Analiza blockchain tehnologija

Mandić, Marko

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Marko Mandić

ANALIZA BLOCKCHAIN TEHNOLOGIJA

ZAVRŠNI RAD

Varaždin, 2019.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Marko Mandić

Matični broj: 0016109959

Studij: Poslovni sustavi

ANALIZA BLOCKCHAIN TEHNOLOGIJA

ZAVRŠNI RAD

Mentor:

Prof. dr. sc. Neven Vrček

Varaždin, srpanj 2019.

Marko Mandić

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Završni rad je rad koji je napravljen kao dokument za završetak preddiplomskog studija. Tema ovog završnog rada je „Analiza blockchain tehnologija“. Unutar rada govoriti će se o osnovnim pojmovima blockchain tehnologija, kao i vrstama blockchain-a, partnerima, te strukturi bloka i algoritmima. Nakon toga nešto više i detaljnije o pametnim ugovorima i ostalim dijelovima vezanim uz pametne ugovore. Za kraj se spominju kriptovalute, te zaključak i literatura rada.

Ključne riječi: blockchain; pametni ugovori; kriptovalute; wallet; NEO; Ethereum; EOS;

Sadržaj

1. Uvod

Kao što je već i naglašeno, tema završnog rada nosi naziv „Analiza blockchain tehnologija“. Blockchain tehnologija se na hrvatski jezik direktno prevodi kao lanac blokova, odnosno niz blokova koji su međusobno povezani i svaki blok ovisi o svom prethodniku. Navedeni se blokovi povezuju, a to se odvija pomoću kriptografije. U današnje vrijeme blockchain tehnologija je sve popularnija i sve više raste, možemo se tako prisjetiti situacije iz 2009.godine kada se prvi put pojavio pojam bitcoin. Bitcoin je zapravo digitalna valuta zbog koje je sama blockchain tehnologija nastala. Na početku, odnosno tek nakon prve pojave bitcoin-a njegova vrijednost bila je nevjerojatno manja nego danas. Iz godine u godinu vrijednost bitcoin-a se mijenjala, zabilježena su velika povećanja vrijednosti bitcoin-a ali i padovi u njegovoj vrijednosti. Početkom 2011. godine bitcoin je dosegao vrijednost od 1$, da bi se ta vrijednost u periodu od 2 godine popela na čak 266$. Najveća vrijednost bitcoin-a je zabilježena krajem 2017.godine kada je iznosila skoro 20 tisuća $. Dok trenutna vrijednost bitcoina, iznosi oko 12 tisuća i 620 $ [1].

Nadalje će se u radu fokus prebaciti na pametne ugovore, pametni ugovori su također dio blockchain tehnologije. To su zapravo digitalni ugovori koji se „sami“ izvršavaju. Oni zapravo predstavljaju određenu sigurnost za poslovanje stranaka koje nemaju međusobnog povjerenja jedna u drugu. No, o tome nešto više u sljedećim poglavljima rada.

2. Metode i tehnike rada

Prilikom izrade rada korišteni su alati koji omogućavaju kreiranje različitih dijagrama, te vlastitu izradu slika na temelju originalnih slika. Korištena je online verzija alata „draw“ koji služi za izradu dijagrama. Također alat nudi različite template-ove kako bi na temelju njih neki dijagram lakše i brže napravili. Primjena odnosno rad u alatu je vrlo jednostavan i ne oduzima puno vremena kako bi se navikli na njegove principe, online verziju alata *Draw.io* [2].

Osim navedenog alata za izradu završnog rada koristili su se i različiti izvori na internetu, bilo da se radi o običnim člancima sa portala ili različitim drugim izvorima. Rad se temelji na podacima pronađenim sa interneta, kao i na nekim osobnim znanjima o navedenoj temi.

3. Blockchain tehnologija

Spominje se dosta definicija blockchain-a pa se postavlja pitanje: što je zapravo blockchain? Blockchain je termin koji označava neki niz ili listu zapisa čije dijelove nazivamo blokovima, koji su povezani uz pomoć kriptografije. Svaki se blok pri tome sastoji od nekoliko različitih dijelova: hash-a prethodnog bloka, vremenske oznake, te podataka o transakciji. Blockchain je zapravo jedinstveno dizajniran, odnosno jedna od njegovih glavnih prednosti je otpornost na modifikaciju podataka [3].

„The blockchain is an incorruptible digital ledger of economic transactions that can be programmed to record not just financial transactions but virtually everything of value.“ (Don & Alex Tapscott, 2016)

Iz navedenog citata možemo zaključiti kako je blockchain zapravo digitalna knjiga koja sadrži ekonomske transakcije koja je programirana na taj način da ne bilježi samo one financijske transakcijske vrijednosti, nego i bilo što drugo što sadrži neku vrijednost ili može predstavljati neku vrijednost.

Kada se pogleda u povijest blockchain-a, prvi se put kriptografski povezani blokovi spominju još početkom 90-ih godina 21.stoljeća. Međutim početak blockchain-a se zapravo pripisuje 2008.godini. Tada je netko pod pseudonimom „Satoshi Nakamoto“ pokrenuo web stranicu „bitcoin.org“. Nakon toga analiziralo se i proučavalo tko se zapravo skriva pod navedenim pseudonimom, razni stručnjaci i ljudi koji se bave tim područjem su se pitali da li je pod tim pseudonimom obični individualac kao osoba, ili neka grupa osoba ili čak organizacija. Još uvijek nije otkriveno tko se nalazi iza navedenog pseudonima, koliko god se istraživalo ili proučavalo nikada se nije došlo do identifikacije. Navodi se kako se prilikom pokušavanja „googlanja“ pseudonima dobivaju fotografije neke osobe, međutim ta je osoba porekla sve veze s kriptovalutama [4].

Sama blockchain tehnologija nastala je onda kada su se pojavile potrebe za rukovanje digitalnom valutom Bitcoin. Blockchain tehnologija odnosi se na podatke koji se distribuiraju na servere na taj način da svi sudionici komunikacije imaju iste podatke zapisane kod sebe. To se odvija pomoću glavne knjige (eng. *Ledger*), ona predstavlja neki oblik baze podataka koja sadrži sve transakcije koje su pohranjene na mnogo mjesta. Transakcije koje se nalaze u njoj, odnosno podaci koji su u njoj mogu biti različitih vrsta iako su to najčešće oni financijski podaci. Kao što smo već naveli, sve je zapravo počelo pojavom bitcoina koji predstavlja virtualnu valutu kojom možemo plaćati različite usluge. No, taj bitcoin ne bi funkcionirao da ne posjeduje odgovarajuću tehnologiju. Upravo zbog toga je i blockchain tehnologija razvijena, tj. razvijena je da bi se lakše plaćale usluge putem bitcoina odnosno virtualnog novca [5].

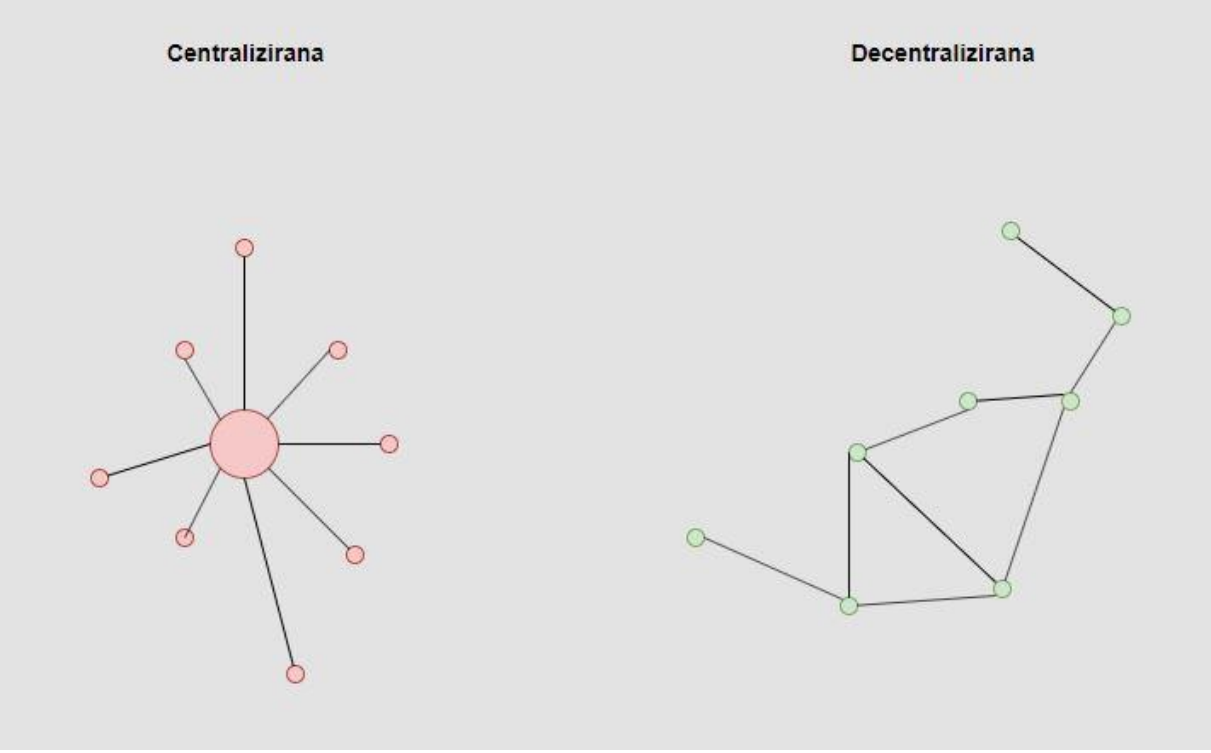
Dakako, navode se i neke od osnovnih značajki blockchain-a a to su [6]:

- Sustav je izgrađen prema modelu ravnopravnih partnera (eng. *Peer-to-peer*)
- Sustav je decentraliziran
- Svi novi zapisi su distribuirani u realnom vremenu
- Čvorovi sustava imaju mogućnost pisanja i čitanja podataka
- Itd.

Osim značajki, navode se i neki osnovni dijelovi blockchain tehnologije. Tj. kako ih nazivaju stupovi na kojima se temelji blockchain tehnologija [3]:

- Decentralizacija
- Transparentnost
- Nepromjenjivost

Decentralizacija se očituje u tome da se u današnje vrijeme decentraliziranom mrežom može direktno komunicirati sa prijateljem bez posredovanja treće strane. Odnosno, upravo to i stoji iz bitcoin-a. Možemo poslati svoj novac bilo kome drugome bez odlaska u banku. Slika 1. prikazuje strukturu centralizirane i decentralizirane mreže.



Slika 1. Struktura centralizirane i decentralizirane mreže „(Prema: Rosic, 2016)“;

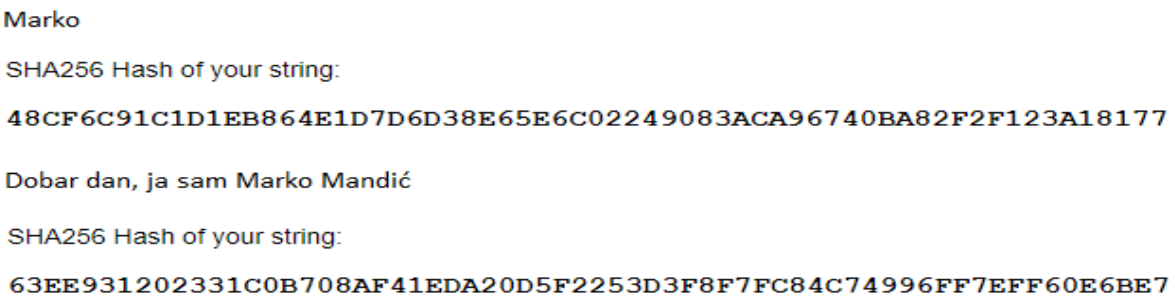
Kao što možemo vidjeti na slici 1. razlika između centralizirane i decentralizirane mreže je ta što kod centralizirane postoji srž odnosno „glavno tijelo“, dok kod decentralizirane ne postoji.

Također kod centralizirane samo privilegirani korisnici mogu vidjeti povijest transakcija ili mogu potvrditi novu transakciju, dok kod decentralizirane to mogu napraviti svi korisnici mreže.

Kada govorimo o transparentnosti, prilikom toga mislimo na privatnost identiteta osobe. Odnosno, identitet osobe je sakriven pomoću kriptografije tako da sadrži samo adresu ali ne i ime osobe. Tako kada govorimo o transparentnosti nećemo vidjeti npr. „Marko je poslao 10 $“, nego će se vidjeti npr. „3MuejFKEO3dmioDK33jIKo je poslao 10 $“.

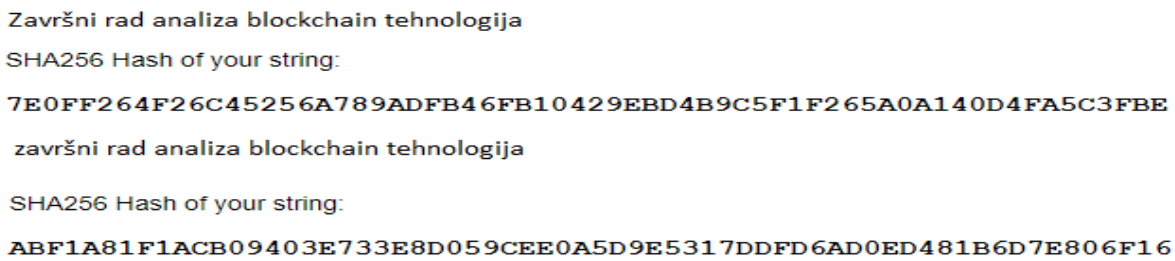
Zadnji „stup“ blockchain tehnologije je nepromjenjivost. To se može vrlo dobro prikazati na primjeru generiranja hash koda. Odnosno, unošenjem određenog inputa generira se output striktno određene duljine. Kako se kod bitcoin-a koristi SHA256, to će se koristiti i u ovom primjeru. Za generiranje hash-a koristit ćemo online verziju SHA256 hash generatora [8].

Sada ćemo navesti nekoliko primjera generiranja hash-a kako bi prikazali kako je duljina hash-a striktno određena bez obzira na duljinu inputa koji unosimo duljina outputa je uvijek jednake duljine. Također biti će prikazano i da bez obzira što je tekst jednak, odnosno razlikuje se u samo jednom velikom u odnosu na malo slovo hash će biti potpuno različit ali i dalje fiksne duljine.



Slika 2. Fiksna duljina output-a (https://passwordsgenerator.net/sha256-hash-generator/)

Slika 2. prikazuje da bez obzira na duljinu inputa, duljina outputa uvijek je fiksna. Na sljedećoj slici 3. možemo vidjeti kako se bez obzira na razliku veličine jednog slova prilikom čega je tekst potpuno jednak generira potpuno različit SHA256 hash.



