Dodajanje poljubnih funkcionalnosti digitalni kripto denarnici MetaMask

Vid Keršič, Andraž Vrečko, Urban Vidovič, Martin Domajnko, Muhamed Turkanović

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor, Slovenija
vid.kersic@um.si, andraz.vrecko@student.um.si, urban.vidovic2@um.si,
martin.domajnko@student.um.si, muhamed.turkanovic@um.si

Sinopsis Kripto denarnice omogočajo interakcijo z verigami blokov (angl. blockchain) in decentraliziranimi aplikacijami (angl. decentralized application, dApp), ter posledično upravljanje z zamenljivi ali nezamenljivimi žetoni (angl. NFT), interakcijo z navideznimi svetovi (angl. metaverse) itn. Danes je vsekakor najpriljubljenejša denarnica MetaMask. Ker se funkcionalnosti in možnosti uporabe tehnologije veriženja blokov zelo hitro povečujejo, so razvijalci denarnice MetaMask leta 2022 izdali razvojni sistem imenovan Snaps. Ta omogoča razvoj in integracijo poljubnih funkcionalnosti v obliki vtičnikov za kripto denarnico MetaMask.

V prispevku bomo predstavili kako lahko na osnovi sistema Snaps snujemo poljubno funkcionalnost h kripto denarnici MetaMask ter le to temu primerno razvijemo in integriramo. Za namen demonstracije bomo predstavili implementacijo razširitev, ki MetaMask denarnici doda podporo decentraliziranim identitetam in identifikatorjem (angl. decentralized identifier, DID) ter preverljivim poverilnicam (angl. verifiable credential, VC), ki so do sedaj za uporabo potrebovali ločeno rešitev.

Ključne besede:
kripto denarnice
digitalna identiteta
tehnologija veriženja blokov
MetaMask
samo-upravljana identiteta

Univerzitetna založba Univerze v Mariboru
ISBN 978-961-286-639-6
DOI https://doi.org/10.18690/um.feri.10.2022.12

1 Uvod

Tehnologija veriženja blokov (angl. blockchain) se je v zadnjih letih razširila na razna področja, na primer finančne trge, dobavne verige, zabavno industrijo, igričarsko industrijo in virtualne svetove [29]. Razlog za njeno uspešno vpeljavo je bila odprtost, nevtralnost, decentralizacija omrežij brez enega samega lastnika in interoperabilnost tehnologije s številnimi standardi, ki so omogočili razvoj kripto valut, decentraliziranih digitalnih identitet, nezamenljivih žetonov (angl. non-fungible token, NFT) kot sredstev za digitalno lastništvo itd. Kljub številnim prednostim je lahko uporaba tehnologije novim uporabnikom zelo zahtevna, saj je v primerjavi s sorodnimi tehnologijami bistveno kompleksnejša in orodja so ljudem na prvi pogled zapletena, podobno kot so bili spletni brskalniki v 90. letih. Za enostavnejšo uporabo tehnologije veriženja blokov so se razvile digitalne kripto denarnice, ki uporabnikom omogočajo interakcijo z verigami blokov in decentraliziranimi aplikacijami (angl. decentralized application, dApp) ter posledično upravljanje z vsemi produkti na verigi blokov, na primer kripto valutami in NFT-ji. Toda hitri razvoj funkcionalnosti in tudi novih verig blokov je povzročil veliko število digitalnih kripto denarnic, kjer vsaka pokriva le določene funkcionalnosti.

Danes prevladujoča kripto denarnica je MetaMask, ki podpira vsa omrežja, ki temeljijo na navideznem stroju Ethereum (angl. Ethereum Virtual Machine, EVM) [17, 3]. MetaMask omogoča upravljanje s kriptografskimi ključi, ki so ena izmed najpomembnejših komponent tehnologije veriženja blokov, pošiljanje kripto valut in NFT-jev, konstrukcijo in pošiljanje transakcij v omrežje, pregled stanja kripto valut in NFT-jev, zamenjavo kripto valut itd. Kljub mnogim funkcionalnostim se MetaMask ne more uporabiti za čisto vse operacije, ki se pogosto dogajajo v svetu verige blokov, na primer ne podpira uporabe omrežja Bitcoin, ali shranjevanja datotek na distribuirano omrežje IPFS. Ker razvijalci ne moremo sami podpreti vseh možnih funkcionalnosti in slediti vsem standardom, ki nastajajo zelo hitro, so leta 2022 izdali razvojni sistem Snaps. Ta omogoča razvoj in integracijo poljubnih funkcionalnosti v obliki vtičnikov za kripto denarnico MetaMask, pri čemer se osnovne funkcionalnosti lahko nadgradijo s poljubnimi, hkrati pa se ohrani varnost operacij, saj kriptografski ključi nikoli ne zapustijo varne hrambe. Tako lahko uporabniki uporabljajo znano rešitev za več namenov in ne potrebujejo novih orodij oz. aplikacij za vsako dodatno funkcionalnost.

Evropska unija razvija specifikacijo za digitalne identitete, ki je usmerjena v končne uporabnike, kjer imajo ti lastništvo nad svojimi podatki, so jim cel čas na voljo, sledi načelom GDPR in omogoča možnosti identifikacije z internetno povezavo ali brez. Eden izmed možnih načinov implementacije takšne digitalne identitete je s samoupravljano identiteto (angl. self-sovereign identity, SSI), ki temelji na podobni kriptografiji kot tehnologija veriženja blokov [21]. Kljub temu danes na trgu izključno prevladujejo kripto denarnice, ki so namenjene le SSI, kar zahteva od uporabnikov še eno dodatno orodje za digitalno življenje. Zato se poraja vprašanje: ali se lahko upravljanje z digitalno identiteto temelječo na SSI integrira v že obstoječo kripto denarnico in razvije enoten produkt za verige blokov ter digitalne identitete?

V prispevku bomo predstavili, kako lahko na osnovi sistema Snaps snujemo poljubno funkcionalnost h kripto denarnici MetaMask, jo implementiramo in integriramo v MetaMask ter posledično v dApp. Čeprav bomo pokazali, da nam sistemi Snaps omogočajo praktično neomejeno možnosti, bomo v prispevku, kot že omenjeno, za namen demonstracije predstavili implementacijo razširitve za podporo digitalni identiteti temelječi na principih in komponentah SSI. Prispevek zaključimo z analizo prednosti in slabosti, ki jih ima predstavljen pristop, in pregledom trenutnega stanja novih funkcionalnosti kripto denarnice MetaMask.

2 Osnovni koncepti

2.1 Tehnologija veriženja blokov

Tehnologija veriženja blokov predstavlja porazdeljene digitalne knjige transakcij (angl. distributed ledgers, DLT), ki nimajo centralne avtoritete in so odporne na poneverbo podatkov (angl. tamper-proof). Leta 2008 je bila ideja

veriženja blokov prvič združena z nekaterimi drugimi tehnologijami in računalniškimi koncepti, ko je Satoshi Nakamoto ustvaril prvo kriptovaluto imenovano Bitcoin [30], ki še danes ostaja na prvem mestu po tržnem kapitalu (angl. market capital). Druga največja po tržnem kapitalu[1] in največja, ko govorimo o ekosistemu javnih tehnologij omrežij blokov, je veriga blokov imenovana Ethereum[2], ki je bila vzpostavljena leta 2015 in predstavila podporo pametnim pogodbam. Omrežje verige blokov v primeru javnega tipa v grobem sestoji iz vozlišč (angl. nodes) porazdeljenih povsod po svetu, pridruži pa se jim lahko teoretično kdorkoli, kar v idealnem primeru pomeni decentraliziran sistem pri katerem zaupanje med uporabniki in vozlišči ni potrebno. Celotno upravljanje je v primeru javnih tipov verig blokov prav tako popolnoma decentralizirano oz. odvisno od 51 % vozlišč. Poznamo pa tudi t. i. zaprte oz. zasebne verige blokov, kjer pa so vozlišča vnaprej definirana s strani centraliziranih upraviteljev, med tem ko je sam dostop do verige blokov lahko javen ali zaprt [31].

