SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 662

SUSTAV ZA ANALITIČKE USLUGE NAD PRIJEDLOZIMA ZA PATENTE SA ZAŠTITOM IZVORNOSTI

Florijan Rusac

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 662

SUSTAV ZA ANALITIČKE USLUGE NAD PRIJEDLOZIMA ZA PATENTE SA ZAŠTITOM IZVORNOSTI

Florijan Rusac

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 11. ožujka 2022.

ZAVRŠNI ZADATAK br. 662

Pristupnik:	**Florijan Rusac (0036525196)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	doc. dr. sc. Mario Brčić

Zadatak: **Sustav za analitičke usluge nad prijedlozima za patente sa zaštitom izvornosti**

Opis zadatka:

Zaštita intelektualnog vlasništva patentima je osjetljiva tema. U sklopu prijave je potrebno provjeriti postojeće patente da se ustanovi prethodno stanje na kojem se gradi i da se osigura originalnost. Danas se prijavljuje i odobrava više patenata no ikad. S druge strane, mogućnosti prihvata i obrade informacija u ljudi su ostale praktički nepromijenjene. Napredne analitičke metode nad patentnim prijavama zahtijevaju od korisnika da se izloži riziku krađe ideje. Stoga je potrebno podržati zaštitu izvornosti predanih podataka na različite analize koristeći principe bazirane na teoriji igara koji povećavaju povjerenje i smanjuju asimetriju u odnosu davatelja i primatelja usluge. U ovom radu će se napraviti sustav koji servira različite analitičke modele te za njih ostvaruje zaštitu izvornosti pohranom potvrde upita s kriptografskim sažetkom predane ideje na blok-lanac. Trebaju biti podržani i javni i privatni blok-lanac te sustav treba podržavati različite modele.

Rok za predaju rada: 10. lipnja 2022.

Sadržaj

Uvod

U ovom radu rješava se problem zaštite izvornosti patenta i izrađuje se sustav za serviranje različitih analitičkih modela. Danas se prijavljuje velika količina patenata, ali mogućnosti prihvata i obrade informacija kod ljudi su ostale iste. Zato treba razviti sustav koji će bilježiti patente koji su predani kako bi se u budućnosti moglo provjeriti je li predani patent već spremljen na primjerice bloklanac koji može biti javni i privatni. Javni bloklanac na koji će se spremati sažetak patenata je Solana, dok je privatni Hyperledger Fabric. Nakon što je sažetak patenta spremljen na bloklanac treba patent proslijediti jednom od više analitičkih modela. Analitički model može primjerice svrstavati patent u jednu od više grupa patenata. Grupe mogu biti primjerice po području industrije u koju patent spada.

Sustav za zaštitu izvornosti patenata i serviranje analitičkih modela je izveden kao backend Spring Boot aplikacija. Aplikacija je povezana s bazom podataka, prima HTTP (*Hypertext Transfer Protocol*) zahtjeve, povezana je s modelima, Hyperledger Fabricom i Solanom. Aplikacija štiti izvornost patenata tako da provodi transakcije na bloklancima i kao odredišnu adresu postavlja sažetak patenta koji će sada ostati nepromijenjen na bloklancu. Aplikacija prosljeđuje patent na analizu modelu koji je zadan u zahtjevu.

U prvom poglavlju se piše o potrebi za provjerom izvornosti podataka, čemu ona služi, o upravljanju podacima i kako ostvariti provjeru izvornosti podataka. U drugom poglavlju se piše o bloklancu koji može poslužiti za upravljanje podacima tako da se podaci stavljaju na bloklanac. U trećem poglavlju riječ je o aplikaciji koja služi za zaštitu izvornosti patenata i serviranje analitičkih modela. U posljednjem poglavlju dan je primjer slanja zahtjeva aplikaciji i što se tada događa.

1. Potreba za provjerom izvornosti podataka

Pojam izvornosti podataka (engl. *data provenance*) se odnosi na trag zapisa koji objašnjava podrijetlo podatka, primjerice u bazi podataka, repozitoriju ili dokumentu. Primjerice u polju poput molekularne biologije mnogo se podataka izvlači iz javnih baza podataka. Te javne baze podataka su možda nastale iz istraživačkih radova, ali nakon nekih transformacija, gdje su samo najvažniji i najrelevantniji podatci su stavljeni u baze podataka. Istraživački radovi su stvoreni iz eksperimentalnih opažanja. Ovo je trag izvornosti podataka [1].

Veliko pitanje koje se postavlja je kako i može li se vjerovati podatcima na koje se oslanjamo. Koliko se možemo pouzdati u podatke ovisi o mnogim varijablama, primjerice od kuda dolaze podatci i kako su procesuirani. Ako znamo odgovore na ta pitanja možemo procijeniti jesu li tvrđeni izvori podataka stvarni, jesu li podaci u originalnom obliku ili izmijenjeni i koriste li se na način koji je objašnjiv i legitiman. U stvarnosti ova pitanja su odgovorena tako da se prati povijest podataka. Za povjesničare umjetnosti i trgovce ova pitanja su jako važna i trag povijesti umjetnina se jako strogo mora bilježiti kako bi se spriječile krivotvorine umjetnina i omogućila restitucija ukradenih umjetnina. U logistici i obradi hrane također je važan trag povijesti kako bi se zaštitilo zdravstvo populacije [2].

U polju analitike podataka trag povijesti podataka postaje manje jasan nego što je u stvarnom svijetu. U praksi, taj trag se ostvaruje serijom obrada koje su načinjene na podatcima [2].

Povijesni trag podataka je područje u kojem su velika komercijalna rješenja još uvijek velikim dijelom nedostupna. Postoji mnogo problema i neki od njih nisu još rješivi ili ih je teško riješiti, ali radi se aktivno na istraživanju tog područja. Neki problemi nisu rješivi iz tehnološke perspektive [2].

Sljedivost podataka je više uobičajena u područjima logistike, projektiranja proizvodnog procesa i inženjerstva zahtjeva. Sljedivost se može podijeliti na dva tipa, unaprijedno i unazadno. Unaprijedno počinje na izvoru podataka, dok unazadno počinje na zadnjem podatku kojeg imamo i slijedi podatke to njenog izvora. U kontekstu bankarstva, sljedivost podataka bi se referirala na praćenje stanja depozita kroz cijeli njegov život, tj. od kada je

novac uplaćen na račun do kada je potpuno povučen s računa uključujući i prijenose na druge račune i djelomične prijenose novca [2].

1.1. Revizija podataka

Postoje dva oblika revizije podataka. Jedan je proces u kojem se podaci vrednuju prema različitim kriterijima koji se odnose na određenu svrhu. Tu spada namjeravana upotreba podataka, postojeća kvaliteta podataka i metodologije prikupljanja podataka. Drugo je proces bilježenja modifikacija podataka, što uključuje praćenje aplikacija i korisnika koji su pristupili podacima. To je relevantno kada se želi znati tko je, kada i kojim redoslijedom izmijenio podatke [2].

Revizije podataka u smislu evaluacije mogu biti korisne za primjerice procjenu kvalitete podataka što pruža uvid u korisnost podataka. Jedan način za procjenu kvalitete podatak je promatranje koliko praznih vrijednosti ima u podacima. Za bilježenje modifikacija podataka, može se koristiti bloklanac. U područjima s osjetljivim informacijama revizije pristupa podatcima su uobičajene, tj. provode se u određenim vremenskim intervalima [2].

1.2. Upravljanje digitalnim pravima (DRM)

Upravljanje digitalnim pravima (engl. *Digital Rights Managemenent*, DRM) se postiže skupinom tehnologija kontrole pristupa koje se koriste za upravljanje materijalima koji su zaštićeni autorskim pravima i za sprječavanje nelegalnog pristupa digitalnom sadržaju. Sljedivost podataka može pomoći forsirati legitimno korištenje i spriječiti digitalno piratstvo jer se postiže trag od izvora podataka do korisnika. Uz ostalo, upravljanje digitalnim pravima je odgovorno za aspekte autentifikacije i autorizacije, dozvola i plaćanja i kontrolu korištenosti. Upravljanje digitalnim pravima opisuje interakciju između tri entiteta, korisnika, sadržaja i prava. Prava znače koji korisnici mogu pristupiti kojem sadržaju.

Primjeri korištenja upravljanja digitalnim pravima se mogu pronaći u medijima, primjerice na internetu kada se kontrolira koliko uređaja pristupa istom sadržaju na internetskim servisima za gledanje filmova ili serija [2].

