Web3 platforma za donacije

Luk, Martin

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Diplomski studij računarstva

Diplomski rad

Web3 platforma za donacije

Rijeka, srpanj 2023.

Martin Luk
0069082648

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Diplomski studij računarstva

Diplomski rad

Web3 platforma za donacije

Mentor: prof. dr. sc. Kristijan Lenac

Rijeka, srpanj 2023.

Martin Luk
0069082648

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA DIPLOMSKE ISPITE

Rijeka, 20. ožujka 2023.

Zavod: **Zavod za računarstvo**
Predmet: **Napredni operacijski sustavi**
Polje: **2.09 Računarstvo**

ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Martin Luk (0069082648)**
Studij: Sveučilišni diplomski studij računarstva
Modul: Računalni sustavi

Zadatak: **Web3 platforma za donacije / Web3 donation platform**

Opis zadatka:

Razviti Web3 platformu za humanitarne donacije. Decentralizirana Web3 platforma temeljena na pametnim ugovorima i blockchain tehnologiji treba omogućavati transparentno i uključivo zaprimanje uplata donacija u kripto valutama. Za razvoj klijentskog dijela web platforme treba koristiti React razvojno okruženje uz upotrebu knjižnica funkcija po izboru. Za razvoj pametnih ugovora treba koristiti Solidity programski jezik na proizvoljno odabranom blockchainu. Implementirati funkcionalnosti stvaranja, ažuriranja, brisanja i pregleda donacijskih kampanja u pametnom ugovoru preko korisničkog sučelja. Također razmotriti mogućnost nagrađivanja donatora izdavanjem nezamjenjivih tokena (NFT).

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 20. ožujka 2023.

Mentor: Prof. dr. sc. Kristijan Lenac

Predsjednik povjerenstva za diplomski ispit: Prof. dr. sc. Miroslav Joler

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, srpanj 2023.

_____

Martin Luk

Zahvala

Zahvaljujem se mentoru na podršci tijekom pisanja ovoga rada i korisnim raspravama i savjetima.

Sadržaj

Popis slika

Poglavlje 1

Uvod

U posljednjem desetljeću značajno su se razvili financijski sustavi s pojavom digitalnih novčanika, Internet bankarstva, mobilnih i beskontaktnih plaćanja. Jedan takav razvoj je i plaćanje kriptovalutama koje pružaju decentralizirano, anonimno i transparentno plaćanje temeljeno na tehnologiji blockchaina. Tom razvoju je pridonijela i vizija decentraliziranog Interneta nazvanog "web3" u kojemu nije potrebno povjerenje između korisnika pri korištenju decentraliziranih aplikacija (eng. Decentralized applications (DApps)).

U ovom radu istražit će se potencijal plaćanja kriptovalutama za poboljšanje učinkovitosti i transparentnosti procesa donacija i razviti će se rješenje za doniranje u obliku decentralizirane aplikacije nazvanom "Web3 platforma za donacije". Prikupljanje sredstava kriptovalutama danas je najbrže rastuća metoda doniranja zbog smanjenih troškova transakcija, anonimnosti, mogućnosti donacije s bilo koje strane svijeta, brzine, porezne efikasnosti i brzorastućeg usvajanja kriptovaluta. Jedan od ciljeva ovog rada je istražiti i shvatiti prednosti i nedostatke tih plaćanja kako bi se maksimizirao njihov potencijal.

U drugom poglavlju ovog rada detaljno će se opisati zadatak i razraditi njegove specifikacije. Analiza trenutno postojećih rješenja i njihova usporedba slijedi u trećem poglavlju, a nakon toga, u četvrtom poglavlju, pobliže će se opisati korištene tehnologije za izradu Web3 platforme za donacije. U petom poglavlju prolazi se kroz sve funkcionalnosti i mogućnosti Web3 platforme za donacije uz objašnjene proce-

dure za njenu upotrebu. Također u petom poglavlju će se objasnit i kod pametnih ugovora te način testiranja Web3 platforme za donacije.

Poglavlje 2

Opis problema

U ovom poglavlju opisat će se ideja Web3 platforme za donacije, motivacija za izradu i njene specifikacije. Istražit će se koncept weba3, raspraviti kako funkcionira te usporediti s web2 Internetom, ističući prednosti i potencijalne nedostatke obiju tehnologija.

2.1 Motivacija

Poticaj za izradu platforme je nastao nakon što sam primijetio da trenutne donacijske platforme ne nude opcije doniranja u kriptovalutama. Netko tko želi sudjelovati u donacijskoj kampanji koja je u drugoj državi s drugom fiat valutom morat će pretvarati lokalnu valutu u valutu strane zemlje što uključuje naknade ili nepovoljne tečajeve što u konačnici rezultira manjem iznosu donacije. Također prilikom doniranja preko granica, banka ili platni servis može naplatiti naknade za međunarodne transakcije. Neke zemlje znaju imati propise ili ograničenja za slanje novca preko granica (posebno za velike iznose). To može zahtijevati dodatnu dokumentaciju ili čak spriječiti vas da napravite donaciju.

U slučajevima kada su donacije bile isključivo u kriptovalutama, kao kada je na primjer za donacije Ukrajini početkom rata, za to doniranje se tada objavilo samo adresu Bitcoin i kasnije još Ethereum kriptonovčanika na Twitter društvenoj mreži. U tom trenutku nije bilo donacijske platforme koja omogućuje doniranje s

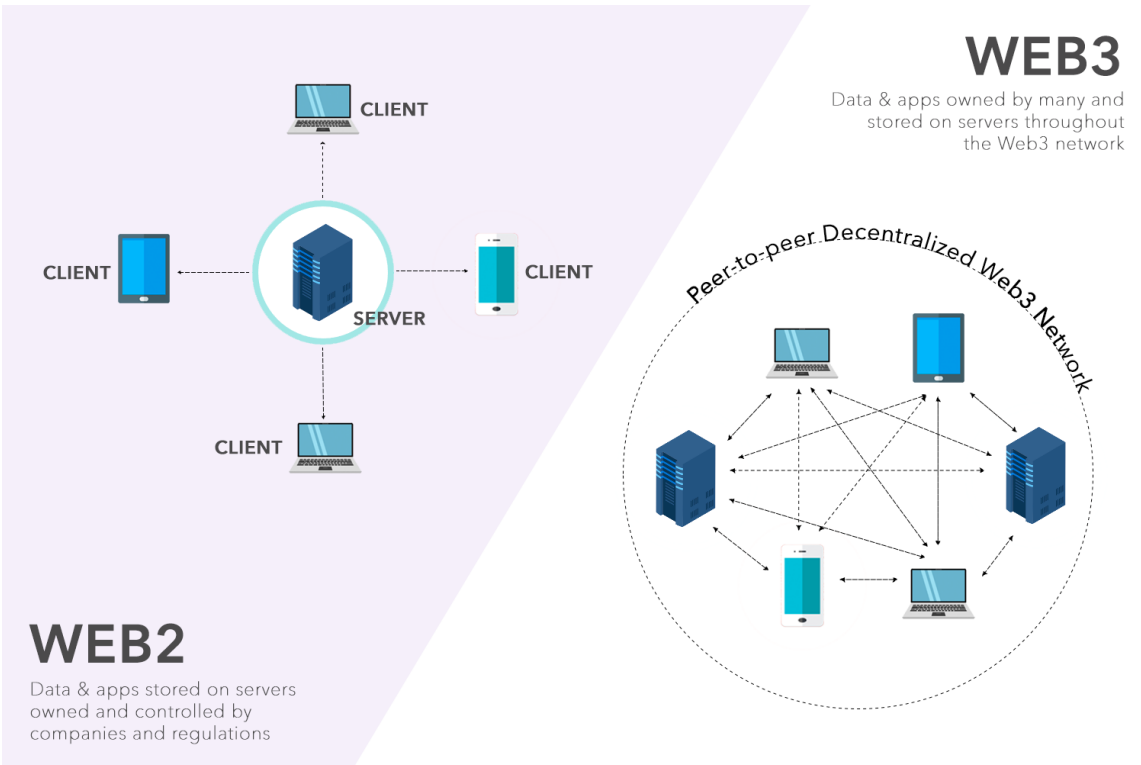
kriptovalutama na kojoj se mogla brzo napraviti donacijska kampanja za Ukrajinu.

Twitter može biti učinkovita platforma za dijeljenje informacija i podizanje svijesti no nije najbolja platforma za kampanje prikupljanja donacija. Jedan od razloga za to bio bi Twitterovo ograničenje od 280 znakova po objavi što može biti značajno ograničenje pri pokušaju prenošenja dovoljno konteksta i informacija potrebnih potencijalnim donatorima kako bi razumjeli i podržali kampanju. Brzi ritam Twittera može uzrokovati da se "tweetovi" brzo zatrpaju novijim sadržajem, smanjujući vidljivost i vijek trajanja kampanja za donacije. To može rezultirati lažnim kampanjama ili pokušajima prijevare što se i dogodilo nakon objave kampanje za donaciju Ukrajini kada se pojavilo mnogo prevara i lažnih adresa za donacije. Twitter ima ograničene multimedijske i informativne mogućnosti u usporedbi s drugim donacijskim platformama što otežava stvaranje zanimljivog i uvjerljivog sadržaja kako bi se uvjerili potencijalni donatori u važnost cilja.

Adrese Bitcoina i Ethereuma koje su bile objavljene za donacije Ukrajini, ograničile su donacije samo na dva blockchaina koja su poznata po visokim troškovima transakcija što je uvelike utjecalo na konačan iznos donacije. Troškovi transakcija na Ethereumu kreću se u vrijednosti od 2$ do 5$ po transakciji za vrijeme manjeg broja transakcija, do 60$ pa i više po transakciji za vrijeme velike količine broja transakcija [1]. Troškovi transakcija na Bitcoin blockchainu su se kretali od par američkih dolara, kada je manja količina prometa, do dvadeset i nešto američkih dolara kod veće količine prometa [2]. Stoga je za ovu Web3 donacijsku platformu odabran Polygon koji ima troškove po transakciji u vrijednosti oko 0.03$ [3]. Prosječna provizija plaćanja kreditnim i debitnim karticama u 2023. godini kreće se od 1.5% do 3.5% za kreditne kartice [4] i oko 1.2% za debitne [5]. Kada bi se izvršila donacija nekoj neprofitnoj udruzi od 1000€ preko kreditne kartice, ta udruga bi dobila iznos donacije između 985 i 965 eura. Ako bi se izvršila ista vrijednost te donacije preko Polygon blockchaina, udruga bi dobila potpuni iznos donacije.

2.2 Web3

Cilj ovog diplomskog rada je naučiti koristiti se web3 tehnologijama za izradu Web3 platforme za humanitarne donacije. Web3 tehnologije odnose se na decentralizirane i distribuirane tehnologije koje zajedno stvaraju novu verziju Interneta, često nazvanog "decentralizirani web", dok se s druge strane web2 oslanja na centralizirane platforme poput Googlea, Amazona i Facebooka koje imaju ogromnu moć i kontrolu nad korisničkim podacima, internetskim sadržajima i uslugama (slika 2.1).



Slika 2.1 *Razlika web2 i web3 arhitekture* [6]

Ova nova verzija Interneta nastoji vratiti snagu u ruke korisnika i poboljšati sigurnost, privatnost i kontrolu nad osobnim podacima s enkripcijom i decentraliziranim rješenjima za pohranu. Uklanjanje ovisnosti o središnjim tijelima i posrednicima osigurava da su korisnički podaci zaštićeni od neovlaštenog pristupa. Nasuprot tome, web2 je suočen s problemima vezanim uz brige o privatnosti zbog trgovanja korisnič-

kim podacima i sigurnosnim propustima pošto su centralizirane platforme atraktivne mete za hakiranje. Jedna od ključnih komponenti web3 tehnologija je Blockchain. Blockchain je distribuirana digitalna knjiga koja bilježi evidenciju transakcija s kriptovalutama poput Bitcoina i Ethereuma na mreži računala na siguran, otporan na manipulacije i transparentan način te je zbog toga temelj za mnoge web3 aplikacije. Kako bi web3 aplikacije, koje su izgrađene na vrhu blockchain tehnologije i rade na Peer-to-peer (P2P) mreži, umjesto na centraliziranom poslužitelju, mogle funkcionirati, koriste pametne ugovore za svoje operacije. Pametni ugovori su samostalni ugovori s uvjetima dogovora izravno napisanim u kodu te se automatski izvršavaju kada su ispunjeni svi određeni uvjeti bez potrebe za posrednicima poput odvjetnika ili bilježnika. Decentralizacija većih podataka na webu3 ostvaruje se rješenjima za distribuciju podataka kroz mreže čvorova kao što je InterPlanetary File System (IPFS). Još jedan aspekt weba3 su digitalni tokeni. Oni predstavljaju vlasništvo nad stvarnim ili digitalnim sredstvima. Jedan od standarda ili vrsta tih tokena su nezamjenjivi tokeni (eng. Non-fungible token (NFT)). To su jedinstveni, nedjeljivi digitalni tokeni koji predstavljaju vlasništvo nad određenom digitalnom ili fizičkom imovinom. Neki od primjera te imovine bile bi razne umjetnosti, kolekcionarstva i predmeti u računalnim igrama. Olakšavaju kreatorima da unovče svoj rad, uspostave porijeklo i olakšaju trgovanje imovinom na decentraliziranim tržnicama. Decentralizirane tržnice mogu ponuditi veću transparentnost, smanjene naknade i otpornost na cenzuru u usporedbi s tradicionalnim e-trgovinskim platformama. Web3 tehnologije također omogućuju korisnicima veću kontrolu nad svojim online identitetima. Stoga u ovom radu će se koristiti decentralizirani digitalni identitet i mehanizmi za očuvanje privatnosti poput kriptonovčanika što omogućava bolju kontrolu podataka, privatnost i sigurnost u usporedbi s centraliziranim identitetskim sustavima. Kriptonovčanik sigurno pohranjuje privatne ključeve za pristup kriptovalutama na blockchainu.

Sve ove tehnologije potrebne su kako bi decentralizirana Web3 platforma bila temeljena na pametnim ugovorima i blockchain tehnologiji da omogući transparentne donacije u kriptovalutama.

2.3 Specifikacije

Cilj Web3 platforme za donacije je da pojednostavi i ubrza izradu donacijskih kampanja i doniranje. Prvi korak u postizanju tog cilja je funkcionalno, lijepo dizajnirano i intuitivno korisničko sučelje. Najčešći generalni moderan dizajn web korisničkog sučelja većinom obično sadrži gradijente boja, djelomično prozirne elemente, animacije i zaobljene kuteve elemenata korisničkog sučelja. Uz moderan dizajn web sučelja, potrebno je da paleta boja na njemu odgovara tematici web aplikacije za koju je namijenjeno sučelje. Stoga za ovu aplikaciju su odabrane većinom svijetle vesele boje. Na vrhu web sučelja će se nalaziti logo, tražilica i gumb za spajanje kriptonovčanika s Web3 platformom. S lijeve strane biti će izborna traka s 3 moguća izbora koja će voditi na početnu stranicu, sučelje za izradu nove kampanje i profil trenutno povezanog korisnika s listom njegovih donacijskih kampanja.

Na početnoj stranici će se nalaziti popis svih donacijskih kampanja koje će se učitati iz pametnog ugovora. Na svakoj kampanji će pisati osnovne informacije o njoj kao što je cilj prikupljanja donacije i koliko je trenutno prikupljeno, ime kampanje, slika, trajanje i kratak opis. Korisnik će moći odabrati sebi zanimljivu kampanju da nauči više o njoj na sučelju s detaljima odabrane kampanje. Na tom sučelju osim proširenih informacija o donacijskoj kampanji, bit će i sučelje za slanje donacija. Na sučelju za slanje doancija će biti informacije o NFT nagradama za donacije što će dodatno potaknuti korisnika na donaciju. Kako bi korisnik mogao saznati više o ciljevima kampanje za koju donira, bit će prikazani razni oblici medija i sadržaja koje će autor kampanje moći postaviti u svojoj kampanji. Taj sadržaj će autor kampanje naknadno moći uređivati ili ažurirati s novostima vezanim uz kampanju. Na dnu detalja o donacijskoj kampanji, bit će prikazane adrese kriptonovčanika donatora za tu kampanju i iznosi donacija. Adresa kriptonovčanika je niz slova i brojeva s kojeg se mogu slati i primati kriptovalute ili NFT-ovi.

Da bi korisnik napravio svoju kampanju, morat će se prvo povezati s kriptonovčanikom na Web3 platformu te otići na sučelje za izradu nove kampanje na izbornoj traci. Ondje će biti ponuđena polja za unos naziva donacijske kampanje, slike, video sadržaja, običnog teksta i PDF sadržaja što može biti prezentacija ili dokument o donacijskoj kampanji. Osim toga, autor će na tom sučelju moći odrediti do kojeg

datuma će kampanja biti aktivna i koja je ciljana vrijednost iznosa koji se želi prikupiti. Prilaganje PDF i video sadržaja će se ostvarivati preko Uniform Resource Locator (URL) poveznica na određene sadržaje. PDF datoteke će imati i mogućnost učitavanja u IPFS mrežu izravno iz sučelja te će se automatski spremiti generirana poveznica preko IPFS prolaza (eng. gateway) do njega. Video sadržaj može biti iz bilo kojeg internetskog izvora (YouTube, Vimeo, Content delivery network (CDN), IPFS...). Kada korisnik popuni sve informacije ispravno, pokrenuti će se transakcija koja će zapisati sve podatke u pametni ugovor te će se uspješnim izvršenjem transakcije prikazati donacijska kampanja na početnoj stranici korisničkog sučelja. Web korisničko sučelje će također biti i responzivno ovisno o veličini ekrana tako da korisnici koji pristupe preko mobilnih uređaja imaju dobro korisničko iskustvo i mogućnost doniranja.

Poglavlje 3

Pregled postojećih rješenja

Web3 i web2 platforme za donacije služe kao sredstva za doprinos humanitarnim ciljevima ili pojedinačnim projektima. Međutim, razlikuju se u vrstama valuta koje podržavaju i osnovnim tehnologijama koje ih pokreću. U nastavku slijedi usporedba tih dviju vrsta platformi.

Web2 platforme za donacije omogućuju donacije pomoću fiat valuta, kao što su dolari, euri ili funte, i putem konvencionalnih načina plaćanja poput kreditnih kartica, debitnih kartica ili bankovnih prijenosa. Ove platforme oslanjaju se na tradicionalne financijske institucije i procesore plaćanja što ih čini centraliziranijima i potencijalno manje transparentnima. Donatori obično moraju dati osobne podatke prilikom doniranja putem web2 platformi što može izazvati zabrinutost za privatnost pojedinaca. Web2 platforme mogu imati ograničenja u pogledu transakcija preko granica što može ograničiti prihvaćanje donacija od međunarodnih podupiratelja. Tradicionalne platforme za donacije možda zahtijevaju ručni nadzor kako bi se osiguralo da su sredstva ispravno raspoređena što može dovesti do povećanih administrativnih troškova i mogućih ljudskih pogrešaka. Platforme temeljene na fiat valutama obično imaju veće transakcijske naknade zbog uključenosti procesora plaćanja i banaka čime se smanjuje iznos sredstava koji na kraju stiže do primatelja.

Web3 platforme za donacije omogućuju doniranje pomoću kriptovaluta poput Bitcoina, Ethereuma, Matica i ostalih tokena. To omogućuje donatorima doprinos sredstava bez prolaska kroz tradicionalne financijske institucije. Te platforme te-

melje se na tehnologiji blockchaina što znači da djeluju decentralizirano. To može dovesti do veće transparentnosti jer su zapisi o donacijama javni i nepromjenjivi. Donatori mogu zadržati svoju anonimnost prilikom doprinosa uzrocima jer kriptovalutne transakcije ne zahtijevaju nužno otkrivanje osobnih podataka. Kripto bazirane platforme omogućuju donacije iz bilo kojeg dijela svijeta ako donator ima pristup Internetu i svome kriptonovčaniku. Web3 platforme mogu koristiti pametne ugovore za automatizaciju distribucije donacija osiguravajući dodjelu sredstava bez posrednika. Transakcije kriptovaluta općenito imaju niže naknade u usporedbi s tradicionalnim načinima plaćanja, što omogućuje da više sredstava dođe do krajnjeg primatelja.

Prednosti web2 platforme za donacije (donacije u fiat valuti):

- **Poznavanje**: Većina ljudi je upoznata s fiat novcem i tradicionalnim sustavima plaćanja što im olakšava doniranje putem web2 platformi.
- **Stabilnost**: Fiat valute su općenito stabilnije od kriptovaluta čime se smanjuje rizik od fluktuacija vrijednosti donacija.
- **Šira prihvaćenost**: Više organizacija prihvaća donacije u fiat valuti što pruža širi spektar projekata koje donatori mogu podržati.
- **Jasne porezne implikacije**: Porezni tretman donacija u fiat valuti uglavnom je dobro utemeljen i razumljiv što olakšava proces za donatore i primatelje.

Nedostaci web2 platforme za donacije:

- **Veće naknade**: Tradicionalni sustavi plaćanja često imaju veće naknade od transakcija s kriptovalutama čime se smanjuje iznos doniranih sredstava koji dosegne krajnjeg primatelja.
- **Sporije transakcije**: Transakcije s fiat valutama, posebno preko granica, mogu potrajati duže u odnosu na transakcije kriptovaluta.
- **Ograničeni globalni doseg**: Neke web2 platforme za donacije možda ne podržavaju međunarodne donacije ili mogu naplaćivati dodatne naknade za transakcije preko granica.
- **Manjak privatnosti**: Donatori moraju pružiti osobne podatke prilikom doniranja u fiat valutama, čime se smanjuje njihova privatnost.

Prednosti Web3 platforme za donacije (kriptodonacije):

- **Globalni doseg**: Donacije kriptovalutama mogu se slati i primati iz bilo kojeg dijela svijeta čime se olakšava ljudima u različitim zemljama da podrže projekte koji ih zanimaju.
- **Niže naknade**: Kriptovalute obično imaju niže transakcijske naknade od tradicionalnih sustava plaćanja što omogućuje da više doniranih sredstava dođe do krajnjeg primatelja.
- **Brže transakcije**: Transakcije kriptovalutama često su brže od tradicionalnih bankovnih prijenosa što omogućuje bržu isporuku sredstava primateljima.
- **Transparentnost i sigurnost**: Tehnologija blockchaina nudi nepromjenjiv i javni zapis transakcija što može pružiti veću transparentnost i povjerenje u proces doniranja.
- **Privatnost**: Donatori mogu donirati anonimno, štiteći svoje osobne podatke od povezivanja s donacijama.

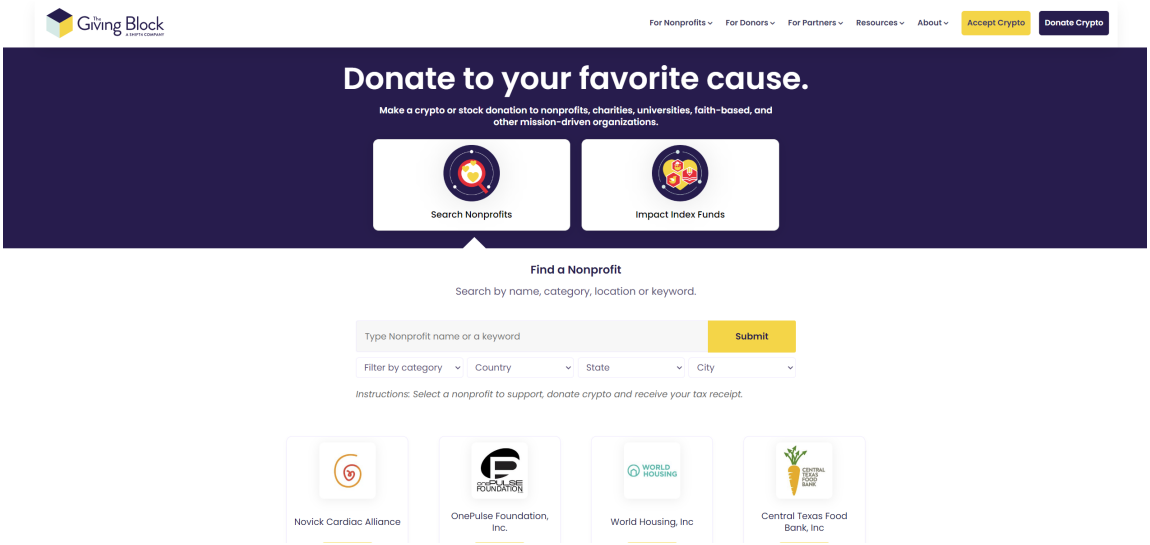
Nedostaci Web3 platforme za donacije:

- **Nestabilnost**: Kriptovalute mogu biti vrlo nestabilne što može dovesti do značajnih fluktuacija vrijednosti donacija.
- **Nedostatak regulacije**: Nedostatak regulacije oko kriptovaluta može otežati nekim organizacijama prihvaćanje ili upravljanje kriptodonacijama.
- **Implementacija**: Nisu sve organizacije spremne prihvatiti donacije kriptovalutama što ograničava broj projekata koji se mogu koristiti ovom tehnologijom.
- **Porezne posljedice**: Porezni tretman donacija kriptovalutama može biti nejasan ili se razlikovati između država što komplicira proces za donatore i primatelje.

