Agentic RAG System - System Design Document

Author: Kiran **Date:** February 2026

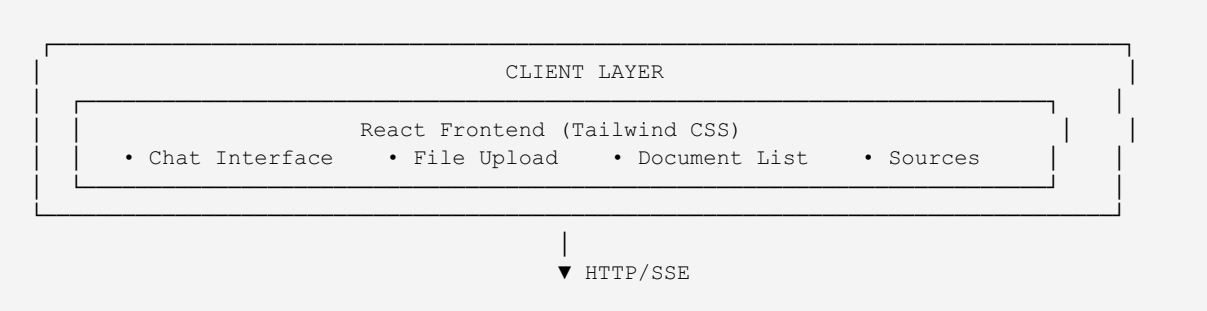
1. Executive Summary

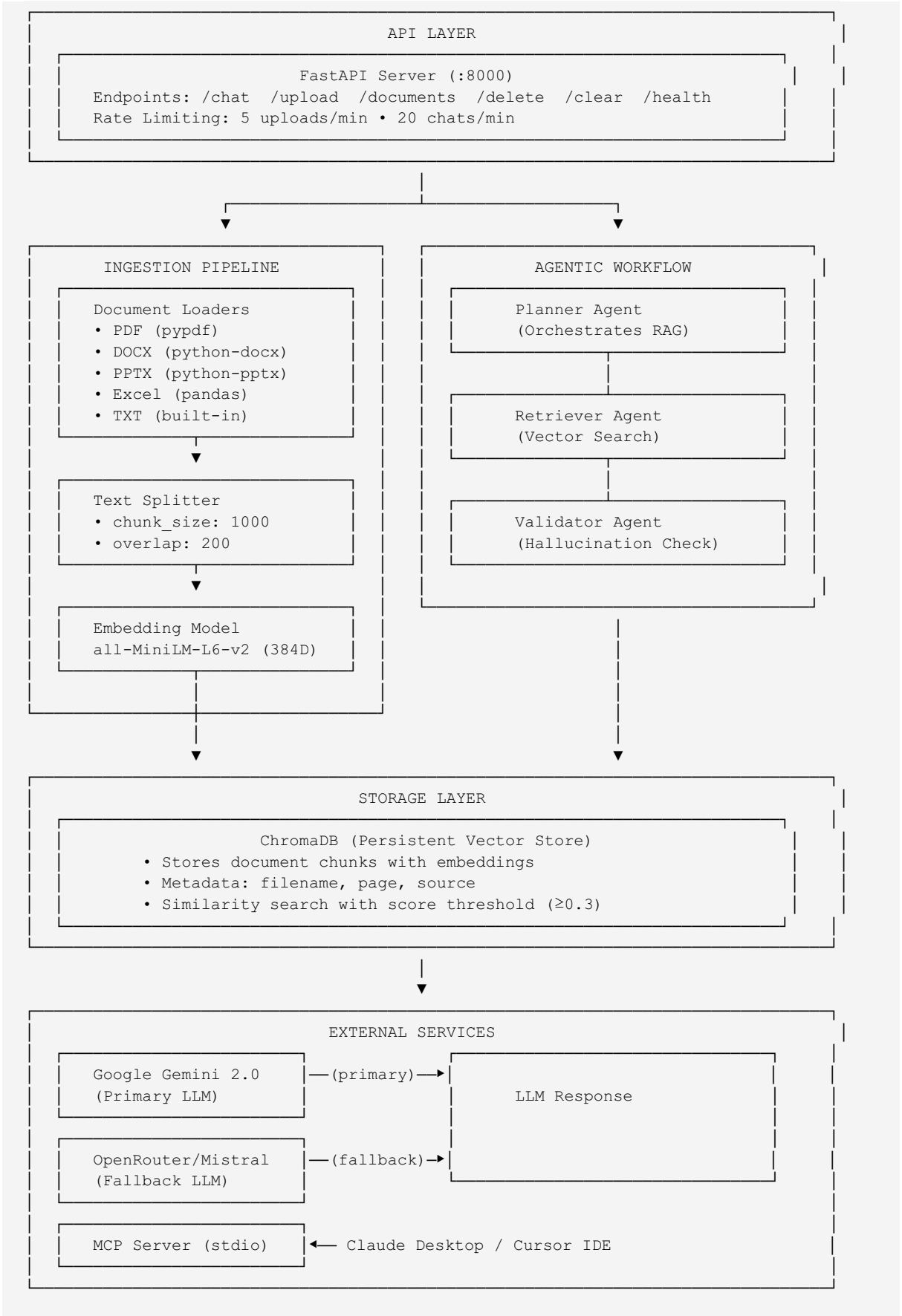
This document describes the architecture of an **Agentic RAG (Retrieval-Augmented Generation) System** that intelligently retrieves and answers questions from multi-format documents. The system combines semantic search, LLM reasoning, and agentic workflows to provide accurate, citation-backed responses.

