Provjera koncepta decentralizirane aplikacije za izdvajanje izračuna na blockchainu

Grenko, Fran

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Preddiplomski studij računarstva

Završni rad

Provjera koncepta decentralizirane aplikacije za izdvajanje izračuna na blockchainu

Rijeka, rujan 2022.

Fran Grenko
0069085585

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Preddiplomski studij računarstva

Završni rad

Provjera koncepta decentralizirane aplikacije za izdvajanje izračuna na blockchainu

Mentor: prof. dr. sc. Kristijan Lenac

Rijeka, rujan 2022.

Fran Grenko
0069085585

Umjesto ove stranice umetnuti zadatak za završni ili diplomski rad

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, rujan 2022.

_____

Fran Grenko

Zahvala

Zahvaljujem profesoru Lencu na podršci tijekom pisanja ovoga rada i korisnim raspravama i savjetima. Zahvaljujem svojoj obitelji i djevojci na podršci tijekom studiranja.

Sadržaj

Popis slika

Popis tablica

Poglavlje 1

Uvod

S povećanom popularnošću kriptovaluta i NFT-a koji oboje koriste blockchain tehnologiju, sve više ljudi se susrelo i s pojmom "blockchain". Blockchain, ili doslovno prevedeno lanac blokova, je distribuirana baza podataka (eng. *ledger*) koja se dijeli između čvorova unutar mreže. Iako se za pohranu podataka može koristiti bilo koja konvencionalna baza podataka, blockchain je jedinstven po tome što je potpuno decentraliziran. Javne blockchain mreže se osiguravaju konsenzus mehanizmima, pomoću kojih sudionici blockchain mreže sami provjeravaju ispravnost podataka. Proof of Work je konsenzus mehanizam čiji sudionici, da bi potvrdili ispravnost podataka, moraju riješiti zahtjevni matematički zadatak. Sudionici su nagrađeni za točno rješenje i time ih se potiče da šalju ispravna rješenja. Najpoznatija kriptovaluta Bitcoin koristi upravo Proof of Work konsenzus mehanizam. Kako je rasla popularnost Bitcoina, tako se i povećavao broj sudionika u njegovoj mreži i time se izrazila njegova loša energetska efikasnost, pošto svaki čvor za svaki matematički zadatak potroši značajnu količinu električne energije. Zbog velike količine sudionika u Bitcoin mreži i dizajna konsenzusa na čijem principu Bitcoin funkcionira, Bitcoin mreža je postala iznimno veliki potrošač električne energije. Ovaj rad se bavi potencijalnim rješenjem koje bi bilo znatno ekonomičnije za implementaciju budućih blockchain mreža, uz zadržavanje prednosti Proof of Work konsenzusa. Takvo rješenje bi bila decentralizirana aplikacija za izdvajanje i provjeru izračuna, u kojoj je umjesto natjecateljskog rješavanja zadataka, implementirano zajedničko rješavanje, te se tako izbjegava uzaludno utrošeni posao. U radu su objašnjeni glavni principi decentralizirane aplikacije

i prikazani su njeni dijelovi i funkcionalnosti.

Poglavlje 2

Blockchain

2.1 Uvod

U ovom poglavlju će se objasniti blockchain, njegove karakteristike i arhitektura. Podijelit ćemo ga na vrste i usporediti razlike.

2.2 Blockchain

Blockchain je slijed blokova koji sadrže kompletnu listu zapisa transakcija, poput javne digitalne knjige (*eng. ledger*). Blokovi su povezani u jednosmjernom lancu, gdje svaki novi blok sadrži informacije o prethodnom. Kod modifikacije podataka unutar jednog bloka je potrebno promijeniti podatke i u prijašnjim blokovima, što je jedno od ključnih sigurnosnih značajka blockchaina. [1]

2.3 Povijest

Blockchain je 2008. godine predložio Satoshi Nakamoto, čiji je pravi identitet do današnjeg dana nepoznat. Napravljen je kako bi služio kao javna distribuirana glavna knjiga za transakcije kriptovalute Bitcoin. Implementacija blockchaina u Bitcoinu je prvi primjer kriptovalute koji je gotovo uklonio rizik od napada "51%" (kojeg

ćemo objasniti u sljedećem poglavlju), bez korištenja glavnog servera. Neposredno nakon su i druge kriptovalute slijedile primjer Bitcoina , što je rezultiralo eksplozijom popularnosti kriptovaluta, a i blockchain tehnologije. [2]

2.4 Blockchain arhitektura

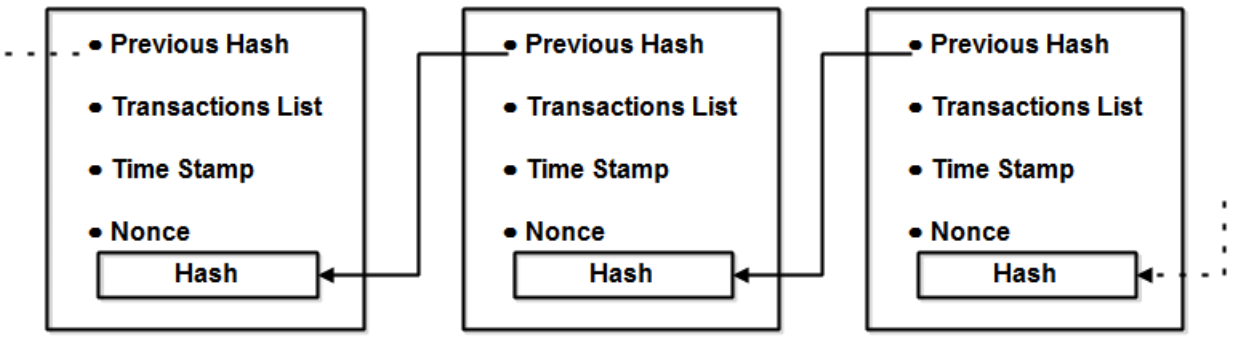
Kao što smo prije spomenuli, blockchain se sastoji od rastućih zapisa koje nazivamo blokovima, koji su jednosmjerno povezani koristeći se kriptografijom. Prvi blok u blockchainu se naziva *genesis* blok , koji ne sadrži informaciju o bloku roditelja (*eng. parent*), dok svi ostali blokovi sadrže i hash. [1] roditelja.

Blok se sastoji od zaglavlja (*eng. header*) i tijela (*eng. body*). *Zaglavlje* sadrži sljedeće informacije:

- *Verzija bloka* - označuje koji skup pravila blok mora ispunjavati
- *Hash prethodnog bloka*
- *Hash Merkle root* - hash generiran na temelju podataka svih transakcija u bloku
- *Vremenska oznaka* - vremenska oznaka kada je blok pridružen blockchainu
- *Bitovi* - težina za traženje noncea
- *Nonce* - enkriptirani broj koji rudar mora riješiti kako bi verificirao blok i zatvorio ga

Tijelo bloka se sastoji od brojača transakcija i transakcija. Maksimalni broj transakcija koje blok može sadržavati zavisi od veličine bloka i veličine svake transakcije.

[1] Hash - string dobiven enkripcijom koji se ne može dekriptirati.



