

Kreirati serversku aplikaciju koja će omogućiti komunikaciju sa Hyperledger Fabric mrežom, kroz pozive chaincode funkcija.

Aplikacija će podržati funkcionalnosti:

- enroll / login korisnika
- upit (query) chaincode-a
- izazivanje (invoke) chaincode-a

Serverska aplikacija mora da koristi SDK (NodeSDK, JavaSDK, GoSDK...) za rad sa chaincode-om.

Aplikacija za interakciju sa chaincode-om, može da koristi samo jedan sertifikat, može i druge, ali je jedan dovoljan.

Potrebno je da Fabric blockchain obezbedi praćenje i poslovnu logiku kojom se obezbeđuju funkcionalnosti opisane u daljem tekstu.

U sistemu inicijalno postoji nekoliko automobila koji su modelovani na sledeći način:

- idAutomobila(jedinstveni identifikator)
- marka
- model
- godiste
- boja
- idVlasnika
- lista kvarova (gde kvar sadrži opis i cenu popravke). Primer kvara je probušena desna prednja guma

Pored automobila u sistemu se čuvaju i podaci o osobama (vlasnik automobila, majstor itd.) .

Osoba ima:

- idOsobe(jedinstveni identifikator)
- ime
- prezime
- adresu elektronske pošte
- količinu novca koju poseduje

Prethodno opisane svojine se upisuju u WorldState, prilikom INIT funkcije chaincode-a, ili

neke druge invoke funkcije koja će postaviti stanje tako da sadrži minimum 6 automobila i minimum 3 osobe.

Funkcionalnosti koje će chaincode obezbediti jestu: promena vlasništva, kreiranje kvara, promena boje automobila i popravka automobila.

Zahtevi za projektni zadatak:

- ❑ Kreiranje Fabric mreže sa jednim kanalom
 - specifikacija uključuje tri organizacije sa po tri peer-a
 - generisanje potrebnih krypto materijala
 - specifikacija docker kontejnera za elemente Fabric mreže
 - kanalu mora biti pridružen bar 1 orderer
 - svi peer-ovi se nalaze na pomenutom kanalu
 - kreiranje svih preostalih neophodnih učesnika u mreži kao i njihove kriptomaterijale (CA, Orderer)
 - korišćenje CouchDB-a nije obavezno
 - svi peer-ovi imaju ulogu endorser-a i commiter-a
- ❑ Chaincode funkcionalnosti (sve podrazumevaju umanjivanje sredstava platioca usluga i povećanje sredstava naručioca usluga odnosno prodavca)
 - promena vlasništva
 - promena vlasništva je moguća samo ukoliko automobil nema kvarove, a u slučaju da ima kvarove, cena se umanjuje za ukupnu cenu kvarova, i to samo ako kupac izrazi želju da kupi auto čak i ako ima kvar (to može biti specificirano kao parametar invoke funkcije)
 - promena boje
 - pri čemu se boja specificira kao parametar funkcije, a boja je predstavljena kao stringovski podatak
 - kreiranje kvara
 - podrazumeva da se funkcija pozove sa parametrom oznake automobila na koji se kvar odnosi, kao i kompletan podatak o kvaru koji će se dodati u listu kvarova pomenutog automobila
 - ukoliko ukupna cena aktivnih kvarova na automobilu postane veća od cene automobila, potrebno je taj automobil izbrisati iz WorldState-a
 - popravka automobila
 - podrazumeva da se kvarovi označe kao popravljeni ili obrišu iz podatka o automobilu
 - query funkcije koje omogućuju pretragu po boji, po vlasniku i boji istovremeno
 - potrebno je testirati sve implementirane funkcionalnosti (kroz pozive funkcije ili REST pozive)
 - potrebno je obraditi sve moguće greške (ukoliko ne postoji automobil ili osoba sa zadatim ključem u WorldState-u, nedovoljna količina sredstava i drugo)

- ❑ U world state-u se čuvaju objekti u JSON obliku, serverska aplikacija vraća JSON podatke takođe.
- server koji koristi SDK treba da omogućiti:
 - enroll – login
 - query
 - invoke

Napomene:

- Preporučuje se pisanje chaincode-a u Golang-u,
- Preporučuje se verzija Fabric-a 2.2, a moguće je koristiti i starije (drugacije su skripte i rad sa SDK-om)
- OrderingService je tipa SOLO, potrebno je da commit bloka radi tako da se bar 2 transakcije nađu u bloku ili na svaki sekund, šta god se desi prvo,
- moguće je koristiti CouchDB (nije obavezno)
 - U tom slučaju potrebno je dozvoliti vršenje upita po proizvoljnom skupu polja o za generisanje genesis bloka, obavezno je korišćenje profila
- Imenovanje elemenata mreže (MSP, Org, Peer, Channel, Orderer itd) je ostavljeno studentu, preporučuje se imenovanje koje je pokazano na vežbama
- Preporučuje se da se za osnovu razvoja mreže koriste test-network i asset-transfer-basic primeri
- Specificiranje pravila prihvatanja transakcije, odnosno endorsement policy za chaincode je ostavljen studentu da specificira kako želi
- Sve što nije ograničeno student može uraditi kako želi, ili se može obratiti asistentu za dodatna pojašnjenja
- Serversku aplikaciju možete testirati kroz pozive u SDK aplikaciji, ili napraviti REST endpoint-e pa testirati preko POSTMAN-a. Pri testiranju aplikacije kroz SDK **obavezno je koristi CA (Certificate Authority) i obavezno je testirati sve funkcionalnosti kroz rad sa peer-ovima i userima koji pripadaju Organizaciji 3.**
- .sh skripte za sve potrebne operacije, koje omogućuju testiranje i pokretanje mreže su obavezne
- Sve potrebne fajlove, zajedno sa uputstvom za testiranje i pokretanje mreže, je potrebno arhivirati i poslati na email horva.n@uns.ac.rs ili dozvoliti asistentu pristup git repozitorijumu (NebojsaHorvat)