

SNS Report : Server - Client Architecture

Andria Puja Pratama, 23157164

Internet Engineering

1 Client - Server Architecture

In the client-server model, the server remains active for incoming connections from clients. Once a client establishes a connection, the server engages in a conversational interface processing requests and delivering predictions based on the inputs received.

The Oracle bot employs TCP for its server architecture, primarily due to TCP's reliability and its capability to transmit data in a sequential order. This ensures that messages exchanged between the client and server are delivered accurately and in the correct sequence, which is crucial for maintaining the conversational flow and integrity of the predictive assistance provided by Oracle.

The server is built utilizing Python's socket and threading libraries, with the **OracleBotThread** Class manages each client connection in distinct threads for parallel communications with numerous clients, allowing for scalable and concurrent clients handling. Various Python functions encapsulate the server's capabilities including client connection establishment, handle the conversation, machine learning integration, and prediction tasks. While The client interface also built using Python's socket library, allows users to connect to the server to ask about weather and number of bike hires, and receive responses.

1.1 Integration to Machine Learning Model

A key aspect of the server is its incorporation of a machine learning model, named **MyModel**, which is tasked with forecasting weather conditions and number of bike hires. This model is activated using a reference date supplied by the client, ensuring personalized and contextually relevant predictions.

- **Model Initialization** : When a new client connects, the server asks for the current date to use as a reference point for predictions. This date initializes the **MyModel** instance, customizing its prediction to fit the specific situation of the client.
- **Model Prediction** : The server queries the ML model for temperature, wind speed, and precipitation forecasts for the weather conditions prediction. While for bike hire predictions, it requests the estimated number of bike

hires. These queries are made for the specified day, according to the client requests.

1.2 Sentence Processing

The server processes client sentences using a combination of keyword identification and regular expressions, enabling it to understand and respond to client requests. It identifies keywords related to the type of prediction (weather or bike hires) and extracts the day for which the prediction is requested using regular expressions. This approach allows the server to handle a variety of request formats flexibly.

- **Keyword Identification:** The server scans client messages for keywords related to the requested prediction type, such as "weather" or "bike hires." This method allows the server to understand what the client is asking for.
- **Regular Expressions:** The server uses regular expressions to identify phrases like "day X" where X represents a number, in order to pinpoint the exact day a prediction is sought for. This method offers adaptability in processing requests.

1.2.1 Interactions and Server Responses

This demonstrate how the server is equipped to interpret sentences in natural language, identify important details, and leverage the machine learning model to deliver predictions that are aware of the context.

1. Client Request Weather Predictions :

- **Client:** "How is the weather on day 3?"
- **Server Process:** The server identifies the keyword "weather" and extracts "3" as the day number. It then generates a weather forecast for day 3 using the machine learning model.
- **Server Response:** "For Day 3, the forecast is: high of 24.5°C, low of 16.2°C, wind speed at 5.4 km/h, and precipitation levels around 0.2 mm."

2. Client Request Number of Bike Hires Predictions :

- **Client:** "Can you estimate the number of bike hires on day 5?"
- **Server Process:** The server recognizes "bike hires" and extracts "5" as the day. It predicts the number of bike hires for day 5 based on the model.
- **Server Response:** "For Day 5, the expected number of bike hires is approximately 289."