Slika 2.1 Pojednostavljeni grafički prikaz blokova u blockchainu

Kako bi se osigurao integritet bloka, transakcija i podataka koje oni sadrže, koristi se **digitalno potpisivanje** (*eng. digital signature.*). Svaki korisnik blockchaina posjeduje javni i privatni ključ, gdje se privatnim ključem potpisuju transakcije. Digitalno potpisivanje sadrži dvije faze: *potpisivanje* i *verificiranje.* U fazi potpisivanja korisnik koji šalje transakciju enkriptira podatke sa svojim privatnim ključem te šalje rezultat enkripcije i originalne podatke. Zatim u fazi verifikacije, drugi korisnik validira podatke s pošiljateljevim javnim ključem. Time se lako može provjeriti jesu li podaci identični onima koje je korisnik poslao. [3]

2.5 Vrste blockchaina

Blockchain se može podijeliti na dva načina: na temelju centralizacije i privatnosti.

U **Decentraliziranom** blockchainu ne postoji centralni autoritet koji upravlja mrežom, što znači da korisnik ima potpunu kontrolu nad svojim podacima. Bilo tko može biti dio blockchaina i slati transakcije. Zbog toga postoje mehanizmi koji osiguravaju točnost transakcija, koji se zovu konsenzus mehanizmi.

Centralizirani blockchain se sastoji od korisnika čiji je identitet poznat. Dakle, sustav je valjan jer samo vjerodostojni sudionici mogu dodavati transakcije u blockchain. Centralizirani blockchaini se koriste u jako reguliranim industrijama poput financijske industrije, zbog smanjenja rizika. Iako decentralizirani i centralizirani blockchaini imaju svoje rizike, centralizirani blockchain je prikladniji u ovom slučaju

zbog znanih korisnika i vođenja revizije.

Blockchaini se mogu podijeliti i na temelju "otvorenosti" na: permissionless (javne) i permissioned (privatne).

Permissionless (javni) blockchain omogućuje slobodno pridruživanje i izlazak iz mreže, dokle god se čvor pridržava zadanih pravila sudjelovanja. Svaki član doprinosi održavanju mreže. Prednost mu je što nije potrebna kontrola pristupa niti zaštita protiv zlonamjernih čvorova. Permissionless blockchaini su transparentni. Dopuštaju korisnicima pristup svim informacijama osim privatnim ključevima, što znači da korisnici imaju pristup svim transakcijama na mreži. Također su anonimni i decentralizirani. Većina permissionless blockchaina ne traže nikakve osobne informacije kod kreiranja čvora. Pošto ne sadrže centralni (glavni) čvor, niti jedan samostalan čvor na mreži ne može promijeniti mrežne protokole, ugasiti mrežu ili uređivati transakcije.

Permissioned (privatni) blockchaini traži dozvolu kako bi se postao dio blockchain mreže. Centralni čvor ili korisnici odlučuju tko može pristupiti mreži, ali je njihov identitet javan. Upravljanje permissioned blockchainom ostavljeno je obično podružnicama jednog entiteta ili grupi entiteta, pa je zbog boljeg upravljanja, veće sigurnosti i povjerenja omogućena implementacija efikasnijih konsenzusnih mehanizama koje nije moguće implementirati na većim mrežama.

Na tablici ispod možemo vidjeti glavne razlike ove dvije vrste blockchaina.

	Permissionless blockchain	**Permissioned blockchain**
Kontrola	Javno	Centralni autoritet / skupina korisnika
Sudjelovanje	Slobodno	Samo ovlašteni
Identitet čvora	Sakriven	Javan
Transparentnost	Da	Može , ali ne mora biti
Veličina mreže	Velika	Mala
Konekcija na mreži	Mala	Velika
Primjer aplikacija	Kriptovalute, decentralizirane aplikacije, pametni ugovori	FinTech, poslovni ugovori

Tablica 2.1 Usporedba permissionless i permissioned vrste blockchaina

Poglavlje 3

Konsenzus mehanizmi

Konsenzus je proces sporazuma, gdje svi čvorovi zajedno odlučuju kako se on postigne da bi se dodao novi blok u postojeći blockchain. Konsenzus mehanizam je jezgra bilo kojeg distribuiranog blockchaina. Blockchain je onoliko pouzdan i siguran koliko je i njegov konsenzus mehanizam. Postoji mnogo vrsta konsenzus mehanizama pomoću kojih čvorovi mogu ostvariti konsenzus da bi dodali novi blok. Oni imaju nekoliko zadaća:

- **Predlaganje blokova** - generiranje blokova i popunjavanje bloka s potrebnim informacijama (timestamp, hash prethodnog bloka, generiranje hasha za trenutni blok , ...)
- **Širenje informacija**, odnosno širenje blokova i transakcija do svih čvorova mreže
- **Validacija blokova** - pregled točnosti informacija koje se nalaze u bloku te njegovih transakcija
- **Finalizacija bloka** - uspoređivanje rezultata s drugim čvorovima kako bi se odredilo stvarno stanje validiranih blokova
- **Dijeljenje nagrada**, odnosno poticanje sudjelovanja u mreži tokenima ili sličnim nagradama

3.1 Vrste konsenzus mehanizama

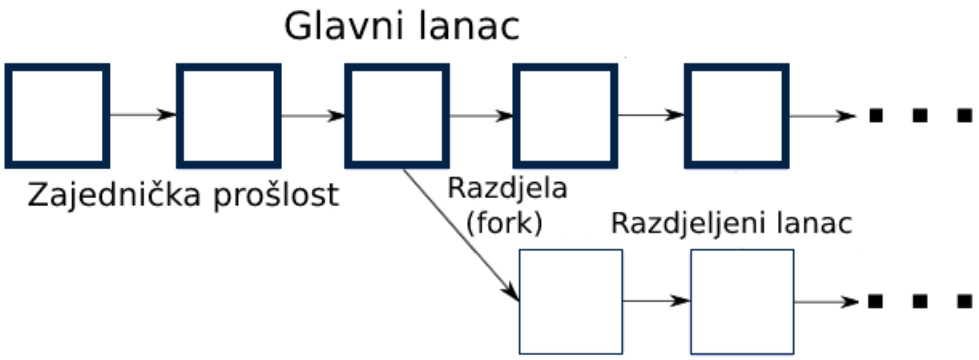
Konsenzus mehanizme možemo podijeliti na dvije vrste: **proof based** (utemeljeni na dokazu) i **voting based** (utemeljeni na glasanju). U sljedećim poglavljima ćemo objasniti najpopularnije konsenzuse.

3.1.1 Proof of Work

Proof of Work ili skraćeno PoW je prvi i najpopularniji mehanizam koji se trenutno koristi u blockchain sustavima. Koriste ga mnoge kriptovalute poput Bitcoina i do nedavno Ethereuma. Rješava problem vjerovanja čvorovima o kojima nisu poznate nikakve informacije, a to se ostvaruje time što je nepoznati čvor potrošio određenu količinu energije za određenu svrhu, stoga je veoma malo vjerojatno da bi zlonamjerni čvor potrošio veliku količinu energije da bi izvršio napad.

