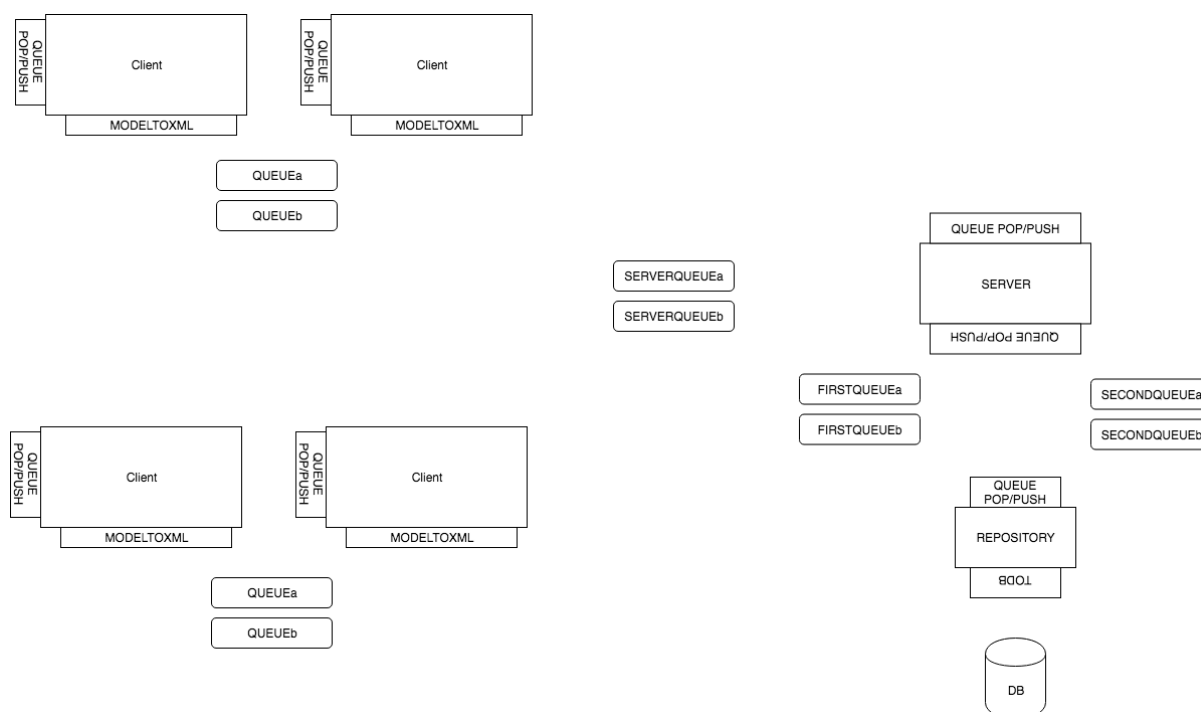


## Projektni Zadatak 2

Potrebno je napraviti dizajn sistema, arhitekturu sistema, implementira i istestirati rešenje koji simulira rad sistema sa slike.



### Opis rada sistema:

Postoji  $n$  klijenata u sistemu i 1 server. Svaki klijent poseduje svoje stanje čiju izmenu deli sa klijentima sa kojima je u vezi kako bi i oni mogli da osveže svoj prikaz. Svi klijenti se registruju na sistem ( server). Prilikom registrovanja oni zapravo trebaju da kažu da li prave novi par redova preko kog će deliti informacije o promenama svog stanja, ili se prijavljuju na već postojeći par redova. Ukoliko klijent pravi novi par redova on mora tu informaciju da pošalje serveru kako bi server mogao na svojoj strani da napravi svoj par redova. Kada se klijenti povežu, svaku promenu svog stanja ubacuju (objavljaju) u red za čitanje (QUEUEa), server čeka dok god nema promena, kada uoči promenu skida je sa reda za čitanje i ubacuje u odgovarajući red za čitavanje u okviru svog para redova namenjenom paru sa kog je poruka skinuta. Sa tog drugog reda za čitanje (FIRSTQUEUEa) komponenta skida poruku i upisuju promenu u bazu, rezultat upisa(promenjeno stanje objekta ili poruku o grešci) stavlja u red za pisanje na serverskog strani (FIRSTQUEUEb) odakle je server skida i ubacuje u red za pisanje na klijentskog strani (QUEUEb), svi klijenti (uključujući i onog koji je promenu objavio) čekaju dok je taj red prazan i kad u taj red stigne poruka sa tog reda je skidaju i ažuriraje odgovarajuće stanje ili obrađuju prosleđenu poruku o grešci. u slučaju da promena stanja iz nekog razloga nije uspela klijent koji je napravio promenu pokušava ponovo tu promenu da objavi (ponavlja se prethodno opisani postupak). Ovo se dešava sa svim klijentima u

sistemu. Svaki klijent kada zeli iz nekog razloga da komunicira sa serverom (CREATE i SUBSCRIBE u nastavku teksta) to radi putem para redova SERVERQUEUEa i SERVERQUEUEb, pri čemu server kada obradi zahtev klijenta stavlja u red odgovor sa naznakom za kog klijenta je taj odgovor i ostali klijentu zanemaruju dobijen odgovor ako nije namenjen njima.

Klijent serveru može da šalje sledeće vrste poruka:

- CREATE - kreiranje novog para redova na serverskoj strani,
- SUBSCRIBE - novi klijent pokušava da se prijavi na postojeći red,
- UPDATE - postupak ažuriranje stanja.

Svaki od ovih tipova poruka prati odgovarajući payload (npr. naziv reda koji treba da se kreira, pri čemu se kreira par redova, automatski se dodaju znakovi a/b na kraj naziva, obratiti pažnju da naziv reda mora biti jedinstven).

Svaki klijent ima model podataka koji se sastoji od sledećih elemenata: items, positions, clients ( lista klijenta koji su u vezi sa ovim klijentom, tj. povezani na isti par redova, klijent se u ovu listu dodaje metodom SUBSCRIBE, tako što se serveru prosledi SBSCRIBE xml\_verzija\_stanja\_klijenta i onda se iz XML-a napravi klijent i ubaci u listu). Svaki item ima svoj naziv, količinu, da li je aktiviran, razornu moć u J, a pozicija je opisana x,y,z koordinatama.

### Kriterijum ocenjivanja

1. Dizajn I arhitektura rešenja
2. Korišćenje Scrum metodologije razvoja – definisanje User Story-a i taskova, planiranje i estimacija
3. Implementacija rešenja
4. CI ciklus
  - a. Build
  - b. UnitTestovi
  - c. Pokrivenost koda testovima