Context Engineering & Reliability

Architectural patterns for scaling context windows in long-running LLM agents.



Context Budgets

Compaction

JIT Retrieval

Subagent Isolation

State vs. Memory

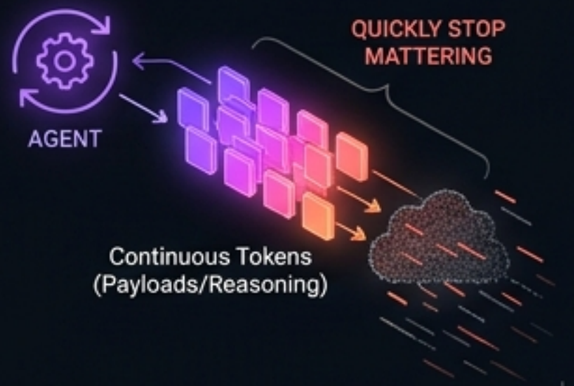
The Stateless Conundrum and Context Rot



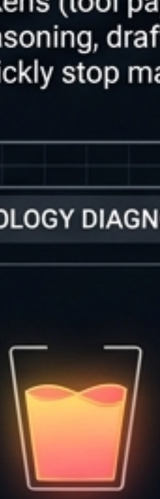
CORE CONCEPT

A model is strictly stateless. At each step, it sees only what the harness placed in the window.

Left alone, an agent in a loop produces continuous tokens (tool payloads, reasoning, drafts) that quickly stop mattering.



PATHOLOGY DIAGNOSTIC FLOW



1. WINDOW FILLS

The agent approaches the hard token limit.



2. RECALL DROPS (CONTEXT ROT)

Recall of specific details degrades gradually as total tokens grow—well before the hard limit is hit.



3. INSTRUCTION DRIFT

The agent slowly forgets early constraints and drifts from its core instructions.

