Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/18BE20Juric

KLIJENTSKA APLIKACIJA ZA PRISTUP ETHEREUM PLATFORMI

CLIENT APPLICATION FOR THE ETHEREUM PLATFORM

Marko Jurić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu opisani su osnovni principi distribuiranih sistema, razvoj blokčejn tehnologija kroz tri faze, pojam Ethereum platforme, kao i arhitektura klijenata za pristup Ethereum mreži. Predstavljeno je rešenje u vidu implementacije promene na Ethereum klijentu kao posledice zahteva Ethereum zajednice za ažuriranje Ethereum platforme.*
Ključne reči: *distribuirani sistemi, blokčejn, pametni ugovori*
Abstract – *This work describes fundamental principles of distributed systems, evolution of blockchain technologies through three phases, Ethereum platform and client architecture for accessing Ethereum. Presented solution is an implementation of a change in Ethereum client which is caused by a request from Ethereum community to update Ethereum platform.*
Keywords: *distributed systems, blockchain, smart contracts*

1. UVOD

U savremenom svetu, finansijske institucije i dalje više ili manje funkcionišu kao centralizovani sistemi.

Centralizacija se odnosi na centralnu tačku kontrole koja se u slučaju finansija odnosi na bankarske institucije. Istorijski gledano, centralizacija u finansijama je bila poželjna kao odrzavač stabilnosti u globalnim finansijskim procesima za koju se smatralo da je sigurnija i stabilnija od ličnog upravljanja [9].

Pod tim podrazumevamo da pojedine korake u proceduri može da radi samo jedan autoritet ili institucija, što može da predstavlja veliki problem u vidu poverenja i performansi. Ukoliko želimo da pošaljemo novac sa jednog računa na drugi, u centralizovanom sistemu postoji nešto što je između, a to je u ovom slučaju banka. Bez posredstva banke kao treće strane nemoguće je izvršiti transakciju, a njeni korisnici su u obavezi da veruju da će sa njihovim sredstvima i podacima biti sve u redu.

Budući da banke kao centralne tačke ovakvih sistema čuvaju resurse i lične podatke svojih korisnika, postavlja se pitanje koliko je zapravo moguće verovati takvom sistemu. Takođe, jedan od problema sa stanovišta bezbednosti jeste što u centralizovanom sistemu imamo jedinstvenu tačku otkaza.

Decentralizovane finansije su oblik finansiranja koji se ne oslanja na centralne finansijske posrednike kao što su brokerske kuće, berze ili banke, već umesto toga koristi blokčejn tehnologiju. Blokčejn tehnologija kreće od pretpostavke da svi učesnici u sistemu imaju pravo da dobiju sve informacije i da učestvuju u svakom koraku donošenja odluka. To se radi upotrebom raznih konsensus algoritama. Baza podataka u blokčejn tehnologiji je jednostruko povezan lanac blokova u kojem se sadrže transakcije. Svaki blok sadrži vezu sa prethodnim blokom i na taj način su povezani u lanac. Povezivanje se izvršava pomoću kriptografije i zato je praktično nemoguće izmeniti već postojeći blok. Umesto korišćenja centralnog entiteta za upravljanje blokčejnom, ova tehnologija koristi *Peer to peer* mrežu u kojoj računari međusobno dele podatke direktno bez potrebe za centralnim serverom. Kada novi član sistema (čvor) pristupi mreži, dobija kopiju blokčejna tako da može da učestvuje u verifikaciji blokova. To što svako može da preuzme kompletnu istoriju svih transakcija i da kontroliše pravljenje novih blokova čini ovaj sistem istinski decentralizovanim. Takođe, u ovakvim sistemima vrlo je teško doći do otkaza, jer čak i da otkažu pojedini čvorovi, ostatak čvorova u mreži može da održava normalan rad sistema. Računi u banci, kao i drugi tajni centralizovani podaci su često meta napada i dešava se da bivaju hakovani. Ukoliko bi neko hteo da hakuje blokčejn, morao bi da hakuje čitav lanac blokova što je praktično nemoguće. Jedna od najpopularnijih platformi koja koristi blokčejn tehnologiju je *Ethereum*.

