Prva kontrolna tacka sep

Jovane,

U sklopu prve dve kontrolne tacke je bitno izraditi sam koncentrator placanja, tako da se on na dalje može posmatrati kao blackbox u koji ce se lako ubaciti pravi servisi za placanje i razni prodavci poput onih koji se nalaze u naucnoj centrali.

Treba razmisliti o tome kako ce se koncentrator placanja koristiti, kako ce se vršiti njegova integracija sa jedne i druge strane, koje podatke KP mora da ima kod sebe i sa kakvom logikom da ih obraduje. S' obzirom da je koncentrator placanja, razumno je da ce voditi evidenciju o svim platnim podacima svojih klijenata - transakcije (no ne nužno njihov sadržaj), tajne i sl.

Iz ovoga možemo ekstrahovati njegove funkcionalne zahteve koji ce diktirati arhitekturu. Medutim, sama arhitektura je dosta složena zbog svojih nefunkcionalnih zahteva. Potrebno je istražiti šta high availability predstavlja i na koje nacine se postiže. Potrebno je istražiti kakvu bezbednost treba podržati kako bi bili PCI DSS-compliant i kako bi uverili sve korisnike našeg KPa da imamo adekvatnu višeslojnu odbranu. Potrebno je razumeti na koji nacin cemo osposobiti sistem da prihvati nove servise za placanje i nove prodavce dok trci, odnosno bez gašenja.

Vaša 1. kontrolna tacka ima smisla da budu proof of concept rešenja za podršku ovih nefunkcionalnih zahteva. Što bolje to uradite, to ce vam lakše biti da posle kontrolne tacke sklopiti sve to zajedno u kompletan KP. Dakle, od vas se ocekuje da mnogo istražujete i ucite, kako bi razumeli na koji nacin da ispunite ove zahteve, i onda implementirate kod koji ce pokazati da ste dobro razumeli sve.

S' obzirom da ste na master (of science) studijama, akcenat je na istraživanju high-level ciljeva koje treba postici, njihovo razlaganje na jednostavnije zahteve i definisanje arhitekture i dizajna spram toga. Samo programiranje je poslednji trivijalan korak koji ste svakako savladali kroz proteklih nekoliko godina. Iako je programiranje jednostavno ako se prethodni koraci urade dobro, bice nemoguce teško ako se ne urade kako treba. Zbog toga treba ozbiljno shvatiti posao, jer je ovo najizazovniji projekat koji ste do sada imali (što je prikladno pošto svakako i najviše znate do sada).

Proof of concept znaci studija izvodljivosti, što podrazumeva realizaciju, odnosno implementaciju elementarne funkcionalnosti - dakle treba da podignete sistem i sa nekom "glupavom" logikom pokažete da komunikacija od endpointa gde ce se NC zakaciti do endpointa gde ce biti servis za placanje radi. Kroz ovo cete zapravo iskonfigurisati visoku dostupnost i ugraditi bezbednost na nivou infrastrukture (npr. šifrovanje komunikacije).

Za 1. KT se pocinje sa istraživanjem - citate PCI DSS, istražujete visoku dostupnost, proucavate mikroservise i razmišljate o integracijama. Treba da kompletirate to istraživanje, da pre svega znate koji deo PCI DSSa jeste relevantan za vas i kako cete adresirati svaki od zahteva iz njega (ako ne i njihova implementacija). Za visoku dostupnost i live upgrade treba da imate jasnu sliku kako ce funkcionisati i test koji ce pokazati ispravnost ideje. Proof of concept je dokaz

Prva kontrolna tacka sep

da dizajn koji je izašao iz vašeg istraživanja ima smisla:

Za osnovnu funkcionalnost, potrebno je podici KP mikroservise i 2 "glupava" servisa (jedan glumi NC, drugi glumi neki servis za placanje, oba imaju po 1 funkciju), te da pokažete da komunikacija može da ide od prve tacke, kroz vašu mikroservisnu arhitekturu, sve do druge tacke.

Za bezbednost, da su ugradene bezbednosne kontrole koje pokrivaju do tada formiranu funkcionalnost (npr. šifrovana komunikacija, ACL-ovi)

Za integracije, treba formulisati integraciju sa obe strane (NC i ostalih prodavaca sa jedne strane, a servisa za placanje sa druge) i mehanizam kako cete raditi live uvodenje novih entiteta u sistem.

PoC je daleko od kompletnog rešenja, ali je konkretniji od samog dizajna i njegova svrha je da pokaže da to što ste smislili radi posao makar za trivijalan slucaj.