EURÓPSKA ÚNIA
Európsky fond regionálneho rozvoja
OP Integrovaná infraštruktúra 2014 – 2020

MINISTERSTVO DOPRAVY A VÝSTAVBY SLOVENSKEJ REPUBLIKY

MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

MINISTERSTVO VNÚTRA SLOVENSKEJ REPUBLIKY

Dokumentácia

pre špecializovaný technický produkt

Mobilné eID mSDK

Dátum a verzia dokumentu:	17.06.2025 V1.0

OBSAH

1 ÚVOD

1.1 IDENTIFIKÁCIA A URČENIE

Tento dokument obsahuje technickú špecifikáciu integračných rozhraní pre integráciu prihlasovania prostredníctvom Mobilného eID do mobilnej aplikácie 3-tej strany.

Dokument je určený pre vývojársky tím mobilnej aplikácie 3tej strany, ktorý integráciu realizuje po technickej stránke.

1.2 ZMENOVÝ LIST

Verzia	Dátum	Autor	Dôvod zmeny	Kapitola, Bod
1.0	17.06.2025	MV SR	Iniciálna verzia	

1.3 DEFINÍCIA POJMOV A SKRATIEK

Skratka	Vysvetlenie
CSRF	Cross-Site Request Forgery
eID	Elektronická identita
EK	Európska komisia
IdP	Poskytovateľ identity, za angl. Identity Provider
MeID	Mobilné eID
mSDK	Mobilné SDK
MV SR	Ministerstvo vnútra Slovenskej republiky
OAuth	Open Authorization
OIDC	OpenID Connect
Passkey	Passkey je prístupový kľúč pre nový, moderný spôsob prihlasovania, ktorý nahrádza tradičné heslá. Funguje na princípe kryptografie verejného a súkromného kľúča.
SAML	Security Assertion Markup Language
SDK	Softvare Development Kit
SP	Poskytovateľ služby, za angl. Service Provider
SPA	Single Page Application
WebAuthn	Web Authentication

1.4 REFERENCIE

[1] W3C Web Authentication: An API for accessing Public Key Credentials, Level 1,2,3
[2] Nariadenie Európskeho parlamentu a Rady (EÚ) č. 910/2014 z 23. júla 2014 o elektronickej identifikácii a dôveryhodných službách pre elektronické transakcie na vnútornom trhu a o zrušení smernice 1999/93/ES v znení platnom ku 31.1.2024
[3] RFC 6749 „The OAuth 2.0 Authorization Framework"
[4] RFC 6750 „The OAuth 2.0 Authorization Framework: Bearer Token Usage"
[5] OpenID Connect Core 1.0 incorporating errata set 2
[6] SAML 2.0 Security Assertion Markup Language (SAML) V2.0 Technical Overview a súvisiace špecifikácie (Assertions and Protocols (Core), Bindings, Profiles, Conformance Requirements, Metadata, Security and Privacy Considerations)
[7] Mobilné eID, Používateľská príručka
[8] Zákon č. 305/2013 Z. z. o elektronickej podobe výkonu pôsobnosti orgánov verejnej moci a o zmene a doplnení niektorých zákonov (zákon o e-Governmente)
[9] Integračná príručka pre špecializovaný technický produkt Mobilné eID, verzia 1.0

1.5 ZOZNAM OBRÁZKOV

1.6 ZOZNAM TABULIEK

2 ROZSAH PLATNOSTI A ÚČEL

Tento dokument popisuje postup integrácie prihlasovania sa prostredníctvom Mobilného eID (MeID) v prostredí natívnych mobilných aplikácií. Dokument je súčasťou SDK, ktoré pozostáva z nasledujúcich častí:

- Vzorová aplikácia predstavujúca príklad integrácie MeID do natívnej mobilnej aplikácie na platformách Android a iOS,
- Dokumentácia k SDK (tento dokument).

