

RAZVOJ I POSTAVLJANJE BLOKČEJN-BAZIRANOG DISTRIBUIRANOG SISTEMA ZA OCENJIVANJE STUDENATA**DEVELOPMENT AND DEPLOYMENT OF A BLOCKCHAIN-BASED DISTRIBUTED SYSTEM FOR GRADING OF ASSIGNMENTS**

Aleksandar Ignjatijević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu će prvo biti predstavljeno istraživanje na temu Hyperledger Fabric blokčejna, a zatim osnove privatnih blokčejnova, kao i razvoj jedne takve mreže i njeno postavljanje. Pošto je akcenat rada na blokčejn i postavljanju istog, biće predstavljen razvoj Hyperledger Fabric mreže kao i postavljanje iste uz pomoć Kubernetes-a.

Ključne reči: distribuirani sistemi, blokčejn, Hyperledger Fabric, deployment, Kubernetes

Abstract – In this paper, an initial presentation of the research on the topic of Hyperledger Fabric blockchain will be provided, followed by the fundamentals of private blockchains, as well as the development and deployment of such a network. Since the focus of the study is on blockchain and its deployment, development of a Hyperledger Fabric network and its deployment utilizing Kubernetes will also be elucidated.

Keywords: distributed systems, blockchain, Hyperledger Fabric, deployment, Kubernetes

1. UVOD

Pojavom blokčejn tehnologije u svetu informacionih tehnologija 21. veka, otvorila se jedna od najintragantnijih tema koja, prema mišljenju nekih, postavlja osnovu za novu fazu interneta (veb 3.0). Dok je blokčejn prepoznat kao stub kriptovaluta, njegove ključne osobine poput postojanosti podataka, autonomnosti, tolerancije na greške, mogućnosti revizije, otpornosti i decentralizovanog izvršavanja otvaraju vrata za njegovu široku primenu u različitim domenima, uključujući vladine sektore, obrazovanje, finansije, industriju, zdravstvo, Internet stvari (IoT) i istraživačke sfere.

Jedan od osnovnih aspekata blokčejn tehnologije je sposobnost da omogući neizmenljivo, transparentno i sigurno čuvanje podataka, čime se otvara put ka novim paradigmama upravljanja informacijama. Univerzitetско okruženje, koje je bogato različitim formama akademskih izvora i informacija, može imati značajne koristi od primene ove tehnologije.

U ovom radu biće opisana implementacija i postavljanje platforme za upravljanje i čuvanje podataka o seminar-

skim radovima studenata. Razvijeno rešenje omogućava praćenje svakog koraka u procesu izrade, evaluacije i arhiviranja seminarskih radova, garantujući ne samo autentičnost i originalnost radova, već i pružajući uvide u akademski razvoj svakog studenta kroz vreme. Pored razvoja i implementacije, biće predstavljen i proces postavljanja ovog rešenja.

Rešenje je implementirano pomoću *Hyperledger Fabric*-a, koji predstavlja jedan od vodećih projekata u okviru *Hyperledger*-a i nudi modularnu i proširivu arhitekturu, omogućavajući efikasno upravljanje i verifikaciju podataka. U savremenim računarskim okolinama, orkestracija kontejnera postala je imperativ za efikasno upravljanje resursima i automatizaciju desetina, stotina ili čak hiljadama kontejnera. Postavljanje blokčejna izvršeno je uz pomoć *Kubernetes*-a koji se izdvojio kao vodeći orkestracioni sistem, pružajući robustne mehanizme za automatizaciju, skaliranje, raspoređivanje i operativnu koordinaciju kontejnerizovanih aplikacija u klasterima.

2. Razvoj i postavljanje privatne blokčejn mreže

Ovo poglavlje posvećeno je teorijskim osnovama razvoja i postavljanja privatnog blokčejna. To pre svega podrazumeva osnove *Hyperledger Fabric*-a i *Kubernetes*-a.

