Sustav za sigurno online glasovanje temeljen na lancu blokova

Gršković, Ana

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 386

SUSTAV ZA SIGURNO ONLINE GLASOVANJE TEMELJEN NA LANCU BLOKOVA

Ana Gršković

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 386

SUSTAV ZA SIGURNO ONLINE GLASOVANJE TEMELJEN NA LANCU BLOKOVA

Ana Gršković

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 386

Pristupnica:	**Ana Gršković (0036523032)**
Studij:	Računarstvo
Profil:	Računalno inženjerstvo
Mentor:	prof. dr. sc. Ivica Botički

Zadatak: **Sustav za sigurno online glasovanje temeljen na lancu blokova**

Opis zadatka:

U okviru diplomskog rada potrebno je implementirati sustav za sigurno online glasovanje koji se temelji na tehnologiji lanca blokova. Kroz vlastitu implementaciju prototipa lanca blokova potrebno je razraditi ključne sigurnosne karakteristike lanca blokova koje ga čine prikladnim temeljem sustava za glasovanje. Sustav pohranjuje glasove u blokove na lancu blokova čime osigurava integritet i neporecivost podataka te onemogućava naknadno krivotvorenje glasova. Sustav treba biti koncipiran kao decentralizirana mreža čime se eliminira ovisnost o trećoj strani, postiže skalabilnost i smanjuju troškovi. Uz rješavanje problema ljudske pogreške i značajno skraćenje trajanja procesa brojanja glasova, sustav treba olakšati glasovanje samim glasačima, pogotovo starima i nemoćnima, kao i glasačima u dijaspori.

Rok za predaju rada: 28. lipnja 2024.

Sadržaj

Uvod

Izbori su temelj demokratskih sustava koji omogućuje građanima da na anoniman i slobodan način izraze svoje političke preferencije te utječu na budući smjer svoje države. Oni predstavljaju sredstvo kojim glasači države legitimno mogu prenijeti svoju volju na izabrane predstavnike vlasti. Međutim, da bi izbori ispunili svoju svrhu, ključno je osigurati pravednost čitavog procesa, kao i njegovu transparentnost i nepristranost, od popisa birača do konačnog prebrojavanja glasova. Svaka sumnja u integritet procesa može ozbiljno narušiti povjerenje javnosti i legitimnost izabranih dužnosnika. Nažalost, u mnogim postojećim izbornim sustavima previše je mogućnosti za manipulaciju glasovima i rezultatima. U Republici Hrvatskoj primjerice, proces glasovanja oslanja se na fizičke glasačke listiće koje glasači ispunjavaju i ubacuju u glasačke kutije na biračkim mjestima. Nakon zatvaranja birališta, glasački odbori ručno prebrajaju listiće za svako biračko mjesto, što otvara vrata ljudskoj pogrešci ili čak namjernoj prijevari. Ovo se pokušava spriječiti sastavljanjem izbornog povjerenstva od pripadnika suprotstavljenih političkih stranaka kako ni u kojem biračkom mjestu ne bi postojala suglasna većina koja bi mogla biti ponukana manipulirati rezultate izbora. Usprkos ovim mjerama, nije neuobičajeno čuti optužbe o manipulacijama izbora poput kupovanja glasova, pritiska na birače, višestrukog glasovanja, lažiranja glasova ili namjernih pogrešaka prilikom brojanja glasačkih listića. Čak i uz strog nadzor, sama logistika prikupljanja i zbrajanja glasova sa stotina biračkih mjesta predstavlja rizik. Tijekom transporta i obrade glasačkih listića postoji prilika da se oni zamijene, oštete ili izmijene. A jednom kad se utvrde rezultati, građani nemaju načina da provjere je li svaki glas stvarno pravilno prebrojan i samim time skeptičniji glasači nemaju razloga vjerovati u legitimnost samih izbora. Dokle god postoji i najmanja mogućnost za takve manipulacije, bit će teško u potpunosti vjerovati ishodima izbora.

Kao potencijalno rješenje nekih od ovih problema, nudi se automatizacija procesa glasovanja, kojom je moguće povećati transparentnost izbora i osnažiti povjerenje građana u legitimnost izbora. Digitalizacija procesa glasovanja mogla bi minimizirati ljudski faktor u samim izborima te time značajno otežati ili čak skroz ukloniti mogućnost manipulacije istima. Osim unaprjeđenja sigurnosnih aspekata izbora, uvođenje sustava za *online* glasovanje umanjilo bi potrebu za ljudskim radnicima na izborima. Nitko ne bi trebao

verificirati identifikacijske dokumente glasača, pružati im glasačke listiće, oprezno pratiti da glasači ne rade ništa sumnjivo niti prebrajati glasove pošto bi sve ove dužnosti obavljao sam sustav. I dalje bi bilo potrebno zaposliti nekoliko pojedinaca koji bi obavljali dužnosti sistemskih administratora i provjeravali da izbori napreduju uredno, no ukupan vremenski i novčani trošak bio bi značajno manji, posebice za veće izbore.

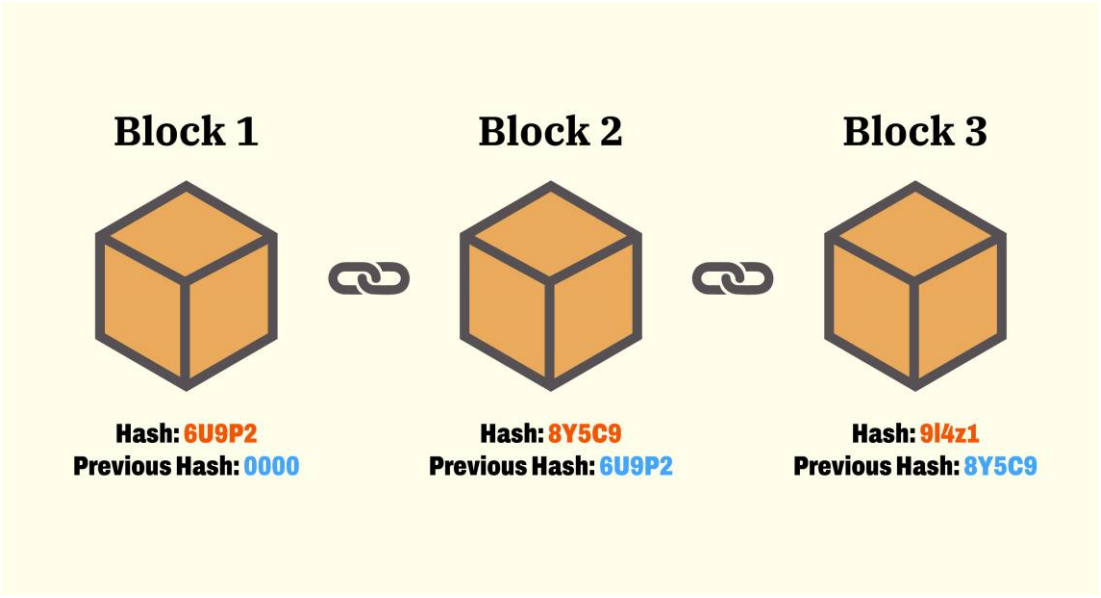
Trenutno vrlo atraktivna tehnologija lanca blokova (engl. *blockchain*) diči se svojim specifičnim sigurnosnim svojstvima, koja ju kvalificiraju i za uporabu u sigurnosno vrlo osjetljivim područjima kao što je digitalno bankarstvo. Ista ova sigurnosna svojstva čine tehnologiju lanca blokova vrlo prikladnom za uporabu u sustavu koji digitalizira proces glasovanja u svrhu povećanja transparentnosti i sigurnosti izbora. U nastavku ovog rada bit će razmotren prijedlog sustava za sigurno *online* glasovanje temeljenog na lancu blokova kao potencijalni idući korak u procesu stvaranja apsolutno transparentnih, sigurnih i poštenih izbora.

1. Teorijska pozadina

1.1. Lanac blokova

Kao što naslov ovog rada indicira, temelj na kojem je sagrađen ovaj sustav leži u tehnologiji lanca blokova (engl. *blockchain*). Prije nego što li se krene u analizu samoga sustava, ključno je da razumjeti što je lanac blokova, po čemu se razlikuje od alternativnih tehnologija i koje su prednosti njegovog korištenja.

Jednostavno rečeno, lanac blokova tip je baze podataka. Ono što razlikuje lanac blokova od standardnih relacijskih ili nerelacijskih baza podataka jest način organizacije podataka. Umjesto u tablice, u lancu blokova podaci su podijeljeni u jedinice zvane blokovima (engl. *blocks*) koje su organizirane u lanac. U lancu svaki blok ima svoje mjesto, a promjenom redoslijeda blokova u lancu, lanac se raspada. Ovaj je redoslijed osiguran kriptografijom. Svaki je blok povezan na blok prije sebe tako što sadrži kriptografski sažetak (engl. *cryptographic hash*) bloka ispred sebe u svojem sadržaju (Sl. 1.1). Kada bi se promijenio prethodni blok, automatski bi se promijenio i njegov kriptografski sažetak, a time informacija o kriptografskom sažetku prethodnog bloka pohranjena u trenutnom bloku ne bi više odgovarala stvarnom kriptografskom sažetku prethodnog bloka i lanac bi se raspao. O ovome će u više tehničkih detalja biti riječi u nastavku (1.1.2).



Sl. 1.1 Povezivanje blokova u lancu kriptografskim sažecima [1]

Dakle, lanac blokova je tip baze podataka koji svoje podatke organizira u blokove i povezuje kriptografskim obilježjima u lanac. Njegovo je drugo bitno svojstvo da je raspodijeljen, odnosno distribuiran. Lanac blokova je pohranjen u mreži čvorova gdje čvorovi sadrže lokalne kopije lanca blokova. U mreži mogu postojati različiti tipovi čvorova, s različitim privilegijama te svi ili samo neki mogu sadržavati kopiju lanca. Lokalne kopije lanca blokova moraju biti sinkronizirane nad svim čvorovima. Kako se novi blokovi dodaju na kraj lanca, čvorovi te promjene međusobno komuniciraju drugim čvorovima u mreži i tako ostvaruju sinkronizaciju po cijeloj mreži. Ovakva replikacija podataka osigurava povjerenje u istinitost istih. Ako se u mreži pojavi zlonamjeran čvor s pogrešnim informacijama u svojoj kopiji lanca blokova, lako će biti razotkriven zahvaljujući velikom broju točnih kopija lanca blokova pohranjenom u svakom od ostalih čvorova u mreži. Nadalje, zahvaljujući ovakvoj raspodijeljenosti, eliminira se potreba za verificiranom trećom stranom koja treba odobriti svaku akciju u sustavu. Čvorovi se sami sinkroniziraju i verificiraju čime se štede resursi i postiže veća efikasnost.

1.1.1. Sadržaj bloka

Svaki se blok u lancu blokova sastoji od sljedećih podataka:

1. Informacija
2. Vremenska oznaka
3. Jednokratna informacija (engl. *nonce*)
4. Sažetak prethodnog bloka
5. Sažetak bloka

Informacija je sam podatak koji blok nosi. Sve ostalo su dodatni podaci koji omogućuju pravilan rad lanca blokova. Vremenska oznaka specificira vrijeme kada je blok dodan u lanac i bitna je jer blokovi u lancu moraju biti poredani kronološki. Jednokratna informacija je nasumičan podatak generiran za svaki blok. Njegova će svrha biti objašnjena kasnije (1.1.3). Zatim, u bloku se nalazi kriptografski sažetak prethodnog bloka, kao što je već navedeno. U trenutku dodavanja bloka u lanac ovo je sažetak posljednjeg bloka u lancu. Konačno, u bloku se nalazi informacija o kriptografskom sažetku samog bloka.

1.1.2. Sažetak bloka

Idući je korak definirati što je sažetak bloka, kako se generira i zašto je bitan. U računarstvu postoji pojam funkcije sažetka (engl. *hash function*). Funkcija sažetka je jednosmjerna deterministička funkcija koji preslikava podatke proizvoljne veličine u niz bitova fiksne veličine. Djeluje kao otisak prsta za digitalne podatke ili dokumente. Ako se ulazna vrijednost modificira ili promijeni, vrijednost sažetka će također biti drukčija. Dovoljna je jedna točka, slovo ili razmak za promjenu cijele vrijednosti sažetka [2]. Funkcije sažetka mogu se koristiti kao metoda maskiranja (npr. u generiranju pseudonima), ali se također mogu koristiti za nesvjesno testiranje ekvivalentnosti dviju privatnih vrijednosti [3].

No, potrebno je razlikovati pojam funkcije sažetka u klasičnoj računalnoj znanosti od pojma kriptografske funkcije sažetka. Iako su u srži iste, kriptografska je funkcija sažetka nadograđena određenim sigurnosnim svojstvima koja ju čine prikladnom za uporabu u sigurnosnim algoritmima. Tri svojstva koja kriptografska funkcija sažetka mora zadovoljavati, a obična ne mora, su:

1. otpor predslike (ili jednosmjernost)

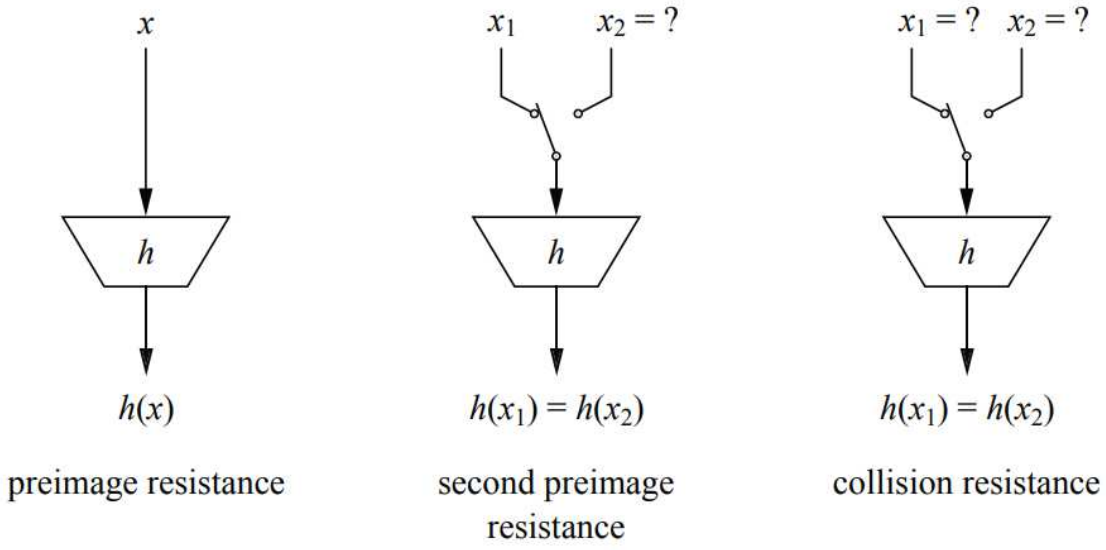
2. druga otpornost predslike (ili slaba otpornost na sudar)

3. otpornost na sudar (ili jaka otpornost na sudar) [4]

Kriptografska funkcija sažetka mora imati svojstvo otpora predslike, tj. mora biti jednosmjerna. Na temelju ulaznog podatka x mora biti računalno izvedivo pronaći izlazni podatak z, ali na temelju izlaznog podatka z mora biti računalno neizvedivo pronaći ulazni podatak x tako da vrijedi $z = h(x)$ [4].

