Modeliranje blockchain aplikacija primjenom Truffle alata

Kukić, Filip

Sveučilište Jurja Dobrile u Puli

Tehnički fakultet u Puli

Filip Kukić

Modeliranje blockchain aplikacija primjenom Truffle alata

Završni rad

Pula, Rujan, 2022

Sveučilište Jurja Dobrile u Puli

Tehnički fakultet u Puli

Filip Kukić

Modeliranje blockchain aplikacija primjenom Truffle alata

Završni rad

JMBAG: 0303090364, redovan student

Studijski smjer: računarstvo

Znanstveno područje: Tehničke znanosti

Znanstveno polje: Računarstvo

Predmet: Programsko Inženjerstvo

Mentor: prof. dr. sc. Tihana Galinac Grbac

Pula, Rujan, 2022

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani_____, kandidat za prvostupnika _____ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljeni način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____

U Puli, _____

IZJAVA O KORIŠTENJU AUTORSKOG DJELA

Ja,_____dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj Završni rad pod nazivom

_____

_____

koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganjejavnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____

Potpis

_____

Sadržaj

1. Uvod

Tema ovog završnog rada je Modeliranje blockchain aplikacija primjenom Truffle alata. Bitcoin se na hrvatski može prevesti kao tehnologija ulančanih blokova [1], u nastavku rada će se koristiti termin *blockchain*. Blockchain i distribuirane glavne knjige se smatraju revolucionarnim tehnologijama koje su se razvile u posljednje vrijeme a razvijene su sa ciljem da se zaobiđe centralizirani posrednik prilikom razmjene sadržaja u internet mreži, a na način da se kod korisnika u komunikaciji instaliraju distribuirane usluge od povjerenja. Prva primjena blockchaina a istodobno i najuspješnija jest kriptovaluta bitcoin. Prvo spominjanje korištenja blockchaina u vezi s digitalnim plaćanjima bilo je u knjizi „Bitcoin: A Peer-to-Peer Electronic Cash System“ 2008. godine [2].

Ali blockchain ima više koristi nego samo razmjenjivanja kriptovaluta. Pomoću blockchaina se mogu i stvarati decentralizirane aplikacije koje mogu imati veliki raspon mogućnosti. Za stvaranje takvih aplikacije mogu pomoći alati Truffle. Truffle suite je razvojno okruženje napravljeno za Ethereum koje sadrži nekoliko alata kao što su Truffle i Ganache pomoću kojih se razvijaju blockchain aplikacije. Ethereum je platforma otvorenog koda pomoću koje se mogu stvarati decentralizirane aplikacije na blockchainu koje rade pomoću takozvanih pametnih ugovora. Pametni ugovori su programi koji rade na blockchainu i koji odobravaju transakcije koje prime na temelju nekoliko postavljenih kriterija. Oni zamjenjuju ljude u odobravanju transakcija te tako ubrzavaju čitav proces i čine ga sigurnijim

Zasad su blockchain aplikacije u ranom razvoju ali se očekuje da će u sljedećih nekoliko godina postati norma te će razmjena sadržaja u nepouzdanom okruženju poput Internet mreže nemoguć zamisliti bez njih. U ovom radu će se definirati ključni pojmovi vezani uz blockchain i pametne ugovore. Nadalje, dati će pregled tehnologija koje se koriste u te svrhe. U studijskom slučaju pokazat će kako se izgrađuje blockchain aplikacija i pamenti ugovor pomoću Truffle tehnologije.

2. Pozadina

2.1. Što je Blockchain

„Blockchain je distribuirana baza podataka ili glavna knjiga koja se dijeli između čvorova računalne mreže.“ [3] Dobio je svoje ime prema načinu na koji funkcionira. Blockchain se može zamisliti kao datoteku u kojoj su podaci lančano povezani. Podaci se spremaju u takozvane blokove, kad se blok napuni stvori se novi koji je povezan sa starim te se na taj način stvara lanac između blokova. Glavna knjiga (leadger) je digitalna datoteka koju blockchain koristi za pohranu i praćenje svih transakcija. Blockchain funkcionira isto kao baza podataka budući da podatke pohranjuje elektronički u digitalnom formatu. Blockchaini imaju ključnu ulogu u sustavima kriptovaluta poput Bitcoina jer održavaju sigurnu i decentraliziranu evidenciju transakcija. Ali umjesto da svaki korisnik sprema podatke transakcije u vlastitim knjigama, koristi se jedna glavna knjiga u kojoj se javno mogu vidjeti sve transakcije na blockchainu. Taj način evidentiranja transakcija je i glavno obilježje blockchaina, a zove se decentralizacija.

