Blockchain - pametni ugovori

Prološčić, Nikola

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni diplomski studij matematike, smjer: Matematika i računarstvo

Nikola Prološčić

Blockchain - Pametni ugovori

Diplomski rad

Osijek, 2022.

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni diplomski studij matematike, smjer: Matematika i računarstvo

Nikola Prološčić

Blockchain - Pametni ugovori

Diplomski rad

Mentor: prof. dr. sc. Ivan Matić

Osijek, 2022.

Sadržaj

1 Uvod **1**

2 Blockchain **2**
2.1 Od jednostavnog knjigovodstva do blockchaina . . . 2
2.2 Bitcoin . . . 3
2.2.1 Dokaz o radu . . . 3
2.2.2 Merkleovo stablo . . . 5
2.2.3 Bitcoin kao sustav tranzicije stanja . . . 6
2.2.4 Bitcoin skripte . . . 7

3 Ethereum **10**
3.1 Povijesni razvoj Ethereuma . . . 10
3.2 Ethereum računi . . . 10
3.3 Transakcije . . . 11
3.4 Ethereum kao tranzicija stanja . . . 13
3.5 Struktura Ethereum mreže . . . 13
3.6 Merkleovo Patricia stablo . . . 14

4 Pametni ugovori **16**
4.1 Ethereum Virtual Machine . . . 16
4.2 Programski jezik Solidity . . . 18
4.3 TestNet . . . 20
4.4 Bibilioteke . . . 20
4.4.1 Kreiranje ERC-20 tokena . . . 20
4.5 ABI specifikacija ugovora . . . 23
4.5.1 Interakcija s pametnim ugovorom . . . 23

5 Primjena blockchain tehnologije i pametnih ugovora **27**
5.1 Financijski sektor . . . 28
5.1.1 CBDC i prekogranična plaćanja . . . 28
5.1.2 Financiranje trgovine . . . 28
5.1.3 Decentralizirane financije . . . 29
5.2 Blockchain u zdravstvenom sustavu . . . 31
5.3 Upravljanje korisničkim identitetima . . . 32
5.4 Upravljanje lancem opskrbe (SCM) . . . 33
5.4.1 Upravljanje rizikom opskrbnog lanca (SCRM) . . . 34
5.4.2 Pomorsko osiguranje . . . 34
5.5 ERP i CRM sustavi . . . 35
5.6 Blockchain as a service (BaaS) . . . 36
5.7 Ostale primjene . . . 36

6 Zaključak **37**

1 Uvod

Transakcije između dviju stranaka u trenutnom financijskom sustavu najčešće se odvijaju u centraliziranom obliku uz obvezno prisustvo treće stranke koja će jamčiti sigurnost transakcije (npr. banke). Sklapanje bilo kakvog ozbiljnog ugovora zahtjeva treću stranu koja će potvrditi valjanost ugovora (npr. javni bilježnik). U takvom sustavu neizbježne su naknade koje zahtjeva treća stranka za svoje usluge. Time nastaju visoki troškovi ugovora, a prisutne su i razne sigurnosne prijetnje. U slučaju da druga stranka ne ispoštuje svoju stranu ugovora, možemo li se pouzdati u sustav i vjerovati da ćemo na kraju dobiti sve što nam ugovorom pripada? Uz to, u centraliziranom sustavu uvijek postoji mogućnost urušavanja cijelog sustava ako središnja institucija zakaže (npr. propast banke).

No kako jamčiti povjerenje između dviju stranaka bez prisustva treće stranke? Upravo taj problem povjerenja dvije stranke riješen je pojavom blockchain tehnologije. 2008. godine začetnik pod nadimkom Satoshi Nakamoto svijetu predstavlja prvu kriptovalutu Bitcoin i njegov blockchain - decentralizirani sustav koji primjenom konsenzusa "Proof of work" svih članova osigurava povjerenje i vjerodostojnost sustava. Po uzoru na Bitcoin blockchain, 2013. godine Vitalik Buterin zajedno sa svojim timom predstavio je Ethereum, decentraliziranu blockchain platformu koja podržava kreiranje pametnih ugovora (engl. smart contracts).

Drugo poglavlje započet ćemo povijesnim razvojem knjigovodstva te predstaviti blockchain kao implementaciju trojnog knjigovodstva. Nakon toga reći ćemo što je Bitcoin, objasniti Bitcoin transakcije i temelj povjerenja blockchain mreže - dokaz o radu. Detaljno ćemo objasniti princip kreiranja lanaca blockchaina i zašto je malo vjerojatno da netko preuzme vlast nad blockchain-om i kreira lažne transakcije. Naše tvrdnje potkrijepit ćemo primjerom Merkleovog stabla. Objasnit ćemo SPV protokol i što nam takav protokol omogućava. Potom ćemo promotriti Bitcoin mrežu kao sustav tranzicije stanja. Navesti ćemo 3 različita načina za izgradnju aplikacija na blockchain-u koja su korištena do pojave Ethereum-a, njihove nedostatke i poteškoće prilikom rada.

Nakon toga, u trećem poglavlju, uvest ćemo Ethereum kao rješenje tih problema, objasniti princip rada i strukturu Ethereum mreže, sličnosti i razlike Bitcoin i Ethereum mreže. U četvrtom poglavlju objasnit ćemo što su pametni ugovori, kako funkcionira Ethereum Virtual Machine, kojim vrstama memorije raspolaže i kako se na njoj izvršavaju skupovi uputa, tzv. opkodovi. Objasnit ćemo kako se pišu pametni ugovori u objektno orijenitiranom programskom jeziku Solidity uz korištenje Remix IDE-a. Na primjeru kreiranja Mathos ERC-20 tokena pokazat ćemo kako iskoristiti Solidity biblioteke za jednostavno kreiranje pametnih ugovora te kako objaviti pametni ugovor na Ethereum testnoj mreži. Reći ćemo što je ABI specifikacija ugovora i kako pomoću nje integrirati pametni ugovor u web aplikaciju.

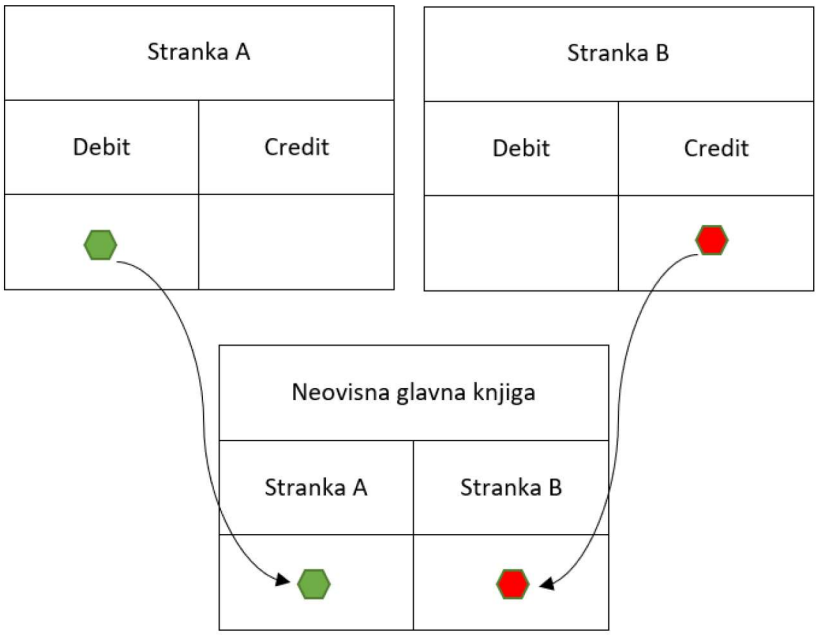
U posljednjim dijelovima rada navest ćemo prednosti blockchain tehnologije i pametnih ugovora te predstaviti konkretne primjene u različitim sektorima. Opisat ćemo kako iskoristiti blockchain i pametne ugovore prilikom prekograničnog plaćanja i financiranja trgovine, za razvoj digitalnog oblika novca centralnih banaka te što su decentralizirane financije. Navest ćemo potencijalne benefite u sustavu zdravstvene skrbi i kontroli pristupa medicinskoj dokumentaciji te kako blockchain i pametni ugovori podižu sigurnost kada se koriste za uprvljanje korisničkim identitetima. Na primjerima ćemo opisati kako već sada blockchain i pametni ugovori podižu upravljanje lancem opskrbe (SCM) na višu razinu te postojeće i moguće primjene u ERP i CRM sustavima. Objasnit ćemo što je BaaS i ukratko navesti ostale moguće primjene blockchain tehnologije i pametnih ugovora.

2 Blockchain

2.1 Od jednostavnog knjigovodstva do blockchaina

Prvi pisani tragovi knjigovodstva[1] odnosno početka računovodstva kao djelatnosti pojavili su se 3200 godina pr. Kr. u Babilonu i Asiriji. Za to najstarije razdoblje računovodstva karakteristično je vrlo jednostavno bilježenje poslovnih događaja koje se primjenjuje sve do početka srednjeg vijeka. Tijekom 13. i 14. stoljeća u Sjevernoj Italiji razvija se jednostavno knjigovodstvo. U jednostavnom knjigovodstvu bilježi se samo dio poslovnih događaja i to samo na jednoj stavci te se sve poslovne promjene evidentiraju kronološkim redom u poslovnim knjigama. Nešto kasnije, 1458. godine, Dubrovčanin Benedikt Kotruljević po prvi puta opisuje dvojno, odnosno dvostruko knjigovodstvo. U dvojnom knjigovodstvu svaki se poslovni događaj knjiži u dvije stavke, od kojih jedna „duguje" a druga „potražuje". Kombinirano korištenje jednostavnog i dvojnog knjigovodstva zadržalo se sve do danas[2].

Godine 1989. profesor Yuji Ijiri u svome radu "Momentum Accounting and Triple-Entry Bookkeeping" predlaže novi način vođenja knjiga koji je nazvao trojno knjigovodstvo. Princip trojnog knjigovodstva zasniva se na povezivanju dvostrukog unosa dvije stranke dodatnim zapisivanjem te iste transakcije na još jedan neovisni, zajednički konto - kako je ilustrirano na slici 1.



Slika 1: Trostruko knjigovodstvo

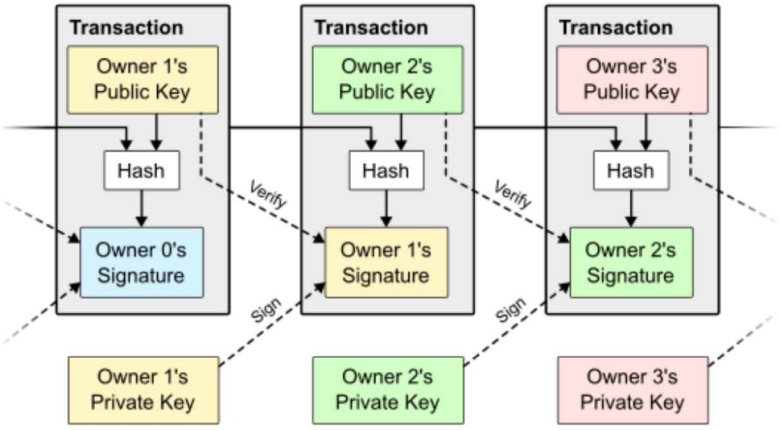
Bilo je više pokušaja implementacije trojnog knjigovodstvenog sustava, no tek 2008. godine pojavom Bitcoin-a i njegove blockchain mreže, ideja profesora Ijirija dobiva uspješnu implementaciju primjenjivu u financijskom sustavu. Blockchain mreža Bitcoin-a postaje javna glavna knjiga u koju će se unositi sve transakcije, a Bitcoin valuta za plaćanje.

[1]Knjigovodstvo (engl. bookkeeping) se kao pojam odnosi na dnevne operacije računovodstvenih sustava, tj. na klasificiranje i zapisivanje rutinskih transakcija. U tom smislu predstavlja uži pojam od računovodstva.[10]

[2]Trenutno u Hrvatskoj jednostavno knjigovodstvo koriste mala poduzeća, obiteljska poljoprivredna gospodarstva i obrti, dok velika poduzeća zakon obvezuje na korištenje dvojnog knjigovodstva.

2.2 Bitcoin

Bitcoin (skraćeno BTC) je prva decentralizirana digitalna valuta. Kombiniranjem digitalnih potpisa i peer-to-peer[3] mreže omogućuje izravno plaćanje preko interneta između dvije stranke, bez posredovanja financijske ustanove. Električni novčić definiran je kao lanac digitalnih potpisa. Svaki vlasnik prenosi novčić na sljedećeg digitalnim potpisivanjem hasha prethodne transakcije i javnog ključa sljedećeg vlasnika te ih dodaje na kraju novčića - kako je ilustrirano na slici 2.



Slika 2: Bitcoin transakcija [6]

Primatelj se može uvjeriti u vjerodostojnost lanca vlasništva provjerom digitalnih potpisa. No, digitalni potpisi ne jamče primatelju da u jednom trenutku neki od prethodnih vlasnika nije dvaput potrošio isti novac. Kada bi jedan od prethodnih vlasnika dvaput potrošio isti novac, nastala bi dva lanca od kojih samo jedan može biti ispravan. U slučaju dvostruke potrošnje, jedino je prva transakcija ona koja se broji. Kako bi se sa sigurnošću znalo koja je transakcija obavljena prva, sve transakcije moraju biti javne i vidljive svim čvorovima blockchain mreže. Dodatno, potreban je sustav u kojem će se sudionici mreže složiti oko jedinstvene povijesti redoslijeda kojim su transakcije primljene. Takav sustav na osnovi peer-to-peer koncepta implementiran je pomoću decentraliziranog konsenzus[4] mehanizma, dokaza o radu (engl. Proof-of-Work).

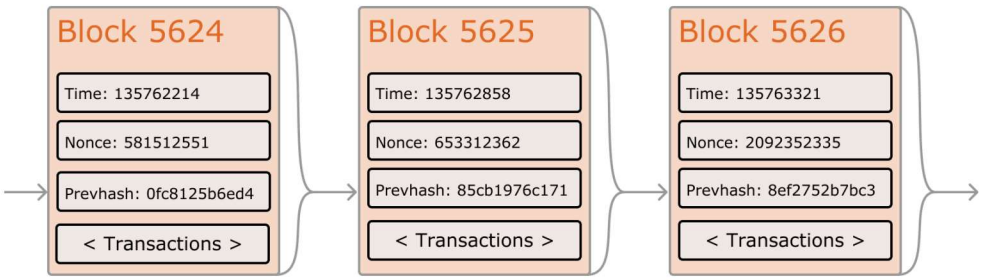
2.2.1 Dokaz o radu

Decentralizirani konsenzus mehanizam Bitcoin mreže zahtjeva konstantne napore čvorova unutar mreže kako bi riješili zahtjevne matematičke jednadžbe i time doprinjeli stvaranju novog skupa transakcija. Transakcije su grupirane u blokove. Svaki blok sadrži vremensku oznaku, referencu na prethodni blok u obliku hash-a, hash Merkelovog korijena, nonce[5] i listu svih transakcija koje su se izvršile nakon prethodnog bloka. Ilustraciju bloka unutar blockchaina vidimo na slici 3.

[3]Peer-to-peer (P2P) mreža je grupa računala od kojih svako djeluje kao čvor za dijeljenje datoteka unutar grupe.

[4]Konsenzus (lat. consensus: slaganje, suglasnost) u političkoj teoriji i praksi, je opća suglasnost pri donošenju odluka. Konsenzus je valjan ako je postignut slobodno, bez sile, straha ili zbog zablude.

[5]Nonce je kratica za "Number Only Used Once" tj. "Broj korišten samo jedanput". Označava jedinstveni broj dodijeljen bloku.



Slika 3: Blok unutar blockchaina [1]

Blockchain mreža Bitcoin-a u prosjeku kreira jedan blok svakih deset minuta. Nove transakcije emitiraju se na sve čvorove mreže te ih svaki čvor sakuplja u blok. Kada čvor pronađe dokaz o radu, on emitira blok svim čvorovima. Ako se utvrdi da je blok valjan te da su sve transakcije unutar bloka važeće i nisu već potrošene, čvorovi prihvaćaju blok. Prihvaćanje iskazuju tako što preusmjere rad na stvaranje sljedećeg bloka u lancu uz korištenje hasha prihvaćenog bloka kao prethodni hash. Hashiranje se provodi korištenjem SHA-256 algoritma za sigurno hashiranje koji za dani ulaz vraća njegov hash veličine 256 bita.

Hash funkcija ima dvije svrhe. Prva je kako bi se sakrio identitet stranaka koje sudjeluju u transakciji. Iako u blockchain mreži svi mogu vidjeti svaku transakciju, istovremeno su transakcije anonimne (osim ako znamo tko su vlasnici adresa koje sudjeluju u transakciji). Druga svrha je samo izvođenje dokaza o radu od strane čvorova mreže u svrhu utvrđivanja konsenzusa mreže. Dokaz o radu uključuje pretraživanje vrijednosti brutforce pristupom, inkrementiranjem nonce vrijednosti sve dok ne dobijemo vrijednost čiji se hash podudara sa zadanom hash vrijednosti. Takav pristup troši velike količine računalnih resursa (CPU, GPU), a samim tim i velike količine električne energije. Prvi čvor koji objavi dokaz o radu na bloku za svoj rad i pronalazak rješenja dobiva nagradu[6] u obliku Bitcoin-a. Iz toga razloga se aktivno sudjelovanje u dokazu o radu naziva rudarenje (engl. mining).