Slika 3. Potpuno različit SHA256 unatoč istom input-u
(https://passwordsgenerator.net/sha256-hash-generator/)

3.1.Vrste blockchain-a

Kada govorimo o vrstama blockchain-a spominje ih se nekoliko. Oni zapravo posjeduju različite karakteristike, ali često čak i slične. Razlikujemo: javni blockchain, privatni blockchain, te konzorcijski blockchain. U nastavku će biti rečeno ponešto o svakom od njih.

3.1.1.Javni blockchain

Kada govorimo o različitostima blockchain-a misli se zapravo na ograničenja u pristupu. Kod javnog blockchain-a nema nikakvih ograničenja pristupa. Mogućnost slanja transakcija i postajanja validatorom je potpuno otvorena, odnosno tu mogućnost posjeduje svatko tko ima pristup internetu. Svaki korisnik javnog blockchain-a posjeduje kopiju javne knjige. Primjeri javnog blackchain-a su npr. ethereum, bitcoin [7].

Kada govorimo o javnom blockchain-u, razlikujemo zapravo dva načina na koje on može biti „javan“. Prvi način je način kod kojeg svi mogu čitati i zapisivati podatke, dok kod drugog načina svi mogu čitati podatke ali zapisivati ih mogu samo validirane osobe.

3.1.2.Privatni blockchain

Da bi se pristupilo privatnom blockchain-u potrebno je biti verificiran, tu postoje određena ograničenja za sudionika i validatora. Privatni blockchain se može uspješno koristiti u organizacijama, na taj način da podacima ne može pristupiti javnost. Privatni je blockchain zapravo namijenjen određenoj skupini ljudi, ili organizaciji te se upravo zbog toga i naziva privatni zato što nije dostupan nikome drugom osim određene skupine ljudi. Razlika između javnog i privatnog blockchain-a je upravo temeljena na sigurnosti, privatnim se blockchainom koriste oni koji nisu zadovoljni sigurnošću koju pruža javna mreža [7].

Jednostavnije rečeno, privatni blockchain koriste organizacije kako bi zaštitile svoje podatke ili transakcije od javnosti. Pomoću privatnog blockchain-a oni obavljaju svoje financijske ili druge transakcije unutar organizacije i na taj način se zapravo štite od šire javnosti kako ona ne bi mogla pristupati njihovim podacima.

3.1.3.Konzorcijski blockchain

Konzorcijski blockchain je zapravo vrlo sličan privatnom blockchain-u ali se razlikuje u jednoj stvari, a to se odnosi na čvorove. Naime, kod konzorcijskog blockchain-a postoji više glavnih čvorova za razliku od privatnog kod kojeg postoji samo jedan. To možemo također objasniti na način da je kod konzorcijskog blockchain-a uključeno više organizacija a ne samo jedna kao kod privatnog. Odnosno, više različitih organizacija je povezano u određenu mrežu

i svaki sudionik mreže ima pristup lancu i mogućnost kreiranja čvora unutar mreže. Unutar konzorcijskog blockchain-a administratori imaju mogućnost ograničavanja prava na način da točno odrede koje dijelove lanca korisnik može samo čitati, te također ograničavaju broj čvorova za konsenzusni protokol [7].

Jednostavnije rečeno, konzorcijski blockchain je zapravo mreža organizacija u kojoj svaki sudionik odnosno član određene organizacije posjeduje određene mogućnosti i također svaki sudionik ima ograničenja vezana uz to što on može raditi unutar navedene mreže.

3.2.Struktura bloka

Kao što smo već naveli u prethodnim dijelovima rada, blockchain je kako mu i ime govori sastavljen od niza blokova. Svaki blok predstavlja strukturu podataka unutar koje su zapisane informacije koje se dijele putem blockchain-a. Blokovi su povezani u lanac. Tablica 1. prikazuje strukturu bloka.

Tablica 1.Struktura bloka

Veličina	Naziv	Opis
4 BAJTA	Veličina bloka	Veličina bloka u bajtovima
80 BAJTOVA	Zaglavlje bloka	Meta-podaci
1-9 BAJTOVA	Brojač zapisa	Koliko zapisa sadrži blok
VARIJABILNO	Zapisi	Zapisi pohranjeni u bloku

(Izrada autora prema: Prasanna, 2018)

U tablici 1. možemo vidjeti kako izgleda struktura bloka. Naime, blok je sastavljen od četiri dijela a to su [9]:

- Veličina bloka – predstavlja veličinu bloka u bajtovima, odnosno daje nam predodžbu koliko je blok velik i koliko informacija sadrži. Možemo vidjeti kako je njegova veličina 4 bajta.
- Zaglavlje bloka – zaglavlje se sastoji od meta podataka o samom bloku, a veličine je 80 bajtova.
- Brojač zapisa – govori nam koliko zapisa sadrži blok, može biti veličine od 1-9 bajtova.
- Zapisi – dio strukture koji predstavlja zapise koji se nalaze u bloku, a njegova veličina je varijabilna upravo zbog toga što su i zapisi varijabilni.

3.2.1.Zaglavlje bloka

Sada ćemo opisati jedan od dijelova strukture bloka, odnosno zaglavlje bloka. Zaglavlje bloka je veličine 80 bajtova, a sadrži meta podatke o zapravo samom bloku kao i informacije potrebne za povezivanje blokova u lanac. Meta podaci su zapravo podaci koji pružaju informacije o drugim podacima.

Chan [10] navodi kako se zaglavlje bloka sastoji od tri skupa meta podataka. Prvi skup meta podataka se sastoji od reference na hash prethodnog bloka u lancu koji služi za povezivanje postojećeg bloka sa prethodnim. Drugi skup meta podataka odnosi se na rudarenje podacima, a uključuje dijelove: timestamp, difficulty targer, te nonce. Dok posljednji, odnosno treći skup metapodataka obuhvaća merkle root. U sljedećoj tablici 2. možete vidjeti strukturu zaglavlja bloka.

Tablica 2. Struktura zaglavlja bloka

Veličina	Naziv	Opis
4 BAJTA	Verzija (eng. *Version*)	Verzija protokola u vrijeme nastajanja bloka
32 BAJTA	Hash prethodnog bloka (eng. *Previouns block hash*)	Referenca na prethodni blok u lancu
32 BAJTA	Korijen binarnog hash stable (eng. *Merkle Root*)	Hash koji sadrži informacije o svim zapisima u bloku
4 BAJTA	Vremenska oznaka (eng. *Timestamp*)	Približno vrijeme kreiranja bloka
4 BAJTA	Težinska oznaka (eng. *Difficulty target*)	Težina algoritma potrebnog za uključivanje bloka u blockchain
4 BAJTA	Nonce	Broj koji služi kao pomoć u rješavanju algoritma za uključivanje bloka u blockchain

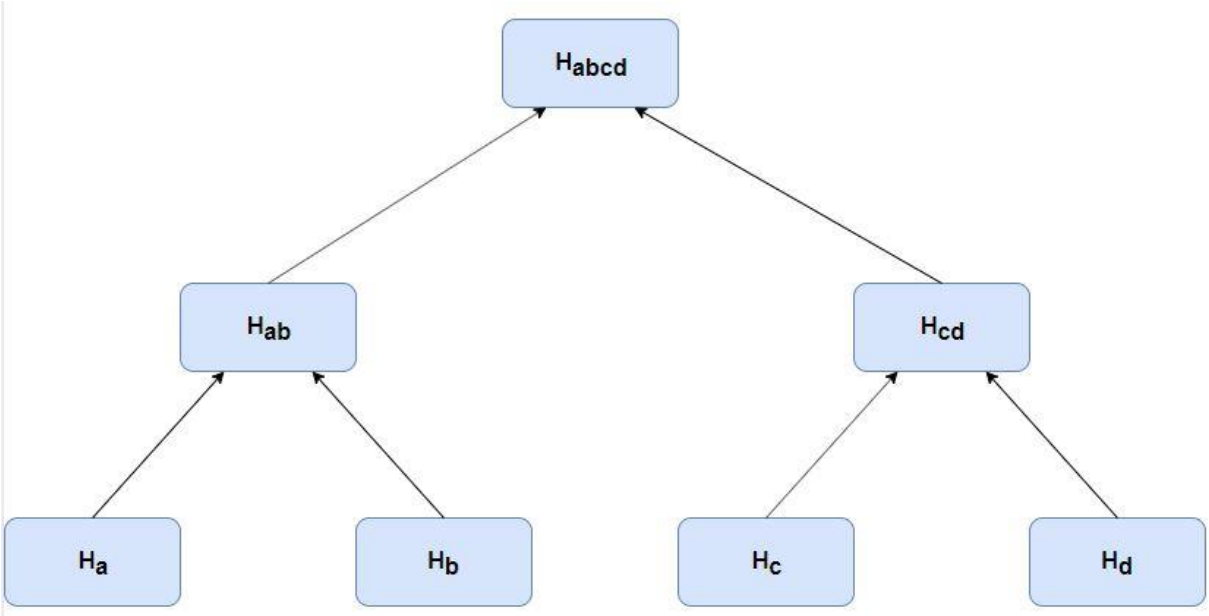
(Izrada autora prema: Chan, 2018)

Kao što možemo vidjeti u tablici 2. zaglavlje bloka sastoji se od šest dijelova prilikom čega svaki dio zaglavlja služi za posebnu funkciju. Možemo vidjeti kako je verzija dio koji se odnosi na verziju protokola u vrijeme nastajanja bloka a veličine je 4 bajta. Sljedeći dio je hash prethodnog bloka koji nam zapravo prikazuje referencu na blok koji prethodi trenutnom bloku u lancu i veličine je 32 bajta. Sljedeći dio prikazuje hash koji sadrži informacije o svim zapisima u bloku a veličine je 32 bajta. Nakon toga slijedi dio zaglavlja koji se odnosi na vremensku

oznaku, odnosno prikazuje približno vrijeme izrade bloka i veličine je 4 bajta. Težinska oznaka prikazuje težinu algoritma za uključivanje bloka u blockchain i veličine je 4 bajta. I zadnji dio naziva nonce se odnosi na broj koji služi kao pomoć prilikom rješavanja algoritma za uključivanje bloka u lanac i veličine je 4 bajta [10].

3.2.2.Binarno hash stablo

Binarno hash stablo je jedan od dijelova koji su usko povezani sa strukturom zaglavlja bloka. Binarno hash stablo je zapravo skup podataka istog tipa, prilikom čega postoji i određena hijerarhija. Svaki blok lanca sadrži korijen binarnog hash stabla, koji prikazuje sve zapise u bloku. Skup podataka unutar binarnog stabla nazivamo čvorovima. Svaki čvor može imati nekog „ispod“ sebe, odnosno djecu. Čvor koji nema djecu, nazivamo listom. Binarno hash stablo je zapravo hash vrijednost podataka kod listova, odnosno kod čvorova nastaje spajanjem vrijednosti njegove djece te kriptiranjem hash funkcijom. Kao hash funkcija koristi se SHA-256 primijenjena dva puta. Na kraju kada pohranimo korijen binarnog hash stabla, kao rezultat dobivamo sažeti prikaz svih zapisa bloka [11].



Slika 4. Binarno hash stablo „(Prema: Škegro, 2019)“;

Slika 4. prikazuje binarno hash stablo na kojem možemo vidjeti tri čvora i četiri lista. Čvorovi su u ovom slučaju H_{abcd}, H_{ab}, H_{cd} zbog toga što oni imaju djecu, odnosno H_{ab} i H_{cd} su zapravo djeca od H_{abcd}. Dok preostala četiri člana na ovoj slici predstavljaju listove zbog toga što oni nemaju djecu. Odnosno, nalaze se na zadnjoj razini hijerarhije i ispod njih nema nikog prema tome upravo ih zato nazivamo listovima. Listovi ovog binarnog hash stabla su: H_a, H_b, H_c, H_d.

Sada ćemo prikazati za primjer sa slike kako se računaju listovi, a zatim na njihovim vrijednostima kako se dobivaju vrijednosti čvorova. Prvi primjer je računanje listova H_a, H_b, H_c, H_d. Prikazat ćemo primjer računanja samo za H_a, a analogno tome tako se računaju i ostali listovi.

$$H_a = SHA - 256(SHA - 256(a))$$

Analogno tom primjeru, tako se računaju i ostali listovi. Sada, nakon što smo izračunali listove možemo prijeći na računanje čvorova. Čvorovi se računaju na način da se zbroje vrijednosti njegovih listova. Primjer računanja čvora H_{ab}.

$$H_{ab} = SHA - 256(SHA - 256(H_a + H_b))$$

Analogno primjeru računanja čvora H_{ab}, isto se računa i čvor H_{cd} samo se zbrajaju njegove vrijednosti. Na kraju se konačni prvi čvor H_{abcd} računa kao zbroj vrijednosti čvorova H_{ab} i H_{cd}. Odnosno, računa se na sljedeći način.

$$H_{abcd} = SHA - 256(SHA - 256(H_{ab} + H_{cd}))$$

Nakon prikazivanja izračuna potrebno je još napomenuti kako je veličina podataka zapisanih u korijenu stabla uvijek fiksna, odnosno uvijek je veličine 32 bajta bez obzira koliko čvorova stablo ima [12].

Na kraju potpoglavlja o binarnom hash stablu, potrebno je još napomenuti kako postoji autentikacijski put. Autentikacijski put je skup čvorova A binarnog hash stabla B za koje vrijedi sljedeće [11]:

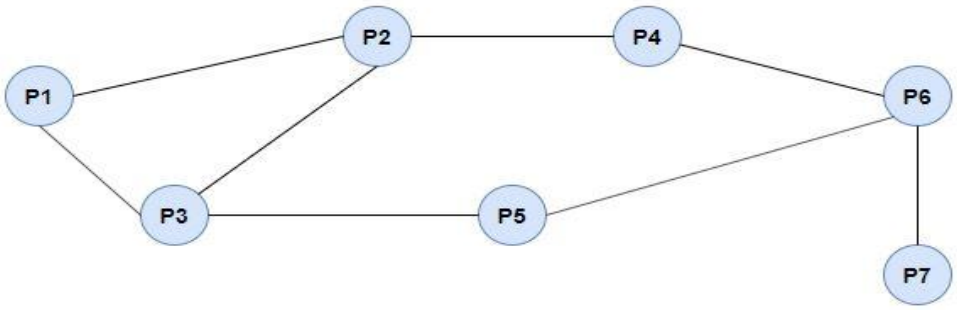
- Postoji točno jedan čvor a koji pripada skupu A za svaku razinu stabla B, osim prve razine
- Korijen binarnog hash stabla B moguće je odrediti ukoliko poznajemo još jedan list stabla B koji ne pripada skupu A

3.2.3.Povezivanje blokova

Kako bi što bolje shvatili povezivanje blokova, to ćemo prikazati na što jednostavniji način. Odnosno, zamislite da su blokovi stranice ovog završnog rada ili stranice bilo kakve knjige prilikom čega je svaka stranica numerirana odnosno sadrži svoj broj ili indeks. Također svaka stranica ima i referencu na svoju prethodnu stranicu, odnosno to bi u pogledu blokova bila referenca na prethodni blok. Upravo to numeriranje, bi nam pomoglo kada bi npr. izgubili redoslijed stranica odnosno kada bi nam se pomiješale mogli bi ih vratiti u prvobitno stanje upravo zbog toga što svaka ima svoj broj i prethodnika. Također moguće je odrediti i validaciju podataka pomoću hash funkcije. Kada bi netko htio napraviti izmjenu podataka nekog bloka, prvo bi morao promijeniti i sve hash-eve od tog trenutka nadalje [7].