2.2. Kakve probleme rješava i zašto je koristan?

Blockchain je koristan iz više razloga. Uklanja uključenost ljudi u proces verifikacije što poboljšava točnost. Blockchain radi na principu „peer-to-peer“ gdje se transakcije odvijaju direktno između dvije osobe. Drugim riječima ne postoji posrednik koji verificira transakcije. To uvelike smanjiva troškove i povećava brzinu obavljanja transakcije. Blockchain je još siguran, privatan i učinkovit. Djeluje kao alternativa bankarstvu, ali i rješava neke od njegovih problema. Za razliku od bankarstva, blockchain je stalno otvoren 24 sata dnevno, korisnici sami postavljaju naknade za transakcije, puno su veće brzine obavljanja transakcija i još mnogo toga. No najveći problem koji bitcoin riješava je problem dvostruke transakcije (double spending). Problem dvostruka transakcije se dogodi kada osoba pokušava poslati isti novac dvjema osobama istodobno. Banke rješavaju taj problem pomoću treće strane koja nadgledava transakciju i pomoću serijskih brojeva, ali to rješenje nije moguće kod blockchaina jer je ono obavlja transakcije bez treće strane. Blockchain rješava taj problem na način da dopusti čitavom sustavu da verificira zakonitost transakcije. Samo ako se većina sudionika složi da je transakcija

zakonita se ona može provesti. Ovaj način rješavanja problema je sličan takozvanom problemu Bizantskih generala. [4]

2.3. Primjena blockchain tehnologije

Kada je blockchain tehnologija prvi put definirana 1991. godine, htjeli su je upotrijebiti za stvaranje sustava u kojem neovlaštene osobe ne bi mogle promijeniti vremenske oznake u dokumentima. No trebalo je gotovo dva desetljeća da blockchain tehnologija konačno dobije stvarnu primjenu kad je 2009. godine lansiran Bitcoin. Bitcoin koristi blockchain kao način za transparentno bilježenje knjige plaćanja, no u teoriji blockchain se može koristiti za bilježenje bilo kojeg broja podatkovnih točaka. Blockchain se tako može koristiti u obliku raznih transakcija, glasovanja na izborima, popisivanje inventara proizvoda, identifikacije stanja, praćenja isprava kuća i u raznim drugim oblicima.

2.4. Kriptovalute

Kriptovalute, skraćeno kripto, su novija vrsta novca koji je digitalan i kojeg pokreta kriptografija. Prva moderna i najpoznatija kriptovaluta je Bitcoin koji je 2009. godine objavljen. Prednost kripta, za razliku od normalnih banki ili Paypal-a, je ta što kripto nema treće strane, odnosno srednjeg čovjeka koji nadgledava transakciju i ima moć ju zaustaviti ili dijeliti informacije o njima. Kripto nam omogućuje sigurno i privatno obavljanje transakcija s bilo kojom osobom na svijetu. Transakcije se obavljaju direktno između pošiljatelja i primatelja.

Ethereum ima vlastitu kriptovalutu koja se zove ether (ETH). Kao i ostale kriptovalute ether je samo digitalan [5]. Količina ethera na tržištu ne kontrolira ni jedna kompanija ili vlada nego je decentralizirano i potpuno transparentno. Nove novčanice ethera, koje se nekad zovu i tokeni, se samo mogu stvoriti rudarenjem ili ih stvaraju dionici koji održavaju mrežu. Rudarenje je proces u kojem globalna mreža kompjutera rade kod koji osigurava da se transakcije odvijaju legitimno i dodaju na blockchain na odgovarajući način, čime su nagrađeni s određenom količinom kripta [6]. Rudarenje je jako zahtjevan proces za kompjutere te zahtjeva velike količine energije kako bi se obavio. Zbog toga je rudarenje samo profitabilno onima koji to rade na veliko sa stotinama procesora odjednom. Proces rudarenja

ethera je jako sličan rudarenju Bitcoina samo što kod ethera rudari mogu sami staviti cijenu koliko će naplaćivati potvrđivanje transakcija [7].

