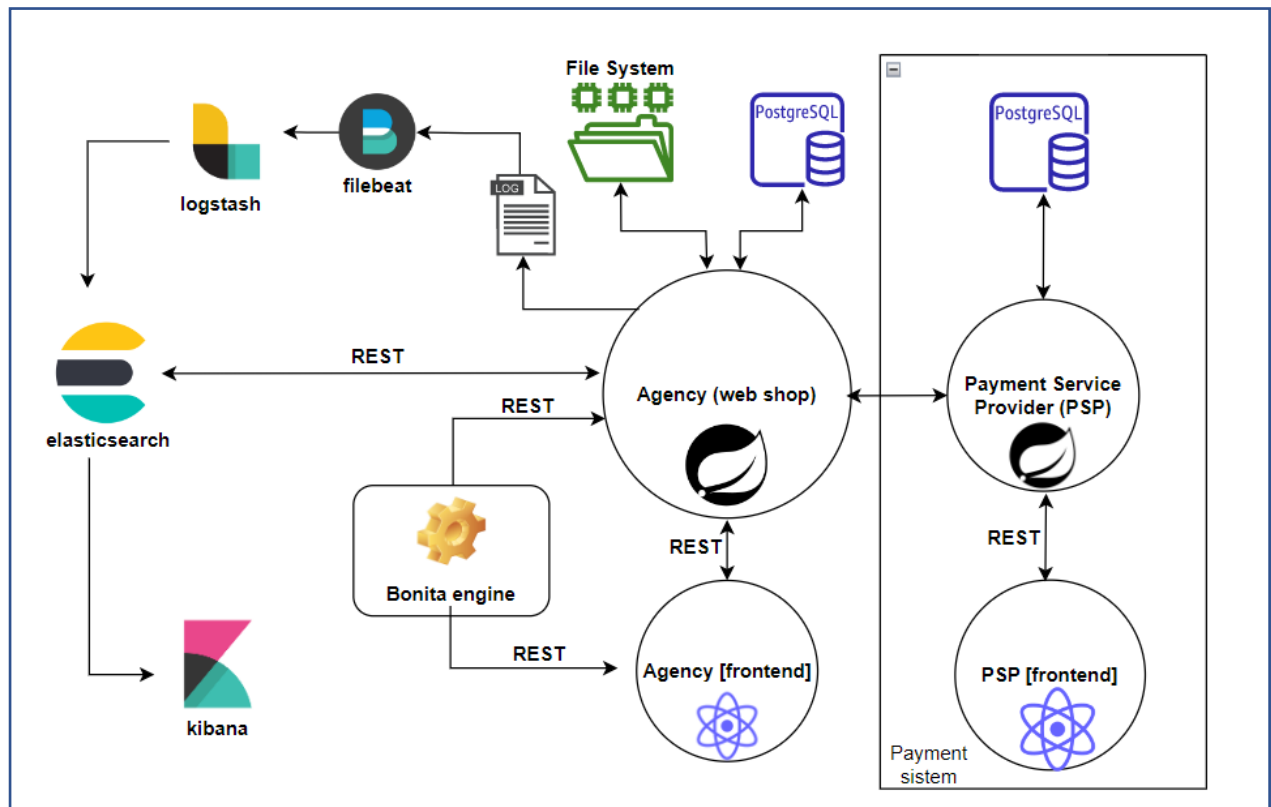


## Upravljanje digitalnim dokumentima – kontrolna tačka

### Arhitektura sistema

Na slici 1 prikazana je arhitektura sistema za podršku rada agencije za zapošljavanje.



