionality pre načítanie dostupných certifikátov z občianskeho preukazu a podpis dát na občianskom preukaze.



Aplikácia integrujúca eID SDK musí zabezpečiť **vytvorenie hash-u** dokumentu pomocou **vlastnej SCA**. Pre načítanie certifikátov, eID SDK poskytuje funkciu **getCertificates**. V tomto scenári prebehne po zadaní BOK-u načítanie certifikátov z občianskeho preukazu. Certifikáty sú vrátené mobilnej aplikácii integrujúcej eID SDK. Po vytvorení hash-u dokumentu, prostredníctvom SCA, je možné hash podpísať volaním funkcie **signData**. Podpísanie dát prebehne po úspešnom **overení podpisového PINu** (KEP PIN). Vytvorenie výslednej obálky podpísaného dokumentu zabezpečí aplikácia prostredníctvom svojej SCA.

Na overenie certifikátu je možné použiť funkciu **verifyCertificate**, ktorá overí zvolený certifikát v rozsahu:

- Overenie certifikačnej cesty
- Overenie časovej platnosti
- Overenie revokácie

## 5.1.1 Príklad volania funkcie a odchytenie výstupu

### 5.1.1.1 Android

#### 5.1.1.1.1 Načítanie certifikátov

```
EIDHandler.getCertificates(certificateType: EIDCertificateType,  
    activity: Activity,  
    activityLauncher: ActivityResultLauncher<Intent>),  
    language: String?)
```

Parameter	Hodnota	Povinný
<b>certificateType</b>	Typ certifikátu, ktoré chceme načítať. Podporované typy sú QES, ES, ENC, ALL (viď <b>EIDCertificateType</b> )	
<b>activity</b>	Activity, z ktorej je proces vyvolávaný	Áno
<b>activityLauncher</b>	Intent launcher pre možnosť získania Activity Result v success scenári alebo Exception v prípade chyby	Áno
<b>language</b>	String, jazyk v ktorom má byť scenár zobrazený (sk/en). V prípade, že nie je uvedený, je zvolený jazyk aplikácie/systémový	Nie

Výstupom je **JSON** (štruktúra JSONU je uvedená nižšie), ktorý je možné získať cez **activityLauncher**, ako string s kľúčom CERTIFICATES. V prípade chyby je možné Exception získať pomocou kľúča EXCEPTION.

**Result code:**

- RESULT\_OK
- RESULT\_CANCELED

**Data:**

- Parameter name – **CERTIFICATES (String)**
- Parameter name – **EXCEPTION (Throwable)**

Príklad získania certifikátov:

```
getCertificatesLauncher =  
    registerForActivityResult (ActivityResultContracts.StartActivityForResult()) {  
        result ->  
            if (result.resultCode == Activity.RESULT_OK) {  
                val certificatesJson =  
                    result.data?.getStringExtra("CERTIFICATES")  
                // Process JSON  
            }  
    }
```

#### 5.1.1.1.2 Overenie certifikátu

```
EIDHandler.verifyCertificate(certificateEncoded: String,  
    onSuccess: ((String) -> Unit),  
    onError: ((Exception) -> Unit))
```

Parameter	Hodnota	Povinný
<b>certificateEncoded</b>	Base64 encoded certifikát	Áno
<b>onSuccess</b>	Success handler, výstupom je JSON String	Áno
<b>onError</b>	Exceptions handler, výstupom je exception	Áno

Výstupom je **JSON** (štruktúra JSONU je uvedená nižšie), ktorý je možné získať cez **activityLauncher**, ako string s kľúčom VERIFICATION. V prípade chyby je možné Exception získať pomocou kľúča EXCEPTION.

Príklad získania výsledku verifikácie:

```
EIDHandler.verifyCertificate(certificate!!.certificateDataEncoded, {  
    // Handle JSON  
}, {  
    // Handle exception  
})
```

### 5.1.1.1.3 Podpis dát

```
EIDHandler.startSign(certIndex: Int,  
    signatureScheme: String,  
    dataToSign: String,  
    activity: Activity,  
    activityLauncher: ActivityResultLauncher<Intent>,  
    Language: String?)
```

Parameter	Hodnota	Povinný
<b>certIndex</b>	Index certifikátu, získaný z funkcie <code>getCertificates</code> , ktorým majú byť dáta podpísané	Áno
<b>signatureScheme</b>	Podpisová schéma, získaná z funkcie <code>getCertificates</code> (môže byť použitá len schéma, ktorú daný certifikát podporuje)	Áno
<b>dataToSign</b>	Base64 encoded dáta na podpis	Áno
<b>activity</b>	Activity, z ktorej je proces vyvolávaný	Áno
<b>activityLauncher</b>	Intent launcher pre možnosť získania Activity Result v success scenári alebo Exception v prípade chyby	Áno
<b>language</b>	String, jazyk v ktorom má byť scenár zobrazený (sk/en). V prípade, že nie je uvedený, je zvolený jazyk aplikácie/systémový	Nie

Výstupom sú **podpísané** Base64 encoded **dáta**, ktoré je možné získať cez **activityLauncher**, ako string s kľúčom `SIGNED_DATA`. V prípade chyby je možné Exception získať pomocou kľúča `EXCEPTION`.