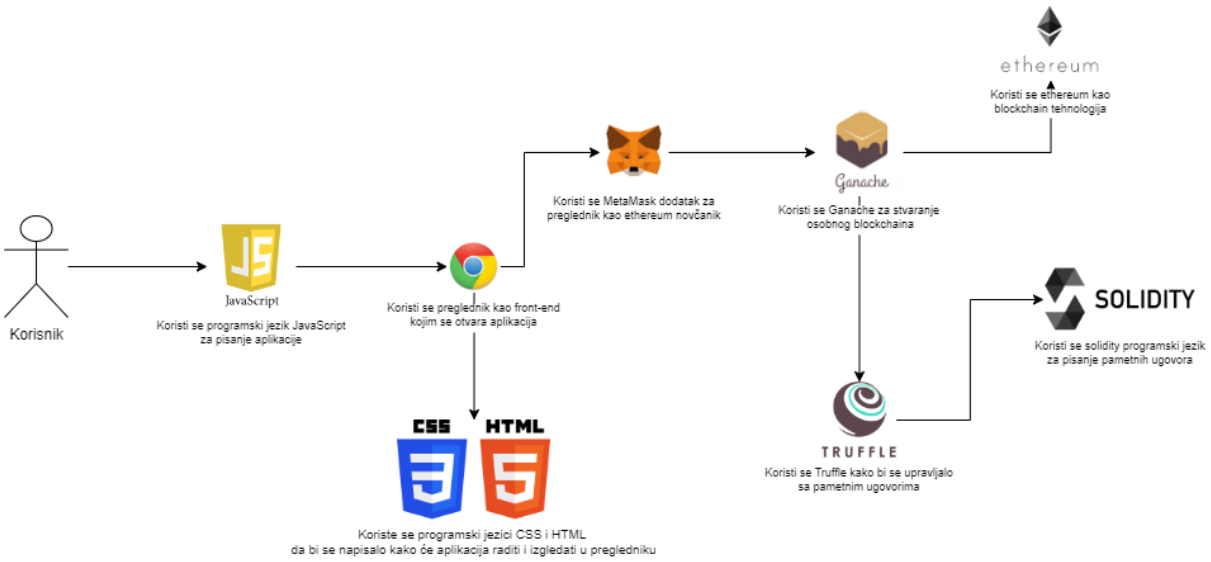
Osnovni nedostatak Ethereuma je to što svaka radnja na mreži zahtjeva neku količinu računalne snage koja se mora platiti. Plaća se s etherom i zove se plin (gas). Zbog toga kako bi se bilo što radilo na Ethereum mreži potrebno je imati neku minimalnu količinu ethera na računu.

3. Blockchain i pametni ugovori

Koristi se Ethereum kao blockchain tehnologija i Solidity programski jezik za pisanje pametnih ugovora. "Ethereum je nova generacija blockchain tehnologije čiji je cilj izgraditi opću nepovjerljivu knjigu s mogućnošću pokretanja programa koji se nazivaju "pametni ugovori".“[8] Ključna razlika od bitcoina je u tome što ethereum koristi EVM (ethereum virtualni stroj) povrh blockchaina. EVM omogućuje stvaranje decentraliziranih blockchain aplikacija koje koriste blockchain kako bi pohranjivali podatke i obrađivali ih. Zbog toga blockchain pomoću ethereuma postaje opće namjene gdje programeri odlučuju šta će aplikacije raditi.

Pametni ugovori su kompjuterski programi koji se nalaze na blockchainu i oni su temeljni dio građe Ethereum aplikacija [9]. Pojam ugovor podrazumijeva sporazum između sudionika ugovora dok pojam pametni označava da se ovi ugovori razlikuju od običnih po tome što ovi rade bez utjecaja ljudi odnosno potpuno samostalno te se sami aktiviraju tek kad se uvjeti ugovora ispune. Transakcije koje pametni ugovori primaju mogu doći od korisnika ili čak drugih pametnih ugovora. U Ethereumu postoje dvije vrste korisničkih računa, a to su oni koji predstavljaju korisnike i oni koji predstavljaju pametne ugovore. Pametni ugovori kao i korisnici imaju vlastitu adresu. Dok korisnici koriste adrese kako bi međusobno slali kriptovalute jedni drugima, pametni ugovori ih koriste kako bi primali transakcije ili druge upite. Pametni ugovori dok čekaju da prime transakciju stoje neaktivni na blockchainu. Tek kad prime upit se aktiviraju, odnosno njihov kod pregledava transakciju da vidi dali ispunjava sve uvjete. Ako ih transakcija ispunjava, onda je ugovor odobri i kriptovalute se prebacuju s računa pošiljatelja na račun primatelja preko blockchaina.

