**X-Service Transaction**

**Table of Contents**:

* Description
* Microservice architecture big picture
* Technology stack

**Description**

The application has 2 main microservices: ModelService and StatisticsService.

The model service is a REST Spring Boot microservice and has a connection to a PostgreSQL database for storing User. Service has a repository layer based on Spring Data JPA.

The statistics service built on Spring web HTTP. The service calculates satistics of all adult User (age >= 21).

The model service and the statistics service have Hystrix. Hystrix is the implementation of Circuit Breaker pattern, which gives a control over latency and failure from dependencies accessed over the network. The main idea is to stop cascading failures in a distributed environment with a large number of microservices. That helps to fail fast and recover as soon as possible - important aspects of fault-tolerant systems that self-heal.

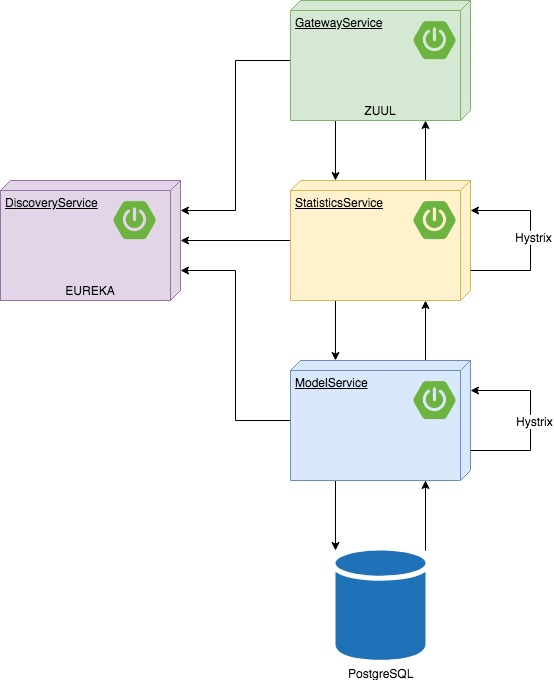
There are 2 util services: GatewayService and DiscoveryService.

The gateway service exposes external API to a client. It is a single entry point into the system, used to handle requests by routing them to the appropriate backend service or by invoking multiple backend services and aggregating the results. The gateway service based on ZUUL.

The discovery service allows automatic detection of network locations for service instances, which could have dynamically assigned addresses because of auto-scaling, failures and upgrades. The key part of Service discovery is Registry. Netflix Eureka is a good example of the client-side discovery pattern, when client is responsible for determining locations of available service instances (and load balancing requests across them.

Each microservice based on Spring Boot.

**Microservice architecture big picture**



**Technology stack**

* Spring Boot
* Spring Data JPA
* Spring Coud
* Zuul
* Eureka
* PostgreSQL
* Flyway
* Hystrix

For good CI and CD there is a proposition to use Docker and Docker compose and CI/CD tool like Gitlab CI.