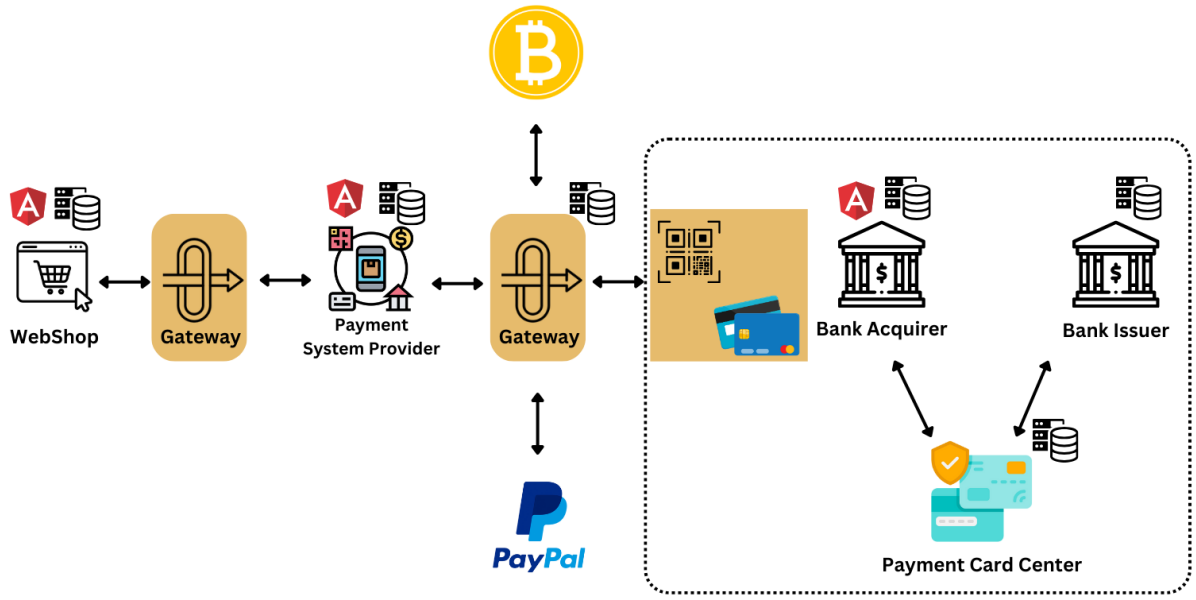
System design – SEP-Telekomunikacioni operater



Kupac inicira proces registracijom na web shop-u i odabirom usluge (paketa, servisa) na koji želi da se pretplati.

Opisi komponenti:

web shop –web aplikacija za čiju smo izradu koristili .NET za serversku stranu aplikacije i Angular za klijentsku. Na klijentskoj aplikaciji korisnik se registruje, bira usluge na koje želi da se pretplati, ima pregled svih uspešnih transakcija usluga na koje je pretplaćen. Za potrebe web shop-a koristi se posebna relaciona baza u kojoj se čuvaju podaci o korisnicima, njihovim pretplatama i transakcijama koje su inicirali.

gateway-web shop-psp – posebna aplikacija koja ima ulogu da obezbedi losely coupling izmedju web shopa i psp-a. Aplikacija je podignuta kao Spring Cloud Gateway sa oslanjanjem na Eureka server koji je takođe podignut. Aplikacija preusmerava zahteve (za pretplatu web shop-a na psp i transakcije) sa web shop-a na psp i odgvoore koji pristižu od odgovarajućeg servisa za plaćanje od psp-a do web shop-a. Aplikacija ima ulogu i load balancer-a time što zahteve raspoređuje ravnomerno. Omogućuje da web shop i psp ne poznaju međusobne API-je, već on poznaje adrese na kojima se oni nalaze.

psp- web aplikacija koja nudi korisniku opcije plaćanja na koje se web shop prethodno pretplatio kod psp-a (za to se na web shop-u nude na klijentskoj strani opcije u zavisnosti od postojećih opcija koje postoje u bazi psp-a). Za klijentsku stranu je korišćen Angular, a za serversku Spring. U bazi se čuvaju podaci o pretplaćenim klijentima (web shop-ovima) i opcijama plaćanja na koje su pretplaćeni, kao i podaci o transakcijama koje su inicirane (ko ih je inicirao, iznos, vreme iniciranja). Ova aplikacija ima zasebnu klijentsku aplikaciju kako bi se osigurao što veći loosely coupling sa kliljentima.