Slika 1. Arhitektura sistema

Posmatrani sistem čine veb aplikacija agencije za zapošljavanje, Bonita *process engine* koji služi za upravljanje poslovnim procesima u okviru sistema i sistem za plaćanje. Veb aplikacija agencije za zapošljavanje ima klijent-server arhitekturu, pri čemu je klijentska strana implementirana u React biblioteci, a serverska strana u Spring Boot okruženju. Aplikacija za skladištenje podataka koristi PostgreSQL relacionu bazu podataka, lokalni fajl sistem za skladištenje CV dokumenata i motivacionih pisama, a podaci neophodni za pretragu dokumenata čuvaju se u Elasticsearch bazi, što će detaljnije naknadno biti objašnjeno. Za potrebe predmeta Upravljanje poslovnim procesima, procesi su razvijeni u Bonita okruženju koje komunicira sa veb aplikacijom agencije putem REST protokola. Prilikom registrovanja kompanije na sistem agencije, kao i prilikom procesa premium zapošljavanja, neophodno je izvršiti plaćanje. Sva plaćanja u sistemu se obavljaju preko nezavisnog platnog sistema, razvijenog za potrebe predmeta Sistemi elektronskog plaćanja. Taj sistem ima mikroservisnu arhitekturu, koja nije od velike važnosti za potrebe ovog predmeta, pa neće biti detaljno

opisana. Bitno je istaći da je centralni servis tog sistema PSP servis, putem koga se vrše sva plaćanja podržana u platnom sistemu, a koji komunicira sa aplikacijom agencije preko REST protokola.

Osim u ovom primjeru, REST komunikacija se koristi i za komunikaciju između agencije i Elasticsearch API-ja, što je predstavljeno na dijagramu arhitekture.

## Konfiguracija Elasticsearch-a

ElasticSearch predstavlja *opensource* engine za pretraživanje tekstualnih dokumenata. Napisan je u Javi, nad Lucene bibliotekom, ali ima podršku i za .NET, Python, Ruby i mnoge druge programske jezike i razvojne okvire. Odlikuju ga visoka skalabilnost i distribuiranost sistema za pretragu i skladištenje podataka. Takođe, ima jednostavan REST API sa kojim aplikacije jednostavno komuniciraju slanjem HTTP zahtjeva.

Danas je ElasticSearch najpopularniji *enterprise engine* za pretragu. Njegovoj popularnosti doprinosi i postojanje ELK steka, odnosno upotreba Logstash-a i Kibane. Logstash omogućava prikupljanje i transformaciju podataka sa više izvora, koji se zatim skladište u Elasticsearch bazi, a dalje se ti podaci vizualizuju u Kibani.

Da bi se koristio pomenuti stek, neophodno je izvršiti odgovarajuću konfiguraciju. U nastavku će biti objašnjena konfiguracija upotrebom Docker *engine*-a. Za svaki od pomenutih *engine*-a (elasticsearch, logstash, kibana) biće preuzeta Docker slika sa DockerHub-a u verziji 7.4.0, zbog kompatibilnosti sa SerbianAnalyzer-om, na osnovu kojih će biti pokrenuti docker kontejneri, sa konfiguracijom opisanom docker-compose.yml fajlom, prikazanim na listingu 1.

```
version: '3.8'

services:
  elasticsearch:
    container_name: elasticsearch
    image: docker.elastic.co/elasticsearch/elasticsearch:${ELASTIC_VERSION}
    command: >
      /bin/sh -c ".bin/elasticsearch-plugin install
      file:///serbian-analyzer/serbian-analyzer-1.0-SNAPSHOT.zip;
      /usr/local/bin/docker-entrypoint.sh"
    volumes:
      - ./serbian-analyzer:/serbian-analyzer
    ports:
      - 9200:9200
      - 9300:9300
    environment:
      node.name: agency
      discovery.type: single-node
    networks:
      - elk
```

```

logstash:
  container_name: logstash
  image: docker.elastic.co/logstash/logstash:${ELASTIC_VERSION}
  ports:
    - 5044:5044
    - 50000:50000/tcp
    - 50000:50000/udp
    - 9600:9600
  networks:
    - elk
  depends_on:
    - elasticsearch

filebeat:
  container_name: filebeat
  image: docker.elastic.co/beats/filebeat:${ELASTIC_VERSION}
  environment:
    - setup.kibana.host=kibana:5601
    - output.elasticsearch.hosts=["elasticsearch:9200"]
  networks:
    - elk
  depends_on:
    - elasticsearch

kibana:
  container_name: kibana
  image: docker.elastic.co/kibana/kibana:${ELASTIC_VERSION}
  ports:
    - 5601:5601
  environment:
    - ELASTICSEARCH_HOSTS=http://elasticsearch:9200
  networks:
    - elk
  depends_on:
    - elasticsearch

networks:
  elk:
    driver: bridge

volumes:
  elasticsearch:

```

Listing 1. Sadržaj docker-compose.yml fajla

Osim konfigurisanja Elasticsearch engine-a, neophodno je omogućiti komunikaciju aplikacije sa njim. Pošto je aplikacija razvijana u SpringBoot okruženju, konfiguracija za komunikaciju sa

ElasticSearch-om je minimalna i sve što je potrebno jeste da se u konfiguracioni fajl pom.xml doda sledeći *dependency*:

```
<!--https://mvnrepository.com/artifact/org.springframework.data/spring  
data-elasticsearch -->
```

```
<dependency>
```

```
    <groupId>org.springframework.data</groupId>
```

```
    <artifactId>spring-data-elasticsearch</artifactId>
```

```
    <version>4.2.2</version>
```

```
</dependency>
```

Dodati *dependency* donosi niz prednosti za rad sa podacima koji se čuvaju u okviru ElasticSearch-a. Slično JPA repozitorijumima, on koristi ElasticSearchRepository koji nas oslobađa pisanja upita za osnovne CRUD operacije. Međutim, nedostatak je što ne pruža veliku fleksibilnost prilikom pisanja upita za pretragu, pa će za tu svrhu biti korišten ElasticSearchTemplate.

## Skladištenje podataka

Prilikom opisa arhitekture sistema, pomenuto je da aplikacija podatke skladišti na 3 mjesta. Podaci o korisnicima sistema (kandidati, kompanije), kredencijali za platni sistem, paketi usluga koje agencija pruža, kao i svi ostali podaci bitni za rad sistema čuvaju se u PostgreSQL bazi podataka. Dokumenti, odnosno CV-evi i motivaciona pisma kandidata, čuvaju se na lokalnom fajl sistemu, pri čemu će folder u kome su oni sačuvani biti namapiran na docker *volume*. Osnovni podaci koji su neophodni za efikasnu pretragu, biće čuvani u ElasticSearch bazi, pri čemu će to biti minimalan skup podataka, opisan u narednoj jedinici Indexing Unit.

## Indexing unit

Za pretraživanje dokumenata putem ElasticSearch-a, neophodno je izvršiti indeksiranje dokumenta, odnosno odrediti koja polja će dokument da sadrži.

U specifikaciji projekta navedeno je da je potrebno omogućiti pretraživanje aplikacija za posao po imenu i prezimenu aplikanta, stepenu stručne spreme, kao i sadržaju CV-a i priloženog pisma. Osim toga, za geoprostornu pretragu neophodno je čuvati podatke o adresi (naziv grada, geografsku dužinu i geografsku širinu), o čemu će više riječi biti u dijelu o geoprostornoj pretrazi. Iz tog razloga će indeksiranje aplikacija u ElasticSearch-u biti implementirano na način predstavljen listingom 1.

```
{
  "id": "applicantId",
  "name": "applicantName",
  "surname": "applicantSurname",
  "educationLevel": "applicantEducationLevel",
  "cv": "textualContentOfApplicantCV",
  "motivationalLetter": "textualContentOfApplicantMotivationalLetter",
  "address": {
    "city": "cityName",
    "location": {
      "lat": "lat",
      "lon": "lon"
    }
  }
}
```