3.3.Blockchain partneri

Vezano uz blockchain partnere spominje se decentralizirani sustav koji se zapravo zasnivan a ideji ravnopravnih partnera. Osim decentraliziranih postoje i centralizirani sustavi koje smo već prikazali u prethodnim poglavljima. Kod decentraliziranog sustava nema poslužitelja za razliku od centraliziranih, pa će tako kod decentraliziranih sustav će se sastojati od niza povezanih partnera. Decentralizirani sustav zapravo predstavlja ravnopravne partnere (eng. *Peer-to-peer*), prilikom čega riječ *peer* označava partnere. Sljedeća slika 5.prikazuje arhitekturu decentraliziranih sustava odnosno model ravnopravnih partnera [6].



Slika 5. Arhitektura modela ravnopravnih partnera „(Prema: Hozjan, 2017)“;

Slika 5. prikazuje arhitekturu sustava građenog na modelu ravnopravnih partnera, možemo vidjeti kako su svi partneri jednaki odnosno ne postoji nikakav poslužitelj između njih što znači da je sustav decentraliziran.

Živković [7] navodi kako postoje četiri funkcije koje partneri mogu obavljati u sustavu blockchain-a, a to su:

- Novčanik (eng. *Wallet*)
- Mrežno usmjeravanje (eng. *Network routing*)
- Održavanje blockchain-a
- Rudarenje (eng. *Mining*)

U sljedećim potpoglavljima biti će detaljno objašnjeni blockchain partneri, vrste blockchain partnera su [6]:

- Potpuni partner
- Rudar
- Jednostavni novčanik
- Blockchain partner

3.3.1. Jednostavni novčanik

Novčanik se u blockchain-u koristi za kupovinu i prodaju, te praćenje stanja nova ili drugih vrijednosti. Novčanik zapravo sadrži sve transakcije, prilikom čega je svaka transakcija bazirana na kriptografiji koja koristi dvije vrste ključeva [12]:

- Javni ključ
- Privatni ključ

Prilikom toga, javni i privatni ključ se razlikuju u sljedećem. Javni ključ je zapravo ključ koji sadrži generator adrese tako da bi se mogla primiti transakcija nekog drugog korisnika. Dok se privatni ključ koristi za pristupanje adresi i sredstvima koja se nalaze u novčaniku. Često se privatni ključ poistovjećuje sa nekim načinom identifikacije, npr. pin bankovne kartice, lozinka nekog računa, šifra na nekoj mreži, itd. Novčanik se također po jednoj stvari razlikuje od ostalih, a to je da je kod njega metoda verifikacije posebna odnosno pojednostavljena. Odnosno, za razliku od ostalih on pohranjuje samo zaglavlje blokova ali ne i cijeli blockchain. Prilikom validacije za pregled nad cjelovitim blockchain-om služe se uz pomoć svojih partnera. Prvo se verificira lanac posebno bez transakcije, te se kasnije povezuje sa transakcijama. Za te veze između lanaca i transakcija koristi se hash binarno stablo. Verifikacije se završava na način da rudari odrade svoju zadaću i tek tada će novčanik znati da je transakcija sigurna, zadatak rudara je da popune lanac potrebnim brojem blokova odnosno dodavanje 6 blokova [7].

Škegro [12] navodi kako postoji nekoliko vrsta različitih novčanika, u nastavku ćemo ih navesti i ukratko napisati ponešto o svakom od njih:

1) Novčanik u oblaku – u jednu ruku najjednostavnija, ali u drugu i najrizičnija mogućnost korištenja novčanika. Koristi se samo email i lozinka, međutim jako je lako izgubiti novac odnosno ostvariti gubitke zbog lažnih internetskih stranica.
2) Novčanik za mobilne uređaje – novčanik koji se može instalirati na mobilne uređaje, prilikom čega je potrebno imati dobru zaštitu na mobilnom uređaju radi sigurnosti digitalnog nova.
3) Desktop novčanik – novčanik kao što i samo ime kaže za desktop verziju bilo to računala ili prijenosnog računala. Sigurniji od novčanika u oblaku, nužan antivirusni program.
4) Hardware novčanik – funkcionira bez povezivanja na Internet, te je zapravo najsigurniji novčanik.

3.3.2. Rudar (eng. *Miner*)

„Rudari su računala koja preuzimaju transakcije od novčanika i pohranjuju ih na blockchain.“
(Škegro, 2019)

Prema navedenom citatu funkcija rudara je prihvaćanje novih zapisa koji dolaze od strane novčanika, nakon toga kreiranje blokova od tih zapisa i zatim na kraju smještanje blokova u blockchain. Postavlja se pitanje kako se ti blokovi zapravo dodaju u blockchain, to se odvija pomoću algoritma proof-of-work kod bitcoina o kojem će nešto više biti rečeno u sljedećem poglavlju. Rudari za svoj odrađeni posao bivaju nagrađeni, odnosno dobivaju određeni iznos bitcoina [6].

Škegro [12] navodi kako je proces rudarenja novih blokova kružni proces, a sastoji se od nekoliko pravila:

- Povećava se broj potvrđenih blokova
- Sustav automatski povećava težinu rudarenja
- Smanjuje se broj potvrđenih blokova
- Sve više rudara se spaja u mrežu
- Vrijeme potrebno za potvrdu bloka se smanjuje

3.3.3. Blockchain partner

Živković [7] navodi kako je glavna funkcija blockchain partnera da održava cijeli lanac sa svim zapisanim podacima u njemu, od prvog do zadnjeg bloka. Također, navodi kako blockchain partner može provjeriti autentičnost podataka i da li oni stvarno pripadaju korisniku prilikom validacije određene transakcije.

Prema Hozjan [6] blockchain partner održava blockchain sa svim zapisima počevši od prvog bloka kojeg nazivamo generički blok, sve do svih ostalih blokova. Nema potrebe za oslanjanjem na partnere, dok se partner oslanjanja na ostatak mreže kako bi primio novokreirane blokove.

3.4.Algoritmi

Algoritmi koji se odnose na blockchain tehnologiju, su zapravo algoritmi za postizanje konsenzusa. Odnosno, predstavljaju postizanje dogovora o vrijednosti podataka među procesima ili sustavima. Konsenzus je može se reći neki način „kompromisa“ odnosno, postizanje dogovora cjeline koja je najbolja za ostale u sustavu. Postizanje konsenzusa

predstavlja najvažniju stavku sigurnosti blockchain-a. Kao što smo već naveli, u prethodnim dijelovima kod bitcoina se koristi algoritam proof-of-work [6].

Škegro [12] navodi nekoliko ciljeva algoritama za postizanje konsenzusa, svaki od njih će biti ukratko objašnjen a to su:

- Dogovor – prikupljanje sporazuma iz grupe koliko god je to moguće
- Suradnja – svaki pojedinac će se založiti za grupu i svoje će interese odložiti sa strane i fokusirat će se na zajedničke
- Sudjelovanje – svaki član mreže mora sudjelovati u glasanju
- Kolaboracija – postizanje dogovora koji će kao rezultat dati interese cijele skupine
- Jednaka prava – jednakost sudionika prilikom glasanja
- Aktivnost – članovi grupe su jednako aktivni, sa jednakim odgovornostima

U sljedećim potpoglavljima ovog poglavlja definirati ćemo nekoliko različitih algoritama i svaki od njih ukratko objasniti. Algoritmi za postizanje konsenzusa su [7]:

- PoW (eng. *Proof-of-work*)
- PoS (eng. *Proof-of-stake*)
- DPoS (eng. *Delegated proof-of-stake*)

3.4.1.PoW (eng. *Proof-of-work*)

Proof-of-work je zapravo konsenzusni algoritam u blockchain tehnologiji, algoritam se koristi za potvrdu transakcija i dodavanje novih blokova u lanac. Može se reći da se takvih algoritmom rudari bore transakcijama kako bi dobili što bolju nagradu u obliku bitcoin-a. Algoritam je implementiran u blockchain na način da rudari rješavaju neki tip zagonetke te dodaju nove blokove i potvrđuju transakcije. Kod proof-of-work algoritma hash svakog bloka sadrži i hash prethodnog bloka upravo zbog sigurnosti. Također, proof-of-work se koristi u kriptovalutama, a najpoznatija kriptovaluta u kojoj se primjenjuje proof-of-work je bitcoin. Vrijeme potrebno da bi proof-of-work algoritam stvorio novi blok je oko 10 minuta. Također, osim kod bitcoin-a proof-of-work se koristi i kod Etheruma [13].

Ideja o algoritmu proof-of-work nastala je još 1993.godine, a razvili su je Cynthia Dward i Moni Naor tako da su objavili znanstveni članak koji je obuhvaćao dijelove borbe protiv neželjene pošte unutar memorije. Proof-of-work predstavlja funkciju koju je teško izračunati, ali lako ju je provjeriti. Osim poruke i adrese, funkcija sadrži i nekoliko drugih parametara. Postoji zanimljivost vezana uz slanje neželjene pošte, naime navodi se kako je računalo u mogućnosti dnevno poslati milijune neželjenih poruka svaki dan međutim ukoliko računalo

mora potrošiti deset sekundi na svaki poruku tada je moguće poslati samo osam tisuća poruka u danu. !999.godine Markus Jakobsson i Ari Juels uveli su pojam proof-of-work i stvorili notaciju [14].

Bruno [15] navodi kako proof-of-work osim određenih prednosti, također ima i neke nedostatke. Kao prednosti navode se:

- Efekt vanjskog faktora
- Jednostavan pool mining
- Korisnost za krajeve s viškom električne energije

Naravno, postoje i nedostaci proof-of-work algoritma a to su:

- Proof-of-work rudarenje nije moguće sa malim uređajima (npr. smartphone)
- Proof-of-work rudarenje blokova je sporo
- Proof-of-work rudarenje troši velike količine struje
- Proof-of-work rudarenje omogućava centralizaciju valute
- Manje tokena za rudare zbog smanjenja block nagrade

3.4.2. PoS (eng. *Proof-of-stake*)

Lisk navodi kako Proof-of-stake predstavlja konsenzusni algoritam koji se koristi kriptovalutama za validaciju blokova u blockchain tehnologiji. Prvi put takav algoritam spominje se 2011.godine, dok je prva implementacija zabilježena 2012.godine kriptovalutom Peercoin. Kada se govori o prednostima, spominju se dvije prednosti a to su sigurnost i energetska učinkovitost [16].

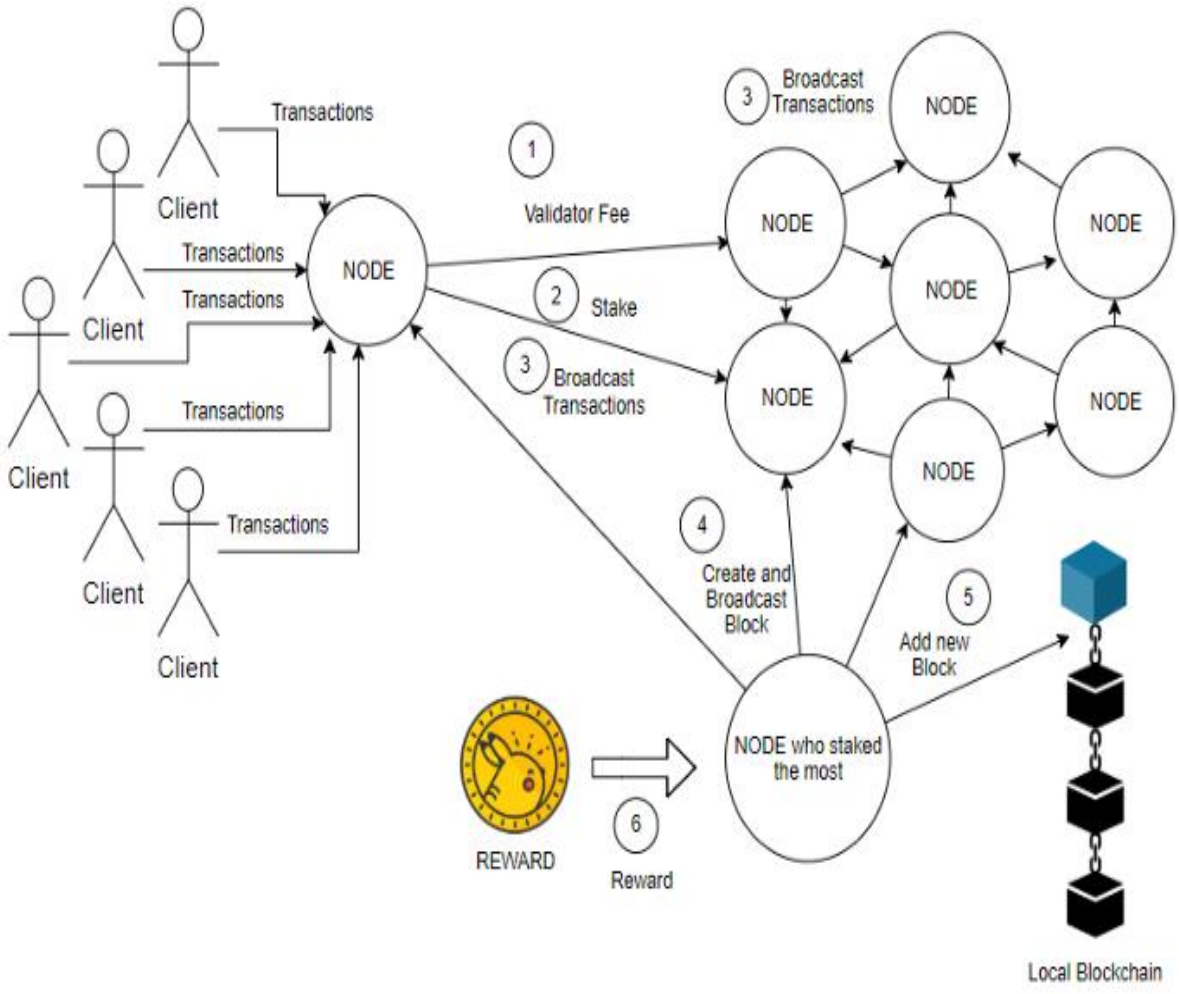
Bruno [15] navodi kako za proof-of-stake algoritam nisu potrebna jaka računala, niti velike količine električne energije. Neke od primjera za koji se algoritam proof-of-stake koristi su: Mintcoin, HyperStake, OKCash, Diamond, Blackcoin, Etherum, itd. Funkcioniranje algoritma proof-of-stake započinje nasumičnim određivanjem validatora, zatim validator ulaže Ether i obvezuje se da će potvrditi istinitost transakcija. Nakon toga, kada stigne nova transakcija potrebno je da ju potvrde ostali validatori a to se odvija tako da ju prvi validator šalje na mrežu ostalima i nakon potvrđivanja validator uzima proviziju transakcije.

