

Smart Service Development - Exercise 3

Wagner, Lukas
01430206

Kissich, Meinhard
11771817

Weinbauer, Klaus
11721179

May 19, 2021

1 Scoping and Structuring

1.1 Motivation

Fast short-range goods delivery seems to be a fast-growing market with numerous examples out there. As in many cases, Amazon is one famous example. It is now possible to receive some goods within just a couple of hours after ordering when living in a large city¹. But it does not end with Amazon. Since the corona crisis forced governments worldwide to close restaurants, food delivery got a boost in popularity². Still it does not end with food. Also, the ambulance can be seen as some short-range, fast delivery service - with some special treatment, of course. However, all these examples combine the same key features: (mostly) short-range, short job duration, source and destination of some matter, (extremely) fast scheduling, many workers ("agents") that do the transportation, optimization of resources, etc. Now, we think that we can provide a service that can do better than others. (Or at least, we want to have a piece of the cake). The provided service aims to give an interface for such transportation jobs, schedules the right resource to the right job for various optimization strategies, and monitors the open missions after assigning the resource. Besides, we also serve the agents on the job with great features - the selection of suggested jobs, automatic route planning, and many more. The main advantage is that the provided service is extremely versatile over many sectors and applications within the same API.

¹<https://tinyurl.com/9sfuh3dc>

²<https://tinyurl.com/4xbzz96d>

1.2 Main Components

Our current plan is to split our service into four micro-services:

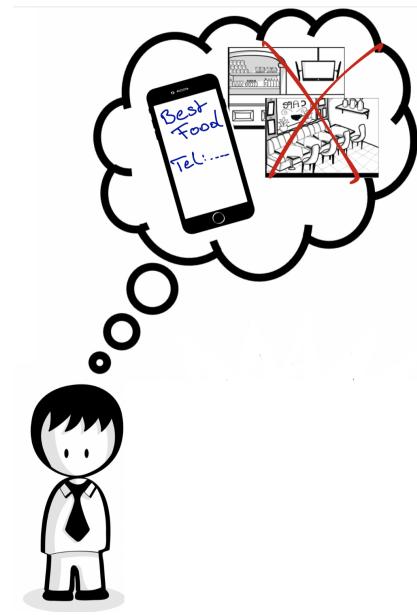
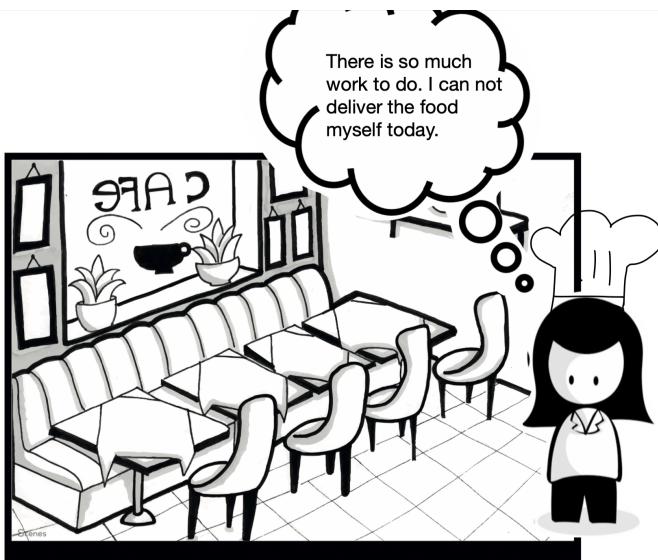
- **Authentication service.** This service is used to manage users and their roles (either provider, agent or admin). It is responsible for handling user logins and also for login token verification for other services.
- **Job service.** This service is used by our providers to create, update and delete jobs. Also, providers can query the state of a job they have created (open/assigned/done). Agents can use this service to ask for a list of jobs that they can take, accept a job and mark jobs as done. End-users can use this service to track their order.
- **Agent service.** This service is used by the agents to update their status (position, availability, etc.). The last known state of an agent is stored in a database. providers can query information such as the current number of available agents, proximity of the next available agent, and also the position of an agent if and only if that agent is currently executing a job created by this particular provider.

