SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Valentin Magdić

RAZVOJ INTELIGENTNOG VIŠEAGENTNOG SUSTAVA ZA ANALIZU TRŽIŠTA KRIPTOVALUTA

DIPLOMSKI RAD

Varaždin, 2018.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Valentin Magdić

Matični broj: 45298/16–R

Studij: Informacijsko i programsko inženjerstvo

RAZVOJ INTELIGENTNOG VIŠEAGENTNOG SUSTAVA ZA ANALIZU TRŽIŠTA KRIPTOVALUTA

DIPLOMSKI RAD

Mentor :

Izv. prof. dr. sc. Markus Schatten

Varaždin, rujan 2018.

Valentin Magdić

Izjava o izvornosti

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

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

U ovom radu pojasnit će se djelovanje višeagentnog sustava za analizu tržišta kriptovaluta. Opisivanju sustava prethodi teorijska osnova potrebna za razumijevanje domene rada. Opisat će se osnovni tehnički koncepti, prednosti i nedostaci te ekosustav izgrađen oko tržišta kriptovaluta. Pomoću niza primjera, obasnit će se koherentno djelovanje višeagentnog sustava kroz uloge pojedinačnih agenata, njihove međusobne interakcije te izvršavane funkcije u smjeru ispunjavanja zajedničke svrhe. Objasnit će se proces kontinuiranog praćenja i pohranjivanja parametara koji utječu na varijabilnost tržišta kriptovaluta. Pritom će se objasniti izračunavanje dodatnih parametara, interakcije s bazom podataka i vanjskim sustavima. Nad prikupljenim podacima provest će se detaljna analiza trenutnog stanja tržišta kriptovaluta te će se agregirani rezultati analize objasniti u sklopu rada.

Ključne riječi: kriptovalute, višeagentni sustav, tržište kriptovaluta, prikupljanje podataka, analiza, bitcoin, rudarenje weba, računalno učenje

Sadržaj

1. Metode i tehnike rada

Metode rada primjenjene u radu su:

- **Metoda analize** je postupak znanstvenog istraživanja raščlanjivanjem složenih pojmova, sudova i zaključaka na njihove jednostavnije sastavne dijelove i elemente. Analiza je proces redukcije nejednakoga na sve veću jednakost.
- **Metoda sinteze** je postupak znanstvenog istraživanja i objašnjavanja stvarnosti putem sinteze jednostavnih sudova u složenije. Sinteza je način sistematiziranja znanja po zakonistima formalne logike, kao proces izgradnje teorijskog znanja u pravcu od posebnog ka općem, odnosno od vrste prema rodu.
- **Povijesna metoda** uzima u obzir kronologiju, razvoj i uzročno-posljedičnu vezu o predmetu istraživanja.
- **Metoda klasifikacije** je podjela općega pojma na posebne, u okviru opsega pojma. Na temelju spoznaja o prirodi stvari, klasifikacija predstavlja sustave skupina predmeta ili raspodjele niza srodnih pojava.
- **Metoda deskripcije** je postupak jednostavnog opisivanja ili očitavanja činjenica, procesa i predmeta u prirodi i društvu te njihovih empirijskih potvrđivanja odnosa i veza, ali bez znanstvenog tumačenja i objašnjavanja. Ova se metoda primjenjuje u početnoj fazi znanstvenog istraživanja, a ima veću vrijednost ako je jednostavno opisivanje povezano s objašnjenjima o uočenim važnijim obilježjima opisivanih činjenica, predmeta i procesa, njihovih zakonitosti i uzročnih veza i odnosa.
- **Metoda kompilacije** je postupak preuzimanja tuđih rezultata znanstveno-istraživačkog rada, odnosno tuđih opažanja, stavova, zaključaka i spoznaja. Metoda kompilacije može se upotrijebiti u kombinaciji s drugim metodama u znanstveno-istraživačkom radu, tako da djelo nosi u što većoj mjeri osobni pečat autora kompilatora, koji će, uz osobni pristup pisanju znanstvenog ili stručnog djela korektno i na uobičajen način citirati sve ono što je od drugih preuzeto.
- **Metoda uzoraka** ispituje dio skupa na temelju slučajnog izbora jedinica. Bit metode uzoraka je stav da se relevantne statističke informacije o masovnoj pojavi mogu odrediti na temelju malog uzorka.
- **Metoda modeliranja** je postupak s pomoću kojega se generira sustav (model) koji može zamijeniti stvarnu pojavu i kojeg eksperimentalno ili simulacijom možemo istraživati te prenositi dobivene podatke sa modela na realnu pojavu. [**51**]

2. Uvod

U posljednjih dvadesetak godina, pojavom i rasprostranjenošću interneta, dogodio se ogroman tehnološki napredak koji predstavlja temelje većine današnjih aktivnosti. Internet se uvukao u sve sfere ljudske aktivnosti, od komunikacije i zabave, učenja i usavršavanja pa sve do trgovine i poslovanja, mikro i makro ekonomskih aktivnosti. Tehnološki napredak također je utjecao na financijski sustav, načine na koje pohranjujemo novac i plaćamo za dobra i usluge koji su nam potrebni. Dugi niz godina postojala je potreba za kvalitetnom implementacijom digitalnih valuta koje žive u virtualnom svijetu interneta, računala i ostalih pametnih uređaja. Digitalne valute imaju neobična imena, nepoznata pravila i za svoja korištenja zahtjevaju određena učenja, prilagođavanja i promjene postojećih navika i načina na koji gledamo na novac. Neke digitalne valute možda su nam već poznate u obliku žetona na nekoj od platformi za elektronička plaćanja, neovisno o tome radilo se o žetonima računalnih igara ili komercijalnim žetonima za kupnju fizičkih dobara i usluga. S druge strane, postoji novi trend pojave određene vrste digitalnih valuta, poznatijih pod imenom kriptovalute. Kriptovalute su u posljednje vrijeme privukle veliki interes javnosti naglim rastom cijena na tržištu kriptovaluta te osigurale medijsku pokrivenost diljem svijeta. Međutim, tehnološke inovacije i primjene kriptovaluta gotovo se i ne spominju. U radu će se detaljnije pokriti tehnička strana kriptovaluta, njeni prednosti i nedostaci te slučajevi korištenja koji proizlaze. Inovacije iza različitih kriptovaluta imaju potencijal promijeniti cjelokupnu ekonomsku aktivnost, od bezgraničnog slanja novca, mikrotransakcija, sigurnih online plaćanja pa sve do načina na koji provodimo poslovanje i potpisujemo ugovore.

Posljednja kriza rezultirala je preispitivanjem nacionalnih valuta i institucija koje njima rukovode, posebice financijskog sektora i vlade. Kriptovalute su se pojavile kao odgovor na posljednju krizu te su sa svim svojim inovacijama pokušale pružiti rješenje. Pritom je posebno naglašen nedostatak tradicionalnih financijskih institucija da pohrane depozite ljudi na siguran način. Drugi problem predstavljaju internacionalni transferi novca, koji su poprilično skupi i jako tromi, što predstavlja nepotrebne troškove za poslovne subjekte u sve povezanijem svijetu. U dodatku, postoji i čimbenik rastućeg vanjskog duga brojnih država zbog kojeg je buduća vrijednost nacionalnih valuta upitna. Zbog navedenih problema razvila se potreba za sigurnim novčanim sustavom, praktičnim za interakcije globalno sve povezanijih zemalja te neovisnim od tradicionalnih financijskih institucija i vlada. Bitcoin je tu pravovremeno uskočio kao rješenje, ali je medijski popraćen zbog financiranja sive ekonomije i kriminalnih radnji izvan dosega nadzornih institucija, regulatora i vlada. Zbog anonimnosti i tajnovitosti bitcoin sustava plaćanja, pretežito se koristio za trgovanje oružjima i drogama. Medijsku pozornost privukla je i činjenica mističnog kreatora, poznatijeg pod pseudonimom Satoshi Nakamoto. čiji identitet niti dan danas nije poznat. Od svog osnutka 2008. godine pa do 2013. godine, bitcoin je bio vrlo uspješan u ispunjavanju svoje svrhe. Poslije 2013. godine, dogodio se niz povezanih problema, od kraha tržišta, prijevara, sigurnosnih propusta pa sve do mogućih regulacija od strane zakonodavnih tijela. Uslijed toga, pojavio se velik broj novih kriptovaluta koje uz različite tehnološke implementacije i novitete nastoje ispraviti propuste bitcoina te pružiti alternativne svrhe korištenja. Neke od njih nastoje zamijeniti ili komplementirati bitcoin kao oblik sredstva razmjene, dok su druge usmjerene na spefičične slučajeve korištenja koji nemaju veze s novcem.

U ovom radu objasnit će se osnovni koncepti i terminologija potrebni za razumijevanje domene kriptovaluta, povijest njihova nastanka, tehnička pozadina te način na koji funkcioniraju. Navest će se njihove glavne prednosti i nedostaci te će se kroz primjere opisati nekoliko najpopularnijih alternativnih kriptovaluta različite namjene. Poznavanjem teorijske pozadine, opisat će se ekosustav izgrađen oko tržišta kriptovaluta, pomoćni alati i načini kako doći u posjed kriptovaluta. Osim navedenog, u ovom radu pojasnit će se djelovanje višeagentnog sustava za analizu tržišta kriptovaluta. Pomoću niza primjera, obasnit će se koherentno djelovanje višeagentnog sustava kroz uloge pojedinačnih agenata, njihove međusobne interakcije te izvršavane funkcije u smjeru ispunjavanja zajedničke svrhe. Objasnit će se proces kontinuiranog praćenja i pohranjivanja parametara koji utječu na varijabilnost tržišta kriptovaluta. Pritom će se objasniti izračunavanje dodatnih parametara, interakcije s bazom podataka i vanjskim sustavima. Nad prikupljenim podacima provest će se detaljna analiza trenutnog stanja tržišta kriptovaluta te će se agregirani rezultati analize objasniti u sklopu rada. Analizirat će se uloga bitcoina na financijskom tržištu, isplativnost investiranja u bitcoin te budućnost ove kriptovalute te cjelokupnog kripto tržišta. Cilj rada bio je izgradnja automatiziranog i samoodrživog višeagentnog sustava koji prikuplja, obrađuje i analizira podatke u stvarnom vremenu sa svrhom pronalaženja obrazaca ili trendova kretanja koji mogu postati temelj kvalitetnih odluka za trgovanje na tržištu kriptovaluta.

3. Kriptovalute

3.1. Povijest kriptovaluta

Pojava digitalnog novca usko je vezana uz razvoje u području kriptografije. Ono predstavlja fundamentalne izazove vezane uz korištenje bitova za predstavljanje vrijednosti koja se može razmjenjivati za dobra i usluge. Bilo je potrebno pronaći rješenje na dva temeljna pitanja: *"Možemo li vjerovati da je digitalni novac autentičan, a ne krivotvoren?"* te *"Možemo li biti sigurni da nitko drugi ne može tvrditi da taj novac pripada njima, a ne meni?"* (poznatije kao problem dvostrukog trošenja). Izdavatelji papirnatog novca rješavaju problem krivotvorenja korištenjem sofisticiranih tehnologija tiskanja. Fizički novac stoga rješava problem dvostrukog trošenja jer ne može biti na dvije lokacije istovremeno. Konvencionalni novac ujedno se nalazi i u digitalnom obliku. U ovom slučaju krivotvorenje i dvostruko trošenje nadziru centralni autoriteti koji imaju globalni pregled cirkulacije valute. Za digitalni novac kriptografija pruža temelj povjerenja u legitimitet korisnikove tvrdnje o posjedovanju određene vrijednosti. Kriptografski digitalni potpisi omogućuju korisniku da potpiše digitalnu imovinu ili transakciju kojom potvrđuje vlasništvo nad tom imovinom. S prikladnom arhitekturom, digitalni se potpisi ujedno mogu koristiti kako bi se adresirao problem dvostrukog trošenja. [**2**]

Nakon što je kriptografija postala široko dostupna 1980-ih godina, mnogi su istraživači pokušavali stvoriti digitalne valute uz pomoć kriptografije. Bili su to najraniji pokušaji razvoja digitalnih valuta i izdavanja digitalnog novca čija se vrijednost temeljila na nacionalnoj valuti ili plemenitom metalu kao što je zlato. Iako su rane digitalne valute radile, bile su centralizirane te su stoga bile ranjive na vanjske napade i podložne manipulacijama. Koristile su centralnu obračunsku kuću (eng. *clearinghouse*) kako bi isplatili sve transakcije u pravilnim intervalima, isto kao i tradicionalni bankarski sustav. Ovakvi oblici najranijih digitalnih valuta bili su predmet oštrih vladinih mjera te su sustavno izbrisane iz postojanja. Kako bi se izbjegli svi ovi problemi, bilo je potrebno stvorini decentraliziranu digitalnu valutu. Bitcoin je primjer po dizajnu potpuno decentralizirane valute oslobođene centralnog autoriteta ili mjesta kontrole koje bi se moglo napasti ili kompromitirati. Bitcoin predstavlja akumulaciju desetljeća istraživanja u kriptografiji i distribuiranim sustavima te objedinjuje četiri ključne inovacije: decentraliziranu čvor-čvor (eng. peer-to-peer) mrežu (bitcoin protokol), javnu transakcijsku knjigu (blockchain), decentralizirano matematičko i determinističko izdavanje valute (distribuirano rudarenje) te decentraliziran sustav verifikacije transakcija (transakcijska skripta). [**2**]

Povijest kriptovaluta oslanja se na dva temelja. Prvi je povijest istraživanja u domeni distribuiranih sustava, a drugi je povijest elektroničkih sustava plaćanja. U najranijim danima, navedena dva područja nisu imala mnogo poveznica. S druge strane, oba su područja vezana uz istraživanja i napretke u kriptogarfiji. Istraživanja u domeni elektroničkog plaćanja prvenstveno su vođena inovacijama u domeni asimetrične kriptografije. Bitcoin je povezao ta dva područja istraživanja te time postao prva decentralizirana kriptografska valuta. Preuzeti su i spojeni točno određeni dijelovi tehnologija navedenih područja u svrhu stvaranja bitcoina. Interes u izraživanjima distribuiranih sustava, sustava elektroničkih plaćanja i valuta naglo je porastao s popularnošću bitcoina. U nastavku poglavlja proći ćemo kroz kratku povijest krip-

tovaluta i početaka istraživanja, od originalnih ideja, razvoja koncepata i implementacija do trenutka nastanka bitcoina. Fokus će prvenstveno biti na tehničkim inovacijama i istraživanjima koja su postojala u promatranom vremenskom periodu. [**1**]

3.1.1. Problem dvostrukog trošenja

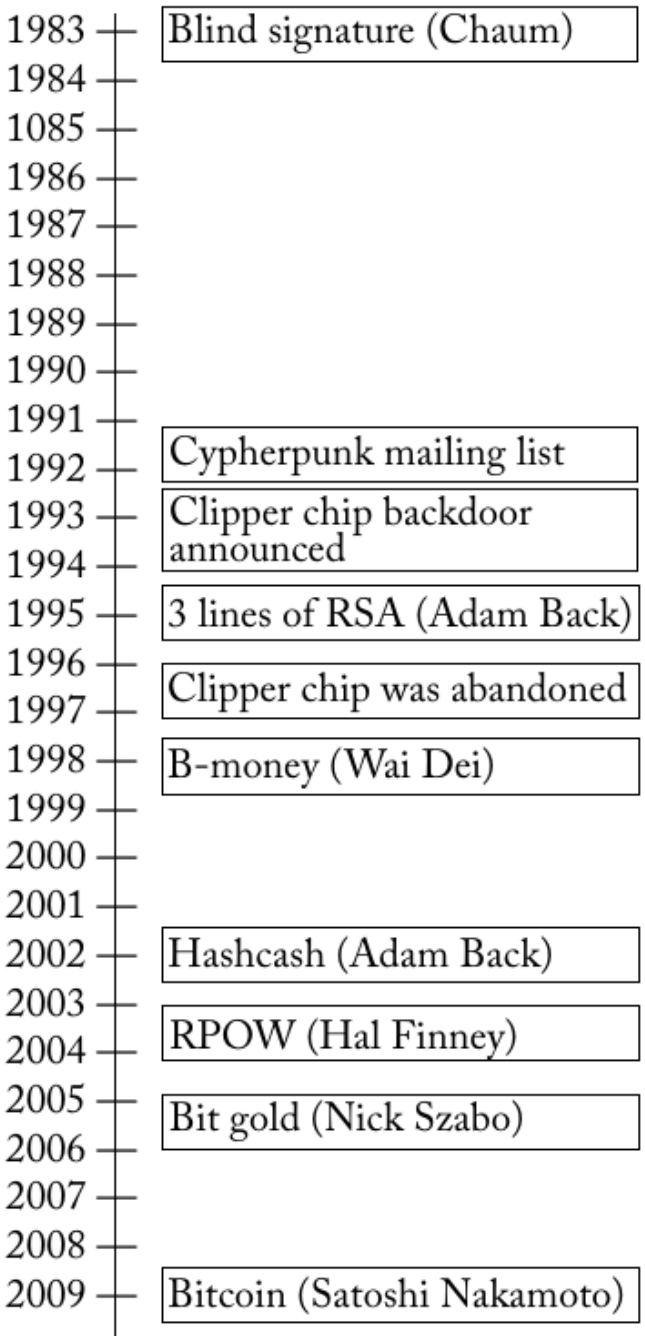
Problem dvostrukog trošenja dugo vremena je bio prepreka u razvoju decentraliziranih digitalnih valuta. Kako bi se ovaj problem pobliže dočarao, pretpostavimo da imamo tehnologiju koja nam omogućava da savršeno kopiramo novačanice neke valute na jednostavan i brz način. Kada bi samo jedna osoba posjedovala ovakvu tehnologiju, iako ilegalno, mogla bi kopirati dovoljno novca da zadovolji sve svoje potrebe. U slučaju široke rasprostranjenosti ovakve tehnologije, nitko više ne bi radio da zaradi novac. Sve dok postoji jedinica valute od koje bi otpočelo kopiarnje, on bi se mogao diplicirati nebrojeno mnogo puta. S druge strane, nitko nikome ne bi ništa htio prodati. Zašto prodati proizvod ili uslugu za nešto što možemo i sami replicirati? Ukratko, izgubila bi se funkcija novca, a ekonomija bi stagnirala. Novac koji je jednostavno kopirati ne bi bio dobar novac. [**3**, **7**]

Navedeni problem dovodi nas do digitalnih valuta. Digitalne su valute u suštini niz nula i jedinica pohranjenih na nekoj fizičkoj lokaciji. Neovisno o tome gdje se nalazi, taj dio podataka pože se kopirati nebrojeno mnogo puta bez da se ošteti originalni podatak. Kada bi novac bio samo elektronički impulsi, ne bi se previše razlikovao od gore navedenog primjera. Digitalna valuta koja bi služila kao novac mora riješiti problem dvostrukog trošenja. Najjednostavnije rješenje bilo bi održavanje glavne knjige, odnosno računa koji bi ispisao svaku jedinicu digitalne valute i pratio vlasnike te valute u bilo kojem trenutku u vremenu. Nakon transakcije, vlasništvo jedinice valute prebacilo bi se s kupca na prodavatelja. Održavanje glavne knjige i dalje ne rješava problem u potpunosti. U digitalnom svijetu, glavna je knjiga samo dio podatka koji se i dalje može jednostavno kopirati. Primjerice, neiskren kupac može kopirati glavnu knjigu prije transakcije te je povratiti na prethodno stanje koje ga izlistava kao vlasnika jedinice valute koju je upravo potrošio. Dakle, samo bi se zamijenio problem kopiranja digitalne valute s problemom održavanja integriteta glavne knjige. Situacija bi se promijenila kada bi se uveo povjerljiv posrednik koji bi bio zadužen za upravljanje glavnom knjigom. U tom slučaju, digitalna valuta bila bi centralizirana te bi povjerljiv posrednik bio jedini entitet koji bi imao prava mijenjati stanja glavne knjige. Sve transakcije morale bis e prijaviti povjerljivom posredniku. Prodavatelji bi uz njegovu pomoć provjeravali ima li potencijalni kupac dovoljno sredstava da izvrši transakciju. Digitalne valute koje se rukovode na ovaj način zapravo djeluju u praksi banaka i bankovnih računa. [**3**, **7**]

Postavljalo se pitanje, da li je moguće dizajnirati decentraliziranu digitalnu valutu koja bi mogla obnašati ulogu novca bez centraliziranog entiteta koji bi rukovodio svim transakcijama? Inicijalno, znanstvenici i informatičari smatrali su da bi to bilo veoma teško izvedivo ili jednostavno nemoguće. Početkom 1980-ih, problem digitalnog novca bio je raširen i poznat problem. Postojali su brojni prijedlozi, pristupi i pokušaji stvaranja digitalnog novca, o čemu više u nastavku. [**7**]

3.1.2. Počeci digitalnog novca

Vremenska linija inkrementalnih razvoja tehnologija digitalnog novca prikazana je sljedećom slikom:



Slika 1: Počeci digitalnog novca [**1**]

Povijest kriptovaluta počinje 1980-ih godina radom Davida Chauma. Zbog svog rada o kriptografskim primitivima slijepih potpisa često se referencira kao inovator sigurnog digitalnog novca. U tom radu, Chaum je predložio novu kriptografsku shemu skrivanja sadržaja poruke prije nego se potpiše, tako da ni potpisnik ne može odrediti sadržaj. Slijepi potpisi mogu se javno verificirati kao bilo koji drugi potpis. Chaumov predloženi pristup digitalnog novca omo-

gućio je korisnicima da anonimno troše digitalne valute, bez mogućnosti praćenja ili razotkrivanja identiteta. U kasnijim objavama, Chaum je unaprijedio ideju omogućavanjem izvanmrežnih (eng. *offline*) transakcija te dodavanjem mehanizama za otkrivanje dvostrukog trošenja (eng. *double spending*). Doduše, ovaj sustav i dalje je zahtjevao povjerljive posrednike za izdavanje i rukovođenje digitalnog novca. Kako bi komercijalizirao ovu ideju, Chaum je 1990. godine pokrenuo DigiCash. Bila je to prva generacija kriptovaluta, ali nije uspjela doseći širi auditorij unatoč brojnim pokušajima komercijalizacije. Glavni je cilj bio izgraditi sustav anonimnog elektroničkog novca za vlade i banke te okončati korupciju i organizirani kriminal. Glavna inovacija ovog sustava bila je mogućnost transporta informacija bežično, čime je bila pogodna za plaćanje cestarine, što je trebala biti prva primjena. Ideja DigiCasha privukla je Nizozemsku vladu, nekoliko banaka, Visu i Microsoft. Unatoč inicijalnom interesu, tvrtka koja je stajala iza DigiCasha ubrzo se raspala. Čak je postojala jedna banka u SAD-u koja je koristila DigiCash, no ubrzo je prekinula s njegovim korištenjem nakon raspada tvrtke. [**1**, **7**]

Uz sva dostignuća Davida Chauma na području kriptovaluta, rodio se pokret cypherpunk (eng. *cypherpunk movement*). Bila je to neformalna grupa koja je komunicirala putem vlastite elektroničke liste adresa (eng. *electronic mailing list*) te se zalagala za korištenje kriptografije i tehnologija koje podupiru privatnost. Uz brojne druge, rad Davida Chauma inspirirao je grupu aktivista da promoviraju korištenje takvih tehnologija. Prije toga, kriptografija nije bila javno dostupna korisnicima, već se ekskluzivno koristila u vojsci i obavještajnim agencijama. Cypherpunk pokret direktno je adresirao teme poput anonimnosti, pseudoanonimnosti, privatnosti komunikacije i skrivanja podataka, cenzuru i nadzor. Velik problem 1990-ih godina bio je Clipper čip. Proizvela ga je NSA s ugrađenim tajnim ulazom (eng. backdoor*backdoor*), što im omogućuje zaobilaženje sigurnosnih mehanizama te izravan pristup računalnom sustavu. Najveći kritičari ovog akta bili su upravo članovi cyberphunk pokreta. Stoga je, 1994. godine, Matt Blaze objavio rad o ranjivostima Clipper čipa. Zaključio je da čip prenosi informacije koje se mogu eksploatirati kako bi se kompromirao enkripcijski ključ. Daljnje ranjivosti otkrili su Moti Yung i Yair Frankel 1995. godine, nakon čega su objavljeni i brojni drugi radovi koji ukazuju na drastične sigurnosne propuste i moguće eksploatacije sustava koji sadrže Clipper čip. Aktivirale su se i druge aktivističke grupe, kao što je Elektronička granična zaklada (eng. *Electronic Frotier Foundation*), gdje su javno kritizirali Clipper čip i vladine pokušaje da limitiraju korištenje enkripcije. Ovaj vremenski period vezanih događaja poznatiji je kao kripto ratovi (eng. *crypto wars*). [**1**, **10**]

Uslijed svih navedenih događaja, Adam Back, kreator kriptovalute Hashcash, predstavio je korištenje ultra kompaktnog koda sa svojim trorednim RSA i datotekom potpisa u Perlu (eng. *Perl signature file*). Ljudi su njegov koncept ispisali na majice i nosili u znak protesta protiv kriptografskih regulacija vlade. Osvještavanjem javnosti o sigurnosnim propustima, Clipper čip nikad nije prihvaćen te je dizajn napušten 1996. godine. Unatoč tome, debate i tajna vrata kontrolirana od strane vlade i dan danas postoje. Snowden-ove objave 2013. godine potaknule su povećanu brigu javnosti o privatnosti koja je rezultirala u povećanoj potražnji za kriptografskim rješenjima. [**1**, **10**]

3.1.3. Pojava kriptovaluta

Prije nego su se pojavili bitcoin i njegovi sljedbenici kao prve decentralizirane kriptovalute, predlagani su brojni pristupi koji su unaprijedili inicijalnu ideju Davida Chauma. Koncepti u nastavku predstavljaju značajnija inkrementalna unaprijeđenja, ali s obzirom da su i dalje sadržavala centralizirane elemente, ne mogu se klasificirati kao potpuno decentralizirane valute. [**1**]

e-cash: Citibank je 1999-ih pokušao razviti sustav elektroničkog novca za vlastitu upotrebu. Nazvali su ga Elektronički Monetarni Sustav (eng. *Electronic Monetary System*), poznatiji kao e-cash. Bio je to drugi pokušaj komercijalnog razvoja decentralizirane digitalne valute nakon DigiCasha. Ova digitalna valuta imala je izuzetno neobično svojstvo. Naime, nakon nekog vremena bi novac nestao, a osoba koja posjeduje taj novac morala bi kontaktirati banku da ga zamijene. Navedenim svojstvom htjeli su izbjeći pranje novca. Bilo je nekoliko testnih pokušaja 1997. i 2001. godine, ali je potom ugašen. [**7**, **11**]

b-money: 1998. godine, Wei Dai objavio je rad za anonimni, distribuirani elektronički sustav plaćanja (eng. electronic cash system) pod imenom b-money. Temeljio se na čvor-čvor mrežnim transakcijama, a transakcije bi pratili članovi mreže u glavnoj knjizi čiju bi kopiju imao svaki član mreže. U svom radu, opisao je dva protokola bazirana na pretpostavci da postoji neprativa mreža gdje se pošiljatelji i primatelji identificiraju isključivo putem digitalnih pseudonima, kao što su javni ključevi, te da se svaka poruka potpiše kod pošiljatelja i enkriptira sve do primatelja. B-money također je omogućavao stvaranje novca baziranog na prethodno neriješenim kriptografskim slagalicama (eng. *cryptographic puzzles*). Kako bi se borio s problemom dvostrukog trošenja, svaki je čvor u sustavu morao napraviti depozit određene svote novca na zajednički račun. Skupljeni novac koristio se kao kazna za krivotvorenje novca i nagrada za dokazivanje krivotvorenja. Doduše, sustav kazni i nagrada teško je održavati bez centralnog autoriteta da donese odluku u slučaju neslaganja. [**1**, **7**]

Hashcash: Adam Back je 2002. godine predložio Hashcash, sustav baziran na dokazu o radu i kriptografskim hash funkcijama kako bi se derivirao probabilistički dokaz procesnog rada kao mehanizam autentikacije. Karakteristike ovakvog sustava značile bi da će biti teško pronaći rješenje, ali jednostavno za provjeriti dano rješenje. U Hashcashu, svrha dokaza o radu bila je da procesno oteža slanje istog sadržaja, čime se rješava problem neželjenih pošiljki (eng. *spam*). Dakle, glavna svrha bila je spriječavanje email spam poruka kroz obvezno ulaganje procesorske moći od strane pošiljatelja prije slanja emaila. S obzirom da bi se identitet pošiljatelja trebao štititi, niti jedana tradicionalna provjera autentikacije nije moguća u ovakvom scenariju. Dakle, mail server je zatjevao rješenje procesnog izazova kao autentikacijsku metodu za prihvaćanje poruke i daljnje prosljeđivanje. U slučaju Hashcash-a, ovo je relizirano uz pomoć dodatnog zaglavlja e-maila. Procesorska moć potrebna za slanje jednog emaila bila bi trivijalna i ne bi utjecala na performans računala. S druge strane, slanje tisuća ili milijuna mailova bilo bi suviše skupo u okvirima potrebne procesorske moći. Hashcash je na ovaj način postizao svoj cilj bez potrebe da novčano naplaćuje slanje email-ova. Back-ova shema dokaza o radu konceptualno je ponovno iskorištenja kod bitcoina u svrhe rudarenja. [**1**, **7**, **10**]

RPOW: Na temelju svih prethodnih radova, Hal Finney je 2004. godine predstavio prvi

valutni sustav baziran na dokazu o radu koji se može ponovno koristiti (eng. *Reusable Proof of work*) i Szabo-voj teoriji kolekcija. Finney-eva shema predstavlja žetonski novac koji je bio u skladu s konceptom vrijednosti zlata. Nakon što je bitcoin mreža pokrenuta, Hal Finney bio je prvi korisnik ove nove distribuirane kriptovalute nakon Satoshi Nakamota. Primio je jednu transakciju u iznosu jednog bitcoina od kreatora, Satoshi-ja Nakamota. [**1**]

bit gold: 2005. godine, Nick Szabo dizajnirao je novu digitalnu valutu imena bit gold. Njegov se sustav također oslanjao na kriptografske slagalice. Nakon što bi se rješile, poslale bi se na javni registar otporan na greške te bi se dodjelile javnom ključu čvora koji bi je rješio. Na ovaj način omogućen je mrežni konsenzus za nove jedinice valute puštene u opticaj. Kako bi adresirao problem dvostrukog trošenja bez centralnog autoriteta, Szabova shema dizajnirana je da oponaša karakteristike povjerenja zlata. Korištenje dokaza o radu i distribuiranog svojstva registra izuzetno je slično kasnijoj implementaciji bitcoina. Rješavanjem slagalica u opticaj su puštane nove jedinice valute, no nije bilo jasne kontrole o količini koja se može kreirati te koliko brzo. Szabo je i sam izjasnio brigu o moćnom računalu koje bi moglo preplaviti tržište jedinicama bit golda te time srušiti njegovu cijenu. Szabo je 2002. godine također predstavio teoriju kolekcija (eng. *theory of collectibles*) utemeljenu na podrijetlu novca. [**1**, **7**]

B-money i bit-gold bile su samo ideje i teoretska razmatranja koja nikada nisu u potpunosti implementirana, stoga je bilo teško procijeniti koliko bi dobro funkcionirala u praksi. Nikada nisu privukle interes javnosti, osim male skupine ljudi kriptografskih entuzijasta. B-money, bit-gold i kasnije bitcoin, kreirali su entuzijasti kako bi zadovoljili potrebu anonimnosti u digitalnim transakcijama. Bilo je i komercijalnih pokušaja kreiranja anonimnih, digitalnih sustava valuta. Slično bitcoinu, ti su se sustavi sastojali su se od neovisnih jedinica valuta, omogućavali su bolju podjelu te su uključivali univerzalnu, trajnu, glavnu knjigu transakcija. Doduše, ovi sustavi većinski su bili centralizirani. Dva najpoznatija primjera bili su DigiCash te e-cash. Bitcoin je uzeo neke elemente svih prethodnih sustava te ih spojio u potpuno novu inovaciju. Očekivani elementi te inovacije bili su čvor-čvor mreža, korištenje enkripcije te javnog i privatnog ključa. Glavna novost bila je spajanje naglašenih koncepata s idejom blockchaina, dokaza o radu i rudarenja, što je stvorilo Bitcoin sustav kakav poznajemo danas. [**7**]

3.1.4. Bitcoin

Povijest bitcoina započela je 2008. godine objavom članka *"Bitcoin: Čvor-čvor Elektronički Novčani sustav" (eng. Bitcoin: A Peer-to-Peer Electronic Cash System)* pod pseudonimom Satoshi Nakamoto. Satoshi Nakamoto objedinio je nekoliko prethodnih inovacija, kao što su b-money i HashCash, kako bi stvorio potpuno decentralizirani elektronički novčani sustav koji se ne oslanja na centralni autoritet za izdavanje valute, nagodbu ili validaciju transackija. Ključna inovacija bila je korištenje sustava za distribuirano procesiranje (eng. *distributed computation system*) s uključenim algoritmom dokaza o radu. Sustav je ujedno provodio globalne "izbore" svakih desetak minuta, omogućujući decentraliziranoj mreži da dosegne konsenzus o stanju transakcija. Na taj se način elegantno rješava problem dvostrukog trošenja te se jedinica valute ne može potrošiti dva puta. Problem dvostrukog trošenja bio je slabost prethodnih digitalnih valuta i adresirao se brisanjem svih transakcija putem centralne obračunske kuće (eng.

clearinghouse). [**1**, **2**, **11**]

Bitcoin je sustav koji se temelji na složenim algoritmima i specifičan je po tome što svaki korisnik ima uvid u svoje transakcije i transakcije ostalih sudionika u mreži, što ga čini tarnsparentnim. Da bi se spriječilo krivotvorenje bitcoina, svaki korisnik ima privatni ključ, koji se povezuje s digitalnim potpisom korisnika, a on se razlikuje u svakoj pojedinoj transakciji. Upravo zbog privatnog ključa i digitalnog potpisa neizvedivo je krivotvorenje i zlouporaba bitcoina. Sve transakcije vezane uz bitcoin zapisuju se u javnu glavnu knjigu. Glavna knjiga je transparentna i dostupna svima. U bilo se kojem trenutku u vremenu može slijediti put svih transakcija određene kriptovalute. U isto vrijeme, stranke uključene u transakciju ne identificiraju se osobnim podacima, već nizom znakova i brojeva. Kreiranjem transakcije, glavna se knjiga proširuje informacijom o broju jedinica valute i adresi na koju se prenose. Adresa je niz alfanumeričkih znakova namjenjena za dijeljenje te se stoga još naziva javnom Bitcoin adresom. Svakom uplatom, transakcija se širi mrežom, zajedno s potpisom baziranim na privatnom ključu pošiljatelja i adresi primatelja. Privatni je ključ također niz alfanumeričkih znakova različitih duljina. S obzirom da je privatni ključ pošiljatelja jedina predispozicija potrebna za kreiranje ispravnog potpisa, izuzevši adresu primatelja, trebao bi se pohraniti na sigurno mjesto. Glavnu knjigu ažurira, odnosno zapisuje i provjerava, bitcoin zajednica. Zajednica se često naziva Bitcoin mreža ili Bitcoin sustav, s početnim velikim slovom. S druge strane, bitcoin napisan malim slovom odnosi se na jedinicu valute, poznatiju pod skraćenicom BTC. Doduše, konvencija pisanja varira od kriptovalute do kriptovalute. Najmanja jedinica valute u bitcoin ekosustavu zove se Satoshi. Jedan bitcoin definiran je kao $1 * 10^8$ satoshija. Prema ISO 4217, službeni simbol valute bitcoin je XBT, ali zajednica i dalje široko koristi BTC kao simbol. Česta je miskoncepcija da je zaliha od 21 milijuna bitcoina osigurana kriptografski. Umjetni limit od 21 milijuna bitcoina definiran je programski. Sve dok se većina korisnika drži pravila definiranih u referentnoj implementaciji, ukupni broj bitcoina bit će limitiran na 21 milijun algoritmom koji izdaje nagrade za rudare. Algoritam koji izdaje nagrade rudarima svakih 210 000 blokova započinje novu eru te se dobivena nagrada rudarenja prepolovljava. Algoritam ukupno definira 33 ere. [**1**, **2**, **4**, **5**, **7**, **11**, **18**]

Službeni početak bitcoina kao decentralizirane valute jest 3. siječnja 2009., trenutak kada prvi blok bitcoin protokola je kreiran. Temelji se na implementaciji koju je objavio Nakamoto, no naknadno su ju mijenjali i unaprijeđivali brojni drugi programeri. Satoshi Nakamoto bio je aktivan u razvoju Bitcoina do prosinca 2010. U međuvremenu, Satoshi Nakamoto se povukao iz javnosti sredinom 2011 godine, nakon čega više nitko nikad nije čuo od njega. Odgovornost za daljnji razvoj koda i mreže prepustio je grupi volontera. Samo ime Satoshi Nakamoto je alias, a identitet osobe ili grupe ljudi koji stoje iza ove inovacije trenutno nije poznata. Vjeruje se kako posjeduje oko milijun bitcoina koje je na početku izrudario, ali nikada nije potrošio niti jedan, što je javno vidljivo unutar mreže. U svakom slučaju, nitko nema kontrolu nad bitcoin sustavom, pa niti Satoshi, jer u potpunosti djeluje na matematičkim principima. Inovacija bitcoina potaknula je nova znanstvena područja distribuiranog procesiranja, ekonomije i ekonometrike. [**1**, **2**, **4**, **5**, **7**, **11**, **18**]

Sredinom 2011. godine, Bitcoin se prvi puta pojavio u medijima tijekom WikiLeaks afere. WikiLeaks je web stranica koja objavljuje informacije, posebice vijesti i tajne informacije

od klasificiranih izvora. S obzirom da su objavljivali informacije protiv vlade, uskoro su im servisi plaćanja (kao što su Američka banka, PayPal i Visa) ukinuli svoju uslugu, čime je postalo gotovo nemoguće primati donacije svojih pristaša. Stoga je vlasnik WikiLeaksa, Julian Assange, odlučio primati donacije u Bitcoinu, čime se naglasila njegova fleksibilnost, anonimnost te neovisnost od tradicionalnih financijskih struktura. Bitcoin je privukao masovnu medijsku pozornost kasne 2013. godine. Njegova je cijena skočila s 15 američkih dolara od početka 2013. godine na preko 1200 dolara do kraja listopada 2013. godine. U to vrijeme je i Baidu, Kineska tražilica i peta najposjećenija stranica na svijetu, odlučila primati Bitcoin za svoje usluge. Još jedan razlog koji je pridonio masivnom medijskom pokriću bila je činjenica da se Bitcoin nalazio u centru nekoliko događaja i skandala, među kojima je najveći bio FBI-jev upad na Silk Road. Silk Road je web stranica koja je spajala kupce i prodavatelje ilegalnih supstanci i usluga. FBI je procijenio ukupni prihod stranice tijekom dvije i pol godine postojanja na 1.2 milijarde dolara. Krajem studenog 2013. godine, Silk Road je ugašen, a Ross William Ulbricht osuđen za pokretanje i održavanje stranice. U tom procesu, FBI je zaplijenio 26 tisuća bitcoina, koji su u to vrijeme bili vrijedni oko 3 i pol milijuna dolara. Svi ovi događaji privukli su pozornost zakonodavaca i regulatora. Unatoč saslušanjima u Američkom senatu, digitalne valute ostavile su generalno pozitivan utisak. Iako su zakonodavci ukazivali na potencijalne rizike, izravne regulacije nisu predlagane. S druge strane, Kineska je centralna banka zabranila financijskim institucijama da dolaze u doticaj s digitalnim valutama. Stoga je i Baidu bio primoren prestati prihvaćati Bitcoin. Osim Kine, Vijetnamski financijski autoriteti zabranili su bitcoin na razini cijele zemlje. Uz sve ove i slične priče, bitcoin je probio svoj put u medije. Iako često nisu diskutirani detalji, javnost je spoznala novu digitalnu valutu, bez centralne banke i nacionalnih granica. Bitcoin se oglašavao kao instantan, anoniman i besplatan način transfera novca, zbog čega je samo rastao u vrijednosti i popularnosti. Bio je smatran kao brža i jeftinija alternativa postojećem novcu. Kao što vidimo danas, dio entuzijazma bio je pogrešan s obzirom da transakcije često nisu besplatne, a niti instantne. No neovisno o valuti, bitcoin je genijalna inovacija koja može riješiti brojne probleme u decentraliziranim mrežama. [**7**]

Od objave članka 2008. godine do danas, bitcoin je doživio značajne promjene obilježene naglim rastom i padom vrijednosti koji su bili popraćeni interesom medija i javnosti. S obzirom da je bitcoin sustav temeljen na složenim kriptografskim algoritmima te nema središnji autoritet koji izdaje novac ili nadzire transakcije, vode se rasprave može li bitcoin postati nova globalna valuta koja neće biti podložna inflaciji niti utjecaju centralnih banaka. Decentraliziranost i nereguliranost sustava smatrala se njegovom prednošću, ali se isto tako pokazala kao glavni uzrok velike promjenjivosti cijena bitcoina jer ovisi isključivo o odnosu ponude i potražnje. Do današnjeg dana, najuspješnija je kriptovaluta što se tiče tržišne kapitalizacije. Više od 1000 alternativnih kriptovaluta baziranih na bitcoinu kreirano je od osnivanja bitcoina. Početkom 2018. godine, tržišna kapitalizacija bitcoina iznosila je 661.2 milijardu dolara s najvišom cijenom u povijesti kriptovaluta od 19665 dolara po jednom bitcoinu. Ovime je dokazana tehnička sposobnost i mogućnost održavanja distribuirane kriptovalute. Osim toga, distribuirano procesiranje koje bitcoinu omogućuje sigurnost eksponencijalno je poraslo te sada premašuje ukupnu računalnu moć svjetskih super računala. [**1**, **4**]

Bitcoin je dizajniran da bude decentralizirana kriptografska valuta koja se ne oslanja na

trećeg posrednika. Kombinacija inovativnog programskog rješenja, kriptografske učinkovitosti i distribuiranog konsenzusa te njihova praktična demonstracija dokazuju da se ova tehnologija može koristiti i u drugim domenama izvan kriptovaluta. Ove implikacije privlače pozornost znanstvene zajednice i povezuju se s drugim sigurnosnim problemima distribuiranih sustava, kao što su distribuirani prostor imena (eng. *distributed name spaces*), sigurnosne vremenske oznake (eng. *secure timestamping*) i mnoge druge. Sve navedene okolnosti čine razvoj bitcoina kao financijskog instrumenta veoma zanimljivim za istraživače u brojnim područjima. Tehnologija iza bitcoina može se koristiti kako bi se postigao konsenzus na decentraliziranim mrežama za poštene izbore, nagradne igre, registre imovine, digitalne ovjere i slično. Tehnologije koje tvore blockchain sve se učestalije pokrivaju u znanstvenim literaturama te pronalaze načine do potrošačkih primjena. Unatoč rastućem interesu unutar akademije i privatnog sektora, i dalje postoje otvoreni problemi u okvirima traženja balansa između performansa, skalabilnosti, sigurnosti, decentralizacije te autonomije takvih sustava. [**1**, **2**, **11**, **18**]

Iz svega do sada navedenog, Bitcoin mogli definirati kao kolekciju različitih koncepata i tehnologija koje formiraju temelj ekosustava digitalnog novca. Predstavlja jedinicu valute i koristi se za pohranu i prijenos vrijednosti između sudionika u bitcoin mreži. Sudionici bitcoin mreže međusobno komuniciraju koristeći bitcoin protokol, iako se mogu koristiti i druge prijenosne mreže. Korisnici mogu izvršiti prijenos bitcoina putem mreže te na taj način simulirati sve moguće radnje konvencionalnih valuta, kao što su kupnja i prodaja dobara, slanje novca i tako dalje. Tehnologija na kojoj je temeljen bitcoin ima ugrađenu enkripciju i digitalne potpise kako bi se osigurala sigurnost bitcoin mreže. Bitcoin se može kupovati, prodavati i razmijenjivati s drugim kriptovalutama na burzama kriptovaluta. Naspram tradicionalnih valuta, bitcoin je u potpunosti virtualan te ne postoji u fizičkom obliku. Prenosi se putem transakcija u kojima se specificira određena vrijednost s računa pošiljatelja na račun primatelja. Svaki korisnik bitcoina posjeduje javni i privatni ključ koji im omogućuju potvrdu vlasništva transakcija u bitcoin mreži. Ključevi su inače spremljeni na digitalnom novčaniku te su jedini preduvjet korištenja bitcoina. Bitcoin je u potpunosti distribuiran sustav temeljen na čvor-čvor (eng. peer-to-peer) mreži, stoga nema centralnu kontrolnu točku. Bitcoin se kreira kroz proces rudarenja (eng. *mining*) koje predstavlja traženje rješenje kompleksnog problema. Svaki sudionik bitcoin mreže može biti rudar i koristiti računalnu moć vlastitog računala kako bi pokušao pronaći rješenje tog problema. U prosjeku svakih desetak minuta rudar uspije validirati trasakciju pronalaskom rješenja te prikupiti nagradu u iznosu određene svote bitcoina. U suštini, rudarenje bitcoina decentralizira izdavanje valute te uklanja potrebu za centralnom bankom. Bitcoin protokol uključuje ugrađen algoritam koji regulira ulogu rudarenja preko cijele mreže. Težina problema koji rudari moraju riješiti prilagođava se dinamički, tako da se pravo rješenje pronađe u prosjeku svakih desetak minuta neovisno o broju rudara koji rade na određenom problemu u nekom trenutku u vremenu. Protokol ujedno prepolovljava brzinu na kojoj se kreiraju novi bitcoini svakih četiri godine te time limitira ukupan broj bitcoina koji će biti kreiran na fiksnih 21 milijun bitcoina. Predviđanja pokazuju da će sav bitcoin biti u opticaju do 2040. godine. S obzirom da se izdaje u opticaj opadajućom krivuljom, u dugom je roku bitcoin kao valuta deflacijski nastrojena. Dakle, bitcoin ne može doživjeti inflaciju printanjem novog novca iznad ili ispod očekivane stope izdavanja. [**2**]

3.2. Tehnička strana kriptovaluta

Bitcoin djeluje nad čvor-čvor mrežom (eng. *peer-to-peer network*) gdje se čvorovi mogu povezati i otići iz mreže po volji. Bitcoin čvorovi povezani su putem TCP/IP protokola. Inicijalno, čvorovi su se povezivali na mrežu tako da su tražili adrese čvorova od poslužitelja imena domena (eng. *Domain Name System - DNS*) koji su im vraćali listu IP adresa trenutno povezanih Bitcoin čvorova. Novo povezani čvorovi propagiraju IP adrese čvora putem Bitcoin-ovih *addr* poruka. Bitcoinov zadani klijent može primiti maksimalno 125 TCP konekcija, od kojih je do 8 izlaznih TCP konekcija. [**8**]

Plaćanja u Bitcoinu provode se putem transakcija koje prenose jedinice valute. Čvorovi se referenciraju u svakoj transakciji uz pomoć pseudonima denotiranih pomoću Bitcoin adresa. Svaka adresa mapira unikatan par javnog i privatnog ključa. Ti se ključevi koriste da bi se prenijelo vlasništvo jedinica valute između pošiljatelja i primatelja, odnosno kupca i prodavatelja. Bitcoin adresa je identifikator od 26 do 35 alfanumeričkih znakova koji uglavnom počinju s brojem 1 ili 3. [**8**]

Svaka bitcoin adresa računa se pomoću javnog ključa algoritma digitalnog potpisa eliptične krivulje (eng. *Elliptic Curve Digital Signature Algorithm - ECDSA*) za kojeg vlasnik adrese zna pripadajući privatni ključ. Korištenjem ECDSA potpisa, čvor može potpisati transakciju koristeći svoj privatni ključ, a svaki drugi čvor u mreži može provjeriti autentičnost tog potpisa koristeći javni ključ potpisatelja. [**8**]

Bitcoin transakcija formira se digitalno potpisivanjem hash-a prethodne transakcije koja pokazuje gdje se ta jedinica valute prethodno potrošila. U transakciju se uključuje i javni ključ budućeg vlasnika. Transakcije kao ulaz uzimaju referencu na izlaz neke druge transakcije koja troši iste jdinice valute i za izlaz daje listu adresa koje mogu skupiti prebačene jedinice valute. Izlaz transakcije može se skupiti samo jednom, nakon čega više nije dostupan drugim transakcijama. Jednom kada je spremna, korisnik potpisuje transakciju i šalje ju putem čvor-čvor mreže. Svaki čvor može provjeriti autentičnost BTC-a provjeravajući lanac potpisa. [**8**]

Razlika između ulaznih i izlaznih vrijednosti transakcije prikuplja se u obliku naknada od strane rudara. Rudari su čvorovi koji sudjeluju u generiranju Bitcoin blokova. Blokovi se generiraju rješavanjem slagalice dokaza o radu. Specifičnije, rudari moraju pronaći vrijednost koja odgovara cilju kompleksnosti. Ukoliko se nađe tražena vrijednost, rudari transakciju uključuju u novi blok te time omogućuju bilo kojem drugom entitetu da provjeri ispravnost dokaza o radu. S obzirom da je se svaki blok povezuje na prethodno generirani blok, Bitcoinov blockchain raste sa svakim novim blokom u mreži. Bitcoinov blok se u prosjeku izrudari svakih desetak minuta. [**8**]

Tijekom normalnog djelovanja, rudari obično rade na produljivanju najduljeg blockchaina u mreži. Najdulji blockchain se izračunava ovisno o lancu koji ima najveći broj blokova kreiranih s najvećom ukupnom težinom od inicijalnog bloka (eng. *genesis block*). U sustavu koji koristi dokaz o radu, moguće je da različiti rudari potencijalo nađu različite blokove u gotovo isto vrijeme. U takvim situacijama događa se grananje blockchaina. Grane se rješavaju unutar Bitcoin sustava, gdje prevladava najdulji blockchain koji ima potporu većine procesne moći u

mreži. [**8**]

3.2.1. Kriptografski alati

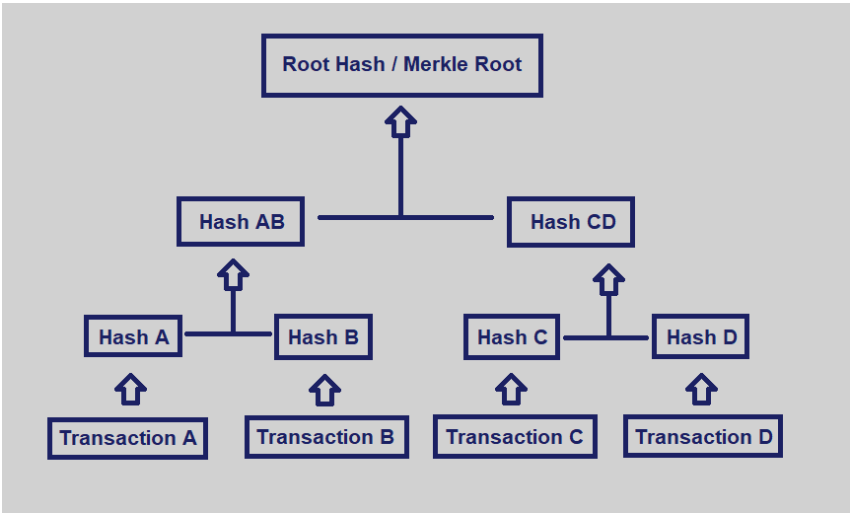
Bitcoin protokol od kriptografskih alata koristi kriptografske hash funkcije, kao što su SHA 256 i RIPEMD160, koristi stabla jele (eng. *Merkle trees*) te algoritam digitalnog potpisa eliptične krivulje (eng. Elliptic Curve Digital Signature Algorithm, skraćeno ECDSA). [**8**]

3.2.1.1. Kriptografske hash funkcije

Kriptografske hash funkcije (eng. *Cryptographic Hash Functions*) mapiraju proizvoljno dugi ulazni niz bajtova na izlaz fiksirane duljine, efektivno ostavljajući otisak na ulaznom nizu. Kriptografske hash funkcije referenciraju se na hash funkcije koje izlažu dva ključna svojstva: jednosmjernost (eng. *one-wayness*) i otpornost na kolizije (eng. *collision-resistance*). Svojstvo otpornosti na kolizije čini važan sigurnosni stup Bitcoina. Primjerice, dokaz o radu u Bitcoinu pretežito je baziran na računanju hash-eva, a *id* transakcije odgovara hashu transakcije. Hash funkcije su ključna komponenta različitih tipova struktura podataka u Bitcoinu, kao što su stabla jele. [**8**]

3.2.1.2. Stabla jele

Stabla jele (eng. *Merkle trees*) omogućuju kombinaciji višestrukih ulaznih nizova u hash stablu konvergenciju u izvorni hash stabla jele. Ova struktura podataka omogućuje kompaktnu reprezentaciju skupa transakcija, kao kada je stablo izgrađeno iz transakcijskih hasheva. Stabla jele mogu se koristiti za inicijalizaciju kriptografskih akomulatora, koji odgovaraju upitu "odgovara li dani kandidat skupu?". Stablo jele jest binarno stablo u kojemu su podaci spremljeni u listove. [**8**, **23**]



Slika 2: Primjer stabla jele [**23**]

Kako bi bolje vizualizirali stablo jele, pratit ćemo primjer sa slike. U ovom slučaju postoje

četiri transakcije skupljene u blok stabla. Najniža ražina su same transakcije apstarhirane na višu razinu pomoću sažimanja dvije transakcije zajedno i dobivanjem izlazne funkcije. Ta je funkcija tada sažeta sa idućom na istoj razini kako bi se dobila nova funkcija više razine. Taj se proces ponavlja sve dok nisu ostale samo dvije funkcije.

