

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Gregor Bučar

**Načrt in izvedba decentraliziranega
dnevnika za stavbe**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Vlado Stankovski

Ljubljana, 2023

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuira, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco GNU General Public License, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses/>.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Kandidat: Gregor Bučar

Naslov: Načrt in izvedba decentraliziranega dnevnika za stavbe

Vrsta naloge: Diplomaska naloga na univerzitetnem programu prve stopnje
Računalništvo in informatika

Mentor: prof. dr. Vlado Stankovski

Opis:

Cilj diplomske naloge je integracija tehnologije OriginTrail, ki je osnovana na tehnologiji veriženja blokov, na domeni gradbeništva. Trenutno je poglavitni razlog za zanemarjanje celovitega pristopa razpršenost informacij o gradbenih proizvodih in storitvah, ki so zapisane v različnih oblikah. Nadgradnja se osredotoča na izdelavo decentraliziranega dnevnika stavb za bolj učinkovit ter transparenten pristop h gradnji energetske in trajnostno učinkovitejših stavb.

Title: Design and implementation of a decentralized digital building logbook

Description:

The aim of the diploma thesis is the integration of the Origin Trail technology, which is based on blockchain technology, in the field of construction. Currently, the main reason for neglecting the complete and comprehensive approach is the dispersion of information on construction products and services, which are currently written in different forms. The proposed solution focuses on producing a decentralized digital building logbook for a more efficient and transparent approach to building more energy-efficient and sustainable buildings.

Rad bi se zahvalil svojemu mentorju prof. dr. Vladu Stankovskemu za odzivnost in napotke tekom izdelave diplomske naloge.

Zahvaljujem se tudi članom OriginTrail skupnosti, ki so mi pomagali pri postavitvi lokalnega okolja za testiranje razvitega sistema.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Semantični splet	3
2.1	Povezani podatki	4
2.2	Uporaba JSON-LD	4
2.3	Primerjava s spletom 2.0	5
3	Decentralizirani splet	7
3.1	Tehnologija veriženja blokov	8
3.2	Decentralizirane aplikacije	9
3.3	OriginTrail tehnologija	11
3.4	Etherium omrežje in platforma	18
4	Načrtovanje decentraliziranega dnevnika za stavbe	21
4.1	Doktorska disertacija Matije Königa	23
4.2	Zasnova decentraliziranega dnevnika	23
4.3	Uporaba tehnologij in orodij	25
5	Izvedba decentraliziranega dnevnika za stavbe	29
5.1	Uporabniški vmesnik	29
5.2	Lokalno razvojno okolje OriginTrail DKG	39

5.3 Vmesnik API decentraliziranega dnevnika	40
6 Sklepne ugotovitve	49
Celotna literatura	51

Seznam uporabljenih kratic

kratica	angleško	slovensko
XML	Extensible Markup Language	Razširljivi označevalni jezik
MEAN	MongoDB, Express, Angular, Node.js	MongoDB, Express, Angular, Node.js
REST	Representational state transfer	Prenos reprezentacijskih stanj
RDF	Resource Description Framework	Okvir za opisovanje virov
OWL	Web Ontology Language	Jezik spletne ontologije
ECDSA	Elliptic Curve Digital Signature Algorithm	Algoritem digitalnega podpisa z eliptično krivuljo
ODN	OriginTrail Decentralized Network	Decentralizirano omrežje OriginTrail
DKG	Decentralized Knowledge Graph	Decentralizirani graf znanja
PTP	Peer-to-peer	Medvrstniška povezava
SPARQL	SPARQL Protocol and RDF Query Language	Protokol SPARQL in poizvedovalni jezik RDF
UAI/UAL	Uniform Asset Identifier/Uniform Asset Locator	Enotni identifikator sredstev/enotni lokator sredstev
JWT	JSON Web Token	JSON spletni žeton
API	Application Programming Interface	Aplikacijski programski vmesnik

Povzetek

Naslov: Načrt in izvedba decentraliziranega dnevnika za stavbe

Avtor: Gregor Bučar

Diplomsko delo obravnava koncept semantičnega spleta in tehnologije, ki se uporabljajo za njegovo izvajanje, ter potencial decentraliziranih tehnologij, kot je veriženje blokov, za rušenje tradicionalnih internetnih modelov in omogočanje novih načinov organiziranja in dostopa do informacij. Naslavlja izzive iskanja zanesljivih in zaupanja vrednih podatkov v hitro rastočem in vse bolj povezanem svetu interneta. V okviru diplomske naloge je poudarek na gradbeništvu, kjer se uvajajo novi izdelki in rešitve, vendar so informacije o njih pogosto nepopolne. Predstavljena je zasnova decentraliziranega dnevnika za stavbe s pomočjo tehnologije OriginTrail. Zajema tudi izvedbo zastavljenega sistema, ki deluje kot spletna aplikacija. Izdelana decentralizirana aplikacija stremi k zagotavljanju transparentnih in sledljivih informacij o gradbenih izdelkih in storitvah ter obenem podpira razvoj bolj trajnostnih in energetsko učinkovitih zasnov stavb in gradbenih praks.

Ključne besede: Semantični splet, veriženje blokov, OriginTrail, decentraliziran dnevnik stavb, spletna aplikacija, transparentnost, sledljivost.

Abstract

Title: Design and implementation of a decentralized digital building logbook

Author: Gregor Bučar

This thesis examines the concept of the Semantic Web and the technologies used to implement it, as well as the potential of decentralised technologies such as blockchain to disrupt traditional Internet models and enable new ways of organising and accessing information. It addresses the challenges of finding reliable and trustworthy information in the fast-growing and increasingly connected world of the Internet. The thesis focuses on the construction industry, where new products and solutions are introduced but information about them is often incomplete. The design of a decentralised building log using OriginTrail technology is presented. It also covers the implementation of the conceived system, which works as a web application. The decentralised application developed aims at providing transparent and traceable information on building products and services, while supporting the development of more sustainable and energy efficient building designs and construction practices.

Keywords: Semantic Web, blockchain, OriginTrail, decentralised building log, web application, transparency, traceability.

Poglavje 1

Uvod

V zadnjih desetletjih sta internet in svetovni svet izjemno hitro rasla in postala neizogiben del našega vsakdana. Obsežna količina podatkov pa s seboj prinaša izzive. Eden izmed realnih problemov, s katerimi se soočamo, je problem iskanja zelenega podatka, storitve ali izdelka izmed ponujene množice. Prav tako postajajo zahteve po trajnosti, kakovosti in zaupanju vrednih podatkov vedno pomembnejše, saj lahko le tako zagotovimo, da bodo naše odločitve, strategije in dejavnosti temeljile na zanesljivih in verodostojnih informacijah. To je še posebej pomembno v okoljih, kjer so podatki povezani z varnostjo, zdravjem, okoljem ali finančnimi transakcijami. Pomanjkanje transparentnosti in odgovornosti pri zbiranju in uporabi podatkov lahko vodi do zlorab in goljufij. Poleg tega se soočamo tudi s težavami, povezanimi s pravicami do zasebnosti in varovanjem osebnih podatkov, ter s težavami pri upravljanju in integraciji podatkov iz različnih virov in sistemov.

Diplomska naloga se konkretno osredotoča na omenjene probleme za gradbeni sektor. Tu se v tržnem okolju pojavljajo novi izdelki in rešitve, vendar so informacije o njih pogosto nepopolne in nezadostne, kar povzroča dvome pri uporabnikih in otežuje razvoj tega področja. Potencialni kupci gradbenih izdelkov in storitev so zato pogosto negotovi in neinformirani o uporabnosti različnih materialov, izdelkov in storitev za izvajanje bolj celo-

vitih in trajnostno učinkovitih načrtov in gradenj. Zgradbe so pomemben dejavnik pri ohranjanju okolja in vzdrževanju energetske učinkovitosti, saj porabijo približno 40% celotne energije, ki se uporablja v EU. Tu se soočamo s številnimi izzivi, ki lahko ogrozijo trajnost in učinkovitost stavb. Med glavnimi je pomanjkanje standardov in smernic za energetske učinkovitost stavb, ki bi zagotavljale, da se pri gradnji uporabi sodobne in kompetentne tehnologije ter pristope. Kljub direktivi o stavbah za energetske učinkovitost pa še vedno obstajajo velike razlike med državami glede na to, koliko se posveča izboljšanju energetske učinkovitosti stavb.

Cilj diplomske naloge je preučiti, ter predvsem preizkusiti možnosti, ki jih ponujajo sodobne tehnologije na področju semantičnega spleta in veriženja blokov. Rezultat, ki ga želimo doseči, je izdelana decentralizirana aplikacija, ki bo delovala kot dnevnik stavb in bo naslavljala predstavljene probleme. Posebna pozornost bo namenjena konceptu Spleta 3.0, kjer bo v ozadju za shranjevanje kritičnih podatkov uporabljena tehnologija OriginTrail, na nivoju verige blokov pa Ethereum. Struktura in povpraševanje po podatkih bo definirano in omogočeno s pomočjo tehnologij semantičnega spleta in povezanih podatkov.

V nadaljevanju bo v Poglavju 2 predstavljen semantični splet, njene tehnologije in primerjava s trenutnim spletom. V Poglavju 3 bodo predstavljeni decentraliziran splet, tehnologija veriženja blokov, decentralizirane aplikacije, tehnologija OriginTrail in platforma Ethereum. Sledi Poglavje 4, v katerem bodo opisane zasnova decentraliziranega dnevnika stavb ter poglobitve tehnologije, in orodja, ki so bila uporabljena za razvoj celotnega sistema. Izvedba zastavljene decentralizirane aplikacije bo nato predstavljena v Poglavju 5. Vsebovalo bo pregled uporabniškega vmesnika, nastavitve lokalnega razvojnega okolja OriginTrail in opis aplikacijskega vmesnika sistema. Poglavje 6 pa bo podalo končni sklep ter predloge za nadaljnji razvoj in izboljšavo izdelane aplikacije.

Poglavje 2

Semantični splet

Semantični splet, velikokrat označen tudi kot Splet 3.0 ali splet povezanih podatkov, je koncept novega spleta, ki temelji na ideji da bo splet berljiv ne samo za ljudi, ampak prav tako za naprave. Strani na spletu bi za programske opreme imele pomen oziroma semantiko, kar bi programom omogočilo interakcijo s spletom na enak način kot ljudem. Programi bi si med seboj po semantičnem spletu lahko izmenjevali podatke, brez da bi jih bilo potrebno izrecno načrtovati za medsebojno komunikacijo. [1]

Temeljna ideja za delovanje semantičnega spleta je bila, da bi vsi začeli uporabljali niz standardov za označevanje svojih spletnih strani, kjer bi te označevali z uporabo delov XML kode. Uporaba delov XML bi bil način uporabe metapodatkov o spletni strani, ki na samo prezentacijo ne bi vplivali, obenem pa bi programski opremi omogočalo dostop do podatkov. Pri tem so se pojavile težave, ki jih je naslovil Cory Doctorow, ko je leta 2001 objavil esej o številnih težavah zaradi kredibilnosti prostovoljnih metapodatkov. Doctorow je pravil, da bi bil svet zanesljivih metapodatkov čudovit, vendar je to le iluzija, ki temelji na samoprevarah in napihnjenih tržnih priložnostih. [1, 4]

2.1 Povezani podatki

Načelo povezanih podatkov podajajo smernice za uporabo standardizirane tehnologije za nastavitev povezav na podatkovni ravni med podatki iz različnih virov v en sam globalni podatkovni prostor. Služijo kot nabor dobrih praks za povezovanje in objavljane strukturiranih podatkov na spletu. [2]

Uporaba povezanih podatkov znotraj aplikacije omogoča, da se od začetnega povezanega podatka sledi vdelanim povezavam do drugih podatkov, ki so na voljo na različnih spletnih mestih po spletu. Prav tako se lahko preko sledenja odkrijejo novi viri podatkov v času izvajanja, s čimer se zagotovi bolj popolne odgovore pri pojavitvi novih virov. Pri tem se zagotavlja tudi zmožnost poizvedovanja po zbirnih podatkih, in sicer na podoben način, kot se poizveduje po lokalnih bazah. Tako lahko iz tehnološkega vidika razberemo, da glavni cilj povezanih podatkov ni samo prepoznavanje spletnih dokumentov, temveč tudi identificiranje poljubnih entitet iz vsakdanjega življenja. [16]

2.2 Uporaba JSON-LD

JSON-LD je lahek format in sintaksa za serializacijo povezanih podatkov v JSON. Sintaksa je zasnovana za enostavno integracijo v nameščene sisteme, ki že uporabljajo JSON, in zagotavlja hitro nadgradnjo v JSON-LD. Je idealen format podatkov za programska okolja, spletne storitve REST in nestrukturirane baze podatkov, kot sta Apache CouchDB in MongoDB. Nekatere izmed dodatnih funkcionalnosti formata so univerzalni identifikatorski mehanizem za objekte JSON, način za razločevanje ključev, ki si jih delijo različni dokumenti JSON in uporaba usmerjenih grafov znotraj dokumentov. [17]

Prednost uporabe JSON-LD pri povezanih podatkih je to, da je enostaven in berljiv ter da se lahko uporablja brez kakršnihkoli predznanj o semantičnem trojčku subjekt-predikat-objekt in samih tehnologijah semantičnega spleta. Format označuje elemente na strani in strukturira podatke, s pomočjo ka-

terih si lahko iskalniki pomagajo za bolj vsebinsko organiziranje zadetkov. Uporablja se tudi pri Schema.org, Googlovem grafu znanja ter tudi za aplikacije v domeni biomedicinske informatike. Poleg tega je vgrajen v kontekstu interneta stvari, kjer se uporablja za opisovanje omrežnih vmesnikov naprav IoT. Iskalnik Google priporoča uporabo JSON-LD za strukturirane podatke namesto mikropodatkov. [17]

2.3 Primerjava s spletom 2.0

Poglavitna razlika med sedanjim semantičnim spletom in trenutnim spletom (Splet 2.0) je v načinu organizacije in predstavitve informacij. Pri trenutnem spletu se gre predvsem za človeku berljiv splet s poudarkom na tem, da so informacije ljudem dostopne in lahko razumljive. Nasprotno pa je semantični splet osredotočen na ustvarjanje strojno berljivega spleta z uporabo semantičnih tehnologij, kot sta RDF in OWL, ki omogočata predstavitev informacij na način, razumljiv strojem. Splet 3.0 se osredotoča tudi na decentralizacijo, kjer so podatki porazdeljeni po več platformah in jih ne nadzoruje ena osrednja enota. To omogoča večjo varnost in zasebnost, posamezniki pa imajo tudi večji nadzor nad svojimi podatki. Tako lahko izpostavimo drugo ključno razliko, in sicer da splet 2.0 večinoma uporablja centralizirane strežnike, splet 3.0 pa večinoma uporablja decentralizirana omrežja in platforme za obdelavo, shranjevanje in izmenjavo informacij.