Kako se u blockchain lancu svaki blok nastavlja na prethodni te se u dokazu o radu koristi hash vrijednost prethodnog bloka, kada bi netko htio promijeniti postojeći blok unutar blockchaina, morao bi obaviti dokaz o radu za sve blokove koji slijede iz tog bloka. Ako bi dva čvora istovremeno emitirala dvije različite verzije sljedećeg bloka, dio čvorova bi prvo zaprimio prvu verziju, a dio drugu. U takvoj situaciji, čvorovi bi nastavili rad na bloku kojeg su prvo zaprimili, no sačuvali bi kopiju lanca koja nastaje iz drugog bloka u slučaju da taj lanac postane duži od onog na kojem trenutno rade. U slučaju da konkurentni lanac postane duži, čvorovi će napustiti trenutni lanac i nastaviti rad na dužem. Stoga se dokazom o radu utvrđuje konsenzus mreže, većinsku odluku predstavlja najdulji lanac koji ujedno sadrži najveći uloženi dokaz o radu.

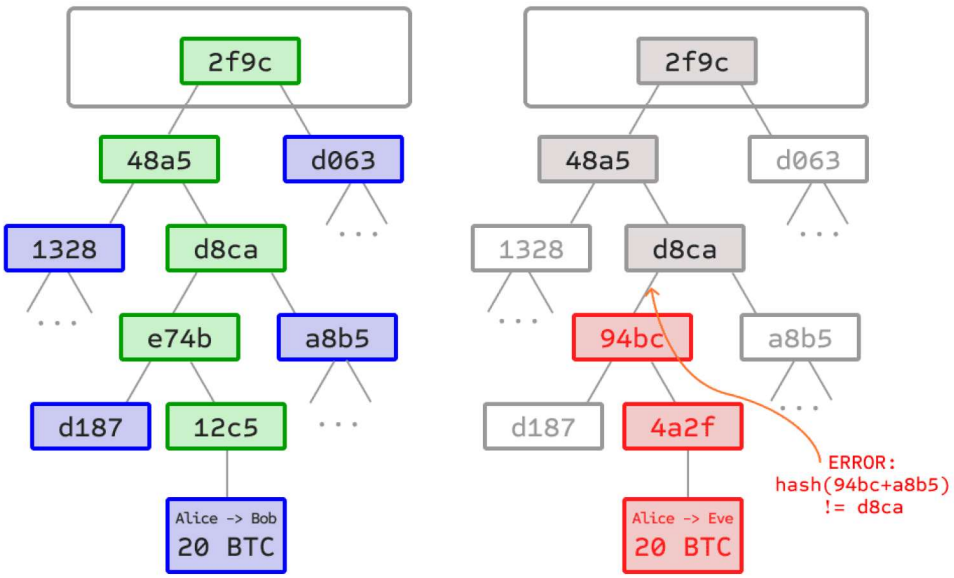
Ako pošteni čvorovi upravljaju većinom procesorske snage mreže, onda će ispravni lanac rasti brže od bilo kojeg drugog lanca u mreži. Kada bi netko pokušao napad na blockchain mrežu namjernim kreiranjem dvostruke potrošnje usmjeravajući transakciju Bitcoin-a u svoj novčanik, morao bi biti jači od polovice mreže, tj. morao bi imati barem 51% ukupne procesorske snage mreže kako bi kreirao blokove brže od ostatka mreže. Čim bi njegov lanac blokova postao kraći od pravog, ostali bi odbacili njegov lanac i prihvatili ispravni.

[6]U trenutku pisanja ovoga rada, nagrada za uspješno rudarenje jednog bloka iznosi 6,25 BTC-a. Nagrada se prepolavlja nakon svakih 210 tisuća kreiranih blokova, dok je gornja granica broja BTC novčića 21 milijun.

2.2.2 Merkleovo stablo

Kako bi se provjerila validnost grane blockchaina, nije potrebno prolaziti kroz cijelu povijest mreže. Dovoljno je provjeriti mali broj čvorova. To nam omogućuje struktura blockchain mreže u obliku **Merkleovog stabla**. Merkleovo stablo je poseban oblik binarnog stabla u kojem je svaki čvor koji nije list hash svoja dva djeteta. U slučaju Bitcoin blockchaina, počevši od roditelja listova, svaki čvor je hash svoja dva djeteta, hashiran SHA-256 funkcijom. Hashiranje se vrši sve dok ne ostane samo jedan čvor kojeg zovemo Merkleov korijen. Upravo je hash Merkleova korijena sastavni dio svakog bloka Bitcoin blockchaina.

Prilikom provjere validnosti grane, vrši se takozvano "podrezivanje" stabla. Dovoljno je uzeti mali broj čvorova kako bi se rekreirali gornji čvorovi potrebni za provjeru grane stabla. Isto tako, kada bi netko pokušao kreirati transakciju koja bi zamijenila postojeću na dnu stabla, ta promjena bi uzrokovala lančanu reakciju promjena nad gornjim čvorovima te bi ubrzo došlo do nekonzistentnosti s ostalim čvorovima stabla. Kako to izgleda na primjeru možemo vidjeti na slici 4. Lijevi prikaz nam ukazuje na mali broj čvorova koje treba provjeriti, a desni da će bilo kakav pokušaj promjene stabla kasnije uzrokovati nekonzistentnost negdje višlje u stablu.

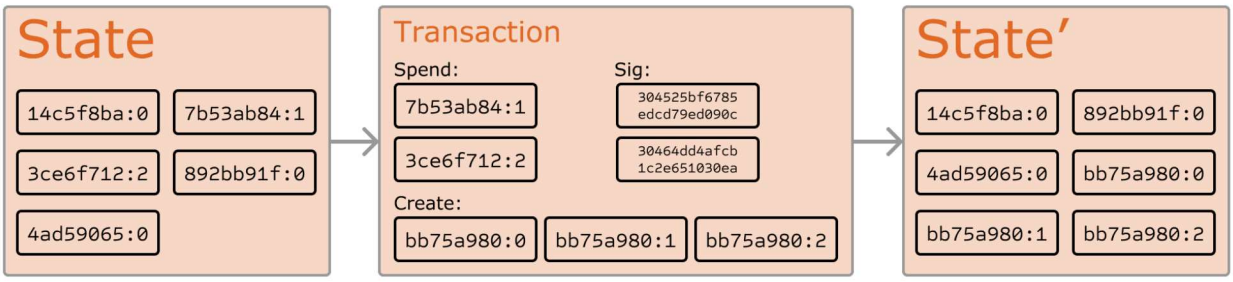


Slika 4: Provjera grana Merkleovog stabla [1]

Blockchain mreža sadrži potpune i lake čvorova, gdje potpuni čvorovi pohranjuju kopiju cijele mreže, dok je lakim čvorovima omogućeno djelomično preuzimanje blockchaina. Laki čvorovi preuzimaju samo zaglavlja čvorova i dijelove grana blockchaina koji su im potrebni za dokaz o radu. Takav protokol naziva se pojednostavljena verifikacija plaćanja ili SPV - "Simplified Payment Verification". SPV protokol jamči lakim čvorovima visoku pouzdanost pri provjeri bilo koje transakcije uz malu količinu podataka koju je potrebno preuzeti za provjeru.

2.2.3 Bitcoin kao sustav tranzicije stanja

Sada kada smo objasnili kako se obrađuju transakcije na Bitcoin mreži, promotrimo Bitcoin mrežu kao sustav tranzicije stanja. Označimo s UTXO[7] svaki kreirani, a nepotrošeni novčić. Svaki UTXO ima svog vlasnika. Vlasnik je definiran 20 bytenom adresom koja čini kriptografski javni ključ. Na stanje u Bitcoin mreži možemo gledati kao kolekciju svih UTXO novčića. Transakcija sadrži jedan ili više ulaza te jedan ili više izlaza. Pri tome svaki ulaz sadrži referencu na postojeći UTXO i kriptografski potpis kreiran iz privatnog ključa vlasnikove adrese, dok svaki izlaz sadrži novi UTXO koji će biti dodan stanju. Prikaz stanja i transakcije opisanog modela vidljiv je na slici 5.



Slika 5: Bitcoin kao sustav tranzicije stanja [1]

Neka je S stanje koje se sastoji od vlasničkog statusa svih postojećih Bitcoin-a te neka je $APPLY$ funkcija tranzicije stanja koja kao varijable prima stanje S i transakciju TX, a kao vrijednost vraća novo stanje S'. Ako je transakcija TX zahtjev za premještanjem određenog broja UTXO-a s adrese A na adresu B, tada će primjena funkcije $APPLY$ na početno stanje i transakciju B rezultirati novim stanjem u kojem će se s početnog stanja adrese A oduzeti traženi iznos UTXO-a, a taj isti iznos dodati na adresu B. Ako transakcija nije valjana, tj. ako je zatražen prijenos UTXO-a veći nego što ih ima na početnom stanju adrese A, funkcija $APPLY$ vratit će grešku. Tada možemo pisati

$$APPLY(S, TX) \rightarrow S' \text{ ili greška.}$$

Djelovanje funkcije $APPLY$ možemo prikazati primjerom.

Primjer 1 *Neka na početnom stanju adresa A sadrži 20 UTXO-a, a adresa B 30 UTXO-a. Ako je transakcija TX zahtjev za prijenosom 10 UTXO-a s adrese A na adresu B, tada imamo:*

$$APPLY(\{\ A\text{: 20 UTXO, B: 30 UTXO}\ \}, TX) = \{A\text{: 10 UTXO, B: 40 UTXO.}\}$$

No, kada bi se transakcijom TX' zahtjevao prijenos 30 UTXO-a s adrese A na adresu B, funkcija APPLY vratila bi grešku zbog nedovoljnog broja UTXO-a na adresi A.

$$APPLY(\{\ A\text{: 20 UTXO, B: 30 UTXO}\ \}, TX') = ERORR.$$

[7]UTXO - engl. Unspent transaction outputs

Prethodno opisana funkcija $APPLY$ može se iskazati algoritmom 1:

Algoritam 1 APPLY funkcija [1]

Podatci: Stanje S, Transakcija TX
Rezultat: Stanje S' ili greška
za svaki *ulaz u* TX **čini**
  ako *referencirani UTXO nije u* S **onda**
    vrati *grešku*
  ako *dani potpis ne odgovara vlasniku UTXO-a* **onda**
    vrati *grešku*
ako *je zbroj svih ulaznih UTXO manji od zbroja svih izlaznih UTXO* **onda**
  vrati *grešku*
vrati S *sa uklonjenim svim ulaznim UTXO-ima i dodanim svim izlaznim UTXO-ima.*

U poglavlju 2.2.1 rekli smo da čvorovi prihvaćaju blok ako se utvrdi da je blok valjan i ako su sve transakcije unutar njega važeće. Sada kada smo uveli funkciju $APPLY$, proces valjanosti bloka i transakcija možemo i formalno zapisati sljedećim algoritmom:

Algoritam 2 Provjera validnosti bloka [1]

1. Provjeri postoji li prethodni blok i je li valjan.
2. Provjeri je li vremenska oznaka bloka veća od vremenske oznake prethodnog bloka.
3. Provjeri valjanost dokaza o radu promatranog bloka.
4. Neka je $S[0]$ stanje na kraju prethodnog bloka.
5. Pretpostavimo da je TX lista transakcija unutar bloka duljine n.
   Za svaki $i \leftarrow 0$ **do** $n-1$ **čini**
   $setS[i+1] = APPLY(s[i], TX[i])$.
   Ako funkcija tranzicije stanja vrati grešku za bilo koji i, vrati grešku.
6. **vrati *ISTINA*** *te registriraj* $S[n]$ *kao stanje na kraju bloka.*

2.2.4 Bitcoin skripte

Nakon pojave Bitcoin-a 2008. godine, počele su nastajati alternativne blockchain aplikacije različitih primjena. Prema pristupu izgradnje konsenzus protokola, sve blockchain projekte do 2014. godine generalno možemo podijeliti u dvije skupine: projekte koji su za svoj konsenzus protokol izgradili nove neovisne blockchain mreže i projekte koji su svoje protokole izgradili nad mrežom Bitcoin-a.

Izgradnja nove blockchain mreže za svaku pojedinu aplikaciju nije praktična zbog zahtjevne implementacije same mreže. Dodatno, interakcija između decentraliziranih aplikacija izgrađenih na različitim blockchain mrežama bila bi otežana. S druge strane, izgradnja protokola povrh Bitcoin blockchain mreže nije praktična jer se time ne nasljeđuje važno svojstvo Bitcoin mreže - pojednostavljena verifikacija plaćanja ili SPV protokol. Projekti koji svoje protokole temelje nad Bitcoin mrežom ne mogu odbaciti transakciju koja nije validna provjerom malog broja čvorova kao što je slučaj u Bitcoin mreži. Stoga je takvim protokolima

potrebno skenirati cijelu mrežu Bitcoin-a kako bi se uvjerili u vjerodostojnost transakcije. Kako Bitcoin blockchain neprestano raste, kompletna provjera mreže prilikom svake provjere transakcije jednostavno nije održiva. Sudeći po tome, protokoli izgrađeni nad Bitcoin mrežom prilikom provjere sigurnosti mreže oslanjaju se na pouzdane poslužitelje.

Iako takav princip rada rješava problem provjere transakcija i sigurnosti mreže, samim uključivanjem poslužitelja mreža prestaje biti decentralizirana i neovisna, čime primarna svrha blockchaina gubi smisao. No, Bitcoin protokol čak i bez ikakvih proširenja pruža mogućnost izrade **pametnih ugovora**. UTXO nije nužno u vlasništvu javnog ključa, može biti i u vlasništvu kompliciranije **skripte** izražene u jednostavnom programskom jeziku baziranom na stogu.

U ovoj paradigmi, transakcija UTXO-a će biti izvršena ako se zadovolje određeni uvjeti iz skripte. Čak je i osnovni mehanizam vlasništva javnog ključa implementiran putem skripte. Pri verificiranju skripta kao ulaz uzima potpis u obliku eliptičke krivulje[8], provjerava dani potpis s transakcijom i adresom koja posjeduje UTXO te vraća 1 ako je verifikacija uspješna, a 0 inače. Skripte se mogu koristiti u razne svrhe, no skriptni jezik kakav je implementiran u Bitcoin mreži ima određena ograničenja. Skriptni jezik Bitcoin-a nije Turing potpun, prvenstveno se ističe nedostatak petlji. To je učinjeno kako bi se izbjegle beskonačne petlje tijekom provjere transakcije. Iako postoje načini da se petlje simuliraju nizanjem if-else naredbi, jasno je kako takav način rada nije efikasan.

Kao drugi problem nameće se nedostatak kontrole broja UTXO-a koji se mogu poslati s jedne adrese na drugu. Objasnimo to na primjeru.

Primjer 2 *Neka je Ana primila tri transakcije od po 3, 4 i 5 BTC-a. Ana nakon zaprimanja transakcija posjeduje 12 BTC-a i želi poslati 2 BTC-a svome prijatelju. Kako bi Ana posla 2 BTC-a na adresu svoga prijatelja, u skripti je potrebno odraditi sljedeće korake:*

- *Izabrati jednu od ulazih transakcija koje je Ana primila, a nije još potrošila.*
- *Dokazati da ta ulazna transakcija stvarno pripada Ani.*
- *Kreirati izlaz do javnog ključa ili adrese kao destinaciju na koju Ana želi poslati 2 BTC-a.*

No, Ana nije zaprimila niti jednu transakciju u iznosu od točno 2 BTC-a pa dolazi do neočekivanog ishoda već nakon prvog koraka. Umjesto jedne transakcije od 2 BTC-a s Anine adrese na adresu njezinog prijatelja, uzimaju se 3 BTC-a iz prve transakcije koju je Ana zaprimila i od ta 3 BTC-a, dva se šalju jednom transakcijom Aninom prijatelju, dok se jedan BTC šalje drugom transakcijom nazad na Aninu adresu.

Treći problem je nedostatak međustanja. UTXO može biti ili potrošen ili nepotrošen. Nije moguće kreirati pametni ugovor s više od dva stanja. Iz tog razloga na Bitcoin mreži nije moguće izgraditi decentralizirane mjenjačnice.

Četvrti problem je Blockchain-blidness. Prilikom programiranja u skripti Bitcoina, nemamo uvid u podatke kao što su nonce i prethodni hash bloka. Time se onemogućava nasumičnost, a samim time i izgradnja bilo kakve igre na sreću.

[8]Elliptic Curve Digital Signature Algorithm (ECDSA) je ANSI standard za generiranje digitalnih potpisa koji pri potpisivanju poruka koristi eliptičke krivulje. Pomoću ECDSA algoritma moguće je generirati ključ te generirati i provjeriti potpis.

Iz gore navedenih razloga, korištenje Bitcoin skripti nije najbolje rješenje za izgradnju decentraliziranih aplikacija na blockchain-u. Naveli smo ukupno tri načina za izgradnju naprednih aplikacija na blockchainu: izgradnja novog blockchaina, izgradnja protokola povrh Bitcoina te korištenjem Bitcoin skripti. Svaki od tri načina ima svojih problema i ograničenja koja ga čine nepraktičnim u primjeni. Kao rješenje tih problema 2014. pojavljuje se Ethereum.

3 Ethereum

Ethereum je decentralizirana platforma za izgradnju decentraliziranih aplikacija i pametnih ugovora. Temelj platforme čini Ethereum blockchain s ugrađenim Turing-potpunim programskim jezikom. Izgradnja pametnih ugovora i decentraliziranih aplikacija znatno je jednostavnija s puno više mogućnosti u odnosu na izgradnju istih pomoću Bitcoin skripti, prvenstveno jer je Ethereum objedinjeno rješenje sva četiri nedostatka Bitcoin skripti koja smo nabrojali u prethodnom poglavlju. Na Ethereum platformi možemo koristiti prednosti Turing potpunog jezika, lako kontrolirati vrijednosti, kreirati međustanja i provjeriti podatke sa same blockchain mreže.