4. Modeliranje blockchaina sa Truffle alatima



Slika 1. Arhitektura Truffle blockchain aplikacije

Izvori: Napravljeno u https://app.diagrams.net/

Javascript: https://quintagroup.com/cms/js/js-image/javascript-logo.png/view

Google Chrome: https://commons.wikimedia.org/wiki/File:Google_Chrome_icon_(February_2022).svg

CSS i HTML: https://commons.wikimedia.org/wiki/File:CSS3_and_HTML5_logos_and_wordmarks.svg

MetaMask: https://commons.wikimedia.org/wiki/File:MetaMask_Fox.svg

Ganache: https://seeklogo.com/vector-logo/426725/ganache

Ethereum: https://artistsatrisk.org/donations/ethereum-logo-2/?lang=en

Truffle:https://medium.com/heartbanklab/how-truffle-works-under-the-hood-f1ff6add416c,

Solidity: https://en.bitcoinwiki.org/wiki/Solidity

4.1. Arhitektura s gledišta korisnika

Arhitektura blockchain modela prikazana je u slici 1. Korisnik otvara aplikaciju u pregledniku. Najkorišteniji preglednik prilikom stvaranja aplikacija je Google Chrome. Najpopularniji je preglednik na svijetu te zbog toga otklanja mogućnost dobivanja problema s kompatibilnošću aplikacije i preglednika koju bi mogli dobiti kod drugih preglednika. Još jedna prednost Google Chrome-a je ta što je jako lagano instalirati dodatak MetaMask koji je potreban prilikom korištenja blockchain aplikacije. Google Chrome i MetaMask čine front end i jedini su dio aplikacije koju

korisnici vide. MetaMask je digitalni novčanik za kriptovalute koji omogućuje korištenje web3 aplikacije. Dostupan je kao aplikacija za telefone i kao dodatak za preglednike koja se češće koristi prilikom stvaranja aplikacija. Potrebno je koristiti dodatak kao što je MetaMask zbog sigurnosnih razloga i jednostavnosti. Zbog toga je potrebno samo programirati što aplikacija radi i kako izgleda, dok način na koji se obavljaju transakcije i upravljaju računi obavlja MetaMask. Kad se radi o novcima i transakcijama korisnici moraju znati da je to sigurno i da nema nikakve mogućnosti krađe što MetaMask omogućuje. Na MetaMask-u korisnik mora napraviti račun nakon čega može kupiti ether s čime obavlja transakcije. Prilikom stvaranja aplikacije uvozi se račun na kojem već ima lažnog ether-a s čime korisnik plaća transakcije u aplikaciji. Nakon što korisnik ima MetaMask instaliran i račun napravljen može koristiti aplikaciju.

4.2. Arhitektura s gledišta programera

4.2.1. JavaScript

Glavni dio aplikacije je napisan u programskom jeziku JavaScript. Pod glavnim dijelom se odnosi na onaj koji govori što će aplikacija ustvari raditi. JavaScript je skriptni programski jezik kojim se može napraviti kompleksne značajke na web stranicama. [10] U aplikaciji, popisi podataka su napisani u obliku JSON-a. JSON (JavaScript Object Notation) je JavaScript-ov format za spremanje podataka u obliku objekta. JSON je jednostavan za razumjeti i pisati programerima i lagan za računala da čitaju i kompiliraju te je zbog tih razloga puno korišten. Jako važno svojstvo JSON-a je da je neovisan o programskom jeziku. Zbog toga aplikacije mogu komunicirati i razmjenjivati podatke pomoću JSON-a iako su obadvije napisane u drugim programskim jezicima. [11]

4.2.2. HTML i CSS

HTML je stilski jezik koji se koristi za stvaranje web aplikacija i stranica. S obzirom na to da se aplikacija otvara i koristi preko preglednika koristi se HTML kako bi stvorili potrebne elemente koje omogućuju da aplikacija funkcionira. Elementi mogu biti tablice, dugmad, odlomci teksta, slike i drugo.

