**Multi-Agent System for Persona-Driven Technical Documentation**

**Problem Statement:**

To build a **multi-agent system** that takes a single source of truth - a **Python function's docstring** - and generates tailored documentations for three different personas, as follows, that work in **parallel**:

1. **Developer Documentation Agent**: Creates precise technical reference documentation for other engineers.
2. **Executive Summary Agent**: Writes high-level, jargon-free summary, focusing on business value.
3. **API User Guide Agent**: Generates a practical "how-to" guide with code examples for end-users.

**Technical Design:**

1. Designed an **autonomous** system using **Autogen** framework that uses the GroupChat, GroupChatManager features to orchestrate and manage the conversations between the **three agents**- Developer Documentation Agent, Executive Summary Agent and the API User Guide Agent. Used **LLM**, ‘GPT-4o-mini', which is deployed on **Azure**.
2. Extended the user interactivity of the framework into **Streamlit**. User input into the system is collected through the Streamlit app. The **interacting agents, intermediate** and **final outputs** are collected and **displayed on the Streamli**t app. This utilizes the ‘register\_reply’ functionality of the agents to collect the response of the agents through reply\_func parameter immediately after they are generated and these are streamed on Streamlit using a function, ‘st\_print’.
3. Also utilized **async** operation, using ‘asyncio’, for parallel processing of the conversations between the agents in Group Chat.
4. Used some Streamlit features to enhance the **user interface** and **user experience** (UI/UX) of the webpage.
5. Set the temperature of LLM to **0.1** to limit its creativity and use only facts and web results knowledge to generate responses.

**Agent Architecture:**