2. TEORIJSKE OSNOVE

Distribuirani sistem je skup autonomnih računara povezanih na mrežu posredstvom mrežnog i distributivnog posredničkog programa koji omogućava računarima da koordiniraju svoje aktivnosti i dele resurse sistema tako da korisnici sistem percipiraju kao jedinstven, integrisani računarski objekat. Komponente distribuiranog sistema komuniciraju i koordiniraju svoje akcije prenošenjem poruka. Međusobno komuniciranje se vrši radi postizanja zajedničkog cilja. Distribuirani sistemi imaju nekoliko ključnih karakteristika. Skalabilnost se ogleda u tome kako distribuirani sistem podnosi rast sa povećanjem broja korisnika sistema. Distribuirani sistem se uglavnom skalira dodavanjem više čvorova u mrežu. Još jedna karakteristika predstavlja nedostatak "globalnog sata". U distribuiranom sistemu postoji mnogo sistema i svaki sistem ima svoj sat. Svaki sat na svakom sistemu radi različitom brzinom što može da dovede do problema sa sinhronizacijom čvorova. U distribuiranom sistemskom softveru, hardveru ili mreži uvek se može dogoditi otkazivanje. Sistem mora biti projektovan na takav način da je dostupan čak i kada nešto otkaže. Transparentnost se ogleda u tome da krajnji ko-

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Goran Sladić, red. prof.

risnik vidi distribuirani sistem kao jednu računsku jedinicu, a ne kao njegove osnovne delove. Krajnji korisnik dakle nije upoznat od kojih čvorova se sistem sastoji, tako da nije ni svestan koji čvor mu obezbeđuje podatke [1].

Tehnologija distribuirane knjige (*Distributed Ledger Technology – DLT)* je digitalni sistem za evidentiranje transakcija različitih dobara u kojem se transakcije i njihovi detalji beleže na više mesta istovremeno. Za razliku od tradicionalnih baza podataka, distribuirane knjige nemaju centralno skladište podataka ili administrativnu funkciju. U distribuiranoj knjizi svaki čvor obrađuje i verifikuje sve podatke, zapisivajući na taj način svaku stavku i stvarajući konsenzus o njenoj verodostojnosti. Tako svaki čvor replicira i čuva identičnu kopiju glavne knjige i samostalno se ažurira. Distribuirana knjiga se može koristiti za evidentiranje statičkih podataka kao sto je registar ali i dinamičkih podataka kao sto su finansijske transakcije. *DLT* koristi kriptografiju za sigurno skladištenje podataka, kriptografskih potpisa i ključeva kako bi omogućio pristup samo ovlašćenim korisnicima. Tehnologija takođe stvara nepromenljivu bazu podataka, što znači da se informacije, nakon što se sačuvaju, ne mogu izbrisati i bilo koje ažuriranje se trajno beleži [7].

Važno je napomenuti da koncept distribuirane knjige nije nov. *DLT* ima svoju najraniju istoriju još u Rimskom carstvu koje je imalo takav bankarski sistem da je omogućavalo ljudima da učestvuju u transakcijama sa drugim regionima carstva. Dalja otkrića su došla u obliku papirnih čekova što je dovelo do poboljšanja u ažuriranju i evidentiranju transakcija. Ali upotreba *DLT-a* bila je sputana zbog dileme koja je postala poznata kao problem vizantijskih generala (*Byzantine generals problem*) koja govori o više generala koji vode svoje armije i strateški su pozicionirani izvan neprijateljske teritorije. Generali moraju komunicirati sa glasnicima kako bi sklopili zajednički sporazum oko započinjanja napada. Da bi napad bio uspešan, potrebno je da svi generali napadnu u isto vreme. Međutim, šta bi se dešavalo ako su neki generali ili glasnici izdajnici? Dilemu su opisali Lamport, Šostak i Pis u svom radu iz 1982.