1.3. Prava nad podacima i korištenjem podataka

Prava nad podacima su važna jer govore o tome tko ima pristup podacima, tko ih može prenijeti i tko ih može promijeniti. Prava nad podacima se mogu odnositi na kontrolu nad podacima i posjed nad podacima. Pojedinac koji ima prava nad podacima može dodijeliti, podijeliti ili maknuti pristup trećim stranama. Korištenje podataka se odnosi na to tko ima dopuštenja za pristup stvaranje i izmjenu podataka [2].

Korištenje podataka se uobičajeno implementira tako da se stvore datoteke kojima određeni korisnik ima određena dopuštenja, poput čitanja ili pisanja u njih [2].

1.4. Upravljanje podacima

Upravljanje podacima se može definirati kao korištenje ovlasti u kombinaciji s politikom kako bi se osiguralo ispravno upravljanje informacijskom imovinom. Upravljanje podacima uobičajeno definira politike, standarde i postupke kako bi se osigurala kvaliteta podataka i omogućilo praćenje usklađenosti. Podaci mogu biti jedno od najvažnije imovine neke tvrtke i to čini upravljanje podacima ključnim instrumentom za upravljanje kvalitetom podataka. Dosljedno upravljanje podacima i visoka kvaliteta podatak jamči da se korisnost podataka održava i da se korisnost povećava. Upravljanje podacima ključno je za usklađivanje s GDPR-om (Općom uredbi o zaštiti podataka) i prijedlogu o uvođenju Zakona o upravljanju podacima [2].

Na mjestima gdje se koristi zajedničko uređivanje, poput Wikidatae, upravljanje podacima ima jako važnu ulogu. Wikidata je baza znanja na kojoj je izgrađena Wikipedija i koja je jako puno puta korištena za zlonamjerne unose od korisnika ili tvrtki radi osobne dobiti. Upravljanje podacima sprječava te zlonamjerne unose jer se oni mogu naknadno otkriti i poništiti. U tablici 1.1 dane su definicije osnovnih pojmova [2].

Tablica 1.1 Definicije pojmova

Pojam	Definicija
Sljedivost podataka	Praćenje kretanja (i promjena) podataka između ishodišta i odredišta.
Revizija podataka	Praćenje izmjena podataka.

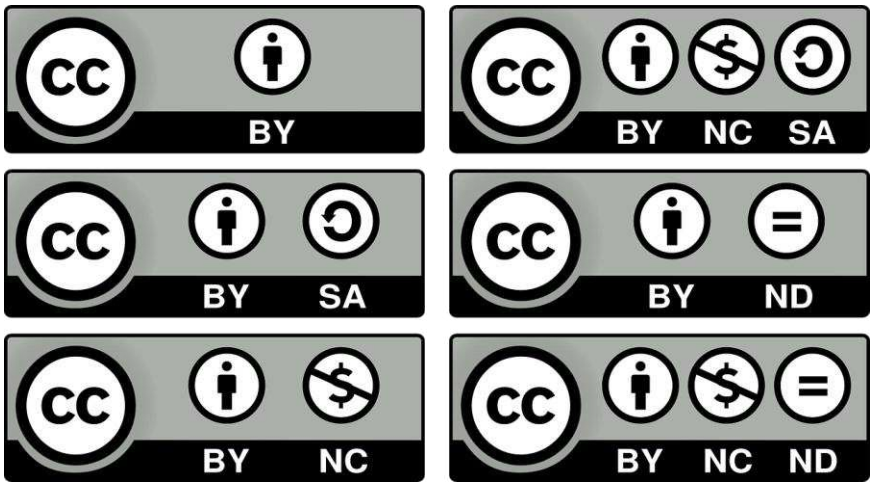
Revizija pristupa podacima	Praćenje pristupa podacima.
Upravljanje digitalnim pravima	Mehanizmi za sprječavanje nezakonitog pristupa materijalu zaštićenom autorskim pravima.
Prava na podatke	Definicija strana koje imaju dopuštenje i kontrolu podataka.
Korištenje podataka	Dozvole za čitanje i/ili izmjenu podataka.
Upravljanje podacima	Skup politika i pravila koja osiguravaju određenu razinu kvalitete podataka.

1.5. Provedba ugovornih uvjeta i ograničenja

Provedba ugovornih uvjeta u današnjem dobu igra veliku ulogu kao i u analognim vremenima. Svaka transakcija između dvije strane često ima i posrednika koji predstavlja pouzdanu treću stranu koji upravlja transakcijom. Pošto se smatra da je posrednik pouzdan dvije strane u transakciji ne moraju brinuti o mogućoj prijevari jer je posrednik odgovoran za zaštitu obje strane, tj. transakcije [3].

Problem se rješava pomoću korištenja pametnih ugovora o kojima će više biti rečeno kasnije, koji su bazirani na bloklancu. Drugo rješenje leži u uporabi ugovornih sporazuma, uglavnom putem licenci kako bi se regulirala ponovna uporaba podataka. U današnjem okruženju dijeljenje podataka je olakšano jer je većina podataka u oblaku, i to od pojedinaca, tvrtki i organizacija [3].

Ako su podatci u oblaku (engl. *cloud*), dijeljenjem podatak s trećim stranama vlasnik gubi kontrolu nad podatcima. CC licence (slika 1.1) djelomično rješavaju taj problem tako da omogućuju dijeljenje podataka pod određenim uvjetima. No javlja se novi problem koji je kako osigurati da su ti uvjeti ispunjeni. To se može riješiti ako imamo dobru sljedivost podataka [3].



Slika 1.1 CC licence [4]

U praksi se koriste funkcije sažetka (engl. *hash functions*) koje izračunavaju vrijednost na osnovu originalnih podataka. Sa svakom daljnjom transformacijom podatak funkcija sažetka se ponovno treba izračunati i zapisati, tako se stvara lanac porijekla. Prednost ovog pristupa je da podatci ne dobivaju previše novog sadržaja jer rezultati funkcije sažetka uzimaju malo prostora [3].

2. Bloklanac

Bloklanac se može promatrati kao decentralizirana baza podataka distribuirana na čvorovima. Čvor je bitna stavka bloklanac mreže jer se na njemu pokreće softver bloklanac protokola i pohranjuje povijest transakcija [13]. Najpoznatija primjena bloklanca je kod kriptovaluta, kao što je primjerice Bitcoin. Uporaba tehnologije bloklanca kod Bitcoina je važna jer osigurava siguran i decentraliziran zapis transakcija tako da se zapis transakcija nalazi na mnogo računala i na svima mora biti isti. Baš u tome što se osigurava sigurnost i točnost podataka (transakcija) bez potrebe za trećom stranom kojoj se vjeruje leži inovativnost bloklanca. Razlika između tipične baze podataka i bloklanca je to što se podaci u pohranjuju u blokovima koji su ograničenog kapaciteta. Kada se jedan napuni stvara se drugi koji se povezuje s prethodnim. Od ovdje dolazi ime bloklanac. Blokovi se pune i stvaraju slijedno tako da promatrajući ih od prvog do posljednjeg čine vremenski tok. Blokovi, jednom kada se ubace u mrežu su neizmjenjivi osim ako se većina mreže ne složi za neku izmjenu [5].

Cilj bloklanca je da podaci koji su pohranjeni u njemu budu nepromjenjivi jednom kada su stavljeni na bloklanac. Tako se bloklanac može koristiti za provođenje transakcija čiji zapisi u blokovima se nazivaju knjiga (engl. *ledger*). Zato se bloklanac naziva i tehnologija distribuirane knjige. Od prve poznatije upotrebe za Bitcoin 2009. godine, bloklanac se koristi još i za ostale kriptovalute, nezamjenjive tokene (NFT) i pametne ugovore [5].

Prednost bloklanca je u tome što pruža visoku redundanciju. Primjerice kada bi postajala baza podataka na 10 000 računala, ako samo jedno od tih računala prestane raditi baza podataka više neće biti važeća jer je bila podijeljena na 10 000 dijelova i sada jedan dio nedostaje. Kod bloklanca svi podaci se nalaze na svim čvorovima i ako se jedan čvor izgubi to ne predstavlja problem pošto postoji mnoštvo drugih čvorova na kojima su pohranjeni isti podaci. Ovo također osigurava da ako netko pokuša napraviti ilegalnu promjenu na jednom čvoru, ona neće biti priznata pošto na većini drugih čvorova ta promjena nije važeća. Zbog ovoga su pohranjene informacije i transakcije na bloklancu nepromjenjive. Ostale informacije koje mogu biti pohranjene su primjerice pravni ugovori, državne identifikacije ili inventar proizvoda tvrtke [5].

