**TITLE OF THE INVENTION**

**KENPIRE COGNITIVE ENGINE: Cognitive Infrastructure for Local-First Agentic Systems with Modular Execution, Persistent Memory, and Trifecta LLM Coordination-----**

**BACKGROUND OF THE INVENTION**

**The field of the invention relates generally to artificial intelligence (AI) systems, particularly those involving multi-agent architectures, persistent memory, and the orchestration of large language models (LLMs). Current agentic systems often lack modular portability across diverse hardware, robust crash-safe recovery, structured long-term memory access, and a reliable consensus mechanism for integrating multiple proprietary LLM services. There is a need for a unified, local-first framework that provides a complete cognitive substrate for persistent, intelligent, and modular AI agents.-----BRIEF SUMMARY OF THE INVENTION**

**The KenPire Cognitive Engine (KCE™) is a novel, multi-layer orchestration framework providing a cognitive substrate for persistent, intelligent, and modular AI agents operating across hardware and software interfaces. Key inventive aspects include the self-contained, portable AI workflows (Capsules); the crash-safe execution kernel (KenFlow™); the consensus-based multi-LLM dispatch system (Trifecta Bridge™); and the tiered, structured memory architecture (Dual RAG Brain™) utilizing a specialized data format (TOON). The system further incorporates cryptographic integrity and IP security through the ProofLock™ and ClauseWitch™ components.-----DETAILED DESCRIPTION**

**The KenPire Cognitive Engine (KCE™) is a multi-layer orchestration framework configured to enable persistent, intelligent, and modular AI agents for robotics, virtual agents, and general modular workflows.**

1. **Capsule Architecture and OS Portability: The core of the system is the Capsule, a self-contained and portable AI workflow. Each Capsule bundles the necessary code, data, memory components, and defined agent traits into a single unit. Capsules are designed for maximum portability, running on diverse operating systems including Linux, macOS, and Windows (via WSL), as well as on resource-constrained devices like Raspberry Pi. Runtime compression is achieved via CBOR/zlib, yielding up to a 41% reduction in size. Capsules also incorporate a Token-Oriented Object Notation (TOON) format for structured data and memory infusion.**
2. **KenFlow Orchestrator (Capsule Kernel): The KenFlow™ Orchestrator serves as the Capsule runtime manager. It implements a crash-safe**

**recovery mechanism that utilizes logged execution data and an automatic recovery script (autonomous_restore.sh) to restore the Capsule state. The kernel supports a unique "Destructive Mode" for controlled teardown and rebuild workflows. The orchestrator exposes a user interface (Commander UI), a backend (FastAPI), and monitoring ports (5173–8080). Agent-to-agent communication is facilitated by the A2A Protocol, a native Redis-based messaging layer.**

3. **Trifecta Bridge™ Multi-Agent Dispatch: The Trifecta Bridge™ is a consensus-based LLM routing system that routes prompts and tasks across multiple distinct large language models (LLMs), such as GPT-5, Claude, and Gemini. This dispatching allows for enhanced task accuracy by utilizing fallback, arbitration, or voting strategies within consensus loops to determine the most reliable output.**
4. **Dual RAG Brain Memory Model: The system employs a tiered memory design: a short-term memory for recent context and a long-term structured memory managed via Redis and a VectorDB. Memory recall is not merely retrieval but is actively infused into LLM prompts using the TOON format, ensuring context persistence and relevance. The architecture further supports local mesh memory across distributed agent environments.**
5. **Proof and IP Security: Intellectual property and integrity are maintained by the ProofLock™ ledger, which cryptographically hashes every Capsule and execution run. The ClauseWitch™ component is embedded to enforce IP restrictions and apply memory watermarking. Agents within the system run with cryptographic signatures, and logs are secured in the /witness/ directory.**
6. **Supporting Components and Features: The system includes a Gameboard UX Layer, a visual tile-based dashboard showing Capsule progress, integrated XP, ProofLock, and unlock systems. For educational applications, the system includes TEKS RAG Bot™ (parsing state standards into JSON), GraphRAG (mapping knowledge progression), and CrosswalkBot (aligning multiple state standards). Manifested Personas like the Dirty Rag Bot (code janitor) and Jarvess (multi-modal coach) are defined in YAML manifests with distinct logic layers.**

**-----CLAIMS**

1. **A system for autonomous agent orchestration, comprising:**
   **a multi-layer execution kernel (KenFlow™) configured to manage and execute self-contained, portable AI workflows (Capsules);**
   **a memory architecture (Dual RAG Brain™) comprising a short-term context memory and a long-term structured memory, wherein memory recall is infused into agent prompts using a Token-Oriented Object Notation (TOON) format; and**

**a multi-agent dispatch unit (Trifecta Bridge™) configured to dynamically route a single task or prompt across a plurality of distinct large language models (LLMs) and determine a final output based on a consensus-based strategy.**

2. **The system of claim 1, wherein the multi-agent dispatch unit is configured to utilize at least one of a fallback strategy, an arbitration strategy, or a voting strategy to achieve consensus across the plurality of LLMs.**
3. **The system of claim 1, wherein the multi-layer execution kernel further comprises a crash-safe recovery mechanism configured to log and auto-recover a Capsule's state.**
4. **The system of claim 3, wherein the crash-safe recovery mechanism supports a "Destructive Mode" for workflow teardown and rebuild.**
5. **The system of claim 1, wherein each Capsule is a self-contained unit that bundles code, data, the Dual RAG Brain memory, and agent traits for execution across operating systems including Linux, macOS, and Windows.**
6. **The system of claim 5, further comprising a compression module configured to reduce Capsule runtime size via CBOR/zlib compression.**
7. **The system of claim 1, further comprising a cryptographic logging system (ProofLock™) configured to generate a secure ledger hash of every Capsule execution and run, thereby creating a verifiable behavioral trail.**
8. **The system of claim 7, further comprising an IP enforcement module (ClauseWitch™) configured to embed intellectual property restrictions and memory watermarking within the Capsules.**
9. **The system of claim 1, further comprising a communication protocol (A2A Protocol) for native Redis-based agent-to-agent messaging.**
10. **The system of claim 1, further comprising an education innovation component configured to align knowledge with state standards, wherein the component includes a CrosswalkBot for aligning multiple state standards (TEKS, CCSS, NGSS, local standards).**
11. **The system of claim 10, wherein the education innovation component includes a TEKS RAG Bot™ configured to parse state standards into JSON format.**
12. **The system of claim 1, further comprising a visualization layer (Gameboard UX Layer) that displays a visual tile-based dashboard of Capsule progress, integrating status, a ProofLock trail, and experience (XP) metrics.**

**-----ABSTRACT OF THE DISCLOSURE**

**A multi-layer orchestration framework, the KenPire Cognitive Engine (KCE™), enables the operation of persistent, intelligent, and modular AI agents across various hardware and software interfaces. The system acts as a cognitive**

**substrate for robotics, virtual agents, and modular workflows by integrating a Dual RAG Brain™ memory architecture for tiered, structured memory; a modular execution kernel called KenFlow™; and the Trifecta Bridge™ for consensus-based multi-Large Language Model (LLM) dispatching. AI workflows are contained within portable, self-contained units called Capsules, which bundle code, data, memory, and agent traits. The system also features a ProofLock™ behavioral logging and cryptographic watermarking system for IP security and crash-safe recovery mechanisms. The core technology provides a local-first automation kernel with verifiable agentic processes.**