Multi-Channel Bot/Agent platform
Architecture Document

Version <1.0>

Revision History

Date	Version	Description
<22/12/2018>	<1.0>	First Draft of the Software Architecture Doc.

Table of Contents

Multi-Channel Bot/Agent Architecture Document

1. Introduction

This introduction provides an overview of the entire *Software Architecture Document* for the Multi-Channel Bot/Agent. It includes a simplified overview of the system mentioned in the assignment document.

1.1 Purpose

This document provides an architectural overview of the Multi-Channel Bot/Agent system.

This document is intended to capture and convey the high-level architectural decisions which have been made in designing and building the system. It includes the explanation of other attached documents as well such as (Swagger API files & Architectural diagram)

1.2 Overview

This deliverable consists of 3 main files, which are described below:

- Software Architecture Document.doc:
  Contains details and steps passed by, during creation of the architecture, selecting components and how they communicate.

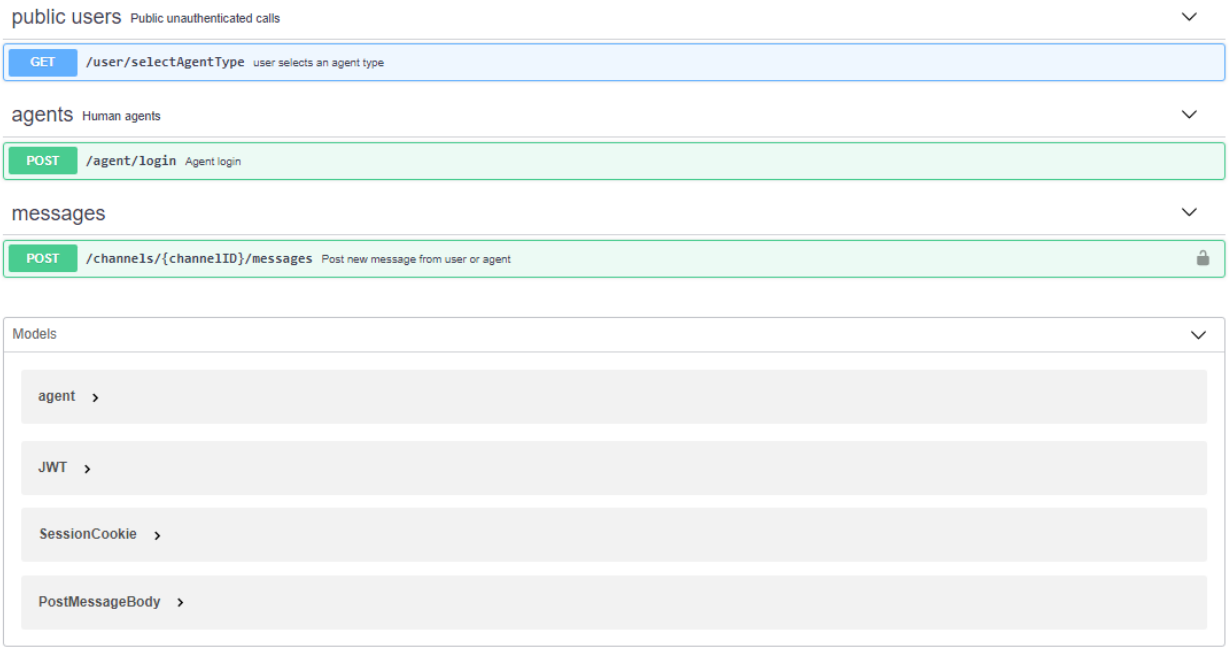
- Chat platform Architecture.pdf:
  Contains high level microservice architecture diagram, as well as the communication flow between the different system components.

- swagger.zip:
  Sample API in Swagger format

2. Architectural Representation

2.1 Architectural Views

A sample of the API is as follows:



More usability details and requirements can be checked though the swagger file, the function of each endpoint is as follows:

SelectAgentType:
This endpoint lets the user select the type of chat to be initiated, whether it's with a human agent or a bot, it doesn't require previous authorization, as it's used by public users.

Agent Login:
This call is used to authorize human agents before they can use the platform, The credentials of the agent needs to be passed in the request body as shown in the swagger example. A JWT token is returned for usage as an authentication token for future requests.