Web3 platforme nude globalni doseg, niže naknade i veću transparentnost, dok web2 platforme nude poznavanje, stabilnost i širu prihvaćenost.

3.1 The Giving Block

Prva postojeća platforma koja će se pregledati je The Giving Block (slika 3.1). Osnovana je 2018. godine i omogućuje neprofitnim organizacijama da jednostavno prihvaćaju donacije u kriptovalutama i time se povežu s novom skupinom potencijalnih donatora. Platforma podržava širok raspon popularnih kriptovaluta. To omogućuje neprofitnim organizacijama pristup rastućem tržištu tehnički potkovanih i ekološki osviještenih donatora koji preferiraju korištenje kriptovaluta u odnosu na tradicionalne načine davanja.



Slika 3.1 *The Giving Block web stranica* [7]

Donacije kriptovalutama za neprofitne organizacije s The Giving Block platformom funkcioniraju na sljedeći način:

1. Neprofitne organizacije koje žele prihvaćati donacije u kriptovalutama prvo se moraju registrirati na platformi The Giving Block tako što će pružiti potrebne informacije o svojoj organizaciji.
2. Nakon registracije, neprofitna organizacija mora postaviti digitalni novčanik za primanje donacija u kriptovalutama. The Giving Block ima smjernice za odabir sigurnog i kompatibilnog novčanika ili preporučuje jedan temeljem potreba

organizacije.

3. Kada je novčanik postavljen, neprofitna organizacija može integrirati donacijski widget The Giving Block na svoju web stranicu [8]. Ovaj widget omogućuje donatorima da izvrše donacije u kriptovalutama izravno putem web stranice organizacije.
4. Kada donatori odluče donirati kriptovalutama, mogu odabrati željenu kriptovalutu od dostupnih na popisu donacijskog widgeta. Dobit će jedinstvenu adresu novčanika na koju trebaju poslati svoju donaciju koja odgovara digitalnom novčaniku neprofitne organizacije.
5. Ovisno o preferenciji neprofitne organizacije, doniranu kriptovalutu mogu zadržati u izvornom obliku ili je pretvoriti u fiat valutu. Konverzija se može izvršiti putem kriptomjenjačnice, a neprofitna organizacija mora se pridržavati lokalnih propisa i poreznih zahtjeva.

Ovim koracima neprofitne organizacije mogu iskoristiti platformu The Giving Block kako bi prihvatile donacije u kriptovalutama od novog skupa potencijalnih donatora.

Za donatore postupak doniranja kriptovaluta preko The Giving Block platforme funkcionira na sljedeći način:

1. Pronađe se neprofitna organizacija kojoj se želi donirati. Provjeri se je li na njihovoj web stranici integriran donacijski widget The Giving Block što označava da prihvaćaju donacije kriptovalutama.
2. Klikne se na donacijski widget i odabere kriptovaluta kojom se želi izvršiti donacija. The Giving Block podržava razne popularne kriptovalute poput Bitcoina (BTC), Ethereuma (ETH) i drugih.
3. Unese se iznos koji se želi donirati u odabranoj kriptovaluti. Može se unijeti vlastita e-mail adresa i osobni podaci kako bi se primila potvrda o donaciji i eventualna porezna potvrda.
4. Widget će prikazati jedinstvenu adresu novčanika na koju se treba poslati donacija. Kopira se adresa novčanika i otvori svoj kriptonovčanik na kojem se unese kopirana adresa kao odredišna adresa za transakciju i upiše iznos do-

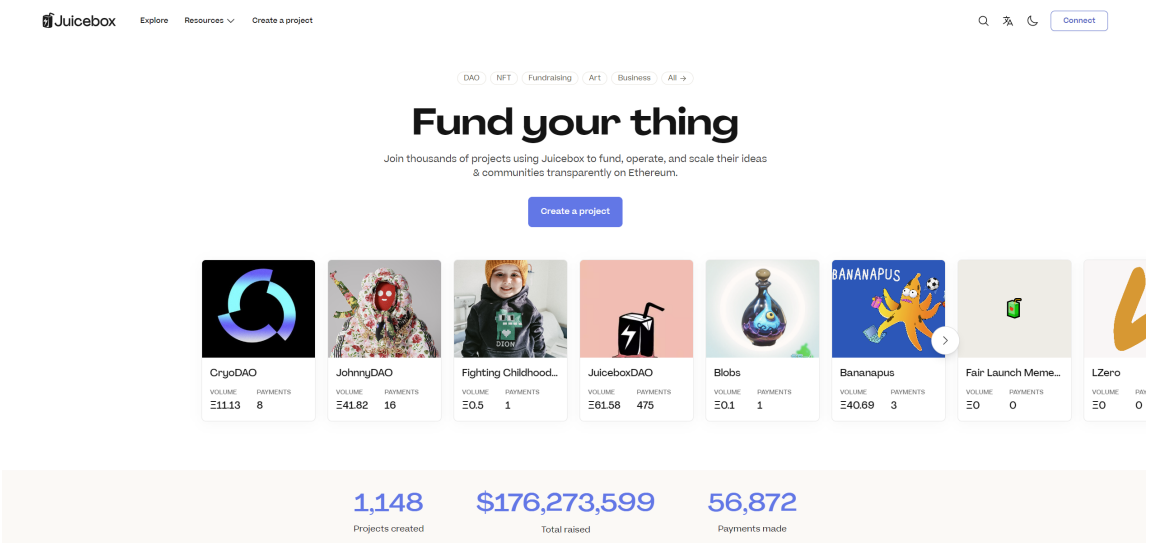
nacije. Provjeri se jesu li svi podaci ispravni, uključujući adresu novčanika, kriptovalutu i iznos donacije, te se potvrdi transakcija.

5. Nakon što se inicijalizira transakcija blockchain mreža će je obraditi. Vrijeme potrebno za potvrdu transakcije može varirati ovisno o kriptovaluti koja se koristi i opterećenju mreže. Nakon što je transakcija potvrđena trebala bi stići e-mail potvrda od neprofitne organizacije o uspješnoj donaciji.

The Giving Block oprema neprofitne organizacije alatima i resursima potrebnim za prihvaćanje kriptovaluta kao donacijama. Donatori na platformi imaju izbor doniranja i u index fondove koji onda raspodijele donaciju na više organizacija koje imaju sličnu svrhu ili cilj. Neke od raznih svrha su Turska i Syria, edukacija, medicina, životinje, okoliš, hrana i voda. Platforma ima solidnu listu kriptovaluta koje se prihvaćaju, ali korisnik mora ručno izvršiti sve donacije iz svog kriptonovčanika na adresu koja mu se predstavi. U takvom postupku mogu se dogoditi razne ljudske pogreške pri slanju donacija na taj način, pogotovo kod neiskusnih korisnika. Sredstva se mogu poslati na pogrešnu adresu, može se odabrati pogrešna kriptovaluta ili koristiti kriva blockchain mreža za slanje na određenu adresu što dovodi do nepovratno izgubljenih sredstava. Na vlastitom rješenju Web3 platforme za donacije u ovom radu, kriptonovčanik se povezuje s platformom na ispravnoj blockchain mreži i donacije se prosljeđuju automatski preko pametnog ugovora na odgovarajuću adresu čime ne može doći do ljudske pogreške i gubitka sredstava.

3.2 Juicebox

Juicebox je protokol otvorenog koda izgrađen na Ethereum blockchainu koji je u vlasništvu zajednice (slika 3.2). Ova platforma nije prilagođena humanitarnim donacijama kao prethodna, ali će se obraditi iz razloga što koristi kriptovalute za financiranje web3 projekata objavljenih na platformi. Juicebox platforma predviđena je za podržavanje NFT projekata i Decentralized Autonomous Organisation (DAO) projekata. Kada korsinik podrži te projekte obično zauzvrat dobije NFT iz kolekcije ili tokene za vladanje ako se radi o DAO projektu. Juicebox ima ugrađen otkup tokena ili NFT-a u platformu što olakšava projektima taj dio posla.



Slika 3.2 *Juicebox web stranica* [9]

Kako bi se pokrenula kampanja financiranja vlastitog projekta na Juiceboxu potrebno je popuniti sve detalje o projektu. Ti detalji su ime projekta, opis, logo, razne poveznice za web stranice projekta, naslovna slika i tekst koji će se prikazati korisnicima kod plaćanja. Nakon toga autor kampanje ima izbor zaključavanja početno izabranih pravila projekta, što bi značilo da se ne može predomisliti usred kampanje i mijenjati pravila koja je postavio pri izradi kampanje, i puštanja opcije izmjene pravila nakon nekog određenog vremena ili odmah. Opcija zaključanih ili otključanih pravila projekta bit će vidljiva na stranici kampanje. Prvi primjer tih pravila projekta bio bi datum pokretanja projekta. Zatim autor može odrediti pravila financiranja vlasnika projekta što može biti fiksna cifra u ciklusima, odmah da mu se isplati koliko god vlasnik želi ili da mu se ništa ne isplati nego da služi za otkup tokena od korisnika. Tokeni se dijele ulagačima koji financiraju projekt. Mogu biti korišteni za vladanje projektom ili pristupe u zajednici projekta. Autor projekta ima izbor kreiranja svog vlastitog tokena kojemu sam određuje pravila token ekonomije (eng. tokenomics) ili može koristiti već ponuđena pravila tokena koja su generalno dobra za većinu projekata. Mintani tokeni će biti ERC-20 standarda. Zatim autor može napraviti vlastitu NFT kolekciju kako bi nagradio svoje pristaše s NFT-ovima.

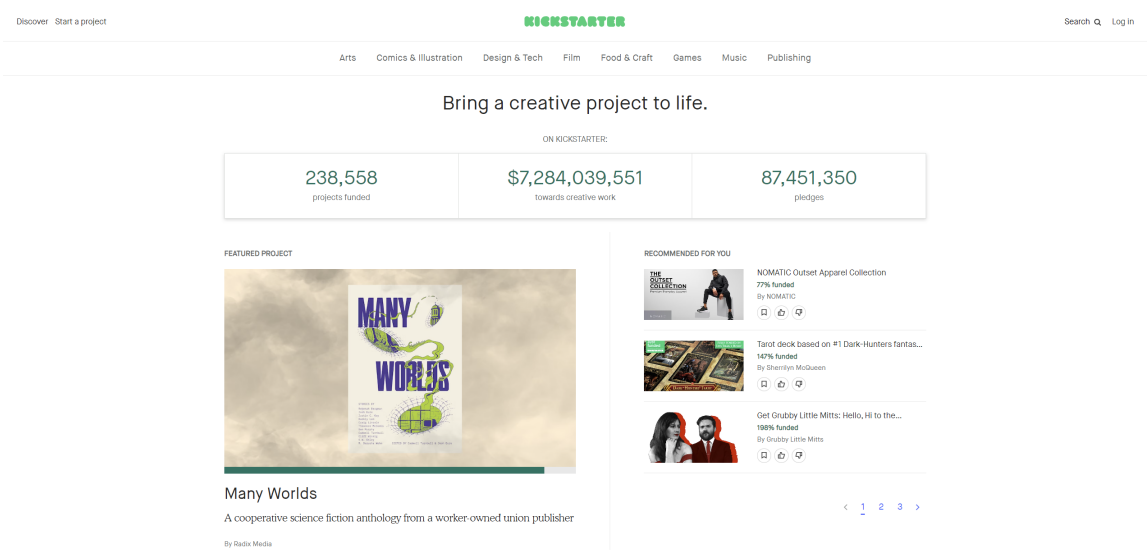
Na kraju autor ima za odrediti pravila naknada, plaćanja i migracije nakon čega se prikaže sučelje kampanje gdje može još jednom pregledati sve opcije koje je odabrao prije nego postavi kampanju na Ethereum glavnu mrežu (eng. mainnet) [10].

Pokrenuti kampanju za projekt na Juicebox platformi je malo kompleksniji postupak nego što će biti za pokretanje donacijske kampanje na Web3 platformi za doniranje. Također je potrebno više tehničkog znanja kako bi se mogle iskoristiti sve dodatne mogućnosti koje pruža Juicebox platforma. Platforma nije prigodna za donacije u humanitarne svrhe jer su troškovi transakcija na Ethereum blockchainu velike i većinom donacijske kampanje nemaju koristi od dodatnih opcija koje pruža Juicebox platforma.

3.3 Kickstarter, GoFundMe, Indiegogo

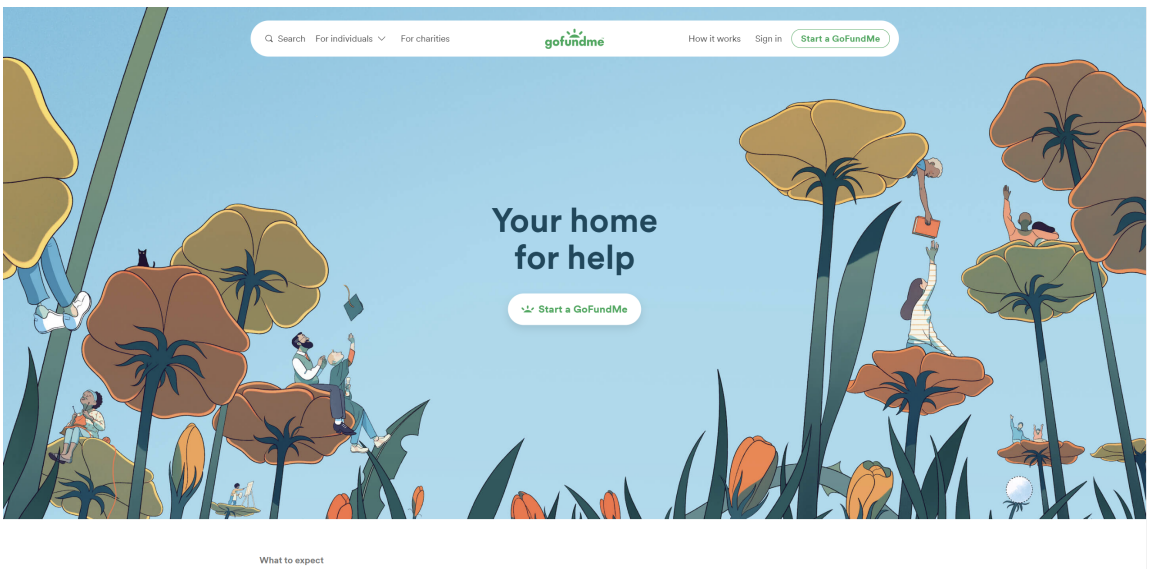
Na webu2 ima puno postojećih riješenja za grupno financiranje od kojih su Kickstarter, GoFundMe i Indiegogo tri najpoznatije platforme. Svaka od ovih platforma omogućuje korisnicima jednostavno stvaranje projekata ili kampanja i podržavanje projekata kako bi im pomogli da postignu svoje ciljeve. Podržavanje projekata na tim platformama se ne može izvršavati s kriptovalutama, već samo fiat valutama za što platforme uzimaju velike naknade. U nastavku će se pobliže objasniti razlike i sličnosti između navedenih platformi.

Kickstarter je platforma koja se usredotočuje na podršku kreativnim projektima poput umjetnosti, glazbe, filma, igrica, tehnologije i sličnog (slika 3.3). To je najveća i najpoznatija platforma za grupno financiranje i ima veći ukupan iznos prikupljenih sredstava nego sve ostale platforme za grupno financiranje zajedno. Oko 65% projekata na Kickstarteru u potpunosti uspije prikupi sredstva u određenom vremenskom razdoblju kampanje [12]. Sustav financiranja projekata na Kickstarteru je sve ili ništa. To znači da ako projekt ne uspije dosegnuti svoj cilj prikupljanja sredstava u određenom vremenskom roku, neće dobiti ništa od prikupljenih sredstava za projekt. Također ne naplaćuju se naknade ako kampanja nije uspješna. Na Kickstarteru se ne mogu prikupljati sredstva za dobrotvorne svrhe i platforma uzima 5% provizije od uspješno financirane kampanje. Također uzima 3% plus dodatnih 30 centi po uplati



Slika 3.3 *Kickstarter web stranica* [11]

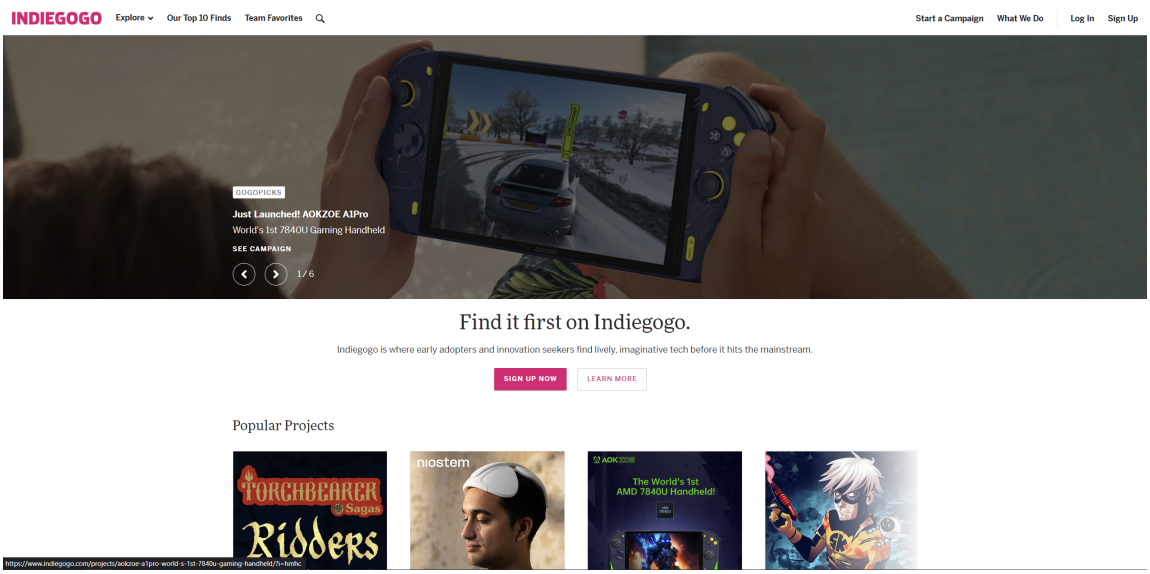
korisnika kampanji (slika 3.6) [12].



Slika 3.4 *GoFundMe web stranica* [13]

Crowdfunding platforma GoFundMe je više usmjerena na pojedince i osobne ci-

ljeve što mogu biti kampanje za financiranje događaja poput obrazovanja, medicinskih tretmana i sportskih timova (slika 3.4). Česta su i financiranja neprofitnih organizacija na platformi. Financiranje na GoFundMe nije sve ili ništa. To znači da se zadržava svaki novac koji se prikupi, bez obzira na to je li dostignut određeni cilj financiranja. Naknade iznose 2.9% od iznosa plus 30 centi po transakciji za osobne kampanje i 1.9% plus 20 centi za donacije neprofitnim organizacijama (slika 3.6). Naknade su iste bez obzira je li kampanja u potpunosti ili djelomično financirana [12].



Slika 3.5 *IndieGoGo web stranica* [14]

Indiegogo platforma se ne specijalizira za nikakve tipove kampanja. Na web stranici se prihvaćaju sve vrste projekata za prikupljanje sredstava (slika 3.5). To znači da su regulacije o tome što je na stranici vrlo slabe što može biti dobro jer svi mogu jednostavno napraviti kampanju čime je baza kampanja vrlo raznolika, a može biti i loše jer se platforma zatrpa s velikim brojem ne uvijek smislenih kampanja koje se natječu za pažnju donatora. Kampanje na Indiegogu imaju niže prosječne prikupljene iznose od nekih drugih crowdfunding stranica. Indiegogo je poseban po tome što jedini pruža dva izbora za model financiranja kampanja. Vlasnik kampanje može odabrati hoće li da prikupljanje sredstva za kampanju bude sve ili ništa, ili da

zadrže sve što prikupe neovisno o uspjehu financiranja kampanje. Naknade se plaćaju bez obzira postigne li se cilj prikupljanja ili ne te iznose 3% od iznosa donacije plus 20 centi za svaku transakciju (slika 3.6) [12].

	KICKSTARTER	INDIEGOGO	gofundme
Naknada platforme	5%	5%	0%
Naknada za obradu plaćanja	3%+0.3c	3%+0.2c	2.9%+0.3c

Slika 3.6 *Provizije za plaćanja na svakoj platformi* [15]

Na IndieGoGo i Kickstarteru kampanje mogu trajati najviše 60 dana ako su se produžile. To znači da ako se kampanja prvotno postavila da traje 30 dana (maksimum), na posljednji dan može ju se produžiti za najviše 30 dodatnih dana. Produljenje kampanje se može napraviti samo jednom. Što se tiče GoFundMe, prikupljanje sredstava za kampanju može trajati koliko god se želi. Nema rokova ni naknada povezanih s ostavljanjem kampanje aktivnom na ovoj platformi [15].

Ukratko, Kickstarter je platforma koja se fokusira na kreativne projekte poput tehnologije, umjetnosti, glazbe, filma, igara itd. Model financiranja projekata je sve ili ništa. GoFundMe platforma namijenjena je pojedincima i osobnim uzrocima. Model financiranja projekata je da autor zadrži sve dodijeljene novce neovisno ako je financiranje dostiglo cilj ili nije. Indiegogo je jedinstvena platforma koja omogućuje mnoge različite kampanje. Autor kampanje odabire ako je model financiranja sve ili ništa ili da zadrži sve što dobije.

Poglavlje 4

Korištene tehnologije

U nastavku su opisane korištene tehnologije i razlozi za njihov odabir. Već je definirano u zadatku da će izrada klijentskog dijela biti u Reactu uz odgovarajuće potrebne knjižnice. Jedna od tih knjižnica koja će omogućiti brzo i učinkovito stvaranje modernog web dizajna da izgleda profesionalno i privlačno za korisnike je Tailwind. Isto je definirano u zadatku da će za razvoj pametnih ugovora biti korišten Solidity programski jezik koji je dizajniran za stvaranje decentraliziranih aplikacija i omogućuje programerima pisanje složenih i sigurnih pametnih ugovora. Da bi se proces izgradnje, implementacije i upravljanja decentraliziranim aplikacijama pojednostavio, poslužiti će ThirdWeb platforma koja nudi alate i infrastrukturu koja uključuje programska sučelja aplikacije (eng. Application Programming Interface (API)), komplet za razvoj softvera (eng. Software development kit (SDK)) i unaprijed izgrađene predloške. Iako je Solidity najkorišteniji programski jezik za razvoj pametnih ugovora na Ethereum platformi, za razvoj ove decentralizirane aplikacije koristit će se Polygon blockchain drugog sloja. Polygon je privlačna alternativa iz više razloga. Neke od prednosti korištenja Polygon blockchaina umjesto Ethereuma uključuju poboljšanu skalabilnost u usporedbi s Ethereumovim prvoslojnim riješenjem. Polygon koristi sporedne lance i razna skalabilna rješenja poput Plasma, Zero-knowledge Rollupsa i Optimistic Rollups-a koja mogu obraditi veći broj transakcija u sekundi što je ključno za decentralizirane aplikacije s velikim prometom. Polygon nudi brže vrijeme potvrde transakcija od Ethereuma, smanjujući vrijeme čekanja korisnika za obradu njihovih transakcija. To je posebno važno za decentralizirane aplikacije koje

zahtijevaju interakcije u stvarnom vremenu poput platformi za decentralizirane financije kao što je slučaj u ovom radu. Zbog svoje arhitekture sporednih lanaca, Polygon značajno smanjuje transakcijske naknade u usporedbi s Ethereumom. To čini interakciju s decentraliziranim aplikacijama izgrađenima na Polygonu ekonomičnijom za developere i korisnike. Polygon je dizajniran za podršku više skalabilnih rješenja i međusobnu povezanost s drugim blockchain mrežama. To poboljšava interoperabilnost decentraliziranih aplikacija izgrađenih na Polygonu, omogućavajući im lako komuniciranje s drugim blockchain mrežama i korištenje imovine iz različitih ekosustava. Iako Polygon koristi vlastiti konsenzusni mehanizam Proof-of-Stake (PoS) za svoje sporedne lance, ipak nasljeđuje sigurnost Ethereum glavnog lanca kroz kontrolne točke što osigurava da decentralizirane aplikacije izgrađene na Polygonu imaju koristi od Ethereumove decentralizirane mreže. Zbog svih tih razloga Polygon platforma postaje sve popularniji ekosustav u prostoru blockchaina s povećanim brojem projekata i developera koji ga biraju za razvoj svojih decentraliziranih aplikacija. Zbog svih ovih prednosti Polygona, specifičnih zahtjeva i ciljeva decentralizirane aplikacije, odabran je Polygon kao platforma na kojoj će biti razvijeni pametni ugovori umjesto Ethereuma. Kako bi se izbjegla centralizirana pohrana podataka i spremanje većih podataka od Web3 platforme za donacije na blockchain (pošto je to skupocjeni oblik pohrane podataka) koristit će se rješenja koja distribuiraju podatke kroz mrežu čvorova što poboljšava sigurnost i privatnost podataka te sprječava cenzuru. Platforme koje će to omogućiti i koje će se koristiti pri izradi Web3 platforme za donacije su IPFS, FileCoin i Web3.storage. IPFS omogućuje distribuciju podataka kroz mrežu čvorova, FileCoin se brine da ti podaci ostanu na mreži čvorova, a Web3.storage povezuje obje platforme i omogućuje da ti podaci budu brzo dostupni Web3 platformi za donacije. Niske naknade transakcija Polygona omogućiti će efikasno mintanje NFT nagrada čiji će metapodaci (slike, atributi, opisi) biti pohranjeni na IPFS-u. Razvijanje Web3 platforme za donacije koristeći ove tehnologije pružit će transparentnost, smanjene troškove, brže transakcije i bolje korisničko iskustvo.