Drugo bitno svojstvo koje kriptografske funkcije sažetka moraju zadovoljavati je druga otpornost predslike, što znači da je za dani ulazni podatak x1 i njegov sažetak h(x1) računalno neizvedivo pronaći drugi ulazni podatak x2 takav da vrijedi da je njegov sažetak jednak sažetku ulaznog podatka x1, odnosno da vrijedi $z1 = h(x1) = h(x2) = z2$ [4].

Konačno, kriptografsku funkciju sažetka nazivamo otpornom na jake sudare ako je računalno neizvedivo pronaći dva različita ulazna podatka $x1 \neq x2$ s jednakim vrijednostima sažetaka $h(x1) = h(x2)$. Ovo je svojstvo teže postići nego slabu otpornost na sudare budući da napadač ima dva stupnja slobode: obje se poruke mogu mijenjati kako bi se postigle iste vrijednosti sažetaka [4] (Sl. 1.2).



Sl. 1.2 Tri svojstva kriptografske funkcije sažetka [4]

Ako funkcija zadovoljava ova tri svojstva, kaže se da je funkcija sažetka kriptografski sigurna. Pri računanju sažetaka blokova koriste se isključivo kriptografski sigurne funkcije sažetka.

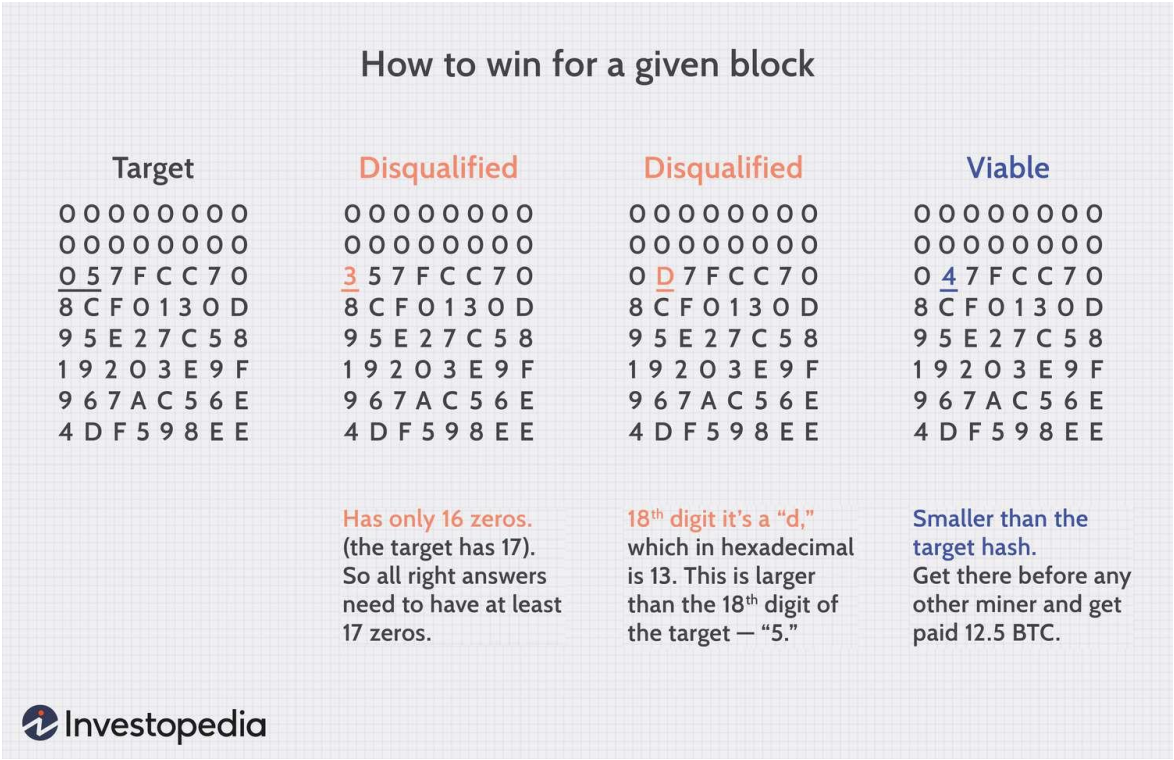
Ključno svojstvo lanca blokova koje proizlazi iz uporabe kriptografskih funkcija sažetka je njegova nepromjenjivost (engl. *immutability*). Nadalje, valja razmotriti slučaj gdje napadač želi promijeniti sadržaj nekog bloka u lancu. Zahvaljujući svojstvu kriptografskih funkcija da dva različita ulazna podatka, koliko god slična bila, imaju sasvim različiti sažetak, promjenom sadržaja bloka sigurno se mijenja i njegov sažetak. Zbog promjene sažetka bloka, sažetak pohranjen u idućem bloku više ne odgovara stvarnom sažetku bloka i lanac se raspada. Sada napadač, ako želi izbjeći raspad lanca, mora obnoviti sažetak bloka u sljedećem bloku, čime se naravno mijenja i sažetak tog bloka, te tako napadač lančano mora promijeniti svaki idući blok sve do kraja lanca. Ako napadač stvarno i promijeni cijeli lanac nakon željenog bloka, njegova kopija lanca blokova sada više uopće ne odgovara kopijama ostalih čvorova i svakako neće proći nezapaženo. Zato se kaže kako struktura lanca blokova ima svojstvo nepromjenjivosti. Jednom uneseni podaci više nisu promjenjivi, nemoguće ih je naknadno lažirati i svatko može provjeriti njihovu točnost.

1.1.3. Rudarenje blokova

Sada kada je funkcija sažetka definirana, valja razumjeti proces dodavanja novih blokova u lanac i uvesti pojam rudarenja blokova. Rudarenje blokova u lancu je proces uporabe

računala za verificiranje valjanosti blokova i kreiranje novih blokova u lancu. Rudarenje blokova u lancu obavljaju entiteti poznati pod imenom rudari (engl. *miners*). To mogu biti ljudi povezani sa svojih kućnih računala, poslužitelji ili automatizirani procesi, ovisno o sustavu. Često su rudari nagrađeni za obavljanje posla rudarenja, što im pruža motivaciju za nastavak rudarenja.

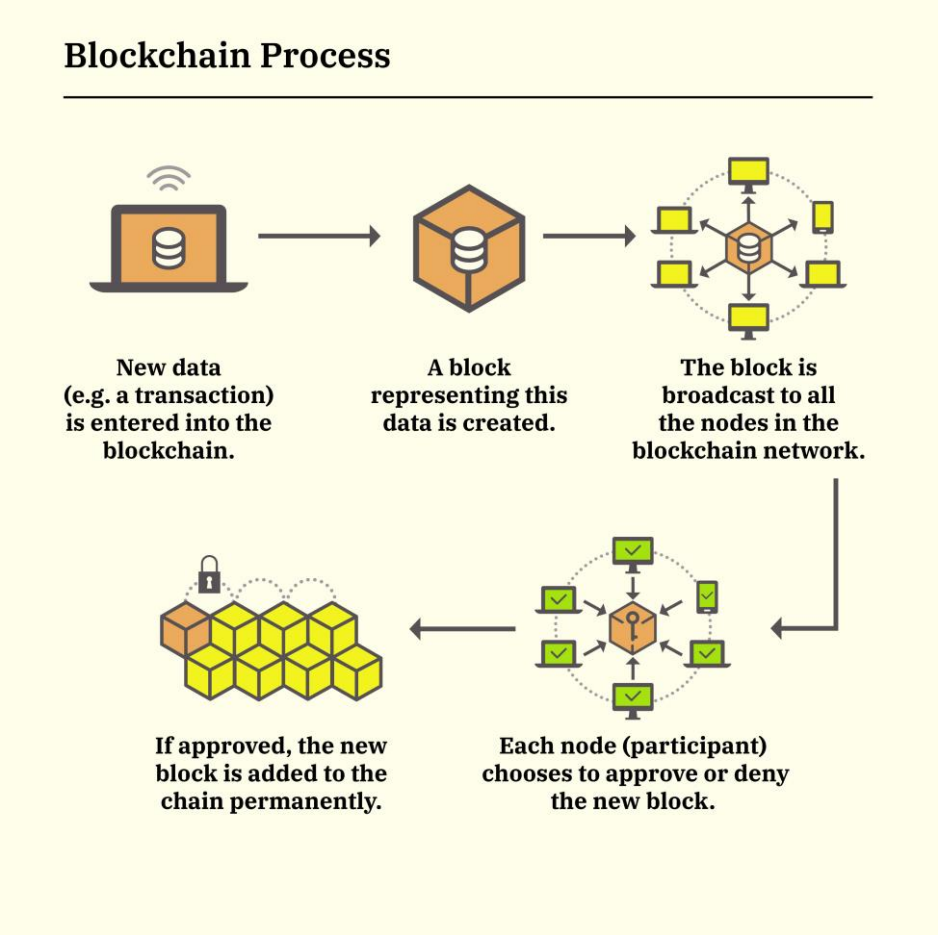
Proces rudarenja sastoji se od rješavanja kompleksnog matematičkog problema u svrhu pronalaska rješenja koje potvrđuje valjanost bloka i omogućava dodavanje bloka u lanac. Implementacija ovog matematičkog problema razlikuje se među različitim implementacijama lanca blokova, ali uvijek je povezana s pojmom kriptografskog sažetka. Problem koji rudari najčešće trebaju riješiti jest pronalazak sažetka bloka koji odgovara zahtjevima definiranim na razini lanca. Ovi zahtjevi su najčešće da sažetak mora počinjati predefiniranim brojem nula te da prvi podatak različit od nule mora biti manji od definirane konstante (Sl. 1.3).



Sl. 1.3 Rješavanje matematičkog problema u svrhu dodavanja bloka u lanac [5]

Postavlja se pitanje kako pronaći ovaj sažetak, pošto je ranije navedeno da je funkcija sažetka deterministička te da postoji samo jedan sažetak za svaki blok. Ako taj sažetak ne odgovara zahtjevima lanca znači li to da blok ne može biti dodan u lanac? Naravno da ne, tu u igru ulazi jednokratna informacija, ranije navedena kao jedan od podataka svakog bloka. Kao što je već rečeno, jednokratna informacija ne nosi nikakav koristan podatak,

ona je svaki put ispočetka nasumično generirana u svrhu pravilnog funkcioniranja lanca blokova. Rudari zapravo iterativno modificiraju jednokratnu informaciju bloka i izračunavaju novi sažetak bloka, ponavljajući ovaj proces sve dok dobiveni sažetak ne zadovolji specifične zahtjeve definirane pravilima lanca. Zahvaljujući svojstvima kriptografske funkcije sažetka i najmanja promjena jednokratne informacije dovoljna je da se sažetak sasvim promijeni. Dakle, kada nova informacija uđe u sustav i treba biti dodana u lanac, svaki od rudara u mreži započinje proces rudarenja nad tom informacijom. Kada jedan od rudara uspješno riješi matematički problem, tako što pronađe valjani sažetak koji zadovoljava zahtjeve lanca, informacija je u bloku dodana u lanac, a rudar je nagrađen. Ako se pak dogodi da dva rudara dođu do rješenja skoro istodobno, odnosno da nijedan od njih ne dobije informaciju o tuđem uspjehu prije nego sam dođe do rješenja, razmatra se vremenska oznaka svakog od blokova. Blok s manjom vremenskom oznakom nastao je prije (pa makar milisekundu prije) te se on dodaje u lanac, a rudar koji ga je kreirao dobiva nagradu. Važno je napomenuti da prije nego što je novi blok stvarno dodan u lanac, svaki od blokova u mreži verificira da je sažetak valjan te da zaista odgovara zahtjevima lanca (Sl. 1.4). Podatak drugog bloka mora se ponovo rudariti kako bi mogao biti dodan u lanac nakon upravo dodanog bloka.



Sl. 1.4 Proces dodavanja bloka u lanac [1]

Bitno je razumjeti poantu ovog pristupa, odnosno zašto svaki od rudara u mreži računa istu stvar, obrađuje isti blok, kada to može učiniti samo jedan rudar i pritom ne mora tražiti sažetak koji odgovara zahtjevima lanca, već naprosto može dodati blok u lanac s prvim sažetkom koji izračuna. Cijeli je ovaj proces uveden kako bi se u sustavu točni podaci razlikovali od netočnih, uvedenih od neke zlonamjerne stranke, primjerice od zlonamjernog rudara koji u lanac želi ubaciti netočne informacije radi vlastite koristi. Kada bi svaki blok u lancu obradio samo jedan rudar, ne bi bilo načina za dokazati da je podatak stvarno istinit. Blok bi bio valjan i uspješno dodan u lanac. Naime, u sustavu gdje svaki rudar obrađuje novi blok, ovaj je problem iskorijenjen. Jedan zlonamjeran čvor računao bi svoje netočne podatke, dok bi svi ostali dobronamjerni rudari računali točne. Naprosto zahvaljujući daleko većem broju dobronamjernih rudara od zlonamjernih, šanse da zlonamjerni rudar završi prije bilo kojeg od dobronamjernih su zanemarivo malene. Čim jedan od dobronamjernih rudara doda blok u lanac, zlonamjerni rudar mora započeti svoj posao ispočetka, jer se zadnji blok u lancu promijenio, a time se mora promijeniti i sažetak zadnjeg bloka u zlonamjernom bloku. Zahvaljujući razlici u broju dobronamjernih i

zlonamjernih rudara, zlonamjerni efektivno nikada neće stići obraditi svoj blok prije nego što to učini neki od dobronamjernih rudara. Ovaj bi se sistem raspao tek ako bi broj zlonamjernih rudara u mreži nadmašio ili se ozbiljno približio broju dobronamjernih rudara.

1.1.4. Dokaz rada

Ovakav se algoritam dolaska do konsenzusa (engl. *consensus mechanism*) naziva dokazom rada (engl. *proof of work*). Dokaz rada je prvi široko korišten mehanizam konsenzusa, a i danas je široko rasprostranjen. Doduše, dokaz rada dolazi s ozbiljnim nedostatkom, velikom potrošnom energije. Repetitivno mijenjanje jednokratne informacije i računanje novih sažetaka definirano je da bude vremenski zahtjevno u svrhu sigurnosti. No, uzme li se u obzir koliko energije jedan rudar troši na računanje jednog bloka, koliko rudara ima u mreži koji obavljaju isti taj posao i koliko se blokova svaki dan dodaje u lanac, jasno je kako su količine energije utrošene alarmantne. Zato danas postoje i alternativni algoritmi konsenzusa, kao što su dokaz udjela (engl. *proof of stake*), dokaz autoriteta (engl. *proof of authority*) i dokaz težine (engl. *proof of weight*). Svaki od njih ima svoje prednosti i nedostatke, a koji koristiti ovisi o vrsti aplikacije. Za ovu je aplikaciju korišten dokaz rada, zbog svoje jednostavnosti. Naravno, kako se sustav dalje bude razvijao, jedna od točaka potencijalnog napretka bit će zamjena algoritma dokaza rada nekim od suvremenijih i energetski efikasnijih mehanizama konsenzusa.

Zbog ovih sigurnosnih svojstava, tehnologija lanca blokova vrlo je prikladna za sustave kriptovaluta, što je danas i najčešća primjena ove tehnologije. No, zbog svojih specifičnih sigurnosnih svojstava, tehnologija lanca blokova ima potencijala biti od koristi i u raznim drugim sustavima, što ocrtava i ovaj sustav.

