

# **Living Implementation Blueprint: Second Brain (Alpha)**

Version: 1.0

Date: July 9, 2025

Status: Approved for Implementation

Authored By: Gemini, Senior Software Architect

A comprehensive plan for building an epistemological inquiry engine to empower strategic knowledge workers.

# Contents

<b>1</b>	<b>The Vision &amp; Architecture</b>	<b>2</b>
1.1	Core Philosophy: The 'Truth-Seeking Engine'	2
1.2	The 'Multi-Axis Analysis' Framework	2
1.3	The Core Algorithm: Codified Mental Models	2
1.4	Alpha Operational Mode: "Mode A - Curated Synthesis"	2
1.5	System Architecture: The 'Modular Agent' Workflow	2
<b>2</b>	<b>Product &amp; Feature Requirements (Alpha PRD)</b>	<b>3</b>
2.1	Target Persona for Alpha	3
2.2	Primary Use Case for Alpha: The "Debate Prep" Scenario	3
2.3	Key Feature 1: The Interactive Report	3
2.4	Key Feature 2: The 'Ingestion Sandbox'	4
2.5	Key Feature 3: Liability & Output Framing	4
<b>3</b>	<b>Actionable Implementation Plan</b>	<b>4</b>
3.1	Phase 1: Foundation & Setup (First Week)	4
3.2	Phase 2: Building the Core Engine - The Agent Chain (Next Two Weeks)	4
3.3	Phase 3: Frontend Integration & UI Build (Final Week)	5

# 1 The Vision & Architecture

This section outlines the core philosophy and technical architecture that will guide all development decisions for the alpha version.

## 1.1 Core Philosophy: The 'Truth-Seeking Engine'

The system's primary purpose is not merely to organize information but to function as an epistemological inquiry engine. It must act as a "thought partner" that empowers the user to question established narratives, analyze the evolution of ideas over time, and form high-conviction opinions based on a multi-faceted analysis of evidence. The guiding principle is to "challenge and guide, not just agree."

## 1.2 The 'Multi-Axis Analysis' Framework

Every piece of synthesized knowledge produced by the engine will be evaluated and presented along four distinct axes:

- **The Temporal Axis:** Pinpointing when a piece of information was considered true and tracking how concepts evolve.
- **The Credibility Axis:** Evaluating the source, author, and potential biases of the information.
- **The Coherence Axis:** Analyzing how information supports or contradicts other knowledge within the user's library and the wider world.
- **The Impact/Risk Axis:** Assessing the potential second-order consequences of acting on the information.

## 1.3 The Core Algorithm: Codified Mental Models

The AI's reasoning process will be explicitly architected to use established mental models, including but not limited to First Principles, Second-Order Thinking, and Inversion. This will be achieved through a chained sequence of prompts within the agentic workflow.

## 1.4 Alpha Operational Mode: "Mode A - Curated Synthesis"

The alpha will focus on perfecting a single, powerful mode. The system uses the user's internal library as the contextual core, which is then enriched, validated, and challenged with external data to produce a final, synthesized output.

## 1.5 System Architecture: The 'Modular Agent' Workflow

The system will be built on a flexible, vendor-agnostic, and asynchronous architecture composed of discrete, swappable components:

- **Presentation Layer:** Next.js with Vite, using server-side rendering (.tsx components).
- **API Layer:** Next.js API Routes, handling incoming requests and validating with Zod.
- **Task Queuing Layer:** BullMQ, using a local Redis instance for the alpha, with the architecture allowing a switch to a cloud provider (e.g., Upstash) via configuration.

- **Data Processing Layer (The Worker):** A Node.js/TypeScript process containing a chain of specialized agents that consume jobs from the queue.
- **Data Persistence Layer:**
  - *Primary:* A single Supabase instance with the pgvector extension enabled, accessed via the Prisma ORM.
  - *Swappable Vector DB:* Pinecone will be maintained as a configurable option for vector storage.
- **AI Layer:** Gemini models (Flash/Pro) will be the default, but the agentic design allows for specific agents to be replaced with other services (e.g., Perplexity API) via configuration.

## 2 Product & Feature Requirements (Alpha PRD)

This section details the specific user-facing features to be built for the alpha, focusing on the primary use case.

### 2.1 Target Persona for Alpha

**The Professional Strategist:** A knowledge worker who needs to analyze complex information, defend a position, and build high-stakes arguments (e.g., project managers, consultants, researchers, marketers).

### 2.2 Primary Use Case for Alpha: The "Debate Prep" Scenario

**User Story:** As a Professional Strategist, I want to upload an opposing document (e.g., a competitor's proposal, a critical project review) so that the system can analyze its weaknesses and help me build the strongest possible counter-argument by leveraging my own trusted knowledge base.

