Workflow

Workflow for a Policy-as-Code RAG System

1. Document Ingestion and Preprocessing

- Ingest all authoritative policy documents (e.g. college bylaws, UGC, AICTE regulations) and preprocess them: segment into clauses; clean, normalize and embed the text.
- Store document chunks as vector embeddings in a scalable vector database.

2. Semantic Retrieval

- When a compliance query or question is entered, convert it into an embedding and use semantic search (e.g. FAISS, Pinecone) to retrieve the most relevant document chunks from the knowledge base.

3. Context-Aware Answer Generation

- Pass the retrieved chunks to a language model (LLM). Synthesize an evidence-backed response that directly references and cites the sources that support its claims.
- Ensure each answer is traceable: cite exact clause and document origins.

4. Conflict Detection and Requirements Alignment

- When queries span multiple frameworks, compare and align retrieved rules to detect overlaps, conflicts, and hierarchical priorities (lex specialis, lex superior, etc.).
- Generate a proof or rationale graph that visualizes the alignment or conflict resolution.

5. Explainability and Verification Module

- Employ RAG-Ex or similar frameworks for post-hoc explanation: show which retrieved documents influenced which parts of the answer.
- Give human-facing outputs that clarify how compliance is determined and what evidence was used.

6. Evaluation and Continuous Improvement

- Regularly evaluate system outputs using benchmarks (RAG Bench). Measure citation precision, evidence completeness, hallucination rate, and overall explainability.
- Use qualitative (manual) and quantitative (automated) evaluation scores to guide further tuning.

7. Modular Update and Domain Adaptation

- Support adding/updating knowledge bases easily; facilitate domain adaptation and multi-lingual extension for broader or changing compliance frameworks.