4.1 Pametni ugovori

Pametni ugovor je programabilni i samoizvršavajući ugovor s uvjetima napisanim u programskom kodu koji se izvodi na blockchain mreži. Drugim riječima, može se protumačiti kao digitalni protokol koji olakšava izvršenje ugovora bez potrebe za posrednicima te automatski slijedi unaprijed određene uvjete i odredbe ugovora. Napisani su u određenom programskom jeziku, ovisno o blockchanu za koji su pisani, a njihovo izvršenje je transparentno, sigurno i decentralizirano. Programski jezici za pisanje pametnih ugovora i blockchain platforme na kojima se koriste [16]:

- **Solidity** (Ethereum i ostale Ethereum Virtual Machine (EVM) kompatibilne blockchain platforme)
- **Rust** (Solana, Polkadot, NEAR i ostali)
- **Vyper** (Ethereum i ostale EVM kompatibilne blockchain platforme)
- **Yul** i **Yul+** (posredni jezik korišten za Solidity kompajler)
- **JavaScript** (NodeJS) (Hyperledger Fabric, NEAR)
- **C++** (EOS)

Koncept pametnih ugovora prvi je predložio znanstvenik Nick Szabo 1996. godine [17]. Međutim, pametni ugovori postali su široko prepoznati i usvojeni tek nakon stvaranja Ethereum blockchain platforme 2015. godine. Tada je potpuni Turingov programski jezik Ethereuma, Solidity, omogućio programerima stvaranje pametnih ugovora koji mogu izvršavati razne zadatke.

Neka ključna obilježja pametnih ugovora uključuju [17]:

- **Nepromijenjivost** - Jednom kada se postave (eng. deploy) pametni ugovori, ne mogu se lako mijenjati čime se osigurava da uvjeti ugovora ostanu nepromijenjeni.
- **Automatizacija** - Pametni ugovori se automatski izvršavaju kada su ispunjeni unaprijed određeni uvjeti što eliminira potrebu za ručnom intervencijom.
- **Sigurnost** - Decentralizirana priroda tehnologije blockchaina osigurava da su pametni ugovori zaštićeni od prijevare, hakiranja (ako programer nije napravio

pogreške) i cenzure.

- **Nema potrebe za povjerenjem** - Pametni ugovori omogućuju transakcije bez potrebe za povjerenjem između stranaka jer ne zahtijevaju središnji autoritet koji bi provjeravao ili provodio uvjete.
- **Učinkovitost** - Uklanjanjem posrednika i automatizacijom procesa, pametni ugovori mogu znatno smanjiti vrijeme i troškove povezane s tradicionalnim provođenjem ugovora
- **Brzina** - Ručna obrada dokumenata oduzima puno vremena što odgađa završetak cilja. Pametni ugovori su automatizirani te ne zahtijevaju osobno uključivanje u proces u većini slučajeva čime se štedi vrijeme.
- **Neovisnost** - Pametni ugovori isključuju uplitanje treće strane. Jamstvo za transakciju je sam program koji, za razliku od posrednika, neće dati razlog za sumnju u svoj integritet.
- **Pouzdanost** - Podaci upisani u blockchain ne mogu se promijeniti niti izbrisati. Ako jedna strana ne ispuni svoju obvezu u transakciji, druga strana je zaštićena uvjetima u pametnom ugovoru.
- **Bez grešaka** - Automatizirani sustav za izvršavanje transakcija i uklanjanje ljudskog faktora osiguravaju visoku točnost pri izvršavanju ugovora.

Nedostaci pametnih ugovora:

- **Nedostatak regulative** - Međunarodna pravna regulativa ne poznaje pojmove "blockchain", "pametni ugovor" i "kriptovaluta".
- **Teškoća implementacije** - Integracija pametnih ugovora s elementima stvarnog svijeta često zahtijeva puno vremena i novca.
- **Nemogućnost promjene pametnog ugovora** - Jedna od prednosti pametnih ugovora može se također smatrati i nedostatkom. Ako stranke postignu povoljniji dogovor ili postoje nekakvi nedostaci i pogreške u postojećem pametnom ugovoru, pametni ugovor se neće moći promijeniti nego će biti potrebno izraditi novi.

Dakle pametni ugovori olakšavaju, provjeravaju i provode izvedbu ugovora bez

potrebe za posrednicima. Solidity omogućuje programerima stvaranje tih samostalno izvršnih sporazuma s uvjetima napisanim u kodu. To se postiže pisanjem pravila, funkcija i struktura podataka koje se međusobno povezuju s EVM okruženjem za izvršavanje pametnih ugovora na Ethereumu i ostalim EVM kompatibilnim blockchain platformama.

Pametni ugovori imaju širok spektar potencijalnih upotreba uključujući upravljanje lancem opskrbe, financijske usluge (Decentralized finance (DeFi)), osiguranje, nekretnine, sustavi glasanja, decentralizirane aplikacije i mnogo više. Razlozi za upotrebu pametnih ugovora u raznim sektorima bili bi: pojednostavljenje procesa, smanjenje troškova, eliminacija potrebe za posrednicima ili središnjim tijelima čime se poboljšava učinkovitost, transparentnost i sigurnost. U nastavku će se istaknuti i opisati sve funkcionalnosti Soliditya koje su važne u razvoju svakog pametnog ugovora.

4.2 Solidity

Solidity je objektno orijentiran, visoko razinski (eng. high level), statički deklariran programski jezik za razvoj i implementaciju pametnih ugovora na blockchain platformama koje podržavaju EVM. EVM je virtualna komputacijska mašina unutar svakog Ethereum čvora koja izvršava bajt-kod ugovora. Za izvršavanje koda na EVM naplaćuje se unaprijed naknada koja se obično naziva "gas fee" i ima nativnu valutu blockchaina kao svoju izvornu valutu. EVM-ovi su svestrani, odnosno omogućuju izvršavanje pametnih ugovora napisanih u više različitih programskih jezika. Pametni ugovori napisani u Solidityu se kompiliraju u EVM bajt-kod, a Ethereum čvorovi izvršavaju EVM instance kako bi se složili oko izvršavanja istog skupa uputa.

Solidity je dominantan jezik za razvoj pametnih ugovora na Ethereumu i drugim kompatibilnim blockchainovima zbog svoje jednostavnosti za korištenje, aktivne zajednice i opsežne dokumentacije.

Ključne značajke Solidityja uključuju [18]:

- **Turingova potpunost**: Solidity je računski univerzalan jezik što znači da može izvršiti bilo koji algoritam koji se može opisati Turingovim strojem, uz

dovoljno resursa.

- **Tipovi varijabli**: Solidity podržava razne tipove varijabli, poput cijelih brojeva, booleovih vrijednosti, nizova, polja i definiranih struktura što omogućuje programerima stvaranje složenih struktura podataka i operacija. Tipovi varijabli koje Solidity ne podržava su float brojevi zbog svoje nepreciznosti i vezane liste.
- **Funkcije i modifikatori**: Solidity omogućuje programerima stvaranje funkcija i modifikatora za ponovnu upotrebu koda, enkapsulaciju i kontrolu pristupa.
- **Nasljeđivanje**: Solidity podržava nasljeđivanje ugovora, omogućavajući programerima stvaranje hijerarhije ugovora te nasljeđivanje svojstava i funkcija od roditeljskih ugovora.
- **Događaji**: Programeri mogu koristiti događaje u Solidityju za bilježenje specifičnih promjena i pokretanje funkcija na front-endu.
- **Obrada pogrešaka**: Solidity uključuje mehanizme za obradu pogrešaka i iznimki, poput naredbi `revert`, `require` i `assert`.

4.2.1 Kako rade Solidity pametni ugovori?

Solidity je programski jezik posebno dizajniran za stvaranje i implementaciju pametnih ugovora na blockchain platformama, prvenstveno Ethereumu. Da bismo razumjeli kako Solidity pametni ugovori rade, bitno je prvo shvatiti koncept pametnih ugovora i Ethereum blockchaina.

Nakon što se pametni ugovor napiše, mora se kompilirati u bajt-kod. Kompajler Soliditya, poznat kao "solc", pretvara visoko razinski (ljudski čitljiv) kod Solidityja u nisko razinski (strojno čitljiv) bajt-kod koji se može izvršiti na EVM [19].

Nakon kompilacije, bajt-kod pametnog ugovora se implementira na Ethereum blockchain. Ovaj postupak uključuje stvaranje posebne transakcije koja sadrži bajt-kod ugovora i bilo kakve argumente konstruktora potrebne za inicijalizaciju ugovora. Jednom kada se ova transakcija provjeri i zapiše u blok, pametni ugovor dobiva svoju jedinstvenu adresu na Ethereum mreži.

EVM je decentralizirano okruženje koje izvršava pametne ugovore na Ethereum blockchainu. Svaki čvor koji sudjeluje u Ethereum mreži pokreće instancu EVM-a što osigurava konsenzus o izvršenju i stanju ugovora. EVM interpretira bajt-kod, izvršava logiku ugovora i ažurira stanje blockchaina sukladno tome. Korisnici i drugi pametni ugovori mogu komunicirati s implementiranim ugovorom šaljući transakcije na njegovu jedinstvenu adresu. Te transakcije mogu uključivati pozive funkcija, podatke i Ether (izvorna kriptovaluta Ethereum mreže) kako bi izvršili određene radnje definirane u kodu pametnog ugovora.

Izvršavanje pametnih ugovora na Ethereum mreži zahtijeva računalne resurse. Kako bi se potaknuli rudari mreže da obrađuju transakcije i izvršavaju pametne ugovore, korisnici moraju platiti troškove transakcije u Etheru. Troškovi transakcije ovise o složenosti funkcija ugovora i trenutnim uvjetima mreže.

4.2.2 Solidity sintaksa

U Solidity sintaksi se može primijetiti utjecaj JavaScript, C++ i Python programskih jezika što ga čini poznatim i pristupačnim raznim programerima. Jezik je osmišljen kako bi olakšao stvaranje i upravljanje pametnim ugovorima na Ethereum blockchainu. U nastavku će se kratko opisati ključni elemenati Solidityja koji su korišteni za razvoj pametnih ugovora u ovom radu.

Pametni ugovori u Solidityju počinju ključnom riječju `contract` nakon čega slijedi ime ugovora i par vitičastih zagrada koje okružuju tijelo ugovora. Globalno deklarirane varijable pohranjuju svoje vrijednosti i podatke na blockchainu. Mogu biti raznih vrsta, kao što su uint, int, bool, string, address, mapping ili prilagođene vrste poput struktura i enuma. Varijable stanja mogu također imati specifikatore vidljivosti poput `public`, `private` ili `internal`.

Mjesta pohrane varijabla:

- **storage**: Trajna pohrana na blockchainu gdje su pohranjene varijable stanja ugovora. To je najskuplji oblik pohrane zbog svoje trajnosti, a pisanje u `storage` troši značajnu količinu goriva.
- **memory**: Privremeno područje pohrane koje postoji samo tijekom poziva

funkcije i njenog izvršavanja. Jeftinija je od `storage pohrane`, ali ne traje nakon što se poziv funkcije završi. Varijable deklarirane unutar funkcije automatski se pohranjuju u memoriji.

- **calldata**: Područje ulaznih podataka za argumente funkcije koji se prenose u vanjskim pozivima funkcija. `Calldata` je samo za čitanje i obično se koristi za velike strukture podataka kako bi se uštedjelo na troškovima goriva prilikom prijenosa podataka između funkcija.

```
contract ExampleContract {
    uint public counter; // Stored on the blockchain
    address private owner;

    function example(uint a, uint b) public {
        uint[] memory numbers = new uint[](2); // Array stored in memory
        numbers[0] = a;
        numbers[1] = b;
    }

    function processData(bytes calldata data) external {
    // Process data without the need to copy it to memory
    }
}
```

Funkcije definiraju ponašanje ugovora. Mogu čitati i mijenjati varijable stanja, interaktirati s drugim ugovorima i emitirati razne događaje. Funkcije također mogu imati specifikatore vidljivosti (`public`, `private`, `internal` ili `external`) i dodatne modifikatore poput `pure`, `view` ili `payable`.

Specifikatori vidljivosti funkcija:

- **public**: Javne funkcije dostupne su unutar ugovora, iz drugih ugovora i izvana putem transakcija i poziva. One su dio vanjskog sučelja (eng. interface) ugovora.
- **private**: Privatne funkcije mogu se pozivati samo unutar ugovora koji ih definira. Nisu dostupne izvedenim (eng. derived) ugovorima ili vanjskim entitetima.

- **internal**: Unutarnje funkcije dostupne su unutar ugovora i izvedenih ugovora. Nisu izravno dostupne izvan ugovora, ali se mogu izložiti putem javnih ili vanjskih funkcija u ugovoru ili izvedenim ugovorima.
- **external**: Vanjske funkcije dostupne su samo iz drugih ugovora i vanjskih transakcija i poziva. Ne mogu se pozivati unutar samog ugovora, osim korištenjem `this.functionName()`.

```
contract ExampleContract {
    function add(uint a, uint b) internal pure returns (uint) {
        return a + b;
    }

    function getCounter() public view returns (uint) {
        return counter;
    }

    function incrementCounter() private {
        counter += 1;
    }
}
```

Dodatni modifikatori funkcija:

- **pure**: Ukazuje da funkcija ne čita niti mijenja stanje ugovora. Ovisi samo o svojim ulaznim parametrima i ne pristupa nijednoj varijabli stanja niti poziva druge nepure funkcije.

```
function add(uint a, uint b) internal pure returns (uint) {
    return a + b;
}
```

- **view**: Ukazuje da funkcija ne mijenja stanje ugovora, ali može čitati iz njega. Funkcije modifikatora `view` obično se koriste za upite o stanju ugovora bez mijenjanja istog.

```
function getBalance(address user) public view returns (uint) {
    return balances[user];
}
```

- **payable**: Ukazuje da funkcija može primati Ether kao dio transakcije koja ju poziva. Kada je funkcija označena kao `payable`, može pristupiti svojstvu `msg.value` kako bi utvrdila količinu Ethera poslanu s transakcijom.

```
function deposit() public payable {
    require(msg.value > 0, "Nema poslanog Ethera");
    balances[msg.sender] += msg.value;
}
```

Solidity nudi nekoliko mehanizama za obradu pogrešaka poput `revert`, `require` i `assert`. Ti mehanizmi pomažu u provjeri unosa, uvjeta i rukovanju pogreškama. "`Revert`" se koristi za poništavanje transakcije kada se određeni uvjet ne ispuni. Poništava sve promjene načinjene na stanju i vraća preostalo gorivo pošiljatelju. Funkcija `require` slična je funkciji `revert`, ali kombinira provjeru uvjeta i akciju poništavanja u jednoj izjavi. Često se koristi za provjeru unosa, osiguravanje ispunjavanja preduvjeta funkcije ili provjeru dosljednosti stanja ugovora prije izvršenja funkcije. Funkcija `assert` se koristi za provjeru uvjeta koji nikada ne bi smjeli biti netočni, npr. unutarnje pogreške. Kada `assert` ne uspije, troši svo preostalo gorivo, a to se može smatrati oblikom kažnjavanja zbog dosezanja nevažećeg stanja. Trebalo bi ga koristiti rijetko i samo za vrlo kritične provjere.

```
contract ExampleContract {
   function setAge(uint _age) public {
        require(_age > 0 && _age < 150, "Neispravna starost");
        age = _age;
    }

    function withdraw(uint amount) public {
        if (balances[msg.sender] < amount) {
            revert("Nedovoljno sredstava");
        }
        balances[msg.sender] -= amount;
```

```
12          msg.sender.transfer(amount);
13      }
14
15      function divide(uint a, uint b) public pure returns (uint) {
16          assert(b != 0);
17          return a / b;
18      }
19  }
```

Solidity podržava nasljeđivanje ugovora, što omogućuje ugovorima nasljeđivanje svojstava i funkcija iz roditeljskih ugovora. Ova značajka omogućuje ponovnu upotrebu koda i stvaranje složenijih hijerarhija ugovora.

```
1  contract A {
2      function foo() public {}
3  }
4
5  contract B is A {
6      // Contract B inherits function foo() from contract A
7  }
```

Sučelja (eng. interfaces) su način definiranja vanjskog API-ja ugovora bez pružanja implementacije. Korisni su za definiranje potrebnih funkcija za ugovor kako bi mogao interaktirati s drugim ugovorima na blockchainu.

```
1  interface IExampleInterface {
2      function doSomething(uint value) external returns (uint);
3  }
4
5  contract ExampleContract is IExampleInterface {
6      function doSomething(uint value) external returns (uint) {
7          // Implementation of the function
8      }
9  }
```

Solidity ima nekoliko globalnih varijabli koje mogu poslužiti u radu i razvoju pametnih ugovora. Jedna od tih globalnih varijabli je `msg` koja pruža informacije o trenutačnoj transakciji ili pozivu funkcije. Osim `msg` globalne varijable, Solidity

pruža još nekoliko drugih globalnih varijabli koje su dostupne unutar pametnih ugovora. Te varijable nude korisne informacije o trenutačnom ugovoru, blockchainu ili okruženju transakcije. Vrijednosti globalnih varijabli koje su korištene pri razvoju pametnih ugovora u ovom radu su:

- `msg.sender`: Vraća adresu u vlasništvu vanjskog korisnika ili drugog pametnog ugovora koji je pokrenuo trenutačni poziv funkcije. Često se koristi za kontrolu pristupa ili praćenje podrijetla transakcije.
- `msg.value`: Vraća količinu Ethera/Matica u "Wei" (10^{-18} ETH/Matic) formatu vrijednosti koja je poslana zajedno s pozivom funkcije. Koristi se u funkcijama s označenim modifikatorom `payable` za dobavljanje vrijednosti poslane s transakcijom.
- `block.number`: Trenutačni broj bloka.
- `block.timestamp`: Vremenska oznaka (u sekundama od Unix epohe) u kojoj je trenutačni blok rudaren.
- `tx.origin`: Vraća adresu računa u vlasništvu vanjskog korisnika koji je pokrenuo izvornu transakciju. `tx.origin` je različit od `msg.sender` po tome što `msg.sender` upućuje na neposrednog pošiljatelja transakcije (koji je mogao biti na primjer drugi pametni ugovor), dok `tx.origin` upućuje na izvorni Externally Owned Account (EOA) koji je započeo prvu transakciju.

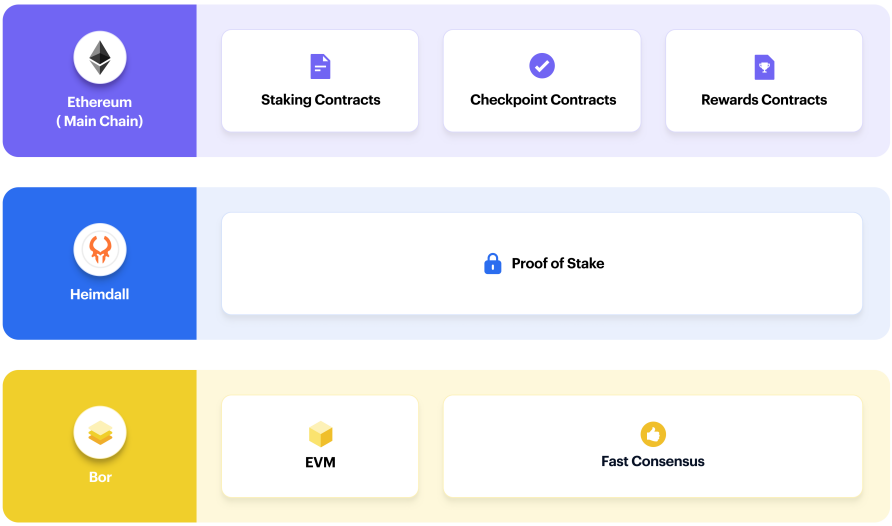
Ove komponente i značajke pružaju osnovni pogled na sintaksu i mogućnosti Soliditya za razvoj pametnih ugovora. Iskorištavanjem ovih alata i najboljim praksama, programeri mogu stvarati svestrane i sigurne pametne ugovore na EVM kompatibilnim blockchainovima.

4.3 Polygon

Ethereum, kao druga najveća blockchain platforma, pridonio je rastu i razvoju decentraliziranih aplikacija i pametnih ugovora. Međutim, problemi sa skalabilnošću Ethereuma doveli su do visokih naknada za transakcije i sporog protoka transakcija. Polygon, rješenje za skalabilnost drugog sloja, cilja riješiti ove izazove zadržavajući

sigurnost i decentralizaciju Ethereum blockchaina. U ovom poglavlju detaljno će se istražiti arhitektura, funkcionalnosti i primjene Polygon blockchaina.

Polygon, ranije poznat kao Matic Network, EVM kompatibilna je blockchain mreža drugog sloja osmišljena za omogućavanje skalabilnih i učinkovitih transakcija izvan glavnog Ethereum lanca. Arhitektura Polygona (slika 4.1) sastoji se od tri osnovna sloja: Bor, Heimdall i Ethereum sloja [21]. Hibridni pristup Polygona nudi rješenje za probleme skalabilnosti i uporabljivosti Ethereuma bez kompromisa u pogledu njegovih ključnih prednosti kao što su decentralizacija, sigurnost i interoperabilnost [21].



Slika 4.1 *Arhitektura Polygona* [21]

Ethereum služi kao temelj za Polygon arhitekturu. Sloj Ethereuma koristi pametne ugovore za olakšavanje funkcija ulaganja (eng. staking) i delegiranja osiguravajući integritet i sigurnost sustava. Omogućuje nesmetan prijenos imovine i podataka između Ethereum i Polygon mreža i djeluje kao čvorište za sve sporedne lance povezane s mrežom. Polygon koristi Ethereum za kontrolne točke, konačnost

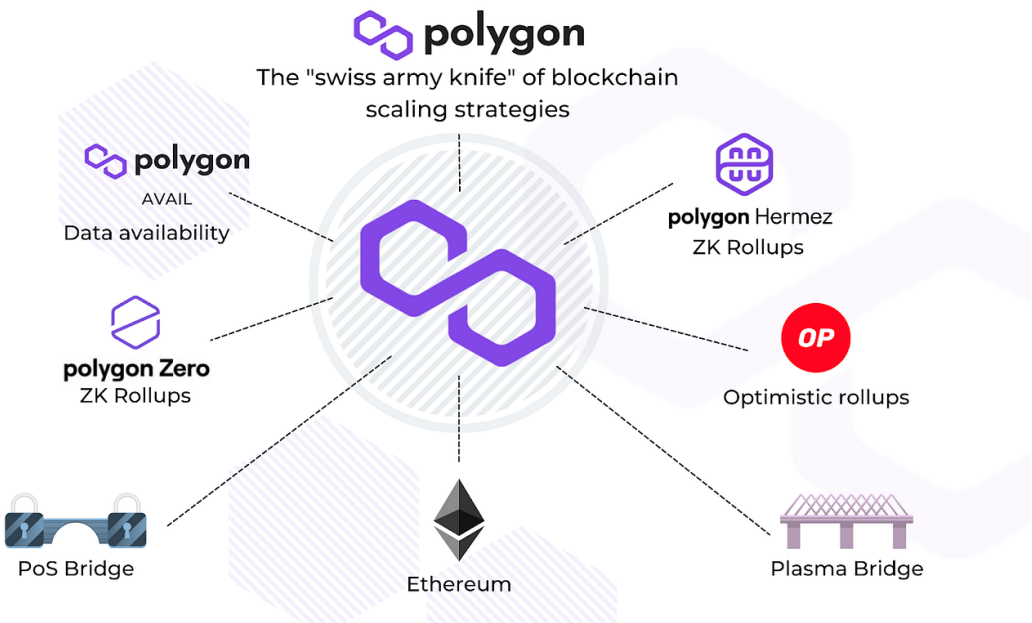
te za ulaganje i delegiranje pomoću pametnih ugovora [21].

