

UNIVERZITET U BEOGRADU  
MATEMATIČKI FAKULTET



Stefan M. Lazović

IMPLEMENTACIJA DECENTRALIZOVANOG  
TRŽIŠTA NEZAMENJIVIH TOKENA U  
INDUSTRIJI VIDEO IGARA

master rad

Beograd, 2024.

**Mentor:**

dr Saša MALKOV, vanredni profesor  
Univerzitet u Beogradu, Matematički fakultet

**Članovi komisije:**

dr Ivan ČUKIĆ, docent  
Univerzitet u Beogradu, Matematički fakultet

dr Stefan MIŠKOVIĆ, docent  
Univerzitet u Beogradu, Matematički fakultet

**Datum odbrane:** \_\_\_\_\_

*Zahvalan sam porodici koja bila tu uz mene tokom celog školovanja, devojci uz čiju podršku sam završio master rad i drugarima sa fakulteta bez kojih ništa ne bi bilo isto*

**Naslov master rada:** Implementacija decentralizovanog tržišta nezamenjivih tokena u industriji video igara

**Rezime:** Ovaj rad istražuje inovativne načine korišćenja blokčejn tehnologije u svrhe trgovine digitalnim dobrima unutar industrije video igara. Rad detaljno analizira koncepte, implementaciju i potencijalne koristi od primene pametnih ugovora i nezamenjivih tokena na blokčejnu Ethereum. Cilj rada je da demonstrira kako decentralizacija tržišta može poboljšati transparentnost, sigurnost i prava vlasništva, čime se otvaraju nove mogućnosti za inovacije i stvaranje globalne ekonomije unutar igara.

**Ključne reči:** tehnologija, pametni ugovori, nezamenjivi tokeni, Ethereum, blokčejn

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Blokčejn Ethereum i osnovni koncepti</b>	<b>4</b>
2.1	Ethereum Virtual Machine ( <i>EVM</i> )	4
2.2	Tokeni i standardi blokčejna Ethereum	6
2.3	Nezamenjivi tokeni	9
2.4	Programski jezik Solidity	12
<b>3</b>	<b>Motivacija i trenutno stanje tržišta</b>	<b>16</b>
3.1	Motivacija i potreba	16
3.2	Trenutno stanje tržišta	17
3.3	Potencijalne primene	21
<b>4</b>	<b>Konceptualno rešenje</b>	<b>23</b>
4.1	Konceptualno rešenje decentralizovanog tržišta	23
4.2	Arhitektura sistema	24
4.3	Životni ciklus nezamenjivog tokena	28
4.4	Implementacija i praktična primena arhitekture	30
<b>5</b>	<b>Implementacija i testiranje pametnih ugovora</b>	<b>31</b>
5.1	Ugovor za kreiranje tokena GOLD	31
5.2	Ugovor za kreiranje nezamenjivih tokena	32
5.3	Ugovor za kupovinu i prodaju tokena GOLD	33
5.4	Ugovor decentralizovanog tržišta nezamenjivih tokena	37
5.5	Implementacija testova za ugovore	40
5.6	Postavljanje ugovora na blokčejn i testiranje	47
<b>6</b>	<b>Diskusija i zaključak</b>	<b>51</b>

*SADRŽAJ*

---

6.1 Zaključak . . . . .	53
<b>Bibliografija</b>	<b>54</b>

# Glava 1

## Uvod

U svetu video igara, tržišta virtuelnih predmeta su postala neizostavan deo iskustva igrača. Igrači širom sveta troše značajne sume novca na kupovinu kostima (eng. *skin*), oružja, profilnih slika (eng. *avatar*) i drugih digitalnih dobara, stvarajući ekonomiju unutar igara koja cveta [1]. Međutim, ova tržišta su uglavnom centralizovana, što donosi sa sobom niz problema.

Jedan od glavnih problema centralizovanih tržišta je nedostatak transparentnosti. Igrači često nemaju uvid u detalje same transakcije, poput raspodele dobiti i visine provizije koju platforme uzimaju [2]. Pored toga, centralizovane baze podataka postaju primamljive mete za hakere, što može rezultirati krađom ili gubitkom vrednih digitalnih predmeta [3]. Još jedan značajan problem je neizvesnost trajanja vlasništva nad digitalnim predmetima. Na mnogim centralizovanim tržištima, igrači gube pristup svojim kupljenim predmetima ukoliko igra prestane sa radom ili platforma promeni pravila [4]. Mogućnosti za preprodaju ovih predmeta su često ograničene ili potpuno nepostojeće, što prisiljava igrače da koriste nesigurne metode preprodaje preko foruma ili tajnih grupa. Ova „crna tržišta” nose sa sobom dodatne rizike od prevara i gubitka sredstava [5].

Blokčejn (eng. *blockchain*) je tehnologija koja omogućava sigurno i transparentno čuvanje podataka putem distribuirane mreže računara. Kombinacijom kriptografskih tehnika i decentralizovane arhitekture, podaci unutar blokčejna ostaju sigurni i otporni na neovlašćene izmene, čineći ceo sistem pouzdanim i zaštićenim od hakovanja [6].

Ova tehnologija je osnova za nezamenjive tokene (eng. *Non-fungible token*, skraćeno *NFT*), koji su digitalna dobra zapisana na blokčejnu sa garantovanim vlasništvom i autentičnošću. Nezamenjivi tokeni su digitalna dobra koja koriste tehnolo-

logiju blokčejn za potvrdu jedinstvenosti i vlasništva. Za razliku od tradicionalnih digitalnih predmeta, koji se mogu kopirati bez ograničenja, svaki nezamenjivi token ima jedinstven identifikator koji ga čini nezamennjivim i individualno vrednim [7].

Nezamenjivi tokeni nude različite funkcionalnosti koje se mogu prilagoditi specifičnim potrebama svake industrije. Tri osnovna načina na koja se nezamenjivi tokeni mogu koristiti uključuju:

- **Zaključavanje resursa:** Nezamenjivi tokeni omogućavaju zaključavanje određenih digitalnih resursa, čime se osigurava da im se može pristupiti samo putem vlasništva nad odgovarajućim tokenom. Ovo je posebno korisno u industriji umetnosti i kolekcionarstva, gde vlasnici mogu kontrolisati pristup ekskluzivnim sadržajima.
- **Dokaz vlasništva ili autorstva:** Nezamenjivi tokeni pružaju nepromenljiv dokaz vlasništva ili autorstva nad digitalnim ili fizičkim dobrima. Ovo je od ključnog značaja u muzičkoj industriji i industriji umetnosti, gde se potvrda autentičnosti i vlasništva može obaviti na transparentan način putem blokčejna.
- **Licenciranje prava:** Nezamenjivi tokeni se mogu koristiti kao potvrda prenosivih ili neprenosivih prava na korišćenje određenog sadržaja ili resursa. Ova funkcionalnost je posebno važna u sektorima kao što su nekretnine, obrazovanje i video igre, gde licenciranje može omogućiti novi način distribucije i monetizacije sadržaja ili usluga.

Ove osnovne funkcionalnosti omogućavaju prilagođavanje tehnologije različitim industrijama, otvarajući vrata za inovacije u načinu na koji se upravlja i trguje digitalnim i fizičkim dobrima [8, 9].

U ovom kontekstu, tehnologija blokčejn nudi inovativna rešenja za mnoge probleme. Decentralizovana tržišta zasnovana na blokčejnu, kao što su ona koja koriste nezamenjive tokene, omogućavaju veću transparentnost, sigurnost i prava vlasništva. Na blokčejnu su sve transakcije javno dostupne, što igračima omogućava jasan uvid u sve trgovinske aktivnosti. Takođe, decentralizovana priroda blokčejna smanjuje rizik od hakerskih napada, jer podaci nisu smešteni na jednoj lokaciji. Jedna od ključnih prednosti korišćenja tehnologije blokčejn je stvarno vlasništvo nad digitalnim predmetima. Koristeći nezamenjive tokene, igrači mogu imati potpunu kontrolu



nad svojim virtuelnim dobrima, što im omogućava da slobodno trguju ovim predmetima bez bojazni da će im biti ukradeni ili da će im se nešto drugo desiti. Ova nova paradigma otvara vrata za globalnu ekonomiju unutar igara, bez ograničenja koja postavljaju centralizovane platforme [10].

Ovaj rad istražuje koncept i implementaciju decentralizovanog tržišta nezamenjivih tokena u industriji video igara koristeći blokčejn *Ethereum*. Kroz analizu i implementaciju, cilj je pokazati osnovne operacije decentralizovanih tržišta, uzimajući u obzir da svako tržište poseduje svoje specifične dodatne funkcionalnosti. Decentralizacija ne samo da može povećati poverenje i sigurnost, već i omogućava igračima da zaista poseduju i upravljaju svojim digitalnim dobrima, otvarajući nove mogućnosti za inovacije u svetu video igara.

## Glava 2

# Blokčejn Ethereum i osnovni koncepti

*Ethereum* je decentralizovana platforma koja omogućava izvršavanje pametnih ugovora i pokretanje decentralizovanih aplikacija (eng. *Decentralised application*, skraćeno *dApps*) bez potrebe za posrednikom. Platforma koristi kombinaciju tehnologija koje uključuju dokaz o radu (eng. *Proof of Work*) i dokaz o ulogu (eng. *Proof of Stake*, virtualna mašina *Ethereum* (eng. *Ethereum Virtual Machine*, skraćeno *EVM*), i poseban sistem za obračun transakcija pomoću gasa.

*Ethereum* je originalno koristio konsenzus mehanizam dokaz o radu, koji zahteva od rudara da reše složene matematičke probleme kako bi potvrdili transakcije i kreirali nove blokove. Međutim, zbog visoke potrošnje energije i težnje ka većoj efikasnosti, *Ethereum* prelazi na mehanizam dokaz o ulogu. Mehanizam dokaz o ulogu smanjuje potrošnju energije i povećava brzinu transakcija tako što dozvoljava korisnicima da „ulože” svoje tokene kao garanciju za pravo da validiraju transakcije [11].

### 2.1 Ethereum Virtual Machine (*EVM*)

*EVM* je srce *Ethereum*-a i on omogućava izvršavanje pametnih ugovora. *EVM* je izolovano kontrolisano okruženje (eng. *Sandbox*) unutar svakog *Ethereum* čvora, što znači da kod koji se izvršava unutar *EVM* ne može direktno komunicirati sa spoljnim svetom. Ovo pruža visok nivo sigurnosti i stabilnosti [12].

*EVM* koristi svoj vlastiti bajt-kod za izvršavanje instrukcija. Pametni ugovori se obično pišu u visokom programskom jeziku, kao što je *Solidity* (u slučaju *Ethereum*-

a), a zatim se kompajliraju u *EVM* bajt-kod, nakon čega se bajt-kod izvršava unutar *EVM* [13].

Jedna od ključnih karakteristika *EVM*-a je koncept *gas*-a. *Gas* je unutrašnja jedinica za merenje koliko resursa je potrebno za izvršenje određene operacije ili skupa operacija unutar *EVM*. Svaka operacija koja se izvršava unutar *EVM* ima određenu cenu izraženu u jedinicama *gas*-a, koja se koristi za izračunavanje ukupne cene transakcije. Ovo pomaže u sprečavanju zloupotrebe resursa mreže i doprinosi njenom stabilnom funkcionisanju [14].

Transakcije su osnovni mehanizam kojim se vrši interakcija sa mrežom *Ethereum*. Transakcije mogu biti jednostavne, kao što je prenos Ethera (kriptovalute *Ethereum*) od jednog naloga do drugog, ili složene, kao što je interakcija sa pametnim ugovorima [15].

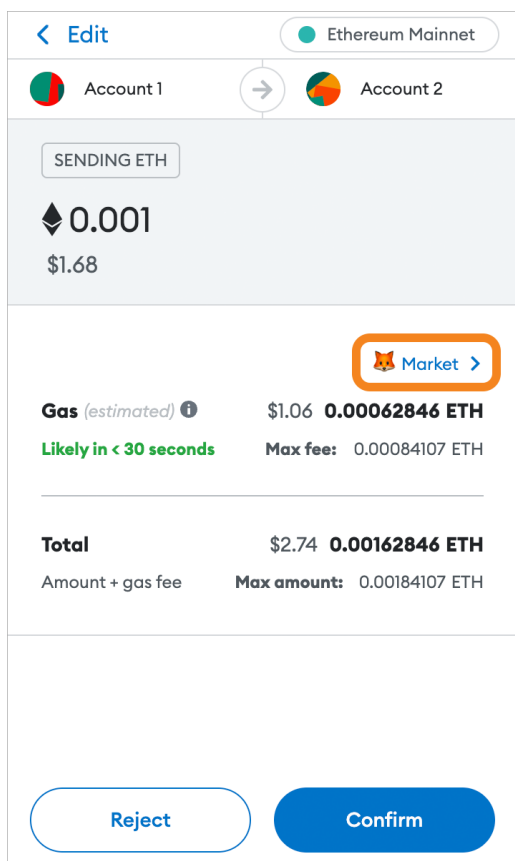
Kada korisnik želi da izvrši transakciju, prvo mora da formira transakcijski zahtev. Ovaj zahtev sadrži sledeće informacije:

- **Adresa primaoca** – adresa na *Ethereum* mreži koja pripada primaocu sredstava ili pametnom ugovoru sa kojim se intereaguje;
- **Količina Ethera** – iznos kriptovalute *Ethereum* koji se šalje;
- **Ograničenje gasa** – maksimalna količina gasa koju korisnik želi da potroši na transakciju;
- **Cena gasa** – količina Ethera koju korisnik želi da plati po jedinici gasa;
- **Opcioni podaci** – koriste se za interakciju sa pametnim ugovorima.

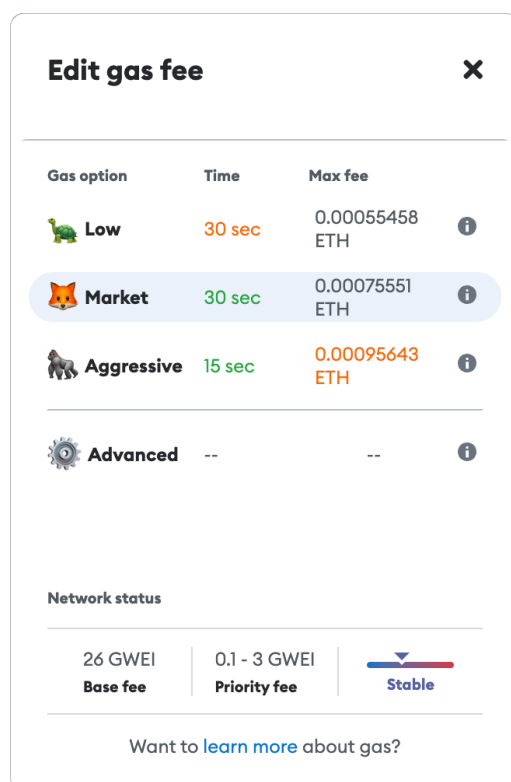
Nakon što je zahtev formiran, korisnik ga potpisuje svojim privatnim ključem. Potpisana transakcija se zatim šalje u *Ethereum* mrežu, gde je rudari (učesnici u mreži koji validiraju transakcije) preuzimaju i dodaju u blok. Rudari biraju koje transakcije će uključiti u blok na osnovu cene *gas*-a; transakcije sa višom cenom *gas*-a su privlačnije jer rudari dobijaju veću nagradu koja se obračunava na osnovu cene *gas*-a.