1.2. Mreža ravnopravnih čvorova

Baziranje sustava na tehnologiji lanca blokova implicira korištenje mreže čvorova koji međusobno komuniciraju i dijele informacije. Na ovaj se način ostvaruje replikacija informacija te osigurava povjerenje u točnost istih. Ova arhitektura naziva se mrežom ravnopravnih čvorova (engl. *peer to peer network*). Mreža ravnopravnih čvorova je decentralizirana mrežna arhitektura u kojoj pojedinačni uređaji, koji se nazivaju čvorovi (engl. *peers*), komuniciraju i surađuju izravno jedni s drugima kako bi dijelili resurse i

informacije [6]. Informacije koje čvorovi razmjenjuju u kontekstu sustava za sigurno *online* glasovanje su lokalne kopije lanca blokova. Postoje tri tipa mreža ravnopravnih čvorova:

1. Nestrukturirane mreže ravnopravnih čvorova
2. Strukturirane mreže ravnopravnih čvorova
3. Hibridne mreže ravnopravnih čvorova

1.2.1. Nestrukturirane mreže ravnopravnih čvorova

U nestrukturiranim mrežama čvorovi nemaju specifičan raspored, što rezultira nasumičnim komunikacijama između čvorova. Ovo čini nestrukturirane mreže ravnopravnih čvorova posebno pogodnim za aplikacije s visokom razinom aktivnosti, kao što su društvene platforme, gdje korisnici često ulaze ili izlaze iz mreže. Međutim, nestrukturirane mreže ravnopravnih čvorova imaju nedostatak. Za učinkovit rad zahtijevaju znatnu količinu CPU-a i memorijskih resursa. Hardver mora podržavati maksimalan broj mrežnih transakcija kako bi se osigurala besprijekorna komunikacija među svim čvorovima [6].

1.2.2. Strukturirane mreže ravnopravnih čvorova

Strukturirane mreže ravnopravnih čvorova razlikuju se od nestrukturiranih po tome što nude organiziranu interakciju među čvorovima. Ostvarene kroz dobro definiranu arhitekturu, te mreže omogućuju korisnicima učinkovitije lociranje i korištenje datoteka, eliminirajući nasumična pretraživanja. Funkcije sažetka često olakšavaju pretraživanje baze podataka u strukturiranim mrežama ravnopravnih čvorova. Iako su općenito učinkovitije, strukturirane mreže ravnopravnih čvorova pokazuju određeni stupanj centralizacije zbog svoje organizirane postavke. To može dovesti do većih troškova održavanja i postavljanja u usporedbi s nestrukturiranim mrežama ravnopravnih čvorova. Unatoč tome, strukturirane mreže pružaju veću stabilnost od svojih nestrukturiranih pandana [6].

1.2.3. Hibridne mreže ravnopravnih čvorova

Hibridne mreže ravnopravnih čvorova kombiniraju arhitekturu ravnopravnih čvorova s modelom klijent-poslužitelj. Ova hibridizacija uvodi središnji poslužitelj, što se pokazalo povoljnim za specifične mrežne scenarije. Hibridne mreže ravnopravnih čvorova nude

brojne prednosti u odnosu na strukturirane i nestrukturirane mreže, uključujući strateške pristupe, poboljšane performanse i druge prednosti [6].

U ovome je sustavu odabrana hibridna mreža ravnopravnih čvorova, primarno zbog jednostavnosti organizacije i implementacije sustava. U mreži se nalazi središnji koordinator čvorova, a njegova je svrha uvođenje novih čvorova u mrežu te koordiniranje informacija o čvorovima u mreži.

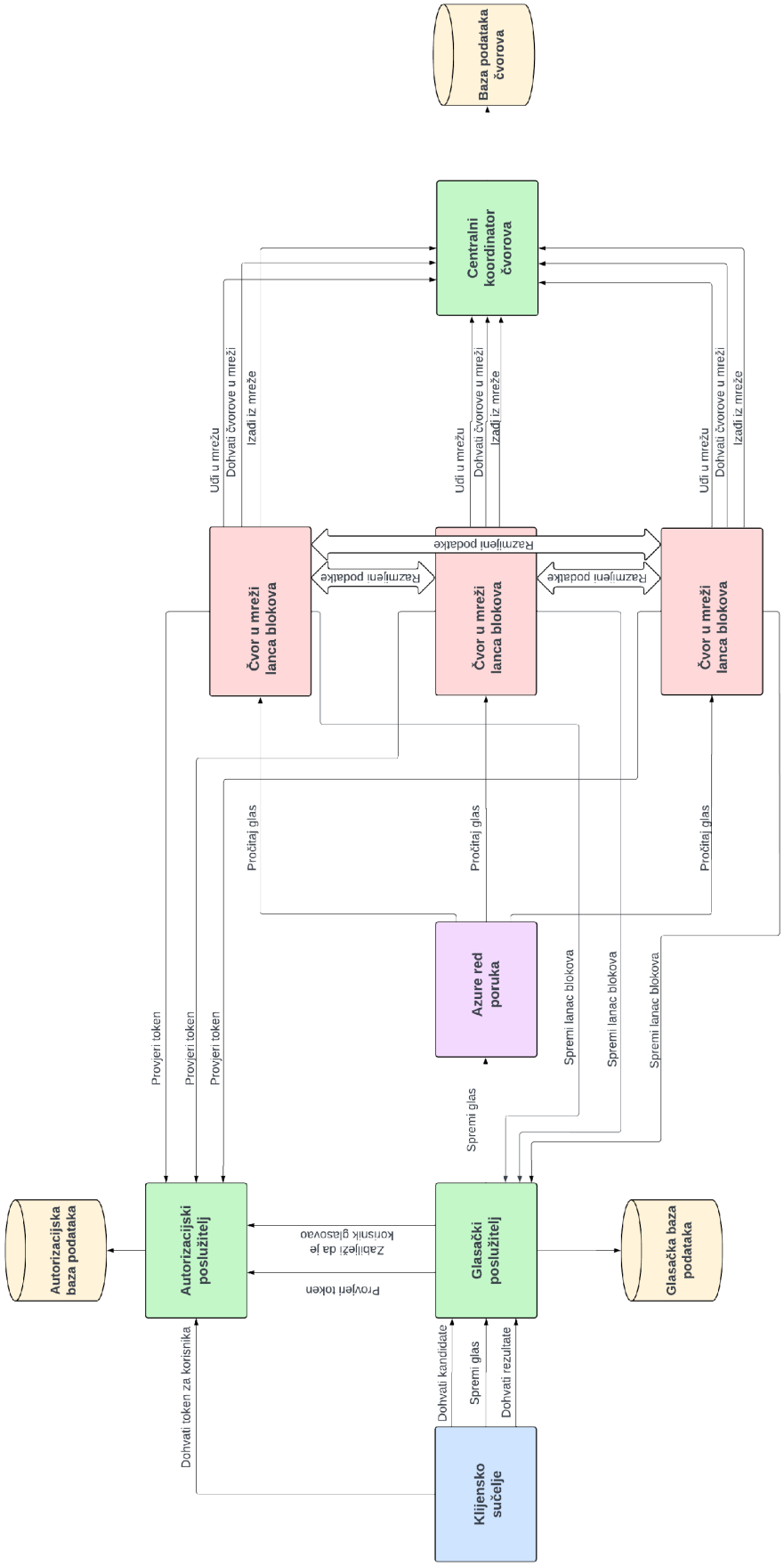
2. Arhitektura sustava

Arhitektura sustava za sigurno *online* glasovanje je mikroservisna. To znači da je cjelokupan sustav rastavljen na manje, logički nezavisne komponente ili usluge koje se jednostavnije razvijaju i održavaju. Svaki mikroservis u arhitekturi je odgovoran za specifičnu poslovnu funkcionalnost i komunicira s ostalim mikroservisima korištenjem mehanizama poput HTTPS/REST API-ja ili asinkronih mehanizama poput redova poruka. Ova decentralizirana arhitektura omogućuje neovisni razvoj, implementaciju i skaliranje pojedinih mikroservisa, poboljšavajući agilnost, fleksibilnost i otpornost na greške sustava.

Sustav za sigurno glasovanje sastoji se od pet mikroservisa:

1. Čvor u mreži lanca blokova
2. Centralni koordinator čvorova
3. Autorizacijski poslužitelj
4. Glasački poslužitelj
5. Klijentsko sučelje

Pregled arhitekture pet mikroservisa u sustavu dan je u dijagramu ispod (Sl. 2.1).



Sl. 2.1 Arhitektura sustavaa

2.1. Čvor u mreži lanca blokova

U samoj srži sustava za *online* glasovanje nalazi se mreža čvorova koji nose informacije o lancu blokova. Svaki od čvorova sadrži vlastitu kopiju lanca blokova, u koju kontinuirano dodaje nove blokove i razmjenjuje je s ostalim čvorovima u mreži.

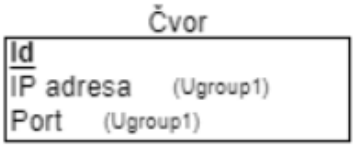
Čvorovi primaju informacije o novim glasovima preko reda poruka. Kada se u redu pojavi nova poruka, jedan od čvorova u mreži je pročita te započinje proces obrade poruke. Za početak, čvor u poruci pronalazi informaciju o autorizacijskom tokenu, koji dokazuje da glasač koji je dao glas trenutno obrađivane poruke uistinu ima pravo na glasovanje. Čvor šalje HTTPS upit na autorizacijski poslužitelj koji mu potvrđuje ili poriče valjanost tokena. Ako token iz bilo kojeg razloga nije valjan, poruka se odbacuje i čvor nastavlja čekati iduću. Ako je token pak valjan, čvor nastavlja s obradom poruke. Idući je korak provjeriti je li glas u poruci prisutan te je li valjan. Glas je valjan ako predstavlja jednog od kandidata tekućih izbora. Kada je čvor zaključio da su i token i glas valjani, započinje proces dodavanja bloka u lanac. Čvor kroz proces rudarenja verificira blok te ga dodaje u lanac blokova. Nakon dodavanja bloka u lanac, vrijeme je za obavijestiti ostale čvorove u mreži o novome bloku. Čvor šalje TCP poruku koja sadrži njegov cijeli lanac blokova svim ostalim čvorovima u mreži. Kada čvor primi poruku s lancem blokova drugoga čvora, prva stvar koju učini validacija je primljenog lanca blokova. Zaprimljeni lanac mora biti valjan sam po sebi, tj. sažetak svakog bloka mora odgovarati sažetku navedenom u idućem bloku te svaki blok lanca, osim prvoga, tzv. bloka geneze (engl. *genesis block*) mora sadržavati glas za jednog od kandidata. Također, zaprimljeni lanac blokova mora biti valjan u odnosu na lokalnu kopiju lanca blokova. Svaki blok zaprimljenog lanca mora biti identičan bloku na istoj poziciji u lokalnoj kopiji lanca blokova, s iznimkom posljednjeg bloka. Ako zaprimljena kopija lanca blokova sadrži jedan blok više od lokalne kopije te ako zaprimljena kopija prolazi svaku validaciju (samostalnu i u odnosu na lokalnu kopiju), lokalna kopija se prepisuje zaprimljenom kopijom lanca blokova. Može se dogoditi da su lokalna i zaprimljena kopija lanca iste dužine, što bi značilo da je lokalno čvor također obradio glas i dodao ga kao čvor u svoj lanac, netom prije primanja druge kopije lanca blokova. U tom se slučaju razmatraju vremenske oznake pohranjene u konačnom bloku obje kopije lanca blokova. Blok s ranijom vremenskom oznakom je prvi dodan u lanac blokova i ima prednost. Ako je to blok zaprimljene kopije lanca blokova, lokalna se kopija prepisuje zaprimljenom kopijom i čvor ispočetka započinje obradu glasa iz svojeg zadnjeg

bloka, sada prebrisanog zbog zaprimljene kopije. Ako je pak zadnji blok lokalne kopije dodan u lanac prije zadnjeg bloka zaprimljene kopije, zaprimljena se kopija naprosto odbacuje. U tom će slučaju drugi čvor sam shvatiti da mora odbaciti svoj zadnji blok kada primi poruku o zadnjem bloku lokalne kopije lanca blokova. Nadalje, ako zaprimljena kopija lanca blokova nije prošla bilo koju od ranije navedenih provjera, ona se naprosto odbacuje. U tom slučaju čvor zaključuje da se radi o napadaču koji pokušava narušiti valjanost lanca blokova ili samo o grešci.

Ovaj se proces ponavlja u svakom od čvorova u mreži tijekom cijelog trajanja izbora. Kada sustav prepozna da je trajanje izbora završeno, svaki od čvorova šalje svoju kopiju lanca blokova (sa svojim identificirajućim podacima) glasačkom poslužitelju na prebrajanje. Pritom svaki čvor potpisuje sadržaj svojega bloka svojim RSA privatnim ključem, čime dokazuje da ovaj HTTPS zahtjev s kopijom lanca blokova zaista dolazi od njega. Digitalni potpis čvor prilaže u zaglavlju HTTPS upita. Pri primitku HTTPS upita s kopijom lanca blokova, autorizacijski poslužitelj verificira digitalni potpis u skladu s listom javnih RSA ključeva svih čvorova u mreži te ga pohranjuje u svoju bazu u svrhu obrade i prikaza rezultata izbora.

2.2. Centralni koordinator čvorova

Kao što je ranije navedeno, sustav za sigurno *online* glasovanje obuhvaća mrežu međusobno povezanih, ravnopravnih čvorova. Ova je mreža hibridna, što znači da ju karakteriziraju neke značajke arhitekture klijent-poslužitelj. U sustavu se nalazi središnji koordinator čvorova koji pohranjuje podatke o svim čvorovima u mreži (Sl. 2.2) te ih na zahtjev dijeli s čvorovima u mreži. S njime čvor koji se želi pridružiti u mrežu prvim uspostavlja vezu te njega obavještava kada želi napustiti mrežu.



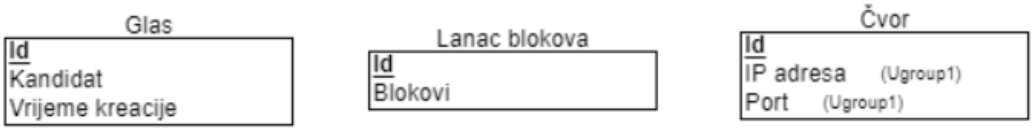
Sl. 2.2 Relacijski dijagram baze podataka centralnog koordinatora čvorova

2.3. Glasački poslužitelj

Glasački je poslužitelj poveznica između sustava za glasovanje i vanjskog svijeta. To je sučelje koje od korisnika prima informacije o glasovima, validira ih i obrađuje te šalje u mrežu čvorova. Glasački je poslužitelj klasičan API poslužitelj. On izlaže krajnju točku (engl. *exposes an endpoint*) koja sluša na HTTPS zahtjeve. Kada primi HTTPS zahtjev, provjerava ima li pošiljatelj uistinu pravo na glasovanje i nije li već glasovao, potom provjerava odgovara li zaista glas u zahtjevu jednom od kandidata na tekućim izborima. Zatim, glasački poslužitelj pohranjuje informaciju o glasu u bazu podataka (Sl. 2.3), u svrhu oporavka podataka u slučaju narušenja sustava. Konačno, glasački poslužitelj sprema informaciju o glasu u poruku, te ju s autorizacijskim podacima šalje na red poruka, gdje će je jedan od čvorova u mreži pročitati i obraditi.

