The Novelist’s Collective

Human-in-the-loop AI system for writing a novel. A central Orchestrator coordinates specialized agents, keeps context tight, and routes work/results through shared storage (Notion + Google Drive) and a vector store for retrieval.

Roles at a glance

Human Author (you): Supplies outline, character seeds, approves direction, revises prose.
Orchestrator Agent: Plans tasks, assembles context, calls sub-agents/tools, tracks versions.
Prose Agent: Writes scene/chapter drafts in target voice; respects outline and continuity.
Character Forge: Builds/updates bios, backstories, motivations, voice/accent, relationships.
The Critic: Reviews outline/plot/drafts; returns structured, actionable notes.
Research Agent: Produces “fact packs” on time/place/industry/culture with citations.

Conceptual Flow (end-to-end)

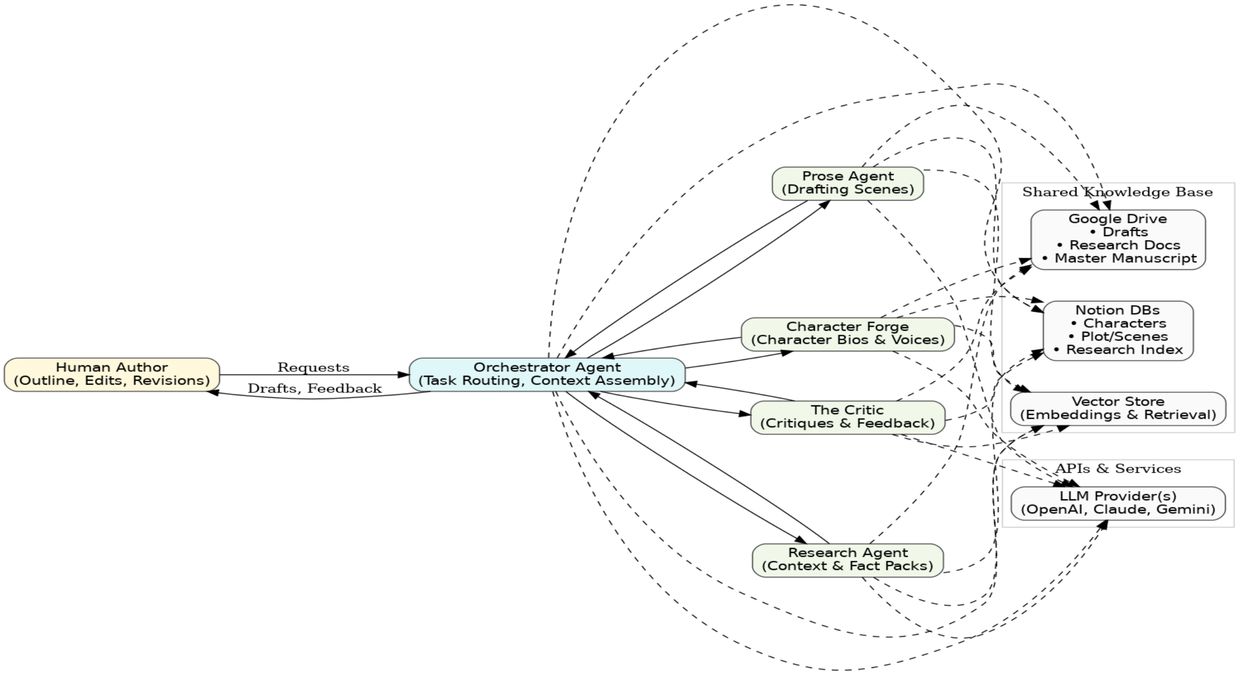
Input → Author creates/updates: outline, scene goals, tone constraints.
Plan → Orchestrator selects tool(s), compiles context from Notion/Drive/Vector store.
Draft → Prose Agent writes a scene (aware of outline + character cards + research).
Critique → The Critic checks voice, pacing, logic, continuity; returns change list.
Revise → Author revises (or Prose Agent revises under author rules).
Persist → Orchestrator commits: drafts to Drive; structured updates to Notion; embeds to vector store.
Loop → Repeat per scene/chapter; maintain status and dependencies in Notion.

Storage & Context Strategy

Structured (Notion)
Characters (DB): name, archetype, desires, fears, virtues/foibles, voice notes, relationships.
Plot/Scenes (DB): act/chapter/scene, summary, beats, POV, setting, included characters, status.
Research Index (DB): topic, sources, quotes, citations, reliability tag, notes, related scenes.
Decisions/Constraints (DB): style rules, banned tropes, continuity decisions, red lines.

Unstructured (Google Drive)
/Drafts/<chapter>.gdoc — rolling drafts per chapter/scene.
/Research/<topic>.pdf|.docx — raw sources, clippings.
/Manuscript/Master.gdoc — the canonical, author-approved manuscript.

Retrieval Layer (Vector Store)
Embed: character cards, scene summaries, prior scene text (chunked), key research notes.
RAG queries by Orchestrator to limit prompts to relevant, recent context.