Mobilné eID je moderné riešenie elektronickej identifikácie, ktoré umožňuje používateľom bezpečne sa prihlasovať a autorizovať v online službách prostredníctvom mobilného zariadenia. Mobilné eID je prostriedkom elektronickej identifikácie s úrovňou zabezpečenia „pokročilá" v zmysle nariadenia EÚ č. 910/2014 (nariadenie eIDAS) [2]. Mobilné eID je od 14.2.2025 zapísané v evidencii autentifikačných prostriedkov v zmysle § 22 zákona č. 305/2013 Z. z. o elektronickej podobe výkonu pôsobnosti orgánov verejnej moci a o zmene a doplnení niektorých zákonov (zákon o e-Governmente), [8].

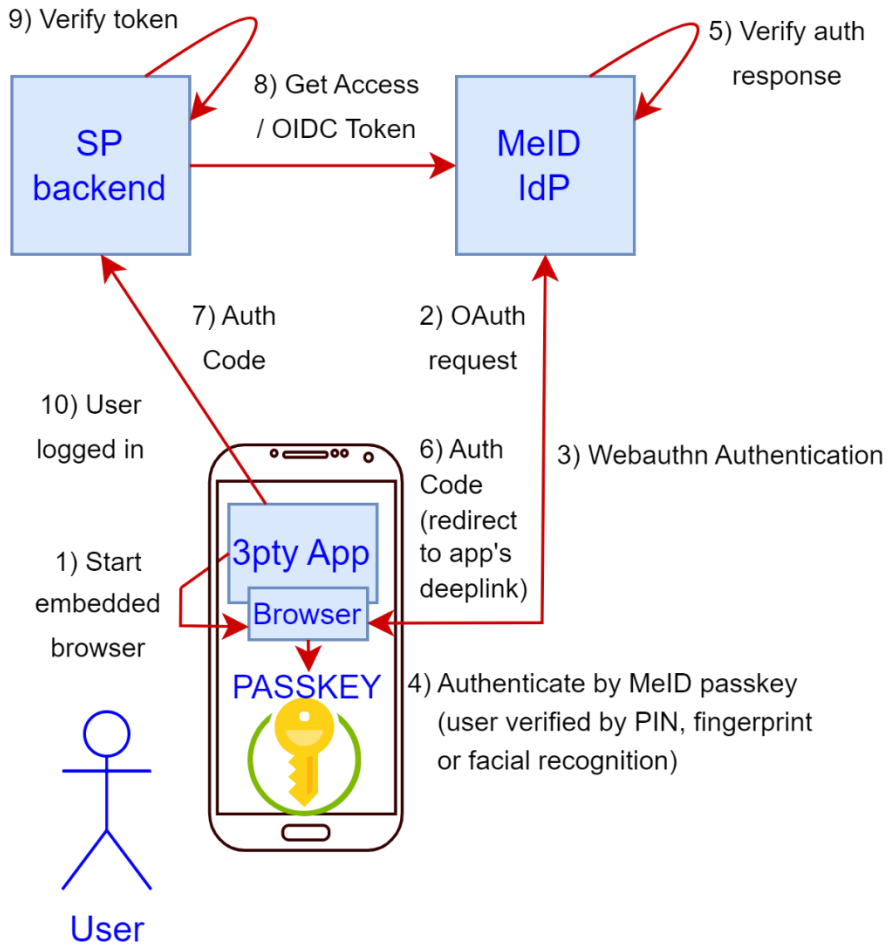
Pred samotnou integráciou MeID do mobilnej aplikácie sa odporúča oboznámiť s:

- konceptom Mobilného eID a jeho základnými vlastnosťami, ktoré sú dostupné na oficiálnej stránke https://meid.minv.sk.
- integračnou príručkou k MeID [9]

3 ARCHITEKTÚRA

Prihlásenie používateľa pomocou MeID do natívnej aplikácie v mobilnom zariadení

Komunikačný kanál pre poskytnutie služby	Natívna aplikácia v mobilnom zariadení
Umiestnenie prístupového kľúča MeID používateľa	Passkey manager v mobilnom telefóne (napr. Správca hesiel Google, Samsung Pass, iCloud kľúčenka)



Postup krokov:

