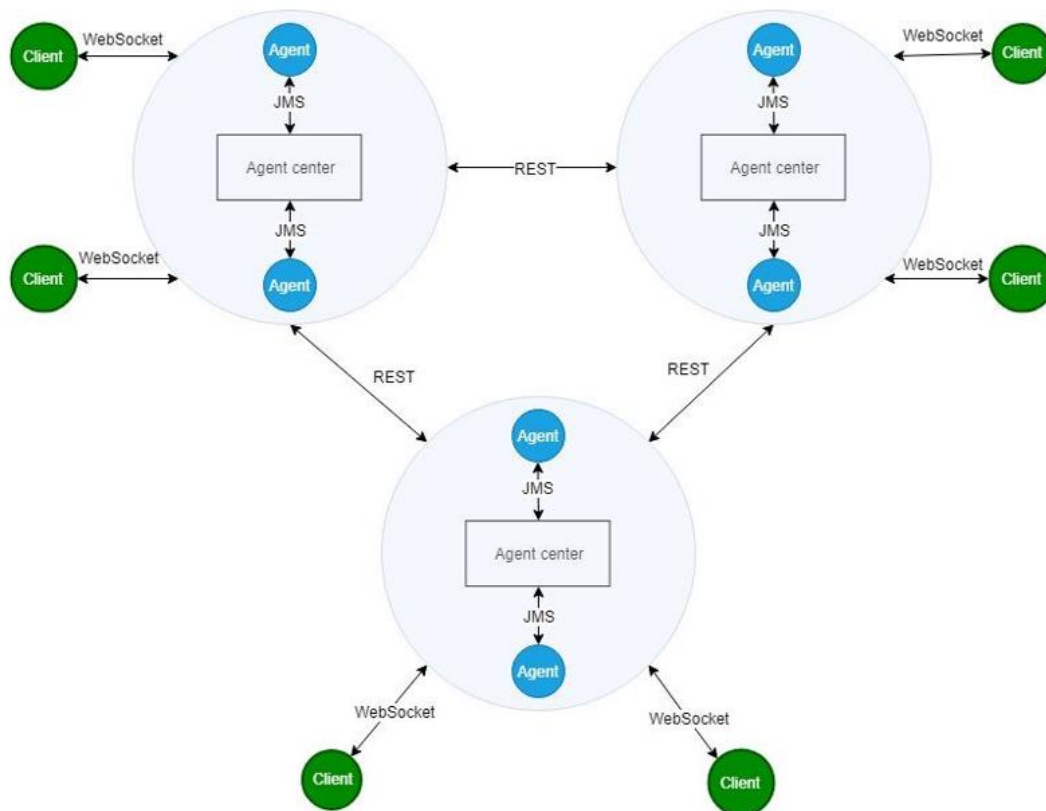


CHAT Zadatak

Implementirati klijent-server aplikaciju za chat koje će predstavljati agentsko okruženje upotrebom JEE platforme.

Sistem se sastoji od sledećih element:

- Host – čvor mreže koja predstavlja agentsko okruženje. Ovi čvorovi su zaduženi za upravljanje životnim ciklusom agenta, kao i za razmenu poruka među agentima
- Agent – softverski entitet koji izvršava zadatak; u našem zadatku to je Korisnik
- Klijent – UI aplikacija preko koje će korisnik moći da se registruje, prijavi na sistem i razmenjuje poruke sa ostalim korisnicima. Treba implementirati uz pomoć neke od sledećih biblioteka: Angular, React, Vue.js ili standardnog html i css.



Model podataka

Korisnik sadrži polja korisničko ime, šifra i host na kojem se prijavio korisnik

Host sadrži polja: alias(naziv čvora) i address

Poruka sadrži polja: primaoc (Korisnik), pošiljalac (Korisnik), datum kreiranja, zaglavlje (subject) i sadržaj.

Komunikacija

Svaki REST zahtev ima prefiks „<host address>:<host port>”.