Postoji više načina za implementaciju Proof of Worka. U Bitcoinu se koristi hash funkcija, koja je jednosmjerna i deterministička funkcija koja uzima ulaz bilo koje duljine i generira string čiji rezultat izgleda nasumičan i nimalo sličan ulazu. Kako bi se validirao blok, rudar (*eng. miner*) mora kombinirati u jedan string: hash zadnjeg rudarenog bloka, blok transakcija kojeg želi validirati, svoj javni ključ i nonce. Zatim taj string uvrštava u hash funkciju i generira hash. Ukoliko blok počinje s n bitova nula, on se smatra točnim i blok se može dodati lancu, gdje n odredi sam blockchain na temelju raspoložive računalne snage koja se koristi za validiranje. Za razliku od drugih konsenzus mehanizama, implementacija PoW-a je dosta jednostavna.

PoW mehanizam se koristi za rješavanje problema razdjeljivanja blockchaina (*eng. forking*). Razdjeljivanje blockchaina je situacija kada se na istoj visini lanca nađu 2 potencijalna bloka. PoW to rješava eliminirajući kraći lanac, što znači da lanac s više računalnih resursa će pobijediti.



Slika 3.1 Pojednostavljeni grafički prikaz razdjeljivanja blockchaina (forking) [4]

Ukoliko zlonamjerni čvor želi poslati lažnu transakciju, prvo će morati posjedovati više računalnih resursa nego dobronamjerni čvorovi, pošto lanac s lažnom transakcijom ne bi funkcionirao na dobronamjernom čvoru.

Uz sve pozitivne strane PoW-a, postoji i nekoliko nedostataka. Najveći problem je upravo energija potrebna da bi se validirao sam lanac. Osim što je potrebno snažno računalo opremljeno naprednim komponentama, za validaciju PoW-a je potrebna velika količina energije. Na primjer, za validaciju jedne Bitcoin transakcije, mreža ukupno potroši 707 kWh električne energije. Također, što je više transakcija i blokova na Proof of Work lancu, količina energija koju je potrebno potrošiti je veća.

Svi blockchain sistemi koji koriste PoW su podložni 51% napadima. Ukoliko napadači posjeduju više od polovice resursa na lancu, mogu sami birati koji će blok i transakciju prihvatiti, a koji ne. Kod velikih blockchaina poput Bitcoina je to gotovo nemoguće, no postoje mnogi manji lanci koji s ne toliko mnogo resursa mogu biti uspješno napadnuti. [3]

3.1.2 Proof of Stake

Proof of Stake ili PoS potječe iz Bitcoin zajednice kao energetski učinkovita alternativa PoW rudarenju. Temelji se na ulogu (*eng. stake*) svakog korisnika, koji je najčešće količina valute unutar mreže. Ulogom korisnici dobivaju pravo glasa u sljedećem stanju blockchaina. Za razliku od PoW-a, čija je jedina vrsta korisnika rudar, u PoS ih postoji nekoliko, poput predlagača i validatora. Predlagači predlažu nove blokove, a validatori provjeravaju ispravnost istih.
Postoji mnogo načina za odabir predlagača. Na primjer, nasumičan odabir, davanje prednosti čvorovima koji duže posjeduju ulog, kombinacijom hash vrijednosti i količinom uloga, i tako dalje. Nakon što je korisnik odabran kao predlagač bloka, njegov se ulog zaključava i on potvrđuje sve transakcije i objavljuje blok. Njegov ulog ostaje zaključan sve dok se svi validatori mreže ne slože da je predloženi blok ispravan. Ukoliko je blok ispravan, predlagač i validatori koji su točno validirali blok dobivaju nagradu za ispravnu validaciju koja proizlazi iz transakcijskog troška mreže. U suprotnom predlagač i validatori koji su pogrešno validirali blok gube svoj ulog i algoritam ih označava malicioznim, što može slijediti i izbacivanju iz mreže.

Korištenjem uloga, PoS zaobilazi korištenje energetski zahtjevnih matematičkih zadataka i time rješava problem energetske učinkovitosti PoW-a, a pritom zadržava visoku razinu sigurnosti. Time se i smanjuje rizik napada "51%", gdje bi izvođenje takvog napada bilo iznimno skupo. Zbog manjeg broja validatora, povećava se brzina slanja transakcija, što omogućava širu primjenu blockchaina u financijskoj industriji.

S obzirom da je i validator nagrađen ako ispravno validira novi blok, može se dogoditi da validator glasa i potvrđuje nekoliko različitih blokova na istoj visini - bez obzira na njihovu točnost, kako bi povećao šansu za dobitkom nagrade, što predstavlja rizik integritetu mreže. Takva situacija je moguća jer, za razliku od PoW-a, u PoS mehanizmu validatori ne ulažu nikakav ulog tokom potvrđivanja blokova. Ukoliko validator potvrdi nekoliko blokova na istoj visini, može doći do razdjeljivanja blockchaina *(eng. multiple forks)*. Ovakav problem se naziva "Nothing at Stake" problem. Recentniji modeli PoS-a rješavaju ovaj problem kažnjavanjem validatora

koji potvrde blokove koji se naposljetku ne pridruže mreži.

Kao što je prije spomenuto, korisnici s više uloga imaju veću šansu biti odabrani kao predlagači novih blokova, što bogatijim korisnicima daje prednost. Sukladno tome, bogatiji korisnici imaju i veću šansu za nagradom, gdje onda bogatiji korisnici postaju još bogatiji. Posljedica toga je manje novih korisnika zbog veće količine početnog uloga, a i veća vjerojatnost izlaska korisnika iz mreže koji će mnogo rjeđe dobivati nagrade. Izlasci manje imućnih korisnika dovodi do veće centralizacije mreže. Centralizacija se može pojaviti i formiranjem skupina validatora.

Proof of Stake može biti i žrtva "Long range" napada, koji predstavlja slučaj kada napadač pokušava izmijeniti već zapisane blokove na blockchainu sve do početnog bloka. Napadač počinje graditi fork od početnog bloka, gdje fork sadrži potpuno drugačiju povijest od glavnog lanca. Nakon što razdijeljeni lanac postane duži nego glavni lanac, napad je uspješan. Jedan od rješenja za ovaj problem je uvođenje kontrolnih točaka *(eng. checkpoint)*, na kojima postoji kvota koliko posljednjih blokova može biti izmijenjeno.

3.1.3 Delegated Proof Of Stake

Delegated Proof of Stake je popularna verzija Proof of Stakea, gdje korisnici mreže glasaju i biraju delegate koji validiraju sljedeći blok. Delegati se isto nazivaju svjedocima ili proizvođačima blokova. Čvoru je omogućeno glasanje delegata kroz udruživanje tokena u tzv. *staking pool* i njega povezati s odabranim delegatom. Tokeni se fizički ne prebacuju u drugi novčanik, ali se koriste kao ulog , odnosno stake.