Polygon za osiguranje svojih mreža koristi mehanizam konsenzusa Proof-of-Stake (PoS), poznat kao Heimdall verifikacijski sloj. Validatori ulažu svoje MATIC tokene kako bi sudjelovali u procesu konsenzusa. Biraju se za provjeru valjanosti transakcija, stvaranje novih blokova i predlaganje kontrolnih točaka na Ethereum glavnom lancu. Heimdall sloj se temelji na Tendermintu, Byzantine Fault Tolerance (BFT) konsenzus mehanizmu [21].

Bor sloj je odgovoran za proizvodnju blokova u Polygonu, za obradu transakcija i stvaranje novih blokova u Polygon dječjim lancima. Bor se temelji na Go Ethereumu (Geth) [22] s prilagođenim izmjenama u algoritmu konsenzusa. Proizvođači blokova su podmreža validatora. Validatori se periodički mijenjaju izborom iz skupine validatora na temelju njihovog udjela Matic tokena na Heimdallu unutar vremenskih razdoblja nazvanih "spanovi" [23].

Polygon SDK je modularni i proširivi okvir za izgradnju i implementaciju Ethereum kompatibilnih blockchainova. Pruža razvojnim inženjerima alate i resurse za stvaranje prilagođenih i samostalnih lanaca koji su povezani s Ethereum glavnim lancem putem Polygon arhitekture. Stoga, Polygon mreža sastoji se od više suverenih i međusobno povezanih blockchain mreža, isto poznatih kao "sidechains" (slika 4.2). Te mreže djeluju neovisno jedna o drugoj i mogu se prilagoditi specifičnim zahtjevima različitih aplikacija. Svaki lanac osiguran je vlastitim mehanizmom konsenzusa i skupom validatora što omogućava visok protok transakcija i smanjenu latenciju. Polygonova arhitektura podržava i PoS i Plasma sporedne lance. Ova hibridna konstrukcija omogućava razvojnim inženjerima korištenje PoS-a, Plasme ili oboje, ovisno o njihovim specifičnim potrebama [24].

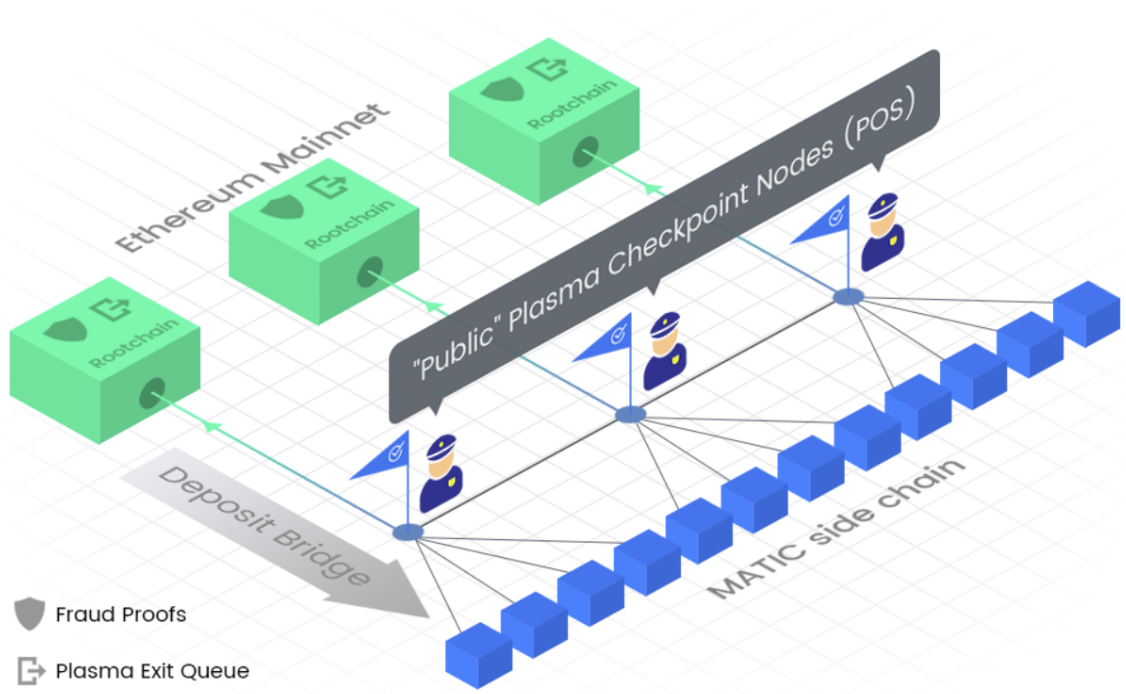
Polygon omogućuje nesmetanu komunikaciju između svojih dječjih lanaca i Ethereum glavnog lanca kroz upotrebu mostova koji omogućuju prijenos tokena, imovine i podataka između različitih lanaca (slika 4.3). Neki od tih mostova su Plasma most, PoS most i prilagođeni mostovi za specifične upotrebe. Mostovi koriste skup pametnih ugovora za zaključavanje imovine na Ethereum blockchainu i izdavanje odgovarajućih tokena na Polygon mreži. Kada korisnici žele premjestiti svoju imovinu natrag na Ethereum, postupak se preokreće i tokeni na Polygon mreži se uništavaju.



Slika 4.2 *Sporedni lanci i mostovi Polygona* [24]

Polygon pruža poznato i jednostavno razvojno okruženje jer nudi potpunu kompatibilnost s alatima, jezicima i standardima Ethereuma (npr. Solidity, Hardhat, MetaMask), čineći ga lakim za premještanje postojećih ili stvaranje novih decentraliziranih aplikacija na Polygon mreži. EVM omogućava kompatibilnost s Ethereum baziranim decentraliziranim aplikacijama i protokolima.

Korisnici mogu sudjelovati u ekosustavu Polygona tako što će uložiti svoje Matic tokene kao validatori ili delegatori. Validatori su odgovorni za provjeru valjanosti transakcija, predlaganje novih blokova i održavanje sigurnosti mreže. Zauzvrat za svoje usluge validatori zarađuju nagrade u obliku Matic tokena. Delegatori, s druge strane, mogu delegirati svoje tokene validatorima, dijeleći odgovornost i nagrade za osiguranje mreže. Delegiranjem svojih tokena korisnici mogu zaraditi dio nagrada validatora bez aktivnog sudjelovanja u procesu konsenzusa. U slučaju da se ulog validatora presječe (eng. slash) zbog njegovih postupaka, delegator koji je uložio svoje Matic tokene kod tog validatora neće izgubiti svoje tokene kao i validator. Jedino može dobivati smanjene nagrade.



Slika 4.3 *Most za komunikaciju između Polygona i Ethereuma* [21]

Proces prijenosa podataka između Ethereum i Polygon mreža olakšava skup oraclea kao mehanizmi sinkronizacije stanja. Ti oraclei prate promjene stanja na Ethereum mreži i podnose relevantne podatke Polygon mreži. Podaci se zatim mogu koristiti u pametnim ugovorima i decentraliziranim aplikacijama za interakciju s Ethereum protokolima i aplikacijama.

Polygon je korišten u ovom projektu zbog svoje interoperabilnosti, kompatibilnosti s Ethereumom i skalabilnosti koja pruža visok protok transakcija s niskim kašnjenjem što ga čini idealnom platformom za DeFi aplikacije koje zahtijevaju velik broj brzih i sigurnih transakcija po niskim troškovima. Kako bi se poticale veće donacije, kao jedna od značajki Web3 platforme za donacije je nagrađivanje donatora NFT-ovima. NFT-ovi predstavljaju jedinstvenu digitalnu imovinu te su posljednjih godina postali znatno popularni za skupljanje. Popularnost NFT-ova je također jednim djelom odgovorna za visoke naknade transakcija na Ethereum mreži što ograničava mintanje i distribuiranje NFT-ova. Niske naknade i brze transakcije Polygona

omogućavaju pristupačnije i skalabilnije okruženje za projekte koji uključuju NFT-e. Najbitniji razlog zašto je odabrana Polygon mreža za ovaj projekt je niska naknada transakcija pošto je u pametnim ugovorima jedna od najskupljih operacija spremanje podataka na blockchain, odnosno pisanje u `storage` mjesto pohrane varijable, a pametan ugovor u ovom slučaju mora spremati korisnički izrađene donacijske kampanje i njihove informacije na blockchain.

Jedinstveni pristup Polygona rješavanju problema skalabilnosti i uporabljivosti Ethereuma pozicionirao ga je kao ključno rješenje drugog sloja u blockchain ekosustavu. Kombinirajući višelančanu arhitekturu s robusnim skupom alata za programere i infrastrukture, Polygon omogućuje skalabilne, sigurne i učinkovite transakcije izvan glavnog lanca održavajući kompatibilnost s Ethereum mrežom. Širok spektar primjena i upotreba platforme, od DeFi-a i NFT-ova do igrica i poslovnih rješenja, dodatno pokazuje njen potencijal za poticanje rasta i usvajanje tehnologije blockchaina.

4.4 NFT

NFT ili Non-Fungible Token, jedinstvena je digitalna imovina koja predstavlja vlasništvo, porijeklo ili autentičnost određenog predmeta ili sadržaja na blockchainu. NFT-ovi su nedjeljivi i ne mogu se zamjenjivati u omjeru jedan naprema jedan kao što mogu kriptovalute poput Bitcoina ili Ethera. Svaki NFT obično posjeduje različite atribute koji ga čine jedinstvenim i vrijednim. Ovi tokeni mogu predstavljati umjetničko djelo, kolekcionarske stvari, virtualnu metaverse nekretninu, sadržaj u igrici, muziku ili bilo koji drugi oblik digitalne pa čak i fizičke imovine.

ERC-721 standard za Non-Fungible Token je široko korišteni standard pametnih ugovora na EVM kompatibilnim blockchainovima koji omogućava stvaranje, upravljanje i razmjenu NFT-ova. Pruža skup standardiziranih funkcija i događaja (eng. events) koje programeri mogu koristiti za izgradnju tržišnih platforma i ostalih aplikacija za NFT-ove. ERC-721 standard osigurava interoperabilnost između različitih platformi što olakšava korisnicima upravljanje i trgovanje NFT-ovima.

Skup ključnih komponenti, funkcija i događaja koje ERC-721 standard specificira

da usklađeni pametni ugovor mora implementirati su sljedeći:

Vlasništvo i upravljanje tokenima:

- `balanceOf()`: Prikazuje broj NFT-ova u vlasništvu određene adrese.
- `ownerOf()`: Prikazuje adresu vlasnika za određeni token ID.
- `safeTransferFrom()`: Prenosi NFT s jedne adrese na drugu, osiguravajući da adresa primatelja može sigurno rukovati tokenom.
- `transferFrom()`: Prenosi NFT s jedne adrese na drugu bez provjere kompatibilnosti adrese primatelja.

Mehanizmi odobravanja:

- `approve()`: Omogućuje određenoj adresi upravljanje NFT-om u ime vlasnika. To omogućuje aplikacijama treće strane olakšavanje prijenosa NFT-ova bez preuzimanja potpunog vlasništva.
- `getApproved()`: Prikazuje odobrenu adresu za određeni token ID, ako postoji.
- `setApprovalForAll()`: Omogućuje vlasniku odobravanje ili opoziv dozvole za operatora (adresu) za upravljanje svim njihovim NFT-ovima.
- `isApprovedForAll()`: Provjerava je li operator (adresa) odobren za upravljanje NFT-ovima određenog vlasnika.

Metapodaci i prebrojavanje (neobavezno):

- `tokenURI()`: Prikazuje URI koji upućuje na metapodatke određenog token ID-a. Metapodaci mogu uključivati informacije poput imena tokena, opisa, atributa i slike.
- `totalSupply()`: Prikazuje ukupan broj NFT-ova izdanih od strane ugovora (ako ugovor podržava prebrojavanje).
- `tokenByIndex()`: Prikazuje token ID za određeni indeks (ako ugovor podržava prebrojavanje).
- `tokenOfOwnerByIndex()`: Prikazuje token ID za određeni indeks tokena u vlasništvu određene adrese (ako ugovor podržava prebrojavanje).

Događaji:

- `Transfer()`: Emitira se kada se NFT prenese s jedne adrese na drugu.
- `Approval()`: Emitira se kada adresa dobije odobrenje za upravljanje NFT-om u ime vlasnika.
- `ApprovalForAll()`: Emitira se kada se operatoru odobri ili opozove dozvola za upravljanje svim NFT-ovima vlasnika.

ERC-721 standard za Non-Fungible Token pruža snažan i fleksibilan okvir za stvaranje i upravljanje NFT-ovima na Ethereum i Polygon blockchainu. Osigurava interoperabilnost i dosljedno iskustvo za korisnike što olakšava razvoj aplikacija i platformi koje podržavaju NFT-ove.

4.5 React

React je JavaScript knjižnica otvorenog koda namijenjena za izgradnju jednostraničnih (eng. single-page) web korisničkih sučelja. Razvijena je od strane Facebooka 2013. godine i održavana je također uz Facebook i velikom open source zajednicom. Danas je React jedna od najčešće korištenih knjižnica za razvoj web sučelja. Glavni cilj Reacta je pojednostaviti razvoj brzih, skalabilnih i održivih aplikacija razdvajanjem User interface (UI) elemenata na ponovno upotrebljive manje komponente koje ažurira samo ako dođe do promjena u komponentama što ga čini vrlo efikasnim u radu. U nastavku su objašnjeni neki od temeljnih koncepata i načina rada Reacta.

Sva React sučelja izrađena su od komponenata te su stoga one temelj UI-a u Reactu. Komponenta predstavlja samostalan i ponovno upotrebljiv dio korisničkog sučelja koji se piše u JavaScript kodu, a za HyperText Markup Language (HTML) i Cascading Style Sheets (CSS) se koristi JavaScript XML (JSX) sintaksa. JSX je proširenje sintakse za JavaScript te ona opisuje strukturu i izgled komponenti. JSX omogućuje pisanje oznaka (poput `<div>`-a) izravno unutar JavaScript koda. Komponente mogu biti jednostavne, poput gumba, ili složenije poput obrasca ili čitave stranice. Kombiniranjem komponenti mogu se stvoriti složeni UI-ovi.

React koristi virtualni Document Object Model (DOM) kako bi poboljšao per-

formanse i učinio ažuriranje sučelja učinkovitijim. Virtualni DOM je reprezentacija stvarnog DOM-a u memoriji, a React ga koristi za praćenje promjena u stablu komponenata. Kada se stanje komponente promijeni, React stvara novi virtualni DOM i provjerava razliku između stare i nove verzije. Ovaj postupak usporedbe omogućuje Reactu da identificira najmanji skup potrebnih promjena za ažuriranje stvarnog DOM-a minimizirajući ponovno iscrtavanje komponenata što poboljšava performanse.

React komponente upravljaju vlastitim podacima putem stanja (eng. state) i svojstava (eng. props). Stanje predstavlja unutarnje podatke koje komponenta sadrži te može njima upravljati i ažurirati ih. Svojstva se koriste za prenošenje podataka iz roditeljske komponente u dječju komponentu. Kada se stanje ili svojstva komponente promijene React automatski ponovno iscrtava komponente na koje se odnose promjene i ažurira sučelje.

React ekosustav sastoji se od brojnih open source knjižnica i alata React zajednice koje nadopunjuju React i ubrzavaju razvoj složenih aplikacija. Jedna od najpopularnijih i najčešće korištenih knjižnica u React ekosustavu je React Router. React Router je knjižnica za upravljanje klijentskom navigacijom u React aplikacijama. Ona mogućuje izradu dinamičkih ruta i upravljanje navigacijskim stanjem što olakšava izgradnju jednostraničnih aplikacija (eng. single-page application (SPA)) s više stranica i navigacijskim strukturama. Nažalost, nisu sve knjižnice i alati jednako kvalitetni zbog prestanka održavanja istih ili nekompatibilnosti s ostalim korištenim tehnologijama. Zbog toga mogu napraviti više problema nego ih riješiti kao što je u ovom radu slučaj s "Uniswap widget" knjižnicom.

U konačnici React je snažna JavaScript knjižnica namijenjena izgradnji korisničkih sučelja pomoću ponovno upotrebljivih komponenti. Njegovi temeljni koncepti koji doprinose njegovoj izvedbi, skalabilnosti i održivosti uključuju komponente, JSX, virtualni DOM, stanje i svojstva te jednosmjerni protok podataka. Programeri koji koriste React mogu s lakoćom i pouzdanošću raditi složene i responzivne aplikacije. React ekosustav je bogat knjižnicama i alatima koji se prilagođavaju različitim potrebama programera te znatno olakšavaju i ubrzavaju izgradnju sofisticiranih i skalabilnih aplikacija. Od upravljanja stanjem aplikacije do stilizacije komponenti i izrade korisničkih sučelja.

4.6 Tailwind

Tailwind CSS je popularan CSS okvir s fokusom na gotovim predefiniranim klasama koje se koriste za pojednostavljenje procesa stiliziranja i izrade responzivnih web aplikacija. U nastavku će se istražiti glavne značajke, prednosti i izazovi upotrebe Tailwind CSS-a.

Glavni cilj okvira je omogućiti brzu izradu prototipova i stiliziranje s minimalnim prilagođenim CSS-om, pružajući programerima sveobuhvatan skup korisničkih (eng. utility) klasa koje se lako mogu primijeniti na HTML elemente. Pristup upotrebe malih, jednostranih korisničkih klasa koje se mogu kombinirati kako bi se stvorile složene i fleksibilne UI komponente razlikuje se od tradicionalnih CSS metodologija kao što su Block-Element-Modifier (BEM) i Object-Oriented CSS (OOCSS), gdje programeri stvaraju prilagođene CSS klase za pojedinačne komponente. Koristeći korisničke klase programeri mogu napraviti širok spektar dizajna bez pisanja prilagođenog CSS-a što rezultira bržim razvojnim ciklusima, manjim problemima s preopterećenjem stilova i većom održivošću.

Tailwind CSS je građen s obzirom na responzivnost. Okvir nudi unaprijed definirani skup točaka prekida koje se mogu koristiti za stvaranje responzivnog dizajna. Programeri mogu primijeniti korisničke klase s odgovarajućim prefiksima točaka prekida kako bi prilagodili stil elementa različitim veličinama ekrana. U sljedećem kodu je kratak primjer korištenja Tailwind CSS-a:

```
<div class="sm:bg-blue-500 md:bg-green-500 lg:bg-red-500">
  Ostale komponente i elementi
</div>
```

U ovom kodu boja pozadine `div`-a mijenja se ovisno o veličini ekrana pomoću Tailwind CSS predefiniranih klasa. Na malim ekranima imat će plavu pozadinu, na srednjim ekranima zelenu pozadinu, a na velikim ekranima crvenu pozadinu.

Tailwind CSS nudi globalne opcije prilagodbe putem svoje konfiguracijske datoteke `tailwind.config.js`. Programeri mogu izmijeniti zadanu temu, dodati prilagođene korisničke klase te čak proširiti postojeće klase. Ova fleksibilnost omogućuje okviru da se prilagodi različitim zahtjevima projekata bez narušavanja njegovog te-

meljnog pristupa usmjerenog na korisničke klase.

Kako bi se minimizirala veličina konačnog CSS paketa, Tailwind CSS integrira se s PurgeCSS-om. Ovaj alat automatski uklanja neiskorištene CSS klase iz produkcijske verzije što rezultira manjim veličinama datoteka i bržim vremenima učitavanja web aplikacija.

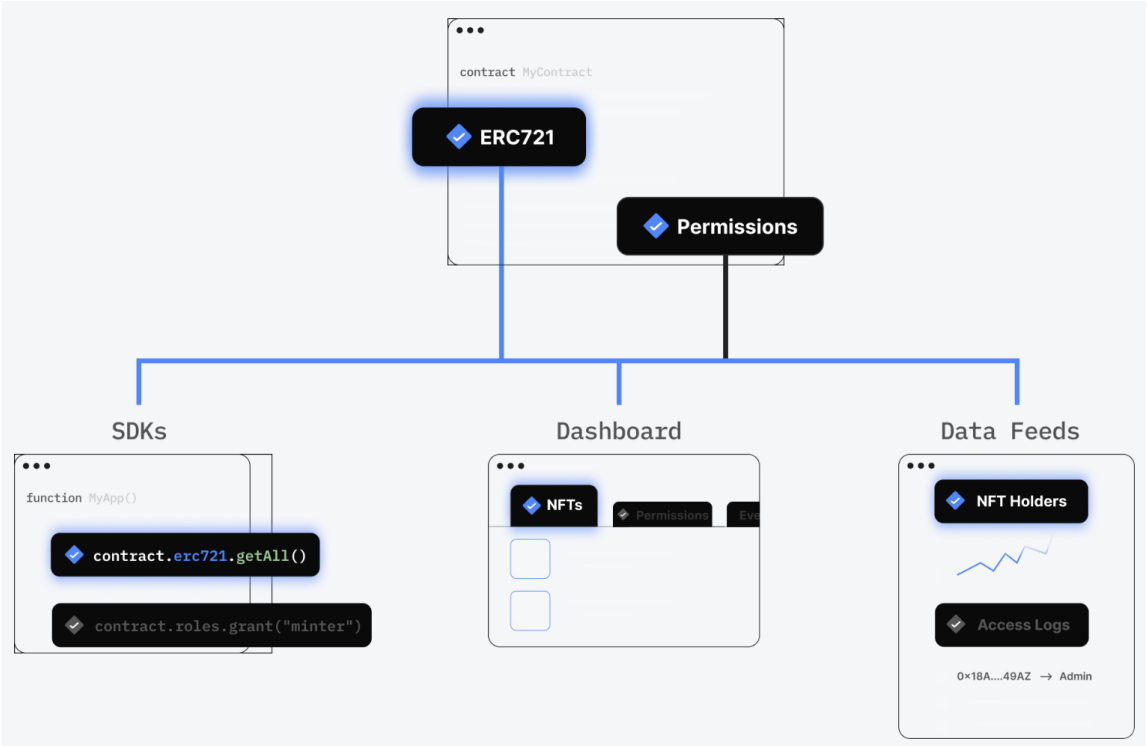
U nekim slučajevima kompleksnijeg detaljnog stiliziranja pristup usmjeren na korisničke klase može dovesti do dugog HTML zapisa jer je za postizanje željenog stiliziranja potrebno mnogo korisničkih klasa. Ova detaljnost može smanjiti čitljivost i povećati složenost oznaka, posebno za neiskusne programere ili one koji nisu upoznati sa sintaksom Tailwind CSS-a.

Tailwind CSS, kao CSS okvir s fokusom na korisničke klase, nudi jedinstven i snažan pristup stiliziranju web aplikacija. Pružajući sveobuhvatan skup unaprijed definiranih korisničkih klasa, okvir omogućava brzi razvoj prototipova, bolju održivost i poboljšanu čitljivost. Sve u svemu, Tailwind CSS je privlačna opcija za programere koji žele pojednostaviti svoj proces stiliziranja, pridržavajući se modernih najboljih praksi.

4.7 ThirdWeb

ThirdWeb je razvojni okvir za web3 aplikacije koji pojednostavljuje proces izrade decentraliziranih aplikacija na blockchain platformama. Web3 se odnosi na sljedeću generaciju Interneta gdje decentralizirane tehnologije poput blockchaina i pametnih ugovora igraju ključnu ulogu u omogućavanju P2P transakcija, vlasništva nad podacima i povjerenju. Razvojni okvir ThirdWeba pruža niz alata i knjižnica koje apstrahiraju veći dio složenosti povezane s izgradnjom decentraliziranih aplikacija čime se ubrzava stvaranje sigurnih i skalabilnih decentraliziranih aplikacija. Platforma nudi alate za ubrzavanje razvojnog procesa koji uključuju unaprijed izgrađene ugovore za najuobičajenije slučajeve upotrebe, SDK-ove za različite programske jezike i intuitivnu nadzornu ploču za upravljanje postavkama ugovora, dozvolama tima, tokovima prihoda i analitikom (slika 4.4).

Ključni alati ThirdWeba objašnjeni su u nastavku [26].