S obzirom na to da je bloklanac decentraliziran, sve transakcije se mogu pogledati putem vlastitog čvora ili tako da se koristi istraživač bloklanca. Svaki čvor ima sve zapise i dodaje nove kada se blokovi potvrde što znači da se primjerice tok bitcoina može pratiti pomoću liste transakcija. Jedan primjer praćenja novih transakcija bitcoina je da ako se provali neka mjenjačnica kriptovaluta, kriptovalute koje su ukradene će moći biti praćene kroz koje račune putuju, iako sam provalnik može ostati anoniman [5].

Novi blokovi u bloklancu su uvijek pohranjeni kronološki na kraj lanca. Nakon što je blok dodan jako je teško vratiti se natrag i promijeniti sadržaj bloka, osim ako se većina mreže složi da to treba napraviti. Uz to svaki blok sadrži svoj hash i hash bloka prije njega. Kada bi se bilo koji dio bloka promijenio, hash bloka bi bio drugačiji i bloklanac ne bi bio važeći. Hash je niz brojeva i slova koji je izračunat na temelju nekog ulaza. Recimo da postoji maliciozni korisnik koji bi htio izmijeniti sadržaj bloka u svoju korist i da taj maliciozni korisnik posjeduje čvor na kojem može staviti svoj blok. Kako su svi podaci podijeljeni na sve čvorove, mreža može jednostavno usporediti sadržaj na svojim čvorovima sa sadržajem čvora malicioznog korisnika i utvrditi da se ne poklapa i tako izdvojiti čvor malicioznog korisnika kao nevažeći. Takav napad bi uspio kada bi maliciozni korisnik kontrolirao više od pola čvorova i onda bi njegov lanac postao „pravi“ lanac. Zbog veličine većine mreža takav napad je nemoguć u stvarnosti zbog količine resursa, novčanih i računalnih, potrebnih za ostvarenje napada. Također, članovi mreže bi primijetili da se nešto zbiva i odlučili napraviti nešto što se naziva „hard fork“, tj. odvojili bi svoj lanac od lanca napadača. Sada bi bilo poznato koji je lanac napadača i njegova vrijednost bi efektivno postala nula, što čini napad neisplativim [5].

2.1. Bitcoin i bloklanac

Bitcoin je prva velika implementacija bloklanca. Bitcoin mreža je stvorena 2009. U istraživačkom radu, izumitelj Bitcoina pod pseudonimom Satoshi Nakamoto rekao je o Bitcoinu da je to „novi elektronički sustav gotovine koji je potpuno ravnopravan, bez potrebe za trećom stranom od povjerenja“. Bitcoin koristi bloklanac za transparentno bilježenje transakcija u knjigu, tj. sprema podatke na bloklanac isto kao što bi se bloklanac mogao koristiti za spremanje ostalih tipova podataka. Primjerice bloklanac bi se mogao iskoristiti za sigurno glasanje u demokratskim izborima zbog toga što su podatci na bloklancu neizmjenjivi do varanja na izborima bi bilo jako teško doći. Glasanje bi moglo funkcionirati tako da svaki građanin dobije jednu jedinicu kriptovalute i pošalje ju na

adresu svog izbornika. Transparentnost i mogućnost praćenja transakcija bi spriječile varanje [5].

2.2. Banke i bloklanac

Blockchain i kriptovalute su nazivane velikim poremećajima u sektoru bankarstva, no banke i sustav kripovaluta su poprilično različiti. Oba sustava imaju sustav plaćanja pristojba za provođenje transakcija. Pristojbe kod kriptovaluta mogu biti manje ili veće od onih kod tipičnih banaka, no ako korisnik odabere premalu pristojbu njegova transakcija možda neće biti provedena ili će trebati dugo dok se ne provede. Kod banaka provođenje transakcija tipično traje i po par dana, dok kod Bitcoina provođenje transakcije traje od 15 minuta do sat vremena, ovisno o tome koliko je mreža zagušena. Banke moraju znati identitet svojih korisnika, dok kod Bitcoina korisnik može ostati anoniman, bilo tko može otvoriti Bitcoin račun. Transakcije kod Bitcoina se mogu provoditi samo uz internetsku vezu, dok kod banaka korisnik treba imati potvrđen identitet, broj mobitela i bankovni račun, kod Bitcoina je provođenje transakcija lakše zbog manjeg broja potvrda identiteta. Privatnost kod banaka je puno manja jer moraju imati potvrdu o identitetu klijenta, dok to nije slučaj kod Bitcoina, kod Bitcoina je anonimnost korisnika moguća, jedni se provođenje transakcija može pratiti. Banke mogu odbiti provođenje transakcija ili zamrznuti račun, dok to nije slučaj kod Bitcoina [5].

2.3. Uporaba bloklanca

Bloklanac ima mnogo uporaba, neke od njih su navedene u nastavku.

Banke mogu jako profitirati od Bitcoin protokola jer ubrzava provođenje transakcija s nekoliko dana na nekoliko minuta, tj. na vrijeme potrebno da se novi blok doda u novi lanac. Također, banke imaju radno vrijeme i ne rade blagdanima, dok je bloklanac uvijek aktivan [5].

Bloklanac je osnova za kriptovalute. Valute, poput američkog dolara, su kontrolirane od strane vlade i to vlada može zlouporabiti u svoju korist. U državama s nestabilnim valutama kriptovalute su ozbiljna alternativa, kao i u državama u kojima je ratno stanje. Bloklanac ne treba centralnu vlast i tako je odvojen od prethodno navedenih problema. To također smanjuje naknade od procesiranja i provođenja transakcija [5].

Medicinski podaci se mogu pohranjivati na bloklanac i tako osigurati njihovu kronologiju i ispravnost. Također, ti podaci se mogu enkriptirati tako da se osigura povjerljivost uz pomoć privatnog ključa [5].

Bloklanac ima potencijal eliminirati traženje fizičkih zapisa vlasništva. Ako se pohrane na bloklanac osigurava se kronologija i ispravnost, pogotovo u državama gdje državna infrastruktura nije na dovoljno visokoj razini da bi se mogla jamčiti ispravnost podataka [5].

Pametni ugovor je kompjuterski kod koji se može ugraditi u bloklanac kako bi provodio ugovor. Primjerice recimo da bi stanar htio unajmiti stan od posjednika koristeći pametni ugovor. U pametni ugovor se može ugraditi da kada stanar plati stanarinu, a posjednik stavi kod vrata stana, kod vrata će se automatski poslati stanaru. Ako neki od uvjeta nije zadovoljen ugovor se neće provesti. Tako se eliminiraju troškovi vezani uz provođenje ugovora u stvarnom svijetu [5].

Nabavljači mogu koristiti bloklanac da zabilježe podrijetlo materijala koje su kupili [5].

Već spomenuto, bloklanac se može upotrijebiti za provođenje glasanja u modernim demokratskim državama. Upotrebom bloklanca se može povećati izlaznost glasača i varanje na izborima jer bi mijenjanje glasova nakon glasanja postalo gotovo nemoguće. Broj osoblja potreban za provođenje izbora bi bio jako smanjen, a rezultati bi bili dostupni odmah, bez potrebe za prebrojavanjem [5].

2.4. Prednosti i nedostatci bloklanca

Prednosti bloklanca su poboljšana točnost jer ljudi nisu umiješani u verifikaciju, smanjen trošak jer nema treće strane kojoj se vjeruje, decentralizacija otežava kvar sustava, transakcije su sigurne, privatne i efikasne, tehnologija je transparentna, pružanje alternativa bankarskim sustavima državama koje su nestabilne ili nedovoljno razvijene. Nedostatci su veliki tehnološko trošak u vezi rudarenja Bitcoina, malo transakcija po sekundi, povijest korištenja u nelegalnim aktivnostima, poput mračnog interneta (engl. *dark net*), regulacija je ovisna o državi i nije potpuno poznata [5].

Transakcije su potvrđene od strane tisuća računala. Kada bi jedno računalo napravilo pogrešku, ona bi se morala ponoviti na više od pola računala kako bi bila priznata, što je u praksi nemoguće. To je velika prednost naspram ljudi koji znaju raditi pogreške [5].

