OID4VC: Izdajanje in deljenje preverljivih poverilnic na osnovi OpenID Connect

Martin Domajnko, Vid Keršič, Muhamed Turkanović

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko,
Maribor, Slovenija
martin.domajnko@student.um.si, vid.kersic@um.si, muhamed.turkanovic@um.si

V današnjem digitalnem okolju je potreba po varnih sistemih za upravljanje identitet postala vse bolj pomembna. Ena takšna rešitev je koncept preverljivih poverilnic. Njihov podatkovni model je tudi definiran kot standard W3C. VC omogočajo dokazovanje in deljenje informacij o digitalni identiteti na kriptografsko preverljiv način. Posameznikom omogočajo tudi nadzor nad njihovimi osebnimi podatki in selektivno razkrivanje le teh za namene preverjanja. Razvoj specifikacij OpenID za izdajanje preverljivih poverilnic je odprla nove možnosti za izboljšanje sistemov za upravljanje identitet. OIDC je sodoben overitveni protokol za varno preverjanje identitete. Uporabnikom omogoča poenostavljen postopek prijave na več spletnih strani s pomočjo uporabe obstoječih računov pri ponudnikih digitalne identitete, kot je na primer Google. V prispevku bomo predstavili specifikacije in izvedbo implementacije OID4VC. Validacija integracije bo izvedena na strežniku implementiranim s pomočjo knjižnice Veramo ter denarnice Masca za sprejem in hrambo VC. S tem uporabimo varen široko razširjen protokol in omogočimo OIDC ponudnikom, da postanejo izdajalci preverljivih poverilnic, uporabnikom pa, da na preprost in varen način prejemajo svoje poverilnice. Implementacija ponuja tudi podporo decentraliziranim identitetam in identifikatorjem. Prispevek zaključimo z analizo prednosti in slabosti, ki jih ima predstavljena rešitev, in pregledom novih integracij, ki jih le ta omogoča.

Ključne besede:
decentralizirana identiteta
digitalna identiteta
samo-upravljana identiteta
preverljive poverilnice
OpenID Connect

Univerzitetna založba Univerze v Mariboru
ISBN 978-961-286-772-0
DOI https://doi.org/10.18690/um.feri.8.2023.13

1 Uvod

Digitalna identiteta je ena izmed najpomembnejših komponent, saj se z njo srečamo na skoraj vsaki spletni strani in služi za identifikacijo uporabnikov na spletu. Že od samega začetka svetovnega spleta ta pretežno temelji na uporabniških imenih in geslih ter se skozi desetletja ni spremenila veliko kljub številnim pomanjkljivostim. Ker naše interakcije in življenje v fizičnem svetu postaja vse bolj prepleteno z digitalnim, potreba po novih, bolj varnih in zanesljivih sistemih za upravljanje digitalnih identitet postaja vse večja. Tradicionalne metode verificiranja digitalnih identitet so pogosto neprijazne in varnostno ranljive, zato se to področje zelo hitro razvija.

Eden izmed novih naprednih načinov za implementacijo sistema za digitalno identiteto je samo-upravljana identiteta (angl. self-sovereign identity, SSI), ki temelju na principih lastništva podatkov s strani uporabnikov, popolno samokontrolo nad upravljanjem identitete in povečano zasebnost uporabnikov [22]. Jedro SSI predstavlja kriptografija in sicer asimetrična kriptografija oz. kriptografija javnega ključa [37], na kateri temelji tudi tradicionalno podpisovanje digitalnih dokumentov kot tudi podpisovanje transakcij pri tehnologiji veriženja blokov (angl. blockchain) [15]. Zaradi podobnih vrednot kot tudi osnovnih tehničnih primitov, sta SSI in blockchain oz. Web3 tehnologije obravnavne skupaj, in sicer kot nova tretja generacija svetovnega spleta, ki razvija internet v smer decentralizacije, digitalnega lastništva in zasebnosti uporabnikov [14].

Ena izmed ključnih komponent SSI so preverljive poverilnice (angl. Verifiable Credential, VC), čigar standard je priporočen tudi s strani organizacije The World Wide Web Consortium (W3C) [40]. Podatkovni model VC služi kot striktno definiran standard za vse možne tipe digitalnih dokumentov in ostale informacije o uporabnikovi identiteti, na primer vozniško dovoljenje, diploma in razni certifikati. VC prav tako omogoča uporabniku deljenje in dokazovanje lastnih informacij na interoperabilen in kriptografsko preverljiv način. Uporaba novih in naprednih kriptografskih pristopov omogoča še povečan nadzor nad podatki in njihovo delitvijo, na primer selektivno razkritje (angl. selective disclosure) in dokazovanje lastništva podatkov namesto njihovih dejanskih vrednosti z uporabo zero-knowledge proofs [40].

Medtem ko sam standard W3C VC definira strukturo digitalnih dokumentov, v njem ni določeno, kako poteka prenos teh dokumentov od uporabnika do entitet, s katerimi je potrebno deliti podatke. V ta namen je bilo razvitih več protokolov, pri čemer sta najbolj znana in uporabljena DIDComm in OpenID protokol za preverljive poverilnice (angl. OpenID for Verifiable Credentials, OID4VC) [3, 18]. Ceprav je protokol DIDComm nastal znotraj SSI skupnosti, se protokol OID4VC navadno pogosteje integrira v sisteme za digitalno identiteto, saj temelji na obstoječem in široko razširjenem protokolu OpenID Connect (OIDC), zaradi česar je njegova integracija lažja in enostavnejša. Samo specifikacijo OID4VC razvija delovna skupina OpenID Connect (angl. OpenID Connect Working Group) in je tako odprla nove možnosti za izboljšanje sistemov za upravljanje digitalne identitete. OIDC je overitvena plast, zgrajena na osnovi protokola OAuth 2.0, in je sodoben overitveni protokol za varno preverjanje identitete in funkcionalnost enkratne prijave (angl. Single Sign-On – SSO) za spletne in mobilne aplikacije [7, 16]. Uporabnikom omogoča poenostavljen postopek prijave na več spletnih strani s pomočjo uporabe obstoječih računov pri večjih ponudnikih digitalne identitete, kot so Google, Facebook in Microsoft. Z dodatno specifikacijo OID4VC je uporabnikom omogočena prijava s SSI.

V prispevku bomo raziskali in predstavili specifikacije ter izvedbo implementacije OID4VC, pri čemer se bomo osredotočili na integracijo samega protokola OIDC. Validacija integracije bo izvedena na strežniku implementiranim z odprto-kodno knjižnico Veramo in digitalno denarnico Masca, s katero bomo prejeli in v njej hranili VC-je. Pri tem uporabimo varen in široko razširjen protokol ter omogočimo OIDC ponudnikom, da postanejo izdajatelji preverljivih poverilnic, uporabnikom pa, da na preprost in varen način prejmejo svoje poverilnice. Implementacija ponuja tudi podporo decentraliziranim identitetam in identifikatorjem (angl. decentralized identifier, DID), kar še dodatno uporabnikom poveča nadzor na svojo osebno digitalno identiteto [2]. Prispevek zaključimo z analizo prednosti in slabosti, ki jih ima predstavljena rešitev in uporaba OID4VC, ter pregledom novih integracij, ki jih le ta omogoča.

2 Osnovni koncepti

2.1 Single Sign-On (SSO)

Single Sign-On (SSO) je mehanizem za avtentikacijo, ki omogoča uporabnikom dostop do več aplikacij oz. storitev z uporabo ene digitalne identitete [1]. Glavni cilj SSO je poenostaviti in izboljšati uporabniško izkušnjo tako, da uporabnikom ni potrebno uporabljati različnih uporabniških in gesel pri uporabi različnih sistemov. S SSO se uporabnik prijavi enkrat, nato pa lahko dostopa do različnih aplikacij, ne da bi moral znova vnašati svoje poverilnice oz. podatke o identiteti. SSO izboljša uporabniško izkušnjo, poveča varnost in zmanjšuje breme za uporabnike, ki ne rabijo več upravljati več prijavnih poverilnic.