Glasački poslužitelj također izvršava ulogu okupljanja svih kopija lanaca blokova od čvorova u mreži te prebrajanja glasova. Glasački poslužitelj u svojim postavkama ima popis javnih ključeva svih čvorova u mreži. Podrazumijeva se da svaki od čvorova u mreži ima zaseban par RSA ključeva, kojima potpisuje svoj lanac blokova i dokazuje svoj identitet. Prilikom primanja lanca blokova od čvora, glasački poslužitelj pronalazi javni ključ kojim je lanac blokova potpisan i verificira od kojeg je čvora kopija lanca blokova došla. Zatim, glasački poslužitelj u svojoj bazi provjerava je li ovaj čvor već poslao svoju kopiju i, ako nije, sprema informaciju o tome da je kopija od navedenog čvora zaprimljena. Ovo je bitno kako bi se spriječilo da zlonamjeran čvor u svrhu narušavanja rezultata pošalje svoju kopiju više puta. Ako je digitalni potpis u redu i ovo je zaista prvi put da dani čvor šalje svoju kopiju lanca blokova, lanac se pohranjuje u bazu podataka u svrhu prikaza na klijentskoj strani.

Konačno, kada vrijeme izbora i obrade rezultata završi, glasački poslužitelj otvara pristupnu točku za prikaz rezultata. Sve kopije lanca blokova u bazi podataka (Sl. 2.3) se tada uspoređuju. Sve kopije pristigle od dobronamjernih čvorova trebale bi biti međusobno identične. Ako se u mreži našao zlonamjeran čvor koji je poslao drukčiju verziju lanca blokova, glasački ju poslužitelj prepoznaje i ignorira. Na glasačkom je poslužitelju definirana granica, maksimalan broj netočnih kopija lanaca blokova koju poslužitelj tolerira prije nego je odlučeno da rezultati nisu valjani zbog prevelikog broja zlonamjernih čvorova u mreži. Ako je broj zlonamjernih kopija lanca blokova dovoljno malen, rezultati su valjani i pristupna točka glasačkog poslužitelja ih uredno vraća klijentu.



Sl. 2.3 Relacijski dijagram baze podataka glasačkog poslužitelja

2.4. Autorizacijski poslužitelj

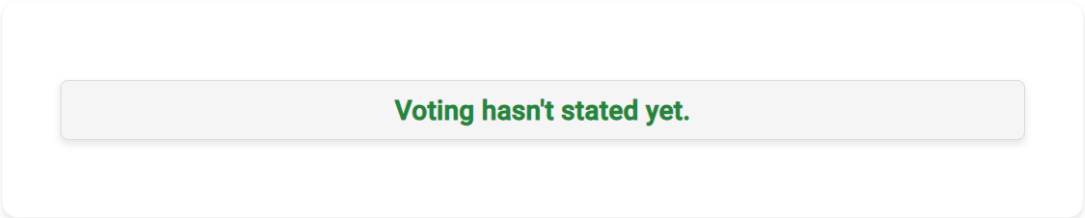
Ranije je spomenuto kako glasački poslužitelj vodi računa o autentifikaciji i autorizaciji glasača. On to radi uz pomoć autorizacijskog poslužitelja, izoliranog API poslužitelja čija je svrha autentifikacija i autorizacija korisnika. Razdvajanjem ova dva poslužitelja, postiže se razdvajanje briga (engl. *separation of concerns*), princip u softverskom inženjerstvu čiji je cilj rastavljanje složenih sustava na manje dijelove kojima se lakše upravlja. Cilj je organizirati komponente sustava tako da se svaki dio bavi jednim problemom ili kohezivnim aspektom funkcionalnosti, umjesto miješanja više problema zajedno. Aktivno primjenjivanje načela razdvajanja briga spriječit će utjecaj drugih struktura prilikom dodavanja ili poboljšanja novih modula ili usluga unutar aplikacije, te će poboljšati modularnost, mogućnost održavanja i skalabilnost sustava.

Ovdje je ključna činjenica da implementacija sustava za sigurno glasovanje, kakva je u trenutku pisanja ovog rada, predstavlja tzv. dokaz koncepta (engl. *proof of concept*), jednostavan primjer čija je svrha dokazati izvedivost projekta. Drugim riječima, ovaj se sustav trenutno ne koristi nigdje, već samo demonstrira kako bi se koristio u stvarnoj situaciji. Kada bi ovakav sustav bio integriran u stvarnoj situaciji, od tijela koje integrira sustav (primjerice vladajućeg tijela države ili organizacije koja organizira izbore) očekivalo bi se da implementira svoj autorizacijski poslužitelj, sa sučeljem nalik autorizacijskom poslužitelju trenutne verzije sustava, ali stvarnim podacima i metodama autentifikacije i autorizacije korisnika. Autorizacijski poslužitelj priložen u ovome radu naprosto oslikava rad takvog poslužitelja: sadrži informacije o setu od 50 umjetnih korisnika (Sl. 2.4) i njihovim podacima te izdaje i verificira jednokratne tokene za svakog od tih korisnika. Ovi se tokeni šalju između komponenata sustava i verificiraju po potrebi, u svrhu utvrđivanja privilegija glasača.

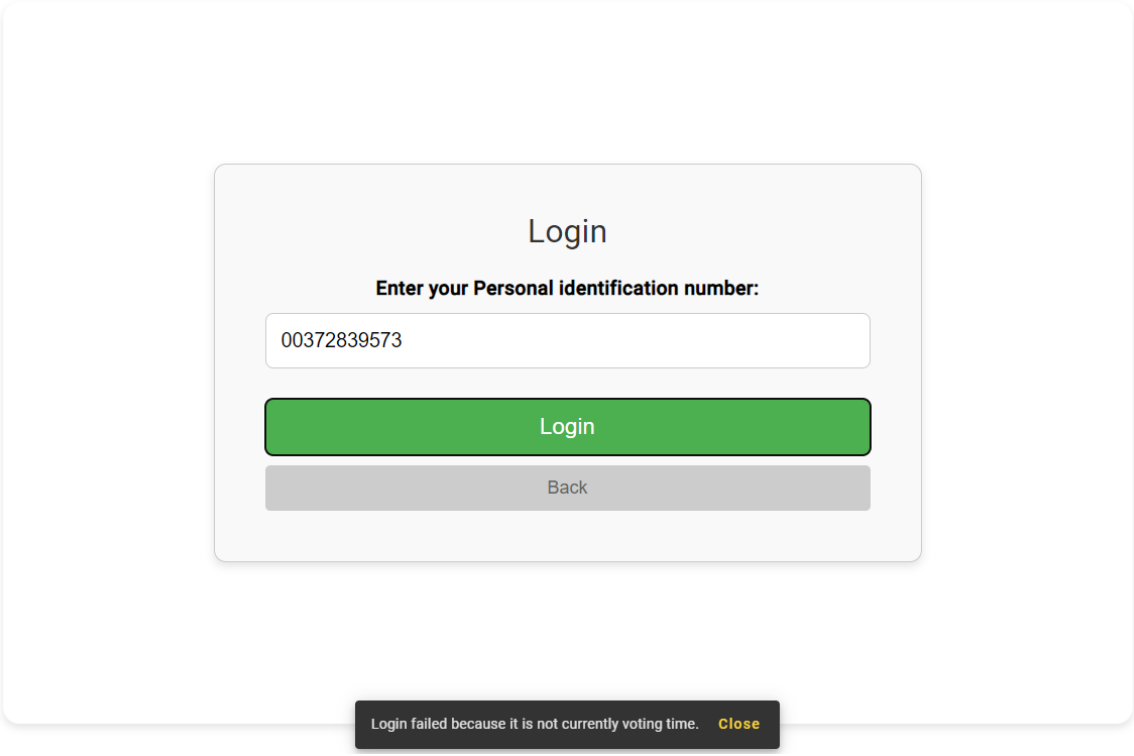
Sl. 2.4 Relacijski dijagram baze podataka autorizacijskog poslužitelja

2.5. Klijentsko sučelje

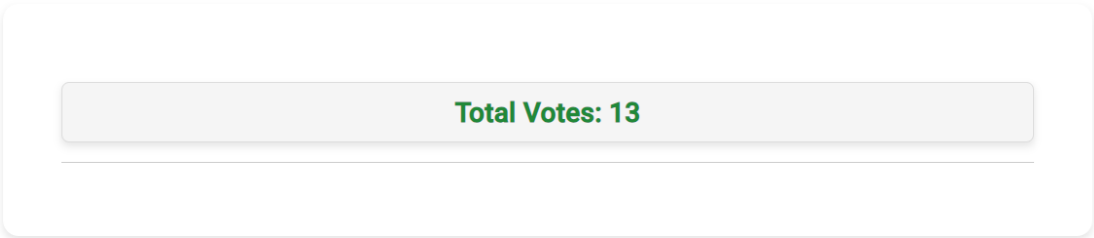
Posljednja komponenta sustava je klijentsko sučelje. Ovo je aplikacija koja se vrti u pretraživaču korisnika te komunicira s autorizacijskim i glasačkim poslužiteljem. Ona korisniku pruža grafičko sučelje za glasovanje te grafičko sučelje za pregled rezultata izbora. Komponente prikazane na klijentskom sučelju ovise o vremenu trajanja izbora u odnosu na trenutno vrijeme. Prije početka izbora, na glavnom je ekranu prikazana prikladna poruka (Sl. 2.5) i glasovanje je na ekranu za glasovanje onemogućeno (Sl. 2.6). Tijekom izbora, na glavnom ekranu prikazan je brojač trenutnih glasova, ali ne i rezultati, odnosno broj glasova po kandidatu (Sl. 2.7), a glasovanje je omogućeno. Kada korisnik želi glasovati, prvo se mora registrirati svojim osobnim identifikacijskim brojem. Ako uneseni osobni validacijski broj nije valjan ili ne odgovara stvarnom glasaču u sustavu, korisnik će o tome dobiti prikladnu poruku (Sl. 2.8) i neće moći pristupiti ekranu za glasovanje. Kada korisnik upiše valjan osobni identifikacijski broj (ako nije već glasovao) bit će proslijeđen na ekran za glasovanje gdje može priložiti i svoj glas (Sl. 2.9). Dok se rezultati izbora obrađuju, na glavnoj je strani i dalje vidljiv brojač glasova (Sl. 2.7), a glasovanje je onemogućeno (Sl. 2.6). Konačno, nakon obrade rezultata izbora, na glavnom je ekranu uz broj ukupnih glasova moguće vidjeti **Error! Reference source not found.** (Sl. 2.10). Glasovanje naravno ostaje onemogućeno (Sl. 2.6).



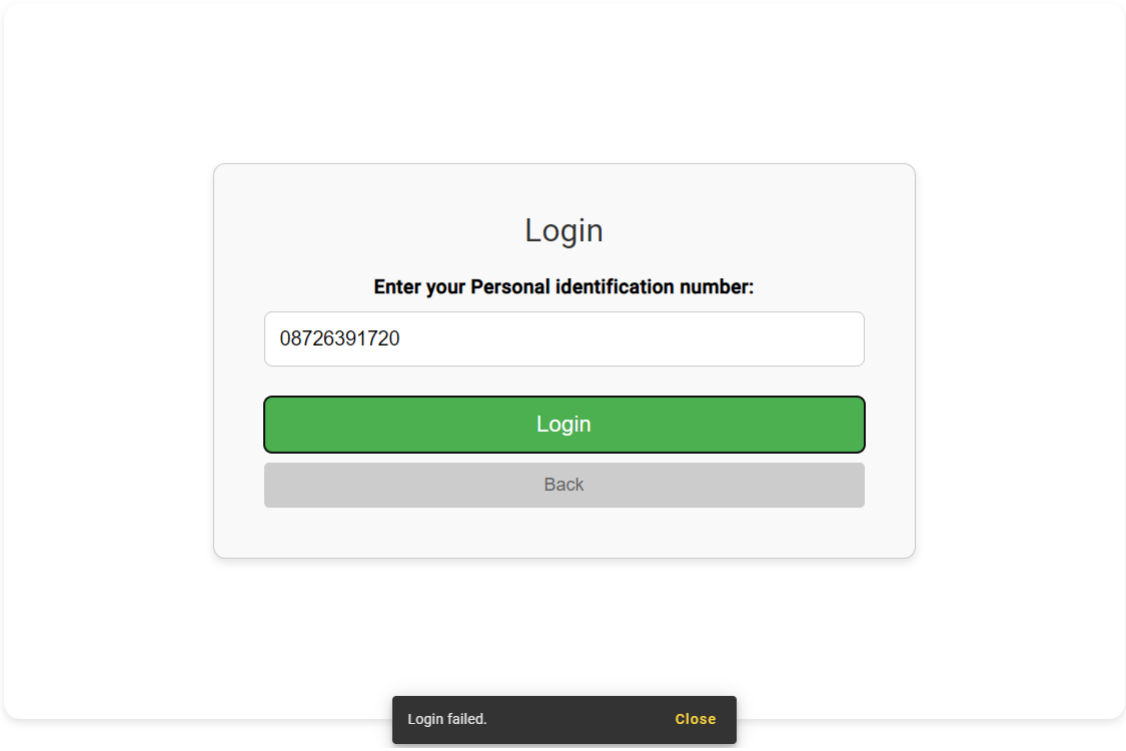
Sl. 2.5 Prikaz glavnog ekrana prije početka izbora



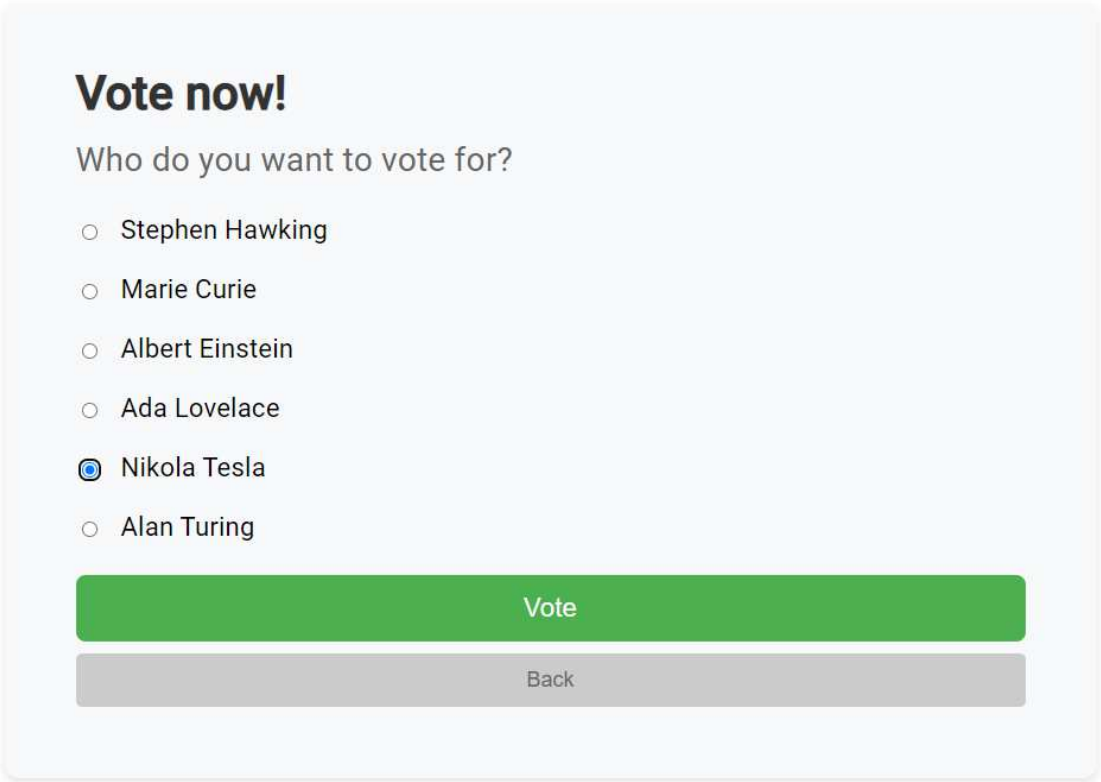
Sl. 2.6 Prikaz onemogućenog ekrana za glasovanje van trajanja izbora