Kot osrednji del trenutnega stanja spleta 3.0 lahko označimo koncept metaverse. Ta se nanaša na virtualni svet, ki je potopljen in interaktiven ter v katerem lahko uporabniki komunicirajo med seboj ter z virtualnimi predmeti. Metaverse velja za naslednji korak v razvoju interneta, kjer so meje med fizičnim in digitalnim svetom zabrisane, uporabniki pa lahko doživijo povezovanje obeh. Za ključnega spodbujevalca metaverse velja tehnologija veriženja blokov, saj omogoča ustvarjanje decentraliziranih virtualnih svetov, v katerih imajo uporabniki popoln nadzor nad svojimi podatki in sredstvi.

Poglavje 3

Decentralizirani splet

Decentralizirani splet, znan kot splet 3, je odprt splet, temelječ na tehnologiji veriženja blokov z osnovno idejo ustvarjanja decentralizirane različice spleta in vračanja nadzora uporabnikom. Poudarek daje predvsem na varnost in vračanju nadzora nad podatki ter nad identiteto uporabnikov. Podatke prisotne na decentraliziranem spletu je težko izbrisati ali spremeniti, saj so razpršeni po več vozliščih.

Poglaviten razlog za razvoj decentraliziranega spleta je dejstvo, da je trenutni splet, ki ga lahko označimo tudi kot socialni splet, centraliziran, tu pa imajo največ vpliva, bogastva in moči velikani, kot so Google, Amazon in Facebook. Tu se pojavljajo problemi glede zbiranja uporabnikovih podatkov brez njihove vednosti in dovoljenja, kar upodablja izgubo demokracije. S tem se tudi zatira konkurenčnost manjših in prihajajočih podjetij, saj nimajo dostopa do enakih količin podatkov končnih uporabnikov. Ena izmed večjih težav je tudi cenzura spletnih strani in medijev, kjer se v nekaterih državah omejuje dostop, kar prav tako prikazuje izgubo nadzora in svobode državljanov. [10]

3.1 Tehnologija veriženja blokov

Tehnologija veriženja blokov temelji na konceptu soglasja, kriptografiji z javnim ključem ter zgoščevalnih funkcijah: [10]

- **Soglasje** - uspešno samo v primeru večinskega sprejetja, lahko se implementira z dokazili, kot so na primer dokazilo o vložku (PoS) ali delu (PoW)
- Uporaba **kriptografske zgoščevalne funkcije** predvsem pri digitalnem podpisovanju, saj so deterministične, se izračunajo hitro, pri uporabi je skoraj nemogoče pridobiti enako zgoščeno vrednost iz različnih podatkov in so enosmerne.
- Pri **kriptografiji javnega ključa** z uporabo javnega ključa, ki je znan vsem, ter privatnega ključa, ki je znan samo lastniku, upravljamo račun na platformi verige blokov.

Veriga blokov je porazdeljena podatkovna baza za shranjevanje zapisov v obliki blokov. Znotraj posameznega zapisa je množica digitalno podpisanih transakcij, kar nam omogoča enostavno dokazovanje pristnosti. Digitalni podpis je v večini implementiran s pomočjo algoritma digitalnega podpisa eliptične krivulje ECDSA. Znotraj blokov je osnovna struktura sestavljena iz povezave do prejšnjega bloka in časovnega žiga, vsebuje tudi zgoščeno vrednost povzetka celotnega bloka ter zgoščeno vrednost prejšnjega bloka. Tako veriga blokov vsebuje najnovejše stanje in zgodovino transakcij. [10]

3.2 Decentralizirane aplikacije

S pojmom decentralizirana aplikacija ali DApp označimo tiste aplikacije, kjer se pri preverjanju transakcije uporablja soglasje - porazdeljena in nezaupljiva oblika dogovora. Pri tem si sodelujoči medsebojno ne zaupajo, transakcija pa uspe takrat, ko se večina udeležencev strinja z izidom transakcije. S tem namesto centralizirane obdelave na enem vozlišču dosežemo decentralizirano obdelavo na neodvisnih vozliščih. Tako se razbremenijo uporabniki, saj mu ni več potrebno zaupati določeni centralni entiteti, ampak samo zasnovi omrežja.

Decentralizirane aplikacije delujejo avtonomno in namesto človeškega posredovanja v večini primerov uporabljajo tako imenovane "žetone", ki se razdelijo v skladu z algoritmom. S tem se zmanjšata lastništvo in nadzor nad aplikacijo. Da lahko določeno aplikacijo označimo kot decentralizirano, mora izpolnjevati pogoje: [8]

1. Biti mora odprtokodna in delovati avtonomno, evidenca delovanja mora biti kriptografsko shranjena v javni verigi blokov.
2. Generirati mora žetone v skladu z določenimi merili, ki se morajo uporabljati znotraj same aplikacije in uporabniki morajo biti za svoje delo plačani v obliki teh žetonov.
3. Protokol aplikacije se lahko spreminja, pri tem pa se o sprejetju predloga odloča z večinskim soglasjem.

Decentralizirane aplikacije lahko delimo glede na to, v katero kategorijo spadajo. Primeri, kjer lahko najdemo uporabo decentralizacije so igre, finance, razvoj, shranjevanje, identiteta in varnost. Lahko jih tudi delimo glede na to, ali v ozadju uporabljajo verigo blokov druge decentralizirane aplikacije, ali pa imajo svojo. Glede na ta kriterij lahko ločimo več skupin: [8]

- Tiste decentralizirane aplikacije, ki imajo v ozadju svojo verigo blokov. Primer so verige Ethereum, Solana, Tezos in druge alternative.
- Decentralizirane aplikacije, ki uporabljajo obstoječe verige blokov. So protokoli, ki za svoje delovanje potrebujejo žetone
- Decentralizirane aplikacije, ki za svoje delovanje uporabljajo obstoječe protokole in so prav tako protokoli, ki za izvajanje funkcionalnosti uporabljajo žetone

3.2.1 Žeton

Na splošno so žetoni digitalne vrednostne enote, ki jih ni mogoče rudariti in obstajajo kot vnosi v verigah blokov. Strogo gledano, žetoni niso kriptovalute, ampak prenosljive enote vrednosti. Namen žetonov je omogočiti dostop do določene funkcionalnosti decentralizirane aplikacije. Vrednost žetona decentralizirane aplikacije se lahko spremeni, vendar žetoni ne predstavljajo lastniških vrednostnih papirjev. Obstajajo različne klasifikacije za žetone glede na njihove značilnosti. Po funkcionalnosti ločimo varnostne in uporabne žetone. Sledeči predstavljajo dostop do storitve, lahko pa tudi delujejo kot menjalno sredstvo znotraj ekosistema. [8]

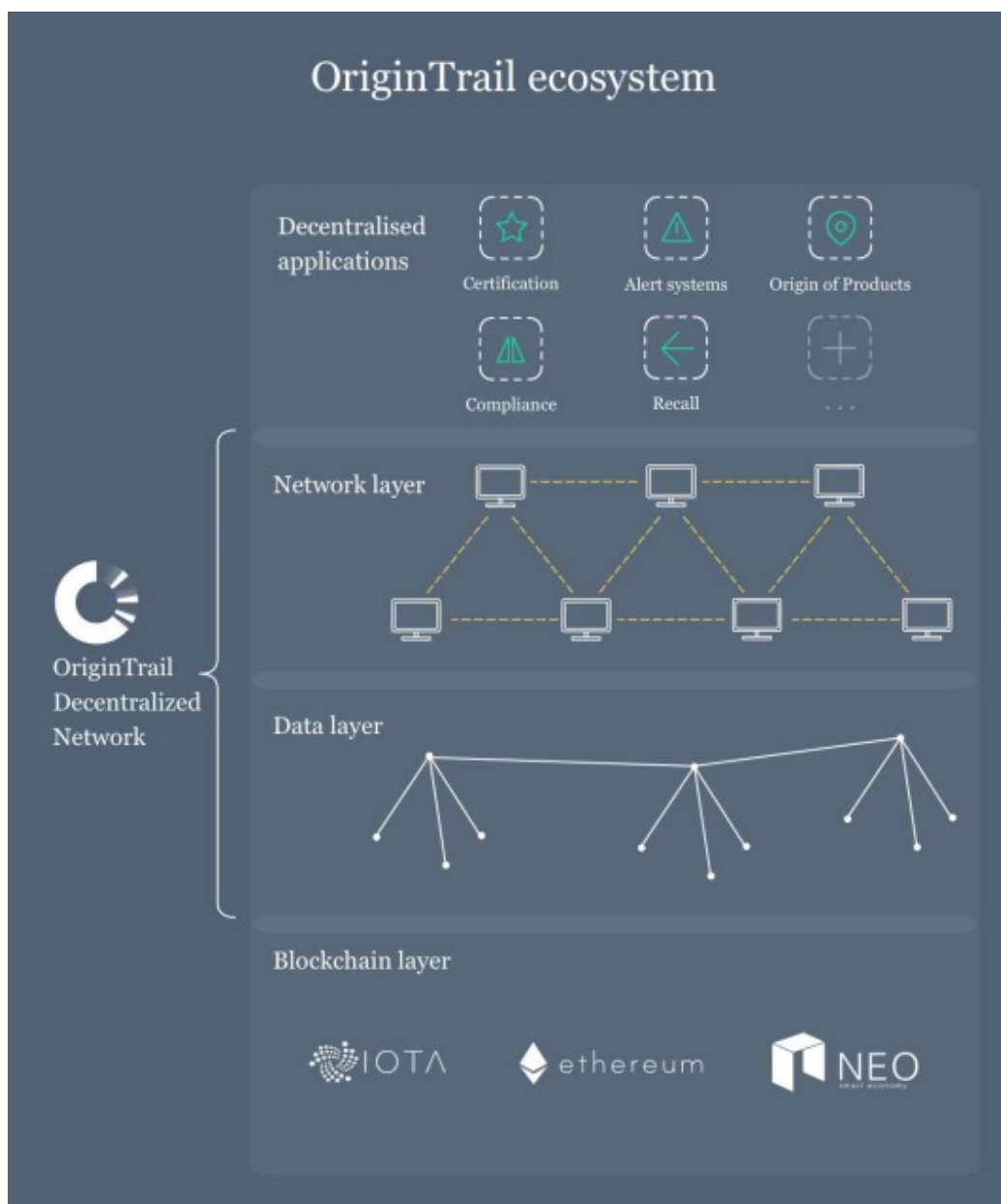
3.3 OriginTrail tehnologija

S pojmom "OriginTrail" označujemo ekosistem, prikazan na sliki 3.1, ki je zasnovan na ekonomiji žetonov z neposrednimi odnosi med uporabniki in vozlišči brez samovoljnih pristojbin. Je prvi namensko izdelan protokol spleta 3 za podporo dobavnim verigam, kjer je največji prispevek v obliki boljše preglednosti, poštenosti in zaupnosti. Razvit je bil s strani iste ekipe, ki je ustanovila tudi Trace Labs, in deluje kot odprtokodni protokol za izmenjavo podatkov. Uporablja decentralizirana vozlišča in omogoča interakcijo z drugimi verigami blokov ter tudi zastarelimi sistemi.