Interesovanje za tehnologiju distribuirane knjige značajno je poraslo u deceniji nakon lansiranja bitkoina, kriptovalute pokrenute tehnologijom blokčejna koja je prva pokazala da ta tehnologija ne samo da funkcioniše, već da može da se povećava i ostane sigurna. Misteriozni pronalazač Bitkoina, pod pseudonimom Satoši Nakamoto, opisao je blokčejn u knjizi za Bitkoin. U najranijim fazama, blokčejn je postavio osnovnu premisu o zajedničkoj javnoj knjizi koja podržava mrežu kriptovaluta. Satošijeva ideja je bila da blokčejn sadrži blokove transakcija koji iznose 1MB. Bitkoin je obezbedio ubedljiv protokol konsenzusa, nazvan dokaz rada (*Proof of Work*), za rešavanje problema vizantijskih generala. Dokaz rada predstavlja algoritam osmišljen tako da ukoliko učesnik u mreži(čvor) želi da verifikuje novi blok, mora da reši kompleksan matematički zadatak. Rešavanjem ovog zadatka troši se određena količina električne energije što ima za cilj da smanji verovatnoću da čvor koji verifikuje bude maliciozan time što se ne isplati trošiti toliko resursa za hakovanje blokčejna koje je samo po sebi veoma teško. Proces verifikacije blokova se takođe popularno naziva rudarenje (*Mining*), jer čvor koji uspe da napravi novi blok biva nagrađen određenom količinom kriptovalute. U ovoj fazi kriptovalute su toliko bile u fokusu, da je bila praktično zanemarena činjenica da blokčejn tehnologija može da se koristi i za druge potrebe. Naravno, kako je vreme prolazilo, programeri su počeli da veruju da blokčejn može učiniti više od pukog dokumentovanja transakcija. Osnivači *Ethereuma* su imali ideju da bi imovina i ugovori o poverenju takođe mogli imati koristi od upravljanja blokčejnom. Na ovaj način, *Ethereum* predstavlja drugu generaciju blokčejn tehnologije. Glavna inovacija koju je *Ethereum* doneo bili su pametni ugovori (*Smart Contracts*). Obično se ugovorima u uobičajenom poslovnom svetu upravlja između dva odvojena entiteta. Međutim, pametni ugovori su oni koji u blokčejnu upravljaju samim sobom. Pokreće ih događaj poput prolaska datuma isteka ili postizanja određene ciljane cene. Kao odgovor, pametni ugovor upravlja sam, prilagođavajući se po potrebi, bez uticaja spoljnih subjekata [4].

Trenutno se svet i dalje nalazi u procesu odgonetavanja svih potencijala pametnih ugovora. Jedan od glavnih izazova sa kojima se blokčejn suočava jeste skaliranje. Dakle, dok su prve dve faze razvoja blokčejna bile izuzetne kada je reč o inovacijama, treća se suočava i sa rešavanjem postojećih problema. Problemi sa uskim grlom javljaju se jer previše ljudi pokušava da izvrši transakciju a premalo je prostora za to na blokčejnu. Blokčejn projekti treće generacije osmišljeni su tako da tehnologija automatski rešava probleme skaliranja ukoliko se pojave. Još jedno pitanje koje blokčejn projekti treće generacije rešavaju je interoperabilnost. Na isti način kao što ne možemo napuniti bateriju jednog mobilnog telefona koristeći punjač telefona drugog proizvođača, prve dve iteracije blokčejna (Bitkoin i Ethereum) ne mogu međusobno komunicirati. Iako ovo možda ne zvuči kao velika stvar, interoperabilnost je zapravo ključna za napredak industrije. Svet se oslanja na kolaboraciju i sisteme u kojima se informacije i podaci mogu razmenjivati na različitim platformama. Projekti poput *Cardano-a* i *Polkadot-a* uveli su funkcije interoperabilnosti u svoj blokčejn od početka, sto znači da mogu normalno sarađivati i sa drugim blokčejnovima. Sa poboljšanjima iz generacije u generaciju, nije preterano optimistično verovati da će u nekoj bližoj budućnosti doći do sveopšteg usvajanja blokčejn tehnologije.