Isto kao i algoritam proof-of-work, tako i algoritam proof-of-stake posjeduje određene prednosti i određene nedostatke. Kao prednosti algoritma proof-of-stake navode se [15]:

- Brzina obrade transakcija
- Ne ranjivost
- Ne štetnost za okoliš
- Mogućnost obrade manjim i slabijim uređajima

Bruno [15] također osim prednosti, navodi i neke nedostatke:

- Nema eksternih faktora – ulaže se samo novčana vrijednost, nije potrebno uložiti ostale faktore npr. znanje, struju, itd.
- Bogatiji se bogate – najviše ethera rezultira najvećom mogućnošću postajanja validatorom



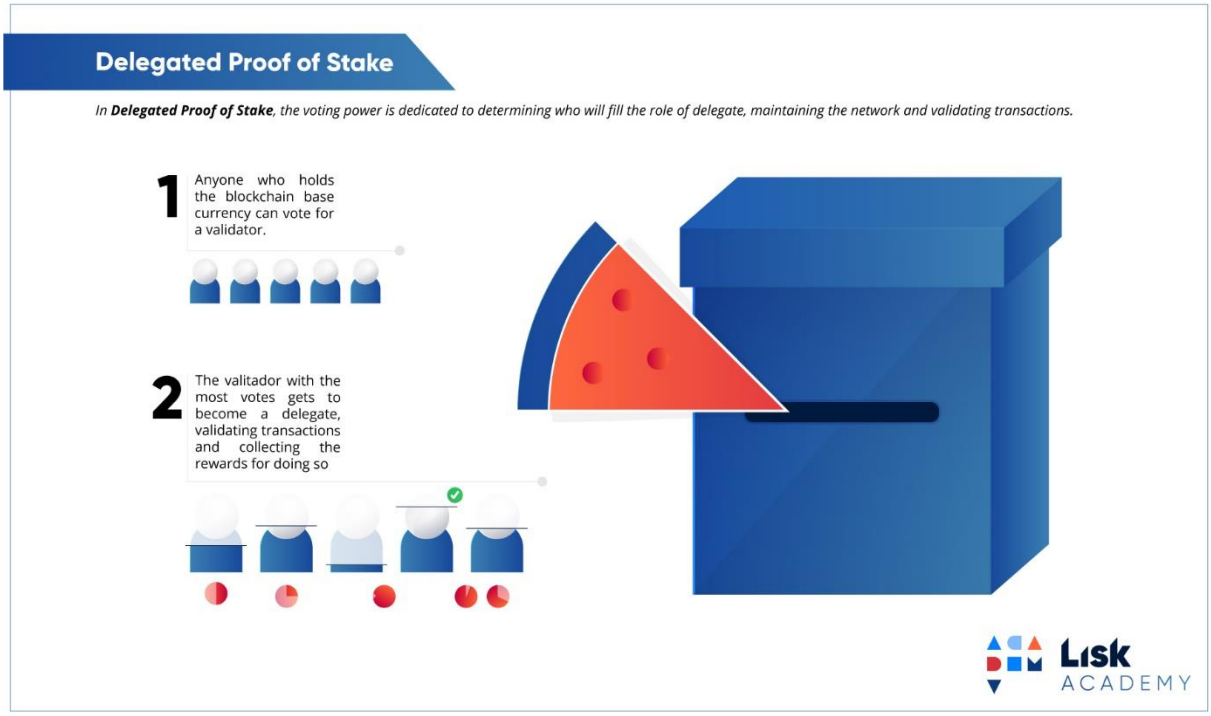
Slika 6. Struktura Proof-of-Stake [17]

Slika 6. prikazuje strukturu algoritma za postizanje konsenzusa proof-of-stake kod blockchain tehnologije. Kao što vidimo na slici, struktura algoritma proof-of-stake se sastoji od šest koraka. Prvo je potrebno od strane čvora koji želi postati validatorom da obavi validatorsku pristojbu odnosno potvrdi transakciju, zatim dolazi do drugog koraka u kojem čvor nakon potvrde transakcije uloži neke novce kako bi se nadmetao sa ostalim validatorima. Nakon toga svaki čvor prenosi transakciju koje je primio od klijenata, zatim nakon dovoljnog broja

transakcija bira se vodeći validator. U pretposljednjem koraku vodeći validator dodaje novi blok, te nakon toga u zadnjem koraku vodeći validator prima nagradu [17].

3.4.3. Delegirani PoS (eng. *Delegated proof-of-stake*)

Delegirani proof-of-stake je algoritam za postizanje konsenzusa koji potvrđuje transakcije i djeluje kao oblik digitalne demokracije, ali pri tome održava i sporazum o istinitosti. Prilikom glasanja validatora koristi se stvarnim vremenom, u usporedbi sa ostalima to je algoritam koji je najmanje centraliziran. Također, problem kod DPoS-a je što nema mogućnost mijenjanja detalja transakcije. Međutim, postoji mogućnost isključivanja određenih transakcija od strane validatora. Kada govorimo o mreži kod delegiranog proof-of-stake algoritma ona je samoupravljana i nadzirana [18].



Slika 7. DPoS (eng. *Delegated Proof-of-Stake*)

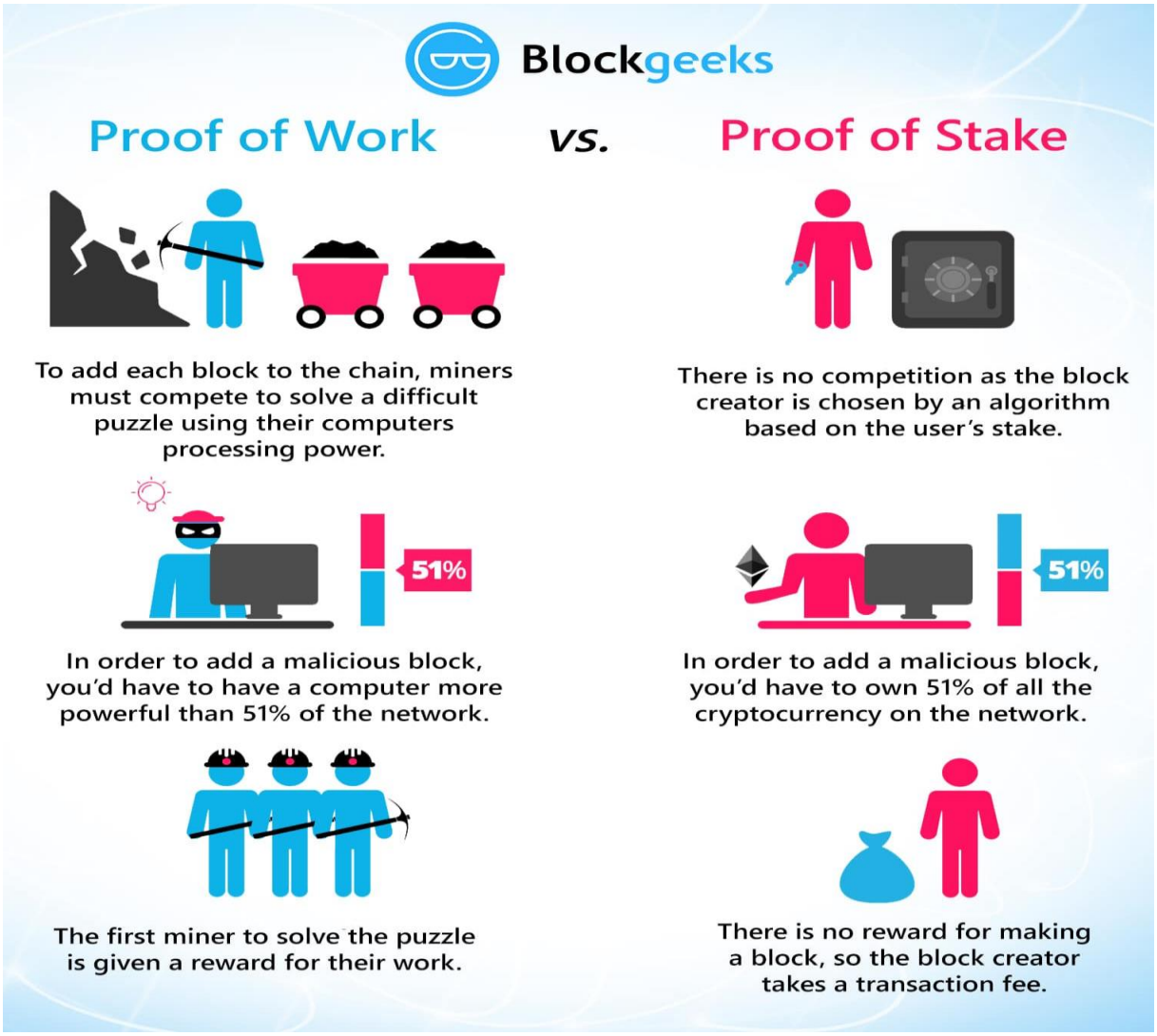
Slika 7. prikazuje način na koji se odvija glasanje prilikom odabira validatora kod algoritma proof-of-stake. Kao što vidimo na slici to je proces koji se odvija u dva koraka. Prvi korak je taj da pravo glasanja za validatora imaju svi oni koji posjeduju osnovnu valutu blockchain-a. Dok u drugom koraku validatorom postaje onaj sa najviše glasova, te zatim on validira transakcije i prikuplja nagrade za obavljeni posao.

3.4.4. Usporedbe navedenih algoritama

U ovom dijelu rada biti će ukratko opisane razlike između tri navedena algoritma konsenzusa kod blockchain-a. Odnosno, biti će ukratko objašnjeni glavni dijelovi po kojima se algoritmi razlikuju.

3.4.4.1. PoW vs PoS (eng. *Proof-of-wok vs proof-of-stake*)

Algoritam proof-of-work je dobro testiran i koristi se u mnogim projektima vezanim uz kripto valute. Kao najbolja usporedba navedena dva algoritma koristiti će se sljedeća slika.



Slika 8. Proof-of-Work vs Proof-of-Stake [20]

Slika 8 prikazuje razliku između navedena dva algoritma konsenzusa a to su proof-of-work i proof-of-stake. Razlika između navedenih algoritama očituje se u tri dijela. Prvi dio je taj da kod algoritma PoW kako bi se dodao blok u lanac rudar mora riješiti zagonetku koristeći

računalnu snagu, dok je kod PoS-a kreator bloka odabran na temelju algoritma koji se bazira na korisničkom udjelu. Drugi dio razlike je vezan uz mrežu, odnosno kod PoW-a je potrebno imati računalo koje je mrežno jače od 51%, dok kod PoS-a je bitno da posjedujete 51% cjelokupnih kriptovaluta na mreži. Zadnji dio razlike odnosi se na nagrađivanje, odnosno kod PoW-a se nagrađuje rudar koji prvi riješi zagonetku, dok kod PoS-a ne postoji nagrada za izradu bloka nego kreator bloka uzima naknadu za transakciju.

3.4.4.2. DPos vs Pos (eng. *Delegated proof-of-stake vs proof-of-stake*)

Glavna razlika između delegiranog PoS-a i PoS-a je ta što kod DPoS-a je uveden sustav demokratskog glasanja prilikom biranja proizvođača blokova. Kod DPoS-a postoji motiviranost delegata vezana uz poštenje i iskrenost zbog toga što suprotno tome moguće je da budu izbačeni iz procesa glasanja. Također, kao jedna od glavnih razlika navodi se i brzina DPoS-a u odnosu na PoS [19].

3.4.4.3. DPoS vs PoW (eng. *Delegated proof-of-stake vs proof-of-work*)

DPoS sustavi mogu obraditi puno veće količine transakcija od PoW-a. Dok se proof-of-work algoritam i dalje smatra najsigurnijim algoritmom. Delegated proof-of-stake je unaprijeđen u odnosu na algoritam proof-of-work na način da je proizvodnja blokova unaprijed određena, dok se kod algoritma proof-of-work ona određuje na temelju tržišnog natjecanja [19].

4. Pametni ugovori

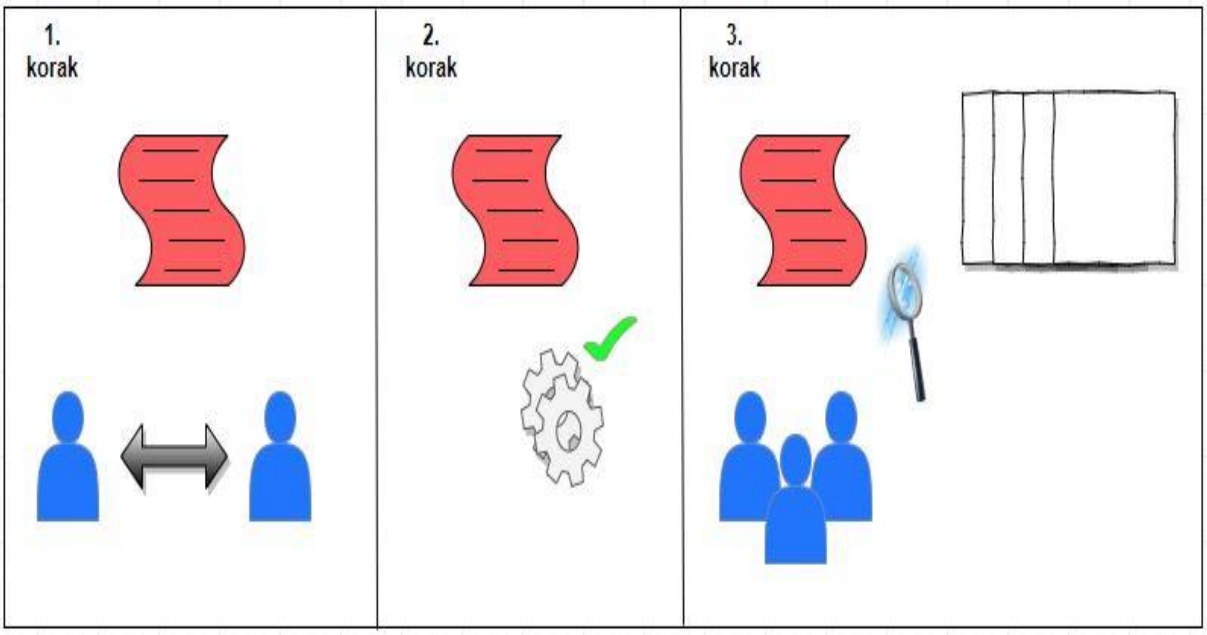
Kada govorimo o povijesti pametnih ugovora, oni se prvi puta spominju od strane američkog kriptografa i programera Nick Szabo-a. On je pojam pametnih ugovora prvi put spomenuo davne 1996.godine što je zapravo i prije nastanka same blockchain tehnologije. Prema njegovim navodima, pojam pametnih ugovora zapravo predstavlja digitalne protokole koji se izvršavaju automatski nakon što su zadovoljeni uvjeti i proces je kontroliran [21].