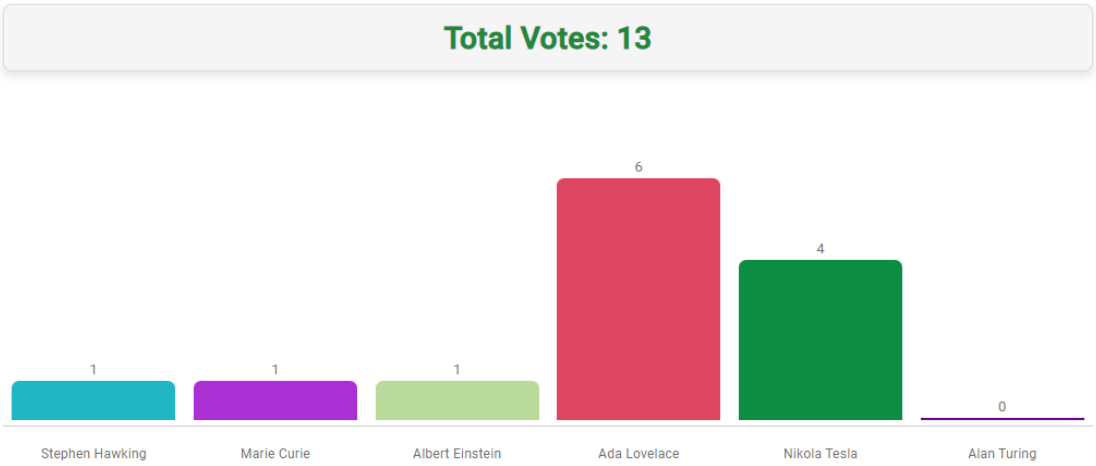
Sl. 2.7 Prikaz glavnog ekrana nakon početka izbora i prije kraja obrade rezultata izbora



Sl. 2.8 Prikaz ekrana za glasovanje s porukom o neuspješnoj registraciji



Sl. 2.9 Prikaz ekrana za glasovanje



Sl. 2.10 Prikaz glavnog ekrana nakon obrade rezultata glasovanja

3. Tehnologije

3.1. Java

Za implementaciju čvorova u mreži lanca blokova odabran je programski jezik Java. Java je široko korišten objektno orijentiran programski jezik visoke razine. Java je programski jezik opće namjene namijenjen da programerima omogući pisanje jednom, a pokretanje bilo gdje što znači da se prevedeni Java kod može izvoditi na svim platformama koje podržavaju Javu bez potrebe za ponovnim kompiliranjem (engl. *compile*). Neki od razloga zašto je Java odabrana kao platforma za čvor u mreži lanca blokova jesu njezina prenosivost (svatko tko želi postaviti čvor u mreži mora samo instalirati JVM i pokrenuti aplikaciju), bogata biblioteka, ugrađene sigurnosne mjere (kvalitetan sustav upravljanja memorijom, rukovanje iznimkama) te podrška višedretvenosti i objektna orijentiranost[1] koji Javu čine odličnim izborom za aplikacije poput ove.

Za automatizaciju i baratanje ovisnostima (engl. *dependency management*), projekt koristi popularan alat Maven. Aplikacija čvora u mreži lanca blokova organizirana je u klasičnu Maven strukturu, s testovima jedinice u AAA formatu priloženim u zasebnom direktoriju. Aplikacija ima dvije *Main.java* klase (ulazne točke u program): glavnu i običnu *main* klasu. Te dvije *main* klase su u suštini iste, a razlikuju se samo po tome što glavna sama kreira prvu kopiju lanca blokova, dok ju obična očekuje primiti od drugih čvorova u mreži. Zbog toga je, prilikom pokretanja sustava, bitno da je glavna *main* klasa pokrenuta prva i samo jedna, a da su nakon nje pokrenute ostale obične *main* klase.

Također valja spomenuti da se aplikacija čvora u mreži lanca blokova odlikuje s nekoliko oblikovnih obrazaca. Za baratanje samo jednom lokalnom kopijom lanca blokova i jednom kopijom liste čvorova u mreži koje su dostupne svim klasama u aplikaciji koristi se oblikovni obrazac jedinstveni objekt (engl. *singleton*). Jedinstveni objekt je kreacijski oblikovni obrazac koji omogućuje da klasa ima samo jednu instancu, dok pruža globalnu pristupnu točku ovoj instanci kroz cijelu aplikaciju[2]. Osim jedinstvenog objekta, u

[1] Oracleova uvodna stranica za programski jezik Java: https://www.java.com/en/

[2] Članak koji opisuje oblikovni obrazac jedinstveni objekt: https://refactoring.guru/design-patterns/singleton

aplikaciji čvora u mreži lanca blokova na više se mjesta koristi i oblikovni obrazac promatrač (engl. *observer*). Promatrač je oblikovni obrazac ponašanja koji omogućuje definiranje mehanizma pretplate za obavještavanje više objekata o svim događajima koji se dogode objektu koji promatraju[3]. Promatrač se u aplikaciji čvora u mreži lanca blokova koristi za obavljanje svih potrebnih akcija po primitku TCP poruke.

3.2. .NET

.NET je popularna platforma za razvoj softverskih aplikacija koju je razvio Microsoft. Uz .NET uglavnom je korišten programski jezik C# koji se može pohvaliti svim ranije navedenim prednostima programskog jezika Jave. Za razvoj svih web API poslužitelja u sustavu za sigurno *online* glasovanje (centralni koordinator čvorova, glasački poslužitelj i autorizacijski poslužitelj), korišten je ASP.NET Core radni okvir (engl. *framework*). ASP.NET radni okvir poznat je po svojim visokim performansama i skalabilnosti, bogatom sustavu biblioteka, ugrađenoj sigurnosti i odličnoj podršci za injekciju ovisnosti (engl. *dependency injection*) te jednostavnoj integraciji s Azureovim rješenjima za računarstvo u oblaku[4]. Ovi su razlozi učinili ASP.NET radni okvir odličnim izborom za brzu i sigurnu implementaciju web API-ja u sustavu za sigurno *online* glasovanje temeljenom na lancu blokova.

Također valja spomenuti da sva tri web API-ja sustava za sigurno *online* glasovanje prate REST standard što ih čini jako preglednima i smislenima kroz standardizirane HTTP metode, beskontekstualnost resursa i korištenje jasnih, intuitivnih, jedinstvenih URI-ja [10]. Svaki od web API-ja sustava organiziran je u jedno ASP.NET rješenje podijeljeno u više funkcijski organiziranih projekata (jedan za sve modele i ugovore, jedan za baratanje bazom podataka, jedan za servise, jedan za unit testove itd.).

3.3. Angular

Treća tehnologija korištena u sustavu za sigurno *online* glasovanje temeljenom na lancu blokova je Angular, popularan okvir prednje strane (engl. *frontend*) koji je razvio i

[3] Članak koji opisuje oblikovni obrazac promatrač: https://refactoring.guru/design-patterns/observer

[4] Microsoftova uvodna stranica za radni okvir ASP.NET: https://dotnet.microsoft.com/en-us/apps/aspnet

održava Google. Angular je odabran za implementaciju klijentskog sučelja zbog svoje modularnosti koja olakšava organizaciju koda, ponovnu upotrebu komponenti i bolje upravljanje složenošću projekta kako on raste, kompatibilnosti s programskim jezikom TypeScript koji donosi određena poboljšanja (statičko tipiziranja, lakše otkrivanje pogrešaka tijekom kompilacije i bolja čitljivost koda) u odnosu na svojeg prethodnika JavaScript te zbog svoje ugrađene podrške za injekciju ovisnosti[5].

Program klijentskog sučelja organiziran je u komponente gdje svaki ekran (glavni ekran, ekran za registraciju i ekran za glasovanje) odgovaraju po jednoj ili dvije komponente.

[5] Uvodna stranica za radni okvir Angular: https://angular.dev/

4. Implementacija sustava za sigurno *online* glasovanje

4.1. Lanac blokova i mreža čvorova

U ranijim poglavljima ovog rada opisani su standardna struktura i način funkcioniranja tehnologije lanca blokova. Ovu strukturu valja promatrati kao niz smjernica, ali samu implementaciju treba prilagoditi zahtjevima i potrebama sustava. Implementacije tehnologije lanca blokova razlikovat će se od sustava do sustava. U nastavku slijede specifičnosti ove implementacije lanca blokova.

Prva i vjerojatno najzamjetnija specifičnost ovog sustava jest da ne postoje rudari u klasičnome smislu. U sustavu za sigurno *online* glasovanje svaki čvor u mreži je ujedno i rudar te za njega nema nagrade kada doda novi blok u lanac. To je tako zato što u ovom sustavu rudari nisu ljudi već automatizirani procesi. U implementaciji lanca blokova za svrhe decentralizirane kriptovalute potrebno je ostvariti računalne resurse za rudarenje blokova s transakcijama te se u tu svrhu koriste računalni resursi korisnika sustava, koji su zauzvrat nagrađeni malenim količinama kriptovalute za svaki blok koji dodaju u sustav. U ovom slučaju nije poželjno od glasača tražiti da pruže svoje računalne resurse kako bi mogli glasovati. To ne bi bilo pošteno prema manje financijski uspješnim glasačima. Umjesto toga, računalne resurse u ovom slučaju osigurava organizator izbora, bila to nekakva organizacija ili vladajuće tijelo države. Zato, a i zbog činjenice da nikakva valuta ionako ne postoji u sustavu, nema smisla niti potrebe nagrađivati rudare.

Nadalje, specifičnost ove implementacije lanca blokova jest da je svaki blok obrađen od strane samo jednog čvora, a ne svakoga čvora u mreži. Ova je odluka donesena iz jednostavnog razloga da je ovaj način puno brži i energetski manje zahtjevan, a benefiti toga da više čvorova računa isti blok naprosto nisu potrebni. U ovom sustavu glasovi dolaze od provjerenog izvora, s enkriptiranog reda poruka, na koji poruke može stavljati isključivo glasački poslužitelj potaknut od strane autoriziranih glasača. Naravno, u svakom bi slučaju činjenica da više čvorova računa isti blok dodala dodatnu razinu sigurnosti koju bi u budućem razvoju ovog sustava svakako trebalo razmotriti, no ovo nije potrebno u

sklopu minimalnog vitalnog proizvoda (engl. *minimum viable product, MVP*), implementiranog u sklopu ovog rada.

Konačna specifičnost ovog sustava je činjenica da postoje dva tipa čvorova u mreži. Prvo postoji glavni čvor (engl. *master peer*), koji učitava listu kandidata iz konfiguracije i kreira inicijalni lanac. U mreži postoji samo jedan glavni čvor i on se za pravilan rad sustava mora pokrenuti prije svih ostalih čvorova u mreži. Ostali čvorovi u mreži su obični čvorovi, koji od glavnog čvora pri pokretanju dobivaju informaciju o inicijalnom lancu. Nakon pokretanja i razmjene informacija o inicijalnom lancu blokova, svi čvorovi funkcioniraju na isti način.

4.2. TCP komunikacija

Ranije navedena mreža ravnopravnih čvorova ostvarena je TCP komunikacijom. Razlozi zbog kojih je odabran TCP protokol su njegova pouzdanost, kontrola protoka, sposobnost očuvanja redoslijeda poruka i prepoznavanja grešaka. Činjenica da u TCP protokolu ne postoji opcija emitiranja poruke svim čvorovima (engl. *broadcast*), već svaka poruka mora biti usmjerena direktno jednom čvoru, jedan je od razloga zašto je odabrana hibridna mreža ravnopravnih čvorova. Koordinator čvorova u ovoj mreži ima svrhu informiranja čvorova o IP adresama i vratima (engl. *port*) ostalih čvorova u mreži.

U sustavu za sigurno *online* glasovanje definirana su tri tipa TCP poruka: „*spajanje*“, „*lanac blokova*“ i „*odspajanje*“. Tip TCP poruke čvorovi lako raspoznaju na temelju identificirajuće oznake na početku poruke. Za svaki od tri tipa TCP poruka, čvorovi imaju definirani odgovor.

Prilikom primanja TCP poruke za spajanje, čvor primatelj dodaje čvor pošiljatelj u svoju TCP mrežu (kolekciju podataka o čvorovima s kojima komunicira) i odgovara mu svojom lokalnom kopijom lanca blokova. Originalni čvor pošiljatelj uspoređuje svoju lokalnu kopiju lanca blokova s nadolazećom i po potrebi prepisuje vlastitu nadolazećom.

TCP poruku tipa „*lanac blokova*“ čvor pošiljatelj šalje povodom dodavanja novog bloka u svoju lokalnu kopiju lanca blokova. Po primitku TCP poruke ovog tipa, čvor primatelj po ranije objašnjenim uputama verificira zaprimljeni lanac blokova i po potrebi prepisuje lokalnu kopiju lanca blokova zaprimljenom. Na ovaj tip lanca blokova je definirani odgovor naprosto potvrda uspješne radnje.

Treći tip TCP poruke je „*odspajanje*“. Nalik prvom tipu, po primitku ove poruke, čvor primatelj uklanja čvor pošiljatelj iz svoje TCP mreže te nastavlja s daljnjim radom.

Budući da TCP komunikacija sama po sebi nije kriptirana, a za potrebe ovog sustava želimo enkripcijom osigurati tajnost sadržaja poruka, svaka TCP poruka poslana u sustavu za sigurno *online* glasovanje enkriptirana je algoritmom za simetričnu enkripciju AES-256, u načinu CBC, s PKCS5 podstavom (engl. *padding*). Ovaj je algoritam današnji standard za sigurnu simetričnu enkripciju i vrlo je široko korišten. Svi čvorovi u mreži dijele isti ključ za AES enkripciju te svaku TCP poruku pri slanju enkriptiraju, a pri primitku dekriptiraju. Ako poruka ne prođe dekripciju uspješno, automatski se odbacuje.

4.3. Red poruka

Važan dio sustava o kojem još nije mnogo rečeno jest komunikacija između glasačkog poslužitelja i mreže čvorova. Glasački poslužitelj od klijenta prima informacije o glasovima, validira ih, sprema u bazu podataka i obavlja ostale nužne operacije nad njima te ih tako priprema za spremanje u blokove na lancu. Idući korak koji glasački poslužitelj u tom trenutku treba obaviti je poslati informaciju o glasu jednom od čvorova u mreži. Tu se postavlja pitanje kako bi glasački poslužitelj trebao znati kojem čvoru poslati poruku. Bi li trebao imati listu svih čvorova u mreži te ju dinamički obnavljati kako čvorovi ulaze u i napuštaju mrežu? Bi li trebao imati i informaciju o tome koji je čvor trenutno nezaposlen, a koji obrađuje drugi glas? Ili samo nasumično određivati kojem će čvoru poslati informaciju o glasu? Svaki od ovih pristupa bi funkcionirao, više ili manje efikasno. Doduše, predviđeno rješenje ovog tipa problema su redovi poruka.

