

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
17.	1.1	Odstránenie podpory QR a deeplinks	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

1. Úvod	5
2. eID SDK – Prehľad funkcií	6
2.1 Android	6
2.2 iOS	6
3. Registrácia, inštalácia a konfigurácia SDK	7
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
4. Autentifikácia pomocou eID SDK	17
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
5. Vyhodenie Kvalifikovaného elektronického podpisu	26
5.1 Scenár App2SDK	26
5.1.1 Príklad volania funkcie a odchytenie výstupu	27
6. Dešifrovanie pomocou encryption certifikátu	36
6.1 Príklad volania funkcie	36
6.1.1 Android.....	36
6.1.2 iOS	37
7. Zobrazenie certifikátov z občianskeho preukazu	38
7.1 Príklad volania funkcie	38
7.1.1 Android.....	38
7.1.2 iOS	39
8. PIN manažment.....	40
8.1 Príklad volania funkcie	40
8.1.1 Android.....	40
8.1.2 iOS	41
9. Zobrazenie tutoriálu	42

9.1	Príklad volania funkcie	42
9.1.1	Android.....	42
9.1.2	iOS	42
10.	<i>Odporúčania pre integrátora eID mSDK</i>	43
10.1	Odporúčania z UX/UI testovania	43
10.2	Odporúčania z bezpečnostného testovania	44

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 scenári **App2SDK** – proces je iniciovaný v natívnej aplikácii integrujúcej eID SDK, pričom SDK zrealizuje vykonanie danej funkcionality

V tomto scenári 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
- **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*

- **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 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/cwxzOMo1bk4TyHlByibiA5E=</string>
        </dict>
        <dict>
          <key>SPKI-SHA256-BASE64</key>
          <string>C5+lpZ7tcVmwQIMcRtPbsQtWLABXhQzejna0wHFr8M=</string>
        </dict>
      </array>
    </dict>
  </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/cwxzOMolbk4TyHI1ByibiA5E=</string>
</dict>
<dict>
<key>SPKI-SHA256-BASE64</key>
<string>C5+lpZ7tcVmwQIMcRtPbsQtWLABXhQzejna0wHFr8M=</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/cwxzOMolbk4TyHI1ByibiA5E=</string>
</dict>
<dict>
<key>SPKI-SHA256-BASE64</key>
<string>C5+lpZ7tcVmwQIMcRtPbsQtWLABXhQzejna0wHFr8M=</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/cwxzOMolbk4TyHI1ByibiA5E=</string>
</dict>
</array>
</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>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+lpZ7tcVwmwQIMcRtPbsQtWLABXhQzejna0wHFr8M=</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+lpZ7tcVwmwQIMcRtPbsQtWLABXhQzejna0wHFr8M=</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"
unknownTag	Priložený NFC tag (karta/zariadenie) nie je podporované		x		
unsupportedCardType	eID karta nie je podporovaná		x		
nfcNotSupported	NFC nie je podporované, komunikácia s eID kartou nebude možná.		x		
jailbreakDetected	Jailbreaknuté zariadenie, eID mSDK by sa nemalo spustiť na takomto zariadení.		x		
certificatesNotIssued	Na eID karte neboli vydané certifikáty a je ich treba vydať na desktopovom eID klientovi.		x		
usedTCTokenQRCode	Nascanovaný QR kód už bol použitý (QR kód obsahuje jednorazový token), preto treba vygenerovať nový QR kód pre ďalšie prihlásenie.			x	
invalidClientIdOrSecret	Nesprávna konfigurácia aplikácie integrujúcej eID mSDK	x			
unsupportedSignatureScheme	Nesprávna konfigurácia podpisovania aplikácie integrujúcej eID mSDK - zle zadaná podpisová schéma	x			
invalidCertificateIndex	Nesprávna konfigurácia podpisovania aplikácie integrujúcej eID mSDK - nesprávny index certifikátu	x			
unsupportedSigningCertificate	Nepodporovaný certifikát na podpis.	x			

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

kepPinInvalid	Nesprávny Podpisový PIN.			x	
kepPinSuspended	Podpisový PIN je suspendovaný, odblokovať sa dá v PIN manažmente.				x
kepPinBlocked	Podpisový PIN je blokovaný, odblokovať sa dá v PIN manažmente.				x
kepPinNotActivated	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