Key Capabilities

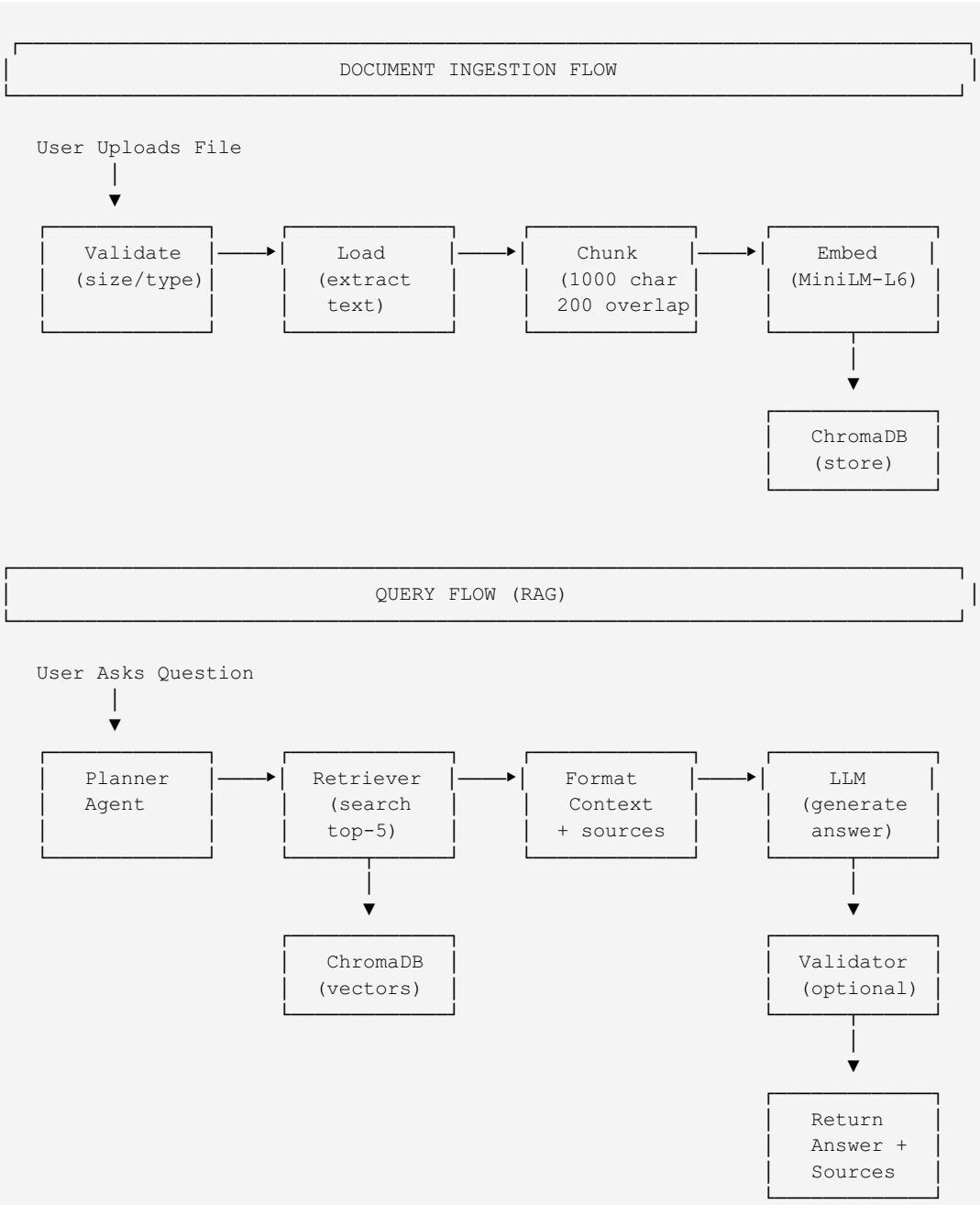
- Multi-format document ingestion (PDF, DOCX, PPTX, Excel, TXT)
- Semantic vector search using ChromaDB
- Agentic query processing with planning and validation
- Real-time streaming responses via SSE
- MCP Protocol integration for external AI tool connectivity
- Intelligent LLM fallback strategy (Gemini → OpenRouter)