1. Používateľ v mobilnej aplikácii SP zvolí prihlásiť sa. Aplikácia iniciuje autentifikáciu otvorením prehliadača (napr. s použitím Custom Tabs na Androide alebo Safari View Controller na iOS) s URL na systém poskytovateľa identity MeID IdP.
2. Požiadavka o autentifikáciu je poslaná na systém poskytovateľa identity MeID IdP.
3. MeID IdP iniciuje autentifikáciu pomocou MeID prostredníctvom API pre webovú autentifikáciu [1]. Na klienta posiela výzvu (challenge).
4. Používateľ zvolí MeID prístupový kľúč (passkey) pre autentifikáciu uložený v jeho mobilnom zariadení a prístup k nemu potvrdí zadaním PIN, odtlačkom prsta alebo overením tváre. Po overení používateľa je výzva (challenge) podpísaná kľúčom MeID a odpoveď spolu s identifikátorom používateľa je poslaná na server MeID
5. MeID IdP overí podpísanú odpoveď z klienta verejným kľúčom používateľa uloženého v databáze

6. MeID IdP vystaví jednorazový autorizačný kód pre stiahnutie prístupového tokenu (Access token) a ID tokenu a presmeruje prehliadač späť do mobilnej aplikácie SP spolu s autorizačným kódom
7. Mobilná aplikácia SP zašle autorizačný kód svojmu backendu
8. Backend SP požiada MeID IdP o poskytnutie prístupového tokenu (Access token) a ID tokenu, pričom v rámci volania posiela svoju identifikáciu (client_id), tajný údaj (client_secret) a autorizačný kód získaný v predchádzajúcom kroku
9. Backend SP overí poskytnuté tokeny, z ID tokenu získa identitu autentifikovaného používateľa. Používateľ je prihlásený.

Pozn.: Pre uvedený flow MV SR sprístupňuje SDK v podobe vzorovej aplikácie a príslušnej dokumentácie.

4 SYSTÉMOVÉ POŽIADAVKY

Pred samotnou integráciou prihlasovania prostredníctvom MeID do mobilnej aplikácie je potrebné detailne sa oboznámiť s integračnou príručkou k MeID [9].

Najmä je potrebné prejsť procesom registrácie a splniť požiadavky definované v [9], kapitola 6 „SYSTÉMOVÉ POŽIADAVKY".

5 INTEGRÁCIA MEID DO NATÍVNEJ MOBILNEJ APLIKÁCIE

V tejto kapitole je popísaný postup integrácie OAuth / OIDC autentifikácie cez systém MeID v prostredí natívnej mobilnej aplikácie.

5.1 SPUSTENIE AUTENTIFIKÁCIE CEZ EMBEDDOVANÝ PREHLIADAČ

Príklad URL pre spustenie autentifikácie

```
https://tmeid.minv.sk/realms/meid/protocol/openid-connect/auth?client_id=test-client&redirect_uri=sk.test.tmeid%3A%2F%2Faccount&response_mode=query&response_type=code&scope=openid+test_all&state=rCOdqiHHtibxZ18j&nonce=44qmuRv3aXmroPqD&code_challenge=P7O5fOSgo7ktI4KzRIw_l-ftVmEu-vpzSIPMWz70aPU&code_challenge_method=S256
```

5.1.1 Inicializácia procesu autentifikácie na platforme Android

Príklad kódu v jazyku Kotlin pre OS Android pre otvorenie Custom Tabs s URL na MeID IdP pre iniciovanie procesu autentifikácie používateľa prostredníctvom MeID:

```
val uri = url.toUri()

// Create custom tab (you can customize colors here and set specific browser)
val customTabsIntent = CustomTabsIntent.Builder().build()

try {
    // Open custom tab
    customTabsIntent.launchUrl(this, uri)
    finish()
} catch (e: Exception) {
    // Fallback to regular browser
    val intent = Intent(Intent.ACTION_VIEW)
    intent.data = uri
    startActivity(intent)
    finish()
}
```

Na platforme Android je MeID oficiálne podporované v prehliadačoch Chrome, Samsung Internet a Firefox (ale môže fungovať aj v iných prehliadačoch, napr. Vivaldi, Brave). Pri integrácii autentifikácie prostredníctvom MeID v prostredí natívnych mobilných aplikácií sa odporúča pri spúšťaní prehliadača pre autentifikáciu s MeID postupovať nasledovne:

1) nájsť v OS inštalované prehliadače a pokúsiť sa spustiť jeden z oficiálne podporovaných prehliadačov;
2) ak sa nepodarí spustiť prehliadač podľa bodu 1), spustiť defaultný prehliadač.

