Vigil.ai

The Multi-Modal Therapeutic Council & Proactive Sentinel System

Ishan Singh • **Daksh Dadeech**

January 22, 2026

---

1 Executive Summary

Vigil.ai represents a paradigm shift in therapeutic Artificial Intelligence, designed to solve the critical "Goldfish Memory" problem inherent in standard Large Language Models (LLMs). Unlike reactive chatbots (e.g., ChatGPT) that reset context with every session, Vigil.ai maintains a persistent, clinically relevant **Memory Core**. It employs a novel **Multi-Agent "Council"** architecture to analyze user input through distinct psychological lenses—Empathy, Logic, and History—simultaneously, ensuring a balanced and human-like response.

Crucially, Vigil.ai introduces the concept of **Proactive Digital Care**. Utilizing a background asynchronous process known as "The Sentinel," the system monitors user well-being even when the application is closed. By analyzing historical memory data for distress patterns or "open loops" (e.g., missed high-stakes events like exams or surgeries), Vigil.ai autonomously dispatches context-aware "nudge" notifications via email. This transforms the AI from a passive tool into an active, always-on guardian of mental well-being.

2 System Architecture Overview

The system operates on a **Hub-and-Spoke** model. A central Orchestrator manages the data flow between the User Interface, the Cognitive Council, the Memory Core, and the asynchronous Safety Layer. This design ensures modularity, allowing individual agents to be upgraded without disrupting the core loop.

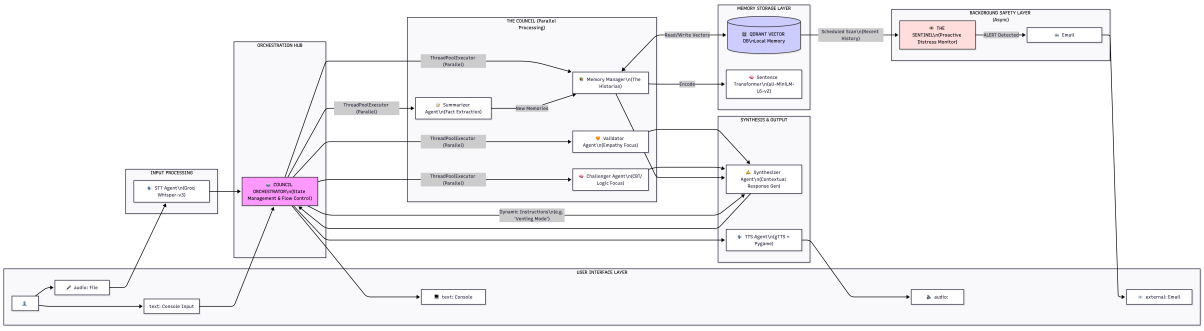

Figure 1: Vigil.ai System Architecture Flowchart

The architecture is divided into four primary layers:

1. **Input/Output Layer:** Handles multi-modal data (Text/Voice).
2. **Orchestration Layer:** Manages state and routes tasks.
3. **Cognitive Layer (The Council):** Performs parallel analysis.
4. **Safety Layer (The Sentinel):** Monitors long-term trends.

3 Detailed Component Analysis

3.1 3.1 The User Interface Layer (Multi-Modal)

The system lowers the barrier to entry for users in distress through seamless multi-modal support, ensuring accessibility for those unable to type.

- **Input Normalization:** The system accepts dual input streams. Standard text is passed directly to the orchestrator. Audio files are intercepted by the **STT Agent**, powered by **Groq's Whisper-large-v3**. This model was chosen for its ultra-low latency and high fidelity in capturing emotional nuance and hesitation in speech.
- **Dual Output:** Responses are delivered as text for readability and simultaneously converted to speech by the **TTS Agent** (using `gTTS` and `pygame`). This creates an immersive, hands-free session akin to a phone call with a therapist.

3.2 3.2 The Vigil Orchestrator

The **Vigil Orchestrator** acts as the central nervous system. It does not generate content but manages state and execution flow to ensure coherent therapy.