Većinu ovih informacija predlaže digitalni novčanik kroz koji korisnik komunicira sa bločkejnom, kako bi bio olakšan pristup ljudima koji su manje tehnički obrazovani, kao što se može videti na slikama 2.1 i 2.2.

Slika 2.1 prikazuje interfejs za podešavanje ograničenja *gas*-a, koji određuje maksimalnu količinu *gas*-a koja se može potrošiti za transakciju. Slika 2.2 ilustruje kako korisnici mogu podešavati cenu *gas*-a, što direktno utiče na brzinu procesiranja transakcije i troškove.



Slika 2.1: Prilagodavanje ograničenja gasa



Slika 2.2: Prilagodavanje cene gasa

## 2.2 Tokeni i standardi blokčejna Ethereum

Ovo poglavlje će pružiti dublje razumevanje o tome šta su digitalni tokeni, kako se kreiraju, i kako se koriste u kontekstu tehnologije blokčejn, fokusirajući se na digitalne tokene i standarde na blokčejnu *Ethereum* odnosno na nezamenjive tokene i standarde ERC-20 i ERC-721.

Standardi ERC-20 i ERC-721 definišu skup pravila koja pametni ugovori na *Ethereum* mreži moraju da prate kako bi implementirali zamenjive (eng. *fungible*) i nezamenjive (eng. *non-fungible*) tokene. Ovi standardi omogućavaju interoperabilnost između različitih pametnih ugovora i distribuiranih aplikacija na platformi *Ethereum* [16].

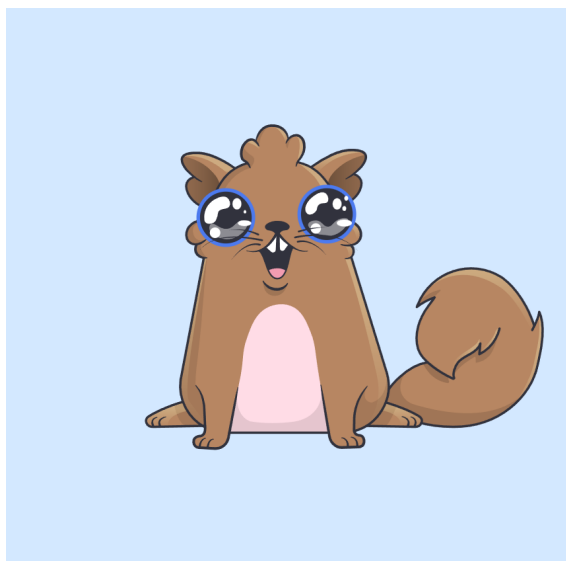
### Standard ERC-721

ERC-721 uvodi standard za nezamenjive tokene, gde svaki token predstavlja jedinstven digitalni sadržaj i samim tim ima svoju vrednost, iako svi potiču iz istog pametnog ugovora. Na primer, jedan pametni ugovor može kreirati različite tokene koji predstavljaju ulaznice za različita sedišta na koncertu, pri čemu svaka ulaznica ima svoju posebnu vrednost u zavisnosti od faktora kao što su lokacija sedišta, retkost ili vizuelni izgled. Svi nezamenjivi tokeni imaju promenljivu tipa `uint256` nazvanu `tokenId`, tako da za bilo koji ugovor ERC-721, parovi adresa ugovora i `tokenId`-ova moraju biti globalno jedinstveni. S tim u vezi, neka decentralizovana aplikacija može imati „konverter” koji koristi `tokenId` kao ulaz i generiše različite oblike metapodataka, poput slika ili drugih digitalnih reprezentacija, koji služe kao pouzdani dokaz vlasništva nad određenom digitalnom imovinom [17].

Kolekcija nezamenjivih tokena<sup>1</sup> i video igra *CryptoKitties* je primer aplikacije koja koristi standard ERC-721. *CryptoKitties* je igra na platformi *Ethereum* u kojoj igrači mogu da kupuju, prodaju, i uzgajaju virtuelne mačke. Svaka mačka je jedinstven token ERC-721, sa svojim jedinstvenim atributima i vrednostima. Na slici 2.3 se može videti primer kako izgleda vizuelno jedan nezamenjivi token iz te kolekcije.

---

<sup>1</sup>Kolekcija u kontekstu nezamenjivih tokena odnosi se na skup tokena koji su kreirani iz istog pametnog ugovora, gde svaki token ima jedinstvene karakteristike i metapodatke, ali svi pripadaju istoj grupi.



Slika 2.3: Crypto kitty nezamenjivi token

Standard ERC-721 dodaje dodatne funkcije za rad sa nezamenjivim tokenima. Ove funkcije uključuju:

- `ownerOf(uint256 _tokenId)`: Vraća vlasnika određenog tokena (`_tokenId`).
- `transferFrom(address _from, address _to, uint256 _tokenId)`: Prebacuje vlasništvo nad određenim tokenom (`_tokenId`) sa jedne adrese (`_from`) na drugu adresu (`_to`).
- `approve(address _approved, uint256 _tokenId)`: Dozvoljava određenoj adresi (`_approved`) da preuzme vlasništvo nad određenim tokenom (`_tokenId`).

## Standard ERC-20

ERC-20 je standard za zamenjive tokene koji imaju svojstvo da je svaki token identičan po tipu i vrednosti kao svaki drugi token (koji pripada istoj adresi ugovora). Primera radi, token ERC-20 se ponaša kao i Ether, što znači da je jedan token uvek jednak svim ostalim tokenima. Tokeni ERC-20 se često koriste za kreiranje novih kriptovaluta, za izdavanje tokena (eng. *Initial Coin Offerings*), za kreiranje tokena za decentralizovane aplikacije, i za mnoge druge svrhe [18].

Standard ERC-20 je definisan kroz skup funkcija koje omogućavaju interakciju sa tokenima unutar pametnih ugovora:

- `totalSupply()`: Vraća ukupan broj tokena.
- `balanceOf(address _owner)`: Vraća broj tokena koji određena adresa (`_owner`) poseduje.
- `transfer(address _to, uint256 _value)`: Prebacuje određeni broj tokena (`_value`) na određenu adresu (`_to`).
- `approve(address _spender, uint256 _value)`: Dozvoljava određenoj adresi (`_spender`) da povuče određeni broj tokena (`_value`).
- `allowance(address _owner, address _spender)`: Vraća broj tokena koje određena adresa (`_spender`) može da povuče sa druge adrese (`_owner`).

Ove funkcije omogućavaju osnovne operacije sa tokenima, kao što su transfer tokena i dozvola za povlačenje tokena. Takođe, omogućavaju dohvatanje informacija o tokenima, kao što su ukupan broj tokena koji se može naći u cirkulaciji (tokeni koji su u opticaju, privremeno blokirani ili rezervisani) i broj tokena koje određena adresa poseduje.

### 2.3 Nezamenjivi tokeni

Nezamenjivi tokeni su jedinstveni digitalni tokeni koji predstavljaju vlasništvo nad različitim vrstama digitalnih sredstava, uključujući umetnička dela, kolekcionarske predmete, virtuelne predmete u igrama [19]. Svaki nezamenjivi token je jedinstven i ne može se zameniti za drugi, što ga čini digitalnim ekvivalentom kolekcionarskog predmeta. Ova jedinstvenost je ono što daje vrednost nezamenjivim tokenima, jer svaki može da predstavlja vlasništvo nad jedinstvenim digitalnim resursom.

Nezamenjivi tokeni su kodirani unutar pametnih ugovora na blokčejnu, što znači da se svaka transakcija koja uključuje nezamenjivi token beleži na blokčejnu. Ovo omogućava neporecivu i nepromenljivu evidenciju o vlasništvu i transakcijama, što je jedan od ključnih aspekata koji čine nezamenjive tokene atraktivnim za korisnike.

Nezamenjivi tokeni su prvi put postali popularni 2021. godine, kada je njihovo tržište doživelo eksplozivni rast. Iako je prihvatanje ove nove forme digitalnog vlasništva još uvek relativno sporo, potencijal tehnologije blokčejn je ogroman. Nezamenjivi tokeni su još uvek u velikoj meri neshvaćeni od strane šire javnosti, što pruža sjajnu priliku za dalje istraživanje ovog tržišta.

Nezamenjivim tokenima se trguje na različitim decentralizovanim tržištima, često uz korišćenje kriptovaluta kao sredstva razmene. Ova tržišta omogućavaju korisnicima da kupuju i prodaju nezamenjive tokene, kao i da prate njihovu istoriju vlasništva i promene u ceni. Svaka transakcija je zabeležena na blokčejnu, što omogućava jasnu istoriju vlasništva svakog nezamenjivog tokena.

Nezamenjivi tokeni se stvaraju kroz proces poznat kao kreiranje tokena (eng. *minting*). Ovaj proces uključuje kreiranje i postavljanje pametnog ugovora na blokčejnu koji sadrži informacije o nezamenjivom tokenu, uključujući njegovu jedinstvenost i vlasništvo, pomoću kog se zatim kreiraju konkretni tokeni.

Svaki nezamenjivi token može da sadrži metapodatke koji definišu informacije poput imena, opisa, slike, i drugih atributa digitalnog sredstva. Kada se nezamenjivi token kreira, on je inicijalno dodeljen kreatoru. Međutim, nezamenjivi token može biti prenet na druge korisnike kroz transakcije na blokčejnu.

Dodatno, pametni ugovori koji upravljaju nezamenjivim tokenima često sadrže mehanizme za isplatu autorskih naknada (eng. *royalties*), koji omogućavaju originalnim kreatorima da ostvare prihod od svake sledeće trgovine njihovih tokena. Ovi mehanizmi su fleksibilni i omogućavaju kreatorima da sami odrede procentualni udeo od prodaje, koji obično varira između 5% i 10%. Ovo ne samo da osigurava kontinuirani prihod za kreatore, već i štiti i promovise autorska prava u digitalnom svetu [20]. Primera radi, kompanija *Yuga Labs*, koja stoji iza kolekcije *Bored Ape Yacht Club*, izdvojila se kao jedan od najvećih primalaca autorskih naknada od trgovine nezamenjivih tokena koje su oni napravili, zaradivši preko 147 miliona dolara [21].

Metapodatke nezamenjivog tokena i povezane digitalne sadržaje, kao što su slike, često nije praktično ili efikasno čuvati neposredno na blokčejnu zbog ograničenja veličine i troškova transakcija. Umesto toga, ovi podaci se obično adresiraju koristeći protokol *IPFS* (eng. *InterPlanetary File System*) uz pomoć dodatnih servisa za trajno skladištenje, poznatih kao *pinning* servisi<sup>2</sup>, *IPFS* koristi mrežu ravnoprav-

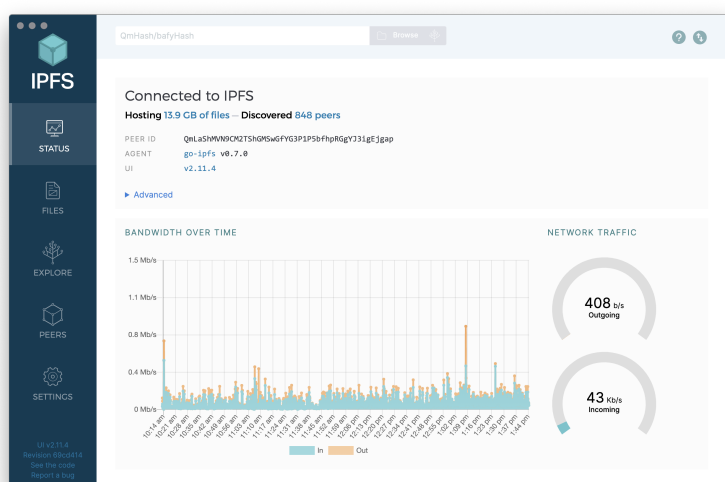
---

<sup>2</sup>*Pinning* servisi omogućavaju trajno skladištenje podataka na *IPFS*-u. Oni osiguravaju da sadržaj ostane dostupan na mreži, tako što ga čuvaju na svojim čvorovima i sprečavaju njegovo brisanje iz mreže zbog neaktivnosti.



nih čvorova (eng. *peer-to-peer*) za deljenje podataka, čime se postiže otpornost na cenzuru i veća brzina pristupa podacima. Na slici 2.4 se može videti kako izgleda interfejs aplikacije *IPFS* na desktop računaru [22].

Kada se kreira nezamenjivi token na blokčejnu, njegovi metapodaci, uključujući informacije o digitalnom sadržaju poput slika, adresiraju se koristeći jedinstveni *Content Identifier* (skraćeno *CID*) na *IPFS*-u. *CID* tog fajla se zatim povezuje sa nezamenjivim tokenom putem pametnog ugovora, osiguravajući da svaki nezamenjivi token ima neporecivu i trajnu referencu na svoj digitalni sadržaj. Važno je napomenuti da se digitalnom sadržaju na *IPFS*-u može pristupiti direktno putem njegovog *CID*-a, bez potrebe za posedom odgovarajućeg nezamenjivog tokena. Nezamenjivi token služi kao dokaz vlasništva i povezanosti sa sadržajem, ali ne ograničava direktan pristup sadržaju na mreži *IPFS* [23].



Slika 2.4: IPFS - početni ekran [24]

Vrednost nezamenjivog tokena je uglavnom subjektivna i određuje se na osnovu različitih faktora, poput jedinstvenosti digitalnog sredstva, reputacije kreatora, i tržišne potražnje. Nezamenjivi tokeni omogućavaju digitalnim sredstvima da imaju jedinstvenu vrednost i da se njima trguje na sličan način kao i fizičkim kolekcionarskim predmetima [25].