3. ETHEREUM

Iz perspektive računarstva, *Ethereum* je deterministička, ali praktično neograničena mašina stanja, globalno dostupnog svima i virtuelne mašine koja primenjuje promene na to stanje. Iz praktičnije perspektive, *Ethereum* je globalno decentralizovana platforma otvorenog koda čija svrha nije samo da bude platna mreža u kojoj se koristi digitalna valuta *Ether*, već i mreža koja izvršava programe koji se zovu pametni ugovori. Kao i u bitkoin mreži, čvorovi koji verifikuju blokove (majneri) dobijaju nagradu. Svaka operacija koja je potrebna da bi se izvršila transakcija ili pametni ugovor zahteva plaćanje u vidu gasa. Gas predstavlja jedinicu u kojoj je izražena količina računarskog napora koji je potreban za izvršavanje nekeodređene operacije. Zato je bitno koliko se gasa troši pri izvršavanju pametnih ugovora, jer što su ugovori

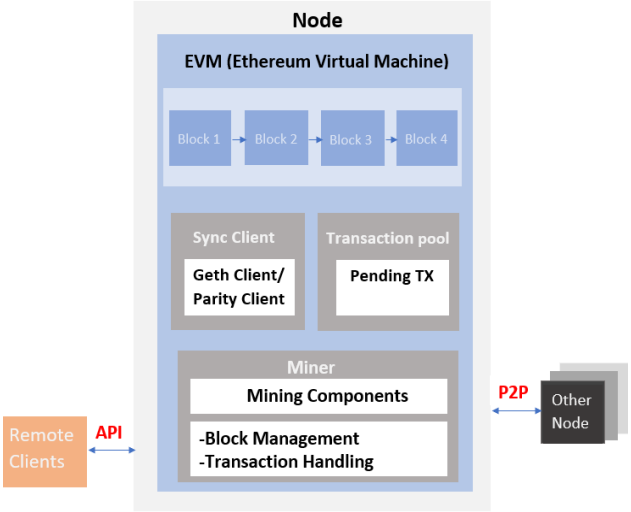
složeniji cena je veća. Naknade za gas se plaćaju u matičnoj valuti *Ethereuma* – eteru [8].

Ethereum klijent je softverska aplikacija koja implementira *Ethereum* specifikaciju i komunicira preko *Peer to Peer* mreže sa drugim *Ethereum* klijentima. Različiti klijenti međusobno komuniciraju ako su u skladu sa referentnom specifikacijom i standardizovanim komunikacionim protokolima. Iako su ovi različiti klijenti implementirani u različitim programskim jezicima, svi oni govore istim protokolom i slede ista pravila. *Ethereum* je definisan formalnom specifikacijom koja se naziva Žuta knjiga (*Yellow paper*).

Ethereumova žuta knjiga dokumentovana je na papiru i napisana je kao kombinacija Engleskog jezika i matematičke formalne specifikacije. Ona definiše standardno ponašanje *Ethereum* klijenta. Žuta knjiga se periodično ažurira ukoliko se zajednica usaglasi da su potrebne promene na *Ethereumu*. Kao rezultat jasnih formalnih specifikacija *Ethereuma*, postoji niz nezavisno razvijenih, ali ipak interoperabilnih, softverskih implementacija *Ethereum* klijenta. Od pomenutih različitih klijent implementacija u različitim programskim jezicima, dve najpoznatije su: *Geth* - napisan u *Go (Golang)* jeziku i *Parity* napisan u *Rust* jeziku [2].