2.1. Hyperledger Fabric

Hyperledger Fabric predstavlja platformu za razvoj distribuiranih rešenja koja se zasnivaju na blokčejn tehnologiji. *Hyperledger Fabric* poseduje modularnu arhitekturu dizajniranu kako bi podržala različite vrste implementacija komponenti. Cilj ovakvog arhitekturnog dizajna je prilagođavanje različitim scenarijima korišćenja ove platforme. Pored pomenutih pogodnosti, prednost ove platforme leži u visokom nivou skalabilnosti i sigurnosti, odnosno privatnosti transakcija, koja je preko potrebna u poslovnim okruženjima. Razvoj *Hyperledger Fabric* platforme je započeo IBM, a verzija 1.0 je objavljena sredinom 2017. godine, dok se u trenutku pisanja ovog rada, *Hyperledger Fabric* nalazi na verziji 2.5.4. objavljenoj u avgustu 2023. godine. Do danas, ni jedna druga implementacije distribuiranog dnevnika transakcija nije korišćena od strane svetskih giganata poput AWS-a, Azure-a, IBM-a, Google i Oracle-a [1].

Jedan od bitnih koncepata ove platforme je postojanje različite vrste čvorova u okviru mreže. Pored toga, *Hyperledger Fabric* je uveo koncept kanala. Kanali imaju svrhu da omoguće privatnost između članova mreže, zato što poseduju sopstvene dnevnike transakcija. Korišćenje

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dušan Gajić, vanr. prof.

kriptografskih metoda omogućava samo članovima, koji učestvuju u transakciji, dekriptovanje i izvršavanje transakcije [2].

Kao što je pomenuto, *Hyperledger Fabric* nudi mogućnost odabira implementacije konsenzus algoritma u zavisnosti od potreba sistema, što predstavlja svakako pozitivnu stranu ovog projekta.

2.2. Kubernetes

Kubernetes je prenosiva, proširiva platforma otvorenog koda za upravljanje kontejnerskim radnim opterećenjima i uslugama. *Kubernetes* olakšava i deklarativno konfigurisanje i automatizaciju i ima veliki ekosistem koji se brzo razvija. *Kubernetes* usluge, podrška i alati su široko dostupni [3].

Ime *Kubernetes* potiče iz grčkog, što znači kormilar ili pilot. *K8s* kao skraćenica je rezultat brojanja osam slova između "K" i "s". *Google* je otvorio projekat *Kubernetes* 2014. godine. *Kubernetes* kombinuje preko 15 godina *Google*-ovog iskustva u vođenju proizvodnih radnih opterećenja u velikom obimu sa najboljim idejama i praksama zajednice [3].

Kontejneri su dobar način za pakovanje i pokretanje aplikacija. U produkciji je potrebno upravljati kontejnerima koji pokreću aplikacije i osigurati da nema prekida u radu. Na primer, ako jedan kontejner prestane da radi, drugi kontejner treba da se pokrene umesto njega [3]. Ovakvo ponašanje bi bilo znatno lakše kontrolisati sistemom, a *Kubernetes* je upravo jedan takav sistem. *Kubernetes* pruža okvir za otporno pokretanje distribuiranih sistema. Brine se o skaliranju i prelasku na rezervnu varijantu aplikacije, nudi oblike primene i još mnogo toga.

3. IMPLEMENTACIJA BLOKČEJN REŠENJA

U ovom poglavlju biće opisana implementacija blokčejn rešenja, kao i proces puštanja u rad jednog takvog rešenja. Biće opisana infrastruktura na kojoj je izvršeno postavljanje, kao i delovi sistema. Implementacija blokčejn rešenja je izvršena uz pomoć *Hyperledger Fabric*-a, pamentni ugovori su pisani u programskom jeziku *Go*, a blokčejn mreža je postavljena uz pomoć *K3s Rancher*-ove distribucije *Kubernetes*-a.

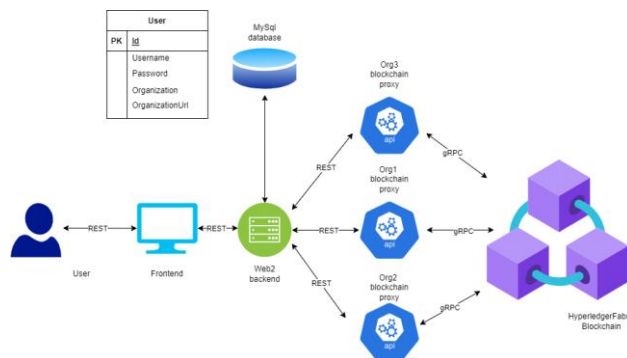
3.1. Open-rev platforma i arhitektura rešenja