3.2.1.3. ECDSA

Bitcoin se trenutno oslanja na algoritam digitalnog potpisa eliptične krivulje (eng. *Elliptic Curve Digital Signature Algorithm, skraćeno ECDSA*) sa secp256k1 krivuljom. ECDSA je varijanta algoritma digitalnog potpisa (eng. *Digital Signature Algorithm - DSA*) koji koristi kriptografiju eliptične krivulje. Traženi secp256k1 privatni ključevi imaju duljinu od 256 bitova i mogu se deterministički pretvarati u pripadajuće secp256k1 javne ključeve. [**8**]

3.2.2. Strukture podataka

Bitcoin i druge vezane valute oslanjaju se na dva različita tipa struktura podataka: transakcije i blokove. Transakcije se zajedno grupiraju u blokove. Blokovi se zajedno ulančavaju pomoću hash-eva prethodnih blokova, time tvoreći autenticiranu strukturu podataka poznatiju kao blockchain. Transakcije i blokovi razbacani su među svim čvorovima koje sudjeluju u čvor-čvor mreži. [**1**]

Novi blok dodaje se u blockchain ako čvor mreže može pružiti ispravan dokaz o radu. Dokaz o radu služi kao mehanizam obrane protiv vanjskih napada te pruža način autentikacije novih blokova, pa tako i blockchaina u cjelini. Ispravni čvorovi usuglašuju se da je u bilo kojem trenutku u vremenu samo najdulji blockchain smatran validnim. Ako čvor ne smatra neki blok ispravnim, onda se taj blok neće dodati u blockchain. Ovakav implicitan proces konsenzusa može se opisati kao "izbor nasumičnog vođe" (eng. *random leader election*) prilikom svakog rješenog dokaza o radu. Vođa može predložiti novi blok te ga nadodati na kraj postojećeg blockchaina. Stoga se bitcoin može opisati kao distribuirani sustav koji koristi dokaz o radu i blockchain kao mehanizam probabilističkog konsenzusa kako bi se prihvatio određeni skup transakcija i njihov redoslijed. Neophodno je da se ispravno rukovode tranzicije vlasništva s bloka na blok putem konsenzusa poznatijeg kao Nakamoto konsenzus. Dakle, vođa je dopušteno da bira jedan blok, nakon čega se bira novi vođa koji je pružio dokaz o radu. Kako bi se ljudi motivirali da pruže svoju procesorske resurse i održavaju čvorove, nagrađuje ih se s jedinicama valute koju rudare za svaki ispravan dokaz o radu pružen za blok i sve vezane transakcije. [**1**]

Tehnologije kriptovaluta mogu se raščlaniti iz brojnih perspektiva. Jedan od pristupa je razdvajanje kriptovaluta u različite razine. Na prvoj razini uvodi se gruba separacija između dvije glavne komponente, a na drugoj razini te se dvije komponente razdvajaju na podsustave. Operacije bitcoina i mnoštva drugih kriptovaluta mogu se razdvojiti na dvije glavne komponente. Prva komponenta jest rukovođenje konsenzusa (eng. *consensus management*) koja sadržava sve što je vezano uz postizanje konsenzusa, kao što su algoritam konsenzusa i povezani komunikacijski aspekti. Druga komponenta je rukovođenje digitalne imovine (eng. *digital asset*

management) te se odnosi na sve aplikacije koje se grade i djeluju nad dogovorenim stanjem, kao što su rukovođenje ključeva i transakcija. Obje se glavne komponente mogu razdvojiti u nekoliko podstustava. Komponenta rukovođenja konsenzusa sadržavala bi mrežni podsustav memorijski podsustav te podsustav algoritma konsenzusa. Komponenta rukovođenja digitalne imovine sadržavala bi podsustav rukovođenja ključeva te podsustav rukovođenja transakcija. [**1**, **6**]

Adrese, transakcije i blokovi su tri osnovne strukture podataka korištene u bitcoinu. Potreba za navedenim strukturama podataka proizašla je iz činjenice da je bitcoin dizajniran kao distribuirana digitalna valuta. Sve kriptovalute koje su bazirane na bitcoinu, neovisno o tome jesu li direktne tenhološke kopije (npr. Namecoin, Litecoin, Zcash) ili samo konceptualno temeljene na bitcoinu (npr. Etherum), uključuju varijante temeljnih struktura podataka s malim modifikacijama. U nastavku će biti objašnjene osnovne gradivne komponente i temelji kriptovaluta. [**1**]

3.2.2.1. Adresa

Na najnižoj razini, Bitcoin adrese, kao i adrese brojnih drugih kriptovaluta, su kriptografski hash-evi javnih ključeva. Predstavlja jedinstveni identifikator koji preuzima ulogu izvorišta i destinacije bilo koje transakcije. Svaka se adresa zapravo sastoji od javnog i privatnog dijela. Javni dio je adresa, koja se može usporediti s brojem računa u tradicionalnom online bankarstvu, a odgovara enkodiranom kriptografskom hashu javnog ključe s bazom 58. Privatni dio je pripadajući tajni ključ te se može usporediti s lozinkom ili potpisom potrebnim da bi se novac preuzeo s tradicionalnog bankovnog računa. Adrese može generirati bilo tko isto kao i parove javnog i privatnog ključa. Time je svakome omogućeno da prima ili šalje bitcoin putem javne adrese bez potrebnog dubljeg znanja o bitcoin protokolu ili algoritmu konsenzusa. Kod bitcoina, adrese su par javnog i privatnog ključa generiranog pomoću ECDSA. Adresa povezana s Bitcoin računom može biti nula ili pozitivna, odnsono u rasponu od najmanje jedinice valute (u slučaju bitcoina, 1 Satoshi) pa do 21 milijuna bitciona (najvećeg iznosa bitcoina koji može biti kreiran prema trenutnom protokolu konsenzusa). Bitcoin adrese inače započinju s decimalnim 1 ili 3 (za adrese s višestrukim potpisima) te se sastoje od 26 do 35 alfanumeričkih znakova. Primjer bitcoin adrese bio bi *1L394jfk32ERK4Qnmqviefnrkdp43EWrneFofNLFa4y*. [**1**, **8**, **12**]

3.2.2.2. Transakcije

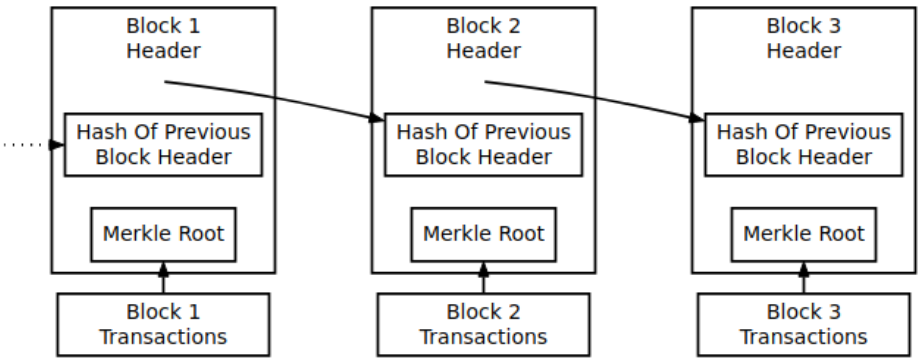
Transakcije su struktura podataka koja može uzeti jedan ili više ulaza i izlaza. Ulaz nadoknađuje BTC koji je referenciran u izlazu prethodne transakcije. Transakcije se koriste kako bi se prenijele jedinice valute s jedne adrese na drugu. Može ih kreirati bilo koji entitet koji je u posjedu jedinica valute. U ovom kontekstu, posjedovanje jedinica valute značilo bi kontrola nad privatnim ključem pripadajuće adrese koja trenutno sadrži jedinice valute za prijenos. Transakcije efektivno predstavljaju lanac transakcija, a bitcoin se tehnički drži samo u izlazima transakcija, a ne unutar adresa. Izlaz transakcije određuje koliko bitcoina ona sadrži te pod kojim uvjetima sljedeća transakcija može nadoknaditi izlaz prethodne transakcije. Pritom enkodira potrebne potrošačke informacije na ulazu transakcije. Uvjeti pod kojima izlaz može

biti potrošen enkodiraju se uz pomoć skripti. Samo sudionici koji su u mogućnosti pružiti točan ulaz skripti mogu potrošiti bitcoin izlaz nekom transakcijom. [**1**, **8**, **12**]

Bitcoin podržava nekoliko različitih tipova transakcija. Samo se podržani transakcijski tipovi propagiraju i validiraju unutar mreže. Transakcije koje ne odgovaraju standardnim tipovima transakcija generalno se odbacuju. Izlaz transakcije mora se potrošiti u potpunosti. Malo je vjerojatno da se ulazne jedinice valute točno podudaraju sa željenom izlaznom količinom. Bitcoin rješava ovaj problem kreiranjem zamjenskih izlaza na kojima se troši razlika između ulaznih i izlaznih jedinica valute. Zamjenski izlazi odgovaraju novim, nasumično generiranim bitcoin adresama čiji su privatni ključevi zadržani kod trenutnog vlasnika ulaznih jedinica valute. Takve adrese referenciraju se kao zasjenjene adrese (eng. *shadow addresses*). U suštini, suma bitcoina ispostavljenog u svim izlazima transakcija ne može biti veća od sume bitciona na ulazima transakcija. Međutim, suma izlaza transakcija može biti manja od sume bitcoina na ulazima transakcija. Razlika između ulaza i izlaza plaća se kao naknada rudaru koji je uključio transakciju u blok. [**8**, **12**]

3.2.2.3. Blok

Glavna struktura podataka kriptovaluta jest blok. Blok se sastoji od zaglavlja bloka i transakcija povezanih u pripadajući blok. Blokovi se ulančavaju uključivanjem kriptografskih hash-eva svojih prethodnika kako bi se stvorila povezana lista, poznatija kao blockchain. Trenutno stanje valute predstavlja se redosljedom blokova u lancu. Oni predstavljaju glavnu knjigu svih izvršenih transakcija, pri čemu su transakcije procesuirane sekvencijalno ovisno o njihovoj poziciji u bloku u kojem se nalaze. [**1**, **8**, **12**]



Slika 3: Pojednostavljen prikaz blockchaina [**24**]

Zaglavlje bloka sadrži različita polja (*nVersion, HashPrevBlock, Hash MerkleRoot, nTime, nBits, nNonce, #vtx, vtx[]*) te ukupno ima 80 bajtova. Iz perspektive integriteta, najbitnije polje zaglavlja bloka jest *HashPrevBlock*. Ono sadrži kriptografski hash (*SHA256*) prethodnog bloka u lancu. Time omogućuje da su blokovi međusobno ulančani u nepromjenjivu strukturu podataka. Integritet blockchaina potom može provjeriti bilo tko tko ima pristup zaglavlju zadnjeg bloka u lancu. Klijent koji ima spremljen samo zadnji blok može provjeriti da lanac do tog trenutka u vremenu nije mijenjan. Iz zadnjeg bloka može zatražiti sve prethodne blokove

i rekreirati hash lanac sve do posljednjeg bloka. Ako se hash posljednjeg bloka podudara, sa sigurnošću se može utvrditi da sadržaj niti jedog prethodnog blok nije mijenjan. [**1**, **8**, **12**]

Redoslijed liste transakcija povezane sa svakim blokom također je veoma bitan jer se one procesiraju sekvencijalno. To omogućuje, primjerice, istim sredstvima da se sekvencijalno prenose nekoliko puta te se pridruže istom bloku. Sve transakcije pridružene određenom bloku vezane su za taj blok putem izvornog hash-a stabla jele (eng. *Merkle tree root hash*) koji je uključen u zaglavlje bloka (*HashMerkleRoot*). Ovo je polje u suštini hash vrijednost svih transakcija u bloku. Ukoliko bi se sadržaj neke od transakcija promjenio nakon što se poveže sa zaglavljem bloka, promjena bi bila vidljiva jer bi se promijenio i izvorni hash stabla jele. [**1**, **8**, **12**]

Blockchain

Najraniji radovi u domeni kriptovaluta većinom su se fokusirali na kriptografske primitive (eng. *cryptographic primitives*) i generalnu sigurnost koja se može postići u takvim sustavima. Najraniji prototipi sustava morali su se oslanjati na povjerljive posrednike (eng. *trusted third parties*) kako bi osigurali ispravno djelovanje sustava i operacija koje se izvode. Pojavom bitcoina, decentralizirane distribuirane kriptovalute, izgubila se potreba za posrednicima. Bitcoin ovo ostvarije kroz kombinaciju poznatih primitiva i tehnika, kao što su dokaz o radu, da bi eventualno uspostavio dogovor (konsenzus) o stanju transakcijske glavne knjige između svih čvorova. Glavna ili javna "knjiga" u koju se zapisuju sve kripto transakcije i vrijednosne izmjene određenih jedinica kriptovaluta zove se blockchain. Svaki zapis temelji se na složenoj matematičkoj kriptografiji te se slijedno zapisuje, jedan blok šifri iza drugoga, gdje svaki blok pokazuje na sljedeći te se tako stvara lanac blokova. Nije moguće promijeniti podatke u lancu jer se pritom uzurpira stanje blokova podataka koji se u njemu nalaze. Blockchain se nalazi na jednom mjestu, stoga svatko tko posjeduje jedinicu kriptovalute ima i svoj primjerak "blockchain knjige" koja se sinkronizira među svim računalima u mreži. Takav pristup pristup konsenzusu, nazvan Nakamoto konsenzus, omogućuje neograničeno sudjelovanje bilo kojem čvoru bez potrebe za dozvoloma. Na početku se izraz blockchain koristio kao referenca na agregaciju i dogovor oko transakcija na nepromjenjivoj glavnoj knjizi. Sada se izraz blockchain koristi kao referenca na skup tehnologija kriptovaluta. Pojam blockchain Satoshi Nakamoto nije direktno uveo u originalnom radu, ali se često koristi na sve koncepte vezane uz tehnologije kriptovaluta. Postoji nekoliko mogućih definicija blockchaina, a svode se na akademsku interpretaciju i kolokvijalnu interpretaciju. [**1**, **5**, **8**, **9**, **10**, **22**]

U akademskoj interpretaciji pojma blockchaina također postoje različite definicije. Preuzeta definicija je poprilično generična i neovisna o algoritmu konsenzusa. Stoga odgovara različitim tipovima blockchaina i pokriva širu sliku ovog pojma. Ova definicija poznatija je kao Princeton definicija: *Blockchain se definira kao povezana lista struktura podataka koja koristi sume hash-eva nad svojim elementima kao pokazivače na odgovarajuće elemente.* Prema ovoj definiciji, sve dok se spremaju i dohvaćaju ispravani blokovi na glavi lanca, konstrukcija blockchaina osigurava da se u potpunosti se mogu provjeriti svi ostali blokovi lancu. [**1**]

U kolokvijalnoj interpretaciji pojam blockchain odnosi se na kategoriju distribuiranih sustava koji su izgrašeni koristeći tehnologije kriptovaluta, kao što su hash lanci, asimetrična kriptografija, teorija igara i slično. Prema ovoj interpretaciji postoje samo dvije različite verzije blockchaina, dozvoljeni (eng. permissioned blockchain) i nedozvoljeni blockchain (eng. permissionless blockchain). Glavno svojstvo nedozvoljenog blockchaina jest da je set čvorova između kojih se dostiže konsenzus oko stanja lanca nije poznat. Poznatiji su još kao blockchain-ovi dokaza o radu (eng. *Proof-of-work blockchains*). Glavno svojstvo dozvoljenog blockchaina jest da je set čvorova između kojih se dostiže konsenzus poznat. Poznatiji su kao blockchainovi otporni na bizantski defekt (eng. *Byzantine Fault Tolerant blockchains*). Daljnja distinkcija može se postići između dozvoljenih blockchain-ova i privatnih blockchain-ova ovisno o kompoziciji i selekciji skupa čvorova. [**1**]

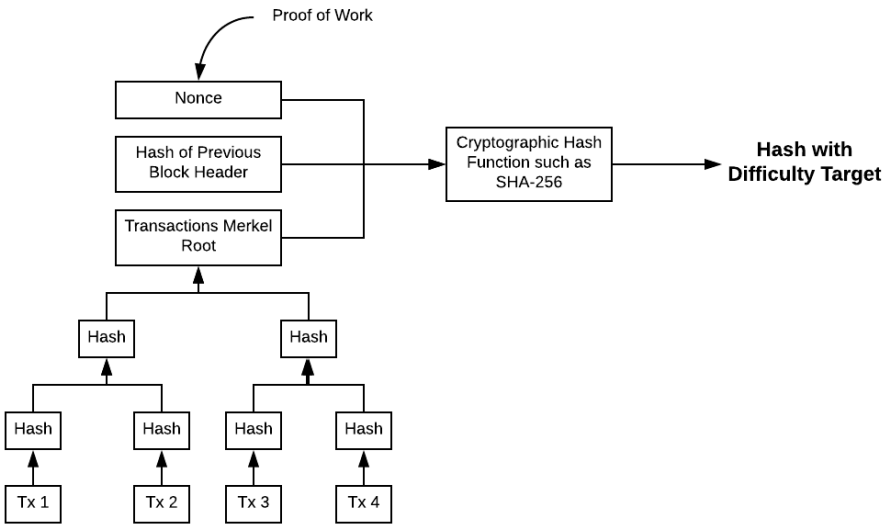
Funckija sažimanja se koristi kroz blockchain protokol kao metoda održavanja dosljednosti podataka jer je pomoću nje vrlo lako sve provjeriti i gotovo nemoguće nešto promijeniti. Gradeći na tim svojstvima možemo riješiti tehnički izazov, pripada li određena transakcija bloku. Postoji više vrsta blokchaina, javni blokchain, privatan blockchain, konzorcijski blokchain i poluprivatni blokchain. Javni blokchain je onaj kojemu mogu svi pristupiti i ažurirati ga. Privatni blokchain je onaj koji je limitiran samo za određenu skupinu ljudi unutar neke organizacije. Konzorcijski blockchain je onaj koji se koristi u suradnji s drugim blockchainima. Poluprivatni blockchain je onaj koji se kombinira između javnog i privatnog blokchaina. [**5**, **9**, **10**]

Kako su ljudi počeli shvaćati načine na koje mogu iskoristi blockchain tehnologiju, krenuli su koristiti u različite svrhe, kao što je pohrana podataka, identiteti, sporazumi i slično. Glavne karakteristike blockchaina su nepromjenjivost, mogućnost samo dodavanja, struktiriranost, vremenska označenost, otvorenost i tarnsparentnost, sigurnost i dosljednost. Kako bi blokchain brzo i lako ovjerio transakcije, one su prestavljene kao stablo jele. Stoga bismo trebali razmišljati o blockchainu kao vrsti tehnologije kao što je internet - opsežna informacijska tehnologija sa slojevitim tehničkim razinama i mnoštvom različitih aplikacija za bilo koji oblik registra imovine, inventara i razmjene, uključujući svako područje financija, ekonomije i novca, opipljive imovine (kao što je fizička imovina, kuća, automobili) te neopipljive imovine (glasovi, ideje, reputacija, intencija, zrdavstveni podaci, informacije itd.). Koncept blockchina je nova organizacijska paradigma za otkrivanje, evaluaciju i prijenos podataka s potencijalom koordinacije svih ljudskih aktivnosti na puno višoj razini nego ikad prije. [**3**, **5**, **10**]

Dokaz o radu

Kako bi se postigao konsenzus između čvorova u Bitcoin mreži, Bitcoin se oslanja na pretpostavku sinkrone komunikacije zajedno s konceptom dokaza o radu. Dokaz o radu svodi se na dokazivanje čorova mreže da su izvršili određeni broj komputacija. Čvorovi koji izvode dokaz o radu nazivaju se rudarima. Što veću količinu hasheva rudar može izvršiti, to su veće šanse da će rudar pronaći blok. Stoga je sposobnost pronalaženja blokova proporcionalna procesorskoj moći rudara. Bitcoinov mehanizam dokaza o radu zahtijeva udvostručen *SHA256* hash sadržaja zaglavlja bloka. Kompleksnost procesa rudarenja prilagođava se dinamički kako bi se novi blok generirao svakih deset minuta te time održao planirani trend rudarenja ograničene količine

bitcoina. [**8**]



Slika 4: Dokaz o radu u blockchainu Bitcoina [**25**]

Kako bi generirali blok, rudari moraju pronaći jedinstvenu vrijednost. Kada se ta vrijednost hashira s dodatnim poljima (primjerice, stablo jele svih ispravnih primljenih transakcija, hash prethodnog bloka, trenutak u vremenu), dobiva se rezultat ispod traženog cilja kompleksnosti. Ako se takva vrijednost nađe, rudari je uključuju (skupa s dodatnim poljima) u novi blok, omogućujući bilo kojem entitetu da provjeri dokaz o radu. Nakon što se uspješno generira blok, rudaru dobije određenu svotu rudarene valute kao nagradu. Ovakvi mehanizmi pružaju jaku inicijativu rudarima da doprinose i održavaju Bitcoin mrežu sa svojom procesorskom moći. Vrijedilo bi napomenuti da bitcoin ima ograničene zalihe. Dakle, bitcoin definira brzinu kojom će se generirati jedinice valute. Primjerice, 2009. godine svaki je rudar dobio 50 bitcoina za generiranje novog bitcoin bloka. Količina bitcoina koju rudari dobivaju kao nagradu prepolovljava se u prosjeku svakih četiri godine, sve dok se ne izrudare svi bitcoini iz zalihe. Jednom kada se blok generira, on se propagira cjelokupnoj mreži. Bilo koji entitet koji primi blok može verificirati ispravnost dokaza o radu računajući hash nad dodatnim poljima bloka. Time ujedno provjerava ispravnost transakcija uključenih unutar bloka te da je izračun ispod ciljane kompleksnosti. Bitcoin mreža ima globalnu kompleksnost blokova (eng. *global block difficulty*) koja se ažurira svakih 2016 blokova. U suštini, kompleksnost se prilagođava ovisno o vremenu u kojemu se generiralo posljednjih 2016 blokova u mreži. U slučaju da je posljednjih 2016 blokova trebalo više od 14 dana da se izračunaju, odnosno više od 10 minuta u prosjeku, onda se kompkesnost smanjuje. U suprotnom, ako je bilo potrebno manje od 14 dana da se izračunaju, onda se kompleksnost povećava. [**8**, **12**]

Blok se generalno smatra ispravnim nakon što je prosljeđen svim čvorovima u mreži te provjeren i prihvaćen od istih. S obzirom da svaki blok pokazuje na prethodno generirani blok, Bitcoinov blockchain svakodnevno raste generiranjem novih blokova u mreži. Kao što je spomenuto u uvodu, ako rudari ne dijele isti pogled mreže zbog grananja blockchaina, može se dogoditi da rudare na različitim granama blockchaina. Grananja se rješavaju unutar Bitcoin sustava, tako da najdulji blockchain eventualno prevlada. Transakcije koje se ne pojavljuju u

blokovima koji su dio glavnog blockchaina (najduljeg), bit će ponovno dodane u bazen transakcija (eng. *pool of transactions*) unutar sustava te potvrđene u nadolazećim blokovima. Trenutno se u Bitcoin sustavu transakcija isplaćuje ako je primila barem šest konfirmacija, odnosno ako je dodano novih pet blokova nad blokom koji potvrđuje transakciju. Ovakav mehanizam pruža zaštitu protiv napada dvostrukog trošenja jer je procesno nemoguće promijeniti povijest transakcije koja je potvrđena sa šest blokova u sustavu. [**8**, **12**]

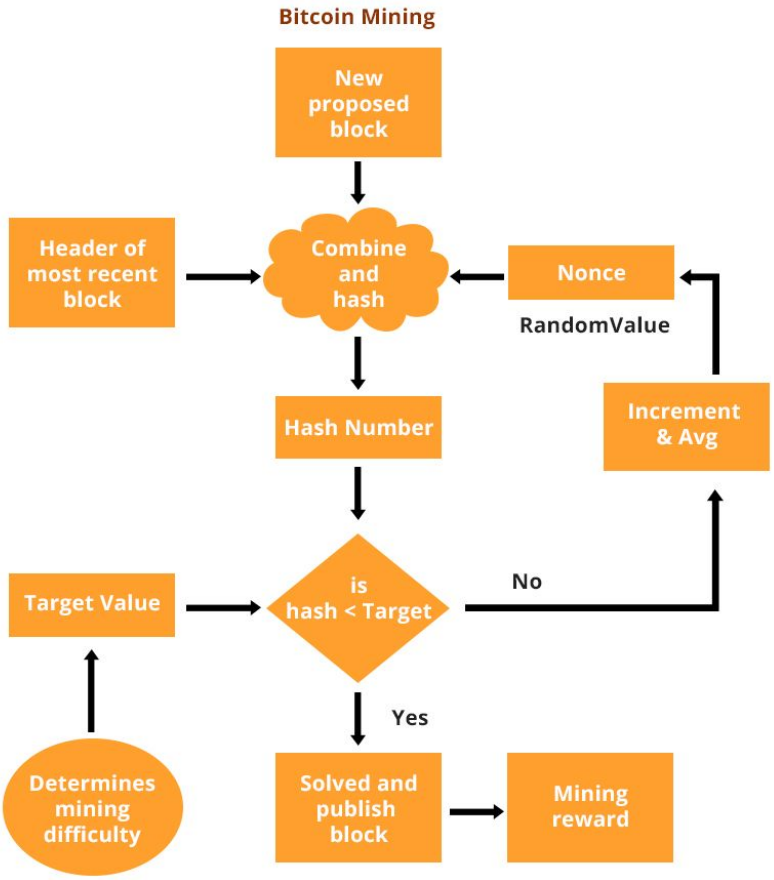
3.2.3. Rudarenje

Rudarenje je proces rješavanja dokaza o radu kao sredstvo dostizanja konsenzusa o trenutnom stanju blockchaina. Čvorovi koji su aktivno uključeni u traženje i pružanje rješenja dokaza o radu nazivaju se rudari (eng. *miners*) Rudari su nagrađeni jedinicama valute u kriptovaluti koju rudare kao kompenzacija za ulaganje procesorske moći u generalnu sigurnost i održavanje kriptovalute. Rudari se mogu pridružiti ili napustiti mrežu u bilo kojem trenutku, čime povećavaju ili smanjuju rudarsku moć mreže. Iz tog razloga blockchain-ovi bazirani na algoritmu dokaza o radu moraju podesiti kompleksnost rješavanja istog kako bi osigurali da generiranje novih blokova u pravilnim intervalima. Probabilnost pronalaska novog bloka eksponencijalno je distribuirana. Stoga su nagrade rudarima isplaćene u nepravilnim intervalima jer blockchain ne može obuhvatiti sve akcije rudara dok se podešava kompleksnost. Kako bi osigurali konstantan izvor prihoda, rudari se udružuju kako bi stvorili rudarske bazene (eng. *mining pools*). Rudarski bazeni sadrže sve resurse rudara, a naknadno dijele profit. [**1**, **9**, **10**, **12**]

Pojasnit ćemo rudarenje na primjeru Bitcoina. Bitcoin funkcionira kao peer to peer mreža u obliku javnog blockchaina, što znači da su svi članovi umreženi bez centralnog poslužitelja i to im omogućuje direktno dijeljenje podataka. Bitcoin mreža funkcionira na kriptografskom protokolu, dok korisnici mreže spremaju, šalju i primaju bitcoin transakcije pomoću računalnog programa u obliku digitalnog novčanika. Rudarenje je proces verificiranja transakcija i stvaranja novih blokova u blockchain mreži koji se spremaju u javnu knjigu transakcija. Ti blokovi su stvoreni rješavanjem funkcije sažimanja baziranog na dokazu u radu, odnosno rudarenju. S obzirom da je svaki blok vezan za prethodni, bitcoin blockchain raste na cjelokupnoj povijesti cijele mreže. Novi blok se generira u prosjeku svakih 10 minuta, dok nagrada za rudarenje ovisi ovisno u broju rudara i onda se tokom vremena mijenja. Kod ostalih kriptovaluta može biti drugačija brzina rješavanja blokova, nagrade i svega ostalog. Rudariti kriptovalute kao što je bitcoin možemo samo s posebnom računalnom opremom, zvanom Asic miner, koji je specifično napravljena samo za rudarenje određenih kriptovaluta. Ostale kriptovalute moguće je rudariti s računalime jake procesorske snage koristeći snagu procesora ili grafičkih kartica. Rudari su članovi mreže koji sudjeluju u procesu rudarenja. Jednog rudara može predstavljati, primjerice, jedna grafička kartica. Često postavljeno pitanje je, od kud vrijednost Bitcoinu? Postoje dva aspekta njegove vrijednosti. Da bi do njega i ostalih kriptovaluta došli, potrebno je uložiti novac u računalnu opremu i trošak struje. Taj trošak predstavlja osnovnu vrijednost ispod koje bi rudarenje bilo neisplativo. Većina ljudi koja ih izrudare nisu voljni prodavati ispod osnovnih troškova. Drugi aspekt njihove vrijednosti jest trgovanje na burzama, gdje tržište, tj.

ljudi, odluče koliko su spremni dati za određenu količinu kriptovalute. [**1**, **8**, **9**, **11**, **12**]

3.2.3.1. Dostizanje konsenzusa



Slika 5: Proces rudarenja [**26**]

Rudarenje je centralna točka Bitcoin protocola, a sastoji se od dvije primarne uloge: dodavenje novih bitcoina ukupnim zalihama valute te provjera transakcija. Nepakirane transakcije koje su se nedavno dogodile u Bitcoin mreži ostaju u transakcijskom bazenu (eng. *transaction pool*) kako bi se zapakirale u blok. Nakon što se blok kreira, potrebno mu je zaglavlje prije nego se može prihvatiti u blockchainu. Rudar koristi zaglavlje prethodnog bloka u blockchainu kako bi konstruirao novo zaglavlje za trenutni blok. Zaglavlje bloka također sadrži ostale elemente, kao što su vremenska oznaka, verzija Bitcoin klijenta te *ID* koji odgovara prethodnom bloku u lancu. Rezultirajući blok još se naziva kandidatski blok (eng. *candidate block*) te se može dodati u blockchain ako su zadovoljeni određeni kriteriji. Rudari se međusobno natječu kako bi pronašli rješenje dokaza o radu za blok koji su kreirali. Prvi rudar koji nađe rješenje je pobijedio. Utrka između rudara zaključuje se kroz desetak minuta te potom započinje novi ciklus. Jednom kada se rješenje Bitcoin slagalice otkrije, rudar može završiti rad s blokom te ga objaviti mreži,

čime ga dodaje na kraj blockchaina. [**9**, **10**]

Kandidatski blok u blockchain se može dodati tek kada se doda zaglavlje. Zaglavlje posljednjeg bloka u blockchainu se spaja s 32 bitnom vrijednošću. Ova se kombinacija usmjerava na hash funkciju (SHA-256) kao ulaz. Hash funkcija računa novi rezultirajući hash kao izlaz. Generirani hash se potom uspoređuje s ciljenom vrijednosti kompleksnosti mreže u danom trenutku. Ako je hash vrijednost veća od ciljane, onda se 32 bitna vrijednost podešava te se šalje novi ulaz hash funkciji kako bi se dobio novi izlaz. Problem pronalaženja prikladne hash vrijednosti koja je manja od ciljane vrijednosti jest srž dokaza o radu, a može se riješiti isključivo na silu (eng. *brute force*). Kada rudar otkrije hash vrijednost manju od ciljane, taj se hash onda koristi u zaglavlju kandidatskog bloka. Time rudar dokazuje rad koji je vršio kako bi otkrio hash i ujedno potvrđuje sve transakcije u bloku. Blok se potom dodaje u blockchain, a rudar dobiva nagradu u jedinici rudarene kriptovalute. [**9**, **10**]

Sada kada imamo generalnu ideju kako se vrši proces rudarenja, možemo govoriti o kompleksnosti rudarenja i ciljanim vrijednostima. Kompleksnost rudarenja (eng. *mining difficulty*) može se definirati kao težina pronalaženja vrijednosti hasha koji je ispod ciljane vrijednosti Bitcoin mreže. Povećanje u kompleksnosti odgovara duljem vremenu potrebnom za otkrivanje hash vrijednosti te samim time rješavanja dokaza o radu. Idealno vrijeme rudarenja postavlja mreža kriptovalute, a u slučaju Bitcoin mreže iznosi 10 minuta. Navedeno implicira da će se novi blok kreirati u mreži svakih 10 minuta. Vrijeme rudarenja ovisno je o tri ključna faktora: ciljanoj vrijednosti, broju rudara u mreži te kompleksnosti rudarenja. Ova tri faktora međusobno su ovisna:

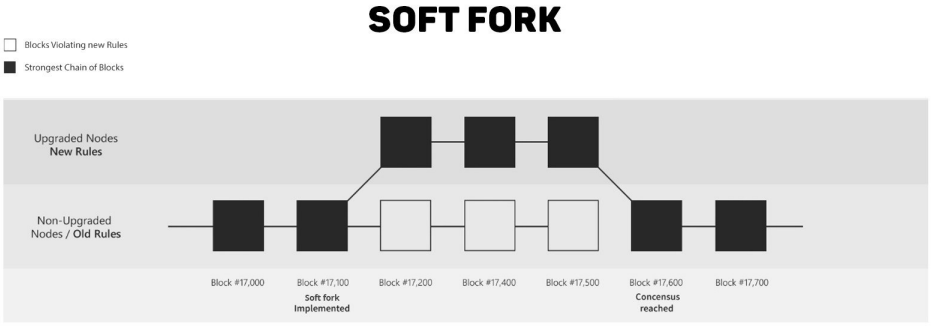
- Povećanje u kompleksnosti rudarenja rezultira u smanjenju ciljane vrijednosti, čime se kompenzira vrijeme rudarenja.
- Povećanje u broju rudara koji se pridružuju mreži rezultira povećanju brzine rješavanja dokaza o radu, što smanjuje vrijeme rudarenja. Kako bi se ovo podesilo, kompleksnost rudarenja se povećava, a vrijeme kreiranja blokova vraća se na postavljeno stanje.
- Ciljana vrijednost se ponovno izračunava i prilagođava svakih 2016 blokova, što se otprilike događa svakih dva tjedna. [**10**]

3.2.3.2. Grananja blockchaina

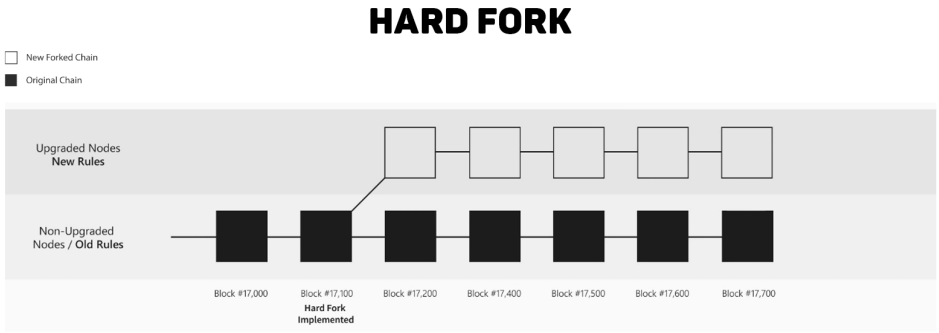
Postoji zanimljiv scenarij u kojemu se nekoliko rudara natječe da riješe dokaz o radu i kreiraju blok. Može se dogoditi da dva rudara nađu ispravnu vrijednost u razmaku od nekoliko sekundi i pošalju riješenje na mrežu. Ovakva situacija poznatija je kao grananje i potpuno je normalan scenarij u djelovanju Bitcoin mreže, posebice s naglim rastom i pritokom novih rudara. Tijekom normalnog djelovanja Bitcoina, rudari rade na proširenju najduljeg blcokchaina u mreži. Ukoliko rudari ne dijele isti pogled mreže, dogodi se da rade na različitom blockchainu, što rezultira grananjima u blockchainu. Grananja se interno rješavaju koristeći pravila konsenzusa unutar Bitcoin sustava. Grananja se inače riješe sa sljedećim nadolazećim blokom. Najdulji blockchain, kojega podržava većina procesne moći, postati će glavni. Transakcije koje se ne pojave u blokovima, a dio su glavnog blockchaina, ponovno će se dodati u bazen transakcija

(eng. *pool of transactions*) sustava te se potvrditi u nadolazećim blokovima. Dakle, grananja blockchaina u suštini su štetna djelovanju Bitcoin sustava. S obzirom da će jedan blockchain prevladati, sve transakcije koje su bile uključene u druge lance neće biti priznate od drugih rudara u sustavu. Bitcoin ne sadrži nikakav mehanizam da olakša ovaj problem. Doduše, normalna grananja nisu zabrinjavajuć događaj jer se inače riješe u roku nekoliko minuta. U slučaju da grana ostane na životu dovoljno dugo vremena, osobe koje održavaju Bitcoin sustav (programeri) moraju donijeti odluku kojom će odabrati jedan lanac na štetu drugoga. To mogu učiniti slanjem poruka upozorenja i hard kodiranjem preferiranog lanca u kodu klijenta. [**8**, **10**]

U ožujku 2013. godine dogodilo se grananje bitcoina koje je zahtijevalo intervenciju progarmera Bitcoina. Nastalo je uslijed ažuriranja verzije i promjene baze podataka na klijentu. Početni blok od kojeg je krenulo grananje sadržavao je oko 1700 transakcija te je bio potvrđen nizom sljedećih blokova. Kao posljedica, svi rudari koji su rudarili verziju prije ažuriranja odbacili su taj blok i sve nadolazeće, dok su rudari koji su prešli na novu verziju prihvatili taj blok i dodali ga u svoj blockchain. Lanac na novijoj verziji podupirala je većina procesorske moći mreže. Unatoč tome, programeri Bitcoina odlučili su, 90 minuta nakon što se grananje dogodilo, da će manji lanac biti onaj ispravan. Ovakva odluka preispituje decentraliziranost Bitcoin sustava i činjenicu da bi većina trebala upravljati Bitcoinom. U konačnici, manje od 10 entiteta nadglasalo je većinu procesorske moći u mreži, što je utjecalo na tisuće transakcija. Takvi utjecajni entiteti imaju moć činiti radikalnije odluke, primjerice, prihvaćati ili odbijati transakcije sustava. [**8**]



Slika 6: Blago grananje [**27**]



Slika 7: Teško grananje [**27**]

Treći scenarij grananja su blaža i teža grananja (eng. *soft and hard forks*). Mogu se dogoditi u sluačju ažuriranja temeljnog koda Bitcoina, pri čemu se događa trajno grananje iz-

među čvorova koji nisu ažurirani i čvorova koji su prešli na novu verziju. Neažurirani čvorovi nisu u mogućnosti validirati novo kreirane blokove, dok ažurirani čvorovi dodaju nove blokove po novim pravilima konsenzusa. Dva potpuno drugačija tipa blokova počinju se pojavljivati u mreži, a mreža nije u mogućnosti spojiti blokove u jedan blockchain sve dok se svi čvorovi ne ažuriraju s ugrađenim novim pravilima. U ovakvim situacijama, postoje dva rješenja. Prvo je rješenje da većina mreže prijeđe na novo rješenje, pri čemu nova pravila omogućuju prijenos dijela ispravnih starih blokova. Ovakav scenarij poznatiji je kao blago grananje (eng. soft fork). Drugom alternativom stari blokovi ostaju neispravni za nove čvorove te se ne prihvaćaju u mreži. Ovakav scenarij poznatiji je kao teško grananje (eng. hard fork). U tom slučaju svi rudari i čvorovi moraju prijeći na novi softver kako bi se rudareni blokovi smatrali ispravnima. Teška grananja mogu izazvati niz problema za korisnike i trgovce koji su kreirali sustave plaćanja i sučelja temeljena na starim pravilima transakcija. Moraju prilagoditi svoj softver novim pravilima kako bi bili u mogućnosti provoditi transakcije. [**10**]

3.2.3.3. Rudarska oprema

Kako je bitcoin rastao u popularnosti i prihvaćenosti kod trgovaca, sve se više rudara počelo pridruživati mreži u nadi da će prikupiti nagrade rudarenja. S vremenom, kompetitivna priroda rudarenja rezultirala je u korištenju specijaliziranog rudarskog hardvera koji može generirati više hash-eva. U nastavku slijedi evolucija procesa rudarenja iz hardverske perspektive:

- **CPU rudarenje:** Ovo je bio najraniji oblik rudarenja dostupan svim Bitcoin klijentima. Postalo je norma rudarenja u ranim verzijama Bitcoin klijenta, ali je nestalo s vremenom jer su se pojavile bolje opcije.
- **GPU rudarenje:** Predstavljalo je sljedeći val unaprijeđenja u rudarenju. Rudarenje s grafičkom procesorskom jedinicom (eng. *graphics processing unit - GPU*) puno je moćnije jer može generirati na tisuću više hash-eva nego središnja procesorksa jedinica (eng. *central processing unit - CPU*). Ovo je sada standard rudarenja za većinu kriptovaluta.
- **FPGA i ASIC:** FPGA (eng. *Field-programmable gated arrays*) su integrirani električni krugovi dizajnirani specifično za rudarenje Bitcoina. FPGA je napisan specifičnih hardverskim jezikom koji im omogućuje da izvršavaju jedan zadatak puno efikasnije u okvirima potrošnje električne energije te generalne efikasnosti rješavanja slagalica. Kratko nakon pojave FPGA, pojavio se ASIC (eng. *application-specific integrated circuits*). ASIC je predstavljao još optimiziraniji komercijalni dizajn dostupan za proizvodnju na veliko. Imao je manje troškove po jedinici uređaja, što mu je omogućavalo masivnu proizvodnju. ASIC uređaji također su veoma kompaktni, tako da ih se može više integrirati u jednom uređaju. Sposobnost niznog spajanja ASIC uređaja po niskim cijenama bila je prekretnica za ubrzavanje brzine rudarenja.
- **Rudarski bazeni (eng. *mining pools*):** Pojavom ASIC uređaja povećavala kompleksnost rudarenja, stoga rudari individualno nisu mogli ostati kompetitivni i nastaviti samostalno rudariti. Bilo je potrebno predugo vrijeme kako bi došli do nagrade, a ta nagrada nije opravdavala troškove koji su pritom nastajali. Rudari su se potom organizirali u grupe,

poznatije kao rudarski bazeni, kako bi udružili procesorske resurse svih članova kao jednu jedinicu. U današnje je vrijeme pridruživanje rudarskom bazenu uobičajen proces kada novi rudar odluči rudariti neku kriptovalutu.

- **Servisi rudarenja na oblaku (eng. *mining cloud services*):** Radi se o dobavljačima specijaliziranim u održavanju platformi za rudarenje. Iznajmljuju svoju opremu rudarima po ugovoru za određenu cijenu i definirani vremenski period. [**10**, **11**]

Jednostavno je za uočiti koliko su ASIC uređaji promijenili proces rudarenja. Bio je to početak "naoružavanja" boljom hardverskom opremom s ciljem povećanja profita rudarenja. Sve se više rudara počelo priključivati Bitcoin mreži sa specijaliziranom opremom za rudarenje. Pojavom novih rudara i procesorske moći u mreži, kompleksnost rudarenja samo se povećavala. U kratkom vremenskom periodu, hardver koji su rudari priuštili u svrhe rudarenja bio je sve manje profitabilan, a u nekim slučajevima je rudario s gubitkom. Značajne investicije bile su potrebne kako bi se održala konkurentnost i profitabilnost. Većina ASIC uređaja već je ostarila, a rudarski bazeni za prosječnog rudara u većini slučajeva prestali su biti profitabilni. [**10**, **11**]

3.3. Stanje kriptovaluta

Riječ "kripto" dolazi od riječi "kriptiranje", odnosno šifriranje. Šifriranje predstavlja prevođenje razgovijetnoga teksta (jasan, otvoreni tekst) ili nekog drugog skupa podataka u nerazgovjetan tekst (kriptirani tekst, kriptogram ili šifrat) kako bi ga onaj koji posjeduje unaprijed utvrđen ključ za odgonetanje (dekriptiranje, dešifriranje) mogao prevesti u izvorni, razgovijetni tekst. Zadaća je kriptografije da omogući dva sudionika komunikacije (pošiljatelju i primatelju) tajnost poruka, čak i u situacijama komunikacije nesigurnim komunikacijskim kanalima kao što su računalna mreža ili telefonska linija. Suvremena kriptologija interdisciplinarna je znanost i uglavnom se oslanja na informatiku, a znatno je potpomognuta teorijom brojeva i drugim matematičkim teorijama. [**30**, **31**]

Iz pravne perspektive, kriptovalute svih oblika spadaju u definiciju virtualnih valuta. Pojam virtualnih valuta definirala je Centralna Europska Banka 2014. godine kao *"digitalnu reprezentaciju vrijednosti koju nije izdala centralna banka ili javni autoritet, niti je nužno vezana uz novčanu valutu, ali je prihvaćena od strane fizičkih ili pravnih osoba kao sredstvo plaćanja te se može prenositi, spremati ili trgovati u elektroničkom obliku".* Kriptovalute su digitalni zapisi o određenim vrijednostima pohranjenim u digitalnim bazama podataka. Jednostavnije, kriptovaluta je digitalno sredstvo razmjene, odnosno, digitalni ekvivalent novca. Postoje samo na internetu i nije ih izdala, niti ih nadzire središnja banka ili država te upravo iz tog razloga formalno nisu smatrane novcem. Za razumijevanje funkcioniranja kriptovaluta potrebna su određena informatička znanja, ali njihovo korištenje i upotreba relativno su jednostavni. Kriptovalute omogućuju pošiljatelju da šalje samo željene podatke primatelju bez dodatnih informacija. Ne zahtjevaju osobna imena niti ostale privatne podatke, već samo digitalnu oznaku novčanika, odnosno ključ. Nema trećih strana, posrednika, kašnjenja s uplatama ili plaćanja naknada. Kriptovalute mogu imati različite aspekte te se mogu promatrati iz brojnih perspektiva, kao što su financijska i ekonomska perspektiva, legalna perspektiva, politička i sociološka perspektiva,

tehnička i socio-tehnička perspektiva. Navedeni različiti pogledi mogu se razdvajati i dalje. Primjerice, tehnički aspekti mogu se podijeliti na područja vezana uz kriptografiju, mrežu i distribuirane sustave, teoriju igara (eng. game theory), podatkovnu znanost (eng. *data science*) te softver i jezičnu sigurnost. Iako je ovaj rad više usmjeren na tehničke perspektive koje su potrebne za razumijevanje ovog širokog područja, mjestimice ćemo se dotaknuti i ostalih perspektiva. Kriptografska zajednica raznovrsna je kao i perspektive na ovu temu. Kriptovalute su, kao što ime upućuje, namijenjene da se koriste kao valute. Stoga privlače različite ljude, kao što su tehnološki entuzijasti, poslovni ljudi i investitori, ideolozi, istraživači, javni autoriteti i zakonske ustanove, financijski regulatori, banke, pa čak i kriminalce koji iskorištavaju anonimnost sustava. Nasuprot tome, distribuirana priroda kriptovaluta poput bicoina ujedno privalči aktiviste i individue koje žive u opresiranim režimima jer mogu rukovoditi svoju digitalnu imovinu neovisno o političkim sankcijama i zakonima. Time se posebno naglašava uloga decentraliziranih valuta za stanovnike takvih zemalja. [**1**, **22**, **31**]

Do današnjeg dana kreirano je preko tisuću različitih kriptovaluta. Neke od njih imale su kratak životni vijek ili su kreirane sa svrhom nekog oblika prijevare, dok su druge donijele dodatne inovacije i slučajeve korištenja. Mehanizmi i osnovni principi većine tih kriptovaluta su, u većoj ili manjoj mjeri, derivirane od originalnog Bitcoin protokola. Većina se razlikuje samo na osnovi različitih konstanti, kao što su ciljni intervali blokova ili maksimalni broj jedinica valute koji će biti u opticaju. Postoje i one koje koriste alternativne algoritme dokaza o radu, kao što su Litecoin i Dogecoin. Druge uključuju dodatne mogućnosti, primjerice Etherum, Namecoin, Zcash i slične. Treće kriptovalute koje koriste različit pristup distribuiranom konsenzusu, kao što su Ripple i PeerCoin. Postoje čak i kriptovalute koje nisu potekle od bitcoina te se ne baziraju na blockchainu, kao što su Iota, Stellar i Hashgraph. Trenutno je napoznatija kriptovaluta bitcoin jer ima najveću tržišnu kapitalizaciju i time najveću pozornost medija. Nagao rast vrijednosti u relativno kratkom roku privukao je brojne ulagače koji su u njemu vidjeli dobru investicijsku priliku. Teško je pronaći financijsku kategoriju o kojoj je u relativno kratkom vremenu napisan toliki broj rasprava kao o bitcoinu. Ovo ne samo da je dovelo do učestale medijske pozornosti, već je i povećanog interesa različitih zajednica, od tehničkih entuzijasta do poslovnih ljudi i investitora, kriminalaca i agencija za provedbu zakona. Cijena bitcoina danas, nakon korekcije i pada s dvadeset tisuća dolara, je i dalje veća od prethodno najveće cijene nekoliko godina ranije. S obzirom da je bitcoin brz, relativno siguran i trenutno najjeftiniji način transfera novca, postavlja se pitanje ima li bitcoin budućnost na svjetskim financijskim tržištima? [**1**, **4**]

3.3.1. Prednosti kriptovaluta

S obzirom da je kroz rad već objašnjen princip rada i glavne inovacije kriptovaluta, u ovom ćemo odlomku sumirati sve prednosti kriptovaluta te tehnologije koje stoje iza istih.