Tipično, klijenti plate banki provedbu transakcije, kako kod bloklanca nema treće strane poput banke takvog troška nema. Primjerice kada tvrtke prihvaćaju kartična plaćanja moraju platiti naknadu izdavaču kartice [5].

Transakcije provedene kroz banku mogu potrajati par dana, primjerice ako se transakcija pokrene u petak, mogla bi se izvršiti tek u ponedjeljak. Dok centralne institucije rade tipično pet dana u tjednu i to ne cijeli dan, bloklanac je uvijek aktivan. Transakcija provedena na bloklancu je najčešće izvršena unutar 10 minuta i može se smatrati sigurnom već nakon nekoliko sati. Ovo je posebno korisno za plaćanja koja prelaze granice država, jer eliminira probleme vezane uz vremenske zone i činjenicu da sve strane moraju potvrditi plaćanje [5].

Mnogo mreža bloklanca funkcionira na način da je povijest svih transakcija javno dostupna. Iako je povijest transakcija dostupna, nije dostupno tko je proveo te transakcije. Česta je zabluda da su bloklanac mreže anonimne, kada su zapravo povjerljive. Kada korisnik napravi transakciju njegov javni ključ je spremljen na bloklanac, njegove osobone informacije nisu [5].

Nakon što je transakcija provedena, njezina autentičnost mora biti potvrđena od strane bloklanac mreže. Na bloklancu svaki blok sadrži svoj hash i hash bloka prije sebe. Kada se podatci u bloku mijenjaju promijenit će se i njegov hash, no hash u sljedećem bloku ostaje isti što razbija lanac. Ovo iznimno otežava izmjenu informacija na bloklancu [5].

Većina bloklanaca su program otvorenog koda (engl. *open source*) što znači da svatko može vidjeti kod bloklanca. To znači da svatko može provjeriti je li kod siguran i svatko može predložiti izmjene koda. Ako se većina slaže da bi kod trebalo izmijeniti, Bitcoin protokol se nadograđuje [5].

Iako su naknade malene za provođenje transakcija, tehnologija je daleko od besplatne. Sustav verifikacije koji Bitcoin koristi troši jako puno računalne snage. U stvarnom svijetu, snaga računala Bitcoin mreže je bliska onoj koju Norveška i Ukrajina potroše u godini dana [5].

Unatoč velikim troškovima, korisnici nastavljaju dodavati računala u Bitcoin mrežu jer bitcoini koje dobiju od dodavanja novog bloka u mrežu su veći od troška struje. Naspram tome, ako je riječ o bloklancima koji ne koriste kriptovalute, rudari će morati biti isplaćeni na neki drugi način kako bi bili potaknuti da provode transakcije [5].

Neka rješenja prevelikog troška energije se javljaju u obnovljivim izvorima energije, primjerice farme za rudarenje bitcoina mogu koristiti energiju iz solarnih elektrana ili vjetroelektrana [5].

Sustav „dokaz rada“ (engl. *proof of work*) kojeg koristi Bitcoin je jako neefikasan jer mu treba 10 minuta da doda novi blok u bloklanac. Tom brzinom, procijenjeno je da mreža može podržavati samo 7 transakcija po sekundi. Visa, primjerice može procesirati 65 000 transakcija po sekundi. Rješenja ovog problema su novi bloklanci koji mogu procesirati 30 000 transakcija po sekundi. Još jedan problem je da u svakom bloku može biti ograničena količina podataka [5]. Veličina bloka je veliko pitanje u tehnologiji bloklanca jer ona utječe na to koliko će bloklanac moći provoditi transakcija u sekundi [14].

Dok povjerljivost bloklanac mreže osigurava privatnost i štiti korisnike od provale, također ilegalna tržišta i aktivnosti na bloklanac mreži. Vjerojatno najpoznatiji primjer uporabe bloklanca za ilegalne aktivnosti je „Cesta svile“ (engl. *silk road*), ilegalna tržnica na kojoj su se prodavale droge i prao novac, uz ostale nelegalne stvari. Tržnica je bila aktivna u periodu od veljače 2011. godine do listopada 2013. godine kada je ugašena od strane FBI-ja. Korisnici su koristili TOR preglednik koji im je omogućavao anonimnost na internetu, a transakcije su se provodile u Bitcoinu ili drugim kriptovalutama. Danas je mali udio transakcija u Bitcoinu ilegalne prirode, i svejedno većina ilegalnih aktivnosti se provodi pomoću gotovine koju se ne može pratiti [5].

Postoji zabrinutost oko vladine regulacije kriptovaluta. Iako je gotovo nemoguće ugasiti nešto toliko veliko i decentralizirano kao što je Bitcoin mreža, moguće je da vlada učini kriptovalute ilegalnima. No, kako kompanije poput Paypala počinju prihvaćati kriptovalute na svojim platformama ta se mogućnost smanjuje [5].

2.5. Razlika između javnog i privatnog bloklanca

U javnom bloklancu se bilo tko može pridružiti mreži i uspostaviti čvor. Zbog otvorene naravi, takvi sustavi trebaju biti kriptografski zaštićeni i imati sustav konsenzusa kao što je dokaz rada. U privatnom bloklancu svaki čvor mora zahtijevati dopuštenje prije nego je uključen u mrežu. Zato jer se smatra da se čvorovima vjeruje, sigurnost može biti nešto olakšana [5].

2.6. Solana

Solanu je stvorio Anatoly Yakovenko 2017. godine. Testnet je izašao 2018. godine, dok je glavna mreža lansirana 2020. godine, iako je ona još uvijek u beta verziji. To je bloklanac čija je namjera procesirati jako puno transakcija u sekundi i održati troškove niskima u isto vrijeme. Solana implementira hibridan sustav konsenzusa koji se oslanja na „Proof of History“ (PoH) algoritam i na „Proof of Stake“ (PoS) algoritam. To omogućava teoretsko procesiranje transakcija od 710 000 transakcija po sekundi, što je puno više od već spomenute Vise, koja omogućava, za usporedbu 65 000 transakcija po sekundi [6].

Arhitektura bloklanca Solane podržava pametne ugovore, decentralizirane aplikacije i NFT tržnice. Decentralizirane aplikacije su aplikacije koje rade na bloklanac mreži računala umjesto na jednom računalu [7]. „Non-fungible token“ (NFT) je unikatan token kojeg se ne može zamijeniti za nešto drugo. Primjerice Bitcoin je zamjenjiv, jer se može zamijeniti jedan Bitcoin za drugi i završiti u istoj poziciji, NFT-ove se ne može tako mijenjati. O NFT-ovima se može razmišljati kao o jedinstvenoj karti koja se može prodavati i posjedovati i pohranjena je na bloklancu [8].

Solana želi riješiti bloklanac „trilemu“ koju je predstavio osnivač Ethereuma Vitalik Buterin. Trilema opisuje tri problema koje developeri moraju riješiti kada grade bloklanac: decentralizacija, sigurnosti i skalabilnost. Vjeruje se da se kod bloklanaca često jedno žrtvuje u korist druga dva, jer je svo troje teško postići u isto vrijeme. Inovativna kombinacija PoS i PoH algoritma kod Solane uspijeva ubrzati procesiranje transakcija u sekundi. Općenito, kod decentraliziranih bloklanaca više čvorova znači da će trebati više vremena da se potvrdi transakcija. Kod Solane taj problem je riješen jer se bira jedan čvor koji će biti vođa pomoću PoS mehanizma koji sekvencira poruke između čvorova, što ubrzava protok podatak i zadržava decentralizaciju. Također, Solana stvara lanac transakcija tako da izračunava sažetak jedne transakcije i koristi ga kao ulaz druge transakcije. Povijest transakcija daje ime algoritmu PoH i omogućava veću skalabilnost protokola [6].

PoS mehanizam funkcionira koristeći algoritam koji odabire sudionike s najvećim ulozima kao validatore, tu je pretpostavka da su sudionici s najvećim ulozima potaknuti da osiguraju obradu transakcije. Ideja je da oni s najviše valute imaju najviše za izgubiti i zato su pozicionirani da rade u interesu mreže. Količina valute koju mreža može zahtijevati se mijenja s vremenom [9].