Kriptovaluta Ethereum mreže naziva se Ether (skraćeno ETH). Ether na Ethereum mreži ima ulogu pokretačkog goriva. Prilikom svake transakcije plaća se transakcijska naknada u određenom iznosu Ethera. Za razliku od Bitcoina, ukupan broj Ethera nije ograničen, no broj Ethera koji se mogu pustiti u opticaj ograničen je na 18 milijuna godišnje. Pritom u prosjeku svakih 15 sekundi nastaje novi blok na Ethereum blockchainu, a kao nagradu za uloženi dokaz o radu, u trenutku pisanja ovoga rada, rudari su nagrađeni s 2 Ethera.

3.1 Povijesni razvoj Ethereuma

Ethereum se prvi puta spominje u studenom 2013. godine kada suosnivač Vitalik Buterin objavljuje Ethereum Whitepaper [1]. Pet mjeseca kasnije, u travnju 2014. godine, dr. Gavin Wood objavljuje Ethereum Yellow Paper [11]. Gavinov rad je bio svojevrsna tehnička dokumentacija za Ethereum Virtual Machine (EVM), o čemu ćemo nešto više reći u potpoglavlju 4.1. Sredinom 2015. godine, u pogon je pušten Ethereum testnet, Olympic, a nakon uspješnog testiranja, 30. srpnja iste godine u pogon je puštena prva faza Ethereumovog razvoja pod nazivom Frontier. Developeri iz cijeloga svijeta počeli su razvijati pametne ugovore i decentralizirane aplikacije, a jedan od prvih projekata na Ethereum platformi bio je decentralizirani fond rizičnog kapitala, DAO.

Projekt je privukao pažnju brojnih ulagača te je pri inicijalnoj ponudi novčića prikupio gotovo 14% svih ETH novčića koji su tada bili u opticaju. No, developeri su napravili brojne greške u kodu što je nepoznati hakerski tim iskoristio kako bi na svoj račun prebacio ETH novčiće u vrijednosti preko 50 milijuna američkih dolara. Kako bi se poništio događaj krađe ETH novčića, Ethereum zajednica odlučila se za hard fork Ethereum blockchaina. Hard forkom mreža je podijeljena na dva lanca, Ethereum i Ethereum Classic. Na Ethereum lancu hakerski napad je poništen te su novčići vraćeni prvotnim vlasnicima, dok je na Ethereum Classic lancu ostala originalna mreža bez ikakvih izmjena. Ethereum mreža se redovito nadograđuje, a posljednja nadogradnja pod nazivom "Arrow Glacier" dogodila se 9. prosinca 2021. godine.

3.2 Ethereum računi

U Ethereumu se svako stanje sastoji od objekata koje zovemo računi (engl. accounts), pri čemu svaki račun sadrži 20 byte-nu adresu. Tranzicijska funkcija je u ovom slučaju direktni prijenos vrijednosti i informacija između različitih računa. Svaki račun sadrži četiri polja: nonce, trenutno stanje računa izraženo u Etheru, kod ugovora i pohranu računa.

Postoje dvije vrste računa: račun u vanjskom vlasništvu koji je kontroliran privatnim ključem i račun ugovora koji je kontroliran programskim kodom ugovora. Svaki put kada račun

ugovora primi poruku, aktivira se programski kod u njemu uz mogućnost čitanja i pisanja u internu pohranu te slanja drugih poruka ili stvaranja novih ugovora. Za razliku od računa ugovora, račun u vanjskom vlasništvu ne sadrži nikakav kod. S računa u vanjskom vlasništvu moguće je poslati poruku slanjem i potpisivanjem transakcije.

3.3 Transakcije

Svaka transakcija sadrži sljedeća polja:

- primatelja poruke
- digitalni potpis pošiljatelja
- iznos Ethera koje pošiljatelj šalje primatelju
- polje podataka (nije obavezno)
- STARTGAS vrijednost
- GASPRICE vrijednost.

Prva tri polja su standardni dio transakcije u svakoj kriptovaluti. Polje podataka kod običnih transakcija s računa u vanjskom vlasništvu na drugi račun u vanjskom vlasništvu nema nikakvu svrhu, no koristi se prilikom transakcija koje uključuju račune ugovora i tada služe kao pohrana podataka. STARTGAS i GASPRICE su ključna polja svake transakcije Ethereum mreže.

Pomoću STARTGAS vrijednosti pošiljatelj osigurava maksimalan broj računalnih koraka koje transakcija može izvršiti, dok se pomoću GASPRICE vrijednosti izražava cijena koju je pošiljatelj spreman platiti prilikom svakog koraka. Postoji naknada za svaki računalni korak koji je potrebno izvršiti. Ta naknada izražava se u jedinicama goriva, a jedinice goriva plaćaju se u Etherima. Kako je riječ o vrlo malim iznosima Ethera, osnovna jedinica u kojoj se izražava cijena jedinice goriva je gwei, gdje jedan gwei iznosi 10^{-9} Ethera[9]. Različite operacije zahtijevaju različit broj računalnih koraka, a tako i različite naknade u broju jedinica goriva. Dio naknada operacija se mijenjao s nadogradnjama mreže, dok su neke naknade ostale nepromijenjene, tj. ostale su iste kakve ih je Gavin definirao u radu [11]. Tako je za svaku običnu transakciju potrebno platiti 21.000 jedinica goriva i ta se naknada nikada nije mijenjala.

Naknade služe kako bi se maksimizirala učinkovitost, imaju ulogu svojevrsne motivacije developerima kako bi pisali kod što je bolje moguće. Minimiziranjem broja potrebnih operacija unutar koda platit će manje naknade kada se kod bude izvršavao na mreži. Time se ujedno rasterećuje mreža od nepotrebnog sadržaja te sprječava pojava beskonačnih petlji. Kada broj računalnih koraka unutar koda transakcije dostigne iznos izražen sa STARTGAS vrijednošću, transakcija će se automatski prekinuti. Nakon prekida transakcije, sve izmjene će se poništiti, izuzev goriva transakcije izraženog u Etherima, koje će se kao naknada za obradu transakcije prenijeti rudarima za utrošeni rad.

Pomoću GASPRICE vrijednosti developeri mogu utjecati na brzinu izvršavanja njihove aplikacije. Povećanjem GASPRICE iznosa kojeg su spremni platiti za svaki korak transakcije,

[9]Mjerne jedinice Ethera: Wei (10^{-18} ETH), KWei (10^{-15} ETH), GWei (10^{-9} ETH), TWei ili szabo (10^{-6} ETH), PWei ili finney (10^{-3} ETH).

povećava se i prioritet njihovih transakcija na mreži. Iako se na Ethereum mreži blokovi kreiraju znatno brže nego li je to slučaj na Bitcoin blockchainu, u trenucima zagušenosti mreže potrebno je platiti veći GASPRICE za nesmetani rad aplikacije. Nakon implementacije "London nadogradnje" 5. kolovoza 2021., osnovni dio naknade transakcije se trajno uništava, a rudaru ostaje samo napojnica koju je pošiljatelj dodao na osnovni dio naknade kako bi se njegova transakcija pozicionirala višlje u prioritetima. Izračun krajnje cijene transakcije dan je sljedećom formulom:

$$\text{Cijena transakcije} = \text{Broj jedinica goriva} \cdot (\text{Osnovni dio} + \text{napojnica})$$

Pokažimo kako to funkcionira na konkretnom primjeru.

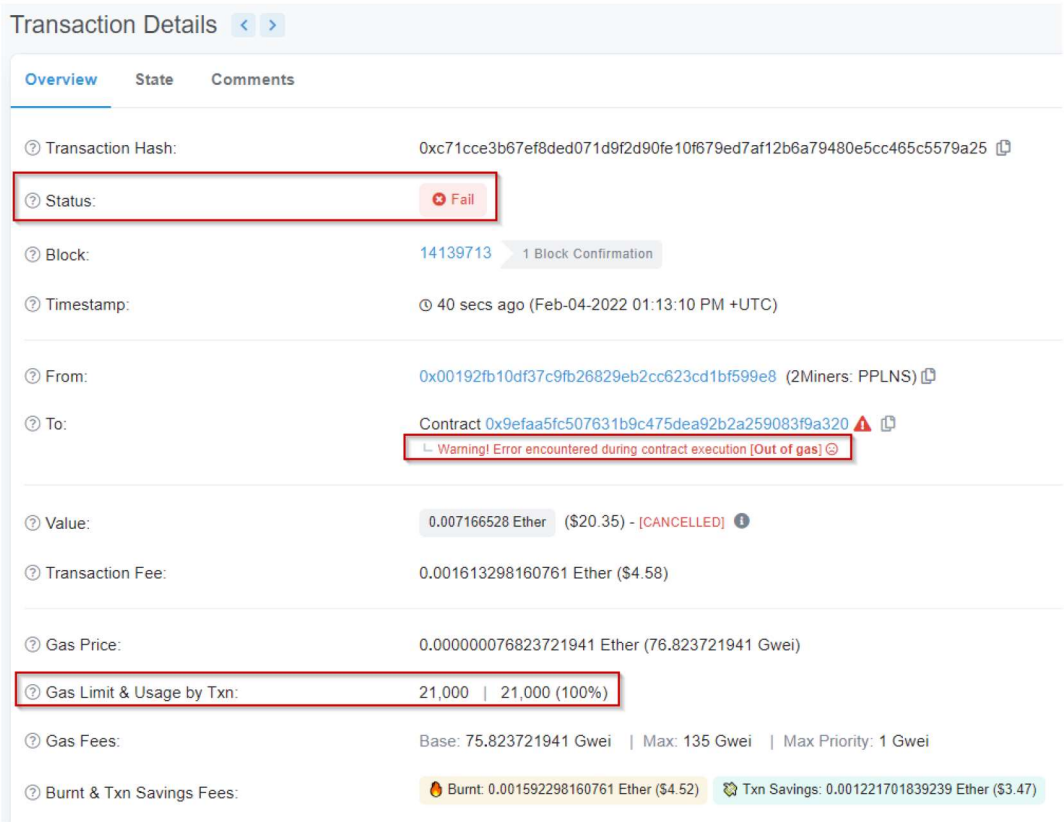
Primjer 3 *Ana mora hitno poslati Marku* 1 *ETH. Ana postavlja vrijednost za STARTGAS na* 21.000 *(jer je toliko potrebno za izvršenje obične transakcije). U trenutku slanja, osnovna cijena jedinice goriva potrebna za uključivanje transakcije u blok iznosi* 100 *gweia. Pošto Ana mora hitno poslati novčiće Marku, uz osnovni iznos dodaje još dodatnih* 20 *gweia kako bi njena transakcija imala što veći prioritet na mreži.*

Tada prema gornjoj formuli ukupna cijena transakcije u gweima iznosi:

$$21.000 \cdot (100 + 20) = 2.520.000.$$

Preračunato u ETH, ukupna cijena transakcije iznosi $0,00252$ *ETH. Od tog iznosa,* $0,0021$ *ETH će biti trajno uništeno, dok će rudaru pripasti* 0.00042 *ETH. S Aninog računa oduzet će se* $1,00252$ *ETH, a kroz par sekundi, Marko će primiti* 1 *ETH.*

Ako pak pošiljatelj ne odredi dovoljno veliku vrijednost za STARTGAS, prijeti mu opasnost od prekida transakcije. Primjer prekinute transakcije zbog nedostatka goriva vidimo na slici 6.



Slika 6: Etherscan - "Out of gas" greška

3.4 Ethereum kao tranzicija stanja

U potpoglavlju 2.2.3 prikazali smo Bitcoin mrežu kao sustav tranzicije stanja. Sada ćemo isto učiniti i za Ethereum mrežu. Započnimo s definicijom funkcije tranzicije APPLY. Jednako kao u slučaju Bitcoina, APPLY funkcija djeluje na stanje S i transakciju TX, a vraća novo stanje S'. Algoritam funkcije APPLY je sljedeći:

Algoritam 3 Ethereum - Funkcija APPLY [1]

1. Provjeri je li transakcija dobro oblikovana, je li potpis validan i odgovara li nonce vrijednost. Ako ne, vrati grešku.
2. Izračunaj cijenu transakcije: STARTGAS · (Osnovni dio + napojnica). Oduzmi taj iznos s računa pošiljatelja i povećaj nonce vrijednost pošiljatelja. Ako pošiljatelj nema dovoljan iznos na računu, vrati grešku.
3. Inicijaliziraj GAS = STARTGAS i oduzmi određeni iznos goriva za svaki byte transakcije.
4. Prenesi Ethere s računa pošiljatelja na račun primatelja. Ako račun primatelja ne postoji, kreiraj ga. Ako je račun primatelja ugovor, izvršavaj naredbe iz koda ugovora do završetka koda ili dok ne ponestane goriva za daljnje izvršavanje.
5. Ako pošiljatelj nije imao dovoljna sredstva za dovršetak transakcije ili ako je izvršavanje koda zaustavljeno zbog ponestalog goriva, poništi sve promjene stanja osim plaćenih naknada i prenesi naknade na račun rudara.
6. Inače, vrati preostalo gorivo vlasniku, a gorivo utrošeno na izvršavanje transakcije pošalji rudaru.

Stanje Ethereum mreže uključuje kolekciju svih računa mreže, sve dosad kreirane ugovore, podatke koji se odnose na informacije iz fizičkog svijeta... Ukratko, sve što je ikad zabilježeno na Ethereum mreži. Kada bismo na izvorno stanje kakvo je bilo u trenutku nastanka mreže postepeno primjenjivali sve transakcije redom kako su se izvršavale, dobili bismo konačno stanje mreže.

3.5 Struktura Ethereum mreže

No, za razliku od Bitcoin mreže, na Ethereum mreži možemo u svakom trenutku provjeriti iznos koji imamo na svome računu, kao i postoji li račun primatelja transakcije. To nam omogućuje posebna struktura Ethereum mreže.

Zaglavlje svakog bloka Ethereum mreže sadrži polja stateRoot[10]. Polje stateRoot sadrži hash korijenskog čvora stabla stanja, a stablo stanja informaciju o cjelokupnom stanju mreže. Provjerom podataka iz polja stateRoot čvorovi relativno brzo dolaze do informacije o stanju blockchaina. Kako bi stablo stanja sadržavalo ispravne informacije o stanju mreže, potrebno ga je redovito ažurirati. Stoga je i validacija bloka Ethereum mreže nešto kompliciranija od validacije bloka Bitcoin mreže. Algoritam provjere bloka je sljedeći:

[10]Zaglavlje bloka Ethereum mreže sadrži 15 polja, a detaljan popis može se pronaći u [11], poglavlje 4.4.

Algoritam 4 Provjera validnosti bloka Ethereum mreže [1]

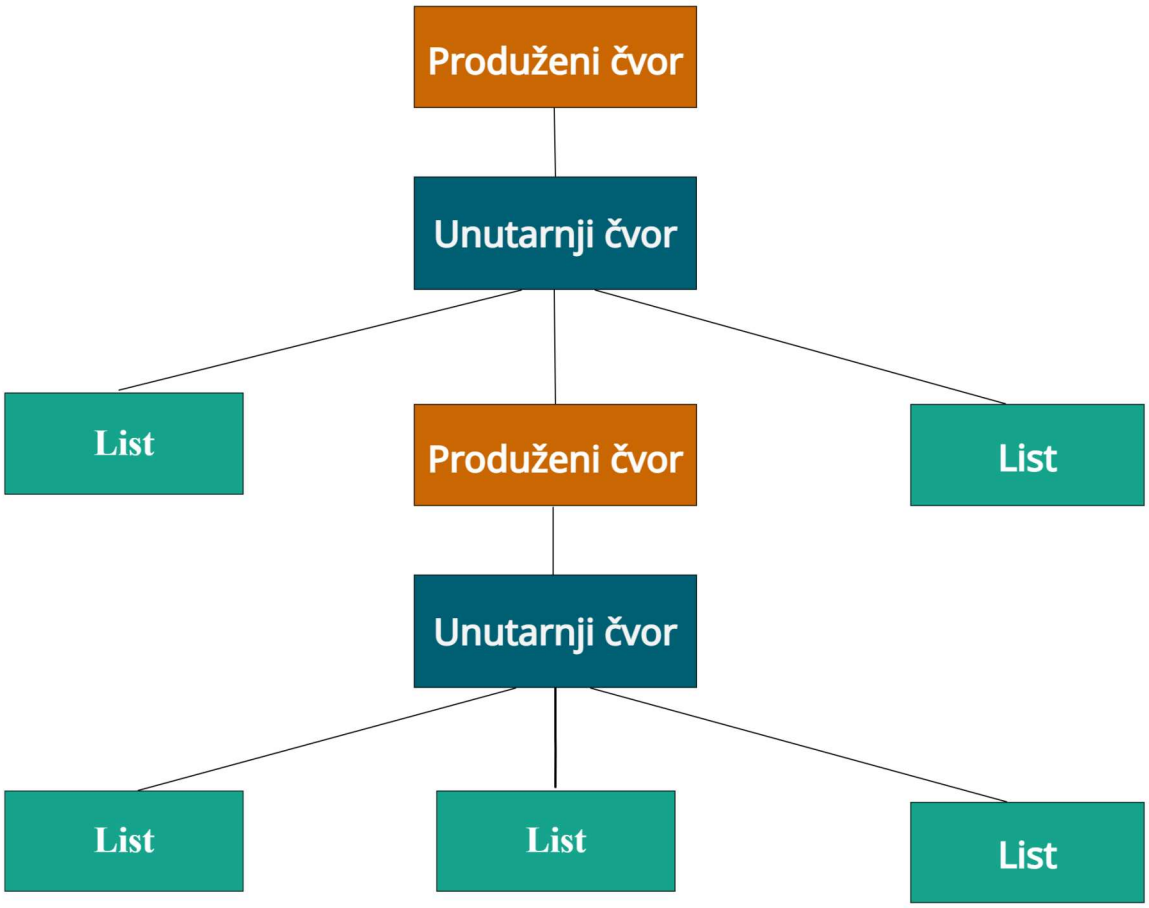
1. Provjeri postoji li prethodni blok i je li valjan.
2. Provjeri je li vremenska oznaka bloka veća od vremenske oznake prethodnog bloka.
3. Provjeri različite koncepte specifične za Ethereum mrežu kao što su broj bloka i težinu bloka, korijen stabla transakcije.
4. Provjeri valjanost dokaza o radu promatranog bloka.
5. Neka je $S[0]$ stanje na kraju prethodnog bloka.
6. Pretpostavimo da je TX lista transakcija unutar bloka duljine n.
   Za svaki $i \leftarrow 0$ **do** $n-1$ **čini**
   $setS[i+1] = APPLY(s[i], TX[i])$.
   Ako funkcija tranzicije stanja vrati grešku za bilo koji i, vrati grešku. Ako tijekom procesa provjere utrošeno gorivo premaši zadani GASLIMIT, vrati grešku.
7. Označimo sa S_FINAL vrijednost $S[n]$ s dodavanom nagradom rudaru za odrađeni dokaz o radu.
8. Provjeri je li korijen Merkleovog stabla od S_FINAL jednak konačnom stanju stabla stanja iz zaglavlja bloka. Ako da blok je validan, inače, blok nije validan.