Transakcije: U tradicionačnom poslovanju poslovnih subjekata, brokera i agenata, postoje brojne komplikacije i troškovi vezani uz transakcije. Uključuju papirologiju, naknade brokerima, komisije i razne druge posebne uvjete koji se odnose na proces transakcije. Jedna od prednosti transakcija kriptovalutama jest njihov "jedan na jedan" (eng. *one-to-one*) odnos koji ne zahtjeva posredništvo trećih osoba. Kao rezultat toga, proces provođenja transakcije mnogo

je jasniji i jednostavniji.

Prijenosi imovine: Blockchain kriptovaluta tehnički može predstavljati bazu podataka za imovinu te se koristiti za ugovaranje dobara kao što su automobili ili nekretnine. Osim toga, ekosustav kriptovaluta može biti korišten za ubrzavanje posebnih tipova transfera. Primjerice, ugovori kriptovaluta mogu se dizajnirati da sadržavaju prihvaćanje od treće strane, referenciranje eksternih činejnica ili završetak transfera u određenom vremenu u budućnosti. Ovakav pristup znatno bi smanjio vrijeme i trošak koji se pojavljuje u prijenosu imovine.

Povjerljivije transakcije: Korištenjem postojećih novčanih i kreditnih sustava, cjelokupna transakcijska povijest korisnika može postati referentni dokument za banku ili kreditnu agenciju. Ovo može uključivati jednostavnu provjeru stanja bankovnih računa kao mjeru osiguranja da je dovoljno sredstava dostupno. Prilikom kompleksnijih transakcija, moguć je i pregled cjelokupne financijske povijesti. Glavni problem jest što su ključne informacije korisnika izložene tijekom cijelog transakcijskog procesa. U slučaju kriptovaluta, svaka transakcija je unikatna razmjena između dvije strane, a njeni uvjeti mogu se pregovarati i dogovoriti u svakom slučaju. Informacije koje proizlaze iz takve transakcije su kontrolirane od strane korisnika, tako da može poslati samo željene informacije primatelju. Princip transakcija kriptovaluta štiti financijsku povijest, ali i korisnika od moguće krađe identiteta.

Transakcijske naknade: U slučaju tradicionalnih valuta, naknade će se pojaviti u slučaju pisanja čekova, transfera sredstava ili bilo koje druge financijske radnje. Situacija postaje posebno problematična za entitete koji na mjesečnoj bazi provode veći broj transakcija, što uzima znatan dio financijskih sredstava. U slučaju kriptovaluta, rudari uglavnom primaju kompenzaciju od mreže kriptovalute, pa se naknade za transakciju korisniku često niti ne naplaćuju. Doduše, mogu postojati eksterne naknade servisa pomoću kojih rukovodimo digitalne novčanike, ali su i dalje znatno manji od prosječnih naknada tradicionalnih financijskih sustava.

Jednostavniji pristup kontu: Internet i digitalni prijenos podataka su glavni mediji olakšavanja razmjene kriptovaluta. Ona je moguća svakoj osobi koja ima pristup internetu, malo znanja o mrežama kriptovaluta te pristup nekom od servisa koji posreduje transakcijama kriptovaluta. Procijenjuje se da postoji minimalno 2.2 milijarde ljudi diljem svijeta koji imaju pristup internetu na ovaj ili onaj način, ali nemaju pristup tradicionalnim bankarskim sustavima niti provođenju transakcija. Ekosustav kriptovaluta omogućuje prijenos imovine i procesiranje transakcija za široko tržište potrošača.

Jednostavnija internacionalna trgovina: Kriptovalute po svojoj prirodi nisu predmet deviznih tečajeva, kamatnih stopa, naknadama transakcija ili bilo kojem drugom obliku pristojbe karakterističnom za određenu državu. Korištenjem čvor-čvor mehanizma, internacionalni transferi i transakcije mogu se provesti bez komplikacija, fluktuacija tečajeva i tako dalje.

Individualno vlasništvo: U tradicionalnim bankarskim i kreditnim sustavima vlasništvo nad vlastitim sredstvima prepuštamo trećim posrednicima. Računi se mogu zatvoriti bez prethodne najave iz niza razloga. Glavna prednost kriptovaluta jest individualno vlasništvo nad privatnim i javnim ključem, odnosno vlastitim novčanim sredstvima unutar mreže.

Prilagodljivost: Trenutno postoji preko 1000 različitih kriptovaluta i altcoinova u cirkula-

ciji diljem svijeta. Većina je prolazna, ali dobar dio njih je kreirao specifične slučajeve korištenja koji odražavaju fleksibilnost kriptovaluta.

Sigurnost: Jednom kada je transfer kriptovalute autoriziran, više se ne može povratiti. Ovaj sustav zaštite protiv prijevare zahtijeva specifičan dogovor između kupca i prodavatelja u vezi povrata novca u slučaju pogreške. U konačnici, tehnike enkripcije diljem distribuiranog blockchaina i transakcijski procesi kriptovaluta su jamstvo od prijevare i krađe identiteta te osiguravatelji potrošačke privatnosti. [**32**]

3.3.2. Nedostaci kriptovaluta

3.3.2.1. Neekonomičnost sustava

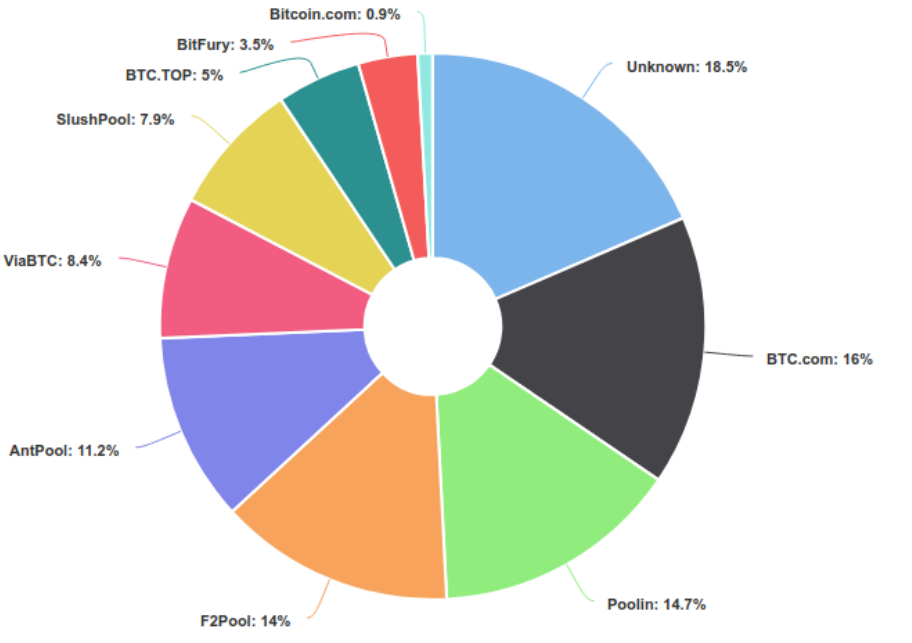
Unatoč svim inovacijama, bitcoin i dalje sadrži nedostatke. Prilikom naglog porasta cijene 2018. godine već smo doživjeli jedan od njih, a to je značajna rastuća cijena rudarenja novih bitcoina. Najveći dio te cijene su troškovi električne energije. Potrebna je poprilično velika investicija kako bi rudar ostao kompetitivan i profitabilan u procesu rudarenja. Više nije niti dovoljno posjedovati klaster računala, već specijalizirane platforme za rudarenje posebno dizajnirane da rješavaju slagalice dokaza o radu (eng. *proof of work puzzle*) što efikasnije. Iz poslovne perspektive rudarenja, već se odvija utrka "naoružavanja". Rudari kontinuirano ulažu u novi hardvare kako bi razvili prednost nad konkurencijom. U ranim počecima, bitcoin se rudario s običnim računalima. S vremenom, rudari su primjetili da mogu steći prednost koristeći grafičke kartice za rudarenje. Tu je počela utrka u dizajniranju sve efikasnijih mašina za rješavanje bitcoin slagalice. Utrka ka novijim i moćnijim tehnologijama pojavila se prvenstveno zbog strukture Bitcoin algoritma. S obzirom da osoba koja riješi slagalicu dobiva cijelu nagradu, čak i najmanja tehnološka unaprijeđenja postavljaju rudara ispred svih drugih. Inkrementalne investicije mogu se činiti male, ali u usporedbi sa svima drugima koji također ulažu u tehnologiju rudarenja, ukupna investicija "rudarske industrije" (eng. *mining industry*) može postati vrijednija od vrijednosti koju rudari mogu dobiti. Utrku za boljom tehnologijom ubrzava i specifično svojstvo Bitcoin sustava. Naime, kompleksnost kriptografske slagalice prilagođava se kako bi održala širenje blockchaina konstantnom brzinom od jednog dodanog bloka svakih 10 minuta. Uvođenje moćnijih platformi za rudarenje (eng. *mining rigs*) efektivno povećava kompleksnost slagalice. Dakle, s obzirom da rudari imaju veću procesorsku moć, riješit će slagalicu u kraćem vremenu. Iz tog razloga slagalica postaje kompleksnija, odnosno zahtijeva više kriptografskih operacija kako bi se riješila, čime usporava rudare. S druge strane, ovaj proces dovodi do povećane potrošnje električne energije. [**7**]

Još jedna zanimljiva posljedica strukture Bitcoin sustava jest pojava rudarskih bazena (eng. *mining pools*). Rudarski bazeni su udruženja rudara koji dijele procesorsku moć svojih mašina te dijele sve dobivene nagrade, tipično proporcionalno procesorskoj moći uloženoj u rudarski bazen. Stoga se samostalni rudari pokušavaju pridružiti rudarskim bazenima jer je malo vjerojatno da će samostalno riješiti slagalicu. Sudjelovanje u rudarskim bazenima omogućuje članovima dijeljenje rizika i međusobno osiguravanje jer će bazen češće riješiti slagalicu od individue. Međutim, nagrade rješavanja slagalice su manje jer se dijele diljem rudarskog bazena,

ali većina rudara preferira kontinuiran prihod čestih manjih nagrada od izuzetno rijetkih većih nagrada. [**7**]

U suštini, ovakav način rudarenja dovodi do tri kategorije troškova. Prvo, tu su troškovi vođeni potrošnjom električne energije za održavanje specijaliziranih rudarskih strojeva. Drugo, sustav potiče neravnomjernu participaciju u mreži. Najveći rudari u konačnici mogu posjedovati cijelu glavnu knjigu te istovremeno rudariti nove bitcoine koji im omogućuju daljnje ulaganje u noviju opremu za rudarenje. Treće, kao posljedicu gore navedenog, dolazi do neefikasnog investiranja u rudarsku opremu. Glavni problem cijelog sustava u konačnici predstavljaju rastući troškovi električne energije koju je potrebno uložiti za održavanje sustava, a odražavaju se na okolinu i generalnu ekonomiju. Ovaj eksterni efekt povećane potrošnje se povećava samom činjenicom da je mnoštvo komputacija kojima se pokušava riješiti slagalica u konačnici beskorisno. Svi rudari koji su se natjecali za rješavanje slagalice i sve komputacije koje su izvršili, dok jedan od njih nije pronašao rješenje, odbacuju se u vjetar. Iz te perspektive, uložena energija potrošena na odbačene komputacije je gubitak u sustavu. [**7**]

Unatoč svemu navedenom, sve dok su prihodu rudarenja veći od troškova električne energije potrebne za održavanje opreme rudarenja, rudarenje će biti ekonomično za rudare. Međutim, ova bi se situacija eventualno mogla promijeniti. Kao što smo već napomenuli, brzina kojom Bitcoin algoritam generira nove bitcoine smanjuje se s vremenom. Malo je vjerojatno da će cijena bitcoina rasti istom brzinom, što bi značilo da će se nagrada rudarenja novih jedinica valuta smanjivati. Eventualno, visoki troškovi električne energije sustići će postepeno smanjivanje profita. Ovaj faktor možda neće srušiti cijeli sustav, ali će naknade učiniti krucijalnim faktorom. Trenutno se bitcoin transakcije provode s minimalnim naknadama. Naknade će se morati povećati kako bi se podmirila razlika povećanih troškova energije i smanjenja profita rudarenja. [**7**]



Slika 8: Distribucija procesorske moći rudarskih bazena [**28**]

Uz do sada navedene nedostatke, rast elitnih rudara mogao bi postati ozbiljan problem. Bitcoin sustav održava integritet blockchaina oslanjajući se na raspršenu mrežu rudara koji se međusobno provjeravaju i održavaju iskrenima. Ovakav sustav distribuiranih provjera doživjeti će neuspjeh kada rudar, ili koordinirana grupa rudara, stekne kontrolu nad više od pola procesorske moći potrebne za održavanje mreže. U tom slučaju, elitni rudar mogao bi preuzeti kontrolu nad glavnom knjigom i izvršiti takozvani 51 postotni napad (eng. 51 percent attack). Time stječe niz moći, od onemogućavanja procesiranja novih transakcija i dodavanje istih u blockchain, do korištenja dvostrukog trošenja. Utrka s najnovijom rudarskom opremom povećava mogućnosti da dođe do ovog problema jer se manje efikasni rudari, koji ne mogu prikupiti sredstva za najnovija unaprijeđenja, tjeraju iz sustava. S vremenom, nekoliko rudara s najvećom procesnom moći dominirati će mrežu. U dodatku, rudarski bazeni također povećavaju prijetnju od 51 postotnog napada. Glavna prednost bitcoina bila je eliminacija potrebe za povjerljivim posrednikom koji bi nadgledao i rukovodio mrežom. Rudar ili rudarski bazen koji kontrolira više od pola mreže ethnički će postati povjerljiv posrednik koji će dominirati mrežom. Ironično, bilo bi upitno bi li uopće bio povjerljiv. Prijetnja 51 postotnog napada nije isključivo teoretske prirode. Sredinom 2004. godine, *Gash.io*, jedan od najvećih bitcoin rudarskih bazena, kratkotrajno je dosegao više od 50 posto procesne moći cjelokupne Bitcoin mreže. Navodno nije bilo loših intencija te nije učinjena nikakva šteta mreži. [**7**]

Još jedan nedostatak bitcoina bio bi potencionalni deflacijski pritisak ugrađen u njegovom algoritmu. Kao što smo već naveli, ukupna zaliha bitcoina u postojanju se povećava, ali se smanjuje brzina kojom se izdaju u opticaj. Ograničena količina bitcoina mogla bi predstavljati pritisak na cijenu. Zbog manje jedinica valute u opticaju, potrošači neće htjeti trošiti previše jedinica valute na neko dobro ili uslugu. Više o ovom fenomenu govori ekonomska teorija pod nazivom kvantitativna teorija novca (eng. *quantity theory of money*). Glavni razlog ograničene količine bitcoina u opticaju jest onemogućavanje inflacije. U kontekstu tradicionalnih valuta, inflaciju često potiče povećanje novca u opticaju. Bitcoin bi trebao slijediti suprotan trend prema strani deflacije. Kako bi se nadoknadila deflacijska priroda, mogli bi zamisliti gradualno povećanje novca u opticaju u Bitcoin algoritmu. Problem onda postaje točna mjera povećanja koja bi osigurala relativno konstantne cijene. Teško da bi specificirana formula mogla samostalno postići ovaj cilj, s obzirom da se to inače prepušta centralnim bankama. Unatoč tome, dio korisnika bitcoina voljan je prihvatiti potencijalne nestabilnosti cijena u zamjenu za neovisnost od institucija kao što su centralne banke. Za takve korisnike, ovo svojstvo bitcoina je pozitivno te bi moglo potaknuti veću primjenu među istomišljenicima. [**7**, **33**]

3.3.2.2. Propusti digtalnih valuta

U nastavku će se objasniti načini zašto kriptovalute koje obnašaju ulogu novca, kao što je bitcoin, ne uspijevaju zamijeniti svojstva tradicionalnih valuta. Uspješna valuta djeluje kao posrednik razmjeni, dijelu računa ili kao pohrana vrijednosti. Bitcoin sadrži slabosti u zadovoljavanju sva tri kriterija. [**11**]

S obzirom da bitcoin nema intrističku vrijednost, njegova vrijednost ovisit će o korisnosti valute u potrošačkoj ekonomiji. Trenutna uporaba bitcoina u trgovini nije proširena, no zna se

koristiti kod manjih poslovnih subjekata koji su voljni prihvaćati bitcoin. Najuspješniji trgovci koji prihvaćaju bitcoin upravo su IT tvrtke koje pružaju aplikacije, burze kriptovaluta ili hardver vezan uz bitcoin. Realističan uvid u adopciju bitcoina može se vidjeti iz podataka prikupljenih s glavne knjige svih transakcija. Većina transakcija uključuju transfere između spekulativnih investitora, a samo mali dio njih koristi se za kupnju dobara i usluga. Čini se da bitcoin i dalje ima neprimjetan udio na globalnom tržištu. Jedan od problema s kojim se susreću trgovci jest prikupljanje novih bitcoina. Ako potrošač nije uspješan rudar, morat će tražiti bitcoin od online burzi ili trećih posrednika te pronaći način da ga sigurno pohrani. Postojeće burze kriptovaluta imaju malu likvidnost, velik raspon između kupovne i potražne cijene te sadrže određene rizike za poslovne subjekte. S druge strane, potrebno je posjedovati određenu svotu kriptovalute kako bi se potrošila na dobra ili usluge. Mogućnost kupnje dobara i usluga bez novca pri ruci česta je pojava u maloprodajnim lancima jer potrošači često kupuju s kreditnim karticama. Navedena mogućnost trenutno nije dostupna s bitcoinom jer još uvijek ne postoje bitcoin kreditne kartice niti potrošački krediti izdani u bitcoinu. U konačnici, razmjena bitcoina između potrošača i trgovca podliježe verifikaciji koja u prosjeku traje 10 minuta. Trgovci mogu potrošaču pružiti povjerenje i prihvatiti transakciju, no to ih može ostaviti ranjivima na prijevare. [**11**]

Drugi problem s kojim se susreće bitcoin u ulozi valute su nagle fluktuacije cijene. S obzirom da se vrijednost bitcoina u odnosu na druge valute mijenja iz dana u dan, trgovci bi često morali ponovno računati cijene. Taj proces bio bi skup i vremenski zahtjevan za trgovca, a zbunjujuć za kupca. U praksi bi ovaj problem nestao da je bitcoin korišten kao glavna valuta, ali takvog mjesta u svijetu još uvijek nema. Povezan problem nalazi se u različitosti tržišnih cijena po kojima se može kupiti bitcoin. Cijena kriptovalute, u ovom slučaju bitcoina, ovisit će o izabranom posredniku za kupnju istoga. Takve varijacije predstavljaju narušavanje klasičnog zakona novca o postojanju jedne cijene. Razvijeno tržište valuta pod ovakvim okolnostima bilo bi predmet jednostavne arbitraže, što bi uvelo još veću pomutnju pri definiranju cijena dobara i usluga. Kao rezultat različitih cijena, trgovci koji prihvaćaju bitcoin ili druge kriptovalute kao sredstvo plaćanja uzimaju prosječnu cijenu svih burzi kriptovaluta kao referentnu točku. Međutim, ovakve agregacije ne prikazuju trgovcima i potrošačima stvarnu cijenu kupnje ili prodaje bitcoina u stvarnom vremenu. Osim svega navedenog, u usporedbi s cijenama proizvoda i usluga, tu je i problem zbog relativno visoke cijene jednog bitcoina. Iz tog razloga trgovci za većinu dobara i usluga moraju navoditi cijene bitcoina u četiri ili više decimala. Iako je matematika preračunavanja vrlo jednostavna, za potrošače bi ovo mogao s vremenom postati problem. Primjerice, ulaskom u restoran možemo si priuštiti glavno jelo za 0.05255 BTC, umak za 0.00529 BTC, desert za 0.01694 BTC, a piće za 0.01393 BTC. Prebrojavati broj decimala te uspoređivati početne brojeve kako bi se dobio uvid u visinu cijena jednostanvo nije praktično. Planiranje troškova i bilanciranje postalo bi problem, posebice jer većina postojećih aplikacija u svijetu, neovisno bile digitalne blagajne, računovodstveni softver ili nešto treće, koristi dva decimalna mjesta. Jedno od rješenja je uvođenje manjih jedinica valute, kao što su mili-bitcoini (mBTC) ili mikro-bitcoini (uBTC). [**11**]

Valuta bi trebala djelovati i kao sredstvo pohrane vrijednosti. Vlasnik valute može ju zamijeniti za dobra ili usluge u nekom trenutku u budućnosti, a kada se valuta potroši, očekuje se da će dobiti istu ekonomsku vrijednost valute u trenutku kada ju je dobio. Kroz povijest,

tretiranje valute kao pohrane vrijednosti predstavljalo je obranu od krađe fizičkim sakrivanjem ili pospremanjem u banci. Bitcoin nema fizičku manifestaciju te se mora pohranjivati na digitalnim novčanicima, što predstavlja sigurnosni problem za cjelokupnu industriju. Ako potrošač pronađe uspješan način za pohranu bitcoina, i dalje ima problem koji nastaje iz fluktuacija cijena istog. Kako bi se bitcoin postavio kao valuta, njegova dnevna vrijednost mora postati stabilnija kako bi pouzdanije služio kao pohrana vrijednosti. Trenutna promjenjivost cijena je konzistentnija s ponašanjima spekulativnih investicija nego s valutama. Osim toga, potrebna su određena informatička znanja kako bi se bitcoin efikasno koristio u svakodnevnoj upotrebi. Nadalje, bitcoinov legitimitet je i dalje upitan. Iako ga ne izdaje neovisna država, bitcoin predstavlja rizik svakom poslovnom subjektu koji ga prihvaća za izvršavanje transakcija. Transakcije su riskantne zbog nedostatka osnovne potrošačke zaštite, kao što je pružanje povrata novca u slučaju nesuglasica između prodavatelja i kupca. Iako se nesuglasice mogu riješiti zakonskim regulativama, vlade nemaju zakonskog načina da preuzmu vlasništvo nad nečijim bitcoinom te samim time nemaju mogućnosti provesti zakone vezane uz potrošačku zaštitu. [**11**]

U zaključku, bitcoin je potpuno odvojen od bankarskog sustava i sustava plaćanja. Većina valuta drži se i prenosi putem bankovnih računa, koji su zaštićeni slojevima regulacija, osiguranja depozita te internacionalnih sporazuma. Bez pristupa ovoj infrastrukturi, bitcoin je ranjiv na prijevare, krađu i subverzije. Međutim, pristaše bitcoina nalažu da bitcoin prelazi sve poznate propuste standardnih financijskih sigurnosnih sustava (eng. *standard financial security systems*), koji su podležni epidemijama krađe identiteta i povezanih problema za prosječne korisnike i veće poslovne subjekte. S druge strane, i dalje postoji dugoročni ekonomski problem vezan uz apsolutni limit od 21 milijuna bitcoina koji će ikada biti u egzistenciji bez mogućnosti povećanja zaliha nakon 2040. godine. Ako bitcoin postane uspješan i zamijeni tradicionalne valute, ima bi deflacijsku ulogu u ekonomiji jer se opskrba novca ne bi povećavala s ekonomskim rastom. Ekonomski stručnjaci smatraju da bi ovakva situacija dovela do godišnjih rezanja plaća, što bi dovelo do političkih nemira. [**11**]

3.3.3. Alternativne kriptovalute

U prethodnom odlomku objasnili smo neugodne eksterne okolnosti neophodne za održavanje Bitcoin sustava, kao što su visoka potrošnja električne energije i deflacijski pritisak. S obzirom da je Bitcoin algoritam javno dostupan i besplatan da se koristi i kopira, pojavio se broj alternativnih kriptovaluta, poznatijih pod imenom *altcoins*. Altcoinovi rješavaju stvarne, a nekad samo pretpostavljene, probleme u dizajnu Bitcoin sustava. U mnoštvu slučajeva, takve kriptovalute djeluju na izuzetno sličan način kao i originalni Bitcoin. Dakle, nemaju centralni autoritet, tj. povjerljivog posrednika koji bi nadgledao i rukovodio transakcijama. Oslanjaju se na kriptografiju kako bi održavale distribuiranu glavnu knjigu (blockchain) koja odražava sve transakcije određene kriptovalute. Većina ima isti sustav rudarenja, odnosno provjeravanja transakcija i rješavanje kompleksnih matematičkih problema kako bi dodali novu transakciju u glavnu knjigu. Kreiranje novog altcoina u današnje vrijeme nema velikih prepreka. Bitcoin je krajem 2013. godine privukao pozornost izvan kriptografske zajednice, nakon čega je broj kriptovaluta baziranih na bitcoinovom protokolu naglo porastao. Neke od tih kriptovaluta čista su preslika bitcoina, a

neke se razlikuju u tehničkim detaljima. Dobar dio takvih kriptovaluta ima sličnu namjenu kao i bitcoin, da zamijenjuju novac i služe kao sredstvo razmjene. Postoje i radikalnija rješenja, od onih koja uvode značajnije promjene u algoritam do onih koje su kreirane s potpuno novim principom rada koji se ne oslanja na blockchain. Nove kriptovalute time ne samo da mijenjaju nedostatke koji se pojavljuju s Bitcoin algoritmom, već i ekonomske aspekte kriptovalute jer se nužno ne koriste kao sredstvo razmjene. [**7**, **11**]

Gotovo sve kriptovalute koje su dizajnirane kao ekvivalent novcu baziraju se na Bitcoin algoritmu i njegovom načinu rješavanja problema dvostrukog trošenja. Međutim, rješenje koje je predložio Satoshi Nakamoto nije ograničeno na digitalne valute. Koncept blockchaina može se generalizirati na širok spektar drugih primjena, a samo neke od njih navest ćemo u nastavku. [**7**, **11**]

Litecoin je jedan od primjera preslika Bitcoinovog sustava s tri ključne razlike. Mreža Litecoina nastoji generirati blok svakih 2 i pol minute, umjesto 10 minutnih intervala Bitcoina. Ovaj pristup omogućuje brže potvrde transakcija te brže dostizanje konsenzusa u mreži. Osim toga, Litecoin bi stoga bio primjereniji za brza plaćanja gdje je vremenski raspon između razmjene dobara, usluga i novca kratak. Nadalje, ukupan limit jedinica valute Litcoin mreže jest 84 milijuna litecoinova, što je četiri puta više nego planirani broj bitcoina u opticaju. Ključna razlika između Litecoina i Bitcoina leži u činjenici da Litecoin koristi scrypt. Scrypt je sekvencijalna memorijska funkcija (eng. *sequential memory-hard function*) koja smanjuje prednost procesorski moćnih rudara. Naime, scrypt zahtijeva velike količine memorije kako bi se efektivno računao, čime se balansira procesorske i memorijske resurse. Vremenski pristup memoriji varira puno manje nego brzine procesora, čime se omogućuje poštenija alternativa pružanja dokaza o radu. Trenutno postoji nekoliko specijaliziranih ASIC rudarskih hardvera dostupnih za sustave dokaza o radu koji su bazirani na scryptu. [**8**, **12**]

Dogecoin još više smanjuje konfirmacijska vremena transakcija na gotovo 1 minutu. Direktna posljedica je malo veća mogućnost generiranja blokova koji ne pripadaju glavnom blockchainu. Slično Litecoinu, Dogecoin se također oslanja na scrypt. Ukupna zaliha dogecoina koja se predviđa izrudariti jest 100 milijardi. Glavna svrha dogecoina jest sustav davanja napojnica za zanimljive i kvalitetne sadržaje. [**8**]

Namecoin je predstavljen 2010. godine s ciljem povećanja anonimnosti aktivnosti na internetu. Namecoin sustav je decentralizirana usluga smještaja (eng. *decentralized hosting*) i održavanja web domene *".bit"*. Naspram tradicionalnih web stranica, cilj je onemogućavanje bilo kojem entitetu da preuzme kontrolu i ugasi web stranicu. Namecoin sustav koristi nativnu valutu, denominiranu u namecoin-ovima, za plaćanja i obnavljanja web stranica u .bit domeni putem Namecoin blockchaina. U ovom kontekstu, dizajn valute namecoina ne razlikuje se od bitcoina. Dapače, mogu se rudariti istodobno u istom procesu. [**7**, **12**]

Ethereum je još jedna blockchain inovacija. Sustav je dizajniran 2011. godine te objavljen 2015. godine. Osobe koje su razvile Ethereum opisuju ga kao platformu za decentralizirane aplikacije. Koristi sličnu tehnologiju kao i druge kriptovalute, ali umjesto kreiranja decentralizirane mreže za slanje transakcija, cilj mu je razviti mrežu koja podržava Ethereum ugovore (eng. *Ethereum contracts*). Ti ugovori mogli bi omogućiti usluge kao što su objave sadržaja, dina-

mično razmjenjivanje poruka i transakcija, ali na potpuno decentraliziran i psudoniman način. Ethereum se više smatra razvojnim orkužjem ili jezikom kojim se mogu pisati pametni ugovori. Ugovori su aplikacije koje imaju vlastita pravila za vlasništvo, transakcije i tako dalje. Pametni ugovori mogu pronaći primjenu u brojnim slučajevima, od sustava glasanja, intelektualne imovine pa do financijskih razmjena. [**7**, **10**]

Koncepti blockchaina i decentralizirane glavne knjige mogu se koristiti s protokolima koji se razlikuju od Bitcoin-ovog. Najpoznatiji takav alternativni sustav jest **Ripple**. Ripple je mreža za plaćanje koju je razvio Ripple Labs, tvrtka u Vancouveru koja je prije bila poznatija pod imenom OpenCoin. Tvrtka je razvila Ripple kako bi olakšala trgovinu u različitim valutama (kriptovalutama i tradicionalnim valutama). Primjerice, omogućavala bi internacionalne doznake koje bi bile jeftinije od onih dostupnih od tradicionalnih pružatelja te usluge, kao što su banke. Ripple mreža uvodi decentraliziranu, otvorenu glavnu knjigu koja prati ponude sudionika za razmjenom različitih valuta. Kako bi se izvršila razmjena, sustav koristi ripple (skraćeno XRP), koji predstavlja posredničku kriptovalutu. Naspram većini kriptovaluta, sav ripple je već izrudaren i u dostupan u opticaju. Ripple se može kupiti isključivo od Ripple Labs-a ili privatnih skupina, no ne mogu se generirati nove jedinice valute, niti postoji ekvivalent rudarenju. U slučaju korištenja ripple-a za prijenos jedinica valute, pošalje se zahtjev na jedan od čvorova mreže. Taj čvor nađe čvor na željenoj lokaciji putem lanca čvorova posrednika koji se nalaze između njih. Umjesto da šalju jedinice valute do destinacije, što bi zahtjevalo infrastrukturu tradicionalnih financijskih institucija i dosta vremena, izvorišni čvor pošalje XRP ekvivalent poslanog novca na destinacijski čvor. Destinacijski čvor onda na kraju transakcije razmijenjuje rippl-ove u željenu valutu. S obzirom da zaobilazi tradicionalnu financijsku infrastrukturu, transferi bi trebali biti jeftiniji od tradicionalnih usluga. Prema svom dizajnu, Ripple mreža bi mogla biti privlačnija financijskim institucijama nego individualnim potrošačima. S obzirom da bi potrošač morao pronaći ripple čvor kako bi mogao koristiti sustav, lakše bi bilo kada bi čvorovi bili locirani u banci po potrošačevom izboru. Prednost Ripple mreže za banku bila bi globalna dostupnost i mogućnost slanja transakcija u stvarnom vremenu. Ripple se time ne postavlja kao konkurent bitcoinu ili drugim kriptovalutama, pa čak niti onima koje izdaju vlade. Ripple je pridobio znatnu popularnost tijekom poslijednjih nekoliko godina, što mu je omogućilo poprilično tržišnu kapitalizaciju. U dodatku, institucije u tradicionalnom financijskom sustavu zainteresirane su za modernizaciju svojih načina plaćanja te se okreću ga Ripple tehnologiji. Prva institucija koja je integrirala Ripple protokol bila je Njemačka banka Fidor 2011. godine. Nekoliko mjeseci kasnije, trend su slijedile dvije banke SAD-a, te je tako pokrenut cijeli lanac integracija diljem svijeta. [**7**]

#	Coin	Price	Direct Vol.	Total Vol.	Top Tier Vol.	Market Cap	7d Chart (USD)	Chg. 24H
1	Bitcoin BTC	$ 10,216.68	$ 250.99 M	$ 2.16 B	$ 1.01 B	$ 183.30 B		0.58%
2	Ethereum ETH	$ 201.29	$ 82.05 M	$ 1.62 B	$ 484.18 M	$ 21.69 B		6.22%
3	EOS EOS	$ 4.103	$ 4.11 M	$ 544.73 M	$ 168.95 M	$ 4.19 B		2.57%
4	XRP XRP	$ 0.2743	$ 27.45 M	$ 358.21 M	$ 161.36 M	$ 27.43 B		6.15%
5	Litecoin LTC	$ 73.74	$ 19.87 M	$ 539.40 M	$ 125.22 M	$ 4.65 B		5.54%
6	Chainlink LINK	$ 1.603	$ 1.61 M	$ 124.75 M	$ 121.99 M	$ 1.60 B		3.42%
7	Bitcoin Cash BCH	$ 314.37	$ 31.71 M	$ 263.15 M	$ 105.24 M	$ 5.66 B		3.89%
8	TRON TRX	$ 0.01620	-	$ 106.97 M	$ 67.81 M	$ 1.08 B		-0.28%
9	Binance Coin BNB	$ 20.94	-	$ 67.52 M	$ 50.29 M	$ 3.26 B		4.17%
10	Ethereum Classic ETC	$ 6.277	$ 1.12 M	$ 128.40 M	$ 47.72 M	$ 712.70 M		1.75%

Slika 9: Top 10 kriptovaluta po ukupnom volumenu [**29**]

3.4. Ekosustav kriptovaluta

Alati za rukovođenje kriptovaluta omogućuju korisnicima da rukovode jednom ili više ključnih zadataka kriptovaluta. Mreža i sloj blockchaina kod bitcoina i drugih kriptovaluta nije bitan samo za integritet cijelog sustava, već ima i značajan utjecaj na sigurnost i privatnost svakog korisnika. Većina trenutno dostupnih alata, poznatijih kao digitalni novčanici, pružaju mogućnosti izvan spremanja ključeva, kao što su provođenje transakcija te preuzimanje dijelova blockchaina. U sučaju bitcoina, interakcija s bitcoin mrežom cjelovit je dio djelovanja distribuiranog sustava. Naspram drugih sustava, bitcoinovi alati moraju sadržavati informacije o obavljenim transakcijama i stanjima računa. Korisnici bitcoina imaju najširi odabir alata za rukovođenje digitalne imovine. U terminologiji bitcoina, takvi se alati trenutno referenciraju kao digitalni novčanici (eng. *wallets*). Originalno, digitalni novčanik bio je kolekcija privatnih ključeva. Stoga, bilo što od reprezentacije privatnog ključa do dediciranog softvera može biti smatrano digitalnim novčanikom. [**1**]

3.4.1. Posrednici transakcija

Jedna od najvećih prednosti decentralizirane prirode kriptovalute jest da bilo tko može početi prihvaćati plaćanja bez potrebe za registracijom računa s nekim centralnim autoritetom. Međutim, mnogim poslovnim subjektima potrebno je izvjesno vrijeme da se prilagode novim tehnologijama i sustavima. Instantna konverzija kriptovaluta u bilo koju novčanu valutu (dolar, euro, yuan) je jedna od najpopularnijih usluga koju pružaju posrednici plaćanja (eng. *payment processors*). Ova je usluga neophodna za sve poslovne subjekte koji prihvaćaju plaćanja putem kriptovaluta, ali i dalje moraju platiti sve ili dio vlastitih troškova koristeći novčane valute. Instantna konverzija smanjuje rizik gubitaka na fluktuacijama burzovnih tečajeva između kriptovaluta i novčanih valuta. Instantne transakcije nisu jedina usluga posrednika transakcija jer inače pružaju cijeli skup alata i usluga koje olakšavaju prihvaćanje i rukovođenje kriptovalutama. [**8**]

Sustavi plaćanja mogu se podijeliti u dvije osnovne kategorije. U prvoj kategoriji su pla-

ćanja između dvije osobe (eng. *person-to-person*) koja pogoduju manjim poslovnim subjektima jer predstavljaju najjednostavniji način prihvaćanja kriptovaluta. U drugoj kategoriji su rješenja ovisna o prodajnoj točki (eng. *point-of-sale - POS*), koja pogoduju većim organizacijama. U kontekstu plaćanja između dvije osobe, korisnik šalje potrebnu količinu kriptovalute direktno na digitalni novčanik trgovca. Kako bi se ubrzao ovaj proces, *CoinBox*, jedna od vodećih trgovačkih platformi, pruža zanimljivo rješenje. Trgovac, koristeći aplikaciju na svom pametnom telefonu, može pretvoriti cijenu dobra ili usluge u QR kod koji sadrži količinu koja mora biti plaćena i adresu primatelja, tj. svog računa. Mušterija potom skenira QR kod sa svojom aplikacijom za digitalni novčanih i time plaća traženu cijenu. Unatoč svojoj jednostavnosti, sustavi plaćanja između dvije osobe teško da će se koristiti u većim organizacijama koje su zainteresirane u prihvaćanje kriptovaluta. Stoga, tržište omogućuje brojna POS rješenja koja trgovac može birati kako bi zadovoljio specifične potrebe. U nastavku će se navesti samo neka u mnoštvu rješenja. [**8**]

Coinify je Danska tvrtka koja pruža POS rješenja, omogućujući prihvaćanje plaćanja bilogdje od bilokoga. Trgovci trenutno mogu biti plaćeni u 17 različitih digitalnih valuta, većini novčanih valuta ili čak kombinaciji toga dvoje. [**8**, **34**]

CoinKite je start-up koji omogućuje sustav plaćanja uz pomoć bitcoina koji izgleda kao tipični čip i PIN terminali (eng. *chip-and-PIN terminals*) koji se nalaze u trgovinama. Ovakav uređaj može čitati debitnu karticu baziranu na bitcoinu te čak služiti kao bankomat za bitcoin i litecoin. U dodatku, ima opcije ispisivanja QR koda korisnicima kako bi ih skenirali mobilnim aplikacijama. [**8**, **35**]

BitPay je internacionalni posrednik transakcija za poslovne subjekte i dobrotvorne ustanove. Integriran je u SoftTouch POS sustav za maloprodajne trgovine. BitPay ima i API koji se može jednostavno integrirati s bilo kojim drugim POS sustavom. Pruža i različite tarife na koje se mogu pretplatiti trgovci te koristiti određene funkcionalnosti na svojim online trgovinama. [**8**, **36**]

Revel Systems pruža različita POS rješenja za iPad-ove kako bi zadovoljio različite kategorije trgovaca, od restorana do maloprodajnih mjesta. Podržavaju bitcoin kao sredstvo plaćanja. [**8**, **37**]

Paystand Bitcoin Merchants razvija sučelje za višestruka plaćanja koji eliminira transakcijske naknade trgovaca te time podupire prihvaćanje digitalnih valuta. [**8**, **38**]

XBTerminal pruža bitcoin POS uređaj koji mušterijama omogućuje plaćanje s bilo kojeg mobilnog digitalnog novčanika putem NFC-a ili QR koda. Omogućuje i plaćanje izvan mreže putem Blutetooth-a. Transakcije se provode kroz platformu tvrtke. Ujedno omogućuju instantnu konverziju novčanih valuta u trenutku kupnje. [**8**, **39**]

3.4.2. Digitalni novčanik

Ekvivalent računa u banci u kripto svijetu jest digitalni novčanik. Digitalni novčanik je softverski program koji pohranjuje privatne i javne ključeve. Djeluje s različitim blockchain tehnologijama kako bi korisnicima omogućio slanje i primanje kriptovaluta. Kreira se na nekoj od

internetskih stranica koje pružaju tu uslugu. Svaka provedena transakcija uređeni je digitalni zapis koji se sastoji od definirane količine prenesenih jedinica kriptovalute te određenih javnih i privatnih ključeva adresa digitalnih novčanika pošiljatelja i primatelja. Svaku transakciju pošiljatelj potpisuje svojim privatnim ključem, čime se transakcija potvrđuje i zapisuje u mreži. Pošiljateljev potpis osigurava da nitko ne može kompromitirati sadržaj transakcije. Dakle, prije kupovine kriptovaluta, potrebno je napraviti digitalni novčanik na koji će kupljene kriptovalute biti uplaćene. Digitalni novčanik je mobilna aplikacija ili računalni program koji nam omogućuje interakciju s mrežom kriptovaluta. Svaki novčanik sadrži adresu, npr. kao što je *1jfkDFF-KELr393fjdLFkdkeirffj34re3RRDFJKR9*, pomoću koje primamo, šaljemo i spremamo kriptovalute. Ujedno sadrži i primarni ključ s kojim možemo pristupiti novčaniku u slučaju da nemamo pristup računalu ili mobitelu na koji smo ga prvotno instalirali. Privatni ključ ne dajemo nikome, čuvamo ga zapisanog ili spremljenog na eksternoj memoriji ili nekom drugom sigurnom mjestu. Većina novčanika ima dodatnu mogućnost zaštite lozinkom ili otiska prsta, što se svakako preporučuje. Ujedno valja napomenuti da ne smijemo slati jednu kriptovalutu na novčanik druge. Smijemo i možemo to napraviti, ali ćemo tim postupkom trajno izgubiti sredstva koja smo poslali bez mogućnosti povrata. Novije verzije digitalnih novčanika automatski upozore ako upišemo adresu druge kriptovalute kojoj ne pripada ona koju šaljemo. No bolje je steći praksu provjere adrese na koju šaljemo i ne raditi takve pogreške. [**1**, **11**, **12**]

Unutar aplikacija za digitalni novčanik, možemo napraviti više novčanika za određenu kriptovalutu, gdje svaki ima svoju zasebnu adresu za primanje, slanje ili spremanje kriptovaluta. Određeni programi za digitalni novčanik imaju funkciju izrade novčanika za više od jedne kriptovalute, dok je većina napravljena samo za jednu. Način na koji šaljemo sredstva i upisujemo adresu primatelja i količinu te ovjeravamo slanje transakcije pinom ili otiskom prsta koji smo postavili. Transakcije također ne sadržavaju nikakve osobne podatke koji su pod rizikom krađe identiteta za razliku od klasičnih bankovnih transakcija. Digitalni novčanici mogu biti podržavani mobilno, putem weba, desktopa ili hardware-a. [**1**, **11**, **12**]

3.4.2.1. Vrste digitalnih novčanika

Kao što smo napomenuli ranije, glavni cilj digitalni novčanika jest sigurna pohrana privatnih ključeva potrebnih za trošenje kriptovaluta koje posjedujemo. Najjeftiniji način sigurne pohrane kriptovaluta jest korištenjem papirnatih novčanika (eng. *paper wallets*). **Papirnati novčanik** jest servis koji generira adresu za korisnika te sliku koja se sastoji od dva QR koda. Prvi enkodira adresu na koju korisnik može primati kriptovalutu, a drugi enkodira privatni ključ te se koristi za slanje kriptovaluta s adrese dodijeljene korisniku. [**8**, **12**, **40**]

S vremenom se pojavio i niz **mobilnih novčanika** koji omogućuju plaćanja putem pametnih telefona. Takvi novčanici lokalno pohranjuju privatne ključeve potrebne za trošenje kriptovaluta te omogućuju plaćanja direktno s mobilnog uređaja. Nadalje, NFC tehnologija može ubrzati proces plaćanja. Prislanjajući mobilni uređaj uz čitač, plaćanje se može izvršiti bez dodatnih interakcija. Mobilni novčanici omogućeni su uz pomoć jednostavnog mehanizma plaćanja (eng. *simple payment mechanism, skraćeno SPV*). SPV omogućuje pametnom mobitelu da provjeri da li je transakcija uključena u blockchainu bez da preuzima cjelokupni blockchain.