Klijentski softver koji je pokrenut na *Ethereum* mreži naziva se čvor (*Node*). Postoje različiti tipovi čvorova koji različito konzumiraju podatke: *Light*, *Full* i *Archive*. *Full Node* (Potpun čvor): čuva ceo blokčejn, učestvuje u validaciji blokova, verifikuje sve blokove i stanja. Iz njega se mogu izvesti sva stanja. Na usluzi je mreži i pruža podatke na zahtev drugih klijenata. *Light Node* (Nepotpun čvor): čuva zaglavlje lanca i sve ostale podatke zahteva od drugih čvorova, validira podatke u odnosu na stanje u zaglavljima blokova. Koristan za uređaje niskog kapaciteta, kao što su mobilni telefoni koji ne mogu skladištiti gigabajte blokčejn podataka. *Archive Node*(Arhivski čvor): čuva sve što se čuva u punom čvoru i pravi arhivu istorijata svih stanja. Koristan je ukoliko je potrebno dobiti informaciju o stanju na računu na nekom određenom bloku. Takođe postoje posebni tipova klijenata, *Remote Client*-i (Udaljeni klijenti), koji nemaju mogućnost skladištenja blokčejna niti validiranja transakcija. Pristup blokčejnu im omogućuju *Full* klijenti kojima u potpunosti veruju ali na taj način gube garancije anonimnosti [3]. Ovakvi klijenti nude funkcionalnost digitalnog novčanika pomoću kojeg mogu da se iniciraju transakcije. Najpoznatiji digitalni novčanik koji koristi Ethereum platformu je *Metamask*.

Postoji nekoliko ključnih arhitekturalnih komponenti *Ethereum* klijenta. *Ethereum* virtuelna mašina (*Ethereum Virtual Machine - EVM*) predstavlja deo *Ethereum*-a koji upravlja razvojem i izvršavanjem pametnih ugovora. Na apstraktnijem nivou, *EVM* koji radi na *Ethereum* blokčejnu može se zamisliti kao globalni decentralizovani računar koji sadrži milione izvršnih objekata od kojih svaki ima svoje stalno skladište podataka. Zadatak *EVM*-a jeste da ažurira stanje *Ethereum*-a računanjem izmena važećih stanja kao rezultat izvršavanja koda pametnih ugovora. Pametni ugovori predstavljaju programe koji se obično pišu u višim programskim jezicima kao što je *Solidity*,a koji se potom kompajliraju u bajtkodove koje *EVM* može da razume. Oni se izvršavaju kao niz opkodova (*opcodes*), tj *EVM* mašinskih instrukcija. Transakcije se pomoću *Peer to Peer (P2P)* sloja propagiraju do ostalih komšijskih čvorova. *Miner* komponenta opisuje kreiranje i majnovanje(rudarenje) blokova. Poput bitkoina, kao algoritam za postizanje konsenzusa, *Ethereum* koristi dokaz o radu. Mesto gde se skupljaju sve transakcije pre nego što uđu u majning proces naziva se *Transaction pool*. To je globalni skup kojeg svi klijenti međusobno dele tako da bi svaki klijent trebao da poseduje identičan skup transakcija u datom momentu. Putem *API*-ja, čvorovi su u mogućnosti da komuniciraju sa *remote* klijentima kao što su digitalni novčanici koji su u mogućnosti da iniciraju nove transakcije [2].



Slika 1. Arhitektura Ethereum klijenta

Hard Fork jeste radikalna promena protokola koja čini da čvorovi koji koriste najnovije verzije blokčejna više ne prihvataju starije verzije blokčejna. Dodavanje novog pravila u kodu stvara situaciju takvu da jedna putanja sledi novi, ažurirani blokčejn, a druga putanja nastavlja po starim pravilima. Čvorovi koji nisu ažurirani će vrlo brzo dobijati blokove koje neće moći da interpretiraju tako da će i oni morati da se ažuriraju ukoliko žele da nastave aktivno da učestvuju na mreži [5]. Radikalne promene ne mogu tek tako da se izvršavaju i postoji detaljna analiza pre samih njihovih implementacija. Sve počinje od predloga za poboljšanja u *Ethereum*-u. Analogno Bitkoinovim *BIP*-ovima, postoji Ethereum Improvement Proposal (*EIP*). *EIP* predstavlja posebno dizajniran dokument koji *Ethereum* zajednici opisuje novu funkcionalnost ili nešto što je potrebno izmeniti u postojećem sistemu.