Postopek SSO navadno vključuje tri glavne komponente oz. entitete: ponudnika identitete (angl. identity provider, IDP), ponudnika storitev (angl. service provider, SP) in končnega uporabnika. Ko uporabnik poskuša dostopati do storitve, SP preusmeri uporabnika na IDP za overjanje oz. avtentikacijo. Ko je uporabnik overjen, IDP vrne SP-ju žeton, ki uporabniku omogoči dostop do zahtevane storitve. Nekateri izmed najbolj priljubljenih protokolov in tehnologij za SSO so SAML (Security Assertion Markup Language), OAuth 2.0 in OpenID Connect (OIDC).

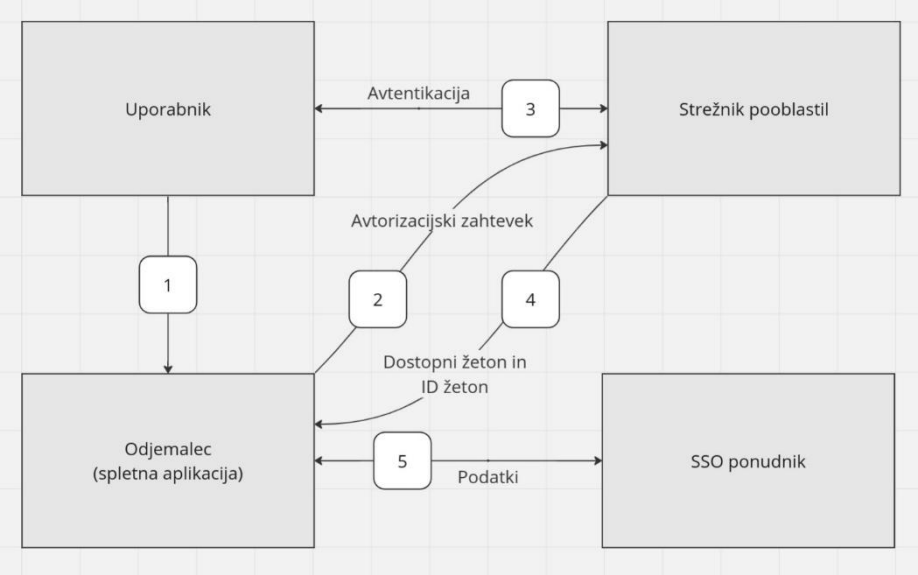
OAuth 2.0

OAuth 2.0 je široko razširjen avtorizacijski protokol, ki omogoča varen dostop do virov v imenu uporabnika, ne da bi ta delil svoje poverilnice [16]. Zasnovan je predvsem kot način za dostop do nabora virov, na primer oddaljenih API-jev ali uporabniških podatkov.

Protokol OAuth 2.0 definira štiri glavne vloge: lastnik vira (uporabnik), odjemalec (aplikacija, ki zahteva dostop), avtorizacijski strežnik pooblastil (izda dostopne žetone po overjanju uporabnika) in strežnik virov (hrani zaščitene vire). OAuth 2.0 vključuje pridobivanje dostopnega žetona od strežnika pooblastil, ki ga odjemalec nato predstavi strežniku virov za dostop do zaščitenih virov. Protokol je široko sprejet zaradi svoje prilagodljivosti, enostavne implementacije in široke podpore za različne scenarije, kot so mobilne aplikacije in spletne storitve.

OpenID Connect (OIDC)

OpenID Connect (OIDC) je avtentikacijski sloj, ki je zgrajen na osnovi protokola OAuth 2.0 [7]. Poleg zagotavljanja dostopa do virov, omogoča odjemalcem tudi preverjanje identitete uporabnika. Medtem ko se OAuth 2.0 osredotoča na dostop do virov, OIDC doda mehanizem za avtentikacijo z uporabo Json Web Token (JWT), ki zagotavlja identiteto uporabnikov. V protokolu OpenID Connect odjemalec zahteva avtentikacijo od strežnika pooblastil, ta pa overi uporabnika in vrne identifikacijski žeton (ID žeton) z informacijami o uporabniku. ID žeton je v formatu JWT, ki je digitalno podpisan in ga odjemalec lahko preveri ter pridobi identiteto uporabnika. OIDC zagotavlja standardiziran način izvajanja avtentikacije in se široko uporablja za vključevanje avtentikacije v aplikacije in storitve, hkrati pa izkorišča varnostne funkcije protokola OAuth 2.0. Celotna arhitektura OIDC je prikazana na sliki 1.

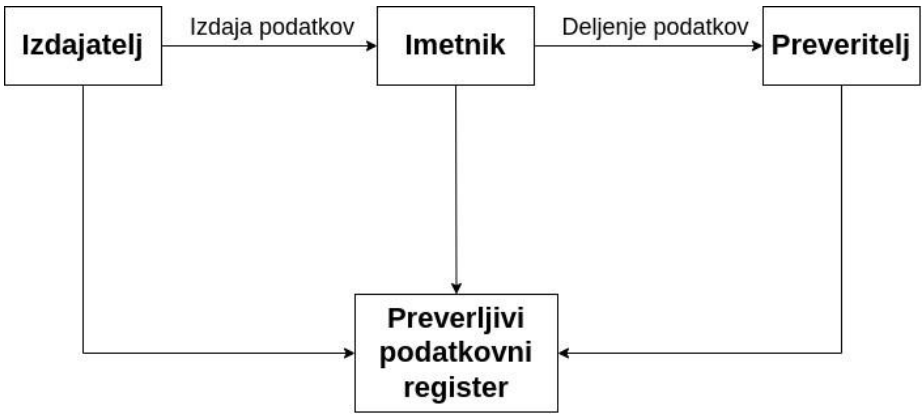


Slika 1: Prikaz arhitekture OpenID Connect.

2.2 Samo-upravljana identiteta (SSI)

Samo-upravljana identiteta (SSI) je eden izmed novih in naprednih načinov implementacije digitalne identitete in visoko-nivojsko preslikava stanje iz realnega fizičnega sveta [22]. Uporabnikom so izdani digitalni dokumenti, ki so nato shranjeni v njihovih digitalnih denarnicah. Svoje digitalnega dokumente lahko nato delijo s komerkoli želijo. SSI v ospredje postavlja lastništvo, zasebnost in popolno kontrolo nad digitalno identiteto. Vsako implementacijo digitalne identitete navadno sestavlja dve glavni komponenti in to so identifikatorji uporabnikov ter njihovi podatki. V primeru SSI sta to standarda DID in VC, pri čemer prvi definirana globalno unikatne identifikatorje uporabnikov in pravila upravljanja z njimi, medtem ko drugi defininra strukturo in lastnosti podatkovne strukture podatkov [2, 40]. Za razvoj skoraj vseh standardov s skupnosti SSI skrbita organizaciji W3C in Decentralized Identity Foundation (DIF) [41, 4].