Mobilni novčanici praktičan su način za prijenos i trošenje kriptovaluta usput. Kod većine mobilnih novčanika ključevi se spremaju i rukovode na serverima, a računi su zaštićeni vlastitim korisničkim imenom i lozinkom. Postoje i mobilni novčanici koji spremaju ključeve na mobitelu. Spremanje ključeva u lokalnoj memoriji mobitela može biti riskantno. U slučaju gubitka ili krađe mobitela, većina mobilnih novčanika pruža opciju zaštite i povratka računa s nasumično odabrane 24 riječi. [**8**, **12**, **40**]

S druge strane, klasični **desktop novčanici** pružaju puno naprednije usluge, kao što su podrška za anonimnost transakcije (eng. *transaction anonymization*) kako bi se onemogućilo praćenje. Namijenjeni su za korisnike koji preferiraju veću kontrolu nad svojim kriptovalutama. Desktop novčanici su aplikacije koje se vrte na računalu i direktno povezuju na mrežu kriptovalute. Lokalna instalacija desktop novčanika omogućuje potpunu kontrolu nad digitalnim novčanikom i privatnim ključevima. Rizici korištenja desktop novčanika svode se na hardverska oštećenja, računalne viruse te neautoriziran pristup. [**8**, **12**, **40**]

Treći tip digitalnih novčanika su online novčanici. **Online novčanici** su bazirani na webu te spermaju privatne ključeve u cloudu, pružajući pritom visoku razinu dostupnosti. Glavne karakteristike online novčanika su: pristup s bilo koje lokacije i uređaja, online kreiranje i pohrana privatnih ključeva, opcije slanja i primanja putem email adrese, često ugrađeni u burze za instantnu kupnju i prodaju kriptovalute, brza i jednostavna registracija te osiguranje dvofaktornom autentikacijom. Međutim, ovakvi novčanici imaju ključni nedostatak. Naime, privatni ključevi nisu pod kontrolom vlasnika, što predstavlja ozbiljan rizik. [**8**, **12**, **40**]

Nadalje, postoje i hardverski novčanici. **Hardverski novčanici** su dedicirani uređaji koji digitalno pohranjuju privatne ključeve te tako asistiraju prilikom plaćanja. Tržište pruža različite uređaje koji su često certificirani protiv različitih vrsta napada, fizičkih i logičkih. U slučaju hardverski novčanika, ključni problem jest vraćanje ključeva u slučaju kvara na hardveru. Kombinacija usluga koje pružaju online novčanici u kombinaciji s sigurnošću koju pružaju hardverski novčanici najbolja su opcija za spriječavanje krađe kriptovaluta. [**8**, **12**, **40**]

S obzirom da postoji mnoštvo kriptovaluta, tehnički je potrebno posjedovati više digitalnih novčanika kako bi se pohranile različite kriptovalute. Jedno od rješenja ovog problema su takozvani **viševalutni novčanici** (eng. *multicurrency wallets*) koji unificiraju rukovođenje mnoštva kriptovaluta unutar jednog sučeja. [**8**, **40**]

3.4.2.2. Kupovina kriptovaluta

Kada napravimo digitalni novčanik, možemo kupiti bitcoin na jedan od sljedećih načina:

- Kupovinom preko bankomata kriptovaluta, gdje fizički moramo doći do bankomata i novčanice uplatiti u njega, nakon čega nam on uplati kriptovalutu na digitalni novčanik koji smo preuzeli na mobilni uređaj. Za tu uslugu bankomat uglavnom uzme oko 10 posto provizije. Bitcoin bankomati se nalaze na više lokacija u Hrvatskoj, a možemo ih pronaći na web stranici *https://coinatmradar.com*. [**11**, **12**, **40**, **41**]
- Drugi način je kupovinom preko online hrvatske mijenjačnice *bitcoin-mjenjacnica.hr* uplatom novca putem bankovnog transfera možemo kupiti bitcoin ili druge kriptovalute. Provi-

zija koju uzimaju je oko 3.5 posto i kupvinu je moguće obaviti na brz i jednostavan način. [**11**, **12**, **40**, **42**]

- Treći način je kupovina preko online Američke mjenjačnice *coinbase.com*. Uplatom novca putem bankovnog transfera možemo kupiti bitcoin ili druge kriptovalute. Provizija koju uzimaju je oko 4 posto i kupovinu je također moguće obaviti na brz i jednostavan način. [**11**, **12**, **40**, **43**]

- Četvrti način kupovine je uplata direktno na burzu. Ovu mogućnost nemaju sve burze. Jedna od onih koja tu mogućnost pruža je *pro.coinbase.com*. Uplatom novca direktno na burzu putem bankovnog transfera možemo kupiti bitcoin i druge kriptovalute. Ovaj način se značajno razlikuje od ostalih koje smo spomenuli po tome što moramo poznavati osnovne funkcionalnosti na burzi i sami postaviti ponudu za kupnju kriptovalute po određenoj cijeni. Ovaj način kupovine uzmima proviziju banke za prijenos novca s računa i oko 0.3 posto provizije prilikom kupovine na burzi. Negativna strana ovog načina kupovine je što ponekad može uzeti više vremena od prethodno navedenih opcija. [**11**, **12**, **40**, **44**]

- Posljednja opcija koju ćemo spomenuti jest kupovina direktno od osobe za koju znamo da nam želi prodati određenu kriptovalutu. Takvu osobu najčešće možemo pronaći putem društvenih mreža, lokalnih poznanstava ili putem web stranice *localbitcoins.com* koja izgleda kao oglasnik za kupovinu i prodaju. Ovaj način kupnje je pod rizikom prijevare, pa ga osobno ne preporučujem. [**11**, **12**, **40**, **45**]

3.4.3. Burze kriptovaluta

Burze kriptovaluta djeluju kao posrednici nacionalnih valuta i kriptovaluta. Stoga su predmet nacionalnih i internacionalnih regulacija, često specifičnih jednoj državi ili ekonomskoj regiji. [**2**, **8**]

1	Coinbase	U.S.A	Stable Coins, Tokens	$ 152.65 M	60.30	AA	★★★☆☆
2	Poloniex	U.S.A	Cryptocurrency, Stable Coins, Tokens	$ 16.29 M	59.90	AA	★★☆☆☆
3	Bitstamp	United Kingdom	Cryptocurrency, Fiat	$ 110.41 M	59.60	AA	★★★☆☆
4	bitFlyer	Japan	Cryptocurrency, Fiat	$ 23.04 M	57.20	AA	★★★☆☆
5	Liquid	Japan	Cryptocurrency, Fiat	$ 163.46 M	56.30	AA	★★★☆☆
6	itBit	U.S.A	Cryptocurrency	$ 4.33 M	56.00	AA	★★★☆☆
7	Kraken	U.S.A	Cryptocurrency, Derivatives, Stable Coins, Tokens, Fiat	$ 108.47 M	54.10	A	★★☆☆☆
8	Binance	Malta	Cryptocurrency, Stable Coins, Tokens	$ 740.91 M	54.00	A	★★★★☆
9	Gemini	U.S.A	Cryptocurrency, Fiat	$ 11.99 M	53.20	A	★★★☆☆
10	Bithumb	South Korea	Cryptocurrency, Tokens, Fiat	$ 339.39 M	53.10	A	★★★☆☆

Slika 10: Top 10 burzi kriptovaluta prema ocijenjenoj kvaliteti [**29**]

Burze kriptovaluta su platforme koje spajaju kupce i prodavatelje te im omogućuju da jednostavno razmijenjuju kriptovalute. Slično tradicionalnim financijskim burzama, pružaju korisnicima priliku da koriste strategije slične onima na burzama dionica. Korisnik s računom na nekoj od burzi kriptovaluta prvo mora izvršiti depozit novca u valuti koju podržava burza koju koristi. Postoje brojna rješenja koja ubrzavaju proces depozita i kupnje kriptovaluta. Primjerice, *Coinbase* pruža opciju spajanja korisničkog bankovnog računa sa svojim digitalnim novčanikom. Time omogućuje automatsku kupnju kriptovaluta u pravilnim intervalima. S druge strane, *BitStamp* i mnoge druge burze djeluju kao posrednici koji omogućuju trgovanje s drugim korisnicima. Nakon izvršenog depozita, korisnik može trgovati s drugim korisnicima iste burze kriptovaluta ili sa samom burzom. Na burzi možemo vidjeti ukupno stanje računa, uplate, isplate te povijest transakcija. Nadalje, možemo vidjeti otvorene narudžbe, povijest narudžbi te ukupnu povijest svog trgovanja. [**7**, **8**, **12**]

Osoba može razmijeniti kriptovalute po trenutnoj cijeni, ali isto tako može staviti i limit narudžbe, tj. dati instrukcije burzi da kupi ili proda određenu količinu kriptovalute u budućnosti ukoliko cijena dosegne zadani cilj. Kupnjom korisnik traži ponudu da zamijeni neku novčanu ili digitalnu valutu koju posjeduje na svom računu za neku drugu kriptovalutu, a prodajom nudi određenu kriptovalutu po zadanoj cijeni za neku drugu kriptovalutu ili novčanu valutu. Razmijena se može izvršiti ako je cijena tražene cijene kupnje (eng. *buy order*) veća od tražene cijene prodaje (eng. *sell order*). Sve dok se burza ponaša u zadanim okvirima, nema rizika od gubitka novca. Ono najvažnije je da burze za razliku od mijenjačnica ne kupuju i prodaju kriptovalute za svoj račun. Oni samo spajaju kupce i prodavače i u tom procesu uzimaju postotak prilikom svake transakcije. Primjerice, *Coinbase* je mijenjačnica. Od njih direktno korisnik kupuje kriptovalutu, dok s druge strane, *pro.coinbase* je burza, gdje korisnik prebaci novac na svoj račun te sami stavljaju ponudu za kupnju kriptovalute po određenoj cijeni. Danas na svjetskom tržištu postoji preko 200 burzi kriptovaluta. Neke od njih su: Binance, Bittrex, Poloniex, Bitfinex, Bitmex, Bitpanda i mnoge druge. [**7**, **8**, **12**]

3.4.3.1. Povijest online burzi

Do sada smo pokrili stjecanje jedinica određene kriptovalute isključivo putem rudarenja, no s obzirom da je postalo izuzetno kompetitivno, potrebni su poprilični resursi i znanje ako netko smatra prikupljati jedinice kriptovalute na ovaj način. Slično, kriptovalute se mogu zaraditi ako prihvaćate plaćanja za svoje proizvode ili usluge putem te kriptovalute. Međutim, rijetki će danas pokrenuti biznis samo kako bi prikupili kriptovalute i potrošili ih na nešto drugo. Srećom, danas postoje jednostavniji načini nabavljanja kriptovaluta kao što je bitcoin, a to je kupnjom od drugih ljudi. Kupnja kriptovaluta konceptualno je slična razmjeni jedne valute (npr. euro) za drugu (npr. kunu). U ovom poglavlju objasnit ćemo različite načine provođenja takvih transakcija, s fokusom na online burzama koje omogućuju takve razmjene. Kada bi kriptovalute postale javno prihvaćene, burze bi postale važan faktor u financijskom sustavu. Bez njih, veće razmjene između kriptovaluta bile bi teške, što bi bila ogromna zapreka daljnjem razvoju kriptovaluta i njihovoj ulozi u ekonomiji. [**7**]

Jednostavan način da prosječna osoba nabavi određenu kriptovalutu jest da direktno

nađe prodavača. Okupljanja su bila najstariji način da ljudi kupe bitcoin bez da sami postanu rudari. Ljudi zainteresirani za trgovanje koordinirali bi se putem interneta, koristeći forume i email. Našli bi se na određenoj lokaciji u stvarnom svijetu i trgovali. Kupac bi platio tradicionalnom valutom, nakon čega bi prodavatelj pokrenuo transfer bitcoina. S vremenom su se u nekoliko država pojavili i bitcoin automati (eng. *Bitcoin Automatic Teller Machine, skraćeno BTM*) koji su omogućavali jednostavniju razmjenu tradicionalnih valuta i bitcoina. Prvi takav bankomat predstavljen je u Vancouveru u Kanadi, a poslije toga u nizu država, od Argentine do SAD-a. Na početku su omogućavale samo kupnju bitcoina, ali nisu bile dizajnirane za njegovu prodaju. Većina istih i dalje ima ovo ograničenje, no većina novijih modela omogućuje dvosmjernu razmjenu. Osobna okupljanja i BTM-ovi mogu zadovoljiti potražnju prosječne osobe za kriptovalutama, ali teško da bi bili dovoljni za potrebe šire ekonomije. Kako bi ekonomija bazirana na kriptovaluti tečno djelovala, potreban je način da se provode veleprodajne transakcije. Online burze kriptovaluta pružaju jedan od načina. [**7**]

Online burza kriptovaluta je dvostrana platforma koja povezuje kupce i prodavače te im omogućuje da trguju kriptovalute koje posjeduju. Konceptualno, burze su slične tradicionalnim financijskim burzama. Pruža korisnicima priliku da razviju strategije slične onima koje bi koristili na tržištu dionica (eng. *stock market*). Primjerice, kriptovaluta se može trgovati po cijeni u datom trenutku u vremenu, ali se mogu i staviti nalozi za nadružbu (eng. *limit order*). Nalozima za narudžbu daje se instrukcija burzi da kupi ili proda na naše ime u nekom trnutku u budućnosti, pod uvjetom da cijena bude dovoljno niska ili visoka, ovisno o tome što smo specificirali. Burze se vežu za tradicionalne financijske sustave, čime omogućuju korisnicima da prebace novac nacionalne valute s bankovnog računa na otvoreni račun burze kriptovaluta i obrnuto. Burze inače ne trguju kriptovalutama s vlastitim rčunom, nego samo spajaju kupce i prodavače. Dakle, burze su samo posrednici koji pružaju uslugu spajanja kupaca i prodavača koji su voljni trgovati po određenoj cijeni. Burze kriptovaluta još su uvijek veoma mlade i dinamične. Često se pojavljuju nove burze kriptovaluta koje se natječu s odveć postojećima i uspješno preuzimaju vodstvo po aktivnosti i broju provedenih transakcija. U nastavku će se objasniti kratka povijest Mt. Gox-a, burze kriptovaluta koja je najpoznatija javnosti, ali i najbitnija po pitanju trgovanja bitcoin-om. [**7**]

MT. Gox bio je burza kriptovaluta sa sjedištem u Tokiju u Japanu. Bila je najbitnija burza za razmjenu bitcoina u prvim godinama postojanja bitcoina. Prema nekim procjenama, Mt. Gox bio je odgovoran za rukovođenje više od 90 posto svih bitcoin transakcija. Njegova veličina i značaj na tržištu nije privlačila samo korisnike, već i napadače. U 2011. godini, burzu je komprimirao haker koji je uspio iskoristiti propuštenu ranjivost stranice. Naime, privremeno je smanjio cijenu bitcoina koju je izlistavala burza i potom si poslao velike količine bitcoina po umjetno sniženoj cijeni. Mt. Gox se oporavio od napada, ali je izgubio velik tržišni udio zbog privremene ranjivosti. Unatoč problemima, Mt. Gox ostao je najdominantnija burza kriptovaluta do sredine 2013. godine. Početkom 2013. godine, korisnici iz SAD-a imali su problema s pristupanjem stranici. Inače, korisnike SAD-a posluživali su putem bankovnog računa jedne od podružnica Mt. Goxa čije je račune zamrznuo FBI. Kroz narednih nekoliko mjeseci, Mt. Gox je izgubio svoju dominantnu poziciju, kontrolirajući samo 27 posto tržišta. Preostali udio tržišta u to vrijeme su podijelile burze BTC China s 35 posto tržišnog udjela, Bitstamp s 24 posto te

BTC-e s 14 posto. Svaka od tih burzi imala je vlastita pravila. Primjerice, BTC China je samo omogućavala trgovanje bitcoina s kineskim yuan-om, dok su preostale burze omogućavale trgovanje bitcoina isključivo s američkim dolarom. Tijekom sljedećih nekoliko mjeseci, pojavilo se još nekoliko burzi kriptovaluta. Dobar primjer je OKCoin, kineska bruza koja je izvjestan vremenski period bila najveća burza na tržištu. S druge strane, neke su ubrze potpuno nestale s tržišta, a među njima i Mt. Gox. Početkom 2014. godine, Mt. Gox opet su napali hakeri i ovoga puta osigurali propast burze. Procijenjuje se da je izgubljeno ili ukradeno bitcoina u vrijednosti od 350 milijuna američkih dolara, što je dovelo do konačnog pada burze. Nakon gašenja Mt. Goxa nastao je kaos na tržištu kriptovaluta te je cijena bitcoina drastično opadala. Međutim, tržište se pokazalo dovoljno snažnim i otpornim da stabilizira cijenu. Pojavile su se nove burze kriptovaluta koje su nadomjestile rupu nastalu nakon nestanka Mt. Goxa. Danas se bitcoin-om može trgovati na preko 100 različitih online burzi kriptovaluta od kojih većina omogućuje trgovanje i drugim kriptovalutama. [**7**]

3.4.3.2. Ekonomski utjecaj

Ekonomska analiza burzi kriptovaluta potpomaže razumijevanju rada kriptovaluta u okvirima generalne ekonomije kroz kvalitetu financijske infrastrukture i pozornosti koju ljudi posvećuju različitim kriptovalutama. Primjerice, na dobro stojećem tržištu cijene na burzama trebale bi odražavati sve informacije dostupne o svakoj od kriptovaluta. Teško je testirati efikasnost tržišta iz te perspektive. Međutim, bez obzira da li je tržište više ili manje efikasno, ono ne bi trebalo omogućavati arbitražne prilike (eng. *arbitrage opportunities*). Arbitraža je vrsta trgovanja koja investitoru osigurava instantan profit, bez da preuzima bilo kakav rizik. U dobro stojećim tržištima, arbitražne prilike trebale bi se pojaviti slučajno. U mjeri u kojoj se pojavljuju u praksi, uglavnom su prouzročene tržišnim fragmentacijama, odnosno specifičnim načinom trgovanja. Slučaj pojavljivanja arbitražnih prilika može biti i malo, neformirano trište. Arbitražnu priliku možemo reprezentirati uz pomoć kratkog primjera. Recimo da trgujemo na burzi kriptovaluta koja omogućuje kupnju bitcoina za 250 američkih dolara. S druge strane, ako želimo kupiti litecoin, to bi nas koštalo 2 dolara. Treća opcija je direktno trgovanje litecoina i bitcoina, bez upotrebe dolara. Za direktnu razmjenu, recimo da je tečaj 100 litecoina za jedan bitcoin. Vidljivo je da cijene nisu konzistentne te da postoji arbitražna prilika. Kako bi iskoristili tu priliku, možemo kupiti 100 litecoina za 200 dolara. Potom možemo kupiti jedan bitcoin s novo stečenih 100 litecoina. Na kraju, prodamo bitcoin za 250 dolara, pri čemu nam profit u odnosu na inicijalnu investiciju od 200 dolara iznosi 50 dolara. Ovakva vrsta arbitraže, ponzatija u stranim burzovnim investicijama, još se zove triangularna arbitraža (eng. *triangular arbitrage*) jer su potrebna tri različita tečaja da bi se izvršila. Kako bi se maksimizirao profit, trgovac bi kupio, u ovom slučaju, litecoina koliko je moguće, pretvorio ih u bitcoin te pretvorio sav bitcoin u dolare. Čim trgovci shvate arbitražnu priliku i krenu trgovati na ovaj način, pomiču cijene dok se ne namjeste na razinu koja onemogućuje arbitražnu priliku (u našem primjeru, dok litecoin ne postane skuplji ili bitcoin jeftiniji). Arbitražne prilike su u tradicionalnim burzama vrlo rijetke, ali se na tržištu kriptovaluta takve prilike pojavljuju učestalije. Ispostavlja se da su u dva posto slučajeva moguće arbitražne prilike koje imaju povrat investicije u iznosu od 1.4 posto. Iako povrat investicije djeluje malo, treba imati na umu da je ovo povrat investicije u izuzetno kratkom

vremenu koji gotovo da nema rizika. Arbitražne prilike najčešće se pojavljuju s kriptovalutama koje nisu baš poznate. Triangularne arbitražne prilike trebale bi se smanjiti povećanjem likvidnosti tržišta kriptovaluta. S druge strane, moguća je i arbitraža između različitih mjenjačnica. Situacija je ista, kupiti određenu kriptovalutu po manjoj cijeni na jednoj burzi, prebaciti kriptovalutu na drugu burzu te je tamo prodati i zaraditi na razlici. Pritisak kupnje i prodaje u ovakvim situacijama stabilizirao bi i ujednačio cijene. [**7**]

Postojanje burzi kriptovaluta također je važno i za konkurenciju među različitim kriptovalutama. Cijene po kojima se trguje na burzama mogu se interpretirati kao tržišna procjena njihove relativne važnosti. Cijena kriptovalute ovisi o procjeni široke mase ljudi koja u nju ulažu jer smatraju da bi mogla biti primjenjiva ne samo na tržištu kriptovaluta, već tradicionalnim segmentima tehnologije. Rast cijena svih kriptovaluta može upućivati i na opće prihvaćanje kriptovaluta te povećano povjerenje da će barem jedna od njih biti šire prihvaćena u ekonomiji. Dodatna potražnja za kriptovalutama može biti vođena i spekulacijama, odnosno nadi otkrivanja "sljedećeg bitcoina" koji će doživjeti nagao rast. [**7**]

3.4.4. Korisni izvori informacija

coinmarketcap.com ili cryptocompare.com Na njima možemo pronaći sve informacije o tržišnoj kapitalizaciji kriptovaluta. Možemo koristiti različite filtere za prikaz, primjerice top 100 kriptovaluta ili sve, prikaz po coinovima ili tokenima. Kriptovalute se dijele na coinove i tokene. Coin je kriptovaluta s vlastitim blockchainom, dok je token kriptovaluta koja koristi blockchain nekog coina. Takvu funckionalnost, primjerice, pruža Ethereum. Možemo ući u svaku kriptovalutu te iz lijevog preglednika vidjeti osnovni pregled kriptovalute - cijena, graf, vijesti i novosti, službene obavijesti. Moguće je vidjeti na kojim se sve burzama može kupiti odabrana kriptovaluta te sortirati najaktivnije burze po najvećem volumenu tj. protoku novca. Moguće je odabrati detaljnu burzu te vidjeti sve kriptovalute koje se nalaze na njoj. Na naslovnoj stranici vidljiv je sažetak svega.

bitcointalk.org Forum na kojemu se mogu pronaći informacije iz zajednice. Kako nešto napraviti, tražiti pomoć ili riješenje problema koji su drugi već imali i pisali o njima, informacije o svim starim i novim kriptovalutama te sve ostalo. Ova je stranica biblija što se tiče izvora informacija. Ako se želite pratiti novosti što se događa s kriptovalutama te što ljudi govore, pravo je mjesto za to. Loša je strana je to što postoji ogromna količina informacija, pa treba znati što tražiti. S druge strane, vrlo je kvalitetan izvor prijevremenih informacija.

coindesk.com Stranica koja prenosi vijesti iz svijeta kriptovaluta. Sličnih stranica ima mnogo, ali ovu sam odabrao posebno zbog brzine i relevantnosti podataka koje prenose. Preporuka je ako ne želite pratiti forume, ali želite pratiti vijesti iz svijeta kriptovaluta.

blockfolio.com ili delta.app Omogućuju praćenje našeg portfelja i cijena kriptovaluta na tržištu. Portfelj ili portfolio predstavlja skup financijske imovine pojedinca sastavljen od različitih financijskih instrumenata, ili u našem kontekstu, kriptovaluta. Unutar aplikacije možemo jednostavno dodavati kriptovalute koje želimo pratiti, postavljati alarme kada cijena dođe do određene razine, pregledavati grafove i transakcije ukoliko ih ručno upisujemo.

tradingview.com Alat za analizu tržišta. U njemu možemo detaljnije pregledati graf određene kriptovalute, crtati po grafu, primjenjivati indikatore i ostale napredne funkcije.

4. Analiza tržišta kriptovaluta

Do sada smo obradili povijest kriptovaluta, objasnili osnovne tehničke koncepte koji predstavljaju temelj njihova djelovanja, pojasnili ekonomski aspekt te prednosti i nedostatke istih, alternativne kriptovalute te ekosustav koji se razvio oko istih. Svi do sada obrađeni dijelovi kripto svijeta, ali i mnogi drugi koji nisu spomenuti u ovom radu, čine tržište kriptovaluta. Tržište kriptovaluta ima ista obilježja kao i tržište dionica, no u ovom slučaju dionice predstavljaju cijene kriptovaluta. Cijene kriptovaluta često su odraz njihovog tehnološkog napretka, učinkovitosti ili slučaja korištenja koji rješavaju svojim sustavom. Postoje različiti načini na koje se može provesti analiza, ovisno o svrsi analize. Za trgovanje na financijskim tržištima, najvažnija je tehnička analiza. Tehnička analiza korištenjem različitih grafova i indikatora proučava kretanje cijena dionica u prošlosti s ciljem predviđanja trenda kretanja cijena u budućnosti. Utemeljitelj tehničke analize bio je Charles Dow čija je tvrtka Dow Jones prva počela sustavno pratiti kretanje cijena. Dow je krajem 19. stoljeća utvrdio osnovne postavke na kojima se temelji današnja moderna tehnička analiza. Osnove tehničke analize leže u očekivanju svih budućih događaja tržišta koji utječu na sadašnju cijenu neke robe. Svaki element koji može imati utjecaj (makroekonomsko okruženje, politički razlozi, očekivanja investitora, promjene u poslovanju) direktno se odražava na cijenu. Gornja pretpostavka često se naziva i pretpostavka efikasnog i transparentnog tržišta. Ona govori da je svaka relevantna informacija jednako i u isto vrijeme dostupna svima te će ju velik broj sudionika na tržištu podjednako interpretirati. Stoga se pretpostavlja da će investicijske odluke velikog broja 'racionalnih' investitora u suštini biti slične. Nadalje, tehnička analiza pretpostavlja da kretanje cijena (vrijednosnica, indeksa, valuta) nije slučajno nego da slijedi određene zakonitosti koje se ponavljaju u ciklusima (cikličko kretanje ekonomije). Proučavanjem takvih zakonitosti u prošlosti, moguće je približno predvidjeti njihovo ponavljanje u budućnosti. Svrha korištenja grafova i različitih indikatora je u olakšavanju određivanja trenda i karakteristika kretanja cijene u prošlosti. Fokusirajući se samo na analizu trenutne tržišne cijene i njenog kretanja u prošlosti, tehnička analiza predstavlja direktan pristup analizi tržišta. Druga vrsta analize jest analiza generalnih trendova i uočavanje generalnih obrazaca. U tu svrhu analizirat će se, u vrijeme pisanja rada, najpopularnije uređene trojke kriptovaluta, valuta i tržišta kriptovaluta. [**14**, **15**]

Promatrane kriptovalute s pripadajućim ID-jevima:

- Bitcoin - BTC (1182)
- Ethereum - ETC (7605)
- Eos.io - EOS (166503)
- Bitcoin Cash - BCH (202330)
- Litecoin - LTC (3808)
- Ethereum Classic - ETC (5324)
- Digital Cash - DASH (3807)

- Ripple - XRP (5031)
- Cardano - ADA (321992)
- Iota - IOTA (127356)

Promatrane valute:

- Euro - EUR
- Dolar - USD
- Bitcoin - BTC

Promatrane burze kriptovaluta:

- Bitfinex
- WavesDex
- Gemini
- Coinbase
- Bitstamp
- Bittrex

4.1. Tehnička analiza na primjeru bitcoina

Za primjer, možemo provesti jednu kratku tehničku analizu bitcoina. Naime, svaki se puta ponavlja ista priča oko korekcije i pada Bitcoina. Ljudi i mediji uporno pričaju i pišu kako je propao, kako nema nikakvu svrhu niti vrijednost i slično. Sada ćemo analizirati koliko je puta do sada Bitcoin bio mrtav i koliko mu je vremena trebalo da prođe cijeli ciklus. Gledat ćemo graf na burzi Bitstamp, koja ima jednu od najduljih povijesti trgovanja Bitcoinom, a to je od 18.8.2011 godine.

Slika 11: Prvi val bitcoina

Slika 12: Korekcija prvog vala bitcoina

Prvi veliki val koji je bitcoin napravio, započeo je u siječnju 2013. godine, trajao je 90 dana i napravio porast od oko 1700 posto, gdje je skočio sa 13 dolara na 260 dolara. Nakon toga je započela korekcija 10.4.2013. i ukupno je trajala oko 187 dana. U tom se periodu cijena stabilizirala u padu do 0.786 fibonacci linije na 65 dolara.

Slika 13: Drugi val bitcoina

Slika 14: Korekcija drugog vala bitcoina

92 dana nakon, cijena probija iznad linije otpora i oko 70 dana nakon toga probija i Ichimoku oblak koji je jednom testirao za potvrdu potpore. Cijena je probila i zadnju razinu otpora na 128 dolara, 13.10.2013., nakon čega je Bitcoin bio slobodan da krene u novi val. Drugi val je krenuo nakon, te u samo 50 dana napravio porast za oko 800 posto, do cijene do oko 1155 dolara. Koliko je god cijena brzo narasla, toliko je višestruko sporije padala. ABC korekcija ovog vala trajala je oko 450 dana, dotaknuvši dno od 156 dolara, što predstavlja skoro 100 posto povrat od razine od koje je krenula. Nakon toga cijena se ustabilila te tako završila tržišni ciklus. Probijanjem otpora 25.2.2015. započeo je period mirovanja na dnu od oko 234 dana kada je cijena konačno probila prvu razinu otpora i krenula u novi ciklus. Sve ovo dovodi do zaključka, da nakon jake euforije i rasta cijene od 2 mjeseca, slijedio je pad koji je trajao čak dvije godine. I Bitcoin je u tom periodu bio mrtav i završio svoj tržišni ciklus. Dvije godine je bilo dovoljno vremena za zaključiti kako je ovo bio samo balon koji je puknuo. U tom periodu nitko nije pričao o njemu, niti obraćao pažnju da postoji. Treći val je krenuo nakon.

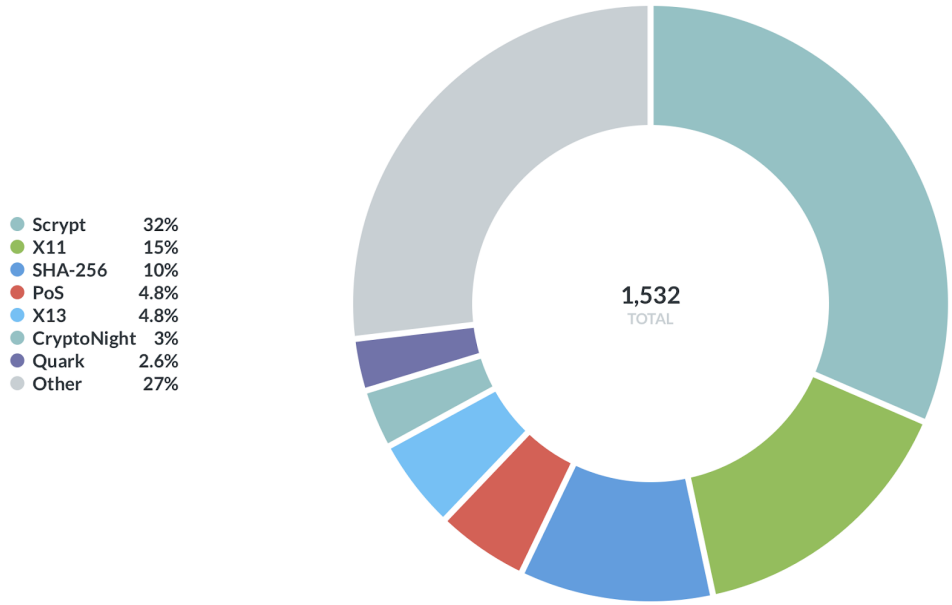
Slika 15: Treći val bitcoina

Oko 1.11.2015. probijanjem otpora i stvaranjem nove tržišne strukture započeo je treći val. Cijena je započela sporije rasti u 5 Elliot valova i u periodu od oko 500 dana napravila

porast od oko 400 posto, do ukupno 1280 dolara. Možemo vidjeti da je taj rast trajao dosta dugo, ali sporo i progresivno. Svi su očekivali korekciju nakon toga, ali se tih 5 valova pretvorilo samo u uvod u ono što slijedi. Idući skok koji je napravio, nije bio predvidiv, ali je bilo jasnije nakon naglog napuštanja vrha vala 5, da je veći tržišni ciklus u igri. Stvorio se novi tržišni ciklus i novih 5 Elliot valova, koji su nakon napuštanja stare rekordne cijene od 1280 dolara, u idućih 240 dana došli do cijene od oko 20000 dolara, što je porast za oko 1300 posto. Nakon tipičnog eksponencijalnog petog vala, slijedila je korekcija i Bitcoin je opet mrtav. Od početka 2013. godine do kraja 2018. godine, Bitcoin je bio mrtav tri puta i preživio tri tržišna ciklusa. Vrijedi se zapitati, dali se povijest ponavlja? Na što je Bitcoin odgovorio u promatranom periodu od 6 godina. Može li se tehničkom analizom vidjeti budućnost i kada nešto savršeno kupiti i prodati? Odgovor na to pitanje je ne. Nije moguće vidjeti točan vrh koliko će cijena porasti, niti točno dno do koje će pasti. Nemamo kristalnu kuglu u koju gledamo i vidimo budućnost. No ono što nam tehnička analiza pruža jest investicijska pismenost. Kada učimo kako čitati graf i provesti kvalitetnu analizu, znamo koje su najbolje zone kada nešto kupiti ili prodati. Tehnička analiza je poput jezika, koju jednom kada naučimo, znamo zauvijek. Dođemo do razine kada pogledamo graf, odmah znamo što se događa u određenoj fazi tržišnog ciklusa. Velik faktor uz sve ovo čini i emocionalno razumijevanje ciklusa i upravljane vlastitim emocijama. [**45**]

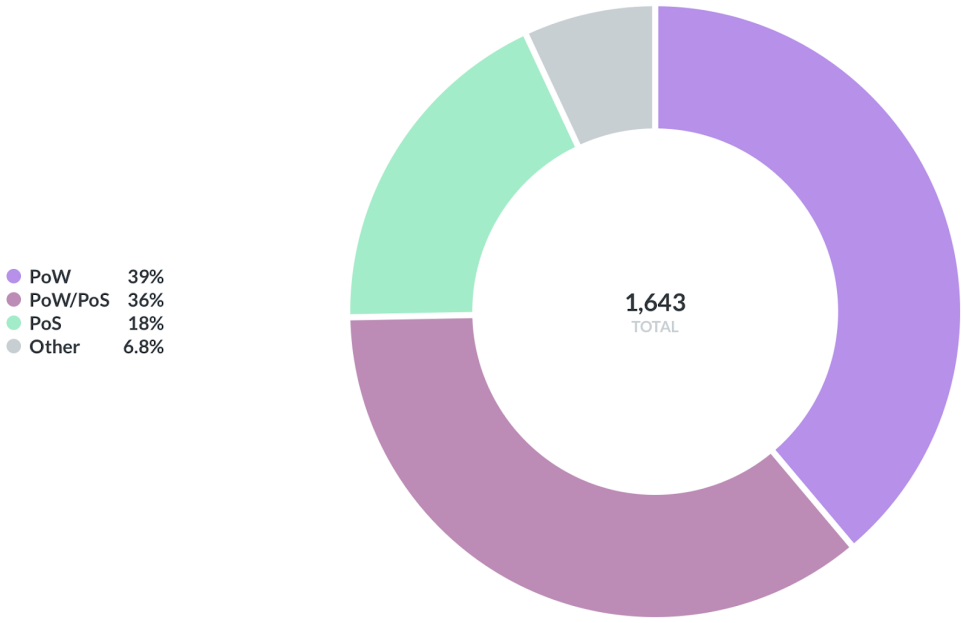
4.2. Analiza kriptovaluta

Među prikupljenim detaljnim podacima kriptovaluta, analizirana su svojstva od ukupno 4199 različitih kriptovaluta. U dostupnim podacima nije postojala niti jedna kriptovaluta čije je broj jedinice valute izrudarenih u startu poznat. Nadalje, niti jedna kriptovaluta još uvijek nije sponzorirana niti prihvaćena od javne institucije. S druge strane, postoji mogućnost da podaci za prethodno navedene zaključke nisu ažurni na API-ju jer nam je iz teoretskog dijela rada poznato da je Ripple početkom rada pustio u opticaj sve jadinice kriptovalute. U nastavku ćemo usporediti svojstva svih kriptovaluta.



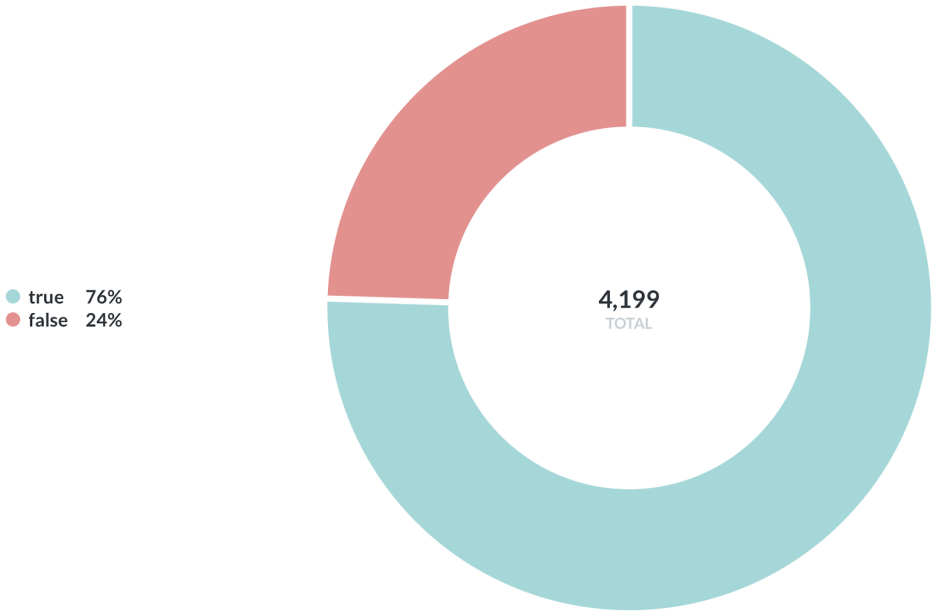
Slika 16: Omjer algoritama hashiranja

Vizualizacija predstavlja omjer korištenih algoritama hashiranja među promatranim kriptovalutama. Od promatranih 4199 kriptovaluta, podaci korištenog algoritma nisu bili dostupni za 2667 kriptovaluta, što čini 64% od ukupnog broja promatranih kriptovaluta, te su izuzeti iz grafičkog prikaza. Od preostalih 1532 kriptovalute, najviše ih koristi Scrypt, čak 489 (32%) kriptovaluta. Slijede X11 sa 233 (15%) i SHA-256 sa 160 (10%) kriptovaluta. S nešto manjom zastupljenošću, istakli su se PoS, X13, CryptoNight the Quark. Iz preostalih 27%, vidljivo je da postoji izuzetno velika distribucija različitih algoritama hashiranja specifičnih za mali broj kriptovaluta.



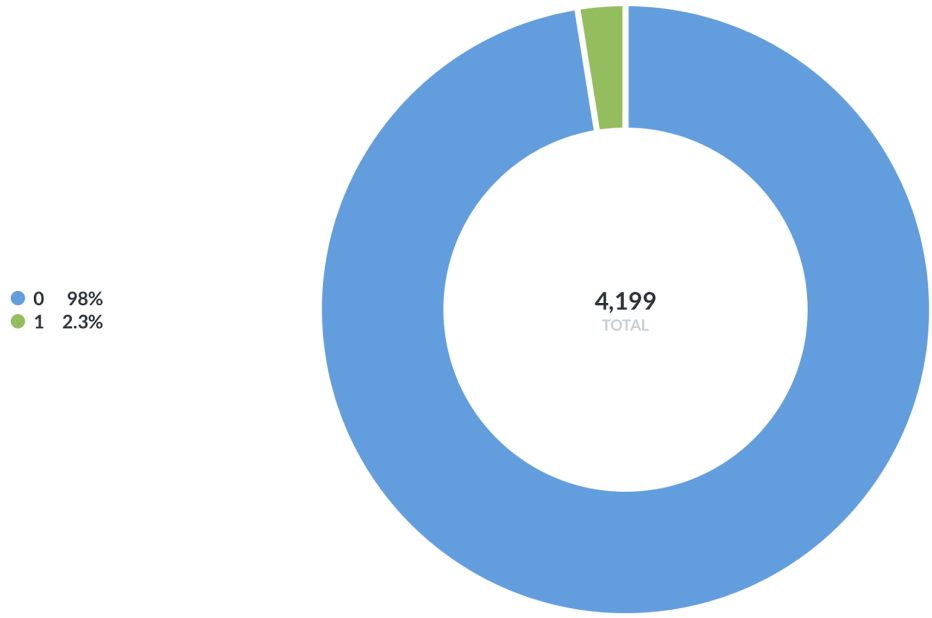
Slika 17: Omjer procesa rudarenja

Vizualizacija predstavlja omjer različitih procesa rudarenja, odnosno procesa dostizanja koncenzusa među promatranim kriptovalutama. Od 4199 kriptovaluta, podaci korištenog procesa rudarenja nisu bili dostupni za 2556 (61%) kriptovaluta te su izuzeti iz grafičkog prikaza. Među preostalih 1643 kriptovalute, 641 (39%) ih koristi PoW (Proof-of-Work), 591 (36%) koristi hibrid PoW i PoS koncenzusa te 300 (18%) kriptovaluta koristi isključivo PoS (Proof-of-Stake) koncenzus. Među preostalih 6.8% kriptovaluta, svaka ima svoju implementaciju algoritma dostizanja koncenzusa među kojima je DPoS (Delegated Proof-of-Stake) najpopularniji.



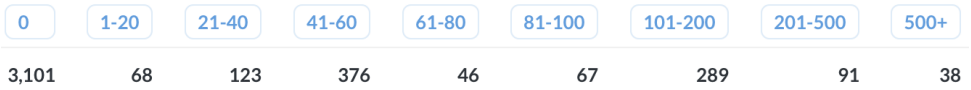
Slika 18: Omjer aktivnih valuta na tržištu kriptovaluta

Vizualizacija predstavlja broj kriptovaluta trenutno aktivnih na tržištu kriptovaluta. Od promatranih 4199 kriptovaluta, sa njih 3176 (76%) se aktivno trguje na barem jednoj burzi kriptovaluta, dok ih 1023 (24%) nije aktivno niti na jednoj burzi kriptovaluta.



Slika 19: Omjer izrudarenih valuta

Vizualizacija predstavlja omjer kriptovaluta koje su u opticaj puštene potpuno izrudarene. Od promatranih 4199 kriptovaluta, ukupno 4104 (98%) kriptovaluta svoje jedinice valute u opticaj izdaje procesom rudarenja. Samo 98 (2.3%) kriptovaluta slijedi princip Ripple kriptovalute, gdje su sve jedinice valute puštene u opticaj zajedno s kriptovalutom pa se ne oslanjaju na proces rudarenja.



Slika 20: Vrijeme rudarenja novog bloka

Tablica predstavlja grupacije kriptovaluta prema prema prosječnom broju sekundi potrebnih za rudarenje novog bloka. Na uzorku od 4199 kriptovaluta, 3101 vrijeme potrebno za rudarenje novog bloka iznosi 0, što bi značilo da kriptovalute ne rade na principu rudarenja ili podaci za iste nisu dostupni. Za 68 kriptovaluta je manje od 20 sekundi potrebno za rudarenje novog bloka. Najveći broj kriptovaluta, njih 376, ima prosječno vrijeme između 41 i 60 sekundi. Između 101 i 200 također postoji poveća grupacija kriptovaluta, što vrijedi i za raspon između 21 i 40 sekundi. U pravilu, većina kriptovaluta rudari novih blok u vremenskom intervalu manjem od dvije minute.

Slika 21: Usporedba kretanja cijena kriptovaluta

Graf prikazuje prosječno kretanje cijena, izraženih u dolarima, između svih burzi po pojedinoj kriptovaluti. Bitcoin evidentno predvodi tržište kao prva kriptovaluta. Drugi je Bitcoin Cash, koji nije uključen u vizualizaciju zbog netočnosti podataka s API-ja. Na trećem je mjestu Ethereum, iako ga je za vrijeme naglog rasta krajem 2018. godine na kratko prestigao DASH. Ostale kriptovalute zauzimaju manji dio tržišta.

4.3. Analiza burzi kriptovaluta

exchange	open	high	close	low
bitfinex	3,955.95	4,143.69	3,989.86	3,760.49
bitstamp	3,963.96	4,142.25	3,998.11	3,777.01
bittrex	3,949.55	4,136.32	3,983.11	3,741.77
coinbase	3,996.63	4,179.75	4,030.81	3,802.48
gemini	3,984.79	4,162.39	4,018.57	3,804.81
wavesdex	6,966.23	7,355.38	7,030	6,543.07

Slika 22: Prosječne OHLC cijene burzi kriptovaluta 2017. godine

Tablica prikazuje prosječne dnevne OHLC cijene za Bitcoin izražen u dolaru po burzi kriptovaluta za cijelu 2017. Godinu. Najviše cijene u svim slučajevima ima burza Wavesdex, koju slijedi Bitfinex. Preostale četiri burze kriptovaluta relativno su ujednačene s varijacijama od nekoliko dolara. Za Wavesdex nisu dostupni podaci za cijelu godinu jer je burza redizajnom promijenila i ime, stoga su prosječne vrijednosti znatno veće od ostatka kriptovaluta.

exchange	open	high	close	low
bitfinex	7,567.59	7,793.84	7,540.79	7,277.18
bitstamp	7,539.6	7,759.06	7,512.12	7,251.35
bittrex	7,440.63	7,631.41	7,413.57	7,174.35
coinbase	7,537.81	7,749.96	7,510.38	7,253.3
gemini	7,538.14	7,752.69	7,510.6	7,252.21
wavesdex	7,885.21	8,466.77	7,860.09	7,497.62

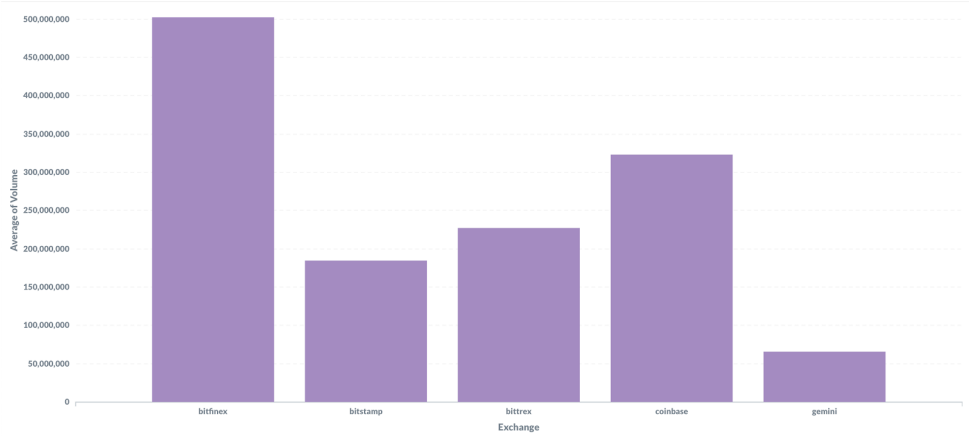
Slika 23: Prosječne OHLC cijene burzi kriptovaluta 2018. godine

Tablica prikazuje prosječne dnevne OHLC cijene za Bitcoin izražen u dolaru po burzi kriptovaluta za cijelu 2018. Godinu. Najviše cijene u svim slučajevima ima burza Wavesdex, koju slijedi Bitfinex. Preostale četiri burze kriptovaluta relativno su ujednačene s varijacijama od nekoliko dolara.

exchange	open	high	close	low
wavesdex	6,164.8	6,247.29	6,181.59	6,104.29
bittrex	7,033.96	7,239.92	7,058.64	6,829.03
bitfinex	7,088.72	7,294.96	7,113.24	6,886.46
bitstamp	7,033.39	7,243.82	7,058.37	6,823.22
gemini	7,034.85	7,242.18	7,059.83	6,828.38
coinbase	7,034.31	7,243.99	7,059.26	6,825.64

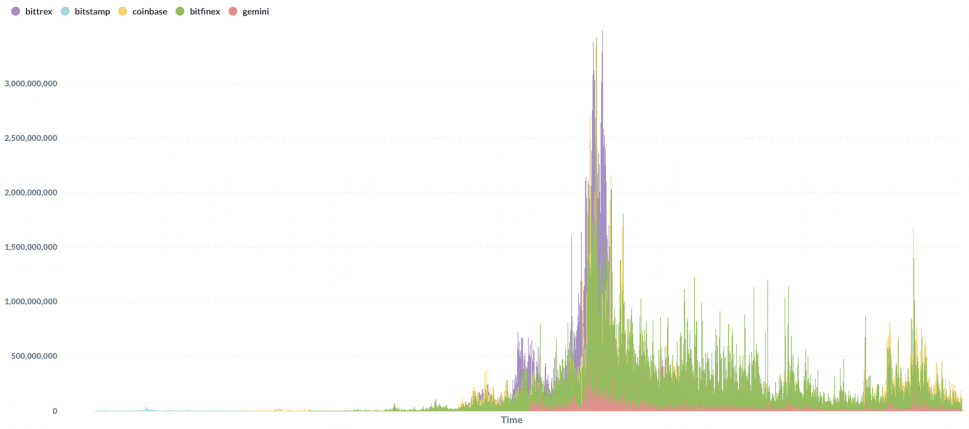
Slika 24: Prosječne OHLC cijene burzi kriptovaluta 2019. godine

Tablica prikazuje prosječne dnevne OHLC cijene za Bitcoin izražen u dolaru po burzi kriptovaluta od 1.1.2019. do 1.9.2019. godine. Najviše cijene u svim slučajevima ima burza Bitfinex, a najniže Wavesdex. Preostale četiri burze kriptovaluta relativno su ujednačene s varijacijama od nekoliko dolara.



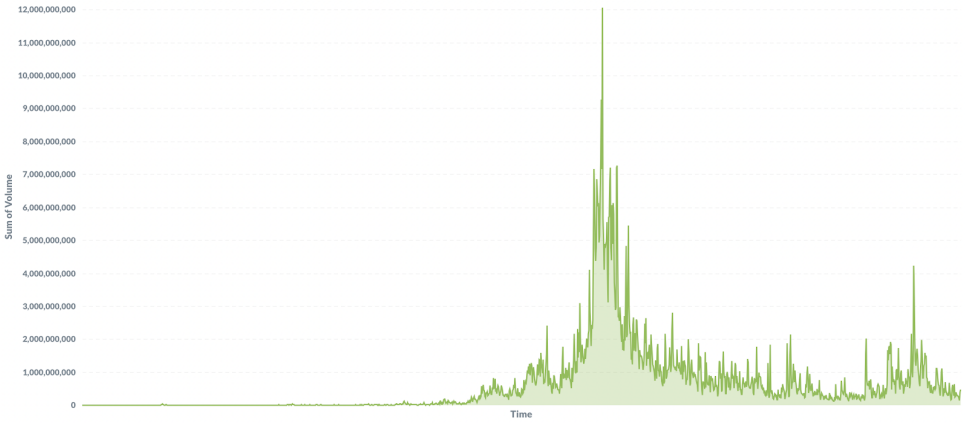
Slika 25: Prosječan dnevni volumen burzi kriptovaluta

Grafikon prikazuje prosječan dnevni volumen prema burzi kriptovaluta, od njihova nastanka do sadašnjeg trenutka. Burza Bitfinex ima evidentno najveći prosječni dnevni volumen koji iznosi 177,875,743.56 dolara. Slijedi Coinbase sa 113,185,211.5 te Bittrex sa 92,248,542.2.



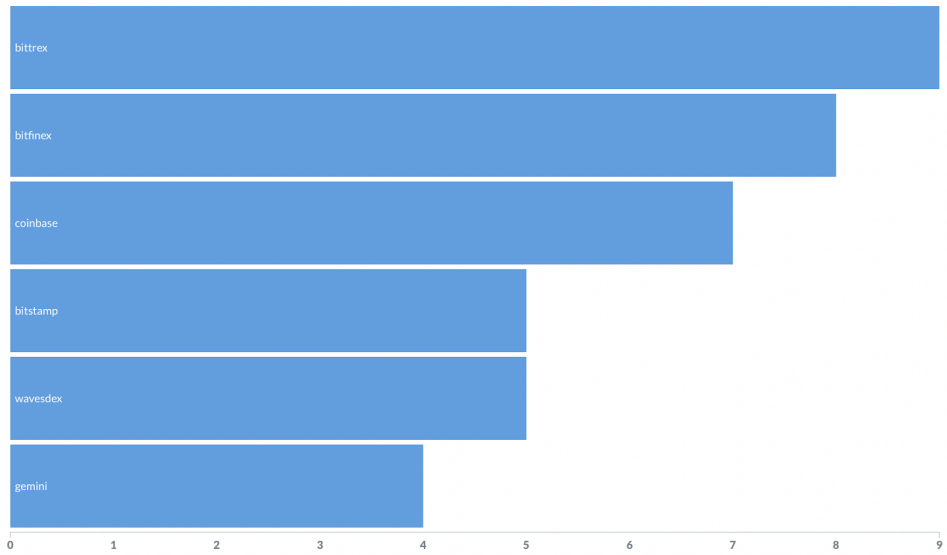
Slika 26: Trend kretanja volumena burzi kriptovaluta

Graf prikazuje prosječan dnevni volumen izražen u dolarima po burzi kriptovaluta kroz vrijeme do sadašnjeg trenutka. U ranim počecima tržišta kriptovaluta, najveći volumen imala je burza Bitstamp. Potom je lansiran Coinbase i preuzeo vodeće mjesto s obzirom na dnevni volumen trgovanja. Pojavom Bitfinex burze kriptovaluta, preuzima vodeću ulogu na tržištu kriptovaluta, uz povremene izmjene s Coinbase burzom. Upečatljivi su nagli porasti volumena trgovanja u vremenskim periodima kada su cijene drastično rasle, posebice onaj krajem 2018. godine u sredini grafikona. Tada je burza Bittrex burza imala najveći volumen.



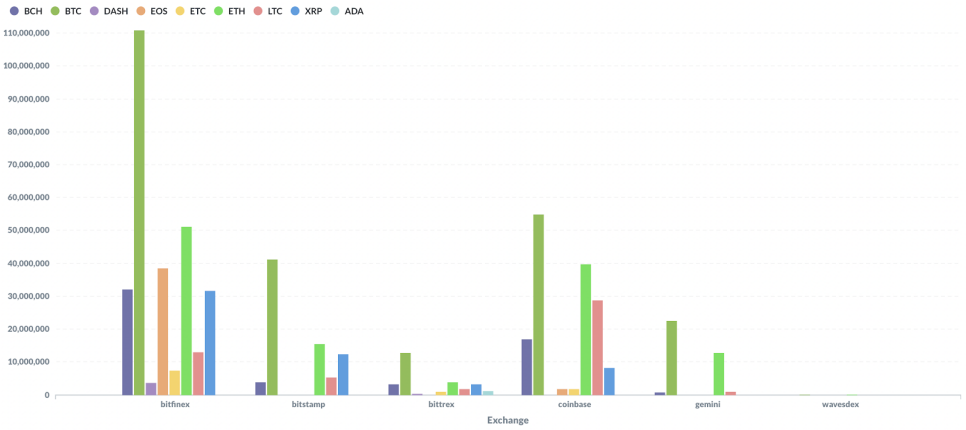
Slika 27: Trend ukupnog volumena burzi kriptovaluta

Graf prikazuje zbroj dnvevnih volumena izraženih u dolarima svih promatranih burzi kriptovaluta kroz vrijeme sve do sadašnjeg trenutka. Od ranih početaka kriptovaluta vidljivi su kontinuirani obrasci naglih porasta i padova volumena koji rezultiraju naglim rastovima i padovima cijena. Generalni najniži volumen ima uočljiv blagi trend rasta. Nakon najvećeg porasta volumena krajem 2018. godine, vidljivi su učestaliji obrasci kratkotrajnih visokih volumena koji se potom vraćaju na median blagog rasta najnižih volumena.