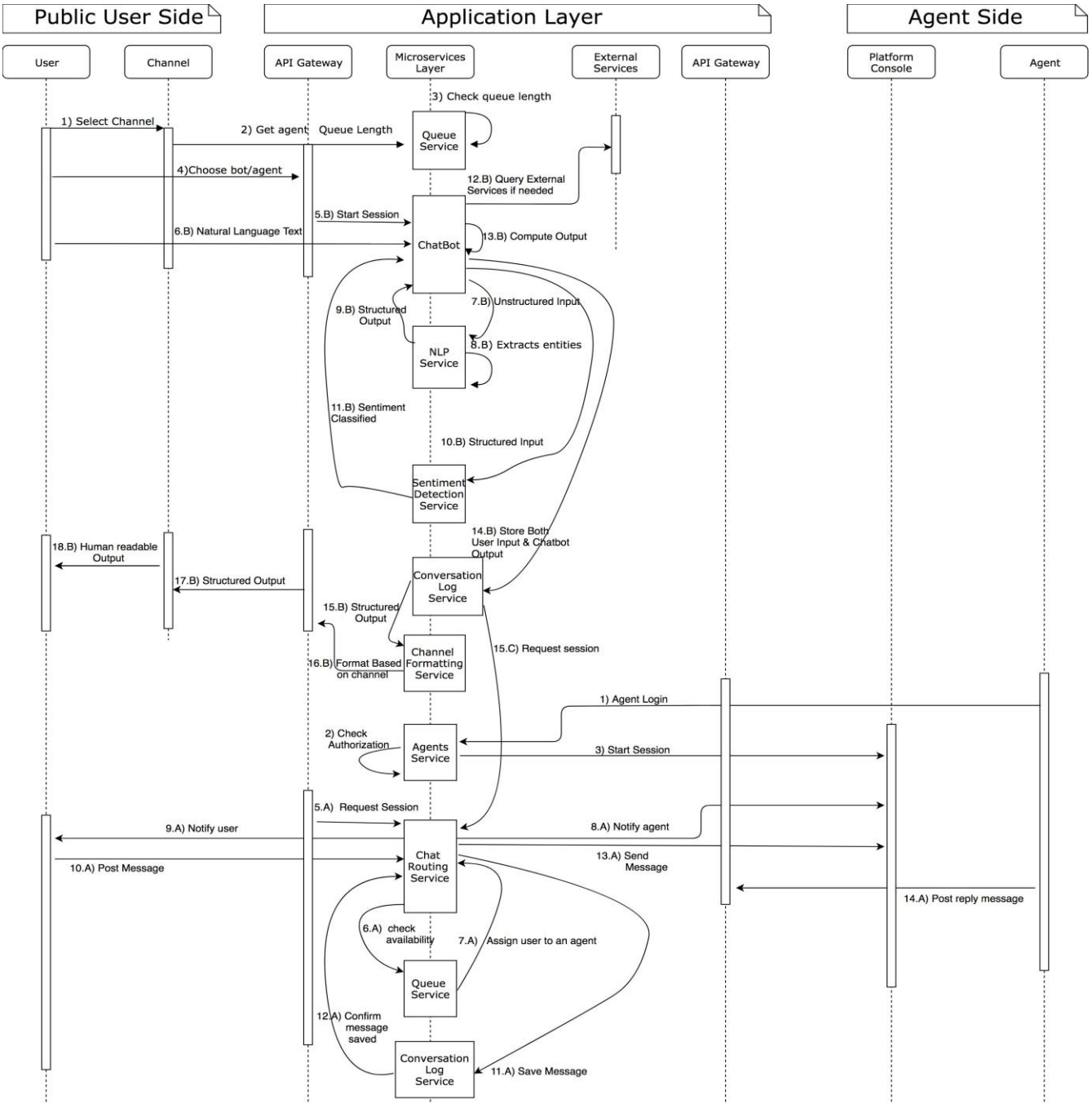
Post message:
This endpoint is used by both agents and public users to post message to the other party, the channel ID and user authentication token (Cookies for pubic users and JWT for agents) are both required.

Models for data representation in requests and responses are included as well.

Please note that this is only a sample of the API calls to be implemented in the system.
An API gateway pattern can act as the single point of entry for such system, the same gateway would be used by both agents and users. In the following Architectural / Communication Flow diagram, Two API gateways are shown, they are the same gateway, but placed twice for easier illustration of agents and users flows.

2.2 Architectural Design Patterns

The Micro-services architecture pattern is used to design the system. The following diagram shows a high-level view of both architecture and communication flow between different system components.