CSS je također stilski jezik koji se koristi kako bi stilizirali HTML elemente. Korištenjem samo HTML-a bi stranica izgledala jako dosadno te bi sve stranice izgledale slično. Korištenjem CSS-a aplikacija može biti puno ljepša te programeri mogu biti puno kreativniji. [12]

Za pisanje HTML-a i CSS-a se u aplikaciji koristio Bootstrap. Bootstrap je besplatan okvir otvorenog koda koji se koristi za pisanje web stranica. On sadrži predloške JavaScript, HTML i CSS koda koji se mogu koristiti za lakše stvaranje stranica. [13]

4.2.3. Ganache i Ethereum

Za rad aplikacije potreban je blockchain na kojem će raditi. A budući da to nije prava aplikacija i ne koristi pravu kriptovalutu, koristi se alat Truffle koji se zove Ganache za stvaranje blockchaina. Truffle Ganache je aplikacija pomoću koje se može stvoriti vlastiti blockchain na kojem se mogu stvarati i testirati web3 aplikacije. Bez Ganache-a se pametni ugovori moraju testirati na stvarnom Ethereum blockchainu što bi koštalo pri svakom testiranju, te bi trebalo više vremena za svaki test. Taj blockchain je pokrenut na jednom računalu lokalno te se pristupa internetskom vezom. Ganache koristi Ethereum tehnologiju kako bi stvorio blockhain. Općenito se koristi kako bi se simulirao blockchain, ali se on može i koristiti kako bi programeri testirali svoje web3 aplikacije.

4.2.4. Truffle i Solidity

Truffle je okvir baziran na javi koji se koristi za implementaciju učinkovitih AST. AST je kratica na engleskom za abstract syntax tree i može se na hrvatski prevest kao stablo sažete sintakse. AST je graf oblika stabla koji prikazuje apstraktnu sintaktičku strukturu teksta ili najčešće izvornog koda. Truffle dinamički specijalizira AST na temelju izvršenih putova i promatranih putova. Truffle okvir je temeljen na JVM odnosno java virtualnoj mašini. Zbog toga Truffle može koristiti njegove usluge kao što su sakupljači smeća, optimizatori strojnog koda, podrška nitima (threads) itd. [14]

Solidity je jedan od glavnih tehnologija kojim se pišu pametni ugovori na Ethereum platformama. Soliditiy je objektno orijentiran jezik visoke razine te mu je sintaksa slična C++. Kako bi razmjenjivali ether Solidity ima nekoliko funkcija od kojih su

najkorištenije call, transfer i send. Funkcija call je sučelje niske razine koje se koristi za slanje poruka pametnim ugovorima, ali se može i koristiti za slane ethera drugim adresama. Funkcija zvana transfer su stvorili 2017. godine i namijenjena je razmjeni kriptovaluta. U slučaju da transakcija ne uspije dogodi se takozvani izuzetak (exception) što znači da se transakcija automatski vraća pošiljatelju. Funkcija send je manja verzija funkcije transfer, jedina razlika je ta što nema mogućnost izuzetka u slučaju neuspjele transakcije šalje grešku programeru da je on riješi. Soliditi također ima i takozvane Solidity Guards što u prijevodu znači čuvari. Ti čuvari su jezične konstrukcije koje nadgledavaju transakcije kako ne bi bilo prijevara. [15]

5. Stvaranje blockchain aplikacije pomoću Truffle

5.1. Truffle

Program Truffle je, zajedno s druga dva programa Ganache i Drizzle dio razvojnog okruženja Truffle Suite. Truffle je program pomoću kojeg se stvaraju i razvijaju decentralizirane aplikacije te se također koristi i za pisanje i testiranje pametnih ugovora. Kako bi programer mogao početi stvarati blockchain aplikaciju mora imati Truffle instaliran. Truffle se instalira pomoću Node.js-a sa naredbom „npm install -g truffle“.

Slika 2. Logo Truffle-a

Izvor: https://medium.com/heartbanklab/how-truffle-works-under-the-hood-f1ff6add416c

5.2. Truffle box

Nakon što se preuzme Truffle, prvi korak stvaranja aplikacije jest preuzimanje Truffle box-a. Truffle box (Truffle kutija) je podloga za stvaranje decentraliziranih aplikacija koja se može preuzeti pomoću Truffle-a. Truffle box sadrži osnovan kod koji je potreban svakoj decentraliziranoj aplikaciji. Truffle je stvorio te kutije kako bi olakšao programerima proces stvaranja aplikacija. Na taj način programeri imaju podlogu od koje početi stvarati, te se mogu koncentrirati na šta će aplikacija raditi i kako izgledati. Truffle nudi mnogo različitih kutija svaka sa različitim kodom koji može pomoći pri stvaranju decentraliziranih aplikacija. Kako bi preuzeo određenu Truffle kutiju,

programer unosi naredbu „truffle unbox ____“ zajedno sa imenom Truffle kutije koje želi preuzeti.

