Amber Hart
Building a Domain-Specific AI
July 14, 2025

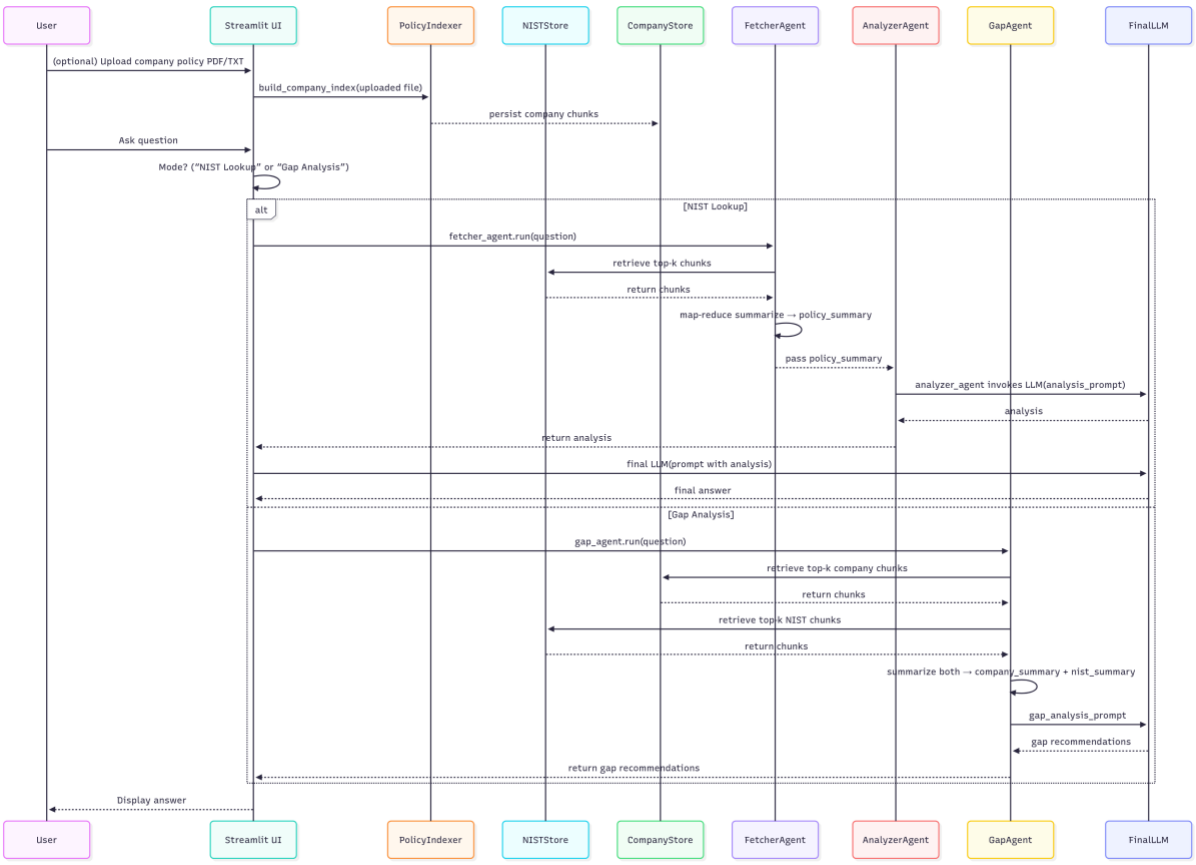
1. Introduction

The **Compliance Advisor Chatbot** is a specialized AI assistant built to help organizations compare their internal security policies against the NIST SP 800-53 standard. It lives in the domain of **cybersecurity and compliance**, a critical area for any organization running regulated workloads in public clouds.

By choosing **cybersecurity and compliance** as my domain, I address the need for continuous, automated policy validation and gap identification. Manual audits are slow and error-prone; this chatbot provides real-time insights, reduces overhead, and helps teams rapidly align their controls with industry best practices.