**User's Choice of AI's First Move:** The system's 'first move' will be a 'Defensive Analysis,' where it first meticulously deconstructs the opponent's document to find its logical fallacies and weak points before proceeding.

### 2.3 Key Feature 1: The Interactive Report

The primary output of the system will be a dynamic, modular web report composed of the following components:

- **EmbeddedContentViewer:** To view uploaded PDFs or source articles directly in the UI.
- **SourceAttributionWidget:** A visual representation of the sources used in the synthesis (e.g., 60% Opposing Document, 40% Your Library).
- **InteractiveCitation:** Clickable footnotes that reveal the exact source text and link.
- **ConfidenceScoreIndicator:** A gauge displaying the system's confidence in its analysis.
- **KeyFindingsPanel:** Sections for "Logical Fallacies Found," "Unsubstantiated Claims," and "Points of Agreement."

## 2.4 Key Feature 2: The 'Ingestion Sandbox'

External sources found during analysis will not be automatically added to the user's library. They will be presented in a "Review Sandbox." The Sandbox UI will provide a comprehensive 'pre-flight analysis report' for each item, including a credibility score and a summary of how it supports or contradicts existing knowledge. The user will have simple "Add to Library" or "Dismiss" actions for each item, keeping them in full control of their knowledge base.

## 2.5 Key Feature 3: Liability & Output Framing

All generated output will be explicitly framed as an "AI-generated opinion for review" and will include standard disclaimers. The AI's persona will be trained via system prompts to use analytical and suggestive language ("This suggests...", "One potential weakness is...") rather than prescriptive commands ("You should...").

# 3 Actionable Implementation Plan

This is the granular, step-by-step plan for building the alpha version.

## 3.1 Phase 1: Foundation & Setup (First Week)

### [ ] Task 1.1: Database Setup:

- Confirm the latest prisma.schema is in place.
- Write and execute a SQL script in Supabase to enable the pgvector extension.
- Run prisma migrate dev to synchronize the database with the schema.
- Run prisma generate to update the Prisma Client.

### [ ] Task 1.2: Configuration:

- Create a central config/system.ts file.
- This file will export configured instances of the Prisma client, Redis client, and AI clients, reading all keys and URLs from the .env file. This ensures all services are initialized from a single, configurable source.

### [ ] Task 1.3: Worker Scaffolding:

- Create the main worker entry point at /app/worker/index.ts.
- This script will connect to the BullMQ defined in lib/queue.ts and set up a processor for the "process-resource" job type.

## 3.2 Phase 2: Building the Core Engine - The Agent Chain (Next Two Weeks)

This phase focuses on implementing the modular agents within the Queue Worker.

### [ ] Task 2.1: Implement the 'Ingestion Agent':

- This agent receives the resourceUrl from the job.
- It will be responsible for fetching the content (e.g., extracting text from a PDF, scraping an article).

- It passes the raw text to the next agent.

[ ] **Task 2.2: Implement the 'Defensive Analysis Agent':**

- Receives the raw text of the opposing document.
- Uses a specific Gemini prompt to analyze the text for logical fallacies, weak arguments, and unsubstantiated claims.
- Passes the original text and this analysis forward.

[ ] **Task 2.3: Implement the 'Internal Coherence Agent':**

- Receives the analysis from the previous agent.
- Performs a pgvector similarity search on the user's existing library to find the most relevant internal documents.
- Uses another Gemini prompt to compare the opposing document's claims with the user's internal knowledge, identifying points of confirmation and contradiction.

[ ] **Task 2.4: Implement the 'Synthesis & Opinion Agent':**

- This is the final agent in the chain. It receives all prior analyses.
- It uses the master "consultant" prompt to synthesize all information into the final, structured "opinion."
- It generates the data for all Interactive Report components (confidence scores, source attribution, etc.).

[ ] **Task 2.5: Implement the 'Persistence Agent':**

- Receives the final structured output.
- Saves the data to the appropriate Supabase tables (*content\_summaries, etc.*) via *Prisma.Cre*

- Updates the master BatchJob status to "completed."

### 3.3 Phase 3: Frontend Integration & UI Build (Final Week)

[ ] **Task 3.1: Build the "Debate Prep" UI:** Create the page/component where a user can upload a document and initiate the analysis.

[ ] **Task 3.2: Build the Interactive Report Viewer:** Create the parent component for the report and all the modular child components (SourceAttributionWidget, etc.). This component will fetch and render the data generated by the worker.

[ ] **Task 3.3: Build the "Ingestion Sandbox" UI:** Create the interface for reviewing and approving/dismissing external sources discovered during analysis.

[ ] **Task 3.4: Final Wiring:** Connect all UI components to the relevant backend APIs and ensure the end-to-end data flow is functional. Complete the .jsx to .tsx migration.

This blueprint provides a comprehensive and actionable plan to deliver the alpha version of the Second Brain application, fully aligned with the sophisticated vision we have developed.