Truffle kutije koje se koriste kao podloga za stvaranje decentraliziranih aplikacija sadrže osnovni kod kako bi počeli stvarati. Sadrže jedan osnovan pametni ugovor koji nam može služiti kao primjer kako napisati svoj, sadrži jednu migracijsku skriptu koja migrira taj pametan ugovor te još sadrži sve potrebne datoteke kojim će se aplikacija pokretati. Te se datoteke sastoje od JavaScript, HTML i CSS koda kojim se aplikacija pokreće u odabranom pregledniku.

5.3. Pisanje pametnih ugovora

Preuzimanjem Truffle box-a programer već ima podlogu te može odmah krenuti pisati šta će njegova aplikacija raditi. Najvažniji dio decentraliziranih aplikacija su pametni ugovori. Pametni ugovori su vrsta programa koji se nalaze na blockchainu, oni odobravaju upite odnosno transakcije koje prime. Decentralizirane aplikacije koriste pametne ugovore kako bi pregledavali legitimnost upita koje im šalje, odnosno dali korisnik aplikacije zadovoljava sve uvijete za obaviti transakciju. Zbog toga su glavno obilježje blockchain aplikacija te se prilikom njihovog pisanja mora obratiti dodatna pozornost.

Pametni ugovori se pišu u programskom jeziku Solidity, te kako bi programer mogao početi pisati mora ga imati preuzetog. Mogu se pisati u bilo kojem uređivaču koda kao naprimjer Visual Studio Code, ali samo ako podržava Solidity programski jezik. Pametni ugovori su vrsta objekta koje Solidity naziva „Contracts“ odnosno ugovor. Ali prije nego što se počne pisati ugovor mora se navesti minimalna verzija Solidity-a na koju će ugovor raditi. To se radi kako bi se osigurao siguran rad ugovora te kako ne bi došlo do grešaka zbog korištenja manjih verzija. Nakon toga se stvara objekt ugovora koji programer imenuje po želji. Unutar objekta se piše kod ugovora koji ovisi o tome šta će ugovor raditi. Radnje koje će ugovor raditi se pišu u obliku funkcija. Ugovor će imati onoliko funkcija koliko programer želi da ima radnji, odnosno koliko stvari će provjeravati. Osim funkcija u ugovoru se i definira njegova adresa. Svaki ugovor na blockchainu mora imati svoju jedinstvenu adresu na kojoj će primati upite. Inače ne bi bilo moguće komunicirati s njima.

Nakon što se napiše pametni ugovor on se mora kompilirati. Kompiliranje je proces u kojem se kod koji je napisao programer pretvara u izvorni kod kako bi bio razumljiv računalu. Umjesto računala pametne ugovore čita i obrađuje EVM. EVM je kratica za Ethereum virtual machine odnosno Ethereum virtualna mašina. Na EVM-u se ne vrte samo pametni ugovori nego i čitava decentralizirana aplikacija. U Truffle-u se kompiliraju pametni ugovori upisom naredbe “truffle compile”.

5.4. Migracijske skripte

Kako bi radili ono za što su namijenjeni, pametni ugovori moraju biti na blockchainu. Kako bi ih pokrenuo tamo Truffle koristi sustav migracija. Migriranje je kod Truffle-a proces u kojem se novi ugovori pokreću na blockchainu ali i proces u kojem se mogu i mijenjati podaci u postojećim ugovorima na blockchainu. Pomoću migracija se također može i mijenjati postojeće ugovore sa potpuno novima. Svaki pametan ugovor mora imati vlastitu migracijsku skriptu. Migracijske skripte nisu komplicirane te se najčešće sastoje od jedne funkcije koja migrira ugovor ključnom riječi deploy. Nakon što se napiše migracijska skripta ona se aktivira komandom „truffle migrate“, čime se pokreće odabrani pametni ugovor na blockchain.

5.5. Ganache

Migriranjem se pametni ugovor pokreće na blockchain. No migriranje košta određenu količinu ethera koja se zove gas odnosno plin. Prilikom stvaranja aplikacije programer konstantno migrira nove ugovore na blockchain ili mijenja postojeće čime bi morao svaki put platiti troškove migriranja. Zbog toga je Truffle stvorio program Ganache. Ganache je zajedno sa Truffle-om dio Truffle Suite programskog okruženja. Ganache je program kojim se lokalno pokreće osobni blockchain koji programer može koristiti kako bi testirao svoje aplikacije prije nego ih pokrene na pravom blockchainu. Koristeći Ganache programer sačuva novac te i vrijeme jer migriranje na pravi blockchain ne samo da košta nego i traje neko vrijeme. S obzirom da Ganache pokreće blockchain lokalno na računalu programera, nema nikakvog čekanja te se migriranje odvija u istom trenu.