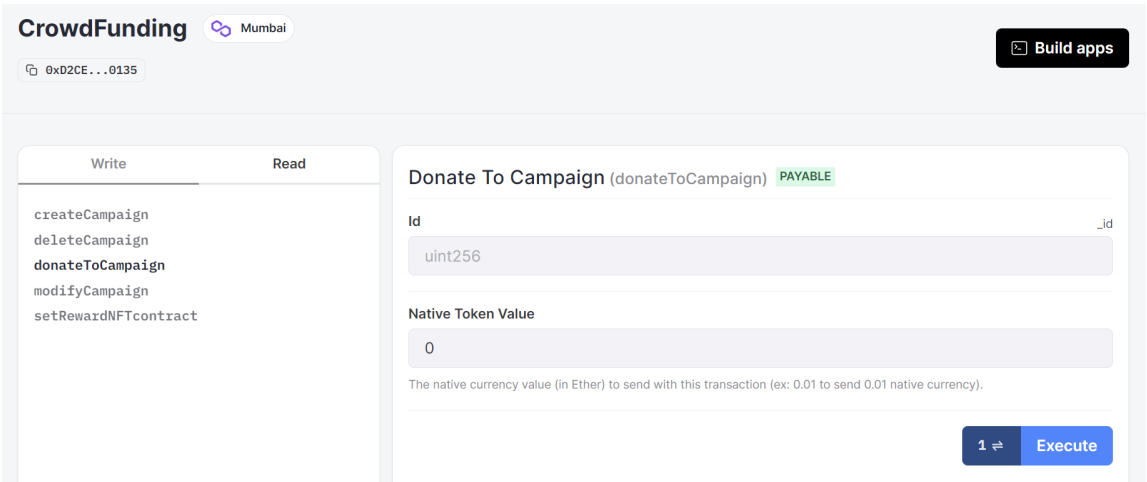


Slika 4.4 *Pregled ThirdWeb platforme* [25]

Kako bi se osigurala kompatibilnost s različitim programskim jezicima i okvirima, ThirdWeb nudi SDK-ove i komponente za povezivanje s korisničkim sučeljem koje pružaju interakciju s pametnim ugovorima i elegantan dizajn. Jedna od tih komponenta na primjer bila bi za autentifikaciju korisnika pomoću web3 novčanika kriptovaluta, poput MetaMaska, čime je omogućeno sigurno upravljanje decentraliziranom aplikacijom. Ono što se najviše koristilo u izradi ovog rada bile su React komponente i kuke (eng. hooks) koje su olakšale integraciju s front-end knjižnicama i okvirima što je u konačnici rezultiralo dobrim korisničkim iskustvom.

ThirdWeb je kompatibilan s više blockchain platformi čime se programerima pruža fleksibilnost u odabiru najbolje platforme za njihove decentralizirane aplikacije i jednostavan prijelaz između blockchainova ako je potrebno.

Platforma također nudi nadzornu ploču za svaki postavljeni pametni ugovor koja omogućuje jednostavno upravljanje administrativnim operacijama poput izdavanja

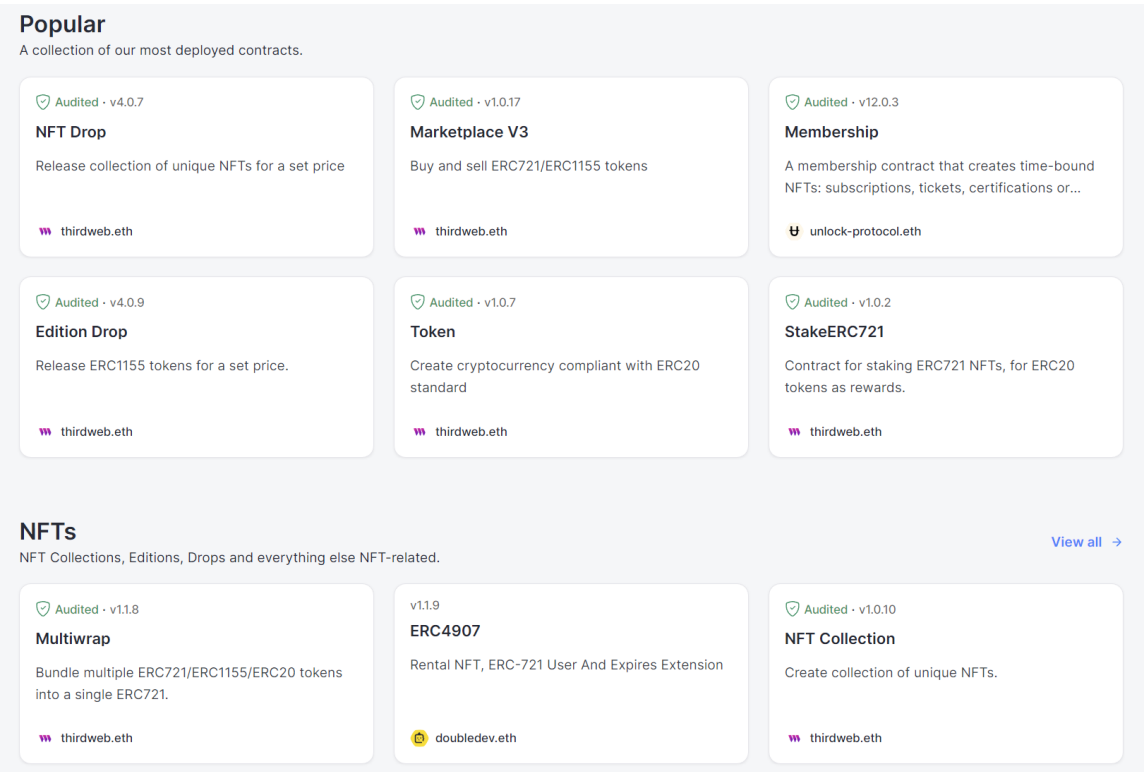


Slika 4.5 *Izgled ThirdWeb web korisničkog sučelja za jednostavno upravljanje pametnim ugovorima*

dozvola za timove i analiziranja uvida iz pametnih ugovora (ove mogućnosti se otključaju s plaćenom verzijom). Nadzorna ploča omogućuje i interakcije s pametnim ugovorima na blockchainu preko web korisničkog sučelja, testiranja (slika 4.5) i automatizirano postavljanje pametnih ugovorima što smanjuje vjerojatnost pogrešaka. Nažalost nisu sve navedene mogućnosti nadzorne ploče besplatne za korištenje.

Temelj bilo koje web3 aplikacije je implementacija pametnih ugovora, a ThirdWeb nudi unaprijed izrađene ugovore i Solidity SDK za izradu prilagođenih ugovora. Unaprijed izrađeni ugovori dostupni su za NFT-ove, tokene pa čak i tržišta za iste. Otvorenog su koda, redovito provjeravani, učinkoviti i u vlasništvu korisnika nakon što se postave jednim klikom (slika 4.6). Ti već izrađeni pametni ugovori imaju vlastite dokumentacije za korištenje i revizije (eng. audit).

Sve u svemu, razvojni okvir ThirdWeb pojednostavljuje proces stvaranja decentraliziranih aplikacija omogućujući da se više usredotočenosti preda izgradnji inovativnih i korisnički prijateljskih (eng. user-friendly) aplikacija dok se većina temeljne složenosti povezane s blockchain tehnologijom apstrahira. Modularna arhitektura ThirdWeba omogućuje odabir samo komponenata koje su potrebne čime se olakšava proširenje funkcionalnosti i prilagođavanje prema specifičnim zahtjevima.



Slika 4.6 *Dio popisa gotovih pametnih ugovora za postavljanje na blockchain* [27]

4.8 IPFS

IPFS ili InterPlanetary File System je decentralizirani protokol za pohranu i dijeljenje podataka putem distribuirane P2P mreže računala. Cilj mu je učiniti web sigurnijim i otvorenijim zamjenom tradicionalnog klijent-server modela, distribuiranom arhitekturom. IPFS koristi adresiranje sadržaja što znači da se podaci adresiraju prema njihovom kriptografskom sažetku (hashu) umjesto prema njihovom fizičkom položaju na određenom poslužitelju. To osigurava da su podaci sigurni, provjerljivi i otporni na cenzuru. U nastavku će se pobliže opisati način rada IPFS-a [28].

Za razliku od tradicionalnih sustava koji koriste adresiranje na temelju lokacije, IPFS koristi adresiranje na temelju sadržaja. Svaka datoteka i njeni dijelovi dobivaju jedinstveni kriptografski identifikator sadržaja (Content Identifier (CID)). To osigurava sigurnost sadržaja jer bi manipulacija sadržajem promijenila identifikator

sadržaja.

IPFS koristi distribuiranu pohranu ključeva i vrijednosti za praćenje koji čvorovi pohranjuju određeni sadržaj. Kada se doda datoteku u IPFS, ona se podijeli na blokove i distribuira među čvorovima u mreži. Distributed hash table (DHT) pomaže pronaći te blokove kada je to potrebno [29].

Bitswap je glavni protokol za razmjenu podataka koji IPFS čvorovi koriste za zahtjev i razmjenu podatkovnih blokova. Kada čvor želi dohvatiti datoteku, šalje zahtjeve za potrebnim blokovima svojim povezanim čvorovima. Ako povezani čvorovi nemaju blokove, proslijedit će zahtjev svojim povezanim čvorovima i tako dalje dok se blokovi ne pronađu [29].

IPFS organizira podatke u MerkleDAG strukturi usmjerenog acikličkog grafa. Svaki čvor u MerkleDAG-u predstavlja podatkovni blok, a čvorovi su povezani pomoću kriptografskog hasha sadržaja. Budući da će identičan sadržaj proizvesti isti hash, struktura MerkleDAG-a osigurava da se pohranjuju samo jedinstveni dijelovi sadržaja, a redundantni podaci se uklanjaju iz sustava. Dakle MerkleDAG povezuje čvorove pomoću sažetaka, omogućujući učinkovitu i sigurnu distribuciju i dohvat podataka.

Dodavanje datoteka u IPFS se odvija u sljedećim koracima:

1. **Razbijanje datoteka**: Kada se datoteka doda u IPFS, podijeli se na manje dijelove (blokove).
2. **Hashiranje**: Svaki blok se zatim hashira stvarajući jedinstveni identifikator (CID) za svaki blok.
3. **Distribucija blokova**: Hashirani blokovi se distribuiraju po IPFS mreži pri čemu svaki čvor pohranjuje kopije blokova koje ga zanimaju.

Dohvaćanje datoteka s IPFS-a:

1. **Zahtjev za blokovima**: Kada korisnik želi pristupiti datoteci, koristi hash (CID) datoteke za zahtjev blokova iz mreže.
2. **Lociranje blokova**: DHT pomaže u lociranju čvorova koji imaju tražene blokove.

3. **Razmjena blokova**: Protokol Bitswap omogućuje razmjenu blokova između čvorova.
4. **Sastavljanje datoteke**: Nakon što su svi blokovi primljeni, datoteka se ponovno sastavlja i prikazuje korisniku.

IPFS čvorovi koriste razne metode za pronalaženje i povezivanje s drugim čvorovima u mreži kao što su Multicast Domain Name System (mDNS), početni čvorovi i ručne veze s čvorovima (eng. manual peer connections). Kada se povežu, čvorovi mogu razmjenjivati podatke i sudjelovati u DHT-u. Transport Layer Security (TLS) je korišten za osiguranje veza između čvorova te jamči integritet i privatnost podataka.

Zahvaljujući svojoj prirodi adresiranja sadržaja, IPFS podržava verzioniranje. Budući da će svaka verzija datoteke imati jedinstveni hash korisnici lako mogu referencirati i dohvaćati prethodne verzije datoteke.

Ukratko, IPFS je decentralizirani, adresabilni sustav datoteka na temelju sadržaja koji kombinira razne koncepte i protokole kako bi stvorio učinkovitiji, sigurniji i otvoreniji web. To postiže korištenjem distribuirane hash tablice, adresiranjem sadržaja, razmjenom blokova, MerkleDAG-om, otkrivanjem čvorova i TLS-om.

4.8.1 FileCoin

Filecoin, s druge strane, je decentralizirana mreža za pohranu podataka izgrađena na vrhu IPFS-a. Koristi tržište temeljeno na blockchainu kako bi se potaknuli korisnici da dijele svoj prostor za pohranu podataka s IPFS mrežom. U nastavku je opisano kako Filecoin funkcionira u kombinaciji s IPFS-om.

Da bi pohranio svoje podatke na Filecoin mreži, korisnik stvara ugovor o pohrani što je ugovor između korisnika i pružatelja usluge pohrane (rudar na Filecoin mreži). Ugovor navodi uvjete pohrane uključujući količinu prostora za pohranu, trajanje pohrane i cijenu koju je korisnik spreman platiti izraženu u attoFIL kriptovaluti (1 attoFIL iznosi 10^{-18} FIL) [30]. Pružatelj usluge prihvaća ugovor, pohranjuje podatke korisnika i prima Filecoin tokene (FIL) kao plaću. Podaci se zatim dodaju na IPFS što uključuje postupak razbijanja podataka na manje dijelove, tzv. "blokove", te sva-

kom bloku dodjeljuje jedinstveni identifikator sadržaja (CID) temeljen na njegovom kriptografskom sažetku. Podaci se potom šifriraju i pohranjuju na više čvorova u IPFS mreži [31].

Kada korisnik želi dohvatiti svoje podatke koji su pohranjeni na Filecoin mreži, identificiraju se rudari koji posjeduju te podatke i od njih se zahtijeva ponuda za dohvaćanje. Ponuda za dohvat podataka sadrži cijenu po bajtu, cijenu otključavanja i interval plaćanja. Zatim se locirani podaci na distribuiranim čvorovima ponovno sastavljaju i šalju korisniku do trenutka kada se dosegne određeni prag. Tada je potrebno platiti Filecoin tokene da bi rudar za dohvaćanje preuzeo ostatak podataka od pružatelja usluge pohrane i isporučio ih korisniku [30].

Kako bi se osiguralo da pružatelji usluge pohrane zaista pohranjuju podatke korisnika i održavaju njihov integritet, Filecoin koristi dva kriptografska dokaza: dokaz o replikaciji (eng. Proof of Replication (PoRep)) i dokaz o prostor-vremenu (eng. Proof of Spacetime (PoSt)). PoRep pokazuje da je pružatelj usluge jedinstveno pohranio podatke, dok PoSt dokazuje da su podaci pohranjeni tijekom određenog razdoblja. Pružatelji usluge moraju podnijeti ove dokaze Filecoin mreži kako bi primili svoje nagrade u FIL tokenima i ne bi bili kažnjeni [32].

Sada vjerojatno razmišljate zašto bi netko plaćao Filecoin tokene za pohranu i dohvat podataka s IPFS-a umjesto da koristi normalnu IPFS mrežu. Dok IPFS omogućuje korisnicima pohranu i dijeljenje podataka na dobrovoljnoj osnovi, Filecoin nudi održivije i pouzdanije rješenje za pohranu potičući sudionike.

Nekoliko razloga zašto bi netko odabrao Filecoin umjesto normalnog IPFS-a za pohranu i dohvaćanje podataka:

1. **Model poticaja**: Filecoin ima ugrađeni ekonomski model koji nagrađuje pružatelje pohrane (rudare) s Filecoin tokenima (FIL) za doprinos prostora za pohranu i propusnost mreže. To potiče veći broj ljudi da ponudi pohranu što rezultira robusnijom i pouzdanijom mrežom.
2. **Pouzdanost**: IPFS se oslanja na dobrovoljne čvorove za pohranu i dijeljenje podataka što može dovesti do nedostupnosti podataka ako su čvorovi koji pohranjuju podatke izvan mreže. Nasuprot tome, Filecoin koristi pametne ugovore kako bi stvorio sporazume između klijenata i pružatelja pohrane što

osigurava da se podaci pohranjuju i održavaju tijekom određenog razdoblja.

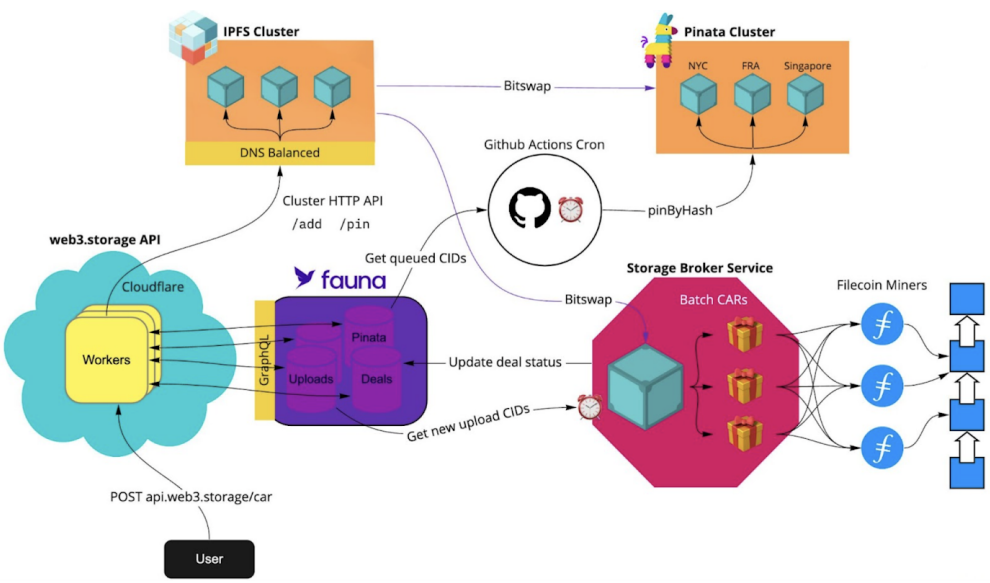
3. **Perzistencija podataka**: Na IPFS-u podaci se možda neće trajno pohranjivati ako ih čvorovi koji poslužuju podatke odluče ukloniti ili izaći iz mreže. Filecoin osigurava perzistenciju podataka uspostavljanjem sporazuma između klijenata i pružatelja pohrane koji uključuju kazne za pružatelje koji ne ispunjavaju svoje obveze pohrane.
4. **Kvaliteta usluge**: Filecoin omogućuje korisnicima odabir pružatelja pohrane na temelju njihovog ugleda, cijene i drugih čimbenika. To omogućuje klijentima da odaberu pružatelje koji udovoljavaju njihovim specifičnim zahtjevima za performanse, pouzdanost i troškove.
5. **Tržište dohvaćanja**: Filecoin također ima tržište za dohvaćanje podataka, na kojem klijenti plaćaju rudarima za dohvaćanje i preuzimanje podataka od pružatelja pohrane. To stvara konkurentno tržište za dohvaćanje podataka što može rezultirati bržem i učinkovitijem pristupu podacima.
6. **Konsenzus za potvdu podataka**: Filecoin koristi algoritam konsenzusa zvan Expected Consensus (EC) kako bi se dogovorio o stanju mreže i distribuirao nagrade [33]. Rudari zarađuju nagrade na temelju svojih kapaciteta za pohranu i količine podataka koje pohranjuju. Kako pružaju više prostora za pohranu i dokazuju da održavaju podatke, povećavaju svoje šanse za dobivanje prava na dodavanje novih blokova u Filecoin blockchain i primanje nagrada za blokove.

IPFS pruža decentraliziranu osnovu za pohranu i dijeljenje podataka s adresiranjem sadržaja, dok Filecoin proširuje tu osnovu stvaranjem poticajnog tržišta za pohranu temeljenog na blockchainu. Ova kombinacija omogućuje sigurnije i otpornije rješenje za pohranu u usporedbi s tradicionalnim klijent-server arhitekturama. Filecoin je dodatak IPFS-u tako što pruža način dokazivanja da korisnik čuva ono što tvrdi da čuva što bi ga učinilo validacijskim dijelom IPFS-a.

4.8.2 Web3.Storage

Web3.storage kombinira snage IPFS-a i Filecoina kako bi pružio besprijekorno i korisnički prijateljsko iskustvo pohrane. Korisnici mogu prenositi svoje podatke na Web3.storage koji se pobrine o distribuciji istih preko IPFS mreže i osiguravanju njihove dostupnosti. Platforma zatim koristi Filecoin mrežu kako bi potaknula pružatelje pohrane da održavaju podatke, stvarajući čvrst i decentraliziran ekosustav pohrane. Web3.storage pruža skalabilnost i performanse koje do sada nisu viđene u IPFS rješenjima.

Kao što je prikazano na slici 4.7, sadržaj koji korisnici šalju u Web3.storage trajno se pohranjuje na mreži pružatelja pohrane na Filecoinu. Također je višestruko prikvačen (eng. pinned) na IPFS-u. Dakle, zajedničke snage Filecoina i IPFS-a pružaju adresabilnost sadržaja što je osigurano putem CID-ova i postojanost putem ekonomskog modela Filecoina. U osnovi, korisnici Web3.storagea mogu se osloniti na provjerljive dokaze o integritetu pohranjenih podataka.



Slika 4.7 *Rad Web3.storage rješenja* [34]

Podaci se pohranjuju tako što Web3.storage šalje podatke na IPFS klaster. IPFS klaster se sastoji od tri čvora Protocol Labsa koji su geografski raspoređeni gdje

sustav stavlja podatke u red za pohranu na Filecoin mreži. Prije nego što protokol pohrani te podatke, sustav ih pakira s drugim podacima u Filecoin dogovoru. Naposljetku, automatski mehanizam pohranjuje paket podataka koristeći najmanje pet geografski raspoređenih rudara (eng. miners). Ipak za dodatnu redundanciju i dostupnost podaci se također prikvačuju na druge IPFS pinning usluge kao što je Pinata [34]. Cijeli taj postupak se automatski izvršava. Sve što se treba savladati je intuitivno sučelje Web3.storagea. Podaci se mogu pohranjivati i putem JavaScript klijentske knjižnice u nekoliko linija koda. Da bi se tako pohranili potrebno je prije izgenerirati Web3.storage API token stvaranjem korisničkog računa i zatim na klijentskoj strani se koristi `put()` metodu.

Nakon pohranjivanja, podaci se mogu dohvatiti na nekoliko načina. To se može učiniti putem samog Web3.storagea, IPFS gatewaya, vlastitog IPFS čvora ili putem Filecoin dohvaćanja. Ove razne opcije pružaju fleksibilnost prilagodbe specifičnim potrebama. Važno je imati na umu da bilo tko može sve pohranjene podatke na IPFS-u dohvatiti putem njihovog CID-a. Stoga se potrebno pobrinuti da se prvo pravilno enkriptira podatke ako se koristiti ova metoda web3 pohrane za privatne ili osjetljive podatke, [34].

Web3.storage, kao i bilo koja druga IPFS usluga ili čvor, i dalje je usluga koja sudjeluje u distribuiranoj peer-to-peer mreži, no pruža performanse weba2 bez kompromisa u prednostima i jamstvima web3 protokola. Za to je zaslužna arhitektura Web3.storagea.

Jedna od stvari koje čine IPFS jedinstvenim je da se podaci referenciraju pomoću sažetka podataka ili CID-a. To se čini tako što se stvara Merkle Directed acyclic graph (DAG) podataka. Ovo zahtijeva obradu podataka prije nego što mogu biti dostupni na IPFS mreži. Ako se podaci obrade kod korisnika (korisnik sam stvori hash podataka), dobiva se snažno kriptografsko jamstvo sigurnosti što omogućuje korisniku da u budućnosti kriptografski provjeri sadržaj. Kada korisnik želi prenijeti datoteku u Web3.storage, klijentska knjižnica od Web3.storage generira MerkleDAG i povezani CID na klijentskoj strani. Stoga korisnik ne mora vjerovati Web3.storageu pri vraćanju ispravnog hasha podataka jer ga sami mogu provjeriti. Nakon što su podaci obrađeni, klijent serijalizira MerkleDAG u Content Addressed Archiver (CAR) datoteku (ili više njih ako je puno podataka). To se šalje u Web3.storage i omogućuje

Web3.storageu da sačuva CID generiran na klijentskoj strani.

CAR datoteka se odmah pohranjuje na dva mjesta koja mogu preuzeti zatraženi sadržaj CID-a, ponovno sastaviti sadržaj iz podataka u CAR i poslati povezani sadržaj. Prvo mjesto je u "Elastic IPFS" instanci koja omogućuje dostupnost sadržaja na javnoj IPFS mreži tako što kada bi se preuzeo sadržaj s IPFS čvora, sadržaj se poslužuje iz Elastic IPFS instance čitajući sadržaj iz CAR datoteka koje su pohranjene tamo. Drugo mjesto je "w3link gateway". W3link je IPFS Hypertext Transfer Protocol (HTTP) gateway koji je dio Web3.storage platforme i može vratiti bilo koji sadržaj dostupan na javnoj IPFS mreži te može posebno brzo posluživati sadržaj pohranjen s Web3.storage. Pohranjivanjem CAR datoteka prijenosa w3link može preskočiti zahtjevavnje šire IPFS mreže o lokaciji podataka ako ima odgovarajuće CAR datoteke za sastavljanje zatraženog sadržaja.

Što se tiče vremena pohranjivanja i dostupnosti, čim je prijenos završen, sadržaj je odmah dostupan na w3linku putem HTTP-a. Na široj IPFS mreži postaje dostupan putem Elastic IPFS-a nekoliko sekundi kasnije jer Elastic IPFS treba malo vremena za obradu prijenosa. Općenito je w3link najbrži i najpouzdaniji način čitanja sadržaja pohranjenog na Web3.storage, budući da infrastruktura koja prima zahtjev za čitanje također posjeduje podatke jer Elastic IPFS i w3link pohranjuju podatke u CAR datotekama tokom mirovanja.