Kada kreator kreira nezamenjivi token, on ostaje vlasnik tog tokena sve dok ne odluči da ga proda. Međutim, u slučaju većine kolekcija nezamenjivih tokena, praksa je da se nezamenjivi tokeni nasumično dodeljuju korisnicima koji učestvuju u procesu njihovog kreiranja. Tokom kreiranja i dodele vlasništva nad tokenima, svi tokeni u

kolekciji imaju identičnu cenu, a kupci nemaju informacije o karakteristikama ili metapodacima tokena koji će dobiti. Tek nakon što se proces kreiranja i dodele tokena završi i otkriju se sve informacije, kupci saznaju koji tačno token su dobili, uključujući njegove atribute, poreklo i druge karakteristike. Ovaj pristup osigurava fer distribuciju retkih i manje retkih nezamenjivih tokena među učesnicima, zato što niko unapred ne zna koji će token dobiti, što može značajno uticati na njihovu tržišnu vrednost. Ipak, kada kreator želi da zadrži vlasništvo nad svojim digitalnim sredstvom dok mu se vrednost ne odredi na tržištu, kreator ostaje vlasnik sve dok samostalno ne odluči da ga proda [26].

## 2.4 Programski jezik Solidity

*Solidity* je programski jezik visokog nivoa, posebno dizajniran za kreiranje pametnih ugovora na blokčejnu *Ethereum*. Ovaj jezik omogućava programerima da pišu ugovore koji definišu pravila i ponašanja unutar decentralizovanih aplikacija. *Solidity* je sintaksno sličan *JavaScript*-u, koristi statičko tipiziranje kao *C++*, a zadržava jednostavnost i čitljivost iz *Python*-a., i omogućava programerima da razvijaju digitalne tokene na mreži *Ethereum*, upravljaju vlasničkim pravima i grade složene poslovne logike koje se izvode na sigurnoj i transparentnoj mreži.

Pametni ugovori napisani u *Solidity*-ju se koriste za automatsko izvršavanje ugovora ili transakcija kada su ispunjeni određeni uslovi, bez potrebe za posrednicima. Ova svojstva čine *Solidity* ključnim alatom za razvoj decentralizovanih aplikacija koje zahtevaju visoku sigurnost i pouzdanost [27].

U nastavku će biti objašnjeni ključni pojmovi i funkcije programskog jezika *Solidity*, koristeći informacije iz zvanične dokumentacije [27].

### Osnovni koncepti jezika Solidity

U programskom jeziku *Solidity* postoje različiti **tipovi podataka** (eng. *data types*) koji se mogu koristiti za definisanje promenljivih unutar pametnih ugovora. Osnovni tipovi podataka uključuju celobrojne, decimalne, logičke, kao i niske i bajtove. Pored ovih osnovnih tipova, *Solidity* omogućava korišćenje složenih tipova podataka kao što su strukture, nizovi i mape, koje se često koriste za organizaciju podataka unutar ugovora.

*Solidity* koristi **modifikatore vidljivosti** (eng. *function Visibility specifiers*) kako bi kontrolisao pristup funkcijama. Dostupni modifikatori su: **public**, **private**, **internal** i **external**. Svaki od ovih modifikatora ima specifičnu ulogu:

- **public**: Funkcija je dostupna svima, kako unutar ugovora, tako i spolja.
- **private**: Ograničava pristup samo na ugovor u kojem je definisana; druge funkcije ili ugovori ne mogu pristupiti.
- **internal**: Omogućava pristup unutar ugovora i ugovora koji nasleđuju taj ugovor.
- **external**: Dozvoljava pristup funkcijama samo spoljnim ugovorima ili korisnicima; funkcije s ovim modifikatorom ne mogu se pozivati unutar istog ugovora.

*Solidity* omogućava definisanje različitih vrsta funkcija kada se radi o promeni stanja (eng. *State Mutability*), uključujući one sa modifikatorima **view** i **pure**, koje ne menjaju stanje ugovora, kao i funkcije sa modifikatorom **payable**, koje omogućavaju primanje kriptovalute Ether.

- **view**: Omogućava čitanje stanja ugovora bez promene tog stanja.
- **pure**: Funkcije sa ovim modifikatorom ne pristupaju stanju ugovora niti ga menjaju već samo funkcionišu na osnovu ulaznih parametara.
- **payable**: Ključan modifikator za funkcije koje omogućavaju primanje kriptovalute Ether unutar ugovora.

Pored toga, **require** se koristi za validaciju uslova unutar funkcija, dok se **emit** koristi za emitovanje događaja koji se mogu pratiti na blokčejnu [28].

### Pisanje pametnih ugovora

U programskom jeziku *Solidity*, **pametni ugovor** se definiše kao klasa koja sadrži promenljive, funkcije i događaje. Struktura ugovora obično započinje sa deklaracijom verzije kompajlera, nakon čega sledi definicija samog ugovora. Unutar ugovora se definišu stanja (eng. *state variables*) koja predstavljaju podatke ugovora, funkcije koje implementiraju logiku ugovora, kao i događaji (eng. *events*) koji omogućavaju praćenje određenih akcija na blokčejnu.

Funkcije za kreiranje tokena su ključne u pametnim ugovorima koji implementiraju digitalne tokene i postoji nekoliko varijacija ovih funkcija:

- `mint`: Ova funkcija se koristi za kreiranje novih tokena i dodeljivanje vlasništva nad njima određenoj adresi.
- `_mint`: Ova funkcija je interna varijanta `mint` funkcije, obično korišćena za implementaciju logike unutar ugovora bez izlaganja te funkcionalnosti spoljnim korisnicima.
- `safeMint`: Ova funkcija je slična `mint` funkciji, ali dodatno proverava uslove za sigurnost pre nego što kreira nove tokene i na taj način osigurava da kreirani tokeni ne naruše integritet ugovora.

U programskom jeziku *Solidity*, **promenljive** mogu biti statičke ili instancirane. Statičke promenljive dele vrednost između svih instanci ugovora, dok instancirane imaju specifičnu vrednost unutar određenog ugovora [28].

Kada se govori o upravljanju paralelnim procesima, kao što su **atomičnost operacija i trke za resursima**, važno je razmotriti kako programski jezik podržava ove aspekte u kontekstu bezbednosti i pouzdanosti. *Solidity* kao programski jezik nema ugrađene mehanizme za upravljanje paralelnim izvršavanjem koda (eng. *thread-safety*). Ovo ograničenje proizlazi iz dizajna *Virtuelne mašine Ethereum (EVM)*, koji ne podržava paralelno izvršavanje transakcija, već sve transakcije obrađuje sekvencijalno. Zbog ove sekvencijalne obrade, transakcije su zaštićene od konflikata koji bi mogli nastati prilikom istovremenog izvršavanja. Na primer, operacije poput inkrementiranja vrednosti su atomične i pouzdane zbog jedinstvenog načina na koji mreža obrađuje transakcije [29].

Kod u nastavku predstavlja jednostavan pametni ugovor napisan u programskom jeziku *Solidity* i omogućava postavljanje i čitanje jednog celobrojnog podatka `storedData` [27].

```
1 // SPDX-License-Identifier: GPL-3.0
2 pragma solidity >=0.4.16 <0.9.0;
3
4 contract SimpleStorage {
5     uint storedData;
6
7     function set(uint x) public {
8         storedData = x;
9     }
10
11    function get() public view returns (uint) {
12        return storedData;
13    }
14 }
```

# Glava 3

## Motivacija i trenutno stanje tržišta

### 3.1 Motivacija i potreba

U savremenom digitalnom dobu, decentralizovane tehnologije su imale znatni uticaj na različite industrije, uključujući i industriju video igara. Postoje brojni motivi za implementaciju decentralizovanog tržišta nezamenjivih tokena unutar industrije video igara, među kojima su:

- **Transparentnost transakcija:** Decentralizovana priroda blokčejna omogućava potpunu transparentnost, pružajući uvid u svaku transakciju što povećava poverenje među korisnicima i smanjuje šansu za potencijalne prevare.
- **Sigurnost digitalnih dobara:** Korišćenjem pametnih ugovora, prava vlasništva nad digitalnim dobrima se sigurno i efikasno prenose. Takav pristup značajno smanjuje rizike od digitalnih krađa i napada.
- **Dostupnost tržišta:** Decentralizovana tržišta omogućuju igračima globalni pristup, nezavisno od geografskog položaja i pristupa tradicionalnim finansijskim uslugama, čime se promoviše veća pristupačnost i ravnopravnost.
- **Inovacije u ekonomskim modelima:** Nezamenjivi tokeni omogućavaju razvoj novih modela zarade unutar igara, poput koncepta „zaradi igrajući” (eng. *play-to-earn*<sup>1</sup>), koji može značajno povećati angažovanost i lojalnost igrača.

---

<sup>1</sup>*Play-to-earn* predstavlja model u video igrama gde igrači mogu sticati stvarne ekonomske vrednosti, poput kriptovaluta ili nezamenjivih tokena, kroz igranje igre, doprinoseći ekonomiji igre i učestvujući u različitim aktivnostima unutar igre [30].

- **Personalizacija tokena:** Trgovina nezamenjivim tokenima omogućava igračima da steknu, prodaju ili zamene unikatne predmete, dodatno personalizujući iskustvo igranja i povećavajući tržišnu vrednost virtuelnog dobra.

Implementacija decentralizovanog tržišta nezamenjivih tokena ne samo da poboljšava transparentnost i sigurnost, već uvodi i potpuno novi način razmišljanja o vrednosti i vlasništvu u digitalnom svetu. Ove promene duboko transformišu način na koji korisnici interaguju unutar digitalnih svetova, otvarajući nove mogućnosti za razvoj igara i njihovih ekosistema.

## 3.2 Trenutno stanje tržišta

*OpenSea* je jedno od najpoznatijih tržišta nezamenjivih tokena (nekada i vodeće), koje omogućava korisnicima kreiranje, kupovinu i prodaju nezamenjivih tokena. Platforma je kompatibilna sa 12 različitih implementacija blokčejna, neke od njih su *Ethereum*, *Polygon* i *Avalanche* [31]. Dnevni obim trgovine na tržištu *OpenSea* često prelazi nekoliko miliona dolara, a mesečni obim trgovine dostiže desetine miliona dolara.

- **Prednosti:** velika korisnička baza, velika likvidnost, raznovrsnost podržanih kolekcija nezamenjivih tokena;
- **Nedostaci:** visoke transakcione naknade (2.5%), problemi sa skalabilnošću zbog visokih troškova *gas*-a na *Ethereum* mreži i neretkim zagušenjima mreže [32].

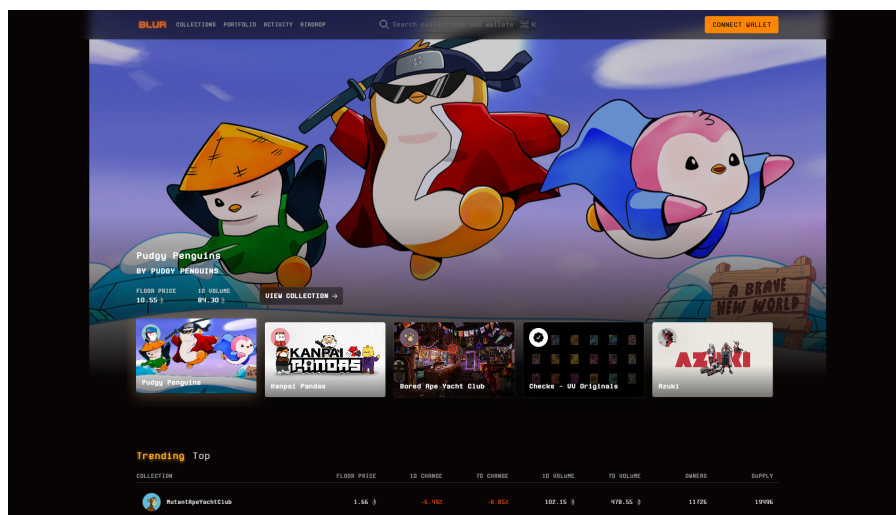
*Magic Eden* je jedno od vodećih tržišta nezamenjivih tokena na blokčejnu *Solana*. Platforma se ističe niskim transakcionim naknadama i brzim transakcijama zahvaljujući efikasnosti mreže *Solana*. Dnevni i mesečni obim trgovine su niži u poređenju sa *OpenSea*, ali *Magic Eden* ima prednost u jeftinijim i bržim transakcijama pa se spekuliše da može da prestigne *OpenSea*, što umnogome zavisi od daljeg razvoja blokčejna *Solana* i zajednice koja ga okružuje.

- **Prednosti:** niske naknade, brze transakcije, rastuća korisnička baza;
- **Nedostaci:** manja likvidnost u poređenju sa tržištem *OpenSea*, samim tim i manja raznovrsnost podržanih kolekcija nezamenjivih tokena [32].

### GLAVA 3. MOTIVACIJA I TRENUTNO STANJE TRŽIŠTA

**BLUR** je trenutno vodeće tržište nezamenjivih tokena na blokčejnu *Ethereum* koje se ističe svojim inovacijama u nagrađivanju trgovaca i kolekcionara nezamenjivih tokena. Ova platforma se fokusira na povećanje likvidnosti i unapređenje korisničkog iskustva. **BLUR** privlači korisnike svojim jedinstvenim sistemom aukcija i analitičkim alatima koji pomažu u boljem donošenju odluka pri trgovanju. Na slici 3.1 se može videti kako izgleda interfejs ove platforme.

- **Prednosti:** napredne analitičke alatke, podsticanje likvidnosti, nagrade za korisnike;
- **Nedostaci:** i dalje postoje visoki troškovi gasa, izazovi u održavanju privatnosti korisnika [32].



Slika 3.1: Početna stranica tržišta BLUR

U tabeli 3.1 se može videti poređenje ključnih karakteristika kada se radi o ova tri decentralizovana tržišta.

Karakteristika	Blur	Magic Eden	OpenSea
Mesečni obim trgovine	\$387M	\$184M	\$107M
Podržani blokčejnovi	<i>Ethereum, Optimism, Blast</i>	<i>Solana, Ethereum, Bitcoin, ...</i>	<i>Ethereum, Polygon, Arbitrum, ...</i>
Napredne funkcije	Alati za trgovanje, nagrađivanje trgovaca	Podrška za <i>Ordinals</i> <sup>2</sup>	Podrška mnoge blokčejnove
Naknade	0% za trgovce, 0.5% za kreatore	2% za trgovce	2.5% za trgovce

Tabela 3.1: Poređenje NFT platformi [33]



## Identifikacija problema

**Visoke transakcione naknade:** Jedan od glavnih problema sa trenutnim tržištima nezamenjivih tokena odnosno onima baziranim na mreži *Ethereum*, su visoke transakcione naknade odnosno cene *gas*-a. Korisnici često plaćaju značajne iznose u naknadama prilikom kreiranja, kupovine i prodaje nezamenjivih tokena, što može obeshrabriti manje korisnike i smanjiti likvidnost tržišta. Na primer, tokom perioda visokog opterećenja mreže, naknade za transakcije odnosno količina *gas*a na mreži *Ethereum* mogu dostići nekoliko stotina dolara po transakciji.