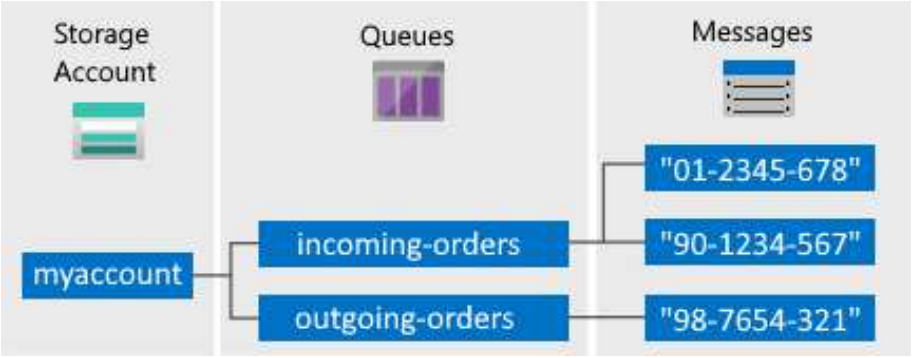
Red poruka (eng. *message queue*) je struktura podataka korištena za asinkronu komunikaciju između različitih komponenti računalnog sustava. Red poruka je privremena memorijska struktura koja skladišti poruke dok se ne obrade. Najčešće je to FIFO struktura (prvi ušao, prvi izašao, engl. *first-in, first-out*), što znači da se prva poslana struktura prva i obradi. Koriste se za razdvajanje procesa slanja i primanja poruka, omogućujući tako decentraliziranu i fleksibilnu distribuciju poslova. Primjena redova poruka je osobito korisna u sustavima gdje postoji potreba za usklađivanjem različitih brzina obrade između proizvođača i potrošača poruka. Proizvođač (npr. aplikacija ili servis) može nesmetano slati poruke u red, bez čekanja na obradu istih. S druge strane, jedan ili više potrošača

(radnika) mogu kontinuirano vaditi poruke iz reda i obrađivati ih u skladu sa svojim performansama.

Uvođenjem reda poruka u sustav za sigurno glasovanje razdajaju se logika slanja i primanja poruka s informacijama o glasovima. Glasački poslužitelj više ne mora znati ništa o čvorovima u sustavu: ne mora znati koji od čvorova će pročitati i obraditi poruku niti kada će to napraviti. Dapače, mreža tehnički može i biti prazna. U tom će slučaju poruke ostati pohranjene u redu dok se prvi čvor ne spoji u mrežu i pročita poruku s reda. Velika je prednost reda podataka kao strukture podataka mogućnost raspodjele posla. Uporabom reda poruka ne može se dogoditi da neki od čvorova dobiva previše glasova koje ne stiže obraditi, da ih mora sam negdje pohranjivati ili paralelizirati posao. Čvor naprosto odradi posao koji mu je poslan i, kada je gotov, čita novu poruku koja mu definira novi posao.

Konkretno, za ovaj je sustav odabran Azure Storage Queue kao pružatelj reda poruka. Azure Storage Queue usluga je reda poruka koju nudi Microsoft Azure, a dio je Azure Storage obitelji. Omogućuje pouzdanu i asinkronu komunikaciju između različitih komponenata sustava. Kao usluga u Azure oblaku (engl. *cloud service*), Storage Queue jamči visoku dostupnost i skalabilnost potrebnu za visoko propusne i opterećene aplikacije, kao i visoku dostupnost replikacijom poruka preko Azureovih podatkovnih centara[6] (Sl. 4.1). Na tržištu postoje i razne druge implementacije reda poruka s vlastitim prednostima i nedostacima, poput RabbitMQ, Apache Kafka ili Amazon SQS, no Azureovo je rješenje odabrano jer zadovoljava sve zahtjeve ovog sustava dok ostaje jednostavno za postavljanje i integraciju. Također, sva tri poslužitelja u sustavu implementirana su i .NET tehnologiji, koja je, kao još jedno Microsoftovo rješenje, dobro integrirana s Azureom i najčešće objavljena (engl. *deployed*) na Azure platformi u oblaku (engl. *cloud platform*).

[6] Članak koji opisuje zalihost Azure Storage Queuea: https://learn.microsoft.com/en-us/azure/storage/common/storage-redundancy



Sl. 4.1 Azure Storage Queue struktura[7]

Jedan od najbitnijih zahtjeva za red poruka sustava za sigurno *online* glasovanje i razloga zašto je Azureovo rješenje reda poruka odabrano jest sigurnost i enkripcija poslanih poruka. Podaci u Azure Storage usluzi enkriptirani su na strani klijenta pri slanju i dekriptirani pri primanju algoritmom AES-256, istim kojim je u ovom sustavu implementirana enkripcija TCP poruka. Zahvaljujući Azure-ovoj enkripciji, u ovom sustavu nije bilo potrebno napraviti ništa kako bi komunikacija između glasačkog poslužitelja i mreže čvorova bila sigurna, ona je takva sama po sebi.

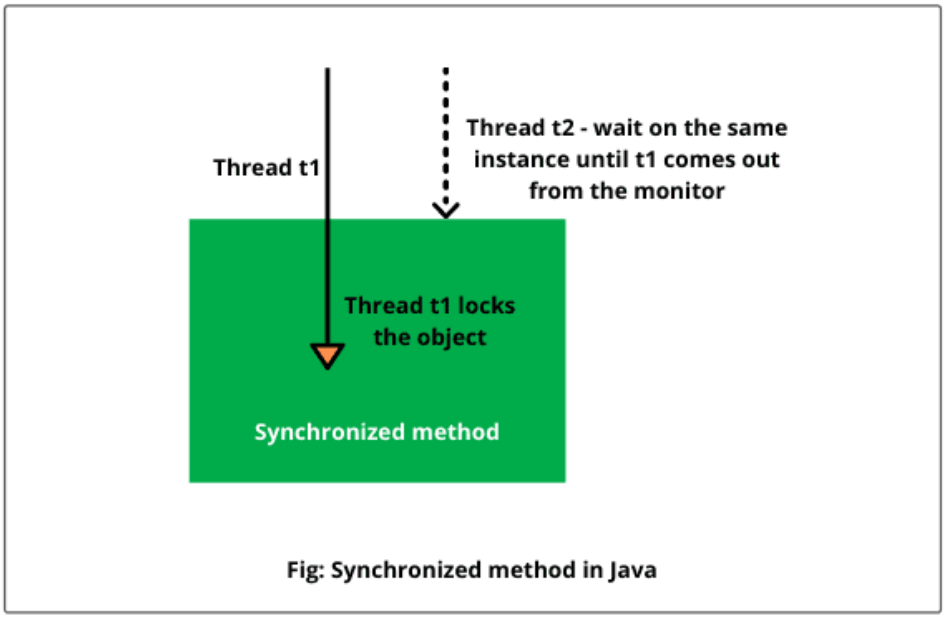
4.4. Problem konkurentnog pristupa i sinkronizacija

Jedan od ključnih izazova s kojim se programeri susreću u raspodijeljenim sustavima, gdje više procesa ili računala surađuju na zajedničkim zadacima, pa tako i u ovome, problem je konkurentnog pristupa. U sustavu za sigurno *online* glasovanje taj problem proizlazi iz činjenice da svaki od čvorova ima dva izvora koja mu mogu mijenjati lokalnu kopiju lanca blokova. Jedan od njih je red poruka, s kojeg čvor čita informaciju o novom glasu te ju obrađuje i dodaje u lanac. Drugi je TCP komunikacija, kojom čvorovi međusobno komuniciraju o novim blokovima u lancu. Većinu vremena rada sustava, svaki od čvorova u sustavu rudariti će jedan glas i istovremeno slušati na dolazeće TCP poruke od drugih čvorova. Dakle, u sustavu će postojati dvije dretve (engl. *thread*) koje izmjenjuju istu strukturu podataka, u ovom slučaju lokalnu kopiju lanca blokova. U ovakvim scenarijima lako može nastupiti problem konkurentnog pristupa. Problemi konkurentnog pristupa javljaju se kada dvije ili više dretvi paralelno pristupaju i pokušavaju promijeniti zajednički

[7] Članak koji istražuje veličiu poruka na Azureovom redu poruka: https://openxmldeveloper.org/unveiling-the-limits-exploring-message-size-in-azure-storage-queue/

resurs (npr. varijablu, podatkovnu strukturu ili datoteku) bez odgovarajuće kontrole pristupa i sinkronizacije. Ovo može dovesti do gubitka podataka ili neefikasnog baratanja računalnim resursima. Da bi se izbjegli ovakvi problemi, potrebni su mehanizmi sinkronizacije poput zaključavanja, monitora, semafora ili atomičkih operacija.

Čvorovi u mreži lanca blokova implementirani su u programskom jeziku Java, u kojem se kao rješenje problema konkurentnog pristupa nudi tzv. sinkronizirani blok (engl. *synchronized block*). Sinkronizirani blok omogućuje dretvi da stekne privremeno vlasništvo nad objektom i time osigurava međusobno isključiv pristup kritičnom odsječku koda među dretvama koje pristupaju tom objektu. Kada dretva uđe u sinkronizirani blok, ona pokušava steći zaključavanje nad objektom. Ako nijedna druga dretva nije zaključala taj objekt, trenutna dretva stječe zaključavanje i nastavlja s izvršavanjem koda unutar bloka. Ako je objekt već zaključan, trenutna dretva čeka dok se zaključavanje ne oslobodi (Sl. 4.2).



Sl. 4.2 Ilustracija sinkroniziranog bloka[8]

Dakle, za pravilan rad sustava za sigurno *online* glasovanje ključno je sinkronizirati tzv. kritične odsječke koda, odsječke koji mijenjaju lokalnu kopiju lanca blokova. Prvi je odsječak (Kôd 4.1), odsječak u kojem se novi blok s glasom pristiglim u redu poruka

[8] Članak koji objašnjava uporabu sinkroniziranog bloka u programskom jeziku Java: https://www.scientecheasy.com/2020/08/synchronized-method-in-java.html/

dodaje u lanac. Bitno je u kritični odsječak uključiti i liniju koda koja dohvaća trenutnu kopiju lanca, kako se ona ne bi promijenila prije završetka funkcije za rudarenje novog bloka.

```
synchronized (VotingBlockChainSingleton.lock) {
	VotingBlockChain blockChain =
VotingBlockChainSingleton.getInstance();
	Block block = new Block(vote, blockChain.getLastBlockHash());
	blockChain.mineBlock(block);
}
```

Kôd 4.1 Odsječak koda koji dodaje blok s glasom pristiglim iz reda poruka u lanac blokova

Drugi je odsječak (Kôd 4.2) koji treba zaključati odsječak koji uspoređuje nadolazeću kopiju lanca blokova s lokalnom i po potrebi prepisuje lokalnu kopiju nadolazećom. Ovaj je blok koda nešto veći i zato nije cijeli priložen u ovome radu, no ključan dio jest. Nakon zaključavanja kopije lanca blokova, uspoređuju se duljine lanaca, provjerava se podudarnost svih njihovih blokova osim zadnjih, a ako su obje kopije dobile jedan novi blok uspoređuju se vremenske oznake novih blokova i onda se konačno lokalna verzija prebrisuje u korist nadolazeće. Potom se poziva funkcija za ponovnu obradu glasa iz zadnjeg bloka lokalne verzije, kako bi bio spriječen gubitak te informacije

```
synchronized (VotingBlockChainSingleton.lock) {
	...
	VotingBlockChainSingleton.setInstance(incomingBlockchain);
	recreateLastBlock(currentLastBlock);
}
```

Kôd 4.2 Odsječak koda koji postavlja nadolazeću kopiju lanca blokova umjesto lokalne i ponovo pokreće rudarenje posljednjeg bloka lokalne kopije lanca blokova

4.5. Autentifikacija i autorizacija

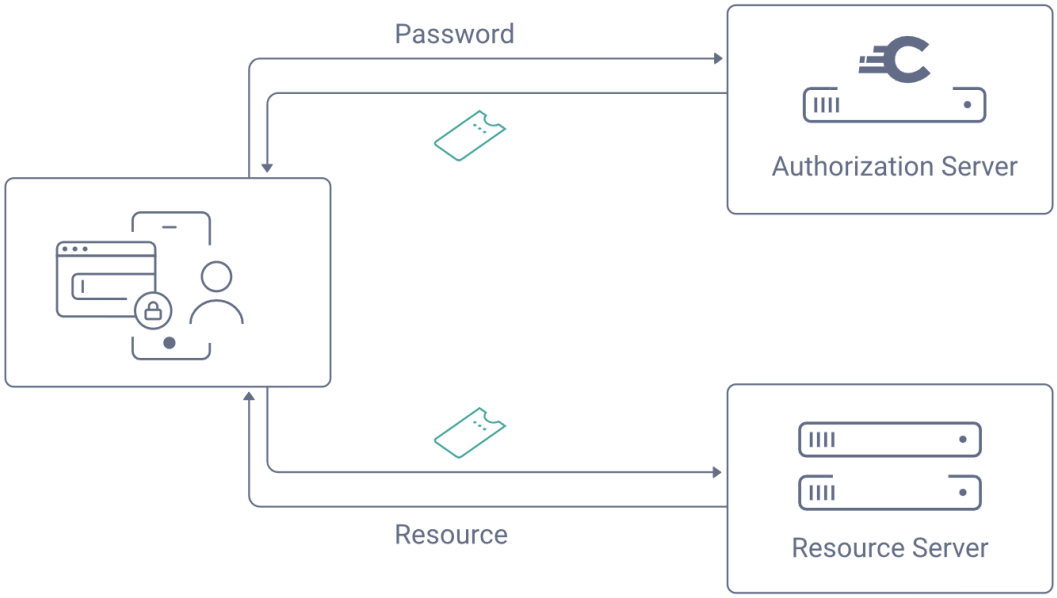
Bitan dio sustava za sigurno *online* glasovanje o kojem nije još mnogo rečeno su autentifikacija i autorizacija. Autentifikacija i autorizacija dva su ključna sigurnosna koncepta u računalnim sustavima i aplikacijama. Oni zajedno osiguravaju da samo ovlaštene osobe ili entiteti mogu pristupiti sustavu i obavljati određene radnje unutar njega.

Autentifikacija je proces provjere identiteta korisnika, uređaja ili sustava. Ona odgovara na pitanje „*Tko si ti?*“. Autentifikacija je postupak provjere korisnikovih vjerodajnica (korisničko ime i lozinka, biometrijski podaci ili drugi oblici identifikacije) s onima pohranjenima u sustavu kako bi se dokazalo da je korisnik zaista ono za što se predstavlja. Jednom kada se pružene vjerodajnice validiraju u odnosu na pouzdan izvor, korisnik ili entitet smatra se autentificiranim.