Omogoča preglednost in sledenje, zaupanja vredno izmenjavo podatkov ter ščiti pred goljufijami. Podatki se varujejo in delijo v popolnoma decentraliziranem omrežju grafov znanja, ki ga nadzira umetna inteligenca. Tehnologija ima trenutno na plasti soglasja možnost uporabe verig blokov Ethereum, Gnosis in Polygon, v teku pa je tudi integracija s Polkadot. [5, 18]

Nekateri primeri, kjer protokol OriginTrail prispeva dodatno vrednost v dobavni verigi so: [5]

- Sledenje poreklu in svežini izdelkov
- Preverjanje pristnosti produkta
- Optimizacija regulativnih postopkov
- Zagotavljanje skladnosti dobavnih verig
- Upravljanje zalog
- Vse ostale aplikacije, ki kot osnovo zahtevajo transparentnost dobavne verige



Slika 3.1: OriginTrail ekosistem [18]

3.3.1 OriginTrail ODN

Tehnologija OriginTrail kot vmesna plast povezuje aplikacijski sloj decentraliziranega spleta skupaj z različnimi verigami blokov. Decentralizirano P2P omrežje OriginTrail (ODN) lahko označimo kot osnovo na kateri deluje več implementacij OriginTrail decentraliziranih grafov znanja (DKG). ODN omogoča prenos, obdelavo in povpraševanje podatkov ter prav tako preverjanje njihove celovitosti in razpoložljivosti. Pridobljene podatke lahko nato vključimo v lokalni graf znanja ali podatkovno shrambo. Omrežje ODN že od konca leta 2018 deluje v produkcijskem okolju "mainnet" in ga upravlja več tisoč vozlišč. Rešitev zmanjšuje količino podatkov, shranjenih v verigi blokov, kar posledično zmanjša tudi stroške. [18]

Podatkovna plast ODN

Ukvarja se s funkcionalnostmi upravljanja podatkov in povezovanjem številnih različnih naborov podatkov. Plast se osredotoča na odnose med podatki. Tehnologija, ki podpira to ključno zahtevo je podatkovna zbirka grafov. Uporaba decentralizirane podatkovne baze grafov tako zagotavlja odlične pogoje za interoperabilnost, zmogljivost in razpoložljivost. Z uporabo prstnega odtisa grafa se omogoča oblikovanje različic, kjer se zagotovi verodostojnost s primerjavo nespremenljivih prstnih odtisov v verigi blokov. [18]

Omrežna plast ODN

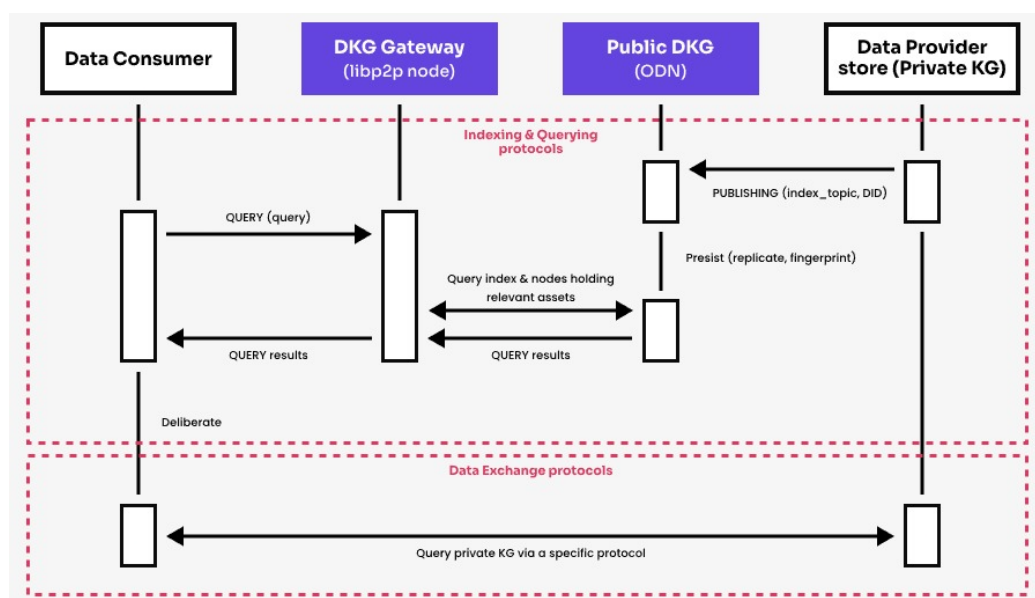
Omrežna plast se ukvarja z upravljanjem in dostopnostjo podatkov za podatkovno plast. Omrežno plast sestavljajo vozlišča, ki hranijo svojo kopijo decentralizirane zbirke grafov. Do podatkov se lahko dostopa z uporabo vmesnika za izmenjavo podatkov. Omrežje razlikuje med dvema tipoma vozlišč, glede na to, ali podatke ustvarjajo ali jih shranjujejo. Vozlišča za ustvarjanje podatkov so odgovorna za vnašanje podatkov in podvojitvi na določeno število vozlišč za shranjevanje podatkov, poleg tega pa si podatke tudi shranijo v lokalno bazo. Sistem na ta način poskuša zagotoviti, da je razmerje med vozlišči, ki shranjuje podatke in vozlišči, ki podatke objavljajo vedno večje od 1. S tem se minimizira možnost goljufij na dogovoru o podatku znotraj dobavne verige. [18]

3.3.2 OriginTrail DKG

Grafi znanja predstavljajo način ustvarjanja konteksta s pomočjo semantičnih metapodatkov. Združujejo značilnosti podatkovnih baz, saj je nad podatki mogoče izvajati strukturirane poizvedbe, grafov, ker jih je mogoče analizirati in baz znanja, kajti imajo formalno semantiko za interakcijo s podatki in sklepanje o novih dejstvih. Podatki v grafih znanj so predstavljeni v obliki trojčkov RDF. Skupaj s tehnologijo OWL omogočajo predstavitev različnih vrst podatkov. So učinkoviti in interoperabilni, saj omogočajo upravljanje z več milijardami povezanih podatkov s tehnologijo SPARQL. Standardizirani so s strani W3C skupnosti.

DKG so decentralizirani grafi znanja brez dovoljenj, ki se izvajajo na vozliščih DKG. To pomeni, da vsako vozlišče znotraj omrežja ODN hrani kopijo podatkov znotraj grafa znanja. Vsa sredstva, indeksirana v DKG, imajo preverljivo identiteto, semantično strukturo in lastništvo, kar omogoča odkrivanje, preverljivost in poizvedovanje po sredstvih. [19]

Uporabljajo se enotni identifikatorji (UAI) in enotni lokatorji sredstev (UAL), ki so razširitve sheme URI/URL. Vključujeta priporočila W3C glede decentraliziranih identifikatorjev (DID). S tem je dosežena preverljiva identiteta, semantična struktura in lastništvo za vsa sredstva znotraj DKG. Decentraliziran graf znanja omogoča poizvedovanje s pomočjo poizvedovalnih jezikov kot sta SPARQL in GraphQL. Osnovni tokovi interakcij so prikazani na sliki 3.2.



Slika 3.2: Osnovni tokovi uporabe decentraliziranih grafov znanja [5]

Vloge

Znotraj omrežja z OriginTrail DKG in uporabnostnim žetonom TRAC sodelujejo vloge: [19]

- **Izdajatelji sredstev** - ustvarjajo in posodabljaajo entitete znotraj DKG, indeksirajo z določenimi ključnimi besedami ter uporabljajo uporabnostni žeton TRAC za plačilo imetnikom vozlišč v zameno za hranjenje in zagotavljanje trajnosti sredstev.
- **Potrošniki sredstev** - ustvarjajo poizvedbe po sredstvih. V primeru podatkov, shranjenih na privatnih vozliščih lahko z uporabo uporabnostnega žetona TRAC dostop do podatkov tudi kupijo. V določenih primerih lahko prav tako kupijo lastništvo nad sredstvom.
- **Imetniki vozlišč DKG** - upravljajo vozlišča OriginTrail DKG, ki gostijo omrežje brez dovoljenj. Za svoje storitve so plačani v obliki žetonov TRAC.
- **Imetniki žetonov TRAC** - v zameno za del nagrad prenesejo žetone TRAC na vozlišča DKG.

3.3.3 TRAC uporabnostni žeton

TRAC je vnaprej rudarjenj ERC-20 neinflacijski uporabnostni žeton z maksimalno oskrbo 500 milijonov žetonov, kar pomeni, da se količina žetona ne bo povečala s časom. Je ključen za delovanje DKG, saj je potreben za izvajanje operacij, kot je objavljjanje podatkov. Uporablja se za: [20]

- **Sodelovanje znotraj ekosistema OriginTrail**, kjer morajo v vozliščih biti vloženi žetoni TRAC. Več žetonov v vozliščih pomeni da se lahko shranjuje in ustvarja več podatkovnih nalog.
- **Objavo in spreminjanje podatkov**, kjer izdajatelji sredstev plačajo vozlišča za opravljene storitve. Višina plačila je odvisna od parametrov, kot sta dolgotrajnost in velikost podatkov.
- **Jamstvo imetnikov podatkov**, kjer se z namenom preprečitve spreminjanja podatkov in obljube hrambe podatkov za določeno časovno obdobje žetone TRAC zakleni za čas trajanja podatkovne naloge. Po preteklem obdobju je vozlišče nato za svojo storitev plačano, ob tem pa se vložene žetone vrni.
- **Vlaganje žetonov** z namenom uvrstitve sredstva na višje mesto pri rezultatih iskanja za določeno ključno besedo
- **Stimulacijo znanja** s pomočjo podatkovnega ekosistema, ki izdajateljem oziroma ustvarjalcem sredstev omogoča monetizacijo preko prodaje podatkov na odprtem trgu.

3.3.4 Primeri uporabe

Tehnologija OriginTrail se uporablja pri izmenjavi informacij o potrdilih podjetij, posameznikov ali izdelkov. Uporabljajo jo tudi svetovna farmacevtska podjetja za povečanje razpoložljivosti, kakovosti in sledljivosti zdravil. Predvsem pa se uporablja pri sledljivosti delov dobavne verige, kjer je eden izmed primerov Švicarska zvezna železnica, ki uporablja tehnologijo za pregled in nadzor nad uporabljenimi železniškimi tirnimi materiali.

3.4 Ethereum omrežje in platforma

Omrežje Ethereum je omrežje enakovrednih (P2P), kjer so si vozlišča med seboj enakovredna in so hkrati strežniki in odjemalci drug za drugega. Vozlišča izpostavljajo skupni vmesnik in medsebojno komunicirajo preko P2P protokola Wire. Vozlišča Ethereum imajo možnost izvajanja aplikacijske kode, ki se nahaja znotraj blokov v verigi blokov. Zaradi te funkcionalnosti lahko platforme, kot je Ethereum, označimo kot programabilne verige blokov. Veriga blokov Ethereum bo v okviru diplomske naloge uporabljena za implementacijo protokola OriginTrail. [13]

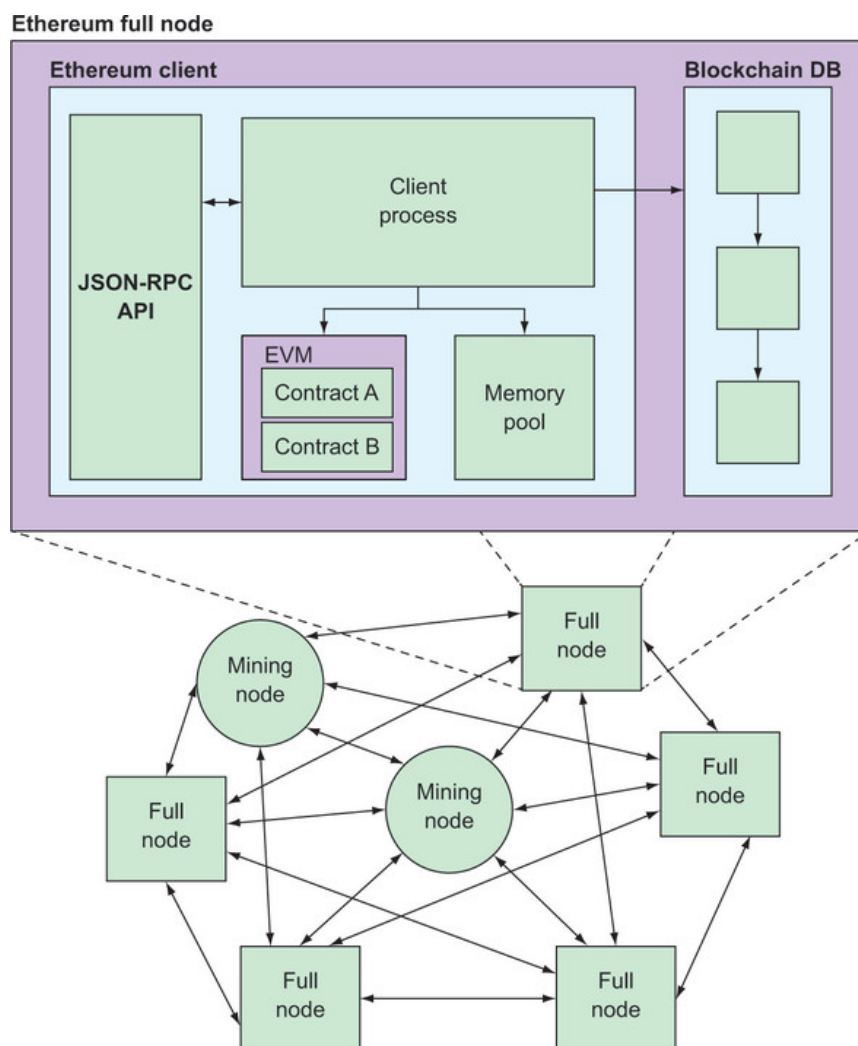
3.4.1 Vozlišče

V splošnem lahko vozlišča razdelimo na dve vrsti in sicer na:

- **Polna vozlišča**, ki hranijo lokalno kopijo verige blokov in jo lahko dodelajo z novim blokom, ki ga prejmejo od ostalih vozlišč
- **Rudarska vozlišča**, ki izvajajo transakcije in so nagrajena v kriptovaluti Ether.

