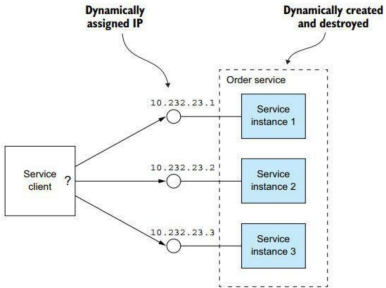
Service discovery, API gateway

Servisno orijentisane arhitekture

Univerzitet u Novom Sadu
Fakultet tehničkih nauka

Service discovery - Problem i razlog postojanja

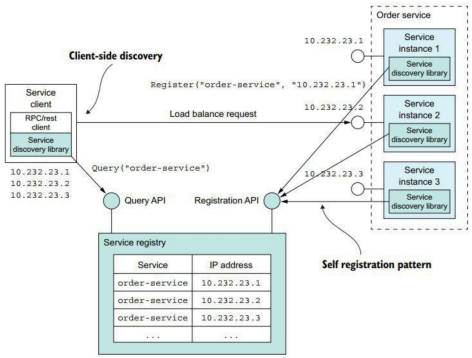
- Monolitna arhitektura podržava samo skaliranje sistema u celosti, pokretanjem više instanci monolita
- Kod mikroservisne aplikacije, svaki servis možemo skalirati nezavisno, prema trenutnim potrebama
- Javlja se problem adresiranja velikog broja servisa, gde svaki može biti pokrenut u više instanci



- Naivni pristup jeste da u nekom konfiguracioom fajlu beležimo adrese svih instanci svih servisa, što je nemoguće održavati
- Bolje rešenje treba da uključuje neki vid automazizacije procesa registracije servisa
- Servisi se mogu registrovati sami prilikom pokretanja ili ih neko drugi može registrovati
- Adresa servisa javlja se centralnom registru
- Registru može da se pošalje upit koji će vratiti adrese svih instanci nekog servisa (ili samo jednu, ako registar obavlja i load balancing)

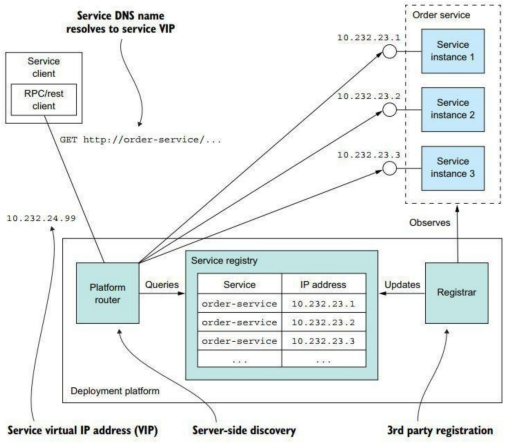
Application-Level Service Discovery

- Klijent i server direktno interaguju sa registrom
- Ne zavisimo od deployment platforme, ali smo često usko vezani za neki framework
- Često je potrebno dodatno konfigurisati servise i dodavati im kod vezan za service discovery
- Dodatna komponenta u sistemu koju moramo održavati i učiniti visoko dostupnom



Platform-Level Service Discovery

- Obavlja se od strane deployment platforme kaošto su Docker i Kubernetes
- U potpunosti mogu da odrade service discovery i rutiranje, ne održavamo dodatne komponente
- Klijent se ne obraća direktno registru, već ruteru koji dalje obavlja posao za njega
- Instance ne moraju same da se registruju, već to može obaviti registrator (3rd party registration)
- Svi servisi koje adresiramo moraju biti deploy-ovani pomoću platforme na koju se oslanjamo za discovery

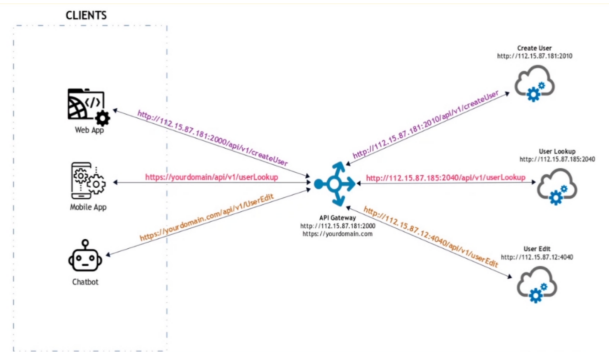


API gateway - Problem

- Neki od problema vezani za mikroservisnu arhitekturu
  - Klijent mora sadržati logiku za otkrivanje i adresiranje svakog servisa koji treba da kontaktira
  - Dekompozicija servisa se komplikuje zbog toga što klijent direktno zavisi od unutrašnje organizacije sistema
  - Protokoli koje servisi koriste ne odgovaraju uvek klijentima
  - . . .
- API gateway je komponenta koja se dodaje u sistem kako bi se rešili navedeni problemi

API gateway - Rešenje

- API qateway je servis koji predstavlja jedinstvenu ulaznu tačku u sistem, enkapsulira unutrašnju arhitekturu i nudi API koji klijenti mogu koristiti
- Neke od osnovnih funkcija:
  - Rutiranje pristiglih zahteva
  - Translacija protokola
  - API composition
  - Logovanje
  - Autorizacija
  - . . .



NGINX

- NGINX je open-souce web server, load balancer i reverse proxy, a mi ćemo ga koristit kao API gateway
- Način rada modula definiše se kroz direktive navedene u konfiguracionim fajlovima
- Direktive se dele na jednostavne i blok direktive, koje grupišu više povezanih direktiva
- Default podešavanja nalaze se u /etc/nginx/nginx.conf fajlu
- Kako bi upravljanje konfiguracijom bilo jednostavnije, fajlovi se čuvaju u /etc/nginx/conf.d/ direktorijumu, koji je učitan include direktivom u nginx.conf fajl

Direktive

- **upstream** - Definište grupe servera koji se mogu referencirati iz drugih direktiva
- **server** - Podešavanje konfiguracije virtualnog servera
- **listen** - Postavlja adresu i port na kom će virtualni server slušati zahteve (podešava se unutar server direktive)
- **location** - Podešava konfiguraciju za zadati URI (podešava se unutar server direktive)
- **proxy_pass** - Postavlja protokol i adresu servera kom se zahtev prosleđuje (podešava se unutar location direktive)

Primer konfiguracionog fajla

- /etc/nginx/conf.d/api_gateway.conf fajl

```
upstream ordering_service {
    server ordering_service:8000;
}
...
server {

    listen 8000 default_server;

    location /api/ordering/ {
        proxy_pass http://ordering_service;
        rewrite ^/api/ordering/(.*)$ /$1 break;
    }
...
}
```

Dockerfile

- Dockerfile za kreiranje NGINX image-a

```
FROM nginx:latest

COPY ./api_gateway.conf /etc/nginx/conf.d/api_gateway.conf
RUN rm /etc/nginx/conf.d/default.con
```