Vsako vozlišče ima praviloma pri sebi shranjeno celotno verigo blokov, ki se posodablja in sinhronizira z vsakim novim dodanim blokom. V blokih verige oz. natančneje logičnih skupkih transakcij so zabeležene nespremenljive transakcije med uporabniki, pametne pogodbe in klici funkcij pametnih pogodb, ki spremenijo stanje računov ali prožijo dogodek (angl. event). Vsaka transakcija se pri vključitvi v blok overi, blok pa se pred dodajanjem v verigo potrdi z določenmim algoritmom konsenza.

Bloki se med seboj povezujejo s pomočjo zgoščenih vrednosti (angl. hash) na način, da ima vsak naslednji blok shranjeno zgoščeno vrednost prejšnjega bloka. V primeru, da napadalci poskušajo v verigo blokov vključiti nov zlonameren blok, ga vozlišča zavrnejo, saj se zgoščene vrednosti več ne ujemajo. Poznamo več tipov konsenza, kot so na primer dokazilo o delu (angl. proof of work, PoW), dokazilo o deležu (angl. proof of stake, PoS) in dokazilo o avtoriteti (angl. proof of authority, PoA) [32, 33]. Na začetku je bila večina omrežij blokov zavarovana s PoW, vendar v zadnjem času prevladuje PoS v javnih oz. PoA zasebnih omrežjih, saj je varnost PoW zagotovljena z rudarjenjem blokov, ki je zelo potraten in računsko zahteven računalniški algoritem.

Kot rečeno, je Ethereum kot nadgradnja Bitcoina uvedel pametne pogodbe (angl. smart contracts). To so deli programske kode, ki so zapisani v bloku, koda pa se izvaja ob klicih funkcij implementiranih v pametnih pogodbah [34]. Pri tradicionalnih pogodbah pogodbeniki potrebujejo zaupanja vredno avtoriteto, ki potrdi pogoje sklenjene pogodbe, medtem, ko se pri pametnih pogodbah izognemo potrebi po omenjeni tretji avtoriteti, saj je koda pogodbe nespremenljiva in vidna vsem, izvrši pa se le, ko so izpolnjeni vsi pogoji.

Vsak uporabnik ima svoj unikaten naslov, ki se enosmerno deterministično izpelje iz privatnega ključa, kar pomeni, da naslov iz privatnega ključa lahko izpeljemo, privatni ključ iz naslova pa ne. Naslovi na nek način predstavljajo identiteto uporabnika ali pametne pogodbe in so vidni v vsakem dogodku na omrežju. Tako lahko točno vidimo, kdo je komu kaj poslal, katero funkcijo pametne pogodbe je izvršil itd.

Na omrežju Ethereum že obstaja veliko t.i. decentraliziranih aplikacij [35]. Te aplikacije se pišejo po določenih standardih z željo, da so interoperabilne med vsemi platformami. Ene izmed najpopularnejših aplikacij so nezamenljivi žetoni. V grobem so to "žetoni", kjer je vsak unikaten in jih ne gre deliti na več delov. V verigi blokov je zapisano, kdo je trenutni lastnik katerega žetona, lastniki jih lahko pošljejo drugim uporabnikom in z njimi trgujejo, kar se prav tako zabeleži v verigo blokov. NFT-ji ponavadi referencirajo določeno medijsko predstavnost - sliko, zvok ali video. Pri tem ima vsak posamezen žeton svojo identifikacijsko številko, kar ga naredi unikatnega in nezamenljivega. Uporabljajo se tudi za druge namene, kot osnova za platforme, ki zagotavljajo globalno zaposljivost ali upravljanje z digitalnimi certifikati [36]. Najpopularnejša standarda, s katerima lahko implementiramo NFT-je sta ERC721 in ERC1155. Na omrežju Ethereum lahko sicer implementiramo tudi žetone po standardu ERC20. Za razliko od NFT-jev so to žetoni, ki so zamenljivi, kar pomeni, da so vsi žetoni med seboj popolnoma enaki. Primer aplikacije ERC20 žetonov je odklepanje novih funckionalnosti na določeni platformi (npr. platforma brezplačno ponuja poslušanje glasbe v nizki kakovosti, vsi uporabniki, ki si lastijo N določenih žetonov pa lahko na platformi glasbo poslušajo v višji kakovosti). Vse pogosteje slišan izraz, ki se prav tako povezuje s tehnologijo veriženja blokov, je Metaverse, ki opisuje splet kot združen digitalni svet s katerim smo v

[1] https://coinmarketcap.com/
[2] https://defillama.com/

stiku s pomočjo virtualne (angl. virtual reality, VR) in obogatene resničnosti (angl. augmented reality, AR). Vsak uporabnik upravlja svojega avatarja, ki je del Metaverse-a in raziskuje virtualni svet. ERC20 žetoni in NFT-ji predstavljajo dobrine v Metaverse-u, ERC20 npr. žetoni oz. valuta za kupovanje vozil, oblačil ipd., NFT-ji - npr. dejanska oblačila, ki jih lahko oblečemo našemu avatarju, vozilo, ki ga s svojim avatarjem vozimo. Tehnologija veriženja blokov se danes vključuje v tehnologije nove generacije svetovnega spleta, imenovane web3 [28].

2.2 Kriptografija

Kriptografija igra pomembno vlogo pri tehnologiji veriženja blokov, saj je poglavitna komponenta, ki zagotavlja, da veriga blokov ostane varna in da ponuja določeno stopnjo zasebnosti. Uporablja se pri šifriranju podatkov oz. za digitalno podpisovanje sleherne transakcije. V grobem zagotavlja, da se transakcije, ki jih uporabniki prožijo na osnovi t. i. blockchain računov, lahko verificirajo, kar je ključnega pomena za platformo, ki omogoča upravljanje s t. i. digitalnimi vrednostmi. Pri veriženju blokov se uporablja predvsem asimetrična kriptografija, ki temelji na šifriranju s parom ključev (javni in zasebni). Vsak akter ima svoj par javnega in zasebnega ključa, ki se generira s pomočjo različnih algoritmov, pri čemer je treba poskrbeti, da zasebni ključ ves čas ostane v rokah lastnika, saj lahko v nasprotnem primeru napadalec lastniku odtuji njegovo digitalno vrednost. V veliki večini platform verig blokov se tako uporablja eden izmed algoritmov eliptičnih krivulij (angl. elliptic curve digital signature algorithm, ECDSA). Javni ključ in s tem povezan tudi t. i. blockchain naslov je izpeljan iz zasebnega ključa enosmerno, kar pomeni, da iz zasebnega ključa izpeljemo javni ključ, iz javnega ključa pa naslov, v nasprotno smer pa izpeljevanje ni možno. Pri tehnologiji veriženja blokov je vsaka transakcija zabeležena na porazdeljeni knjigi in digitalno podpisana s privatnim ključem. Tako lahko z imetjem na določenem naslovu razpolaga le lastnik zasebnega ključa, medtem ko lahko veljavnost digitalnega podpisa preverijo vsi uporabniki omrežja, saj validacija poteka z javnim ključem, ki se lahko pridobi iz digitalno podpisane transakcije. Asimetrična kriptografija tako predstavlja višjo raven varnosti, saj si je s prejemnikom potrebno izmenjati le javni del kriptografskega para. Prav ta princip digitalnih podpisov se uporablja tudi pri upravljanju preverljivih poverilnic, šifriranje podatkov pa se v našem primeru uporablja za varno shranjevanje decentraliziranih identitet [1].