#### Status:

- `Activity.RESULT_OK`
- `Activity.RESULT_CANCELED`

#### Data:

- Parameter name – **SIGNED\_DATA (String)**
- Parameter name – **EXCEPTION (Throwable)**

Príklad získania podpísaných dát:

```
signLauncher =  
    registerForActivityResult(ActivityResultContracts.StartActivityForResult()) {  
        result ->  
            if (result.resultCode == Activity.RESULT_OK) {  
                val signedDataEncoded =  
                    result.data?.getStringExtra("SIGNED_DATA")  
                // Process data  
            }  
    }
```



## 5.1.1.2 iOS

### 5.1.1.2.1 Načítanie certifikátov

```
eIDHandler().getCertificates(from viewController: UIViewController,
                             types: [eIDCertificateIndex],
                             completion: (Result<String, eIDError>) -> ())
```

Parameter	Hodnota	Povinný
<b>viewController</b>	ViewController, z ktorého je proces vyvolávaný	Áno
<b>types</b>	Pole typov certifikátov, ktoré chceme načítať. Podporované typy sú .ES, .QES, .Encryption. (viď <b>eIDCertificateIndex</b> )	Áno
<b>completion</b>	Closure (completion block) volaný po ukončení procesu, ktorý vracia Result<String, eIDError>, teda JSON string obsahujúci certifikáty podľa štruktúry popísanej vyššie, prípadne chybu, ktorá nastala počas procesu	Áno

### 5.1.1.2.2 Overenie certifikátu

```
eIDHandler().verifyCertificate(from viewController: UIViewController,
                               environment: eIDEnvironment,
                               certificateBase64String: String,
                               completion: (Result<String, eIDError>) -> ())
```

Parameter	Hodnota	Povinný
<b>viewController</b>	ViewController, z ktorého je proces vyvolávaný	Áno
<b>environment</b>	Prostredie, nad ktorým volanie prebehne - .plautDev, .plautTest, .minvTest, .minvProd. (Viď <b>eIDEnvironment</b> )	Áno
<b>certificateBase64String</b>	Base64 encoded certifikát	Áno
<b>completion</b>	Closure (completion block) volaný po ukončení procesu, ktorý vracia Result<String, eIDError>, teda JSON string obsahujúci výsledok overenia, prípadne chybu, ktorá nastala počas procesu	Áno

### 5.1.1.2.3 Podpis dát

```
eIDHandler().signData(from viewController: UIViewController,
                      certIndex: Int,
                      signatureScheme: String,
                      dataToSign: String,
                      completion: (Result<String, eIDError>) -> ())
```

Parameter	Hodnota	Povinný
<b>viewController</b>	ViewController, z ktorého je proces vyvolávaný	Áno
<b>signatureScheme</b>	Podpisová schéma, získaná z funkcie <code>getCertificates</code> (môže byť použitá len schéma, ktorú daný certifikát podporuje)	Áno
<b>dataToSign</b>	Dáta ako base64 encoded string na podpis	Áno
<b>certIndex</b>	Index certifikátu (získaný z <code>getCertificates</code> ), ktorým sa majú dáta podpísať	Áno
<b>completion</b>	Closure (completion block) volaný po ukončení procesu, ktorý vracia <code>Result&lt;String, eIDError&gt;</code> , teda podpísané dáta ako base64 encoded string, prípadne chybu, ktorá nastala počas procesu	Áno

### 5.1.1.3 Výstupy

Výstupom funkcie **getCertificates** je JSON s dostupnými certifikátmi na občianskom preukaze.



















```
{
  "cardType": "eID",
  "QSCD": true,
  "certificates": [
    {
      "slot": "QES",
      "certIndex": 1,
      "certData": "MIIEKTCCApGgAwIBAgIQSgZY5ITBQKGZFyR5ZN8...",
      "isQualified": true,
      "supportedSchemes": [
        "1.2.840.113549.1.1.1",
        "1.2.840.113549.1.1.11",
        "1.2.840.113549.1.1.12",
        "1.2.840.113549.1.1.13"
      ]
    },
    {
      "slot": "ES",
      "certIndex": 2,
      "certData": "MIIEKTCCApGgAwIBAgIQSgZY5ITBQKGZFyR5ZN8...",
      "isQualified": false,
      "supportedSchemes": [
        "1.2.840.113549.1.1.1",
        "1.2.840.113549.1.1.11",
        "1.2.840.113549.1.1.12",
        "1.2.840.113549.1.1.13"
      ]
    },
    {
      "slot": "ES",
      "certIndex": 3,
      "certData": "MIIEKTCCApGgAwIBAgIQSgZY5ITBQKGZFyR5ZN8...",
      "isQualified": false,
      "supportedSchemes": [
        "1.2.840.113549.1.1.1"
      ]
    }
  ]
}
```

```
}
  ]
}
```

Výstupom funkcie **signData** sú podpísané Base64 encoded dáta.

Výstupom funkcie **verifyCertificate** je JSON s výsledkom overenia.

```
{
  "result": {
    "expiration": "VALID",
    "verification": "UNKNOWN"
  },
  "timestamp": "2022-09-09T16:18:42.363578700Z"
}
```

Verification	Text	Info
CHAIN_FAILED	 Nedôveryhodný	 Zlyhalo overenie certifikačnej cesty. Certifikát nebol vydaný žiadnou z dôveryhodných certifikačných autorít, alebo jeho integrita bola narušená.
GOOD	 Certifikát je platný	 Všetko je v poriadku.
REVOKED	 Zrušený	 Platnosť Vášho certifikátu bola zrušená certifikačnou autoritou. K zrušeniu certifikátu môže dôjsť na základe Vašej žiadosti, alebo na základe rozhodnutia certifikačnej autority v prípade podozrenia na ohrozenie bezpečnosti.
UNKNOWN	 Neznámy	 Certifikačná autorita neeviduje stav tohto certifikátu.
NETWORK_ERROR	 Overenie zlyhalo	 Overenie zlyhalo z dôvodu technickej chyby pri overovaní revokácie certifikátu. Skúste prosím neskôr.
SERVER_ERROR	 Overenie zlyhalo	 Overenie zlyhalo z dôvodu technickej chyby. Overte, či je Váš počítač pripojený k internetu.
SERVICE_UNAVAILABLE	 Nepodarilo sa overiť, služba je nedostupná	 Služba pre overenie stavu certifikátu nie je dostupná, skúste neskôr.  <i>Pozn.: navrhujem rozlišovať stavy, keď občan nie je pripojený do internetu (NETWORK_ERROR) od prípadu, keď nas backendový servis CertificateVerifier nebude vedieť komunikovať s OCSP (SERVICE UNAVAILABLE)</i>
UNABLE_TO_VERIFY	 Nepodarilo sa overiť	 Overenie stavu certifikátu bolo neúspešné.
MISSING_VERIFICATION_INFO	 Vydavateľ neposkytuje službu overenia.	 Aplikácia eID klient overuje stav certifikátu voči službe OCSP, ktorú zvyčajne prevádzkuje certifikačná autorita vydávajúca daný certifikát. Certifikačná autorita, ktorá aktuálne overovaný certifikát vydala, neposkytuje službu OCSP na overenie stavu certifikátu.