Open-rev je platforma za deljenje seminarskih radova kreirana sa idejom da studenti, profesori, kao i potencijalni poslodavci, na jednom mestu imaju uvid u profesionalni razvoj studenata kroz njihove seminarske radove. Svaki rad je moguće oceniti, a kako bi se smanjila mogućnost pristrasnog ocenjivanja radova, uvedeno je i ocenjivanje samih recenzija radova, kao i kredibilitet ocenjivača.

Blokčejn implementacija se svodi na migraciju već postojećeg *web2* rešenja *Open-rev*, koje se nalazi na URL-u <https://www.open-rev.com>. Sastavni delovi sistema predstavljaju:

- korisnički interfejs,
- *back-end*,
- *MySQL* bazu podataka,
- *Hyperledger Fabric* proxy i,
- *Hyperledger Fabric* blokčejn mreža.

Na slici 1 dat je prikaz arhitekture rešenja.



Slika 1. Arhitektura Open-rev platforme

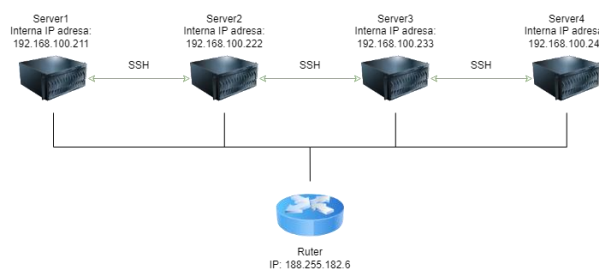
Hyperledger Fabric blokčejn mreža je organizovana u 3 organizacije. Svaka organizacija ima po jednog *peer*-a.

Već postojeći *back-end* se koristi za autentifikaciju i autorizaciju korisnika, kredencijali se drže centralizovani u *MySQL* bazi podataka, a komunikacija sa *blokčejnom* se odvija preko *proxy* servera. Postavljen je jedan *proxy* server-a, a planirano je postavljanje dodatna dva *proxy* server-a koji odgovaraju ostalim organizacijama koje su učesnici u blokčejn mreži. U pomenutoj centralizovanoj bazi, pored kredencijala i uloge u sistemu, nalazi se i podatak o tome kojoj organizaciji pripada korisnik, te se uz pomoć njega vrši rutiranje na *proxy* server odgovarajuće organizacije.

3.2. Infrastruktura

Infrastruktura se sastoji od četiri identična servera koji se nalaze na istoj lokalnoj mreži. Kako bi *bootstrap*-ovanje *Kubernetes* klastera, kao i funkcionisanje istog, bilo moguće, omogućena je *SSH* komunikacija između svih servera.

Na svim serverima je instalirana *Ubuntu 20.04* distribucija *Linux*-a. Serveri raspolazu sa 16Gb RAM-a, kao i sa 2Tb internog skladišta, i procesorom *Intel Xeon CPU E3-1220 v3* koji poseduje 4 jezgra i radi na 3.10GHz. Serveri su povezani na isti ruter i dostupni uz pomoć posvećenog port-a otvorenog na ruteru. Na slici 2 dat je prikaz infrastrukture sistema.



Slika 2. Prikaz infrastrukture

Pored *on-premise* infrastrukture, iznajmljen je i *droplet* na platformi *DigitalOcean*. *Droplet* raspolaze sa jednim virtualnim jezgrom, 25Gb skladišta i 1Gb RAM-a. Operativni sistem na *droplet*-u je *Ubuntu 20.04*. Svrha *droplet*-a je da bude *NFS* server (engl. *Network File System*), na kome će se čuvati sertifikati organizacija, blokovi postanka (engl. *Genesis blocks*) i sve ostalo što je osetljivo za samu blokčejn mrežu. *NFS* je distribuirani sistem datoteka. Svrha *NFS*-a je da obezbedi deljenje

datoteka putem mreže, na način koji veoma podseća na deljenje datoteka koje su na lokalnom sistemu [4].