Na prvu se algoritam čini dosta neučinkovitim jer se pri kreiranju svakog bloka kompletno stanje mreže pohranjuje u blok. No, kako se stanje mreže pohranjuje u strukturi stabla, prilikom kreiranja bloka potrebno je promijeniti samo mali dio toga stabla. Tako je između dva susjedna bloka razlika u stablu stanja vrlo mala, tj. veliki dio stabla se podudara. Dovoljno je podatke spremiti jednom, a dohvaćati ih pomoću dva pokazivača. To je riješeno pomoću modificiranog Merkleovog Patricia stabla.

3.6 Merkleovo Patricia stablo

Uz hash korijena stabla stanja, svaki blok sadrži hash korijena stabla transakcija (polje transactionsRoot) i hash korijena stabla transakcijskih računa (polje receiptsRoot). U sva tri slučaja riječ je hashiranju pomoću Keccak 256-bit hash funkcije. U poglavlju 7 objasnili smo strukturu Bitcoina na principu Merkleovog stabla. Tamo je bila riječ o jednostavnom binarnom Merkleovom stablu, no za potrebe Ethereum mreže, takva vrsta stabla nije zadovoljavajuća. Prvenstveno zbog stabla stanja koje se mora konstantno ažurirati.

Iz tog razloga, na Ethereum mreži koristi se modificirano Merkleovo Patricia stablo koje omogućava umetanje i brisanje čvorova stabla. Nastalo je kombiniranjem modificiranog Merkleovog stabla i Patricia stabla. Postoje četiri vrste čvorova u Merkleovom Patricia stablu: prazni čvorovi, listovi, produženi čvorovi i unutarnji čvorovi. Listovi nemaju djecu te su u njima pohranjeni podaci u obliku para ključa i vrijednosti. Produženi čvorovi mogu imati samo jedno dijete, dok unutarnji čvorovi mogu imati do šesnaest direktnih potomaka. Primjer strukture Merklovog Patricia stabla vidimo na slici 7.



Slika 7: Merkleovo Patricia stablo

Na samom početku rada, u poglavlju 2.1, opisali smo Bitcoin mrežu kao javnu glavnu knjigu u koju se unose sve transakcije. Ethereum nije samo distribuirana glavna knjiga, ne prate se samo stanja računa nego cjelokupno stanje mreže. Kao što smo već rekli, na Ethereum platformi omogućeno je kreiranje decentraliziranih aplikacija i pametnih ugovora. Objasnili smo osnovnu strukturu mreže i kako se na mreži izvršavaju transakcije. Sada ćemo nešto više reći o izvršavanju pametnih ugovora.

4 Pametni ugovori

Pojam pametni ugovor prvi put objasnio je Nick Szabo 1994. u svome radu "Smart Contracts" [7]. Prema njemu, pametni ugovor je računalni transakcijski protokol koji izvršava uvjete ugovora. Szabo je već tada predvidio kako bi pomoću takvog protokola mogli minimizirati potrebu za posrednikom prilikom sklapanja ugovora između dvije stranke te smanjiti broj pogrešaka prilikom izvršavanja ugovora. Kao jednostavne oblike pametnih ugovora tada je naveo POS uređaje i elektroničku razmjenu podataka (EDI)[11]. Iako oba primjera značajno olakšavaju transakcije i razmjenu podataka te smanjuju broj grešaka tijekom istih, posrednik je i dalje prisutan. Baš kao i u slučaju Ijirijeve ideje trostrukog knjigovodstva, za napredniju implementaciju Szabine ideje pametnih ugovora trebalo je pričekati pojavu blockchain tehnologije.

Na Ethereum pametne ugovore možemo gledati kao na programe koji su pohranjeni na Ethereum blockchain mreži te se pokreću kada su ispunjeni unaprijed određeni uvjeti. Najčešće se koriste za automatizaciju izvršenja sporazuma tako da svi sudionici mogu odmah biti sigurni u ishod, bez uključivanja posrednika ili gubitka vremena. Pomoću pametnih ugovora može se automatizirati tijek rada povezivanjem više pametnih ugovora koji će redom pokretati jedan drugoga kako se budu ispunjavali uvjeti.

4.1 Ethereum Virtual Machine

Okruženje za implementaciju i izvođenje pametnih ugovora na Ethereum mreži naziva se Ethereum Virtual Machine (EVM). Tisuće računala diljem svijeta zajedno pokreću EVM i omogućuju neprekidno izvršavanje pametnih ugovora. Programski kod pametnih ugovora pisan je u bytecode jeziku niske razine, baziranom na stogu, kojeg nazivamo Ethereum virtual machine code (EVM kod). EVM kod je Turing potpun, što znači da može riješiti bilo koji rješivi računalni problem, bez obzira na njegovu složenost. Svaki program napisan u EVM kodu u jednom trenutku će završiti, ili kad odradi sve što je trebao, ili od ponestanka goriva za daljnje izvršavanje. Dodatno, EVM je deterministički stroj tako da će za dani ulaz uvijek dati isto rješenje, bez obzira koliko ga puta pokrenuli.

EVM za izvršavanje određenih zadataka koristi skup uputa koje zovemo opkodovi (engl. opcodes). U trenutku pisanja ovog rada postoji 141 opkod EVM-a. Prema radnjama koje izvršavaju, opkodove možemo podijeliti u sljedeće kategorije:

- usporedni i bitwise logički operatori (ADD, SUB, GT, LT, AND, OR)
- informacije o okruženju (CALLER, CALLVALUE, BALANCE)
- informacije o bloku (BLOCKHASH, TIMESTAMP, DIFFICULTY)
- opkodovi za manipulaciju stogom (POP, SSTORE, SLOAD)
- PUSH opkodovi (PUSH1, PUSH2, ... PUSH32)
- opkodovi dupliciranja (DUP1, DUP2, ... DUP16)
- SWAP opkodovi (SWAP1, SWAP2, ... SWAP16)

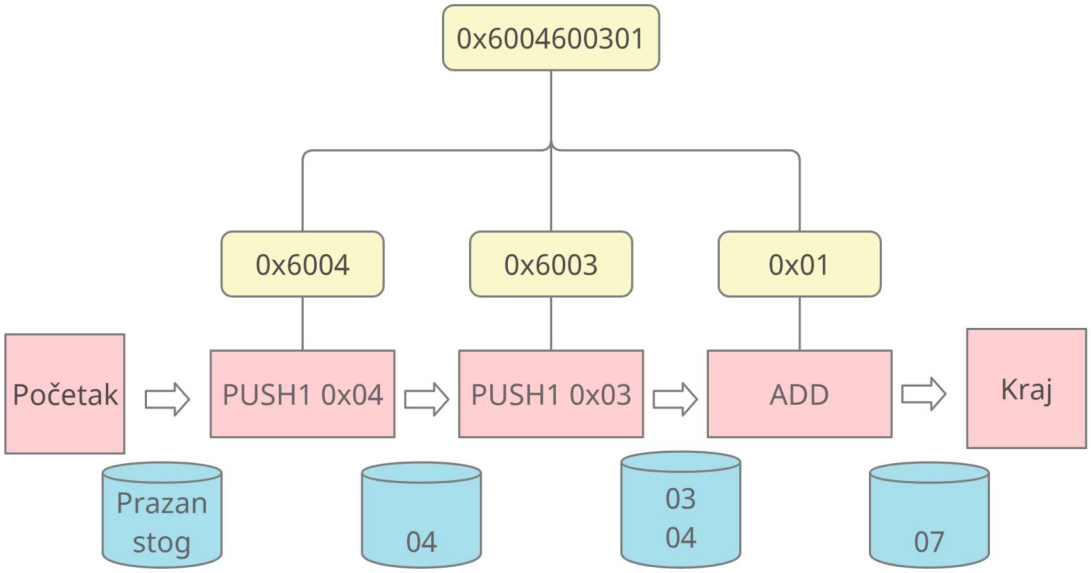
[11] EDI sustav omogućuje elektroničku razmjenu podataka i dokumenata između poslovnih partnera. Primarni cilj sustava je automatizirati sustav nabave i fakturiranja (npr. elektroničko izdavanje računa poznato kao e-Račun).

- opkodovi logiranja (LOG0, LOG1, ... LOG4)
- sistemski opkodovi (CREATE, CALL, SELFDESTRUCT)
- SHA-3 opkod (SHA3).

Popis svih opkodova možemo pronaći na Ethereum stranici [21]. Izvršavanje većine opkodova zahtjeva naknadu u jedinicama goriva. Tako je za jednostavne aritmetičke operacije potrebno izdvojiti 3 jedinice goriva, dok je za složenije operacije kao što je kreiranje pametnog ugovora (opkod CREATE) potrebno izdvojiti čak 32000 jedinice goriva. No, za neke opkodove kao što je opkod za zaustavljanje izvršavanja i naknadno uništenje SELFDESTRUCT, dobiva se povrat goriva kao nagrada za rasterećenje mreže.

Kako bi se učinkovito pohranili, opkovi su kodirani u bytecode. Svaki opkod ima svoj heksadecimalni par. Tako je heksadecimalni zapis opkoda ADD jednak 0x01, a 0x60 heksadecimalni zapis opkoda PUSH1. Prikažimo to na primjeru:

Primjer 4 *Dan nam je bytecode u heksadecimalnom zapisu* 0x6004600301. *Rastavimo ga na byteove i zapišimo u obliku opkodova.*



Slika 8: Bytecode rastavljen na byteove, ekvivalentni opkodovi

PUSH *opkodovi primaju vrijednosti i dodaju ih na vrh stoga pa se iza njih uvijek očekuje određen broj byteova koje će primiti kao vrijednost. Tako* PUSH1 *očekuje* 1 *byte, dok* PUSH32 *očekuje* 32 *bytea. Zbog toga se iza heksadecimalnog ekvivalenta* 0x60 *opkoda* PUSH1 *uzima dodatan byte, u ovom slučaju* 0x04 *i* 0x03. *Byte* 0x01 *označava Opkod* ADD *koji na kraju zbraja dvije vrijednosti, te je konačni rezultat* 0x07, *u decimalnom zapisu broj* 7.

U primjeru je kao prostor za pohranu podataka korišten stog. Stog EVM-a može pohraniti najviše 1024 elementa pri čemu je svaki element veličine 256 bita. Pametni ugovori na EVM uz stog na raspolaganju imaju još i memoriju ugovora (engl. contract memory) te pohranu ugovora (engl. contract storage).

Memorija ugovora se koristi za pohranu podataka prilikom izvršavanja pametnog ugovora. Kada završi izvršenje ugovora, sadržaj memorije se briše. EVM pruža tri opkoda za interakciju s memorijom ugovora:

- MLOAD - učitava riječ iz memorije na stog,
- MSTORE - pohranjuje riječ u memoriju,
- MSTORE8 - pohranjuje jedan byte u memoriju.

Memorija ugovora je neograničena, no treba uzeti u obzir cijenu goriva koju je potrebno platiti za alokaciju memorije i interakciju s memorijom.

Kako bi se podaci trajno pohranili i učinili dostupnim u budućnosti, koristi se pohrana ugovora (storage). Unošenje podataka u pohranu ugovora znatno je skuplje od unošenja podataka u privremenu memoriju ugovora. Opkodovi za interakciju s pohranom ugovora su:

- SLOAD - učitava riječ iz pohrane ugovora na stog,
- SSTORE - pohranjuje riječ u pohranu ugovora.

Za razliku od klasičnih virtualnih strojeva, EVM-a ne podržava standardne biblioteke. Na primjer, u JVM (Java Virtual Machine) mnoge uobičajene osnovne funkcije pohranjene su u standardnoj "rt.jar" biblioteci. Zbog tog nedostatka, prilikom kreiranja pametnih ugovora potrebno je svaki puta iznova implementirati neke standardne funkcionalnosti. Osim što je nepraktičan, takav način rada u kombinaciji s visokim naknadama čini izradu pametnih ugovora izrazito skupom. Dodatno, ako nisu dobro provjereni svi mogući ishodi pametnog ugovora, postoji opasnost od hakerskih napada.

Radi povećanja sigurnosti mreže i smanjenja cijene izrade pametnih ugovora, Ethereum mreža planira prelazak s EVM-a na EWASM - Ethereum WebAssambly[12]. Nakon prelaska na EWASM, developeri će za kreiranje pametnih ugovora moći vrlo jednostavno koristiti bilo koji programski jezik visoke razine koji bude podržan od strane Ethereum WebAssambly-a. Dotada, na raspolaganju imaju dva jezika visoke razine, Solidity i Vyper. U ovom radu objasnit ćemo programski jezik Solidity.

4.2 Programski jezik Solidity

Solidity je objektno orijentirani jezik visoke razine za implementaciju pametnih ugovora koji najviše sliči programskom jeziku C++, no sadrži i elemente JavaScripta, Pythona i drugih jezika. Utjecaj C++ jezika vidljiv je u sintaksi: deklariranje varijabli, for petlje, preopterećenje operatora i funkcija, koncept nasljeđivanja, ...

Kako bi se mogao izvršiti, programski kod napisan u Solidity programskom jeziku potrebno je kompajlirati u bytecode EVM-a. Za razliku od EVM-a, Solidity sadrži razne biblioteke koje olakšavaju programiranje. Vrlo je prilagođen korisniku, a kako bi se dodatno olakšao rad, postoje razna integrirana razvojna okruženja (IDE) za Solidity. U ovom radu koristit ćemo online IDE Remix.

[12]WebAssambly (skraćeno Wasm) je binarni format instrukcija za virtualni stroj baziran na stogu. Wasm omogućuje implementaciju na webu za klijentske i poslužiteljske aplikacije u bilo kojem programskom jeziku te se trenutno aktivno koristi u četiri najveća web preglednika (Mozzila, Google Chrome, Safari i Edge).

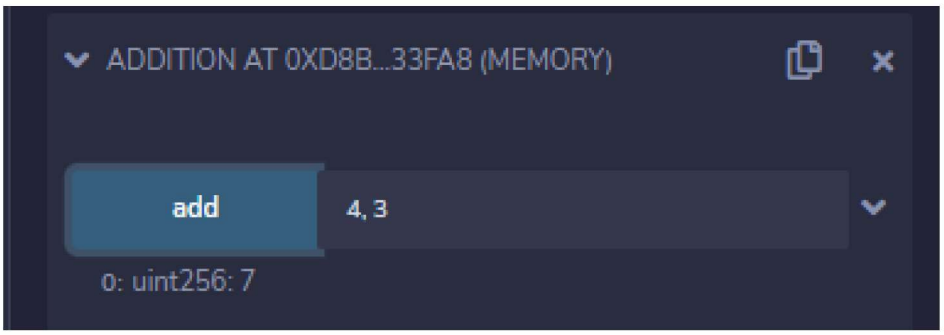
Kako bi se kompajlirani kod mogao izvršiti, pametni ugovor potrebno je poslati kao Ethereum transakciju bez adrese primatelja. Time pametni ugovor postaje dio Ethereum blockchaina.