Završna faza u procesu je pohranjivanje podataka u Filecoin ugovorima pohranjivanja. Kada se klijent za pohranu poput Web3.storagea i pružatelj pohrane dogovore o poslu za pohranu, pružatelj počinje pohranjivati CAR datoteke koje sadrže podatke klijenta. Tijekom trajanja posla pružatelj pohrane mora periodično podnositi kriptografske dokaze da stvarno pohranjuje točne podatke i broj replika koje su dogovorene ugovorom. Ovi dokazi se matematički mogu provjeriti od strane drugih pružatelja skladištenja u mreži i jednom kada se provjere, ovi dokazi se upisuju u javni Filecoin blockchain. Korisnik Web3.storagea može provjeriti Filecoin blockchain kako bi se uvjerio da su podaci pohranjeni kao što je obećano. Budući da podaci moraju biti agregirani u poslove, može proći 1 do 3 dana prije nego što podaci budu preneseni i pohranjeni u Filecoin mrežu. Filecoin se smatra hladnom pohranom podataka jer čitanje podataka iz Filecoin pružatelja može potrajati neko vrijeme i uzrokovati prepreke u korisničkom iskustvu [35].

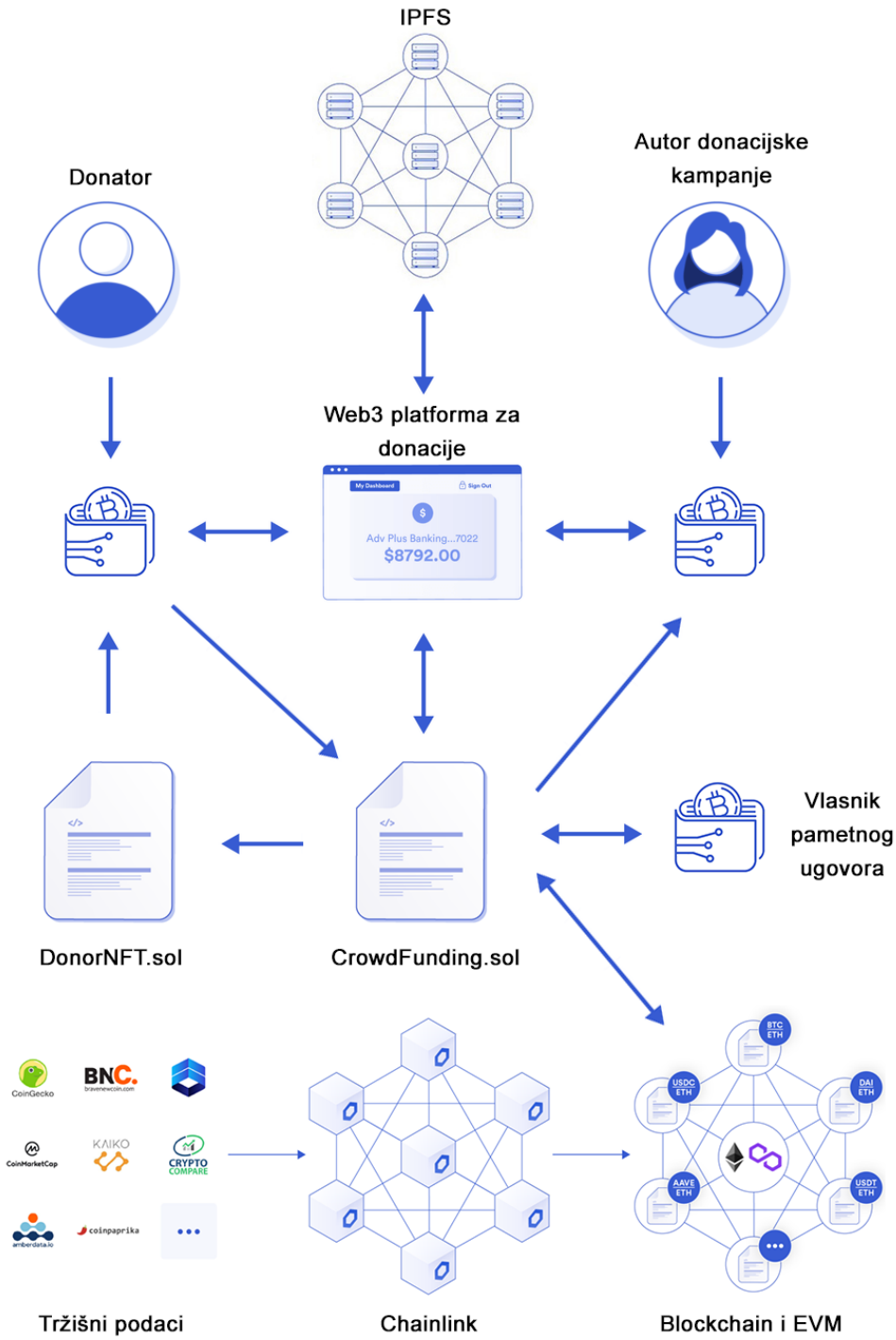
Poglavlje 5

Web3 platforma za donacije

U ovom poglavlju će se opisati Web3 platforma za donacije i način na koji funkcionira. Platforma je u potpunosti na engleskom jeziku jer je razvijana s naumom da bude predviđena i dostupa cijeloj javnosti za korištenje, a ne samo u Hrvatskoj. Web3 platforma za donacije realizira kompletno rješenje za donacije pomoću pametnih ugovora. Platformom upravljaju dva pametna ugovora "CrowdFunding.sol" i "DonorNFT.sol". CrowdFunding pametni ugovor upravlja donacijskim kampanjama i donacijama dok DonorNFT služi za mintanje i upravljanje NFT nagradama koje se dodjeljuju donatorima. Na Web3 platformi se očitavaju i smisleno prezentiraju korisniku svi podaci iz CrowdFunfing pametnog ugovora. Intuitivno korisničko sučelje Web3 platforme olakšava korisniku doniranje, izradu kampanja i ostale interakcije s pametnim ugovorom.

Na slici 5.1 je prikazano međusobno funkcioniranje svih potrebnih komponenata i tehnologija za rad Web3 platforme za doniranje. Autor donacijske kampanje izrađuje svoju kampanju koristeći korisničko sučelje Web3 platforme za donacije s kojim je povezan preko svojeg kriptonovčanika. Nakon izdrade kampanje, sve informacije o kampanji zapisuju se u "CrowdFunding.sol" pametni ugovor s kojeg Web3 platforma za donacije čita kampanje i prikazuje ih potencijalnim donatorima. Donator, koji je također povezan na Web3 platformu za donacije sa svojim kriptonovčanikom, pregledava objavljene kampanje iz "CrowdFunding.sol" pametnog ugovora preko Web3 platforme za doniranje. Odabirom jedne kampanje, učitavaju se dodatne informacije i detalji o toj kampanji iz IPFS-a i ostalih izvora koje je autor dodao. Davanjem dona-

cije kampanji pokreće se transakcija za koju "CrowdFunding.sol" analizira vrijednost doniranog iznosa pomoću Chainlink oraclea koji dobiva informacije iz različitih vanjskih izvora te ih ažurira na blockchainu. Ako je vrijednost donacije iznad određene vrijednosti, poziva se "DonorNFT.sol" pametni ugovor koji će mintati donatoru određeni NFT kao nagradu za donaciju. Donacija se prosljeđuje autoru kampanje i mali postotak te donacije se prosljeđuje na kriptonovčanik vlasnika "CrowdFunding.sol" pametnog ugovora.

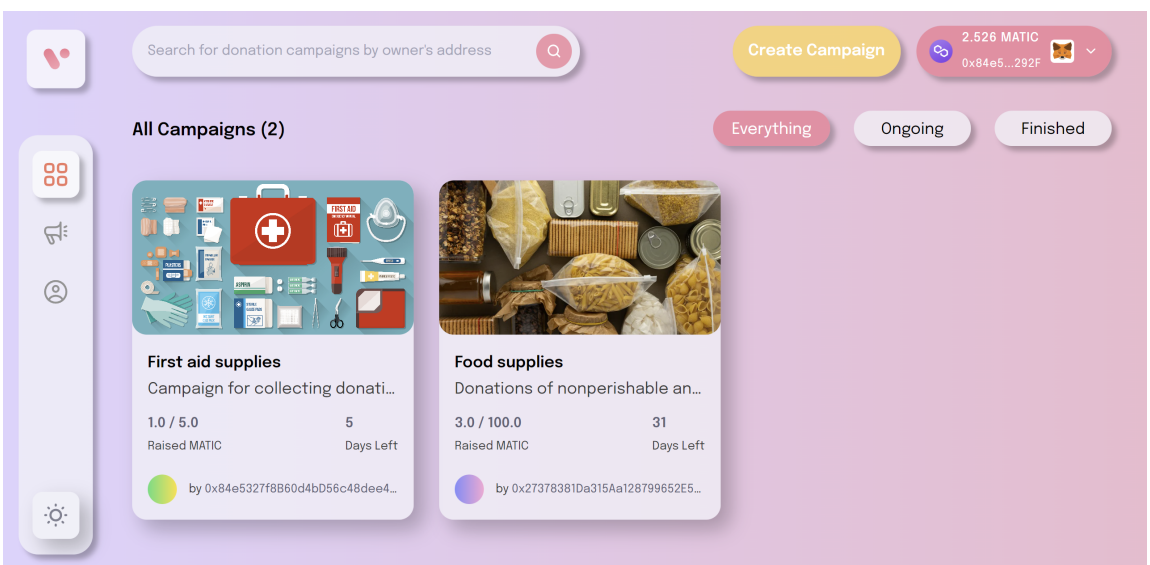


Slika 5.1 *Skica načina funkcioniranja Web3 platforme za doniranje*

5.1 Korištenje razvijene platforme

Cilj Web3 platforme za donacije je da olakša i ubrza stvaranje donacijskih kampanja sa svim potrebnim informacijama na blockchainu te da pojednostavi proces doniranja tim kampanjama. Korisnik koji želi donirati nekoj kampanji više ne mora ručno upisivati adresu kojoj želi donirati u svome novčaniku i pri tome paziti na točnost podataka, već može jednim klikom donirati preko korisničkog web sučelja platforme i prepustiti pametnom ugovoru da se pobrine za ostalo.

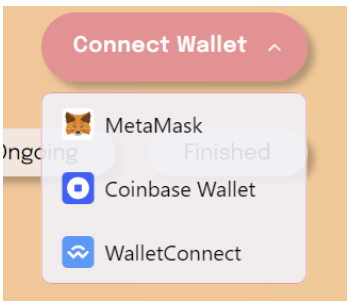
5.1.1 Popis donacijskih kampanja i povezivanje novčanika



Slika 5.2 *Popis donacijskih kampanja*

Prvo što korisnik vidi kada pristupi Web3 platformi za donacije je popis svih kampanja koje su poslože u karticama (slika 5.2). Na svakoj kartici kampanje su ukratko prikazane osnovne informacije o kampanji. Svaka kampanja ima svoju sliku kojom korisnik može jednostavno pretpostaviti na što se donacijska kampanja odnosi. Ispod slike je naziv kampanje, a ispod naziva je kratak početak teksta opisa kampanje. Zatim je prikazan ukupan iznos trenutno prikupljenih donacija za kampanju

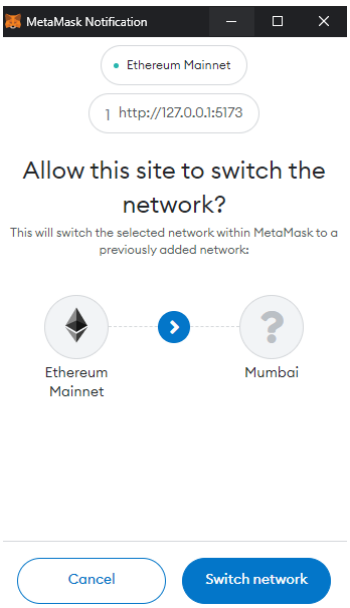
i preostalo vijeme trajanja kampanje. Na dnu je napisana adresa autora kampanje pored koje je slika gradijenta koji je ovisan o adresi autora kampanje što olakšava prepoznavanje donacijskih kampanja od istih autora pošto će boje gradijenta biti iste. Korisnik može kartice kampanja filtrirati na aktivne i neaktivne kampanje klikom na 3 gumba u gornjem desnom kutu. Iznad ta tri gumba, ako korisnik još nije spojio svoj novčanik s web3 platformom, prikazuje mu se "Connect Wallet" gumb. Klikom na taj gumb otvori se padajući izbornik s podržanim novčanicima za spajanje (slika 5.3). Podržani novčanici su MetaMask, Coinbase Wallet i WalletConnect protokol koji podržava povezivanje za više od 270 kriptonovčanika uključujući mobilne i desktop novčanike (slika 5.3) [36].



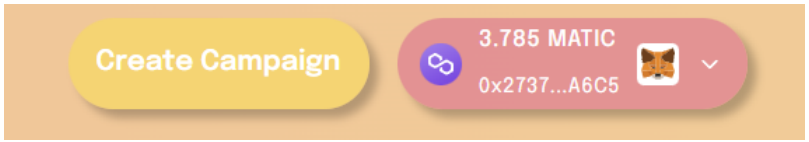
Slika 5.3 *Opcije za povezivanje novčanika*

Nakon odobrenja spajanja novčanika na Web3 platformu ako je novčanik trenutno na krivoj blockchain mreži, automatski će se pojaviti upit za promjenu blockchain mreže na odgovarajuću Polygon mrežu koja se koristi na Web3 platformi (slika 5.4). U slučaju da kriptonovčanik nema spremljene podatke o mreži, ponudit će se i opcija automatskog dodavanja mreže u kriptonovčanik.

Nakon što je novčanik uspješno spojen, na Web3 platformi pojavit će se gumb koji upućuje korisnika na izradu svoje donacijske kampanje (slika 5.5), a gumb za povezivanje novčanika postat će gumb za upravljanje raznim opcijama novčanika (slika 5.6) na kojemu su prikazane informacije o trenutno povezanom novčaniku (količina nativnog tokena ili kriptovalute i nekoliko početnih i završnih znakova adrese trenutno povezanog novčanika).



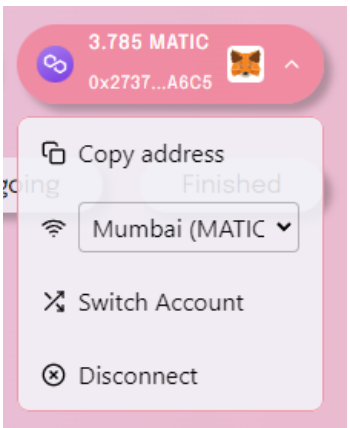
Slika 5.4 *Promjena trenutne blockchain mreže u MetaMask novčaniku*



Slika 5.5 *Uspješno povezan novčanik s Web3 platformom*

U slučaju kada je novčanik povezan na Web3 platformu, ali je na krivoj blockchain mreži, pojavit će se i gumb za promjenu mreže na ispravnu mrežu (slika 5.7). Nakon pritiska na gumb, ponovo će se u povezanom korisničkom novčaniku ponuditi zamjena mreže na ispravnu (slika 5.4).

Na vrhu je još i tražilica za ostale korisnike Web3 platforme za donacije koji imaju svoje donacijske kampanje (slika 5.2). Tražilica radi tako da, nakon što se upiše adresa novčanika autora od kojeg se želi vidjeti donacijska kampanja, odvede do profila tog korisnika gdje su prikazane sve kampanje koje je izradio. U slučaju da korisnik još nema ni jednu izrađenu kampanju, onda se ispiše da korisnik još nema svojih donacijskih kampanja.



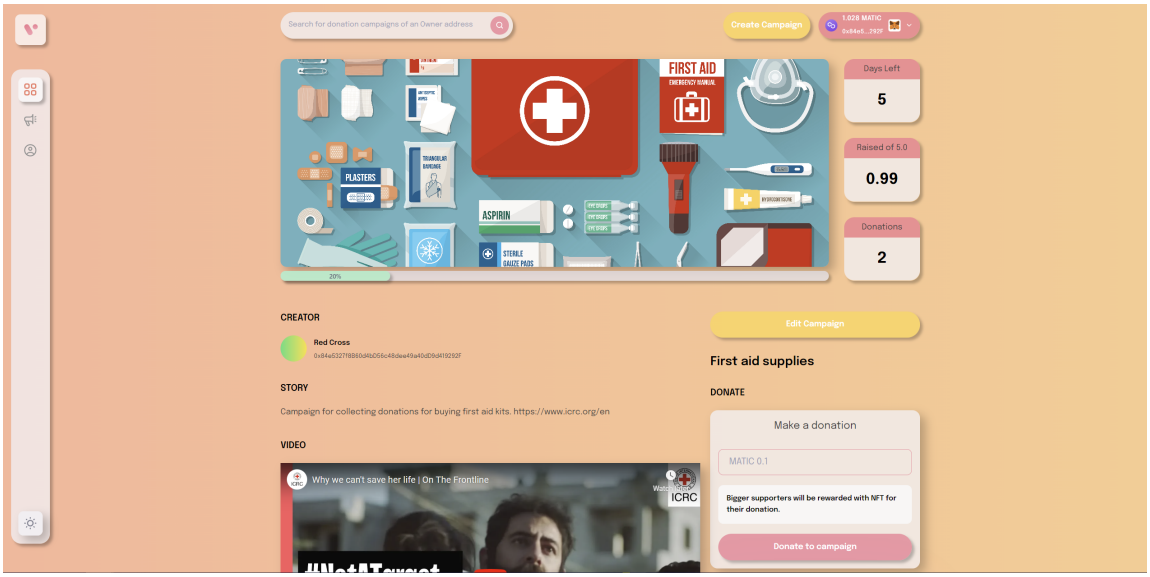
Slika 5.6 *Opcije za upravljanje trenutno povezanim novčanikom*

Slika 5.7 *Uspješno povezani novčanik, ali na krivoj blockchain mreži*

S lijeve strane korisničkog sučelja se nalazi navigacijska traka (slika 5.2). Prvi gumb na navigacijskoj traci vodi na popis svih kampanja što je ujedno i početna stranica Web3 platforme. Drugi gumb vodi na sučelje za izradu nove donacijske kampanje. Treći gumb služi za pregled svih vlastitih donacijskih kampanja dok četvrti gumb, na dnu navigacijske trake, služi za promjenu palete boja "hue" efekta pozadine.

5.1.2 Detalji kampanje i doniranje

Kada korisnik odabere kampanju za koju je zainteresiran, prikaže se sučelje s potpunim informacijama o kampanji. Ovdje korisnik može jasno vidjeti cijelu naslovnu fotografiju kampanje pored koje je prikazano par ključnih statistika. S desne strane fotografije kampanje redom je:



Slika 5.8 *Detalji donacijske kampanje*

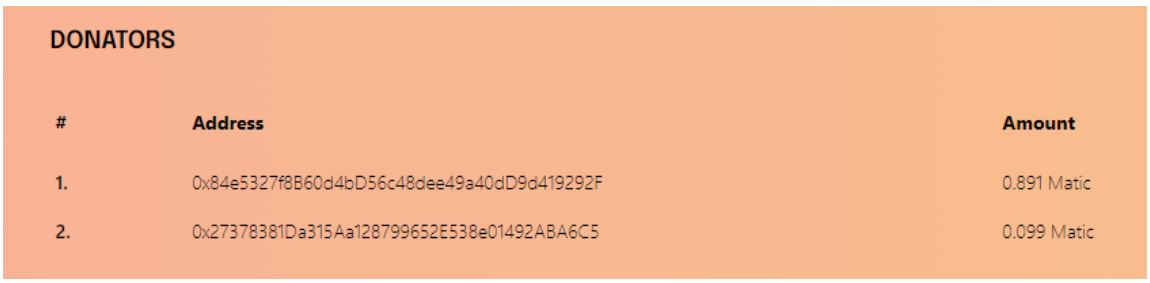
- Preostali dani trajanja kampanje do isteka roka
- Ukupna vrijednost svih donacija kampanji
- Broj donacija za kampanju

Ispod fotografije je zelena traka koja prikazuje postotak financiranja kampanje.

Ispod trake je sučelje podijeljeno na 2 stupca. Lijevi stupac sadrži sve informacije potrebne za kampanju, a desni stupac ima sučelje za slanje donacija kampanji u Matic tokenima i UniSwap sučelje za konvertiranje različitih tokena, koje korisnik ima u novčaniku, u Matic tokene.

Na početku lijevog stupca je ime autora kampanje i njegova adresa novčanika. Klikom na autora donacijske kampanje vodi se korisnika na sve kampanje tog autora. Ispod informacija o autoru nalaze se informacije o kampanji. Tu može pisati opis kampanje, dodatne udruge, organizacije, osobe koje su zadužene za donacijsku kampanju, vanjske poveznice koje vode na dodatni sadržaj, priča o povodu nastanka kampanje i mnogo drugih stvari. Ispod toga se nalazi video sadržaj ako ga je autor privezao za kampanju. Video može biti iz bilo kojeg izvora i ima kontrole upravljanja.

Nakon video sadržaja prikazuje se dokument s većom količinom informacija o kampanji. Dokument je PDF formata koji se učitava iz bilo kojeg izvora. Autor može u dokumentu pobliže objasniti okolnosti kampanje, dodati još grafičkog sadržaja, slika, shema... Ako autor ne pristavi Portable Document Format (PDF) dokument, neće se ništa prikazati. Na dnu lijevog stupca nalazi se popis svih donacija za trenutnu kampanju (slika 5.9). Ispisane su samo adrese kriptonovčanika s kojih su poslane donacije i iznosi, a donatori su ostali anonimni.



Slika 5.9 *Popis svih adresa kriptonovčanika donatora i iznosa koji su donirali*

Na početku desnog stupca je puni naslov donacijske kampanje. Ispod toga je kartica s poljem za unos iznosa donacije i gumb za slanje donacije (slika 5.8). U kartici za slanje donacije korisnici će vidjeti da postoje i NFT nagrade za veće donatore što ih možda potakne da povećaju svoj donacijski iznos kako bi zadovoljili preduvjete za dobitak odgovarajuće nagrade za koju su zainteresirani (slika 5.10).

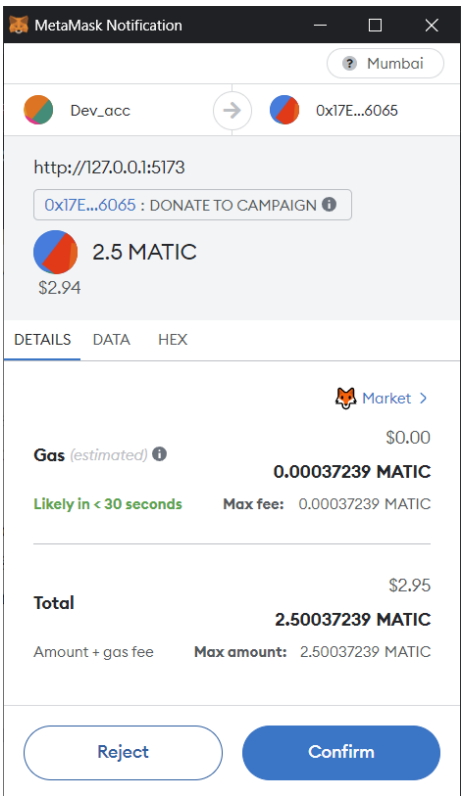
Klikom na gumb za doniranje, pojavit će se u novčaniku informacije i vrijednost transakcije te će korisnik moći potvrditi ili odbiti transakciju (slika 5.11). Najvažnije informacije o transakciji koje se prikazuju su vrijednost transakcije u Matic tokenima, ukupna vrijednost Matic tokena u United States dollar (USD) valuti, pretpostavljeni trošak prijenosa transakcije koji se izračunava tako što se predvidi izvršavanje funkcije u pametnom ugovoru te na osnovu toga i trenutnog prometa na blockchainu daje dobru estimaciju za količinu goriva koje je potrebno da se transakcija uspješno izvrši u naznačenom vremenu. Na ovoj transakciji vidi se da je trošak transakcije manji od jednog centa i da bi se transakcija trebala izvršiti u roku manjem od 30 sekundi (slika 5.11). Kako korisniku ne bi bilo dosadno čekati uspješno izvršenje transakcije, prikazuje se WebGL interaktivna šarena animacija sa svjetlosnim efektima dok

Slika 5.10 *NFT nagrade za donacije*

transakcija nije izvršena ili prekinuta.

Ispod kartice za doniranje nalazi se UniSwap sučelje za razmjenu raznih tokena u Matic tokene koji se koriste za donacije. UniSwap je najveća i najpoznatija decentralizirana mjenjačnica koja olakšava zamjenu tokena tako što uklanja potrebu za posrednicima jer koristi sustav automatskog tržišnog stvaratelja. Automated market maker (AMM) zamjenjuje tradicionalnu knjigu naloga koja se koristi na centraliziranim mjenjačnicama i omogućuje svakome da stvori bazen likvidnosti (eng. liquidity pool) za token, čineći ga dostupnim za trgovanje. U ovom sustavu pružatelji likvidnosti polažu parove tokena u pametni ugovor kako bi stvorili bazen likvidnosti. Kada korisnici žele zamijeniti tokene, izravno interaktiraju s pametnim ugovorom bazena likvidnosti.