Listing 2. JSON struktura index unit-a

## Konfiguracija SerbianAnalyzer-a

Sa repozitorijuma <https://github.com/chenejac/udd06> preuzet je source kod elasticsearch *plugin*-a za pretprocesiranje teksta na srpskom jeziku. U okviru repozitorijuma, nalazi se README.md fajl sa uputstvima za *build*-ovanje *plugin*-a. Kao *build* alat korišten je gradle, pa je build izvršen pozicioniranjem u *root* folder projekta komandom `./gradlew clean build`. Za uspješan *build* neophodno je podesiti JAVA\_PATH kao *environment variable*, a kao verziju Java koristiti verziju 13. Postupak *build*-ovanja prikazan je na slici 2.

```
PS C:\Users\hp\Desktop\UDD\udd06> ./gradlew clean build
Starting a Gradle Daemon, 3 incompatible and 1 stopped Daemons could not be reused, use --status for details

> Task :printGlobalBuildInfo UP-TO-DATE
=====
Elasticsearch Build Hamster says Hello!
Gradle Version      : 6.0
OS Info             : Windows 10 10.0 (amd64)
JDK Version         : 13 (Oracle Corporation 13.0.2 [Java HotSpot(TM) 64-Bit Server VM 13.0.2+8])
JAVA_HOME           : C:\Program Files\Java\jdk-13.0.2
See https://docs.gradle.org/6.0/userguide/command\_line\_interface.html#sec:command\_line\_warnings

BUILD SUCCESSFUL in 2m 32s
31 actionable tasks: 29 executed, 2 up-to-date
```

Slika 2. Postupak build-ovanja SerbianAnalyzer-a

Rezultat *build*-a je `serbian-analyzer-1.0-SNAPSHOT.zip` fajl u folderu `./build/distributions`. Nakon toga, neophodno je pomenuti zip ubaciti u *root* folder projekta kako bi se mogao pokrenuti pri podizanju elasticsearch kontejnera, u okviru docker compose-a, komandom `./elasticsearch-plugin install file: <apsolutna putanja do zip fajla>`.

## Geoprostorna pretraga

Po specifikaciji projekta, neophodno je obezbjediti geoprostornu pretragu. Ovakav vid pretrage je podržan u Elasticsearch-u, ali je neophodno prilagoditi indeksiranje dokumenata.

Prilikom registracije na aplikaciju agencije, korisnik mora da unese svoju adresu. Na osnovu unijete adrese, uz pomoć PositionStack-a i GeoCode-a, biće izvučene geografska širina i geografska dužina, koje će biti sačuvane u bazu podataka zajedno sa svim ostalim podacima o korisniku. Takođe, index unit će biti proširen tako da pored osnovnih podataka o aplikantu sadrži i podatke o nazivu grada i geografskim koordinatama adrese, što je prethodno predstavljeno na listingu 2.

Prilikom geoprostorne pretrage, šalje se upit čiji su parametri radijus i naziv grada. Na osnovu naziva grada, pomoću PositionStack-a, biće izvučene geografske koordinate grada i na osnovu njih formiran upit za pretragu svih aplikanata u okviru zadatog radijusa.

Različiti primjeri formiranja upita za pretragu, kao i primjeri mapiranja i indeksiranja polja koja predstavljaju lokaciju, mogu se pronaći u dokumentaciji na linku [Geo-distance query | Elasticsearch Guide \[8.5\] | Elastic](#). Na listingu 3 nalazi se primjer jednog upita koji vrši pretragu u radijusu od 200km, na osnovu zadatih geografskih koordinata.

```
GET /my_locations/_search
{
  "query": {
    "bool": {
      "must": {
        "match_all": {}
      },
      "filter": {
        "geo_distance": {
          "distance": "200km",
          "pin.location": {
            "lat": 40,
            "lon": -70
          }
        }
      }
    }
  }
}
```

Listing 3. Primjer geoprostorne pretrage