U PoS-u blokove ne stvaraju rudari koji rade posao, nego sudionici koji ulažu svoje tokene kako bi se kladili na to koji su blokovi valjani. Pod pretpostavkom da će većina glasati za ispravni novi blok, oni koji ne glasaju za ispravan blok gube svoj novac [9].

Umjesto da se vjeruje vremenskoj oznaci transakcije, može se dokazati da se transakcija dogodila prije nekog događaja i nakon nekog događaja. PoH je visokofrekventna funkcija odgode. Takva funkcija zahtijeva određeni broj koraka da se evaluira i proizvodi jedinstveni izlaz koji se može javno i efikasno potvrditi [10].

Dokaz povijesti je slijed računanja koji može pružiti način za kriptografsku provjeru prolaska vremena između dva događaja. Koristi se funkcija koja je kriptografski sigurna i napisana tako da se izlaz ne može predvidjeti iz ulaza. Funkcija se mora potpuno izvršiti da bi se generirao izlaz. Funkcija radi u nizu na jednoj jezgri, uzimajući prijašnji izlaz kao trenutni ulaz. Periodično snima izlaz i koliko puta je bio pozvan. Izlaz se tada može ponovno izračunati i provjeriti na drugim računalima paralelno provjeravanjem svakog segmenta sekvence na različitoj jezgri. Podaci se mogu označiti, ili njihov sažetak, dodavanjem podataka u stanje funkcije. Snimanje stanja, indeksa i podataka kako su dodani u sekvence daje vremensku oznaku koja može jamčiti da su podaci stvoreni prije nego je generiran sljedeći sažetak u nizu [10].

Kriptovaluta u Solani je SOL, lansiran u ožujku 2020. godine, SOL je postao jedna od top deset najviših kriptovaluta po ukupnoj tržišnoj kapitalizaciji, što je ukupna vrijednost kriptovalute. SOL omogućuje razmjenu vrijednosti i očuvanje sigurnosti bloklanca kroz ulaganja (eng. staking). Solana token se koristi za isplatu nagrada i za plaćanje naknada kod provođenja transakcija. U cirkulaciji je više od 500 milijuna tokena, od čega je 60% u kontroli osnivača Solane, a oko 38% je dostupno ostalima [6].

2.7. Hyperledger Fabric

Hyperledger Fabric je radni okvir za bloklanac koji traži dopuštenja za ulazak u mrežu i otvorenog je koda. Pokrenut je 2015. godine od strane The Linux Foundation. To je modularni radni okvir kojemu je svrha biti korišten za općenitu uporabu. Koristan je za puno stvari u industriji, poput praćenja lanaca opskrbe, trgovinske razmjene, lojalnosti i nagrada i rješavanja problema vezanih uz imovine [11].

Postoji knjiga koja je transakcijski dnevnik i ona ne može biti mijenjanja jednom kada se nešto upiše u nju. Sadržava cijeli zapis transakcija koje su obavljene do sada i novi podaci se mogu samo dodavati. Svaki član bloklanac mreže održava neovisnu kopiju [11].

Algoritmi konsenzusa omogućavaju upis novih podataka u knjigu i izvođenje pametnih ugovora. Također omogućava članovima da imaju dogovorenu metodu koja omogućuje izvršavanje novih transakcija. Ako metoda koja je dogovorena nije ispunjena onda se transakcija ne provodi, tj. podaci se ne upisuju u knjigu [11].

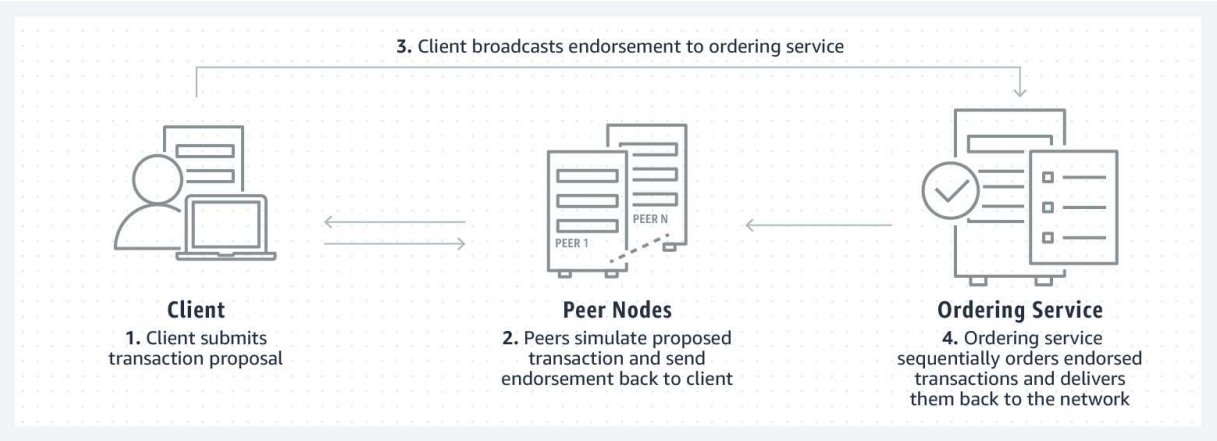
Pametni ugovori su kod koji se izvršava kada su određeni uvjeti zadovoljeni. Kod se izvršava na bloklanac mreži. Često definiraju pravila poslovnog ugovora [11].

To je radni okvir otvorenog koda sa rastućom zajednicom developera. Bloklanac traži dopuštenje za ulazak u mrežu što znači da su identiteti sudionika poznati i autentificirani što je posebno dobro u slučaju korištenja u industriji poput bankarstva, zdravstva i osiguranja gdje je jako važno da su podaci sigurni. Primjerice osiguravateljska kuća može podijliti podatke samo sa stranama kojima vjeruje pomoću bloklanca. Hyperledger Fabric podržava brzo provođenje transakcija [11].

Hyperledger Fabric se sastoji od jedinstvenih članova koji komuniciraju jedni s drugima na mreži. Primjerice članovi mogu biti banke u mreži financijskih institucija ili poštanske kompanije u lancu nabave. Svaki član ima jedan ili više čvorova. Fabric također ima sustav naručivanja kojeg dijele svi članovi u mreži. Svaki član u mreži ima svoj certifikat koji identificira korisnika ili čvor. Certifikat specificira dopuštenja na mreži poput dopuštenja samo za čitanje ili potpunog pristupa [11].

Organizacija također stvara jedan ili više ravnopravnih čvorova koji se koriste za provođenje transakcija i za pohranjivanje i izvođenje pametnih ugovora (također znano kao „chaincode“) i pohranjuje lokalnu kopiju ljestvice. Klijenti Fabrica upotrebljavaju sve navedeno, tj. provode transakcije (slika 2.1), pohranjuju kod za pametne ugovore i čitaju ljestvicu [11].

Postoji i naručiteljski servis kojeg koriste svi članovi mreže. Taj servis osigurava da su nove transakcije u pravom redoslijedu zapisane u novom bloku. Servis onda pošalje svim ravnopravnim čvorovima blok kojeg treba zapisati i ravnopravni čvorovi zapišu taj blok [11].



Slika 2.1 Komunikacija klijenta, članova i naručiteljskog servisa [11]

Hyperledeger Fabric se može koristiti kod nabavnih lanaca. Nabavni lanci su globalna mreža nabavljača, proizvođača i prodavača. Hyperledger Fabric može poboljšati transparentnost i sljedivost transakcija. Tvrtke na istoj mreži mogu vidjeti iste nepromojenjive podatke, čime se povećava odgovornost i smanjuje rizik od krivotvorenja [11].

Trgovanje zahtjeva uvoznike, izvoznike, banke, poštanske kompanije koji svi trebaju surađivati jedni s drugima. Uz pomoć Hyperledger Fabrica može se stvoriti mreža gdje se mogu zapisivati informacije vezane uz trgovanje koje bi se inače zapisivale na papire, bez potrebe za centralnim autoritetom. U ostalim procesima koji ne koriste Hyperledger Fabric papirologija treba ići naprijed i natrag više puta dok se na Hyperledger Fabricu transakcije provode instantno [11].

Hyperledger Fabric se može koristiti u industriji osiguranja. Veliki problem u industriji osiguranja su duplikatni ili falsificirani zahtjevi, no s Hyperledger Fabricom podaci zapisani na ljestvici ne mogu biti mijenjani tako da se lako provjeri koji zahtjevi su pravi ili već obrađeni [11].