**Problemi sa skalabilnošću:** Skalabilnost je veliki izazov za većinu blokčejn mreža koje podržavaju tržišta nezamenjivih tokena. *Ethereum*, kao najkorišćenija mreža za nezamenjive tokene, često pati od zagušenja mreže, što rezultira visokim naknadama i sporim transakcijama. Ograničenja u propusnosti mreže negativno utiču na korisničko iskustvo i smanjuju ukupnu efikasnost tržišta.

**Sigurnosni problemi:** Tržišta nezamenjivih tokena često su meta sigurnosnih napada, uključujući napade pecanja (eng. *phishing*), prevare i zloupotrebe funkcionalnosti. Na primer, iako je *OpenSea* bio u centru nekoliko značajnih sigurnosnih incidenata, problemi nisu bili uzrokovani slabostima blokčejna, već propustima platforme i timova zaduženih za razvoj i održavanje pametnih ugovora. Stare verzije pametnih ugovora i nedovoljno efikasna migracija na novije verzije dovele su do ranjivosti koje su napadači iskoristili. Ovi incidenti naglašavaju potrebu za poboljšanom sigurnošću i proaktivnim održavanjem infrastrukture, što je ključno za dalji rast i prihvatanje tržišta nezamenjivih tokena [34, 35]. Kako bi se smanjila učestalost ovakvih napada, *OpenSea*, kao i ostatak kripto zajednice, aktivno rade na edukaciji korisnika, osnažujući ih da prepoznaju i izbegnu napade pecanja i slične prevare u kojima napadači imitiraju legitimne entitete.

*Ethereum* se izdvaja kao pionir u domenu nezamenjivih tokena zahvaljujući svojim naprednim pametnim ugovorima, što omogućava realizaciju složenih projekata. Međutim, iako *Ethereum* nudi bogat ekosistem alata i široko prihvaćenu platformu, suočava se sa značajnim problemima u pogledu visokih troškova *gas*a i sporijih transakcija tokom perioda velike potražnje. Ovi problemi postavljaju ograničenja na skalabilnost sistema i utiču na efikasnost.

S druge strane, *Solana* nudi izuzetno brzu obradu transakcija i niske troškove, što je čini privlačnom alternativom za one koji traže efikasnost. Uprkos svojoj rastućoj popularnosti i podršci za raznovrsne projekte koji se tiču nezamenjivih tokena, *Solana* još uvek nije dostigla veličinu ekosistema koju poseduje *Ethereum* i povre-

meno se suočava sa problemima zagušenja mreže. Ova brzina i ekonomičnost čine Solanu privlačnom opcijom, ali i postavljaju pitanja o njenoj dugoročnoj održivosti i sigurnosti u poređenju sa mrežom *Ethereum*, koja i dalje drži prvo mesto kada je u pitanju podrška zajednice i sigurnost.

### Potencijal za unapređenje

**Smanjenje transakcionih naknada:** Korišćenjem alternativnih blokčejn mreža sa nižim transakcionim naknadama, kao što su *Solana* ili *Polygon*, može se značajno smanjiti trošak transakcija za korisnike. Uvođenjem optimizovanih pametnih ugovora koji smanjuju potrebnu količinu gasa može se dodatno smanjiti trošak transakcija na mrežama kao što je *Ethereum*.

**Poboljšanje skalabilnosti:** Implementacija rešenja drugog sloja kao što su *zk-Rollups* ili *Optimistic Rollups* može pomoći u povećanju skalabilnosti *Ethereum* mreže.

*zk-Rollups* obavlja veći deo obrade van osnovne mreže, koristeći složene matematičke dokaze za verifikaciju. Ova metoda obezbeđuje visok nivo privatnosti i sigurnosti bez zagušenja centralne mreže.

*Optimistic Rollups* funkcioniše na pretpostavci da su sve transakcije ispravne osim ako ne dođe do konkretnog osporavanja. Ovaj pristup uvodi potrebu za određenim periodom čekanja kako bi se omogućilo vreme za eventualne prigovore.

Upotreba *Sharding* tehnologije i prelazak na *Ethereum 2.0* može dodatno poboljšati propusnost mreže i smanjiti zagušenja. *Sharding* je tehnika koja deli mrežu na manje delove, poznate kao *shards*. Svaki *shard* može procesuirati svoje transakcije i pametne ugovore, što znači da više *shard*-ova može raditi paralelno, umesto da cela mreža radi na jednoj velikoj sekvencijalnoj operaciji. Ovo drastično povećava kapacitet mreže i smanjuje opterećenje na svakom čvoru, omogućavajući veći broj transakcija po sekundi [36].

**Unapređenje sigurnosti:** Razvijanje naprednijih sigurnosnih protokola i edukacija korisnika može pomoći u smanjenju rizika od napada pecanjem. Uvođenje dvofaktorske autentifikacije (skraćeno *2FA*<sup>3</sup>) i poboljšanih verifikacionih procedura može povećati sigurnost korisnika i transakcija.

---

<sup>3</sup>Dvofaktorska autentifikacija (2FA) predstavlja dodatni sloj sigurnosti koji zahteva ne samo korisničko ime i lozinku, već i nešto što samo korisnik poseduje, kao što je mobilni telefon ili drugi lični uređaj, na koji se šalje kod za potvrdu identiteta prilikom prijavljivanja.

### 3.3 Potencijalne primene

Ove tehnologije pružaju brojne mogućnosti za inovacije ne samo u industriji video igara, već i u mnogim drugim oblastima. Njihova primena može biti raznovrsna:

1. **Industrija umetnosti i kolekcionarstva:** Blokčejn i nezamenjivi tokeni menjaju tržište umetnina, omogućavajući umetnicima da digitalno potvrde autentičnost i vlasništvo nad svojim delima, što takođe otvara vrata za nova tržišta i načine trgovine umetničkim delima [37].
2. **Muzička industrija:** Muzičari mogu koristiti nezamenjive tokene za prodaju ekskluzivnih prava na svoje pesme, albume ili ekskluzivne sadržaje vezane za njihova dela, omogućavajući sebi direktan ali i transparentan prihod od svojih fanova [37].
3. **Sektor nekretnina:** Nezamenjivi tokeni i blokčejn mogu se koristiti za tokenizaciju nekretnina, što pojednostavljuje procese kupoprodaje, iznajmljivanja i upravljanja imovinom, pružajući transparentnost i efikasnost u transakcijama koje su inače složene i birokratski umeju biti veoma otežane [37].
4. **Obrazovanje i sertifikacija:** Institucije mogu koristiti blokčejn za izdavanje verifikovanih digitalnih diploma i sertifikata koji su neizbrisivi i lako proverljivi, što smanjuje mogućnost prevara [37].
5. **Industrija video igara:** Osim u tradicionalnim video igrama, nezamenjivi tokeni i tehnologije blokčejn se mogu koristiti za stvaranje potpuno novih ekonomskih modela unutar igara, gde igrači mogu trgovati, iznajmljivati ili prodavati svoje digitalne predmete i likove unutar ili izvan igre, kreirajući snažnu zajednicu i ekosistem [37].

Svaka od ovih primena ne samo da demonstrira širinu mogućnosti koje tehnologija donosi, već i potencijal za stvaranje potpuno novih ekonomskih i društvenih struktura zasnovanih na transparentnosti, sigurnosti i efikasnosti koje pruža tehnologija blokčejn.

## Razlika između trgovine resursima i trgovine prava kod nezamenjivih tokena

Kada se razmatra trgovina nezamenjivim tokenima, važno je razlikovati trgovinu koja uključuje resurse, kao što su digitalna umetnička dela ili kolekcionarski predmeti, od trgovine koja se fokusira na prava, kao što su licence za korišćenje sadržaja. U slučaju licenciranja, nezamenjivi tokeni ne predstavljaju direktno vlasništvo nad resursima, već služe kao potvrda prava na njihovo korišćenje, što može uključivati prenosiva ili neprenosiva prava. Ova vrsta trgovine može uključivati dodatne pravne i tehničke izazove, jer se ne trguje resursima direktno, već pravima vezanim za te resurse [9].

# Glava 4

## Konceptualno rešenje

Glavni elementi decentralizovanog tržišta su pametni ugovori koji omogućavaju automatizaciju i sigurnost transakcija, i njihova integracija sa delovima sistema neophodnim za funkcionalnost tržišta.

### 4.1 Konceptualno rešenje decentralizovanog tržišta

Decentralizovano tržište nezamenjivih tokena na blokčejnu *Ethereum* predstavlja inovativni koncept koji koristi tehnologiju distribuiranih registara u industriji video igara.

Ovo tržište je zasnovano na pametnim ugovorima (konkretno na standardima ERC-20 i ERC-721) koji automatizuju i osiguravaju sve transakcije, eliminišući potrebu za posrednicima. Korišćenje pametnih ugovora doprinosi smanjenju troškova transakcija i povećanju efikasnosti, pružajući korisnicima pouzdanu platformu za trgovinu nezamenjivim tokenima.

Jezgro sistema čine dva tipa tokena: ERC-20 i ERC-721. Tokeni GOLD, koji su zasnovani na standardu ERC-20, služe kao valuta pomoću koje korisnici intereaguju sa tržištem. Svaki nezamenjivi token predstavljen je jedinstvenim tokenom ERC-721, i omogućava jasnu verifikaciju vlasništva i autentičnosti digitalnih predmeta, a samim tim ima i odgovarajuću tržišnu vrednost. Integracija sa blokčejnom *Ethereum* omogućava da se svaka transakcija zabeleži na transparentan i nepromenljiv način, što dodatno jača poverenje među korisnicima. Digitalni novčanici olakšavaju upravljanje i skladištenje tokena i nezamenjivih tokena, pružajući korisnicima jednostavan i siguran način da kontrolišu svoja digitalna dobra.

Ovaj konceptualni model ne samo da potencira inovacije u načinu trgovine digitalnim dobrima već i promovise nove ekonomske modele koji mogu biti od značajne koristi kako za pojedince, tako i za širu zajednicu igrača.

Za razvoj, implementaciju i lokalno testiranje pametnih ugovora, koristi se razvojno okruženje *Hardhat*, specijalizovano za *Ethereum*, koje definiše i okvir za arhitekturu sistema. *Hardhat* omogućava lako podizanje lokalnih blokčejn čvorova, kompajliranje, testiranje i razvijanje pametnih ugovora unutar izolovanog okruženja [38]. Ovo okruženje je neophodno za efikasno upravljanje projektom i testiranje funkcionalnosti pre postavljanja na javnu mrežu, osiguravajući da su svi procesi unutar tržišta konfigurisani i bezbedni.

*Hardhat* pruža podršku za napredne funkcije poput automatskog postavljanja ugovora i interakcije sa mrežom *Ethereum*, što značajno pojednostavljuje proces razvoja. Korišćenje ovog alata može efikasno simulirati ponašanje korisnika i interakcije unutar tržišta, što omogućava detaljno testiranje i verifikaciju svih aspekata decentralizovanog tržišta nezamenjivih tokena.

Ovaj pristup ne samo da ubrzava razvoj i implementaciju, već i jača sigurnost i pouzdanost sistema, sto ga čini idealnim za aplikacije koje zahtevaju visoku transparentnost i nepromenljivost, kao što su tržišta digitalnih dobara u industriji video igara.

## 4.2 Arhitektura sistema

Dijagram na slici 4.1 prikazuje arhitekturu decentralizovanog tržišta za video igre zasnovanog na blokčejnu *Ethereum*. Sistem se sastoji od više ključnih komponenti.

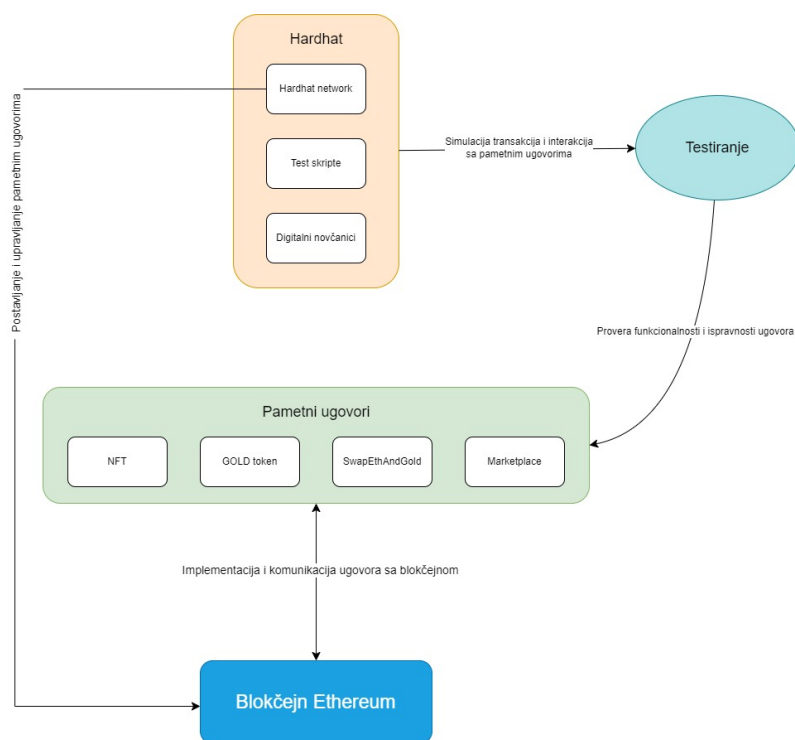
**Fiksni delovi arhitekture** su komponente koje ostaju nepromenjene i koje obezbeđuju stabilnost i funkcionalnost sistema:

- **Hardhat:** Služi kao razvojno i testno okruženje za projekte na mreži *Ethereum*. Uključuje:
  - **Hardhat network:** Simulira mrežu *Ethereum* za razvoj i testiranje.
  - **Test skripte:** Koriste se za automatsko testiranje logike i funkcionalnosti pametnih ugovora.
  - **Digitalni novčanici:** Simuliraju korisničke novčanike za testiranje transakcija unutar okruženja *Hardhat*.

- **Blokčejn Ethereum:** Glavna infrastruktura koja omogućava implementaciju i komunikaciju pametnih ugovora, na kojoj se izvršavaju sve transakcije i logika ugovora.

**Promenljive delove arhitekture** predstavljaju pametni ugovori kojima se definiše funkcionalnost decentralizovanog tržišta.

**Testiranje** nije komponenta već je ključan deo razvojnog procesa koji osigurava pravilnu funkcionalnost svih komponenti sistema i bezbednost pametnih ugovora pre njihovog postavljanja na javnu mrežu. Ovo testiranje uključuje simulaciju transakcija kroz digitalne novčanike unutar okruženja *Hardhat*.