Expiration	Text	Info
EXPIRED	 Exspirovaný	 <i>Platnosť certifikátu skončila.</i>
VALID	 Certifikát je platný	 <i>Všetko je v poriadku.</i>
EXPIRES_SOON	 Certifikát je platný, avšak čoskoro expiruje	 <i>Váš certifikát čoskoro expiruje. Odporúčame Vám požiadať o vydanie nového certifikátu.</i>

#### 5.1.1.4 Parametre certifikátu

SupportedSchemes obsahuje **OID** podporovaných podpisových resp. šifrovacích schém, ktoré môžu byť v spojení s daným certifikátom a jeho privátnym kľúčom aplikované. Nasledujúca tabuľka obsahuje zoznam možných schém:

OID	Názov schémy	Popis
1.2.840.113549.1.1.1	rsaEncryption	RSAS-PKCS1-v1_5 encryption scheme
1.2.840.113549.1.1.11	sha256WithRSAEncryption	PKCS#1 version 1.5 signature algorithm with Secure Hash Algorithm 256 (SHA256) and Rivest, Shamir and Adleman (RSA) encryption
1.2.840.113549.1.1.12	sha384WithRSAEncryption	PKCS#1 version 1.5 signature algorithm with Secure Hash Algorithm 384 (SHA384) with Rivest, Shamir and Adleman (RSA) Encryption
1.2.840.113549.1.1.13	sha512WithRSAEncryption	PKCS#1 version 1.5 signature algorithm with Secure Hash Algorithm SHA-512 with Rivest, Shamir and Adleman (RSA) encryption

Obsah parametra **dataToSign** by mal vyzerat' vzhľadom na zvolenú schému nasledovne:

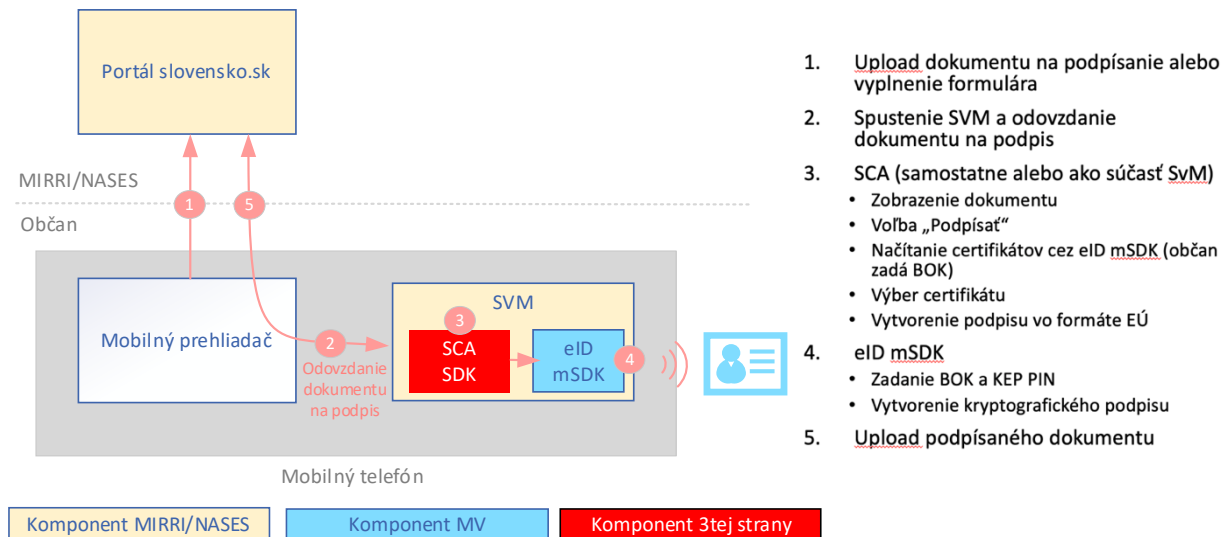
OID	Názov schémy	Obsah parametra dataToSign
1.2.840.113549.1.1.1	rsaEncryption	Obsahuje hodnotu hash zakódovanú v štruktúre DigestInfo podľa RFC 3447
1.2.840.113549.1.1.11	sha256WithRSAEncryption	Obsahuje len hodnotu hash (32 bajtov)
1.2.840.113549.1.1.12	sha384WithRSAEncryption	Obsahuje len hodnotu hash (48 bajtov)
1.2.840.113549.1.1.13	sha512WithRSAEncryption	Obsahuje len hodnotu hash (64 bajtov)

**správanie** SDK/eID karty pri podpisovaní vzhľadom na zvolenú schému:

OID	Správanie SDK / eID Karty
1.2.840.113549.1.1.1 (rsaEncryption)	Dáta budú poslané priamo do RSA podpisovej operácie bez zmeny
1.2.840.113549.1.1.11 (sha256WithRSAEncryption)	SDK / eID karta zakóduje poskytnutú hash hodnotu do štruktúry DigestInfo podľa RFC 3447 s parametrom digestAlgorithm nastaveným na OID 2.16.840.1.101.3.4.2.1 (id-sha256). Takto vytvorená štruktúra je poslaná do RSA podpisovej operácie.
1.2.840.113549.1.1.12 (sha384WithRSAEncryption)	SDK / eID karta zakóduje poskytnutú hash hodnotu do štruktúry DigestInfo podľa RFC 3447 s parametrom digestAlgorithm nastaveným na OID 2.16.840.1.101.3.4.2.2 (id-sha384). Takto vytvorená štruktúra je poslaná do RSA podpisovej operácie.
1.2.840.113549.1.1.13 (sha512WithRSAEncryption)	SDK / eID karta zakóduje poskytnutú hash hodnotu do štruktúry DigestInfo podľa RFC 3447 s parametrom digestAlgorithm nastaveným na OID 2.16.840.1.101.3.4.2.3 (id-sha512). Takto vytvorená štruktúra je poslaná do RSA podpisovej operácie.