3. Aplikacija za serviranje patenata modelima

U ovom poglavlju je objašnjena izvedba aplikacije koja služi za prosljeđivanje patenata različitim modelima koji te patente obrađuju na svoj način i vraćaju rezultat aplikaciji koja ga prosljeđuje natrag korisniku. Prvo je objašnjen Spring Boot na kojemu se temelji aplikacija.

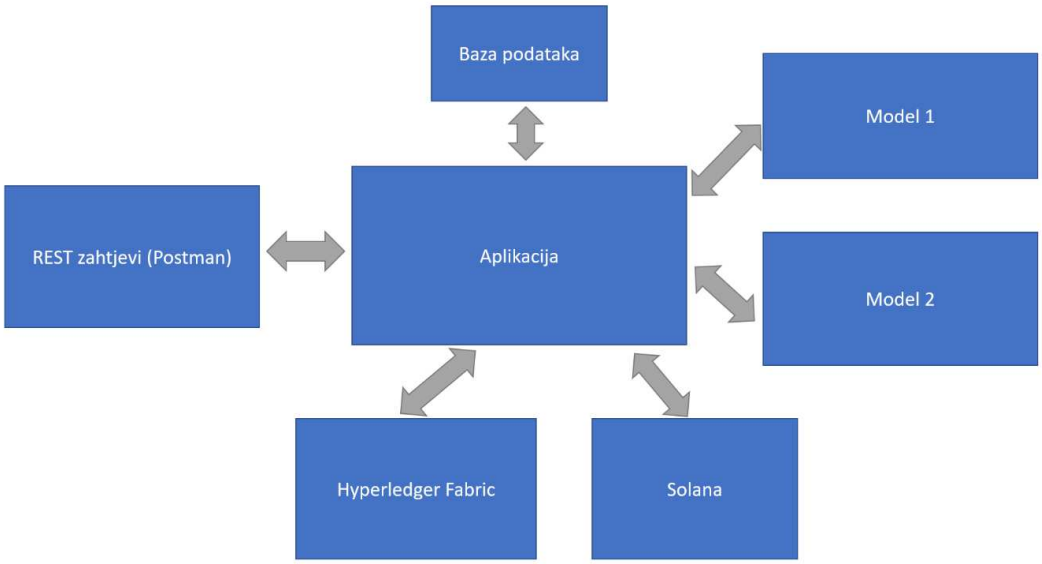
3.1. Spring Boot

Spring Boot je radni okvir baziran na programskom jeziku Javi uz pomoć kojeg se može izraditi mikro servis. Mikro servis je arhitekturni i organizacijski pristup razvoju softvera gdje se softver sastoji od malih neovisnih usluga koje komuniciraju preko dobro definiranih API-ja [15]. Mikro servisi nude sljedeće pogodnosti: lako se pokreću, lagano skaliraju, potrebna je minimalna konfiguracija i manje produkcijskog vremena [12].

Spring Boot automatski konfigurira aplikaciju ovisno o ovisnostima koji su dodani u projekt tako da se koristi anotacija @EnableAutoConfiguration. Ulazna točka Spring Boot aplikacije je klasa koja je anotirana sa @SpringBootApplication i ima main metodu. Spring Boot automatski skenira komponente uključene u projekt tako da se koristi anotacija @ComponenetScan [12].

3.2. Aplikacija

Aplikacija se sastoji od backenda koji komunicira pomoću REST zahtjeva s korisnikom i modelima. Također komunicira s PostgreSQL bazom podataka i privatnim bloklancom Hyperledger Fabric i javnim bloklancom Solana (slika 3.1).



Slika 3.1 Shema aplikacije

```
public static void main(String[] args) {
		SpringApplication.run(PatentiApplication.class,
args);
}
```

Ispis 3.1 – *Main* metoda

Ispis 3.1 prikazuje kod za pokretanje aplikacije.

```
public class PatentDTO {
	private String patentText;
	private String modelName;

	public String getPatentText() {
		return patentText;
	}

	public void setPatentText(String patentText) {
		this.patentText = patentText;
	}

	public String getModelName() {
		return modelName;
	}

	public void setModelName(String modelNumber) {
		this.modelName = modelNumber;
```

```
	}
}
```

Ispis 3.2 - *PatentDTO*

Ispis 3.2 prikazuje strukturu podataka koju prima aplikacija.

Aplikacija preko post zahtjeva koji se može poslati iz Postmana prima tekst patenta i ime modela kojemu se želi proslijediti patent.

```
@PostMapping("")
public Result req(@RequestBody PatentDTO info) {
	return myService.req(info);
}
```

Ispis 3.3 – funcija *req()*

Kada aplikacija dobije post zahtjev na ruti „/“ poziva se metoda `req()` nad objektom `myService`, kako je prikazano u ispisu 3.3.

Metoda `req()` ima više dijelova, prvo se poziva metoda `solana()` pa `hyperledgerFabric()` koje će biti objašnjene u nastavku. Također se na početku stvara objekt `res` koji predstavlja rezultat koji će biti vraćen korisniku, a sadržava grupu patenta ili slične patente i zapis transakcije na solani kako je prikazano u ispisu 3.4.

```
@Entity
@Table(name = "resultTable")
public class Result {
	@Id
	@GeneratedValue
	private Long id;

	@Column(name="valueGroup")
	private String group;
	@Column(name="valueHash")
	private String hash;

	public String getGroup() {
		return group;
	}

	public void setGroup(String group) {
		this.group = group;
	}
```

```
	public String getHash() {
		return hash;
	}

	public void setHash(String hash) {
		this.hash = hash;
	}
}
```

Ispis 3.4 – klasa *Result*

U metodi `solana()` prvo se računa SHA-256 hash patenta kako je prikazano u ispisu 3.5.

```
MessageDigest digest = null;
try {
	digest = MessageDigest.getInstance("SHA-256");
} catch (NoSuchAlgorithmException e1) {
	// TODO Auto-generated catch block
	e1.printStackTrace();
}
byte[] encodedhash =
digest.digest(info.getPatentText().getBytes(StandardCharsets.
UTF_8));
String patentHash = bytesToHex(encodedhash);
```

Ispis 3.5 – računanje hasha patenta

Onda se taj hash pretvara pomoću metode `calculatePatentHash()` u adresu koju je moguće postaviti kao adresu primaoca transakcije na solana mreži.

```
private String calculatePatentHash(String patentHash) {
	patentHash = patentHash.substring(0, 43);
	patentHash = patentHash.replaceAll("0", "A");
	patentHash = patentHash.replaceAll("I", "B");
	patentHash = patentHash.replaceAll("O", "C");
	patentHash = patentHash.replaceAll("l", "D");
	patentHash = "2" + patentHash;

	System.out.println(patentHash);
	return patentHash;
}
```

Ispis 3.6 – pretvaranje hasha

To se radi tako da se preuzmu prva 43 znaka SHA-256 sažetka i zamijene slova i brojke redom 0, I, O, l sa slovima A, B, C, D. Na kraju se još nadodaje 2 na početak sažetka kako u nekim funkcijama ne bi došlo do „buffer overflowa“. To se sve radi kako bi hash bio u formatu Base58, koji je standard za odredišne adrese računa na solana mreži. Ideja je da se pohrani sažetak patenta na bloklanac kako bi se moglo u budućnosti odrediti da taj patent postoji, tj. da ne može netko drugi predati isti patent, što bi bila krađa.

```
RpcClient client = new
RpcClient("https://api.testnet.solana.com");

PublicKey fromPublicKey = new
PublicKey("BgGL1Zbs16coZypgFbo1QH32axJYgkcyfuYuYwGEqRXF");
PublicKey toPublickKey = new PublicKey(patentHash);
int lamports = 1_000_000; // 1_000_000_000 = 1 sol

Account signer = new
Account(Base58.decode("5V6HFgkzdVqcafo5Vu4i5J1RrXs72eJ1gHbjQ4
djBkKuqvtuBiN3pu38Sr7S3UvxfCKxJA9qeVn8eMWKbkXN8zuw"));

Transaction transaction = new Transaction();
transaction.addInstruction(SystemProgram.transfer(fromPublicK
ey, toPublickKey, lamports));

String signature = null;
try {
      signature =
client.getApi().sendTransaction(transaction, signer);
} catch (RpcException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
}

return signature;
```

