Chatbot

Architecture (components)

- **Ingest pipeline:** parser → chunker (semantic + sliding windows) → embedder → vector DB (Faiss/Weaviate/Pinecone).
- **Retriever layer:** metadata filters + semantic retriever → top-K candidates.
- **Fusion layer:** score normalization + reranker + fusion (RRF/Borda-like).
- **Agent/Planner:** LangChain agent with tool registry (Pydantic models) & function-calling support.
- **Tools:** internal search, summarizer, policy-extractor, OCR (OpenAI Vision / Tesseract), DB reads/writes.
- **Frontend:** Agent UI for agents with trace view + “source docs” panel.
- **Observability:** token/cost logging, latency, groundedness metric.

LangChain tool wrapper pattern:

- Tools register with `name`, `schema`, `execute()`; LangChain function-calling uses JSON schema derived from Pydantic.

Chunker approach:

- Produce semantic chunks ~500–1,200 tokens using paragraph boundaries + transformer sentence embeddings; overlapping sliding windows for context continuity.

Fusion:

- Normalize cosine scores and metadata match score, then RRF-style final rank.

Tests & CI

- Unit: chunker edge cases, embedder consistency, tool input/output validation (Pydantic).
- Integration: mock LLM + deterministic vector DB (small dataset) to assert tool selection & final answer includes citation.

- E2E: staged test with a sample transcript → expected agent action.
- CI: pytest + mypy + contract tests run on PR; also automatic smoke E2E using mocked LLM.

Metrics & success criteria

- Retrieval recall@5 > X (baseline)
- Groundedness (fraction of answers with correct citation) > 95%
- Average time-to-answer for agent < 8s
- Agent adoption: % of calls where agents used the assistant
- Cost per call (tokens + inference) reduced vs naïve prompt by 25%

Interview talking points / likely follow-ups

- Why Pydantic? — guarantees typed I/O, catches schema drift, enables safe function-calls.
- How to prevent hallucination? — fusion + reranking + forcing citation + grounding heuristics + refusal policy.
- How to handle new policy uploads? — streaming ingestion, incremental embedding updates, versioning & reindexing window.
- Failure modes & mitigation: stale embeddings, broken OCR → fallback to human + alerting.