Slika 4.1: Dijagram arhitekture sistema

## Pametni ugovori

Implementacija pametnih ugovora počinje određivanjem osnovnih funkcionalnosti koje su potrebne za rad decentralizovanog tržišta. Ovde će biti razmotreno kako su razvijeni ključni ugovori za kreiranje, prodaju, i trgovinu nezamenjivim tokenima, kao i za upravljanje tokenima ERC-20.

## Ugovori tokena NFT i GOLD

Ovi ugovori omogućavaju kreiranje i upravljanje digitalnim tokenima koji se koriste u okviru decentralizovane aplikacije. Ugovor `GoldToken` koristi standard ERC-20 za kreiranje tokena GOLD, koji funkcionišu kao glavna valuta unutar tržišta. Koriste se za trgovinu nezamenjivim tokenima i obavljanje drugih transakcija unutar sistema.

Ugovor NFT je implementacija standarda ERC-721, koji omogućava kreiranje nezamenjivih tokena. Ovi tokeni predstavljaju jedinstvene digitalne predmete koji se mogu trgovati na tržištu. Osnovne funkcionalnosti ovog ugovora uključuju kreiranje novih tokena, prenos vlasništva između korisnika, i emitovanje događaja prilikom svake transakcije.

## Ugovor za kupovinu i prodaju tokena GOLD

Ugovor `SwapEthAndGold` je centralni deo ekonomskog modela decentralizovane aplikacije koji omogućava korisnicima da konvertuju sredstva između tokena Ether i GOLD. Ovaj ugovor olakšava trgovinu i povećava likvidnost na tržištu. Korisnici mogu brzo i sigurno razmenjivati tokene koristeći ovaj ugovor, uz garanciju da će razmena biti obavljena po unapred definisanom kursu.

`SwapEthAndGold` koristi biblioteku *OpenZeppelin* za bezbednost i matematičke operacije kako bi se osiguralo pravilno izvršavanje transakcija. Ključne funkcionalnosti ovog ugovora uključuju:

- **Dodavanje likvidnosti:** Omogućava vlasniku ugovora da doda Ether u ugovor, čime se obezbeđuje likvidnost za dalju trgovinu tokenima GOLD.
- **Kupovina tokena GOLD:** Korisnici mogu zameniti Ether za tokene GOLD, pri čemu se iznos tokena GOLD određuje na osnovu kursa definisanog u ugovoru.
- **Prodaja tokena GOLD:** Korisnici mogu prodati svoje tokene GOLD i dobiti odgovarajući iznos Ethera.



### Ugovor decentralizovanog tržišta nezamenjivih tokena

Ugovor Marketplace je ključni element za trgovanje nezamenjivim tokenima na decentralizovanom tržištu. Ovaj ugovor omogućava korisnicima da objave (eng. *list*) odnosno ponude ili da kupe nezamenjive tokene koristeći tokene GOLD kao valutu trgovine.

Ključne funkcionalnosti ugovora Marketplace uključuju:

- **Objavljivanje tokena na tržištu:** Korisnici mogu postaviti svoje nezamenjive tokene na tržište, definišući cenu po kojoj žele da ih prodaju.
- **Izračunavanje cene tokena:** Ugovor automatski izračunava cenu nezamenjivog tokena uračunavajući naknadu za decentralizovano tržište.
- **Sigurna razmena sredstava i tokena:** Omogućava korisnicima da izvrše sigurnu razmenu sredstava i nezamenjivih tokena putem tržišta, osiguravajući da se transakcije izvršavaju u skladu sa dogovorenim uslovima.
- **Emitovanje događaja o prodaji:** Svaka transakcija na tržištu emituje događaj koji beleži prodaju ili kupovinu nezamenjivog tokena, čime se osigurava transparentnost i praćenje na mreži.

### Struktura proizvoda na decentralizovanom tržištu

U okviru decentralizovanog tržišta, svaki proizvod je opisan koristeći strukturu podataka koja omogućava efikasno i organizovano čuvanje svih relevantnih informacija o proizvodu. Ova struktura je implementirana unutar pametnog ugovora Marketplace i služi kao temelj za sve transakcije i interakcije unutar tržišta.

```
1 struct Item
2 {
3     uint itemId;
4     IERC721 nft;
5     uint tokenId;
6     uint256 price;
7     address payable seller;
8     bool sold;
9 }
```

- **itemId**: Jedinstveni identifikator svakog predmeta na tržištu,
- **nft**: Adresa pametnog ugovora nezamenjivog tokena koji je na listi za prodaju ili kupovinu,
- **tokenId**: Identifikator tačno određenog nezamenjivog tokena unutar kolekcije implementirane od strane NFT pametnog ugovora,
- **price**: Vrednost nezamenjivog tokena koji je objavljen ili prodat,
- **seller**: Adresa novčanika koji je stavio nezamenjivi token na prodaju,
- **sold**: Atribut koje označava da li je nezamenjivi token prodat ili ne.

Ova struktura omogućava da se svaki proizvod na tržištu opisuje na standardizovan i konzistentan način, što olakšava upravljanje i interakciju sa proizvodima unutar pametnog ugovora.

Prikaz definicije i implementacije ovih ugovora će biti izložen u poglavlju 5.

### 4.3 Životni ciklus nezamenjivog tokena

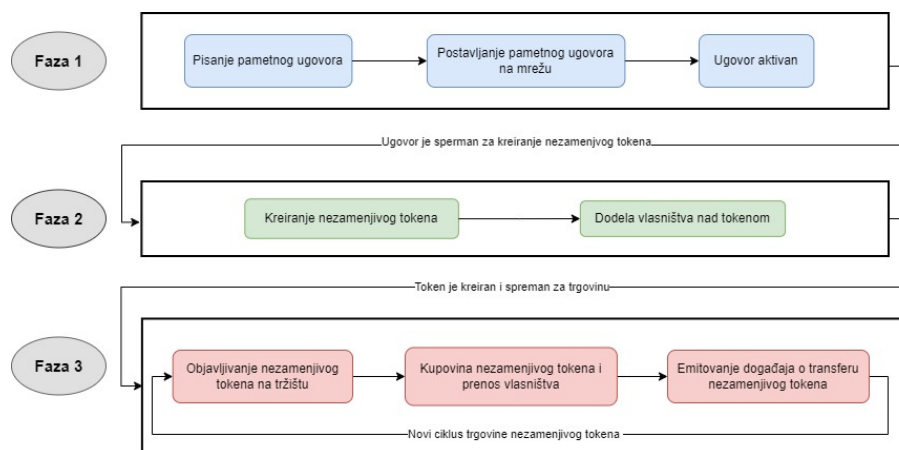
Nezamenjivi tokeni prolaze kroz nekoliko ključnih faza tokom svog životnog ciklusa, od pisanja pametnog ugovora, preko kreiranja tokena, do trgovine tokenom između različitih korisnika. Na slici 4.2 je prikazan dijagram koji ilustruje ove faze i glavne korake unutar svake faze.

#### Faza 1: Pisanje i postavljanje pametnog ugovora

Prva faza u životnom ciklusu nezamenjivog tokena uključuje kreiranje pametnog ugovora koji definiše pravila i funkcionalnosti tokena. Ova faza se sastoji od sledećih koraka:

- **Pisanje pametnog ugovora**: Programer piše kod pametnog ugovora koji definiše pravila i uslove za kreiranje i trgovinu nezamenjivim tokenom.
- **Postavljanje pametnog ugovora**: Nakon što je kod napisan, pametni ugovor se postavlja na blokčejn mrežu.

Nakon postavljanja, pametni **ugovor postaje aktivan**, čime započinje sledeća faza u životnom ciklusu tokena – kreiranje tokena i dodela vlasništva nad njim.



Slika 4.2: Dijagram životnog ciklusa nezamenjivog tokena

## Faza 2: Kreiranje nezamenjivog tokena i dodela vlasništva

Nakon što je pametni ugovor postavljen i aktiviran, sledeća faza uključuje kreiranje nezamenjivog tokena i dodelu vlasništva nad tim tokenom.

- **Kreiranje nezamenjivog tokena:** Pametni ugovor kreira novi nezamenjivi token na osnovu pravila definisanih u ugovoru.
- **Dodela vlasništva nad tokenom:** Nakon što je token kreiran, vlasništvo nad tokenom se dodeljuje prvom korisniku, čime postaje vlasnik tokena.

Prelaskom iz faze 2 u fazu 3, token koji je kreiran postaje spreman za trgovinu na tržištu.

## Faza 3: Trgovina nezamenjivim tokenom

U poslednjoj fazi, token se stavlja na tržište i prolazi kroz nekoliko iteracija trgovine, gde vlasništvo nad tokenom prelazi sa jednog korisnika na drugog.

- **Objavljivanje nezamenjivog tokena na tržištu:** Token se stavlja na tržište, gde postaje dostupan za kupovinu drugim korisnicima.
- **Kupovina nezamenjivog tokena i prenos vlasništva:** Drugi korisnik kupuje token, nakon čega se vlasništvo prenosi na novog korisnika.
- **Emitovanje događaja o transferu nezamenjivog tokena:** Pametni ugovor emituje događaj koji beleži transfer vlasništva na blokčejn mreži.

Nakon emitovanja događaja o transferu, token se ponovo objavljuje na tržištu, čime započinje novi ciklus trgovine.

## 4.4 Implementacija i praktična primena arhitekture

Arhitektura sistema, kako je definisana kroz okvir *Hardhat* i blokčejn *Ethereum*, osigurava da se sve komponente sistema međusobno povezuju i funkcionišu u sinergiji. U praksi, ovo znači da su svi pametni ugovori razvijeni i testirani u lokalnom okruženju *Hardhat* pre nego što su implementirani na glavnu mrežu *Ethereum*. Ovakav pristup omogućava stabilan i siguran sistem, gde su sve potencijalne greške otkrivene i ispravljene pre javne upotrebe.

Zahvaljujući modularnosti arhitekture i upotrebi okvira *Hardhat*, implementacija i puštanje sistema u rad je jednostavan proces. *Hardhat* omogućava razvoj i testiranje unutar simuliranog okruženja *Ethereum*, što omogućava razvijateljima softvera da optimizuju pametne ugovore i potvrde njihovu funkcionalnost pre nego što ih puste u rad na glavnoj mreži. Na taj način se umanjuju rizici i osigurava da sistem bude spreman za operativnu upotrebu.

## Glava 5

# Implementacija i testiranje pametnih ugovora

Za implementaciju, testiranje i postavljanje pametnih ugovora korišćeno je razvojno okruženje *Hardhat*, što je zahtevalo podešavanje određenih parametara u njenoj konfiguraciji, uključujući lokacije pametnih ugovora, testova i skripte za postavljanje pametnih ugovora na blokčejn.

```
1 require("@nomiclabs/hardhat-waffle");
2
3 module.exports = {
4   solidity: "0.8.4",
5   paths: {
6     artifacts: "./src/backend/artifacts",
7     sources: "./src/backend/contracts",
8     tests: "./src/backend/test",
9   },
10  };
```

### 5.1 Ugovor za kreiranje tokena GOLD

Kreiranje tokena uključuje kreiranje novih instanci tokena ERC-20, u ovom slučaju tokena GOLD. Ugovor `GoldToken` koristi standard ERC-20 za kreiranje tokena i njegova osnovna funkcionalnost je implementirana kroz konstruktor koji se poziva prilikom kreiranja ugovora.

```

1 contract GoldToken is ERC20 {
2     constructor(uint256 initialSupply) ERC20("GOLD Token", "GOLD")
3     {
4         _mint(msg.sender, initialSupply * (10 ** decimals()));
5     }
6 }

```

- `initialSupply`: Inicijalna količina tokena koja se kreira.
- `decimals()`: Funkcija koja vraća broj decimala za token, standardno za tokene ERC-20 je postavljena na 18<sup>1</sup>.
- `_mint`: Funkcija iz standarda ERC-20 koja se koristi za kreiranje novih tokena i dodavanje istih na adresu postavljača (u ovom slučaju na adresu `msg.sender` koja je kreirala ugovor).

Ovaj konstruktor pri postavljanju pametnog ugovora na blokčejn kreira početnu količinu tokena, pomnoženu sa 10 u stepenu broja decimala, što omogućava precizno predstavljanje tokena u manjim jedinicama, i šalje je na adresu onoga ko postavlja ugovor. Dok niske „GOLD Token” i „GOLD” predstavljaju imena tokena odnosno njihove simbole.

## 5.2 Ugovor za kreiranje nezamenjivih tokena

Ugovor NFT je implementacija standarda ERC-721, koji omogućava kreiranje nezamenjivih tokena. U ovom slučaju, konstruktor definiše ime i simbol ugovora odnosno nezamenjivog tokena pri postavljanju istog na blokčejn.

```

1 contract NFT is ERC721URIStorage {
2     uint public tokenCount;
3     constructor() ERC721("NFT", "NFT") {}
4     function mint(string memory _tokenURI) external returns (uint)
5     {
6         tokenCount++;
7         _safeMint(msg.sender, tokenCount);
8         _setTokenURI(tokenCount, _tokenURI);
9         return (tokenCount);
10    }
11 }

```

<sup>1</sup>U nastavku će biti objašnjeno i zašto upravo 18

- `tokenCount`: Brojač koji se koristi za generisanje jedinstvenog identifikatora za svaki novi nezamenjivi token. Tip ovog brojača je `uint`, čiji je opseg 256 bita<sup>2</sup>.
- `_tokenURI`: URI metapodataka nezamenjivih tokena, često smešten na IPFS, koji sadrži informacije kao što su slika, opis i druge karakteristike.
- `_safeMint`: Funkcija iz standarda ERC-721 koja bezbedno mintuje nezamenjivi token i dodeljuje ga vlasniku adrese `msg.sender`.
- `_setTokenURI`: Funkcija koja povezuje nezamenjivi token sa njegovim metapodacima.

Funkcija za kreiranje tokena inkrementira brojač, kreira novi nezamenjivi token sa sledećim dostupnim identifikatorom, postavlja URI metapodataka za taj token, i vraća identifikator novokreiranog tokena.

### 5.3 Ugovor za kupovinu i prodaju tokena GOLD

Ugovor `SwapEthAndGold` omogućava korisnicima razmenu između digitalnih sredstava: tokena Ether i GOLD.

#### Konstruktor

Konstruktor ugovora inicijalizuje ugovor tako što dohvata adresu tokena GOLD koja se koristi za dalje operacije:

```
1 constructor(address gldTokenAddress) {  
2     gldToken = GoldToken(gldTokenAddress);  
3 }
```

Parametar `gldTokenAddress` je adresa tokena GOLD koja se koristi za interakciju sa tim ugovorom.

---

<sup>2</sup>Tip `uint` u *Solidity*-ju je skraćenica za `uint256`, što znači da je to celobrojni tip podataka koji zauzima 256 bita. Postoje i drugi `uint` tipovi, kao što su `uint8`, `uint16`, itd., koji se povećavaju za 8 bita sve do 256 koji je podrazumevan opseg kada se koristi `uint` tip.

