🏛️ The Three-Agent Architecture

You will now build three specialized Python "Agents" that work together to audit a new protocol.

1. The Auditor Agent (The "Inspector")

- **Goal:** Scan a new protocol for risks.
- **Python Logic:** Use a **Vector Database** (like ChromaDB or FAISS) to store your `llm_master_analysis.json`. When you input a new protocol, the agent performs a "Similarity Search" to find past FDA letters that match the new protocol's design.
- **Output:** A list of "Red Flags" (e.g., "This sample size is suspiciously similar to the one rejected in FDA Letter X").

2. The Biostats Optimizer (The "Expert")

- **Goal:** Provide the math-heavy solution to the Red Flags.
- **Python Logic:** This agent is programmed with "Chain-of-Thought" prompts specifically for **Biostatistics literature** (ICH E9).
- **Functionality:** If the Auditor flags a p-hacking risk, this agent will suggest specific corrections:
    - **Alpha Spending:** "Apply a Bonferroni or O'Brien-Fleming boundary."
    - **Power:** "Increase N from 50 to 128 to achieve $1-\beta=0.8$."
    - **Causal Inference:** "Switch from Per-Protocol to Intent-to-Treat (ITT) analysis."

3. The Orchestrator (The "Manager")

- **Goal:** Summarize the conversation into a final report for a human.
- **Python Logic:** A final LLM call that synthesizes the "Risk" and the "Correction" into a professional PDF or Dashboard.

💻 The Python Workflow

Here is the technical sequence:

1. **Vectorization:** Convert the JSON findings into "Embeddings" (numerical representations of the text) so the AI can "search" them quickly.
2. **Protocol Parsing:** Write a function to take a `.pdf` or `.docx` clinical protocol and break it into segments (Background, Endpoints, Statistical Plan).
3. **Agent Loop:** * Send a segment to the **Auditor**.
    - If a flag is raised, pass the flag to the **Optimizer**.
    - Collect the "Optimized" version.

4. **Verification:** Compare the original protocol against the optimized one to ensure the "Guardrails" are met.

🧠 Deep-Dive: The Biostats "Checkpoints"

Emphasize how the agent handles these specific statistical pitfalls:

1. P-Hacking & Multiple Comparisons

The agent flags when a protocol lists 10+ secondary endpoints but doesn't specify a **Multiplicity Adjustment** (e.g., Benjamini-Hochberg). This is a top reason for FDA rejection.

2. Power & Sample Size Logic

The agent doesn't just check for a number; it checks for the **Effect Size assumptions**. If the protocol expects a 40% improvement (highly optimistic) with only 50 patients, the agent flags "Underpowered/Over-optimistic assumptions."

3. Causal Inference & Covariates

The agent looks for the **Statistical Analysis Plan (SAP)** to include **Intent-to-Treat (ITT)** analysis vs. **Per-Protocol**. It also flags if major confounders (like age, baseline severity, or previous treatments) are not listed as covariates in the regression models.

4. Representativeness

Using the LLM's reasoning, the agent compares the **Inclusion/Exclusion criteria** against the general population demographics of the disease. If a trial for a disease prevalent in elderly populations excludes anyone over 65, the agent flags a **Generalizability (External Validity) Risk**.

📈 Why this is valuable for a Job Posting:

1. **Domain Expertise:** Shows I understand the high-stakes world of FDA 21 CFR Part 11 and E9 statistical principles.
2. **Agentic Design:** Demonstrates I can build a system that doesn't just "process text" but "reasons through tools."
3. **Risk Mitigation:** Companies love tools that act as a "Pre-submission Gatekeeper," potentially saving them millions in failed filings.