SSI temelji na dveh glavnih procesih: izdajanje VC in njihovo deljenje. Ta dva procesa se izvjata med tremi tipi entitet in sicer izdajateljem (angl. issuer), imetnikom (angl. holder) in preveriteljem (angl. verifier) [22]. Celoten SSI cikel je prikazan na sliki 2. Na preprostem primeru visokošolske diplome bi bila v vlogi izdajatelja fakulteta oz. univerza, imetnik bi bil diplomant in preveritelj podjetje, kamor se bo po študiju zaposlil diplomant. Imetnik pridobi VC od izdajatelja in jo shrani v svojo digitalno denarnici, od koder jo nato lahko deli s preveritelji. Pri izdajanju in deljenju vse entitete verificirajo digitalne podpise podatkov in posledično tudi identifikatorje entitet, s katerimi komunicirajo. Verifikacija temelji na kriptografiji in sicer kriptografiji javnega ključa. Izdajatelj s svojim zasebnim ključem digitalno podpiše podatke, prav tako jih imetnik digitalno podpiše pri njihovem deljenju. Tako lahko vsakdo kriptografsko preveri, da so podatki bili izdani oz. deljeni s strani lastnika podatkov. Medtem ko je zasebni ključ entitet shranjen na varni lokaciji, na primer digitalni denarnici, mora biti javni ključ javno dostopen, saj preveritelj in izdajatelj ne komunicirata neposredno. V ta namen se javni ključ (z ostalimi potrebnimi podatki) registrirajo na preverljivi podatkovni register (angl. Verifiable Data Registry, VDR) [22]. Na VDR se zabeleži tudi stanje oz. veljavnost VC v obliki binarne vrednosti. VDR je pogosto implementiran kot decentralizirano omrežje, saj le v takšnem primeru SSI ostane povsem v decentralizirani obliki. Kadar uporabnik deli svoje podatke oz. dokumente s preveriteljem, generira preverljivo prezentacijo (angl. Verifiable Presentation, VP), ki je prav tako definirana v standardu VC [40]. Pri deljenju lahko uporabi tudi eno izmed naprednih kriptografskih metod, na primer selektivno razkritje ali ZKP [40].



Slika 2: Model zaupanja pri digitalni identiteti SSI.

Sistem in ekosistem za digitalno identiteto, ki temelji na principih SSI, razvija tudi Evropska komisija (angl. European Commission) v sodelovanju z Evropskim blockchain partnerstvom (angl. European Blockchain Partnership, EBP) v sklopu projektov European Blockchain Services Infrastructure (EBSI) in European Self-Sovereign Identity Framework (ESSIF) [6]. Evropska komisija je prav tako pred kratkim izdala navodila za razvoj digitalnih denarnic za decentalizirane digitalne identitete, pri čemer je vključenih veliko standardov iz skupnosti SSI [5]. Omenjena digitalna denarnica je namenjena prebivalcem Evropske unije (angl. European Union, EU), pri čemer je poudarek na storitvah v javni upravi.

Decentralizirani identifikatorji (DID)

DID so globalno unikatni identifikatorje uporabnikove oz. entitetine samoupravljane identitete, pri čemer so ti generirani naključno oz. iz kriptografskega ključa. Struktura identifikatorjev temelji na široko razširjenem spletnem standardu Uniform Resource Identifier (URI) [2]. Specifikacijo DID je prav tako pod okriljem organizacije W3C. Primer DID identifikatorja je did:ebsi:zxAFkP2dnB5oL2A5iaaK5K3, kjer:

- prvi del (did) striktno izraža, da je prikazan identifikator decentraliziran
- drugi del (ebsi) sporoča izbrano DID metodo (angl. DID method); tj. podvrsta oz. specifikacija za en tip decentraliziranih identifikatorjev, ki definira pravila njihovega upravljanja in pogosto tudi omrežje, ki se uporablja za omenjene DID-e
- tretji del (zxAFkP2dnB5oL2A5iaaK5K3) predstavlja vrednost, ki mora biti globalno unikatna znotraj izbrane DID metode

Prikazan identifikator spada pod DID metodo ebsi in se uporablja znotraj omenjenega SSI ekosistema, pri čemer so dodatne informacije o identifikatorju in entiteti za njim shranjene na EBSI blockchain omrežju [6]. Vsak DID se s pomočjo namenske storitve imenovane resolver transformira v DID dokument (angl. DID document), ki vsebuje vse dodatne informacije o DID-u [2]. To so vsi javni ključi identifikatorja, trenutni administrator oz. kontroler (angl. controller), URL pri izdajateljih in preveriteljih, itd. DID dokument je navadno v obliki JSON-LD in vse njegove možne lastnosti (angl. property) so definirane v DID metodi. Na tem mestu je potrebno omeniti, da vse DID metode ne podpirajo posodabljanja DID dokumentov - to sta na primer DID metodi did:key in did:pkh. Te DID metode se navadno uporabljajo za enkrat uporabo, preprostejše primere uporabe ali so tesno povezane z uporabnikovimi blockchain naslovi. Ostale DID metode, kot so na primer did:ebsi, did:polygonid in did:web, po drugi strani omogočajo posodabljanje DID dokumentov, kar omogoča napredne uporabe identifikatorjev, na primer rotacijo ključev ali kontrolerjev.

Preverljive poverilnice (VC in VP)

VC definira podatkovni format oz. kontejner, s katerim je mogoče predstaviti katerikoli digitalni dokument [40]. Tudi ta specifikacija je pod okriljem organizacije W3C. Medtem ko DID definirana unikatne identifikatorje uporabnikov, VC-ji predstavljajo podatke (najpogosteje podatke o njihovi identiteti). Ene izmed glavnih prednosti

uporabe preverljivih poverilnic so zmožnost njihove strojne berljivosti, enoten standard za vse tipe digitalnih dokumentov in kriptografska preverljivost izdajatelja ter imetnika dokumentov. Vsak VC navadno vsebuje informacije o izdajatelju, času izteka, tipu digitalnega dokumenta, informacije o imetniku, vrsto kriptografskega podpisa in ostale dodatne informacije specifične za določen tip digitalnega dokumenta [40].

Uporabniki svoje VC-je navadno hranijo v svojih digitalnih denarnicah. Priporočen in najpogostejši format za VC je JSON, pri čemer je ta najpogosteje v obliki JWT ali JSON-LD. Izbira formata VC je v večini odvisna od potreb in zahtev po uporabi določene vrste kriptografskega digitalnega podpisa. Vsak tip VC oz. digitalnega dokumenta mora slediti vnaprej definirani JSON Schemi, v kateri se zabeleži vse zahtevane lastnosti JSON-a, kot tudi katera polja so obvezna in katera opcijska. Verifikacija digitalnega podpisa poteka z resolucijo DID izdajatelja, kjer se iz DID dokumenta pridobi njegov javni ključ, ki se uporabi ob preverjanju digitalnega podpisa. Kadar imetnik VC-jev svoje dokumente deli s preveriteljem, zgenerira preverljivo prezentacijo (VP), v katero se lahko vključi en ali več digitalnih dokumentov. Z uporabo novih in naprednih kriptografskih pristopov, se lahko varnost in zasebnost uporabnika še dodatno izboljša, predvsem z uporabo selektivnega razkritja, kjer se razkrijejo le nujno potrebne vrednosti, in ZKPs, kjer se dokazuje prisotnost vrednosti in ne njihove dejansko vrednosti.

Sam standard VC ne definira enotnega načina, kako prenesti digitalne dokumente med različni entitetami v SSI ciklu, na primer izdajanje VC od izdajatelja do imetnika in deljenje od imetnika do preveritelja. V ta namen so bili razviti različni protokoli, pri čemer po uporabi prevladujeta DIDComm in OID4VC [3, 18]. Medtem ko je prvi nastal znotraj skupnosti decentralizirane identitete in podpira naprednejše funkcionalnosti DID-ov in VC-jev, se pogosteje uporablja in integrira v obstoječe sisteme OID4VC. Razlog za to je, da ta protokol temelji na obstoječem protokolu OIDC, ki je industrijski standard za sisteme za upravljanje digitalne identitete. Oba protokola sicer vsebujeta podobne določene korake, na primer dokaz o lastništvu privatnih ključev, ki kontrolirajo DID, in zahteve po pridobitvi določenih tipov oz. shem VC-jev in VP-jev.