Autorizacija je proces definiranja i dodjeljivanja posebnih dozvola ili prava pristupa autentificiranim korisnicima, uređajima ili sustavima. Ona odgovara na pitanje „*Što si ovlašten raditi?*“. Nakon što je korisnik ili entitet autentificiran, proces autorizacije određuje resurse, podatke ili radnje kojima oni smiju pristupiti ili ih obavljati unutar sustava na temelju njihovih dodijeljenih uloga, ovlasti ili razina pristupa. Autorizacija može dodijeliti ili uskratiti dopuštenje za izvršavanje zadataka ili pristup područjima aplikacije.

Autentifikacija i autorizacija djeluju kao prva linija obrane u osiguravanju sustava i aplikacija, omogućujući samo ovlaštenim osobama ili entitetima pristup sustavu i obavljanje određenih radnji na temelju dodijeljenih dozvola. Ovaj slojeviti sigurnosni pristup pomaže u sprječavanju neovlaštenog pristupa, curenja podataka i potencijalnih zlonamjernih aktivnosti unutar sustava.

Moderne aplikacije obično ne odrađuju same svoju autentifikaciju i autorizaciju, već tu dužnost delegiraju vanjskom sustavu, svojevrsnom autorizacijskom poslužitelju. U takvim scenarijima, autorizacijski poslužitelj obavlja autentifikaciju korisnika te za njega kreira token, naizgled nasumičan podatak koji u sebi sadrži informacije o radnjama koje korisnik smije obavljati. Prilikom pristupa aplikaciji, korisnik šalje token koji je dobio od autorizacijskog poslužitelja. Na aplikaciji je samo da provjeri token i utvrdi smije li korisnik zaista pristupiti njenim resursima i operacijama. To pomaže aplikacijama da ne moraju upravljati osjetljivim korisničkim podacima i tako smanjuje rizik od povrede podataka. Osim toga, omogućuje različitim aplikacijama da dijele isti sustav provjere autentičnosti tako da korisnici ne moraju pamtiti više skupova vjerodajnica. **Error! Reference source not found.** (Sl. 4.3).



Sl. 4.3 Autentifikacija bazirana na tokenu[9]

[9] Članak koji opisuje autentifikaciju i autorizaciju te razlike među njima: https://curity.io/resources/learn/authentication-vs-authorization/

5. Sigurnosne značajke

Kao što i naslov ovog rada sugerira, jedna od najbitnijih odlika ovog rada te ujedno i motivacija iza njegovog nastanka jest potreba za sigurnim, sasvim digitaliziranim sustavom za glasovanje. Za pravilan rad sustava od ključne je važnosti da su svi dijelovi sustava sigurni i da ne postoje ranjivosti koje mogu biti iskorištene za lažiranje izbora.

Kroz rad više je puta istaknuto kako sustav za glasovanje mora biti siguran, ali još nije objašnjeno točno što to znači niti kakve odlike siguran sustav mora imati. Da bi računalni sustav bio siguran, sljedeći temeljni sigurnosni zahtjevi moraju biti zadovoljeni:

- povjerljivost (engl. *confidentiality*), tajnost (engl. *secrecy*)
- cjelovitost, integritet (engl. *integrity*)
- raspoloživost (engl. *availability*)
- autentičnost (engl. *authenticity*)
- neporecivost (engl. *non-repudiation*) [7]

Prva tri navedena sigurnosna zahtjeva smatraju se temeljnim sigurnosnim zahtjevima i preduvjetima za sigurnost sustava. Druga dva zahtjeva nekad se pridodaju u skup temeljnih sigurnosnih zahtjeva, ali nisu prihvaćeni kao prva tri [7]. Za svrhe ovog rada bit će razmotreno svih pet sigurnosnih zahtjeva.

Prvi zahtjev, povjerljivost ili tajnost, znači da određene informacije moraju biti dostupne samo ovlaštenim identitetima [7]. U sustavu za *online* glasovanje zapravo ne postoji puno podataka koji su tajni, odnosno povjerljivi. Svi podaci u lancu blokova namijenjeni su da budu dostupni svima i samim time nisu tajni. Doduše, komponenta povjerljivosti očitava se u vremenskim oznakama sustava. Naime, tijekom samog glasovanja (i prije dovršetka obrade rezultata) privremeni rezultati glasovanja su tajni. Ovaj je zahtjev sustava definiran kako privremeni rezultati ne bi utjecali na birače, baš kao na klasičnim izborima. Tajnost privremenih rezultata osigurana je autentifikacijom i autorizacijom korisnika koji pokušaju pristupiti bilo kojoj od pristupnih točaka glasačkog poslužitelja. Kako je ranije rečeno, ova su autentifikacija i autorizacija ostvarene s pomoću tokena.

Idući je temeljni sigurnosni zahtjev, a potencijalno i najbitniji sigurnosti zahtjev u ovom sustavu, cjelovitost, tj. integritet. Integritet je jamstvo da su informacije poslane, primljene ili pohranjene u izvornom i nepromijenjenom obliku [7]. Konkretno, u sustavu za sigurno *online* glasovanje, očuvanje je integriteta glasova od naročite važnosti. Integritet glasova je jamac sustava da su rezultati glasovanja netaknuti i nepromijenjeni. Integritet je jedno od bitnih svojstava tehnologije lanca blokova same po sebi. Zbog ranije objašnjene strukture lanca blokova, u raspodijeljenoj mreži nemoguće je naknadno promijeniti vrijednosti blokova u lancu. Jednom kada je informacija pohranjena u lancu, ona je nepromjenjiva te njena cjelovitost ostaje garantirana.

Idući je sigurnosni zahtjev raspoloživost i njegovu je zadovoljenost u ovom sustavu nešto teže garantirati. Sustav je raspoloživ ako su njegove informacije raspoložive, a usluge u operativnom stanju, usprkos mogućim neočekivanim i nepredvidljivim događajima [7]. Replikacija lanca blokova po svim čvorovima u mreži definitivno ide u korist očuvanju raspoloživosti sustava, no u principu o raspoloživosti nema previše smisla pričati u fazi lokalnog razvoja. Pitanje raspoloživosti postaje puno bitnije u trenutku objave (engl. *deployment*) sustava. U daljnjem razvoju sustava za sigurno *online* glasovanje planirana je objava na Azure usluži računarstva u oblaku (engl. *cloud computing service*). Između ostaloga, Azure svojim korisnicima pruža visoku raspoloživost konceptom pod imenom zona raspoloživosti (engl. *availability zones*). Zone raspoloživosti su odvojene grupe podatkovnih centara unutar Azure regije (Sl. 5.1), koje su dovoljno blizu da imaju veze niske latencije s drugim zonama dostupnosti. Povezane su mrežom visokih performansi s povratnom latencijom manjom od 2 ms. Međutim, zone dostupnosti dovoljno su udaljene jedna od druge kako bi se smanjila vjerojatnost da će više od jedne biti pogođene lokalnim prekidima ili vremenskim uvjetima. Zone dostupnosti imaju neovisnu infrastrukturu napajanja, hlađenja i umrežavanja. Osmišljeni su tako da ako jedna zona doživi prekid rada, ostale zone podržavaju regionalne usluge, kapacitet i visoku dostupnost. [10]

[10] Microsoftov članak koji opisuje Azureove zone raspoloživosti: https://learn.microsoft.com/en-us/azure/reliability/availability-zones-overview?tabs=azure-cli

Sl. 5.1 Ilustracija Azure regija i zona dostupnosti [8]

Četvrti sigurnosni zahtjev sustava za *online* glasovanje jest autentičnost. Autentičnost implicira potvrdu identiteta korisnika, autentifikaciju sudionika komunikacije te provjeru izvora podataka [7]. Kako je ranije navedeno, ova su svojstva zadovoljena uporabom autorizacijskog poslužitelja koji autentificiranim korisnicima izdaje tokene za pristup resursima.

Konačni sigurnosni zahtjev je neporecivost, koji je također od velike važnosti u sustavu za sigurno *online* glasovanje. Neporecivost znači da sudionici nikako ne mogu poreći akciju u kojoj su sudjelovali, tj. ne mogu naknadno otkazati akciju u koju su krenuli. Ovaj je zahtjev poput integriteta osiguran uporabom lanca blokova, jednom kad je glas zapakiran u blok u lancu, glasač ga više nikako ne može poreći, to je neporeciva radnja.

Sustav koji zadovoljava ovih 5 sigurnosnih zahtjeva smatra se sigurnim. Bitno je naglasiti da je sustav siguran samo ako su sigurnosni zahtjevi zaista zadovoljeni, duž cijelog sustava. Na posljetku, sustav je siguran koliko je siguran njegov najmanje siguran dio.

U svrhu utvrđivanja da su podaci zaista zaštićeni u svakom trenutku koji provedu u sustavu, valja proći još jedno kroz tok podataka u sustavu za sigurno *online* glasovanje. Podaci ulaze u sustav preko klijentskog sučelja. Klijentsko sučelje i glasački poslužitelj komuniciraju i razmjenjuju tokene za svakog korisnika s autorizacijskim poslužiteljem u svrhu utvrđivanja identiteta glasača i njihovog prava na glasovanje te tako zadovoljavaju svojstvo autentičnosti u tom dijelu sustava. Komunikacija između klijentskog sučelja i web poslužitelja te među web poslužiteljima bazirana je na protokolu HTTPS. HTTPS protokol je sigurna verzija protokola HTTP, koji je primarni protokol koji se koristi za slanje podataka između web-preglednika i web-mjesta. HTTPS protokol enkriptira poslane podatke čime zadovoljava svih pet sigurnosnih zahtjeva između dva servisa koji

komuniciraju HTTPS protokolom. Zatim, glasački poslužitelj komunicira s čvorovima u lancu preko Azure reda poruka. Kao što je ranije navedeno, poruke u redu Azure automatski enkriptira, osiguravajući tako sve sigurnosne zahtjeve i u tom dijelu sustava. Konačno, čvorovi međusobno komuniciraju TCP porukama. TCP protokol sam po sebi ne enkriptira sadržaj svojih poruka, no kao što je ranije rečeno, ova je enkripcija u sustavu za sigurno *online* glasovanje ostvarena ručno, čime je i u zadnjem koraku prijenosa podataka u sustavu zadovoljeno svih pet sigurnosnih zahtjeva.

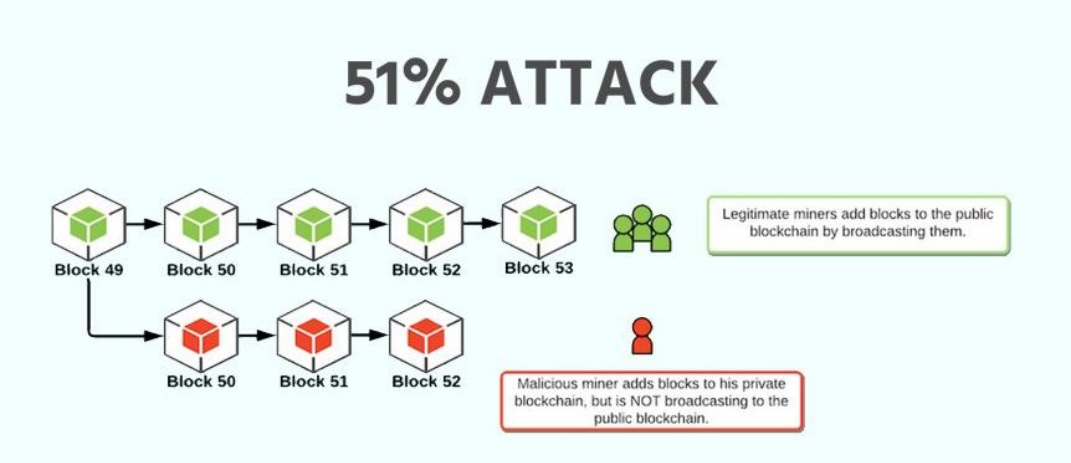
5.1. Potencijalni napadi na sustav

Usprkos zadovoljenosti svih sigurnosnih zahtjeva, i u ovom sustavu postoje ranjivosti. Jedine (dosad poznate) ranjivosti sustava za sigurno *online* glasovanje su ujedno i vrlo česte ranjivosti svih sustava baziranih na lancu blokova, ranjivosti koje proizlaze iz tehnologije lanca blokova kao takve. Ove ranjivosti najčešće imaju veze s pojavom zloćudnih čvorova u mreži lanca blokova.

Sustav za sigurno *online* glasovanje pri predaji rezultata glasovanja na glasački poslužitelj filtrira rezultate po tome dolaze li od predefiniranih validnih čvorova ili ne, kako bi se spriječio napad gdje napadači šalju lažne (ali validne) lance blokova na poslužitelj. No, prije ove predaje rezultata, mreži čvorova se može priključiti bilo tko, pošto priključivanje mreži blokova ne zahtijeva nikakvu provjeru identiteta. To također znači da napadač može modificirati kod svojega čvora da radi nešto drugo te se priključiti mreži i narušiti tok razmjene kopija lanaca.

Jedan od najopasnijih napada na sustav baziran na lancu blokova, te također napad koji bi mogao narušiti i sustav za sigurno *online* glasovanje temeljen na lancu blokova jest 51% napad. 51% napad je napad na mrežu lanca blokova gdje jedan entitet stječe kontrolu nad više od polovice (51%) računalne snage. Ova nerazmjerna kontrola omogućuje im provedbu značajnih promjena, što je u suprotnosti s načelom decentralizacije temeljnim za tehnologiju lanca blokova [9]. Prvi korak ovog napada uključuje napadača koji akumulira više od polovice (51%) računalne snage mreže, odnosno ima kontrolu nad više od 51% čvorova u mreži ako govorimo o čvorovima jednake računalne snage. To se može postići stjecanjem znatnih hardverskih resursa ili uvjeravanjem velikog broja rudara da se pridruže grupi pod kontrolom napadača [9]. Napadač, koji sada upravlja većinom mrežne moći raspršivanja, učinkovito odvaja svoju grupu od glavne mreže dok i dalje održava internu

komunikaciju. Unatoč ovom odvajanju, hakerska skupina nastavlja s operacijama rudarenja, ali se suzdržava od dijeljenja svog napretka s primarnom mrežom ili primanja ažuriranja od nje. Posljedično, dvije paralelne verzije lanca blokova počinju se razvijati neovisno (Sl. 5.2) [9]. Zbog svoje superiorne računalne moći koja uzrokuje brže računanje sažetaka novih blokova, napadačeva skupina može dodavati blokove svojoj verziji lanca blokova brže od ostatka mreže [9]. Nakon što se hakerska skupina ponovno pridruži mreži, dvije konkurentske verzije lanca blokova šire se cijelom mrežom. Prema pravilima konsenzusnog protokola, čvorovi zadržavaju najdulji lanac blokova, a kraći se odbacuje [9]. To znači da svi blokovi koje je glavna mreža dodala tijekom razdoblja odvajanja odbacuju, dok se blokovi napravljeni od zloćudnog dijela mreže zadržavaju. Važno je napomenuti da je napad od 51% među najznačajnijim sigurnosnim prijetnjama lancima blokova, posebno onima koji koriste konsenzusne algoritme dokaza rada (engl. *proof of work*) i dokaza udjela (engl. *proof of stake*) [9]. Postoji nekoliko mogućih rješenja za prevenciju 51% napada u sustavima baziranim na tehnologiji lanca blokova. Preporučeno je rješenje promjena mehanizma konsenzusa između čvorova na neki gdje konačna prevlast ne ide grupi s najvećom računalnom moći. Također, uvođenje revizije (engl. *audit*) lanca blokova omogućilo bi administratorima sustava rano prepoznavanje i sprječavanje ovakvih napada. Ove točke svakako bi trebalo razmotriti u daljnjem razvoju sustava za sigurno *online* glasovanje temeljenog na tehnologiji lanca blokova.



