

Welbo assesment

Juan Buhagiar

April 14, 2019

1 Introduction

This document describes the design decision made for the implemented system described in Task 1 and also a proposed system to answer the question of Task 2. The task was completed in approximately 4 hours.

2 Task 1

Figure 1 shows the proposed architecture for a robot checking-in guests to an office. In the developed version the Auth/API modules were not developed and therefore there is no connection with the internet. This was replaced by the Appointment File System described in the task. Nonetheless these modules would be essential to answer questions in Task2. A description of the most essential modules is attached below.

After the main flow of the chat was developed, I focused on the extraction of names from the chat. This has been achieved by a two-step operation, the first step is to pass the response containing the name through a pretrained Named-entity recognition model which outputs person names found. The second step involves a matcher which tries to find names from the appointment list in the query.

The main modules used by the system are described below:

2.1 Parser

This modules consumes raw text input from the user, the parsers task is the extract the intent of the user by using NLP techniques.

2.2 Response Generator

The response generator keeps track of the conversation, a makes decisions on what to say.

2.3 Appointment File System

This modules is used in communicating with the 3rd party systems. In our case this would save checkins and load appointments. This would be replaced with an AUTH and API modules in real world scenario.

3 Task 2

This part of the assignment is dedicated to answer three questions about developing a system with one application for different client requirements, authentication credentials and 3rd party servers.

The approach used in this assignment was a modular one, this is due to the strict requirement that the same application is used for all client with different requirements yet a similar process. The main process consists of the modules described in Figure 1, these can be implemented as interfaces in an OOP language. These interfaces would then be implemented for different client requirements such as having different Response Generators and API modules depending on the client needs and technology used. Apart from fitting the requirements, this design could allow for the use of micro-services.

To provide for a more secure infrastructure, it would be ideal to set up a token service which requests an authentication token (be it user/pass or a key) and returns a token to be sent with any API requests made to client servers. This can be seen in Figure 1.

Finally, to keep 3rd party services/servers in check, an idea would be develop an integration test pipeline on a server which runs periodically. The developers can develop integration tests that would verify that request made by the robot would be successfully handled by the 3rd party services/servers.

4 Future Improvements

If more time was available for this assignment, there are some things I would have done differently. As described above, the main focus of the assignment was given to the extraction of names from the users response. This could be further extended as a three step process which extends the above-mentioned pipeline to include a rule based matcher. The rule based matcher would include grammatical rules which are used to extract names from the input. Secondly, more effort would be given to things such as different greetings, different verification phrases etc.

On a more high level it, as seen in Figure 1 A logging server would be introduced to be able to collect statistics as to how the robots perform in conversations.

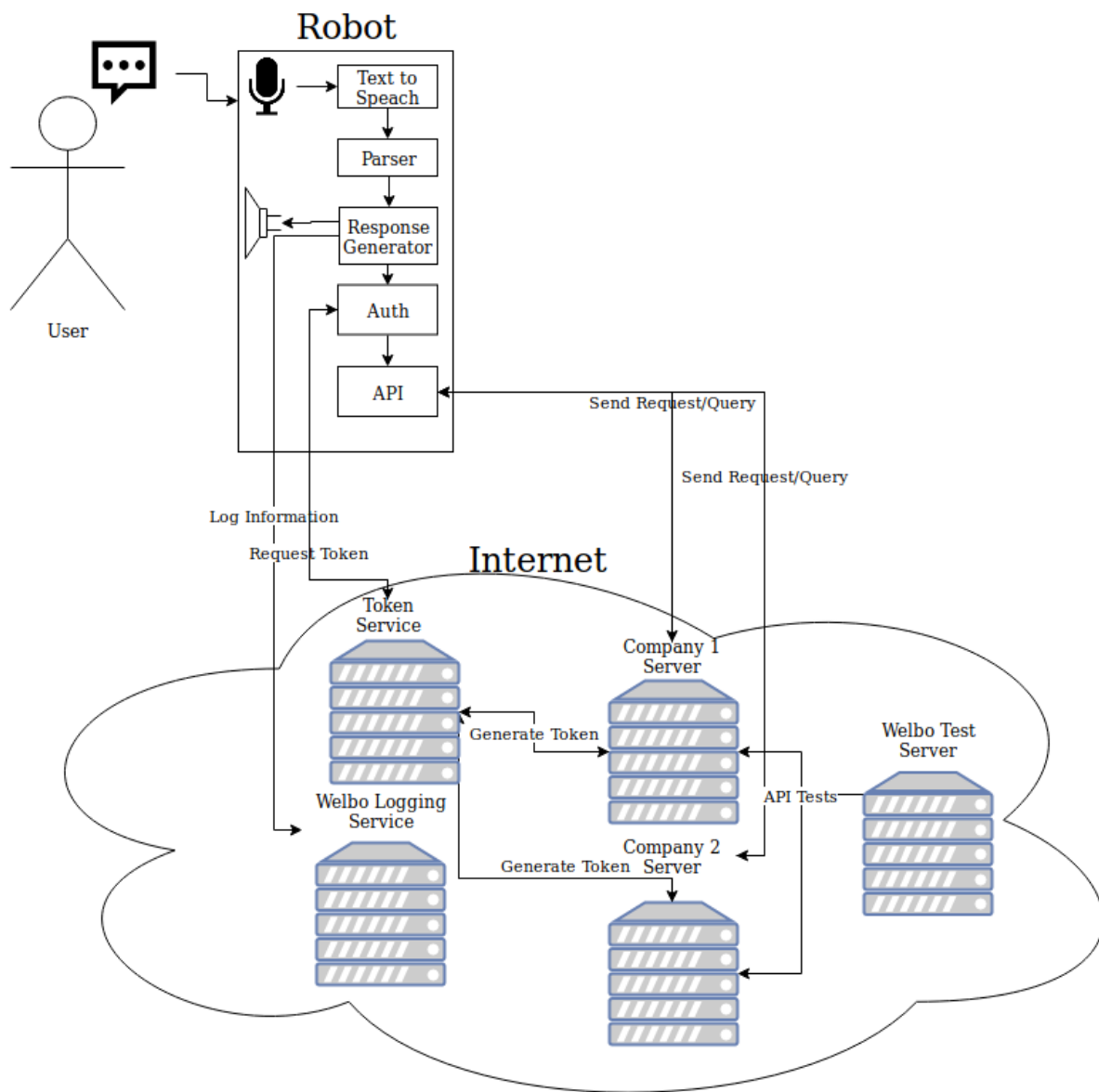


Figure 1: System architecture