## 5.2 Scenár Web2App

Autorizácia je iniciovaná z **webovej aplikácie** v mobilnom prehliadači, pričom autorizácia dokumentu prebehne v **mobilnej aplikácii** integrujúcej eID SDK.



V tomto scenári musí mobilná aplikácia integrujúca eID SDK zabezpečiť **odchytenie deeplinku** z webovej aplikácie, **získať dokument** na podpis (pričom spôsob jeho získania nie je predmetom tohto dokumentu) a rovnako ako v predchádzajúcom scenári musí zabezpečiť vyhotovenie KEP pomocou vlastnej SCA s využitím eID SDK. Pre načítanie certifikátov, eID SDK poskytuje funkciu **getCertificates**. V tomto scenári prebehne po zadaní BOK-u načítanie certifikátov z občianskeho preukazu. Certifikáty sú vrátené mobilnej aplikácii integrujúcej eID SDK. Po vytvorení hash-u dokumentu, prostredníctvom SCA, je možné hash podpísať volaním funkcie **signData**. Podpísanie dát prebehne po úspešnom **overení podpisového PINu** (KEP PIN). Vytvorenie výslednej obálky podpísaného dokumentu zabezpečí aplikácia prostredníctvom svojej SCA, rovnako sa postará aj o **vrátenie** podpísaného dokumentu späť na portál.

### 5.2.1 Príklad volania funkcie

Volanie funkcie je rovnaké ako v predchádzajúcom scenári.

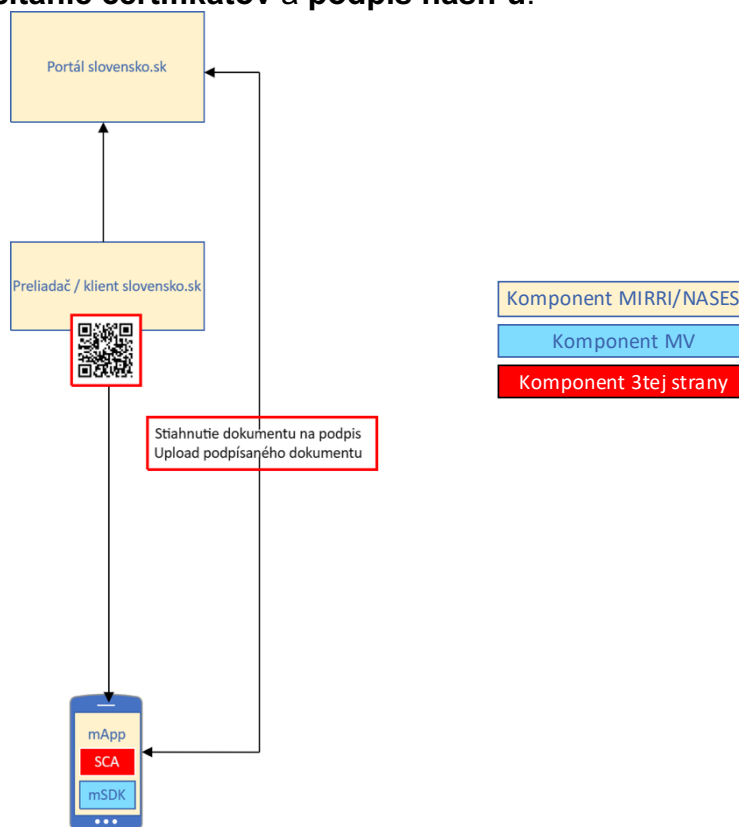
### 5.3 Scenár Desktop2Mobile

Autorizácia je iniciovaná z **webovej aplikácie** na **desktape**, pričom autorizácia (vyhotovenie el. podpisu) dokumentu prebehne v aplikácii integrujúcej **eID SDK**.

V nasledujúcich kapitolách sú popísané varianty, ako možno integrovať eID mSDK v scenároch pre autorizáciu. Samotná implementácia týchto scenárov je **mimo rozsah funkcionality eID mSDK**. Nasledujúce kapitoly obsahujú iba **návrhy možných spôsobov integrácie**.

#### 5.3.1 Variant A

V tomto scenári prebieha komunikácia **priamo** medzi webovým portálom a mobilnou aplikáciou **bez nutnosti** inštalácie ďalších komponentov na desktape, za pomoci QR kódu generovaného priamo webovým portálom. Aplikácia integrujúca eID SDK musí zabezpečiť **naskenovanie QR kódu** z webového portálu, **stiahnutie dokumentu** na podpis, **vytvorenie hash-u** dokumentu pomocou SCA tretej strany, prípadne vlastnou implementáciou a po úspešnom podpise hash-u, vyskladanie obálky podpísaného dokumentu a následný upload na portál. eID SDK zabezpečí komunikáciu s kartou pre **načítanie certifikátov** a **podpis hash-u**.



