🌌 SeedCore: Engineering Deep Dive

A Cognitive Operating System for Distributed, Self-Evolving Intelligence

SeedCore is a cognitive operating system built on **Kubernetes + Ray**, designed around a persistent **cognitive organism** capable of real-time adaptation, structural reasoning, and autonomous self-repair. It unifies distributed systems engineering, cognitive modeling, and biological metaphors into a single architecture grounded in **control theory**, **energy dynamics**, and **formal routing contracts**.

This document provides an engineering-deep analysis of all major components, including the new **Holon Memory Fabric**, which brings human-like memory behavior with provable stability guarantees.

---

1. 🧊 Planes of Control Architecture

SeedCore avoids monolithic agent runtimes and instead uses a strict **Planes of Control** model that separates cognition, strategy, execution, and raw computation.

```
Cognitive (Brain)
↑
Control (Strategy Cortex)
↑
Execution (Distributed Reflex Agents)
↑
Infrastructure (Math + Models)
```

---

2. 🧠 Intelligence Plane — DSPy Cognitive Core v2

The Cognitive Service acts as a **Brain-as-a-Service**, providing domain reasoning, planning, retrieval, and structural anomaly detection.

Key capabilities:

- **Fact schema w/ provenance + trust**
- **RRF + MMR optimized retrieval**
- **Dynamic token budgeting informed by OCPS drift**
- **HGNN structural reasoning** for root-cause analysis
- **Post-condition validation** for safe reasoning
- **Cache governance w/ TTL-per-task-type**
- **Escalation hints** (but never routing)
- **Lie Group Capability Models** (see Section 4)

Invariant:
Cognitive Service never picks agents, routes tasks, or maintains private memory.

---

3. 🎮 Control Plane — Coordinator (Strategic Cortex)

The Control Plane governs **global behavior**, decides **Fast vs Deep path**, and orchestrates the **Plan → Execute → Audit** loop.

Core mechanisms:

- **OCPS (Online Change-Point Sentinel)**
  Surprise = Information entropy on incoming stimuli
- **Policy Knowledge Graph (PKG)**
  Graph-based policies for routing, constraints, safety
- **Plan → Execute → Audit loop**
  Produces structured subplans for the organism

- **Decision governance**
  Never executes tools or holds long-lived state

Invariant:
Coordinator never executes work; it delegates and governs.

---

4. ⚙️ Infrastructure Plane — Mathematical Substrate

Provides the physics behind cognition:

- **XGBoost Engines** for regime detection
- **Drift Detectors** for feature distribution monitoring
- **HGNN** for hypergraph-based retrieval
- **Lie Group Capability Evaluation Pipeline**

4.1 Group-Theoretic Capability Measurement

SeedCore embeds text embeddings into **Lie groups** to measure structural “normality” and capability boundaries.

Pipeline:
`Embedding → PCA → so(m) → SO(m) → Tangent Space → Gaussian Model → Semantic Score`

This model quantifies:

- domain capability
- explanation quality
- dataset / distill-set fitness
- anomaly detection

Mathematically grounded, domain-agnostic, and provably stable.

---

5. ⚡ Execution Plane — Organism Service (Distributed Reflex Layer)

Execution is handled by persistent Ray actors representing **agents**.

Agents handle:

- Local reflex execution (System 1)
- Tool calls with RBAC enforcement
- Salience scoring
- Specialization broadcasting
- Sticky session affinity (`agent_tunnel`)

Agents do *not* handle:

- Routing
- Global reasoning
- Cross-agent coordination
- Memory governance

Invariant:
Agents execute; they do not decide.

---

6. 🌐 Unified TaskPayload Contract (JSONB)

All cognition, routing, memory signals, and telemetry are described through a **schema-less, contract-first envelope** stored in JSONB.

Envelopes:

- `params.risk` — canonical high-stakes classification
- `params.routing` — router inbox
- `params.interaction` — agent-tunnel, coordinator-routed, or one-shot
- `params.cognitive` — LLM provider, decision kind, cog type
- `result.meta.routing_decision` — router output
- `result.meta.exec` — execution metrics

No schema churn.
All evolution happens via JSONB path indexes.