Skozi leta razvoja sistemov za decentralizirano identiteto je bilo ustvarjenih več odprto-kodnih knjižnic, ki poenostavijo delo in integracijo različnih komponent SSI. Ene izmed najbolj priljubljenih knjižnic so Veramo, Hyperledger Aries, DID JWT VC, itd. Zanimivost in prednost standardov SSI je tudi, da se lahko uporabijo ločeno, kar pomeni, da se v določenih primerih uporabe lahko integrira le DID, v določenih pa le VC.

2.3 Digitalne denarnice

Digitalna denarnica predstavlja vstopno točko za imetnike za samo-upravljano identiteto oz. decentralizirani tehnologijami nasploh [21]. Denarnice so programska ali strojno oprema, s katero se uporabniki identificirajo in njihova jedrna naloga je upravljane s kriptografskimi ključi. Delitev navadno lahko poteka na dva načina in sicer glede na namembnost ali glede na tehnične specifikacije. Prvi prvem načinu prevladujejo kripto denarnice ali Web3 denarnice, ki se uporabljajo za podpisovanje transakcij in uporaba različnih blockchain omrežij (na primer MetaMask in Trust Wallet), in na denarnice za decentralizirano identiteto, na primer Lissi in Masca. Pri drugem načinu se denarnice grobo ločijo na programske denarnice, tj. vse denarnice omenjene do sedaj, in denarnice v obliki strojne opreme, na primer Ledger in Trezor. Kot je bilo že omenjeno, ne glede na namembnost in tehnično specifikacijo, je glavna naloga varno upravljano s kriptografskimi ključi, ki se lahko uporabijo za različne namene. Ključi se lahko iz denarnic varnostno kopirajo na različne načine, najpogosteje s semensko frazo (angl. seed phrase) ali s pomočjo novejših pristopov kot so Passkeys. Poleg upravljanja s kriptografskimi ključi še denarnice podpirajo več funkcionalnosti kot so generiranje DID in blockchain naslovov, generiranje in podpisovanje VP, podpisovanje in pošiljanje blockchain transakcij, komunikacija z blockchain omrežji in podpora različnim protokolom kot sta DIDComm in OID4VC [9].

3 OID4VC

OID4VC je nabor treh specifikacij ki definirajo standarden način za izdajanje, predstavljanje in deljenje VC in VP. Te tri specifikacije so:

- OpenID za izdajanje VC (angl. OpenID for Verifiable Credential Issuance, OID4VCI) [19], ki določa API in ustrezne avtorizacijske mehanizme, ki temeljijo na OAuth, za izdajanje VC. Podpira tako W3C VC podatkovni model, kot tudi mobilna vozniška dovoljenja (angl. mobile driving license, mDL) predstavljena v standardu ISO/IEC 18013-5:2021 [12].
- OpenID za predstavitev VP (angl. OpenID for Verifiable Presentations, OID4VP) [20] določa mehanizem zgrajen na OAuth 2.0, ki omogoča predstavitev trditev v obliki VP kot del protokola.
- Samoizdani OpenID ponudnik v2 (angl. Self-Issued OpenID Provider v2, SIOP) [38] omogoča uporabnikom, da sami delujejo kot svoj OpenID ponudnik in opravijo avtentikacijo s samoizdanimi ID žetoni, ki so podpisani s ključi pod njihovim nadzorom, in s tem neposredno predložijo samopotrjene trditve.

3.1 Primeri uporabe

OID4VCI se lahko uporabi za različne namene, na primer izdajo digitalnih potrdil oz. digitalnih dokumentov, ki vsebujejo informacije o identiteti ali kvalifikacijah posameznika. Primeri takšnih potrdil so osebne izkaznice, ki vsebuje informacije o starosti, in diplome oz. drugi certifikati o opravljenih tečajih. Uporabniki lahko nato pridobljene VC-je uporabijo kot dokazilo o opravljenem študij, usposobljenosti ali za drugačne namene.

OID4VP se medtem uporablja za varno predstavitev VC v obliki preverljivih predstavitev (VP) in omejevanje dostopa do raznih virov. Lahko se uporabi za omejitev dostopa do spletne strani, ki se omejene s starostjo ali izobrazbo uporabnika. V teh primerih uporabnik ustvari VP, ki vsebuje enega ali več veljavnih VC ter selektivno dokaže trditve oz. potrebne podatke in si tako pridobi dostop do želenega vira.

3.2 Potek

V prispevku se zaradi omejenega časa osredotočimo na sledeči dve specifikaciji: OID4VCI in OID4VP. Ti dve specifikaciji predstavljata tudi jedro OID4VC.

OID4VCI

Vsak izdajatelj VC za avtorizacijo dostopa uporablja OAuth 2.0 avtorizacijski strežnik, pri čemer se lahko isti strežnik uporabi s strani enega ali več izdajateljev. Izdajanje potrdil je moč izvesti na več različnih načinov, pri čemer se načini v grobem delijo na naslednje [19]:

- Avtoriziran kodni tok (angl. authorized code flow) ali pred-avtoriziran kodni tok (angl. pre-authorized code flow). Izdajatelj potrdila pridobi uporabniške podatke, ki jih pretvori v VC z uporabo uporabniške avtentikacije na OAuth 2.0 avtorizacijskem strežniku (avtoriziran kodni tok) ali z uporabo mehanizmov zunaj območja izdaje (pred-avtorizirani kodni tok).
- Sproženo s strani denarnice (angl. Wallet initiated) ali izdajatelja (agnl. Issuer initiated). Zahteva iz denarnice se lahko pošlje brez interakcije oz. akcije izdajatelja (sproženo s strani denarnice) ali kot odgovor po začetni komunikaciji z izdajateljem potrdil (sproženo s strani izdajatelja).
- Ista naprava (angl. same-device) ali več naprav (angl. cross-device). Spletna stran ali aplikacija izdajatelja lahko prebiva na isti napravi kot digitalna denarnica, ki prejema VC (ista naprava), ali na drugi napravi (več naprav).

- Pravočasno (angl. just-in-time) ali odloženo (angl. deferred). Izdajatelj potrdila lahko potrdilo izda neposredno v odgovor na zahtevo (pravočasno) ali pa z zakasnitvijo, pri čemer uporabnik s svojo digitalno denarnico prevzame kasneje (odloženo).

OID4VP

Načini pri postopku OID4VP imajo podobne lastnosti in se v grobem delijo v iste kategorije kot pri OID4VCI. Glavna razlika je, da v tem postopku imetnik pošilja podatke, medtem ko jih pri prejšnjem prejema. Glavna razširitev specifikacije OID4VP je vpeljava VP žetona [20]. Slednji vsebuje enega ali več VP in je v vlogi kontejnerja, ki omogoča končnim uporabnikom deljenje VP. Deljenje lahko podobno kot pri OID4VCI poteka na isti napravi, kot se nahaja denarnica, ki hrani VC, ali pa se nahaja na drugi napravi.

3.3 Arhitektura

Tako pri OID4VCI, kot tudi pri OID4VP poteka komunikacija s strežnikom preko https povezave. Obe specifikaciji uporabljata lastne formate URI naslovov za začetek tokov izmenjave poverilnic ali predstavitev. URI za ponudbo izdaje nove poverilnice se začne z predpono openid − credential − offer : //, URI za zahtevo pa avtorizaciji pa z predpono openid : //.