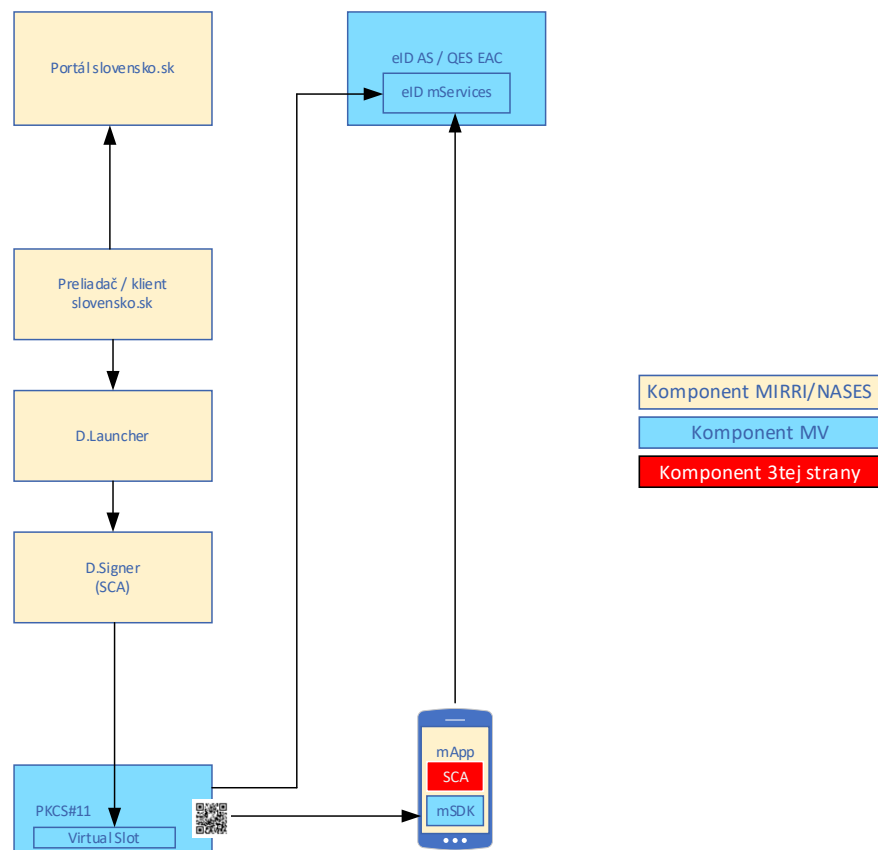
#### 5.3.2 Príklad volania funkcie

Volanie funkcie je rovnaké ako v predchádzajúcom scenári.



### 5.3.3 Variant B

V tomto scenári prebieha komunikácia medzi webovým portálom a mobilnou aplikáciou za pomoci komponentov desktop-ovej **signer aplikácie** a desktop-ového **eID klienta**, ktoré musí mať používateľ nainštalované a správne nakonfigurované. Aplikácia integrujúca eID SDK musí zabezpečiť **naskenovanie QR kódu** a **odovzdanie jeho obsahu do eID SDK** pre spustenie procesu podpisovania.



Aplikácia integrujúca eID SDK musí zabezpečiť **naskenovanie QR kódu**, ktorý sa zobrazí na desktpe. Naskenovaný QR kód v nezmenenom formáte **odovzdá** eID SDK pomocou volania funkcie **handleQRCode**, SDK zabezpečí spracovanie dát z QR kódu a prepojenie s desktop eID klientom (resp jeho modulom PKCS#11) v móde vzdialenej čítačky. Za týmto účelom eID SDK zabezpečí vytvorenie session a komunikáciu medzi desktopom a mobilným telefónom, načítanie a odovzdanie certifikátov signer aplikácii, podpis hash-u dokumentu, interakciu s používateľom pri zadávaní BOK a KEP PIN.

### 5.3.4 Príklad volania funkcie

#### 5.3.4.1 Android

```
EIDHandler.handleQRCode(apiKeyId: String,  
                        apiKeyValue: String,  
                        qrCodeData: String,  
                        activity: Activity,  
                        activityLauncher: ActivityResultLauncher<Intent>)
```

Parameter	Hodnota	Povinný
<b>apiKeyId</b>	Registrované ID API kľúča pre gateway	Áno
<b>apiKeyValue</b>	Registrovaná hodnota API kľúča pre gateway	Áno
<b>qrCodeData</b>	Naskenovaný string	Áno
<b>activity</b>	Activity, z ktorej je proces vyvolávaný	Áno
<b>activityLauncher</b>	Intent launcher pre možnosť získania Activity Result v success scenári alebo Exception v prípade chyby	Áno

V tomto scenári nie je podpísaný dokument vracaný aplikácii, nakoľko proces pokračuje na desktope. Výstupom procesu je **Activity result** (RESULT\_CANCELED – back button click) a je možné ho odchytiť cez **activityLauncher**,

**Result code** - Activity.RESULT\_CANCELED

#### 5.3.4.2 iOS

```
eIDHandler().handleQRCode(from viewController: UIViewController,  
                          qrCodeData: String,  
                          apiKeyId: String?,  
                          apiKeyValue: String?,  
                          completion: (eIDError?) -> ())
```

Parameter	Hodnota	Povinný
<b>viewController</b>	ViewController, z ktorého je proces vyvolávaný	Áno
<b>qrCodeData</b>	Naskenovaný string	Áno
<b>apiKeyId</b>	Registrované ID API kľúča pre gateway	Nie
<b>apiKeyValue</b>	Registrovaná hodnota API kľúča pre gateway	Nie
<b>completion</b>	Closure (completion block) volaný po ukončení procesu, ktorý vracia chybu ak proces neprebehol úspešne alebo nil	Nie

V tomto scenári nie je podpísaný dokument vracaný aplikácii, nakoľko proces pokračuje na desktope. Výstupom procesu je prípadná chyba, ktorá by počas procesu nastala.

## 6. Dešifrovanie pomocou encryption certifikátu

eID SDK poskytuje možnosť dešifrovať dáta encryption certifikátom. Encryption certifikát (jeho verejný kľúč) je možné vyčítať z karty volaním **getCertificates**, aplikácie tretích strán podpíšu týmto certifikátom dáta a eID mSDK je schopné tieto dáta dešifrovať na úrovni eID karty.