The motivation behind splitting this service from the Job service are mainly separation of concerns and scalability. In general, there are much more agents than there are providers. Also, agents will update their state quite frequently due to position updates. Thus, the split between job service and agent service allows us to spawn a large number of agent service instances (horizontal scaling) to deal with the traffic.

- **Geocoding.** The (reduced) geocoding service is used as a wrapper for an externally provided service by a 3rd party to translate addresses into coordinates. In addition it provides caching to reduce the traffic to our partners.

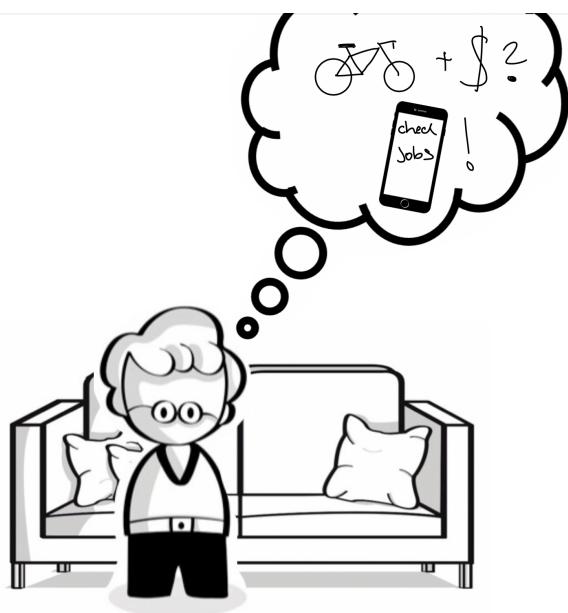
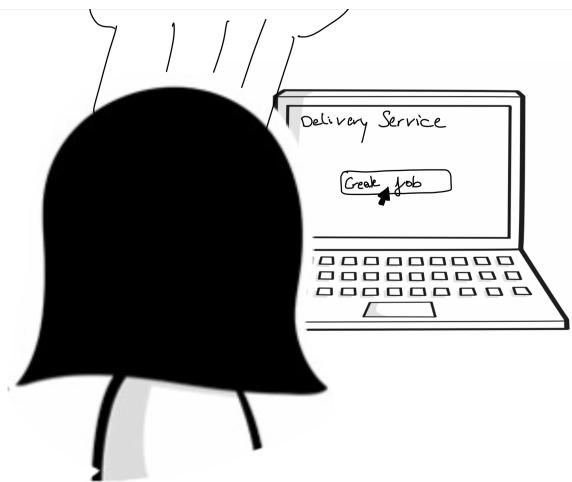
1.3 Interaction

John was busy with meetings all day long. So he wants to get some delicious food without bothering with cooking tonight. As Covid-regulations doesn't permit going into a restaurant, John orders a meal at his favorite bar.



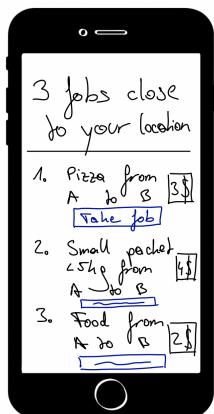
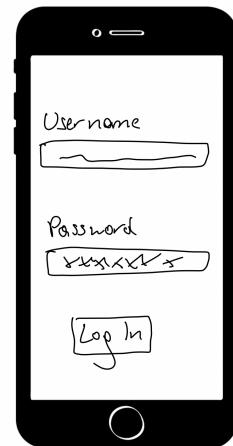
Silvia, the bar chef, is already stressed. Since everyone orders the meals online, she doesn't have enough stuff to deliver the food at peak times.

But she has a plan. Some weeks ago, she discovered a new online service for short-range delivery. Best time to find out how it works.



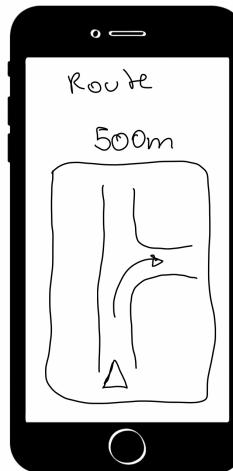
Meanwhile, Michael wants to have a ride on his bike. He speculates about whether there is a way to earn some money while exercising.