- **State Detection Algorithms:** The Orchestrator analyzes metrics such as *Turn Count* (session duration) and *Input Token Length* to classify the user's current psychological state.

- **Dynamic Instruction Injection:** Based on the state (e.g., "Venting" vs. "Probing"), it injects meta-instructions into the Council. For example, if "Venting" is detected, it instructs the Synthesizer to *"suppress logical challenges and maximize validation."*

3.3 3.3 The Cognitive Council (Parallel Processing)

To emulate complex human reasoning, Vigil.ai utilizes Python's `ThreadPoolExecutor` to run four specialized agents simultaneously. This reduces latency while maximizing analytical depth.

- **The Validator (Empathy Engine):** A specialized instance prompted to ignore logic. Its sole function is to reflect the underlying emotion (e.g., fear, exhaustion) to build rapport.
- **The Challenger (CBT Logic Engine):** A "Cognitive Behavioral Therapy" specialist. It scans the input for cognitive distortions (e.g., *Catastrophizing*, *All-or-Nothing Thinking*) and offers analytical rebuttals.
- **The Historian (Context Manager):** Interfaces with the Memory Core. It performs a **Hybrid Retrieval** (Vector + Keyword) to find relevant past interactions that provide context to the current statement.
- **The Summarizer (Memory Encoder):** An extraction agent that runs in the background. It strips away conversational filler and condenses the user's input into concise "Memory Keys" for efficient storage.

3.4 3.4 The Memory Core ("The Hippocampus")

Vigil.ai abandons standard JSON storage for a **Hybrid Vector Database** powered by **Qdrant**. This ensures the AI remembers *concepts*, not just words.

- **Semantic Vector Search:** User text is embedded into a 384-dimensional vector space using `SentenceTransformer('all-MiniLM-L6-v2')`. This allows the system to understand that "I feel blue" is semantically related to a past memory of "User felt depressed," even if the words differ.
- **The "Rescue" Mechanism:** A custom lexical fallback ensures specific entities (names, medications, places) are retrieved even if vector similarity is low, preventing the "hallucination" of personal details.

4 Background Safety Protocols: The Sentinel

Vigil.ai includes an asynchronous safety layer that operates independently of the active chat session, acting as a fail-safe mechanism.

1. **Scheduled Scanning:** A background process wakes up periodically to execute a "Scroll" operation on the Qdrant database, retrieving the 10 most recent memory entries.
2. **The Diagnosis:** These memories are fed into a "Clinical Supervisor" LLM which analyzes the chronological trend for:
   - **High Distress Patterns:** Rapidly escalating negative sentiment.
   - **Open Loops:** Mentions of high-stakes events (e.g., "Surgery tomorrow") that lack a subsequent "resolution" memory.
3. **The Intervention:** If an alert is triggered, the **Email Dispatcher** utilizes `smtplib` to generate an HTML-formatted email. The content is dynamically drafted to specifically reference the detected trigger (e.g., *"Just checking in on how the exam went..."*), bridging the digital-physical divide.

5 Technology Stack Summary

Component	Technology	Purpose
LLM Inference	Groq API	Ultra-fast inference (¡0.5s) for Llama-3.1 and Whisper.
Orchestration	Python Futures	Managing parallel agent execution.
Vector Database	Qdrant (Local)	Storing and retrieving semantic memories in RAM.
Embeddings	SentenceTransformers	`all-MiniLM-L6-v2` for text-to-vector conversion.
Speech-to-Text	Whisper-large-v3	High-accuracy audio transcription.
Text-to-Speech	gTTS & Pygame	Audio generation and playback.
Notifications	SMTP (`smtplib`)	Sending HTML-formatted proactive emails.

Table 1: Vigil.ai Technical Stack

6 Key Differentiators

- **Memory Permanence:** Unlike standard bots, Vigil.ai remembers specific details (names, events) across sessions indefinitely.
- **Proactive vs. Reactive:** Vigil.ai reaches out to the user when they spiral or go silent; standard bots only respond when spoken to.
- **Multi-Modal Accessibility:** Full voice support lowers the barrier for users in acute distress who may find typing difficult.