## Dodavanje likvidnosti u ugovor

Funkcija `addLiquidity` omogućava vlasniku ugovora da doda Ether u ugovor, koji se može koristiti za zamenu sa korisnicima koji žele da prodaju svoje tokene GOLD:

```
1 function addLiquidity() public payable onlyOwner {}
```

Ova funkcija je ograničena modifikatorom `onlyOwner`, što znači da samo vlasnik ugovora može da dodaje likvidnost i ovde se prvi put spominje ključna reč `payable` kao opis funkcije za koju je rečeno da omogućava funkciji da primi Ether.

## Kupovina GOLD tokena

Funkcija `buyGLDTokens` omogućava korisnicima da kupe tokene GOLD u zamenu za Ether. Kurs između tokena GOLD i Ether je 100 prema 1, što znači da 1 Ether može biti zamenjen za 100 tokena GOLD, i obrnuto. Ovaj kurs je definisan koristeći statičko polje na početku ugovora:

```
1 uint256 public gldTokensPerEth = 100;
```

Definicija funkcije `buyGLDTokens` je prikazana u kodu [5.1](#).

Ovde se prvi put nailazi na metodu `require` koja označava obavezan uslov koji mora biti ispunjen kako bi se nastavilo izvršavanje pametnog ugovora. Ukoliko uslov nije ispunjen, izvršavanje se zaustavlja i sve komande izvršene pre nje se poništavaju.

Promenljive `msg.value` i `msg.sender` odgovaraju količini Ethern transakcije koja je pozvala ugovor odnosno adresi pozivaoca ugovora. Ove dve promenljive su članice objekta `message` koji služi za promenu stanja odnosno izvršavanje transakcija na mreži *Eteheruem*.

Kako bi sva izračunavanja bila sigurna koristi se biblioteka *SafeMath*. *SafeMath* je biblioteka za *Solidity* koja pruža funkcije za sigurno izvođenje aritmetičkih operacija nad celobrojnim vrednostima. Dizajnirana je da spreči uobičajene greške kao što su prekoračenje (eng. *overflow*) i potkoračenje (eng. *underflow*), koje mogu dovesti do neočekivanih ponašanja i sigurnosnih propusta u pametnim ugovorima [\[39\]](#).

U ovom ugovoru se može primetiti i ključna reč `emit` koja označava emitovanje određenog događaja na blokčejn. Ova emitovanja omogućavaju aplikacijama koje interaguju sa ugovorom da „slušaju” određene događaje i reaguju na njih u realnom vremenu, kao što su na primer ažuriranje korisničkog interfejsa ili izvršavanje dodatnih akcija kada se neki predmet proda.



```
1 function buyGLDTokens() public payable returns (uint256 amountGLD)
2 {
3     require(
4         msg.value > 0.1 ether,
5         "Minimum amount should equal to 0.1 ETH."
6     );
7
8     uint256 amountOfGLDTokensToBeBought = SafeMath.mul(
9         msg.value,
10        gldTokensPerEth
11    );
12
13    uint256 contractBalanceOfGLDTokens =
14        gldToken.balanceOf(address(this));
15    require(
16        contractBalanceOfGLDTokens >= amountOfGLDTokensToBeBought,
17        "Contract doesn't have enough GOLD tokens."
18    );
19
20    bool isTransferred = gldToken.transfer(
21        msg.sender,
22        amountOfGLDTokensToBeBought
23    );
24    require(
25        isTransferred,
26        "Transfer of the GOLD tokens has failed."
27    );
28
29    emit Swap(msg.sender, "buy", msg.value,
30        amountOfGLDTokensToBeBought);
31    return amountOfGLDTokensToBeBought;
32 }
```

Listing 5.1: Definicija funkcije buyGLDTokens

- `msg.value`: Količina Ethernu koju korisnik šalje.
- `gldTokensPerEth`: Konverzioni odnos između tokena Ether i GOLD koji je u ovom slučaju unapred određen.
- `isTransferred`: Promenljiva tipa `bool` koja označava da li je transfer tokena GOLD uspešan.

## Prodaja GOLD tokena

Funkcija `sellGLDTokens` omogućava korisnicima da prodaju svoje tokene GOLD u zamenu za Ether:

```
1 function sellGLDTokens(uint256 amountGLD) public returns (bool
   success)
2 {
3     require(amountGLD > 0, "Amount must be greater than 0.");
4     require(
5         gldToken.balanceOf(msg.sender) >= amountGLD,
6         "Not enough GOLD tokens."
7     );
8
9     uint256 amountETH = SafeMath.div(amountGLD, gldTokensPerEth);
10    require(
11        address(this).balance >= amountETH,
12        "Contract has not enough ETH."
13    );
14
15    bool isTransferred = gldToken.transferFrom(
16        msg.sender,
17        address(this),
18        amountGLD
19    );
20    require(
21        isTransferred,
22        "Transfer of the GOLD tokens has failed."
23    );
24
25    payable(msg.sender).transfer(amountETH);
26
27    emit Swap(msg.sender, "sell", amountETH, amountGLD);
28
29    return true;
30 }
```

- `amountGLD`: Količina tokena GOLD koje korisnik želi da proda.
- `amountETH`: Količina Ethern koja se dobija u zamenu.

Ove metode osiguravaju da korisnici mogu efikasno i bezbedno trgovati tokenima GOLD i Ether, pri čemu se koriste matematičke funkcije za precizno izračunavanje i sigurnosne provere da bi se osigurao pravilan transfer sredstava.

U ovom kontekstu, `payable(msg.sender)` pretvara adresu pošiljaoca (koji je prodao tokene GOLD) u „payable” adresu, što omogućava ugovoru da na tu adresu isplati odgovarajuću količinu Ether-a. Metoda `transfer` se koristi za slanje Ether-a, i ona automatski vrši transakciju u iznosu koji odgovara `amountETH`.

Dakle, `payable` je ključna za upravljanje i transfer Ether-a unutar pametnih ugovora *Ethereum*, koja omogućava kako prijem sredstava tako i njihovu isplatu.

## 5.4 Ugovor decentralizovanog tržišta nezamenjivih tokena

Ugovor Marketplace omogućava korisnicima da trguju digitalnim dobrima koristeći nezamenjive tokene i tokene GOLD. Ovaj ugovor obezbeđuje funkcije za objavljivanje, kupovinu i prodaju nezamenjivih tokena. Marketplace ugovor se direktno povezuje sa ugovorima NFT (kroz polje `nft` strukture `Item` koje mora imati svaki proizvod na na tržištu) i GOLD Token kako bi omogućio korisnicima da trguju nezamenjivim tokenima.

### Konstruktor

Konstruktor ugovora inicijalizuje osnovne parametre:

```
1 constructor(uint _feePercent, address _goldTokenAddress) {
2     feeAccount = msg.sender;
3     feePercent = _feePercent;
4     goldToken = IERC20(_goldTokenAddress);
5 }
```

- `_feePercent`: Procenat provizije koji se naplaćuje za svaku prodaju i direktno prosleđuje onom ko je postavio ugovor na blokčejn.
- `_goldTokenAddress`: Adresa tokena ERC-20 (GOLD) koji se koristi za transakcije.

Promenljiva `_feeAccount` je adresu novčanika koji je postavio ovaj ugovor i na koju će biti uplaćena provizija za svaku trgovinu na ovom decentralizovnom tržištu.

## Objavlivanje nezamenjivog tokena na tržištu

Funkcija `makeItem` omogućava korisnicima da objave svoje nezamenjive tokene za prodaju na tržištu:

```
1 function makeItem(IERC721 _nft, uint _tokenId, uint _price)
   external
2 {
3     require(_price > 0, "Price must be greater than zero");
4     itemCount++;
5     _nft.transferFrom(msg.sender, address(this), _tokenId);
6
7     Items[itemCount] = Item(
8         itemCount,
9         _nft,
10        _tokenId,
11        _price,
12        payable(msg.sender),
13        false
14    );
15
16    emit ItemListed(itemCount, address(_nft), _tokenId, _price, msg
17        .sender);
18 }
```

## Kupovina nezamenjivog tokena sa tržišta

Funkcija `purchaseItem` omogućava korisnicima da kupe nezamenjivi token koji je objavljen na tržištu:

```
1 function purchaseItem(uint _itemId) external
2 {
3     Item storage item = Items[_itemId];
4     uint256 totalPrice = getTotalPrice(_itemId);
5     require(!item.sold, "Item is already sold.");
6     require(
7         goldToken.balanceOf(msg.sender) >= totalPrice,
8         "Not enough GOLD tokens to cover the item price."
9     );
10
11    uint256 fullFeeAmount = SafeMath.mul(item.price, feePercent);
12    uint256 percentFeeAmount = SafeMath.div(fullFeeAmount, 100);
13    uint256 sellerAmount = item.price - percentFeeAmount;
```

```
14     goldToken.transferFrom(msg.sender, item.seller, sellerAmount);
15     goldToken.transferFrom(msg.sender, feeAccount, percentFeeAmount
16         );
17     item.sold = true;
18     item.nft.transferFrom(address(this), msg.sender, item.tokenId);
19     emit ItemPurchased(
20         item.itemId,
21         address(item.nft),
22         item.tokenId,
23         item.price,
24         item.seller,
25         msg.sender
26     );
27 }
```

## Funkcija za izračunavanje cene nezamenjivog tokena na tržištu

Funkcija `getTotalPrice` izračunava ukupnu cenu nezamenjivog tokena, uključujući tržišnu proviziju:

```
1 function getTotalPrice(uint _itemId) internal view returns (uint)
2 {
3     (bool addSuccess, uint256 addResult) = SafeMath.tryAdd(100,
4         feePercent);
5     require(addSuccess, "Adding feePercent failed");
6     (bool divSuccess, uint256 divResult) = SafeMath.tryDiv(
7         addResult, 100);
8     require(divSuccess, "Dividing by 100 failed");
9     (bool mulSuccess, uint256 mulResult) = SafeMath.tryMul(
10         Items[_itemId].price,
11         divResult
12     );
13     require(mulSuccess, "Multiplying price and feePercent failed");
14
15     return mulResult;
16 }
```

Ove funkcije omogućavaju efikasno i sigurno upravljanje trgovinom nezamenjivih tokena na tržištu, osiguravajući da su transakcije transparentne i da su svi učesnici sigurni.

## 5.5 Implementacija testova za ugovore

Pre postavljanja ugovora na blokčejn ali i radi potvrde da sve radi kako treba, neophodno je detaljno testiranje svih pametnih ugovora kako bi se osiguralo da su funkcionalni i sigurni za upotrebu. Ovo uključuje:

- **Jedinične testove:** Testiraju pojedinačne funkcije unutar ugovora kako bi se osiguralo da svaka od njih radi prema očekivanjima.
- **Integracione testove:** Proveravaju interakcije između različitih ugovora i drugih komponenti sistema kako bi se osiguralo da celokupna aplikacija funkcioniše kao što bi trebalo.

### Jedinični testovi

Jedinični testovi (eng. *unit tests*) su osnovni alat za proveru ispravnosti individualnih funkcija unutar pametnih ugovora. Testovi su napisani u *JavaScript*-u koristeći okvir *Hardhat* i biblioteku *Chai*, omogućavajući proveru ponašanja pametnih ugovora u različitim uslovima.

*Chai* je biblioteka za testiranje u *JavaScript*-u i *Node.js*-u koja pruža jasan i intuitivan način za pisanje pretpostavki (eng. *assertions*) u testovima. *Chai* podržava različite stilove pisanja, kao što su *expect*, *assert* i *should*, dajući više izbora pri odabiru stila za testiranje [40].

Prvo će biti objašnjeni testovi postavljanja ugovora, zatim funkcionalnosti tržišta i na kraju funkcionalnost kupovine i prodaje tokena GOLD.

### Provera pravog postavljača ugovora za tržište

```
1 it("Should set the right deployer of the NFT marketplace",
2   async function () {
3     expect(await marketplace.feeAccount()).to.equal(deployer.
4       address);
5   });
```

Ovaj test proverava da li je adresa koja je postavila ugovor ispravno postavljena kao vrednost statičke promenljive `feeAccount` u ugovoru `Marketplace` i da su prava na prijem naknada od tržišnih transakcija pravilno dodeljena. Adresa `deployer` je ona koja će postaviti sve ugovore na blokčejn, koja će u ovom slučaju dobijati proziviju od tržišta i kojoj će biti dodeljeni svi kreirani tokeni `GOLD` kada budu kreirani.

### Proveri ispravnog kreiranja `GOLD` tokena

```
1 it("Should mint correct amount of the gold tokens to the right
  deployer",
2   async function () {
3     const balance = await goldToken.balanceOf(deployer.address)
4     ;
5     expect(balance).to.equal(ethers.utils.parseUnits("100000",
6     18));
7   });
```

Ovaj test proverava da li je inicijalna količina tokena `GOLD` kreirana i da li je dodeljena adresi postavljачa, što potvrđuje ispravnost funkcije kreiranja u ugovoru `GoldToken`.