OID4VCI

V sklopu avtorizacijskega strežnika je potrebno implementirati sledeče vmesnike:

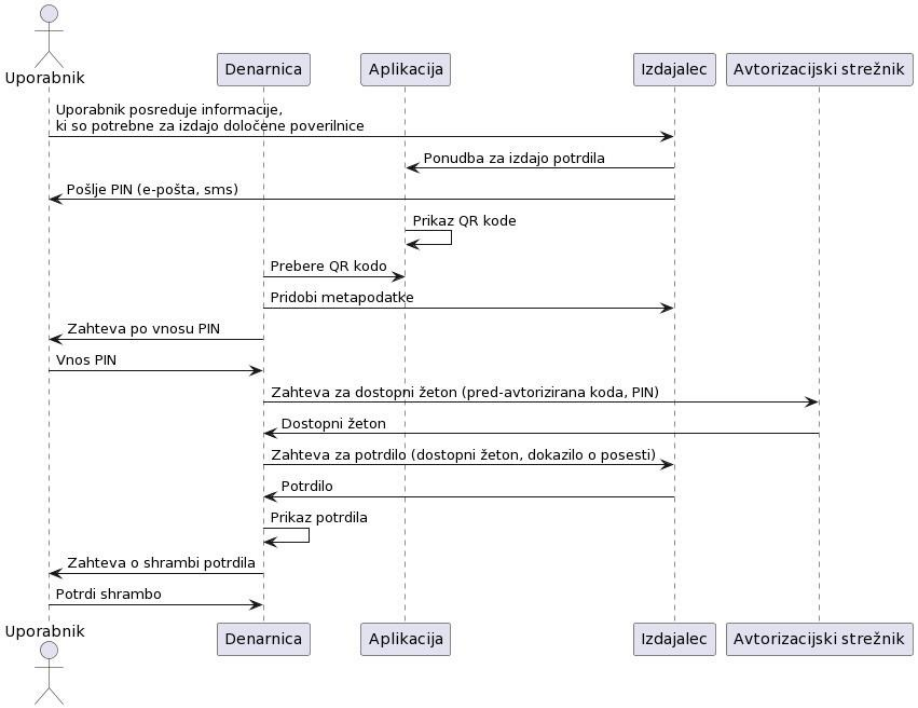
- Vmesnik za avtorizacijo (angl. authorization endpoint), ki se uporablja za avtorizacijo uporabnika pred izdajanjem VC.
- Vmesnik za pridobitev žetona (angl. token endpoint), ki se uporablja za pridobitev žetonov za dostop (angl. access token) in opcijsko tudi osvežitvenih žetonov (angl. refresh token).

V sklopu izdajatelja VC so implementirani sledeči vmesniki:

- Vmesnik za poverilnice (angl. credential endpoint), ki se uporablja za izdajanje posameznih VC v zameno za veljaven žeton. Opcijsko lahko ta vmesnik tudi veže izdan VC na kriptografski ključ uporabnikove identitete.

Opcijsko se lahko na strani izdajatelja VC implementirajo tudi sledeči vmesniki:

- Vmesnik za ponudbo potrdila (angl. credential offer endpoint), ki se uporablja za kreiranje ponudb uporabniku za izdajanje potrdila. Ponudba lahko vključuje informacije o VC, način zahtevane avtorizacije in URL izdajatelja, kjer denarnica lahko pridobi VC.
- Vmesnik za metapodatke (angl. metadata endpoint), ki se uporablja za objavljanje metapodatkov o sposobnostih izdajatelja potrdil. Ti lahko vključujejo informacije o vrstah VC, podprtih formatih VC, podprtih kriptografskih algoritmih in seznam vseh dostopnih vmesnikov.
- Vmesnik za množična potrdila (angl. batch credential endpoint), ki se uporablja za izdajanje večih potrdil v enem zahtevku. To je koristno za izdajo potrdil skupini uporabnikov ali za izdajo večih potrdil enemu uporabniku.
- Vmesnik za odloženo potrdilo (angl. deferred credential endpoint), ki se uporablja za izdajanje potrdil, ki niso takoj na voljo uporabnikom.



Slika 3: Potek izdaje VC (uporaba kombinacije pred-avtoriziranega kodnega toka, sproženega s strani izdajatelja, pri uporabi večih naprav in s pravočasnim izdajanjem).

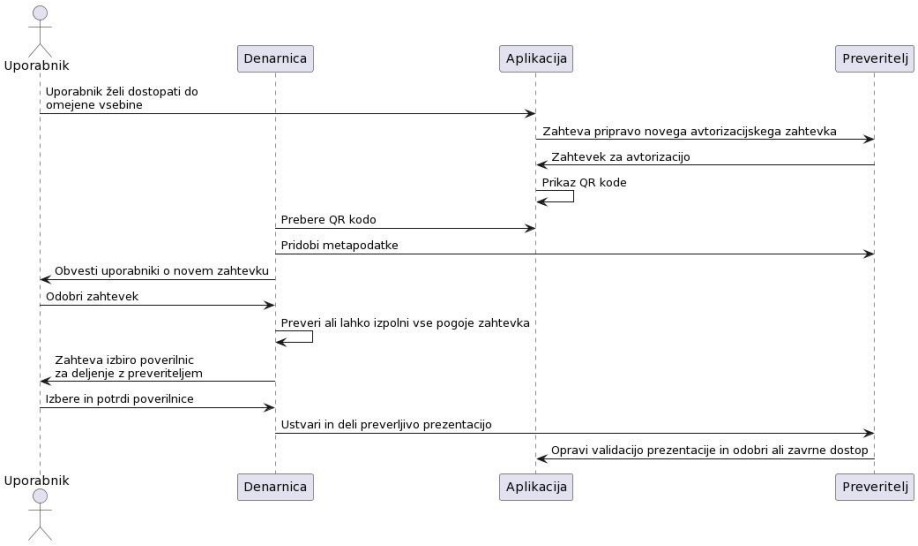
OID4VP

Na strani preveritelja je potrebno implementirati sledeče vmesnike:

- Vmesnik za avtorizacijski zahtevek (angl. authorization request endpoint), ki se uporablja za kreiranje zahtev za predstavitev VP od uporabnika.
- Vmesnik za avtorizacijski odgovor (angl. authorization response endpoint), ki se uporablja za prejemanje in validacijo uporabnikovih VP s pomočjo preusmeritev.

Opcijsko se lahko na strani preveritelja VP implementirajo tudi sledeči vmesniki:

- Vmesnik za metapodatke (angl. metadata endpoint), ki se uporablja za objavljanje metapodatkov o sposobnostih preveritelja VP. Ti lahko vključujejo informacije o podprtih formatih VP, podprtih kriptografskih algoritmih in seznam vseh dostopnih vmesnikov.
- Vmesnik za direkten avotorizacijski odgovor (angl. direct post response endpoint), ki se uporablja za prejemanje in validacijo uporabnikovih VP preko metode HTTP POST.



Slika 4: Potek deljenja VP pri uporabi večih naprav.

4 Uvajanje protokola OID4VC v digitalno denarnico Masca

4.1 Masca

Masca je MetaMask Snap [13] (vtičnik), ki razširi delovanje kripto denarnice MetaMask s funkcionalnostmi SSI oz. decentralizirane identitete [10]. Glavne funkcionalnosti zavzemajo upravljanje z decentraliziranimi identifikatorji, hrambo in upravljanje z VC, generiranje VP, deljenje VP preko protokola OID4VP, pridobivanje VC preko protokolov OID4VCI in Iden3comm [8] in avtentikacijo z ZKP preko protokola Iden3comm. Dodatna prednost kripto denarnice Masca je tudi preprosta integracija v (decentralizirane) aplikacije, saj je z uporabo knjižnice imenovane masca-connector možno integracijo izvesti le z nekaj vrsticami kode [11].