Za svaki novi blok je određen broj delegata , obično između 20 i 100, tako da delegati jednog bloka ne moraju nužno biti delegati sljedećeg. Izabrani delegati primaju naknade za transakcije od validiranog bloka, te se ta nagrada dijeli s korisnicima koji su glasali za tog delegata. Što čvor uloži više , to će njegova nagrada biti proporcionalno veća. Ukoliko delegat propusti svoj red za validiranjem, drugi delegati će preuzeti njegov blok i taj korisnik će izgubiti status delegata.

Za razliku od drugih konsenzusa, DPoS koristi demokraciju za biranje validatora, što dovodi do raznolikije grupe korisnika koji sudjeluju u procesu, pošto se konsenzus bazira na reputaciji, a ne na uloženim tokenima. DPoS je pokazao mnogo potencijala za poboljšanje efikasnosti, brzine transakcija i propusnosti blockchain protokola. [5]

3.2 Usporedba Proof of Work-a i Proof of Stake-a

U Proof of Work-u novi blok se stvara na temelju informacija prethodnog bloka, gdje pritom rudari računaju rješenje zahtjevnog matematičkog problema. Za isplativo rudarenje nekih od popularnih blockchaina koji koriste PoW potrebno je uložiti velika financijska sredstva za specijalnu računalnu opremu, čime se smanjuje dolazak novih rudara. Uz to, za stvaranje novog bloka je potrebna velika količina električne energije. Po nekim procjenama Bitcoin koristi 0.55% svjetske proizvodnje električne energije.
S druge strane, glavna prednost Proof of Stakea nad PoW-om je upravo mala potrošnja električne energije. Umjesto ulaganja u računalnu opremu i električnu energiju, korisnici ulažu u mrežne tokene kako bi povećali šansu za validacijom blokova. To znači da bogatiji korisnici mogu osigurati veću šansu za osvajanjem blokova, što također odvraća nove korisnike.
Za razliku od PoS-a , PoW može postati žrtva "51%" napada, a on se manifestira kada zlonamjerni čvorovi sadrže većinu računalnih resursa. Napad čiji je također cilj izmijeniti glavni lanac lažnim podacima je "Long range" napad, koji se pojavljuje isključivo na PoS-u. Pojavljuje se kada se glavni čvor zamijeni forkom koji je potpuno različit od glavnog čvora.

Proof of Work nema nikakve zahtjeve za sudjelovanje u validaciji bloka. Sudionici također mogu biti podijeljeni u razne grupe zavisno o njihovim karakteristikama, što pomaže decentralizaciji. Decentralizacija PoS-a se s vremenom smanjuje, zbog toga što su glavni faktori za generiranje bloka broj tokena i starost tokena.

Proof of Stake je skalabilniji od PoW-a i transakcije su na njemu validirane puno brže. Skalabilnost znači da sustav validira više transakcija po sekundi (*eng. TPS*).

PoS postiže skalabilnost tako što uspostavlja konsenzus prije nego što su blokovi generirani, što dopušta obrađivanje tisuće zahtjeva po sekundi s manje od milisekunde latencije.

Kod razdjele blockchaina, odnosno pojave više mogućih blokova na istoj visini, PoW odabire duži lanac (objektivno), dok u PoS validatori sami odlučuju koji će lanac odabrati (subjektivno).

Proof of Stake može postati žrtva napada "Nothing at Stake", koji je objašnjen u poglavlju 3.2 . Proof of Work ne može biti žrtva tog napada jer od svojih validatora traži rješavanje energetski zahtjevnih zadataka, dok u PoS-u to nije slučaj.

Uzevši sve u obzir, PoS naizgled ima veću prednost pred PoW-om, no to ne mora biti nužno tako. Najveća mana PoS-a je to što još nije dovoljno testiran, što znači da i u budućnosti može doći do novih vrsta napada. PoW je sigurniji i provjereniji nego PoS, no očito je zalaganje zajednice za razvijanjem energetski efikasnijih konsenzusa.

	Proof of Work	**Proof of Stake**
Rudarenje i validiranje blokova	Količina računalnih resursa određuje vjerojatnost rudarenja bloka	Količina uloga (stake) ili broj tokena određuje vjerojatnost validacije novog bloka
Distribucija nagrade	Kompetitivno - tko prvi izrudari blok, dobiva nagradu	Validator ne dobiva nagradu za uspješnu validaciju, nego dobiva mrežnu pristojbu
Kompetitivnost	Validatori se moraju natjecati rješavanjem kompleksnih matematičkih zadataka	Algoritam odlučuje pobjednika zavisno o njihovom stakeu
Specijalizirana oprema	Specifični integrirani krugovi i grafičke kartice	Nepotrebna
Dodavanje malicioznog bloka	Maliciozni čvorovi moraju posjedovati 51% računalnih resursa	Maliciozni čvorovi moraju posjedovati 51% mrežnih tokena
Energetska efikasnost	Troše više energije	Troše manje energije
Pouzdanost	Više pouzdani	Manje pouzdani
Sigurnost	Što je hash veći, to je mreža sigurnija	Ulaganje zaključava imovinu korisnika kako bi se osigurala mreža
Forking	Objektivno	Subjektivno

Tablica 3.1 Usporedba Proof of Work i Proof of Stake konsenzus mehanizma

Poglavlje 4

Tendermint

4.1 O Tendermintu

Tendermint ili Tendermint Core je blockchain platforma koja pruža usluge servera, baze podataka i podržane knjižnice za blockchain aplikacije koje su napisane u bilo kojem programskom jeziku. Slično kao što web server pruža web aplikacije, tako Tendermint pruža blockchain aplikacije.

Tendermint se sastoji od dvije glavne komponente: blockchain konsenzus i sučelje za aplikacije. Konsenzus mehanizam, koji se zove **Tendermint Core**, osigurava da se iste transakcije nalaze u istom redoslijedu na svim čvorovima. Aplikacijsko sučelje, koje se zove **Application BlockChain Interface (ABCI)** omogućuje obradu transakcija u bilo kojem programskom jeziku. Za razliku od drugih blockchain i konsenzus rješenja, koji uglavnom dolaze s ugrađenim konačnim automatima (*eng. state machines*), programeri mogu koristiti Tendermint za replikaciju BFT (Byzantine-fault tolerant) konačnog automata aplikacija napisanih u bilo kojem programskom jeziku i programskom okruženju koje im odgovara. Tendermint je lak za korištenje, jednostavan za razumijevanje, pruža visoke performanse i koristan za široki spektar distribuiranih aplikacija. [6]

4.2 Application BlockChain Interface (ABCI)

4.2.1 Motivacija

Application BlockChain Interface omogućuje BFT [1] replikaciju aplikacija koje su napisane u bilo kojem programskom jeziku. Do Tenderminta, svi blockchaini (poput Bitcoina) su imali dizajn monolita. Što znači da je svaki blockchain bio jedan program koji rješava sve dijelove decentraliziranog ledger-a; što uključuje: P2P vezu, emitiranje transakcija iz "mempoola" [2], konsenzus na najnovijem bloku, stanje na računu korisnika, dozvole za korisnike i tako dalje. Korištenje dizajna monolita je uglavnom loša praksa u računarstvu. U dizajnu monolita teško je ponovno koristiti komponente bez kompleksnog održavanja koda i ograničava razvoj isključivo na programski jezik u kojem je napisan.