Ispis 3.7 – funkcija *solana()*

U ispisu 3.7 je prikazano spajanje na solanu, konkretnije na testnet solane koji služi testiranju funkcija solane. Prikazuje se stvaranje dva računa uz pomoć javnih ključeva i stvaranje računa koji će potpisati transakciju iz para ključeva. Taj račun odgovara onome koji šalje kriptovalutu u transakciji. Dalje u kodu je prikazano provođenje transakcije i dohvat potpisa transakcije koji se vraća pozivajućoj funkciji i naposljetku prosljeđuje

natrag onome koji je poslao početni zahtjev iz primjerice Postmana. Dalje u funkciji `req()` poziva se funkcija `hyperledgerFabric()`. Prvi dio računanja hasha iz patenta je isti dok se spajanje na Hyperledger Fabric razlikuje od spajanja na Solanu.

```
// Load a file system based wallet for managing identities.
Path walletPath = Paths.get("C:\\fabric-samples-
repo\\3\\fabric-samples\\fabcar\\java\\wallet");
Wallet wallet = null;
try {
      wallet = Wallets.newFileSystemWallet(walletPath);
} catch (IOException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
}
      // load a CCP
Path networkConfigPath = Paths.get(
            "C:\\fabric-samples-repo\\3\\fabric-
samples\\test-
network\\organizations\\peerOrganizations\\org1.example.com\\
connection-org1.yaml");

Gateway.Builder builder = Gateway.createBuilder();
try {
      builder.identity(wallet,
"appUser").networkConfig(networkConfigPath).discovery(true);
} catch (IOException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
}

try (Gateway gateway = builder.connect()) {

      // get the network and contract
      Network network = gateway.getNetwork("mychannel");
      Contract contract = network.getContract("fabcar");

      byte[] result = null;

      try {
            contract.submitTransaction("createCar",
patentHash, patentHash, "", "", "");
```

```
        } catch (ContractException | TimeoutException |
InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
        }
                try {
                result = contract.evaluateTransaction("queryCar",
patentHash);
        } catch (ContractException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
        }
        System.out.println(new String(result));
}
```

Ispis 3.8 – funkcija *hyperledgerFabric()*

U ispisu 3.8 je prikazano učitavanje postojećeg novčanika iz direktorija na disku. Također se učitava konfiguracija mreže koja se inače koristi za spremanje tipova automobila, no ovdje je iskorištena za spremanje sažetaka patenata. Ovdje se provodi transakcija pomoću pametnog ugovora i sprema se sažetak patenta na bloklanac. Rezultat se sprema i ispisuje se što je spremljeno pod tim sažetkom.

Nakon što je sažetak patenta spremljen na Solanu i Hyperledger Fabric gleda se kojem modelu će se poslati tekst patenta na obradu prema zaprimljenim podacima iz početnog zahtjeva, kako je prikazano u ispisu 3.9.

```
String model = info.getModelName();
String address = "http://127.0.0.1:5000/";
if (model.equalsIgnoreCase("prvi")) {
        address = "http://127.0.0.1:5000/";
}
if (model.equalsIgnoreCase("drugi")) {
        address = "http://127.0.0.1:5001/";
}
```

Ispis 3.9 – biranje modela

Na kraju se šalje POST zahtjev odabranom modelu. U tijelu post zahtjeva šalje se tekst patenta, a dobiveni rezultat se prosljeđuje natrag početnome zahtjevu kako je prikazano u ispisu 3.10.

```
CloseableHttpClient httpclient = HttpClients.createDefault();
```

```
HttpPost httppost = new HttpPost(address);

// Request parameters and other properties.
List<NameValuePair> params = new ArrayList<NameValuePair>(2);
            params.add(new BasicNameValuePair("patentText",
      info.getPatentText()));
try {
      httppost.setEntity(new UrlEncodedFormEntity(params,
"UTF-8"));
} catch (UnsupportedEncodingException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
}

// Execute and get the response.
HttpResponse response = null;
try {
      response = httpclient.execute(httppost);
} catch (IOException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
}
HttpEntity entity = response.getEntity();

if (entity != null) {
      try (InputStream instream = entity.getContent()) {
            int bufferSize = 1024;
            char[] buffer = new char[bufferSize];
            StringBuilder out = new StringBuilder();
            Reader in = new InputStreamReader(instream,
StandardCharsets.UTF_8);
            for (int numRead; (numRead = in.read(buffer, 0,
buffer.length)) > 0;) {
                  out.append(buffer, 0, numRead);
            }
            res.setGroup(out.toString());
            resultRepo.save(res);
            return res;
      } catch (UnsupportedOperationException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
```

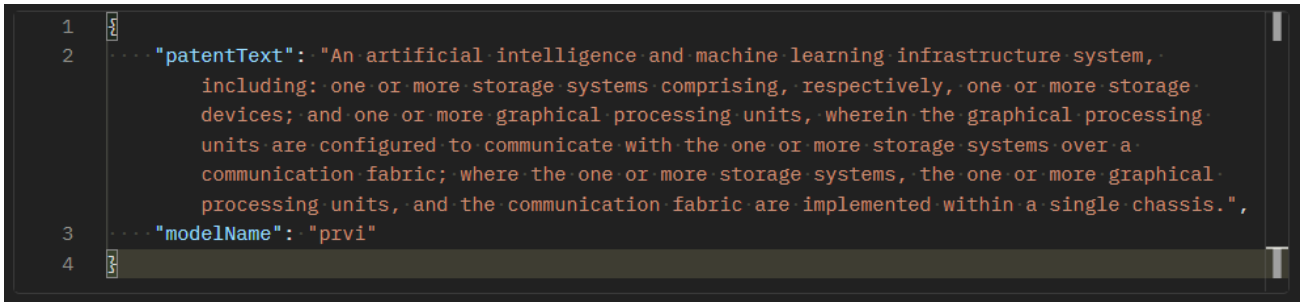
```
        } catch (IOException e) {
              // TODO Auto-generated catch block
              e.printStackTrace();
        }
}
```

Ispis 3.10 – slanje POST zahtjeva i vraćanje iz funkcije

Aplikacija je povezana i s bazom podataka u koju se sprema potpis transakcija i rezultat obrade modela.

4. Demonstracija slanja zahtjeva

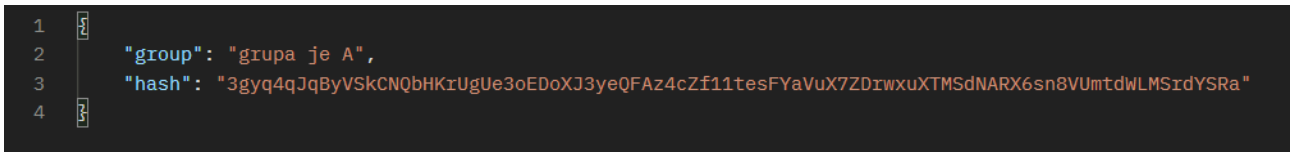
Aplikacija prima POST zahtjev s dva polja, tekstom patenta i imenom modela kojem će se proslijediti patent kako je prikazano na slici 4.1. Na slici 4.1 za tekst patenta je postavljen sažetak (engl. *abstract*) patenta za bolji prikaz, iako se može postaviti i puni tekst patenta. Za ime modela bira se model koji se zove „prvi“.



Slika 4.1 Tijelo POST zahtjeva

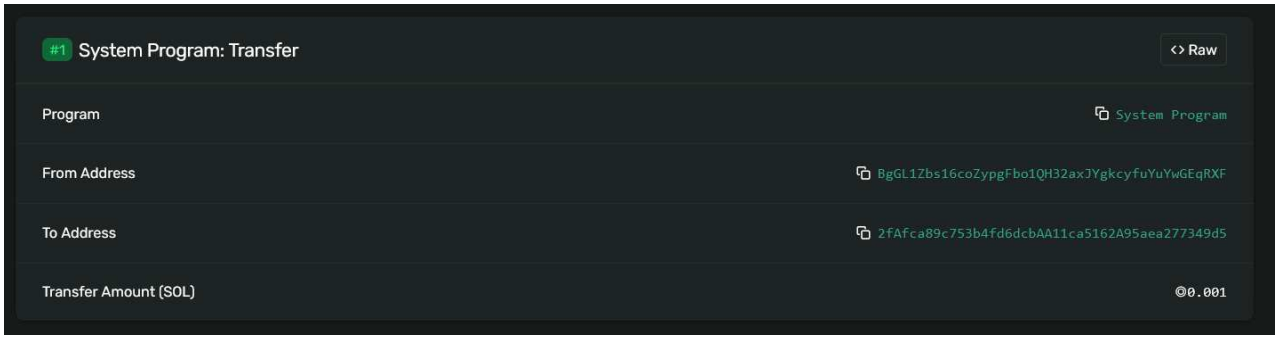
Da bi Hyperledger Fabric radio treba pokrenuti Docker i onda u Windows Subsystem for Linux (WSL) terminalu upisati naredbu `./startFabric` u direktoriju „fabcar“. Također korisnici admin i appUser trebaju biti stvoreni. Baza podataka i model kojem se šalje patent trebaju biti pokrenuti.

