Textual Documentation

Application 1

This is an application showcasing the microservice architecture and its capabilities. It implements a simple use-case where users can create tasks and add comments to them.

Microservices

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

- *comments-webservice*
- *user-webservice*
- *task-webservice*

There are six 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
- *api-gateway* – API gateway that proxies all external calls to the microservices; implemented with Zuul
- *configserver* – central server where configuration information is stored; implemented with Spring Cloud Config
- *webservice-registry* – service discovery server; implemented with Eureka
- *zipkin-tracing* – Single Page Application that shows tracing information about the connected services; implemented with Zipkin

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 *webservice-registry*. When a request for an internal service arrives at the *api-gateway*, the gateway will query the *webservice-registry* 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 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 *user-webservice*, *task-webservice*, and *comments-webservice*. The user-service manages all users that are registered with the application. The tasks-service allows creating new tasks and retrieving information about existing ones. The comments-service provides comments that are associated with each task. The task-service queries the comments-service by formulating a REST request (using Spring's *RestTemplate*) to the comment-service's address. Only the task-service and the user-service are reachable via the API gateway.

All services receive their configuration information from the central configuration service *configserver*, which in turn retrieves it from an external GitHub repository (https://github.com/anilallewar/microservices-basics-cloud-config).