Voras [22] navodi kako su zapravo pametni ugovori koncept koji se zadnjih godina često spominje, naime najčešće ga se povezuje sa pojmom blockchain-a. Navodi se kako su pametni ugovori jedna od mogućih nadogradnji blockchain-a. To je zapravo dogovor između dvije strane prilikom čega će zapravo o programskom kodu ovisiti da li će se nešto dogoditi. Naime, ako se zadovolje uvjeti koji su postavljeni vezani uz izvršavanje tada će se to desiti. Odnosno, za razliku od običnih ugovora nije potrebna ona „treća“ strana koja će ugovoru svjedočiti, odobriti, itd.

Pametni ugovori (eng. *Smart contracts*) se koriste za lakšu i sigurniju razmjenu različitih vrsta sredstava bez potrebe posrednika. Naime, oni sami izvršavaju određene procese nakon što su zadovoljeni uvjeti koje su strane ugovora postavile. Različite vrste sredstava se mogu razmijeniti pametnim ugovorima, poput: novca, robe, nekretnina, imovine, kao i drugih [23].

Hozjan [6] navodi proces kreiranja pametnih ugovora. Naime, proces kreiranja pametnih ugovora sastoji se od tri koraka:

1. Razmjena dobara između dvije (ili više) strane zapisuje se u obliku programskog koda i pohranjuje, prilikom čega je sadržaj ugovora javan dok su strane tajne
2. Prema pravilima i uvjetima definiranim i zapisanim u programskom kodu, na izvršavanje ugovora utječu različite varijable kao što je npr. datum
3. Ostali korisnici imaju mogućnost pretraživanja blockchain-a, kako bi detaljno proučili i razumjeli ugovor ili vidjeli njegov rezultat



Slika 9. Proces kreiranja pametnog ugovora „(Prema: Hozjan, 2017)“;

Slika 9. prikazuje proces kreiranja pametnog ugovora koji je opisan na prethodnoj stranici, naime proces se sastoji od tri koraka. Plavi oblici „čovjeka“ odnosno korisničke eng. *user* ikone predstavljaju strane koje sudjeluju u ugovoru. Crveno su označeni ugovori, u drugom koraku vidimo zelenu kvačicu koja označava da su uvjeti definirani u ugovoru zadovoljeni. Dok u trećem koraku možemo vidjeti kako više strana može pregledati ugovor kao i rezultate.

Nosikov [21] navodi nekoliko prednosti pametnih ugovora koja će biti nabrojane u nastavku, te ukratko objašnjene. Prednosti pametnih ugovora su:

- Brzina
  - Ušteda vremena u obliku automatiziranja procesa, odnosno za razliku od običnih ugovora ne zahtijeva se ljudska uključenost pa se smanjuje vrijeme potrebno za obradu
- Pouzdanost
  - Ne mogućnost brisanja ili mijenjanja podataka, te zaštićenost odnosno sigurnost strane koja je ispunila svoje uvjete
- Neovisnost
  - Ne postojanje treće strane, odnosno očuvanje integriteta
- Bez pogrešaka
  - Kako ne postoji ljudski faktor, odnosno automatizacija osigurava točnost
- Štednja
  - Štednja odnosno ušteda troškova posrednika i operativnih troškova

Nosikov [21] navodi kako pametni ugovori posjeduju i određene nedostatke. Naime, nedostaci kod pametnih ugovora su:

- Nedostatak regulacije
  - Nedovoljno točna definiranost koncepata vezanih uz pojmove poput blockchain-a, pametnih ugovora i kriptovaluta
- Poteškoće implementacije
  - Zahtijeva se puno vremena, novca i truda
- Nemogućnost mijenjanja pametnog ugovora
  - Ne postoji mogućnost mijenjanja odnosno izmjene ugovora

4.1.Platforme za pametne ugovore

Navodi se kako u današnje vrijeme postoji više vrsta različitih platformi za pametne ugovore, međutim u ovom radu je odlučeno da će se spomenuti one koje se u današnje vrijeme najviše spominju na različitim internetskim stranicama. Osim općenitih i osnovnih informacija o svakoj platformi ponaosob, također će biti navedene i njihove karakteristike kao i dan primjer za svaku platformu. Nakon pregleda internetskih stranica vezanih uz ovu temu, platforme koje će biti obrađene su: NEO, Ethereum, te EOS. Također, u današnje vrijeme je najpopularnija platforma za pametne ugovore Ethereum pa će ona biti nešto više obrađena od ostalih.

4.1.1.Ethereum Virtual Machine (EVM)

Bisade [24] navodi kako je Ethereum Virtual Machine zapravo platforma koja se koristi za kreiranje pametnih ugovora, te je ujedno i najčešće primijenjena platforma koja se za to koristi. Ethereum Virtual Machine je neki oblik kvazi Turingovog stroja, koji je nazvan po Turingu odnosno znanstveniku koji je izumio Turingov stroj. Programski jezici i procesor omogućuju pristup i manipulaciju podacima. Ethereum Virtual Machine je zapravo kvazi Turingov stroj zbog toga što postoje ograničenja izračuna.

King [25] navodi kako je zapravo Ethereum nastao, odnosno 2012.godine Vitalik Buterin se prvi put susreo sa Bitcoin kriptovalutom koju mu je predstavio otac i na taj način se on počeo zanimati za tu tehnologiju. Vitalik je odlučio uvesti unaprjeđenja u postojeću tehnologiju i poboljšati platformu, tu je zapravo razvio ideju Ethereum-a. Ethereum se pojavio 2015.godine i od nastanka je sve više i više rastao i stjecao sve veću vrijednost. Ethereum Internet verzija

je verzija koja se razlikuje od standardne po tome što su poslužitelji i oblaci zamijenjeni mrežom sustava koja se naziva čvorovima. Te se tu zapravo pojavljuje pojam blockchain baze.

Bartolović [23] navodi da Ethereum platforma koristi veliku količinu ljudi koji moraju pokrenuti softver na vlastitim računalima kako bi se ostvarilo napajanje mreže. Ethereum Virtual Machine djeluje poput operacijskog sustava i izvršava programe, prilikom čega su programi zapisani u programskom jeziku Solidity. Ti programi koje Ethereum Virtual Machine izvršava zapravo nazivamo pametnim ugovorima, o kojima smo nešto više spomenuli u prethodnim poglavljima. Da bi se određeni dio izvršio potrebno je ispuniti uvjete plaćanja odnosno platiti „cijenu“, koja je izražena kriptovalutom naziva ETHER. Unutar Ethereum Virtual Machine postoji potpuna jednakost korisnika odnosno strana koje sudjeluju u pametnom ugovoru tj. jednakost korisnika i pametnog ugovora. Navodi se kako svaka radnja koju želimo da se izvrši unutar platforme Ethereum Virtual Machine ima svoju cijenu, odnosno cijenu koju treba platiti kako bi se ta radnja izvršila. Također, moguće je izbjeći beskonačne petlje definiranjem maksimalne cijene tako da u slučaju kada platforma dostigne tu definiranu cijenu prekine sa svojim izvršavanjem.

Joudrey [26] navodi da je Ethereum zapravo blockchain platforma koja korisnicima omogućuje izgradnju i korištenje decentraliziranih aplikacija (eng. *DApps*) koje se pokreću na blockchain tehnologiji. Razlika Ethereum-a u odnosu na Bitcoin je ta što je Ethereum prilagodljiv i fleksibilan. Platforma omogućuje različite tipove decentraliziranih aplikacija te naravno i kriptovalute.

Slika 10. Ethereum logo [26]

Slika 10. prikazuje Ethereum logo koji je zapravo službeni logo Ethereum platforme od 2014.godine.

4.1.2.NEO

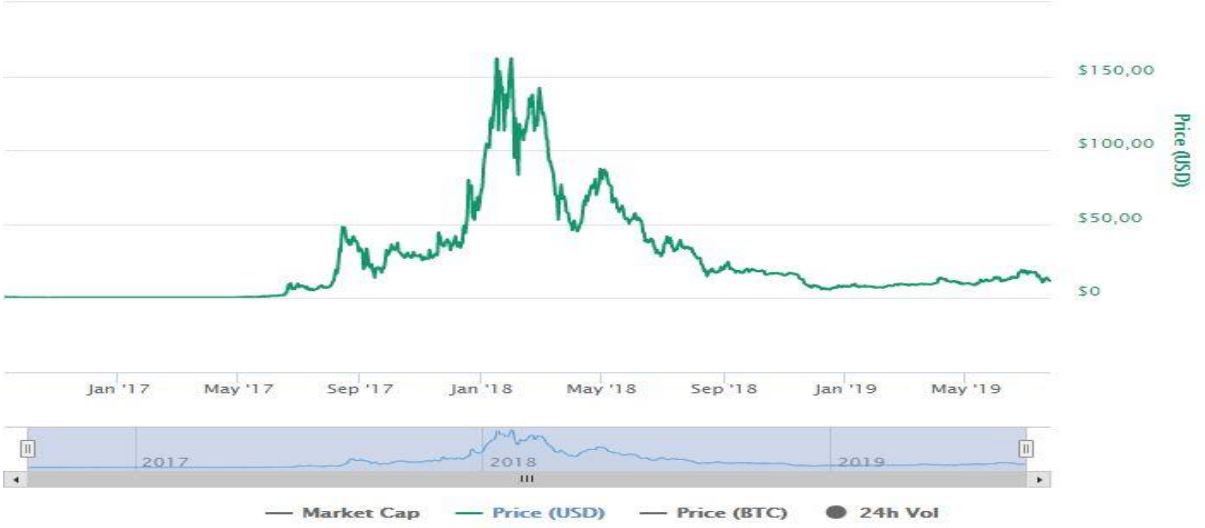
NEO ili otprije poznat i kao Antshares je u posljednje dvije godine zabilježio ogroman porast vrijednosti. NEO je također vrsta platforme koja omogućuje pametne ugovore koji su pisani u programskim jezicima Java i C#. NEO platformu još nazivaju i kineskim Ethereumom upravo zbog toga što je ideja razvijena u Kini ali je i zabilježila ogroman napredak [27].

Rogina [27] navodi kako je NEO platforma za pametne ugovore pokrenuta od tvrtke koja se bavi istraživanjem i razvojem blockchain tehnologija naziva Onechain. Na čelu tvrtke Onechain su Da Hongfei i Erik Zhang. Originalni naziv Antshares je u kolovozu 2017.godine primijenjen u današnji naziv NEO. Kada se govori o transakcijama, NEO podržava 1000 transakcija/sekundi. Što se tiče cijena vezanih uz NEO platformu, plaćanje se izvršava GAS-om koji predstavlja oblik plaćanja provizije. Trenutna cijena jednog GAS-a je 36 dolara, kako je za objavu pametnog ugovora potrebno platiti 500 GAS-a dolazimo do cijene objave pametnog ugovora koja iznosi 18 tisuća dolara.

Slika 11. NEO logo [31]

Slika 11. prikazuje logo platforme NEO. Dok sljedeća slika 12. prikazuje kretanje 1 NEO vrijednosti koja je izražena u američkim dolarima. Slika 12. prikazuje kretanje vrijednosti u

razdoblju od druge polovice 2016.godine do danas



Slika 12. Kretanje NEO vrijednosti u razdoblju od kraja 2016.godine do danas [28]

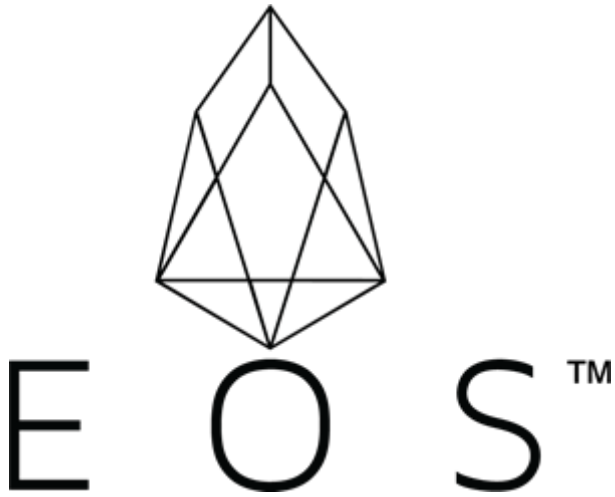
4.1.3.EOS

Posljednja blockchain platforma o kojoj će biti napisano nešto više je EO platforma. To je platforma koja je još uvijek u razvoju. Velika su očekivanja od EOS platforme zbog toga što će podržavati programski jezik C++. Slična je Ethereum platformi ali ipak postoje određene razlike. Kao i ostale platforme EOS platforma će omogućiti podržavanje decentraliziranih aplikacija [23].

Kod blockchain sustava kod ugovora se kompajlira u bytecode pa je potreban izvršitelj za izvršenje pametnih ugovora. EOS je zapravo izvršitelj koji izvršava ugovore u blockchain sustavu, navodi se kako EOS Virtual Machine zapravo ima određene odgovornosti [29]:

1. Odgovoran je za učitavanje i razdvajanje kompajliranog bytecode-a pametnog ugovora
2. Odgovoran je za dodjeljivanje resursa bytecode-u
3. Odgovoran je za davanje ili pridruživanje API-ja izvan baze podataka bytecode-a pametnih ugovora
4. Odgovoran je za izvršavanje bytecode-a prilikom izračuna rezultata pametnih ugovora

Slika 13. prikazuje logo EOS platforme.



Slika 13. EOS logo [32]

4.2.Područja primjene pametnih ugovora

U ovom poglavlju biti će navedeno nekoliko primjera u kojima se koristi pametni ugovor. Primjeri će biti vezani uz različita područja primjene. Primjeri su preuzeti sa različitih internetskih stranica i služe kako bi neka osoba što bolje razumjela pojam pametnog ugovora. U sljedećim potpoglavljima definirati će se nekoliko različitih vrsta primjera vezanih uz pametne ugovore.

4.2.1.Primjer iz turizma

Navedeni primjer odnosi se na primjer iznajmljivanja apartmana. Zamislimo da imamo dvije strane, odnosno osobu X koja želi iznajmiti apartman i osobu Y koja iznajmljuje apartman. Situacija je takva da osoba Y mora platiti svotu koja je potrebna za iznajmljivanje apartmana, samim time osoba Y dobiva digitalni račun koji je predmet ugovora između navedene dvije strane. Nakon toga osoba X mora osobi Y poslati digitalni ključ, ako ga ne dostavi na vrijeme pametnim ugovorom se odmah vraćaju sredstva koja je osoba Y uplatila. Pametni ugovor pamti definirane uvjete i na temelju toga se odvija daljnje izvršavanje. Na taj način se osigurava sigurnost ili nemogućnost prijevare. Ukoliko su zadovoljeni svi uvjeti tada se isplaćuju sva sredstva definirana u ugovoru. Odnosno, ukoliko osoba X pošalje ključ biti će mu omogućena usluga iznajmljivanja apartmana [30].