Uvedený postup je implementovaný v sprievodnej vzorovej aplikácii.

5.1.2 Inicializácia procesu autentifikácie na platforme iOS

Príklad kódu v jazyku Swift pre OS pre vyskladanie URL inicializácie autentifikácie

```
static let authUrl = https://tmeid.minv.sk/realms/meid/protocol/openid-connect/auth
static let clientId = "test-client"
static let clientSecret = "udPb01a5N6f8cq1hMv9IqQIoEE0SMt6S"
static let authRedirectUri = "sk.test.tmeid://account"

func getAuthUrl(state:String, nonce: String) -> URL {
    var urlComps = URLComponents(string: Constants.authUrl)!
    urlComps.queryItems = [URLQueryItem(name: "client_id", value: clientId),
                           URLQueryItem(name: "redirect_uri", value: authRedirectUri),
                           URLQueryItem(name: "response_mode", value: "query"),
                           URLQueryItem(name: "response_type", value: "code"),
                           URLQueryItem(name: "scope", value: "openid"),
                           URLQueryItem(name: "nonce", value: nonce),
                           URLQueryItem(name: "state", value: state),
                           URLQueryItem(name: "code_challenge", value: getPKCE()),
                           URLQueryItem(name: "code_challenge_method", value: "S256")]
    return urlComps.url!
}
```

Kód pre spustenie procesu autentifikácie v SFSafariViewController

```
let authUrl = getAuthUrl(state: UUID().uuidString, nonce: UUID().uuidString)

let safariVC = SFSafariViewController(url: authUrl)
safariVC.modalPresentationStyle = .automatic
present(safariVC, animated: true)
```

5.2 ODCHYTENIE AUTHORIZATION CODE

5.2.1 Odchytenie authorization code v aplikácii na platforme Android

```
override fun onNewIntent(intent: Intent) {
    super.onNewIntent(intent)
    if (intent.action == Intent.ACTION_VIEW) {
        val deeplink = intent.dataString
        // Parse query parameter "code"
    }
}
```

5.2.2 Odchytenie authorization code v aplikácii na platforme iOS

Pre správne odchytenie authorization code musí mať aplikácia zaregistrovanú „custom scheme“ **sk.minv.meid**

Po úspešnej autentifikácii v SFSafariViewController a zaregistrovanej schéme sa na SceneDelegate zavolá delegačná funkcia **scene(_:openURLContexts:)**, v ktorej je potrebné z URL získať query item **code:**

```
func scene(  scene: UIScene, openURLContexts URLContexts: Set<UIOpenURLContext>) {
    guard let deeplink = URLContexts.first?.url else {
        print("unknown url")
        return
    }

    if deeplink.host == "account",
       let urlComponents = URLComponents(url: deeplink, resolvingAgainstBaseURL: true),
       let code = urlComponents.queryItems?.first(where: { $0.name == "code" })?.value {
           // handle the authorization code (sent to server to get the tokens)
    }
}
```

5.3 ZÍSKANIE ID TOKENU

Mobilná aplikácia SP odovzdá autorizačný kód získaný podľa postupu uvedeného v kap. 5.2 Odchytenie authorization code svojej backendovej službe. Backend SP získa ID token volaním príslušného endpointu na MeID IdP, príklad:

```
curl -X POST "https://tmeid.minv.sk/realms/meid/protocol/openid-connect/token" -H
"Authorization: Basic dGVzdC1jbGllbnQ6dWRQYjAxYTVONmY4Y3ExaE12OUlxUUlvRUUwU010NlM="
-H "Content-Type: application/x-www-form-urlencoded" -d
"grant_type=authorization_code&redirect_uri=sk.test.tmeid%3A%2F%2Faccount&code=81cd
fa1d-45c9-4ea1-a825-5477b0859318.1d6f5b9d-aa64-41ec-815e-b6266ef2706b.5af56050-
b1ff-4714-8321-
3b06fc623166&code_verifier=DLNjPHWkYO_dJpR28QWdRUybsEgSpvJNtRVo1J8_vSuDVWuVxqJR_nel
oOb-n8hgWIly5fupWBZBfF5FwsaHdw"
```