Slika 3. Logo Ganache-a

Izvor: https://trufflesuite.com/blog/ethereum-gas-exactimation/

Ganache se preuzima s službene stranice Truffle-a. Nakon što se preuzme i otvori može se stvoriti blockchain. Ganache ima dva načina na koji stvara blockchain, a to su quickstart odnosno brzo pokretanje i novi radni prostor. Quickstart je funkcija koju Ganache ima kojom se može jednim klikom stvoriti blockchain. Ta se opcija koristi kada je brzo potreban blockchain za testiranje a nije važno koje su mu postavke, odnosno želi se koristiti zadane postavke. Ako se više razumije u Ganache i za stvaranje aplikacije je potrebno napraviti blockchain s nekim posebnim postavkama onda se koristi novi radni prostor. Kolikom na to Ganache prije stvaranje blockchaina otvara prozor s postavkama u kojim programer bira kakve postavke želi.

5.6. Testiranje pametnih ugovora

Nakon što su se migrirali na blockchain, pametni ugovori su spremni za korištenje. Ali prije nego što se koriste moraju se testirati kako bi bili sigurni da su dobro napisani. S obzirom da programeri pišu vlastite pametne ugovore s jedinstvenim funkcijama, moraju i napisati testove koji će testirati dali svaka funkcija radi ono čemu je namjenjena. Testovi se mogu pisati u JavaScript-u ili Solidity-u. Test je kao i pametan ugovor objekt koji se sastoji od funkcija. Svaka funkcija je poseban test koji testira jednu funkciju pametnog ugovora. Testovi općenito rade na način da pošalju varijablu pametnom ugovoru te čekaju odgovor. Nakon toga varijablu iz odgovor pametnog ugovora uspoređuju s poslanom kako bi utvrdili dali je ugovor dobro odradio zahtjev.

Nakon što se napišu testovi pokreću se naredbom „truffle test“. Svi testovi radi lakšeg testiranja se pišu u jedan dokument te broj testova ovisi o broju funkcija. Output naredbe jesu popis svih testova zajedno s x ako nije prošao ili kvačicom ako jeste. Ako neki test ne prođe programer mora pogledati šta je taj test testirao te popraviti to

u kodu pametnog ugovora nakon čega mora ponovno migrirati ga na blockchain kako bi spremio promjene i ponovno pokrenuo test.

5.7. MetaMask

Kako bi korisnik plaćao transakcije prilikom korištenja aplikacije mora imati kripto novčanik na kojem ima odgovarajuću količinu ethera. Jedan od takvih novčanika je MetaMask. MetaMask je dodatak pregledniku koji služi za plaćanje transakcija s kripto valutama. MetaMask je dostupan na većini preglednika ka naprimjer Google Chrome te se preuzima na njihovim trgovinama. Novac se na MetaMask uplaćuje kupnjom određene kripto valute. Prilikom stvaranja blockchain aplikacije MetaMask se također može i koristiti za testiranje korištenjem lažnog kripta. To se radi na način da se spoji s Ganache-om te se uveze jedan od računa koji ima 100 lažnog ethera. Uvozi se s lozinkom koja se može dobiti u Ganache-u. Pomoću MetaMask-a tako programer može da simulira korištenje aplikacije i obavljanja transakcija kako bi vidio dali ima grešaka.

Slika 4. Logo MetaMask-a

Izvor: https://1000logos.net/metamask-logo/

5.8. Korisničko sučelje

Blockchain aplikacije se pokreću preko preglednika kao što je Google Chrome. Zbog toga se one pišu u JavaScript-u, HTML-u i CSS-u. Preuzimanjem Truffle kutije se dobije početan kod koji se može koristiti za stvaranje vlastite aplikacije. S obzirom da je to blockchain aplikacija preglednik na kojem se otvara mora biti kompatibilan s web3-om. Web3 je verzija WWW-a u kojoj se odvijaju decentralizirane aplikacije i druge blockchain tehnologije. Zbog toga je potrebno dodati u aplikaciju dio koda koji to provjerava.

Nakon što se napravi sučelje pokreće se lokalno kako bi se testiralo dali je ispravno napravljeno. Za pokretanje se mogu koristiti različiti programi kao što je naprimjer Lite server koji se nalazi u Truffle kutiji.