Vsako vozlišče omrežja pa kot nakazuje slika 3.3 vsebuje:

- **Izvajalno okolje**, sestavljeno iz virtualnega stroja EVM, pomnilniškega bazena, odjemalniškega procesa in vmesnika JSON-RPC
- **Bazo verige blokov**, ki hrani podatke o transakcijah in bajtno kodo EVM pametnih pogodb, nameščenih v omrežju



Slika 3.3: Struktura polnega vozlišča Ethereum [21]

Poglavje 4

Načrtovanje decentraliziranega dnevnika za stavbe

Projekti v domeni gradbeništva se običajno razlikujejo glede svojih specifičnih zahtev, predvsem zaradi raznolikosti uporabljenih gradbenih procesov. Zato je potrebno doseči sledljivost celotnega procesa in preglednost od začetka do konca. Pri tem bi se lahko to funkcionalnost implementiralo s pomočjo pametnih aplikacij, ki bi tako pri obdelavi osebnih in industrijskih podatkov prispevale k večji stopnji zaupanja celotnemu procesu. Decentralizirane aplikacije oziroma vsebovane pametne pogodbe bi se tako lahko uporabljale na nivoju upravljanja poslovnih procesov, sledljivosti dobavne verige in prenosa lastništva gradbenih materialov.

Dandanes se informacije glede trajnostne gradnje med različnimi deležniki ne izmenjujejo zlahka, ob sami gradnji pa se tudi zanemarja celovit pristop. Razlogov je več, med ključnimi predvsem omejenost in razpršenost informacij o gradbenih proizvodih in storitvah, ki so zapisane v različnih oblikah ter shranjene v raznolikih bazah podatkov. Zaradi neenotnosti formatov dokumentov so zato potrebni mehanizmi za pretvorbo. Tu se prav tako pojavljajo pomanjkanja podrobnosti in neaktualnosti informacij. [12]

Openly share	Yes	Maybe	No
Product and service offers	60	6	2
References of projects, products, services	41	15	12
Quality certificates	49	16	3
Reviews of products or practices	21	45	2
Opinions, comments, ratings and taggings	40	26	2
Free advice on projects, e.g. "how to do" instructions	12	27	29
Detailed information	11	16	41

Tabela 4.1: Pripravljenost o deljenju informacij ponudnikov storitev in izdelkov [12]

Glede na izvedeno anketo o pripravljenosti za odprto izmenjavo informacij, prikazano na sliki 4.1, lahko razberemo, da so ponudniki storitev in izdelkov pripravljeni deliti naslednje stvari:

- Informacije o različnih izdelkih in storitev z namenom primerjanja cen ter kakovosti oziroma učinkovitosti
- Svoja najboljša priporočila in strokovna mnenja
- Pridobljene certifikate kakovosti

Deležniki in drugi sodelujoči si želijo bolj organiziran in učinkovitejši dostop do informacij o trajnostni gradnji. Tako lahko izpostavimo nekatere ključne lastnosti o aplikacijskih in sistemskih zahtevah načrtovane aplikacije, med katere lahko uvrstimo:

1. Možnost objave informacij o izdelkih in storitvah
2. Možnost objave strokovnega mnenja
3. Funkcionalnost prostega iskanja
4. Možnost primerjave sorodnih izdelkov, nadgradnja na temo trajnostnega kriterija
5. Analiziranje parametrov trajnosti stavbe

4.1 Doktorska disertacija Matije Königa

Ideja decentraliziranega dnevnika stavb delno temelji na doktorski disertaciji Matije Königa. Vsebina disertacije se osredotoča na področje trajnostne gradnje. Obravnava pomanjkljivosti dostopnosti informacij glede storitev znotraj posameznih procesov prenove stavb, kjer je lahko iskanje časovno neučinkovito in neorganizirano. Analizira stanje zgradb, kjer rezultati nakazujejo potrebo po prenovi večjega deleža. Prav tako analizira pripravljenost deležnikov do deljenja informacij glede proizvodov, storitev, mnenj, ocen in drugih informacij o projektih.

Opisani so tudi prispevki gradbeništva na področju trajnostnega razvoja, zgodovinsko uporabljeni in najbolj razširjeni pristopi ocenjevanja trajnostnega vrednotenja ter možne nadgradnje in ukrepi za energetske izboljšanje trajnosti stavb. Podrobno sta predstavljena razvoj in arhitektura prototipa sistema delotokov, ki deluje kot zaporedje povezanih storitev in nudi podporo tistim, ki se ukvarjajo s trajnostno gradnjo. Za razvoj informacijskega sistema so bile uporabljene semantične tehnologije. [11]

4.2 Zasnova decentraliziranega dnevnika

Decentraliziran dnevnik bo razvit s pomočjo informacijskega sistema, ustvarjenega v okviru doktorske disertacije, tehnologij semantičnega spleta, tehnologije verige blokov in ekosistema OriginTrail. Rešitev bo vsebovala funkcionalnosti za področje bolj energetske učinkovite in trajnostne gradnje, pri tem pa bo poudarek na decentraliziranem delovanju. Temelj bosta predstavljala tudi sledljivost in pregled nad že opravljenimi storitvami ter prav tako preverljivost storitev in izdelkov.

Eden izmed glavnih ciljev decentraliziranega dnevnika stavb bo implementacija nekaj različnih primerov uporabe decentraliziranih grafov znanj s pomočjo OriginTrail decentraliziranega omrežja vozlišč. Aplikacija bo podpirala več

načinov uporabe, in sicer jo bodo lahko uporabljali proizvajalci gradbenih materialov, izvajalci storitev, povpraševalci po storitvah za prenovu izbranih konstrukcijskih sklopov in gosti, ki bodo lahko po zbranih podatkih brskali. Uporabnikom bo kot za osnovo omogočala obdelavo nad podatki, izdelki in storitvami glede prenove streh, oken, tal in sten.

Sama struktura podatkov bo zasnovana s pomočjo doktorata in dosežena z uporabo semantičnih tehnologij. Za boljšo uporabniško izkušnjo bo pri povpraševanju in ustvarjanju podatkov dodana enostavna navigacija do zelenih izdelkov ali storitev. Energetska učinkovitost in trajnostna gradnja bo vključena na način analize trenutnega in končnega stanja podatkov izbrane stavbe ter algoritmov in smernic za takšno gradnjo.

4.2.1 Transparentnost in sledljivost

Glavni funkcionalnosti, ki bosta nadgrajevali dosedanje rešitve na temo iskanja in povpraševanja po storitvah in izdelkih za prenovu stavbnih konstrukcij, sta transparentnost in sledljivost. Transparentnost oziroma preglednost bo vgrajena kot pregled do sedaj opravljenih storitev nad določeno zgradbo. Sledljivost bo omogočila uporabnikom pregled nad izvorom določenega materiala. Poleg tega pa se z uporabo tehnologije veriženja blokov zagotovita še kredibilnost in preverljivost zapisov.

4.2.2 Preverljive poverilnice in identiteta

S pomočjo tehnologija veriženja blokov oziroma preverljivih poverilnic (VC) znotraj tehnologije OriginTrail se bo dosegla preverljiva digitalna identiteta (DID) izvajalcev in proizvajalcev. S tem uporabnikom, ki jih zanimajo storitve ali izdelki, ne bo potrebno zaupati ponudniku glede na njihove ocene, ki se lahko enostavno ponaredijo. Zaupati bo moral samo tehnologiji preverljivih poverilnic oziroma zapisu znotraj nespremenljivega bloka v verigi blokov.

4.2.3 Monetizacija

Deljenje znanja in podatkov se bo spodbujalo tako, da bo določen uporabnik lahko svoje podatke, kot so povpraševanja in zgodovine opravljenih storitev nad določeno stavbo, zaklenil. Za to se bo uporabila tehnologija OriginTrail, ki omogoča obstoj privatnih vozlišč DKG. Za dostop do podatkov, ki so na voljo na privatnih vozliščih DKG, bo potrebno plačati določeno število uporabnostnih žetonov TRAC.

4.3 Uporaba tehnologij in orodij

Skozi celotni razvoj aplikacije decentraliziranega dnevnika stavb je bilo uporabljenih več različnih tehnologij in orodij za lažji in hitrejši razvoj. Vključenih je veliko knjižnic, ki glede na celotno aplikacijo ne nosijo velikega vpliva. Zaradi tega je veliko izmed teh tehnologij oziroma orodij izpuščenih in so v nadaljevanju opisane le tiste, brez katerih izdelana aplikacija ne bi delovala.

MEAN - Sklad MEAN je okvir, ki temelji na JavaScriptu, za razvoj spletnih aplikacij. Poimenovan je po začetnicah tehnologij, ki plasti sklada sestavljajo.

MongoDB je dokumentno usmerjena baza podatkov NoSQL, uporablja pa se shranjevanje podatkov v binarnem zapisu objekta JSON - BSON. Vsak zapis je predstavljen kot dokument in je identificiran s ključem. Vrednosti predstavljajo podatki, ki so vsebovani znotraj dokumenta. MongoDB podpira tudi več-dokumentne transakcije ACID za primere, ki zahtevajo integriteto in varnost.[15]

Express.js služi kot zaledno spletno ogrodje okolja Node.js za gradnjo spletnih aplikacij in aplikacijskih programskih vmesnikov. Konfigurira vmesna orodja oziroma funkcije, ki uporabljajo objekte zahteve, odgovora in pokličejo naslednjo vmesno programsko opremo v nizu. Omogoča tudi uporabo sej in vsebuje preprost vmesnik za usmerjanje, kjer zahtevo preusmeri v tisti segment kode, ki za to zahtevo posluša.[15]

Angular je ogrodje Javascript, napisano v programskem jeziku TypeScript. Temelji na komponentah in se uporablja za izdelavo robustnih enostranskih spletnih aplikacij, ki se izvajajo na strani odjemalca - v brskalniku.[22]

Node.js je odprtokodno izvajalno okolje JavaScript in izvaja kodo JavaScript zunaj spletnega brskalnika. Temelji na asinhronskem modelu dogodkov, ki je zasnovan za razvoj skalabilnih aplikacij. Deluje enonitsko, vendar lahko služi več odjemalcem. Vsebuje različne module, ki uporabljajo zmožnost večprocesorskega okolja kot sta npr. ustvarjanje podrejenih procesov in souporaba vtičnic. [15]

jsonwebtoken - Knjižnica, ki uporablja storitve node-jws za uporabo spletnega žetona JWT. Kreira ga spletni strežnik, uporablja pa se za avtentikacijo in avtorizacijo uporabnikov za dostop do storitev in podatkov. Žeton JWT je šifriran, v osnovi pa ga sestavljajo glava, koristni podatki in podpis. Glava vsebuje tip žetona in informacijo o kodirnem algoritmu. V koristni vsebini so podatki, ki se prenašajo med klientom in strežnikom. Podpis vsebuje kodirano zgoščeno vrednost sestavljeno iz glave, podatkov ter skrivnosti, katero pozna samo izdajatelj žetona. Je izjemno varna tehnika za preverjanje identitete in dovoljenj, kljub temu da lahko žeton dekodira vsak. [9]

MySQL - Odprtokodni sistem za upravljanje podatkovnih baz. Deluje kot relacijska zbirka podatkov, kjer so podatki shranjeni v tabelah. Tabele so sestavljene iz stolpcev, ki služijo kot lastnosti tabel, in vrstic, ki predstavljajo zapise. Uporabniki lahko definirajo, manipulirajo z in poizvedujejo po podatkih s pomočjo programskega jezika SQL.[14]

GraphDB - Zbirka podatkov grafov in orodje za odkrivanje znanja. Uporablja se za shranjevanje in upravljanje semantičnih podatkov grafov znanj. Skladen je z RDF in SPARQL in je v uporabi pri različnih evropskih raziskovalnih projektih.[7]

Ganache - Je privatno okolje verige blokov Ethereum, ki se lahko uporablja za namestitev pogodb, razvijanje aplikacij in izvajanje testov. Omogoča komunikacijo s pametnimi pogodbami v svoji zasebni verigi blokov. Ob zagonu je uporabniku na voljo 100 računov, ki so pripravljeni za uporabo.[6]

dkg.js - Odprtokodna knjižnica za programski jezik Javascript. Omogoča interakcijo z vozlišči OriginTrail DKG. Knjižnica je v času pisanja diplomske naloge še v izdelavi, zaenkrat pa podpira osnovne funkcionalnosti, kot so dodajanje, posodabljanje, branje, iskanje, poizvedovanje in preverjanje celovitosti zahtevanih podatkov.[3]