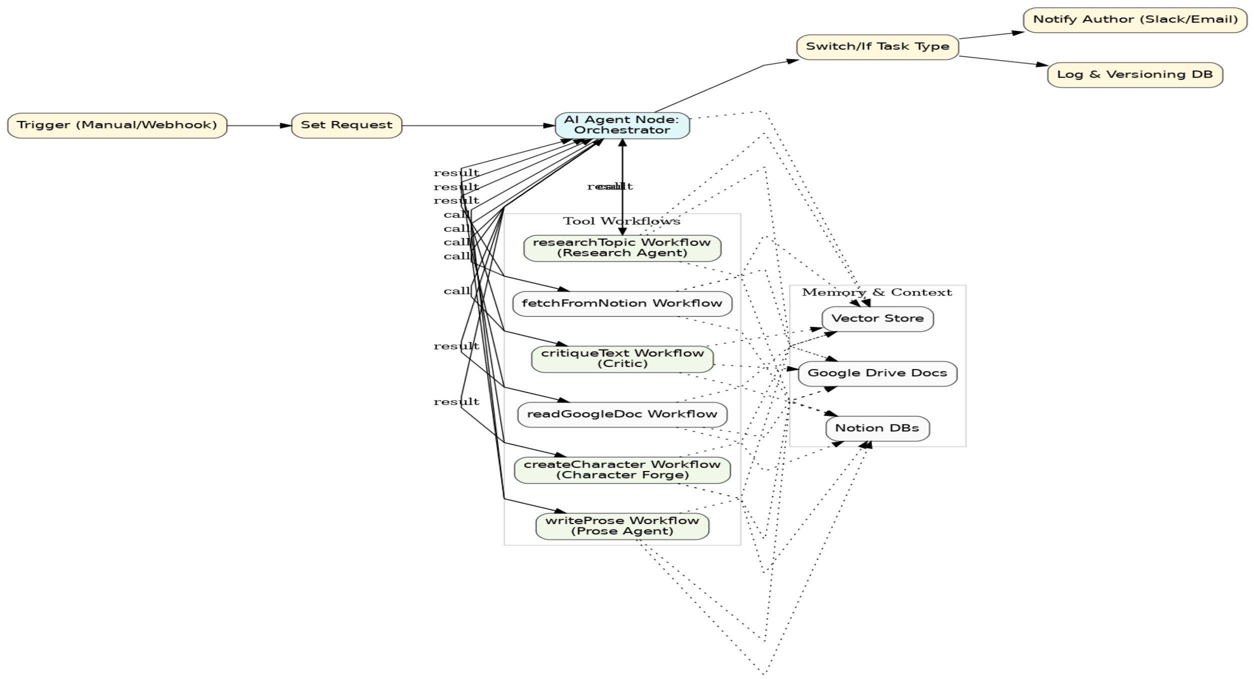


n8n Implementation (agent-tool pattern)

Main workflow: Orchestrator Agent
Trigger: Manual or Webhook (from a simple UI).
Set (Normalize Request): Clean/validate inputs; attach user/project IDs.

AI Agent Node (Orchestrator)
System: “You are the Orchestrator for a novel-writing team. Analyze requests, select tools, assemble minimal, precise context, call tools, validate results, and persist updates. Always keep the author in the loop.”
Tools (each is another n8n workflow): writeProse, critiqueText, createCharacter, researchTopic, fetchFromNotion, readGoogleDoc, (optional) updateVectorStore, queryVectorStore.

Switch/If (Task Type): Route results, handle edge cases (missing context, policy).
Notify: Slack/Email to author with draft + notes.
Log/Version: Append run metadata and hashes for reproducibility.


Trigger (Manual/Webhook)
Set Request
AI Agent Node: Orchestrator
Switch/If Task Type
Notify Author (Slack/Email)
Log & Versioning DB
result
call
Tool Workflows
researchTopic Workflow (Research Agent)
fetchFromNotion Workflow
critiqueText Workflow (Critic)
readGoogleDoc Workflow
createCharacter Workflow (Character Forge)
writeProse Workflow (Prose Agent)
Memory & Context
Vector Store
Google Drive Docs
Notion DBs

Tool workflows (sub-agents)

writeProse
Input: scene goal, POV, style constraints; relevant character cards; last scene chunk(s).
Fetch: fetchFromNotion → Plot/Scenes + Characters; readGoogleDoc → previous text.
LLM (Prose persona). Return: draft + self-check.

critiqueText
Input: draft, outline, character constraints.
LLM (Critic persona). Return: JSON notes + optional diff block.

createCharacter
Input: seed (name, role), relationships, sample dialogue.
LLM (Character Forge). Persist: Notion Characters + embeddings.

researchTopic
Input: topic, scene(s), required specificity.
Tasks: scrape/APIs, summarize, cite. Return: fact pack + glossary.

fetchFromNotion / readGoogleDoc
Utility workflows returning clean JSON/text for prompts.

Configuration Snippets

Notion DB properties (example)

```
{
  "Characters": ["Name","Role","ArcStage","Motivation","Fear","Virtues","Foibles","VoiceNotes","Accent","Relations
  "Scenes": ["Act","Chapter","Scene","POV","Goal","Conflict","Setting","CharactersPresent","Beats","Status","Links"
  "Research": ["Topic","Summary","Citations","Reliability","RelatedScenes","Tags","LastUpdatedISO"]
}
```

Orchestrator tool contract (recommended)

```
// request
{ "tool":"writeProse", "projectId":"...", "sceneId":"...", "context":{...} }
// response
{ "ok": true, "tool":"writeProse", "outputs": { "draft":"...", "checks": {...} }, "persist": [{ "type":"drive.write", ... }, { "type":"
```

Guardrails & QA

Continuity tests: Orchestrator checks character facts/timelines.

Style lints: ban-lists, cadence targets, POV enforcement.

Fact flags: Research Agent tags low-confidence facts.

Versioning: all writes carry run ID + content hash.

Operating the loop (practical)

Start a scene (UI/Slack).

Orchestrator assembles context → writeProse.

Draft returns → sent to critiqueText.

Author receives draft + critique. Accept/modify.

Orchestrator persists final, updates status, schedules next scene.