4.2.2 ABCI arhitektura

ABCI se sastoji od 3 primarne vrste poruka koje se isporučuju iz Tendermit Corea u aplikaciju. Aplikacija odgovara s odgovarajućim response porukama.

- **DeliverTx** poruka je glavni dio aplikacije. Svaka transakcija u blockchainu je dostavljena preko ove poruke. Aplikacija validira svaku transakciju koja dolazi u DeliverTx s trenutnim stanjem, protokolom aplikacije i kriptografskim vjerodajnicama transakcije. Validirana transakcija mora ažurirati trenutno stanje aplikacije vezanjem vrijednosti u Key Value Store [3].
- **CheckTx** poruka je slična DeliverTx, ali služi samo validiranju transakcija. Tendermint Core prvo provjerava validnost transakcije s CheckTx porukom, a potom prosljeđuje tu poruku ostalim čvorovima. Na primjer , aplikacija može provjeravati inkrementirajući redni broj u transakciji i vratiti error ukoliko je redni broj zastario.

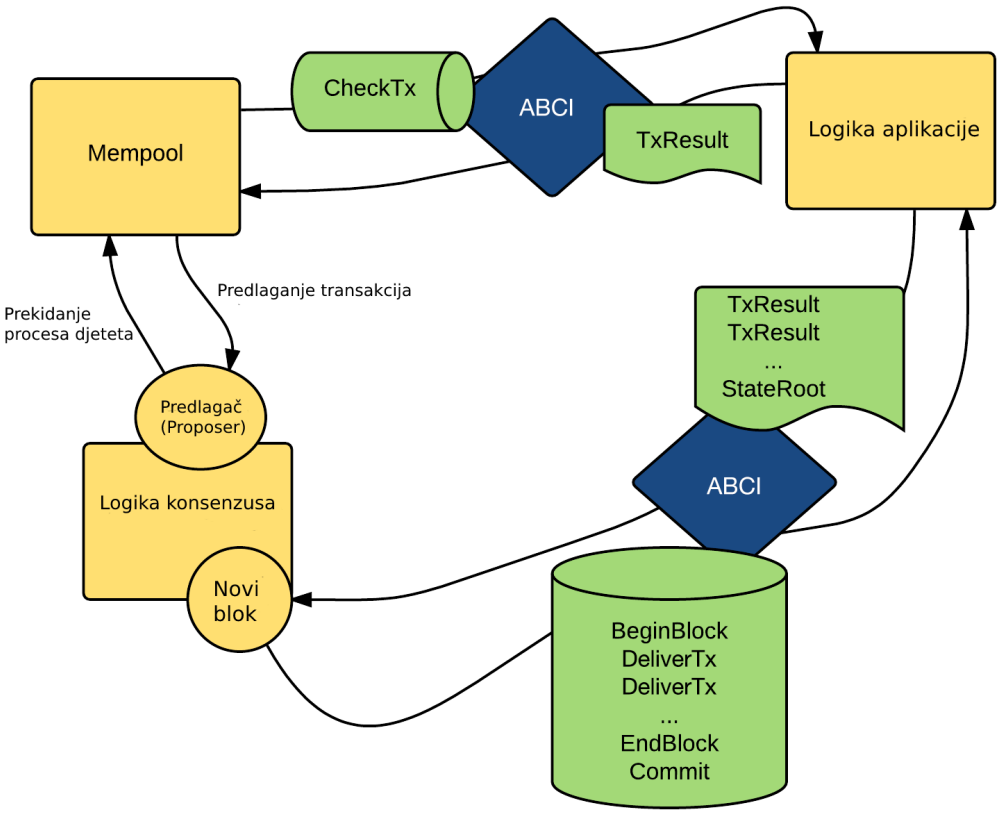
[1] Byzantine Fault Tolerant - svojstvo sustava da nastavi funkcionirati iako se neki njegovi čvorovi ponašaju zlonamjerno

[2] Mempool - skupina nepotvrđenih transakcija

[3] Key Value Store - tip nerelacijske baze podataka koja za svaki ključ sprema određenu vrijednost

- **Commit** poruka se koristi za izračunavanje kriptografske obveze (*eng. commitment*) na trenutno stanje aplikacije, koje će se nalaziti u zaglavlju sljedećeg bloka. Nedosljednosti u ažuriranju stanja sada će se pojaviti kao blockchain forkovi koji hvataju cijelu klasu programskih pogrešaka. Ovo također pojednostavljuje razvoj sigurnih klijenata [4], jer se Merkle-hash dokazi mogu verificirati provjerom u odnosu na blok hash i potpisom kvoruma.

U aplikaciji može postojati nekoliko ABCI veza. Tendermint Core se povezuje s ABCI tri puta; za validaciju transakcija kod emitiranja u mempool, za konsenzus mehanizam da bi mogao predlagati nove blokove i za slanje upita o stanju aplikacije. Sam tok poruka kroz ABCI je prikazan na dijagramu:



Slika 4.1 Dijagram komunikacije ABCI-a s aplikacijom i konsenzusom

[4]Klijent - oznaka je za bilo koji računalni program koji služi za prihvat i prikaz podataka

4.3 Tendermint Core

Tendermint Core je (uglavnom) asinkroni BFT konsenzus protokol. Sudionici protokola se zovu validatori. Validatori naizmjence predlažu blokove transakcija i međusobno ih glasuju. Blokovi se pridružuju lancu s jednim blokom na svakoj visini. Blok može biti neuspješno pridružen lancu te ga u tom slučaju protokol prebacuje u sljedeću rundu, a za to vrijeme drugi validator predlaže blok na trenutnoj visini. Postoje dvije faze kako bi se blok uspješno pridružio lancu: pre-vote (prije glasanja) i pre-commit (prije pridruživanja). Blok se pridružuje lancu kad 2/3 validatora uđu u pre-commit fazu za isti blok u istoj rundi. Svakom pre-commitu mora prethoditi *polka* u istoj rundi, odnosno 2/3 validatora moraju ući u pre-vote stanje za isti blok.

Ukoliko validator ne uspije pridružiti (commit) blok, ostali validatori mogu odlučiti da se taj validator preskoči. Validator malen dio vremena čeka kompletan predloženi blok prije nego što se glasanje prebaci na sljedeću rundu. Upravo taj timeout znači da Tendermint nije asinkroni , već sinkroni protokol. Ostatak protokola je asinkron i validatori napreduju samo nakon što se veza uspostavila s barem 2/3 validatora.

Kako ne bi došlo do situacije da se na istoj visini nađu više od jednog bloka, Tendermint ima pravilo zaključavanja (*eng. locking*), koje modulira putanje koje se mogu pratiti u dijagramu toka. Nakon što validator dođe u pre-commit stanje za jedan blok, on je zaključan na isključivo taj blok. Zatim mora napraviti pre-vote na bloku na kojem je zaključan te se može otključati ukoliko se desila polka za taj blok u kasnijoj rundi. [7]

4.4 Odabir tehnologije

Upravo zbog lakoće korištenja, dobre dokumentacije i programerskih zajednica na društvenim mrežama, za našu aplikaciju smo odlučili koristiti Tendermint. To nam je omogućilo fleksibilnost kod odabira programskog jezika za razvoj aplikacije i daje nam

mogućnost promjene programskog jezika u budućnosti, ukoliko će to biti potrebno. Programeri jednostavno mogu povezati protokol s korisnikom i Tendermint je idealan za razvijanje aplikacije koje se nalaze na vrhu blockchaina.

