

Hvala.

Sada cu vam prezentovati svoj diplomski rad. U dogovoru sa mentorom Željkom Vukovićem, ova prezentacija neće biti previše dugačka.

Naslov mog rada je Integracija servisa za pretragu letova primenom Enterprise Service Bus obrasca.

Prvo cemo se osvrnuti na teorijsku osnovu Enterprise Service Bus obrasca. Enterprise Service Bus je arhitektonski obrazac kojim se koristi za integraciju aplikacija i servisa unutar sistema i olakšava njihovu komunikaciju. ESB služi kao centralni komunikacioni kanal koji omogućava da aplikacije i servisi međusobno komuniciraju i razmenjuju podatke bez direktnih veza.

Glavne uloge ESB su:

- Posredovanje i transformacija: ESB posreduje i transformiše podatke između različitih servisa, koji koriste različite formate podataka i protokole.
- Rutiranje poruka: Rutira poruke između servisa na osnovu unapred definisanih pravila.
- Konverzija protokola: Podržava komunikaciju između servisa koji koriste različite protokole.
- Orkestracija servisa: Koordinira složene interakcije između servisa, često formirajući tok rada ili poslovni proces.
- Sigurnost: Obezbeđuje bezbednu komunikaciju između sistema, često pružajući autentifikaciju, autorizaciju i enkripciju.
- Upravljanje greškama i beleženje (logging): Centralizovano upravljanje greškama i beleženje omogućavaju praćenje statusa poruka, servisa i događaja u sistemu.

Prednosti ESB:

- Skalabilnost: Omogućava lako integrisanje novih sistema bez potrebe za promenom postojećih.
- Odvajanje (decoupling): Aplikacije su slabo povezane što znači da ne utiču jedna na drugu.
- Fleksibilnost: Organizacija može da dinamično menja, skalira i upravlja servisima.

Tipični slučajevi korišćenja:

- Integracija starih, legacy sistema sa modernim aplikacijama.
- Omogućavanje komunikacije između servisa u servisno-orijentisanoj arhitekturi.

- Rukovanje složenim tokovima poruka.

Sada ćemo preći na arhitekturu mog sistema. Na dijagramu možete videti celokupan sistem sastavljen od 6 servisa koji su integrisani uz pomoć Enterprise Service Bus-a koji opslužuje klijentsku aplikaciju. Sve sisteme osim Exchange-Rate API-ja sam implementirao ja, sa fokusom na raznovrsnost protokola, formata i struktura poruka.

Najbolje bi bilo da se kratko osvrnemo na svaki servis pojednicano.

U sistemu su prisutna 3 sistema avio kompanija. Prvi sistem je klasičan RESTful servis koji komunicira uz pomoć JSON formata poruka. Drugi sistem je SOAP Web servis, koji komunicira uz pomoć SOAP protokola i XML formata poruka koji su jasno definisani XML semama i WSDL-om. Sistem avio kompanije je GraphQL servis koji takođe koristi JSON format za komunikaciju kao i prvi sistem, ali sa znatnim razlikama u strukturi poruka u odnosu na prvi sistem. Uloga ovih sistema za klijente je pružanje informacija o letovima kroz pretragu u okviru jedne avio-kompanije, kao i rezervisanje karata i njihovo dostavljanje putem email-a u PDF formatu.

Sistem za pretragu aerodroma je mali pomoćni sistem koji pomaže korisniku pri odabiru aerodroma polaska i destinacije. U pitanju je RESTful servis koji komunicira uz pomoć CSV formata poruka.

Sistem za pretragu letova sa presedanjem učestvuje u sistemu ukoliko korisnik omogući prikaz letova koji ne stižu do destinacije u jednom koraku, već zahtevaju presedanje. U pitanju je RESTful servis koji koristi JSON format podataka. Ovaj sistem dobija podatke od ESB-a nakon što su dobavljeni podaci iz sve 3 avio-kompanije i nakon što su transformisani u univerzalni JSON format i grupisani u listu.

Exchange-Rate API je third party RESTful servis koji pruža informacije o kursu. Komunikacija se odvija u JSON formatu. Korisniku je omogućeno da odabere valutu u kojoj želi da mu bude prikazana cena. Pre dobavljanja letova ESB dobavlja koeficijente kursa kojim se množe cene letova nakon dobavljanja i transformacije podataka. Pretpostavka je da sva 3 sistema avio-kompanija daju letova u drugacijoj valuti.

Još malo bih se osvrnuo na sistem za pretragu letova sa presedanjem i način na koji on pronalazi validne putanje. Od dobijenih podataka formira se graf u kom su aerodromi predstavljeni čvorovima, a letovi su predstavljani granama. Dva čvora mogu biti međusobno povezana sa više grana u slučaju da postoji više letova koji ih povezuju. Za pretragu grafa je iskorišćen BFS algoritam. Algoritam pronalazi sve validne putanje od početnog do krajnjeg aerodroma. Na primeru ovog grafa, možemo zamisliti da putujemo iz Nju Jorka do Beograda. Iako uočavamo direktan let, možemo uočiti i neke putanje koje nisu direktne, na primer Nju Jork – Luton – Beograd.

Što se tiče same intergacije i ESB-a, ESB je implementiran uz pomoć alata Mulesoft Anypoint Studio. Integracija se postiže tokovima operacija u okviru kojih se prenosi i transformiše porukasa podacima. Na slici je dat primer jednog od 30 tokova, ovaj je konkretno vezan za pretragu letova. Tokovi mogu pozivati ostale tokove, tako se u ovom toku pozivaju drugi tokovi na 6 mesta, a i sam tok je pozvan nakon parsiranja klijentskog zahteva. Tokovi su sastavljeni od komponenti koje imaju razlicitu funkciju, pa tako postoje komponente za transformaciju poruka, grananje, logiku uslova, memorisanje varijabli, slanje HTTP zahteva,...

Obrada podataka se vrši uz pomoć DataWeave-a, jezika za transformaciju, filtriranje, mapiranje i kombinovanje podataka pri čemu je rad sa podacima dodatno olakšan definisanjem metapodataka (podaci o podacima) u vidu JSON i XML sema.

To bi bilo to što se tiče moje prezentacije. Hvala na pažnji!