4.2.2.Primjer bankarskih usluga

Sljedeći primjer odnosi se na uslugu bankarstva, odnosno također kao i kod prethodnog primjera postoje dvije strane ugovora. U ovom slučaju to je prijenos novčanih sredstava sa jednog računa na drugi račun. Računi predstavljaju dvije strane ugovora. Kako pametni

ugovori ne zahtijevaju potrebu posrednika tako se transakcije prijenosa novčanih sredstava sa jednog računa na drugi odvijaju brzo, efikasno i bez dodatnih troškova. Također, smanjen je rizik od prijevare zbog toga što se transakcija brzo završava i novčana sredstva se isplaćuju. Glavna razlika korištenja pametnog ugovora očituje se u tome što je kod običnog prijenosa novca s jednog računa na drugi potrebno platiti neku naknadu i pričekati nekoliko dana da se transakcija obradi [23].

4.2.3. Ostali primjeri

Upotreba pametnih ugovora je široko rasprostranjena, odnosno primjenjuju se u različitim područjima. Tako se npr. pametni ugovori također mogu upotrebljavati kod automatizacije različitih sustava kao što je sustav glasanja pri čemu se dobiva na sigurnosti i povjerljivosti prilikom glasanja. Također primjena može biti i u različitim istraživanjima unutar različitih područja zbog toga što se na taj način zaštićuju osobni podaci ispitanika i dobiva se na sigurnosti i anonimnosti anketiranih korisnika. I još jedan od primjera korištenja je u logistici koje omogućava praćenje robe i/ili proizvoda koji smo naručili/isporučili i na taj način lakše mogli obavljati sljedeće aktivnosti. Tako se navedena roba i/ili proizvodi preko blockchain tehnologije prate automatski te nam je omogućeno izvještavanje.

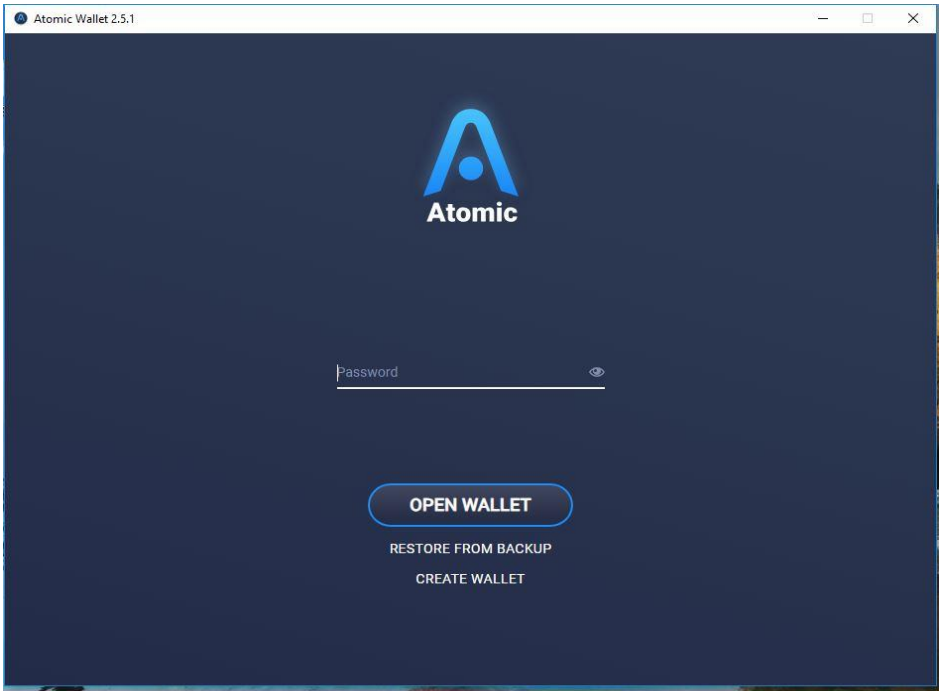
5. Praktični primjer

Nakon obrađenih teorijskih dijelova teme, te prikazanih pojedinosti vezanih uz teorijsku obradu teme. Sada prelazimo na praktični dio rada, u kojem će biti prikazan proces izrade pametnog ugovora. Kako bi se proces izrade ugovora prikazao što bolje, za pripremu se koristio video materijal (eng. *Tutorial*) koji je vezan uz proces izrade pametnog ugovora. Prilikom procesa kreiranja pametnog ugovora prvo je potrebno odabrati platformu unutar koje će se sam pametni ugovor zapravo i kreirati. U ovom slučaju odabrana je platforma Ethereum o kojoj je u prethodnim poglavljima navedeno nešto više. Ethereum platforma je odabrana upravo radi njene trenutno najveće primjene u praksi i zbog širokih mogućnosti vezanih uz alate programiranja.

Nakon toga, drugi korak prilikom kreiranja pametnog ugovora je korak u kojem je potrebno instalirati novčanik (eng. *Wallet*) pomoću kojeg nam je olakšan uvid u transakcije. Prilikom odabira novčanika koristila se web stranica na kojoj su prema određenim performansama navedeni najbolji novčanici koji se mogu koristiti uz Ethereum platformu. Na stranici je navedeno deset najboljih novčanika, neki od njih su desktop novčanici, neki mobilne verzije, a neki hardware verzije. U nastavku će biti navedeni ti novčanici [33]:

1) Ledger Nano S (Hardware Wallet)
2) Trezor (Hardware Wallet)
3) Atomic Wallet (Desktop and Mobile)
4) Exodus (Desktop Wallet + mobile)
5) Jaxx (Mobile Wallet)
6) Mist (Desktop Wallet)
7) MetaMask (Desktop Wallet)
8) MyEtherWallet (Web Wallet)
9) Coinbase (Web Wallet)
10) KeepKey (Hardware Wallet)

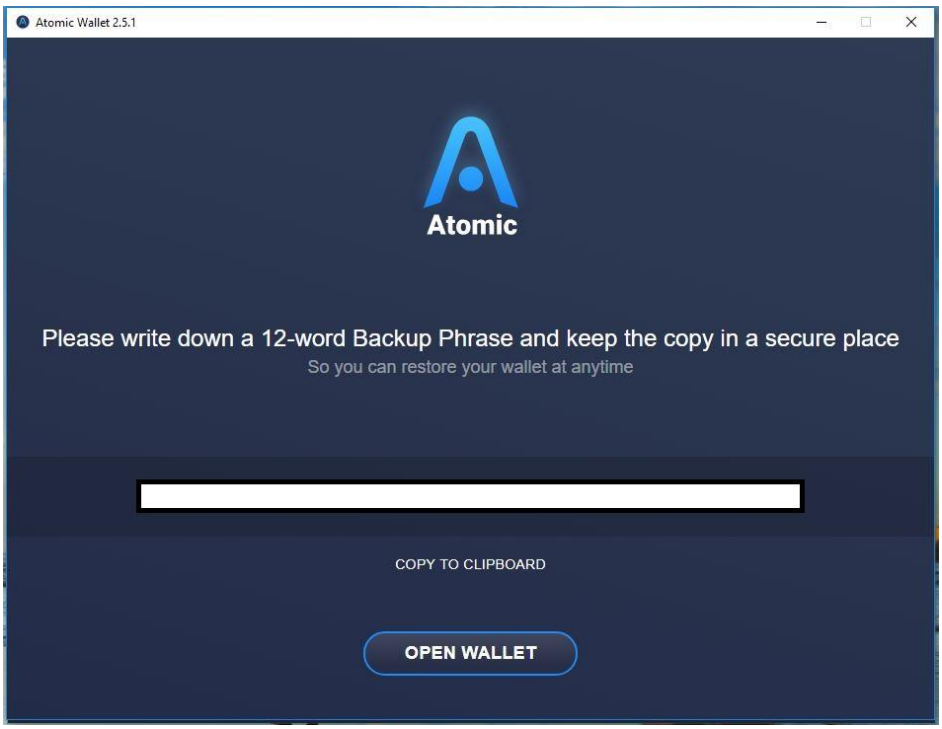
U ovom slučaju odlučeno je kako će se odabrati novčanik pod rednim brojem 3, odnosno Atomic Wallet. Taj je novčanik odabran zbog toga što nudi desktop verziju odnosno može se njime raditi na prijenosnom računalu, ali i kao dodatak mobilnu verziju te se transakcije mogu uvijek pratiti preko mobitela. Desktop verzija novčanika je preuzeta sa službene stranice Atomic Wallet, te ju možete preuzeti na poveznici: https://atomicwallet.io/ethereum-wallet. U našem slučaju preuzeta je verzija za operacijski sustav windows. Osim toga, moguće je preuzeti i verzije: macOS, Ubuntu, Debian, Fedora. U nastavku će biti prikazan rad u navedenom novčaniku.



Slika 14. Početni zaslon Atomic Wallet-a (Desktop screenshot)

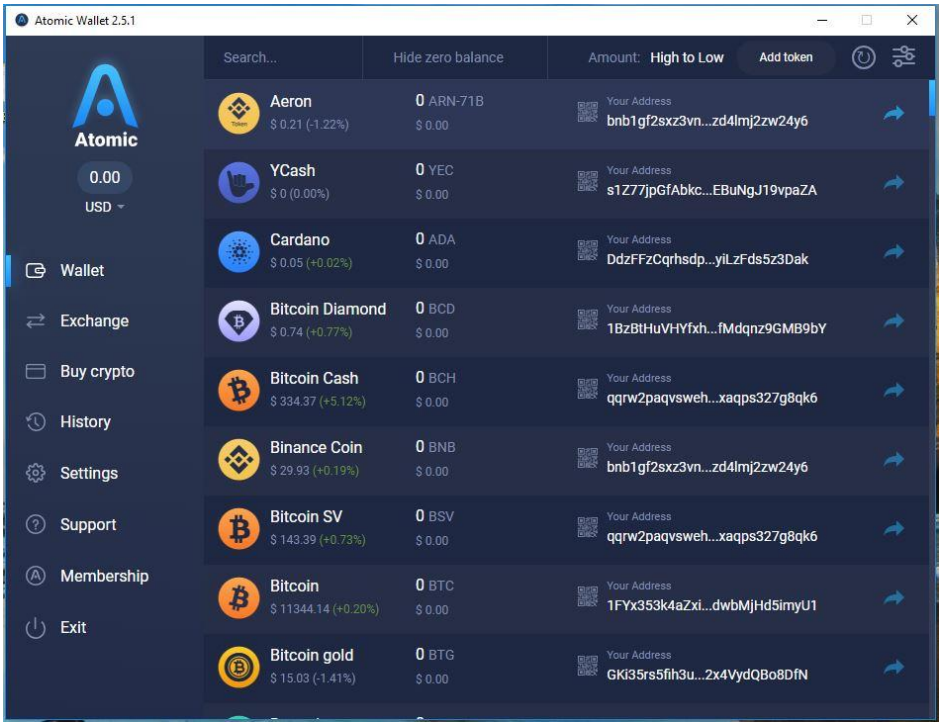
Slika 14. prikazuje početni zaslon Atomic Walleta, odnosno prvi zaslon koji se otvara nakon pokretanja desktop verzije. U ovom slučaju potrebno je unijeti lozinku i otvoriti novčanik

ukoliko imamo kreiran profil već. Osim toga nudi se i mogućnost kreiranja novog novčanika opcijom create wallet. Sada se unosi lozinka i otvara se novčanik.



Slika 15. Atomic Wallet fraza (Desktop screenshot)

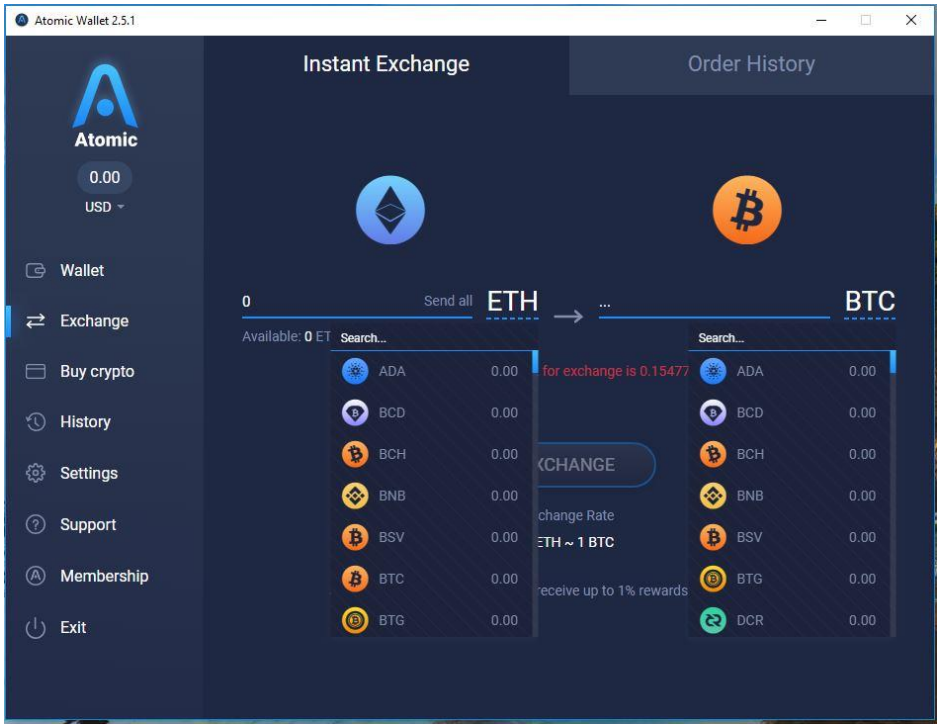
Slika 15. prikazuje backup frazu koja se sastoji od 12 riječi. Fraza se nalazi unutar bijele ispune na slici gdje je u ovom slučaju zaštićena zbog privatnosti. Frazu je potrebno pohraniti negdje kako bi u slučaju gubitka lozinke mogli vratiti svoj novčanik.



