Textual Documentation

Application 2

This is an application showcasing the microservice architecture and its capabilities. It implements a simple use-case where one microservice manages registered users and a second one holds detailed information about them.

Microservices

The application's business logic is implemented in two *internal* services:

- *account-service*
- *customer-service*

There are three additional *infrastructural* services which implement common functionality found in most microservice applications:

- *auth-server* – OAuth2 authorization server realizing access control decisions; has a database connected to it to store access tokens; implemented with Spring OAuth 2.0
- *gateway-server* – API gateway that proxies all external calls to the microservices; implemented with Zuul
- *discovery-server* – service discovery server; implemented with Eureka

Structure

The user interacts with the API Gateway, which serves as entry point to the system. The Gateway performs load balancing to distribute calls across all running instances of a microservice. It further has a built-in circuit breaker, a common security mechanism in microservice applications to prevent cascading failures.

In order for the gateway to know, which service instances are up and running, all instances register with the service-discovery server *discovery-server*. When a request for an internal service arrives at the *gatewy-server*, the gateway will query the *discovery-server* to retrieve a list of running instances of that service. It will then forward the request to one of them, according to its load-balancing scheme.

Before requests are forwarded, however, they need to be authenticated. For this, the client is referred to the authorization server *auth-server,* where it authenticates itself and receives an access token in return. The access token is sent along the query to the business logic service, where it is validated by again querying the authorization server. If everything is successful, the business logic service handles the request from the client.

The business logic is implemented in the internal microservices *account-service* and *customer-service*. The customer-service manages all users that are registered with the application. The account-service hold further details about the registered users. The customer-service queries the account-service to retrieve this information. For that, it uses a *FeignClient*, which is a technology for direct inter-service communication without the gateway as proxy. Only the customer-service is reachable via the API gateway.