2.3 Kripto denarnice

Ena ključnih komponent za interakcijo s tehnologijo veriženja blokov so kripto denarnice. Primarna funkcionalnost kripto denarnic je varna hramba kriptografskih ključev , v večini primerov pa omogočajo tudi uporabo le teh za namene podpisovanja transakcij in v nekaterih primerih tudi šifriranje podatkov. Glavna razlika v primerjavi z digitalnimi denarnicami je ta, da kripto denarnice ne hranijo dejanskih sredstev. Ta so namreč shranjena na verigi blokov in do njih dostopamo s pomočjo naših zasebnih ključev s katerimi dokažemo lastništvo. Prav tako se kripto denarnice razlikujejo od trenutnih sistemov, npr. bančnih, pri katerih z uporabniškim imenom in geslom pridobimo dostop do centralizirane platforme, na kateri lahko izvedemo različne operacije. V kripto denarnicah se vse operacije z našimi ključi podpišejo digitalno lokalno, nato pa se pošljejo v omrežje verige blokov, kjer jih izvedejo vozlišča omrežja. Ta razlika krepko pripomore k varnosti teh sistemov, hkrati pa je verjetnost, da dve denarnici ustvarita enake ključe med 2^{128} in 2^{256}, kar dodatno podkrepi omenjeno trditev. Kripto denarnice lahko omogočajo tudi generiranje večih parov ključev, med katerimi ni povezave, in med katerimi lahko preprosto preklapljamo, kot med računi v aplikaciji.

Kripto denarnice generirajo zasebne ključe deterministično glede na naključno semensko frazo (angl. seed phrase). Ta je sestavljena iz 12 do 24 naključnih besed in služi tudi za obnovitev denarnice v primeru izgub. Ideja uporabe semenskih fraz za deterministično generiranje zasebnih ključev izhaja iz motivacije, da so besede preprostejše za človeško uporabo kot števila v binarni ali šestnajstiški obliki, in je bila predstavljena v 39. predlogu za izboljšanje Bitcoina (angl. Bitcoin improvement proposal, BIP) [20].

Kripto denarnice delimo na več načinov. Najpogostejša delitev razdeli kripto denarnice v tri kategorije, tj., (1) kripto denarnice v obliki programske opreme, (2) kripto denarnice v obliki strojne opreme in (3) kripto denarnice v papirni obliki. Slednja (angl. paper wallet) predstavlja list papirja, ki hrani natisnjene zasebne in javne ključe, ponavadi v obliki QR kod. Ta oblika denarnice sodi tudi pod kategorijo hladne hrambe (angl. cold storage), kar pomeni, da so ključi hranjeni na mediju, ki ni povezan z internetom. Kripto denarnice v obliki strojne opreme prav tako spadajo v kategorijo hladne hrambe, saj so kriptografski ključi hranjeni na ločenih fizičnih napravah, ki so ponavadi v obliki podobni USB ključem z dodatnimi mikrokrmilniki. Ta metoda hrambe se danes uveljavlja kot najvarnejši način hrambe kriptografskih ključev. Dve najbolj znani napravi iz te kategorije sta Ledger [18] in Trezor [23]. V kategorijo kripto denarnic v obliki programske opreme sodijo (1) mobilne kripto denarnice, (2) kripto denarnice kot aplikacije naložene na računalniku in (3) spletne kripto denarnice. Glavna razlika spletnih kripto denarnic od ostalih dveh kategorij je ta, da se kriptografski ključi ne hranijo pri uporabniku, ampak so pod okriljem tretje osebe, ponavadi so to kripto menjalnice (angl. crypto exchange), ki upravljajo s temi ključi v imenu uporabnika. Med najbolj znane kripto menjalnice, ki ponujajo spletne denarnice, sodijo Binance, FTX in Coinbase.

Kripto denarnice se razlikujejo tudi v tem, katera omrežja podpirajo. Nekatere kripto denarnice podpirajo le eno vrsto omrežja, primer sta kripto denarnici Electrum, ki podpira zgolj omrežje Bitcoin, in MetaMask, ki podpira le omrežja, ki temeljijo na navideznem stroju Ethereum. Kripto denarnice, kot so Trust Wallet, Coinbase Wallet in ZenGo, ter večina spletnih kripto denarnic, pa hkrati podpirajo hrambo kriptografskih ključev za interakcijo z različnimi vrstami omrežij.

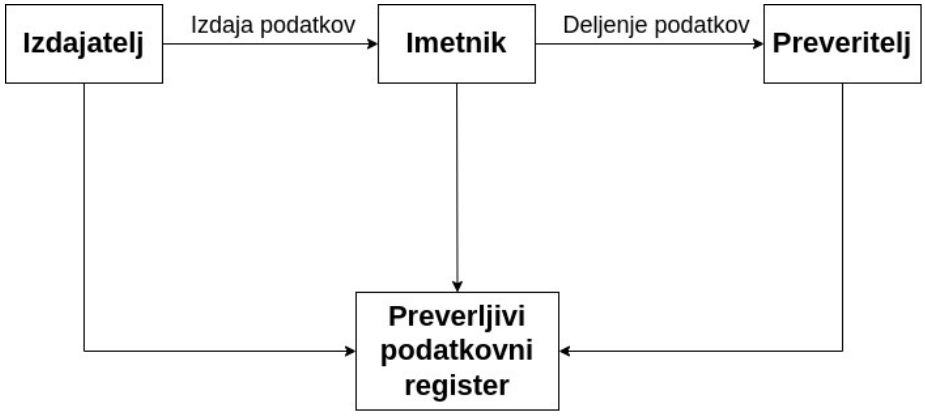
2.3.1 MetaMask

Kripto denarnica MetaMask sodi v kategorijo kripto denarnic v obliki programske opreme. Namesti se lahko kot razširitev v brskalniku ali kot samostojna mobilna aplikacija. Uporabniku zraven osnovnih funkcionalnosti za hrambo in upravljanje s ključi omogoča tudi povezavo s kripto denarnicami v obliki strojne opreme, nakup podprtih kripto valut s fiat denarjem (angl. fiat money), zamenjavo kripto valut, interakcijo s pametnimi pogodbami in digitalno podpisovanje. Nakup kripto valut je omogočen s pomočjo zunanjih storitev Transak, MoonPay in Wyre. Pri zamenjavi kripto valut pa kripto denarnica preišče izbrane decentralizirane menjalnice (angl. decentralized exchange, DEX) in izbere najboljše pretvorbeno razmerje z najnižjimi stroški menjave. Samo interakcijo s pametnimi pogodbami podpira na osnovi podpisovanja transakcij, hkrati pa s tem omogoča tudi funkcionalnost za prikaz stanj ERC20 žetonov v sami denarnici. Implementacija digitalnega podpisovanja temelji na standardih EIP712 [9] in EIP191 [7]. Mobilna različica denarnice podpira tudi protokol WalletConnect [26] in ponuja vgrajen brskalnik, s katerim lahko dostopamo do decentraliziranih aplikacij. MetaMask podpira tudi že nov standard EIP4361 [8], ki definira uporabo Ethereum računov za overjanje in vzpostavljanje sej brez uporabe klasičnih centraliziranih ponudnikov identitet (angl. identity providers, IdPs).