Kada bismo željeli na blockchain mreži kreirati pametni ugovor koji bi u sebi sadržavao funkciju koja bi jednostavno zbrajala dva broja, implementacija unutar Solidity programskog jezika bila bi sljedeća:

```
1  // SPDX-License-Identifier: GPL-3.0
2
3  pragma solidity >=0.7.0 <0.9.0;
4
5  contract Addition {
6
7          function add(uint a, uint b) public pure returns (uint)
8      {
9
10          uint sum = a + b ;
11
12          return sum;
13      }
14 }
```

Kod 1: Primjer koda pisanog u Solidity programskom jeziku

Pomoću Remix IDE-a gornji kod možemo isprobati bez stvarnog objavljivanja ugovora na Ethereum mreži. Remix uz implementaciju i kompajliranje Solidity koda omogućuje i testno okruženje za pokretanje pametnih ugovora koristeći JavaScript Virtual Machine. Pomoću JavaScript VS-a simulira se rad Ethereum mreže, a pokretanjem koda 1 dobivamo:



Slika 9: Rezultat pokretanja koda 1 na Remix Javascript VS okruženju

Remix JavaScript VS je vrlo koristan kada želimo na brzinu napisati i testirati par linija koda, no simulirani blockchain kojeg nam omogućuje JavaScript VS nije praktičan za ozbiljnija testiranja pametnih ugovora. Prvenstveno jer se blockchain pokreće u web pregledniku korisnika te se sa svakim osvježenjem stranice pokreće nova simulacija blockchaina. Za testiranje pametnih ugovora u kojima vidimo kako naš ugovor utječe na cijelu mrežu koristimo testnu mrežu, tzv. TestNet.

4.3 TestNet

Postoji samo jedna glavna Ethereum mreža (MainNet) na kojoj se ugovori izvršavaju plaćanjem jedinica goriva pomoću Ethera. No, s obzirom na to da cijena Ethera nije mala, testiranje aplikacija na glavnoj mreži nije opcija. Umjesto glavne mreže, za testiranje pametnih ugovora koriste se različite testne mreže (TestNet) koje simuliraju glavnu. U trenutku pisanja ovoga rada, četiri najkorištenije testne mreže su:

- Rinkeby
- Ropsten
- Kovan
- Goerli.

Rinkeby TestNet kreiran je odvajanjem (forkom) od Ethereum MainNeta 2017. godine od strane neprofitne organizacije Ethereum Foundation. Za konsenzus mreže koristi "Proof of Authority" (PoA) model.

Za razliku od dokaza o radu koji zahtjeva velike napore računalnih resursa, PoA zahtjeva znatno manje napore, a samim tim omogućuje puno bržu validaciju blokova. U PoA modelu blokove potvrđuju čvorovi koji svojim identitetom jamče da su sve transakcije unutar potvrđenog bloka validne. Mreža unaprijed određuje mali broj čvorova mreže koji su svojom reputacijom zaslužili potvrđivati blokove. Ropsten, Kovan i Goerli za konsenzus mreže također koriste PoA model.

Na svakoj od gore nabrojanih testnih mreža i dalje je potrebno imati valutu kojom ćemo plaćati jedinice goriva utrošene za naše transakcije i naredbe koje će se izvršavati unutar pametnih ugovora. No, kako mreže za konsenzus koriste PoA model umjesto dokaza o radu, valute testnih mreža moguće je dobiti besplatno jednostavnim zahtjevom novčića od same mreže. Potrebno je samo unijeti adresu Ethereum računa i broj tokena koji želimo. Za što lakšu interakciju s Ethereum računima i računima ostalih kriptovaluta, postoje razni kriptonovčanici. U ovom radu koristit ćemo Ethereum novčanik MetaMask.

4.4 Bibilioteke

Kako je rečeno na početku potpoglavlja 4.2, Solidity nam pruža opciju korištenja različitih biblioteka. Nije potrebno pisati svaki pametni ugovor u svom projektu ispočetka, dovoljno je učitati biblioteku koja nam je potrebna i koristiti njene funkcionalnosti. Pokažimo to na primjeru kreiranja ERC-20 tokena.

4.4.1 Kreiranje ERC-20 tokena

Na Ethereum mreži token može biti virtualna reprezentacija raznih vrijednosti kao što su:

- nagradni bodovi na online platformi
- financijska imovina poput udjela u poduzeću
- razine i vještine lika unutar igrice
- udjeli u zlatnim polugama

- srećke lutrije
- različite kriptovalute...

Kako bi aplikacije koje imaju svoje tokene mogle pružati i primati usluge od drugih sustava, potreban je standard po kojem će se takvi tokeni kreirati. Jedan od takvih standarda je ERC-20. ERC-20 je standard za zamjenjive tokene, njime osiguravamo da svaka jedinica tokena kreiranog po tom standardu vrijedi jednako kao i svaka druga jedinica tog tokena.

ERC-20 standard osigurava sljedeće funkcionalnosti nad tokenom:

- prijenos tokena s jednog računa na drugi
- dohvaćanje trenutnog stanja tokena na računu
- dohvaćanje ukupnog broja tokena koji cirkuliraju na mreži
- mogućnost odobravanja trošenja tokena s nekog računa od strane drugog računa.

Ugovor kojim su tokeni pušteni u opticaj naziva se ERC-20 ugovor. Takav ugovor odgovoran je za praćenje kreiranih tokena na Ethereum mreži te mora sadržavati sljedeće metode i evente definirane od strane začetnika Ethereum mreže Buterina i Vogelstellera u radu [9]:

```
1 function name() public view returns (string)
2 function symbol() public view returns (string)
3 function decimals() public view returns (uint8)
4 function totalSupply() public view returns (uint256)
5 function balanceOf(address _owner) public view returns (uint256 balance)
6 function transfer(address _to, uint256 _value) public returns (bool
      success)
7 function transferFrom(address _from, address _to, uint256 _value) public
      returns (bool success)
8 function approve(address _spender, uint256 _value) public returns (bool
      success)
9 function allowance(address _owner, address _spender) public view returns (
      uint256 remaining)
```

Kod 2: Metode ERC-20 ugovora

```
1 event Transfer(address indexed _from, address indexed _to, uint256 _value)
2 event Approval(address indexed _owner, address indexed _spender, uint256
      _value)
```

Kod 3: Eventi ERC-20 ugovora

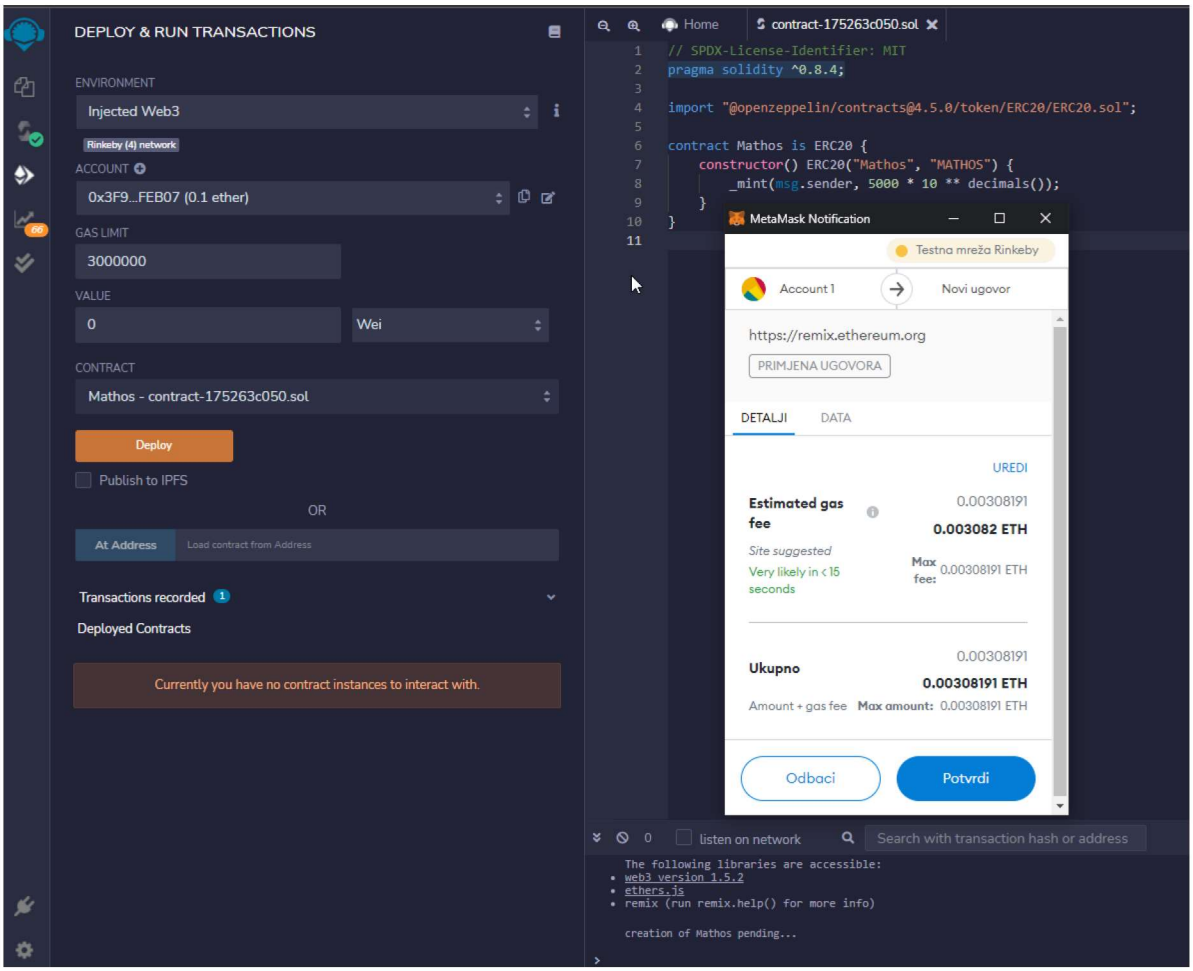
Kada bi željeli kreirati Mathos token kojim bi studenti mogli kupovati promo materijal Odjela, a tokene dijeliti studentima na temelju njihova postignuća na fakultetu, kreiranje istog bilo bi vrlo jednostavno.

Umjesto definiranja svih gore nabrojanih metoda i evenata, dovoljno je učitati poznatu OpenZeppelin biblioteku za kreiranje ERC-20 tokena, unijeti ime i simbol tokena te broj tokena koji želimo inicijalizirati - kao u sljedećem kodu.

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.4;

import "@openzeppelin/contracts@4.5.0/token/ERC20/ERC20.sol";

contract Mathos is ERC20 {

    constructor() ERC20("Mathos", "MATHOS") {

        _mint(msg.sender, 5000 * 10 ** decimals());

    }
}
```

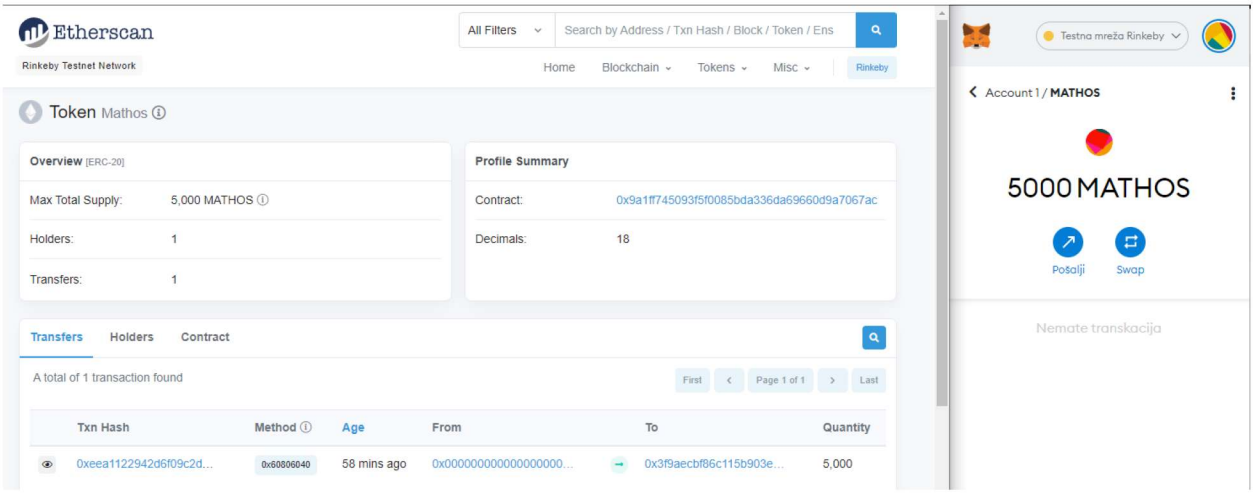
Kod 4: Kreiranje ERC-20 tokena

Sada u Remix IDE-u nakon kompajliranja gornjeg koda umjesto JavaScript VS-a odaberemo "Injected Web3" kao okruženje i povežemo se s našim MetaMask računom Rinkeby mreže (mogli smo koristiti bilo koji Ethereum TestNet). Klikom na akciju "Deploy" i potvrdom transakcije s našeg Rinkeby računa pametni ugovor će se objaviti na Rinkeby mrežu - slika 10.



Slika 10: Objavljivanje ERC-20 ugovora na Rinkeby TestNet-u

Transakcijom "0xeea1122942d6f09c2d13fc3ef9aaf296b69d9e1117486445539316b386afbdd2" kreiran je pametni ugovor "0x9a1ff745093f5f0085bda336da69660d9a7067ac", a s ugovorom i ERC-20 token Mathos - slika 11.



Slika 11: Mathos token - Etherscan i MetaMask

Na isti način možemo kreirati ERC-20 token na Ethereum MainNetu, naravno uz plaćanje jedinica goriva pravim Etherima.

Nakon kreiranja, s tokenima možemo vršiti sve operacije koje osigurava ERC-20 standard. Iako je metode moguće odmah isprobati u sučelju Remixa, željeli bismo da metode možemo izvršavati iz konkretne aplikacije ili web stranice. To je moguće pomoću ABI specifikacije.

4.5 ABI specifikacija ugovora

ABI (engl. Application Binary Interface) specifikacija ugovora je standardni način interakcije s pametnim ugovorima Ethereum mreže, kako za aplikacije izvan blockchaina, tako i u međusobnoj komunikaciji samih ugovora mreže. Nastaje kompajliranjem pametnog ugovora zajedno s bytecode-om. ABI je JSON datoteka koja opisuje pametni ugovor i sve njegove funkcionalnosti. Pokažimo kako pomoću ABI specifikacije možemo koristiti sve metode ugovora kojeg smo kreirali korištenjem OpenZepellin ERC-20 biblioteke.

4.5.1 Interakcija s pametnim ugovorom

ERC-20 ugovor kojim smo kreirali Mathos token sadrži 9 metoda potrebnih za praćenje i održavanje tokena na mreži. Kada bismo pojedine metode htjeli pozivati klikom na akcije web stranice, za interakciju s našim ERC-20 ugovorom potrebna nam je adresa ugovora i ABI specifikacija koja je nastala kompajliranjem ugovora. Kreirajmo web aplikaciju koja će nam to omogućiti. Prije samog početka rada, potrebno je instalirati web3.js JavaScript biblioteku. Biblioteka web3.js je kolekcija modula koji omogućuju razne funkcionalnosti specifične za interakciju s Ethereum mrežom. Nakon učitavanje biblioteke valja provjeriti podržava li web preglednik komunikacijski protokol EIP-1102[13]. Ako je EIP-1102 protokol podržan,

[13]Komunikacijski protokol između aplikacija i Ethereum mreže. Omogućuje korisnicima da odobre ili odbiju aplikaciji pristup do njihovih računa čime se korisnici osiguravaju od mogućih napada zlonamjernih aplikacija.

možemo kreirati web3 objekt pozivom `new` operatora. Metamask je osigurao API pomoću kojeg korisnici vrlo jednostavno mogu pristupiti svojim računima, a kako bi integrirali API u svoju aplikaciju dovoljno je u kodu naznačiti da ćemo koristiti `window.ethereum` Metamask pružatelj usluga.

Iz sigurnosnih razloga, korisnik mora aplikaciji dati odobrenje pristupu njegovom računu. Pozivom naredbe `window.ethereum.enable` pokreće se korisničko sučelje koje omogućuje korisniku da odobri ili odbije aplikaciji pristup računu. Za nesmetani rad aplikacije opisane naredbe pozivamo unutar asinkrone funkcije - kod 5.

```
async function loadWeb3() {
            //Provjera podrzivosti EIP-1102 protokola
            if (window.ethereum) {
                //kreiranje web3 objekta
                window.web3 = new Web3(window.ethereum);
            //omogucuje pristup racunu
             window.ethereum.enable();
            }
        }
```

Kod 5: Kreiranje web3 objekta

Nakon što smo kreirali web3 objekt, potrebno je povezati se s našim pametnim ugovorom kojeg smo prethodno kreirali na blockchainu. To činimo prosljeđujući ABI specifikaciju ugovora i njegovu adresu objektu `web3.eth.contract` - kao u kodu 6.

```
async function loadContract() {

        return await new window.web3.eth.Contract(abi, adresa ugovora);

        }
```

Kod 6: Dohvaćanje pametnog ugovora

Sada možemo koristiti sve metode našeg pametnog ugovora unutar naše aplikacije. Prije nego pozovemo neku od metoda ugovora, kreirajmo jednostavnu HTML stranicu koja će sadržavati tri gumba za pozive metoda ERC-20 ugovora, jedno tekstualno polje za unos ETH adrese i element u koji ćemo ispisivati status komunikacije s ugovorom - kod 7.

```
<h1>Mathos ERC-20 token</h1><br>
<button onclick="getName();">Ime tokena</button>
<button onclick="tokenSupply();">Ukupan broj tokena u cirkulaciji</button>
<div>
 <p>Unesite adresu ETH racuna za provjeru stanja.</p>
 <fieldset>
   <label for="wBalance">ETH Adresa</label>
   <input type="text" name="wBal" id="wBal" class="text" size="45">
   <button onclick="balanceOfAcc();">Stanje racuna</button>
 </fieldset>
</div>
<h5> Status: <span id="status">Ucitavanje...</span></h5>
```

Kod 7: HTML web stranice

Dodatno, dodajmo pomoćnu funkciju `updateStatus` kojom ćemo ispisivati povratne vrijednosti metoda pametnog ugovora unutar DOM elementa HTML-a i na konzoli - kod 8.