Slika 28: Broj podržanih kriptovaluta po burzi kriptovaluta

Graf prikazuje broj podržanih promatranih kriptovaluta po burzi. Od promatranih top 10 kriptovaluta na kojima se temelji analiza, Bittrex podržava čak 9 kriptovaluta, Bitfinex 8, a slijedi ga Coinbase sa 7. Bitstamp i Wavesdex podržavaju svega 5 kritpovaluta, a Gemini 4.

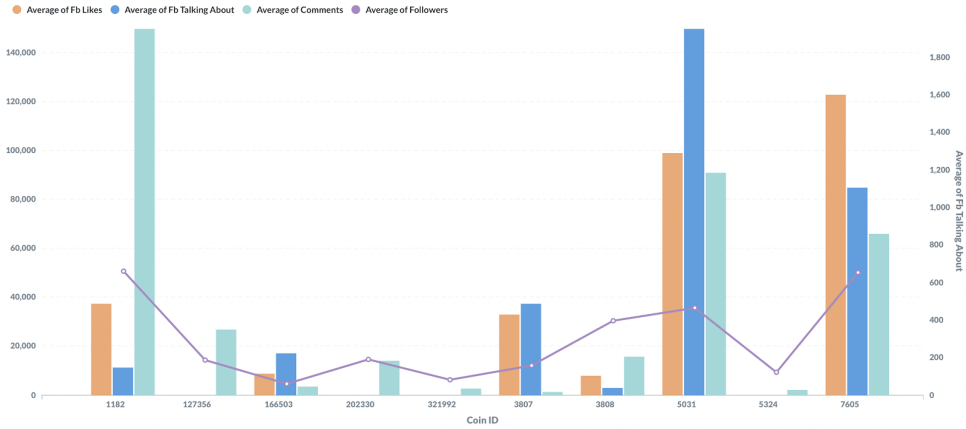


Slika 29: Prosječan dnevni volumen kriptovaluta po burzi kriptovaluta

Graf prikazuje prosječan dnevni volumen pojedinih kriptovaluta izražen u dolarima po svakoj burzi kriptovaluta. Evidentno je da Bitcoin ima najveći volumen na svakoj burzi kriptovaluta, a uvijek ga slijedi Ethereum. Na trećem mjestu je Eos.io, a iza njega se, ovisno o burzi, izmjenjuju Litecoin, Ripple i Bitcoin Cash. Preostale kriptovaluta nemaju značajan volumen o odnosu na navedene.

4.4. Analiza društvenih podataka

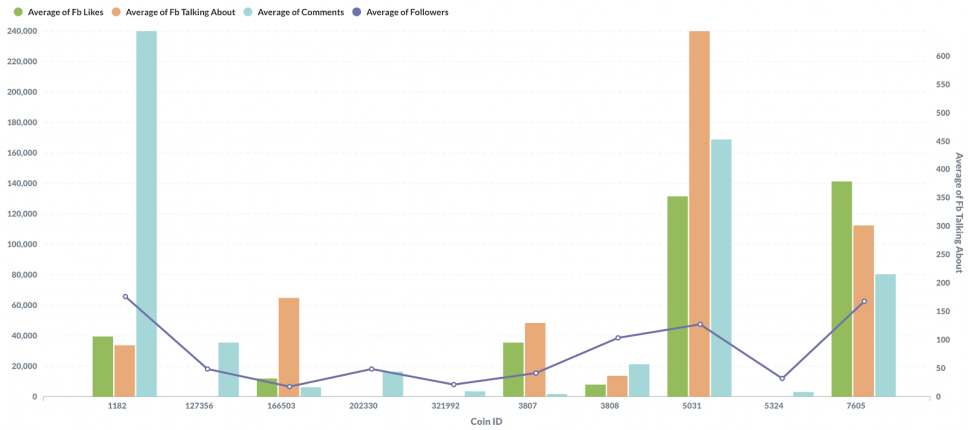
4.4.1. Facebook



Slika 30: Prosječan broj sviđanja, komentara, diskusija i pratitelja u danu

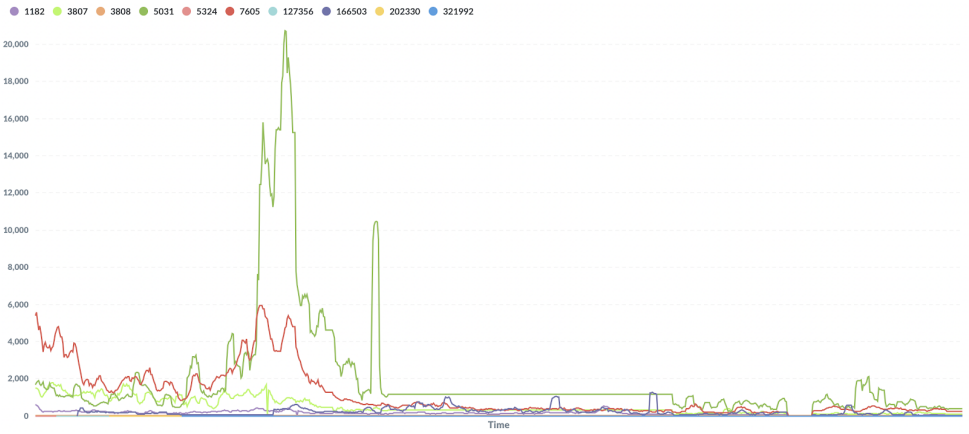
Graf predstavlja prosječan broj sviđanja, komentara, diskusija i pratitelja na društvenoj platformi Facebook u vremenskom intervalu od jednog dana. Desna skala vrijednosti odnosi se na broj prosječnih diskusija, dok se lijeva odnosi na sve ostale promatrane varijable. Najveći prosječni broj sviđanja ima Ethereum s 122,899, a blizu mu je i Ripple s 99,086. Slijede Bitcoin i Dash s podjednakim brojem prosječnih sviđanja. S druge strane, najveći broj diskusija vodi se na temu Ripple-a s 1951 te zatim Ethereum-a s 1105. Nešto manje imaju Eos.io i Dash, dok je Bitcoin na petom mjestu slabo zastupljen. Najviši prosjek komentara na bazi jednog dana

na Facebook stranici ima Bitcoin s 149,791, a slijede ga Ripple i Ethereum. Najveći prosječan broj pratitelja Facebook stranice također ima Bitcoin s 65709 pratitelja, a slijedi ga Ethereum s 62353 te Ripple s 47597.



Slika 31: Prosječan broj sviđanja, komentara, diskusija i pratitelja u satu

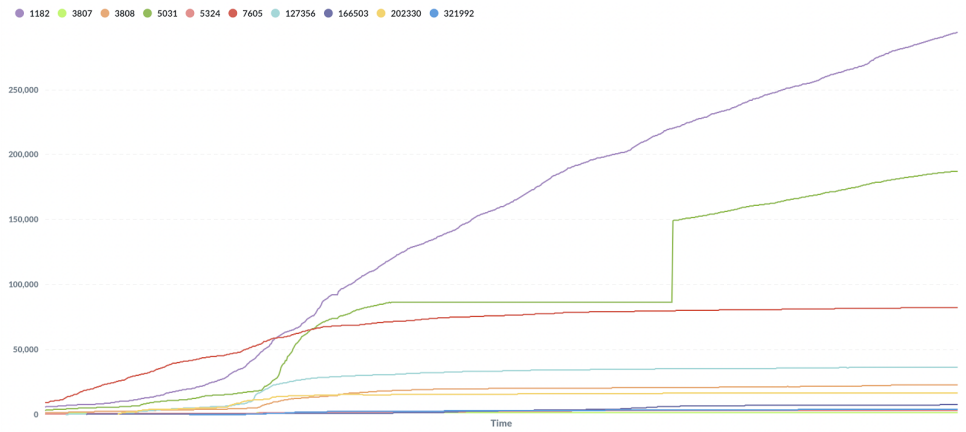
Graf predstavlja prosječan broj sviđanja, komentara, diskusija i pratitelja na društvenoj platformi Facebook u vremenskom intervalu od jednog sata. Desna skala vrijednosti odnosi se na broj prosječnih diskusija, dok se lijeva odnosi na sve ostale promatrane varijable. Najveći prosječni broj sviđanja ima Ethereum s 141,176, a blizu mu je i Ripple s 131,490. Slijede Bitcoin i Dash s podjednakim brojem prosječnih sviđanja. S druge strane, najveći broj diskusija vodi se na temu Ripple-a s 644 te zatim Ethereum-a s 301. Nešto manje imaju Eos.io i Dash, dok je Bitcoin na petom mjestu slabo zastupljen. Najviši prosjek komentara na bazi jednog sata na Facebook stranici ima Bitcoin s 240,148, a slijede ga Ripple s 169082 i Ethereum s 80535. Najveći broj pratitelja Facebook stranice također ima Bitcoin s 65709 pratitelja, a slijedi ga Ethereum s 62353 te Ripple s 47597.



Slika 32: Usporedba broja Facebook diskusija

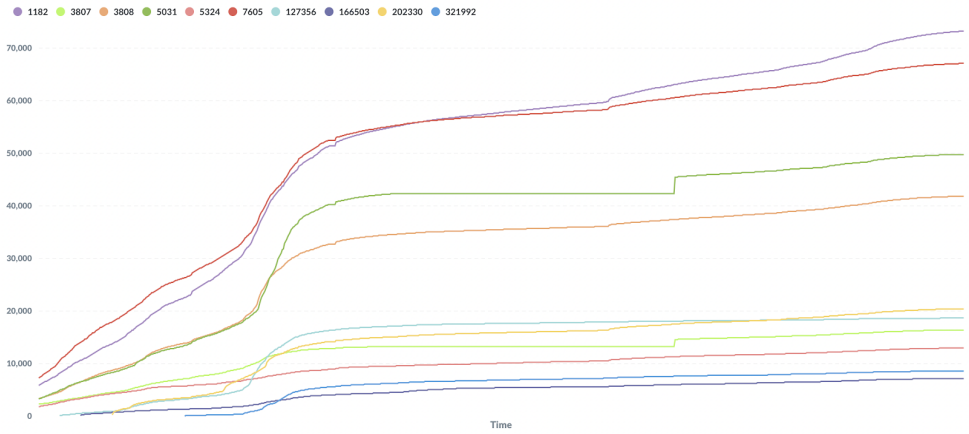
Graf prikazuje broj diskusija o pojedinim kriptovalutama na društvenoj platformi Facebook kroz promatrani vremenski period. Na Početku je najveći broj diskusija imao Ethereum, no ostao je dovoljno zastupljen kroz promatrani vremenski period da se većinski nađe na drugom

mjestu. Najveći broj diskusija ima Ripple, posebice za vrijeme naglog rasta svih kritpovaluta krajem 2018. godine.



Slika 33: Usporedba broja Facebook komentara

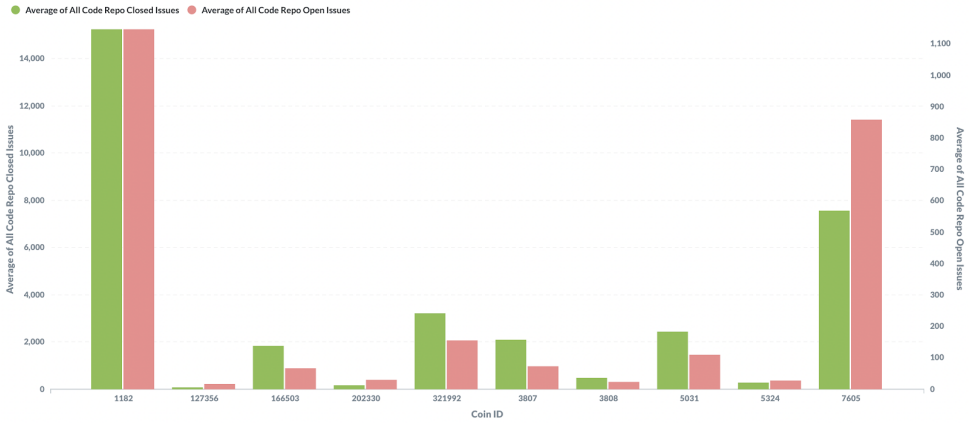
Graf prikazuje broj komentara na društvenoj platformi Facebook u promatranom vremenskom periodu. Broj komentara na Facebooku za Bitcoin je u konstantnom porastu te nastavlja trend rasta. Ripple je doživio nagli porast i nastavio trend rasta, dok je Ethereum krenuo s navišim brojem Facebook komentara, nakon čega je rast postepeno usporen.



Slika 34: Usporedba broja Facebook pratitelja

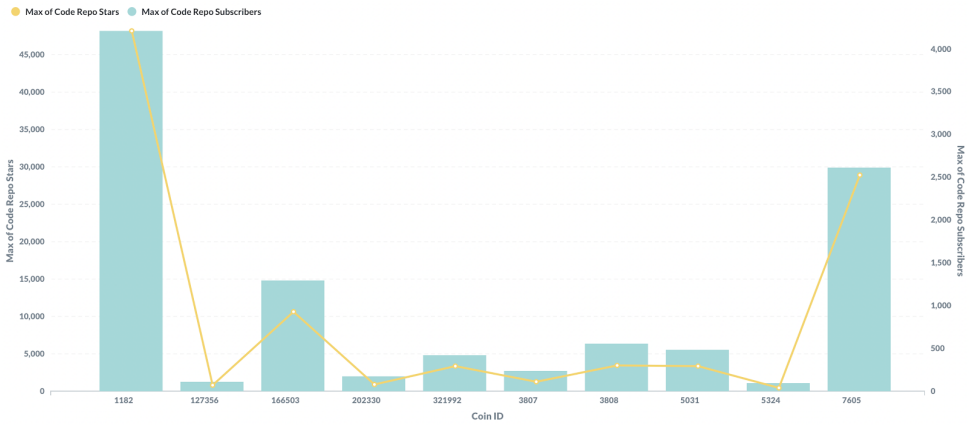
Graf prikazuje broj pratitelja na društvenoj platformi Facebook u promatranom vremenskom periodu. Bitcoin i Ethereum imaju najveći broj pratitelja koji je u stalnom porastu. Slijede Ripple koji slijedi trend rasta te Ethereum Classic paralelno s Rippleom.

4.4.2. GitHub



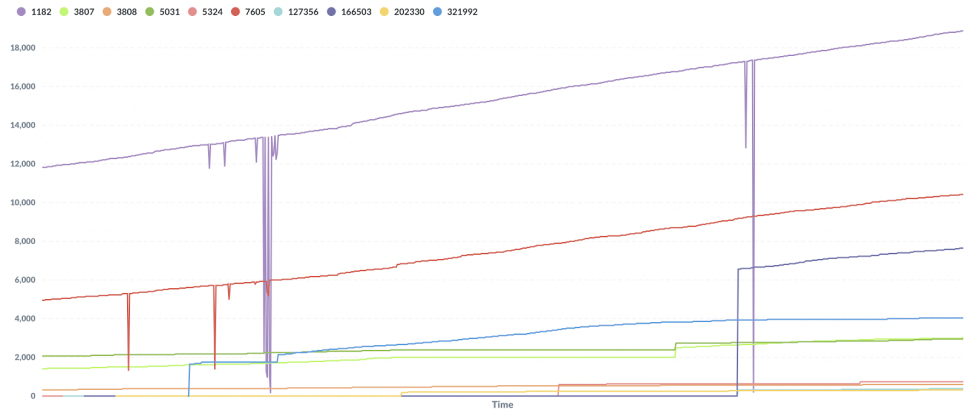
Slika 35: Usporedba broja otvorenih i riješenih problema

Graf predstavlja agregirani prikaz otvorenih i riješenih problema na Github repozitoriju. Lijeva skala predstavlja broj riješenih problema, a desna broj otvorenih problema. Polazišna točka za usporedbu je Bitcoin s obzirom da ima najveći broj otvorenih i riješenih problema. Sve kriptovalute rješavaju više problema nego ih se otvori. U odnosu na Bitcoin, Ethereum ima sporiji trend rješavanja otvorenih problema s obzirom na trend njihova pojavljivanja. Preostale kriptovalute ili rješavaju otvorene probleme dovoljnom brzinom ili se suočavaju s manjim brojem otvorenih problema, zbog čega imaju pozitivan trend u odnosu na Bitcoin.



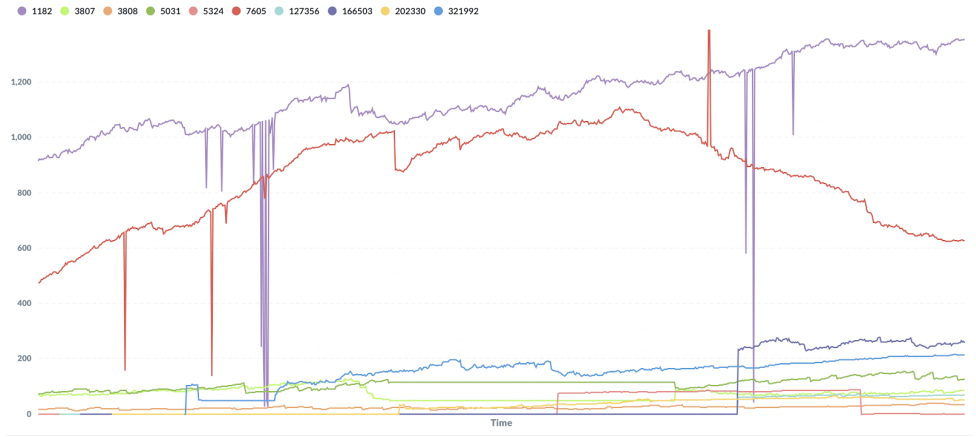
Slika 36: Broj pretplatnika i favorita Github repozitorija

Graf predstavlja trenutni broj pretplatnika i favorita Github repozitorija za svaku kritpovalutu. Lijeva skalda predstavlja broj favorita, a desna broj pretplatnika. Najveći broj pretplatnika i favorita ima Bitcoin te je zbog toga i polazište usporedbe. Slijedi ga Ethereum s proporcionalnim omjerom favorita i pretplatnika. Eos.io je sljedeći popularan projekt, ali s nešto manje favorita od očekivanog broja. Ostale kriptovalute nisu privukle preveliku pozornost.



Slika 37: Trend rješavanja problema na Github repozitoriju

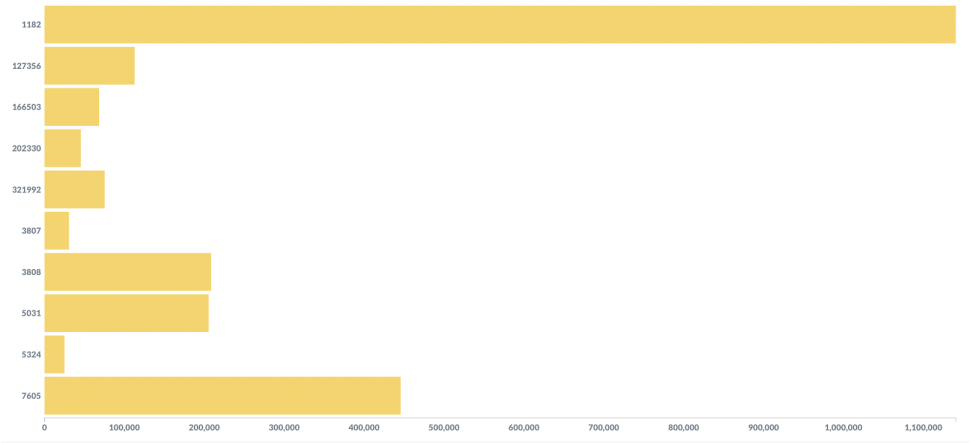
Graf predstavlja prosječan trend rješavanja problema na Github repozitoriju u promatranom vremenskom periodu. Bitcoin i Ethereum imaju stabilan pozitivan trend, a pridružio im se i Eos.io. S druge strane, Cardano prelazi u stanje stagnacije, kao većina preostalih kriptovaluta.



Slika 38: Trend otvaranja problema na Github repozitoriju

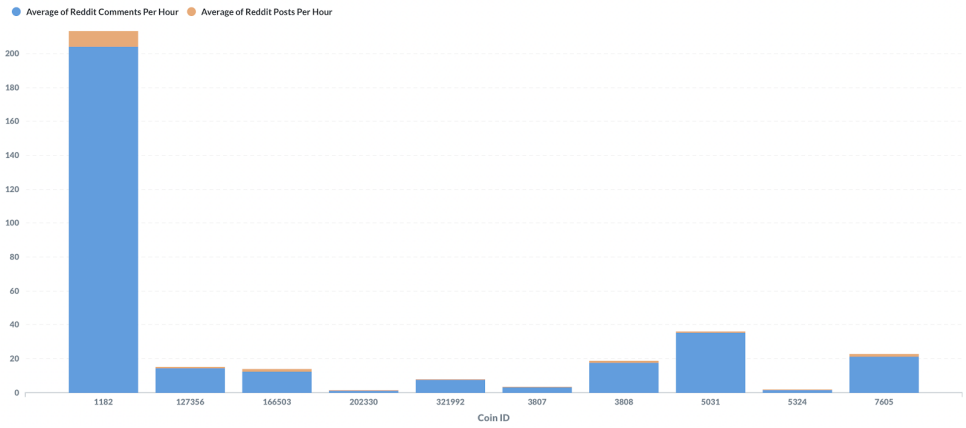
Graf predstavlja prosječan trend otvaranja problema na Github repozitoriju u promatranom vremenskom periodu. Broj otvaranih problema kod Bitcoina i dalje ima trend rasta, kao i većina preostalih kriptovaluta. S druge strane, Ethereum je u nagloj krivulji opadanja, što bi značilo su na dobrom putu ka stabilnom rješenju.

4.4.3. Reddit



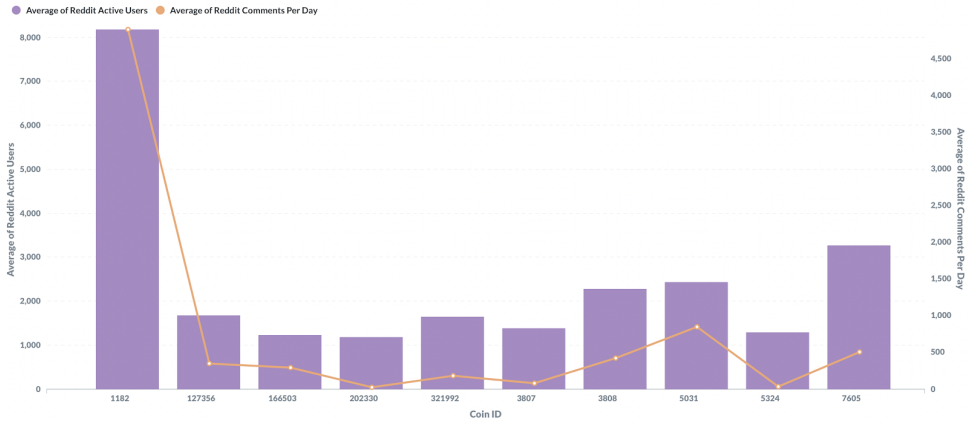
Slika 39: Broj pretplatnika Reddit stranice

Graf predstavlja ukupan broj pretplatnika Reddit stranice kriptovalute. Najveći broj pretplatnika ima Bitcoin s 1,140,182, a slijedi ga Ethereum s 445,705. Iza njih su Litecoin s 208,393 te Ripple s 205,427.



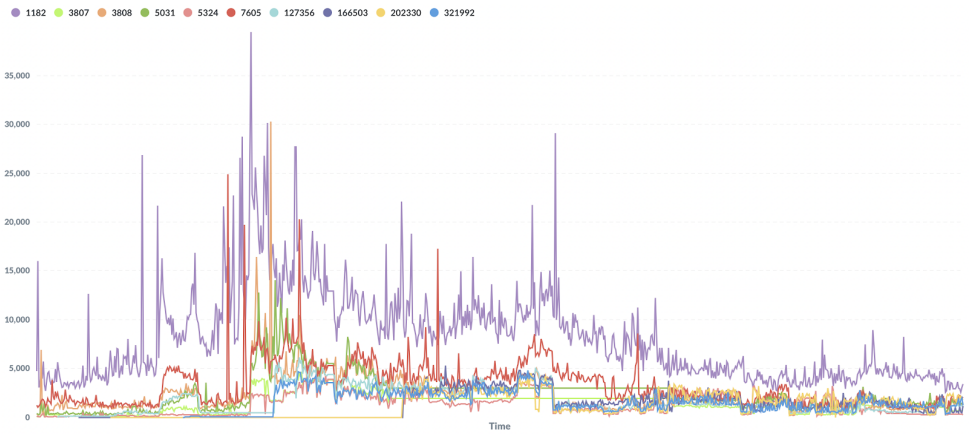
Slika 40: Prosječan broj objava i komentara Reddit stranice

Graf predstavlja prosječan broj objava i komentara na Reddit stranici svake kriptovalute u vremenskom razdoblju od jednog sata. Bitcoin ima najaktivniju Reddit stranicu. Ripple je po ukupnom broju komentara i objava na drugom mjestu, a potom slijede Ethereum i Litecoin.



Slika 41: Omjer prosječnog dnevnog broja korisnika i komentara Reddit stranice

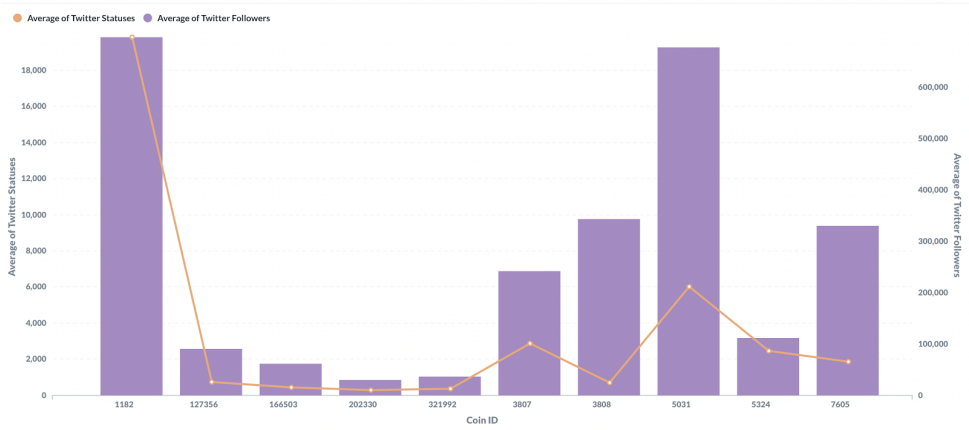
Graf predstavlja omjer prosječnog dnevnog broja Reddit korisnika i prosječnog broja dnevnih komentara na Reddit stranici određene kriptovalute. Lijeva skala predstavlja prosječan broj aktivnih korisnika, dok desna predstavlja prosječan broj dnevnih komentara. Bitcoin je baza za usporedbu omjera jer ima najveći broj prosječnog dnevnog broja korisnika i komentara na Reddit stranici. Ethereum ima velik aktivan broj korisnika, ali mali broj komentara. Analizirajući sadržaj Reddit stranice, može se primjetiti da zajednica na Bitcoin stranici objavljuje apsolutno sve: od vijesti, događaja, tehnoloških dostignuća, analiza i predikcija pa sve do šaljivih sadržaja i zabavnih materijala. S druge strane, Ethereum Reddit stranica isključivo je orijentirana na tehnologiju te imaju odvojenu stranicu za analize cijene i zabavu, zbog čega je prosječan broj komentara na glavnoj Ethereum Reddit stranici manji od očekivanog. S brojem pratitelja još se mogu istaknuti Ripple i Litecoin.



Slika 42: Trend kretanja dnevnih posjetitelja Reddit stranice

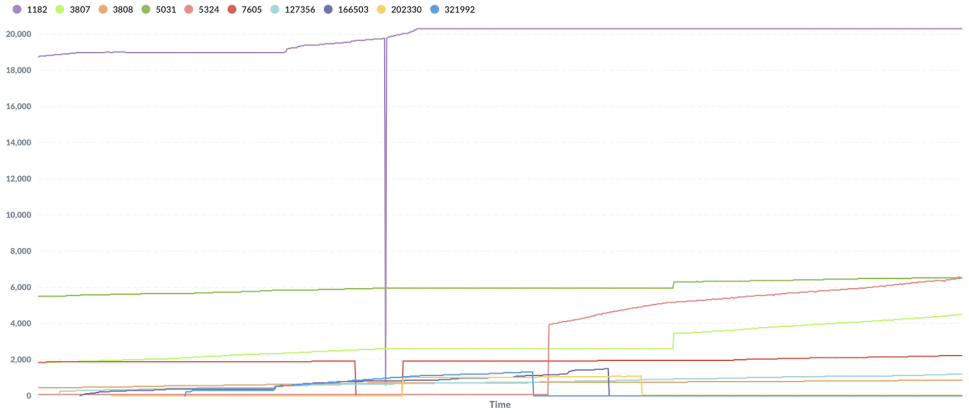
Graf predstavlja trend kretanja dnevnih posjetitelja Reddit stranica kriptovaluta. Najveći porast broja aktivnih korisnika sve su kriptovalute doživjele za vrijeme naglog rasta 2018. godine. Tada je naglo porastao interes za Ethereum Classic te Ripple kriptovalutama, nakon čega je entuzijazam ponovno splasnuo. Bitcoin uvjerljivo ima najveći broj aktivnih posjetitelja, a slijedi ga Ethereum. Generalno je vidljiv pad interesa javnosti za kriptovalutama nakon pada početkom 2019. godine.

4.4.4. Twitter



Slika 43: Omjer prosječnog dnevnog broja Twitter pratitelja i statusa

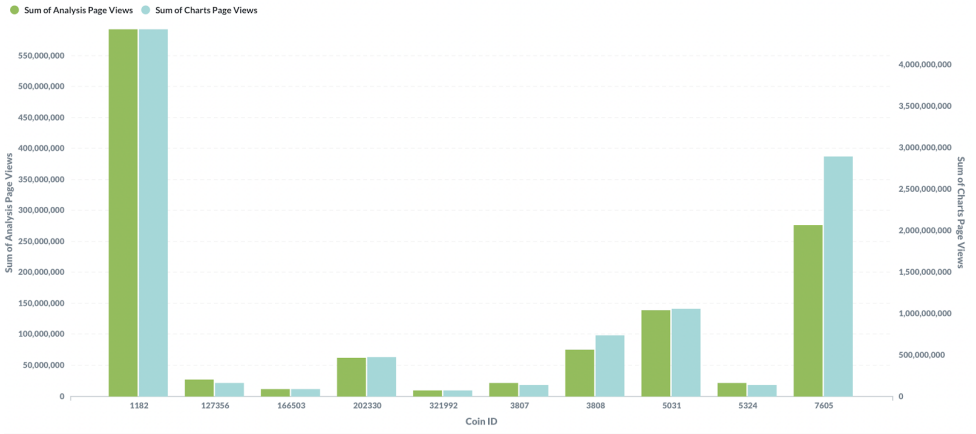
Graf predstavlja omjer prosječnog dnevnog broja Twitter pratitelja i statusa prema kriptovaluti. Lijeva skala predstavlja prosječan dnevni broj statusa, a desna prosječan broj pratitelja. Najveći broj pratitelja ima Bitcoin s 698,171, a slijedi ga Ripple s 768,303 pratitelja. Doduše, Bitcoin ima daleko veći broj Twitter statusa od Ripple-a. Na trećem mjestu je Litecoin s 343,349 pratitelja te potom Ethereum s 329,668 pratitelja.



Slika 44: Prosječan broja Twitter statusa

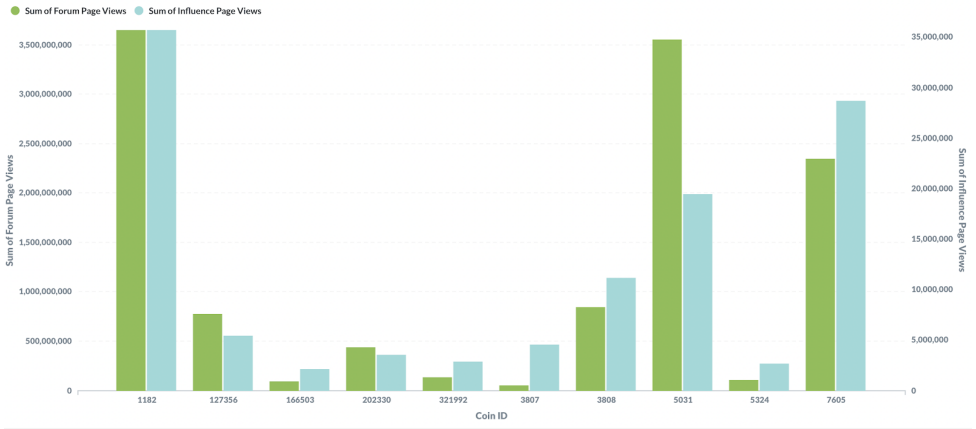
Graf prikazuje prosječan broj Twitter statusa kroz promatrani vremenski period. Najviše statusa ima Bitcoin, a slijedi ga Ripple. Ethereum je u naglom porastu te bi ovim trendom uskoro mogao prestići Ripple.

4.4.5. Broj pregleda



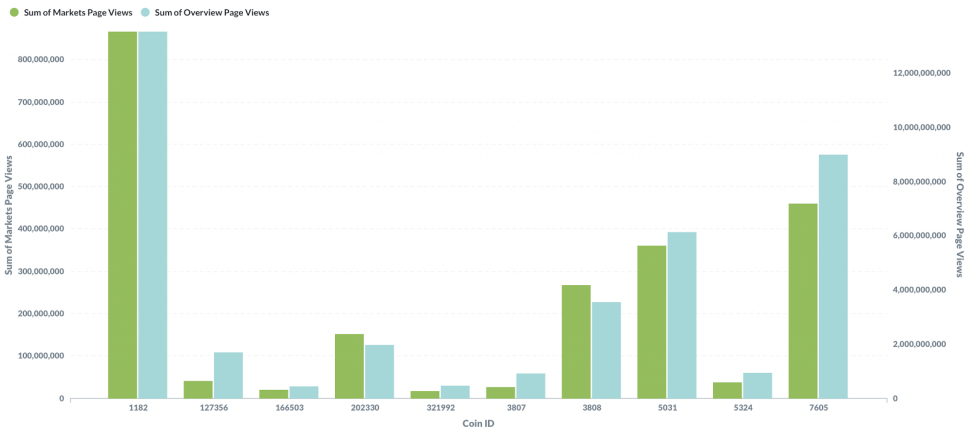
Slika 45: Pregledi stranica analiza i grafova

Graf predstavlja ukupan broj pregleda stranica vezanih uz analize i grafove za pojedinu kriptovalutu. Lijeva skala predstavlja ukupan broj pregleda analitilčkih stranica, a desna ukupan broj pregleda grafova. Temelj usporedbe je Bitcoin jer ima najveće sume oba promatrana parametra. Omjer pregleda promatranih parametara kod gotovo svih kriptovaluta je isti, izuzetak su Ethereum i Litecoin koji imaju veći broj pregleda grafova.



Slika 46: Pregledi foruma i utjecajnih stranica

Graf predstavlja ukupan broj pregleda foruma i utjecajnih stranica za pojedinu kriptovalutu. Lijeva skala predstavlja ukupan broj pregleda foruma, a desna ukupan broj pregleda utjecajnih stranica. Temelj usporedbe je Bitcoin jer ima najveće sume oba promatrana parametra, iako po broju pregleda foruma Ripple ne stoji daleko iza Bitcoina. Zajednica na forumima je aktivna i za Ethereum, nešto manje Litecoin i Iotu.



Slika 47: Pregledi tržišnih stranica i općih pregleda

Graf predstavlja ukupan broj pregleda tržišnih stranica i stranica općeg pregleda za pojedinu kriptovalutu. Lijeva skala predstavlja ukupan broj pregleda tržišnih stranica, a desna ukupan broj pregleda stranica općeg pregleda. Temelj usporedbe je Bitcoin jer ima najveće sume oba promatrana parametra. Ethereum je na drugom mjestu po oba promatrana parametra, a slijede ga Ripple te Litecoin.

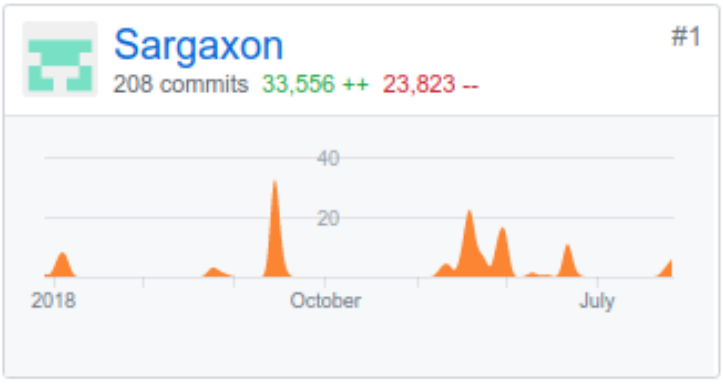
5. Tehnička dokumentacija

Sustav je napisan u programskom jeziku Python verzije 3.7 u razvojnom okružju (eng. *framework*) Flask. Za bazu podataka korišten je PostgreSQL. Korištene su vanjske biblioteke većinski pisane u programskom jeziku C. Osnovne biblioteke potrebne za funkcioniranje sustava su:

- **Flask** je razvojno okruženje za web aplikacije. [**52**]
- **SQLAlchemy** je objektni relacijski mapper (eng. *Object Relational Mapper*) koji pruža potpunu fleksibilnost nad SQL upitima. [**47**]
- **requests** je jednostavna HTTP biblioteka za Python. [**53**]
- **NumPy** je temeljni paket za kalkulacije u Pythonu. [**54**]
- **TA-lib** je omotač (eng. *wrapper*) za izračunavanje različitih tehničkih indikatora. [**55**]
- **Numba** pretvara dio Python ili Numpy koda u brzi strojni kod. [**56**]
- **Seaborn** je biblioteka za vizualizaciju podataka temeljena na matplotlib-u. [**57**]
- **Pandas** je biblioteka za analizu i vizualizaciju podataka. [**58**]

Cjelokupan je kod dokumentiran Python-ovom dokumentacijom (docstrings) jer pruža jednostavan način asociranja dokumentacije s Python-ovim modulima, funkcijama, klasama i metodama. Projekt se sastoji od tri glavna dijela: glavnog projekta, višeagentnog sustava i analize. Glavni projekt predstavlja temelj logike cijelog sustava te povezuje i omogućuje jednostanvu interakciju zasebnih cjelina. Višeagentni dio sadrži isključivo agente i njihove interne logike. Analiza je provedena uz pomoć alata *Metabase*. Svaki dio projekta detaljnije ćemo opisati u nastavku.

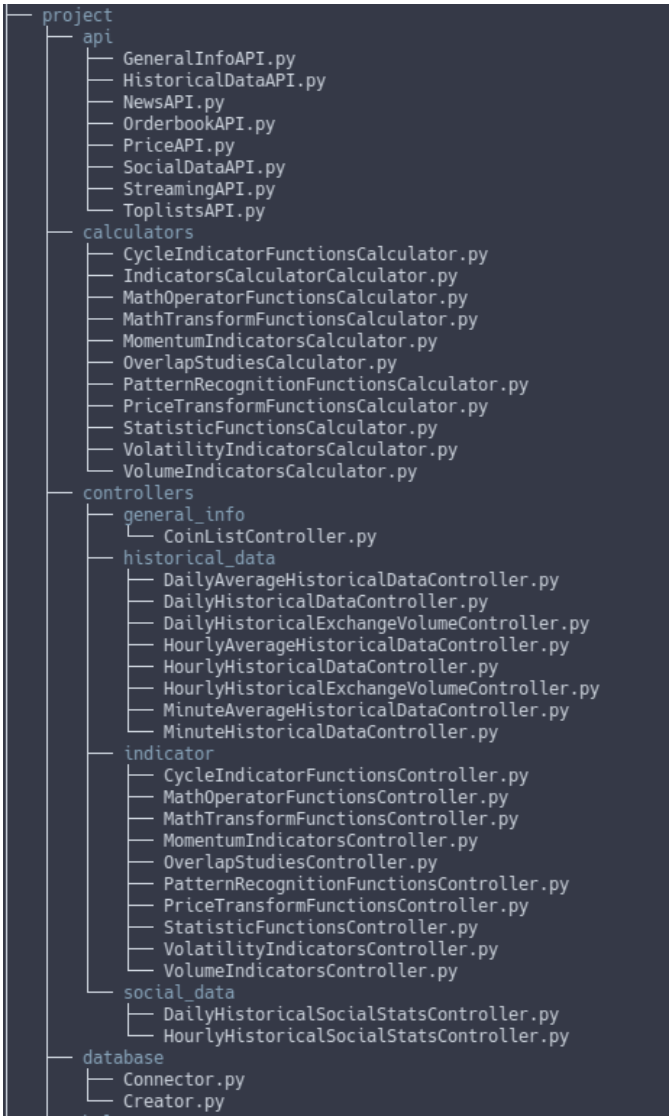
Projekt je kontinuirano testiran, čime su utvrđeni propusti i nedostaci sustava te glavne točke slabosti koje je bilo potrebno refaktorirati kako bi se osigurao optimalan performans sustava. Pritom su za provođenje svih kalkulacija, vizualizacija i ostalih procesorski težih radnji korištene vanjske biblioteke pisane u programskom jeziku C. Nakon svih refaktoriranja, sustav trenutno sadrži gotovo deset tisuća linija koda:

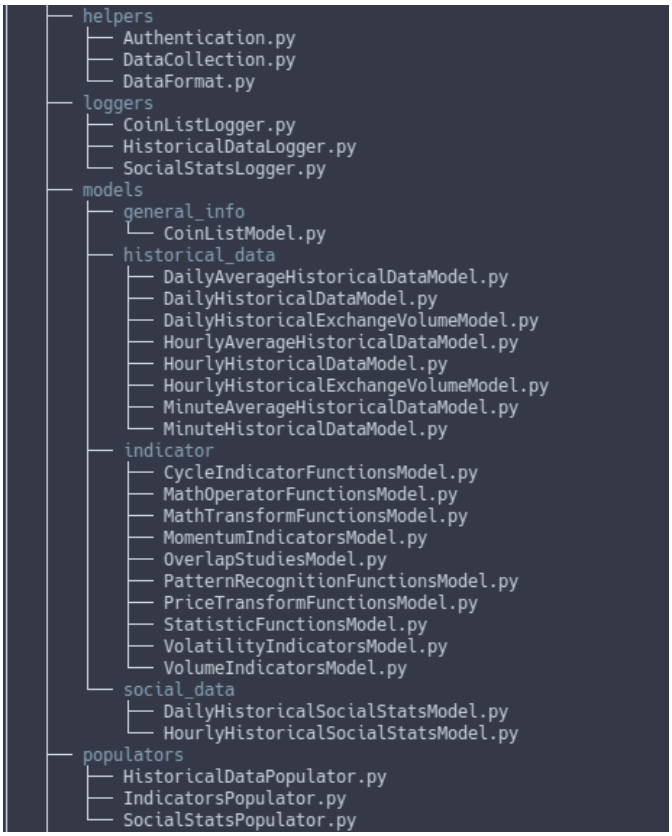


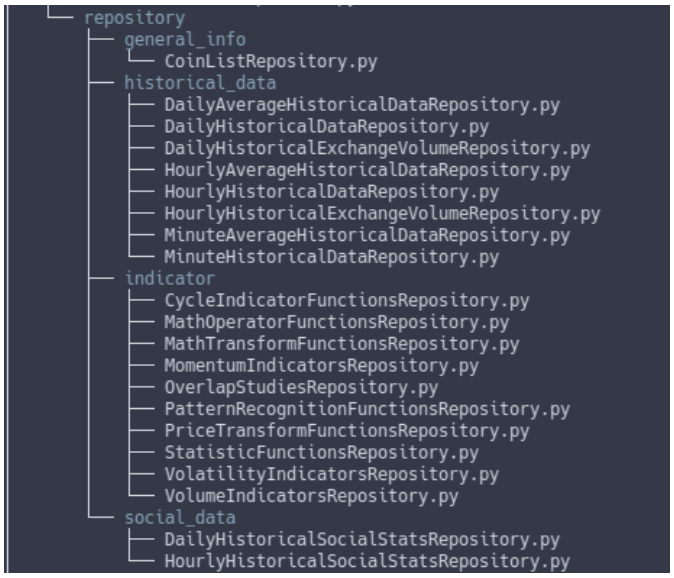
Slika 48: Kontribucije projektu

5.1. Projekt

Struktura projekta izgleda ovako:







Slika 49: Struktura projekta

5.1.1. API

Podaci potrebni za neometan rad sustava preuzeti su s *Cryptocompare* API-ja. Cryptocompare je jedna od vodećih globalnih institucija koja pruža povijesne podatke vezane uz kriptovalute, ali i podatke u stvarnom vremenu. U trenutku pisanja rada, sadrže sve podatke za 3219 različitih kriptovaluta, 236 burzi kriptovaluta, 149458 različitih parova za trgovanje (eng. *trading pairs*) te 44 izvora novosti i vijesti. Putem plaćenog Cryptocompare API-ja moguće je

ukupno poslati više od 40000 zahtjeva po sekundi i izvršiti više od 800 kupoprodaja po sekundi. Podržavaju ukupno 52 krajnje točke preko 8 različitih kategorija podataka koje se redovno ažuriraju s novim krajnjim točkama. [**48**]

U projektu je osigurana podrška za sve postojeće kategorije s pripadajućim krajnjim točkama API-ja. Svaka kategorija krajnjih točaka API-ja u pravilu ima istoimenu vlastitu klasu u projektu s metodama koje vrše poziv ka krajnjim točkama. Metode sadrže sve parametre koje je moguće poslati krajnjoj točki. Parametri su predefinirani na zadane vrijednosti, tako da se metodi mogu proslijediti samo željene izmjene. Prije poziva, u svakoj se metodi proširuje generirani API ključ. Odgovor se parsira i iz metode koja obrađuje krajnje točke izlaze formatirani podaci pogodni za daljnji rad. Nazivi metoda u projektu također se podudaraju s nazivima krajnjih točaka API-ja. Osim toga, svaka je metoda sa svim ulaznim parametrima dokumentirana sukladno dokumentaciji dostupnoj na API-ju.

5.1.1.1. Generalni podaci (*General info*)

Generalni podaci sadrže opće podatke o radnjama vezanim uz račun korisnika i ograničenja definirana po pojedinom API ključu. Osim osobnih uvida, sadrže podatke o trenutno podržanim podacima i radnjama samog API-ja. Za generalne podatke postoji ukupno 18 krajnjih točaka (eng. *endpoint*) iz kojih je moguće dobiti uvide u sljedeće:

- U slučaju da se koristi besplatno generiran API ključ, pruža uvid u broj mogućih i preostalih poziva po sekundi, minuti, satu, danu i mjesecu. Omogućuje 50 poziva po sekundi, 2500 po minuti, 25000 po satu, 50000 po danu i 100000 u jednom mjesecu.
- U slučaju da se koristi besplatno generiran API ključ, pruža uvid u broj preostalih poziva u trenutnom satu za pojedinu kategoriju krajnjih točaka.
- Popis podržanih burzi kriptovaluta.
- Sve sastavne burze kriptovaluta prema zadanoj kategoriji.
- Sve podržane parove kriptovaluta i burzi kriptovaluta.
- Sve ne podržane parove kriptovaluta i burzi kriptovaluta.
- Sve parove kriptovaluta i burzi kriptovaluta za kriptovalute koje nisu podržane na određenim burzama kriptovaluta.
- Popis svih podržanih kriptovaluta.
- Opći podaci i 24 satni volumen podržanih burzi kriptovaluta.
- Opći podaci o podržanim digitalnim novčanicima.
- Opći podaci o stanju rudarskih radnji.

Postoji još nekoliko krajnjih točaka koje nisu spomenute, ali sadrže male varijacije u odnosu na navedene pa su grupirane skupa. Primjer poziva jedne krajnje točke generalnih podataka u projektu za parove kriptovaluta i burzi kriptovaluta:

```
def fetch_all_exchanges_and_trading_pairs(self, coin_symbol='', exchange=''):
    """
    Returns all the exchanges that API has integrated with. You can filter by
        exchange and from symbol

    :param coin_symbol: The cryptocurrency symbol of interest [ Min length - 0] [
        Max length - 10] [ Default - ]
    :param exchange: The exchange to obtain data from (our aggregated average -
        CCCAGG - by default) [ Min length - 0] [ Max length - 30] [ Default - ]
    :return:
    """
    url = 'https://min-api.cryptocompare.com/data/v2/all/exchanges?fsym={}&e={}' \
    .format(coin_symbol.upper(), exchange)
    page = requests.get(url)
    data = page.json()['Data']
    return data
```

Metoda prima dva parametra, simbol kriptovalute i burzu. U slučaju da nisu prosljeđene vrijednosti, podaci će se vratiti za zadane vrijednosti. Krajnjoj točki pridružuje se API ključ te se zadana vrijednost simbola kriptovalute formatira na velika slova. Šalje se zahtjev te se iz odgovora dohvaćaju samo podaci.

5.1.1.2. Povijesni podaci (*Historical data*)

Povijesni podaci sadrže povijesne vrijednosti za parove kriptovaluta, valuta, burzi kriptovaluta u promatranim vremenskim intervalima od minute, sata ili dana. Za povijesne podatke postoji ukupno 8 krajnjih točaka (eng. *endpoint*) iz kojih je moguće dobiti uvide u sljedeće:

- Povijesne OHLCV cijene za zadanu kriptovalutu, valutu i burzu kriptovaluta u vremenskim intervalima od minute, sata ili dana.
- Povijesne OHLCV cijenu za zadanu kriptovalutu, valutu i burzu kriptovaluta te dani trenutak u vremenu (eng. *timestamp*)
- Prosječnu dnevnu OHLCV cijenu za zadanu kriptovalutu, valutu i burzu kriptovaluta te dani trenutak u vremenu. Podržava različite metode računanja dnevnog prosjeka.
- Povijesne volumene za zadanu valutu i burzu kriptovaluta u vremenskim intervalima od sata ili dana.
- Završnu tržišnu cijenu za sve parove kriptovaluta i burzi kriptovaluta.