---

7. 🧬 Holon Memory Fabric

A Control-Theoretic, Human-Like Memory System for SeedCore

The Holon Memory Fabric extends SeedCore with **human-like memory behavior**, formally grounded in **control theory** and the global energy functional:

> **$E(s_t)$** — the organism's global energy state
> Memory is part of the organism state, not a subsystem.

This ensures **provably stable**, contractive memory operations that do not amplify entropy.

7.1 Core Principles

1. Control-Theoretic Stability

All memory operations behave as **contractive maps**:
`||f(x) - f(y)|| ≤ k ||x - y|| with k < 1`

This ensures:

- no error amplification
- no runaway memory
- guaranteed convergence

2. Unified State & Energy Model

Memory operations are embedded into the organism's global energy functional:

```
E_total = E_compute + E_memory + E_latency + E_drift + ...
```

Every memory action has energy cost:

- consolidation
- recall
- compression
- forgetting

System dynamics naturally push the organism toward **low-energy wells** (stable states).

3. Human-Like Behavior via Energy Minimization

Human-like memory phenomena (forgetting, consolidation, associative recall) emerge from energy minimization, not heuristics.

7.2 Memory Tiers

Tier	Type	Holon Enhancement
L0	Organ-local	Hot-item prewarming based on meta-controller predictions
L1	Node cache	Shared volatile on-node tier
L2	SharedCacheShard	Atomic ops for cluster-wide single-flight sentinels
Mw	Working Memory	4–8× effective capacity via meta-adaptive compression (VQ-VAE)
Mlt	Long-Term Memory	Durable semantic store; consolidation target
Mfb	Flashbulb Memory	Salience-gated storage for rare, high-impact events
Ma	Agent Private Memory	128-D EWMA-based identity embedding

7.3 Holon Cognitive Memory Processes

1. Encoding & Consolidation

- Write-through from Mw → Mlt
- Mfb stores high-salience outliers
- Consolidation uses **scheduled gradient descent** (sleep-replay)
- Consolidation frequency $\gamma(t)$ decreases during drift for stabilization

2. Recall — Hierarchical + Associative

- Fast path: L0 → L1 → L2
- Deep path: Mlt → Mfb

- Associative recall via HGNN from system state `h_system`
  → enables *reminding* behavior

3. Forgetting — Value-Weighted Decay

- TTL is replaced by retention proportional to:
  `(TD-priority × execution_utility)`
- Meta-controller sets decay curriculum κ
- Always satisfies global freshness guarantee
  `Δt_stale ≤ 3s`

4. Compression — Meta-Adaptive Capacity Expansion

- VQ-VAE or vector-quantization model
- Lipschitz constraint:
  `||Dec||_Lip ≤ 1`
  ensures stability
- Compression cost β_mem is a term in global energy
- Meta-controller throttles capacity vs compute

7.4 Memory Meta-Controller

The Tier0 memory managers evolve into a unified **Memory Meta-Controller**.

Responsibilities:

- Set consolidation cadence (γ)
- Set forgetting decay schedule (κ)

- Adjust compression throttle (β_mem)
- Trigger hot-item prewarming
- Maintain energy stability across all memory tiers

Operates on **∇E (energy gradient)**, not fixed rules.

---

8. Summary Table

Feature	Old System	Holon Memory Fabric
Control Model	Heuristics	Control-theoretic energy minimization
Consolidation	Background cron	Adaptive, drift-aware consolidation
Forgetting	Fixed TTL	Value-weighted selective decay
Recall	Hierarchical	Hierarchical + associative (HGNN)
Capacity	Fixed	4–8x via contractive compression
Tuning	Manual	Self-tuning meta-controller

---

9. Conclusion

The Holon Memory Fabric elevates SeedCore from a high-performance distributed cognitive system to a **human-like, control-theoretically stable memory organism**. It enables:

- predictable dynamics
- self-tuning behavior
- continuous adaptation
- safe, contractive cognition
- structural generalization

- domain-level capability measurement

SeedCore now has a **closed adaptive cognitive loop** where memory, reasoning, routing, and execution co-evolve under a unified theoretical model.