```
function updateStatus(status) {
    const statusEl = document.getElementById('status');
    statusEl.innerHTML = status;
    console.log(status);
    }
```

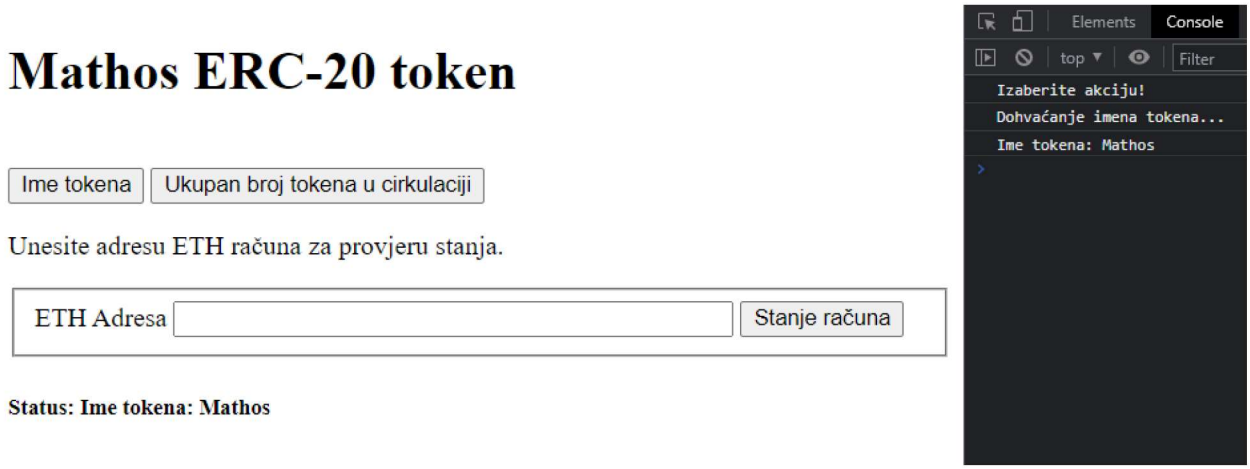
Kod 8: Pomoćna funkcija `updateStatus`

Pozovimo sada metodu `name` ERC-20 ugovora i ispišimo njenu povratnu vrijednost unutar konzole i DOM elementa html-a.

```
async function getName() {
            updateStatus('Dohvacanje imena tokena...');
            const tokenName = await window.contract.methods.name().call();
            updateStatus(`Ime tokena: ${tokenName}`);
        }
```

Kod 9: Poziv `name` metode ERC-20 ugovora

Jednostavan prikaz poziva metode `name` ERC-20 ugovora s web stranice vidimo na slici 12.



Slika 12: Dohvaćanje imena ERC-20 tokena

Jednako tako možemo pozvati metodu `totalSupply` ERC-20 ugovora koja će vratiti ukupan broj tokena u cirkulaciji, što je u ovom slučaju 5000. Valja napomenuti da se broj tokena izražava pomoću 18 decimalnih mjesta. Kod 10 prikazuje poziv metode `totalSupply`.

```
async function tokenSupply() {
        updateStatus('Dohvacanje ukupnog broja tokena u cirkulaciji...');
        const supply = await window.contract.methods.totalSupply().call();
        updateStatus(`Ukupan broj tokena u cirkulaciji: ${supply}`);
        }
```

Kod 10: Poziv `totalSupply` metode ERC-20 ugovora

Za razliku od gornjih metoda koje uvijek vraćaju istu vrijednost, metoda `balanceOf` vraća broj tokena koji se nalaze na adresi koja je proslijeđena kao parametar. Pošto se trenutno svi tokeni nalaze na adresi "0x3f9aecbf86c115b903e3e944a7d6ad7bffefeb07", metoda će za tu adresu vratiti 5000 (uz dodatnih 18 nula), a za sve ostale adrese vratit će 0. Poziv metode vidimo u kodu 11, a rezultat testiranja na slici 13.