Kada se pošalje zahtjev dobije se odgovor prikazan na slici 4.2. Vidi se da je model svrstao patent u grupu „A“ i aplikacija je vratila sažetak transakcije pomoću kojega se može provjeriti je li sažetak patenta zapisan na bloklanac.



Slika 4.2 Odgovor na zahtjev

Ako provjerimo na https://explorer.solana.com/?cluster=testnet (solana bloklanac explorer) što je zapisano pod tim sažetkom vidimo da je kao odredišna adresa zapisan sažetak patenta, kako je prikazano na slici 4.3.



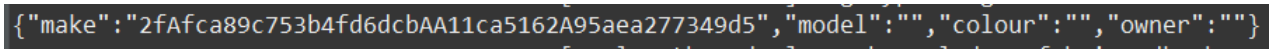
Slika 4.3 Detalji provedene transakcije

Ako pogledamo što se je zapisalo u bazu podataka, na slici 4.4 vidimo također sažetak transakcije u bazi podataka i grupu patenta.

105	grupa je A	3gyq4qJqByVSkCNQbHKrUgUe3oEDoXJ3yeQFAz4cZf11tesFYaVuX7ZDrwxuXTMSdNARX6sn8VUmtdWLMSrdYSRa

Slika 4.4 Novi zapis u bazi podataka

Još ostaje pogledati što se je zapisalo na Hyperledger Fabric, to ispisuje aplikacija u terminalu. Na slici 4.5 vidi se da je zapisan sažetak patenta.



Slika 4.5 Novi zapis na Hyperledger Fabricu

Zaključak

U sklopu ovog rada dan je opis sljedivosti podataka i opis bloklanaca i kako oni mogu biti iskorišteni za pohranu podataka. Napravljen je poslužiteljski dio aplikacije u Spring Boot radnom okruženju koji služi za zaštitu izvornosti patenata i serviranje analitičkih modela. Zaštita izvornosti patenata je izvedena pomoću računanja SHA-256 sažetka patenta i onda pretvorbe tog patenta u oblik koji može poslužiti kao odredišna adresa transakcije na bloklancu. Kada se transakcija provede, taj zapis ostaje u nepromijenjenom obliku na bloklancu. Serviranje analitičkih modela je izvedeno tako da se pošalje POST zahtjev modelu koji je naveden u početnom zahtjevu, koji će u sebi sadržavati opis patenta.

U budućnosti može se izraditi klijentski dio aplikacije tako da se ne treba koristiti Postman. Može se ostvariti pohrana podataka na bloklanac na neki drugi način koji omogućuje pohranu originalnog SHA-256 sažetka. Također se može dodati podrška za više od dva modela.

Literatura

[1] A. Gupta, "Data Provenance," in *Encyclopedia of Database Systems*, L. LIU and M. T. ÖZSU, Eds. Boston, MA: Springer US, 2009, pp. 608–608. doi: 10.1007/978-0-387-39940-9_1305.

[2] "Data provenance & lineage: technical guidance on the tracing of data - Part 1 | Support Centre for Data Sharing." https://eudatasharing.eu/technical-aspects/data-provenance-part-1 (accessed Jun. 01, 2022).

[3] "Data provenance & lineage: technical guidance on the tracing of data - Part 2 | Support Centre for Data Sharing." https://eudatasharing.eu/technical-aspects/data-provenance-part-2 (accessed Jun. 01, 2022).

[4] "What is the creative commons license? Types and how to find CC content." https://candid.technology/what-is-the-creative-commons-license-types-how-to-find/ (accessed Jun. 01, 2022).

[5] "Blockchain Definition: What You Need to Know." https://www.investopedia.com/terms/b/blockchain.asp (accessed Jun. 01, 2022).

[6] "What is Solana, and how does it work?" https://cointelegraph.com/news/what-is-solana-and-how-does-it-work (accessed Jun. 01, 2022).

[7] "Decentralized Applications (dApps) Definition." https://www.investopedia.com/terms/d/decentralized-applications-dapps.asp (accessed Jun. 01, 2022).

[8] "NFTs, explained: what they are, and why they're suddenly worth millions - The Verge." https://www.theverge.com/22310188/nft-explainer-what-is-blockchain-crypto-art-faq (accessed Jun. 01, 2022).

[9] "Proof of Stake (PoS) - consensus." https://tokens-economy.gitbook.io/consensus/chain-based-proof-of-stake/proof-of-stake-pos (accessed Jun. 01, 2022).

[10] "Proof of History - consensus." https://tokens-economy.gitbook.io/consensus/chain-based-proof-of-capacity-space/proof-of-history (accessed Jun. 01, 2022).

[11] "What is Hyperledger Fabric?" https://aws.amazon.com/blockchain/what-is-hyperledger-fabric/ (accessed Jun. 01, 2022).

[12] "Spring Boot - Introduction." https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm (accessed Jun. 01, 2022).

[13] "What are blockchain nodes?" https://blog.bitstamp.net/post/what-are-blockchain-nodes/ (accessed Jun. 05, 2022).

[14] "Bitcoin Block Size, Explained." https://cointelegraph.com/explained/bitcoin-block-size-explained (accessed Jun. 05, 2022).

[15] "What are Microservices? | AWS." https://aws.amazon.com/microservices/ (accessed Jun. 05, 2022).

Sažetak

Naslov: Sustav za analitičke usluge nad prijedlozima za patente sa zaštitom izvornosti

Sažetak: Patenti predstavljaju ključnu ulogu u zaštiti intelektualnog vlasništva. Prilikom njihove prijave, s ciljem osiguranja originalnosti, potrebno je utvrditi različitosti u odnosu na sve preostale patente. S obzirom na povećan porast patentnih prijava, ljudski kapaciteti u prihvatu i obradi istih su ograničeni. S druge strane, uporaba naprednih analitičkih metoda izlaže prijavitelja riziku krađe intelektualnog vlasništva. Kako bi se doskočilo danom problemu, potrebno je podržati zaštitu izvornosti predanih podataka. Jedan od načina za ostvarenje tog cilja je baziran na teoriji igara gdje se povećava povjerenje smanjenjem asimetrije u odnosu davatelja i primatelja usluge. U ovom je radu razrađena problematika prijave velikog broja patenata te je opisana aplikacija koja servira različite analitičke modele grupiranja patenata i pritom ostvaruje zaštitu izvornosti. Također, dan je primjer uporabe javnog (*Solana*) ili privatnog (*Hyperledger Fabric*) bloklanaca za pohranu potvrde upita s kriptografskim sažetkom predane ideje s ciljem zaštite izvornosti.

Ključne riječi: sljedivost podataka, upravljanje podacima, upravljanje digitalnim pravima, bloklanac, javni bloklanac, privatni bloklanac, Solana, Hyperledger Fabric, Spring Boot, Java, zaštita izvornosti, serviranje modela

Summary

Title: Patent Proposal Analytical Services System With Protection of Authenticity

Abstract: Patents play a crucial role in protecting intellectual property. When applying for them, it is necessary to identify differences from all other patents to ensure originality. Due to the growing number of patent applications, the human capacity to receive and process them is limited. On the other hand, advanced analytical methods expose the applicant to the risk of intellectual property theft. To address this problem, it is necessary to support the authenticity of the submitted data. One way to achieve this goal is a game theory-based where trust is increased by reducing the asymmetry between the service provider and recipient. This paper deals with the issue of filing a large number of patents and describes the application that serves different analytical models of grouping patents while achieving protection of authenticity. Also, an example is of using public (Solana) or private (Hyperledger Fabric) blockchain to store query confirmation with a cryptographic summary of the submitted idea to protect authenticity.

Keywords: data provenance, data governance, Digital Rights Management, blockchain, public blockchain, private blockchain, Solana, Hyperledger Fabric, Spring Boot, Java, protection of authenticity, serving models