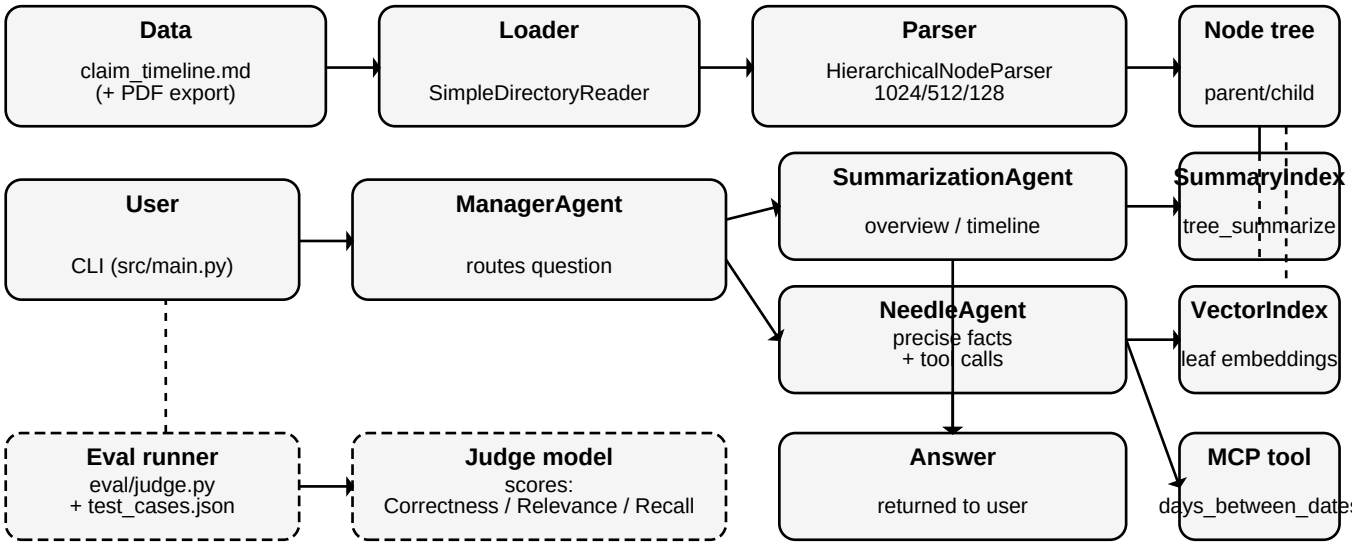
Midterm – Insurance Claim Agents

Agentic Patterns • Hierarchical Retrieval■Augmented Generation (RAG) + Model Context Protocol (MCP) tool

System overview

A small agentic system that answers questions about an insurance claim. Input is a single claim timeline file structured as **Claim** → **Document** → **Section** headings, including a **second/minute■resolution** incident log and “needle” details embedded in longer text; the dataset is also exported as a **10+ page PDF**. LlamaIndex builds a **hierarchical chunk tree** (1024/512/128 tokens) so the system can answer both high■level summaries and precise lookups.

Architecture (one■page view)



Main results

Evaluation uses **src/eval/judge.py** with fixed test cases (**test_cases.json**) and a judge model scoring **Correctness**, **Relevance**, and **Recall**. Latest run: **Correctness 4.88**, **Relevance 5.00**, **Recall 4.88** (average over 8 queries), including “needle” retrieval and a tool■assisted date■difference question.

MCP usage (grader■verifiable)

The date■difference function (**days_between_dates**) is exposed as an MCP server over **STDIO** and invoked via **mcp_integration**. Set **USE_REAL_MCP=1** and **ALLOW_MCP_FALLBACK=0** (strict mode) to require MCP: the client logs **[REAL MCP]** when used and fails if MCP cannot run.

How to run

Install: `pip install -r requirements.txt` (and set `OPENAI_API_KEY`)
App: `python src/main.py`
Eval: `python src/eval/judge.py`
Strict MCP: `$env:USE_REAL_MCP="1"; $env:ALLOW_MCP_FALLBACK="0"; python src/main.py`