```
async function balanceOfAcc() {
 updateStatus('Dohvacanje stanja racuna ...');
 var wAddress = document.getElementById("wBal").value;
 const balance= await window.contract.methods.balanceOf(wAddress).call();
 updateStatus(`Stanje ra una ${wAddress} je: ${balance}`);
   }
```

Kod 11: Poziv `balanceOf` metode ERC-20 ugovora

Slika 13: Interakcija s metodama ERC-20 ugovora

Cijeli kod prethodno opisanog primjera interakcije s ERC-20 ugovorom moguće je pronaći na [23].

Osim samog dohvaćanja informacija o stanju na blockchainu, pozivom različitih metoda ugovora možemo vršiti različite promjene na blockchainu. Konkretno, pozivom metoda `transfer` i `transferFrom` ERC-20 ugovora možemo prenijeti tokene s jedne adrese na drugu čime će se u pozadini izvršiti transakcija na blockchainu.

Kako smo naveli na početku potpoglavlja 4.4.1, kreiranje vlastitog tokena može poslužiti u razne svrhe, no primjena pametnih ugovora ne staje samo na tome. Pametni ugovori i blockchain tehnologija mogu poslužiti u raznim područjima: bankarski sektor, osiguranja, medicina, graditeljstvo, ... Sljedeće poglavlje posvetit ćemo konkretnim primjenama pametnih ugovora i blockchain tehnologije.

5 Primjena blockchain tehnologije i pametnih ugovora

Do sada smo govorili samo o pametnim ugovorima na Ethereum mreži. Ethereum je prva mreža koja je omogućila jednostavno kreiranje pametnih ugovora te je vodeća po udjelu projekata koji uključuju blockchain tehnologiju, no nije jedina mreža na koju se developeri mogu osloniti. Po uzoru na Ethereum naknadno su nastale razne blockchain mreže. Prema načinu pristupa mreži možemo ih podijeliti na javne (engl. public) i privatne blockchainove.

Kod javnih blockchainova svatko može pristupiti mreži dok se kod privatnih blockchainova pristup odobrava provjerom autentičnosti korisnika, svatko mora potvrditi svoj identitet i zadovoljiti unaprijed zadane uvjete kako bi pristupio mreži (npr. biti povezan na zadani VPN).

Jednako tako, prema razini prava koje korisnici imaju na mreži, blockchainove možemo podijeliti u dvije kategorije: permissioned i permissionless. Na permissionless mreži svaki čvor može vršiti sve akcije i sudjelovati u konsenzusu mreže, dok je na permissioned mreži za vršenje određenih akcija potrebno odobrenje od strane ostatka mreže. Ovisno o specifičnosti platforme, postoje različite razine ograničavanja. Sudionici permissioned mreže imaju mogućnost reguliranja tko može sudjelovati u konsenzus mehanizmu mreže, koji će čvorovi izvršavati obične transakcije, a koji pametne ugovore. Pri tome jedan ili više sudionika mreže ima ovlasti za upravljanje različitim razinama pristupa.

Kombiniranjem podjele prema načinu pristupa mreži i podjele prema razini prava dobivamo 4 kategorije:

- Javne permissionless mreže: Svatko može pristupiti mreži, postaviti pametni ugovor i potvrđivati blokove transakcija (Bitcoin, Ethereum).
- Javne permissioned mreže: Svatko može pristupiti mreži, no samo čvorovi koji zadovoljavaju određene uvjete mogu sudjelovati u konsenzus mehanizmu (Hyperledger Indy, EBSI).
- Privatne permissionless mreže: Mreži mogu pristupiti samo korisnici koji imaju odgovarajuće digitalne certifikate te su povezani na odgovarajući VPN. Svi članovi mreže imaju jednaka prava (privatni Enterprise Ethereum).
- Privatne permissioned mreže: Pristup mreži je ograničen te se za vršenje određenih akcija mora dobiti odobrenje (Hyperledger Fabric, Corda).

Pristup \ Prava	Permissionless	Permissioned
Javne	Bitcoin Ethereum	Hyperledger Indy EBSI
Privatne	Privatni Enterprise Ethereum	Hyperledger Fabric Corda

Slika 14: Kategorije blockchainova

Ovisno o tome u koju svrhu planiramo koristiti blockchain tehnologiju, izabrat ćemo jednu od četiri navedene kategorije. Tako ćemo u slučaju decentraliziranih financija (potpoglavlje 5.1.3) koristiti javnu permissionless mrežu, za upravljanje korisničkim identitetima (potpoglavlje 5.3) javnu permissioned mrežu dok ćemo za upravljanje poduzećem (5.5) odabrati privatnu permissioned mrežu.

5.1 Financijski sektor

Blockchain tehnologija se sve više i više integrira u različite sektore djelatnosti, no najveća primjena pametnih ugovora i blockchain tehnologije trenutno je u financijskom sektoru.

5.1.1 CBDC i prekogranična plaćanja

Pojava Bitcoina i različitih oblika kriptoimovine, posebice stablecoin novčića potaknula je centralne banke brojnih država svijeta na razvoj digitalnih valuta. Do sada su građani imali pristup novcu središnje banke samo u fizičkom obliku, digitalni oblik novca bio je rezerviran samo za poslovne banke. Digitalni oblik novca centralne banke koji je široko dostupan sveopćoj javnosti naziva se centralnobankarski digitalni novac ili CBDC[14].

Valja napomenuti kako CBDC može, ali ne mora biti baziran na blockchainu. Primjer CBDC-a baziranog na blockchain tehnologiji je DCash kreiran od strane Istočnokaripske valutne unije te se u sklopu pilot projekta koristi kao platno sredstvo na području osam karipskih zemalja. DCash platforma koristi Hyperledger Fabric Framework kao temeljnu blockchain tehnologiju pri čemu svi korisnici platforme moraju proći proces autentifikacije i autorizacije. Platforma se nedavno susrela s velikim problemima kada je gotovo dva mjeseca bila izvan uporabe zbog tehničkih poteškoća.

Države planiraju uvoditi digitalne valute iz različitih razloga te će vrlo vjerojatno prilikom razvoja svojih digitalnih valuta koristiti različite tehnologije i protokole. Te razlike mogle bi ograničiti interoperabilnost između različitih CBDC-ova što bi u konačnici moglo dodatno zakomplicirati prekogranična plaćanja. Taj problem već je razmotrila Banka za međunarodna poravnanja (skraćeno BIS[15]), što se može vidjeti u [20] i kao jedno od izglednih rješenje ponudila korištenje blockchain tehnologije i pametnih ugovora.

BIS-ovo navođenje blockchain tehnologije kao mogućeg rješenja prekograničnog plaćanja između CBDC-ova ne treba čuditi s obzirom da se blockchain tehnologija već uvelike koristi prilikom prekograničnih plaćanja. RippleNet, IBM World Wire, Onyx by J.P. Morgan, XinFin samo su neka od postojećih blockchain rješenja prekograničnog plaćanja. Prema broju financijskih institucija koje ih koriste još su daleko od tradicionalnih načina prekograničnog plaćanja. Ipak, s obzirom na znatno nižu cijenu naknada i brzinu transakcija, kao i smanjenje broja posrednika prilikom transakcija, izgledno je da će se s vremenom sve više financijskih institucija odlučiti za blockchain rješenja.

5.1.2 Financiranje trgovine

Tradicionalni model financiranja trgovine koji vode banke nije se značajnije mijenjao desetljećima. Banke ne mogu jednostavno proširiti svoju platformu kako bi je učinile dostupnom

[14]CBDC - akronim za "Central Bank Digital Currency".

[15]BIS - akronim od "Bank for International Settlements". BIS je međunarodna financijska institucija u vlasništvu središnjih banaka koja potiče međunarodnu monetarnu i financijsku suradnju. HNB je članica BIS-a od 1997. godine.

svim klijentima, dok mnoge tvrtke ne žele biti izložene riziku koji nosi međunarodno poslovanje bez bankovnih jamstava. Zbog toga je konzorcij velikih banaka Europe 2017. godine u suradnji s IBM-om osnovao platformu za financiranje međunarodne trgovine we.trade.

We.trade je platforma za financiranje trgovine temeljena na blockchainu koja okuplja sve strane uključene u trgovinu: kupca, prodavatelja, njihove banke i transportne tvrtke. Pritom se na platformi bilježi cijeli trgovinski proces, od narudžbe do plaćanja te jamči automatsko plaćanje pomoću pametnih ugovora kada se ispune svi ugovorni uvjeti. Tvrtke koriste we.trade digitalnu platformu kako bi poboljšali novčani tijek i digitalizirali svoje postojeće procese te za rješavanje izazova kao što su zakašnjelo plaćanje faktura, kibernetičke prijevare i slično.

Platforma je izgrađena na IBM blockchain platformi koja koristi Hyperledger Fabric Framework kao temeljnu blockchain tehnologiju. We.trade koristi značajku Hyperledger Fabric Frameworka pod nazivom kanali (engl. channels). Svaki kanal predstavlja privatni permissioned blockchain na kojem se blokovi transakcija prenose samo između čvorova koji su članovi kanala. Pri tome se osobni podaci nikada ne pohranjuju na blockchainu nego samo podaci o pravnim osobama koje sudjeluju u trgovini.

5.1.3 Decentralizirane financije

Decentralizirane financije ili DeFi je kolektivni naziv za sve financijske proizvode i usluge koje su izgrađene na blockchain tehnologiji. Trenutni financijski sustav je centraliziran, za bilo kakvu akciju na financijskom tržištu potrebno je prisustvo treće stranke u obliku banke, osiguravajućeg društva, burze. Moramo vjerovati financijskoj instituciji da će pravilno raspolagati našom financijskom imovinom i u samu stabilnost te institucije. Ako se institucija nađe u problemima, automatski smo i mi zahvaćeni time.

Za razliku od tradicionalnog centraliziranog financijskog sustava, DeFi nam omogućuje financijske proizvode i usluge bez centralne vlasti. Svatko može koristiti DeFi usluge direktnim pristupanjem tržištu u bilo koje doba dana. Svatko ima potpunu kontrolu nad svojom imovinom, nema centralne institucije o kojoj ovisi. Prilikom korištenja takvih usluga treba imati na umu da nedostatak centralne institucije znači nepostojanje korisničke podrške. Ako se korisnik nađe u problemu prepušten je sam sebi, nitko mu ne može pomoći.

Izravna razmjena i pružanje usluga vrši se pomoću blockchain peer-to-peer koncepta. Najveći dio DeFi platformi izgrađen je na Ethereum mreži, no zbog velike cijene koju je potrebno platiti prilikom svake transakcije na Ethereum mreži, sve je veći broj DeFi platformi izgrađenih na drugim mrežama. Uz blockchain infrastrukturu, DeFi platformama potrebna je digitalna valuta čija cijena ne oscilira brzo kao što je slučaj s Etherom. Zbog toga su kreirani različiti stablecoin novčići koji su fiksirani za neku fiat valutu ili imovinu.

Tako imamo novčiće fiksirane uz tečaj američkog dolara (USDT), tečaj eura (EURS) ili vrijednost zlata (PAXG). Prije konverzije fiat valute u njen stablecoin par valja provjeriti koliko je određeni stablecoin zaista stabilan, koje je pokriće njegove vrijednosti. U petom mjesecu 2022. godine algoritamski stablecoin UST izgubio je paritet s američkim dolarom te mu je vrijednost u svega par dana pala za preko 90%.

Najkorišteniji oblik DeFi usluga su decentralizirane mjenjačnice (DEX - engl. Decentralized Exchange). Decentralizirane mjenjačnice pomoću pametnih ugovora omogućuju korisnicima da kupuju, prodaju i razmjenjuju kriptovalute. Specifičnost decentraliziranih mjenjačnica je u tome što ne zahtijevaju nikakav oblik autorizacije, nema registracije ni potvrde identiteta

kao što je slučaj kod centraliziranih mjenjačnica. Umjesto centralne institucije pravila po kojima se vrši razmjena definirana su pametnim ugovorima. Kada se zadovolje unaprijed definirani uvjeti pametnih ugovora, ugovor se automatski izvršava. Najpoznatija decentralizirana mjenjačnica je Uniswap.

Uniswap platforma izgrađena je na Ethereum mreži i omogućuje korisnicima razmjenu ERC-20 tokena pomoću tzv. bazena likvidnosti (engl. liquidity pool). Kako bi omogućili razmjenu dva tokena potrebno je položiti određenu količinu oba tokena u bazen likvidnosti. Jednom kada se aktivira bazen likvidnosti omjer vrijednosti dvaju tokena unutar bazena likvidnosti mora ostati konstantan, a to je osigurano pomoću Automated Market Maker algoritma napisanog unutar pametnog ugovora. Pokažimo kako algoritam funkcionira na konkretnom primjeru.

Primjer 5 *Uprava Odjela za matematiku odlučila se za trgovanje Mathos tokenom na Uniswap decentraliziranoj mjenjačnici. U bazen likvidnosti položili su* 5000 *Mathos tokena i* 2 *Ethera. Omjer vrijednosti tokena unutar bazena likvidnosti mora uvijek biti* 50 : 50*. Ako uzmemo da* 2 *Ethera ukupno vrijede* 5000 USD*, onda i* 5000 *Mathos tokena vrijedi jednako toliko, tj. svaki Mathos token na početku vrijedi* 1 USD*. Dodatno, u svakom trenutku umnožak broja Mathos tokena i Ethera unutar bazena likvidnosti mora biti konstantan tj. mora zadovoljavati formulu*

$$X \cdot Y = k \tag{1}$$

gdje je k konstanta. U ovom slučaju imamo:

$$5000 \cdot 2 = 10000.$$

Kada bi netko odlučio kupiti 1000 *Mathos tokena, broj Ethera koje bi morao izdvojiti izračunao bi se uvrštavanjem novog broja Mathos tokena u jednadžbu* (1) *uz konstantan iznos k:*

$$(5000 - 1000) \cdot Y = 10000.$$

Iz prethodne jednadžbe slijedi da je $Y = 2.5$ *što odgovara broju Ethera koji bi trebali biti unutar bazena likvidnosti nakon razmjene. Kako je prije razmjene unutar bazena likvidnosti bilo samo* 2 *Ethera, jasno je da kupac za* 1000 *Mathos tokena mora izdvojiti* 0.5 *Ethera.*

Nakon razmjene unutar bazena likvidnosti imat ćemo 4000 *Mathos tokena i* 2.5 *Ethera. Ova razmjena nije utjecala na globalnu cijenu Ethera, ali je značajno utjecala na cijenu našeg Mathos tokena. Sada* 4000 *Mathos tokena vrijedi jednako kao* 2.5 *Ethera, što je* 6250 USD*, tj. svaki Mathos token vrijedi* 1.5625 USD*. Kada bi kupac ponovno htio kupiti* 1000 *Mathos tokena, kako bi zadovoljio jednadžbu* 1 *morao bi položiti dodatnih* 0.833 *Ethera u bazen likvidnosti. Nasuprot tome, kada bi netko zamijenio svoje Mathos tokene nazad u Ethere, cijena Mathos tokena bi se smanjila. Ovim primjerom pokazali smo kako Automated Market Maker algoritam osigurava stabilnost bazena likvidnosti na principu ponude i potražnje.*

Svatko može postaviti svoj ERC-20 token na Uniswap i time omogućiti trgovanje tokenom. No, upravo je to najveća mana Uniswapa i decentraliziranih mjenjačnica. Naime, nitko ne provjerava tokene kojima se trguje na platformi tako da su pokušaji prevare dosta česti. ERC-20 standard ne jamči jedinstvenost imena tokena. Kada smo kreirali Mathos token u potpoglavlju 4.4.1, niti u jednom trenutku nije došlo do provjere postoji li već token istog imena na mreži. Ako bismo odlučili postaviti Mathos token na Uniswap mjenjačnicu, nitko nam ne može jamčiti da će naš token biti jedini token kojim će se trgovati na Uniswapu pod tim imenom. Netko bi mogao zlonamjerno kreirati bezvrijedan token istog imena i

time zavarati buduće kupce Mathos tokena. Zbog toga je prije bilo kakve razmjene na decentraliziranim mjenjačnicama potrebno usporediti adresu ugovora koja je navedena na mjenjačnici s adresom ugovora kojim je token prvotno kreiran (provjerom na Etherscanu kao na slici 11).

Uz decentralizirane mjenjačnice, DeFi platforme omogućuju štednju, pozajmljivanje i zaduživanje u kriptovalutama, trgovanje tokeniziranim dionicama i druge financijske usluge. Kako je DeFi u razvoju te nijedan projekt ne jamči potpunu sigurnost, nastale su razne decentralizirane platforme na kojima možemo osigurati svoju kriptoimovinu od mogućih prevara ili grešaka unutar pametnih ugovora.

5.2 Blockchain u zdravstvenom sustavu

Jedno od glavnih područja potencijalne primjene pametnih ugovora vezano je uz zdravstvenu skrb i kontrolu pristupa medicinskoj dokumentaciji. Blockchain tehnologija može u značajnoj mjeri olakšati interoperabilnost zdravstvenog sustava na nacionalnoj razini umrežavanjem elektroničkih zdravstvenih zapisa kao što su osobni zdravstveni kartoni pacijenta.

Osim što bi pridonijela smanjenju troškova zdravstvenog sustava, primjena blockchain tehnologije značajno bi smanjila mogućnost grešaka te bi osigurala dosljednost podataka. Čak i mala greška prilikom prijenosa informacija s jedne zdravstvene institucije na drugu može dovesti do pogrešnog načina liječenja i time ugroziti život pacijenta. Kada bi se takve informacije zapisivale na blockchain, a pristup informacijama kontrolirao pametnim ugovorima i digitalnim potpisima, medicinsko osoblje bi na raspolaganju imalo puno ispravnije podatke.

Dodatno, blockchain tehnologija mogla bi se primijeniti i u provjeri zdravstvenog osiguranja te automatskoj naplati medicinskih računa. U Sjedinjenim Američkim državama česte su situacije kada pacijentima nakon medicinskog tretmana stignu fakture na naplatu iako imaju puno zdravstveno osiguranje. Tada pacijent mora kontaktirati osiguravateljsku kuću s kojom ima sklopljenu policu zdravstvenog osiguranja i zatražiti povrat troškova. Opisana procedura mogla bi se automatizirati primjenom pametnih ugovora kojima bi se naplata automatski preusmjerila na osiguravateljsku kuću.

Zbog visoke cijene implementacije malo je vjerojatno da će u skoro vrijeme blockchain tehnologija postati dio javnog zdravstva, no kada su u pitanju privatne klinike i poliklinike već postoje brojna rješenja. Jedno od njih nudi Medicalchain platforma izgrađena na permissioned Hyperledger Fabric blockchainu. Putem Medicalchain platforme pacijenti određuju tko će i kada vidjeti informacije iz njihovih osobnih zdravstvenih kartona. Uz sigurnu razmjenu podataka platforma omogućuje pacijentima kontakt s njihovim doktorima putem video konzultacija.

Samo prošle godine, u SAD-u je više od 40 milijuna kartona pacijenata kompromitirano zbog raznih incidenata, kako slučajno tako i namjerno putem hakerskih napada. Platforma Medicalchain smanjuje mogućnost krađe zdravstvenih podataka pacijenata, no bolnice nisu jedino mjesto u kojem su naši osobni podaci izloženi mogućem napadu. Vlasti pokušavaju kontrolirati situaciju raznim zakonima i uredbama o privatnosti kao što je opća uredba o zaštiti podataka Europske unije (GDPR), no curenju osobnih podataka teško je stati na kraj. Blockchain tehnlogija nudi rješenje i tog problema te se sve češće integrira u različite sustave upravljanja korisničkim identitetima.

5.3 Upravljanje korisničkim identitetima

Svatko od nas ima pravo na zaštitu osobnih podataka i sam odlučuje kome će i u kojoj mjeri podijeliti svoje osobne podatke. No, često smo prisiljeni otkriti više osobnih podataka nego što bismo željeli. Nerijetko pružateljima usluga te informacije služe samo kao potvrda da možemo koristiti njihove usluge, npr. otkrivanje datuma rođenja kako bi potvrdili da smo punoljetni. Stoga su razvijene različite metode pomoću kojih možemo potvrditi istinitost neke tvrdnje bez otkrivanja podataka koji to dokazuju. Takve metode nazivaju se dokazi nultog znanja (engl. Zero-knowledge proofs). Mreže koje koriste autentikaciju protokolom nultog znanja osiguravaju korisnicima potpunu anonimnost. Najpoznatije mreže toga tipa u trenutku pisanja ovoga rada su Zcash i Monero.

Potpuna anonimnost prilikom korištenja različitih servisa pojedincu može zvučati sjajno, no vlasti se ne bi s tim složile. Blockchainovi koji omogućuju potpunu privatnost nisu u skladu s propisima brojnih država kada je u pitanju politika protiv pranja novca. Ako na mreži ne postoji način kako uči u trag transakcijama, mreža bi se mogla koristiti u ilegalne svrhe kao što su trgovanje drogom, kockanje i pranje novca. Stoga su takvi blockchainovi zabranjeni u brojnim zemljama svijeta.

Ipak, vlasti raznih zemalja svijeta sve više ulažu u istraživanja mogućih primjena blockchain tehnologije. Sve češće krađe identiteta i iskorištavanje privatnih podataka građana prilikom različitih online aktivnosti primorale su njemačke vlasti da podrže IDunion projekt baziran na Hyperledger Indy blockchain mreži. Kako bi korisnicima osigurao suverenitet nad vlastitim podacima, IDunion stvara otvoren i siguran ekosustav za digitalne identitete. Zadatak IDunion projekta je migrirati centralizirane sustave upravljanja identiteta ka decentraliziranom samosuverenom upravljanju digitalnim identitetima za ljude, organizacije i strojeve.

Samosuvereni identitet ili SSI (engl. Self-sovereign identity) je model za upravljanje digitalnim identitetima u kojem pojedinac ili tvrtka imaju isključivo vlasništvo i kontrolu nad svojim računima i osobnim podacima. Uz samosuvereni identitet, korisnici imaju potpunu kontrolu nad načinom na koji se njihovi osobni podaci čuvaju i koriste. Nema centralne institucije koja izdaje vjerodajnice kao što je slučaj kod klasične izrade osobne iskaznice, vozačke dozvole ili studentske iskaznice. Nema čekanja i nepotrebnih troškova kao ni straha od gubitka vjerodajnice. Samosuverene vjerodajnice mogu se provjeriti bilo gdje i u bilo koje vrijeme. Dodatno, nema potrebe za stvaranjem više različitih računa za svaku pojedinu online uslugu koju koristimo, SSI omogućuje korisnicima da vrlo jednostavno potvrde svoj identitet na više različitih platformi i lokacija.

Projekt IDunion kroz sljedeće tri godine testirat će specifične slučajeve upotrebe i pokušati implementirati SSI tehnologiju u svakodnevni život njemačkog stanovništva. No, SSI rješenja bazirana na blockchain tehnologiji već se uvelike koriste u razne svrhe, jedno od takvih rješenja implementirano je u Turskoj od strane turske vlade i nizozemske kompanije Tykn.

Nakon rata u Siriji i migrantske krize, tursko Ministarstvo vanjskih poslova u suradnji s Tykn kompanijom pokrenulo je platformu za decentralizirani digitalni identitet kako bi optimiziralo i ubrzalo proces izdavanja dokumentacije radnih dozvola za izbjeglice. Turska je tim potezom željela omogućiti financijsku neovisnost tri milijuna izbjeglica u zemlji kojima je većina dokumenata uništena, kao i institucije koje su ih izdale. Sličan problem s dokazivanjem identiteta prilikom korištenja mikrokredita u Africi pokušava se riješiti pomoću Kiva protokola.

U dijelovima Afrike za većinu financijskih institucija, operativni troškovi upisivanja i provjere novih korisnika nadilaze očekivane prihode od pružanja usluga korisnicima s nižim prihodima i klijentima koji prethodno nisu koristili bankovne usluge. Financijske institucije nemaju povjerenja u povijesne podatke ili ne mogu sa sigurnošću provjeriti istinitost podataka.

Zbog toga su pojedinci primorani uzimati mikrokredite s velikim kamatama kako bi financirali školovanje djece, otvaranje obrta i slične životne situacije unatoč dobroj kreditnoj povijesti. U rujnu 2019. godine vlada Sijera Leone u suradnji s UN-om pokrenula je nacionalnu platformu za digitalni identitet temeljen na Kiva protokolu. Pomoću nacionalne platforme građani mogu jednostavnim otiskom palca i svojim nacionalnim identifikacijskim brojem otvoriti račun ili mu pristupiti u bilo kojoj financijskoj instituciji u zemlji.