### 6.1 Príklad volania funkcie

#### 6.1.1 Android

```
EIDHandler.startDecrypt(dataToDecrypt: String,
    activity: Activity,
    activityLauncher: ActivityResultLauncher<Intent>,
    Language: String?)
```

Parameter	Hodnota	Povinný
<b>dataToDecrypt</b>	Base64 encoded dáta na dešifrovanie	Áno
<b>activity</b>	Activity, z ktorej je proces vyvolávaný	Áno
<b>activityLauncher</b>	Intent launcher pre možnosť získania Activity Result v sucsess scenári alebo Exception v prípade chyby	Áno
<b>language</b>	String, jazyk v ktorom má byť scenár zobrazený (sk/en). V prípade, že nie je uvedený, je zvolený jazyk aplikácie/systémový	Nie

Výstupom sú **dešifrované** Base64 encoded **dáta**, ktoré je možné získať cez **activityLauncher**, ako string s kľúčom DECRYPTED\_DATA. V prípade chyby je možné Exception získať pomocou kľúča EXCEPTION.

#### Status:

- Activity.RESULT\_OK
- Activity.RESULT\_CANCELED

#### Data:

- Parameter name – **DECRYPTED\_DATA** (String)
- Parameter name – **EXCEPTION** (Throwable)

Príklad získania dešifrovaných dát:

```
decryptLauncher =
registerForActivityResult(ActivityResultContracts.StartActivityForResult()) {
result ->
    if (result.resultCode == Activity.RESULT_OK) {
        val decryptedDataEncoded = result.data?.getStringExtra("DECRYPTED_DATA")
        // Process data
    }
}
```

## 6.1.2 iOS

```
eIDHandler().decryptData(from viewController: UIViewController,  
                           certIndex: Int,  
                           dataToDecrypt: String,  
                           completion: (Result<String, eIDError>-> ()))
```

Parameter	Hodnota	Povinný
<b>viewController</b>	ViewController, z ktorého je proces vyvolávaný	Áno
<b>certIndex</b>	Index encryption certifikátu získaný z volania getCertificates. V prípade indexu iného ako encryption certifikátu vráti SDK chybu.	Áno
<b>dataToDecrypt</b>	Base64 encoded dáta na dešifrovanie	Áno
<b>completion</b>	Closure (completion block) volaný po ukončení procesu, ktorý vracia decryptované dáta ako base64 encoded string, alebo chybu ak nastala počas procesu.	Áno

Výstupom funkcie **decryptData** sú decryptované Base64 encoded dáta.

## 7. Zobrazenie certifikátov z občianskeho preukazu

Nakoľko eID SDK v rámci funkcionality vyhotovenia kvalifikovaného elektronického podpisu umožňuje vyčítať certifikáty, je možné využiť aj zobrazenie certifikátov v UI poskytovanom v eID SDK. Proces je možné spustiť zavolaním funkcie **startCertificates**. SDK zabezpečí komunikáciu s občianskym preukazom, pri ktorej prebehne po korektnom zadaní znalostných faktorov, načítanie certifikátov z občianskeho preukazu a overenie certifikátov. Na konci procesu sú načítané dáta certifikátov a výsledok overenia, **zobrazené** používateľovi.

Podporované scenáre sú:

- **App2SDK** – kombinácia natívna aplikácia < -- > eID SDK

### 7.1 Príklad volania funkcie

#### 7.1.1 Android

```
EIDHandler.startCertificates(activity: Activity,  
                             activityLauncher: ActivityResultLauncher<Intent>,  
                             Language: String?)
```

Parameter	Hodnota	Povinný
<b>activity</b>	Activity, z ktorej je proces vyvolávaný	Áno
<b>activityLauncher</b>	Intent launcher pre možnosť získania Activity result v success scenári alebo Exception v prípade chyby	Áno
<b>language</b>	String, jazyk v ktorom má byť scenár zobrazený (sk/en). V prípade, že nie je uvedený, je zvolený jazyk aplikácie/systémový	Nie

V tomto scenári nie sú údaje vračané aplikácii, ale sú zobrazené v UI SDK. Výstupom procesu je **Activity result** (RESULT\_CANCELED – back button click) a je možné ho odchytiť cez **activityLauncher**.

**Result code** – Activity.RESULT\_CANCELED

## 7.1.2 iOS

```
eIDHandler().startCertificates(from viewController: UIViewController,  
                               environment: eIDEnvironment,  
                               completion: (eIDError?) -> ())
```

Parameter	Hodnota	Povinný
<b>viewController</b>	ViewController, z ktorého je proces vyvolávaný	Áno
<b>environment</b>	Prostredie, nad ktorým volanie prebehne - .plautDev, .plautTest, .minvTest, .minvProd. (Vid' <b>eIDEnvironment</b> )	Áno
<b>completion</b>	Closure (completion block) volaný po ukončení procesu, ktorý vracia chybu, ak nastala počas procesu alebo nil	Áno

V tomto scenári nie sú údaje vračané aplikácii, ale sú zobrazené v UI SDK.  
Výstupom procesu je **prípadná chyba**.

## 8. PIN manažment

eID SDK poskytuje kompletnú funkcionálnosť s vlastným UI pre manažment znalostných faktorov (BOK, KEP PIN, PUK). Proces je možné spustiť zavolaním funkcie **startPinManagement**. Po zavolaní tejto funkcie SDK zabezpečí komunikáciu s občianskym preukazom, pri ktorej prebehne načítanie stavov znalostných faktorov. Na základe načítaných stavov je používateľovi zobrazené menu s dostupnými funkciami manažmentu znalostných faktorov – napr. Zmena BOK, Odblokovanie BOK, Odsuspendovanie BOK, Zmena KEP PIN, Odblokovanie KEP PIN, zmena PUK.

