PROBLEM STATEMENT 2: Multi-Agentic System with Dynamic Decision Making

Name:	**Arshiya Mubarakali Saiyyad**

1. Introduction

This document explains the architecture of the multi-agent system designed for processing and answering questions based on PDFs, web data, and academic papers. It details the system components, decision-making logic, safety and privacy considerations, and discusses current limitations.

2. System Architecture

A) Overview:

The system is built using a modular structure with a FastAPI backend that exposes REST APIs for interaction. Core components include:

- Main API (`main.py`): Handles user requests, file uploads, and logs.
- Controller (`agents/controller.py`): Orchestrates multiple specialized agents based on user queries.
- Agents:
  1) **PDF RAG Agent (`agents/pdf_rag.py`):** Extracts and processes PDF content for question answering.
  2) **Web Search Agent (`agents/web_search.py`):** Performs web searches for real-time information.
  3) **ArXiv Agent (`agents/arxiv_agent.py`):** Retrieves academic papers relevant to the query.
  4) **Sample PDFs (`sample_pdfs/`):** Repository of PDFs for testing.

B) Data Flow:

1. User Interaction: Users upload PDFs or ask questions via API endpoints.
2. Question Handling: The controller assesses which agents to invoke.
3. Agent Processing:
- PDF content is extracted and embedded for retrieval.
- Web search results are fetched via external APIs.
- ArXiv searches find relevant academic papers.

4. Response Aggregation: Responses are combined, reasoned about, and formatted.
5. Logging: Interactions are recorded for future reference.

3. Controller Decision Logic

The controller (`agents/controller.py`) determines which agents to invoke based on the user's question:

- **Question Analysis:**
  - Simple keyword detection or NLP classification to decide whether a question requires PDF info, web data, or academic papers.
  - For example, questions mentioning “latest news” trigger web search; questions about research papers trigger ArXiv.

- **Agent Invocation:**
  - Calls relevant agents asynchronously or sequentially.
  - Collects responses and evaluates their relevance and confidence.

- **Response Synthesis:**
  - Combines agent responses.
  - Uses language models to generate a coherent answer.

- **Decision Logging:**
  - Records which agents were used and the reasoning behind the selection.
  - Stores the final answer and intermediate responses.

This logic ensures that the system dynamically adapts to different question types, optimizing accuracy and relevance.

4. Safety and Privacy Handling

A) Data Privacy

- **File Storage:**
  - PDFs uploaded by users are stored temporarily in a designated folder (`sample_pdfs/`) and are deleted after processing if not needed.

- **Data Handling:**
  - User questions and responses are processed in-memory or stored securely.
  - Sensitive data is not shared externally unless explicitly configured.

- **External APIs:**
  - Web searches via SerpAPI or similar services may transmit query data.
  - The system ensures API keys and user data are stored securely, with access limited.

B) Safety Considerations

- **Content Filtering:**

- Responses generated from web or PDF data are filtered for harmful or inappropriate content using heuristics or NLP filters.

- **Rate Limiting:**
  - API endpoints are rate-limited to prevent abuse.

- **Logging and Monitoring:**
  - All interactions are logged for audit and anomaly detection.

- **User Consent:**
  - Clear instructions are provided for users regarding data privacy.

C) Compliance

- The system aligns with privacy standards such as GDPR by not storing personally identifiable information longer than necessary and providing options for data deletion.

5. Limitations

A) Technical Limitations

- **Accuracy of Extraction:**
  - PDF text extraction may be imperfect, especially with scanned documents or complex layouts.

- **Agent Coverage:**
  - The system relies on the availability and accuracy of external APIs (e.g., SerpAPI, ArXiv API).

- **Response Latency:**
  - Multiple agent calls, especially web searches, can introduce delays.

- **Context Handling:**
  - Currently processes each question independently without maintaining conversation context.

B) Functional Limitations

- **Limited NLP Understanding:**
  - The question classification might not handle complex or ambiguous queries effectively.

- **Scalability:**
  - Designed for small to medium scale; high concurrency or large datasets require optimization.

- **Security:**
  - Not hardened for production environments; additional security layers are recommended for deployment.

C) Ethical and Privacy Concerns

- Users should be cautious when uploading sensitive documents, as extraction involves temporary storage.

- External API data transmission may expose queries; sensitive information should be avoided or anonymized.

6. Conclusion

This multi-agent system provides a flexible and extensible framework for answering questions based on PDFs, web data, and academic papers. While effective in many scenarios, ongoing improvements are needed to address current limitations, enhance accuracy, and ensure robust privacy and security measures.