Postoji još nekoliko krajnjih točaka koje nisu spomenute, ali sadrže male varijacije u odnosu

na navedene pa su grupirane skupa. Primjer poziva jedne krajnje točke povijesnih podataka u projektu za dnevne povijesne cijene:

```
def fetch_historical_daily_ohlcv(self, coin_symbol, comparison_symbol, exchange='
    CCCAGG', limit=30, aggregate=1, aggregate_periods=True, try_conversion=True,
    all_data=False, to_ts=''):
    """
    Get open, high, low, close, volumefrom and volumeto from the daily historical
        data.
    The values are based on 00:00 GMT time.
    It uses BTC conversion if data is not available because the coin is not trading
        in the specified currency.
    If you want to get all the available historical data, you can use limit=2000 and
         keep going back in time using the toTs param.
    You can then keep requesting batches using: &limit=2000&toTs={the earliest
        timestamp received}.
    :param coin_symbol: [REQUIRED] The cryptocurrency symbol of interest [ Min
        length - 1] [ Max length - 10]
    :param comparison_symbol: [REQUIRED] The currency symbol to convert into [ Min
        length - 1] [ Max length - 10]
    :param exchange: The exchange to obtain data from (our aggregated average -
        CCCAGG - by default) [ Min length - 2] [ Max length - 30] [ Default - CCCAGG
        ]
    :param limit: The number of data points to return [ Min - 1] [ Max - 2000] [
        Default - 30]
    :param aggregate: Time period to aggregate the data over (for daily it's days,
        for hourly it's hours and for minute histo it's minutes) [ Min - 1] [ Max -
        30] [ Default - 1]
    :param aggregate_periods: True by default, only used when the aggregate param is
         also in use. If false it will aggregate based on the current time.If the
        param is false and the time you make the call is 1pm - 2pm, with aggregate
        2, it will create the time slots: ... 9am, 11am, 1pm.If the param is false
        and the time you make the call is 2pm - 3pm, with aggregate 2, it will
        create the time slots: ... 10am, 12am, 2pm.If the param is true (default)
        and the time you make the call is 1pm - 2pm, with aggregate 2, it will
        create the time slots: ... 8am, 10am, 12pm.If the param is true (default)
        and the time you make the call is 2pm - 3pm, with aggregate 2, it will
        create the time slots: ... 10am, 12am, 2pm. [ Default - true]
    :param try_conversion: If set to false, it will try to get only direct trading
        values [ Default - true]
    :param all_data: Returns all data (only available on histo day) [ Default -
        false]
    :param to_ts: Returns historical data before that timestamp. If you want to get
        all the available historical data, you can use limit=2000 and keep going
        back in time using the toTs param. You can then keep requesting batches
        using: &limit=2000&toTs={the earliest timestamp received}
    :return:
    """
    url = self.append_api_key('https://min-api.cryptocompare.com/data/histoday?fsym
        ={}&tsym={}&limit={}&aggregate={}&aggregatePredictableTimePeriods={}&
        tryConversion={}&allData={}' \
```

```
    .format(coin_symbol.upper(), comparison_symbol.upper(), limit, aggregate, str(
        aggregate_periods).lower(), str(try_conversion).lower(), str(all_data).lower
        ()))
    if exchange:
    url += '&e={}'.format(exchange)
    if to_ts:
    url += '&toTs={}'.format(to_ts)
    page = requests.get(url)
    data = page.json()
    return data
```

Metoda prima devet parametara: simbol kriptovalute, valutu, burzu kriptovalute, broj podataka u odgovoru, broj vrijednosti za agregaciju, vremensku oznaku do koje da vrati podatke te nekoliko zastavica. U slučaju da nisu prosljeđene vrijednosti, podaci će se vratiti za predefinirane vrijednosti. Krajnjoj točki pridružuje se API ključ te se vrše potrebna formatiranja prije slanja zahtjeva. Šalje se zahtjev te se iz odgovora dohvaćaju svi podaci.

5.1.1.3. Novosti (*News*)

Novosti sadrže sve podatke vezane uz najnovije vijesti u kripto svijetu. Postoje ukupno 4 krajnje točke (eng. *endpoint*) iz kojih je moguće dobiti sljedeće uvide:

- Najnovije vijesti za zadane parametre.
- Listu izvora koje API trenutno podržava.
- Listu kategorija vijesti koje API sadrži.
- Listu izvora koje API trenutno podržava s pripadajućim kategorijama vijesti.

Primjer poziva jedne krajnje točke novosti u projektu za listu najnovijih članaka:

```
def fetch_latest_news_articles(self, feeds="ALL_NEWS_FEEDS", categories="
    ALL_NEWS_CATEGORIES", exclude_categories="NO_EXCLUDED_NEWS_CATEGORIES", last_ts
    =0, sort_order="latest", lang="EN"):
    """
    Returns news articles from the providers that API has integrated with.

    :param feeds: Specific news feeds to retrieve news from [ Min length - 1] [ Max
        length - 1000] [ Default - ALL_NEWS_FEEDS]
    :param categories: Category of news articles to return [ Min length - 3] [ Max
        length - 1000] [ Default - ALL_NEWS_CATEGORIES]
    :param exclude_categories: News article categories to exclude from results [ Min
         length - 3] [ Max length - 1000] [ Default - NO_EXCLUDED_NEWS_CATEGORIES]
    :param last_ts: Returns news before that timestamp [ Min - 0] [ Default - 0]
    :param sort_order: The order to return news articles - latest or popular [ Min
        length - 1] [ Max length - 8] [ Default - latest]
    :param lang: Preferred language - English (EN) or Portuguese (PT) [ Min length -
         1] [ Max length - 4] [ Default - EN]
```

```
    :return:
    """
    url = self.append_api_key('https://min-api.cryptocompare.com/data/v2/news/?feeds
        ={}&categories={}&excludeCategories={}&lTs={}&sortOrder={}&lang={}' \
    .format(feeds, categories, exclude_categories, last_ts, sort_order, lang.upper()
        ))
    page = requests.get(url)
    data = page.json()['Data']
    return data
```

Metoda prima šest parametara: izvore vijesti, kategorije, isključene kategorije, vremensku oznaku, redoslijed sortiranja te željeni jezik. U slučaju da nisu prosljeđene vrijednosti, podaci će se vratiti za zadane vrijednosti. Krajnjoj točki pridružuje se API ključ te se vrše potrebna formatiranja prije slanja zahtjeva. Šalje se zahtjev te se iz odgovora dohvaćaju svi podaci.

5.1.1.4. Knjiga narudžbi (*Orderbook*)

Knjige narudžbi pružaju informacije za rukovođenje kupoprodajnim vrijednostima. Postoje ukupno 3 krajnje točke vezane uz knjigu narudžbi pomoću kojih možemo dobiti uvid u sljedeće:

- Popis svih burzi kriptovaluta koje imaju knjigu narudžbi.
- Najnovije kupoprodajne vrijednosti za zadanu kriptovalutu, valutu i burzu kriptovaluta.
- Snimku kupoprodajnih vrijednosti za zadanu kriptovalutu, valutu i burzu kriptovaluta.

Primjer poziva jedne krajnje točke knjige narudžbi u projektu za snimku kupoprodajnih vrijednosti:

```
def fetch_orderbook_l2_snapshot(self, coin_symbol="BTC", comparison_symbol="USDT",
    exchange='Binance', limit=30):
    """
    You can only use this endpoint with a valid api_key and correct permissions,
        please get in touch if you want to test it.
    This is in BETA so the format and access rules might change. Returns latest
        orderbook L2 data snapshot for the requested exchange.
    The data format is seq, key, array of bids historical_data~quantity, array of
        asks historical_data~quantity.

    :param coin_symbol: The cryptocurrency symbol of interest [ Min length - 1] [
        Max length - 10] [ Default - BTC]
    :param comparison_symbol: The currency symbol to convert into [ Min length - 1]
        [ Max length - 10] [ Default - USDT]
    :param exchange: The exchange to obtain data from [ Min length - 2] [ Max length
         - 30] [ Default - Binance]
    :param limit: The number of top bids and asks to return. If you want them all,
        just pass in -1 for the limit param. [ Min - -1] [ Max - 2000] [ Default -
        30]
```

```
    :return:
    """
    url = self.append_api_key('https://min-api.cryptocompare.com/data/ob/l2/snapshot
        ?fsym={}&tsym={}&limit={}' \
    .format(coin_symbol.upper(), comparison_symbol.upper(), limit))
    if exchange:
    url += '&e={}'.format(exchange)
    page = requests.get(url)
    data = page.json()['Data']
    return data
```

Metoda prima četiri parametra: simbol kriptovalute, simbol valute, burzu kriptovaluta te ograničenje. U slučaju da nisu proslijeđene vrijednosti, podaci će se vratiti za zadane vrijednosti. Krajnjoj točki pridružuje se API ključ te se vrše potrebna formatiranja prije slanja zahtjeva. Šalje se zahtjev te se iz odgovora dohvaćaju samo podaci.

5.1.1.5. Cijene (*Price*)

Krajnje točke vezane uz cijene omogućuju konverzije između kriptovaluta. Postoje ukupno 4 krajnje točke vezane uz cijene pomoću kojih možemo dobiti uvid u sljedeće:

- Cijenu kriptovalute izraženu u bilo kojoj drugoj valuti.
- Cijenu mnoštva kriptovaluta izraženih u bilo kojoj drugoj valuti.
- Listu OHLCV cijena kriptovaluta izraženih u bilo kojoj drugoj valuti.
- Trenutni prosjek OHLCV cijena za zadanu kriptovalutu i burzu kriptovaluta izražeu u bilo kojoj drugoj valuti.

Primjer poziva jedne krajnje točke cijena u projektu za cijenu kriptovalute izraženu u bilo kojoj drugoj valuti:

```
def fetch_single_symbol_price(self, coin_symbol, comparison_symbols, exchange='
    CCCAGG', try_conversion=True):
    """
    Get the current historical_data of any cryptocurrency in any other currency that
         you need.
    If the crypto does not trade directly into the toSymbol requested, BTC will be
        used for conversion.
    If the opposite pair trades we invert it (eg.: BTC-XMR)

    :param coin_symbol: [REQUIRED] The cryptocurrency symbol of interest [ Min
        length - 1] [ Max length - 10]
    :param comparison_symbols: [REQUIRED] Comma separated cryptocurrency symbols
        list to convert into [ Min length - 1] [ Max length - 500]
    :param exchange: The exchange to obtain data from (our aggregated average -
        CCCAGG - by default) [ Min length - 2] [ Max length - 30] [ Default - CCCAGG
        ]
```

```
    :param try_conversion: If set to false, it will try to get only direct trading
        values [ Default - true]
    :return:
    """
    url = self.append_api_key('https://min-api.cryptocompare.com/data/
        historical_data?fsym={}&tsyms={}&tryConversion={}' \
    .format(coin_symbol.upper(), comparison_symbols.upper(), str(try_conversion).
        lower()))
    if exchange:
    url += '&e={}'.format(exchange)
    page = requests.get(url)
    data = page.json()
    return data
```

Metoda prima četiri parametra: simbol kriptovalute, simbol valute, burzu kriptovaluta te zastavicu za konverziju. U slučaju da nisu prosljeđene vrijednosti, podaci će se vratiti za zadane vrijednosti. Krajnjoj točki pridružuje se API ključ te se vrše potrebna formatiranja prije slanja zahtjeva. Šalje se zahtjev te se iz odgovora dohvaćaju svi podaci.

5.1.1.6. Društveni podaci (*Social data*)

Krajnje točke vezane uz društvene podatke pružaju statistike za mnoštvo društvenih stranica, kao što su Facebook, Reddit, Twitter i druge. Postoje ukupno 3 krajnje točke vezane uz cijene pomoću kojih možemo dobiti uvid u sljedeće:

- Statistike društvenih stranica za zadanu kriptovalutu.
- Povijesne statistike društvenih stranica za zadanu kriptovalutu u vremenskim intervalima od sata ili dana.

Primjer poziva jedne krajnje točke društvenih podataka u projektu za dnevnu povijesnu statistiku:

```
def fetch_historical_day_social_stats_data(self, coin_id=1182, limit=30, aggregate
    =1, aggregate_periods=True, to_ts=''):
    """
    You can only use this endpoint with a valid api_key.
    Returns daily social stats data for the coin requested

    :param coin_id: The id of the coin you want data for. [ Default - 1182]
    :param limit: The number of data points to return [ Min - 1] [ Max - 2000] [
        Default - 30]
    :param aggregate: Time period to aggregate the data over (for daily it's days,
        for hourly it's hours and for minute histo it's minutes) [ Min - 1] [ Max -
        30] [ Default - 1]
    :param aggregate_periods: True by default, only used when the aggregate param is
         also in use. If false it will aggregate based on the current time.If the
        param is false and the time you make the call is 1pm - 2pm, with aggregate
```

```
        2, it will create the time slots: ... 9am, 11am, 1pm.If the param is false
        and the time you make the call is 2pm - 3pm, with aggregate 2, it will
        create the time slots: ... 10am, 12am, 2pm.If the param is true (default)
        and the time you make the call is 1pm - 2pm, with aggregate 2, it will
        create the time slots: ... 8am, 10am, 12pm.If the param is true (default)
        and the time you make the call is 2pm - 3pm, with aggregate 2, it will
        create the time slots: ... 10am, 12am, 2pm. [ Default - true]
    :param to_ts: Returns historical data before that timestamp. If you want to get
        all the available historical data, you can use limit=2000 and keep going
        back in time using the toTs param. You can then keep requesting batches
        using: &limit=2000&toTs={the earliest timestamp received}
    :return:
    """
    url = self.append_api_key('https://min-api.cryptocompare.com/data/social/coin/
        histo/day?coinId={}&limit={}&aggregate={}&aggregatePredictableTimePeriods={}
        ' \
    .format(coin_id, limit, aggregate, str(aggregate_periods).lower()))
    if to_ts:
    url += '&toTs={}'.format(to_ts)
    page = requests.get(url)
    data = page.json()['Data']
    return data
```

Metoda prima pet parametara: id kriptovalute, broj povijesnih podataka u odgovoru, broj vrijednosti za agregaciju, zastavicu za agergaciju te vremensku oznaku. U slučaju da nisu prosljeđene vrijednosti, podaci će se vratiti za zadane vrijednosti. Krajnjoj točki pridružuje se API ključ te se vrše potrebna formatiranja prije slanja zahtjeva. Šalje se zahtjev te se iz odgovora dohvaćaju samo podaci.

5.1.1.7. Strujanje (*Streaming*)

Krajnje točke vezane uz strujanja pružaju agregirane informacije volumena i cijena kriptovaluta. Postoje ukupno 5 krajnjih točki vezanih uz strujanja pomoću kojih možemo dobiti uvid u sljedeće:

- Definirani broj kriptovaluta na svim burzama kriptovaluta prema ukupnom volumenu u posljednja 24 sata.
- Definirani broj kriptovaluta na top 20 burzi kriptovaluta prema ukupnom ranku volumena u posljednja 24 sata.
- Definirani broj kriptovaluta na svim burzama kriptovaluta prema tržišnoj kapitalizaciji
- Listu svih dostupnih kanala strujanja (eng. *streaming channels*).
- Kombinacije strujanja i informacija o cijenama.

Postoji još nekoliko krajnjih točaka koje nisu spomenute, ali sadrže male varijacije u odnosu na navedene pa su grupirane skupa. Primjer poziva jedne krajnje točke strujanja u projektu za listu najboljih kriptovaluta prema volumenu:

```
def fetch_toplist_by_24h_volume_subscriptions(self, comparison_symbol, limit=10,
    pagination=0):
    """
    Get a number of top coins by their total volume across all markets in the last
        24 hours. Default value is first page (0) and the top 10 coins.


    :param comparison_symbol: [REQUIRED] The currency symbol to convert into [ Min
        length - 1] [ Max length - 10]
    :param limit: The number of coins to return in the toplist, default 10, min 10,
        max 100 will round to steps of 10 coins [ Min - 10] [ Max - 100] [ Default -
         10]
    :param pagination: The pagination for the request. If you want to paginate by 50
         for example, pass in the limit_toplist param the value 50 and increasing
        page_toplist integer values, 0 would return coins 0-50, 1 returns coins
        50-100 [ Min - 0] [ Default - 0]
    :return:
    """
    url = self.append_api_key('https://min-api.cryptocompare.com/data/top/totalvol?
        tsym={}&limit={}&page={}' \
    .format(comparison_symbol.upper(), limit, pagination))
    page = requests.get(url)
    data = page.json()['Data']
    return data
```

Metoda prima tri parametra: simbol kriptovalute, broj kriptovaluta u odgovoru te broj paginacije. U slučaju da nisu prosljeđene vrijednosti, podaci će se vratiti za zadane vrijednosti. Krajnjoj točki pridružuje se API ključ te se vrše potrebna formatiranja prije slanja zahtjeva. Šalje se zahtjev te se iz odgovora dohvaćaju samo podaci.

5.1.1.8. Top liste (*Toplists*)

Krajnje točke vezane uz top liste pružaju agregirane informacije volumena i tržišnih kapitalizacija kriptovaluta za posljednja 24 sata. Postoje ukupno 7 krajnjih točki vezanih uz top liste pomoću kojih možemo dobiti uvid u sljedeće:

- Najbolje kriptovalute na svim burzama kriptovaluta u posljednjih 24 sata prema volumenu.
- Najbolje kriptovalute prema tržišnoj kapitalizaciji.
- Najbolje burze kriptovaluta prema volumenu za zadani par kriptovalute i valute.
- Najbolje kriptovalute prema paru volumena.

Postoji još nekoliko krajnjih točaka koje nisu spomenute, ali sadrže male varijacije u odnosu na navedene pa su grupirane skupa. Primjer poziva jedne krajnje točke top čiste u projektu za najbolje kriptovalute prema tržišnoj kapitalizaciji:

```
def fetch_toplist_by_market_cap_full_data(self, comparison_symbol, limit=10,
    pagination=0):
    """
    Get a number of top coins by their market cap. Default value is first page (0)
        and the top 10 coins.

    :param comparison_symbol: [REQUIRED] The currency symbol to convert into [ Min
        length - 1] [ Max length - 10]
    :param limit: The number of coins to return in the toplist, default 10, min 10,
        max 100 will round to steps of 10 coins [ Min - 10] [ Max - 100] [ Default -
         10]
    :param pagination: The pagination for the request. If you want to paginate by 50
         for example, pass in the limit_toplist param the value 50 and increasing
        page_toplist integer values, 0 would return coins 0-50, 1 returns coins
        50-100 [ Min - 0] [ Default - 0]
    :return:
    """
    url = self.append_api_key('https://min-api.cryptocompare.com/data/top/mktcapfull
        ?tsym={}&limit={}&page={}' \
    .format(comparison_symbol.upper(), limit, pagination))
    page = requests.get(url)
    data = page.json()['Data']
    return data
```

Metoda prima tri parametra: simbol kriptovalute, broj kriptovaluta u odgovoru te broj paginacije. U slučaju da nisu prosljeđene vrijednosti, podaci će se vratiti za zadane vrijednosti. Krajnjoj točki pridružuje se API ključ te se vrše potrebna formatiranja prije slanja zahtjeva. Šalje se zahtjev te se iz odgovora dohvaćaju samo podaci.

5.1.2. Baza podataka *(Database)*

Odabrani sustav za upravljanje bazom podataka jest PostgreSQL. U projektu se koristi SQLAlchemy biblioteka za spajanje i komunikaciju s bazom podataka. SQLAlchemy je objekti relacijski maper (eng. *Object Relational Mapper*) koji pruža potpunu moć i fleksibilnost nad SQL-om. Omogućuje niz poznatih obrazaca dizajniranih za brz i efikasan pristup bazama podataka pomoću Python programskog jezika. [**46**, **47**]

Kroz projekt se koristi zadana baza podataka i korisnik *'postgres'* koji dolaze predefinirani i konfigurani s instalacijom postgreSQL-a. Baza podataka nije normalizirana s obzirom da su podaci međusobno neovisni. Spajanje na bazu i kreiranje sesije koja se nadalje koristi u projektu za rad s bazom podataka prikazana je u sljedećem isječku koda:

```
"""
Connects to database engine (dialect+driver://username:password@host:port/database)
"""
engine = create_engine(app.config['DATABASE'], encoding="utf-8")
session = scoped_session(sessionmaker(autocommit=False,
    autoflush=False,
```

```
    bind=engine))

Base = declarative_base()
Base.query = session.query_property()
```

Vrijednosti potrebne za povezivanje na bazu podataka privremeno su pohranjene u konfiguraciji projekta. Pokretanjem projekta, stvara se instanca sesije baze podataka putem koje se izvršavaju upiti, spremaju, ažuriraju ili dohvaćaju podaci iz baze podataka. Prilikom prvog pokretanja projekta, izvršit će se niz SQL skripti koje kreiraju sve potrebne tablice (sa zadanim atributima i tipovima podataka) korištene u projektu. Dakle, instaliran postgreSQL sa zadanim korisnikom i konfiguracijom jedina je predispozicija normalnom djelovanju cjelokupnog sustava. Naravno, prije toga valjalo bi preuzeti i instalirati sve ostale biblioteke korištene u projektu. Nakon toga, dovoljno je pokrenuti projekt i sve ostalo kreira se samo od sebe. U nastavku primjer sintakse kreiranja jedne tablice u bazi podataka s pripadajućim atributima i tipovima podataka:

```
def create_general_info_tables(db):
    db.execute('CREATE TABLE IF NOT EXISTS coin_list'
    '(idx serial primary key, '
    'name text, '
    'full_name text, '
    'coin_name text, '
    'symbol text, '
    'id int,'
    'content_created_on text,'
    'total_coin_supply text, '
    'pre_mined_value text, '
    'fully_premined text, '
    'algorithm text, '
    'url text, '
    'sort_order text, '
    'sponsored text, '
    'image_url text, '
    'proof_type text, '
    'total_coins_free_float text, '
    'is_trading text, '
    'built_on text, '
    'smart_contract_address text, '
    'total_coins_mined text, '
    'block_number text, '
    'net_hashes_per_second text, '
    'block_reward text, '
    'block_time text);')
```

Svakoj krajnjoj točki API-ja koja ima povijesni prikaz, listu ili neki oblik agregacije podataka dodjeljena je vlastita tablica u bazi podataka. Jedan od razloga jest performans sustava zbog količine podataka pohranjene permutacijama svih kriptovaluta, valuta, burzi kriptovaluta i ostalih parametara, ali i zbog pojednostavljivanja arhitekture. Osim podataka s API-ja, trenutno

se pohranjuju i određene ručno rađene kalkulacije koje nisu bile dostupne na API-ju, kao što su tehnički indikatori ili prosječne povijesne cijene. Popis svih postojećih tablica u pazi podataka:

- **coin_list:** sadrži sve dostupne kriptovalute i podatke o istima. Podaci su preuzeti s krajnje točke generalnih podataka.
- **histo_day_social_stats:** sadrži sve statistike društvenih podatka za vremenski interval od jednog dana. Podaci su preuzeti s krajnje točke društvenih podataka u vremenskom intervalu od jednog dana.
- **histo_hour_social_stats:** sadrži sve statistike društvenih podatka za vremenski interval od jednog sata. Podaci su preuzeti s krajnje točke društvenih podataka u vremenskom intervalu od jednog sata.
- **histo_day:** sadrži sve povijesne OHLCV podatke za vremenski interval od jednog dana. Podaci su preuzeti s krajnje točke povijesnih OHLCV podataka u vremenskom intervalu od jednog dana.
- **histo_hour:** sadrži sve povijesne OHLCV podatke za vremenski interval od jednog sata. Podaci su preuzeti s krajnje točke povijesnih OHLCV podataka u vremenskom intervalu od jednog sata.
- **histo_minute:** sadrži sve povijesne OHLCV podatke za vremenski interval od jedne minute. Podaci su preuzeti s krajnje točke povijesnih OHLCV podataka u vremenskom intervalu od jedne minute.
- **histo_day_average:** sadrži izračunati prosjek za sve povijesne OHLCV podatke svih burzi kriptovaluta za vremenski interval od jednog dana. Podaci su izračunati iz povijesnih OHLCV podataka u vremenskom intervalu od jednog dana.
- **histo_hour_average:** sadrži izračunati prosjek za sve povijesne OHLCV podatke svih burzi kriptovaluta za vremenski interval od jednog sata. Podaci su izračunati iz povijesnih OHLCV podataka u vremenskom intervalu od jednog sata.
- **histo_minute_average:** sadrži izračunati prosjek za sve povijesne OHLCV podatke svih burzi kriptovaluta za vremenski interval od jedne minute. Podaci su izračunati iz povijesnih OHLCV podataka u vremenskom intervalu od jedne minute.
- **histo_day_exchange_volume:** sadrži sve povijesne volumene burzi kriptovaluta za vremenski interval od jednog dana. Podaci su preuzeti s krajnje točke povijesnih podataka volumena burzi kriptovaluta u vremenskom intervalu od jednog dana.
- **histo_hour_exchange_volume:** sadrži sve povijesne volumene burzi kriptovaluta za vremenski interval od jednog sata. Podaci su preuzeti s krajnje točke povijesnih podataka volumena burzi kriptovaluta u vremenskom intervalu od jednog sata.
- **overlap_studies:** sadrži izračune 17 različitih tehničkih indikatora kroz povijest u kategoriji funkcija preklapanja (eng. *overlap studies function*). Računa se na osnovi povijesnih

OHLCV podataka za parove kriptovaluta, valuta i burzi kriptovaluta za vremenske intervale od minute, sata ili dana.

- **momentum_indicators:** sadrži izračune 40 različitih tehničkih indikatora kroz povijest u kategoriji indikatora trenda (eng. *momentum indicators*). Računa se na osnovi povijesnih OHLCV podataka za parove kriptovaluta, valuta i burzi kriptovaluta za vremenske intervale od minute, sata ili dana.
- **volume_indicators:** sadrži izračune 3 različita tehnička indikatora kroz povijest u kategoriji indikatora volumena (eng. *volume indicators*). Računa se na osnovi povijesnih OHLCV podataka za parove kriptovaluta, valuta i burzi kriptovaluta za vremenske intervale od minute, sata ili dana.
- **volatility_indicators:** sadrži izračune 3 različita tehnička indikatora kroz povijest u kategoriji indikatora promjenjivosti (eng. *volatility indicators*). Računa se na osnovi povijesnih OHLCV podataka za parove kriptovaluta, valuta i burzi kriptovaluta za vremenske intervale od minute, sata ili dana.
- **pattern_recognition_functions:** sadrži izračune 60 različitih tehničkih indikatora kroz povijest u kategoriji funkcija prepoznavanja obrazaca (eng. *pattern recognition functions*). Računa se na osnovi povijesnih OHLCV podataka za parove kriptovaluta, valuta i burzi kriptovaluta za vremenske intervale od minute, sata ili dana.
- **cycle_indicator_functions:** sadrži izračune 7 različitih tehničkih indikatora kroz povijest u kategoriji cikličnih indikatorskih funkcija (eng. *cycle indicator functions*). Računa se na osnovi povijesnih OHLCV podataka za parove kriptovaluta, valuta i burzi kriptovaluta za vremenske intervale od minute, sata ili dana.
- **statistic_functions:** sadrži izračune 9 različitih tehničkih indikatora kroz povijest u kategoriji statističkih funkcija (eng. *statistic functions*). Računa se na osnovi povijesnih OHLCV podataka za parove kriptovaluta, valuta i burzi kriptovaluta za vremenske intervale od minute, sata ili dana.
- **price_transform_functions:** sadrži izračune 4 različita tehnička indikatora kroz povijest u kategoriji funkcija promjene cijene (eng. *price transform functions*). Računa se na osnovi povijesnih OHLCV podataka za parove kriptovaluta, valuta i burzi kriptovaluta za vremenske intervale od minute, sata ili dana.
- **math_transform_functions:** sadrži izračune 15 različitih tehničkih indikatora kroz povijest u kategoriji funkcija matematičke promjene (eng. *math transform functions*). Računa se na osnovi povijesnih OHLCV podataka za parove kriptovaluta, valuta i burzi kriptovaluta za vremenske intervale od minute, sata ili dana.
- **math_operator_functions:** sadrži izračune 11 različitih tehničkih indikatora kroz povijest u kategoriji funkcija matematičkih operatora (eng. *math operator functions*). Računa se na osnovi povijesnih OHLCV podataka za parove kriptovaluta, valuta i burzi kriptovaluta za vremenske intervale od minute, sata ili dana.

Do sada su aktivno prikupljeni svi povijesni podaci (do sadašnjeg trenutka) za listu kriptovaluta te njihove pripadajuće dnevne i satne društvene podatke. Zbog ograničenja broja poziva besplatnog API ključa, za sve ostale tablice populirane su i redovno ažurirane permutacije 6 najpopularnijih burzi kriptovaluta, 10 kriptovaluta koje su u vrijeme pisanja rada imale najveću tržišnu kapitalizaciju te 3 valute. Iako, postojeća arhitektura dopušta jednostavno dodavanje, populiranje i ažuriranje novih parova burzi kriptovaluta, kriptovaluta i valuta. U bazu podataka trenutno su pohranjene permutacije vrijednosti:

- **burze kriptovaluta:** Bitfinex, Wavesdex, Gemini, Coinbase, Bitstamp te Bittrex.
- **kriptovalute:** "BTC" (Bitcoin), "ETH" (Ethereum), "EOS" (EOSIO), "BCH" (Bitcoin Cash), "LTC" (LiteCoin), "ETC" (Ethereum Classic), "DASH" (Dash), "XRP" (Ripple), "ADA" (Cardano), "IOT" (Iota)
- **valute:** "EUR" (Euro), "USD" (Dolar), "BTC" (Bitcoin)

5.1.3. Modeli *(Models)*

Modeli predstavljaju apstrakciju podataka koji su korišteni u sklopu projekta. Grupirani su po direktorijima, pri čemu svaki direktorij predstavlja jednu kategoriju krajnjih točaka, a svaki model predstavlja podatkovnu strukturu odgovora pojedine krajnje točke. Dakle, svaki model predstavlja jednu tablicu u bazi podataka. U slučaju SQLAlchemy arhitekture, za rad s bazom podataka struktura prosječnog modela izgleda ovako:

```
class CoinListModel(Base):
    """
    Model for list of crypto currency coins.
    """

    idx = Column(Integer, primary_key=True)
    name = Column(Text)
    full_name = Column(Text)
    coin_name = Column(Text)
    symbol = Column(Text)
    id = Column(Integer)
    content_created_on = Column(Text)
    total_coin_supply = Column(Text)
    pre_mined_value = Column(Text)
    fully_premined = Column(Text)
    algorithm = Column(Text)
    url = Column(Text)
    sort_order = Column(Text)
    sponsored = Column(Text)
    image_url = Column(Text)
    proof_type = Column(Text)
    total_coins_free_float = Column(Text)
    is_trading = Column(Text)
```

```
built_on = Column(Text)
smart_contract_address = Column(Text)
total_coins_mined = Column(Text)
block_number = Column(Text)
net_hashes_per_second = Column(Text)
block_reward = Column(Text)
block_time = Column(Text)

def __init__(self,
        name=None,
        full_name=None,
        coin_name=None,
        symbol=None,
        id=None,
        content_created_on=None,
        total_coin_supply=None,
        pre_mined_value=None,
        fully_premined=None,
        algorithm=None,
        url=None,
        sort_order=None,
        sponsored=None,
        image_url=None,
        proof_type=None,
        total_coins_free_float=None,
        is_trading=None,
        built_on=None,
        smart_contract_address=None,
        total_coins_mined=None,
        block_number=None,
        net_hashes_per_second=None,
        block_reward=None,
        block_time=None,
        ):
    self.name = name
    self.full_name = full_name
    self.coin_name = coin_name
    self.symbol = symbol
    self.id = id
    self.content_created_on = content_created_on
    self.total_coin_supply = total_coin_supply
    self.pre_mined_value = pre_mined_value
    self.fully_premined = fully_premined
    self.algorithm = algorithm
    self.url = url
    self.sort_order = sort_order
    self.sponsored = sponsored
    self.image_url = image_url
    self.proof_type = proof_type
    self.total_coins_free_float = total_coins_free_float
    self.is_trading = is_trading
    self.built_on = built_on
    self.smart_contract_address = smart_contract_address
```

```
        self.total_coins_mined = total_coins_mined
        self.block_number = block_number
        self.net_hashes_per_second = net_hashes_per_second
        self.block_reward = block_reward
        self.block_time = block_time
```

Svaki model nasljeđuje baznu klasu (Base) opisanu u prethodnom poglavlju baze podataka. Modeli sadrže atribute i tipove podataka istih u obliku u kojem se nalaze u tablici baze podataka. Svaki model ima odgovarajući konstruktor za postavljanje inicijalnih vrijednosti prilikom kreiranja novih objekata ove klase.

5.1.4. Repozitoriji *(Repositories)*

Repozitorij je sloj pristupa bazi podataka, sadrži logiku za CRUD operacije nad entitetima u sustavu i time omogućuje perzistenciju podataka kroz vrijeme. Repozitoriji su razina apstrakcije između kontrolera i modela, a rade direktno s bazom podataka. Isto kao i modeli, grupirani su po direktorijima kategorija krajnjih točaka. Svaki repozitorij sadrži metode za manipuliranje sadržaja jednog modela. U biblioteci SQLAlchemy to izgleda ovako:

```
class DailyHistoricalDataRepository(Base):
    """
    Repository for daily historical crypto currency data.
    """
    __tablename__ = "histo_day"
```

Svaki repozitorij nasljeđuje baznu klasu (Base) i sadrži naziv tablice baze podataka nad kojom provodi radnje. Osnovne radnje sadržane u svakom repozitoriju su kreiranje (eng. *create*), ažuriranje (eng. *update*) te brisanje (eng. *delete*). Pomoću SQLAlchemya, osnovne radnje realiziraju se na sljedeći način:

```
@staticmethod
def insert(daily_historical_data_model):
    """
    Save record to database.
    :return:
    """
    session.add(daily_historical_data_model)
    session.commit()

@staticmethod
def update():
    session.commit()

@staticmethod
def delete(entity_id):
    """
```

```
    Delete record with given ID.
    :param entity_id:
    :return:
    """
    DailyHistoricalDataModel.query.filter(DailyHistoricalDataModel.idx == entity_id)
        .delete()
    session.commit()
```

Za dodavanje novih podataka u odgovarajuću tablicu, dovoljno je proslijediti instancu objekta modela. Ažuriranje se provodi nad samim objektom, a za brisanje je u ovom slučaju potrebno proslijediti id retka u tablici koji želimo izbrisati. Na primjeru brisanja može se vidjeti način formiranja upita pomoću biblioteke SQLAlchemy. Osim osnovnih radnji, repozitorijima su često dodane nove metode za brži i lakši pristup dohvaćanju ili agregiranju željenih podataka. U nastavku slijedi nekoliko klasičnih primjera koji se često pojavljuju u repozitorijima.

```
@staticmethod
    def find_id(entity_id):
    """
    Find record with given ID.
    :param entity_id:
    :return:
    """
    return DailyHistoricalDataModel.query.filter(DailyHistoricalDataModel.idx ==
        entity_id).first()

@staticmethod
def find_exact_price(entity_time, entity_coin_symbol, entity_currency_symbol,
    entity_exchange):
    """
    Find exact historical historical_data for given time, coin, currency and
        exchange.
    :param entity_time:
    :param entity_coin_symbol:
    :param entity_currency_symbol:
    :param entity_exchange:
    :return:
    """
    return DailyHistoricalDataModel.query.filter(and_(DailyHistoricalDataModel.time
        == str(entity_time),
                                DailyHistoricalDataModel.coin_symbol == str(
                                    entity_coin_symbol),
                                DailyHistoricalDataModel.currency_symbol == str(
                                    entity_currency_symbol),
                                DailyHistoricalDataModel.exchange == str(
                                    entity_exchange))).first()
```

U prvom slučaju (*find_id*) dohvaća se jedan unos s prosljeđenim id-jem. Druga metoda (*find_exact_price*) dohvaća točnu cijenu za kriptovalutu, valutu i burzu kriptovalute u točno određenom trenutku u vremenu.

```
@staticmethod
def find_coin_symbol(coin_symbol):
    """
    Find all data for given coin.
    :param coin_symbol:
    :return:
    """
    return DailyHistoricalDataModel.query.filter(DailyHistoricalDataModel.
        coin_symbol == coin_symbol)

@staticmethod
def find_min_ts(entity_coin_symbol, entity_currency_symbol, entity_exchange):
    """
    Find the earliest database record.
    :return:
    """
    return DailyHistoricalDataModel.query.filter(and_(DailyHistoricalDataModel.
        coin_symbol == str(entity_coin_symbol),
        DailyHistoricalDataModel.currency_symbol == str(entity_currency_symbol),
        DailyHistoricalDataModel.exchange == str(entity_exchange))).with_entities(
            func.min(DailyHistoricalDataModel.time)).first()


@staticmethod
def find_max_ts(entity_coin_symbol, entity_currency_symbol, entity_exchange):
    """
    Find the latest database record
    :return:
    """
    return DailyHistoricalDataModel.query.filter(and_(DailyHistoricalDataModel.
        coin_symbol == str(entity_coin_symbol),
        DailyHistoricalDataModel.currency_symbol == str(entity_currency_symbol),
        DailyHistoricalDataModel.exchange == str(entity_exchange))).with_entities(
            func.max(DailyHistoricalDataModel.time)).first()
```

U prvoj metodi (*find_coin_symbol*) dohvaćaju se svi podaci vezani uz prosljeđenu kriptovalutu. Druga metoda (*find_min_ts*) dohvaća najraniji (prvi) trenutak u vremenu određene kriptovalute, valute i burze kriptovaluta, a treća metoda (*find_max_ts*) dohvaća najkasniji (posljednji) trenutak u vremenu. Postoje metode, tj. upiti, specifični za određeni repozitorij, kao što je sljedeći:

```
@staticmethod
def find_averages(entity_coin_symbol, entity_currency_symbol):
    """
    Calculate average prices from all exchanges for specific time, coin and currency
        .
    :return:
    """
    return DailyHistoricalDataModel.query.filter(
```

```
    and_(DailyHistoricalDataModel.coin_symbol == str(entity_coin_symbol),
        DailyHistoricalDataModel.currency_symbol == str(entity_currency_symbol),
        DailyHistoricalDataModel.open > 0,
        DailyHistoricalDataModel.high > 0,
        DailyHistoricalDataModel.low > 0,
        DailyHistoricalDataModel.close > 0,
        DailyHistoricalDataModel.volumefrom > 0,
        DailyHistoricalDataModel.volumeto > 0),
        ).with_entities(DailyHistoricalDataModel.time,
            DailyHistoricalDataModel.coin_symbol,
            DailyHistoricalDataModel.currency_symbol,
            func.avg(DailyHistoricalDataModel.open),
            func.avg(DailyHistoricalDataModel.high),
            func.avg(DailyHistoricalDataModel.low),
            func.avg(DailyHistoricalDataModel.close),
            func.avg(DailyHistoricalDataModel.volumefrom),
            func.avg(DailyHistoricalDataModel.volumeto)
            ).group_by(DailyHistoricalDataModel.time,
                DailyHistoricalDataModel.coin_symbol,
                DailyHistoricalDataModel.currency_symbol).all()
```

Ova metoda sadrži upit koji izračunava prosječne OHLCV vrijednosti svih burzi kriptovaluta za prosljeđenu kriptovalutu i valutu. Dakle, upit na temelju dostupnih povijesnih OHLCV podataka izračuna prosjek za svaki trenutak u vremenu. U obzir uzima samo unose čije su OHLCV vrijednosti veće od 0. Razlog tome stoji u činjenici da ne podržavaju sve burze iste parove kriptovaluta i valuta ili nemaju dovoljno dugu povijest cijena kao neke druge burze kriptovaluta. Na ovaj način se ignoriraju kako ne bi utjecale na izračun prosjeka ostalih burzi kriptovaluta koje sadrže cijene za par kriptovalute i valute u određenim vremenskim intervalima.

5.1.5. Kontroleri *(Controllers)*

Kontroleri su ulazna točka projekta. Nasljeđuju repozitorije i apstrahiraju njihove metode. Isto kao repozitoriji i modeli, grupirani su po direktorijima kategorija krajnjih točaka. Svaki kontroler sadrži metode nasljeđenog repozitorija. Primjer jednog kontrolera iz projekta u nastavku:

```
class DailyHistoricalDataController(DailyHistoricalDataRepository):

    def __init__(self):
        super().__init__()

    def insert_action(self, daily_historical_data_model):
        return self.insert(daily_historical_data_model)

    def update_action(self):
        return self.update()

    def delete_action(self, id):
```

```
        return self.delete(id)

    def find_id_action(self, id):
        return self.find_id(id)

    def find_exact_price_action(self, time, coin_symbol, currency_symbol, exchange):
        return self.find_exact_price(time, coin_symbol, currency_symbol, exchange)

    def find_coin_symbol_action(self, coin_symbol):
        return self.find_coin_symbol(coin_symbol)

    def find_min_ts_action(self, coin_symbol, currency_symbol, exchange):
        return self.find_min_ts(coin_symbol, currency_symbol, exchange)

    def find_max_ts_action(self, coin_symbol, currency_symbol, exchange):
        return self.find_max_ts(coin_symbol, currency_symbol, exchange)

    def find_averages_action(self, coin_symbol, currency_symbol):
        return self.find_averages(coin_symbol, currency_symbol)
```

5.1.6. Populatori *(Populators)*

Direktorij s imenom *"populators"* sadrži nekoliko klasa čija je glavna svrha populiranje baze podataka za tek kreiranu bazu podataka, odnosno inicijalno pokretanje projekta. Svaka klasa sadrži niz metoda koje odgovaraju krajnjim točkama API-ja u odnosu jedan na jedan. Svaka metoda automatski populira jednu tablicu u bazi podacima krajnje točke, od prvog povijesnog podatka sve do posljednjeg dostupnog podatka. Trenutno su podržani populatori za sve krajnje točke povijesnih OHLCV podataka, društvenih podataka te izračunavanje i populiranje tehničkih indikatora, neovisno o promatranom vremenskom intervalu.

5.1.6.1. Povijesni OHLCV podaci

Primjer metode za populiranje dnevnih povijesnih OHLCV podataka:

```
def populate_daily_historical_ohlcv(self, coin_symbols, currency_symbols, exchanges)
    :
    """
    Fetch all daily historical data from API and save to database.
    :return:
    """
    print("Populating daily historical data...")
    for coin_symbol in coin_symbols:
        for currency_symbol in currency_symbols:
            for exchange in exchanges:
            data = []
            print("*DAY* Coin: {}, Currency: {}, Exchange: {}".format(coin_symbol,
                currency_symbol, exchange))
```

```
            for row in self.fetch_historical_daily_ohlcv(coin_symbol,
                currency_symbol, exchange[0], all_data=True):
            row["coin_symbol"] = coin_symbol
            row["currency_symbol"] = currency_symbol
            row["exchange"] = exchange[0]
            data_obj = DailyHistoricalDataModel(**row)
            if not data_obj.find_exact_price(data_obj.time, coin_symbol,
                currency_symbol, exchange[0]):
                data_obj.insert()
            data.append(row)
```

Metoda prima tri ključna parametra: listu simbola kriptovaluta, listu simbola valuta te listu burzi kriptovaluta. Potom prolazi kroz tri iteracije, jednu za svaku listu parametara. Nakon što iz svake liste uzme jedan element, preuzima sve podatke krajnje točke API-ja za dnevne povijesne OHLCV podatke, prosljeđujući pritom pojedine elemente svake liste. Primljenom nizu podataka proširuje pojedine elemente listi (simbol kriptovalute, simbol valute i burzu kriptovaluta) te ih, ukoliko takav unos već ne postoji u bazi, sprema u bazu podataka. Nakon toga, iterira se sljedeći element u listi i taj se proces ponavlja sve dok podaci nisu pohranjeni za sve permutacije zadanih simbola kriptovaluta, simbola valuta i burzi kriptovaluta. Na istom principu djeluje metoda za druge vremenske intervale.

5.1.6.2. Prosječni povijesni OHLCV podaci

Nakon što se populiraju dnevni povijesni OHLCV podaci, u pravilu se poziva metoda za izračunavanje OHLCV prosjeka svih burzi kriptovaluta za zadane simbole kriptovaluta i simbole valuta:

```
def populate_daily_average_historical_ohlcv(self):
    """
    Calculate average daily historical data from stored historical daily ohlcv and
        save result to database.
    :return:
    """
    print("Populating average historical data...")
    results = HourlyHistoricalDataModel.find_averages()
    items = [dict(zip(DailyAverageHistoricalDataModel.column, row)) for row in
        results]
    for row in items:
        print("*AVG* Time: {}, Coin: {}, Currency: {}".format(row['time'], row['
            coin_symbol'], row['currency_symbol']))
        data_obj = DailyAverageHistoricalDataModel(**row)
        if not data_obj.find_exact_price(data_obj.time, data_obj.coin_symbol,
            data_obj.currency_symbol):
            data_obj.insert()
```

Ova metoda populira prosječne dnevne povijesne OHLCV podatke na osnovi dnevnih povijesnih OHLCV podataka pohranjenih u bazi podataka. Na razini modela već postoji upit koji računa prosječne podatke. Dobivenim podacima potrebno je dodati zaglavlja, nakon čega se, ukoliko takav unos već ne postoji u tablici, spremaju u vlastitu tablicu s prosječnim OHLCV cijenama.

Na istom principu djeluje metoda za druge vremenske intervale.

5.1.6.3. Povijesni volumeni burzi kritpovaluta

Slično prvoj metodi, provodi se populiranje povijesnih dnevnih volumena burzi kriptovaluta:

```
def populate_daily_historical_exchange_volume(self, currency_symbols, exchanges):
    """
    Fetch all daily historical exchange volume data from API and save to database.
    :return:
    """
    print("Populating daily historical exchange volume data...")
    for currency_symbol in currency_symbols:
        for exchange in exchanges:
            data = []
            print("*DAY* Currency: {}, Exchange: {}".format(currency_symbol,
                exchange))
            for row in self.fetch_historical_daily_exchange_volume(currency_symbol,
                exchange[0]):
            row["currency_symbol"] = currency_symbol
            row["exchange"] = exchange[0]
            data_obj = DailyHistoricalExchangeVolumeModel(**row)
            if not data_obj.find_exact_record(data_obj.time, currency_symbol,
                exchange[0]):
                data_obj.insert()
            data.append(row)
```

Metoda prima dva ključna parametra: listu simbola kriptovaluta te listu burzi kriptovaluta. Potom prolazi kroz dvije iteracije, jednu za svaku listu parametara. Nakon što iz svake liste uzme jedan element, preuzima sve podatke krajnje točke API-ja za dnevne povijesne volumene burzi kriptovaluta, prosljeđujući pritom pojedine elemente svake liste. Primljenom nizu podataka proširuje pojedine elemente listi (simbol kriptovalute i burzu kriptovaluta) te ih, ukoliko takav unos već ne postoji u bazi, sprema u bazu podataka. Nakon toga, iterira se sljedeći element u listi i taj se proces ponavlja sve dok podaci nisu pohranjeni za sve permutacije zadanih simbola kriptovaluta, simbola valuta i burzi kriptovaluta. Na istom principu djeluje metoda za vremenski interval od jednog sata.

5.1.6.4. Povijesni duštveni podaci

Populiranje društvenih podataka vrši se na sljedeći način:

```
def populate_historical_day_social_stats_data(self, coin_id_list):
    """
    Fetch all available daily historical social data from API and save to database.
    :return:
    """
    for coin_id in coin_id_list:
        data = []
```

```
        print("*DAY SOCAL STATS* Coin id: {} ".format(coin_id))
        for row in self.fetch_historical_day_social_stats_data(coin_id):
            row["coin_id"] = coin_id
            data_obj = DailyHistoricalSocialStatsModel(**row)
            if not data_obj.find_exact_record(data_obj.time, coin_id):
                data_obj.insert()
            data.append(row)
```

Metoda za populiranje dnevnih društvenih podataka prima samo jedan parametar: listu id-jeva kriptovaluta za koje želimo populirati društvene podatke. Id kriptovalute može se dohvatiti putem krajnje točke API-ja generalnih podataka za listu svih kriptovaluta ili iz baze podataka ako su prethodno populirani generalni podaci. Metoda iterira sve id-jeve kriptovaluta, preuzima podatke krajnje točke API-ja za povijesne društvene podatke te ih, ukoliko već ne postoje u ciljanoj tablici, sprema u bazu podataka. Proces se ponavlja sve dok ne prođe listu zadanih kriptovaluta. Na istom principu djeluju metode za populiranje povijesnih društvenih podataka u drugim vremenskim intervalima.

5.1.6.5. Podaci tehničkih indikatora

Preostaje primjer metode populiranja tehničkih indikatora:

```
def populate_volume_indicators(self, coin_symbol, currency_symbol, data):
    """
    Calculate indicators and store results to database.
    :param coin_symbol:
    :param currency_symbol:
    :param data:
    :return:
    """
    results = self.calculate_all_volume_indicators(coin_symbol, currency_symbol,
        data)
    for row in results:
        data_obj = VolumeIndicatorsModel(**row)
        if not data_obj.find_exact_price(data_obj.time, data_obj.coin_symbol,
            data_obj.currency_symbol):
            data_obj.insert()
```

Metoda prima tri parametra: simbol kriptovalute, simbol valute te OHLCV podatke. U navedenom primjeru, na temelju prosljeđenih podataka istovremeno se računaju svi indikatori volumena, pri čemu svaka metoda za izračun indikatora volumena uzima dio sebi potrebnih podataka za računanje svakog pojedinog indikatora. S obzirom da se svi volumeni indikatora nalaze u jednoj tablici, nakon što se indikatori izračunaju, kolektivno se pohranjuju u bazu podataka.

5.1.7. Loggeri *(Loggers)*

Direktorij s imenom *"loggers"* zamišljen je kao produžetak populatora. S obzirom da populatori iteriraju vrijednosti od prvog povijesnog zapisa sve do posljednjeg dostupnog zapisa, inicijalna svrha loggera bila je pohranjivanje jedne vrijednosti i ažurno održavanje sustava (eng.

up-to-date). Inkrementalnim poboljšanjima rada, svrhu loggera interno su zamijenili agenti, stoga se trenutno ne koriste u sustavu. No loggeri će svakako u budućnosti ostati dio sustava u malo drugačijem obliku s posebno dediciranim agentima koji će koristiti njihovu logiku.

5.1.7.1. Podaci o kriptovalutama

Pohranjivanje pojedinog detaljnog podatka o kriptovalutama provodi se na sljedeći način:

```
def save_coin_list(data):
    """
    Save coin list data to database.
    :param data:
    """
    log = {}
    for row in data.values():
        for k, v in row.items():
            log[DataFormat.convert_lower_underscore(k)] = v
        data_obj = CoinListModel(**log)
        data_obj.insert()
```

Metoda prima samo jedan parametar, a to su podaci s krajnje točke API-ja generalnih podataka za listu svih kriptovaluta. Iterira listu primljenih vrijednosti, formatira ih u prikladan oblik te pohranjuje u bazu podataka.