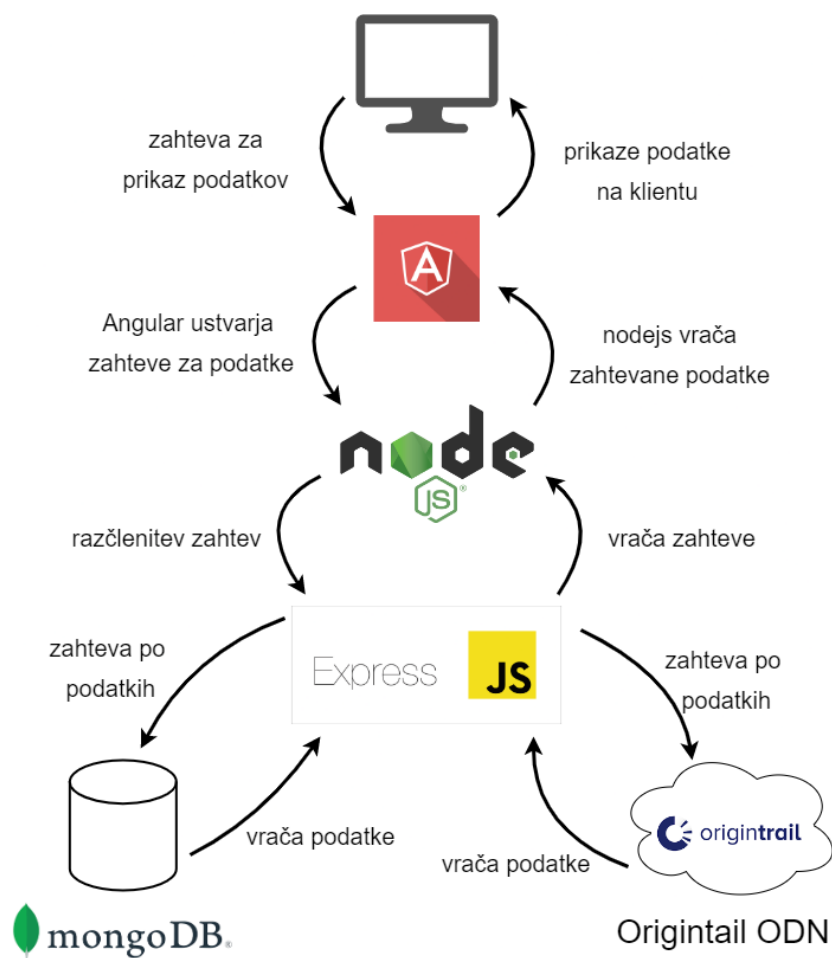
Poglavje 5

Izvedba decentraliziranega dnevnika za stavbe

Aplikacija decentraliziranega dnevnika stavb je realizirana s pomočjo sklada tehnologij MEAN. Poleg MongoDB je za podatkovni nivo določenih podatkov uporabljen OriginTrail DKG. Skeletna struktura je prikazana na sliki 5.1. Za dostop do podatkovnih nivojev se uporablja storitev REST API, ki je implementirana s pomočjo ogrodja Express, ki teče na izvajalnem okolju NodeJS.

5.1 Uporabniški vmesnik

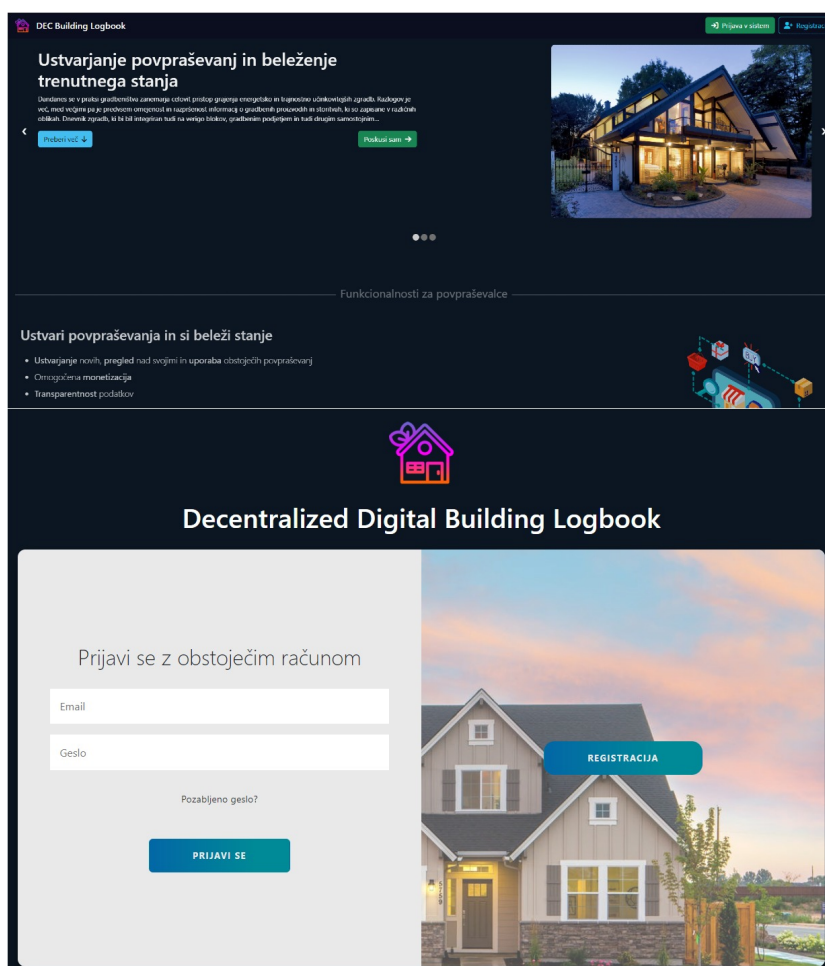
Uporabniški vmesnik je ustvarjen s pomočjo razvojne platforme Angular, ki je zgrajena na osnovi TypeScripta. Decentraliziran dnevnik stavb je zasnovana kot skalabilna spletna aplikacija, ki temelji na komponentah. Pri tem izkorišča zbirko integriranih knjižnic in nabor orodij za razvijalce, ki jih ponuja Angular. Uporabljeno je tudi CSS ogrodje Bootstrap za razvoj odzivnega in mobilnega spletnega mesta.



Slika 5.1: Osnovna struktura aplikacije z uporabo MEAN sklada tehnologij.

5.1.1 Osnovne strani

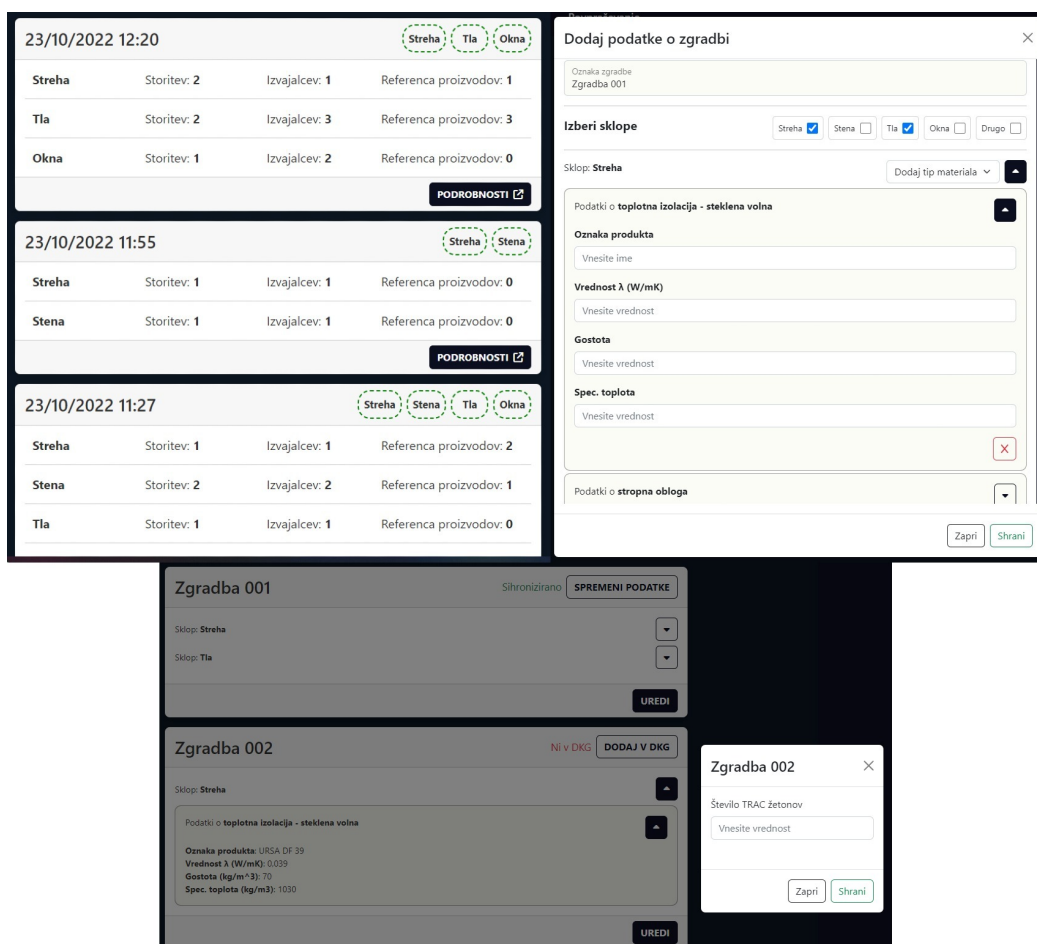
Osnovne strani, prikazane na sliki 5.2, predstavljajo domača stran, stran za prijavo in registracijo ter stran za nastavitvev. Domača stran predstavlja vstopno točko. Tu si lahko uporabniki preberejo o funkcionalnostih, ki jih decentraliziran dnevnik stavb ponuja ali pa v sistem vstopijo s prijavo oziroma registracijo. Pri registraciji imajo uporabniki na voljo izbiro med nivoji uporabe spletne aplikacije. V primeru, da bi uporabnik želel spremeniti uporabniški nivo, lahko to stori na strani z nastavitvami, kjer se v primeru degradacije določeni podatki arhivirajo.



Slika 5.2: Domača stran in stran za prijavo ter registracijo

5.1.2 Stran za hitri pregled

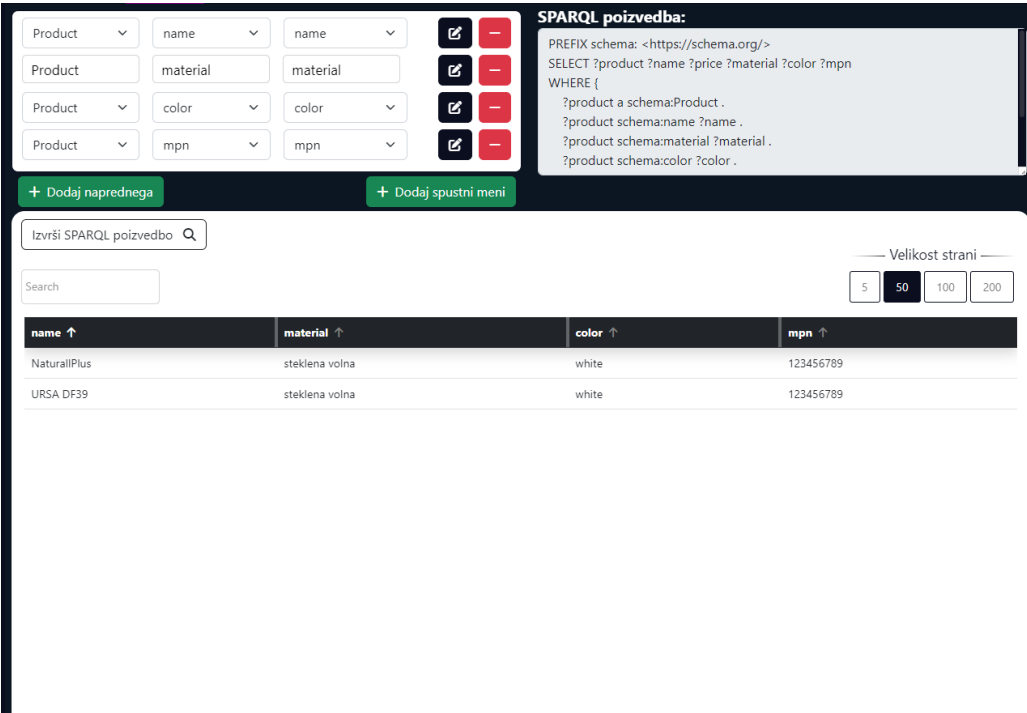
Stran za hitri pregled je namenjena navadnim prijavljenim uporabnikom. Sestavljena je iz pogleda za pregled poslanih povpraševanj ter pogleda za urejanje statusa podatkov obstoječih zgradb. Vsi podatki, predstavljeni na strani za pregled povpraševanj, so shranjeni v OriginTrail decentraliziranem grafu znanja. Uporabnik lahko status izbrane zgradbe in povezana povpraševanja javno objavi ali pa shrani v privatno vozlišče ter določi število žetonov, ki jih je potrebno plačati za dostop do teh podatkov. S tem je v aplikacijo decentraliziranega dnevnika stavb vpletena tudi monetizacija znanja. Primeri uporabe omenjenih funkcionalnosti so prikazani na sliki 5.3.



Slika 5.3: Strani za pregled povpraševanj in urejanje podatkov zgradb

5.1.3 Stran za iskanje

Na strani za iskanje lahko uporabniki prosto iščejo po vseh podatkih, shranjenih v OriginTrail DKG. Uporabniški vmesnik ponuja nekaj osnovnih izbir, po katerih lahko uporabnik išče, kot so izbira tipa zapisa ter vnaprej znane lastnosti, ki jih tip vsebuje. V primeru bolj kompleksnega iskanja pa lahko uporabnik sam zapiše lastnosti, po kateri želi poizvedovati, ter zanje vpiše vrednosti. Primer uporabniškega vmesnika iskanja je prikazan na sliki 5.4. Poizvedba se izvrši takrat, ko uporabnik pritisne na gumb Izvrši SPARQL poizvedbo.