Slika 16. Glavni zaslon Atomic Wallet-a (Desktop screenshot)

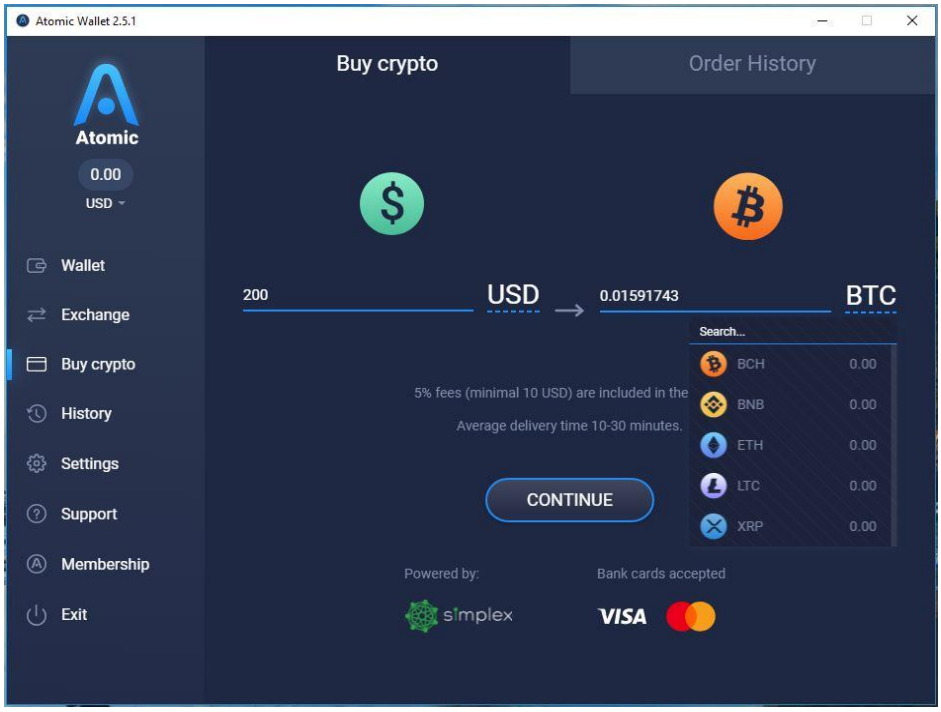
Slika 16. prikazuje glavni zaslon Atomic Wallet-a u kojem možemo vidjeti popis kriptovaluta kao i njihovu vrijednost u dolarima te postotak promjene vrijednosti. Osim toga Atomic Wallet nudi mogućnosti koje možemo vidjeti na lijevom dijelu slike. To su npr. razmjena različitih valuta, kupovanje različitih kriptovaluta, pregled povijesti novčanika, podešavanje različitih opcija, podrška, itd. Također u gornjem dijelu slike možemo vidjeti kako se nude i mogućnosti pretraživanja kriptovaluta po nazivu, mogućnost sakrivanja, mogućnost sortiranja bilo od najveće do najmanje ili obrnuto, filtriranje kriptovaluta, itd.

Sljedeća slika, odnosno slika 17. prikazuje mogućnost mjenjačnice tj. razmjene različitih kriptovaluta. Na slici možemo vidjeti kako možemo odabrati koju kriptovalutu želimo razmijeniti u koju kriptovalutu. Odabire se kriptovaluta koju mi dajemo i unosi se vrijednost koliko je želimo razmijeniti, te se zatim odabire u koju kriptovalutu ju želimo razmijeniti te novčanik sam računa koja je vrijednost koju možemo dobiti za uloženo i samo je još potrebno odabrati opciju exchange.



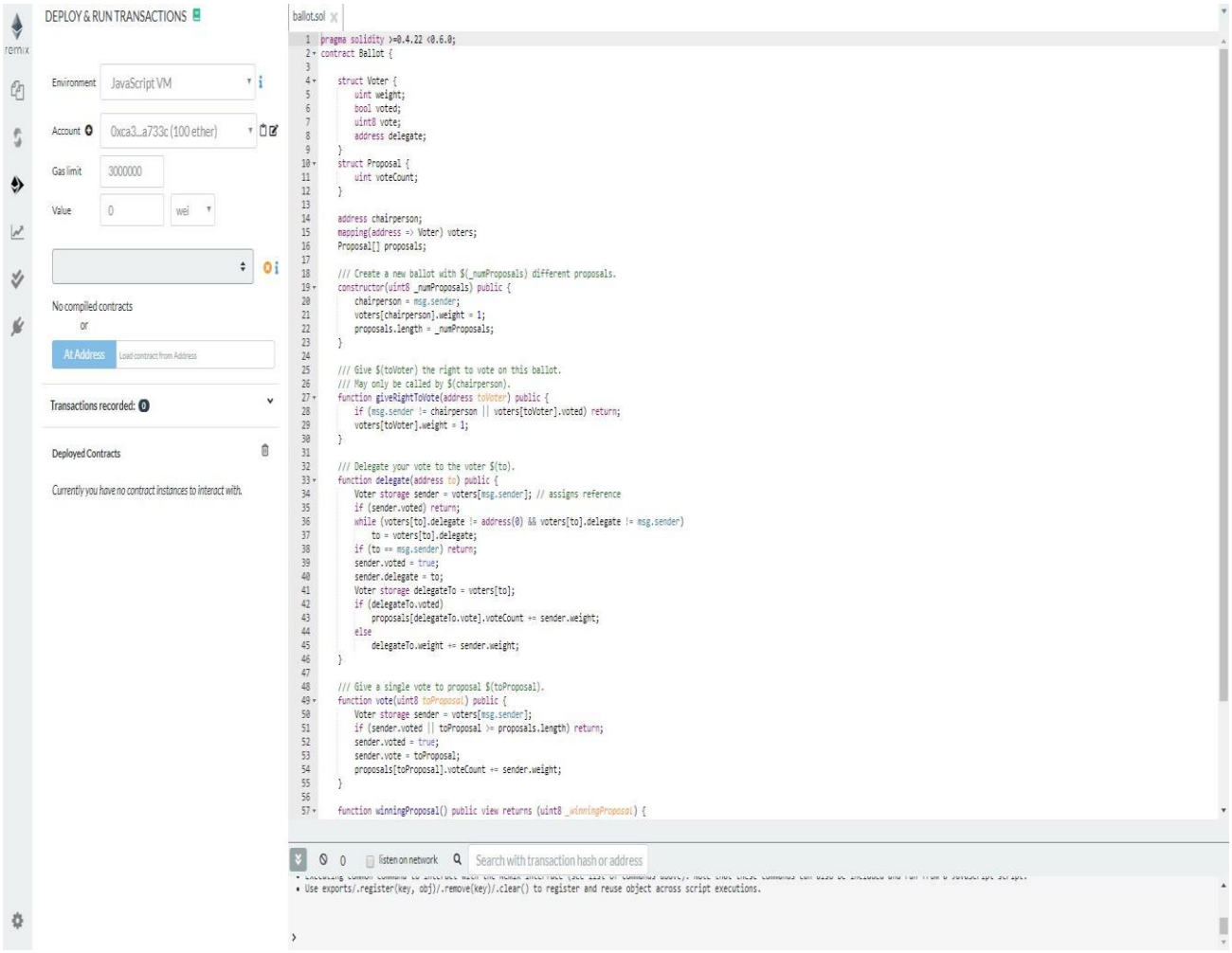
Slika 17. Opcija mjenjačnice (Desktop screenshot)

Slika 18. prikazuje mogućnost kupnje različitih kriptovaluta. Kao što možemo vidjeti na slici 18. unosi se vrijednost novca koji smo spremni uložiti. U ovom slučaju moguće je unijeti vrijednost u dolarima ili eurima, dok je izbor kriptovaluta malo širi. Također kriptovalute koje možemo kupiti su prikazane na slici. Nakon odabira kriptovalute i unosa vrijednosti koju plaćamo otvara se prozor unutar kojeg se unose podaci za plaćanje i način plaćanja.



Slika 18. Opcija kupnje kriptovalute (Desktop screenshot)

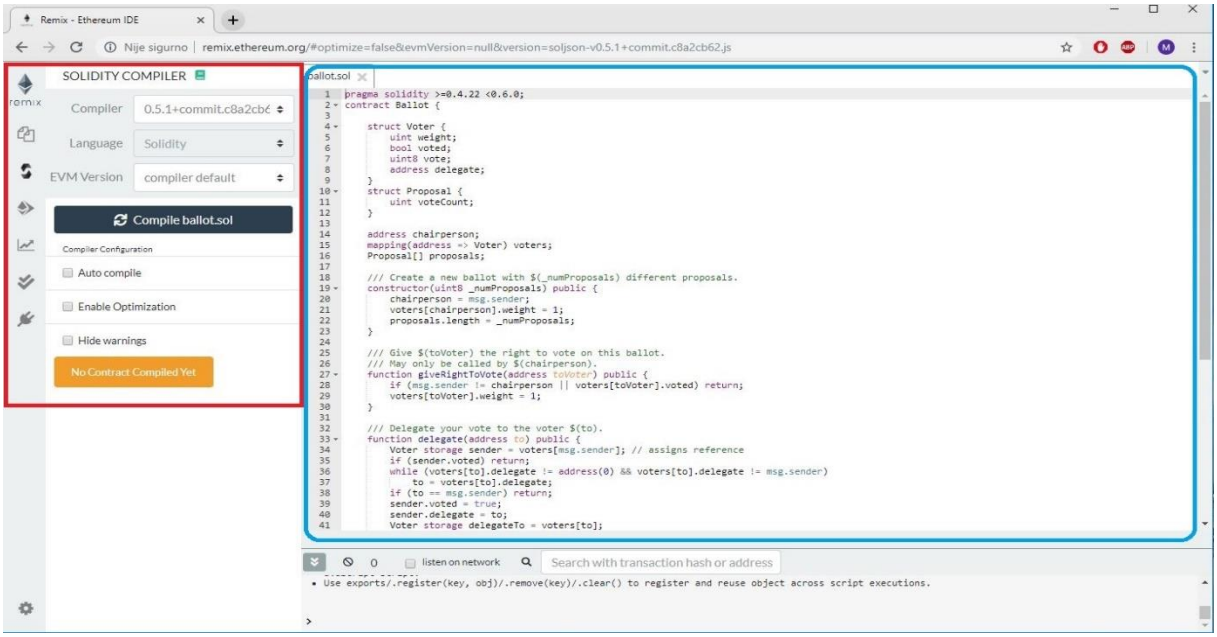
Na nekoliko prethodnih stranica prikazan je Atomic Wallet kao novčanik koji se koristi za Ethereum platformu. Prikazano je nekoliko glavnih mogućnosti koje se nude unutar novčanika, te će se sada preći na drugi dio praktičnog rada. Unutar drugog dijela praktičnog rada biti će prikazan proces izrade pametnog ugovora. Odnosno, korak u kojem se programira ugovor. Za programiranje pametnog ugovora koristi se web oblik IDE okruženja pod nazivom „Remix“ koji je moguće pronaći na poveznici: https://remix.ethereum.org/. To je web verzija koja nudi sve potrebne alate koji omogućuju pisanje ugovora programeru. Osim samog dijel za pisanje i uređivanje programskog koda, također se nudi i opcija kompajliranja i interpretiranja, te ispravljanja pogrešaka.



Slika 19. Primjer ugovora napisanog u Remix IDE (http://remix.ethereum.org/)

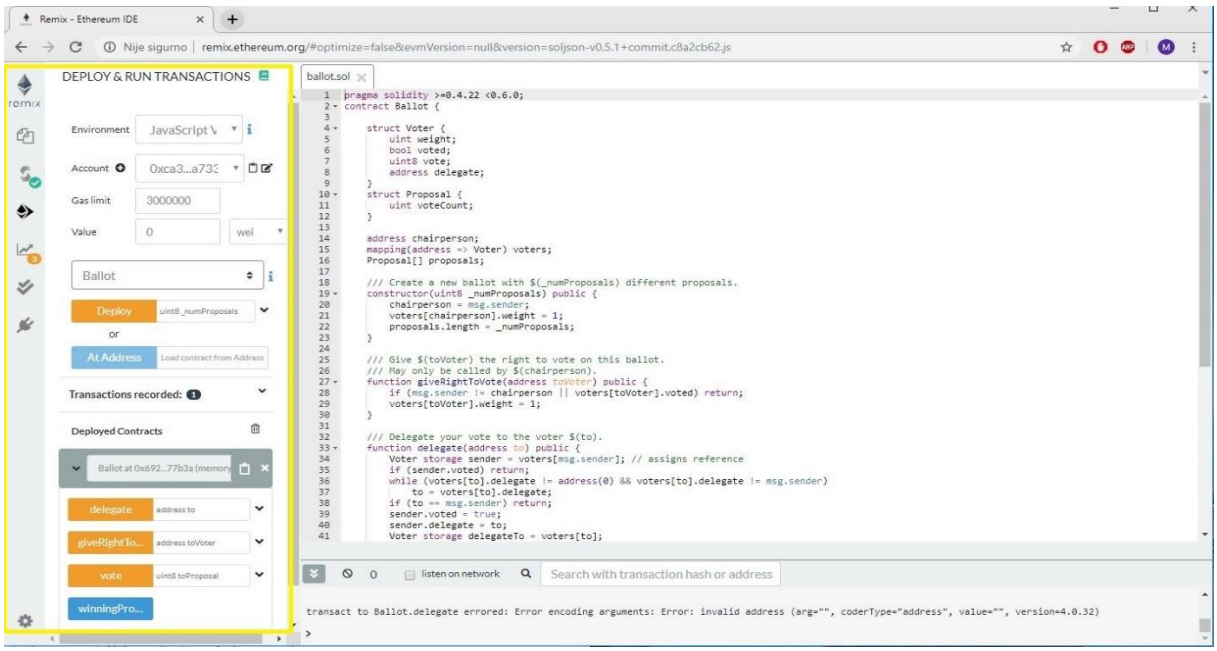
Slika 19. prikazuje primjer programskog koda napisanog u Remix IDE. U nastavku će se kreirati novi file unutar kojeg će se napisati programski kod koji će predstavljati primjer pametnog ugovora od strane autora ovog rada.

Odabire se opcija kreiranja novog file-a i daje se naziv, nakon toga otvara se novi prozor unutar kojeg se trenutno ne nalazi niti jedna linija koda. Sada se prelazi na korak pisanja programskog koda koji će biti prikazan na sljedećoj slici.



Slika 20. Rad u Remix IDE 1 (http://remix.ethereum.org/)

Na slici 20. možemo unutar plavog okvira vidjeti programski kod ugovora, odnosno napisan primjer pametnog ugovora. Dok na lijevoj strani slike, odnosno unutar crvenog okvira možemo vidjeti mogućnost podešavanja kompajliranja file-a te dodatne dijelove vezane uz kompajliranje. Također, možemo ukoliko želimo označiti opciju auto kompajliranja kako bi na okruženje automatski pokretalo kompajliranje koda. Osim auto kompajliranja, također je moguće uključiti i opciju sakrivanja upozorenja. To su dijelovi koda koji rade, ali nam se ipak daje određen oblik upozorenja. Konkretno ovaj primjer je preuzet sa weba, autor rada ga ne koristi u radu nego je kod čisto reprezentativne primjene.



