Multi-Agent Competitor Analysis - Solution Design Document

1. System Overview

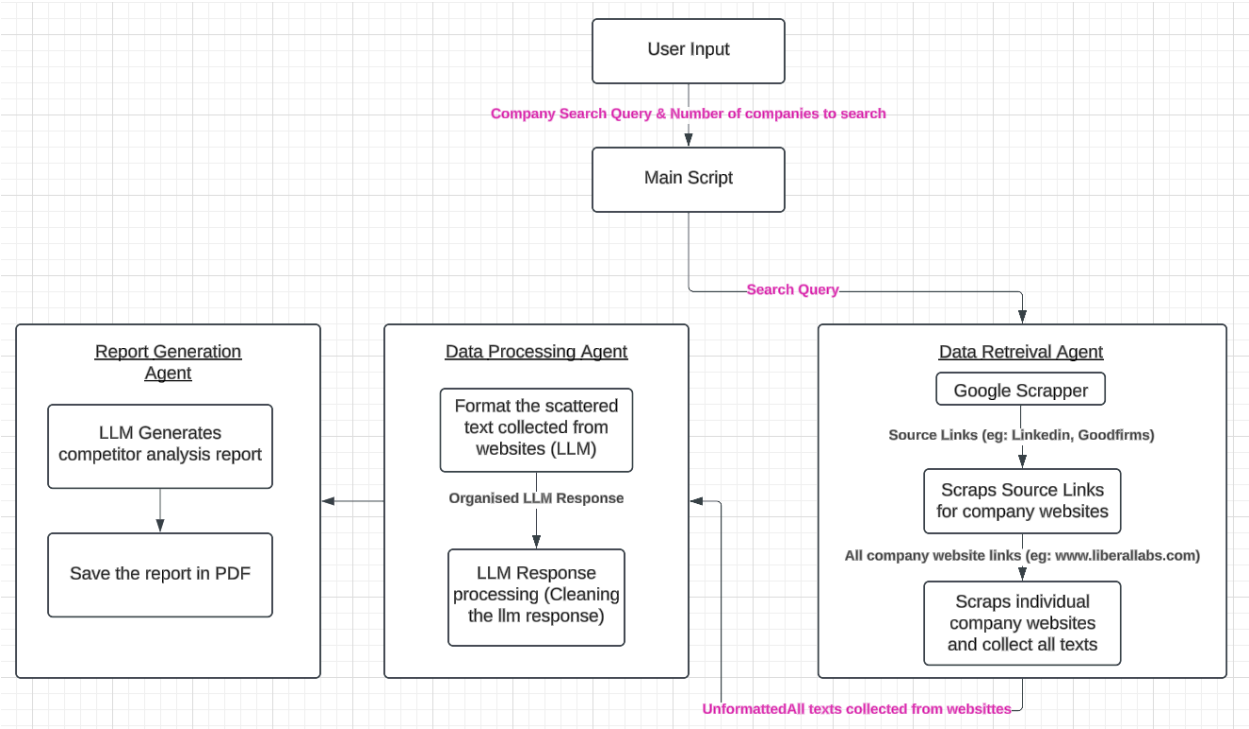
This document outlines the solution design for a **Multi-Agent Competitor Analysis System**, designed to collect, process, and generate comprehensive reports on competitors based on user input. The system integrates three main agents: **Data Retrieval Agent**, **Data Processing Agent**, and **Report Generation Agent**, orchestrated by a main script.

Objective

To create a robust competitor analysis system that:

- Scrapes and aggregates data from multiple company websites.
- Processes unstructured data into an organized format.
- Generates comprehensive competitor analysis reports in PDF format.

2. Architectural Design



The system consists of the following key components:

2.1. User Input

- **Input Parameters:**
    - Search Query (e.g., "AI startups in healthcare")
    - Number of Companies to Analyze
- This input initiates the workflow, triggering the main script to orchestrate tasks across agents.

2.2. Main Script

The central orchestrator coordinates the flow of data between the agents:

- Sends the search query to the **Data Retrieval Agent**.
- Passes the unstructured data to the **Data Processing Agent**.
- Directs the organized data to the **Report Generation Agent**.

2.3. Data Retrieval Agent

- **Tool:** Google Scraper
- **Functionality:**
    - Scrapes source links from platforms like LinkedIn, GoodFirms, and company websites.
    - Crawls individual company websites to collect all textual data.
- **Output:** A collection of unformatted important text for further processing.

2.4. Data Processing Agent

- **Tool:** OpenAI API
- **Functionality:**
    - Formats unstructured text collected by the Data Retrieval Agent.
    - Cleans and organizes the data into a coherent structure for analysis. The information which was retrieved are:
        - Services they provide
        - Technologies used
        - There targetted market
        - SWOT Analysis
- **Output:** Organized text ready for report generation.

2.5. Report Generation Agent

- **Tool:** OpenAI API
- **Functionality:**
    - Generates a detailed competitor analysis report based on the processed data.

    - Converts the report into a PDF format.
- **Output:** A professionally formatted PDF report.

3. Justification for Chosen Technologies

3.1. OpenAI API

- **Advantages:**
    - Superior language understanding and processing capabilities.
    - Efficient in transforming unstructured text into meaningful insights.
    - High accuracy in generating detailed and coherent reports.
- **Alternatives Considered:**
    - GPT-based open-source models (e.g., LLaMA, Falcon). However, they would require extensive setup and tuning, increasing complexity and resource consumption.
    - Chosen OpenAI API for its ease of integration and robust performance.

3.2. Google Scraper

- **Advantages:**
    - It used beautifulsoup at its core, works really well for collecting source links.
- **Alternatives Considered:**
    - Existing scraping tools like Scrapy or Beautiful Soup. While they are flexible, but a lot of manual engineering is needed to get desired output.

3.3. Frameworks

- **Selenium:** For dynamic web scraping and interaction with web elements during data retrieval.
- **OpenAI:** Essential for language processing and text summarization tasks, providing seamless integration for LLM-based operations.
- **PDFKit:** Used for converting text reports into PDF format.

4. Alternative Approaches and Trade-offs

4.1. Alternative LLMs

- **Considered:** Open-source LLMs (e.g., LLaMA, GPT-J).
- **Trade-offs:**
    - Open-source models would reduce API costs but require high **computational resources.**
    - I have built another similar type of project with pretrained LLM (LLaMa 3 - 8b). LLM Watch Expert

4.2. Prebuilt Data Processing Pipelines

- **Considered:** Prebuilt pipelines like spaCy and NLTK for text cleaning and structuring.
- **Trade-offs:**
    - These libraries are efficient but lack the context-aware processing capability of an LLM.
    - OpenAI's end-to-end language processing provided a more streamlined solution.

4.3. Alternative Report Generation Tools

- **Considered:** LaTeX for professional PDF formatting.
- **Trade-offs:**
    - While LaTeX offers advanced formatting, the integration complexity outweighed its benefits for this use case.

6. Conclusion

This multi-agent system integrates state-of-the-art technologies to provide an efficient, scalable, and user-friendly solution for competitor analysis. The chosen design balances functionality, efficiency, and resource usage while ensuring compliance with ethical and security standards.