A clearer view can be checked through the file ("Chat platform Architecture.pdf")

2.3 System components

The system consists of several components as follows:

API gateway:

The single-entry point for the system, it receives REST API requests and redirect them to a microservice according to the requested data.

Queue service:

This service contains a queue of waiting users to be served by human agents, it matches agents with the users in the queue.

Further characteristics can be implemented in this service such as (Multiple levels of priority for users, matching agents according to their technical skills and specialization).

Chatbot service:

The chatbot service is one of the main services, it is used to manage the conversation with the users through communication with other services such as (NLP, Sentiment analysis and external services). The chatbot output is calculated after processing inputs from all the previously mentioned services.

NLP service:

NLP expects unstructured input from the user message including slangs and idioms, then outputs main keywords that can be understood by the chatbot service. The NLP service can be developed or provided by an external vendor according to maturity levels required.

Sentiment Detection Service:

It expects terms from the chatbot that were previously extracted by NLP to classify the current user sentimental state. The Sentiment Detection Service can be developed or provided by an external vendor according to maturity levels required as well.

Conversation Log Service:

This service is used to log and store all the messages by any of the involved parties (Users, agent and bots). All messages must pass by this service to be transferred to the other party to ensure consistent conversation history. This service can be implemented in a NOSQL pattern to ensure scalability.

Channel Formatting Service:

This service is used as a middleware between the platform and different channels, as each channel will have its own data format, this will ensure system extendibility with new channels.

Agents Service:

This service is used for all actions related to agents such as adding, editing, deleting and authentication of agents. It will include its own relational database to store agents data.

Chat Routing Service:

This service handles initiation of user – agent chat session and routes messages according to the current conversation held between the 2 parties (users and agents).

External Services:

Third party services such as CRM, can be extended with new services to support new features.

2.4 Communication Flow

The communication flow will have 2 starting points in the system, the first point is the flow of data when a public user initiates a session, the other is when the agent login the platform to start receiving messages.

User Data Flow:

1) The user will select one of the available channels.

2) A request will be sent to the API gateway to query users to be served queue length before letting the user to choose his next action, as the number of waiting users will heavily affect his decision and to encourage users to choose the bot solution.

3) User will choose a bot or agent.

4) According to the user's selection, one of two paths will start, whether a bot or an agent conversation. Case A will be for agent conversation while case B for bot conversation.

5.A) A session initiation request is sent to the chat routing service

6.A) The chat routing service will check the agents availability through the queue service.

7.A) When an idle agent is detected, the queue service will assign him a user from the queue and notify the chat routing service to proceed with session.

8.A) The agent is notified that he has been assigned a user to start the conversation.

9.A) The user is notified that the chat session has begun.
Note: The chat routing service will keep track of conversation using 3 main parameters (conversation ID, Cookie of the user and JWT token of the agent).

10.A) User post a message, the message is received by the chat routing service.

11.A) Message is sent to the conversation log service to be stored.

12.A) Confirmation of saving the message is sent back to the chat routing service.

13.A) The chat routing service redirects the message to the assigned agent.

14.A) Agent reply with a message that follow the same flow of the user message (starting from step 10.A)

The scenario of choosing a bot is as follows:

5.B) A session initiation request is sent to the chat bot service for maintaining the session with this user.

6.B) User post message that is received by the Chatbot.

7.B) The chatbot sends the user's input to NLP service as an unstructured input.

8.B) NLP extracts entities from the unstructured input

9.B) A structured output containing the extracted entities is returned from the NLP to the chatbot service.

10.B) Extracted entities are sent from the chatbot to the sentiment detection services for analyzing the user sentimental state.

11.B) A classification of the user sentimental state is returned to the chatbot.

12.B) The chat bot queries any external services if needed.

13.B) The output text is calculated based on the previous input and the business logic embedded within the chatbot.

14.B) The user input and chatbot output are both sent from the chatbot service to the Conversation Log Service for storage.

15.B) The output is passed to the channel formatting service before sending it back to the channel.

15.C) A special case can occur in case of detecting human intervention request, A session request will be sent to the chat routing service.

16.B) A channel-based format is sent from the channel formatting service to the API gateway.

17.B) The API gateway redirects the output to the designated channel

18.B) The output is sent to the user in a human readable format.

Agent Data Flow:

The agent data flow will have two extra steps in the begging which are associated with agent authorization.

1) Agent login request is sent to the API gateway, then re routed to the agents service.

2) Agent service check authorization for the entered credentials and start a session in case of granting access.

Rest endpoints can be implemented in each micro service for internal communication and data flow between the services.