Prethodno opisani primjeri iz Turske i Sijera Loene samo su neki od oblika korištenja decentraliziranih identifikatora (skraćeno DID). Primjena DID-a planirana je i u Hrvatskoj u sklopu Europskog blockchain partnerstva (skraćeno EBP).

Dvadeset sedam zemalja Europske unije (uz Norvešku i Lihtenštajn) 2018. godine potpisalo je deklaraciju o kreiranju Europskog blockchain partnerstva i time se obvezalo da će zajednički raditi na razvoju strategije Europske unije o blockchainu. Od 2020. godine zajedno s Europskom komisijom, EBP je pokrenula inicijativu za izgradnju europske blockchain infrastrukture za javne usluge (EBSI). Četiri primarna cilja EBSI inicijative su:

- ESSIF[16] - implementirati jedinstveni SSI model za cijelu Europu.
- Digitalne diplome - omogućiti građanima kontrolu nad svojim digitalnim obrazovnim vjerodajnicama.
- Praćenje dokumentacije - automatiziranje provjere podataka.
- Pouzdano dijeljenje podataka - omogućiti građanima sigurno dijeljenje osobnih podataka carinskim i poreznim tijelima Europske unije.

Gornja rješenja implementirat će se nad javnim permissioned blockchainom, no valja napomenuti kako osobni podaci neće biti objavljeni direktno na blockchain mreži te će sustav biti usklađen sa svim propisima Europske unije kao što su GDPR i eIDAS[17].

5.4 Upravljanje lancem opskrbe (SCM)

Za razliku od osobnih podataka koji nikako ne bi trebali biti objavljeni na javnom mjestu te ih nerado dijelimo s drugima, poželjno je da podaci o proizvodima budu što transparentniji. Htjeli bismo što jednostavnije doći do što više informacija o nekom proizvodu kojeg koristimo. No procesom globalizacije postalo je gotovo nemoguće ući u trag podrijetlu svih sastavnica pojedinih proizvoda, npr. mobilnog telefona. Primjena blockchain tehnologije i pametnih ugovora mogla bi značajno doprinijeti rješenju ovog i sličnih problema upravljanja lancem opskrbe[18].

[16] ESSIF - engl. European Self Sovereign identity framework

[17] eIDAS je uredba Europske unije o elektroničkoj identifikaciji i uslugama povjerenja za elektroničke transakcije na jedinstvenom tržištu Europske unije.

[18] Upravljanje lancem opskrbe ili SCM (eng. Supply Chain Management) je rukovanje cjelokupnim tijekom proizvodnje robe ili usluge od početnih sirovina pa sve do isporuke konačnog proizvoda potrošaču.

U idealnom scenariju, kada bi svi sudionici proizvodnog procesa zapisivali potrebne podatke na blockchain, mogli bismo lako ući u trag svim komponentama gotovog proizvoda. Naravno, malo je vjerojatno da će se to dogoditi u složenom proizvodnom procesu kao što je proizvodnja mobilnog telefona. No kada su u pitanju prehrambeni proizvodi, američki lanac maloprodaje Walmart već je dokazao prednosti detaljnog praćenja proizvoda putem blockchaina.

Kada dođe do pojave zarazne bolesti koja se prenosi hranom, npr. salmonele ili E.coli, može potrajati danima, čak i tjednima da se pronađe njezin izvor. Sve dok istražitelji točno ne odrede s kojih farmi je potekla bolest, vlasti najčešće savjetuju potrošače da izbjegavaju proizvode uzgojene u cijelom području na kojeg sumnjaju, a svi postojeći proizvodi s tog područja moraju se uništiti. Osim financijske štete nanesene proizvođačima, distributerima i lancima maloprodaje, prisutna je i opasnost po zdravlje potrošača ako se istragom ne utvrdi pravi izvor bolesti. Sve je to navelo Walmart na praćenje prehrambenih proizvoda zapisivanjem podataka na blockchainu. Takvim sustavom praćenja, otkrivanje porijekla određenih prehrambenih proizvoda svedeno je s prosječnih sedam dana na tek par sekundi. Dodatno, potrošači mogu biti sigurni da je porijeklo proizvoda uistinu ono koje piše na deklaraciji.

5.4.1 Upravljanje rizikom opskrbnog lanca (SCRM)

Uz porijeklo proizvoda, potrošači često nisu sigurni je li proizvod koji kupuju zaista te robne marke ili je riječ o krivotvorini. Prema nekim istraživanjima, ukupni iznos krivotvorina samo prošle godine nanio je proizvođačima štetu veću od dva trilijuna američkih dolara. Rješenje tog problema ponudila je VeChain platforma. VeChainovo rješenje omogućuje brendovima da digitaliziraju proizvode na blockchainu uspostavljanjem veze između fizičkog proizvoda i jedinstvenog blockchain identiteta pomoću pametnih NFC oznaka. Pomoću takvog sustava moguće je jednostavnim skeniranjem proizvoda i provjerom informacija na blockchainu utvrditi je li riječ o pravom proizvodu ili krivotvorini.

Iz VeChaina dolazi rješenje još jednog oblika prevare, a radi se o neovlaštenom mijenjanju brojača kilometara na osobnim automobilima. Prema Europskom parlamentu, zbog neovlaštenog mijenjanja brojača kilometara, kilometraža je smanjena na gotovo 50% rabljenih automobila kojima se trguje u EU, čime je na prijevaru cijena pojedinog vozila u prosjeku porasla od 2.000 do 5.000 eura. Rješenje tog problema VeChain je predstavio u obliku digitalne putovnice vozila. Digitalna putovnica vozila pohranjuje se na blockchainu te omogućuje decentraliziranu razmjenu podataka između proizvođača automobila, servisera, osiguravajućeg društva, banke i samog vlasnika vozila. Uz kilometražu, prate se podaci o obavljenim popravcima, izmjenama dijelova, police osiguranja, sudjelovanja u nesreći, itd.

5.4.2 Pomorsko osiguranje

Prekidi opskrbnog lanca uzrokuju velike troškove poduzećima, posebice ako proizvod nikada ne stigne na predviđenu adresu. Veliki udio prekida opskrbnog lanca uzrokovan je nesretnim događajima prilikom pomorskog prijevoza robe i tereta. Pomorsko osiguranje pokriva gubitak ili oštećenje brodova, tereta, terminala i svakog transporta kojim se imovina prenosi, nabavlja ili drži između mjesta polaska i krajnjeg odredišta. No pomorsko osiguranje nije se posebno mijenjalo proteklih tristo godina otkada se primjenjuje. Zbog administrativnog opterećenja i papirologije sustav je poprilično neučinkovit. Svi ugovori potpisuju se više puta, idu od broda do broda, luke do luke. Kao potencijalno rješenje tog problema nameće se korištenje pametnih ugovora i blockchaina.

Insurwave je softverska platforma koja povezuje osiguranike, osiguravatelje i reosiguravatelje te podržava administraciju i servisiranje posebnih ugovora o osiguranju. Razvijena je od strane tvrtki EY i Guardtime, a koristi blockchain tehnologiju temeljenu na Microsoft Azureu. Kao pilot projekt odlučili su zadovoljiti potrebe pomorskog osiguranja jednog od najvećih operatora kontejnerskih linija i plovila u svijetu, brodarske tvrtke Maersk.

Sve informacije o statusu plovidbe pojedinog broda pratit će se putem blockchaina. Cijeli proces obavještavanja, potpisivanja dokumenata i izdavanja potvrde će se automatizirati. Pametni ugovori automatski će izdavati potvrde i fakture ovisno o GPS lokaciji broda.

5.5 ERP i CRM sustavi

U prethodnom poglavlju naveli smo neka od postojećih blockchain rješenja integriranih u sustav upravljanja lancem opskrbe. Logično je zapitati se može li se blockchain tehnologija integrirati i u ERP i CRM sustave.

ERP (Enterprise resource planning) je sustav koji tvrtkama omogućuje da upravljaju i integriraju važne dijelove svojeg poslovanja: proizvodnju, usluge, lanac opskrbe, financije, ljudske resurse. Blockchain tehnologija i ERP sustavi imaju dosta toga zajedničkog. Oboje osiguravaju jedinstveni izvor informacija te teže automatizaciji procesa uz što veću sigurnost podataka. Blockchain tehnologija i pametni ugovori mogli bi dodatno unaprijediti postojeće ERP sustave u pogledu transparentnosti, sigurnosti i brzine.

ERP sustavi uvelike olakšavaju internu razmjenu podataka između različitih odjela poduzeća, no kada je riječ o business to business (B2B) komunikaciji između dvije tvrtke, ostalo je još dosta prostora za napredak. Razni servisi i B2B protokoli kao što je EDI omogućuju razmjenu podataka između različitih tvrtki, no takvi protokoli često zahtijevaju posrednika, ne jamče visoku sigurnost te nerijetko zakažu. Alternativa se nazire u permissioned blockchain mrežama. Umjesto da tvrtke međusobno šalju i primaju podatke pomoću raznih protokola i servisa te naknadno usklađuju svoj ERP sustav s podacima koje su primile, mogle bi bilježiti dio informacija na zajedničkom permissioned blockchainu. Osim što bi takav blockchain osigurao jedinstveni izvor podataka uz visoku sigurnost, procesi između dvije tvrtke mogli bi se dodatno ubrzati automatizacijom pomoću pametnih ugovora.

Poslovni procesi sve su složeniji te samim tim potreba za ERP sustavima sve više raste. S porastom potražnje rastao je i broj pružatelja ERP usluga. Velike kompanije koje su nastale spajanjem i akvizicijama više različitih poduzeća nerijetko posluju na više različitih ERP sustava. U takvim okolnostima zatvaranje financijskih izvješća nije jednostavno te iziskuje puno vremena. Rješenje tog problema za kompaniju General Electric ponudio je Oracle u obliku međukompanijske distribuirane računovodstvene knjige. Pomoću blockchain mreže i aplikacija kojima upravljaju pametni ugovori, povezali su nezavisne ERP sustave te omogućili djelatnicima General Electrica da međukompanijskim podacima pristupe preko jednostavnog web sučelja. Uz prethodno opisano rješenje, Oracle svojim korisnicima nudi mogućnost korištenja blockchain usluga putem Oracle Blockchain Platforme.

Oracle nije jedini pružatelj ERP usluga koji svojim korisnicima nudi mogućnost integracije blockchain tehnologije. Tako vodeći pružatelj ERP usluga SAP svojim korisnicima nudi opciju integracije s MultiChain i Hyperledger Fabric mrežama, dok Microsoft omogućava pristup blockchainu preko Quorum Blockchain Servicea.

CRM (Customer relationship management) ili upravljanje odnosima s klijentima je pristup upravljanju tvrtke kroz interakciju sa sadašnjim i budućim kupcima. Vodeći svjetski pružatelj CRM usluga Salesforce pokrenuo je 2019. godine Salesforce Blockchain platformu izgrađenu na Hyperledger Sawtooth protokolu. Salesforce Blockchain platforma integrirana je direktno u Salesforce platformu te omogućuje CRM klijentima da vrlo jednostavno koriste benefite blockchain tehnologije i pametnih ugovora u svojem poslovanju.

Ovisno o svojim potrebama, klijenti mogu birati koji konsenzus protokol žele koristiti. API za pametne ugovore omogućuje developerima da kreiraju i odrede skup poslovnih pravila i time automatiziraju postojeće poslovne modele. Unazad tri godine otkako je Salesforce Blockchain platforma pokrenuta, nastalo je više uspješnih integracija blockchain tehnologije od strane Salesforce klijenata. Tako je državno sveučilište Arizona Salesforce Blockchain platformu iskoristilo za dijeljenje podataka o akademskim uspjesima, kompanija IQVIA za označavanje i regulaciju lijekova, dok je S&P Global Ratings iskoristio platformu kako bi smanjio vrijeme potrebno za pregled i odobravanje novih poslovnih bankovnih računa.

5.6 Blockchain as a service (BaaS)

Prednosti blockchain tehnologije su brojne, no blockchain rješenja nije tako jednostavno integrirati u postojeće aplikacije i servise. Zbog toga su vodeći svjetski pružatelji IT usluga svojim klijentima odlučili ponuditi gotovu blockchain infrastrukturu u obliku BaaS-a (Blockchain-as-a-Service). BaaS je temeljen na SaaS (Software-as-a-Service) modelu te radi na sličnom principu. Omogućuje korisnicima da iskoriste prednosti blockchain tehnologije bez razmišljanja o samoj infrastrukturi u pozadini. Pružatelj usluga održava infrastrukturu operativnom i jamči nesmetani rad, naravno uz određenu naknadu.

Tako IBM korisnicima putem IBM Blockchain Platforme nudi podršku prilikom integracije s Hyperledger Fabric blockchainom, Microsoft svojim korisnicima nudi opciju Quorum Blockchain Servicea (QBS), dok je Amazon u svoj Amazon Web Service (AWS) integrirao alat Amazon Managed Blockchain. Najveći pružatelj blockchain usluga je Amazon Web Service. Koliki utjecaj ima na blockchain tehnologiju govori nam podatak da se gotovo četvrtina svih punih čvorova Ethereum mreže pokreće na AWS-u. Pritom valja napomenuti kako uz Ethereum, AWS također podržava i Hyperledger Fabric. Pomoću Amazon Managed Blockchain alata kreirani su brojni projekti iz različitih sektora, od poljoprivrede do autoindustrije.

5.7 Ostale primjene

Iako primjena blockchain tehnologije neprestano raste, ostalo je još puno prostora za daljnji rast. Uz primjene koje smo već naveli u ovom poglavlju, blockchain tehnologija ima potencijalnu primjenu i u drugim sferama. Recimo, tržište nekretninama moglo bi blockchain tehnologiju iskoristiti za lakšu i bržu pretragu nekretnina, ubrzanje procesa dubinske analize (eng. due diligence) te upravljanje leasingom i tijekom novca (vidjeti [12]). U telekomunikaciji blockchain može zaustaviti roaming prevare i lažne pretplate te doprinijeti daljnjem razvoju 5G mreže (vidjeti [13]).

Korištenjem blockchain tehnologije mogu se minimizirati mogućnosti prevare i manipulacije sustavima glasovanja (vidjeti [5]), a u kombinaciji s IoT-om (Internet of Thing), umjetnom inteligencijom i računarstvom u oblaku (eng. cloud computing), blockchain tehnologija može značajno unaprijediti naš svakodnevni život (vidjeti [15]).

6 Zaključak

Satoshi Nakamoto 2008. godine predstavio je svijetu decentralizirani oblik novca Bitcoin i njegov blockchain. Od tada razvoj blockchain tehnologije ubrzano raste, posebice otkako su blockchain tehnologiju pojavom Ethereum platforme 2013. godine dodatno obogatili pametni ugovori. Pomoću pametnih ugovora moguće je automatizirati određene procese prema unaprijed dogovorenim uvjetima. U ovom radu objasnili smo kako funkcioniraju pametni ugovori na Ethereum mreži jer je ona prva koja je omogućila jednostavno kreiranje istih. Po uzoru na Ethereum, nastale su brojne druge mreže različitih tipova. Ovisno o tome u koju svrhu planiramo koristiti blockchain tehnologiju i pametne ugovore, odabrat ćemo tip mreže koji nam najviše odgovara.

Blockchain tehnologija u javnosti se često pogrešno poistovjećuje s kriptovalutama. Iako je nastao zajedno s Bitcoinom koji je znan kao najveća kriptovaluta, blockchain je puno više od toga. Korištenjem blockchain tehnologije i pametnih ugovora moguće je ubrzati protok podataka, smanjiti mogućnost pogrešaka te povećati transparentnost i sigurnost podataka. Moguće primjene su brojne, od financija i medicine pa sve do praćenja lanca opskrbe i upravljanja osobnim podacima.

Dok su mnogi blockchain projekti tek u povojima, postoje brojni projekti koji su već uspješno integrirani u razne poslovne procese, neki od njih spomenuti su u ovom radu. Prvenstveno je riječ o projektima izgrađenim na Hyperledger, Ethereum, IBM, ConsenSys Quorum i R3 Corda blockchain platformama. Ne sumnjam u primjenu i rast blockchain tehnologije i pametnih ugovora u narednim godinama. Mislim da će u naš svakodnevni život ponajprije doći putem projekata vladinih organizacija u pogledu upravljanja korisničkim identitetima te preko budućih proizvoda i usluga velikih IT korporacija koje ulažu značajna sredstva u razvoj blockchain tehnologije kao što su IBM, Oracle i Amazon.

Literatura

[1] V. Buterin, *A Next-Generation Smart Contract and Decentralized Application Platform*, 2013.
https://ethereum.org/en/whitepaper

[2] M. Dameron, *Beigepaper: An Ethereum Technical Specification*, 2019.
https://github.com/chronaeon/beigepaper/blob/master/beigepaper.pdf

[3] L. Hollander, *The Ethereum Virtual Machine: How does it work?*, 2019.
https://medium.com/mycrypto/the-ethereum-virtual-machine-how-does-it-work-9abac2b7c9e

[4] Y. Hu, *Blockchain-based Smart Contracts - Applications and Challenges*, University of New South Wales, 2019.
https://arxiv.org/pdf/1810.04699.pdf

[5] K. M. Khan, J. Arshad, M. M. Khan, *Secure Digital Voting System based on Blockchain Technology*, NED University of Engineering and Technology, University of West London,
https://core.ac.uk/download/pdf/155779036.pdf

[6] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008.
https://bitcoin.org/bitcoin.pdf

[7] N. Szabo , *Smart Contracts*, 1994.
https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

[8] L. Thomas, *An interpretation of the ethereum project yellow paper*,
https://ethereum.stackexchange.com/questions/268/ethereum-block-architecture

[9] F. Vogelsteller, V. Buterin, *EIP-20: Token Standard*, 2015.
https://eips.ethereum.org/EIPS/eip-20

[10] H. Volarević, M. Varović, *Osnove računovodstva*, MATE d.o.o., Zagreb, 2013.
https://www.bib.irb.hr/981264

[11] G. Wood, *Ethereum: A Secure Decentralised Generalised Transaction Ledger*, 2014.
https://gavwood.com/paper.pdf

[12] Deloitte, *Blockchain in commercial real estate*,
https://www2.deloitte.com/content/dam/Deloitte/us/Documents/financial-services/us-dcfs-blockchain-in-cre-the-future-is-here.pdf

[13] Deloitte, *How Blockchain can impact the telecommunications industry*,
https://www2.deloitte.com/content/dam/Deloitte/za/Documents/technology-media-telecommunications/za_TMT_Blockchain_TelCo.pdf

[14] Hyperledger, *Case Study: Walmart food supply chain*,
https://www.hyperledger.org/wp-content/uploads/2019/02/Hyperledger_CaseStudy_Walmart_Printable_V4.pdf

[15] IBM, *How blockchain adds trust to AI and IoT*,
https://www.ibm.com/blogs/blockchain/2020/08/how-blockchain-adds-trust-to-ai-and-iot/

[16] IDunion, *IDunion - About the project*,
https://idunion.org/projekt/?lang=en

[17] Oracle, *Modernizing Intercompany Billing using Hyperledger Fabric Distributed Ledger*,
https://analyticsanddatasummit.org/wp-content/uploads/2020/10/Crisci_8.20.20_Presentation.pdf

[18] PwC,*DeFi: Defining the future of finance*, 2021.
https://www.pwc.ch/en/publications/2021/defi-defining-the-future-of-finance-may-2021.pdf

[19] Tykn, *Self-Sovereign Identity*,
https://tykn.tech/self-sovereign-identity/

[20] Visa, Inc., *Cross-Border Payments for Central Bank Digital Currencies via Universal Payment Channels*,
https://www.bis.org/events/cpmi_ptfop/proceedings/paper14.pdf

[21] *Opcodes For The EVM*, 2021.
https://ethereum.org/en/developers/docs/evm/opcodes/

[22] *Blockchain As A Service: Enterprise-Grade BaaS Solutions*
https://101blockchains.com/blockchain-as-a-service/

[23] Gitlab
https://gitlab.com/proloscic/smart-contract-erc-20-token-interaction

Sažetak

Tema ovog diplomskog rada je blockchain tehnologija i pametni ugovori. Glavni cilj rada bio je objasniti kako funkcionira blockchain, što su pametni ugovori te koja je primjena blockchain tehnologije i pametnih ugovora. Strukturu blockchaina objasnili smo na primjeru Bitcoina - prve decentralizirane digitalne valute. Opisali smo rad Ethereuma, najveće platforme za izgradnju decentraliziranih aplikacija i pametnih ugovora te objasnili kako funkcionira Ethereum Virtual Machine (EVM). Na primjeru kreiranja ERC-20 tokena objasnili smo kako iskoristiti biblioteke Solidity programskog jezika za razvoj pametnih ugovora. Na istom primjeru pokazali smo kako objaviti pametni ugovor na Ethereum TestNetu te kako pomoću ABI specifikacije integrirati pametni ugovor u web aplikaciju. U nastavku smo naveli konkretne primjene blockchain tehnologije i pametnih ugovora u različitim sektorima: financijski sektor, medicina i zdravstvo, upravljanje korisničkim identitetima, upravljanje lancem opskrbe, ERP i CRM sustavi. Na samom kraju rada objasnili smo što je Blockchain as a Service (BaaS).

Ključne riječi: blockchain, pametni ugovori, Bitcoin, Ethereum, Solidity, Remix IDE, TestNet, Rinkeby, MetaMask, Merkleovo stablo, Merkleovo Patricia stablo, EVM, opkod, ERC-20, ABI specifikacija, CBDC, permissioned mreže, BaaS

Blockchain - Smart Contracts

Summary

The topic of this master thesis is blockchain technology and smart contracts. The main goal of this paper was to explain how blockchain functions, what are smart contracts, and what is the application of blockchain technology and smart contracts. We explained the blockchain structure on the example of Bitcoin, the first decentralized digital currency. We described the work of Ethereum which is the biggest platform for building decentralized applications and smart contracts, and explained the functioning of Ethereum Virtual Machine (EVM). We explained how to use Solidity programming language libraries to develop smart contracts by using the example of creating an ERC-20 token. With the same example we demonstrated how to publish a smart contract on Ethereum TestNet and how with the ABI specification we can integrate a smart contract into a web application. Furthermore, we have listed concrete applications of blockchain technologies and smart contracts in different sectors: financial sector, medicine and healthcare, customer identity management, supply chain management, ERP and CRM systems. In the end, we explained what Blockchain as a Service (BaaS) is.

Keywords: blockchain, smart contracts, Bitcoin, Ethereum, Solidity, Remix IDE, TestNet, Rinkeby, MetaMask, Merkle tree, Merkle patricia tree, Ethereum Virtual Machine, opcode, ERC-20, ABI specification, CBDC, permissioned blockchain, Blockchain as a Service

Životopis

Rođen sam 19. lipnja 1996. godine u Našicama. Nakon završene Osnovne škole "Matija Gubec" Magadenovac upisujem Srednju školu Isidora Kršnjavoga Našice, smjer Opća gimnazija. Preddiplomski studij matematike upisujem 2015. godine na Odjelu za matematiku u Osijeku te ga završavam 2019. sa završnim radom na temu RxJS pod mentorstvom izv.prof.dr.sc. Domagoja Matijevića i komentorstvom mag. Jurice Maltara. Iste godine upisujem diplomski studij matematike, smjer Matematika i računarstvo. Tijekom diplomskog studija radio sam preko studentskog ugovora kao aplikacijski inženjer u IT kompaniji Be-terna u Osijeku. Nakon svih položenih ispita odlazim na Erasmus+ praksu u kompaniju Be-terna Ljubljana gdje se dodatno usavršavam na poziciji aplikacijskog inženjera. Trenutno sam zaposlen na istoj poziciji u kompaniji Be-terna Zagreb.