2.4 Samo-upravljanje identitete

Samo-upravljane identitete je nov princip implementacije digitalne identitete v računalništvo, ki sledi principom decentralizacije in usmerjenosti v uporabnika [21, 2]. Prvotni namen SSI je sama digitalna identiteta, torej digitalni identifikatorji in dokumenti oz. certifikati, vendar se lahko uporabi tudi za omejevanje dostopa v računalniških sistemih. Večina standardov nastaja pod Svetovnim spletnim konzorcijem (angl. World Wide Web Consortium, W3C) in Fundacijo za decentralizirano identiteto (angl. Decentralized Identity Foundation, DIF) [27, 6]. Sistem zaupanja v SSI temelji na treh akterjih in sicer izdajatelju (angl. issuer), imetniku (angl. holder) in preveritelj (angl. verifier). Njihova interakacija in prenos podatkov sta prikazana na sliki 1. Naloga izdajatelja je izdaja podatkov v kriptografski preverljivi obliki, kar pomeni, da so digitalno podpisani. Vsi podatki, ki so potrebni za delovanje sistema, na primer javni ključi za verificiranje digitalnih podpisov in register, ki hrani status digitalnih dokumentov (v obliki ZgoščenaVrednost(dokument) → veljaven ali neveljaven), so shranjeni v zaupanja vrednem preverljivem podatkovnem registru (angl. verifiable data registry, VDR), ki ga navadno predstavlja decentralizirano omrežje

tehnologije veriženja blokov, s katerim je mogoče implementirati sistem brez centralne točke. Izdajatelji izdajo podatke imetniku, ki jih navadno hrani pri sebi v lokalni hrambi ali šifrirano na spletnem strežniku oz. oblačni storitvi. S takšnim upravljanjem in hrambo podatkov ima le imetnik dostop do svojih podatkov. Imetnik lahko nato sam deli podatke s preveriteljem, pri čemer lahko slednji preveri verodostojnost podatkov brez neposredne interakcije z izdajateljem, saj lahko pridobi informacije o izdajatelju, tj. njegove javne ključe oz. digitalni certifikat X.509, iz skupnega podatkovnega registra VDR. Pri deljenju podatkov s preveriteljem, lahko imetnik tudi izbere le njihovo podmnožico, kar se imenuje izbirno razkritje (angl. selective disclosure), ki v kombinaciji z Zero Knowledge Proof-i (ZKP) omogoča razkritje in dokaz o lastništvu podatkov ter dokazovanje predikatov, na primer starejši od 18 let, brez razkritja njihove dejanske vrednosti in digitalnega podpisa izdajatelja [15, 37]. Osnovni komponenti, ki sestavljata SSI, in skrbita za standardno strukturo oz. obliko podatkov so decentralizirani identifikatorji (angl. decentralized identifier, DID) in preverljive overilnice (angl. verifiable credential, VC).



Slika 1: Model zaupanja pri digitalni identiteti SSI.
Vir: lasten

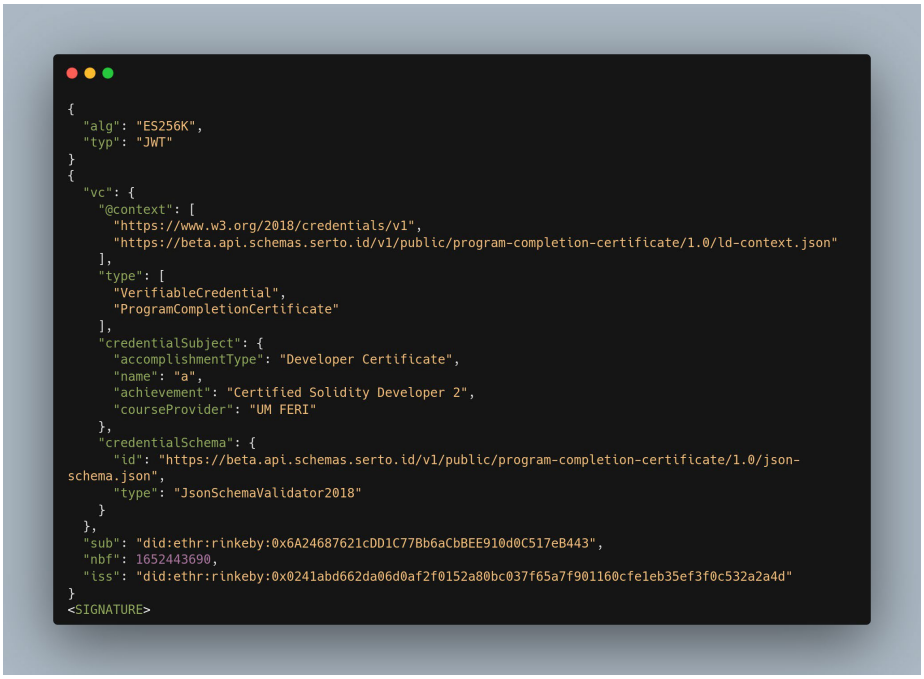
2.4.1 DID

DID je nova oblika identifikatorja, ki temelji na Uniform Resource Identifier (URI) in enolično identificirajo vsako entiteto v SSI sistemu [4]. Primer DID je did:ebsi:zvHWX359A3CvfJnCYaAiAde, kjer prvi del did definira, da URI prestavlja decentraliziran identifikator, drugi del, imenovan DID metoda (angl. DID method), pove kateri VDR se uporablja za hrambo podatkov, v tem primeru omrežje European Blockchain Services Infrastructure (EBSI), in tretji del predstavlja unikaten identifikator v tem omrežju [16]. Entitete pri ustvarjanju digitalne identitete registrirajo svoj identifikator v omrežje z DID dokumentom (angl. DID document). DID sam po sebi navadno ne hrani podatkov, na primer javnega ključa entitete, ampak ga je potrebno razrešiti (angl. DID resolution) s klicem na VDR. Ta klic vrne DID dokument, navadno v obliki JavaScript Object Notation Linked Data (JSON-LD), ki vsebuje vse potrebne informacije o entiteti za verificiranje njenih digitalnih podpisov. JSON-LD je podatkovni format, ki poleg podatkov v obliki JSON vsebuje tudi polje @context, s katerim se sklicuje na opis semantičnih definicij posameznih polj JSON-a. To omogoča interoperabilnost digitalnih dokumentov različnih podatkovnih shem.

Samo entiteta, ki ima komplementarni ključ, tj. zasebni ključ, od tistega v DID dokumentu, se lahko predstavi kot lastnik DID identifikatorja. DID dokument lahko entiteta tudi spremeni, kar se pogosto uporablja za rotacijo ključev (angl. key rotation). Za različne namene je bilo ustvarjenih več DID metod, na primer did:ebsi, did:ethr, did:key in did:peer, pri čemer vsaka služi svojemu namenu in vse sledijo W3C specifikaciji [4].