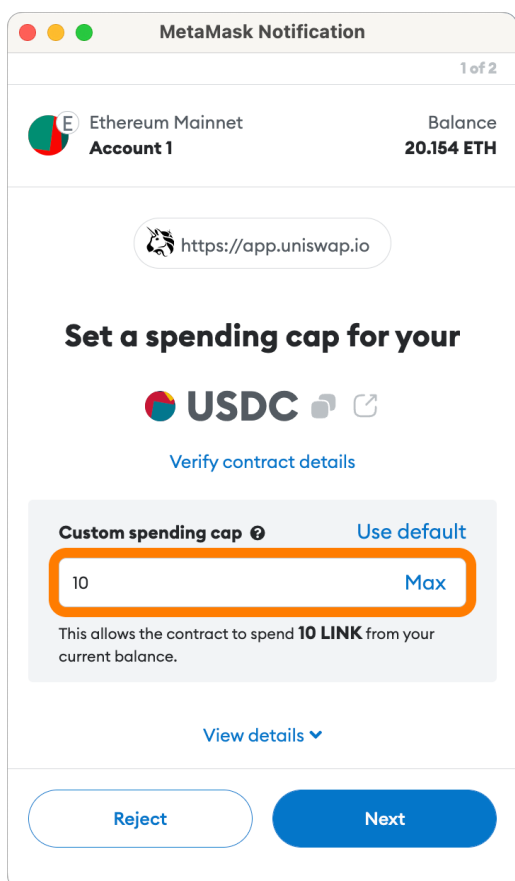
### Objavljivanje proizvoda na tržište

```
1 it("Should allow to list an item", async function ()
2 {
3   await nft.connect(addr1).mint(uri);
4   await nft.connect(addr1).approve(marketplace.address, 1);
5   await marketplace.connect(addr1).
6     makeItem(nft.address, 1, ethers.utils.parseEther("200"));
7
8   const item = await marketplace.Items(1);
9   expect(item).to.exist;
10  expect(item.seller).to.equal(addr1.address);
11  expect(ethers.utils.parseEther("200")).to.be.equal(item.price);
12 });
```

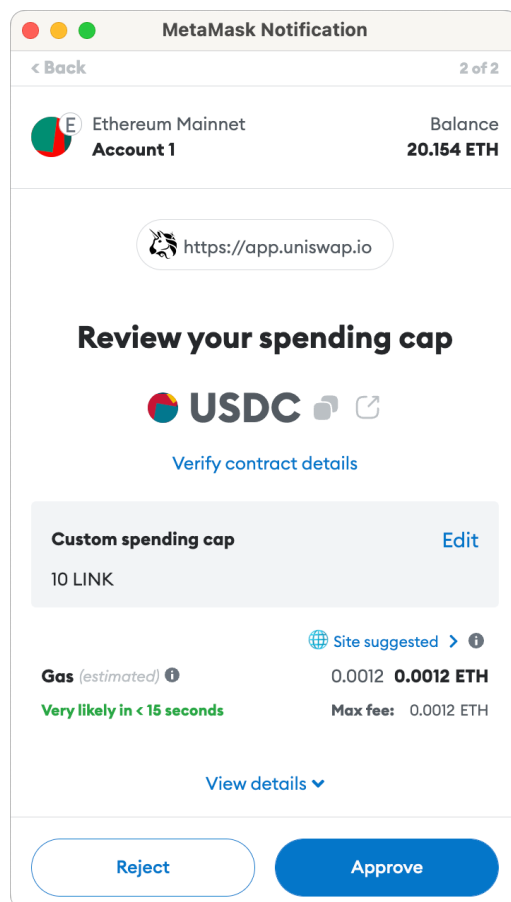
Testira se da li korisnik može da objavi nezamenjivi token na tržištu. Uključuje njegovo kreiranje, odobravanje tržištu da upravlja tokenom, i kreiranje ponude za taj nezamenjivi token. Proverava se da li su svi podaci o artiklu ispravno zabeleženi. Funkcija `approve` omogućava korisniku da dodeli pametnom ugovoru dozvolu za transfer određene količine tokena s njegovog naloga na drugi nalog ili na sam ugovor.

## GLAVA 5. IMPLEMENTACIJA I TESTIRANJE PAMETNIH UGOVORA

Na slici 5.1 se može videti kako u novčaniku *Metamask* izgleda ekran za odabir broja tokena za koje se daje dozvola, a zatim na slici 5.2 se može videti kako izgleda ekran kojim se to potvrđuje [41].



Slika 5.1: Odabir broja tokena



Slika 5.2: Potvrda davanja dozvole za odabrane tokene

### Objavljivanje proizvoda koji nema cenu

```
1 it("Should fail to list an item with zero price", async function ()
2 {
3   await expect(
4     marketplace.connect(addr1).makeItem(nft.address, 1, 0)
5   ).to.be.revertedWith("Price must be greater than zero");
6 });
```



Ovaj test proverava da li sistem uspešno sprečava objavljivanje nezamenjivog tokena za prodaju na tržištu čija je cena postavljena na nulu - što je neispravno stanje.

### Objavljivanje nezamenjivog tokena sa neispravnom adresom ugovora

```
1 it("Should fail with an invalid NFT address", async function ()
2 {
3   const invalidNFTAddress = "0
4     x0000000000000000000000000000000000000000000000000000000000000000";
5   await expect(
6     marketplace
7       .connect(addr1)
8       .makeItem(invalidNFTAddress, 1, ethers.utils.parseEther("1"))
9   ).to.be.revertedWith("function call to a non-contract account");
10 });
```

Poruka „function call to a non-contract account” označava da je neki od poziva bio usmeren ka nepostojećoj adresi ugovora.

### Kupovina proizvoda

```
1 it("Should allow to purchase an item", async function ()
2 {
3   await goldToken.connect(deployer).
4     approve(marketplace.address, ethers.utils.parseEther("500"))
5     );
6   await marketplace.connect(deployer).purchaseItem(1);
7
8   const item = await marketplace.Items(1);
9   expect(item.sold).to.equal(true);
10 });
```

Testira proces kupovine proizvoda tako što se proverava se da li je korisnik odobrio dovoljno tokena GOLD i da li je artikal pravilno označen kao prodat nakon transakcije.

### Kupovina već prodatog proizvoda

Test, prikazan u kodu 5.2, proverava scenario u kojem korisnik pokušava da kupi artikal koji je već prethodno prodat, očekuje se da ugovor pravilno spreči transakciju.

```
33 it("Should fail to purchase an already sold item",
34   async function () {
35     await nft.connect(addr1).mint(uri);
36     await nft.connect(addr1).approve(marketplace.address, 1);
37     await marketplace.connect(addr1).makeItem(nft.address, 1,
38       ethers.utils.parseEther("100"));
39     await goldToken.connect(addr1).approve(marketplace.address,
40       ethers.utils.parseEther("100"));
41     await marketplace.connect(addr1).purchaseItem(1);
42     await expect(marketplace.connect(addr2).purchaseItem(1)).
43       to.be.revertedWith("Item is already sold");
44   });
```

Listing 5.2: Testiranje kupovine već prodatog artikla

### Kupovina tokena GOLD za ETH

```
1 it("Should allow to BUY Gold tokens for ETH", async function ()
2 {
3   const initialBalanceEth = await ethers.provider.getBalance(
4     addr1.address);
5   const tx = await swapEthAndGold.connect(addr1).
6     buyGLDTokens({ value: ethers.utils.parseEther("0.2") });
7   const finalBalanceEth = await ethers.provider.getBalance(addr1.
8     address);
9   const gasUsed = tx.gasUsed.mul(tx.gasPrice);
10  expect(finalBalanceEth).to.equal(initialBalanceEth.
11    sub(ethers.utils.parseEther("0.2")).sub(gasUsed));
12 });
```

Testira funkciju za kupovinu tokena GOLD koristeći Ether, proveravajući da li su Ether sredstva umanjena uzimajući u obzir i troškove *gas*-a.

### Prodaja GOLD Tokena za ETH

Test, prikazan u kodu 5.3, proverava funkcionalnost prodaje tokena GOLD za Ether, osiguravajući da su tokeni pravilno preneseni i da je stanje tokena GOLD kod korisnika umanjeno za odgovarajući iznos.

```

1 it("Should allow to SELL Gold tokens for ETH", async function ()
2 {
3     await goldToken.connect(addr1).
4         approve(swapEthAndGold.address, ethers.utils.parseEther("
5             100"));
6     await swapEthAndGold.connect(addr1).
7         sellGLDTokens(ethers.utils.parseEther("100"));
8
9     const finalBalanceGLD = await goldToken.balanceOf(addr1.address
10        );
11    expect(finalBalanceGLD).to.
12        equal(initialBalanceGLD.sub(ethers.utils.parseEther("100"))
13        );
14 });

```

Listing 5.3: Testiranje prodaje GOLD tokena

## Integracioni testovi

Integracioni testovi su ključni za verifikaciju kako se pojedinačni delovi sistema ponašaju kada rade zajedno. Cilj ovih testova je da otkriju bilo kakve probleme u interakciji između različitih pametnih ugovora ili komponenti sistema, osiguravajući da celokupna aplikacija funkcioniše kako treba. U nastavku se mogu videti integracioni testovi i njihovi opisi:

## Kupovina nezamenjivog tokena koristeći GOLD tokene

```

1 it("Should handle a full transaction buying an NFT with GOLD tokens
2 ",
3     async function () {
4         await nft.connect(addr1).mint(uri);
5         await nft.connect(addr1).approve(marketplace.address, 1);
6         await marketplace.connect(addr1).
7             makeItem(nft.address, 1, ethers.utils.parseEther("200"));
8
9         await goldToken.connect(deployer).
10            transfer(addr2.address, ethers.utils.parseEther("500"));
11
12        await goldToken.connect(addr2).
13            approve(marketplace.address, ethers.utils.parseEther("200"))
14            );

```

```

13     await marketplace.connect(addr2).purchaseItem(1);
14     const item = await marketplace.Items(1);
15     expect(item.sold).to.equal(true);
16     expect(await nft.ownerOf(1)).to.equal(addr2.address);
17 });

```

Ovaj test simulira kompletnu kupovinu nezamenjivog tokena koristeći tokene GOLD, počevši od kreiranja i objavljivanja nezamenjivog tokena na tržištu, do njegove prodaje. Proverava da li je artikal pravilno označen kao prodat i da li je vlasništvo nad nezamenjivim tokenom prešlo na kupca.

## Refundacija sredstava usled neuspešne kupovine

```

1 it("Should refund on a failed purchase", async function ()
2 {
3     await nft.connect(addr1).mint(uri);
4     await nft.connect(addr1).approve(marketplace.address, 1);
5     await marketplace.connect(addr1).
6         makeItem(nft.address, 1, ethers.utils.parseEther("100"));
7
8     const initialBalance = await goldToken.balanceOf(addr2.address);
9     await expect(marketplace.connect(addr2).purchaseItem(1)).to.be.
10        reverted;
11     const finalBalance = await goldToken.balanceOf(addr2.address);
12     expect(finalBalance).to.equal(initialBalance);
13 });

```

Testira scenario u kojem transakcija kupovine ne uspeva, fokusirajući se na to da li su tokeni GOLD vraćeni kupcu, što je ključno za osiguranje da greške u transakciji ne dovode do finansijskih gubitaka za korisnike.

## Automatska distribucija tržišne naknade na odgovarajuću adresu

Test koji je prikazan u kodu 5.4 proverava da li su naknade od prodaje nezamenjivih tokena pravilno distribuirane između prodavca i adrese `feeAccount` odnosno onoga ko je postavio ugovor za tržište na blokčejn (u ovom slučaju adresa pod imenom `deployer`), koristeći procenat definisan u ugovoru `Marketplace`. Fokus je na proveru pravilnog računanja i transfera naknada, što je neophodno za održavanje transparentnosti i fer prakse unutar tržišta.

```
1 it("Should automatically distribute transaction fees", async
2   function ()
3 {
4   await goldToken.connect(deployer).
5     transfer(addr2.address, ethers.utils.parseEther("10"));
6   const initialBalance = await goldToken.balanceOf(deployer.address
7     );
8
9   const salePrice = ethers.utils.parseEther("5");
10  await nft.connect(addr1).mint(uri);
11  await nft.connect(addr1).approve(marketplace.address, 1);
12  await marketplace.connect(addr1).makeItem(nft.address, 1,
13    salePrice);
14
15  await goldToken.connect(addr2).approve(marketplace.address,
16    salePrice);
17  await marketplace.connect(addr2).purchaseItem(1);
18
19  const fee = salePrice.mul(await marketplace.feePercent()).div
20    (100);
21  const expectedFinalBalance = initialBalance.add(fee);
22  const feeAccountFinalBalance = await goldToken.balanceOf(deployer
23    .address);
24
25  expect(feeAccountFinalBalance).to.equal(expectedFinalBalance);
26 });
```

Listing 5.4: Testiranje automatske distribucije tržišne naknade

Svaki od ovih testova igra ključnu ulogu u proveru složenih procesa koji obuhvataju više funkcionalnosti, ugovora i interakcija. Oni pomažu u identifikaciji problema u interakcijama između ugovora pre nego što sistem postane aktivan, čime se smanjuje rizik od grešaka koje bi mogle negativno uticati na korisnike i celokupni integritet platforme.

## 5.6 Postavljanje ugovora na blokčejn i testiranje

Nakon što su pametni ugovori razvijeni i testirani, sledeći korak je njihovo postavljanje na *Ethereum* mrežu. Ovaj proces, poznat kao *deploy*, zahteva da se ugovori postave na javni blokčejn, čime postaju dostupni za interakciju sa korisnicima i drugim ugovorima.

Skripta `Deploy.js` se koristi za automatsko postavljanje pametnih ugovora na mrežu *Ethereum* koristeći okruženje *Hardhat*. U ovom slučaju, ugovori će biti postavljeni na lokalnu mrežu. Skripta upravlja procesom kompilacije, konfiguracije i postavljanja ugovora, kao i čuvanjem podataka tih ugovora nakon postavljanja. U nastavku se mogu videti ključni delovi skripte:

```
1 async function main() {
2   const [deployer] = await ethers.getSigners();
3   const initialSupply = "100000";
4
5   const NFT = await ethers.getContractFactory("NFT");
6   const Marketplace = await ethers.getContractFactory("
7     Marketplace");
8   const GoldToken = await ethers.getContractFactory("GoldToken");
9   const SwapContract = await ethers.getContractFactory("
10     SwapEthAndGold");
11
12   const nft = await NFT.deploy();
13   const goldToken = await GoldToken.deploy(initialSupply);
14   const marketplace = await Marketplace.deploy(1, goldToken.
15     address);
16   const swapEthAndGold = await SwapContract.deploy(goldToken.
17     address);
18
19   saveContractsData(marketplace, "Marketplace");
20   saveContractsData(nft, "NFT");
21   saveContractsData(goldToken, "GoldToken");
22   saveContractsData(swapEthAndGold, "SwapEthAndGold");
23 }
```

Funkcija `GetContractsFactory` dohvata ugovore, a zatim ih pomoću druge funkcije `deploy` postavlja na lokalni blokčejn.

Ovde se mogu videti konstruktori ugovora i vrednosti njihovih parametara koje su korišćene pri njihovom postavljanju na blokčejn:

- `NFT.deploy()`, `GoldToken.deploy(initialSupply)`, `Marketplace.deploy(1, goldToken.address)`, i `SwapContract.deploy(goldToken.address)` su pozivi konstruktora svakog od ugovora. Ovi pozivi stvaraju nove instance ugovora na blokčejnu.
- `GoldToken.deploy(initialSupply)` postavlja početnu količinu tokena GOLD.

- `Marketplace.deploy(1, goldToken.address)` inicijalizuje decentralizovano tržište nezamenjivih tokena sa 1% prozivije i adresom tokena GOLD.
- `SwapContract.deploy(goldToken.address)` povezuje ugovor `SwapGoldAndEth` sa adresom tokena GOLD.