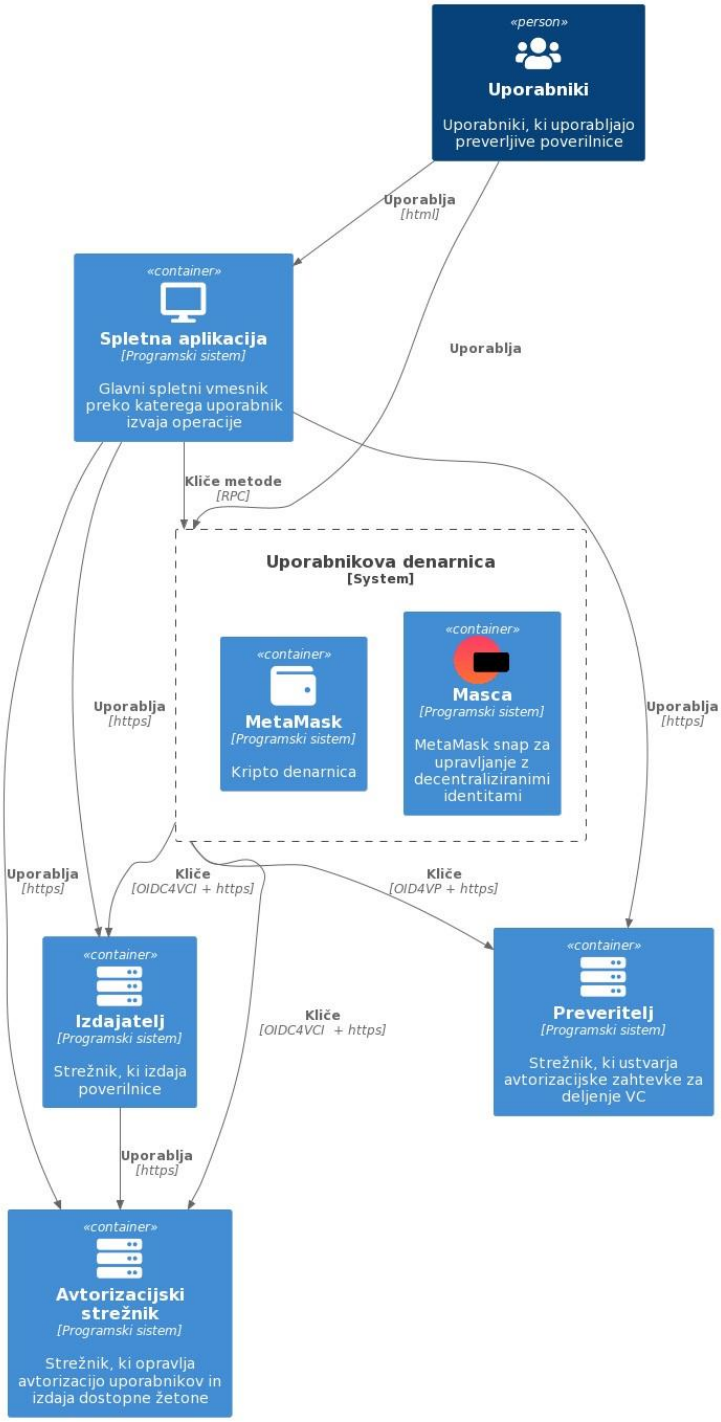
4.2 Veramo

Veramo je JavaScript knjižnica za SSI in na splošno preverljive podatke [39]. Njena zasnova temelji na fleksibilnost in modularnosti, kar omogoča enostavno integracijo z različnimi sistemi in v različnih okoljih. Knjižnica lahko tako teče na strežniku, v brskalniku ali na mobilnih napravah z uporabo okolja React Native. Uporaba knjižnice poteka tako, da razvijalec na začetku ustvari tako imenovanega agenta, kateremu se nato dodajo vtičniki z želenimi funkcionalnostmi, ki so nato dosegljive preko agentovega API. Nekatere izmed teh funkcionalnosti so:

- kreiranje in upravljanje s kriptografskimi ključi za namen digitalnega podpisovanja in šifriranja,
- kreiranje in upravljanje z DID,
- generiranje VC in VP,
- verificiranje VC in VP ter
- predstavljanje VP s selektivnim razkrivanjem.

4.3 Implementacija

Arhitektura implementacije je prikazana na sliki 5. Spletna aplikacija tukaj služi kot točka za pridobivanje ponudb za izdajo novih poverilnic, za pridobivanje zahtevkov za avtorizacijo in za avtentikacijo uporabnikov. Prav tako mora ta implementirati funkcionalnosti za klicanje RPC metod Masca vtičnika. Vtičnik od spletne aplikacije prejme pridobljene ponudbe za izdajo poverilnic in zahtevke za avtorizacijo, ter opravi vse nadaljnje korake, da se le te ti uspešno opravijo.



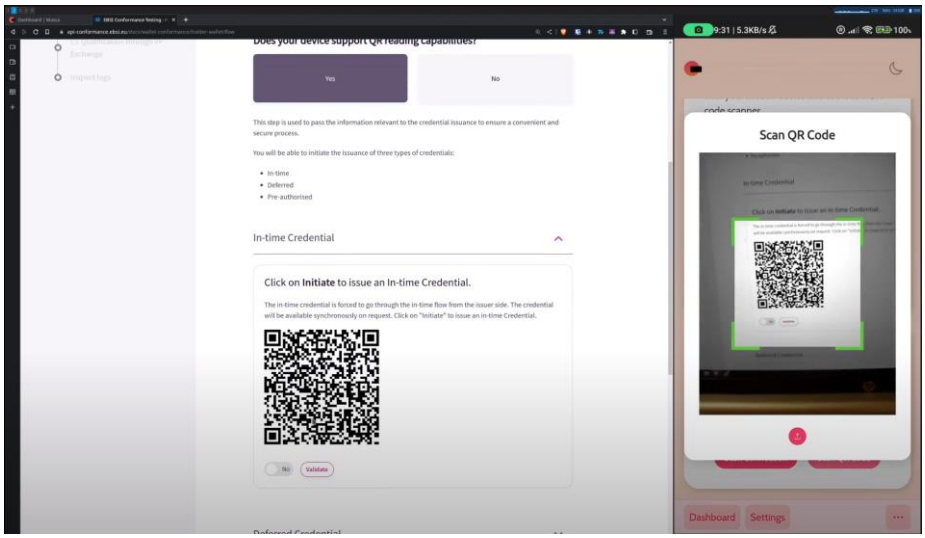
Slika 5: Arhitektura celotnega implementiranega sistema.

5 Validacija

5.1 Izmenjava poverilnic

Postopek izdaje nove poverilnice z uporabo pred-avtoriziranega kodnega toka se vedno prične pri izdajatelju. Slednji si je predhodno že pridobil potrebne informacije za izdajo potrdila uporabniku in v prvem koraku pripravi ponudbo za izdajo potrdila. Ta ponudba je v obliki URI naslova in se lahko prikaže uporabniku v obliki QR kode, katero mora skenirati na Masca aplikaciji, primer viden na sliki 6, ali pa se neposredno kliče RPC metoda Masca vtičnika. Hkrati izdajatelj pošlje uporabniku PIN kodo in sicer po drugem kanalu kot je poslal ponudbo za izdajo