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:

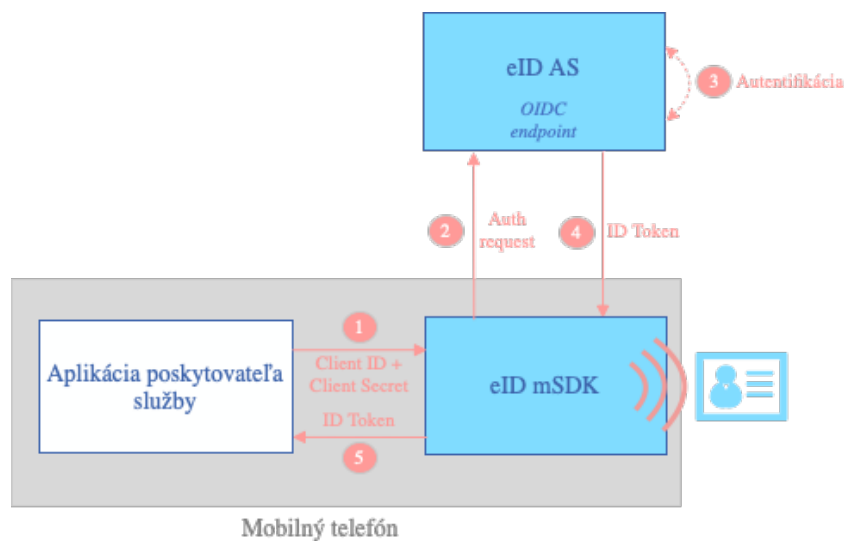
- Implicit Flow
- Authorization Code Flow.

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



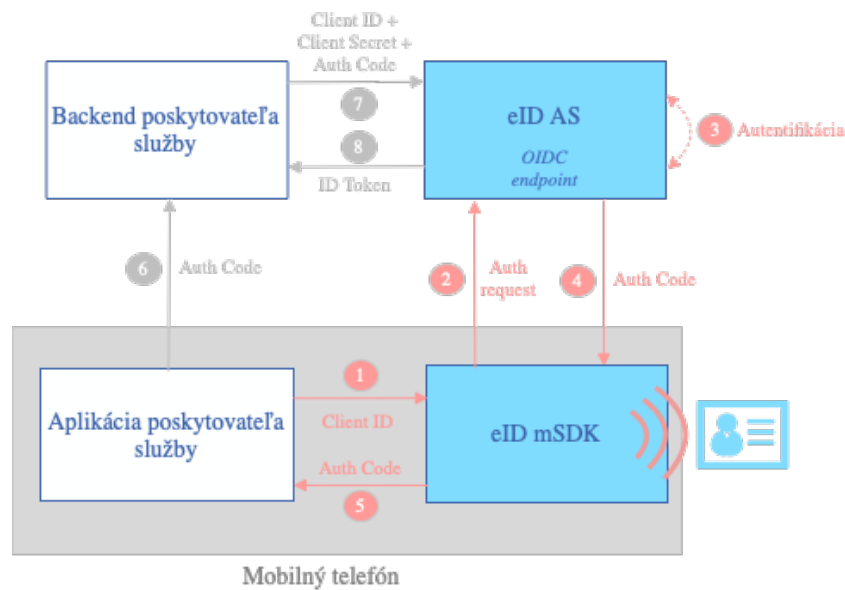
V tomto scenári musí aplikácia integrujúca eID SDK zavolať funkciu **startAuth** a odovzdať jej registrované **Client ID**, **Client secret**, **ApiKeyId** a **ApiKeyValue**. Výstupom procesu je **podpísaný ID token** obsahujúci údaje identity autentifikovaného používateľa.