2.4.2 VC

Podatkovni model VC definira skupen splošen standard za vse digitalne dokumente [25]. Specifikacijo prav tako razvija W3C [27]. VC model sledi temeljem SSI. Njegova prednost je strojno berljiva oblika podatkov, saj se v specifikaciji priporoča format JSON, kateremu lahko natančno določimo strukturo z JSON shemo. Medtem ko DID predstavlja identifikator osebe, VC predstavlja dejanske podatke in informacije o entiteti. VC navadno vsebuje naslednje podatke: identifikator lastnika podatkov, identifikator izdajatelja, tip digitalnega dokumenta, jedrne informacije za ta tip dokumenta in digitalni podpis izdajatelja. VC-ji so najpogosteje shranjeni v digitalnih denarnicah uporabnika, kjer so uporabniku in le njemu cel čas na voljo ter samo on jih lahko deli z drugimi entitetami. Uporabnik pri delitvi VC-ja preveritelju generira preverljivo predstavitev (angl. verifiable presentation, VP), ki lahko vsebuje vse ali le podmnožico podatkov v VC. Preveritelj lahko verificira podatke z resolucijo DID identifikatorja izdajatelja, s čimer pridobi njegov javni ključ, in preverjanjem digitalnega podpisa. Jedrni podatki VC-jev so navadno v obliki JSON-LD, tako kot DID dokumenti. Toda digitalni podpis dokumenta se lahko doda na različne načine, pri čemer sta najpogostejši obliki Json Web Token (JWT) in JSON Linked Data Proofs (JSON-LD Proofs). Slika 2 prikazuje primer VC-ja v obliki JWT.



Slika 2: Primer preverljive overilnice v formatu JWT.
Vir: lasten.

Za celotno delovanje SSI ekosistema je nujno potrebnih še veliko ostalih standardov. Za pošiljanje in izmenjavo VC ter VP med vsemi akterji se uporabljajo različni protokoli, na primer DIDComm ali Self-Issued OpenID Connect Provider (SIOP OIDC), ki definirajo zaporedje in strukturo sporočil potrebnih za varno izmenjavo dokumentov [5, 22]. Ena izmed najpomembnejših komponent so digitalne denarnice, pogosto imenovani tudi agenti, ki hranijo zasebne kriptografske ključe in digitalne dokumente uporabnikov. Primeri takšnih denarnic so uPort, Gataca in Lissi [38]. Denarnice so navadno samostojne (mobilne) aplikacije, ki implementirajo vse potrebne operacije in protokole za SSI. Poleg denarnic se razvija tudi več ogrodij za enostavnejše razvijanje aplikacij temelječih na SSI, pri čemer ima vsako določene prednosti, na primer boljša podpora za integracijo v mobilne

aplikacije in razvoj strežniške aplikacije. Ena izmed najbolj uporabljenih orodij so Veramo[3], Hyperledger Aries[4], DID JWT VC[5] itd.

3 Razširitve MetaMask

Denarnica MetaMask [19] ponuja veliko število funkcionalnosti in načinov uporabe, vendar razvoj novih funkcionalnost poteka počasneje od razvoja ekosistema, v katerem se uporablja. Ta težava se lahko reši z razvojem in implementacijo lastne denarnice, kar pa je precej zahteven in dolgotrajen proces. MetaMask v te namene nudi popolno virtualizirano izvajalno okolje, v katerem se lahko izvajajo razširitve, ki razširijo funkcionalnost osnovne kripto denarnice. Izvajalno okolje, ki ga uporabljajo, je podnabor programskega jezika Javascript imenovan Secure EcmaScript (SES)[6], ki ga razvija podjetje Agoric[7].

3.1 Snaps

Najpopularnejša kripto denarnica, MetaMask, je s sistemom Snaps omogočila varen razvoj razširitev. Ta sistem omogoča razvoj poljubne razširitve, od preprostih aplikacij, ki omogočajo shranjevanje podatkov v stanje denarnice MetaMask, do kompleksnejših aplikacij, ki omogočajo podporo za druge verige blokov. Nekaj takšnih Snap-ov je bilo že implementiranih, med njimi Btcsnap [12], Solsnap [13], ter Filsnap [14]. Kot je razvidno iz imen, ti Snap-i omogočajo interakcijo z omrežji Bitcoin, Solana in Filecoin. Seznam vseh implementiranih Snap-ov se nahaja v aplikaciji Awesome Snaps [10].

MetaMask Snaps je JavaScript program, ki se izvaja v izoliranem okolju znotraj denarnice MetaMask. Ob obstoječih MetaMask metodah za oddaljen klic postopka (angl. remote procedure call, RPC) lahko s Snap-i ustvarimo nove RPC metode, katere lahko uporabijo dApp-i za nove funkcionalnosti. Trenutno je razvoj Snap-ov podprt samo v denarnici MetaMask Flask, namenjeni razvijalcem, s ciljem, da se v bližnji prihodnosti integrirajo v glavno denarnico MetaMask.

4 Dodajanje funkcionalnosti elektronskega podpisovanja poverilnic

4.1 Zasnova

Uporaba tehnologij VC omogoča številne inovativne rešitve za obstoječe probleme. Preprost primer takšnega problema je preverljiv dokaz polnoletnosti za namen vhoda v diskoteko. Ce želi Anita danes dokazati polnoletnost, bo morala pokazati svoj osebni dokument. Iz več razlogov to ni idealna rešitev. Anita lahko pokaže izposojen ali celo ponarejen osebni dokument, kar ni nujno, da varnostnik pred diskoteko tudi opazi. Prav tako pridobi varnostnik iz Anitinega osebnega dokumenta več podatkov, kot je dejansko potrebnih. Pri dokazu polnoletnosti ni potreben EMŠO, naslov bivališča ali točen datum rojstva. Potrebno je samo dejstvo, da je Anita starejša od 18 let ali ne.

Ta banalen problem lahko rešimo z uporabo tehnologije SSI. Z uporabo VC-ja in svojega DID-a, lahko Anita dokaže, ne samo da je ona lastnica tega VC-ja, temveč tudi da je polnoletna, brez da varnostniku izda več osebnih podatkov. Pristnost in veljavnost tega VC-ja lahko prodajalec preveri kadarkoli in kjerkoli. S trenutnimi tehnologijami takšne rešitve ne bi mogli implementirati. Entiteta, ki je izdala VC, bi morala biti dostopna pri

[3] https://veramo.io/
[4] https://www.hyperledger.org/use/aries
[5] https://github.com/decentralized-identity/did-jwt-vc
[6] https://github.com/endojs/endo/tree/master/packages/ses
[7] https://agoric.com/

verifikaciji podatkov, kar pri uporabi tehnologije SSI ni potrebno. Prav tako veliko prednost tehnologije SSI predstavlja zasebnost in varnost podatkov. Izdajalec VC-jev ne vidi, kje so podatki kasneje uporabljeni. Uporabnik ima popoln nadzor nad tem, s kom deli VC, in katere podatke iz samega VC-ja bo delil.

Žal je uporaba tehnologije SSI trenutno zelo zahtevna. Če želi uporabnik uporabiti SSI koncepte, mora namestiti dodatno aplikacijo. Zaupati mora, da aplikacija zagotavlja varnost podatkov in kriptografskih ključev. Trenutno še ne obstaja web3 aplikacija, ki bi omogočala preprosto in varno uporabo DID-ov ter VC-jev in hkrati služila kot kripto denarnica. Sicer obstaja ogromno denarnic za kripto valute, vendar nobena ne omogoča uporabe tehnologije SSI. Z željo, da bi omogočili uporabo tehnologije SSI milijonom obstoječih MetaMask uporabnikov, smo se lotili razvoja t. i. SSI Snap-a. Pri snovanju SSI Snap-a smo si zamislili preprost primer poteka delovanja; VC ponudnik izda VC izbranemu računu v denarnici MetaMask. Ta VC je nato s pomočjo RPC metod SSI Snapa shranjen v stanje MetaMask denarnice. Po potrebi lahko uporabnik s pomočjo RPC metod z VC-jem generira VP in ga posreduje aplikaciji, ki ga zahteva.

