TourGuide

Requirements & Design

TourGuide

Requirements & Design

1. Project Overview

Describe the solution being proposed. Include the target audience and the business advantages of the solution.

Tour Guide is an application where target audience is tourists. The application allows users to get package deals on hotel stays, find nearby tourist attractions and get admission to them.

1.1 Project Goals

Describe the goals of the project (2 to 3 sentences), including the problem(s) being solved.

The project's objective is to improve application performance, support at least 100,000 users per day and fix multiple bugs that have been reported. Slitting the TourGuide app into multiple services and deploying different Docker containers to reduce host monthly costs.

1.2 Out of Scope

Which objectives have been considered but are not covered by this project?

- ✓ Adding a security framework to provide both authentication and authorization to regulate roles and access to the application.
- ✓ The "getAllUsers” feature is designed for an administrator role.

1.3 Project Metrics

How will you measure success?

- ✓ Ensure that the tests have been written sufficiently to cover all the needs of the application.
- ✓ Check that the updated app version got the incrested performance to meet the requirements stated above.

2. Features

Include a list of features, user stories, and acceptance criteria here.

- ✓ Improved the performance of gps and rewardws services.
- ✓ List of 5 attractions closest to the user based on the his/her geolocation with reward points for each attraction.
- ✓ Management of user preferences.
- ✓ Get a list of every user's most recent location

1. User story : As the user, I want to be able to geolocate myself in the shortest period.

   Acceptance tests : When the user clicks on the “Find my location” button then the system shows the longitude and latitude of user geolocation.

2. User story : As the user, based on my location, I want to be able to find 5 nearby tourist attractions in the shortest period.

   Acceptance tests : When the user clicks on the “Find nearby attractions” button, a system based on the user’s location retrieves 5 attractions’ GPS coordinates and information about them and displays them to the user in the shortest period.

3. User story : As the user, I want to set my trip preferences to get appropriate trip offers.

   Acceptance tests : The user wants to enter their preferences: user name, trip duration, ticket quantity, number of adults, number of children and get choices of suppliers.

4. User story : As the user, I want to set my trip preferences to get appropriate trip offers.

   Acceptance tests : The user wants to enter their preferences: user name, trip duration, ticket quantity, number of adults, number of children and get choices of suppliers.

5. User story : As admin, I want to locate all users at once.

   Acceptance tests: When admin clicks on a “View all users” button, then the system displays a list of all users and their location.

3. Proposed Solution

The application is divided into four microservices for the following tasks:

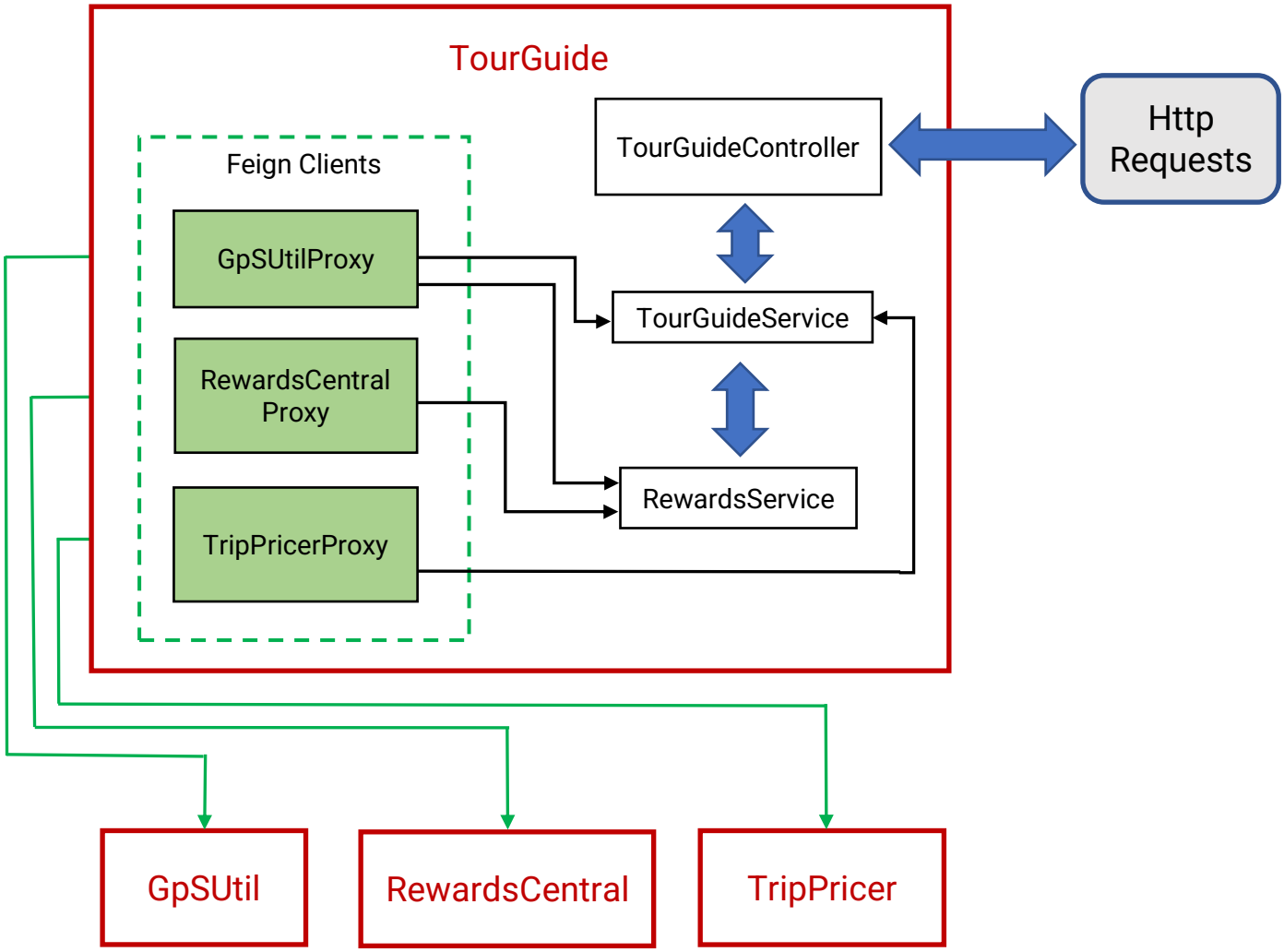
TourGuide - manages user data and sends requests to the other microservices

GpSUtil - manages location and attraction data

RewardsCentral – calculates user and attraction reward points

TripPricer – manages trip deals

3.1 Technical Design Diagrams



3.2 Glossary

Any domain vocabulary goes here.

Attraction: Represents a tourist attraction
User: Represents a user who can use the application
UserReward: Represents user rewards
UserPreferences: Represents user preferences
Location: Represents the location of the user
VisitedLocation: Represents the places visited by the user

3.3 Technical Specifications

Explain which technologies and languages can be used as a solution.

Java as a development language
Spring Boot framework.
Spring Cloud: openfeign.
Docker.

3.4 Alternative Solutions

Explain alternative directions for the solution and why these were not chosen.

The RestTemplate could be used, but openfeign has been chosen because it's easier to implement.

An alternative solution, to solve the performance issues could be more powerful hardware which could also lead to significant expenses, this was not the right approach for a problem that was solved by improving the code.