Poglavlje 5

Decentralizirana aplikacija za izdvajanje izračuna

5.1 Uvod

Održivi mehanizam za postizanje konsenzusa ili **SPOW** (*eng. Sustainable Proof of Work*), je projekt koji se razvija u sklopu Riteh Blockchain Teama na Tehničkom fakultetu u Rijeci. U ovom poglavlju se bavimo osnovnom temom ovog rada: razvojem Proof of Concepta decentralizirane aplikacije koja je bazirana na SPOW-u. Objasniti ćemo ideju i motivaciju za razvoj konsenzusa i objasniti funkcionalnosti same aplikacije.

5.2 SPOW i motivacija

SPOW je mehanizam za postizanje konsenzusa koji ima većinu karakteristika PoW mehanizma, ali ima znatno manje štetan utjecaj na okolinu. Glavna razlika s Proof of Workom je to što bi se uzalud utrošena energija na rad mehanizma zamijenila s korisnim poslom. Proof of Work se bazira na gađanju hash kodova, gdje samo prvi rudar koji pronađe točan hash dobiva nagradu, a električna energija ostalih rudara je uzalud potrošena. Takva okolina daje prednost korisnicima s većim resursima. Ideja SPOW-a je da se beskoristan posao u PoW, zamjeni s korisnim poslom. Tako

bi se natjecateljsko rudarenje zamijenilo kooperativnim rješavanjem zadataka, gdje bi se nagrada dijelila sa svim korisnicima koji su pronašli točno rješenje zadatka, proporcionalno njihovom ulogu. SPOW bi zapravo bio Proof of Stake konsenzus, gdje bi obavljeni posao označavao ulog, odnosno *stake*. Ovim navedenim radnjama bismo riješili problem energetske učinkovitosti i održivosti.

5.3 Decentralizirana aplikacija za izdvajanje izračuna

Decentralizirana aplikacija za izdvajanje izračuna ili DAPP je aplikacija koja služi kao Proof of Concept SPOW konsenzusu, testira osnovne funkcionalnosti, ispituje ograničenja rješenja i ostvaruje uvid u moguće probleme, te predlaže poboljšanja. DAPP pruža sljedeće funkcionalnosti:

- registracija korisnika
- prijavljivanje resursa za rješavanje
- dijeljenje i raspodjela zadataka
- praćenje rješavanja
- provjeravanje rješenja
- rješavanje rubnih situacija
- mehanizam za nagrađivanje sudionika

5.4 Arhitektura

DAPP je napisana koristeći prije spomenute Tendermint Core i ABCI. Napisana je u programskom jeziku Go. Sam projekt se sastoji od dvije glavne datoteke: *main.go* i *app.go*. Kako bi projekt funkcionirao, prvo je potrebno inicijalizirati nekoliko Tendermint čvorova koji će služiti kao validatori i jedan čvor koji će služiti kao tzv. *seed* čvor koji povezuje sve prijavljene validatore.

5.4.1 Čvor

Datoteke čvora su podijeljene u 3 direktorija: *badger, config i data.* U *badger* i *data* direktorijima su informacije vezane za bazu podataka i podatke.
U datoteci *priv_validator_state.json* u *data* direktoriju se nalaze informacije o trenutnom stanju čvora. Ovdje možemo pronaći nekoliko podataka: najnovija visina u kojoj je validator prisustvovao konsenzusu, u kojoj se rundi i koraku nalazi i posljednji digitalni potpis čvora. Ti podaci omogućuju čvoru da nastavi validaciju na posljednjoj visini ukoliko je iz nekog razloga izvršenje čvora zaustavljeno.
Config direktorij sadrži potrebnu konfiguraciju čvorova. U *config.toml* datoteci se nalaze postavke timeouta, adrese na kojima će se RPC server za pojedini čvor nalaziti, maksimalni broj povezanih čvorova i tako dalje. Kako bi više čvorova bilo povezano, *genesis.json* datoteka mora biti jednaka na svim čvorovima. U njoj se nalaze općenite informacije i postavke za cijeli lanac i popis svih prijavljenih validatora. U *config* direktoriju se nalaze i datoteke koje sadrže javni i privatni ključ čvora.

5.4.2 main.go

Svrha main.go datoteke je pokrenuti čvor, odnosno aplikaciju, sve potrebne komponente i prikupiti sve potrebne informacije od korisnika. Prilikom pokretanja aplikacije potrebno je specifirati koji čvor pokrećemo, te se od korisnika se traži odabir vrste čvora kojeg želi pokrenuti. Trenutno postoje funkcionalnosti samo za "validator" tip. Nakon toga se od korisnika traži količina resursa u khash-per-block, odnosno broj izračunatih tisuća hasheva po bloku. Zatim se pripremaju svi potrebni argumenti kako bi se pokrenuo key value store: čitanje config datoteke, čitanje genesis datoteke, pokretanje baze podataka i inicijalizacija loggera. Key value store povezujemo s ABCI i pokrećemo sam čvor.

```
db, err := badger.Open(badger.DefaultOptions(dbPath))
if err != nil {
    log.Fatalf("Opening database: #{err}")
}
defer func() {
    if err := db.Close(); err != nil {
        log.Fatalf("Closing database: #{err}")
    }
}()
app := NewKVStoreApplication(db, homeDir, config, resource)
acc := abciclient.NewLocalCreator(app)

logger := tmlog.MustNewDefaultLogger(tmlog.LogFormatPlain, tmlog.LogLevelInfo, trace: false)
node, err := nm.New(config, logger, acc, gf)
if err != nil {
    log.Fatalf("Creating node: #{err}")
}

err = node.Start()
```

Slika 5.1 Kod pokretanja čvora iz main.go datoteke

5.4.3 app.go

Tendermint Core komunicira s aplikacijom preko Application BlockChain Interfacea. App.go posjeduje sve poruke (*messages*) kao i ABCI, s odgovarajućim requestom i responseom. App.go se sastoji od sljedećih funkcija:

- **CheckTx** - kada se transakcija doda u Tendermint Core, aplikacija provjerava njen format i potpise kako bi ju validirala. Ukoliko transakcija nema format *bytes=bytes*, funkcija vraća vrijednost 1. Ukoliko već postoji vrijednost za taj ključ, funkcija vraća vrijednost 2, a za ostale situacije vraća vrijednost 0, što označava validnu transakciju. Ova funkcija se poziva samo na čvoru koji je poslao zadanu transakciju.
- **BeginBlock** - kreira se varijabla koja će spremati transakcije koje su kreirane u bloku. Informacije iz BeginBlock requesta se mogu koristiti kako bi se pokrenuo neki kod na početku bloka. U ovoj funkciji šalje se i hash trenutnog bloka i

zaglavlje bloka aplikaciji, prije nego što se šalju bilo koje transakcije.

