AgentOS Memory Architecture at a Glance

Neutral context for Craig's review | August 10, 2026

The ask: We are looking for candid architecture advice, not approval of a finished design. A memory SOP is currently live so work is governed while we learn; it is explicitly open to replacement or revision after this review.

PAGE 1 — Environment, scale, and current flow

What this system is

AgentOS is Sean's internal research and operations environment: one coordinating AI-agent process, one more-independent reviewing process, and four bounded specialist worker processes across six computers. It is not authorized as a client-production or regulated-data platform. The memory architecture review is preventive, not a response to a known breach or data-loss incident.

Current scale snapshot

- 108 Markdown notes in the private canonical vault; 33 notes in the sanitized shared view.
- 13 dated local daily-memory files and roughly 130 workspace deliverable/evidence files.
- 7 files currently in the reviewer-submission area and 2 approved shared-memory files.
- Promotion volume is currently low and has not yet been measured systematically.
- Four worker nodes are connected; one coordinator and one independent reviewer handle synthesis and review.

Current flow — plain engineering terms

Bounded workers (4) → task results and evidence → coordinator/review gate → reviewed institutional knowledge.

Independent reviewer (1) → candidate additions through a bounded shared lane → coordinator/review gate.

Project repositories remain authoritative for code and versioned implementation files. A private Markdown notes vault, viewed through Obsidian, currently holds reviewed decisions, procedures, project state, evidence links, and durable lessons. Search indexes, embeddings, graphs, and generated answers would be rebuildable derivatives, not authority.

Current trust model

One coordinator process has sole agent write/promotion authority for shared approved memory. The independent reviewer can propose but cannot self-approve. Worker output is treated as a claim until checked against evidence. Sean remains final authority for consequential decisions.

PAGE 2 — Evolution, constraints, alternatives, and known gaps

Compressed evolution

- **July 20:** coordinator/reviewer roles established; manual bounded file handoffs.
- **July 22–23:** repositories/files identified as project truth; draft closeout, transcript, and reviewed-memory rules created.
- **July 25–26:** first scoped request/response handoff succeeded; Node 1 Obsidian designated canonical; evidence and restart-point closeouts formalized.
- **August 3–5:** sanitized end-to-end encrypted shared view implemented; read/response/submission acceptance test passed; reviewer proposals separated from approved memory.
- **August 7–9:** three optional layers clarified—dated evidence, reusable knowledge, distilled current memory; raw logs do not automatically become long-term truth.
- **August 10:** one fleet SOP consolidated the procedure; daily 04:30 full and 15:00 light reviewer checkpoints added; the SOP is live but expressly revisable after Craig's input.

Constraints we care about

- Keep private, customer, banking, email, credential, and raw-transcript data out of the shared memory lane.
- Preserve human readability, provenance, correction history, and an exact path back to source evidence.
- Avoid six competing memory authorities and avoid making Sean a manual courier.
- Keep maintenance realistic for one primary operator.
- Make recovery and rollback testable; sync alone is not backup or permission enforcement.
- Keep operational burden proportionate to the actual value and risk; use measurement to justify either added structure or deliberate simplicity.

Architectures considered — no selection implied

1. Human-readable files/Obsidian with disciplined metadata and backup.
2. Files plus Git-based change control and rollback.
3. Files plus rebuildable hybrid keyword/vector retrieval.
4. Structured relational or append-only event storage with human-readable publication.
5. A graph layer for dependency, lineage, or authority queries.
6. A hybrid of these layers with explicit ownership and rebuild boundaries.

Verified or observed internally

The private and shared vaults exist; proposed and approved lanes are separate; the independent reviewer completed a synthetic read/response/submission acceptance test; four workers currently connect; and a candidate packet was recovered without self-promotion. These are internal observations, not external validation.

Known gaps

- No fleet-wide backup/restore exercise has been completed.
- Repeated scheduled-memory collection and maximum-deferral behavior are still being reliability-tested.
- There is no automated detection of unique unsaved session knowledge.
- Audit evidence remains split among Markdown, logs, session records, and deliverables; there is no unified append-only event ledger.
- Growth, promotion frequency, retrieval accuracy, and operator-maintenance time are not yet measured consistently.

What we want from Craig

Recommend the architecture you would actually operate for the next 6–12 months—whether simpler or more structured than the current design. Identify the three largest risks and tell us what evidence or thresholds should trigger a different database, Git workflow, vector/RAG layer, or graph.