6 NÁVRATOVÉ KÓDY

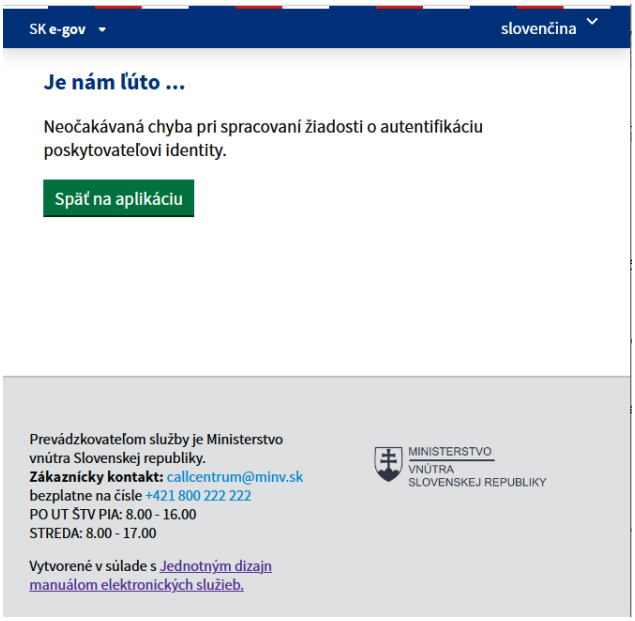
6.1 CHYBY PRI PRIHÁSENÍ

V prípade neúspešného prihlásenia / problémov pri prihlasovaní, je používateľ o dôvode problému informovaný priamo na prihlasovacej stránke MeID IdP.

Na uvedenej stránke je zároveň zobrazené tlačidlo „Pokračovať". Po jeho stlačení je prehliadač používateľa presmerovaný na štandardnú redirect URI aplikácie SP tak, ako bolo uvedené v registračných údajoch SP v rámci jeho registrácie na MeID IdP.

V prípade chyby / problému prihlásenia nie je v rámci spätného presmerovania do URL pridaný žiaden dodatočný parameter.

Vzor obrazovky s informáciou o probléme pri prihlasovaní, ktorú zobrazuje priamo MeID IdP:



Obrázok 1: Vzor obrazovky s informáciou o probléme pri prihlasovaní

6.2 CHYBY PRI ZÍSKANÍ ID TOKENU

V prípade, že na strane IdP nastane nejaká chyba pri spracovaní požiadavky od SP o získanie ID tokenu, vráti IdP štruktúrovanú informáciu o chybe vo formáte JSON nasledovne:

```
{
   "error":"<kód_chyby>",
   "error_description":"<popis_chyby>"
}
```

Nasledovná tabuľka obsahuje zoznam možných hodnôt chybových kódov:

Tabuľka 1: Zoznam možných chybových kódov pri volaní endpoint MeID IdP pre získanie ID tokenu

Hodnota „error"	**Význam**
`invalid_request`	Chýba povinný parameter alebo má chybný formát.
`unauthorized_client`	Klient nemá oprávnenie použiť daný `grant_type` alebo endpoint.
`access_denied`	Používateľ alebo IdP zamietol požiadavku.
`unsupported_response_type`	Server nepodporuje požadovaný `response_type`.
`invalid_scope`	Požadovaný rozsah (`scope`) je neplatný, neexistuje alebo nie je povolený.
`server_error`	Interná chyba servera na strane IdP.
`temporarily_unavailable`	IdP je dočasne nedostupný – preťaženie, údržba atď.
`invalid_client`	Chybná autentifikácia klienta (napr. nesprávny `client_secret`)
`invalid_grant`	Problém s autorizačným kódom, napr. exspirovaný, už použitý, neexistujúci.
`unsupported_grant_type`	Poslaný `grant_type` nie je podporovaný IdP.

Príklad:

```
{
   "error":"invalid_grant",
   "error_description":"Code not valid"
}
```