- **DeliverTx** - Tendermint šalje DeliverTx requeste asinkrono, ali po redu. TCP protokol osigurava da su requesti primljeni redoslijedom kojim su i poslani. DeliverTx vraća abci.Result, koji sadrži kod, podatke i log. Ukoliko je kod različit od nule, to označava da će transakcija biti odbijena. U ovom projektu DeliverTx funkciju koristimo za validaciju transakcije od strane validatora, te za slanje evenata za lakše dohvaćanje podataka.
- **EndBlock** - funkcija koja završava blok. Može se koristiti za dodavanje novog validatora ili ažuriranje postojećeg.
- **Commit** - nakon što je završena obrada bloka, Tendermint šalje Commit request i blokove koji čekaju response. Mempool može nastaviti raditi istovremeno, no on je zaključan dok konsenzus ne primi Commit response. To znači da aplikacija ne smije ostvarivati nikakvu komunikaciju s mempoolom za vrijeme Commit funkcije, inače će se dogoditi zastoj.

5.5 Funkcionalnosti

5.5.1 Registracija korisnika

Pri pokretanju projekta, korisnik odabire količinu resursa koju želi koristiti tokom izvođenja aplikacije. Potom unutar BeginBlock funkcije dolazimo do sljedećeg dijela koda:

```
conn := checkIfRpcAddressIsReachable(app)
if conn != nil && registered == 0 {
    if findRegisterTransaction(app) == false {
        sendRegisterTransaction(app)
    } else {
        registered = 1
    }
}
```

Slika 5.2 Registracija korisnika

U prvoj liniji provjeravamo je li RPC server dostupan. To je potrebno jer ukoliko pokrećemo već prije pokrenuti čvor, on će imati već neke podatke u bazi podataka. Dok čvor validira postojeće transakcije, nije u stanju primati nove transakcije. Nakon što je RPC server dostupan, provjeravamo je li korisnik registriran. Svaka transakcija koju šaljemo ima svoj ključ i svoju vrijednost. Ključ je standardiziran i za svaku transakciju se koristi ista struktura za generiranje ključa:

```
type Key struct {
    Type int    `json:"type"`
    Node string `json:"node"`
}
```

Slika 5.3 Struktura ključa transakcije

Ključ se sastoji od: typea, odnosno vrste, i nodea, odnosno jedinstvene oznake čvora. Vrsta transakcije se označava konstantama koje su definirane u aplikaciji, a jedinstvena oznaka čvora se učitava iz njegove config datoteke. Vrijednost transakcije se sastoji od količine resursa koje je korisnik unio pri pokretanju aplikacije.

5.5.2 Izračun hasheva

Ukoliko je korisnik registriran, čvor će krenuti rješavati zadatak pri generiranju novog bloka. Kao što je prije spomenuto, trenutno aplikacija ima samo jednu vrstu zadatka,

a to je običan hashcash zadatak. Prvo konkatiniramo sljedeće tri vrijednosti: hash trenutnog bloka, jedinstvenu oznaku čvora i nonce. Nonce se inkrementira nakon svakog izračuna bloka. Dobiveni rezultat enkriptiramo SHA-256 algoritmom, te smo time dobili jedan izračun hasha. Ukoliko hash na svojih prvih 4 bita ima vrijednost 0, on se smatra točnim rješenjem zadatka. Težina zadatka, odnosno količina 0 na početku hasha je trenutno fiksna, no u budućnosti je plan tu vrijednost mijenjati zavisno o računalnim resursima čvora. Za vrijeme jednog bloka, čvor izračuna onoliko tisuća hasheva koliko smo specificirali pri pokretanju aplikacije. Na slici možemo vidjeti kako petlja izgleda u aplikaciji.

```
for i < 1000*app.resource {
    i++
    blockHash := hex.EncodeToString(req.GetHash())
    nodeKey, _ := getNodeKeyIDFromDirectory(homeDir)
    app.nonce++
    nonce := strconv.Itoa(int(app.nonce))
    hash := sha256.Sum256([]byte(blockHash + string(nodeKey) + nonce))
    hexString := fmt.Sprintf( format: "%x", hash)
    first4Chars := hexString[0:4]
    if first4Chars == "0000" {
        fmt.Println( a...: "************************************************************")
        fmt.Println( a...: "CORRECT HASH!")
        fmt.Println( a...: "************************************************************")
        sendAwardTransaction(blockHash, string(nodeKey), nonce, app)
    }
}
```

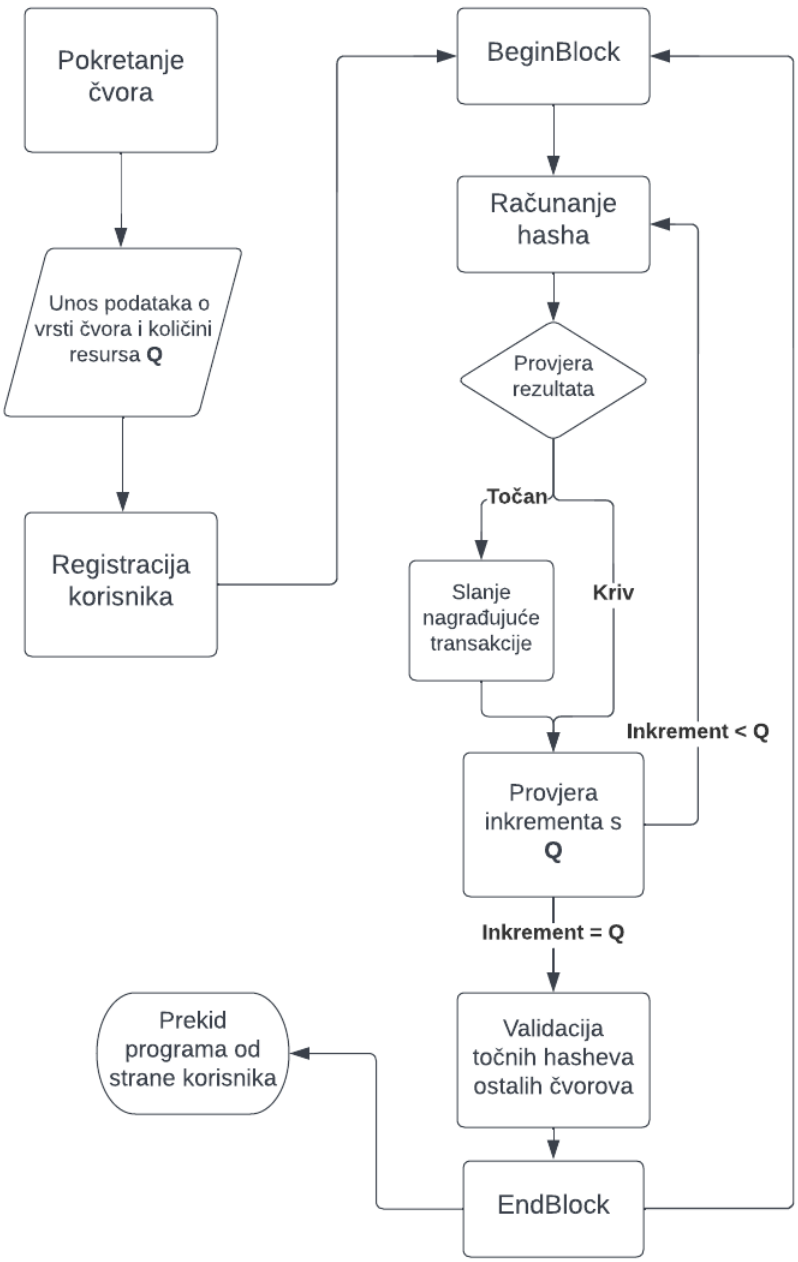
Slika 5.4 Petlja za izračun hasha

Nakon što čvor pronađe točan hash, on će slati transakciju o pronađenom hashu. Vrijednost transakcije sadržavaju: hash trenutnog bloka, nonce i količina resursa. Te vrijednosti su potrebne kod validacije transakcije i nagrade korisnika.