5.9. Stavljanje aplikacije u stvarnu upotrebu

Tek kad je programer siguran da svaki dio aplikacije radi kako treba može ga staviti u stvarnu upotrebu. To čini na način da prvo pametne ugovore migrira na pravi blockchain. Nakon toga sučelje aplikacije pokreće online umjesto lokalno kako bi svi mogli njemu pristupiti. I na kraju se može koristiti aplikacija pomoću MetaMask-a s pravim kriptom.

6. Zaključak

Cilj ovog rada bio je detaljno opisati proces stvaranja blockchain aplikacija primjenom Truffle alata. Stvaranje blockchain aplikacije je težak proces i zahtjeva znanje ne samo o blockchainu nego i o web stranicama. Ali korištenjem Truffle alata nam uvelike olakšava čitav proces. Čak i osobe koje nemaju nikakvo predznanje mogu pratiti neke od mnogih tutoriala za svaki program te razviti blockchain aplikaciju. Razvojno okruženje Truffle Suite sadrži tri programa: Truffle, Ganache i Drizzle od kojih smo prva dva koristili tijekom razvoja naše aplikacije u radu. Pomoću Ganache smo napravili privatni blockchain na kojem se vršile transakcije naše aplikacije. A koristili smo Truffle kako bismo skinuli Truffle box kojim smo dobili već godov dio aplikacije za početak te smo ga koristili prilikom pisanja i testiranja pametnih ugovora. Blockchain aplikacije su budućnost interneta, a pomoću alata kao Truffle neiskusni programeri ih mogu početi razvijati i naučiti važne vještine koje će im pomoći tijekom stvaranja budućih blockchain aplikacija.

Literatura

[1] Andro Babić, „Terminologija Ulančanih Blokova na Hrvatskome Jeziku“, https://www.ffzg.unizg.hr/hieronymus/wp-content/uploads/2020/03/H6-2019_2_Babic.pdf

[2] Nakamoto, Satoshi. 2008. Bitcoin: A peer-to-peer electronic cash system.URL: https://bitcoin.org/bitcoin.pdf

[3] ”What Is a Blockchain?” https://www.investopedia.com/terms/b/blockchain.asp (pristupljeno 8.7.2022-)

[4] Tschorsch F i B. Scheuermann, „Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies“, 2016.

[5] Ethereum.org, https://ethereum.org/en/what-is-ethereum/ (pristupljeno 9.9.2022.)

[6] Matt Whittaker, „How Does Bitcoin Mining Work?“ https://www.forbes.com/advisor/investing/cryptocurrency/bitcoin-mining/ (pristupljeno 8.9.2022.)

[7] plus500, „What is the difference between Ethereum and Bitcoin?“ https://www.plus500.com/en-CZ/Instruments/ETHUSD/What-is-the-difference-between-Ethereum-and-Bitcoin~2 (pristupljeno 8.9.2022.)

[8] Kirkman S. i R. E. Newman, "Using Smart Contracts and Blockchains to Support Consumer Trust Across Distributed Clouds", 2017. https://www.researchgate.net/publication/317057712_Using_Smart_Contracts_and_Blockchains_to_Support_Consumer_Trust_Across_Distributed_Clouds, (pristupljeno 1. 7. 2022.)

[9] Ethereum.org, „Introduction to smart contracts“, https://ethereum.org/en/smart-contracts/ (pristupljeno 9.9.2022.)

[10] mdn web docs, „What is JavaScript?“, https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript (pristupljeno 10.9.2022.)

[11] json.org, „Introducing JSON“, https://www.json.org/json-en.html (pristupljeno 10.9.2022.)

[12] Aryan Gupta, „HTML vs. CSS: The Best Guide to Understand the Difference" https://www.simplilearn.com/tutorials/html-tutorial/html-vs-css (pristupljeno 11.9.2022.)

[13] bootstrap, „Build fast, responsive sites with Bootstrap“ https://getbootstrap.com/ (pristupljeno 11.9.2022.)

[14] S. Marr, T. Pape i W. De Meuter, „Are We There Yet? Simple Language Implementation Techniques“, 2014.

[15] Verheijke D. i H. Rocha, „An Exploratory Study on Solidity Guards and Ether Exchange Constructs“, 2022.

[16] „Truffle suite Pet Shop“, https://trufflesuite.com/guides/pet-shop/

[17] Schär F. i A. Berentsen, *Bitcoin, Blockchain, and Cryptoassets A Comprehensive Introduction*, The MIT Press, 2020.

Popis slika