Podporované scenáre sú:

- **App2SDK** – kombinácia natívna aplikácia < -- > eID SDK

### 8.1 Príklad volania funkcie

#### 8.1.1 Android

```
EIDHandler.startPinManagement(activity: Activity,  
                                activityLauncher: ActivityResultLauncher<Intent>,  
                                language: String?)
```

Parameter	Hodnota	Povinný
<b>activity</b>	Activity, z ktorej je proces vyvolávaný	Áno
<b>activityLauncher</b>	Intent launcher pre možnosť získania Activity result v success scenári alebo Exception v prípade chyby	Áno
<b>language</b>	String, jazyk v ktorom má byť scenár zobrazený (sk/en). V prípade, že nie je uvedený, je zvolený jazyk aplikácie/systémový	Nie

V tomto scenári nie sú údaje vračané aplikácii, ale sú zobrazené v UI SDK. Výstupom procesu je **Activity result** (RESULT\_CANCELED – back button click) a je možné ho odchytiť cez **activityLauncher**.

**Result code** - Activity.RESULT\_CANCELED

### 8.1.2 iOS

```
eIDHandler().startPinManagement(from viewController: UIViewController,  
                                completion: (eIDError?) -> ())
```

Parameter	Hodnota	Povinný
<b>viewController</b>	ViewController, z ktorého je proces vyvolávaný	Áno
<b>completion</b>	Closure (completion block) volaný po ukončení procesu, ktorý vracia chybu, ak nastala počas procesu alebo nil	Nie

V tomto scenári nie sú aplikácii vračané žiadne údaje. Výstupom procesu je **prípadná chyba**.



## 9. Zobrazenie tutoriálu

Nakoľko je komunikácia s eID cez NFC veľmi náchylná na chyby (spôsob priloženia eID, stabilita NFC spojenia, pohyby rukou), odporúčame aj na základe UX testov zobraziť používateľom tutoriál, ktorý v pár krokoch vysvetľuje spôsob práce s eID kartou. Tutoriál odporúčame zobraziť pred prvým použitím, poprípade aj pravidelnejšie, napr. po viac procesoch, ktoré skončili neúspešne z dôvodu prerušenia NFC komunikácie.

### 9.1 Príklad volania funkcie

#### 9.1.1 Android

```
EIDHandler.startTutorial(activity: Activity,  
    activityLauncher: ActivityResultLauncher<Intent>,  
    Language: String?)
```

Parameter	Hodnota	Povinný
<b>activity</b>	Activity, z ktorej je proces vyvolávaný	Áno
<b>activityLauncher</b>	Intent launcher pre možnosť získania Activity result v success scenári alebo Exception v prípade chyby	Áno
<b>language</b>	String, jazyk v ktorom má byť scenár zobrazený (sk/en). V prípade, že nie je uvedený, je zvolený jazyk aplikácie/systémový	Nie

V tomto scenári nie sú údaje vračané aplikácii, ale sú zobrazené v UI SDK. Výstupom procesu je **Activity result** (RESULT\_CANCELED – back button click) a je možné ho odchytiť cez **activityLauncher**.

**Result code** - Activity.RESULT\_CANCELED

#### 9.1.2 iOS

```
eIDHandler().showTutorial(from viewController: UIViewController,  
    completion: (() -> ())? = nil)
```

Parameter	Hodnota	Povinný
<b>viewController</b>	ViewController, z ktorého je proces vyvolávaný	Áno
<b>completion</b>	closure (completion block) volaný po ukončení procesu	Nie

V tomto scenári nie je vyžadovaná komunikácia s eID kartou, zobrazuje sa len UI.

## 10. Deeplinky a QR kódy

Pre zabezpečenie komunikácie medzi desktopom/externými aplikáciami (napr. webový prehliadač) a aplikáciou integrujúcou eID SDK bola zadefinovaná štruktúra **QR kódov** (pre komunikáciu medzi desktopom a mobilnou aplikáciou) a **deeplinkov** (komunikácia z externých aplikácií).

### 10.1 Autentifikácia

Podporované spôsoby spustenia procesu identifikácie a autentifikácie používateľa:

- **Deeplink** - možnosť využitia v scenári **Web2App**,
- **QR kód** - možnosť využitia v scenári **Desktop2Mobile**.

**Deeplink** aj **QR kód** sú generované eID autentifikačným systémom (eID AS) MV SR v kontexte prebiehajúcej autentifikácie.

**Deeplink** je spustený v **mobilnom prehliadači** za účelom presmerovania do aplikácie integrujúcej eID SDK. Vid' krok 3. v obr. v kap. **Autentifikácia pomocou eID SDK -> [Scenár Web2App](#)**.

**QR kód** je zobrazený v prehliadači na **desktope/notebooku** za účelom spustenia autentifikačného procesu v eID SDK. Vid' krok 3. v obr. v kap. **Autentifikácia pomocou eID SDK -> [Scenár Desktop2Mobile](#)**. QR kód obsahuje zakódovaný deeplink v rovnakej štruktúre ako v scenári **Web2App**.

Deeplink pre **spustenie autentifikácie** používateľa prostredníctvom **eID SDK v mobile**:

eid://auth

**Parametre:**

Parameter	Hodnota	Povinný	Logika eID SDK ak nie je parameter zadany
<b>tcTokenUrl</b>	string, url endpointu, na ktorom je inicializovaný proces autentifikácie	áno	-
<b>qr</b>	boolean, hodnoty: true / false, pomocou tohto parametru sa rozlišuje:  false = používateľ je z eID app presmerovaný na refreshUrl  true = používateľ je hláškou vyzvaný k pokračovaniu vo web browseri	nie	používateľ je presmerovaný z eID app na refreshUrl

## Príklady:

### 1. deeplink obsahujúci len tcTokenUrl

eid://auth?tcTokenUrl=http%3A%2F%2Flocalhost%3A8080%2Fedoc%2Feac%2Finit%3FtcTokenId%3D4a0dd7337225b8c9dcda

### 2. deeplink s tcTokenUrl aj qr

eid://auth?tcTokenUrl=http%3A%2F%2Flocalhost%3A8080%2Fedoc%2Feac%2Finit%3FtcTokenId%3D4a0dd7337225b8c9dcda&qr=true