5.5.3 Validacija točnog hasha

Pronalaskom točnog hasha šalje se transakcija nagrade. Ona dolazi do DeliverTx funkcije svakog od čvorova i ukoliko u toj funkciji nemamo potrebnu validaciju, bilo koji čvor bi mogao falsificirati transakciju i dobivati nagradu bez utrošenog rada. Zbog toga po dolasku transakcije za nagradu u DeliverTx funkciju, validiramo transakciju na temelju podataka koje su poslane u vrijednosti transakcije. Događa se isti proces konkatiniranja i hashiranja kao i u prethodnom poglavlju. Tek ako je dobiveni rezultat zaista točan, onda se ta transakcija dodaje u mempool i dodaje se u bazu podataka čvora. Ovaj proces prolazi svaki validator blockchaina.

Zbog lakše provjere i pretrage transakcija, nakon uspješne valiadacije šalje se ABCI Event. ABCI posjeduje funkciju abci_query, koja dohvaća samo posljednje pridodanu transakciju s određenim ključem. Zbog određivanja ukupnog broja točno pronađenih hasheva, moramo pristupiti svih transakcijama koje imaju tip nagrađujuće transakcije za određeni čvor. U Eventu možemo dodati koje god informacije želimo kako bi lakše i preglednije mogli dohvaćati podatke. Zbog toga vrlo jednostavno možemo poslati upit s ključem nagradne transakcije kao argumentom i dobiti sve nagrađujuće transakcije za određeni čvor.



Slika 5.5 UML dijagram procesa u decentraliziranoj aplikaciji

5.6 Ideje za budućnost

U budućnosti je plan uvesti novi tip čvora, a to je predlagač *(eng. proposer)*. Svrha predlagača je predlaganje zadataka. Time bismo postigli raznolikost u vrstama zadataka, što bi solveru, odnosno rješavaču omogućilo da odabere zadatak koji najbolje odgovara njegovim resursima i ambicijama.

Umjesto fiksne težine, plan je uvesti varijabilnu težinu zadatka, koja bi se mijenjala ovisno o računalnim resursima čvora. Time bismo spriječili preopterećenje čvora i spriječili iznenadno zaustavljanje čvora.

Kako bi se privukli sudionici u blockchainu, potrebno je napraviti mehanizam nagrađivanja. Trenutno nagrada postoji samo kao zapis transakcije, koji nema neku praktičnu svrhu. Postoji mnogo načina na koji bi se mogli nagrađivati sudionici: razni tokeni, NFT i slično.

Poglavlje 6

Zaključak

U ovom radu je objašnjena blockchain tehnologija, njezina arhitektura i podijeljena je na vrste. Također su objašnjeni i konsenzus mehanizmi, podijeljeni su na vrste te su uspoređeni dva najpoznatija konsenzusa: Proof of Work i Proof of Stake. Zaključeno je da Proof of Work, najpopularniji konsenzus mehanizam, troši veliku količinu električne energije kako bi funkcionirao, zbog čega je predloženo rješenje u Sustainable Proof of Worku (SPOW). Objašnjen je koncept SPOW-a i kao Proof of Concept konsenzusa je razvijena decentralizirana aplikacija, s kojom dokazujemo principe SPOW-a. Za razvijanje decentralizirane aplikacije je korištena Tendermint blockchain platforma, koja pruža fleksibilnost kod odabira programskog jezika za razvoj aplikacije i jednostavna je za korištenje. Objasnili smo dijelove Tendermint blockchain platforme i nabrojali najbitnije funkcionalnosti.

Predstavljena je decentralizirana aplikacija za izdvajanje izračuna, u kojoj čvorovi zajedno rješavaju zadatke i tako je eliminirano uzaludno trošenje energije. Objašnjene su glavne funkcionalnosti aplikacije: registracija, zajedničko računanje hasheva i validacija točnih rješenja zadataka. Aplikacija uspješno implementira SPOW i energetski je efikasnija nego običan PoW, što znači da će svaki čvor dobiti nagradu za točno izračunate hasheve. Navedeni su zadaci i funkcionalnosti koji će se u budućnosti razvijati na aplikaciji kako bi ona bila bolje implementirana i efikasnija.

Bibliografija

[1] O'Reilly: "Chapter 7. The Blockchain", s Interneta, https://www.oreilly.com/library/view/mastering-bitcoin, 20.5.2021

[2] Blockchain, U *Wikipedia*, https://en.wikipedia.org/wiki/Blockchain, 22. srpnja 2022.

[3] Sudhani V, Divakar Y, Girish C: "Introduction of Formal Methods in Blockchain Consensus Mechanism and Its Associated Protocols", s Interneta,

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9801830, 21. lipnja 2022.

[4] Blocknet dokumentacija, slika, s Interneta, https://docs.blocknet.co/wallet/forking/

[5] S. Saad, R.Radzi, S. Othman: "Comparative Analysis of the Blockchain Consensus Algorithm Between Proof of Stake and Delegated Proof of Stake", https://ieeexplore.ieee.org/document/9617549, 6. listopad 2021.

[6] Tendermint Core: "What is Tendermint?", s Interneta, https://docs.tendermint.com/v0.34/introduction/what-is-tendermint.html

[7] Tedermint, s Interneta, https://github.com/tendermint/tendermint

Sažetak

Najpopularniji i najprovjereniji konsenzus mehanizam Proof of Work ima veliku manu energetske neefikasnosti. U ovom radu se predlaže alternativa u Sustainable Proof of Worku, gdje se uzaludan posao natjecateljskog rudarenja, mijenja kooperativnim rješavanjem zadataka. Predstavljena je decentralizirana aplikacija za izdvajanje izračuna koja implementira koncept Sustainable Proof of Worka i dokazuje njegove principe.

Ključne riječi **— Proof of Work, održivost, SPOW, decentralizirana aplikacija**

Abstract

The most popular and proven consensus mechanism Proof of Work has a large drawback of energy inefficiency. This paper proposes an alternative in Sustainable Proof of Work, where the futile work of competitive mining is changed with cooperational task solving. The paper introduces a decentralized application for outsourcing computational tasks that implements the concept of Sustainable Proof of Work and proves its principles.

Keywords **— Proof of Work, sustainability, SPOW, decentralised application**