3.3. Pametni ugovori

Pametni ugovori predstavljaju transakciju, odnosno parče koda koji sadrži neku transakcionu logiku koja kontrolise stanje objekta u okviru *world state*-a. Postoje dva tipa transakcije:

- *Evaluate* transakcija i
- *Submit* transakcija.

Evaluate transakcije su transakcije koje ne menjaju *world state*, već samo zahtevaju neke podatke koji se već nalaze u *world state*-u. Pored toga što ne menjaju *world state*, *evaluate* transakcije se ne pamte na *ledger*-u.

Submit transakcije su prvenstveno transakcije koje menjaju *world state*, a mogu i da vraćaju podatke sa *ledger*-a. O izvršavanju ovih transakcija se čuva podatak na *ledger*-u.

3.4. Proxy server

Kako je *Golang* programski jezik korišćen za implementaciju pametnih ugovora, odlučeno je da se i za implementaciju *proxy* servera koristi isti programski jezik. Za implementaciju *proxy* servera korišćena je *Gin* biblioteka za *web* servere napisana u *Golang*-u.

Komunikaciju između *proxy* servera i *blokčejna* je realizovana upotrebom *gRPC*-a, kao i sva ostala komunikacija u okviru *blokčejna*. *gRPC* (engl. *Remote Procedure Call*) je *open-source RPC framework* koji se koristi za komunikaciju visokih performansi između servisa razvijen od strane *Google*-a.

3.5. Testna mreža i njene komponente

U toku razvoja softverskog rešenja korišćena je blokčejn testna mreža [5]. Za tu svrhu iskorišćen je *docker-compose* alat, a testna mreža se podiže i spušta po potrebi. *Hyperledger Fabric* na svom *Github* nalogu poseduje repozitorijum po imenu *hyperledger/fabric-samples* i u okviru tog repozitorijuma priložene su implementacije osnovnih funkcionalnosti koje nudi *Hyperledger Fabric* privatna blokčejn mreža, poput primera pametnih ugovora, primera pozivanja transakcija od strane eksternog sistema, kao i primer testne mreže i pokretanja iste [6]. Testna mreža, ponuđena od strane *Hyperledger*-a sastoji se od dve organizacije koje poseduju po jednog *peer*-a, te je bilo potrebno proširiti postojeće ponuđene *docker-compose yml* datoteke, kao i skripte koje se koriste za pokretanje testne mreže, kako bi bilo moguće podržati tri organizacije sa tri *peer*-a. Kako je sadržaj potrebnih datoteka za postavljanje testne mreže poprilično obiman, neće biti prikazan u okviru ovog rada, međutim sve potrebne datoteke nalaze se na repozitorijumu projekta.

4. POSTAVLJANJE MREŽE UZ POMOĆ KUBERNETES-A

Postavljanje *Hyperledger Fabric*-a je obavljeno uz pomoć *Kubernetes* orkestratora. S obzirom da postavka *Hyperledger Fabric*-a podrazumeva postavljanje više komponenti, proces postavke je izvršen u nekoliko faza. Proces postavke se može okvirno podeliti na:

- fazu pripreme,

- fazu postavljanja *Fabric CA*,
- fazu generisanja sertifikata i početnih artifakata,
- fazu postavljanja *orderer*-a i *peer*-ova,
- fazu instaliranja *chaincode*-a,
- fazu postavljanja *proxy* servera i
- fazu postavljanja *ingress* kontrolera.

Manifesti se primenjuju sa izvršavanjem komande *kubectl apply -f naziv_manifesta* u okviru direktorijuma u kom se nalazi manifest. Manifesto su dostupni na *Github* repozitorijumu projekta.

4.1. Faza pripreme

U okviru faze pripreme izvršeno je par ključnih koraka koji su potrebni za dalje faze. Prvi korak u ovoj fazi podrazumeva prebacivanje već postojećih skripti, kao i prebacivanje ključnih konfiguracionih *yml* datoteka na zakupljeni *droplet* na platformi *DigitalOcean*, koji služi kao *NFS* server. Nakon uspešnog prebacivanja svih potrebnih datoteka, odvojen je deo skladišta na klasteru u vidu *persistentVolume*-a (skr. *PV*). Naredni korak u pripremnoj fazi je postavljanje *persistentVolumeClaim*-a (skr. *PVC*). *PVC* predstavlja zahtev za korišćenje *PV*-a. U narednim koracima, svi *Deployment*-i *orderer*-a, *peer*-ova i *CA*-eva će koristiti taj kreirani *PVC*.