## 10.2 Autorizácia - kvalifikovaný elektronický podpis

Keďže implementácia SCA, vytvorenie dokumentu a jeho zobrazenie je na strane mobilnej aplikácie a eID SDK poskytuje len podporné funkcie pre načítanie certifikátov a podpis dát, je na integrátorovi aby si zadefinoval štruktúru deeplinkov a QR kódov a implementoval túto funkcionality na strane mobilnej aplikácie. Ako bolo spomínané v časti [Vyhodenie kvalifikovaného elektronického podpisu](#), v rámci dema je možné poskytnúť samostatné SDK integrujúce mobilnú SCA, ktoré zastrešuje celú funkcionality podpisu dokumentu a informatívneho overenia podpisu. Pre tieto scenáre boli zadefinované Deeplinky, QR kódy, Android intenty, iOS Action extensions a Share extensions, ktoré sú popísané v dokumente „Sign SDK.pdf“ a možno ich v rovnakej štruktúre využiť aj v tomto prípade.

## 11. Odporúčania pre integrátora eID mSDK

eID mSDK knižnice prešli dvoma kolami UX/UI testovania na reálnej vzorke používateľov, rovnako ako aj bezpečnými/penetračnými testami. Postrehy a nálezy, ktoré bolo možné zohľadniť a zapracovať v knižniciach, boli zapracované. Zopár odporúčaní, ktoré zostávajú v rukách integrátora týchto knižníc uvádzame nižšie v tabuľke.

### 11.1 Odporúčania z UX/UI testovania

Dotknutá oblasť	Popis nedostatku	Odporúčanie
<b>Prikladanie karty a úvodný tutoriál</b>	Viacerí respondenti/ky si nevšimli, že sa im pri prvom spustení aplikácie otvoril tutoriál, považovali animácie za pokyn k prikladaniu karty.	Pri spustení tutoriálu manuálne z menu odporúčame rovno zobrazíť tutoriál. Avšak pri automatickom spustení (napr pri prvom použití aplikácie) pred samotným spustením zobrazíť používateľovi otázku, či si chce prejsť tutoriálom (návodom na použitie) aby vedel, že ide prejsť návodom.
<b>Zabudnutý BOK</b>	Časť respondentov/iek hľadala zmenu kódov v časti Osobné údaje.	Údaje na eID, Certifikáty či správa PIN kódov sú 3 samostatné sekcie. Závisí od integrátora eID mSDK kde a ako ich zobrazí.  Na základe testov odporúčame, aby integrujúca aplikácia zobrazila menu Nastavenia, kde budú odkazy na tieto 3 sekcie pod sebou zoradené a teda bude evidentné, že pre zmenu kódov netreba ísť do Osobné údaje ale kliknúť na <b>Správa kódov na OP</b>
<b>Certifikáty</b>	Časť respondentov/iek hľadala "moje" certifikáty v časti Osobné údaje.	Údaje na eID, Certifikáty či správa PIN kódov sú 3 samostatné sekcie. Závisí od integrátora eID mSDK kde a ako ich zobrazí.  Na základe testov odporúčame, aby integrujúca aplikácia zobrazila menu Nastavenia, kde budú odkazy na tieto 3 sekcie pod sebou zoradené a teda bude evidentné, že pre zobrazenie mojich certifikátov netreba ísť do Osobné údaje, ale kliknúť na <b>Moje certifikáty</b>

Kritická funkcionálnosť z pohľadu UX je práve spôsob priloženia eID karty a komunikácie cez NFC. Aj napriek animovanému tutoriálu a možnosti si odskúšať prikladania OP a komunikáciu cez NFC, je potrebné, aby sa používatelia naučili správny spôsob prikladania a držania. Pre väčšinu používateľov bude nepochopiteľné, že OP treba držať niekoľko sekúnd presne bez pohybu, čo je na rozdiel napr. oproti bezkontaktným platbám veľmi nepohodlné a ani úspešnosť nebude stopercentná.

Samotný tutoriál v eID mSDK nemusí byť v niektorých prípadoch dostatočná pomoc, odporúčame teda niekoľko možností, ako môže integrujúca aplikácia pomôcť používateľom správne sa naučiť používať bezkontaktné OP:

- Zobrazíť tutoriál pri/pred prvým použitím eID mSDK
- Komunikovať používateľovi, že sa treba naučiť správne prikladať OP k mobilnému zariadeniu a že každé zariadenie má inú anténu a treba si preto aj viackrát odskúšať, v ktorom mieste prebieha NFC komunikácia bezproblémovo
- Používať analytics nástroje, sledovať počty neúspešných NFC operácií (connection lost) a zobrazíť tutoriál po viac neúspešných pokusoch znova
- Na stránke podpory integrátora natočiť reálne videá z používania aj s komentárom, prípadne ku konkrétnym modelom mobilných zariadení zobrazíť spôsob prikladania OP. Na danú stránku potom odkázať tých používateľov, ktorí dlhodobo majú problémy s prikladaním OP k mobilu.

## 11.2 Odporúčania z bezpečnostného testovania

Dotknutá oblasť	Popis nedostatku	Odporúčanie
<b>Možný únik informácií z automaticky vytváraných snímkov aplikácie presunutej do pozadia (iba iOS)</b>	iOS operačný systém vytvára snímky aktuálne zobrazenej mobilnej aplikácie zakaždým, keď je táto poslaná do pozadia a zachytáva pritom súčasný stav obrazovky aplikácie pre rýchlejšie zobrazovanie pri prechode znovu do popredia. Z takto uložených snímkov môžu uniknúť citlivé osobné, finančné a iné informácie o používateľovi v prípade, že je telefón napríklad odcudzený.	Integrátorovi SDK odporúčame zabrániť zachyteniu citlivých dát napr tým, že sa prekryjú v momente, keď prechádza aplikácia do pozadia.