The screenshot displays a search interface with a SPARQL query editor and a results table. The query editor includes a PREFIX section and a WHERE clause. The results table shows two rows of data with columns for name, material, color, and mpn.

SPARQL poizvedba:

```
PREFIX schema: <https://schema.org/>
SELECT ?product ?name ?price ?material ?color ?mpn
WHERE {
  ?product a schema:Product .
  ?product schema:name ?name .
  ?product schema:material ?material .
  ?product schema:color ?color .
```

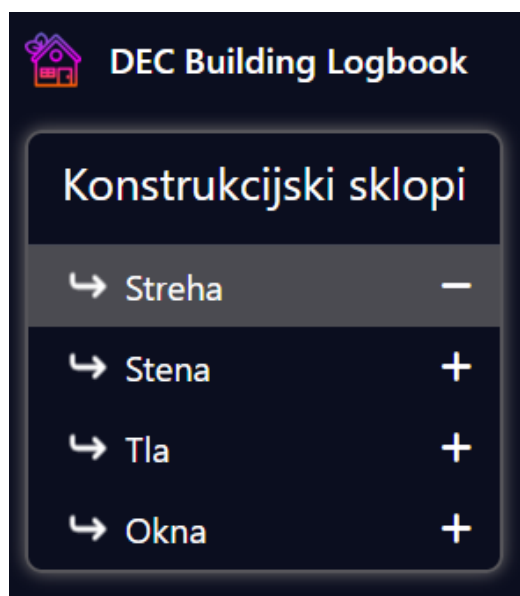
Results Table:

name ↑	material ↑	color ↑	mpn ↑
NaturallPlus	steklena volna	white	123456789
URSA DF39	steklena volna	white	123456789

Slika 5.4: Stran za iskanje s pomočjo SPARQL poizvedb

5.1.4 Stran za povpraševanje

Povpraševanje predstavlja eno izmed glavnih funkcionalnosti decentralizirane dnevnika stavb. Na voljo je vsem prijavljenim uporabnikom. Omogoča povpraševanje po storitvah za enega ali več konstrukcijskih sklopov. Konstrukcijske sklope, prikazane na sliki 5.5, predstavljajo streha, stene, tla in okna. Vsak izbrani sklop je sestavljen iz naslednjih korakov: Izbira storitev, izbira proizvodov in izbira izvajalcev. Uporabnik se lahko med koraki prosto premika nazaj, na naslednji korak pa se lahko premakne le v primeru da je trenutni korak pravilno izpolnjen.



Slika 5.5: Izbira konstrukcijskih sklopov

1. Izbira storitev

V tem koraku uporabnik izbira med ponujenimi storitvami za izbran konstrukcijski sklop. Storitve, ki so med seboj povezane, so združene v ukrepe. Ukrep služi kot način za lažjo predstavo povezanih storitev v primeru velike količine podatkov. Primer izbire ukrepov in storitev je prikazan na sliki 5.6. Korak je veljaven takrat, ko je izbran vsaj en ukrep in vsaj ena storitev za vsak izbran ukrep.

Streha

Izbira ukrepov in storitev Izbira izdelkov Izbira izvajalcev

Izbira ukrepov

Zamenjava toplotne izolacije Dodatna toplotna izolacija Prezračevanje podstrešja

Izbira storitev

Zamenjava toplotne izolacije storitve:

Zamenjava parne ovire Zamenjava sekundarne kritine

Dodatna toplotna izolacija storitve:

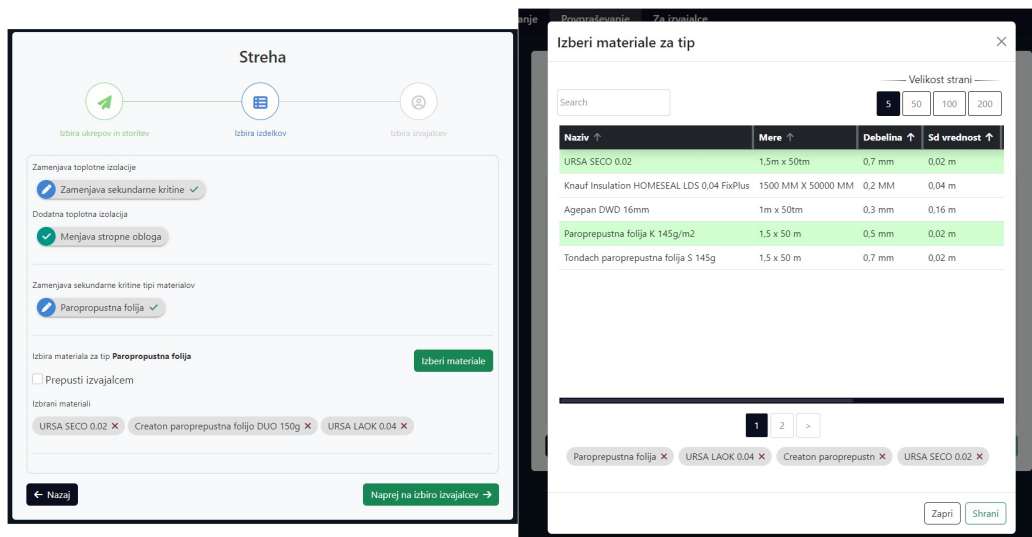
Inštalacija strešnih oken Menjava stropne obloga

Naprej na izbiro materialov →

Slika 5.6: Izbira ukrepov in storitev določenega konstrukcijskega sklopa

2. Izbira proizvodov

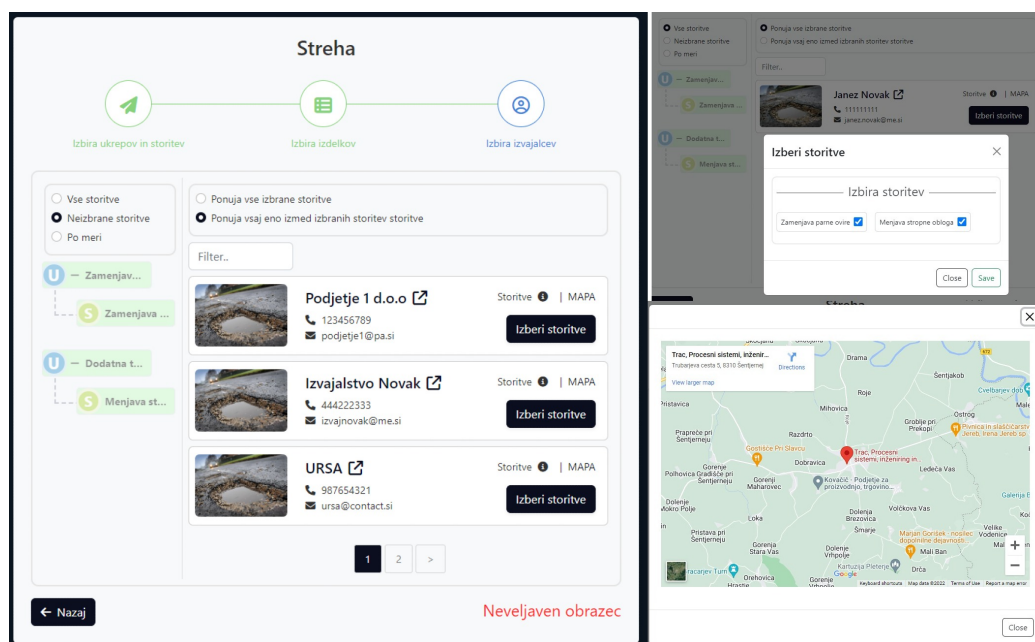
Po uspešni izbiri storitev so uporabniku za vsako izbrano storitev prikazani tipi materialov, potrebni v okviru storitve. Uporabnik lahko izbiro proizvodov prepusti izvajalcem, ali pa si jih iz seznama, prikazanega na sliki 5.7, izbere sam. Izbrani proizvodi niso končni in so izvajalcem samo v pomoč. Korak je veljaven, ko uporabnik pri vsaki storitvi izbiro proizvodov prepusti izvajalcem, ali pa jih izmed prikazanih nekaj izbere.



Slika 5.7: Izbira proizvodov za določen tip materiala izbrane storitve

3. Izbira izvajalcev

Na koncu mora uporabnik za veljaven obrazec konstrukcijskega sklopa izbrati izvajalce. Za boljšo uporabniško izkušnjo lahko filtrira izbrane storitve v 1. koraku. Filtracija je omogočena tudi nad storitvami, ki jih mora izvajalec ponujati. Storitve je veljavna, ko je zanjo izbran vsaj 1 izvajalec. Izbira storitev in prikaz zemljevida sta predstavljena na sliki 5.8.

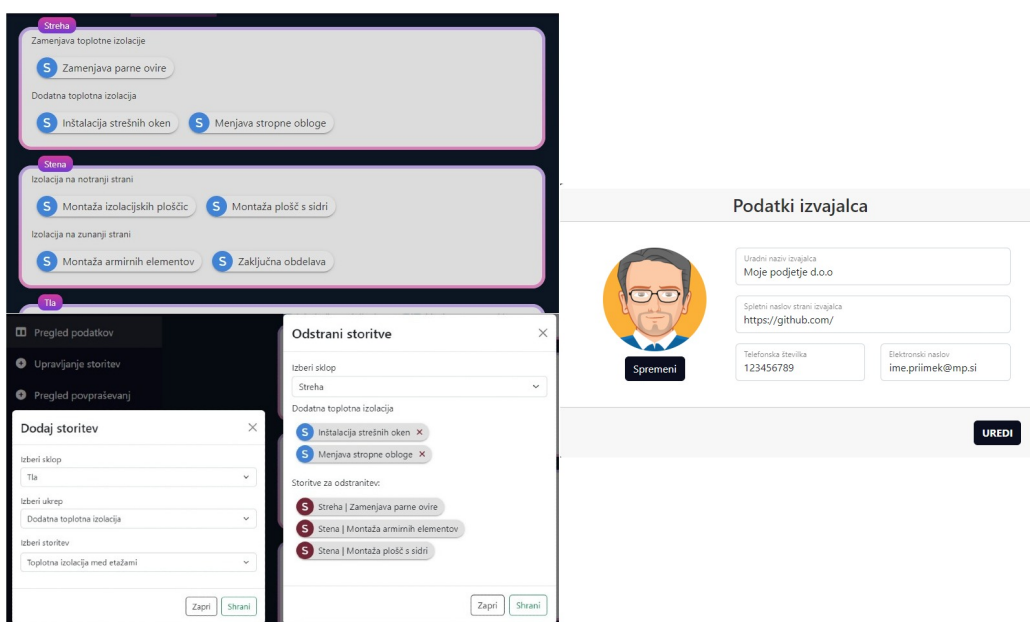


Slika 5.8: Izbira izvajalcev, validacija obrazca in zemljevid

5.1.5 Strani za izvajalce

Uporabniki, označeni kot izvajalci, imajo dostop do naslednjih dodatnih funkcionalnosti, prikazanih na sliki 5.9:

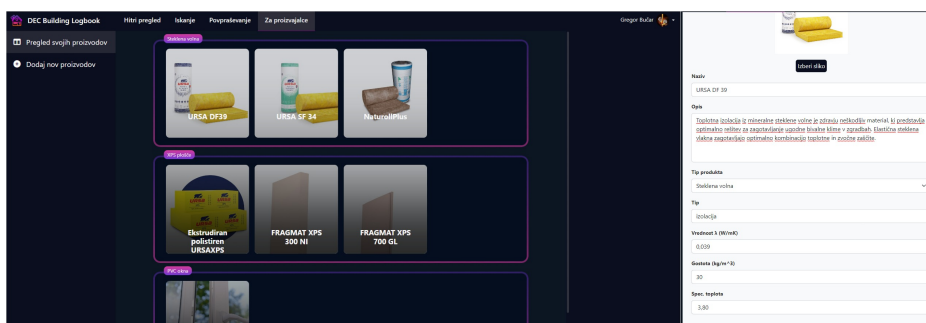
- **Pregled in urejanje podatkov izvajalca** - omogočeni ogled, dodajanje ali pa urejanje podatkov, s katerimi se izvajalci želijo predstavljati,
- **Pregled in upravljanje s storitvami** - možnost pregleda nad trenutnimi storitvami, ki jih izvajalec ponuja, dodajanje nove storitve za določen ukrep in konstrukcijski sklop, ter brisanje obstoječe ponujene storitve,
- **Pregled povpraševanj** - prikazana vsa povpraševanja, katerih del je tudi ta izvajalec.



Slika 5.9: Strani za pregled podatkov izvajalca in storitev ter uporabniški vmesniki za dodajanje oz. odstranjevanje storitev

5.1.6 Strani za proizvajalce

Dodatne funkcionalnosti uporabnikov, označenih kot proizvajalci, so pregled trenutnih proizvodov, ki se ponujajo, arhiviranje aktivnega proizvoda ter dodajanje novega proizvoda. Pogleda sta prikazana na sliki 5.10. Glede na tip produkta se razlikujejo tudi vnosna polja. S tem se podpirajo različni tipi produktov, ki pa so še vedno v osnovi produkt.



