Multi-Agent RAG System — Architecture Report

Generated: 2025-10-07 09:19:16

This report accompanies the demo skeleton for the Multi-Agent Retrieval-Augmented Generation system. It documents architecture, controller logic, agents, security, and deployment notes.

Key components: - Controller Agent: Receives user requests and routes to PDF RAG, Web Search, ArXiv, or combinations. - PDF RAG Agent: Ingests PDFs, chunks, embeds, stores vectors in FAISS, and retrieves relevant chunks. - Web Search Agent: Queries SerpAPI or fallback search and returns top results with summaries. - ArXiv Agent: Queries the arXiv API for recent papers and returns metadata and summaries. - Synthesizer: LLM-based answer composition (placeholder in demo).

Routing rules (examples): - If request includes uploaded PDF and user asks to summarize: use PDF RAG. - If query contains 'arxiv' or 'recent papers': use ArXiv agent. - If query contains 'latest' or 'recent developments': use Web Search. - Otherwise: default to RAG if a PDF context exists, else Web Search + synthesize.

Security & Privacy: - Uploaded PDFs limited in size and lifespan. By default, retention is short and PII redaction is recommended. - API keys must be stored in environment variables (.env or deployment secrets).

Deliverables included in this repo: - backend/: Flask app and agent stubs - frontend/: minimal UI - sample_pdfs/: 5 demo PDF dialogs - REPORT.pdf (this document)