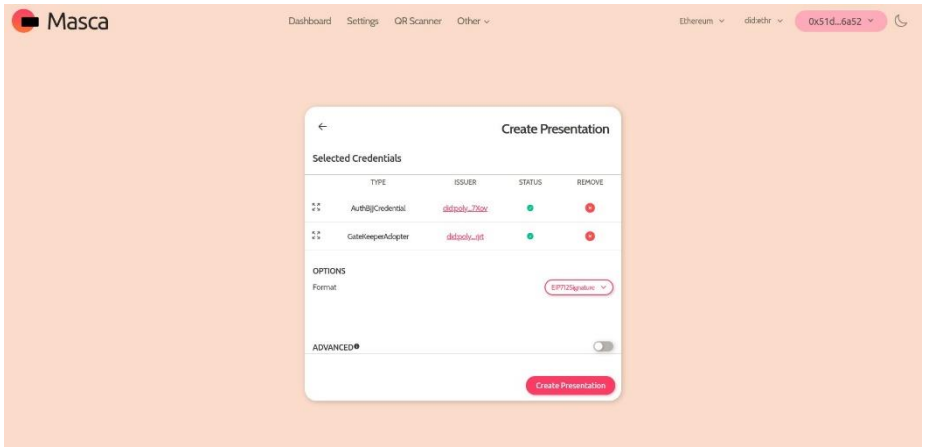
potrdila. PIN koda je potrebna za pridobitev dostopnega žetona. V naslednjem koraku denarnica pridobi metapodatke izdajatelja poverlinic, kjer je tudi zabeležen naslov avtorizacijskega vmesnika. Denarnica nato pripravi zahtevek za pridobitev dostopnega žetona, ki nato od uporabnika zahteva vnos PIN kode in vse skupaj se pošlje na avtorizacijski strežnik. Kot odgovor uporabnik pridobi dostopni žeton, kateri se v naslednjem koraku uporabi za pridobitev poverilnice s strani izdajatelja. Poverilnica se prikaže uporabniku, kateri jo nato potrdi in jo shrani v digitalno denarnico ali jo zavrne.



Slika 6: Skeniranje QR kode na strani EBSI WCT, ki vsebuje ponudbo za izdajo preverljive poverilnice.

5.2 Deljenje poverilnic

Postopek izmenjave preverljivih prezentacij (VP) se začne na strani preveritelja, ki pripravi zahtevek za avtorizacijo. Ta je prav tako kot v primeru izdajanja nove poverilnice v obliki URI naslova. Zahtev vsebuje vse potrebne informacije o poverilnicah, ki jih želi prejeti od uporabnika. Ponovno se lahko URI prikaže uporabniku v obliki QR kode, katero mora skenirati z Masca aplikacijo, ali pa se neposredno kliče RPC metoda Masca vtičnika. Vtičnik obvesti uporabnika o novem zahtevku za avtorizacijo in ko ta zahtevek odobri, preveri ali lahko zadosti vsem pogojem v njem (npr. deljenje specifičnega tipa poverilnice). Od uporabnika zahteva tudi, da izbere poverilnice, katere želi deliti s preveriteljem in nato ustvari preverljivo prezentacijo, ki se pošlje preveritelju. Na sliki 7 je prikazano deljenje poverilnic preko spletne aplikacije Masca.



Slika 7: Deljenje poverilnice na spletni aplikaciji Masca.

5.3 Varnostni pogledi

OID4VCI

Digitalne denarnice ne smejo pri uporabi vmesnika za ponudbo potrdil tem slepo zaupati, saj izvor ponudb ni avtoriziran in integriteta sporočil ni zavarovana. Zlonamerni akterji bi potencialno lahko uporabili te ponudbe za napad z lažnim predstavljanjem (angl. phishing attack) ali napad z vrivanjem (angl. injection attack) [19].

Prav tako se pojavi tudi vprašanje o zaupanju med denarnicami in izdajatelji VC. Izdajatelji pogosto želijo vedeti, kateri denarnici izdajajo VC in kako le ta upravlja s svojimi zasebnimi kriptografskimi ključi. Razlogi za to so sledeči:

- Izdajatelj želi zagotoviti, da so kriptografski ključi uporabnika pravilno zaščiteni pred uhajanjem (angl. exfiltration) in ponovnim predvajanjem (angl. replay). Tako je zlonamerni osebi preprečeno, da se pretvarja kot legitimen imetnik potrdila in predloži uporabnikovo potrdilo.
- Izdajatelj želi zagotoviti, da digitalna denarnica, ki upravlja potrdila, upošteva določene pravilnike in je bila po potrebi revidirana ter odobrena v skladu z določeno regulativno in/ali komercialno shemo.

Pred-avtoriziran kodni tok je ranljiv na ponovno predvajanje, saj ni vezan na specifično napravo, kot je to značilno pri avtoriziranem kodnem toku v kombinaciji s PKCE [30]. V ta namen je zato priporočljivo dodatno uporabiti naključno generirano uporabniško kodo, ki se uporabniku posreduje po drugem kanalu (npr. elektronski pošti ali SMS-u) kot ponudba za prevzem VC. Ta koda se nato zahteva v naslednjem koraku, ko denarnica želi pridobiti žeton za prevzem VC [19].

OID4VP

Pri deljenju VP je pomembno, da preveritelji implementirajo funkcionalnosti, ki preprečujejo napadalcem uporabo prestreženih VP žetonov, za namene impersonacije uporabnika. To se doseže z uporabo dveh parametrov. Vsak avtorizacijski zahtevek mora vsebovati naključno kriptografsko število, imenovano nonce, ki ima dovolj veliko entropijo in uporabnik jo mora vključiti v VP žeton. Prav tako mora VP žeton vsebovati parameter imenovan client id, ki predstavlja identifikator preveritelja, kateremu je žeton namenjen [20].

Pri uporabi deljenja VP, kjer se namesto preusmeritev uporablja metoda HTTP POST, lahko pride do napada fiksacije seje (angl. session fixation). V tem primeru napadalec začne postopek deljenja VP na svoji napravi in posreduje prejet avtorizacijski zahtevek pravemu uporabniku. Napadalec nato periodično poskuša zaključiti tok na svoji napravi in ko uporabnik zaključi le tega, dobi napadalec dostop do uporabnikove seje. Ta napad se lahko prepreči z uporabo preusmeritve, ki vključuje naključno kodo vezano na sejo, uporabnikove denarnice na čelni (angl. frontend) del aplikacije preveriteljevega strežnika [20].

5.4 Prednosti in slabosti

Prednosti

Ena večjih prednosti specifikacije OID4VCI je uporaba protokola OAuth 2.0, saj je ta široko razširjen, varen, preprost in fleskibilen. Prav tako omogoča, da se lahko obstoječe namestitve OAuth 2.0 in OIDC strežnikov razširijo s funkcionalnostmi izdajanja VC. Specifikacija s poenotenim in standardiziranim načinom izdajanja VC pripomore tudi k interoperabilnosti rešitev v tem ekosistemu.

Dobra lastnost specifikacij je tudi ta, da omogočajo razširitev implementacij z dodatnimi standardi. Na primer pri uporabi avtoriziranega kodnega toka se priporoča razširitev s standardom RFC7636 [30], ki omogoča preprečitev napada na prestrezanje avtorizacijskih kod, in standardom RFC9126 [35], ki zagotavlja celovitost in pristnost zahtev za avtorizacijo. Ostali standardi na katerih gradijo specifikacijo so sledeči: RFC6750 [23], RFC7515 [25], RFC7517 [27], RFC7519 [28], RFC7591 [29], RFC8152 [31] in RFC8725 [33].

Poglavitni doprinos specifikacij OID4VP je, da omogoča višjo zasebnost in varnost uporabnikov, uvede poenoten in standardiziran način deljenja in verifikacije VP, ter omogoča nadgradnjo že obstoječih rešitev. Standardi, na katerih specifikacija gradi oz. se lahko dodatno razširi so sledeči: RFC6819 [24], RFC7516 [26], RFC7591 [29], RFC8252 [17], RFC8414 [32], RFC9101 [34], RFC9126 [35] in RFC9207 [36].