Slika 5.10: Strani za pregled in dodajanje proizvodov

5.2 Lokalno razvojno okolje OriginTrail DKG

OriginTrail je odprtokodna programska oprema, h kateri lahko prispeva vsakdo. Pri razvoju decentraliziranega dnevnika stavb je bila uporabljena verzija v6. Za delovanje potrebuje programsko opremo GraphDB, ki je semantična zbirka podatkov grafov in orodje za odkrivanje znanja, skladna z standardi W3C. Prvi pogoj je tudi nameščen sistem za upravljanje relacijskih baz podatkov MySQL, ki mora vsebovati prazno tabelo z imenom **operationaldb**. Z namenom izolacije razvojnega okolja je na nivoju verige blokov uporabljeno zasebno okolje verige blokov Ethereum - **Ganache**.

Po uspešni namestitvi in zagonu programskih oprem MySQL in GraphDB sledi zagon lokalne instance verige blokov Ganache. Pri tem je potrebna datoteka `.env`, ki vsebuje dostopno točko in privatni ključ.

Po uspešnem zagonu je na instanco Ganache potrebno uvesti potrebne pogodbe za delovanje vozlišč OriginTrail DKG. Sledita zagon vozlišča ki služi za ustvarjanje podatkov in zagon dovolj vozlišč za hranjenje podatkov. Pri zagonih, prikazanih na sliki 5.11, spreminjamo le konfiguracijsko datoteko.

```

[2022-12-11 21:39:30] INFO: All modules initialized!
[2022-12-11 21:39:30] INFO: Creating blockchain identity on network: ganache
[2022-12-11 21:39:30] INFO: Creating profile on network: ganache
[2022-12-11 21:39:30] INFO: ganache blockchain identity is 0xa789fc797d0fbd0e057f132748548f0269ca6d
[2022-12-11 21:39:30] INFO: Initializing http api router
[2022-12-11 21:39:30] INFO: Node listening on port: 8984
[2022-12-11 21:39:30] INFO: Initializing rpc router
[2022-12-11 21:39:30] INFO: Enabling network protocol: /store/1.0.0
[2022-12-11 21:39:30] INFO: Enabling network protocol: /get/1.0.0
[2022-12-11 21:39:30] INFO: Replay pending/started commands from the database.
[2022-12-11 21:39:30] INFO: Telemetry Injection Service initialized successfully
[2022-12-11 21:39:30] INFO: Node is up and running!
[2022-12-11 21:40:16] TRACE: Node QmXVh3stZBp8F27dK63j5GvRx1dm4VbXjdtij7qpY3BzMY connected to QmYf3dtqJnhuCPzEDTnmFYcStjXtRtXhGcUUmGHdg2gtj
[2022-12-11 21:40:16] TRACE: Node QmXVh3stZBp8F27dK63j5GvRx1dm4VbXjdtij7qpY3Bz

[2022-12-11 21:37:08] WARN: Cannot connect to Triple store (GraphDB), retry number: 2 /10. Retrying in 10 seconds.
[2022-12-11 21:36:44] WARN: Cannot connect to Triple store (GraphDB), retry number: 3 /10. Retrying in 10 seconds.
[2022-12-11 21:37:08] INFO: tripleStore module initialized with implementation: ot-graphdb
[2022-12-11 21:37:08] INFO: All modules initialized!
[2022-12-11 21:37:08] INFO: Creating blockchain identity on network: ganache
[2022-12-11 21:37:08] INFO: Creating profile on network: ganache
[2022-12-11 21:37:08] INFO: ganache blockchain identity is 0x168ae0554bc8b88c3D8d54c52372e8888e8e260
[2022-12-11 21:37:08] INFO: Replay pending/started commands from the database...
[2022-12-11 21:37:08] INFO: Telemetry Injection Service initialized successfully
[2022-12-11 21:37:08] INFO: Initializing http api and rpc router
[2022-12-11 21:37:08] INFO: Enabling network protocol: /store/1.0.1
[2022-12-11 21:37:08] INFO: Enabling network protocol: /get/1.0.0
[2022-12-11 21:37:08] INFO: Node listening on port: 8980
[2022-12-11 21:37:08] INFO: Routers initialized successfully
[2022-12-11 21:37:08] INFO: Node is up and running!

8800 GANACHE_TOKENS and 5000000000000 gTRAC.
[2022-12-11 21:38:26] INFO: blockchain module initialized with implementation: ganache
[2022-12-11 21:39:29] INFO: repository module initialized with implementation: sequelize-repository
[2022-12-11 21:38:28] INFO: All modules initialized!
[2022-12-11 21:38:28] INFO: Creating blockchain identity on network: ganache
[2022-12-11 21:38:28] INFO: Creating profile on network: ganache
[2022-12-11 21:38:28] INFO: ganache blockchain identity is 0xa9f2ce471Efe1F7390657695b2601F7C52978ebc
[2022-12-11 21:38:28] INFO: Initializing http api router
[2022-12-11 21:38:28] INFO: Node listening on port: 8981
[2022-12-11 21:38:28] INFO: Initializing rpc router
[2022-12-11 21:38:28] INFO: Enabling network protocol: /store/1.0.0
[2022-12-11 21:38:28] INFO: Enabling network protocol: /get/1.0.0
[2022-12-11 21:38:28] INFO: Replay pending/started commands from the database...
[2022-12-11 21:38:28] INFO: Telemetry Injection Service initialized successfully
[2022-12-11 21:38:28] INFO: Node is up and running!
[2022-12-11 21:39:06] TRACE: Node QwY285XeJbw2EQutl8Zdw7U5aoE02tjFfwkRmNScKdTP connected to QmYf3dtqJnhuCPzEDTnmFYcStjXtRtXhGcUUmGHdg2gtj
[2022-12-11 21:39:06] TRACE: Node QwY285XeJbw2EQutl8Zdw7U5aoE02tjFfwkRmNScKdTP connected to QmYf3dtqJnhuCPzEDTnmFYcStjXtRtXhGcUUmGHdg2gtj
[2022-12-11 21:39:06] TRACE: Node QwY285XeJbw2EQutl8Zdw7U5aoE02tjFfwkRmNScKdTP connected to QmYf3dtqJnhuCPzEDTnmFYcStjXtRtXhGcUUmGHdg2gtj

[2022-12-11 21:39:11] INFO: ganache blockchain identity is 0x8f26f57DeF244dc858813ac3488818911e8333
[2022-12-11 21:39:11] INFO: Initializing http api router
[2022-12-11 21:39:11] INFO: Node listening on port: 8983
[2022-12-11 21:39:11] INFO: Initializing rpc router
[2022-12-11 21:39:11] INFO: Enabling network protocol: /store/1.0.0
[2022-12-11 21:39:11] INFO: Enabling network protocol: /get/1.0.0
[2022-12-11 21:39:11] INFO: Replay pending/started commands from the database...
[2022-12-11 21:39:11] INFO: Telemetry Injection Service initialized successfully
[2022-12-11 21:39:11] INFO: Node is up and running!
[2022-12-11 21:39:56] TRACE: Node QmQvFFVXJSRkXWmoSCAbtzppjUj29Nz9hC8Gr7p61uPha connected to QmYf3dtqJnhuCPzEDTnmFYcStjXtRtXhGcUUmGHdg2gtj
[2022-12-11 21:39:56] TRACE: Node QmQvFFVXJSRkXWmoSCAbtzppjUj29Nz9hC8Gr7p61uPha connected to QmYf3dtqJnhuCPzEDTnmFYcStjXtRtXhGcUUmGHdg2gtj
[2022-12-11 21:40:26] TRACE: Node QmQvFFVXJSRkXWmoSCAbtzppjUj29Nz9hC8Gr7p61uPha connected to QmYf3dtqJnhuCPzEDTnmFYcStjXtRtXhGcUUmGHdg2gtj
[2022-12-11 21:40:26] TRACE: Node QmQvFFVXJSRkXWmoSCAbtzppjUj29Nz9hC8Gr7p61uPha connected to QmYf3dtqJnhuCPzEDTnmFYcStjXtRtXhGcUUmGHdg2gtj
[2022-12-11 21:40:26] TRACE: Node QmQvFFVXJSRkXWmoSCAbtzppjUj29Nz9hC8Gr7p61uPha connected to QmYf3dtqJnhuCPzEDTnmFYcStjXtRtXhGcUUmGHdg2gtj

```

Slika 5.11: Zagon vozlišč OriginTrail

5.3 Vmesnik API decentraliziranega dnevnika

Nudi storitve za izmenjavo podatkov uporabnikov, sklopov, ukrepov, storitev in materialov. Poleg tega prav tako upravlja podatke, shranjene na decentraliziranem grafu znanja OriginTrail. Spodnji odsek kode predstavlja modele mongoose za sklop, ukrep, storitev in material.

```
const sklopSchema = new mongoose.Schema({
  name: { type: String, required: true },
  ukrepi: { type: [ukrepSchema], required: true },
});

const ukrepSchema = new mongoose.Schema({
  name: { type: String, required: true },
  sklopId: { type: ObjectId, ref: "Sklop" },
  storitve: { type: [storitevSchema], required: true }
});

const storitevSchema = new mongoose.Schema({
  name: { type: String, required: true },
  ukrepId: { type: ObjectId, ref: "Ukrep" },
  materiali: { type: [materialSchema], required: true }
});

const materialSchema = new mongoose.Schema({
  name: { type: String, required: true },
  specPodatki: { type: [Mixed], required: true} // Posebnost
});
```

Schema za model material vsebuje lastnost **specPodatki**, ki kot ime napeljuje, opisuje lastnosti določenega materiala. Primer je Steklena volna z naslednjimi lastnostmi, kot so toplotna prevodnost λ , gostota, CO_2 odtis in specifična toplota c_m .

5.3.1 Avtentikacija uporabnika

Schema uporabnika, prikazana na spodnjem odseku kode, vsebuje osnovne podatke kot so ime, priimek in elektronski naslov. Za potrebe avtentikacije sta vključeni tudi lastnosti zgoščena vrednost (Hash) in naključna vrednost (Salt). Dodani sta tudi zastavici, ki opisujeta ali ima uporabnik tudi privilegije izvajalcev oziroma proizvajalcev.

V primeru, da je uporabnik označen kot izvajalec, mu pripada tudi naslov UAL, s katerim lahko pridobi podatke o izvajalcu iz OriginTrail decentraliziranega grafa podatkov. Podobno velja tudi za proizvajalca, le da mu v tem primeru pripada seznam naslovov produktov UAL.

```
const uporabnikSchema = new mongoose.Schema({
  name: { type: String, required: true },
  surname: { type: String, required: true },
  email: { type: String, unique: true, required: true },
  isContractor: { type: Boolean, default: false },
  isManufacturer: { type: Boolean, default: false },
  contractorUAL: { type: String },
  manufacturerProductUALs: { type: [String] },
  zgoscenaVrednost: { type: String, required: true },
  nakljucnaVrednost: { type: String, required: true },
});
```

Pri registraciji uporabnika se ob ustreznih podatkih generirata naključno geslo, iz gesla in generiranega naključnega gesla pa še zgoščena vrednost. Vrednosti se poleg posredovanih podatkov nato shranita v podatkovno bazo, uporabniku pa se nazaj vrne ustvarjenega uporabnika brez generiranih varnostnih vrednosti ter žeton JWT.

Pri prijavi, ki je prikazana v sledečem odseku kode, se iz obstoječe naključne vrednosti za posredovanega uporabnika in gesla preverita vrednosti generirane in obstoječe zgoščene vrednosti. V uspešnem primeru se uporabniku nazaj vrne generiran žeton JWT.

```
//Passport uporabljen kot vmesna programska oprema
//za preverjanje pristnosti
const login = (req, res) => {
  passport.authenticate("local", (err, uporabnik, info) => {
    if (err) return res.status(500).json(err);

    delete uporabnik.nakljucnaVrednost;
    delete uporabnik.zgoscenaVrednost;

    if (uporabnik) res.status(200).json(
      { uporabnik: uporabnik,
        zeton: uporabnik.generirajJwt()
      });
    else res.status(401).json(info);
  })(req, res);
```

Ko uporabnik uspešno pridobi žeton JWT, lahko izvaja tudi tiste funkcionalnosti, ki za delovanje potrebujejo žeton z veljavno vsebino. Primer uporabe je predstavljen v naslednjem odseku kode.

```
const { expressjwt: jwt } = require('express-jwt');
const authentication = jwt({
  secret: process.env.JWT_GESLO,
  userProperty: "payload",
  algorithms: ["HS256"],
});
const usersController = require("../controllers/users");

//Omejitev na avtentificirane uporabnike
//v primeru posodabljanja podatkov uporabnika
router.put("/user/:userId", authentication,
  usersController.updateUser);
```


5.3.2 Upravljanje s podatki OriginTrail DKG

Produkti, izvajalci in povpraševanja so z namenom transparentnosti ter sledljivosti shranjeni v decentraliziranih grafih znanja. Interakcija z OriginTrail DKG je omogočena in implementirana s pomočjo knjižnice `dkg.js`

Knjižnica za delovanje potrebuje vozlišče OriginTrail, ki je lahko oddaljeno ali pa lokalno. V spodnjem odseku kode je uporabljeno zagonsko vozlišče, ki deluje znotraj lokalnega razvojnega okolja na vratih 8900. Za namen razvoja je tudi uporabljena privatna veriga blokov Ganache, ter privatni in javni ključ izbrane denarnice za izvajanje operacij s pomočjo `dkg.js`.

```
const DKG = require('dkg.js');

const OT_NODE_HOSTNAME = "http://localhost"; //Ime gostitelja
const OT_NODE_PORT = "8900"; //Vrata
const PUBLIC_KEY = "0x2f2697b2a7BB4555687...";
const PRIVATE_KEY = "b1c53fd90d0172ff60f1...";

const blockchain = {
  name: "ganache",
  publicKey: PUBLIC_KEY,
  privateKey: PRIVATE_KEY,
};

const dkg = new DKG({
  endpoint: OT_NODE_HOSTNAME,
  port: OT_NODE_PORT,
  useSSL: false,
  loglevel: "trace",
});
```

Ustvarjanje in Posodabljanje

Za ustvarjanje novega zapisa v DKG uporabimo funkcijo `create`, ki kot parametre zahteva shemo s podatki ter opcije. Znotraj opcij lahko nastavimo ključne besede, vidnost, število žetonov in čas hranjenja zapisa. Potrebuje tudi informacije o podatkih denarnice, ki želimo lastništvo dodeliti in iz nje poslati zahtevana sredstva.