4.2. Faza postavljanja Fabric CA

Ova faza podrazumeva kreiranje *Deployment* i *Service* manifesta za *Fabric CA* koji je zadužen za svaku organizaciju pojedinačno, kao i samog *orderer*-a. *Deployment* je korišćen kako bi se omogućilo potencijalno skaliranje ili ažuriranje podešavanja postavljenih *Pod*-ova. Drugi deo postavljanja *Fabric CA* predstavlja postavljanje *Service*-a, kako bi ti *Pod*-ovi koji su kreirani uz pomoć *Deployment*-a, bili dostupni na mreži unutar klastera

4.3. Faza generisanja sertifikata i početnih artifakata

U ovoj fazi postavljanja *Hyperledger Fabric*-a generišu se sertifikati koji će koristiti *orderer* i organizacije, a za to je kreiran manifest koji će kreirati *Job*. *Kubernetes Job* kreira *Pod* kako bi obavio određeni posao, a to u ovoj konkretnoj fazi predstavlja generisanje potrebnih sertifikata. U tu svrhu koristi se *hyperledger/fabric-ca-tools Docker* slika za kreiranje kontejnera u kome će se izvršiti skripte zadužene za generisanje pomenutih sertifikata. U narednom koraku ove faze generišu se početni artifakti uz pomoć drugog *Job*-a. Rezultat ovog *job*-a su datoteke *genesis.block*, *Org1MSPanchors.tx*, *Org2MSPanchors.tx*, *Org3MSPanchors.tx* i *mychannel.tx* datoteke.

4.4. Faza postavljanja orderer-a i peer-ova

Poput postavljanja *Fabric CA*, postavljanje *orderer*-a se svodi na kreiranje *Deployment*-a i *Service*-a na veoma sličan način kao i pomenuti *Fabric CA*. Za kreiranje kontejnera koristi se *hyperledger/fabric-orderer Docker* slika. Kako je *Hyperledger Fabric* platforma za rešenja sa distribuiranom glavnom knjigom i mrežom sa kontrolisanim pristupom, uređivački servis je veoma bitna komponenta sistem i čini ga grupa *orderer* čvorova. Ovaj servis je zadužen za uređivanje transakcija u blokove. Postavljeno je pet *orderer Pod*-ova uz pomoć *Deployment*-a i *Service*-a.

4.5. Faza instaliranja *chaincode*-a

U pripremnoj fazi, prebačen je *chaincode* direktorijum u okviru koga se nalazi direktorijum *packaging*. Sadržaj ovog direktorijuma predstavljaju tri *tgz* datoteke: *basicv1-org1.tgz*, *basicv1-org2.tgz* i *basicv1-org3.tgz*. Ove datoteke se sastoje iz dve datoteke: *code.tar.gz* i *metadata.json*. Instalacija *chaincode*-a koristi ove datoteke. Rezultat instalacije je *hash* instaliranog *chaincode*-a i on je potreban za uspešno postavljanje *Deployment*-a eksternih *chaincode* servera, te je veoma bitno zapamtiti taj *hash*. Nakon uspešnog instaliranja *chaincode*-a, sledeći korak u ovoj fazi je postavljanje *Deployment*-a i *Service*-a za *chaincode* servere. Nakon postavljanja *Deployment*-a i *Service*-a, potrebno je potvrditi (engl. *approve*) *chaincode* na svakom od *peer*-ova. Kako svaki *peer Pod* koristi i kontejner za *CouchDb*, moguće je povezati se na interfejs *CouchDb*-a na adresi http://localhost:5984/_utils/.