Sl. 5.2 Ilustracija 51% napada na lanac blokova[11]

[11] Članak koji objašnjava česte napade na sustave bazirane na lancu blokova i ranjivosti tehnologije lanca blokova: https://kingslanduniversity.com/blockchain-attack-vectors-vulnerabilities

6. Potencijalni dodaci

Sustav za sigurno glasovanje temeljen na lancu blokova već u svojoj trenutnoj, nedovršenoj fazi zadovoljava pohvalan set sigurnosnih zahtjeva te nudi sigurnu i transparentnu alternativu trenutnom sistemu glasovanja. Doduše, ovaj je sustav još daleko od dovršenosti i stvarne mogućnosti uporabe na pravim izborima. Jasno je da su izbori bilo koje države od prevelike važnosti da se proces njihove provedbe stavi u ruke sustava napravljenog od strane jednog studenta u manje od godinu dana. Zato će u ovom poglavlju biti definirane iduće točke razvoja ovog sustava, koje bi ga jednoga dana zaista mogle dovesti do uporabe na stvarnim izborima.

Počnimo s točkama daljnjeg razvoja sustava vezanim za kriptografiju. Svaki od čvorova u mreži lanca blokova ima vlastiti par RSA ključeva, a čvorovi svoje privatne RSA ključeve pohranjuju u .txt formatu u „*resources*“ direktoriju. Jedan od prvih koraka k povećanju sigurnosti u sustavu bio bi sigurnija pohrana privatnih RSA ključeva te promjena .txt formata u base64 enkodirani .pem format. .pem format je standard za pohranu i slanje kriptografskih ključeva i certifikata te bi osigurao tajnost pohranjenih ključeva. Također, sustav za AES enkripciju i dekripciju TCP poruka trenutno uvijek koristi istu lozinku i inicijalizacijski vektor. Ovo nije u skladu s AES standardom i za ostvarenje potpunih benefita korištenja AES standarda, inicijalizacijski vektor trebao bi se mijenjati za svaku enkriptiranu poruku.

Ranije je navedeno da je jedna od najbitnijih točaka daljnjeg razvoja sustava prelazak s mehanizma konsenzusa dokaza rada na neki od alternativnih sigurnijih i energetski efikasnijih mehanizama konsenzusa. Kao dobre alternative, nude se razne varijante dokaza udjela (engl. *proof of stake*): delegirani, vezani ili zakupljeni dokaz udjela, dokaz autoriteta (engl. *proof of authority*) i dokaz težine (engl. *proof of weight*).

Nadalje, jasno je da snaga i otpornost na napade tehnologije lanca blokova leži u broju dobroćudnih čvorova u mreži. Kao posljedica toga, svako tijelo koje odluči koristiti sustav za sigurno *online* glasovanje temeljen na lancu blokova za organizaciju svojih izbora trebalo bi nastojati postaviti što veći broj čvorova u mreži. Naravno, jasno je da je svaka država i organizacija financijski ograničena te da si možda ne može osigurati broj čvorova koji bi htjela za zadovoljavanje svojih sigurnosnih očekivanja. Očito rješenje ove prepreke

osigurala bi suradnja više država prilikom organizacije izbora. Kada bi primjerice Europska Unija odlučila uvesti sustav za sigurno *online* glasovanje temeljen na lancu blokova za izbore u svim svojim zemljama članicama, mogla bi zahtijevati od svake zemlje članice da podigne određen broj čvorova u mreži, sukladan njenoj veličini ili financijskim resursima. Time bi izbori svake od zemalja članica imali značajno veći broj čvorova u mreži nego što bi si država sama mogla priuštiti i izbori bi bili robusniji nego da država sama organizira izbore koristeći ovaj sustav. Naravno, ovakva bi uporaba implicirala određene preinake u sustavu. Lanac blokova onda više ne bi pohranjivao informacije samo o jednim izborima, već bi na jednom dijeljenom lancu blokova bile pohranjenje informacije o svim izborima provedenim od uvođenja sustava. Ovaj bi se lanac eventualno mogao skraćivati (uklanjanjem početnog dijela) s vremena na vrijeme kako bi se izbjegao pretjeran rast strukture lanca podataka.

Također, sigurnost sustava značajno bi se poboljšala uvođenjem revizije (engl. *audit*) lanca blokova koja bi olakšala prepoznavanje sumnjivih akcija u sustavu i rješavanje istih. Bilo bi dobro čvorove u mreži lanca blokova obogatiti funkcionalnošću raspoznavanja zloćudnog ponašanja drugog čvora od benigne greške te prijavljivanja zloćudnih čvorova u svrhu njihovog izbacivanja iz mreže. Uvođenjem procesa revizije u ovaj sustav, njegovi bi administratori postali svjesni malicioznih entiteta u mreži čak i prije nego što oni naprave ikakvu pravu štetu podacima ili sustavu te bi na vrijeme mogli suzbiti tu štetu. Ovime bi se osigurala veća otpornost na napade, a ranom reakcijom i izbacivanjem zloćudnih entiteta iz mreže, sustav bi se mogao oporaviti od napada bez gubitka podataka.

Nadalje, za uporabu sustava za sigurno *online* glasovanje na stvarnim izborima, novi autorizacijski poslužitelj bi trebao biti implementiran, s podacima o stvarnim glasačima i generiranjem tokena pristupa u skladu s kriptografskim standardima. Uz to, cijeli bi sustav trebao biti objavljen javno na internet, što trenutno nije, najlakše preko objave u oblaku koja bi, osim globalne dostupnosti, sustavu zagarantirala i razne druge zahtjeve poput manjeg vremena odziva (engl. *response time*), veće propusnosti (engl. *throughput*) i visoke raspoloživosti (engl. *availability*). Nad objavljenom aplikacijom također bi trebalo provesti testiranje opterećenja (engl. *load testing*). Testiranje opterećenja je vrsta testiranja performansi sustava kroz postavljanje simuliranih zahtjeva na sustav kako bi se osiguralo da sustav može podnijeti velik broj zahtjeva korisnika. Podnošenje izrazito velikog broja zahtjeva u kratkom vremenskom intervalu svakako je preduvjet za pravilan rad ovog sustava budući da rušenje sustava prilikom početka bitnih izbora nikako nije prihvatljivo.

Implementacijom ovih točaka, sustav za sigurno *online* glasovanje bio bi potpuniji i korak bliže uporabi na stvarnim izborima.

Zaključak

Izbori su temeljni element demokratskih sustava koji omogućuje građanima slobodno izražavanje političkih preferencija i utjecaj na budući smjer svoje države. Međutim, u mnogim je postojećim izbornim sustavima, pa tako i u hrvatskom, previše prostora za manipulaciju glasovima i rezultatima, što narušava povjerenje javnosti u validnost rezultata izbora.

Ovaj rad predstavio je potencijalno rješenje tih problema uvođenjem sustava za sigurno *online* glasovanje temeljenog na tehnologiji lanca blokova. Zahvaljujući svojstvima tehnologije lanca blokova, uporabom enkripcije, digitalno potpisanih poruka te autentifikacije i autorizacije korisnika, sustav uspješno štiti osjetljive podatke od neovlaštenog pristupa, omogućuje glasovanje isključivo verificiranim glasačima, osigurava siguran prijenos i pohranu glasova te onemogućuje njihove naknadne izmjene.

Naravno, treba razumjeti da digitalni sustav za glasovanje ne može spriječiti svaku vrstu manipulacije izbora. Postoje načini manipulacije koji se događaju van sustava, neovisno o samom procesu glasovanja kao što su predizborno podmićivanje birača i kupovanje glasova, na koje ovakav sustav ne može imati utjecaja. No, ovakav sustav uvelike povećava sigurnost podataka tijekom procesa glasovanja i time predstavlja odličan korak prema transparentnim, pravednim, sigurnim i pouzdanim izborima. Uz daljnji razvoj, predloženi sustav mogao bi postati vrijedan alat u osnaživanju povjerenja javnosti u izborne procese te promicanju istinski demokratskih izbora.

Literatura

[1] Rodriguez Cruz, G. O., *What Is Blockchain?*, Money, (2022, lipanj). Poveznica: https://money.com/what-is-blockchain/; pristupljeno: 11. lipnja 2024.

[2] Bahalul Haque, A.K.M., Bhushan, B., *Blockchain for medical insurance: Synthesizing current knowledge and problematizing it for future research avenues.* 1. izdanje, Cambridge: Academic Press, 2023.

[3] Niksefat, S., Kaghazgaran, P., Sadeghiyan, B., *Privacy issues in intrusion detection systems: A taxonomy, survey and future directions*, Computer Science Review, 25,1 (2017), str. 69-78.

[4] Paar, C., Pelzl, J. *Understanding Cryptography*. 2. izdanje. Berlin: Springer, 2024.

[5] Hong, E., *How Does Bitcoin Mining Work?*, Investopedia, (2024, travanj). Poveznica: https://www.investopedia.com/tech/how-does-bitcoin-mining-work/; pristupljeno 8. lipnja 2024.

[6] Dham, M., *What are the different types of P2P networks?*, PrepBytes Blog, (2023, kolovoz). Poveznica: https://www.prepbytes.com/blog/computer-network/what-are-the-different-types-of-p2p-networks/; pristupljeno 4. lipnja 2024.

[7] Groš, S., Đerek, A., Mikuc, M., Vuković, M., *Sigurnost računalnih sustava*, 3. izdanje, Zagreb: Fakultet elektrotehnike i računarstva, 2024.

[8] Vasireddy, S., *Protecting Azure VM against Zonal/Regional outages using Azure Site Recovery and Azure Backup*, Microsoft, (2024, siječanj). Poveznica: https://techcommunity.microsoft.com/t5/azure-storage-blog/protecting-azure-vm-against-zonal-regional-outages-using-azure/ba-p/4033280; pristupljeno 18. lipnja 2024.

[9] Barwikowski, B., *51% Attack: The Concept, Risks & Prevention*, Hacken, (2024, veljača). Poveznica: https://hacken.io/discover/51-percent-attack/; pristupljeno 16. lipnja 2024.

[10] Gupta, L., *What is REST?*, REST API Tutorial, (2023, prosinac). Poveznica: https://restfulapi.net/; pristupljeno 16. lipnja 2024.

Sažetak

Ovaj rad predstavlja sustav za sigurno *online* glasovanje temeljen na tehnologiji lanca blokova. Predloženi sustav povezuje distribuiranu mrežu čvorova s procesima enkripcije, digitalnog potpisivanja poruka, autentifikacije i autorizacije korisnika te strukturom lanca blokova kako bi osigurao temeljne sigurnosne značajke poput povjerljivosti, integriteta, raspoloživosti, autentičnosti i neporecivosti te ponudio transparentnu, pouzdanu, pravednu i sigurnu alternativu postojećim izbornim procesima.

Ključne riječi: lanac blokova, raspodjeljeni sustav, mreža ravnopravnih čvorova, sigurnost, enkripcija, glasovanje

Summary

This paper presents a secure online voting system based on blockchain technology. The proposed system combines a distributed network of nodes with encryption processes, digital message signing, authentication and authorization of users, and the blockchain structure to ensure fundamental security requirements such as confidentiality, integrity, availability, authenticity, and non-repudiation, thus offering a transparent, reliable, and secure alternative to existing electoral processes.

Keywords: blockchain, distributed system, peer-to-peer network, security, encryption, voting

Skraćenice

AAA	*Arrange-Act-Assert*	uzorak organizacije testova jedinice
AES	*Advanced Encryption Standard*	napredni standard šifriranja
API	*Application Programming Interface*	programsko sučelje aplikacije
CBC	*Cipher Block Chaining*	način rada AES standarda šifriranja
CPU	*Central Processing Unit*	središnja procesorska jedinica
FIFO	*First-In, First-Out*	prvi ušao, prvi izašao
HTTP	*Hypertext Transfer Protocol*	protokol prijenosa hiperteksta
HTTPS	*Hypertext Transfer Protocol Secure*	siguran protokol prijenosa hiperteksta
IDE	*Integrated Development Environment*	integrirano razvojno okruženje
JVM	*Java Virtual Machine*	virtualna mašina jezika Java
MVN	M*inimum Viable Product*	minimalni vitalni proizvod
PKC	*Public Key Cryptography*	kriptografija javnog ključa
REST	*Representational State Transfer*	reprezentacijski prijenos stanja
RSA	*Rivest–Shamir–Adleman*	standard za asimetrično šifriranje
TCP	*Transmission Control Protocol*	protokol kontrole prijenosa
URI	*Uniform Resource Identifier*	jedinstveni identifikator resursa

Privitak

Upute za korištenje programske podrške

1. Za lokalno pokretanje sustava, prvo morate instalirati sljedeće ovisnosti:
   - .NET 8 ili noviji
   - Java 17 ili novija
   - Maven 3.9.4 ili noviji
   - Angular 17.3.9
   - Microsoft SQL server 2022
   - opcionalno neki IDE-ovi poput Visual Studio 2022, Visual Studio Code, IntelliJ IDEA, itd.
2. Stvorite Azure račun za pohranu, prijavite se lokalno i stvorite Azure red za pohranu.
3. Nakon instaliranja ovisnosti, trebali biste urediti postavke sljedećih komponenti:
   1. Čvor u mreži lanca blokova - diplrad.constants.Constants.java
   2. Centralni koordinator čvorova - CentralPeerCoordinator.API.appsettings.json
   3. Autorizacijski poslužitelj - DummyAuthorizationProvider.API.appsettings.json
   4. Glasački poslužitelj - VotingApp.API.appsettings.json
4. Pobrinite se da postavke komponenti odgovaraju jedna drugoj:
   - vremenske oznake lanca blokova moraju biti sinkronizirane među svim komponentama
     - BlockChainCalculationStartTime mora biti prije BlockChainCalculationEndTime, koji mora biti prije BlockChainStabilizationEndTime
   - BlockChainCalculationEndTime na čvorovima lanca blokova trebao bi biti nekoliko minuta (npr. 5) nakon BlockChainCalculationEndTime na poslužitelju za glasovanje, ali i dalje prije BlockChainStabilizationEndTime

- popis kandidata mora biti isti u poslužitelju za glasovanje i klijentu
- popis javnih ključeva čvorova u poslužitelju za glasovanje mora odgovarati javnim ključevima čvorova lanca blokova
- postavke Azure pohrane i postavke krajnjih točaka moraju se podudarati u svim komponentama

5. Zatim morate pokrenuti migracije u sljedeće tri komponente koristeći EntityFramework Core:
    1. Centralni koordinator čvorova
    2. Autorizacijski poslužitelj
    3. Glasački poslužitelj
6. Dodajte certifikate centralnog koordinatora čvorova, autorizacijskog poslužitelja i glasačkog poslužitelja u pouzdana korijenski certifikacijska tijela na klijentskoj strani.
7. Konačno, možete pokrenuti komponente sljedećim redoslijedom:
    1. Autorizacijski poslužitelj
    2. Glasački poslužitelj
    3. Centralni koordinator čvorova
    4. Čvor u mreži lanca blokova
    5. Klijent