Nakon podešavanja konfiguracionog fajla okruženja *Hardhat* i napisane skripte za postavljanje ugovora na blokčejn, potrebno je da se podigne lokalni čvor na mreži *Ethereum*, da se postave ugovori na njega i onda mogu svi testovi da se pokrenu, a to se radi sledećim komandama:

1. Podizanje lokalnog blokčejna:

```
1 npx hardhat node
```

2. Postavljanje ugovora na testni blokčejn:

```
1 npx hardhat run .\src\backend\scripts\deploy.js --network  
localhost
```

3. Pokretanje svih testova:

```
1 npx hardhat test
```

Komandom za podizanje lokalnog blokčejna se kreira 20 adresa sa javnim i privatnim ključevima i balansom od 10000 Ether-a po adresi i u tom terminalu se loguju sve transakcije koje se odvijaju na lokalnom blokčejnu. Pošto lokalni blokčejn ne radi u pozadini već zauzima jedan terminal, za ostale komande mora da se koristi druga instanca terminala.

Nakon uspešnog postavljanja ugovora se dobija poruka koliko je fajlova kompajlirano i mogu se ispisati razne informacije, u ovom slučaju su tu početni balans tokena GOLD na adresi postavjača i adrese na blokčejnu svakog od postavljenih ugovora, što se može videti na slici 5.3. Balans tokena GOLD se implementira i prikazan je sa 18 decimalnih mesta kako bi se omogućile veoma male transakcije i precizna raspodela tokena što dalje omogućava fleksibilnost u transakcijama i izračunavanjima, čime se izbegavaju greške zaokruživanja koje bi mogle nastati sa manjim brojem decimalnih mesta. Na blokčejnu *Ethereum*, sve jedinice su podignute na 18. stepen kako bi se omogućila visoka preciznost u radu sa tokenima. Ovo je postignuto korišćenjem najmanje jedinice nazvane *wei*, gde je 1 Ether ekvivalentan  $10^{18}$  *wei* [42].

## GLAVA 5. IMPLEMENTACIJA I TESTIRANJE PAMETNIH UGOVORA

---

```
Compiling 19 files with 0.8.4
Solidity compilation finished successfully
Deploying contracts with the account: 0xf39Fd6e51aad88F6F4ce6aB8827279cFfFb92266
Account balance: 10000000000000000000000
GoldToken contract address: 0xe7f1725E7734CE288F8367e1Bb143E90bb3F0512
NFT contract address: 0x5FbDB2315678afecb367f032d93F642f64180aa3
Marketplace contract address: 0x9fE46736679d2D9a65F0992F2272dE9f3c7fa6e0
SwapEthAndGold contract address: 0xCf7Ed3AccA5a467e9e704C703E8D87F634fB0Fc9
```

Slika 5.3: Postavljanje ugovora na blokčejn

Kada se pokrenu testovi dobija se izlaz sa slike 5.4, u kom može da se vidi da li su svi testovi prošli, a ako neki nije - zašto nije prošao.

```
NFTMarketplace
Deployment
  ✓ Should set the right deployer of the NFT marketplace
  ✓ Should mint the correct amount of the gold tokens to the right deployer
Marketplace functionality
  ✓ Should allow to list an item (43ms)
  ✓ Should allow to purchase an item (91ms)
  ✓ Should fail to list an item with zero price (45ms)
  ✓ Should fail with an invalid NFT address
  ✓ Should fail to purchase an already sold item (106ms)
  ✓ Should fail to purchase multiple items without sufficient funds (56ms)
Swap functionality
  ✓ Should allow to BUY Gold tokens for ETH
  ✓ Should allow to SELL Gold tokens for ETH
Integration Tests
  ✓ Should handle a full transaction buying an NFT with GOLD tokens (86ms)
  ✓ Should refund on a failed purchase (59ms)
  ✓ Should automatically distribute transaction fees (72ms)
```

Slika 5.4: Testiranje



# Glava 6

## Diskusija i zaključak

### Prednosti i ograničenja decentralizovanog tržišta

Implementacija decentralizovanog tržišta nezamenjivih tokena na blokčejnu *Ethereum* donosi značajne prednosti u smislu transparentnosti i sigurnosti. Svaka transakcija se zapisuje na blokčejnu, što omogućava svim korisnicima da vide istoriju vlasništva i transakcija bez mogućnosti izmene, čime se smanjuje rizik od prevara i povećava poverenje u sistem. Takođe, eliminacija posrednika ne samo da smanjuje troškove transakcija već i ubrzava procese, čime se korisnicima omogućava brži i jeftiniji pristup tržištu.

Ipak, postoje i izazovi. Jedan od glavnih izazova je, sada već nekoliko puta spomenut, visok trošak *gas*-a na mreži *Ethereum*, što može biti prepreka za česte transakcije koje su tipične za tržište video igara. Takođe, kompleksnost implementacije i potreba za visokim stepenom tehničkog znanja ograničavaju pristupačnost ovakvih rešenja široj populaciji razvijачa i korisnika.

### Integracija sa tradicionalnim platformama za video igre

Integracija nezamenjivih tokena sa postojećim platformama poput *Steam*-a ili *PlayStation Network*-a predstavlja izazov zbog razlika u digitalnoj infrastrukturi i poslovnim modelima. Ova integracija zahteva saradnju između kompanije koje koriste tehnologiju blokčejn i tradicionalnih kompanija iz industrija video igara ali najverovatnije i odricanje dela profita. Integracija bi omogućila igračima da svoje digitalne predmete koriste i trguju njima preko različitih platformi, što bi moglo u potpunosti promeniti način na koji se digitalna imovina koristi i percipira u industriji video igara.

### Potencijalni uticaj na industriju video igara

Decentralizovano tržište može radikalno promeniti industriju video igara tako što omogućava igračima da pravno poseduju digitalne predmete i trguju njima na otvorenom tržištu. Ovo bi moglo dovesti do novih modela monetizacije gde igrači mogu zarađivati prodajom ili čak iznajmljivanjem svojih digitalnih predmeta, što dodatno povećava njihovu angažovanost i lojalnost prema igri.

### Pravni i regulatorni izazovi

Razvoj decentralizovanog tržišta nosi sa sobom određene pravne i regulatorne izazove, posebno u pogledu autorskih prava i prava intelektualne svojine. Pored toga, regulatorni okviri za kriptovalute i tehnologiju blokčejn još uvek se razvijaju, što može dovesti do pravne neizvesnosti za korisnike ovih platformi u budućnosti.

### Skalabilnost i potencijalna poboljšanja

Razmatranje skalabilnosti i optimizacije troškova *gas*-a su ključni za dugoročnu održivost i prihvatljivost decentralizovanih tržišta. Implementacija rešenja kao što je grupna kupovina nezamenjivih tokena ili prelazak na efikasnije implementacije tehnologije blokčejn može značajno smanjiti troškove i poboljšati iskustvo korisnika. Takođe, razvoj i primena tehnologija kao što su rešenja drugog sloja, mogu dodatno poboljšati performanse i smanjiti troškove transakcija.

Pored *Ethereum*-a, koji je trenutno vodeća platforma, mogu se razmatrati i druge blokčejn mreže kao što su *Solana (SOL)*, *Polygon (MATIC)*, *Arbitrum (ARB)* i *Optimism (OP)*, koji su poznati po svojim rešenjima drugog sloja (eng. *Layer 2*<sup>1</sup>).

Na tabeli 6.1 se može videti poređenje broja transakcija po sekundi i okvirne cene transakcija vodećih blokčejnova odnosno rešenja drugog sloja, kao uvod u diskusiju gde će između ostalog biti spomenuta potencijalna poboljšanja koja se tiču cene transakcija i skalabilnosti.

---

<sup>1</sup>Layer 2 rešenja su tehnologije koje rade „na” osnovnom blokčejnu (Layer 1), kao što je *Ethereum*, sa ciljem da poboljšaju skalabilnost i smanje troškove transakcija. Ova rešenja omogućavaju brže i jeftinije transakcije, izvršavajući ih van glavnog lanca i povremeno sinhronizujući sa osnovnim blokčejnom.

Blokčejn	Transakcije po sekundi	Cena transakcija (USD)
<i>Ethereum</i>	15-30	\$1.50 - \$5.00
<i>Solana</i>	50,000	\$0.00025 - \$0.01
<i>Polygon</i>	7,000	\$0.01 - \$0.05
<i>Arbitrum</i>	2,000	\$0.02 - \$0.50
<i>Optimism</i>	2,000	\$0.01 - \$0.20

Tabela 6.1: Uporedna analiza performansi blokčejnova [43]

## 6.1 Zaključak

U radu je objašnjena motivacija iza decentralizovanog tržišta nezamenjivih tokena i prikazana je njegova implementacija na blokčejnu *Ethereum*, koristeći pametne ugovore za kreiranje, prodaju, i razmenu nezamenjivih tokena i tokena ERC-20. Projekat ilustruje kako tehnologija blokčejn omogućava razvoj sigurnih i transparentnih tržišta digitalne imovine, nudeći inovativne ekonomske modele u industriji video igara.

Implementirani sistem demonstrira ključne prednosti decentralizacije, uključujući smanjenje potrebe za posrednicima i povećanje sigurnosti digitalnih transakcija. Pored toga, diskusija o skalabilnosti i optimizaciji troškova transakcija naglašava potencijal za dalje unapređenje i efikasnost. Analiza i poređenje sa tradicionalnim tržištima pokazuje da blokčejn može radikalno promeniti način trgovine digitalnom imovinom, unatoč izazovima poput visokih troškova gasa i skalabilnosti.

Kako se tehnologija razvija i usvaja na globalnom nivou, pitanje budućnosti tradicionalnih, centralizovanih tržišnih modela postaje sve relevantnije, uz razmatranje mogućnosti da sva tržišta evoluiraju ka potpunoj decentralizaciji i procene da li je to scenario kojem bi trebalo težiti. Ovaj rad otvara prostor za dalja istraživanja o potencijalnim uticajima decentralizovanih tržišta na globalnu ekonomiju i društvene strukture.

# Bibliografija

- [1] Igor Bojanic, Jelena Blazevic, and Josip Budak. The role of human resource analytics in the financial sector, 2024.
- [2] Xiang Li, Chunyang Gao, Rui Yang, and Zhenhua Li. Blockchain-based governance models supporting corruption prevention: A case study of a public finance ecosystem, 2023.
- [3] NFTEvening. How blockchain is revolutionizing cyber security, 2023. Pristupljeno: 10. 8. 2024.
- [4] Justin Birnbaum. Why video game makers see huge potential in blockchain—and why problems loom for their new nfts, 2022. Pristupljeno: 3. 8. 2024.
- [5] Immutable. Web3 marketplaces: Unlocking the potential of digital commerce, 2023. Pristupljeno: 3. 8. 2024.
- [6] Stuart Smadnick and Others. Systematic approach to analyzing security and vulnerabilities of blockchain, 2019. Pristupljeno: 3. 8. 2024.
- [7] United States Patent, Trademark Office (USPTO), and United States Copyright Office (USCO). Non-fungible tokens and intellectual property: A report to congress, 2024. Pristupljeno: 3. 8. 2024.
- [8] Kaspersky. Nfts definition & explanation. Pristupljeno: 10. 8. 2024.
- [9] The Verge. Nft, crypto, copyright, ownership. Pristupljeno: 10. 8. 2024.
- [10] Syamsu Rijal and Fajar Saranani. The role of blockchain technology in increasing economic transparency and public trust, 2023. Pristupljeno: 3. 8. 2024.
- [11] WhiteBIT. What is ethereum? Pristupljeno: 17. 8. 2024.

- [12] Ethereum Foundation. Ethereum documentation - evm overview. Pristupljeno: 10. 8. 2024.
- [13] Ethereum Foundation. Ethereum documentation - evm and smart contracts. Pristupljeno: 10. 8. 2024.
- [14] Ethereum Foundation. Ethereum documentation - gas and fees. Pristupljeno: 10. 8. 2024.
- [15] Ethereum Foundation. Ethereum documentation - transactions. Pristupljeno: 10. 8. 2024.
- [16] Ethereum Foundation. Ethereum documentation - erc standards. Pristupljeno: 10. 8. 2024.
- [17] Ethereum. Standard erc-721 za nezamenjive tokene.
- [18] Ethereum. Standard erc-20 za zamenjive tokene.
- [19] Matthieu Nadini, Laura Alessandretti, Flavio Di Giacinto, Mauro Martino, Luca Maria Aiello, and Andrea Baronchelli. Mapping the nft revolution: market trends, trade networks, and visual features. *Nature Scientific Reports*, 2021.
- [20] What are nft royalties? Pristupljeno: 17. 8. 2024.
- [21] Decrypt. Nft creators earned \$1.8 billion in royalties: Galaxy digital, 2023. Pristupljeno: 14. 5. 2024.
- [22] IPFS Team. Defining ipfs. Pristupljeno: 17. 8. 2024.
- [23] IPFS Team. Content identifiers (cids). Pristupljeno: 17. 8. 2024.
- [24] IPFS Team. Ipfs desktop installation.
- [25] Nft valuation models and management. Pristupljeno: 17. 8. 2024.
- [26] Martin Spann Sophie M. Berghueser. The value of distinctiveness: Product uniqueness in crypto marketing. *International Journal of Research in Marketing*, 2023. Pristupljeno: 17.8.2024.
- [27] Solidity Documentation. Introduction to smart contracts. Pristupljeno: 10. 8. 2024.

- [28] Solidity Documentation. Solidity documentation - structure of a contract. Pristupljeno: 10. 8. 2024.
- [29] Vitalik Buterin. Ethereum whitepaper, 2013. Pristupljeno: 10. 8. 2024.
- [30] Adeola Adebo. Understanding play-to-earn, the blockchain game. Pristupljeno: 17. 8. 2024.
- [31] OpenSea. Blockchains compatible with opensea. Pristupljeno: 5. 8. 2024.
- [32] SpaceSeven. An overview of the 3 most popular nft blockchains. *Medium*, 2021. Pristupljeno: 11. 4. 2024.
- [33] XP Network. Explore the 3 biggest nft marketplace of 2024: Opensea is only no. 3, 2024. Pristupljeno: 10. 8. 2024.
- [34] Halborn. Explained: The opensea nft listing vulnerability. Pristupljeno: 10. 8. 2024.
- [35] Cointelegraph. Opensea phishing scandal reveals a security need across the nft landscape. Pristupljeno: 10. 8. 2024.
- [36] Ethereum Foundation. Sharding: Simplified consensus. Pristupljeno: 5. 8. 2024.
- [37] Vibhortechnwriter. 10 popular nft use cases in 2024. Pristupljeno: 28. 8. 2024.
- [38] Hardhat. Hardhat: Ethereum development environment.
- [39] OpenZeppelin. Safemath: Arithmetic library for secure integer operations.
- [40] Chai Contributors. Chai assertion library.
- [41] Metamask. How to customize token approvals with a spending cap.
- [42] Ethereum Foundation. Introduction to ether.
- [43] Obvious Wallet. Understanding gas fees in crypto - obvious wallet, 2022. Pristupljeno: 10. 8. 2024.