S prej omenjeno tehnologijo MetaMask Snaps smo se lotili razvoja aplikacije SSI Snap. Vendar smo pred samim začetkom implementacije morali razrešiti dilemo, katero DID metodo uporabiti. Vsak uporabnik mora biti namreč v sistemu unikatno identificiran in imeti popoln nadzor nad svojo identiteto. V SSI svetu se ta identiteta imenuje DID. Kot smo prej omenili, DID temelji na DID metodi, ki pove, kje in kako so DID dokumenti ustvarjeni, razrešeni, posodobljeni ter deaktivirani.

Na trgu je trenutno že veliko DID metod. Ena najpopularnejših se imenuje ”did:ethr” [11]. Ta metoda uporabi obstoječe Ethereum naslove (angl. Ethereum address) kot samostojne DID-e. Z drugimi besedami, vsak Ethereum račun je DID, kjer se ta začne s prepono ”did:ethr:<omrežje>:”, ter nadaljuje s samim Ethereum naslovom. V praksi to pomeni, da že imajo vsi uporabniki denarnic MetaMask svoje DID-e, katerim manjka le funkcionalnost za pravilno uporabo in izkoriščanje njihovega potenciala. Spremembe DID dokumentov se objavijo in hranijo na verigi blokov Ethereum. Za to verigo blokov smo se odločili zaradi več razlogov:

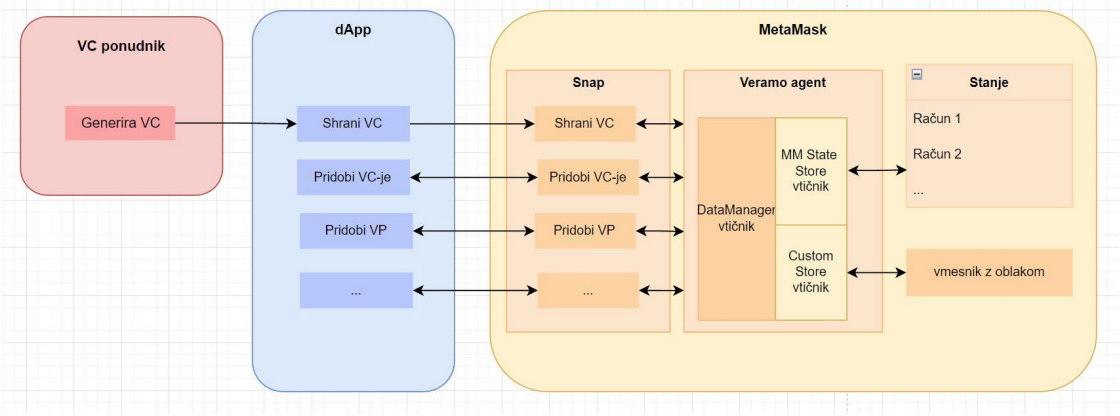
- ob Bitcoin-u je Ethereum najbolj decentralizirana veriga blokov,
- je najpopularnejša in najbolj uporabljena veriga blokov,
- ima veliko skupnost razvijalcev in veliko preizkušenih orodij, med katere spadajo tudi številna SSI orodja ter DID metoda did:ethr,
- v praksi se DID dokumenti spreminjajo zelo redko, zato pristojbine (angl. fees) za njihove spremembe na verigi blokov Ethereum niso visoke,
- denarnica MetaMask ima najboljšo podporo za Ethereum.

Metodo did:ethr bi sicer lahko zamenjali s katerokoli drugo metodo, npr. z did:ebsi, ki se uporablja za razvoj digitalne identitete na Evropskem trgu.

4.2 Arhitektura nadgrajene kripto denarnice

Jedro SSI Snap-a sestavlja ogrodje (angl. framework) Veramo [24], ki je zmogljiv in modularen API za preverljive podatke (angl. verifiable data) in SSI. Z ogrodjem razvijamo agenta, ki omogoča ustvarjanje in uporabljanje DID-ov, VC-jev in VP-jev. Razvijalci ogrodja Veramo so implementirali tudi več vtičnikov (angl. plugins) imenovane DIDManager, KeyManager, ter KeyPairManager, ki zagotavljajo istoimenske funkcionalnosti. Agent Veramo je prav tako zmožen validirati VC-je glede na podano shemo in ustvariti VP-je iz podanih VC-jev. Ker za ogrodje Veramo ne obstaja vtičnik, ki bi omogočal shranjevanje VC-jev, smo ta vtičnik implementirali sami. Implementirali smo razširljiv vtičnik imenovan Veramo VC Manager, ki omogoča upravljanje z VC-ji na podoben način kot delujejo vtičniki DIDManager, KeyManager ter KeyPairManager. Jedro našega vtičnika predstavlja abstrakten razred s funkcijami *import*, *get*, *delete* in *list*. Razvili smo tudi vtičnik imenovan SnapDataStore, ki implementira prej

omenjen abstraktni razred z zmožnostjo shranjevanja podatkov v stanje denarnice MetaMask. Razširljivost teh vtičnikov predstavlja veliko prednost za nadaljnje implementacije, kot je npr. shranjevanje podatkov v oblaku. Končna arhitektura SSI Snap-a je prikazana na sliki 3.



Slika 3: SSI Snap arhitektura.
Vir: lasten

SSI Snap shranjuje DID-e, VC-je in pare ključev (angl. keypairs) v stanje denarnice MetaMask. Pri prvi spremembi stanja se ustvari objekt imenovan SSISnapState. V ta objekt so dodani novi objekti, poimenovani po naslovu računa MetaMask, ki jih je ta naslov ustvaril. Znotraj teh objektov se hranijo DID-i, VC-ji, in pari ključev, kot je prikazano na sliki 4.

```
{
  ...
  objectCreatedByOtherSnaps,
  SSISnapState:
    {
      0xBea807A8...e59D:
        {
          snapKeyStore: Record<string, IKey>,
          snapPrivateKeyStore: Record<string, IKey>,
          identifiers: Record<string, IIdentifier>,
          vcs: VerifiableCredential[]
        },
      0x8Db2a08D...caD7:
        {
          snapKeyStore: Record<string, IKey>,
          snapPrivateKeyStore: Record<string, IKey>,
          identifiers: Record<string, IIdentifier>,
          vcs: VerifiableCredential[]
        },
       ...,
    },
}
```

Slika 4: Arhitektura stanja denarnice MetaMask.
Vir: lasten

Pri implementaciji SSI Snap-a se je pojavilo vprašanje, kako izvesti podpisovanje VP. Edini način, da bi VP podpisali z računom MetaMask, bi zahteval izvoz zasebnega ključa, saj MetaMask ne ponuja potrebnih metod za

podpisovanje VP-jev. Zaradi varnostnih pomislekov z omenjenim pristopom smo se odločili, da namesto izvoza zasebnega ključa ustvarimo nov DID. Vloga tega DID-a bo izključno podpisovanje VP-jev. Toda tukaj se pojavijo dodatni izzivi. DID lahko uporabi samo VC-je, ki si jih lasti, in ker so v večini primerov VC-ji izdani DID-u računa MetaMask, je podpis novega DID-a neveljaven. Ta problem rešimo tako, da novemu DID-u dodelimo pravico podpisovanja DID-ov od računa MetaMask. To storimo tako, da DID-u od računa MetaMask s pomočjo knjižnice ethr-did ustvarimo nov delegat. Vse omenjene akcije se zgodijo znotraj SSI Snap-a.