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

Implicit Flow: 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

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
clientID	Registrované Client ID	Áno
clientSecret	Registrovaný Client Secret	Áno – Pre Implicit flow Nie – Pre Authorization code flow
apiKeyId	Registrované ID API kľúča pre gateway	Nie
apiKeyValue	Registrovaná hodnota API kľúča pre gateway	Nie
activity	Activity, z ktorej je proces vyvolávaný	Áno
activityLauncher	Intent launcher pre možnosť získania ID Tokenu v success scenári alebo Exception v prípade chyby	Áno
language	String, jazyk v ktorom má byť scenár zobrazený (sk/en). V prípade, že nie je uvedený, je zvolený jazyk aplikácie/systémový.	Nie
nonce	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
viewController	ViewController, z ktorého je proces vyvolávaný	Áno
environment	Prostredie, nad ktorým volanie prebehne - .plautDev, .plautTest, .minvTest, .minvProd. (Vid' eIDEnvironment)	Áno
clientId	Registrované Client ID	Áno
clientSecret	Registrovaný Client Secret <ul style="list-style-type: none"> Iba pre implicit flow 	Áno
apiKeyId	Registrované ID API kľúča pre gateway	Nie
apiKeyValue	Registrovaná hodnota API kľúča pre gateway	Nie
nonce	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
completion	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_NefwplglecjozbBTjplK0itKjvd4RuvuZuM66w06NkRxY7lPJz284tf7V86tPwAq8M75sdAkD
dApvtm49x8FfLZY0ZWyEk8Y2WS96WrAtoo3JMPP2GWYkrTE4an4AvVbID70f3PuPdgmKJg4VHFQ"
    },
    {
      "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
ZmFnfdidi4pCdQE05zopw-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>

5. Vyhodenie Kvalifikovaného elektronického podpisu

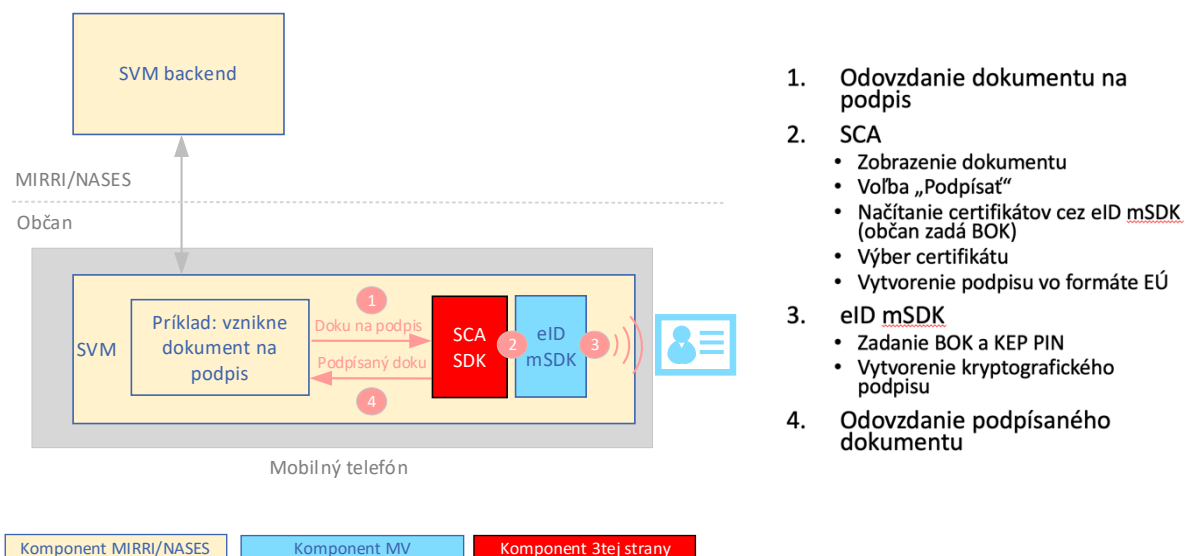
V tejto kapitole sú popísané funkcie eID SDK pre integráciu funkcionality vyhotovenia kvalifikovaného elektronického podpisu pomocou **občianskeho preukazu s bezkontaktným čipom**.