4. KORIŠĆENE TEHNOLOGIJE I ALATI

Go-Ethereum (*Geth*) predstavlja implementaciju koju aktivno razvija Ethereum fondacija, koja predstavlja "oficijelnu" implementaciju Ethereum klijenta. Obično svaki *Ethereum* blokčejn ima svoju Geth implementaciju. U praktičnom smislu Geth predstavlja interfejs komandne linije za pokretanje potpunog čvora implementiranog u *Go* jeziku. Instaliranjem i pokretanjem *Geth*-a može se započeti učestvovanje na *Ethereum* mreži i postoje sledeće mogućnosti: rudarenje (majnovanje) kriptovalute *Ether* validiranjem transakcija, prenošenje sredstava sa jedne adrese na drugu, kreiranje pametnih ugovora, istraživanje istorije blokova, kao i mnoge druge. Za skladištenje potpune kopije Ethereum blokčejna potrebno je imati najmanje 300 GB prostora na disku [6].

5. ARHITEKTURA PREDLOŽENOG REŠENJA

Svaka velika promena koja se usvoji na *Ethereum* protokolu obavezuje *Ethereum* klijente da ažuriraju svoje implementacije sa ciljem da nastave da rade na mreži. *London Hard Fork* predstavlja jednu od najvećih promena na mreži koja se dogodila nakon bloka 12 965 000 i sadrži skup od pet *EIP*-ova. Usvajanje ovih predloga značilo je da se u velikoj meri menja način procesuiranja transakcija. Ranije, svako ko je inicirao transakciju morao je rudarima priložiti *Gas fee* odnosno naknadu za gas. Međutim, dešavalo se da ove naknade mogu da narastu kada je velika gužva na mreži, tj kad veliki broj ljudi šalje transakcije u sličnom vremenskom periodu. U ovakvim situacijama desavalo se da majneri biraju transakcije koje su imale veći *Gas fee* što je dovodilo do toga da mnogi drugi korisnici dugo čekaju na overu transakcija. *EIP 1559* menja ovaj mehanizam tako što je uvedena osnovna naknada za svaku transakciju, *base fee*, koja biva "spaljena" umesto da ide rudarima. Cilj je zadržati mrežu na oko 50% korišćenja, što bi značilo da ako upotreba mreže počne da premašuje 50% kapaciteta, *base fee* se automatski povećava. Obrnuto, kako korišćenje mreže pada ispod 50%, *base fee* se blago smanjuje.

```
opcodes.go
core > vm > opcodes.go > POP
    // 0x40 range - block operations.
    const (
        BLOCKHASH OpCode = 0x40 + iota
        COINBASE
        TIMESTAMP
        NUMBER
        DIFFICULTY
        GASLIMIT
        CHAINID       OpCode = 0x46
        SELFBALANCE   OpCode = 0x47
        BASEFEE       OpCode = 0x48
    )
```

Slika 2. Dodavanje BASEFEE opcode-a

Predstavljeno rešenje pokazuje implementaciju promene u arhitekturi *Ethereum Geth* klijenta specificirane *EIP-om 3198*. *EIP 3198* predstavlja deo implementacije London hard fork-a i usko je povezan sa pomenutim *EIP-om 1559*. Specifikacija izmena za *EIP 3198* zahtevala je dodavanje novog opcode-a koji Ethereum virtuelnoj mašini daje mogućnost da pristupi base fee vrednosti bloka sa kojim u tom trenutku operiše.

```
if !chain.Config().IsLondon(header.Number) {
    // Verify BaseFee not present before EIP-1559 fork.
    if header.BaseFee != nil {
        return fmt.Errorf("invalid baseFee before fork: have %d, want <nil>", header.BaseFee)
    }
    if err := misc.VerifyGaslimit(parent.GasLimit, header.GasLimit); err != nil {
        return err
    }
} else if err := misc.VerifyEip1559Header(chain.Config(), parent, header); err != nil {
    // Verify the header's EIP-1559 attributes.
    return err
}
```