4.3 Validacija

S prototipno platformo smo želeli prikazati delovanje SSI Snap-a. Ustvarili smo platformo, na katero se lahko uporabnik poveže s svojo denarnico MetaMask, namesti oziroma posodobi SSI Snap, opravi test imenovan Solidity Developer Course in po uspešni opravitvi tega testa prejme VC. Ta VC vsebuje podatke, da je lastnik povezane denarnice MetaMask uspešno opravil test. Na tej platformi lahko uporabnik tudi izpiše seznam vseh shranjenih VC-jev. Implementirali smo tudi "skrivno sobo", v katero lahko uporabnik dostopa samo z veljavnim VP-jem. Demo se lahko brezplačno preizkusi[8]. Za namen testiranja se mora uporabiti MetaMask Flask (različica >10.18.0) in imeti nekaj ETH na testnem omrežju Rinkeby. Platforma je sestavljena iz obličja (angl. frontend), ki komunicira z zalednim delom (angl. backend), ki skrbi za VC-je in VP-je.

Uporabniški vmesnik, ki je prikazan na sliki 5, je implementiran s pomočjo ogrodja React, pri čemer je izgled strani implementiran s pomočjo ogrodja Tailwind CSS. Zaledje je implementirano s pomočjo ogrodja Express.js. Jedro zaledja predstavlja agent Veramo, ki je uporabljen za generiranje VC-jev in preverjanje VP-jev.

Slika 5: Uporabniški vmesnik platforme.
Vir: lasten

Za dostop do vsebine platforme je potrebna povezava z denarnico MetaMask. Ob prvotni povezavi aplikacija preveri, ali ima MetaMask že nameščen SSI Snap. V primeru, da ga nima, zaprosi uporabnika za njegovo namestitev. V primeru, da ima MetaMask nameščeno starejšo različico SSI Snap-a, bo posodobljena na najnovejšo verzijo. Po uspešni povezavi lahko uporabnik začne uporabljati platformo. V zavihku "Profile" lahko izpiše vse VC-je shranjene v stanju svoje denarnice MetaMask. V primeru, da nima še nobenega, lahko v zavihku "Course" opravi preprost Solidity test, in si pridobi VC, ki se generira na zaledju, kjer so tudi na varen način shranjeni kriptografski ključi izdajatelja. Ta VC vsebuje podatke, da je povezana denarnica MetaMask uspešno opravila test. Uporabnikom, ki so uspešno opravili test se pojavi nov zavihek "Secret Room", do katerega lahko dostopajo samo, če priložijo VP, ki vsebuje VC za opravljen Solidity test. Če je preveritev VP-ja na zaledju uspešna, bo uporabniku odobren dostop do skrivne sobe.

[8] https://blockchain-labum.github.io/course-dapp/

Za generiranje VC-jev sta na obličju potrebna ime uporabnika in DID povezanega račun. Ti podatki so nato posredovani zaledju, kjer se DID-u uporabnika izda VC, ki ga nato zaledje vrne obličju, da se shrani v denarnico MetaMask. Del vsebine izdanega VC-ja je prikazana na sliki 6. Pri preverjanju VP-jev mora obličje posredovati VP in Ethereum naslov povezanega računa. Pri preverjanju VP-ja zaledje preveri pravilnost JWT podpisov VP-ja in prisotnost vseh VCjev. Preveri se tudi, ali VP sploh vsebuje pravilen VC, in če je lastnik VC-ja res povezan račun MetaMask. Na koncu se še preveri, če je VP podpisal DID, kateremu je bil izdan VC. V primeru, da to ne drži, se preveri, ali je DID, ki je podpisal VP, prisoten v DID dokumentu od prejemnika VC-ja kot delegat. Če so izpolnjeni vsi pogoji, zaledje vrne odgovor, da je bil VP uspešno preverjen.

Podrobnejši opis prototipa je na voljo v repozitoriju GitHub[9].


```
"verifiableCredential": [
    {
        "credentialSubject": {
            "accomplishmentType": "Developer Certificate",
            "learnerName": "a",
            "achievement": "Certified Solidity Developer 2",
            "courseProvider": "https://blockchain-lab.um.si/",
            "id": "did:ethr:rinkeby:0x6A24687621cDD1C77Bb6aCbBEE910d0C517eB443"
        },
        "issuer": {
            "id": "did:ethr:rinkeby:0x0241abd662da06d0af2f0152a80bc037f65a7f901160c
        },
        "type": [
            "VerifiableCredential",
            "ProgramCompletionCertificate"
        ],
        "credentialSchema": {
            "id": "https://beta.api.schemas.serto.id/v1/public/program-completion-c
            "type": "JsonSchemaValidator2018"
        },
        "@context": [
            "https://www.w3.org/2018/credentials/v1",
            "https://beta.api.schemas.serto.id/v1/public/program-completion-certifi
        ],
        "issuanceDate": "2022-05-13T12:08:10.000Z",
        "proof": {
            "type": "JwtProof2020",
```


Slika 6: Del vsebine izdanega VC-ja.
Vir: lasten

5 Zaključek

V prispevku smo podrobno predstavili številne nove koncepte povezane s tehnologijo veriženja blokov oz. s konceptom decentralizacije. Izhajali smo iz trenutne kompleksnosti področja povezanega z digitalnimi (kripto) denarnicami, kjer so tako v raziskavah kot na trgu številne različice in tipi, pri čemer pa vsaka rešuje zgolj en del celote oz. zagotavlja zgolj specifične funkcionalnosti. Posledično smo ugotavljali, kako je kripto denarnica MetaMask de-facto standardna na programski opremi temelječa kripto denarnica, ki pa kot rečeno ponuja zgolj omejen nabor funkcionalnosti, povezane z verigami blokov oz. zamenljivimi in nezamenljivimi žetoni itn. Prav tako smo predstavili novo področje upravljanja digitalnih identitet, ki prav tako izhaja iz tehnologije veriženja blokov in temelji na decentralizaciji, tj. samo-upravljane identitete oz. SSI. Tako smo identificirali izziv, kako zagotoviti hitrejše sprejemanje novih tehnologij, kot je SSI, ki prav tako zahteva določeno vrsto digitalnih denarnic. Podrobneje smo izziv razvili na način, da smo iskali rešitev za nadgradnjo priljubljene kripto denarnice MetaMask s funkcionalnostmi, ki bi podprle zahteve, ki jih prinaša tehnologija SSI. Izziv smo rešili na način, da smo uspešno načrtovali in tudi razvili rešitev, ki s pomočjo t. i. Snap-ov razširi funkcionalnosti kripto denarnice MetaMask. Na

[9] https://github.com/blockchainlab-um/ssi-snap

ta način smo s kompleksnim primerom tudi demonstrirali, kako bi se lahko na osnovi sistema Snap MetaMask poljubno nadgradil.

Predstavljeni in validirani koncepti odpirajo pot do novih razširitev MetaMask, kakor tudi do reševanja drugih izzivov, kot je npr. glasovanje v decentralizirano avtonomnih organizacijah (angl. decentralized autonomous organisation, DAO) zgolj na osnovi količine žetonov, ki si jo posameznik lasti, kar samo po sebi predstavlja t.i. plutokratičen problem. V prihodnosti tako načrtujemo nadgraditi SSI Snap ter platformo Snapshot na način, da bosta omogočali uporabnikom, da z eno kripto denarnico upravljajo tudi DAO glasovanje, pri čemer pa se bo le to izvedlo zgolj v primeru, ko uporabnik na osnovi primerih VC/VP-jev pokaže svojo kompetentnost na področju tematike glasovanja.