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
certificateType	Typ certifikátu, ktoré chceme načítať. Podporované typy sú QES, ES, ENC, ALL (viď EIDCertificateType)	
activity	Activity, z ktorej je proces vyvolávaný	Áno
activityLauncher	Intent launcher pre možnosť získania Activity Result v success scenári alebo Exception v prípade chyby	Áno
language	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
certificateEncoded	Base64 encoded certifikát	Áno
onSuccess	Success handler, výstupom je JSON String	Áno
onError	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
certIndex	Index certifikátu, získaný z funkcie <code>getCertificates</code> , ktorým majú byť dáta podpísané	Áno
signatureScheme	Podpisová schéma, získaná z funkcie <code>getCertificates</code> (môže byť použitá len schéma, ktorú daný certifikát podporuje)	Áno
dataToSign	Base64 encoded dáta na podpis	Áno
activity	Activity, z ktorej je proces vyvolávaný	Áno
activityLauncher	Intent launcher pre možnosť získania Activity Result v success scenári alebo Exception v prípade chyby	Áno
language	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
viewController	ViewController, z ktorého je proces vyvolávaný	Áno
types	Pole typov certifikátov, ktoré chceme načítať. Podporované typy sú .ES, .QES, .Encryption. (viď eIDCertificateIndex)	Áno
completion	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
viewController	ViewController, z ktorého je proces vyvolávaný	Áno
environment	Prostredie, nad ktorým volanie prebehne - .plautDev, .plautTest, .minvTest, .minvProd. (Viď eIDEnvironment)	Áno
certificateBase64String	Base64 encoded certifikát	Áno
completion	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
viewController	ViewController, z ktorého je proces vyvolávaný	Áno
signatureScheme	Podpisová schéma, získaná z funkcie <code>getCertificates</code> (môže byť použitá len schéma, ktorú daný certifikát podporuje)	Áno
dataToSign	Dáta ako base64 encoded string na podpis	Áno
certIndex	Index certifikátu (získaný z <code>getCertificates</code>), ktorým sa majú dáta podpísať	Áno
completion	Closure (completion block) volaný po ukončení procesu, ktorý vracia <code>Result<String, eIDError></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. 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)
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	 Expirovaný	 <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.

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
dataToDecrypt	Base64 encoded dáta na dešifrovanie	Áno
activity	Activity, z ktorej je proces vyvolávaný	Áno
activityLauncher	Intent launcher pre možnosť získania Activity Result v sucsess scenári alebo Exception v prípade chyby	Áno
language	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
viewController	ViewController, z ktorého je proces vyvolávaný	Áno
certIndex	Index encryption certifikátu získaný z volania getCertificates. V prípade indexu iného ako encryption certifikátu vráti SDK chybu.	Áno
dataToDecrypt	Base64 encoded dáta na dešifrovanie	Áno
completion	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
activity	Activity, z ktorej je proces vyvolávaný	Áno
activityLauncher	Intent launcher pre možnosť získania Activity result v success scenári alebo Exception v prípade chyby	Áno
language	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
viewController	ViewController, z ktorého je proces vyvolávaný	Áno
environment	Prostredie, nad ktorým volanie prebehne - .plautDev, .plautTest, .minvTest, .minvProd. (Vid' eIDEnvironment)	Áno
completion	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
activity	Activity, z ktorej je proces vyvolávaný	Áno
activityLauncher	Intent launcher pre možnosť získania Activity result v success scenári alebo Exception v prípade chyby	Áno
language	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
viewController	ViewController, z ktorého je proces vyvolávaný	Áno
completion	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
activity	Activity, z ktorej je proces vyvolávaný	Áno
activityLauncher	Intent launcher pre možnosť získania Activity result v success scenári alebo Exception v prípade chyby	Áno
language	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
viewController	ViewController, z ktorého je proces vyvolávaný	Áno
completion	closure (completion block) volaný po ukončení procesu	Nie

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

10. 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.

10.1 Odporúčania z UX/UI testovania

Dotknutá oblasť	Popis nedostatku	Odporúčanie
Prikladanie karty a úvodný tutoriál	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.
Zabudnutý BOK	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 Správa kódov na OP
Certifikáty	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 Moje certifikáty

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.

10.2 Odporúčania z bezpečnostného testovania

Dotknutá oblasť	Popis nedostatku	Odporúčanie
Možný únik informácií z automaticky vytváraných snímok aplikácie presunutej do pozadia (iba iOS)	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 snímok 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.