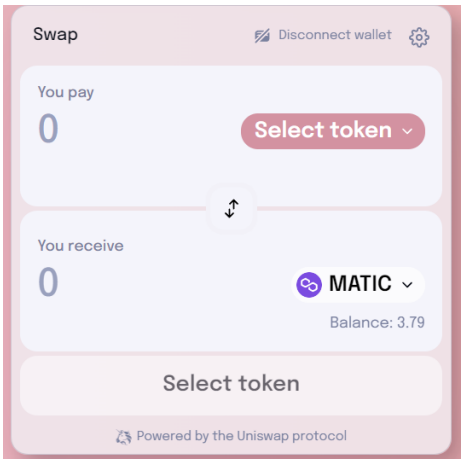
Da bi se zamijenili tokeni preko Uniswapa, korisnik jednostavno određuje ulaz



Slika 5.11 *Potvrda transakcije za donaciju*

(token koji želi prodati) i izlaz (token koji želi kupiti). AMM algoritam određuje cijenu zamjene na temelju trenutne rezerve oba tokena u likvidnom bazenu. Cijena se izračunava pomoću formule konstantnog produkta, koja se prikazuje kao X * Y = K, gdje su X i Y količine dvaju tokena u bazenu, a K je konstantna vrijednost količine oba tokena. Kada korisnik izvrši zamjenu, pametni ugovor prilagođava rezerve bazena. Korisnik dobiva željene tokene, a prodani tokeni dodaju se u bazen. Naplaćuje se mala naknada za svaku zamjenu koja se raspodjeljuje pružateljima likvidnosti kao nagrada za pružanje likvidnosti bazenu.

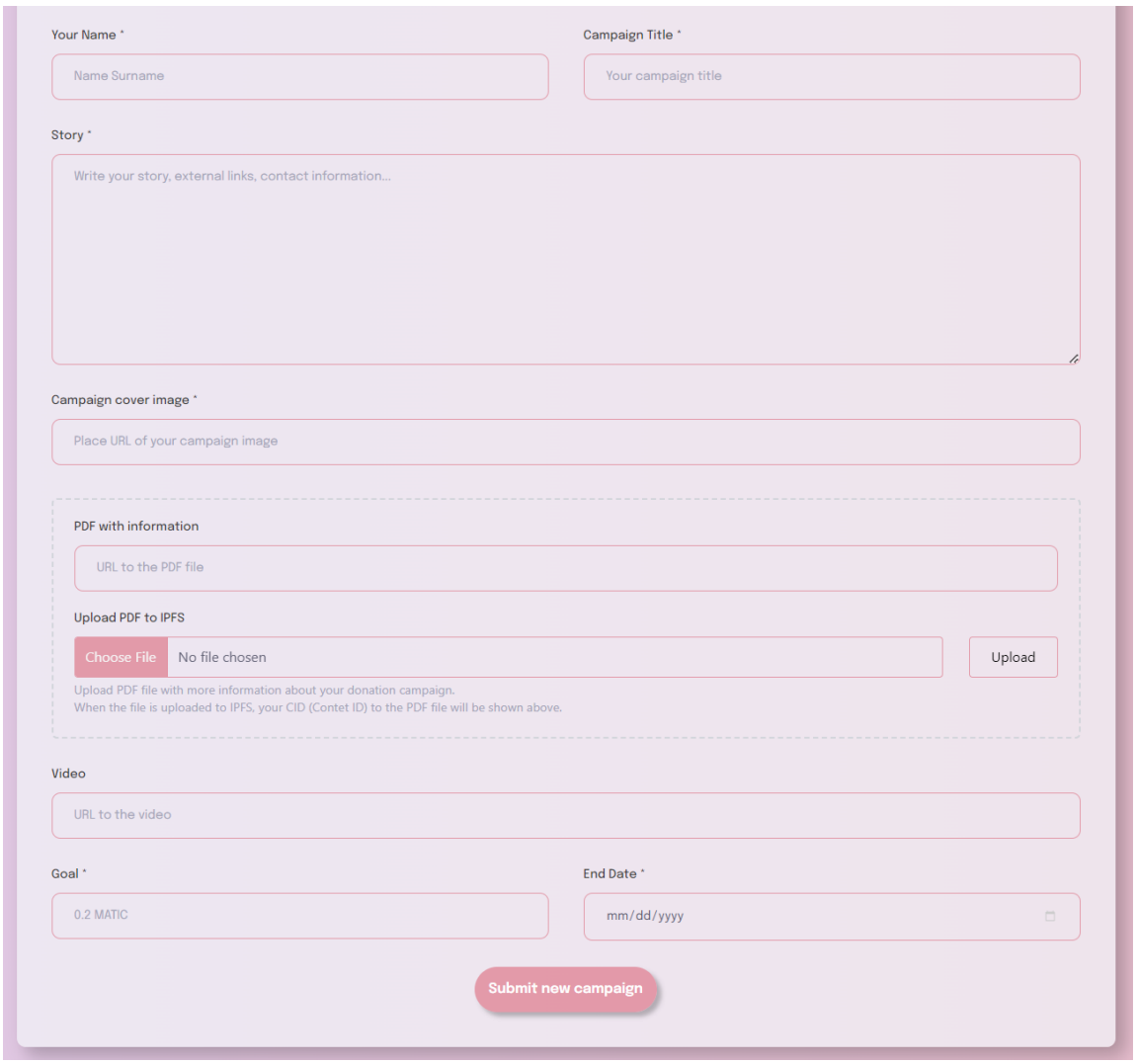
Slika 5.12 *WebGL interaktivna šarena animacija sa svijetlosnim efektima*



Slika 5.13 *UniSwap widget*

5.1.3 Izrada i uređivanje donacijske kampanje

Pri izradi nove donacijske kampanje korisnik mora pripaziti koji novčanik koristi jer će sve transakcije od donacija biti poslane na tu adresu novčanika te će jedito taj novčanik biti autoriziran za kasnije uređivanje donacijske kampanje. Polja koja

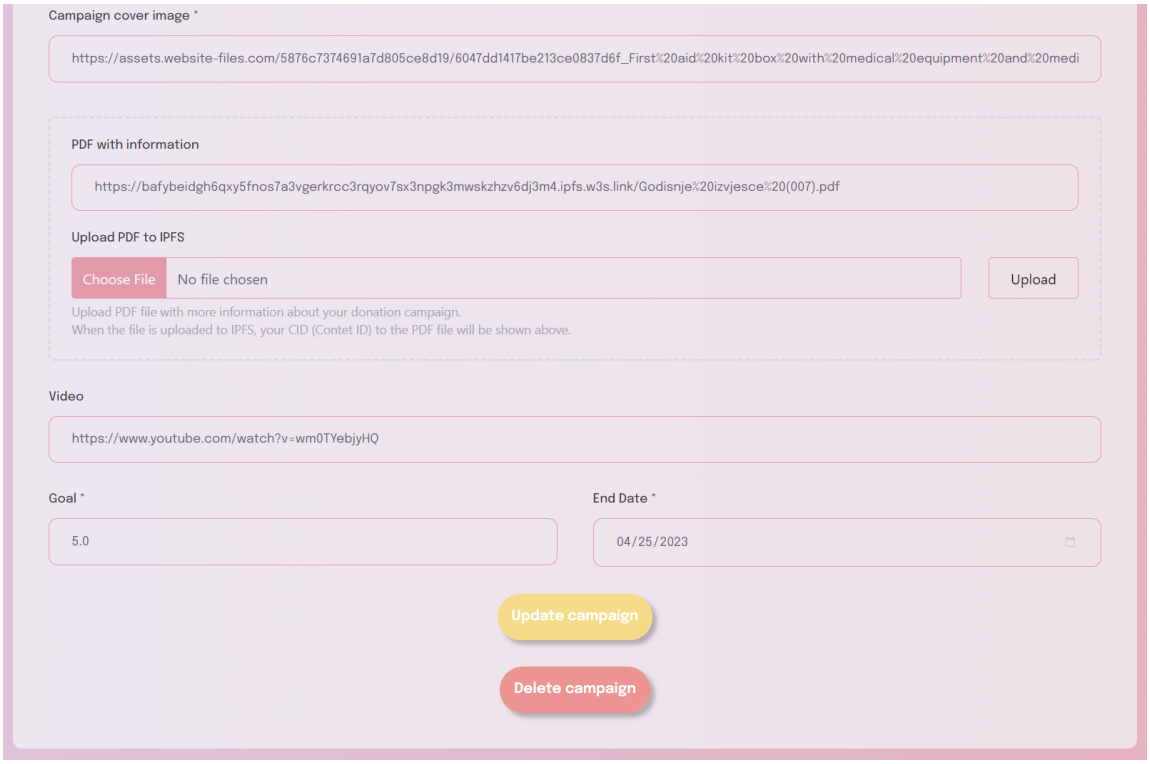


Slika 5.14 *Forma za izradu nove donacijske kampanje*

su označena sa zvjezdicom su obavezna, a polja bez zvjezdice se mogu ostaviti i prazna ako autor donacijske kampanje ne planira koristiti te medije za prikaz njegove donacijske kampanje (slika 5.14). Prvo što autor kampanje upisuje je svoje ime i prezime, nadimak ili ime organizacije koja je odgovorna za donacijsku kampanju i naziv kampanje. Zatim se u sljedećem polju daje kratki opis kampanje, priča, dodatne informacije, kontakti, eksterni linkovi... Svaka kampanja mora imati svoju odgovarajuću naslovnu fotografiju kako bi se bolje prepoznala i istakla među svim

ostalim objavljenim donacijskim kampanjama. Sljedeća polja su djelomično međuovisna. Ako korisnik želi dodati PDF dokument koji detaljnije opisuje donacijsku kampanju, može to učiniti tako što sam upisuje link do PDF dokumenta ili ako nema nigdje dostupan PDF može ga učiniti dostupnim tako da ga pošalje u IPFS nakon čega će se generirati link do PDF dokumenta i prikazati u polju iznad. Video polje služi kako bi autor mogao dodati, ako poželi, video sadržaj svojoj donacijskoj kampanji. Video može biti objavljen bilo gdje na Internetu te je potrebno upisati samo URL poveznicu do njega. Na kraju korisnik unosi cilj u Matic tokenima koji planira prikupiti donacijama i vrijeme do kojeg to želi ostvariti. Pritiskom na gumb za podnošenje nove kampanje, autor mora potpisati transakciju u svom novčaniku i pričekati stvaranje njegove kampanje. Dok se transakcija odvija, korisnika će zabaviti ona ista interaktivna WebGL animacija (slika 5.12) te će se maknuti po završetku ili prekidu transakcije.

U slučaju da autor u jednom trenutku želi promijeniti svoju donacijsku kampanju, to može učiniti tako da se spoji s novčanikom kojim je izradio kampanju koju želi izmijeniti. Svoju kampanju lako može pronaći kada ode na profil u navigacijskoj traci gdje će biti prikazane sve njegove izrađene donacijske kampanje. Odabirom kampanje, ako je spojen s ispravnim novčanikom, pojavit će se gumb iznad naziva donacijske kampanje (slika 5.8) koji vodi na korisničko sučelje za uređivanje postojeće kampanje gdje nakon unesenih promjena autor mora potvrditi u novčaniku transakciju za promjene te čekati potvrdu transakcije (slika 5.15).



Slika 5.15 *Forma za uređivanje postojeće donacijske kampanje*

5.2 Kod pametnih ugovora

U nastavku će biti pojašnjeni Solidity programski kodovi dvaju pametnih ugovora koji su napisani za svrhe ove decentralizirane aplikacije. Prvi pametni ugovor imena "CrowdFunding" upravlja svim aspektima donacijskih kampanja. Drugi pametni ugovor služi za mintanje NFT nagrada koje su dio NFT kolekcije nazvane "DonorNFT".

5.2.1 CrowdFunding pametni ugovor

```
1 // SPDX-License-Identifier: UNLICENSED
2 pragma solidity ^0.8.18;
3
4 import
  ↪ "@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol";
```

```
5  import "@openzeppelin/contracts/access/Ownable.sol";
6
7  interface IDonorNFT {
8      function mintReward(address _to, string calldata _rewardTier)
   ↪  external;
9  }
```

Prije koda implementacije pametnog ugovora potrebno je definirati nekoliko stvari. Prva stvar koja je definirana je vrsta licence koda s ovom zakomentiranom linijom `// SPDX-License-Identifier: UNLICENSED`. U ovom slučaju kod nije licenciran, što znači da nije objavljen pod nekom posebnom open source licencom.

`pragma solidity ^0.8.18` linija određuje verziju Solidity kompajlera u kojoj je napisan kod. Navodi da se kod treba kompajlirati s verzijom kompajlera 0.8.18 ili novijom, ali ispod verzije 0.9.0. To se radi iz razloga što starije verzije Soliditya ne podržavaju sve sigurnosne zakrpe, a buduće 0.9.x verzije mogu uvesti prijelomne (eng. braking) promijene.

Zatim se uvozi `AggregatorV3Interface` ugovor iz Chainlink knjižnice pametnih ugovora. Chainlink je decentralizirana oracle mreža koja omogućuje pametnim ugovorima siguran pristup "off-chain" izvorima podataka. Uvezeno sučelje pruža interakciju pametnog ugovora s Chainlinkovim izvorima cijena kriptovaluta i tokena.

Zatim se uvozi `Ownable` pametan ugovor iz OpenZeppelin knjižnice. OpenZeppelin je popularna i široko korištena knjižnica sigurnih i provjerenih implementacija standardnih pametnih ugovora za Ethereum i ostale blockchainove. Ugovor `Ownable` pruža jednostavan mehanizam za kontrolu pristupa tako što omogućuje ograničavanje određenih funkcija ugovora na vlasnika ugovora.

Na kraju je sučelje pod nazivom `IDonorNFT`. Sučelja se koriste za definiranje strukture ugovora, navodeći funkcije koje moraju implementirati svi ugovori koji žele biti u skladu sa sučeljem. U ovom slučaju, sučelje `IDonorNFT` predstavlja ugovor koji se bavi izdavanjem NFT-ova kao nagrada. Funkcija `mintReward` prihvaća dva parametra: adresu novčanika i string vrste nagrade. Ključna riječ `external` označava da se ova funkcija može pozivati samo iz drugih ugovora, a ne iz ugovora koji ju implementira.

```
contract CrowdFunding is Ownable {

    struct Campaign {
        address owner;
        string title;
        string description;
        uint256 target;
        uint256 deadline;
        uint256 amountCollected;
        string image;
        address[] donators;
        uint256[] donations;
        uint256 campaignId;
        string pdf;
        string video;
        string name;
    }

    mapping(uint256 => Campaign) public campaigns;

    uint256 public numberOfCampaigns = 0;
    uint256 public realNumberOfCampaigns = 0;
```

Nakon definiranja licence, kompajlera, uvezenih ugovora i sučelja, deklarira se pametni ugovor pod nazivom `CrowdFunding` koji nasljeđuje `Ownable` ugovor koji pruža osnovne funkcije kontrole autorizacije te postavlja stvaratelja ugovora kao vlasnika i ograničava određene funkcije da ih može pozvati samo vlasnik. Na početku pametnog ugovora definirana je struktura pod nazivom `Campaign`. Struktura `Campaign` predstavlja jednu donacijsku kampanju i sastoji se od sljedećih polja:

- `address owner`: Adresa vlasnika kampanje.
- `string title`: Naslov kampanje.
- `string description`: Opis kampanje.
- `uint256 target`: Ciljni iznos prikupljanja sredstava u Wei-u (najmanja jedinica Ethra/Matica).
- `uint256 deadline`: Vremenska oznaka roka za kampanju predstavljena u Unix vremenu (sekundama od Unix epohe).

- `uint256 amountCollected`: Ukupan iznos dosad prikupljenih donacija za kampanju.
- `string image`: Poveznica na sliku koja predstavlja kampanju.
- `address[] donators`: Polje adresa donatora koji su doprinijeli kampanji.
- `uint256[] donations`: Polje pripadajućih iznosa donacija u Wei-u.
- `uint256 campaignId`: Jedinstveni identifikator kampanje.
- `string pdf`: Poveznica na PDF dokument povezan s kampanjom.
- `string video`: Poveznica na video kampanje.
- `string name`: Ime autora kampanje.

Linijom `mapping(uint256 => Campaign) public campaigns` deklarira se mapiranje pod nazivom `campaigns` koje mapira jedinstvene ID-ove kampanje (`uint256`) na odgovarajuće `Campaign` strukture. To omogućuje jednostavan pristup kampanji pomoću njenog ID-a i trajnu pohranu informacija na blockchainu o svakoj kampanji. Globalna varijabla `numberOfCampaigns` služi za praćenje ukupnog broja stvorenih kampanja na platformi, dok globalna varijabla `realNumberOfCampaigns` služi za praćenje trenutnog broja kampanja na platformi.

```
address public rewardNFTcontract;

AggregatorV3Interface internal maticUsdPriceFeed =
↪  AggregatorV3Interface(0xd0D5e3DB44DE05E9F294BB0a3bEEaF030DE24Ada);

function setRewardNFTcontract(address _rewardNFTcontract) external
↪  onlyOwner {
    rewardNFTcontract = _rewardNFTcontract;
}

function getLatestPrice() public view returns (uint256) {
    (
        /* uint80 roundID */,
        int price,
        /*uint startedAt*/,
        /*uint timeStamp*/,
        /*uint80 answeredInRound*/
    ) = maticUsdPriceFeed.latestRoundData();
```

```
17      return uint256(price);
18  }
```

`rewardNFTcontract` varijabla sadrži adresu `DonorNFT` pametnog ugovora za dodjelu NFT nagrada za veće donacije. `setRewardNFTcontract()` funkcija uzima adresu pametnog ugovora kao ulazni parametar i omogućuje vlasniku pametnog ugovora postavljanje `rewardNFTcontract` adrese. Modifikator `onlyOwner` osigurava da funkciju može pozvati samo vlasnik ugovora.

`maticUsdPriceFeed` je varijabla tipa `AggregatorV3Interface` koja se koristi za interakciju s Chainlinkovim cjenovnim agregatorom. Adresa koja je dana kao argument `AggregatorV3Interface()` funkcije je adresa Aggregator ugovora na Mumbai testnoj mreži koji pruža podatke o cijeni Matic-USD.

`getLatestPrice()` funkcija je javna `view` funkcija, pošto ne mijenja stanje ugovora, koja vraća najnoviju cijenu Matic-USD iz Chainlinkovog agregatora.

```
1   function createCampaign(
2       address _owner,
3       string memory _title,
4       string memory _description,
5       uint256 _target,
6       uint256 _deadline,
7       string memory _image,
8       string memory _pdf,
9       string memory _video,
10      string memory _name
11      ) public returns (uint256) {
12          require(_deadline > block.timestamp, "Deadline in the past" );
13          require(_target > 0, "Goal value must be greater than zero");
14          Campaign storage campaign = campaigns[numberOfCampaigns];
15
16          campaign.owner = _owner;
17          campaign.title = _title;
18          campaign.description = _description;
19          campaign.target = _target;
20          campaign.deadline = _deadline;
21          campaign.amountCollected = 0;
22          campaign.image = _image;
```

```
        campaign.campaignId = numberOfCampaigns;
        campaign.pdf = _pdf;
        campaign.video = _video;
        campaign.name = _name;

        numberOfCampaigns++;
        realNumberOfCampaigns++;
        return numberOfCampaigns - 1;
    }
```

Funkcija `createCampaign(...)` omogućuje stvaranje i spremanje novih donacijskih kampanja u pametnom ugovoru ako su zadovoljeni određeni uvjeti i parametri. Jedan od tih uvjeta je `require(_deadline > block.timestamp)` koji provjerava je li zadani rok za kampanju veći od trenutne vremenske oznake (`block.timestamp`). Ako nije, funkcija će baciti grešku s porukom "Deadline in the past". `require(_target > 0)` provjerava je li ciljni iznos prikupljanja donacija veći od nule. Ako nije, funkcija će baciti grešku s porukom "Goal value must be greater than zero". Iako se na korisničkom sučelju sprječava korisnika da upiše nepogodne vrijednosti, potrebno je svejedno i u pametnom ugovoru postaviti uvjete za slučajeve kada netko pokuša interaktirati direktno s pametnim ugovorom bez korisničkog sučelja.

Linija `Campaign storage campaign = campaigns[numberOfCampaigns]` stvara referencu na novu kampanju u `campaigns` mapiranju, koristeći trenutnu vrijednost varijable `numberOfCampaigns` kao ključ. Na kraju funkcija vraća ID nove kampanje koji je jednak vrijednosti `numberOfCampaigns` umanjenoj za 1 pošto se prije vraćanja vrijednosti `numberOfCampaigns` povećao za 1.

```
function donateToCampaign(uint256 _id) public payable {
    uint256 amount = msg.value;

    Campaign storage campaign = campaigns[_id];

    require(campaign.owner != address(0), "Donation failed. This campaign
↪   does not exist.");

    uint256 onePercent = amount / 100;
    uint256 remainingAmount = amount - onePercent;

```

```
11      campaign.donators.push(msg.sender);
12      campaign.donations.push(remainingAmount);
13      campaign.amountCollected = campaign.amountCollected + remainingAmount;
14
15      (bool sentToOwner,) = payable(owner()).call{value: onePercent}("");
16      require(sentToOwner, "Failed to send 1% to contract owner");
17
18      (bool sent,) = payable(campaign.owner).call{value:
    ↪  remainingAmount}("");
19      require(sent, "Failed to send Matic");
20
21      if(sent) {
22          uint256 maticPriceInUSD = getLatestPrice();
23          uint256 totalDonationValueInUSD = (msg.value * maticPriceInUSD) /
    ↪  1e18;
24          IDonorNFT NFTcontract = IDonorNFT(rewardNFTcontract);
25          if (totalDonationValueInUSD > (500 * 1e8)) {
26              NFTcontract.mintReward(msg.sender, "6");
27          } else if (totalDonationValueInUSD > (200 * 1e8)) {
28              NFTcontract.mintReward(msg.sender, "5");
29          } else if (totalDonationValueInUSD > (100 * 1e8)) {
30              NFTcontract.mintReward(msg.sender, "4");
31          } else if (totalDonationValueInUSD > (50 * 1e8)) {
32              NFTcontract.mintReward(msg.sender, "3");
33          } else if (totalDonationValueInUSD > (25 * 1e8)) {
34              NFTcontract.mintReward(msg.sender, "2");
35          } else if (totalDonationValueInUSD > (10 * 1e8)) {
36              NFTcontract.mintReward(msg.sender, "1");
37          }
38      }
39  }
```

Funkcija `donateToCampaign(uint256 _id)` omogućuje korisnicima doniranje sredstava kampanji. Prije daljnjeg izvršavanja funkcije, provjerava se ako kampanja kojoj se želi poslati donacija postoji. Funkcija je označena kao `payable` jer će prihvatiti u ovom slučaju Matic kao uplatu. 1% doniranog iznosa šalje se vlasniku pametnog ugovora dok se preostali iznos šalje vlasniku kampanje. Ako su transakcije uspješno izvršene, kreće provjera ako je donacija dovoljno velika da zadovolji uvjete za NFT nagradu.

`getLatestPrice()` dohvaća najnoviju vrijednost Matica u američkim dolarima koja služi da bi se izračunala ukupna vrijednost donacije u američkim dolarima. Ovisno o vrijednosti njihove donacije u američkim dolarima zauzvrat će dobiti NFT nagradu za koju zadovoljavaju uvjet.

```
function getDonators(uint256 _id) view public returns (address[] memory,
↪  uint256[] memory) {
    return (campaigns[_id].donators, campaigns[_id].donations);
}
```

Funkcija `getDonators(uint256 _id)` vraća popis donatora i njihove donacije za kampanju čiji ID je predan kao ulazni parametar. Funkcija je označena kao `view` jer ne mijenja stanje ugovora, već samo čita podatke iz njega.

```
function getCampaigns() public view returns (Campaign[] memory) {
    Campaign[] memory allCampaigns = new
↪  Campaign[](realNumberOfCampaigns);

    uint j = 0;
    for(uint i = 0; i < numberOfCampaigns; i++) {
        if (campaigns[i].owner == address(0)) continue;

        allCampaigns[j] = campaigns[i];
        j++;
    }

    return allCampaigns;
}
```

`getCampaigns()` vraća polje svih kampanja koje su pohranjene u pametnom ugovoru. Funkcija je označena kao `view` jer samo čita podatke. Prvo se stvara prazno polje u memoriji za spremanje svih kampanja koje će se vratiti. Polje se inicijalizira s veličinom `realNumberOfCampaigns`, što je broj stvarnih kampanja koje su pohranjene u pametnom ugovoru. Za svaku kampanju se provjerava ako je izbrisana te ako nije, dodaje se u `allCampaigns` polje.

```
function getCampaign(uint256 _id) view public returns (Campaign memory) {
    return campaigns[_id];
}
```

Pomoću `getCampaign(uint256 _id)` se, koristeći ID kampanje kao ulazni parametar, dohvaća kampanja koja je pohranjena u pametnom ugovoru. Kampanja se vraća kao `Campaign` struktura u memoriji.

```
function modifyCampaign(
    uint256 _id,
    string memory _title,
    string memory _description,
    uint256 _target,
    uint256 _deadline,
    string memory _image,
    string memory _pdf,
    string memory _video,
    string memory _name
) public {

    if(msg.sender != campaigns[_id].owner) {
        revert("Can't be modified. You are not the owner.");
    }
    require(_deadline > block.timestamp, "Deadline in the past instead of
    ↪ future" );
    require(_target > 0, "Goal value must be greater than zero");

    campaigns[_id].title = _title;
    campaigns[_id].description = _description;
    campaigns[_id].target = _target;
    campaigns[_id].deadline = _deadline;
    campaigns[_id].image = _image;
    campaigns[_id].pdf = _pdf;
    campaigns[_id].video = _video;
    campaigns[_id].name = _name;
}
```

`modifyCampaign(...)` omogućuje vlasnicima kampanje da mijenjaju podatke svoje kampanje. Ulazni parametri uključuju `_id` kampanje i sve podatke koji će se mijenjati.