gateway-psp-payment services – posebna web aplikacija koja bi obezbedila plagabilnost sistema odnosno lako dodavanje novih opcija za placanje. Aplikacija je podignuta kao Spring Cloud Gateway sa oslanjanjem na Eureka server koji je takođe podignut. Za potrebe ove aplikacije podignuta je posebna baza podataka u kojoj se čuvaju opcije plaćanja i putanje na kojima se nalaze servisi koji su zaduženi za obradu zahteva za transakcijama korišćenjem te usluge. Iz header-a Payment iz pristiglih zahteva se čita opcija plaćanja, a zatim se pronalazi putanja u bazi i prosleđuje zahtev dalje (Za ovo je

napravljen filter koji presreće zahteve). Aplikacija takođe preusmerava podatke o klijentima koji su se pretplatili kod psp-a dalje u banku i odgovore iz banke u psp.

Opcije plaćanja su realizovane kao mikroservisi (Mikroservis za PayPal, Bitcoin, Banka(QR kod i kartica) kako bi se omogućilo lako dodavanje i uklanjanje opcija plaćanja, jer svaki mikroservis sadrži poseban vid obrade transakcija. Svakako time se postiže i mogućnost za dobre performanse sistema, podizanjem novih instanci servisa koji su opterećeni.

Banka --->Banka obuhvata aplikaciju banke prodavca, banke kupca i PCC.

Banka prodavca i **banka kupca** su podignute kao 1 web aplikacija sa 2 različite konfiguracije. Za serversku stranu je korišćen Spring, za klijentsku Angular. Na klijentskoj aplikaciji korisnik (inicijator transakcije) unosi podatke o sredstvu plaćanja (podatke o kartici ili mu se prikazuje QR kod). Svaka banka koristi posebnu bazu u kojoj se čuvaju podaci o računu klijenta u toj banci i svim transakcijama za svaki račun. Registrovanjem klijenta kod psp-a u banci prodavca se otvara račun za tog klijenta. Po iniciranju plaćanja i unosu podataka o računu serverska strana proverava ispravnost podataka kako bi se inicirala transakcija.

Ukoliko se na osnovu prvih 6 cifara pan-a ustanovi tokom te provere da kupac ima račun u drugoj banci, inicirana transakcija se prosleđuje do **Payment Card Centra**. PCC je realizovan kao posebna web aplikacija koja ima samo serversku stranu, na kojoj se vrši pronalazak odgovarajuće banke kojoj treba proslediti transakciju na osnovu računa. PCC koristi posebnu bazu u kojoj se čuvaju adrese servisa na kojima se nalaze banke kojima treba proslediti transakciju, kao i identifikator banke (6 cifara). PCC preusmerava i odgovor banke kupca, banci prodavca.

Dakle kao odgovor na tri nefunkcionalna zahteva sistema nudimo sledeće rešenje:

1. **Losely coupling** – Kako bismo obezbedili loosely coupling između web shop-a i psp-a postavili smo gateway između njih. On omogućava da web shop ne poznaje api psp-a, load balancing, nezavisan razvoj i izmene ova dva sistmea, izolaciju poslovne logike web shopa od specifičnih implementacija PSP-a. Odabrali smo Spring Cloud Gateway sa Eurekom jer omogućava dinamičko otkrivanje servisa, centralizovano rutiranje i balansiranje opterećenja, kreiranje filtera, ugrađene mehanizme za rate limiting i token validaciju, što dodatno osigurava bezbednost transakcija prema PSP-ovima. Dodatno, loosely coupling smo povećali razdvajanjem klijentskih aplikaija web shop-a i psp-a.
2. **Plagabilnost** – Lako dodavanje novih opcija plaćanja smo postigli dodavanjem gateway-a između psp-a i banke. Svaka nova opcija plaćanja i adresa na kojoj se nalazi servis zadužen za tu opciju plaćanja se lako doda u bazu ovog gateway-a, pa se na osnovu opcije pročitane iz header-a zahteva, zahtev dalje preusmerava. Takođe opcije plaćanja se čuvaju u bazi psp-a, te je lako njihovo dodavanje i uklanjanje.
3. **Visoka dostupnost sistema** – Monitoringom sistema mogla bi se pratiti opterećenost i stanje svake od komponenti, te se ustanovit za koju komponentu će biti potrebno podizanje dodatnih instanci. Ako bismo koristili docker i kubernetes, mogli bismo da podizanjem novih instanci docker-a podižemo nove instance servisnih komponenti i orkestriramo ih.