Za posodabljanje obstoječega zapisa pa poleg sheme s podatki in opcij potrebujemo še unikatni identifikator UAL. V primeru, da denarnica, ki smo jo posredovali, ni lastnik zapisa, funkcija podatkov ne posodobi. Pri ustvarjanju in posodabljanju entitet uporabniku nazaj vrnemo identifikator UAL in podatke ustvarjenega oziroma posodobljenega zapisa. Implementacija omenjenih funkcionalnosti je prikazana v spodnjem izseku kode.

```
const createAssetResult = await saveProduct(/*UAL*/, req.body);
...
//Za posodabljanje samo zamenjamo funkcijo create v update
//Pri tem pa še na koncu dodamo UAL
const res = await dkg.asset.create(
  assertion, {
    keywords: ["product"],
    visibility: "public",
    holdingTimeInYears: 1,
    blockchain: blockchain
  }/*UAL*/)
...
res.status(201).json({UAL: createAssetResult.UAL, ...asset});
```

Branje

Branje je najbolj enostavna operacija, saj je vse, kar potrebuje, identifikator UAL, s katerim nato pridobimo podatke o zapisu, identificiranem s podanim parametrom. Implementacija je prikazana v spodnjem izseku kode.

```
const UAL = req.params.UAL;
...
const asset = await readProduct(UAL);
...
async function readProduct(ual) {
  var result = await dkg.asset.get(ual);
  return result.assertion;
}
...
res.status(200).json(asset);
```

Iskanje

Funkcija search omogoča iskanje po ključnih besedah. Kot parameter ji podamo iskalni izraz, vrne pa zapise, katerih ključne besede vsebujejo podane besede v iskalnem izrazu. Implementacija je prikazana v spodnjem izseku kode.

```
const result = await searchForProducts();
...
//Iskanje po keywords -> vrne produkte oz. customQuery
async function searchForProducts(customQuery = undefined) {
  return await dkg.search(//Podatki znotraj itemListElement
    { query: customQuery ?? 'product',
      responseType: 'assertions' });
}
...
res.status(200).json(result.itemListElement);
```

Poizvedovanje

Poizvedovanje po DKG je omogočeno z uporabo poizvedovalnega jezika SPARQL. Rezultat izvedene poizvedbe je prenosni graf RDF. V spodnjem odseku kode je predstavljen primer implementirane funkcionalnosti. Za lažje razumevanje je uporabljen primer iz uradne dokumentacije knjižnice dkg.js. [5]

```
const result = await queryGraph(queryParams);
...
async function queryGraph(generatedQuery) {
  options = { /* query: generatedQuery*/
    query: `PREFIX schema: <http://schema.org/>
      CONSTRUCT { ?s ?p ?o }
      WHERE {
        GRAPH ?g {
          ?s ?p ?o .
          ?s schema:hasVisibility ?v
        }
      }`
  };
  return await dkg.query(options)
}
...
//Rezultati (n-quads) - ustrezajo predikatu schema:hasVisibility
//{{X}} = http://schema.org/, {{Date}} = ISO-8601 format
{ "status": "COMPLETED", "data": {
  "response": [
    "<did:dkg:43a..7435> <{{X}}hasDataHash> \"01904..c801\" .",
    "<did:dkg:43a..7435> <{{X}}hasIssuer> \"0xbd..646f\" .",
    "<did:dkg:43a..7435> <{{X}}hasSignature> \"0x13..b41b\" .",
    "<did:dkg:43a..7435> <{{X}}hasTimestamp> \"{{Date}}\" .",
    "<did:dkg:43a..7435> <{{X}}hasType> \"default\" ."]}]
}
...

```

Monetizacija znanja

Monetizacija znanja je enostavno omogočena z zamenjavo opcije vidnost in dodatne lastnosti za število žetonov, katere je potrebno plačati enkratni zahtevan znesek za pridobitev podatkov zapisa z denarnic, katerih naslovi so različni oh naslova lastnika. Funkcionalnost je na voljo vsem uporabnikom, kjer lahko podatke povpraševanj in zgodovine opravljenih storitev nad določeno stavbo delijo z drugimi. Primer uporabe opisane funkcionalnosti je predstavljen v spodnjem izseku kode.

```
const createAssetResult = await save(req.body);
...
//Zamenjamo vidnost, določimo število žetonov
const res = await dkg.asset.create(assertion, {
  ...
  visibility: "private",
  tokenAmount: 5,
  ...
})
...
res.status(201).json({UAL: createAssetResult.UAL, ...asset});
```


Poglavje 6

Sklepne ugotovitve

V diplomskem delu smo obravnavali tematiko bolj učinkovitega pristopa h grajenju trajnostno učinkovitejših zgradb s pomočjo tehnologij semantičnega in decentraliziranega spleta. Predstavljena sta načrt in implementacija celotnega sistema, imenovanega decentraliziran dnevnik stavb. Pri izdelavi prototipa smo se osredotočili na uporabo decentraliziranega grafa znanja, OriginTrail DKG. Prototip podpira tri vrste uporabnikov, in sicer navadnega uporabnika, izvajalca in proizvajalca. Pri tem so v uporabi varnostni mehanizmi, ki preprečujejo zlonamerno uporabo. Podatke, za katere želimo zagotoviti preverljivost in transparentnost, se shranjuje v decentraliziranih grafih znanja. Za podatke uporabnikov in drugih redko spremenljivih podatkov pa se uporablja centralizirana podatkovna baza.

Aplikacija je enostavno nadgradljiva s stališča funkcionalnosti in skalabilnosti, saj je bila ustvarjena s pomočjo sklada MEAN in sodobnega pristopa spletnega programiranja. Na nivoju verige blokov je bilo uporabljeno okolje Ganache, ki je privatno okolje verige blokov Ethereum. Delovanje decentralizirane aplikacije je bilo testirano v lokalnem razvojnem okolju. V izvedenih testnih scenarijih je bila precej opazna predvsem hitrost delovanja. Ta se je izmenjala za zagotavljanje transparentnosti, sledljivosti in preverljivosti entitet. Implementirana je tudi monetizacija znanja podatkov obstoječih stavb.

V okviru načrtovanja decentraliziranega dnevnika stavb se sklicujemo na pripravljenost o deljenju informacij glede storitev in izdelkov. Ob tem bi bilo dobro izvesti dodatno anketiranje glede uporabe tehnologij decentraliziranega spleta za izdelavo aplikacije, ki bi omenjene informacije obdelovala. Ena izmed pomanjkljivosti je tudi razvoj in testiranje samo v lokalnem razvojnem okolju. Za bolj realistične rezultate bi lahko decentralizirano aplikacijo uporabili v testnem okolju "testnet" in nato še v produkcijskem okolju "mainnet" z veliko večjim naborom podatkov, kasneje pa bi poleg tega lahko tudi preverili delovanje na drugih, podprtih verigah blokov.

Izdelan prototip decentraliziranega dnevnika stavb zadostuje ciljem diplomske naloge, ki so bili zadani. Poleg uporabe dnevnika za beleženje stanj stavb lahko služi kot primer uporabe tehnologije decentraliziranih grafov, ki uporabljajo sodobne tehnologije spleta 3.0. Prav tako je lahko prototip uporabljen kot izhodišče za nadaljnje raziskovanje in implementacijo na drugih področjih. Pri tem bi lahko uvedli nekaj izboljšav, in sicer bi prva bila eliminacija ročnega vnašanja podatkov storitev, ki jih določen izvajalec ponuja. Namesto tega bi se uvedel uvoz podatkov iz obstoječih virov, kjer bi nato pridobljene podatke JSON preoblikovali v shematični zapis, primeren za dodajanje na decentraliziran graf znanja. Drug način bi lahko bil strganje delcev kode JSON-LD iz podane spletne strani, kjer bi pridobili shemo s podatki. Naslednja izboljšava je registracija oziroma prijava v sistem preko avtentikacijske storitve, kjer bi se preverjala identiteta uporabnika.

Izboljšali bi lahko uporabniški vmesnik za prosto iskanje po DKG. Ustvarili bi uporabniku prijazen generator poizvedb SPARQL. Znotraj same aplikacije bi lahko uvedli dodatni funkcionalnosti ocenjevanja in komentiranja določenih produktov, izvajalcev in njihovih storitev. S tem bi povečali interakcijo z decentralizirano aplikacijo, produktom in storitvam dodali novo vrednost v obliki povprečja ocen in mnenj ter posledično izboljšali uporabniško izkušnjo. Enako bi lahko tudi storili s plačljivimi podatki obstoječih stavb.

Celotna literatura

- [1] Tim Berners-Lee, James Hendler in Ora Lassila. “The semantic web”. V: *Scientific american* 284.5 (2001), str. 34–43.
- [2] Christian Bizer. “The emerging web of linked data”. V: *IEEE intelligent systems* 24.5 (2009), str. 87–92.
- [3] *DKG Client*. URL: <https://github.com/OriginTrail/dkg.js> (pridobljeno 3. 8. 2022).
- [4] Cory Doctorow. “Metacrap: Putting the torch to seven straw-men of the meta-utopia”. V: *Retrieved June 10* (2001), str. 2003.
- [5] *OriginTrail*. URL: <https://origintrail.io/> (pridobljeno 19. 3. 2022).
- [6] *Ganache*. URL: <https://trufflesuite.com/ganache/> (pridobljeno 3. 8. 2022).
- [7] *GraphDB*. URL: <https://graphdb.ontotext.com/documentation/10.1/> (pridobljeno 3. 8. 2022).
- [8] David Johnston in sod. “TheGeneralTheoryofDecen-tralizedApplications, DApps”. V: (2014).
- [9] *jsonwebtoken*. URL: <https://www.npmjs.com/package/jsonwebtoken?activeTab=readme> (pridobljeno 3. 8. 2022).
- [10] Abdul Ghaffar Khan in sod. “A journey of WEB and Blockchain towards the Industry 4.0: An Overview”. V: *2019 International Conference on Innovative Computing (ICIC)*. IEEE. 2019, str. 1–7.

- [11] Matija König. “Sistem delotokov kot informacijska podpora podjetjem, dejavnim na področju trajnostne gradnje”. Doktorsko delo. FGG - Fakulteta za gradbeništvo in geodezijo, 2014.
- [12] Matija König, Jaka Dirnbek in Vlado Stankovski. “Architecture of an open knowledge base for sustainable buildings based on Linked Data technologies”. V: *Automation in construction* 35 (2013), str. 542–550.
- [13] William Metcalfe in sod. “Ethereum, smart contracts, DApps”. V: *Blockchain and Crypt Currency* (2020), str. 77.
- [14] *MySQL*. URL: <https://www.mysql.com/> (pridobljeno 3. 8. 2022).
- [15] N Nirgudkar in P Singh. “The MEAN stack”. V: *International Research Journal of Engineering and Technology* (2017), str. 2395–56.
- [16] Leo Sauermann, Richard Cyganiak in Max Völkel. “Cool URIs for the semantic web”. V: (2007).
- [17] Manu Sporny in sod. “JSON-LD 1.0”. V: *W3C recommendation* 16 (2014), str. 41.
- [18] *First purpose built protocol for supply chains based on blockchain*. URL: <https://origintrail.io/storage/documents/OriginTrail-White-Paper.pdf> (pridobljeno 3. 8. 2022).
- [19] *OriginTrail Ecosystem White Paper 2.0, Layer 2: OriginTrail DKG*. URL: <https://parachain.origintrail.io/whitepaper?section=layer-2-origintrail-dkg> (pridobljeno 3. 8. 2022).
- [20] *OriginTrail Explained*. URL: <https://origintrailexplained.info/> (pridobljeno 3. 8. 2022).
- [21] *Chapter 2. Understanding the blockchain*. URL: <https://livebook.manning.com/book/building-ethereum-dapps/chapter-2/> (pridobljeno 3. 8. 2022).
- [22] Sufyan bin Uzayr, Nicholas Cloud in Tim Ambler. “Angular”. V: *JavaScript Frameworks for Modern Web Development*. Springer, 2019, str. 209–223.