5.1.7.2. OHLCV podatak

Pohranjivanje pojedinih povijesnih OHLCV podataka vrši se pomoću sljedeće metode:

```
def save_daily_historical_ohlcv(self, coin_symbol, currency_symbol, exchange, data):
    """
    Save daily historical_data record to database.
    :param coin_symbol:
    :param currency_symbol:
    :param exchange:
    :param data:
    :return:
    """
    for row in data:
        row["exchange"] = exchange
        row["coin_symbol"] = coin_symbol
        row["currency_symbol"] = currency_symbol
        data_obj = HourlyHistoricalDataModel(**row)
        data_obj.insert()
```

Metoda standardno prima tri parametra: simbol kriptovalute, simbol valute, burzu kriptovalute te set podataka vezan uz prethodne parametre. S obzirom da među podacima ne postoje informacije o kriptovaluti, valuti, a niti burzi kriptovalute, oni se ručno pridružuju podacima te se potom spremaju u bazu podataka.

5.1.7.3. Volumen burze kriptovalute

Pohranjivanje pojedinog volumena burze kriptovalute izvršava se pomoću sljedeće metode:

```
def save_daily_historical_exchange_volume(self, currency_symbol, exchange, data):
    """
    Save daily historical exchange volume to database.
    :param currency_symbol:
    :param exchange:
    :param data:
    """
    for row in data:
        row["exchange"] = exchange
        row["currency_symbol"] = currency_symbol
        data_obj = DailyHistoricalExchangeVolumeModel(**row)
        data_obj.insert()
```

Metoda prima tri parametra: simbol kriptovalute, burzu kriptovalute te set podataka vezan uz prethodne parametre. S obzirom da među podacima ne postoje informacije o kriptovaluti, a niti burzi kriptovalute, oni se ručno pridružuju podacima te se potom spremaju u bazu podataka.

5.1.7.4. Društveni podatak

Pohranjivanje pojedinog volumena burze kriptovalute izvršava se pomoću sljedeće metode:

```
def save_daily_historical_social_stats(self, coin_id, data):
    """
    Save daily historical_data record to database.
    :param coin_id:
    :param data:
    :return:
    """
    for row in data:
        row["coin_id"] = coin_id
        data_obj = DailyHistoricalSocialStatsModel(**row)
        data_obj.insert()
```

Metoda prima dva parametra: id kriptovalute te set podataka vezan uz istu. S obzirom da u prosljeđenim podacima ne postoji id kriptovalute, on se ručno pridružuje podacima te se potom sprema u bazu podataka.

5.1.8. Kalkulatori *(Calculators)*

Direktorij s imenom *"calculators"* sadrži niz pomoćnih klasa za izračunavanje tehničkih indikatora. Tehnički indikatori računaju se na temelju jednog ili serije OHLCV podataka, ovisno o vrsti indikatora. Kao pomoć u izračunavanju korištena je biblioteka *talib*. *TA-Lib* jest omot

(eng. *wrapper*) Python programskog jezika baziran na Cython-u umjesto SWIG-u. *TA-Lib* je na široko korišten za provođenje tehničke analize podataka financijskih tržišta. Biblioteka sadrži niz metoda koje izračunavaju tehničke indikatore na osnovi proslijeđenih OHLCV parametara. U projektu je iz prikupljenih povijesnih OHLCV podataka za parove kriptovalute, burze kriptovalute i valute izračunat trend 169 različitih indikatora: [**55**]

- 17 indikatora **funkcije preklapanja** (eng. *overlap studies functions*),
- 40 **indikatora trenda** (eng. *momentum indicators*),
- 3 **indikatora volumena** (eng. *volume indicators*),
- 3 **indikatora promjenjivosti** (eng. *volatility indicators*),
- 60 indikatora **funkcije prepoznavanja obrazaca** (eng. *patter recognition functions*),
- 7 indikatora **cikličkih indikatorskih funkcija** (eng. *cycle indicator functions*),
- 9 indikatora**statističkih funkcija** (eng. *statistic functions*),
- 4 indikatora **funkcije promjene cijene** (eng. *price transform functions*),
- 15 indikatora **funkcije matematičke promjene** (eng. *math transform functions*),
- 11 indikatora **funkcije matematičkih operatora** (eng. *math operator functions*)

U nastavku slijedi nekoliko grupiranih primjera izračunavanja različitih kategorija tehničkih indikatora.

```
# BBANDS - Bollinger Bands
bbands_upperband, bbands_middleband, bbands_lowerband = BBANDS(inputs['close'],
    timeperiod=5, nbdevup=2, nbdevdn=2, matype=0)

# MACD - Moving Average Convergence/Divergence
macd, macdsignal, macdhist = MACD(inputs['close'], fastperiod=12, slowperiod=26,
    signalperiod=9)

# OBV - On Balance Volume
obv_real = OBV(inputs['close'], inputs['volume'])

# CDLLONGLINE - Long Line Candle
cdllongline_integer = CDLLONGLINE(inputs['open'], inputs['high'], inputs['low'],
    inputs['close'])

# HT_DCPERIOD - Hilbert Transform - Dominant Cycle Period
ht_dcperiod_real = HT_DCPERIOD(inputs['close'])

# STDDEV - Standard Deviation
stddev_real = STDDEV(inputs['close'], timeperiod=5, nbdev=1)
```

U primjeru su nasumično odabrani tehnički indikatori iz različitih kategorija. Pomoću *talib* biblioteke, svaki tehnički indikator izračunava se uz pomoć jedne metode, kojoj se prosljeđuju potrebni parametri. Neke metode dopuštanju dodatna "naštimavanja" preciznosti, broja vremenskih intervala razmatranih za izračunavanje, agregiranja i tako dalje. U projektu su za sve prikupljene OHLCV povijesne podatke u vremenskim intervalima minute, sata i dana, izračunati tehnički indikatori, čime su povijesnom trendu kretanja cijena pridružene vrijednosti tehničkih indikatora za promatrani vremenski period.

5.1.9. Pomoćne klase *(Helpers)*

U direktoriju *"helpers"* postoji nekoliko pomoćnih klasa koje uglavnom sadrže metode za oblikovanje podataka, dohvaćanje API ključa i slične univerzalne radnje koje se koriste kroz cijeli projekt. Glavna svrha je održavanje konzistentnosti arhitekture koda te jedne točke pristupa za pomoćne metode. Primjeri nekoliko metoda u nastavku.

```
def convert_lower_underscore(name):
    """
    Convert camelcase to underscore and lower all characters
    :param name:
    :return:
    """
    s1 = re.sub('(.)([A-Z][a-z]+)', r'\1_\2', name)
    return re.sub('([a-z0-9])([A-Z])', r'\1_\2', s1).lower()
```

Priložena metoda za ulazni niz znakova postavlja donju povlaku ispred velikog slova te potom sve pretvara u mala slova. Metoda se uglavnom koristi za formatiranje *CamelCase* formata pisanja u *snake_case*.

```
def zip_data(keys, values):
    """
    Merge keys with corresponding values.
    :param keys:
    :param values:
    :return:
    """
    values = [[float(elem) for elem in item] for item in np.nditer(values)]
    return [dict(zip(keys, row)) for row in values]
```

Gore navedena metoda uglavnom se koristi nakon izračunavanja tehničkih indikatora jer metode za izračun istih zahtijevaju poseban format podataka. U suštini, spaja niz zaglavlja s pripadajućim vrijednostima natrag u json format pogodan za daljnji rad. U nastavku je priložena dulja inačica metode s istom svrhom, češće korištena zbog memorijske utilizacije i performansa sustava:

```
def zip_large_data(keys, values, data):
    """
    Merge keys with corresponding values for larger quantities of data.
    :param keys:
    :param values:
    :param data:
    :return:
    """
    json = []
    for i, item in enumerate(zip(*values)):
        values = [data[i]['time'], data[i]['coin_symbol'], data[i]['currency_symbol'
            ]]
        for elem in item:
            values.append(float(elem))
        json.append(dict(zip(keys, values)))
    return json
```

5.2. Višeagentni sustav

Za Python 2.x verzije postoji biblioteka SPADE (Smart Python Agent Development Environment) koja pruža platformu za razvijanje višeagentnih sustava. Omogućuje instantnu razmjenu poruka, uvid u stanje pojedinih agenata u stvarnom vremenu, asinkronu komunikaciju, agentni model utemeljen na ponašanjima, sučelje na webu, podršku sa XMPP serverom i mnoštvo drugih. [**50**] Nažalost, za vrijeme pisanja rada SPADE nije bio dostupan za Python 3.x verzije, zbog čega se višeagentni sustav morao manualno improvizirati od temelja. Cilj je bio simulirati rad većeg broja agenata određene uloge kako bi se kolektivno ispunila zajednička svrha. U svom djelovanju, svaki je tip agenta nezamijenjivi dio cjeline. Ispunjavaći predefiniranu ulogu u sustavu, agenti provode skup radnji kojima u cjelini samostalno rukovode i održavaju sustav te time osiguravaju njegov integritet. Sustav se u cjelini samostalno koordinira: kreira bazu podataka s tablicama i atributima potrebnim za neometano djelovanje sustava, populira bazu podataka od najudaljenije dostupne točke u povijesti i prikuplja podatke različitih kategorija prema željenim parametrima i pohranjuje ih u bazu podataka, održava bazu podataka ažurnom sukladno najnovijim dostupnim podacima u stvarnom vremenu, vrši izračune iz dostupnih podataka te u bazu podataka pohranjuje parametre koji inicijalno nisu dostupni u prikupljenim podacima, dohvaća opće informacije integriranih vanjskih sustava, prati promjene na tržištu kriptovaluta i propagira ih unutar sustava te trguje u skladu s definiranim generalnim i specifičnim strategijama trgovanja.

Višeagentni sustav organiziran je hijerarhijski na tri razine, gdje na prvoj razini postoji jedan glavni agent (koordinator) na čelu jednog sustava agenata. Koordinator rukovodi tri agenta na sljedećoj razini hijerarhije: glavnog tehničkog agenta (CTO) koji rukovodi tehničku skupu agenata, glavnog izvršnog agenta (CEO) koji rukovodi izvršnu skupu agenata te glavnog financijskog agenta (CFO) koji rukovodi financijsku skupinu agenata. Dakle, rukovođenje agenata propagira se hijerarhijski, dok je komunikacija među agentima istih ili različitih razina na principu modela objave i pretplate (eng. *publish and subscribe*). Primjerice, agent zadužen za dohvaćanje najnovijih cijena uređene trojke kriptovalute, valute i burze kriptovalute propagirat

će cijenu cijelom sustavu, a na njegove obavijesti pretplatit će se samo oni agenti koji djeluju s navedenom uređenom trojkom. Nadalje, svaki agent nasljeđuje klase podržane u projektnom dijelu rada, čime mu se omogućuje pristup nizu funkcionalnosti sustava, od poziva krajnjih točaka pa sve do rada s bazom podataka. Na taj način agenti sadrže isključivo logiku uloge koju izvršavaju, a za sve vanjske radnje oslanjaju se na postojeće metode. Svaki agent na najnižoj razini sustava nasljeđuje isključivo jednu kategoriju krajnjih točaka te jedan kontroler, ovisno o podacima nad kojima agent djeluje, čime imaju pristup svim metodama potrebnim za ispunjavanje svoje specifične svrhe u sustavu. U suštini, agent predstavlja vanjskog korisnika koji pokreće neku radnju projektnog dijela sustava.

Višeagentni sustav je u svojoj svrsi ujedno i distribuiran, a moguće ga je jednostavno dalje skalirati. Naime, svaki koordinator može pokrenuti proizvoljno mnogo CEO-a, CTO-a i CFO-a, a svaki od njih može pokrenuti proizvoljno mnogo sebi podređenih agenata. Pokrenuti agenti mogu biti i istog tipa s različitim prosljeđenim parametrima, čime pokrivaju različitu ulogu u sustavu. Svaki agent vrti se u vlastitoj dretvi, čime su mu osigurani resursi potrebni za rad. Zaustavljanjem agenta, gasi se i njegova dretva. S obzirom na prirodu implementacije agenata, postoji nekoliko zajedničkih metoda koje se dijele među svim agentima sustava. Svaki agent sadrži konstruktor kojemu se prosljeđuju parametri specifični isključivo toj instanci agenta. S obzirom da je svaki agent dretva, sadrže osnovnu metodu (*run*) koja predstavlja glavninu logike njihova djelovanja:

```
def run(self):
    while not self._stop_event.is_set():
        #logika čspecifine svrhe agenta
```

Agent će izvršavati svoju ulogu sve do prekida njegova rada. Provjeravanje ili zaustavljanje rada agenata vrši se pomoću zastavica:

```
def stop(self):
    self._stop_event.set()

def stopped(self):
    return self._stop_event.is_set()
```

U prvom slučaju, postavlja se zastavica koja označava kraj rada agenta, čime prestaje njegova funkcija i gasi se dretva. Druga metoda služi za provjeravanje aktivnog stanja agenta, izvršava li još uvijek svoju funkciju ili je završio s radom. U nastavku će se opisati pojedini agenti koji djeluju u sklopu višeagentnog sustava zajedno sa sebi specifičnim metodama, međusobnim interakcijama te ulogama koje izvršavaju unutar sustava. Struktura direktorija agenata izgleda ovako:

Slika 50: Struktura višeagentnog sustava

5.2.1. Koordinator *(Coordinator)*

Glava sustava i agent na najvišem stupnju hijerarhije jest koordinator *coordinator*. Koordinator je ulazna točka sustava čije pokretanje predstavlja početak rada cjelokupnog sustava. Pokreće, rukovodi, nadzire i zaustavlja tri agenta niže razine hijerarhije, a to su *CTO*, *CEO* i *CFO*. Omogućuje pokretanje proizvoljno mnogo agenata na nižoj razini hijerarhije. Isto tako, može se pokrenuti proizvoljno mnogo koordinatora, koji imaju proizvoljno mnogo agenata na nižoj razini hijerarhije, ukoliko za time ima potrebe. Svaki od njih dobit će potrebne resurse za rad i izvršavanje svoje uloge u sustavu. U slučaju višekorisničke potpore, zamisao je da jedan koordinator bude jedinstveno identificiran s jednim korisnikom. Korisnik putem dodjeljenog koordinatora može voditi vlastiti sustav agenata i ostvariti željeni cilj sustava te pritom analizirati provedene radnje u sustavu. Postojeći koordinator predstavlja simulaciju rada višeagentnog sustava. Implementacijom vanjske podrške rukovođenju sustava, otvara se potpuno novi spektar mogućnosti. Pokretanjem koordinatora inicijaliziraju se liste i brojači za praćenje agenata niže razine hijerarhije. Liste sadrže sve potrebne podatke o pokrenutim agentima u obliku uređenih parova, a brojači služe za evidenciju broja pokrenutih agenata svakog tipa i eventualno dodjeljivanje slobodnog id-a novo inicijaliziranim agentima.

```
def __init__(self):
    self.active_CEO_agents = {}
    self.active_CTO_agents = {}
```

```
    self.active_CFO_agents = {}
    self.ceo_agent_count = 0
    self.cto_agent_count = 0
    self.cfo_agent_count = 0
```

Nakon inicijalizacije, u glavnoj dretvi koordinatora pokreću se i zaustavljaju agenti niže razine hijerarhije s pripadajućim metodama.

```
def start_CEO_agent(self, agent_id, agent_name):
    agent = CEO(agent_id, agent_name)
    agent.start()
    print("CEO {}.{} started.".format(agent_id, agent_name))
    self.active_CEO_agents[agent_id] = [agent_name, agent]
    self.ceo_agent_count += 1
```

U gornjem isječku koda naveden je primjer pokretanja CEO agenta. Za pokretanje agenta potrebno mu je dodijeliti jedinstveni identifikator *agent_id* te proizvoljan naziv *agent_name*. U praksi, jedinstveni identifikator može se proslijediti iz brojača tog tipa agenta. Potom se inicijalizira i pokreće dretva željenog agenta, o čemu se obavještava korisnika. Pokretanjem novog agenta, u listu aktivnih agenata tog tipa dodaje se uređeni par s ključem jedinstvenog identifikatora agenta te vrijednostima imena i instance objekta, što nam kasnije omogućuje čisto zaustavljanje agenta. Povećavanjem brojača aktivnih agenata tog tipa završava pokretanje novog agenta. Na istom principu pokreće se i CTO agent, ali CFO ima malu različitost:

```
def start_CFO_agent(self, agent_id, agent_name, exchange):
    agent = CFO(agent_id, agent_name, exchange)
    agent.start()
    print("CFO {}.{}[{}] started.".format(agent_id, agent_name, exchange))
    self.active_CFO_agents[agent_id] = [agent_name, exchange, agent]
    self.ceo_agent_count += 1
```

Naime, financijski agenti mogu djelovati na točno specificiranim burzama kriptovaluta, pa se tako i CFO agentima može dodijeliti dodatan parametar za ciljanu burzu kriptovaluta. Osim toga, logika pokretanja agenta ostaje ista kao u prethodno opisanom primjeru. Tijekom rada koordinatora, može se dobiti uvid u broj trenutno aktivnih agenata:

```
def get_active_cfo_agents(self):
    return self.active_CFO_agents

def get_active_cto_agents(self):
    return self.active_CTO_agents

def get_active_ceo_agents(self):
    return self.active_CEO_agents
```

Metode samo vraćaju vrijednost brojača aktivnih agenata određenog tipa. Do sada smo o pokretanju i evidenciji aktivnih agenata. Zaustavljanje agenata može se izvršiti na dva različita načina:

```
def stop_CEO_agent(self, agent_id=None, agent_name=None):
    for k, v in list(self.active_CEO_agents.items()):
        if k == agent_id or v[0] == agent_name:
            v[1].stop()
            v[1].join()
            if v[1].stopped:
                self.active_CEO_agents.pop(k)
                print("CEO {}.{} stopped.".format(k, v[0]))
```

Metodi za zaustavljanje se mogu proslijediti jedinstveni identifikator agenta, njegov naziv ili oboje. Ukoliko neki od prosljeđenih parametara odgovara nekom od identifikatora u listi agenata, dretva će se zaustaviti te time prekinuti rad željenog agenta i svih njegovih podanika. Zaustavljanjem agenta, on se briše iz liste aktivnih agenata, čime završava ova radnja. Ukoliko se metodi ne proslijedi ništa ili parametar koji ne postoji u listi aktivnih agenata, ništa se neće dogoditi. Situacija je ponovno ista za CTO agenta, dok CFO ima malu različitost:

```
def stop_CFO_agent(self, agent_id=None, agent_name=None, exchange=None):
    for k, v in list(self.active_CFO_agents.items()):
    if k == agent_id or v[0] == agent_name or v[1] == exchange:
        v[2].stop()
        v[2].join()
        if v[2].stopped:
            self.active_CFO_agents.pop(k)
            print("CFO {}.{}[{}] stopped.".format(k, v[0], v[1]))
```

U slučaju CFO agenta, može se proslijediti naziv burze, čime će se zaustaviti rad svih CFO agenata koji djeluju na toj burzi.

5.2.2. Tehnički agenti *(technical)*

Generalno, tehnički agenti zaduženi su za sve radnje vezane uz održavanje sustava i rad s bazom podataka. U trenutnoj verziji sustava, tehnički agenti pohranjuju odluke sustava, populiraju bazu podataka i zapisuju pojedine vrijednosti. Svaki od agenata izvršava točno određenu ulogu, što ga čini nezamijenjivim dijelom sustava.

5.2.2.1. CTO

CTO je glavni tehnički agent koji pokreće, rukovodi, nadzire i zaustavlja sve tehničke agente. Glava je tehničkog dijela sustava, a direktno je podređen koordinatoru koji može provjeravati njegovo aktivno stanje. U slučaju da koordinator odluči obustaviti rad CTO-a, CTO će

prvo zaustaviti sve podređene agente te tek onda prekinuti s radom. Jedan CTO agent može rukovoditi s proizvoljno mnogo tehničkih agenata, neovisno o kategoriji i ulozi tehničkog agenta. Svakom tehničkom agentu može proslijediti proizvoljne parametre, kao što su parovi određenih kriptovaluta, valuta i burzi kriptovaluta nad kojima će podređeni tehnički agent djelovati. Na taj način CTO definira specifičnu ulogu podređenog tehničkog agenta u sustavu. U trenutnoj inačici sustava, kategorije agenata u domeni kontrole CTO-a su:

- populatori općih podataka (detaljna lista kriptovaluta),
- populatori povijesnih podataka (OHLCV cijena, prosječnih OHLCV cijena, volumena burzi kriptovaluta) za sve vremenske intervale (minuta, sat i dan),
- populatori tehničkih indikatora (svi do sad navedeni u projektu),
- populatori društvenih podataka za sve vremenske intervale,
- loggeri općih podataka,
- loggeri povijesnih podataka

Kako bi se inicijalizirao CTO agent, potrebno ga je jednoznačno identificirati id-jem i proizvoljnim nazivom:

```
def __init__(self, agent_id, agent_name):
    threading.Thread.__init__(self)
    self.agent_id = agent_id
    self.agent_name = agent_name

    self.active_general_info_agents = {}
    self.active_historical_data_agents = {}
    self.active_average_historical_data_agents = {}
    self.active_historical_exchange_volume_agents = {}
    self.active_historical_social_stats_agents = {}
    self.tech_agent_count = 0

    self._stop_event = threading.Event()
```

Pokretanjem CTO-a, instanciraju se prazne liste koje sadrže evidenciju svih pokrenutih tehničkih agenata te brojač koji prati količinu podređenih agenata jednog CTO-a. Pokretanje tehničkih agenata vrši se pomoću generičkih metoda, pri čemu svaka kategorija agenata ima vlastitu metodu. Primjerice, pokretanje tehničkog agenta koji radi s povijesnim OHLCV podacima izvršava se pomoću sljedeće metode:

```
def start_historical_data_agent(self, agent_id, agent_name, agent_type, coin_symbols
    , currency_symbols, exchanges):
    agent = agent_type(agent_id, agent_name, agent_type.__name__, coin_symbols,
        currency_symbols, exchanges)
```

```
    agent.start()
    print("[Technical] {}_{}({}) started.".format(agent_id, agent_name, agent_type.
        __name__))
    self.active_historical_data_agents[agent_id] = [agent_name, agent_type.__name__,
         agent]
    self.tech_agent_count += 1
```

Kako bi pokrenuo tehničkog agenta koji radi s povijesnim OHLCV podacima, potrebno ga je jednoznačno identificirati id-jem, proizvoljnim nazivom, tipom agenta te mu proslijediti listu simbola kriptovaluta, simbola valuta te burzi kriptovaluta nad kojima će agent djelovati. Na ovaj način CTO može podijeliti zadatke između većeg broja agenata istog tipa. Pokretanjem novog tehničkog agenta, njegovi podaci se dodaju u listu kategorije agenata kojoj pripada te se povećava brojač aktivnih tehničkih agenata. Metode za pokretanje različitih kategorija tehničkih agenata uglavnom variraju u prosljeđenim parametrima specifičnim za tu kategoriju tehničkih agenata. Pokretanje agenta uz pomoć gore navedene metode izgledalo bi ovako:

```
self.start_historical_data_agent(self.tech_agent_count, "Daily historical ohlcv",
    DailyHistoricalDataPopulatorAgent, crypto_list, currency_symbols, exchanges)
```

Dakle, pokreće se tehnički agent u kategoriji populatora povijesnih podataka s jedinstvenim id-jem i proizvoljnim nazivom s parametrima predefinirane liste kriptovaluta, valuta i burzi kriptovaluta nad kojima će djelovati. Metode za zaustavljanje tehničkih agenata također su veoma slične i variraju ovisno o parametrima koji se prosljeđuju toj kategoriji tehničkih agenata. Na našem primjeru, populator povijesnih podataka može se zaustaviti na tri načina pomoću sljedeće metode:

```
def stop_historical_data_agent(self, agent_id=None, agent_name=None, agent_type=None
    ):
    for k, v in list(self.active_historical_data_agents.items()):
        if k == agent_id or v[0] == agent_name or v[1] == agent_type:
            v[2].stop()
            v[2].join()
            if v[2].stopped:
                self.active_historical_data_agents.pop(k)
                print("[Technical] {}_{}({}) stopped.".format(k, v[0], v[1]))
```

Metodi se može proslijediti jedan do tri različita parametra, a to su:

- id agenta, čime se zaustavlja s radom tog specifičnog agenta,
- naziv agenta, čime se zaustavlja s radom svih agenata koji imaju isto ime,
- tip agenta, čime se zaustavljaju svi agenti istog tipa

Zaustavljanjem agenta, on se briše iz liste aktivnih agenata te sustav dalje nastavlja s radom. Broj i detalji tehničkih agenata određene kategorije podređenih specifičnom CTO-u mogu se dobiti pomoću metoda koje vraćaju stanje listi i brojača, čime koordinator može dobiti uvid u djelovanje određenog CTO agenta. Primjer nekoliko metoda za dohvaćanje detalja:

```
def get_active_historical_social_stats_agents(self):
    return self.active_historical_social_stats_agents

def get_active_active_technical_agents_count(self):
    return self.tech_agent_count
```

Prva metoda vraća detalje svih aktivnih povijesnih agenata, a druga ukupni broj tehničkih agenata. U slučaju da dođe do gašenja CTO agenta, prije vlastitog zaustavljanja terminirati će rad svih podređenih tehničkih agenata na sljedeći način:

```
def stop(self):
    for k in list(self.active_general_info_agents):
        self.stop_general_info_agent(k)
    for k in list(self.active_historical_data_agents):
        self.stop_historical_data_agent(k)
    for k in list(self.active_average_historical_data_agents):
        self.stop_historical_data_agent(k)
    for k in list(self.active_historical_exchange_volume_agents):
        self.stop_historical_exchange_volume_agent(k)
    for k in list(self.active_historical_social_stats_agents):
        self.stop_historical_social_stats_agent(k)
    self._stop_event.set()
```

Iteriranjem kroz liste detaljnih kategorija tehničkih agenata zaustavit će se svi podređeni agenti. Koordinator ujedno može provjeriti ukoliko je određeni CTO zaustavljen ili još uvijek radi uz pomoć metode:

```
def stopped(self):
    return self._stop_event.is_set()
```

5.2.2.2. Populatori generalnih podataka

Ova kategorija tehničkih agenata poziva krajnje točke u kategoriji generalnih podataka te nasljeđuje kontrolere u direktoriju generalnih podataka, čime im je omogućen niz metoda za rad s bazom podataka. Specifičnije, populatori generalnih podataka preuzimaju sve dostupne podatke s krajnjih točaka generalnih podataka te ih pohranjuju u bazu podataka. Za vrijeme verzije API-ja za koje je pisan ovaj rad, postojala je samo jedna krajnja točka u kategoriji generalnih podataka koja sadrži detaljan skup podataka vrijedan pohranjivanja i analiziranja. Stoga postoji jedan populator generalnih podataka, a to je populator dostupnih kriptovaluta i njihovih detaljnih informacija. Za pokretanje agenta potrebno je proslijediti id agenta, proizvoljan naziv te model agenta:

```
def __init__(self, agent_id, agent_name, agent_model):

    self.agent_id = agent_id
    self.agent_name = agent_name
    self.agent_model = agent_model

    self._stop_event = threading.Event()
```

Agent sadrži jednu metodu za populiranje liste kriptovaluta, dok su ostale metode ovog agenta klasične su svim ostalim tehničkim agentima. Populiranje liste kriptovaluta vrši se pomoću:

```
def populate_coin_list(self):
    """
    Fetch all coin list data from API and save to database.
    :return:
    """
    log = {}
    for row in self.fetch_all_coins().values():
        for k, v in row.items():
            log[DataFormat.convert_lower_underscore(k)] = v
        print(log)

        data_obj = CoinListModel(**log)

        if self.find_by_id(data_obj.id):
            self.update_action()
        else:
            self.insert_action(data_obj)
```

Za početak se dohavaćaju sve vrijednosti s krajnje točke te se formatiraju na način prihvatljiv za pohranu u bazu podataka. U slučaju da kriptovaluta s postojećim id-jem postoji, ažurirat će se njen sadržaj, a u suprotnom će se pohraniti novi unos u bazu podataka.

5.2.2.3. Populatori povijesnih podataka

Ova kategorija tehničkih agenata poziva krajnje točke u kategoriji povijesnih podataka te nasljeđuje kontrolere u direktoriju povijesnih podataka, čime im je omogućen niz metoda za rad s bazom podataka. Specifičnije, populatori povijesnih podataka preuzimaju sve dostupne podatke s krajnjih točaka povijesnih podataka te ih pohranjuju u bazu podataka. Postoje tri glavne kategorije populatora povijesnih podataka:

- populatori OHLCV cijena u promatranim vremenskim intervalima minute, sata i dana.
- populatori volumena burzi kriptovaluta u promatranim vremenskim intervalima od sata i dana.
- populatori prosječnih OHLCV cijena u promatranim vremenskim intervalima minute, sata i dana.

Populatori OHLCV cijena

Tri su glavna populatora OHLCV cijena. Razlikuju u vremenskim intervalima u kojima djeluju, korištenim krajnjim točkama i kontrolerima. Za inicijalizacija populatora OHLCV cijena potrebno je proslijediti id agenta, proizvoljno ime agenta, tip agenta te parametre koji definiraju njegovu specifičnu ulogu. Parametri su lista kriptovaluta, lista valuta te burze kriptovaluta za čije će međusobne permutacije agent izvršavati populiranje podataka. Proći ćemo kroz primjer populatora dnevnih OLHCV podataka:

```
def __init__(self, agent_id, agent_name, agent_model, coin_symbols, currency_symbols
    , exchanges):
    self.agent_id = agent_id
    self.agent_name = agent_name
    self.agent_model = agent_model
    self.coin_symbols = coin_symbols
    self.currency_symbols = currency_symbols
    self.exchanges = exchanges

    self._stop_event = threading.Event()
```

Pokretanjem agenta, vršit će populiranje i održavati bazu ažurnom sve dok se ne zaustavi. Glavnina rada prikazana je sljedećom metodom:

```
def run(self):
    while not self._stop_event.is_set():

        for coin_symbol in self.coin_symbols:
            for currency_symbol in self.currency_symbols:
                for exchange in self.exchanges:
                    try:
                        timestamp = self.fetch_historical_daily_ohlcv(coin_symbol,
                            currency_symbol, exchange, 1)["TimeFrom"]
                    except:
                        print("[Error] Daily historical data not available for pair
                            - Coin: {}, Currency: {}, Exchange: {}".format(
                            coin_symbol, currency_symbol, exchange))
                        continue

                    print("*Started - populate daily historical data* Coin: {},
                        Currency: {}, Exchange: {}".format(coin_symbol,
                        currency_symbol, exchange))
                    self.populate_daily_historical_ohlcv(coin_symbol,
                        currency_symbol, exchange, timestamp)
                    print("*Finished - populater daily historical data* Coin: {},
                        Currency: {}, Exchange: {}".format(coin_symbol,
                        currency_symbol, exchange))
```

```
        print(self.agent_id, self.agent_name)
        time.sleep(86400)
```

Dakle, populiranje će se iterativno izvršavati sve dok agent ne dobije naredbu da prekine sa svojim radom. Sve dok je agent aktivan, prolazit će kroz permutacije vrijednosti prosljeđenih u listi kriptovaluta, valuta i burzi kriptovaluta. Najprije će s krajnje točke pokušati dohvatiti trenutak u vremenu zadnjeg dostupnog podatka za vremenski interval na kojemu djeluje. U slučaju da on ne postoji za par kriptovalute, valute i burze kriptovalute, prijeći će na sljedeći par. U slučaju da je uspio dohvatiti posljednji trenutak u vremenu, proslijedit će ga vlastitoj metodi za populiranje podataka, zajedno sa simbolom kriptovalute, simbolom valute i burzom kriptovalute za koje je potrebno populirati podatke. Ukoliko agent populira sve podatke za prosljeđene liste parametara, čekat će jedan dan te preuzeti najnovije vrijednosti i na taj način održavati bazu ažurnom. Metoda za populiranje povijesnih dnevnih OHLCV podataka izgleda ovako:

```
def populate_daily_historical_ohlcv(self, coin_symbol, currency_symbol, exchange,
    timestamp):
    """
    Fetch all available daily historical data from API and save to database.
    :return:
    """
    last_ts = self.find_max_ts_action(coin_symbol, currency_symbol, exchange)[0]
    last_ts = last_ts if last_ts is not None else 0
    while True:
        api_data = self.fetch_historical_daily_ohlcv(coin_symbol, currency_symbol,
            exchange, 2000, to_ts=timestamp)
        api_data["Data"] = api_data["Data"][::-1]
        for row in api_data["Data"][1:]:
            row["coin_symbol"] = coin_symbol
            row["currency_symbol"] = currency_symbol
            row["exchange"] = exchange
            data_obj = DailyHistoricalDataModel(**row)
            if int(last_ts) == data_obj.time:
                return
            elif (row["close"] and row["close"] and row["close"] and row["close"])
                is 0:
                return
            else:
                self.insert_action(data_obj)
            timestamp = api_data["TimeFrom"]
```

Prije nego započne s populiranjem, agent će iz baze podataka dohvatiti posljednji trenutak u vremenu za koji postoji zapis u bazi podataka za uređenu trojku kriptovalute, valute i burze kriptovalute. Dohvaćena vremenska oznaka služi kao polazište od kojega treba započeti populiranje podataka. Podatke s krajnje točke potrebno je iterativno preuzimati s obzirom da je maksimalna količina podataka koja se može preuzeti s krajnje točke točno 2000. Krajnjoj točki prosljeđuju se simbol kriptovalute, valute i burze kriptovalute, zajedno s vremenskom oznakom do koje želimo dohvatiti podatke. Dobivenim podacima pridružuju se vrijednosti parametara s

obzirom da se ne nalaze u odgovoru, a u bazi podataka postoji jedna tablica za sve uređene trojke povijesnih OHLCV podataka istog vremenskog intervala. U slučaju da se vremenska oznaka podatka podudara s vremenskom oznakom posljednjeg zapisa u bazi podataka ili više ne postoji podataka za uređenu trojku, populiranje ove uređene trojke se završava te agent nastavlja sa sljedećom trojkom. Princip rada za druge populatore OHLCV cijena koji djeluju u različitim vremenskim intervalima je isti.

Populatori prosječnih OHLCV cijena

Tri su glavna populatora prosječnih OHLCV cijena. Razlikuju u vremenskim intervalima u kojima djeluju i korištenim kontrolerima. Za inicijalizacija populatora prosječnih OHLCV cijena potrebno je proslijediti id agenta, proizvoljno ime agenta, tip agenta te parametre koji definiraju njegovu specifičnu ulogu. Parametri su lista kriptovaluta te lista valuta za čije će međusobne permutacije agent izvršavati populiranje podataka. Proći ćemo kroz primjer populatora prosječnih dnevnih OLHCV podataka:

```
def __init__(self, agent_id, agent_name, agent_model, coin_symbols, currency_symbols
    ):
    self.agent_id = agent_id
    self.agent_name = agent_name
    self.agent_model = agent_model
    self.coin_symbols = coin_symbols
    self.currency_symbols = currency_symbols

    self._stop_event = threading.Event()
```

Pokretanjem agenta, vršit će populiranje prosječnih OHLCV podataka i održavati bazu ažurnom sve dok se ne zaustavi. Glavnina rada prikazana je sljedećom metodom:

```
def run(self):
    while not self._stop_event.is_set():
        for coin_symbol in self.coin_symbols:
            for currency_symbol in self.currency_symbols:
                print("*Started - populate daily average historical data* Coin: {},
                    Currency: {}".format(coin_symbol, currency_symbol))
                self.populate_daily_average_historical_ohlcv(coin_symbol,
                    currency_symbol)
                print("*Finished - populate daily average historical data* Coin: {},
                     Currency: {}".format(coin_symbol, currency_symbol))

        print(self.agent_id, self.agent_name)
        time.sleep(86400)
```

Dakle, populiranje će se iterativno izvršavati sve dok agent ne dobije naredbu da prekine sa svojim radom. Sve dok je agent aktivan, prolazit će kroz permutacije vrijednosti prosljeđenih

u listi kriptovaluta i valuta. Za svaki par kriptovalute i valute pozvat će vlastitu metodu za populiranje. Ukoliko agent populira sve podatke za prosljeđene liste parametara, čekat će jedan dan te preuzeti najnovije vrijednosti i na taj način održavati bazu ažurnom. Metoda za populiranje prosječnih povijesnih dnevnih OHLCV podataka izgleda ovako:

```
def populate_daily_average_historical_ohlcv(self, coin_symbol, currency_symbol):
    """
    Fetch all available daily average historical data from API and save to database.
    :return:
    """
    last_ts = self.find_max_ts_action(coin_symbol, currency_symbol)[0]
    last_ts = last_ts if last_ts is not None else 0

    averaged_data = self.find_averages_action(coin_symbol, currency_symbol, last_ts)
    for row in averaged_data:
        row = DataFormat.zip_single_key_value(DailyAverageHistoricalDataModel.column
            [1:], row)
        data_obj = DailyAverageHistoricalDataModel(**row)

        self.insert_action(data_obj)
```

Prije nego započne s populiranjem, agent će iz baze podataka dohvatiti posljednji trenutak u vremenu za koji postoji zapis u bazi podataka za uređeni par kriptovalute i valute. Dohvaćena vremenska oznaka služi kao polazište od kojega treba započeti populiranje podataka. Simbol kriptovalte, valute i vremenska oznaka prosljeđuju se kontroleru kojega nasljeđuje agent. Kontroler nadalje zahtjev prosljeđuje sve do repozitorija koji sadrži upit za izračunavanje prosječnih OHLCV vrijednosti na temelju poslanih parametara. Dobivene prosječne vrijednosti potom se iteriraju te im se pridružuju zaglavlja atributa, nakon čega se pohranjuju u bazu podataka. Populiranje uređenog para završava nakon što se pohrane sve prosječne vrijednosti te agent potom nastavlja sa sljedećim parom. Princip rada za druge populatore prosječnih OHLCV cijena koji djeluju u drugim vremenskom intervalima je isti.

Populatori volumena burzi kriptovaluta

Dva su glavna populatora volumena burzi kriptovaluta. Razlikuju u vremenskim intervalima u kojima djeluju, korištenim krajnjim točkama i kontrolerima. Za inicijalizacija populatora volumena burzi kriptovaluta potrebno je proslijediti id agenta, proizvoljno ime agenta, tip agenta te parametre koji definiraju njegovu specifičnu ulogu. Parametri su lista kriptovaluta te burze kriptovaluta za čije će međusobne permutacije agent izvršavati populiranje podataka. Proći ćemo kroz primjer populatora dnevnih volumena burzi kriptovaluta:

```
def __init__(self, agent_id, agent_name, agent_model, currency_symbols, exchanges):

    self.agent_id = agent_id
```

```
    self.agent_name = agent_name
    self.agent_model = agent_model
    self.currency_symbols = currency_symbols
    self.exchanges = exchanges

    self._stop_event = threading.Event()
```

Pokretanjem agenta, vršit će populiranje i održavati bazu ažurnom sve dok se ne zaustavi. Glavnina rada prikazana je sljedećom metodom:

```
def run(self):
    while not self._stop_event.is_set():

        for currency_symbol in self.currency_symbols:
            for exchange in self.exchanges:
                try:
                    timestamp = self.fetch_historical_daily_exchange_volume(
                        currency_symbol, exchange, 1)["TimeFrom"]
                except:
                    print("[Error] Daily historical volume not available for pair -
                        Currency: {}, Exchange: {}".format(currency_symbol, exchange
                        ))
                    continue

                print("*Started - populate daily historical exchange volume*
                    Currency: {}, Exchange: {}".format(currency_symbol, exchange))
                self.populate_daily_historical_exchange_volume(currency_symbol,
                    exchange, timestamp)
                print("*Finished - populater daily historical exchange volume*
                    Currency: {}, Exchange: {}".format(currency_symbol, exchange))

        print(self.agent_id, self.agent_name)
        time.sleep(86400)
```

Dakle, populiranje će se iterativno izvršavati sve dok agent ne dobije naredbu da prekine sa svojim radom. Sve dok je agent aktivan, prolazit će kroz permutacije vrijednosti prosljeđenih u listi kriptovaluta i burzi kriptovaluta. Najprije će s krajnje točke pokušati dohvatiti trenutak u vremenu zadnjeg dostupnog podatka za vremenski interval na kojemu djeluje. U slučaju da on ne postoji za par kriptovalute i burze kriptovalute, prijeći će na sljedeći par. U slučaju da je uspio dohvatiti posljednji trenutak u vremenu, proslijedit će ga vlastitoj metodi za populiranje podataka, zajedno sa simbolom valute i burzom kriptovalute za koje je potrebno populirati podatke. Ukoliko agent populira sve podatke za prosljeđene liste parametara, čekat će jedan dan te preuzeti najnovije vrijednosti i na taj način održavati bazu ažurnom. Metoda za populiranje povijesnih dnevnih volumena burzi kriptovaluta izgleda ovako:

```
def populate_daily_historical_exchange_volume(self, currency_symbol, exchange,
    timestamp):
```

```
    """
    Fetch all available daily historical exchange volume from API and save to
        database.
    :return:
    """
    last_ts = self.find_max_ts_action(currency_symbol, exchange)[0]
    last_ts = last_ts if last_ts is not None else 0
    while True:
        api_data = self.fetch_historical_daily_exchange_volume(currency_symbol,
            exchange, 2000, to_ts=timestamp)
        api_data["Data"] = api_data["Data"][::-1]
            for row in api_data["Data"][1:]:
            row["currency_symbol"] = currency_symbol
            row["exchange"] = exchange
            data_obj = DailyHistoricalExchangeVolumeModel(**row)
            if int(last_ts) == data_obj.time:
                return
            elif row["volume"] is 0:
                return
            else:
                self.insert_action(data_obj)
        timestamp = api_data["TimeFrom"]
```

Prije nego započne s populiranjem, agent će iz baze podataka dohvatiti posljednji trenutak u vremenu za koji postoji zapis u bazi podataka za uređeni par valute i burze kriptovalute. Dohvaćena vremenska oznaka služi kao polazište od kojega treba započeti populiranje podataka. Podatke s krajnje točke potrebno je iterativno preuzimati s obzirom da je maksimalna količina podataka koja se može preuzeti s krajnje točke točno 2000. Krajnjoj točki prosljeđuju se simbol valute i burze kriptovalute, zajedno s vremenskom oznakom do koje želimo dohvatiti podatke. Dobivenim podacima pridružuju se vrijednosti parametara s obzirom da se ne nalaze u odgovoru, a u bazi podataka postoji jedna tablica za sve uređene parove povijesnih volumena burzi kriptovaluta istog vremenskog intervala. U slučaju da se vremenska oznaka podatka podudara s vremenskom oznakom posljednjeg zapisa u bazi podataka ili više ne postoji podataka za uređeni par, populiranje uređenog para se završava te agent nastavlja sa sljedećom trojkom. Princip rada za populatore volumena burzi kriptovaluta koji djeluju u različitim vremenskim intervalima je isti.

5.2.2.4. Populatori tehničkih indikatora

Ova kategorija tehničkih agenata poziva dohvaća OHLCV cijene određenog vremenskog intervala temeljem kojih izračunavaju niz tehničkih indikatora tržišta. Svaki indikator nasljeđuje jednog kalkulatora koji sadrži sve potrebne metode za izračuavanje indikatora te jedan od kontrolera u direktoriju indikatora, čime im je omogućen niz metoda za rad s bazom podataka. Postoji ukupno 9 populatora tehničkih indikatora, pri čemu je svaki zadužen za jednu kategoriju tehničkih indikatora. Za inicijalizacija populatora tehničkih indikatora potrebno je proslijediti id agenta, proizvoljno ime agenta, tip agenta te uređenu trojku kriptovalute, valute i burzu kriptovaluta za koje da izračuna i populira tehničke indikatore.

```
def __init__(self, agent_id, agent_name, agent_model, currency_symbols, exchanges):

self.agent_id = agent_id
self.agent_name = agent_name
self.agent_model = agent_model
self.currency_symbols = currency_symbols
self.exchanges = exchanges

self._stop_event = threading.Event()
```

Pokretanjem agenta, vršit će populiranje indikatora i održavati bazu ažurnom sve dok se ne zaustavi. Glavnina rada prikazana je sljedećom metodom:

```
def run(self):
    while not self._stop_event.is_set():

        for coin_symbol in self.coin_symbols:
            for currency_symbol in self.currency_symbols:
                for exchange in self.exchanges:

                    print("*Started - populate overlap studies indicators* Coin: {},
                         Currency: {}, Exchange: {}".format(coin_symbol,
                        currency_symbol, exchange))
                    self.populate_overlap_studies(coin_symbol, currency_symbol,
                        exchange)
                    print("*Finished - populate  overlap studies indicators* Coin:
                        {}, Currency: {}, Exchange: {}".format(coin_symbol,
                        currency_symbol, exchange))

        print(self.agent_id, self.agent_name)
        time.sleep(5)
```

Dakle, populiranje će se iterativno izvršavati sve dok agent ne dobije naredbu da prekine sa svojim radom. Sve dok je agent aktivan, prolazit će kroz permutacije vrijednosti proslijeđenih u listi kriptovaluta, valuta i burzi kriptovaluta. Za svaku uređenu trojku kriptovalute, valute i burze kriptovalute pozvat će vlastitu metodu za populiranje. Metoda za populiranje tehničkih indikatora izgleda ovako:

```
def populate_overlap_studies(self, coin_symbol, currency_symbol, exchange):
    """
    Calculate indicators and store results to database.
    :param coin_symbol:
    :param currency_symbol:
    :param exchange:
    :return:
    """
```

```
    ohlcv_data = self.find_all_prices_action(coin_symbol, currency_symbol, exchange)
    results = self.calculate_all_overlap_studies(coin_symbol, currency_symbol,
        exchange, ohlcv_data)
    for row in results:
        data_obj = OverlapStudiesModel(**row)
        if not data_obj.find_exact_price(data_obj.time, data_obj.coin_symbol,
            data_obj.currency_symbol):
            data_obj.insert()
```

Prije nego započne s populiranjem, agent će iz baze podataka dohvatiti sve OHLCV cijene za prosljeđenu uređenu trojku. Dobivene parametre te dohvaćene podatke proslijedit će kalkulatoru, koji će izvršiti računanje tehničkih indikatora. Potom će se iterirati izračunati rezultati i, ukoliko već ne postoje, pohraniti u bazu podataka. Populiranje uređene trojke završava nakon što se pohrane svi izračunati tehnički indikatori te agent potom nastavlja sa sljedećim parom. Princip rada za druge populatore tehničkih indikatora je isti.

5.2.2.5. Populatori društvenih podataka

Ova kategorija tehničkih agenata poziva krajnje točke u kategoriji društvenih podataka te nasljeđuje kontrolere u direktoriju društvenih podataka, čime im je omogućen niz metoda za rad s bazom podataka. Točnije, populatori društvenih podataka preuzimaju sve dostupne statističke podatke s krajnjih točaka društvenih podataka te ih pohranjuju u bazu podataka. Populatori društvenih podataka razlikuju u vremenskim intervalima u kojima djeluju, korištenim krajnjim točkama i kontrolerima. Za inicijalizacija populatora društvenih podataka potrebno je proslijediti id agenta, proizvoljno ime agenta, tip agenta te listu id-jeva kriptovaluta čije statistike društvenih podataka želimo populirati. Proći ćemo kroz primjer populatora društvenih podataka:

```
def __init__(self, agent_id, agent_name, agent_model, coin_symbols, currency_symbols
    , exchanges):
    self.agent_id = agent_id
    self.agent_name = agent_name
    self.agent_model = agent_model
    self.coin_symbols = coin_symbols
    self.currency_symbols = currency_symbols
    self.exchanges = exchanges

    self._stop_event = threading.Event()
```

Pokretanjem agenta, vršit će populiranje i održavati bazu ažurnom sve dok se ne zaustavi. Glavnina rada prikazana je sljedećom metodom:

```
def run(self):
    while not self._stop_event.is_set():

        for coin_id in self.coin_ids:
```

```
            timestamp = self.fetch_historical_day_social_stats_data(coin_id, 1)[0]["
                time"]

            print("*Started - populate daily historical social stats* CoinId: {}".
                format(coin_id))
            self.populate_daily_historical_social_stats(coin_id, timestamp)
            print("*Finished - populate daily historical social stats* CoinId: {}".
                format(coin_id))

        print(self.agent_id, self.agent_name)
        time.sleep(86400)
```

Dakle, populiranje će se iterativno izvršavati sve dok agent ne dobije naredbu da prekine sa svojim radom. Sve dok je agent aktivan, prolazit će kroz ' vrijednosti prosljeđenih u listi id-jeva kriptovaluta. Najprije će s krajnje točke pokušati dohvatiti trenutak u vremenu zadnjeg dostupnog podatka za vremenski interval na kojemu djeluje. U slučaju da on ne postoji za id kriptovalute, prijeći će na sljedeću kriptovalutu. U slučaju da je uspio dohvatiti posljednji trenutak u vremenu, proslijedit će ga vlastitoj metodi za populiranje podataka, zajedno s id-jem kriptovalute za koju je potrebno populirati podatke. Ukoliko agent populira sve podatke za prosljeđene id-jeve kriptovaluta, čekat će jedan dan te preuzeti najnovije vrijednosti i na taj način održavati bazu ažurnom. Metoda za populiranje povijesnih društvenih podataka izgleda ovako:

```
def populate_daily_historical_social_stats(self, coin_id, timestamp):
    """
    Fetch all available daily historical social stats from API and save to database.
    :return:
    """
    last_ts = self.find_max_ts_action(coin_id)[0]
    last_ts = last_ts if last_ts is not None else 0
    while True:
        api_data = self.fetch_historical_day_social_stats_data(coin_id, 2000, to_ts=
            timestamp)
        api_data = api_data[::-1]
        for row in api_data[1:]:
            row["coin_id"] = coin_id
            data_obj = DailyHistoricalSocialStatsModel(**row)
            if int(last_ts) == data_obj.time:
                return
            elif (row["comments"] and row["posts"] and row["followers"] and row["
                points"]) is 0:
                return
            else:
                self.insert_action(data_obj)
            timestamp = row["time"]
```

Prije nego započne s populiranjem, agent će iz baze podataka dohvatiti posljednji trenutak u vremenu za koji postoji zapis u bazi podataka za promatrani id kriptovalute. Dohvaćena vre-

menska oznaka služi kao polazište od kojega treba započeti populiranje podataka. Podatke s krajnje točke potrebno je iterativno preuzimati s obzirom da je maksimalna količina podataka koja se može preuzeti s krajnje točke točno 2000. Krajnjoj točki prosljeđuje se id kriptovalute zajedno s vremenskom oznakom do koje želimo dohvatiti podatke. Dobivenim podacima pridružuju se vrijednosti parametara s obzirom da se ne nalaze u odgovoru, a u bazi podataka postoji jedna tablica za sve društvene podatke istog vremenskog intervala. U slučaju da se vremenska oznaka podatka podudara s vremenskom oznakom posljednjeg zapisa u bazi podataka ili više ne postoji podataka za uređenu trojku, populiranje ovog id-a se završava te agent nastavlja sa sljedećim. Princip rada za druge populatore društvenih podataka koji djeluju u različitim vremenskim intervalima je isti.