4.6. Faza postavljanja proxy servera

Poput postavljanja *orderer*-a i *peer*-ova, postavljanje *proxy* servera se svodi na kreiranje *Deployment*-a i *Service*-a, na veoma sličan način kao i pomenuti resursi. Za kreiranje kontejnera koristi se prethodno kreirana *Docker* slika *oakenknight/openrev-gateway-api*. Uz pomoć *environment* varijabli postavlja se identitet organizacije kojoj pripada određeni proxy server, kanal na koji će se povezivati, naziv *chaincode*-a koji će biti korišćen i podaci o *MinIO* bucket-u koji će biti korišćen za skladištenje radova. Port koji će biti otvoren na *api-staging Pod*-u je port 80. *MinIO* predstavlja distribuirani datotečki sistem postavljen sa svrhom perzistentnog čuvanja seminarskih radova na *Kubernetes* klasteru.

4.6. Faza postavljanja proxy servera

Ingress kontroler predstavlja objekat koji služi za eksterni pristup *Service*-ima u okviru *Kubernetes* klastera, a kako je potrebno da *proxy* server koristi *back-end Open-rev* platforme, potrebno je postaviti i ovaj *Kubernetes* resurs. Prilikom kreiranja *Ingress* kontrolera, postavlja se putanja na kojoj će biti dostupan *API* koji se izlaže uz pomoć *Ingress* kontrolera. Kao *Ingress* kontroler korišćen je *Nginx Ingress* controller.

5. ZAKLJUČAK

U okviru ovog rada, razmatrana je primena blokčejn tehnologije u okruženju akademskih institucija, konkretno u procesu praćenja i upravljanja seminarskim radovima studenata. Postavljanje takvog sistema implementirano je koristeći *Hyperledger Fabric* platformu, koja je omogućila kreiranje transparentnog i sigurnog okruženja za čuvanje i praćenje podataka o seminarskim radovima. Ovaj sistem je pokazao značajne prednosti u pogledu integriteta, revizije i decentralizovanog pristupa podacima, dok je u isto vreme obezbedio mehanizme za očuvanje privatnosti i autentičnosti informacija.

Implementacijom blokčejn rešenja uz pomoć *Hyperledger Fabric* platforme, omogućena je pouzdana i transparentna razmena informacija među relevantnim učesnicima - studentima, mentorima i akademskim osobljem. *Hyperledger Fabric* se pokazao kao robustna platforma koja može da odgovori na zahteve takvog sistema u smislu performansi, sigurnosti i skalabilnosti.

Ovaj rad otvara prostor za dalja istraživanja u primeni blokčejn tehnologije u obrazovnim institucijama, i pruža temelj za razvoj proširenih sistema koji mogu uključivati i druge aspekte akademskog života. Takav razvoj može doprineti stvaranju efikasnijih, transparentnijih i sigurnijih okruženja za akademsku zajednicu, čime se unapređuje celokupni akademski proces. Takođe, analiza izazova i prednosti u implementaciji blokčejn tehnologije pružila je vredne uvide koji mogu voditi buduće inicijative u oblasti obrazovanja i tehnologije.

Kod dostupan na linku:

<https://github.com/OakenKnight/OpenRev>.

6. LITERATURA

- [1] Hyperledger Fabric Whitepaper, <https://8112310.fs1.hubspotusercontent-na1.net/hubfs/8112310/hyperledger-fabric-whitepaper.pdf>, pristupljeno 06.09.2023.
- [2] What is Hyperledger Fabric, <https://hyperledger-fabric.readthedocs.io/en/latest/blockchain.html>, pristupljeno 06.09.2023.
- [3] Kubernetes documentation: Overview, <https://kubernetes.io/docs/concepts/overview/>, pristupljeno 19.09.2023
- [4] Network File System, <https://www.ibm.com/docs/en/aix/7.1?topic=management-network-file-system>, pristupljeno 19.08.2023
- [5] Fabric Samples, <https://github.com/hyperledger/fabric-samples>, pristupljeno 19.08.2023.
- [6] Hyperledger Fabric documentation: Test network, https://hyperledger-fabric.readthedocs.io/en/release-2.5/test_network.html, pristupljeno 19.08.2023.

Kratka biografija:



Aleksandar Ignjatijević rođen je u Novom Sadu 19. aprila 1998. godine. Fakultet tehničkih nauka u Novom Sadu, studentski program Računarstvo i automatika upisao je 2017. godine. Nakon završenih osnovnih studija, 2021. godine, upisao je master akademske studije iz iste oblasti.

kontakt: a.ignjatijevic@gmail.com