`if(msg.sender != campaigns[_id].owner)` redak provjerava je li adresa novčanika osobe koja pokreće funkciju jednaka adresi vlasnika kampanje s tim ID-om. Ako adrese nisu jednake, izvršava se `revert()` koji poništava sve promjene transakcije i vraća poruku o pogrešci. Ovdje su i provjere novog datuma kraja roka donacijske kampanje i novog cilja prikupljanja kampanje. Ažuriranje podataka kampanje može biti korisno za ispravljanje pogrešaka ili ažuriranje informacija u skladu s promjenama u kampanji.

```
function deleteCampaign(uint256 _id) public {
    if(msg.sender != campaigns[_id].owner) {
        revert("Can't be deleted. You are not the owner of this.");
    }
    delete campaigns[_id];
    realNumberOfCampaigns--;
}
```

Zadnja funkcija u pametnom ugovoru omogućuje vlasnicima kampanja da izbrišu svoje kampanje iz pametnog ugovora. Jedina potrebna provjera u ovoj funkciji je dolazi li zahtjev za brisanjem kampanje od vlasnika kampanje. Jer ako je kampanja već izbrisana, onda će adresa vlasnika kampanje biti jednaka nuli čime uvjet za brisanje nikada neće biti zadovoljen. Ispod je izgled JavaScript Object Notation (JSON) datoteke izbrisane kampanje u pametnom ugovoru.

```
[
  "0x0000000000000000000000000000000000000000",
  "",
  "",
  {
    "type": "BigNumber",
    "hex": "0x00"
  },
  {
    "type": "BigNumber",
    "hex": "0x00"
  },
  {
    "type": "BigNumber",
    "hex": "0x00"
```

```
  },
  "",
  [],
  [],
  {
    "type": "BigNumber",
    "hex": "0x00"
  },
  "",
  "",
  ""
]
```

`delete campaigns[_id]` redak briše kampanju s ID-om iz mapiranja `campaigns`. To se postiže postavljanjem vrijednosti za taj ključ (ID kampanje) na njegovu pretpostavljenu vrijednost što u ovom slučaju znači da će svi podaci o kampanji biti postavljeni na njihove pretpostavljene vrijednosti kao u primjeru iznad (npr. adresa vlasnika će biti `address(0)`).

Na kraju se smanjuje `realNumberOfCampaigns` broj za 1 kako bi se odražavala činjenica da je jedna kampanja izbrisana.

5.2.2 Donor NFT pametni ugovor

Donor NFT služi kao potvrda i nagrada donatoru za njegov donirani iznos. Također motivira donatore da povećaju svoju donaciju kako bi zadovoljili uvjete dobitka određene NFT nagrade.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.18;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import
↪  "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/utils/Counters.sol";
```

"ERC721.sol" pametni ugovor iz OpenZeppelin knjižnice je standard za nezamje-

njive i jedinstvene tokene na Ethereum blockchainu. Uvozom ovog ugovora nasljeđuju se funkcionalnosti za stvaranje i upravljanje NFT-ovima. "ERC721URIStorage.sol" je proširenje koje omogućuje da se pohranjuje Uniform Resource Identifier (URI) za ERC721 tokene. Token URI obično sadrži metapodatke o tokenu poput JSON objekta koji opisuje svojstva i atribute tokena te se obično nalazi na decentraliziranom sustavu za pohranu podataka poput IPFS-a. Nasljeđivanjem ovog ugovora može se povezati NFT-ove s njihovim odgovarajućim URI-ima te upravljati ažuriranjima token URI-a. "Ownable.sol" je, kako je već navedeno, jednostavan mehanizam kontrole pristupa. "Counters.sol" pruža jednostavan alat za povećavanje i smanjenje unsigned integera (uint256). To je korisno za upravljanje brojačima kao što su ID-ovi tokena ili drugi sekvencijalni podaci u pametnom ugovoru.

```
contract DonorNFT is ERC721URIStorage, Ownable {
    address private CONTRACT_MINTER;
    string internal baseURIreward;
    using Counters for Counters.Counter;
    Counters.Counter private _tokenIdCounter;

    mapping(address => uint256) public mintedWallets;
```

U kodu iznad je definiran pametni ugovor za NFT nagrade pod nazivom "DonorNFT". Nasljeđuje `ERC721URIStorage` i `Ownable` pametne ugovore. To znači da će ugovor imati funkcionalnost ERC721 NFT standarda s mogućnostima pohrane URI-a i mehanizme kontrole pristupa koje pruža Ownable ugovor.

`CONTRACT_MINTER` je privatno deklarirana varijabla tipa address koja će se koristiti za pohranu adrese ugovora zaduženog za stvaranje novih tokena u ugovoru "DonorNFT". `baseURIreward` je interna varijabla tipa string koja pohranjuje osnovni URI za metapodatke tokena koji se koristi kao prefiks pri izradi potpunog URI-a za svaki token. To olakšava upravljanje i ažuriranje osnovnog URI-a ako je potrebno zamijeniti NFT kolekciju koja će se mintati.

`using Counters for Counters.Counter` ukazuje da će ugovor koristiti knjižnicu `Counters` za upravljanje instancama `Counters.Counter`. `Counters.Counter` je struktura koja uključuje neoznačenu cjelobrojnu vrijednost i pruža funkcionalnost za sigurno povećavanje i smanjenje te vrijednosti. Privatna varijabla `_tokenIdCounter`

tipa `Counters.Counter` je brojač koji će se koristiti za upravljanje i generiranje jedinstvenih ID-ova tokena za NFT-ove stvorene unutar ovog ugovora.

Mapiranje pod nazivom `mintedWallets` mapira `wallet` adrese na cjelobrojne vrijednosti (`uint256`) koje prate broj mintanih NFT-a za svaku adresu novčanika.

```
constructor() ERC721("DonorNFT", "DONR") {
    CONTRACT_MINTER = address(0);
    baseURIreward =
↪   "ipfs://QmRcEHh4ndzemEZQgDD1Vny8SvTU23RVLYq6yZfh7prQxF/";
}
```

Ovaj dio Solidity pametnog ugovora definira konstruktor za ugovor "DonorNFT". Konstruktor je posebna funkcija koja se poziva samo jednom kada se ugovor postavi na blockchain i obično postavlja početno stanje i vrijednosti ugovora. Poziva se konstruktor naslijeđenog ugovora ERC721 kojemu se prosljeđuje ime "DonorNFT" i simbol "DONR" kao argumente NFT kolekcije.

`CONTRACT_MINTER` varijabla će biti postavljena na nultu adresu (`address(0)`) dok se ručno ne doda adresa "CrowdFunding.sol" pametnog ugovora za donacijske kampanje od strane vlasnika, a do tad nijedna specifična adresa nema dozvolu za stvaranje tokena. Nulta adresa je posebna adresa koja obično predstavlja neinicijalizirani ili nepostojeći entitet u Ethereum ekosustavu.

`baseURIreward` će biti postavljen na IPFS URI gdje su pohranjeni metapodaci NFT-ova. Ovaj osnovni URI koristit će se kao prefiks pri izradi potpunog URI-ja za svaki token.

```
function setBaseURI(string calldata _newUri) public onlyOwner {
    baseURIreward = _newUri;
}

function setContractAddress(address _newContractAddress) public onlyOwner
↪   {
    CONTRACT_MINTER = _newContractAddress;
}
```

Ove dvije funkcije omogućuju vlasniku ugovora ažuriranje osnovnog URI-a za

metapodatke tokena i adrese odgovorne za izradu novih nagradnih tokena. Obe funkcije mogu biti pozvane samo od vlasnika pametnog ugovora zbog modifikatora `onlyOwner`.

```
modifier onlyContract() {
    require(msg.sender == CONTRACT_MINTER, "Mint not allowed");
    _;
}
```

`modifier onlyContract()` je modifikator koji provjerava je li pošiljatelj poruke (`msg.sender`) jednak adresi `CONTRACT_MINTER`. Ako nije, izvršavanje funkcije će biti poništeno s porukom o pogrešci "Mint not allowed". Ova provjera osigurava da samo adresa definirana u varijabli `CONTRACT_MINTER` može pozivati funkcije koje koriste modifikator `onlyContract`.

Oznaka "`_;`" služi za označavanje mjesta stvarnog tijela funkcije kada se primijeni modifikator, odnosno zamijenit će se kodom funkcije kada se pozove funkcija s ovim modifikatorom.

```
function mintReward(address _to, string calldata _rewardTier) external
↪   onlyContract {
    uint256 tokenId = _tokenIdCounter.current();
    string memory fullRewardURI = string.concat(baseURIreward,
↪   _rewardTier, ".json");
    _mint(_to, tokenId);
    _setTokenURI(tokenId, fullRewardURI);
    _tokenIdCounter.increment();
    mintedWallets[_to]++;
}
```

`mintReward(...)` je vanjska funkcija s dva ulazna parametra: `_to` tipa address i `_rewardTier` tipa string. Ključna riječ `calldata` ukazuje na to da će se podaci parametra samo čitati, bez promjena. Modifikator `onlyContract` osigurava da samo adresa `CONTRACT_MINTER` može pozvati ovu funkciju.

`_tokenIdCounter.current()` dohvaća trenutnu vrijednost `_tokenIdCounter` i koristiti kao ID za novi token koji se stvara.

`string.concat(...)` spaja `baseURIreward, _rewardTier` i "`.json`" kako bi se stvorio `fullRewardURI` koji predstavlja potpuni URI za metapodatke tokena.

`_mint(_to, tokenId)` poziva unutarnju funkciju `_mint` koja se nasljeđuje iz ugovora ERC721. Stvara novi token s `tokenId` i dodjeljuje ga `adresi _to`.

`_setTokenURI(tokenId, fullRewardURI)` poziva `_setTokenURI` koja se nasljeđuje iz ugovora "ERC721URIStorage". Povezuje `fullRewardURI` s `tokenId`, postavljajući time URI metapodataka za novostvoreni token. Na kraju se povećava broj izrađenih tokena za adresu `_to` u mapiranju `mintedWallets`.

5.3 Testiranje

Testne mreže (eng. testnets) omogućuju programerima temeljito otklanjanje pogrešaka i optimizaciju koda pametnih ugovora prije postavljanja na glavnu mrežu. Korištenje testnih mreža obično je besplatno za razliku od korištenja glavne mreže. Testne mreže su predviđene za opsežno testiranje decentralizirane aplikacije bez brige o troškovima povezanim s transakcijama ili postavljanjem pametnih ugovora. Testiranje na tastnoj mreži omogućuje prikupljanje povratnih informacija od budućih korisnika pošto pruža sigurno i kontrolirano okruženje u kojem se decentralizirana aplikacija može testirati bez rizika i troška stvarnih vrijednosti u slučaju pogrešaka ili ranjivosti. Na testnim mrežama su također postavljeni pametni ugovori za testiranje integracije decentralizirane aplikacije s drugim uslugama web3 ekosustava kao što su novčanici, oraclei i mjenjačnice. Također testne mreže se često koriste za testiranje novih nadogradnji mreže i promjena protokola prije nego što se implementiraju na glavnu mrežu kao što je primjer s Ethereumovim prelaskom s Proof-of-Work (PoW) na PoS konsenzus mehanizam. Time se osigurava da aplikacije budu kompatibilne s nadolazećim promjenama. Bitno je provođenje i testova prodiranja kako bi se identificirale ranjivosti u pametnim ugovorima. Rješavanjem tih ranjivosti prije postavljanja na glavnu mrežu smanjuje se rizik od napada i štiti imovina korisnika.

Stoga testiranje web3 decentraliziranih aplikacija na testnetu jedan je od najvažnijih koraka u razvojnom procesu. Kako bi Web3 platforma za doniranje radila glatko i sigurno prije postavljanja na glavnu mrežu, sve funkcionalnosti navedene u

prethodnom poglavlju su razvijene i testirane na Mumbai testnoj mreži i u React produkcijskom okruženju. Jedino Uniswap widget radi u razvojnom okruženju, a ne radi u React produkcijskom okruženju zbog problema s minimizacijom i kompajliranjem koda Uniswap widget knjižnice. Funkcionalnosti su testirane izradom kampanja koje su imale različite opise, ciljeve iznosa, datume kraja prikupljanja, izvore video i PDF sadržaja. Učitavanje PDF sadržaja testirano je učitavanjem PDF dokumenta od 132 stranice. Transakcije donacija testirane su s različitim iznosima donacija (0.05 - 12.5 Matic) čime je testirano i nagrađivanje donatora s NFT nagradom za donaciju veću od 10$ vrijdnosti. U slučajevima kada je krivo upisana svota doniranja (ostavljeno prazno polje ili vrijednost donacije 0 Matic tokena) ili donator nema dovoljno Matic tokena u povezanom kriptonovčaniku za donaciju, transakcija se neće ni pokrenuti. Kod forma izrade i ažuriranja kampanja, testirana je ispravnost provjeravanja upisa potrebnih informacija u forme. Transakcija za izradu nove ili postojeće kampanje se neće pokrenuti ako nisu popunjena sva obavezna polja u formi s točnim infromacijama. Također je testiran pokušaj brisanja i mjenjanja informacija potojeće kampanje s adrese kriptonovčanika koja nije vlasnik te kampanje. Riješen je i slučaj direktnog pristupa informacijama kampanje s linkom tako da se učitaju podaci iz pametnog ugovora za kampanju, umjesto da se proslijede kao svojstvo iz roditeljske komponente. Responzivnost sučelja je testirana na različitim dimenzijama ekrana računala, mobitela i tableta.

Ukratko, testiranje web3 decentralizirane Polygon aplikacije na Mumbai testnetu ključno je za osiguravanje sigurnosti, performansi i pouzdanosti pri korištenju.

Poglavlje 6

Zaključak

U ovom radu opisana je i razvijena Web3 platforma za donacije. Za njenu izradu korišten je Solidity programski jezik za razvoj pametnih ugovora i React zajedno s Tailwind knjižnicom da bi se postigao responzivan i moderan dizajn korisničkog web sučelja. Polygon omogućuje brze i jeftine transakcije što pridonosi korisničkom iskustvu. ThirdWeb olakšava proces razvoja pametnih ugovora, a Web3.storage daje potrebnu brzinu IPFS-u da bi se koristio za pohranu i pristup većih podataka na Web3 platformi. Aspekt kolekcionarstva NFT-ova dodatno potiče korisnike da doniraju, a intuitivnost platforme omogućuje korisnicima brzo savladavanje i lako korištenje čime se otvaraju vrata novoj generaciji donatora. Uza sve to izvedena je sveobuhvatna analiza potencijalnih prednosti usvajanja web3 platformi. Pojavom web3 platformi u području filantropije otvaraju se značajne mogućnosti za poboljšanje učinkovitosti i transparentnosti procesa donacija. Tijekom izrade diplomskog rada ispunjeni su i osobni ciljevi, a to su da naučim Solidity programski jezik i napravim svoju prvu web aplikaciju u Reactu. Web3 neće zamijeniti web2, ali će otvoriti novi kanal za korištenje Interneta.

Bibliografija

[1] s Interneta, Prosječni troškovi Ethereum transakcija, https://bitinfocharts.com/comparison/ethereum-transactionfees.html#3y, 5. travnja 2023.

[2] s Interneta, Prosječni troškovi Bitcoin transakcija, https://www.statista.com/statistics/1224286/transaction-fees-bitcoin/, 5. travnja 2023.

[3] s Interneta, Naknada za transakcije na Polygonu, https://dune.com/queries/1538703/2579471, 2. svibnja 2023.

[4] s Interneta, Naknada za transakcije kreditnim karticama, https://www.forbes.com/advisor/business/credit-card-processing-fees/, 2. svibnja 2023.

[5] s Interneta, Prosjek naknada za transakcije debitnim karticama, https://merchantcostconsulting.com/lower-credit-card-processing-fees/debit-card-processing-fees-explained/, 2. svibnja 2023.

[6] s Interneta, Razlika između web3 i web2, https://www.skiplevel.co/blog/What-is-Web3-Breaking-down-Web2-vs-Web3, 4. travnja 2023.

[7] s Interneta, The Giving Block web stranica, https://thegivingblock.com/, 4. travnja 2023.

[8] s Interneta, Primjer The Giving Block widgeta za donacije na web stranici Save the children neprofitne organizacije, https://support.savethechildren.org/site/SPageNavigator/donation__crypto.html, 4. travnja 2023.

[9] s Interneta, Juicebox web platforma za financiranje kripto projekata, https://juicebox.money/, 4. travnja 2023.

[10] s Interneta, Juicebox izrada projekta, https://docs.juicebox.money/user/, 4. travnja 2023.

[11] s Interneta, Kickstarter web stranica, https://www.kickstarter.com/, 2. svibnja 2023.

[12] s Interneta, Usporedba Kickstartera, GoFundMe i Indiegogo platforma za podupiranje projekata, https://grasshopper.com/resources/tools/crowdfunding-platforms-kickstarter-gofundme-indiegogo/, 4. travnja 2023.

[13] s Interneta, GoFundMe web stranica, https://www.gofundme.com/, 2. svibnja 2023.

[14] s Interneta, IndieGoGo web stranica, https://www.indiegogo.com/, 2. svibnja 2023.

[15] s Interneta, Detaljnija usporedba Kickstartera, GoFundMe i Indiegogo https://www.perkfection.com/Blog/indiegogo-vs-kickstarter-vs-gofundme, 4. travnja 2023.

[16] s Interneta, Progamski jezici pametnih ugovora, https://moralis.io/top-smart-contract-programming-languages-for-blockchain-developers/, 4. travnja 2023.

[17] s Interneta, Što su pametni ugovori, https://www.cryptoninjas.net/what-are-smart-contracts/, 4. travnja 2023.

[18] s Interneta, Solidity dokumentacija, https://docs.soliditylang.org/en/v0.8.19/, 4. travnja 2023.

[19] s Interneta, Članak o Solidity kompajleru, https://www.alchemy.com/overviews/solidity-compiler, 4. travnja 2023.

[20] s Interneta, Globalne varijable u Solidityu, https://docs.soliditylang.org/en/v0.8.19/cheatsheet.html#global-variables, 4. travnja 2023.

[21] s Interneta, Polygon arhitektura, https://wiki.polygon.technology/docs/pos/polygon-architecture, 4. travnja 2023.

[22] s Interneta, Go Ethereum, https://www.mycryptopedia.com/what-is-go-ethereum-a-detailed-guide/, 5. svibnja 2023.

[23] s Interneta, Polygon Bor dokumentacija, https://wiki.polygon.technology/docs/pos/bor/, 8. travnja 2023.

[24] s Interneta, Polygon sporedni lanci, https://medium.com/momentum6/polygon-overview-layer-2-or-sidechain-a888104f5ffc, 5. travnja 2023.

[25] s Interneta, Slika ThiedWeb platforme, `https://portal.thirdweb.com/solidity`, 6. travnja 2023.

[26] s Interneta, ThiedWeb dokumentacija, `https://portal.thirdweb.com/`, 6. travnja 2023.

[27] s Interneta, Popis ThiedWeb gotovih pametnih ugovora, `https://thirdweb.com/explore`, 6. travnja 2023.

[28] s Interneta, IPFS dokumentacija, `https://docs.ipfs.tech/`, 7. travnja 2023.

[29] s Interneta, Kako radi IPFS, `https://docs.ipfs.tech/concepts/how-ipfs-works/#subsystems-overview`, 7. travnja 2023.

[30] s Interneta, Filecoin dogovori, `https://filecoin.io/blog/posts/how-storage-and-retrieval-deals-work-on-filecoin/`, 15. svibnja 2023.

[31] s Interneta, Filecoin dokumentacija, `https://docs.filecoin.io/basics/what-is-filecoin/overview/`, 7. travnja 2023.

[32] s Interneta, Filecoin dokazi, `https://docs.filecoin.io/basics/the-blockchain/proofs/`, 7. travnja 2023.

[33] s Interneta, Filecoin konsenzus, `https://docs.filecoin.io/basics/the-blockchain/consensus/`, 7. travnja 2023.

[34] s Interneta, članak o Web3 storage, `https://moralis.io/an-introduction-to-web3-storage-what-is-it-and-how-does-it-work/`, 17. travnja 2023.

[35] s Interneta, Web3 storage arhitektura, `https://blog.web3.storage/posts/web3-storage-architecture`, 17. travnja 2023.

[36] s Interneta, Wallet Connect popis podržanih kripto novčanika, `https://explorer.walletconnect.com/`, 17. travnja 2023.

Pojmovnik

AMM Automated market maker. 61, 62

API Application Programming Interface. 20, 30, 50

BEM Block-Element-Modifier. 40

BFT Byzantine Fault Tolerance. 33

CAR Content Addressed Archiver. 50, 51

CDN Content delivery network. 8

CID Content Identifier. 44, 45, 47, 49–51

CSS Cascading Style Sheets. 38, 40, 41

DAG Directed acyclic graph. 50

DAO Decentralized Autonomous Organisation. 14

DApps Decentralized applications. 1

DeFi Decentralized finance. 24, 35, 36

DHT Distributed hash table. 45, 46

DOM Document Object Model. 38, 39

EC Expected Consensus. 48

EOA Externally Owned Account. 31

EVM Ethereum Virtual Machine. 22, 24–26, 31, 32, 34, 36

HTML HyperText Markup Language. 38, 40, 41

HTTP Hypertext Transfer Protocol. 51

IPFS InterPlanetary File System. 6, 8, 21, 44–52, 65, 77, 78, 82

JSON JavaScript Object Notation. 75

JSX JavaScript XML. 38, 39

mDNS Multicast Domain Name System. 46

NFT Non-fungible token. 6, 7, 14, 15, 21, 35–38, 43, 52, 60, 66, 67, 70, 72, 73, 76–78, 81, 82

P2P Peer-to-peer. 6, 41, 44

PDF Portable Document Format. 60, 65, 81

PoRep Proof of Replication. 47

PoS Proof-of-Stake. 33, 80

PoSt Proof of Spacetime. 47

PoW Proof-of-Work. 80

SDK Software development kit. 20, 33, 41, 43

SPA single-page application. 39

TLS Transport Layer Security. 46

UI User interface. 38

URI Uniform Resource Identifier. 77, 78, 80

URL Uniform Resource Locator. 8, 65

USD United States dollar. 60, 70

Sažetak

Ovaj rad pokušava poboljšati sustav donacija implementacijom Web3 platforme za donacije. Problemi koji se riješavaju tom implementacijom su: velike naknade kod slanja donacija, ograničeni globalni doseg, manjak privatnosti, spore transakcije, manjak transparentnosti i sigurnosti. Također su istražene i implementirane sve potrebne tehnologije za riješavanje tih problema. Razvijene funkcionalnosti Web3 platforme za donacije su: izrada i uređivanje donacijskih kampanja, učitavanje PDF dokumenata donacijske kampanje na IPFS, prikaz informativnog sadržaja o kampanji, tražilica donacijskih kampanja, mogućnost razmjene tokena prije donacije, dodjela NFT nagrada za veće donacije, anonimna identifikacija korisnika putem povezivanja na platformu s kriptonovčanikom.

Ključne riječi — **Blockchain, Web3, React, Solidity, Pametni ugovori**

Abstract

This paper attempts to improve the donation system by implementing Web3 platform for donations. The problems that are being addressed by this implementation are: high fees for sending donations, limited global reach, lack of privacy, slow transactions, lack of transparency and security. Additionally, all necessary technologies for addressing these problems have been researched and implemented. Developed functionalities of the Web3 donation platform are: creating and editing donation campaigns, uploading donation campaign PDF documents to IPFS, displaying informative content about the campaign, a search bar for donation campaigns, the ability to exchange tokens before donation, awarding of larger donations with NFT rewards, anonymous user identification by connecting to the platform with a crypto wallet.

Keywords — **Blockchain, Web3, React, Solidity, Smart Contracts**

Dodatak A

Izvorni kod pametnih ugovora i web aplikacije

Na ovoj poveznici https://github.com/Ostix626/Web3-platforma-za-donacije priložen je dodatak A koji sadrži izvorni kod web3 aplikacije platforme za donacije i kodove CrowdFunding i DonorNFT pametnih ugovora.