Literatura

[1] BUTERIN Vitalik »Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform«.

[2] ČUČKO Špela, TURKANOVIĆ Muhamed »Decentralized and self-sovereign identity: Systematic mapping study«, IEEE Access 9, 2021, pp. 139009-139027.

[3] DANNEN Chris »Introducing Ethereum and solidity«, Springer, številka 1, 2017,

[4] https://www.w3.org/TR/did-core, Decentralized Identifiers (DIDs) v1.0, obiskano 1. 7. 2022.

[5] https://identity.foundation/didcomm-messaging/spec, DIDComm Messaging Specification, obiskano 1. 7. 2022.

[6] https://identity.foundation, DIF – Decentralized Identity Foundation, obiskano 1. 7. 2022.

[7] https://eips.ethereum.org/EIPS/eip-191, EIP-191: Signed Data Standard, obiskano 7. 7. 2022.

[8] https://eips.ethereum.org/EIPS/eip-4361, EIP-4361: Sign-In with Ethereum, obiskano 7. 7. 2022.

[9] https://eips.ethereum.org/EIPS/eip-712, EIP-712: Ethereum typed strucuted data hashing and signing, obiskano 7. 7. 2022.

[10] https://github.com/piotrroslaniec/awesome-metamask-snaps, Github page for Awesome Snaps, obiskano 7. 7. 2022.

[11] https://github.com/uport-project/ethr-did, Github page of did:ethr, obiskano 7. 7. 2022.

[12] https://github.com/KeystoneHQ/btcsnap, Github page of btcsnap, obiskano 7. 7. 2022.

[13] https://github.com/cavanmflynn/solsnap, Github page of solsnap, obiskano 7. 7. 2022.

[14] https://github.com/ChainSafe/filsnap, Github page of filsnap, obiskano 7. 7. 2022.

[15] GOLDREICH Oded, OREN Yair »Definitions and properties of zero-knowledge proof systems«, Journal of Cryptology 7.1, 1994, pp. 1-32.

[16] https://ec.europa.eu/digital-buildingblocks/wikis/display/ebsi, Home – EBSI, obiskano 1. 7. 2022.

[17] https://docs.metamask.io/guide, Introduction – MetaMask Docs, obiskano 1. 7. 2022.

[18] https://ledger.com, Ledger hardware wallet, obiskano 7. 7. 2022.

[19] https://metamask.io, MetaMask, obiskano 7. 7. 2022.

[20] https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki, Mnemonic code for generating deterministic keys, obiskano 7. 7. 2022.

[21] PREUKSCHAT Alex, REED Drummond »Self-sovereign identity«, Manning Publications, 2021.

[22] https://openid.net/developers/specs, Specifications – OpenID, obiskano 1. 7. 2022.

[23] https://trezor.io, Trezor hardware wallet, obiskano 7. 7. 2022.

[24] https://veramo.io, Veramo – A JavaScript Framework, obiskano 7. 7. 2022.

[25] https://www.w3.org/TR/vc-data-model, Verifiable Credentials Data Model v1.1, obiskano 1. 7. 2022.

[26] https://walletconnect.com, WalletConnect web3 standard, obiskano 7. 7. 2022.

[27] https://www.w3.org, World Wide Web Consortium (W3C), obiskano 1. 7. 2022.

[28] YAGA Dylan, MELL Peter M., ROBY, Nick, SCARFONE Karen, »Blockchain Technology Overview«, NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, 2018, https://doi.org/10.6028/NIST.IR.8202.

[29] KASPARS Zīle, STRAZDINA Renāte »Blockchain Use Cases and Their Feasibility«, Applied Computer Systems 23.1, 2018, pp. 12-20, https://doi.org/10.2478/acss2018-0002.

[30] NAKAMOTO Satoshi »A Peer-to-Peer Electronic Cash System, 2008.

[31] PODGORELEC Blaž, REK Patrik, STREHAR Miha, TURKANOVIĆ, Muhamed, HERIČKO Marjan (ur.), KOUS, Katja (ur.) »Vzpostavitev konzorcijskega omrežja Ethereum«, Sodobne informacijske tehnologije in storitve: OTS 2019: zbornik štiriindvajsete konference, Maribor, 18. in 19. junij 2019. 1. izd. Maribor: Univerzitetna založba Univerze, 2019. Str. 69-79. ISBN 978-961-286-283-1.

[32] SRIMAN B., KUMAR Ganesh S., PRABAKARAN Shamili »Blockchain Technology: Consensus Protocol Proof of Work and Proof of Stake«, Intelligent Computing and Applications, Proceedings of ICICA 20ver (pp.395-406), 2020, https://doi.org/10.1007/978-981-15-5566-4_34.

[33] https://apla.readthedocs.io/en/latest/concepts/consensus.html, Proof of Authority conesnsus, obiskano 7. 7. 2022.

[34] PODGORELEC Blaž, TURKANOVIĆ Muhamed, HERIČKO Marjan (ur.), KOUS Katja (ur.) »Implementacija nadgradljivosti in zamenljivosti pametnih pogodb na platformi Ethereum«, Sodobne informacijske tehnologije in storitve: OTS 2018: zbornik triindvajsete konference, Maribor, 19. in 20. junij 2018. 1. izd. Maribor: Univerzitetna založba Univerze: Fakulteta za elektrotehniko, računalništvo in informatiko, 2018. Str. 8-19. ISBN 978-961-286-163-6, ISBN 978-961-286-162-9.

[35] REK Patrik, TURKANOVIĆ Muhamed, HERIČKO Marjan (ur.), KOUS Katja (ur.) »Orodja za podporo celovitemu razvoju decentraliziranih aplikacij«, Sodobne informacijske tehnologije in storitve: OTS 2019: zbornik štiriindvajsete konference, Maribor, 18. in 19. junij 2019. 1. izd. Maribor: Univerzitetna založba Univerze, 2019. Str. 173-180. ISBN 978-961-286-283-1.

[36] KERŠIČ Vid, ŠTUKELJ Primož, KAMIŠALIĆ Aida, KARAKATIČ Sašo, TURKANOVIĆ Muhamed, PRIETO Javier (ur.) »A blockchain- and ai-based platform for global employability«, Blockchain and applications: international congress [on Blockchain and applications, BLOCKCHAIN 2019, held in Avila, Spain, from 26th to 28th June, 2019]. Cham [etc.]: Springer, cop. 2020. Vol. 1010, str. 161-168. Advances in intelligent systems and computing (Print), 1010. ISBN 978-3-030-23812-4, ISBN 978-3-030-23813-1. ISSN 2194-5357.

[37] PODGORELEC Blaž, TURKANOVIĆ Muhamed, HERIČKO Marjan (ur.), KOUS Katja (ur.) »ZKP (zero-knowledge proof) pod drobnogledom«, Sodobne informacijske tehnologije in storitve: OTS 2019: zbornik štiriindvajsete konference, Maribor, 18. in 19. junij 2019. 1. izd. Maribor: Univerzitetna založba Univerze, 2019. Str. 109-118. ISBN 978-961-286-283-1.

[38] ČUČKO Špela, ŠUMAK Boštjan, TURKANOVIĆ Muhamed, HÖLBL Marko (ur.) »Načrtovalski vzorci uporabniškega vmesnika samoupravljanih identitet«, "Digitalno desetletje: varno, zeleno in odporno": zbornik: 29. konferenca Dnevi slovenske informatike: Portorož, 11. in 12. maj 2022. 1. izd. Ljubljana: Slovensko društvo Informatika, 2022. 8 str., ilustr. ISBN 978-961-6165-59-4.