Slabosti

Skupna slabost OID4VCI in OID4VP specifikacij je to, da sta še vedno v razvoju in se pogosto spreminjata. To povzroča nekompatibilnosti med različnimi implementacijami in nameščenimi rešitvami. Nobena od specifikacij prav tako ne omenja, kako se lahko v postopek izdajanja VC in preverjanja VP vključi uporaba ZKP.

6 Zaključek

V prispevku smo predstavili komponento novega in inovativnega sistema za upravljanje digitalne identitete, ki temelji na konceptih in principih SSI. Na začetku smo izpostavili potrebo bo varnih protokolih za izdajanje in deljenje preverljivih poverilnic (VC) ter predstavili prednosti uporabe protokolov OID4VC. V nadaljevanju smo predstavili specifikacijo in standarde OID4VC v podrobnosti, pri čemer smo se osredotočili na protokola OID4VCI in OID4VP.

Predstavljene koncepte smo validirali na lastni implementaciji izdajatelja in preveritelja v kombinaciji s kripto denarnico Masca, v katero smo implementirali podporo za protokole OID4VC. Med validacijo smo prišli do več zaključkov in povzeli prednosti ter slabosti omenjenega pristopa, pri čemer je ena izmed glavnih prednosti uporaba preverjenih standardov kot je OAuth 2.0 in velika zmožnost razširitve jedrnega postopka, medtem ko je ena izmed večjih slabosti hitro spreminjajoče specifikacije, saj še le te niso v dalj časa stabilni različici.

Literatura

[1] CLERCQ Jan De "Single sign-on architectures", International Conference on Infrastructure Security. Springer. 2002, str. 40–58.

[2] https://www.w3.org/TR/did-core/, Decentralized Identifiers (DIDs) v1.0, obiskano 7. 6. 2023.

[3] https://identity.foundation/didcomm-messaging/spec/, DIDComm Messaging Specification, obiskano 7. 6. 2023.

[4] https://identity.foundation/, DIF - Decentralized Identity Foundation, obiskano 7. 6. 2023.

[5] https://digital-strategy.ec.europa.eu/en/library/ european-digital-identity-architecture-and-reference-frameworkoutline, European Digital Identity Architecture and Reference Framework – Outline, obiskano 7. 6. 2023.

[6] https://ec.europa.eu/digital-buildingblocks/wikis/display/ebsi, Home - EBSI -, obiskano 7. 6. 2023.

[7] https://openid.net/developers/ how-connect-works/, How OpenID Connect works, obiskano 7. 6. 2023.

[8] https://0xpolygonid.github.io/tutorials/ wallet/wallet-sdk/polygonid-sdk/iden3comm/overview/, Iden3comm Overview, obiskano 7. 6. 2023.

[9] KERSIC Vid, et al. "Orchestrating Digital Wallets for On-and Off-chain Decentralized Identity Management", IEEE Access 2023.

[10] KERSIC VID et al. "Dodajanje poljubnih funkcionalnosti digitalni kripto denarnici MetaMask". In: Nasl. z nasl. strani. Univerza v Mariboru, Univerzitetna založba; Fakulteta za elektrotehniko, računalništvo in informatiko, 2022, pp. 141–154. url: https://dk.um.si/IzpisGradiva. php?id=82880.

[11] https://www.npmjs.com/package/@blockchain-lab-um/mascaconnector, Knjižnica za integracijo Masca (@blockchain-lab-um/masca-connector), obiskano 7. 6. 2023.

[12] https://www.iso.org/standard/ 69084.html, mDL - ISO/IEC 18013-5:2021, obiskano 7. 6. 2023.

[13] https://metamask.io/snaps/, MetaMask Snaps, obiskano 7. 6. 2023.

[14] MURRAY Alex et al. "The promise of a decentralized internet: What is Web3 and how can firms prepare?", Business Horizons 66.2 (2023), str. 191–202.

[15] https://bitcoin.org/bitcoin, Satoshi Nakamoto. "Bitcoin whitepaper", obiskano 7. 6. 2023.

[16] https://oauth.net/2/, OAuth 2.0 - OAuth, obiskano 7. 6. 2023.

[17] https://www.rfc-editor.org/info/ rfc8252, OAuth 2.0 for Native Apps, obiskano 7. 6. 2023.

[18] https://openid.net/openid4vc/, OpenID for Verifiable Credentials - OpenID Foundation, obiskano 7. 6. 2023.

[19] https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html, OpenID for Verifiable Credentials Issuance, obiskano 7. 6. 2023.

[20] https://openid.net/specs/openid-4-verifiable-presentations-1_0.html, OpenID for Verifiable Presentations, obiskano 7. 6. 2023.

[21] PODGORELEC Blaz, et al. "What is a (digital) identity wallet? a systematic literature review", 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC). IEEE. 2022, str. 809–818.

[22] PREUKSCHAT Alex, REED, Drummond. "Self-sovereign identity", Manning Publications, 2021.

[23] https://www.rfc-editor.org/info/rfc6750, RFC 6750: The OAuth 2.0 Authorization Framework: Bearer Token Usage, obiskano 7. 6. 2023.

[24] https://www.rfc-editor.org/info/rfc6819, RFC 6819: OAuth 2.0 Threat Model and Security Considerations, obiskano 7. 6. 2023.

[25] https://www.rfceditor.org/info/rfc7515, RFC 7515: JSON Web Signature (JWS), obiskano 7. 6. 2023.

[26] https://www.rfceditor.org/info/rfc7516, FC 7516: JSON Web Encryption (JWE, obiskano 7. 6. 2023.

[27] https://www.rfc-editor. org/info/rfc7517, RFC 7517: JSON Web Key (JWK), obiskano 7. 6. 2023.

[28] https://www.rfc-editor. org/info/rfc7519, RFC 7519: JSON Web Token (JWT), obiskano 7. 6. 2023.

[29] https://www.rfc-editor.org/info/rfc7591, RFC 7591: OAuth 2.0 Dynamic Client Registration Protocol, obiskano 7. 6. 2023.

[30] https://www.rfc-editor.org/info/rfc7636, RFC 7636: Proof Key for Code Exchange by OAuth Public Clients, obiskano 7. 6. 2023.

[31] https://www.rfc-editor.org/info/rfc8152, RFC 8152: CBOR Object Signing and Encryption (COSE), obiskano 7. 6. 2023.

[32] https:// www.rfc-editor.org/info/rfc8414, RFC 8414: OAuth 2.0 Authorization Server Metadata, obiskano 7. 6. 2023.

[33] https://www.rfc-editor.org/info/rfc8725, RFC 8725: JSON Web Token Best Current Practices, obiskano 7. 6. 2023.

[34] https://www.rfc-editor.org/info/rfc9101, RFC 9101: The OAuth 2.0 Authorization Framework: JWT-Secured Authorization Request (JAR), obiskano 7. 6. 2023.

[35] https://www.rfc-editor.org/info/rfc9126, RFC 9126: OAuth 2.0 Pushed Authorization Requests, obiskano 7. 6. 2023.

[36] https://www.rfc-editor.org/info/rfc9207, RFC 9207: OAuth 2.0 Authorization Server Issuer Identification, obiskano 7. 6. 2023.

[37] SALOMAA Arto, "Public-key cryptography". 1996.

[38] https://openid.net/specs/openid-connect-self-issued-v2-1_0.html, Self-Issued OpenID Provider v2, obiskano 7. 6. 2023.

[39] https://veramo.io, Veramo - A JavaScript Framework, obiskano 7. 6. 2023.

[40] https://www.w3.org/TR/vc-data-model/, Verifiable Credentials Data Model v1.1, obiskano 7. 6. 2023.

[41] https://www.w3.org/, World Wide Web Consortium (W3C), obiskano 7. 6. 2023.