Slika 21. Rad u Remix IDE 2 (http://remix.ethereum.org/)

Na slici 21. desni dio slike je identičan kao i na prethodnoj slici. Dok na lijevom dijelu slike, odnosno unutar žutog okvira možemo vidjeti deploy & run opcije unutar remix okruženja. Nakon odabira deploy opcije ugovora, u podnožju žutog okvira možemo vidjeti koji su ugovori deploy-ani odnosno razvijeni. Prilikom toga možemo otvoriti ugovor i unutar tog dijela vidimo kako naš ugovor izgleda odnosno od kojih se dijelova sastoji te što se u kojem dijelu nalazi.

Kao zadnji dio praktičnog rada, u nastavku će biti prikazan jednostavan primjer pametnog ugovora koji će biti napisan od strane autora ovog rada. Također, svaki dio koda biti će interpretiran kako bi se što bolje prikazala funkcionalnost ugovora te svi njegovi dijelovi.

Marko1.sol

```
pragma solidity ^0.5.0;

contract MarkoZavrsniRad {

    string public valuta = 'markovaluta';

    string public oznakaValute = 'mv';

    mapping (address => uint) iznosi;

    event TransferNovca(address posiljatelj, address primatelj, uint256 iznos);

    constructor public(){

        iznosi[msg.sender] = 5000;

    }

    function slanjeNovca (address prima, uint iznosNovca) public returns (bool dovoljan){

        if (iznosi[msg.sender] < iznosNovca) return false;

        iznosi[msg.sender] -= iznosNovca;

        iznosi[prima] += iznosNovca;

        emit TransferNovca (msg.sender, prima, iznosNovca);

        return true;

    }

    function provjeriStanje(address adresa) public view returns (uint) {

        returns iznosi[adresa];

    }

}
```

Slika 22. Pametni ugovor napisan od strane autora rada (http://remix.ethereum.org/)

Slika 22. prikazuje primjer pametnog ugovora odnosno programski kod ugovora napisan od strane autora ovog završnog rada. Kako bi se lakše i bolje objasnili pojedini dijelovi koda na slici je kod podijeljen na nekoliko dijelova od kojih se svaki dio nalazi unutar određeno obojanog okvira. Tako se unutar crvenog okvira nalazi zapravo sami pametni ugovor koji nosi naziv kao što možemo vidjeti „MarkoZavrsniRad“. Nakon crvenog okvira možemo vidjeti prvi dio ugovora koji se nalazi u ljubičastom okviru, to je dio unutar kojeg je definiran naziv valute kao i simbol navedene valute. Nadalje, sljedeća linija unutar ljubičastog okvira definira adrese sa iznosima, te zadnja linija unutar tog okvira predstavlja deklaraciju jednog događaja, koji zapravo ne radi ništa dok ga se god ne pokrene.

Sljedeći dio koda nalazi se unutar zelenog okvira, tamo se nalazi konstruktor koji se pokreće automatski nakon što je ugovor kreiran. Unutar konstruktora se nalazi linija koja definira iznos kreatora ugovora. Nakon toga slijedi žuti okvir, koji sadrži funkciju naziva „slanjeNovca“, funkcija prvo provjerava transfer novca odnosno vraća izraz false ukoliko se ne zadovoljava iznosi definirani u konstruktoru. Također unutar tog dijela možemo vidjeti i kako se dešavaju promjene vezane uz sredstva, odnosno određena smanjenja kao i povećanja. Te se u zadnjem dijelu tog okvira poziva prethodno kreirani događaj „TransferNovca“ sa svojim parametrima, te funkcija vraća vrijednost true. Te na kraju, plavi okvir sadrži funkciju provjere stanja koja vraća broj, odnosno iznos kojim raspolaže određena adresa. Slika je zapravo screenshot unutar Remix IDE, dok je kod napisan od strane autora ovog rada. Na kraju bih još samo htio napomenuti kako su sve slike u praktičnom dijelu rada slike ekrana, te se kao takve ne navode u popisu literature.

Na kraju još možemo reći i ponešto o cijenama odnosno troškovima izvršavanja ugovora tj. objave ugovora na ethereum platformu. Troškovi se dobivaju kao umnožak broja potrebnih GAS-a sa cijenom GAS-a. GAS na ethereum platformi predstavlja osnovnu jedinicu kao oblik plaćanja pametnih ugovora na ethereum platformi. Ovisno o tome kako brzo želimo da naš ugovor bude u funkciji takvu ćemo i cijenu GAS-a platiti, odnosno ako želimo da se što prije izvrši onda plaćamo skuplju cijenu GAS-a i obrnuto. Znači, prilikom računjanja troškova objave pametnog ugovora u obzir je potrebno uzeti GAS cijenu, GAS limit odnosno broj potrebnih GAS jedinica, te tržišnu vrijednost ethereum-a. Međutim, to nisu svi troškovi objave pametnog ugovora. Kada bi uzeli u obzir sve troškove i sumirali ih došli bi do ukupnog troška objave pametnog ugovora. Shodno tome, jednostavni pametni ugovor stoji oko 7 500 $, dok neki od njih mogu koštati čak i do 45 000 $ [34].

6. Zaključak

Na kraju nakon svih dosad definiranih poglavlja, potrebno je još donijeti i određeni zaključak kako bi se ovaj završni rad zaokružio na jedan cjelovit način. Nakon dosada definiranih pojmova možemo zaključiti kako je zapravo blockchain tehnologija široko primijenjena, te su njene mogućnosti velike i često primjenjivane. Na početku rada je definiran sama blockchain tehnologija, kao i njene vrste, struktura, te algoritmi koji se koriste. Tako razlikujemo nekoliko algoritama, to su: proof-of-work, proof-of-stake, te delegirani proof-of-stake. O svakom od njih je rečeno nešto više te su donesene kratke usporedbe navedenih algoritama. Nakon toga, uslijedio je drugi dio rada odnosno dio koji se odnosi na pametne ugovore te je taj dio zapravo obrađen i teoretski i praktično. Prvo su definirani osnovni pojmovi, te platforme pomoću kojih je moguće zapravo i kreirati pametni ugovor. To su platforme: ethereum virtual machine, NEO, te EOS. O njima je navedeno nešto više, te se zatim prešlo na praktični dio rada.

Praktični dio rada sastojao se od dva dijela, u prvom dijelu bilo je potrebno odabrati koja će se platforma koristiti za izradu pametnog ugovora te je tako odabrana ethereum platforma. Nakon toga preuzeta je verzija novčanika pod nazivom Atomic Wallet, koji nudi različite mogućnosti vezane uz kriptovalute. Tako smo vidjeli mogućnost razmjene različitih vrsta kriptovaluti, zatim mogućnost kupnje neke kriptovalute, itd. Nakon toga prikazano je i pojašnjeno Remix IDE kako bi se što bolje razumio način rada unutar njega. Te je kao sami završetak prikazan primjer koda napisan od strane autora ovog rada, sa svojim objašnjenim pripadnim dijelovima.

Na kraju kao autor ovog rada, mogu reći kako mi se tema svidjela i kako je dosta zanimljiva te ukoliko želimo možemo puno čitati i naučiti o njoj. Smatram da je blockchain tehnologija zapravo vrlo značajna te u potpunosti razumijem sve veću popularnost njezine upotrebe.

7. Popis literature

[1] O. Godbole, „Bitcoin (BTC)“, 2019. [Na internetu]. Dostupno: https://www.coindesk.com/price/bitcoin [pristupano 28.6.2019.].

[2] Draw. Draw.io (online verzija) (2019) [Na internetu]. Dostupno: https://www.draw.io/ [pristupano 28.6.2019.]

[3] A. Rosic, „What is Blockchain Tehnology? A Step-by-Step Guide for Beginners“, 2019. [Na internetu]. Dostupno: https://blockgeeks.com/guides/what-is-blockchain-technology/ [pristupano 29.6.2019.].

[4] D. Arunović, „Što je u stvari blockchain i kako radi?“, 2018. [Na internetu]. Dostupno: https://www.bug.hr/tehnologije/sto-je-u-stvari-blockchain-i-kako-radi-3011 [pristupano 29.6.2019.].

[5] B. P., „Uvod u blockchain tehnologiju“, 2017. [Na internetu]. Dostupno: https://pcchip.hr/ostalo/tech/uvod-u-blockchain-tehnologiju/ [pristupano 30.6.2019.].

[6] D. Hozjan, „Blockchain“, 2017. [Na internetu]. Dostupno: https://repozitorij.pmf.unizg.hr/islandora/object/pmf%3A779/datastream/PDF/view [pristupano: 30.6.2019.].

[7] S. Živković, „Blockchain tehnologija“, 2018. [Na internetu]. Dostupno: https://zir.nsk.hr/islandora/object/infri:289/preview [pristupano 31.6.2019.].

[8] PasswordsGenerator. SHA256 Hash Generator. (2018). [Na internetu]. Dostupno: https://passwordsgenerator.net/sha256-hash-generator/ [pristupano: 31.6.2019.].

[9] Prasanna, „What is the Blockchain dana structure?“, 2018. [Na internetu]. Dostupno: https://cryptoticker.io/en/blockchain-data-structure/ [pristupano: 1.7.2019.].

[10] R. Chan, „Blockchain Dana Structure“, 2018. [Na internetu]. Dostupno: https://www.linkedin.com/pulse/blockchain-data-structure-ronald-chan/ [pristupano: 1.7.2019.].

[11] Škola koda (bez dat.) Binarno hash stablo [Na internetu]. Dostupno: https://skolakoda.org/binarno-hash-stablo [pristupano: 2.7.2019.].

[12] A. Škegro, „Primjena blockchain tehnologije u prehrambenoj industriji“, 2019. [Na internetu]. Dostupno: https://zir.nsk.hr/islandora/object/algebra%3A233/datastream/PDF/view [pristupano: 2.7.2019.].

[13] A. Tar, „Proof-of-Work, Explained“, 2018. [Na internetu]. Dostupno: https://cointelegraph.com/explained/proof-of-work-explained [pristupano: 4.7.2019.].

[14] A. Huskanović, „Proof of Work – What it I san How Does it Work?“, 2018. [Na internetu]. Dostupno: https://www.asynclabs.co/blog/proof-of-work-what-it-is-and-how-does-it-work/ [pristupano: 4.7.2019.].

[15] Bruno, „Po čemu se razlikuju Proof of Work, Proof of Stake, i Delegated PoS metoda?“, 2018. [Na internetu]. Dostupno: https://bitfalls.com/hr/2018/04/24/whats-the-difference-between-proof-of-work-pow-proof-of-stake-pos-and-delegated-pos/ [pristupano: 4.7.2019.].
[16] Lisk (bez dat.) Proof of Stake [Na internetu]. Dostupno: https://lisk.io/academy/blockchain-basics/how-does-blockchain-work/proof-of-stake [pristupano: 4.7.2019.].
[17] K. Khullar, „Implementing Proof of Stake Part – 1“, 2019. [Na internetu]. Dostupno: https://medium.com/coinmonks/implementing-proof-of-stake-e26fa5fb8716 [pristupano: 4.7.2019.].
[18] Lisk (bez dat.) Delegated Proof of Stake [Na internetu]. Dostupno: https://lisk.io/academy/blockchain-basics/how-does-blockchain-work/delegated-proof-of-stake [pristupano: 4.7.2019.].
[19] Binance Academy, „Delegated Proof of Stake Explainded“, 2018. [Na internetu]. Dostupno: https://www.binance.vision/blockchain/delegated-proof-of-stake-explained [pristupano: 4.7.2019.].
[20] A. Rosic, „Proof of Work vs Proof of Stake: Basic Mining Guide“, 2017. [Na internetu]. Dostupno: https://blockgeeks.com/guides/proof-of-work-vs-proof-of-stake/ [pristupano: 4.7.2019.].
[21] S. Nosikov, „What Are Smart Contracts?“, (bez dat.). [Na internetu]. Dostupno: https://www.cryptoninjas.net/what-are-smart-contracts/ [pristupano 15.7.2019.].
[22] I. Voras, „Što su pametni ugovori – uvod“, 2018. [Na internetu]. Dostupno: https://ubik.hr/2018/03/26/sto-su-pametni-ugovori-uvod/ [pristupano: 15.7.2019.].
[23] Z. Bartolović, „Pametni ugovori“, 2018. [Na internetu]. Dostupno: https://zir.nsk.hr/islandora/object/veleri%3A1594/datastream/PDF/view [pristupano: 16.7.2019.].
[24] A. Bisade, „Ehereum Virtual Machine Explained“, 2018. [Na internetu]. Dostupno: https://www.mycryptopedia.com/ethereum-virtual-machine-explained/ [pristupano: 18.7.2019.].
[25] R. King, „Whai is Ethereum? Understanding How Does Ethereum Work“, 2019. [Na internetu]. Dostupno: https://www.bitdegree.org/tutorials/what-is-ethereum/ [pristupano: 18.7.2019.].
[26] S. Joudrey, „Guide to Ethereum virtual Machines“, 2019. [Na internetu]. Dostupno: https://hedgetrade.com/guide-to-ethereum-virtual-machines/ [pristupano: 18.7.2019.].
[27] N. Rogina, „NEO (Antshares) – kineski Ethereum?“, 2018. [Na internetu]. Dostupno: https://www.kriptovaluta.hr/altcoin/neo-antshares-kineski-ethereum/ [pristupano: 19.7.2019.].
[28] CoinMarketCap (bez dat.) NEO [Na internetu]. Dostupno: https://coinmarketcap.com/currencies/neo/ [pristupano: 19.7.2019.].

[29] EOSForce, „The Blockchain Industry Will Have the First Tehnical Standard Debate: EOS-VM Take the Lea din Virtual Machine Race“, 2019. [Na internetu]. Dostupno: https://medium.com/@eosforce/the-blockchain-industry-will-have-the-first-technical-standard-debate-eos-vm-take-the-lead-in-8619c5b84601 [pristupano: 20.7.2019.].
[30] Škola koda (bez dat.) Pametni ugovori [Na internetu]. Dostupno: https://seeklogo.com/vector-logo/322892/eos [pristupano: 20.7.2019.].
[31] GitHub (bez dat.) NEO Virtual Machine [Na internetu]. Dostupno: https://github.com/neo-project/neo-vm [pristupano: 20.7.2019.].
[32] Steemit, „EOS whitepaper walk-trought: Scripts and Virtual Machines“, 2018. [Na internetu]. Dostupno: https://steemit.com/eos/@bluabaleno/eos-whitepaper-walk-through-scripts-and-virtual-machines [pristupano: 20.7.2019.].
[33] H. Argawal, „The Top 10 Best Ethereum Wallets (2019 Edition)“, 2019. [Na internetu]. Dostupno: https://coinsutra.com/best-etherum-wallets/ [pristupano: 12.8.2019.].
[34] R. Shome, „How Much Does It Cost To Deploy A Smart Contract on Ethereum?“, 2019. [Na internetu]. Dostupno: https://www.btcwires.com/round-the-block/how-much-does-it-cost-to-deploy-a-smart-contract-on-ethereum/ [pristupano: 19.8.2019.].

8. Popis slika

9. Popis tablica