2. System Architecture Diagram




API LAYER
FastAPI Server (:8000)
Endpoints: /chat /upload /documents /delete /clear /health
Rate Limiting: 5 uploads/min • 20 chats/min
INGESTION PIPELINE
Document Loaders
• PDF (pypdf)
• DOCX (python-docx)
• PPTX (python-pptx)
• Excel (pandas)
• TXT (built-in)
Text Splitter
• chunk_size: 1000
• overlap: 200
Embedding Model
all-MiniLM-L6-v2 (384D)
AGENTIC WORKFLOW
Planner Agent
(Orchestrates RAG)
Retriever Agent
(Vector Search)
Validator Agent
(Hallucination Check)
STORAGE LAYER
ChromaDB (Persistent Vector Store)
• Stores document chunks with embeddings
• Metadata: filename, page, source
• Similarity search with score threshold (≥0.3)
EXTERNAL SERVICES
Google Gemini 2.0
(Primary LLM)
(primary)
OpenRouter/Mistral
(Fallback LLM)
(fallback)
LLM Response
MCP Server (stdio)
Claude Desktop / Cursor IDE

3. Data Flow Diagram



4. Agentic Workflow Design

The system uses a **Plan-Retrieve-Generate-Validate** pattern:

Step	Agent	Action
1	**Planner**	Receives user query, coordinates the workflow
2	**Retriever**	Searches ChromaDB for top-5 relevant chunks
3	**Planner**	Formats context with source citations
4	**Planner**	Sends context + query to LLM (Gemini/OpenRouter)
5	**Validator**	(Optional) Checks answer is grounded in sources
6	**Planner**	Returns answer + sources to frontend

LLM Fallback Strategy: - Primary: Google Gemini 2.0 Flash (best reasoning, free tier) - Fallback: OpenRouter Mistral (if Gemini fails)

---

5. Context Construction Strategy

1. **Semantic Search**: Query embedding → ChromaDB similarity search → Top-5 chunks
2. **Score Filtering**: Only chunks with similarity ≥ 0.3 are included
3. **Metadata Preservation**: Each chunk includes filename, page number, source
4. **Prompt Augmentation**: Context injected into system prompt with citation IDs
5. **Citation Mapping**: LLM references source IDs → Frontend displays verified references

---

6. Technology Choices & Rationale

Component	Technology	Rationale
Frontend	React + Tailwind	Modern UI, responsive, good DX
Backend	FastAPI	

Component	Technology	Rationale
		Async-native, auto OpenAPI docs, Pydantic validation
Vector DB	ChromaDB	Zero-setup, persistent, ideal for < 10k docs
Embeddings	all-MiniLM-L6-v2	Local (no rate limits), 384D, fast inference
Primary LLM	Gemini 2.0 Flash	Best reasoning, large context window, free tier
Fallback LLM	Mistral (OpenRouter)	High availability, cost-effective
Streaming	Server-Sent Events	Native browser support, real-time UX

---

7. Key Design Decisions

1. **Streaming SSE Responses**: Real-time "thinking" states improve UX
2. **Multi-Format Loaders**: Custom loaders for Excel (row-by-row) and PPTX (slide-by-slide)
3. **Windows-Optimized ChromaDB**: `allow_reset=True` prevents file-lock issues
4. **Modular Architecture**: Loaders, embeddings, and vector DB can be swapped independently
5. **Rate Limiting**: Protects against API abuse (5 uploads/min, 20 chats/min)
6. **MCP Integration**: Exposes search tool for Claude Desktop/Cursor integration

---

8. Limitations

Limitation	Description	Future Solution
No Image Analysis	Only extracts text from documents	Integrate Vision LLM
Single-Tenant	All documents in one collection	Add user namespaces
Scale Ceiling	ChromaDB optimal for < 10k files	Migrate to Milvus
No OCR	Scanned PDFs not supported	Add Tesseract preprocessing
Sequential Workflow	Single plan-retrieve-generate cycle	Add iterative self-correction

---

9. API Reference

Endpoint	Method	Description
`/chat`	POST	Submit question, get answer with sources
`/chat/stream`	POST	Streaming SSE response
`/upload`	POST	Upload documents (PDF, DOCX, PPTX, XLSX, TXT)
`/documents`	GET	List all ingested documents
`/delete/{filename}`	DELETE	Remove a document

Endpoint	Method	Description
`/clear`	POST	Clear all data
`/health`	GET	Health check

This system demonstrates agentic RAG workflows with production-ready architecture for intelligent document Q&A.