There is one! So he takes his mobile phone and log-in to the app to fetch some pending jobs.



The app suggest a couple of different jobs to choose from.

While picking up the goods and bringing them to the desired location, Michael is guided by the agent app, which shows pick-up and drop-off location on a map. Also Silvia can query the live status of the job as Michael's app periodically pushes an update. The app developer can optionally implement a full route planner based on these location points.



2 Services and their Interfaces

Figure 1 depicts the components used to implement the provided functionality.

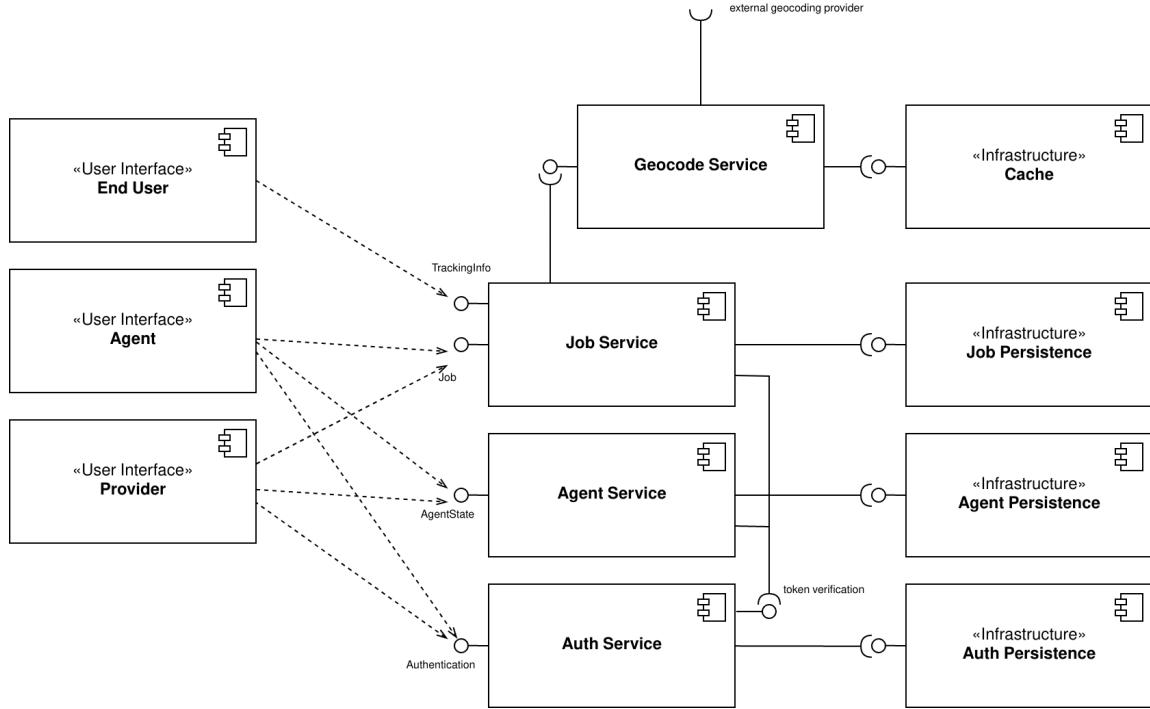


Figure 1: UML component diagram of services..

2.1 Authentication Service

Users need certain privileges (roles) to trigger some actions. This service handles the authentication by a REST API.

2.2 Job Service

Using this service, the provider can create a job and monitor its status via a REST API.

2.3 Agent Service

Agents need to be informed about pending jobs. Once a job is taken, a frequent update of status and location shall be published. This happens with the agent service. A REST API exposes it.

2.4 Geocoding Service

Provides a REST API to query coordinates for a given address. Serves as a proxy to an external provider, providing an abstraction layer and additional caching.

3 API

The API can be found at <https://app.swaggerhub.com/apis/lwagner94/DeliverySystem/1.0.0/> and also in the submitted YAML file.