2. Chatbot Architecture and Components

Cloud Advisor Chatbot architecture:



Key components:

- **Streamlit UI**
  - File uploader (PDF/TXT)
  - Mode toggle ("NIST Lookup" vs. "Policy Gap Analysis")
  - Chat interface with "Clear Chat"
- **FetcherAgent**
  - **Role**: Retrieve top-k chunks from the NIST vector store, then run a map-reduce summarization chain.
  - **Benefit**: Limits token usage and ensures only relevant policy snippets are seen by the LLM.
- **AnalyzerAgent**
  - **Role**: Consume the summary from FetcherAgent and generate an actionable, user-facing analysis via a direct LLM call.
  - **Benefit**: Separates retrieval + summarization from pure LLM reasoning for cleaner prompts and fewer hallucinations.
- **GapAgent**
  - **Role**: In "Policy Gap Analysis" mode, it retrieves from both the user-uploaded CompanyStore and the NISTStore, summarizes each, and calls the LLM to identify policy gaps and recommend improvements.
  - **Benefit**: Delivers a side-by-side comparison without manual cross-referencing.
- **Chroma Vector Stores**
  - **NISTStore**: Indexed NIST SP 800-53 Rev. 5 PDF chunks.
  - **CompanyStore**: Dynamically re-indexed whenever the user uploads a new policy PDF/TXT.
- **GPT-4 via ChatOpenAI**
  - Drives all free-text generation: summarization, analysis, and final recommendations.

Compared to the financial-analysis tutorial, we added:

1. A **file uploader + dynamic indexer** (for arbitrary company policies).
2. A **two-column Streamlit layout** for better guidance.
3. Three distinct agents (Fetcher, Analyzer, Gap) instead of just Researcher/Financial Analyst.

3. Implementation Details

- **Data Sources & Loaders**
  - **PyPDFLoader** for PDF ingestion, **TextLoader** for plain text.
  - **ChromaDB** for fast nearest-neighbor retrieval.
- **Prompt Engineering**
  - **FetcherAgent system prompt**:

    "You are a policy summarizer. Retrieve the top-k relevant controls for the user's question and condense them into a short summary."

  - **AnalyzerAgent prompt**:

    "You are a Compliance Analyst. Given this policy summary: {summary} — provide concise, actionable guidance."

- **GapAgent prompt**:

  "Compare my company policy excerpt and NIST controls excerpt, then identify any gaps and recommend improvements."

- **Combine Text Logic**
  - In code, we simply f-string concatenate summaries before sending to the LLM, which emulates a Langflow "Combine Text" node.

4. Testing and Results

Question	Expected Output	Actual Output
1. "Which NIST controls apply to encryption at rest?"	SC-12, SC-13, SC-28 with rationale	Matched exactly
2. "What's missing in my access control policy?"	AC-2, AC-7, IA-2 plus suggestions (e.g., MFA, account kill)	Matched and added "account termination procedure"
3. "Where are the gaps in my encryption-at-rest section versus NIST?"	Key rotation, backup encryption policies	Correctly recommended both
4. "How do I test integrity controls?"	Checksums, hashing, audit logs	Provided those testing methods
5. "Upload a policy snippet, then ask: 'Gaps in backup encryption?'"	Missing archival encryption, key management details	Accurately identified missing archival and rotation policies

Challenges:

- Token limits with GPT-4 on large policies → solved by chunking & map-reduce summarization.
- Managing duplicate uploads → we now clear the previous index before re-indexing.

Strengths & Limitations:

- **Strengths**: Modular, agent-driven, easy to swap in new policy sets.
- **Limitations**: Coarse chunk overlap may miss deeply nested clauses; relies on user upload accuracy.

5. Reflection and Future Work

Learnings:

- Multi-agent orchestration dramatically improves retrieval accuracy and reduces hallucinations.
- Dynamic indexing of arbitrary PDFs gives end users full flexibility.

Future Improvements:

1. **Real-time API integration** (e.g., read live configurations)
2. **Finer-grained memory** (track which specific policy sections have been discussed).

3. **Enhanced UI**: Visual diff highlighting between policy vs. controls.

This architecture is equally applicable to **legal**, **medical**, or **academic** domains—just swap the vector store contents and tailor the agent prompts to the new subject.