# **Technical Design Specification: Tri-Phasic Agentic System**

Version: 3.1 (Revised)

Date: 15 June 2025

Status: Proposed (Based on Technical Research Report v16)

### 1. Architectural Overview

This specification details a tri-phasic, hierarchical agentic system implementing a Plan -> Synthesize -> Evaluate workflow. The architecture uses three distinct computational graphs, with two Human-in-the-Loop (HITL) gates for supervision. A key feature is the use of an independent committee of evaluator agents (using a different LLM family) to ensure unbiased auditing and verifiability.

* Workflow: Constitutional Planning -> HITL Plan Approval -> Grounded-Source Synthesis -> Independent Evaluation -> HITL Evaluation Review & Optional Restart.
* Supervisor (supervisor.py): Orchestrates the complete tri-phasic workflow, managing the state and HITL gates between the three distinct graphs. It will be responsible for initializing and managing clients for multiple LLM providers (e.g., Gemini for Planning/Synthesis, DeepSeek for Evaluation).
* Agent Core (agent\_core.py): Will now contain three factory functions: build\_planning\_graph(), build\_synthesis\_graph(), and build\_evaluation\_graph().

### 2. State Management (GraphState)

The GraphState is updated to manage the full feedback loop.

class GraphState(TypedDict):

# Core Inputs

user\_prompt: str

task\_mode: str

documents: List[Document]

# Stage 1: Planning

plan: Optional[List[str]]

# Stage 2: Synthesis

output: str # The final assembled artifact with in-line citations

# Stage 3: Evaluation & Feedback Loop

evaluation\_report: Optional[str] # The report from the evaluation graph

user\_feedback: Optional[str] # User's notes to guide the next loop

# Universal State

# (critique, num\_revisions, etc. will be managed within graph scopes)

log: List[str]

### 3. Computational Graphs

#### 3.1. build\_planning\_graph() (Gemini)

* Input: user\_prompt, documents, and optionally evaluation\_report and user\_feedback from a previous loop.
* Nodes: generate\_plan\_draft, critique\_plan.
* Logic: Engages in a constitutional debate to produce a single, high-quality plan. If user\_feedback is present, the initial prompt is amended to instruct the agent to create a new plan that addresses the feedback.
* Output: A plan.md artifact.

#### 3.2. build\_synthesis\_graph() (Gemini)

* Input: The user-approved plan.
* Nodes:
  + execute\_synthesis\_step: Crucially, the prompt for this node will now instruct the LLM to generate content AND provide an in-line citation in the format [Source: doc\_name] for each major piece of information.
  + assemble\_draft: No change.
  + holistic\_review: No change.
* Logic: Executes the plan and produces a single, internally-reviewed artifact containing Grounded-Source Citations.
* Output: A report\_with\_citations.md artifact.

#### 3.3. build\_evaluation\_graph() (DeepSeek)

This is a new, independent graph.

* Input: The report\_with\_citations.md and the original documents.
* Nodes:
  + parse\_citations\_from\_report: A utility node (regex-based) that extracts all [Source: ...] tags and the associated sentences into a structured list.
  + check\_consistency: An LLM node that reviews the full report for logical flow and coherence.
  + verify\_citations: For each extracted citation, this node retrieves the cited document chunk and asks an LLM to answer: "Does the source text support the claim made in the report?" It generates a list of (citation, claim, source\_text, verification\_status [PASS/FAIL]).
  + check\_goal\_alignment: An LLM node that compares the final report to the original user\_prompt to assess if the strategic goal was met.
  + assemble\_evaluation\_report: Takes the outputs of the three check nodes and formats them into a single, human-readable Markdown report with sections for Consistency, Citation Audit, and Goal Alignment.
* Output: An evaluation\_report.md artifact.

### 4. Supervisor and UI Logic

The Supervisor's state machine becomes more complex:

1. Initial Request: User clicks "Generate Research Report."
2. Run Planning -> HITL Gate 1:
   * Supervisor runs the Planning Workflow.
   * UI displays the generated plan.md.
   * UI displays an "Approve Plan & Begin Synthesis" button.
3. Run Synthesis & Evaluation:
   * On button click, the Supervisor runs the Synthesis Workflow, generating report\_with\_citations.md.
   * The Supervisor immediately passes this output to the Evaluation Workflow, generating evaluation\_report.md.
4. HITL Gate 2 (Final Review & Refinement Loop):
   * The UI displays the final synthesized report and the evaluation report side-by-side.
   * The UI presents a text box for user\_feedback.
   * The UI presents two buttons: "Accept Final Report" and "Re-Plan with This Feedback".
5. Loop or Terminate:
   * If "Accept" is clicked, the process ends.
   * If "Re-Plan" is clicked, the supervisor bundles the evaluation\_report and the user\_feedback text, updates the GraphState, and returns to Step 2, triggering the Planning Workflow again with the new feedback.

Technical Design Specification v3.0