# **Technical Design Specification: Hierarchical Agentic System**

Version: 3.0

Date: 15 June 2025

Status: Proposed (Based on Technical Research Report v15)

### 1. Architectural Overview

This specification details a hierarchical, two-stage agentic system designed to maximize user control and artifact quality. The core principle is the formal separation of Planning from Synthesis. The system will use two distinct LangGraph computational graphs, planning\_graph and synthesis\_graph, orchestrated by a master supervisor. A mandatory Human-in-the-Loop (HITL) approval step connects the two stages.

* Workflow: Constitutional Planning -> User Approval -> Constitutional Synthesis.
* Supervisor (supervisor.py): The master orchestrator. Manages the overall two-stage workflow. It is responsible for invoking the correct graph, managing state between stages, handling the user interaction for plan approval, and orchestrating the final constitutional debate.
* Agent Core (agent\_core.py): The library containing all node functions and graph-building logic. It will now contain two factory functions: build\_planning\_graph() and build\_synthesis\_graph().
* Configuration (prompts.yaml): External configuration holding all prompts, now including separate prompts for the planning and synthesis phases.

### 2. State Management (GraphState)

The GraphState will be enhanced to manage the new hierarchical workflow.

class GraphState(TypedDict):

# Core Inputs

user\_prompt: str

task\_mode: str # {research, design\_spec, code}

documents: List[Document]

# Stage 1: Planning Artifact

plan: Optional[List[str]] # The final, user-approved plan

plan\_critique: str # For debate during the planning stage

# Stage 2: Synthesis Artifacts

working\_memory: Optional[Dict[str, str]] # For section-by-section generation

completed\_plan\_items: List[str]

output: str # The final assembled artifact

output\_critique: str # For debate during the synthesis stage

# Universal State

num\_revisions: int

log: List[str]

### 3. Computational Graphs

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

This graph's sole responsibility is to produce a high-quality plan.

* Entry Point: retrieve\_context
* Nodes:
  1. generate\_plan\_draft: Takes the user\_prompt and documents, and generates an initial plan (a list of strings).
  2. critique\_plan: Takes a peer's plan and critiques it based on the constitution and grounding in the documents.
* Logic:
  1. The supervisor will call this graph for Agent A and Agent B to get two initial plans.
  2. The supervisor will then orchestrate a constitutional debate, using the critique\_plan node and a refinement prompt to have the agents converge on a single, final plan.
* Output: A string containing the final, debated plan in Markdown format.

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

This graph's responsibility is to execute a pre-approved plan.

* Entry Point: execute\_synthesis\_step
* Input State: Must be initialized with the user-approved plan.
* Nodes:
  1. execute\_synthesis\_step: Generates the content for a single item from the plan.
  2. assemble\_draft: Stitches together the content from the working\_memory into a full output draft.
  3. holistic\_review: Critiques the fully assembled draft for coherence and flow.
* Logic:
  1. The graph will loop through the plan items, calling execute\_synthesis\_step for each one.
  2. Once all items are complete, it will call assemble\_draft.
  3. It will then enter a self-correction loop, using holistic\_review to refine the assembled draft up to MAX\_REVISIONS.
* Output: The fully generated and internally reviewed artifact.

### 4. Supervisor and UI Logic

The supervisor.py script will be the state machine for this entire process.

1. Initial Request: The user provides a prompt (e.g., clicks "Generate Research Report").
2. Run Planning Workflow:
   * The supervisor invokes the constitutional planning debate using build\_planning\_graph.
   * It receives the final, converged plan.md string.
3. Human-in-the-Loop Gate:
   * The Streamlit UI displays the plan.md.
   * All processing stops.
   * The UI will present two buttons: "Approve Plan & Begin Synthesis" and "Cancel".
4. Run Synthesis Workflow:
   * If the user clicks "Approve," the supervisor initializes the synthesis\_graph with the approved plan.
   * It runs the synthesis workflow for Agent A and Agent B to get two complete drafts.
   * It orchestrates a final constitutional debate on the two complete drafts.
   * It saves and displays the final, converged artifact.

This new design explicitly builds in the checks and balances that our previous iterations lacked. It makes planning a deliberate, validated, and supervised act, ensuring that all subsequent work is built on a solid and user-aligned foundation.