Klijent-Server

Klijent je front-end aplikacija preko koje se korisnik registruje, prijavljuje na sistem i šalje poruke drugim korisnicima. Implementirati sledeće REST protokole:

- POST /users/register – registruj korisnika (zadaje se korisničko ime i šifra)
- POST /users/login – prijava korisnika na sistem
- GET /users/loggedIn – spisak svih prijavljenih korisnika na sistemu
- GET /users/registered – spisak svih registrovanih korisnika na sistemu
- POST /messages/all – pošalji poruku svim prijavljenim korisnicima
- POST /messages/user – pošalji poruku korisniku
- GET /messages/{user} – spisak svih poruka (prijavljenog korisnika)
- DELETE /users/loggedIn/{user} – odjavi korisnika sa sistema

Klijentska aplikacija treba da podržava automatsko osvježavanje liste prijavljenih korisnika putem WebSocket-a.

Server-Server

Server čini jedan čvor u mreži. Kada se pokrene prvi host on postaje master čvor mreže. Svaki naredni host koji se podigne treba da kontaktira master čvor kako bi postao deo mreže. Prilikom aktiviranja novog ne-master čvora potrebno je da se izvrši handshake, baziran na REST pozivima, u kom će:

- POST /register – nov ne-master čvor kontaktirati master čvor koji ga registruje
- POST /node – Master čvor javlja ostalim ne-master čvorovima da je nov ne-master čvor ušao u mrežu
- POST /nodes – Master čvor dostavlja spisak ostalih ne-master čvorova novom ne-master čvoru
- POST /users/loggedIn – Master čvor dostavlja spisak prijavljenih korisnika novom ne-master čvoru koji se nalaze kod njega ili nekog od preostalih ne-master čvorova; Nije neophodno da naveden handshake bude u potpunosti ispoštovan što se tiče navedenih metoda (moguće je smanjiti broj zahteva upotrebom povratnih vrednosti funkcija). Ako se u bilo kom trenutku desi otkaz (tako što se, npr. desi timeout) vrši se ponovni pokušaj datog zahteva. Ako se otkaz ponovo desi potrebno je izvršiti rollback, što podrazumeva „čišćenje“ svakog čvora sistema od informacija koje je nov ne-master čvor ubacio u sistem sa sledećim REST pozivom:
- DELETE /node/{alias} – Master čvor javlja ostalim ne-master čvorovima da obrišu čvor koji nije uspeo da izvrši handshake, kao i sve tipove agenata koji su potencijalno dostavljeni ostalim čvorovima. Ova operacija se takođe treba eksplicitno pokrenuti kada se neki čvor priprema za gašenje i želi da se odjavi iz klastera. Prilikom gašenja čvora treba pogasiti i sve agente koji trče na datom čvoru

Svaki put kada se desi prijava korisnika na sistem potrebno je izvršiti POST /users/loggedIn zahtev na sve preostale čvorove u mreži, kako bi svi imali informaciju o najnovijem korisniku. Kada se ažurira lista prijavljenih korisnika potrebno je obavestiti WebSocket klijetne da ažuriraju svoje liste. Kada neki čvor prihvati poruku namenjenu za određenog korisnika on prosleđuje tu poruku čvoru na kom se korisnik nalazi.

Svaki čvor u klasteru treba da implementira heartbeat protokol kojim periodično proverava da li su ostali članovi klastera i dalje živi. Protokol se svodi na sledeći REST poziv:

- GET /node – Ukoliko se desi da čvor ne odgovori zahtev se izvršava još jednom i ukoliko čvor ni tada ne odgovori smatra se da je ugašen i javlja se ostalim čvorovima da izbace zapis o ugašenom čvoru.

BODOVANJE

Za 20 bodova treba implementirati agentsko okruženje koje ne radi u klasteru (samo Klijent-Server komunikacija).

Za 30 bodova potrebno je podržati klaster između Host-ova. (Komunikacija Klijent-Server i Server-Server)