5.2.3. Izvršni agenti *(executive)*

Izvršni agenti zaduženi su za sve radnje vezane uz prihvaćanje naredbi, vanjsku komunikaciju sustava, dohvaćanje najnovijih vijesti iz svijeta kriptovaluta, generiranje izvješća, analizu i slično. U trenutnoj verziji sustava, izvršni agenti samo dohvaćaju opće podatke o stanju sustava.

5.2.3.1. CEO

CEO je glavni izvršni agent koji pokreće, rukovodi, nadzire i zaustavlja sve izvršne agente. Glava je izvršnog dijela sustava, a direktno je podređen koordinatoru koji može provjeravati njegovo aktivno stanje. U slučaju da koordinator odluči obustaviti rad CEO-a, CEO će prvo zaustaviti sve podređene agente te tek onda prekinuti s radom.

Jedan CEO agent može rukovoditi s proizvoljno mnogo izvršnih agenata, neovisno o kategoriji i ulozi izvršnog agenta. Svakom izvršnom agentu može proslijediti proizvoljne parametre nad kojima će podređeni tehnički agent djelovati. Na taj način CEO definira specifičnu ulogu podređenog izvršnog agenta u sustavu. Kako bi se inicijalizirao CEO agent, potrebno ga je jednoznačno identificirati id-jem i proizvoljnim nazivom:

```
def __init__(self, agent_id, agent_name):
    threading.Thread.__init__(self)
    self.agent_id = agent_id
    self.agent_name = agent_name

    self.active_general_info_agents = {}
    self.active_exec_agents = {}
    self.general_info_agent_count = 0
    self.exec_agent_count = 0

    self._stop_event = threading.Event()
```

Pokretanjem CEO-a, instanciraju se prazne liste koje sadrže evidenciju svih pokrenutih izvršnih agenata te brojač koji prati količinu podređenih agenata jednog CEO-a. Pokretanje izvršnih agenata vrši se pomoću generičkih metoda, pri čemu svaka kategorija agenata ima vlastitu

metodu. Primjerice, pokretanje izvršnog agenta koji radi s općim podacima izvršava se pomoću sljedeće metode:

```
def start_general_info_agent(self, agent_id, agent_name, agent_type):
    agent = agent_type(agent_id, agent_name, agent_type.__name__)
    agent.start()
    print("[Executive] {}_{}({}) started.".format(agent_id, agent_name, agent_type.
        __name__))
    self.active_general_info_agents[agent_id] = [agent_name, agent_type.__name__,
        agent]
    self.general_info_agent_count += 1
```

Kako bi pokrenuo izvršnog agenta koji radi s općim podacima, potrebno ga je jednoznačno identificirati id-jem, proizvoljnim nazivom i tipom agenta. Pokretanjem novog izvršnog agenta, njegovi podaci se dodaju u listu kategorije agenata kojoj pripada te se povećava brojač aktivnih izvršnih agenata. Metode za pokretanje različitih kategorija izvršnih agenata uglavnom variraju u prosljeđenim parametrima specifičnim za tu kategoriju izvršnih agenata. Metode za zaustavljanje izvršnih agenata također su veoma slične i variraju ovisno o parametrima koji se prosljeđuju toj kategoriji izvršnih agenata. Na našem primjeru, agent općih podataka može se zaustaviti na tri načina pomoću sljedeće metode:

```
def stop_general_info_agent(self, agent_id=None, agent_name=None, agent_type=None):
    for k, v in list(self.active_general_info_agents.items()):
        if k == agent_id or v[0] == agent_name or v[1] == agent_type:
        v[2].stop()
        v[2].join()
        if v[2].stopped:
            self.active_general_info_agents.pop(k)
            print("[Executive] {}_{}({}) stopped.".format(k, v[0], v[1]))
```

Metodi se može proslijediti jedan do tri različita parametra, a to su:

- id agenta, čime se zaustavlja s radom tog specifičnog agenta,
- naziv agenta, čime se zaustavlja s radom svih agenata koji imaju isto ime,
- tip agenta, čime se zaustavljaju svi agenti istog tipa

Zaustavljanjem agenta, on se briše iz liste aktivnih agenata te sustav dalje nastavlja s radom. Broj i detalji izvršnih agenata određene kategorije podređenih specifičnom CEO-u mogu se dobiti pomoću metoda koje vraćaju stanje listi i brojača, čime koordinator može dobiti uvid u djelovanje određenog CEO agenta. Primjer nekoliko metoda za dohvaćanje detalja:

```
def get_active_general_info_agents(self):
```

```
    return self.active_general_info_agents

def get_general_info_agent_count(self):
    return self.general_info_agent_count
```

Prva metoda vraća detalje svih aktivnih općih agenata, a druga ukupni broj izvršnih agenata. U slučaju da dođe do gašenja CEO agenta, prije vlastitog zaustavljanja terminirati će rad svih podređenih izvršnih agenata na sljedeći način:

```
def stop(self):
    for k in list(self.active_general_info_agents):
        self.stop_general_info_agent(k)
    for k in list(self.active_exec_agents):
        self.stop_executive_agent(k)
    self._stop_event.set()
```

Iteriranjem kroz liste detaljnih kategorija izvršnih agenata zaustavit će se svi podređeni agenti. Koordinator ujedno može provjeriti ukoliko je određeni CEO zaustavljen ili još uvijek radi uz pomoć metode:

```
def stopped(self):
    return self._stop_event.is_set()
```

5.2.3.2. Rukovoditelj generalnih podataka *(general info agent)*

Rukovoditelj generalnih podataka poziva krajnje točke u kategoriji generalnih podataka te nasljeđuje kontroler u direktoriju generalnih podataka. Točnije, rukovoditelj generalnih podataka preuzimaju sve dostupne statističke podatke s krajnjih točaka generalnih podataka vezane uz stanje sustava, broj iskorištenih API poziva po pojedinoj kategoriji krajnjih točaka, opće informacije o dostupnim burzama i digitalnim novčanicima, parove podržanih kriptovaluta i burzi kriptovaluta, rudarske ugovore i tako dalje. Namijena ovog agenta jest odgovaranje na upite od strane sustava kako bi se omogućio nesmetan rad. U trenutnoj verziji sustava nije ostvarena komunikacija s vanjskim entitetima, pa izvršni agenti nemaju značajnu ulogu. Pokretanjem agenta, čekat će na upit od strane sustava sve dok se ne zaustavi njegov rad. U slučaju da primi poziv, prosljeđuje ga putem jedne od svojih metoda na krajnje točke generalnih podataka te vratiti odgovor u prikladnom obliku.

5.2.4. Financijski agenti *(financial)*

Financijski agenti zaduženi su za sve radnje vezane uz rad s digitalnim novčanikom, praćenjem cijena kriptovaluta izraženih u određenim valutama na specificiranim burzama kriptovaluta u stvarnom vremenu, obavještavanje ostatka sustava o najnovijim promjenama cijena,

trgovanje na tržištu kriptovaluta i tako dalje. Svaki od agenata izvršava točno određenu ulogu, što ga čini nezamijenjivim dijelom sustava.

5.2.4.1. CFO

CFO je glavni financijski agent koji pokreće, rukovodi, nadzire i zaustavlja sve financijske agente te vodi glavnu evidenciju stanja sredstava na računu. Glava je financijskog dijela sustava, a direktno je podređen koordinatoru koji može provjeravati njegovo aktivno stanje. U slučaju da koordinator odluči obustaviti rad CFO-a, CFO će prvo zaustaviti sve podređene agente te tek onda prekinuti s radom. Jedan CFO agent može rukovoditi s proizvoljno mnogo financijskih agenata, neovisno o kategoriji i ulozi financijskog agenta. Svakom financijskom agentu može proslijediti proizvoljne parametre, kao što su parovi određenih kriptovaluta, valuta i burzi kriptovaluta nad kojima će podređeni financijski agent djelovati. Na taj način CFO definira specifičnu ulogu podređenog financijskog agenta u sustavu. U trenutnoj inačici sustava, kategorije agenata u domeni kontrole CFO-a su operatori i trgovci. Kako bi se inicijalizirao CFO agent, potrebno ga je jednoznačno identificirati id-jem i proizvoljnim nazivom te mu se može proslijediti i burza kriptovaluta za koju je zadužen:

```
def __init__(self, agent_id, agent_name, exchange):
    threading.Thread.__init__(self)
    self.agent_id = agent_id
    self.agent_name = agent_name
    self.exchange = exchange

    self.account_balance = 0
    self.coin_balance = {}
    self.active_finance_agents = {}
    self.active_trader_agents = {}
    self.finance_agent_count = 0
    self.trader_agent_count = 0

    self._stop_event = threading.Event()
```

Pokretanjem CFO-a, instanciraju se prazne liste koje sadrže evidenciju svih pokrenutih financijskih agenata te brojač koji prati količinu podređenih agenata jednog CFO-a. Pokretanje financijskih agenata vrši se pomoću generičkih metoda, pri čemu svaka kategorija agenata ima vlastitu metodu. Primjerice, pokretanje financijskih agenta koji trguju sa zadanom kriptovalutom, valutom i burzom kriptovaluta izvršava se pomoću sljedeće metode:

```
def start_trader_agent(self, agent_id, agent_name, agent_type, coin_symbol="BTC",
    currency_symbol="USD", exchange="bitfinex"):
    agent = agent_type(agent_id, agent_name, agent_type.__name__, coin_symbol,
        currency_symbol, exchange)
    agent.start()
```

```
    print("[Financial] {}_{}({}): {}-{}-{} started.".format(agent_id, agent_name,
        agent_type.__name__, coin_symbol, currency_symbol, exchange))
    self.active_trader_agents[agent_id] = [agent_name, agent_type.__name__, agent,
        coin_symbol, currency_symbol, exchange]
    self.trader_agent_count += 1
```

Kako bi pokrenuo financijskog agenta koji trguje na tržištu kriptovaluta, potrebno ga je jednoznačno identificirati id-jem, proizvoljnim nazivom, tipom agenta te mu proslijediti simbola kriptovalute, simbol valute te burzu kriptovaluta nad kojima će agent djelovati. Na ovaj način CFO može podijeliti zadatke između većeg broja agenata istog tipa. Pokretanjem novog financijskog agenta, njegovi podaci se dodaju u listu kategorije agenata kojoj pripada te se povećava brojač aktivnih financijskih agenata. Metode za pokretanje različitih kategorija financijskih agenata uglavnom variraju u prosljeđenim parametrima specifičnim za tu kategoriju financijskih agenata. Pokretanje agenta uz pomoć gore navedene metode izgledalo bi ovako:

```
self.start_trader_agent(self.trader_agent_count, "ProfitMeister 3000",
    MinuteTradeAgent, "ETH", "USD", "bitfinex")
```

Dakle, pokreće se financijski agent u kategoriji trgovaca s jedinstvenim id-jem i proizvoljnim nazivom s parametrima predefinirane kriptovalute, valute i burze kriptovaluta nad kojima će djelovati. Može se definirati neograničen broj trgovaca istog tipa koji će djelovati na različitim uređenim trojkama. Metode za zaustavljanje financijskih agenata također su veoma slične i variraju ovisno o parametrima koji se prosljeđuju toj kategoriji financijskih agenata. Na našem primjeru, agent trgovac može se zaustaviti na šest načina pomoću sljedeće metode:

```
def stop_trader_agent(self, agent_id=None, agent_name=None, agent_type=None,
    coin_symbol=None, currency_symbol=None, exchange=None):
    for k, v in list(self.active_finance_agents.items()):
        if k == agent_id or v[0] == agent_name or v[1] == agent_type or v[2] ==
            coin_symbol or v[3] == currency_symbol or v[4] == exchange:
            v[2].stop()
            v[2].join()
            if v[2].stopped:
            self.active_finance_agents.pop(k)
            print("[Trader] {}_{}({}): {}-{} stopped.".format(k, v[0], v[1], v[3], v
                [4]))
```

Metodi se može proslijediti jedan do šest različita parametra, a to su:

- id agenta, čime se zaustavlja s radom tog specifičnog agenta,
- naziv agenta, čime se zaustavlja s radom svih agenata koji imaju isto ime,
- tip agenta, čime se zaustavljaju svi agenti istog tipa,
- simbol kriptovalute, čime se zaustavljaju svi agenti koji trguju s tom kriptovalutom,

- simbol valute, čime se zaustavljaju svi agenti koji trguju s tom valutom,
- burza kriptovalute, čime se zaustavljaju svi agenti koji trguju na toj burzi kriptovaluta,

Zaustavljanjem agenta, on se briše iz liste aktivnih agenata te sustav dalje nastavlja s radom. Broj i detalji financijskih agenata određene kategorije podređenih specifičnom CFO-u mogu se dobiti pomoću metoda koje vraćaju stanje listi i brojača, čime koordinator može dobiti uvid u djelovanje određenog CFO agenta. Primjer nekoliko metoda za dohvaćanje detalja:

```
def get_active_trader_agents(self):
    return self.active_trader_agents

def get_trader_agent_count(self):
    return self.trader_agent_count
```

Prva metoda vraća detalje svih aktivnih agenata trgovaca, a druga ukupni broj agenata trgovaca. U slučaju da dođe do gašenja CFO agenta, prije vlastitog zaustavljanja terminirati će rad svih podređenih tehničkih agenata na sljedeći način:

```
def stop(self):
    for k in list(self.active_finance_agents):
        self.stop_financial_agent(k)
    for k in list(self.active_trader_agents):
        self.stop_trader_agent(k)
    self._stop_event.set()
```

Iteriranjem kroz liste detaljnih kategorija financijskih agenata zaustavit će se svi podređeni agenti. Koordinator ujedno može provjeriti ukoliko je određeni CFO zaustavljen ili još uvijek radi uz pomoć metode:

```
def stopped(self):
    return self._stop_event.is_set()
```

5.2.4.2. Operatori *(trackers)*

Ova kategorija financijskih agenata poziva krajnje točke u kategoriji cijena. Operatori ažuriraju sustav najnovijim cijenama, volumenima i ostalim vrijednostima potrebnim za nesmetano funkcioniranje sustava za uređene trojke kriptovaluta, valuta i burzi kriptovaluta. Za svaku kategoriju krajnjih točaka koja se ažurira kontinuiranim slijedom podataka, postojao bi operator koji bi pratio promjene i propagirao ih dalje u sustavu na principu modela objave i pretplaćivanja (eng. *publish-subscribe model*). Dakle, operator bi sustavu objavljivao najnovije informacije, a agenti bi se pretplaćivali na dio njima potrebnih informacija, čime bi svi agenti dobivali najažurnije informacije. U trenutnoj verziji sustava, napravljena su dva tipa operatora koji rade s

praćenjem i propagiranje OHLCV cijena i najnovijih vrijednosti tehničkih indikatora baziranih na tim cijenama. S obzirom da još ne postoje agenti koji bi se pretplatili na informacije koje pružaju operatori, njihova trenutna uloga u sustavu nije značajna.

5.2.4.3. Trgovci *(traders)*

Ova kategorija financijskih agenata jedna je od onih koja bi se pretplatila na dio informacija koje propagiraju operatori. Točnije, primali bi najnovije cijene, volumene i ostale vrijednosti potrebne za nesmetano trgovanje na burzi kriptovaluta. Svaki trgovac djeluje s uređenom trojkom kriptovalute, valute i burze kriptovalute te nizom parametara koji definiraju ulogu agenta. Teoretska podloga razrađenih strategija trgovanja trenutno se svodi na nagađanja i pretpostavke ili stohastičke agente koji djeluju na tržištu regularnih dionica. U usporedbi s regularnim dionicama, tržište kriptovaluta puno je varijabilnije, nestabilnije i teže za predvidjeti, stoga je teoretska strategija trgovanja koja vrijdi za regularne burze gotovo neuporabljiva. U nastavku rada predstavit će se nekoliko proizvoljnih agenata koji imaju različite generalne strategije trgovanja. Svakoj se mogu podesiti različiti parametri koji u konačnici rezultiraju promjenom specifične strategije trgovanja. Uspješnost pojedinog trgovca ovisit će o kvaliteti odabrane generalne strategije trgovanja i parametrima specifične strategije trgovanja. Između nekoliko implementiranih generalnih strategija, cilj je pronaći najprofitabilniju specifičnu strategiju s optimalnim parametrima. CFO može inicijalizirati proizvoljan broj trgovaca istog tipa s različitim parametrima te na taj način doći do najbolje strategije trgovanja. U nastavku rada opisat će se jedan trgovac s vlastitim parametrima trgovanja koji trguje u razdoblju od jedne minute. Trgovac se inicijalizira prosljeđivanjem niza parametara:

```
def __init__(self,
    agent_id,
    agent_name,
    agent_model,
    coin_symbol,
    currency_symbol,
    average_minutes=7200,
    refresh_price_sec=15,
    stop_loss_percentage=35,
    buy_percentage=10,
    sell_percentage=10,
    next_buy_wait_sec=5,
    next_sell_wait_sec=5,
    available_funds=10,
    *args):

    threading.Thread.__init__(self)
    self.agent_id = agent_id
    self.agent_name = agent_name
    self.agent_model = agent_model
    self.coin_symbol = coin_symbol
    self.currency_symbol = currency_symbol
```

```
self.average_minutes = average_minutes
self.refresh_price_sec = refresh_price_sec
self.stop_loss_percentage = stop_loss_percentage
self.buy_percentage = buy_percentage
self.sell_percentage = sell_percentage
self.next_buy_wait_sec = next_buy_wait_sec
self.next_sell_wait_sec = next_sell_wait_sec
self.available_funds = available_funds

# self.start_time = 0.0
self.average_open = 0.0
self.average_close = 0.0
self.average_high = 0.0
self.average_low = 0.0
self.current_price = 0
self.last_buy_time = 0.0
self.last_sell_time = 0.0
self.coin_balance = 0.0

self._stop_event = threading.Event()
```

Ulazni parametri agenta koji utječu na formiranje specifične strategije trgovanja su:

- **coin_symbol:** kratica naziva kriptovalute s kojom će agent trgovati
- **currency_symbol:** željena valuta trgovanja
- **account_balance:** početno novčano stanje na digitalnom novčaniku
- **coin_balance:** početna količina kriptovalute na digitalnom novčaniku
- **average_minutes:** parametar za limit poziva krajnje točke, a označava kvantitetu prethodnih cijena koje će se dohvatit
- **refresh_price_sec:** broj sekundi nakon kojih će agent provjeriti trenutnu cijenu
- **stop_loss_percentage:** postotak sniženja od prosjeka najnižih cijena za koji agent prodaje sve kriptovalute kako bi spriječio gubitak
- **buy_percentage:** postotak od ukupnog novčanog stanja na digitalnom novčaniku koji će agent uložiti prilikom sljedeće kupnje
- **sell_percentage:** postotak od ukupne količine kriptovaluta na digitalnom novčaniku koje će agent prodati prilikom sljedeće prodaje
- **nex_buy_wait_sec:** broj sekundi koji će pričekati prije nego napravi sljedeću kupnju
- **next_sell_wait_sec:** broj sekundi koji će pričekati prije nego napravi sljedeću prodaju

Glavna metoda koja definira generalnu strategiju trgovanja:

```
def run(self):
    while not self._stop_event.is_set():
        print(self.agent_id, self.agent_name)
        time.sleep(2)
        if time.time() - self.start_time > 60:
            self.calculate_prices()
            self.start_time = time.time()

        avg_open = float("{0:.02f}".format(self.average_open))
        avg_close = float("{0:.02f}".format(self.average_close))
        avg_high = float("{0:.02f}".format(self.average_high))
        avg_low = float("{0:.02f}".format(self.average_low))
        stop_loss = avg_low * (1.0 - float(self.stop_loss_percentage) / 100)
        self.current_price = float(self.fetch_historical_minute_ohlcv(self.
            coin_symbol, self.currency_symbol).values()[0])
        current_price = float("{0:.02f}".format(self.current_price))
        print("Current historical_data: {0}".format(current_price))

        if avg_low > current_price > stop_loss and int(self.available_funds) > 0:
        self.buy(current_price,
        (float(self.available_funds) / float(self.buy_percentage)) / current_price)

        elif current_price > avg_high and float("{0:.02f}".format(self.coin_balance)
            ) > 0.00:
        self.sell(current_price, float(self.coin_balance / float(self.
            sell_percentage)))

        elif current_price < stop_loss and float("{0:.02f}".format(self.coin_balance
            )) > 0.00:
        self.sell(current_price, self.coin_balance)

        time.sleep(self.refresh_price_sec)
```

Agent trguje u stvarnom vremenu ovisno o postavljenim ulaznim parametrima s cjelokupnim raspoloživim novčanim iznosom *(account_balance)* ili količinom kriptovaluta *(coin_balance)*. Trguje s postavljenom kriptovalutom *(coin_symbol)* izraženom određenom valutom *(currency_symbol)*. Agent izračunava prosjek najnižih i najviših cijena prema OHLCV principu za prethodni vremenski period zadan parametrom prosječnih minuta *(average_minutes)*. Ukoliko trenutna cijena prijeđe prosijek najviših cijena, agent će u intervalima čekanja sljedeće prodaje *(next_sell_wait_sec)* početi prodavati postotak ukupnih zaliha kriptovaluta s digitačnog računa definiran s postotkom prodaje *(sell_percentage)*. U suprotnom, padne li cijena ispod prosjeka najnižih cijena, agent će u intervalima čekanja sljedeće kupnje *(next_buy_wait_sec)* početi kupovati postotak od ukupnog novčanog iznosa s računa definiranog sa postotkom kupnje *(buy_percentage)*. U suštini, agent nastoji kupiti kriptovalute ispod prosječne niske cijene i prodati iznad prosječne visoke cijene. U slučaju da cijena na tržištu krene nekontrolirano padati, stupanj rizika može se odrediti sigurnosnim parametrom sprječavanja gubitka *(stop_loss_percentage)* kojim se definira potreban postotak pada cijene za izvršavanje instantne prodaje svih zaliha kriptovaluta koje agent posjeduje. U jednominutnim intervalima, agent će preuzeti podatke najviših i najnižih cijena tržišta prethodne minute te izračunati novi prosjek temeljem kojega će donositi odluke o

kupnji i prodaji. Računanje novog prosjeka vrši se pomoću sljedeće metode:

```
def calculate_prices(self):
    self.price_list = self.fetch_historical_minute_ohlcv(self.coin_symbol, self.
        currency_symbol, self.exchange, limit=self.average_minutes)
    self.average(self.price_list)
    self.start_time = time.time()
```

Dakle, agent ažurira vlastitu listu cijena nizom povijesnih OHLCV vrijednosti definiranim parametrom *average_minutes*. Potom poziva vlastitu metodu za izračunavanje prosjeka cijena te resetira brojač za čekanje nove minute. Metoda za populiranje prosjeka izgleda ovako:

```
def average(self, price_list):
    for row in price_list:
        if (self.average_open or self.average_close or self.average_high or self.
            average_low) == 0:
            self.average_open = row["open"]
            self.average_close = row["close"]
            self.average_high = row["high"]
            self.average_low = row["low"]
        else:
            self.average_open = (self.average_open + row["open"]) / 2
            self.average_close = (self.average_open + row["close"]) / 2
            self.average_high = (self.average_open + row["high"]) / 2
            self.average_low = (self.average_open + row["low"]) / 2
            print("***AVG.OPEN: {0}, ***AVG.CLOSE: {1}, ***AVG.HIGH: {2}, ***AVG.LOW
                : {3}"
            .format(self.average_open, self.average_close, self.average_high, self.
                average_low))
```

Za niz prosljeđenih vrijednosti izračunati će prosjek te ga pohraniti u vlastite varijable, temeljem kojih će se vršiti usporedba i donositi daljnje odluke o kupnji ili prodaji. Kupnju agent izvršava pomoću metode:

```
def buy(self, price, amount):
    if float(time.time() - self.last_buy_time) >= float(self.next_buy_wait_sec):
        self.coin_balance += amount
        self.available_funds -= (self.available_funds / self.buy_percentage)
        print("[BUY] CoinName: {0}, BuyPrice: {1}, Currency: {2}, BuyAmount: {3},
            BuyValue: {4}"
        .format(self.coin_symbol, price, self.currency_symbol, amount, price*amount)
            )
        print("[BALANCE] Coin: {0}, Account: {1}".format(self.coin_balance, self.
            available_funds))
        self.last_buy_time = time.time()
```

Prvo ispituje da li je uistinu vrijeme za kupnju, a ako jest, ažurira varijable stanja na računu, dostupnih sredstava te resetira brojač za posljednju kupnju. Prodaja se vrši na sličan način:

```
def sell(self, price, amount):
    if float(time.time() - self.last_sell_time) >= float(self.next_sell_wait_sec):
        self.available_funds += amount * price
        self.coin_balance -= amount
        print("[SELL] CoinName: {0}, SellPrice: {1}, Currency: {2}, SellAmount: {3},
             SellValue: {4}"
        .format(self.coin_symbol, price, self.currency_symbol, amount, price*amount)
            )
        print("[BALANCE] Coin: {0}, Account: {1}".format(self.coin_balance, self.
            available_funds))
        self.last_sell_time = time.time()
```

Prvo ispituje da li je uistinu vrijeme za prodaju, a ako jest, ažurira varijable stanja na računu, dostupnih sredstava te resetira brojač za posljednju prodaju.

5.3. Analiza

Kako bi se provela analiza nad podacima pohranjenim u bazi podataka, korišten je *Metabase*. Metabase je alat poslovnog izvješćivanja (eng. *business intelligence tool*) otvorenog koda (eng. *open source*). Podržava postavljanje pitanja vezanih uz promatrane podatke te pruža odgovore u smisaonom formatu. Postavljena pitanja mogu se pohraniti za kasnije, što omogućava jednostavan povratak na iste. Za analizu podataka, mogu se izraditi proizvoljni grafički prikazi, od stupčastih grafikona do detaljnih tablica. Grafički prikazi se potom mogu grupirati na nadzorne ploče (eng. *dashboard*), čime se dobiva istovremeni pregled svih grafičkih prikaza dodjeljenih promatranoj nadzornoj ploči. Nadalje, one se mogu, skupa s grafičkim prikazima, jednostavno podijeliti s vanjskim suradnicima. Prije nego li krenemo raditi s ovim alatom, potrebno je konfigurirati pristup vlastitoj bazi podataka. Nakon što je omogućen pristup s definiranim ulogama, *Metabase* ima pregled svih podataka do kojih mu je omogućen pristup. Moguće ih je revidirati, filtrirati samo željene informacije prema zadanim parametrima te birati način na koji će informacije biti prikazane. Osim toga, podaci se mogu dohvatiti ručno napisanim upitom te se potom vizualizirati na željeni način. Podržane vizuaizacije su: tablice (eng. *table*), linijski dijagrami ()eng. *line*), stupčasti dijagrami (eng. *bar*), tortni grafikon (eng. *pie*), prostorni grafikon (eng. *area*) te karte (eng. *maps*). *Metabase* je brz i jednostavan način vizualiziranja prikupljenih podataka, ali je zato i ograničen u mogućnostima dublje analize. Podloga za detaljniju analizu je osigurana projektom, no trenutno nije provodena. Rezultati dubinske analize nekom drugom prilikom. [**49**]

Bibliografija

[1] A. Judmayer, N. Stifter, K. Krombholz, E. Weippl (2017), *Blocks and Chains: Introduction to Bitcoin, Cryptocurrencies, and Their Consensus Mechanisms*, Morgan & Claypool

[2] Andreas M. Antonopoulos (2015), *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*, O'Reilly Media

[3] Melanie Swan (2015), *Blockchain: Blueprint For a New Economy*, O'Reilly Media

[4] D. Buterin, E. Ribarić, S. Savić (2015), *Bitcoin: Nova globalna valuta, investicijska prilika ili nešto treće?*, Zbornik Veleučilišta u Rijeci

[5] Bambara J.J., Allen P.R. (2018), *Blockchain: A practical guideto developing business, law, and technology solutions*, McGraw-Hill Education

[6] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gün (2016), *On scaling decentralized blockchains*, Springer, Berlin, Heidelberg

[7] H. Halaburda, M. Sarvary (2016), *Beyond Bitcoin: The Economics of Digital Currencies*, Palgrave Macmillan

[8] G. Karame, E. Androulaki (2017), *Bitcoin and Blockchain Security*, Artech House

[9] D. Drescher (2017), *Blockchain Basics: A Non-Technical Introduction in 25 Steps*, Apress

[10] V. Dhillon, D. Metcalf, M. Hooper (2017), *Blockchain enabled Applications: Undrestand the Blockchain Ecosystem and How to Make it Work for You*, Apress

[11] Lee Kuo Chuen D. (2015), *Handbook of Digital Currency: Bitcoin, Innovation, Financial Instruments, and Big Data*, Elsevier Inc

[12] R. Caetano (2015), *Learning Bitcoin: Embrace the new world of finance by leveraging the power of crypto-currencies using Bitcoin and the Blockchain*, Packt Pubishing

[13] A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder (2016), *Bitcoin and Crypto-currency Technologies*, Princeton University Press

[14] R. Bertschi (2018), *Technical Analysis: Explained*, Credit Suisse

[15] W. J. Podobas (2018), *The Advanced Technical Analysis: The complex technical analysis of assets*, Nepoznato

[16] D. Shah, K. Zhang (2014), *Bayesian regression and Bitcoin*, Laboratory for Information and Decision Systems

[17] E. Samanidou, E. Zschischang, D. Stauffer, T. Lux (2007), *Agent-based models of financial markets*, IOP Publishing

[18] L. Feng, B. Li, B. Podobnik, T. Preis, H. E. Stanley (2012), *Linking agent-based models and stochastic models of financial markets*, PNAS

[19] T. Lux, M. Marchesi (1999), *Scaling and criticality in a stochastic multi-agent model of a financial market*, Nature

[20] Leksikografski zavod Miroslav Krleža, *bitcoin*, preuzeto 10.6.2019. s `http://www.enciklopedija.hr/natuknica.aspx?id=70775`

[21] Nikola Rogina (2017), *Princip rada kriptovaluta*, preuzeto 10.6.2019. s `https://www.kriptovaluta.hr/bitcoin/princip-rada-kriptovaluta/`

[22] Europska Komisija, Predstavništvo u Hrvatskoj (2019), *Kriptovalute i blockchain – sve što trebate znati*, preuzeto 10.6.2019. s `https://ec.europa.eu/croatia/cryptocurrencies_and_blockchain_all_you_need_to_know_hr/`

[23] Shaan Ray (2017), *Merkle Trees*, preuzeto 13.6.2019. s `https://hackernoon.com/merkle-trees-181cb4bc30b4`

[24] Bitcoin.org (2018), *Blockchain*, preuzeto 13.6.2019. s `https://bitcoin.org/en/blockchain-guide#introduction`

[25] Ajitesh Kumar (2018), *Bitcoin Blockchain – What is Proof of Work?*, preuzeto 13.6.2019. s `https://vitalflux.com/bitcoin-blockchain-proof-work/`

[26] Kriptonesia.com (2019), *Bitcoin Mining Diagram*, preuzeto 14.6.2019. s `https://www.kriptonesia.com/bitcoin-mining-diagram/`

[27] Cryptog (2018), *Cryptocurrency Forks Explained*, preuzeto 14.6.2019. s `http://cryptosuss.com/cryptocurrency-forks-explained/`

[28] Blockchain.com (2019), *Hashrate Distribution: An estimation of hashrate distribution amongst the largest mining pools*, preuzeto 1.9.2019. s `https://www.blockchain.com/en/pools`

[29] Cryptocompare.com (2019), *Overview*, preuzeto 1.9.2019. s `https://www.cryptocompare.com/`

[30] Leksikografski zavod Miroslav Krleža (2019), *kriptografija*, preuzeto 15.6.2019. s `http://www.enciklopedija.hr/natuknica.aspx?ID=33988`

[31] Europska komisija - predstavništvo u Hrvatskoj (2019), *Kriptovalute i blockchain – sve što trebate znati*, preuzeto 15.6.2019. s `https://ec.europa.eu/croatia/cryptocurrencies_and_blockchain_all_you_need_to_know_hr`

[32] Finjan Team (2018), *Advantages of Cryptocurrency*, preuzeto 2.7.2019. s https://blog.finjan.com/advantages-of-cryptocurrency/

[33] Adam Barone (2019), *What Is the Quantity Theory of Money?*, preuzeto 15.7.2019. s https://www.investopedia.com/insights/what-is-the-quantity-theory-of-money/

[34] coinify.com (2019), *Conify: Blockchain payments*, preuzeto 1.9.2019. s https://www.coinify.com/

[35] coinkite.com (2019), *Coinkite: bitcoin wallet with multi-signature bank-grade security*, preuzeto 1.9.2019. s https://coinkite.com/

[36] bitpay.com (2019), *Bitpay: A bitcoin payment processor*, preuzeto 1.9.2019. s https://bitpay.com/

[37] revelsystems.com (2019), *Revel ipad point-of-sale software*, preuzeto 1.9.2019. s http://revelsystems.com/

[38] paystand.com (2019), *Paystand: 0% business payments*, preuzeto 1.9.2019. s www.paystand.com

[39] xbterminal.io (2019), *Bitcoin pos system*, preuzeto 1.9.2019. s https://xbterminal.io/

[40] Investing.hr (2019), *Što je digitalni novčanik?*, preuzeto 23.7.2019. s https://www.investing.hr/sto-je-digitalni-novcanik

[41] coinatmradar.com (2019), *Bitcoin ATM Map*, preuzeto 23.7.2019. s https://coinatmradar.com

[42] bitcoin-mjenjacnica.hr (2019), *Bitcoin mjenjačnica*, preuzeto 23.7.2019. s https://bitcoin-mjenjacnica.hr/#exchange

[43] coinbase.com (2019), *Buy and sell cryptocurrency*, preuzeto 23.7.2019. s https://www.coinbase.com/

[44] pro.coinbase.com (2019), *The most trusted platform for trading cryptocurrency*, preuzeto 23.7.2019. s https://pro.coinbase.com/

[45] localbitcoins.com (2019), *Buy and sell bitcoins near you*, preuzeto 23.7.2019. s https://localbitcoins.com/

[46] tradingview.com (2019), *TradingView*, preuzeto 1.9.2019. s https://www.tradingview.com

[47] postgresql.org (2019), *PostgreSQL: The World's Most Advanced Open Source Relational Database*, preuzeto 1.9.2019. s https://www.postgresql.org/

[48] sqlalchemy.org (2019), *The Python SQL Toolkit and Object Relational Mapper*, preuzeto 1.9.2019. s https://www.sqlalchemy.org/

[49] cryptocompare.com (2019), *The Ultimate API Solution*, preuzeto 1.9.2019. s https://min-api.cryptocompare.com/documentation

[50] metabase.com (2019), *Metabase is the easy, open source way for everyone in your company to ask questions and learn from data*, preuzeto 1.9.2019. s https://metabase.com/

[51] SPADE (2019), *Metabase is the easy, open source way for everyone in your company to ask questions and learn from data*, preuzeto 1.9.2019. s https://spade-mas.readthedocs.io/en/latest/readme.html

[52] unizd.hr (2019), *Metode znanstvenih istraživanja*, preuzeto 1.9.2019. s http://www.unizd.hr/portals/4/nastavni_mat/1_godina/metodologija/metode_znanstvenih_istrazivanja.pdf

[53] palletsprojects.com (2019), *Flask*, preuzeto 1.9.2019. s https://palletsprojects.com/p/flask/

[54] python-requests.org (2019), *requests*, preuzeto 1.9.2019. s https://2.python-requests.org/en/master/

[55] numpy.org (2019), *NumPy*, preuzeto 1.9.2019. s https://numpy.org/

[56] TA-Lib (2019), *Python wrapper for TA-Lib*, preuzeto 1.9.2019. s https://mrjbq7.github.io/ta-lib/index.html

[57] numba.pydata.org (2019), *Numba*, preuzeto 1.9.2019. s http://numba.pydata.org/

[58] seaborn.pydata.org (2019), *Seaborn*, preuzeto 1.9.2019. s https://seaborn.pydata.org/

[59] pandas.pydata.org (2019), *Pandas*, preuzeto 1.9.2019. s https://pandas.pydata.org/

6. Zaključak

U ovom radu opisali su se osnovni koncepti i terminologija potrebni za razumijevanje domene kriptovaluta, od povijesti njihova nastanka i razvoja, tehničke pozadine i načina rada, alternativnih kriptovaluta pa sve do ekosustava koji se razvio oko kriptovaluta. Objasnili su se nedostaci bitcoina i povezanih kriptovaluta, ali i njihove prednosti u odnosu na tradicionalne sustave plaćanja. Pokazao se način rada višeagentnog sustava kroz uloge pojedinačnih agenata, njihove međusobne interakcije i izvršavane radnje koje u cjelini ispunjavaju svrhu sustava. Analizirao se dio prikupljenih podataka na obrascu od 10 kriptovaluta, 6 burzi kriptovalute te 3 valute kako bi se dobio uvid u obrasce i trendove kretanja elemenata tržišta kriptovaluta. Na uzorku od 4199 promatranih kriptovaluta zaključili smo da većina kriptovaluta koristi Scrypt algoritam hashiranja, da ih većina trguje na tržištu kriptovaluta te da postoji samo nekolicina kriptovaluta koja je u startu izdala sve jedinice kriptovalute u opticaj. Primjetili smo da je najučestalije vrijeme rudarenja novog bloka između 40 i 60 sekundi. Zaključili smo da Bitfinex burza kriptovaluta ima najveće prosječne cijene, WavesDex najniže, a da su preostale četiri promatrane burze kriptovaluta podjednako balansirane. Ovakva distribucija cijena ostavlja prostora za arbitražna trgovanja među burzama kriptovaluta. Također, Bitfinex je od svog osnutka u većini vremena imao najveći volumen među promatranim burzama kriptovaluta. Izuzeci su bili u periodima naglih rasta cijena, kao što je bio onaj krajem 2018. godine, kada su Coinbase i Bittrex privremeno premašile ukupni volumen Bitfinexa. Na uzorku promatranih kriptovaluta, Bittrex ih je podržavao najviše, a slijedili su ga Bitfinex i Coinbase. Na svim burzama, Bitcoin je imao daleko najveće cijene i volumen za promatrano vremensko razdoblje, a slijedio ga je Ethereum te potom Ripple. Međutim, primjetili smo da Bitcoin stranica slabije zastupljena na Facebooku i ima slabo aktivnu zajednicu, gdje pretežito dominiraju Ethereum i Ripple. Broj prosječnih Facebook komentara i diskusija još je uvijek na strani Bitcoina, a jako su mu blizu Ethereum i Ripple. S druge strane, Bitcoin daleko dominira na svim aspektima Reddita, gdje se nalazi većina njegove zajednice. Isto vrijedi i za Twitter, gdje je Ripple iznenađujuće zastupljen, gotovo kao i Bitcoin, unatoč manjoj generalnoj zajednici. Što se tiče općeg razvoja kriptovaluta, analizom GitHub repozitorija zaključeno je da Bitcoin ima najveći broj otvaranih problema, ali ih i najbrže rješava. Prema trendu rješavanja problema, ističu se Bitcoin, Ethereum i Eos.io s pozitivnim trendom rasta. Cardano je kroz vrijeme vidljivo usporio s brojem riješenih problema. Valja zamijetiti da je Ethereum jedina kriptovaluta čiji broj novootvorenih problema nije u konstantnom porastu. U suštini, Bitcoin i dalje dominira tržištem kao vodeća kriptovaluta, a često ga slijede Ethereum i Ripple. Ostale se kriptovalute u odnosu na tri navedene rijetko kada ističu.

Bitcoin je svojim primjerom dokazao da su decentralizirane kriptovalute tehnički izvedive. Od svojih početaka 2008. godine, Bitcoin protokol i njegova zajednica pokazali su da je moguće rukovoditi i održavati decentraliziranu globalnu valutu. U isto vrijeme, cijeli ekosustav izgrađen oko kriptovaluta pokazao se otpornim na različite pokušaje napada i ostalih malicioznih radnji. Kao rezultat toga, postalo je vrlo jednostavno kreirati valutu koja se može globalno koristiti bez potrebe za posrednikom da distribuira jedinice valute. Ovakva promjena paradigme razmišljanja o novcu može nas uvesti u budućnost gdje postoji veći broj različitih kriptovaluta koje pokrivaju različit skup primjena prihvaćenih unutar zajednice. Sve dok postoje načini jed-

nostavnog korištenja i razmjene različitih kriptovaluta, nije se neophodno oslanjati na samo jednu kriptovalutu za sve radnje. Broj novih kriptovaluta u stalnom je porastu, iako još uvijek nemaju značajan utjecaj na ekonomiju. Među njima, pojavio se i velik broj novih kriptovaluta koji imaju različite načine rješavanja niza izazova i problema. Svaka nova generacija na bolji način adresira postojeće probleme i komplementira neke nove slučajeve korištenja. Slijedeći ovakav trend, pojavit će se i kriptovalute koje rudarenje čine efikasnijim za decentralizirane valute ili rješavaju neki drugi nedostatak. Stotine godina monetarne povijesti uči nas da je korištenje bilo koje valute utemeljeno na povjerenju, a danas još uvijek postoji premalo iskustava s digitalnim valutama da bi se razvilo univerzalno povjerenje. Naravno, i dalje postoje manje zajednice koje su određenu kriptovalutu već prihvatile kao interno sredstvo plaćanja. S vremenom, sve će se veći broj korisnika priviknuti na inovacije koje pružaju kriptovalute te ih eventualno početi tretirati kao validna sredstva plaćanja. S druge strane, kroz povijest su istovremeno postojala različita sredstva plaćanja, pa bi tako mogle koegzistirati digitalne valute određene namjene zajedno s tradicionalnim sredstvima plaćanja. S obzirom da je posljednjih godina niz priznatih banaka počeo eksperimentirati s kriptovalutama, mogle bi se pojaviti i digitalne, centralizirane alternative tradicionalnom novcu. Doduše, tehnologije iza kriptovaluta u posljednje se vrijeme sve više primjenjuje u druge svrhe. Razvojem tehnologija kriptovaluta, inovacije će se uvijek koristiti u svrhe poboljšanja postojećih modela. Iako novim tenhologijama treba neko vrijeme da zamijene potojeća rješenja, u dugom će roku ljudi nastaviti pronalaziti nova komercijalno i društveno korisna rješenja pomoću kriptovaluta.

Unatoč navedenim postignućima, i dalje postoje prepreke i izazovi s kojima će se kriptovalute u budućnosti morati suočiti. U radu su objašnjeni problemi skalabilnosti, neefikasne potrošnje resursa, decentralizacije, rukovođenja jedinica valute i ažuriranja softvera, no postoje i oni koji nisu isključivo tehničke prirode. Sigurnost i ostala svojstva koja pružaju kriptovalute su kombinacija tehničkih aspekata i povjerenja koje ljudi dodjeljuju kriptovalutama, čime se formira baza njihove vrijednosti. Dodatnu prijetnju predstavljaju regulacije i vladina ograničenja. Bitcoin možda nije odgovor na sve probleme, ali već sada ima velik utjecaj na brojna područja i različite zajednice. Kombinacija sastavnih tehnologija i metoda otvorila je nove mogućnosti u različitim domenama istraživanja. Neki su uzbuđeni oko bitcoina zbog njegove tenhologije, dok se drugi uzbuđeni oko njegovih komercijalnih mogućnosti, a treći zbog društvenih i političkih implikacija. Nakon Bitcoina se otvorio novi svijet alternativnih dizajnova kriptovaluta koji smo tek počeli istraživati, a od kojih će neki postati važniji i od Bitcoina. Teško da će se jedna od kriptovaluta probiti kao globalno prihvaćena valuta koju će svi koristiti, no moguće je da nekolicina uspješnih kriptovaluta koegzistira. Postavlja se pitanje, hoće li kriptovalute zamijeniti tradicionalne valute? Imaju potencijala da to učine, ali s obzirom na sve neriješene probleme, teško je to za očekivati u skoroj budućnosti. U svakom slučaju, bit će zanimljivo promatrati kompetenciju između kriptovaluta i valuta za pojedine segmente tržišta. Predviđanje budućnosti nije jednostavan zadatak, posebice kada je u pitanju nova tehnologija koja je predmet aktivnog istraživanja i daljnjeg razvoja. Doduše, mogu se izvesti pretpostavke i implikacije za moguće buduće scenarije. Na temelju podataka o kretanju vrijednosti bitcoina i njegovu razvoju, teško je predvidjeti hoće li njegova cijena dalje rasti ili padati. Ekonomski stručnjaci smatraju da će njegova cijena u budućnosti padati te da ne predstavlja dobru investicijsku priliku, da nema budućnost kao sigurno sredstvo čuvanja vrijednosti niti kao globalna valuta, no da postoji mo-

gućnost da se nametne kao pouzdan i jeftin način transfera novca. U svakom slučaju, neovisno o tome opstane li tržište kriptovaluta nadolazeće izazove, tehnologija koja stoji iza kriptovaluta zasigurno će pronaći široke primjene u različitim domenama ljudske djelatnosti. Kakva god da je budućnost kriptovaluta, možemo sudjelovati i svjedočiti uzbuđujućoj i izazovnoj tehnološkoj tranziciji koja ima potencijal da postane jedna od najvećih pojava od osnutka interneta.

Popis slika

1. Prilog: Riječnik pojmova

bitcoin adresa Bitcoin adresa izgleda ovako 1DSrFJDE2AHdE4ERLlaoEZC2m6443kdJafV. Sastoji se od niza slova i brojeva te počinje s brojem 1. Isto kao što bismo tražili druge da nam pošalju email na našu email adresu, tako bi ih i tražili da nam pošalju bitcoin na bitcoin adresu.

bitcoin Naziv jedinice valute, mreže i softvera.

blok (eng. *block*) Grupa transakcija, označenih vremenskom oznakom i otiskom prethodnog bloka. Zaglavlje bloka se kriptira kako bi se pronašao dokaz o radu (eng. *Proof of work*) i time validirale transakcije. Ispravni blokovi dodaju se u glavni blockchain mrežnim konsenzusom.

blockchain Lista validiranih blokova, gdje svaki pokazuje na svog prethodnika sve do prvog bloka.

konfirmacije (eng. *confirmations*) Nakon što je transakcija uvrštena u blok, ima dodatnu konfirmaciju. Čim je neki drugi blok izrudaren na istom blockchainu, transakcija ima dvije konfirmacije i tako dalje. Šest ili više konfirmacija smatrano je dovoljnim dokazom da se transakcija ne može povratiti.

kompleksnost (eng. *difficulty*) Postavka na razini mreže koja kontrolira koliko je komputacije potrebno da bi se pronašao dokaz o radu (eng. *Proof of work*).

cilj kompleksnosti (eng. *difficulty target*) Razina kompleksnosti na kojoj će sve komputacije u mreži pronaći blokove otprilike svakih 10 minuta.

ponovno ciljanje kompleksnosti (eng. *difficulty re-targeting*) Ponovno izračunavanje kompleksnosti na razini mreže koje se događa jednom svakih 2106 blokova.

naknada (eng. *fee*) Pošiljatelj transakcije često uključuje naknadu mreži za procesiranje njegove transakcije. Većina transakcija zahtjeva izuzetno malu naknadu.

hash Digitalni otisak nekog binarnog unosa.

hash funkcija (eng. *hash function*) Funkcija koja se koristi za računanje hasha. U Bitcoin protokolu, ta je funkcija SHA-256.

hash vrijednost (eng. *hash value*) Rezultirajući hash izlaz iz hash funkcije.

blok postanka (eng. *genesis block*) Prvi blok na blockchainu, korišten da se inicijalizira kriptovaluta.

rudar (eng. *miner*) Mrežni čvor koji pronalazi ispravne dokaze o radu za nove blokove ponavljajući proces hash-iranja.

mreža (eng. *network*) Čvor-čvor (eng. *peer-to-peer*) mreža koja propagira transakcije i blokove do svakog bitcoin čvora na mreži.

dokaz o radu (eng. *proof of work*) Dio podatka koji zahtjeva značajno procesiranje kako bi se pronašao. U bitcoinu, rudari moraju pronaći numeričko rješenja SHA256 algoritma koje odgovara cilju kompleksnosti na razini mreže.

nagrada (eng. *reward*) Iznos uključen u svakom novom bloku kao nagrada mreže rudaru koji je pronašao rješenje za dokaz o radu.

tajni ključ (eng. *secret key*) Tajni broj koji otključava bitcoine poslane pripadajućoj adresi.

transakcija (eng. *transaction*) Prijenos bitcoina s jedne adrese na drugu. Preciznije, transakcija je potpisana struktura podataka koja izražava prijenos vrijednosti. Transakcije se prenose preko bitcoin mreže, rudari ih prikupljaju te ih uključuju u blokove koji čine trajni blockchain.

digitalni novčanik (eng.*digital wallet*) Softver koji sadrži sve korisnikove bitcoin adrese i tajne ključeve. Koristi se za slanje, primanje i pohranu bitcoina.

distribuirana kriptovaluta (eng.*distribiuted cryptocurrency*) Distribuirana kriptografska valuta, odnosno kriptovaluta jest sustav digitalne imovine dizajniran da radi kao sredstvo razmjene koje koristi kriptografske primitive kako bi osigurao kontrolu i kreiranje jedinica valute.

Nakamoto konsenzus (eng. *Nakamoto consensus*) Pojam Nakamoto konsenzusa referencira temeljni mehanizam konsenzusa koji stoji iza bitcoina. Omogućava dinamički skup anonimnih sudionika u distribuiranom sustavu da dođu do uzajamnog dogovora uzimajući u obzir svojstva dokaza o radu i ekonomske faktore.

povjerljiv posrednik (eng. *trusted third party - TTP*) Povjerljiv posrednik označava potrebu za posrednikom između dvije strane. Posrednik mora biti povjerljiv kako bi dvije strane provele transakciju ili sigurno komunicirale.

virtualna valuta (eng. *virtual currency*) Centralna Europska Banka redefinirala je ovaj pojam 2014. godine, te je formalno "digitalna reprezentacija vrijednosti koju ne izdaje centralna banka ili neki drugi javni autoritet, niti je nužno vezana za fizičku valutu, ali je prihvaćena od strane fizičkih ili pravnih osoba kao sredstvo plaćanja te se može prenositi, spremati ili trgovati elektroničkim putem".

kandidatski blok (eng. *candidate block*) Kandidatski blok je nedovršeni blok, kreiran kao privremeni konstrukt rudara u svrhu pohranjivanja transakcija iz bazena nepotvrđenih transakcija. Postaje potpuni blok nakon što se doda zaglavlje sa završnim rješenjem dokaza o radu.