Slika 3. Provera konfiguracije Ethereum blokčejna

Na slici br. 3 vidi se način prelaska na novu konfiguraciju izazvanom *Hard Fork-om*. Kao što je napomenuto, prelazak treba da se dogodi nakon određenog bloka. Promena mora biti spremna, ali dok se blok sa rednim brojem predviđenim za novu konfiguraciju ne pojavi, sistem nastavlja da radi po starom.

6. ZAKLJUČAK

U ovom radu prikazana je evolucija blokčejn tehnologija od samog početka pa sve do danas. Prikazane su prednosti koje ovakvi sistemi imaju u odnosu na današnje koji su i dalje mahom centralizovani. Ekspanzijom interneta se proporcionalno povećao i broj malicioznih pokušaja krađa, kompromitacija podataka i slično. Tehnologija blokčejna osim sto nudi rešenja za takve probleme, ne zahteva od korisnika lične informacije kako bi mogli da učestvuju u digitalnim transakcijama. Sam fokus bio je na *Ethereum* platformi kao i arhitekturi njenih klijenata. Predstavljena je najpopularnija implementacija Ethereum kllijenta – *Go Ethereum(Geth).*

Blokčejn tehnologija je i dalje u fazi razvoja i bori se sa problemima kao što je skalabilnost. Potrebno je pronaći odgovore na što veći broj korisnika koji će se priključivati blokčejn mrežama. Takođe, nastavlja se potraga i za idealnim konsenzus algoritmima. Interoperabilnost je još jedna komponenta koju treba imati u vidu jer je potrebno obezbediti komunikaciju između različitih mreža, a različiti modeli često prave dodatne probleme u komunikaciji. Budućnost samog Ethereum protokola je teško predvideti, ali ono što je sigurno jeste da je njegov doprinos blokčejn tehnologijama već sad enorman. *Ethereum* će nastaviti svoj razvoj paralelno sa ostalim blokčejn platformama koje će vremenom biti sve mnogobrojnije. Ono što je evidentno je da će primena ovih tehnologija rasti u godinama koje su pred nama.

7. LITERATURA

[1] Tanenbaum, A.S. and Van Steen, M, Distributed systems: principles and paradigms. Third Edition, Februar 2022.

[2] Andreas M. Antonopoulos, Gavin Wood, Mastering Ethereum, https://github.com/ethereumbook/ethereumbok, Februar 2022

[3] Nodes and Clients, https://ethereum.org/en/developers/docs/nodes-and-clients, Februar 2022

[4] Blockchain Generations, https://www.investopedia.com/tech/blockchain-technologys-three-generations , Februar 2022.

[5] Hard Fork, https://www.investopedia.com/terms/h/hard-fork.asp, Februar 2022

[6] Geth Implementation, https://github.com/ethereum/go-ethereum , Februar 2022

[7] Distributed Ledger https://searchcio.techtarget.com/definition/distributedledger , Februar 2022.

[8] Ethereum Gas https://ethereum.org/en/developers/docs/gas/ , Februar 2022.

[9] Centralized, Decentralized and Distributed systems https://www.geeksforgeeks.org/comparison-centralized-decentralized-and-distributed-systems/ , Februar 2022

Kratka biografija:

Marko Jurić rođen je u Novom Sadu 1995. godine. Fakultet tehničkih nauka upisao je 2014. godine. Osnovne akademske studije završio je 2018. godine.