RESULT: SYSTEM INSTABILITY & FAILURE

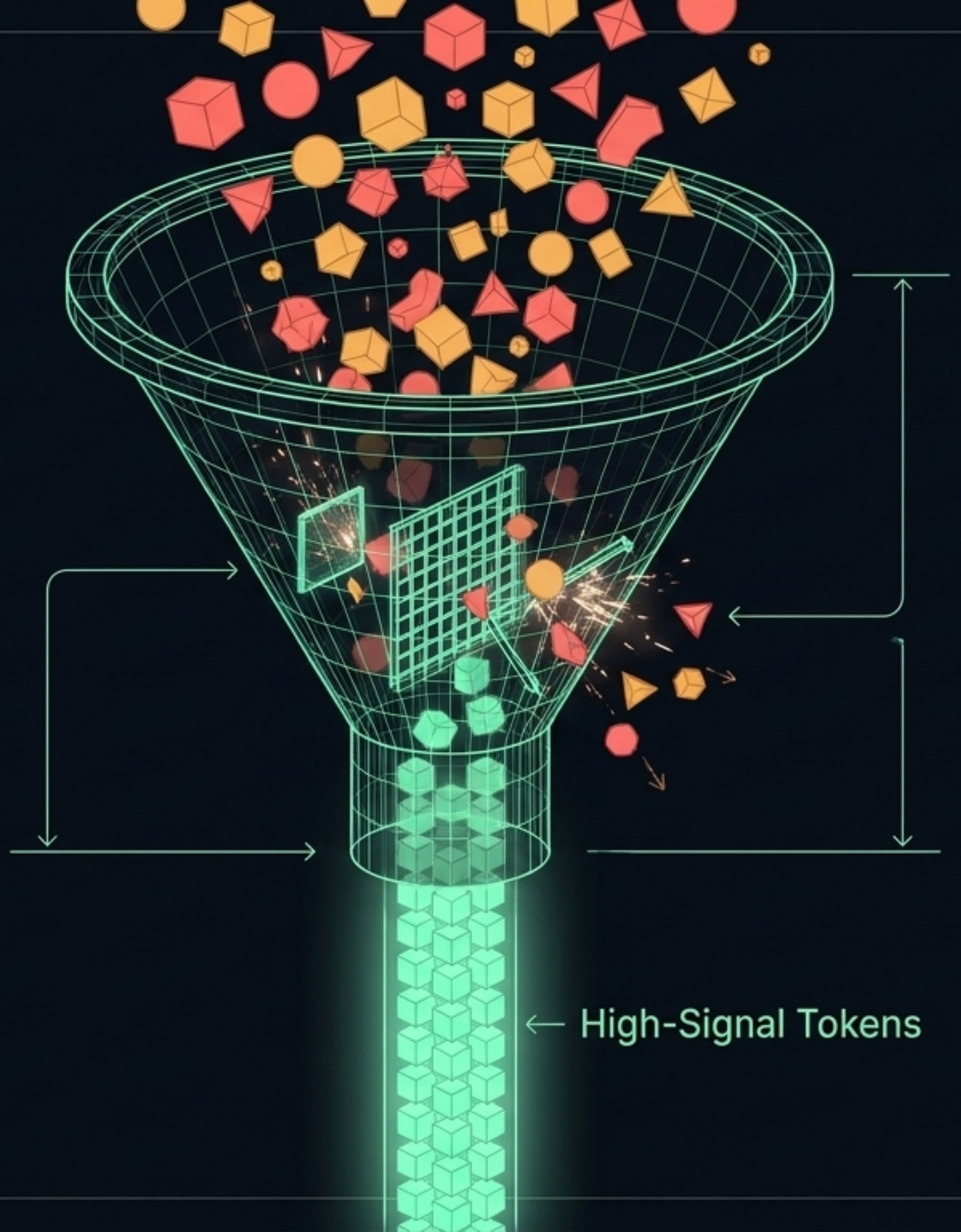
Defining the Per-Turn Discipline

Definition:

Context engineering is the per-call discipline of choosing which tokens enter a model's finite context window, in what form, in what order, and what is moved outside the window.

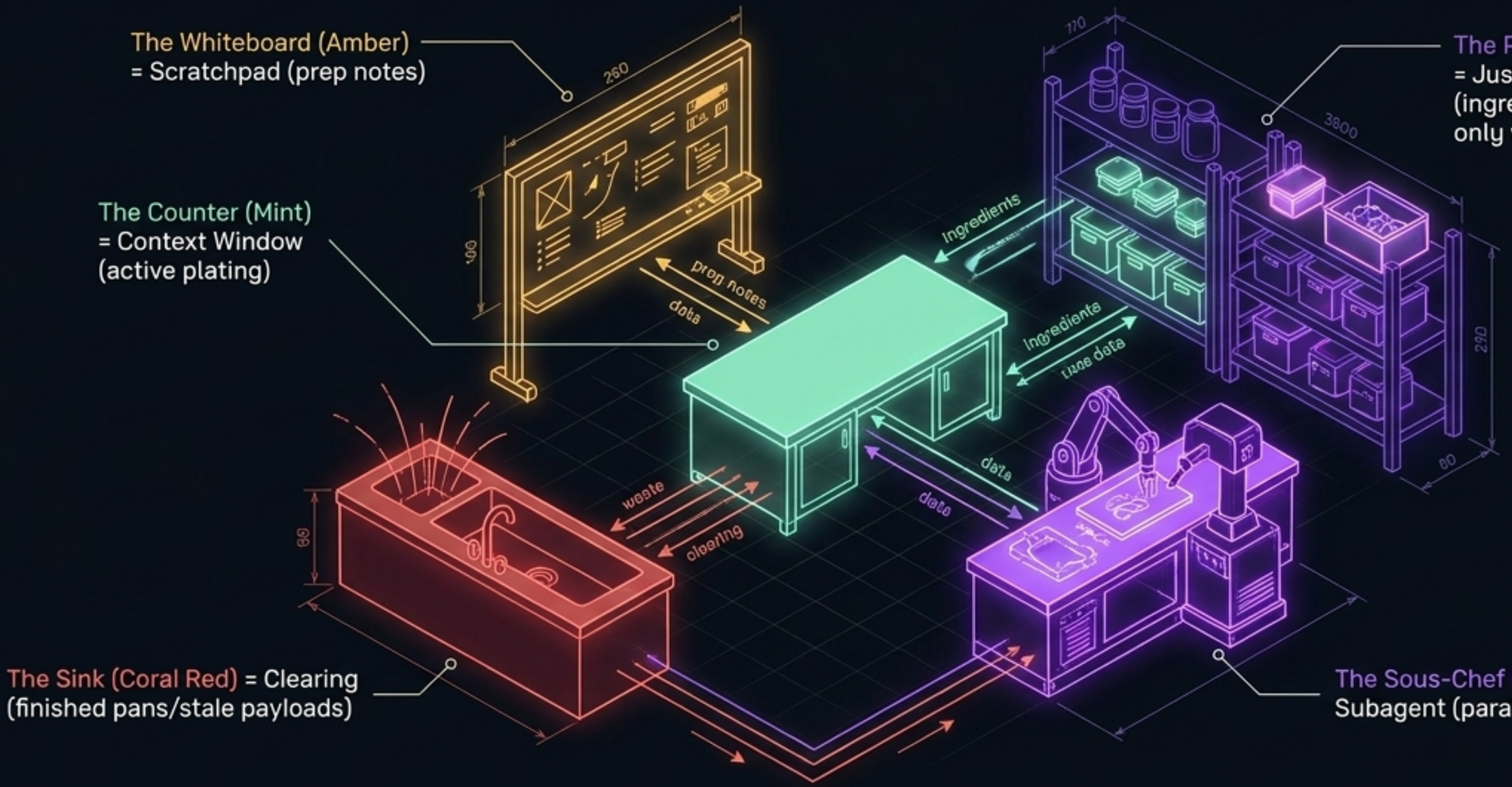
The Goal:

Keep the window small, current, and exceptionally high-signal for the entire run, preventing an agent from hallucinating or drifting over hundreds of turns.



The Kitchen Counter Mental Model

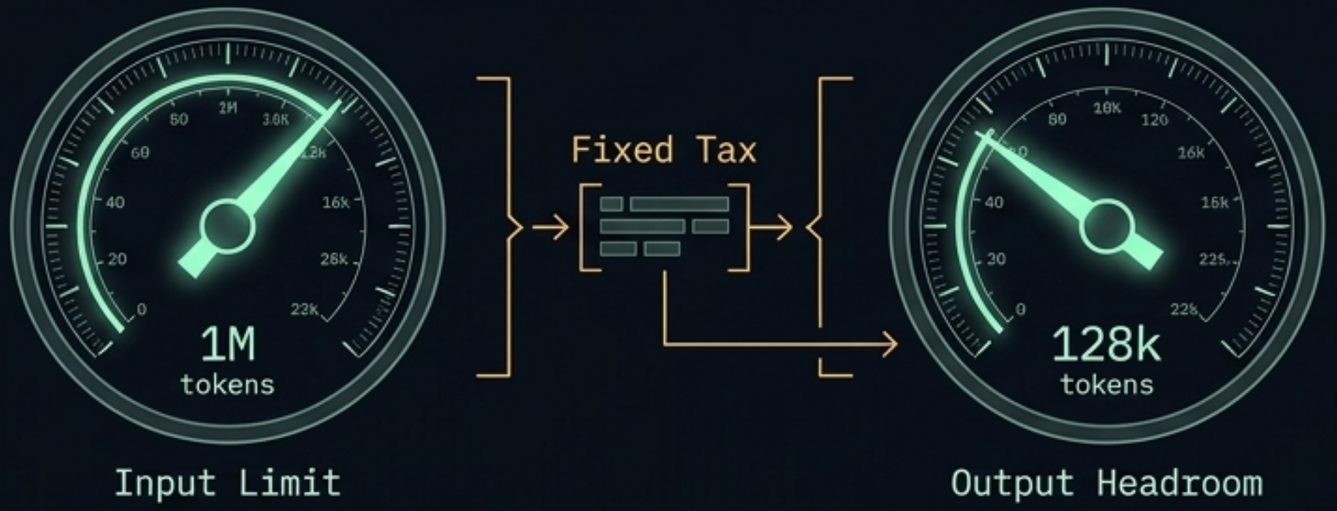
Mapping the Metaphor: Treat the LLM context window strictly as a working surface, not a storage unit. Only the elements required for the exact step being executed remain on the counter. Everything else must be routed to specialized architectural layers.



Architectural Taxonomy Matrix

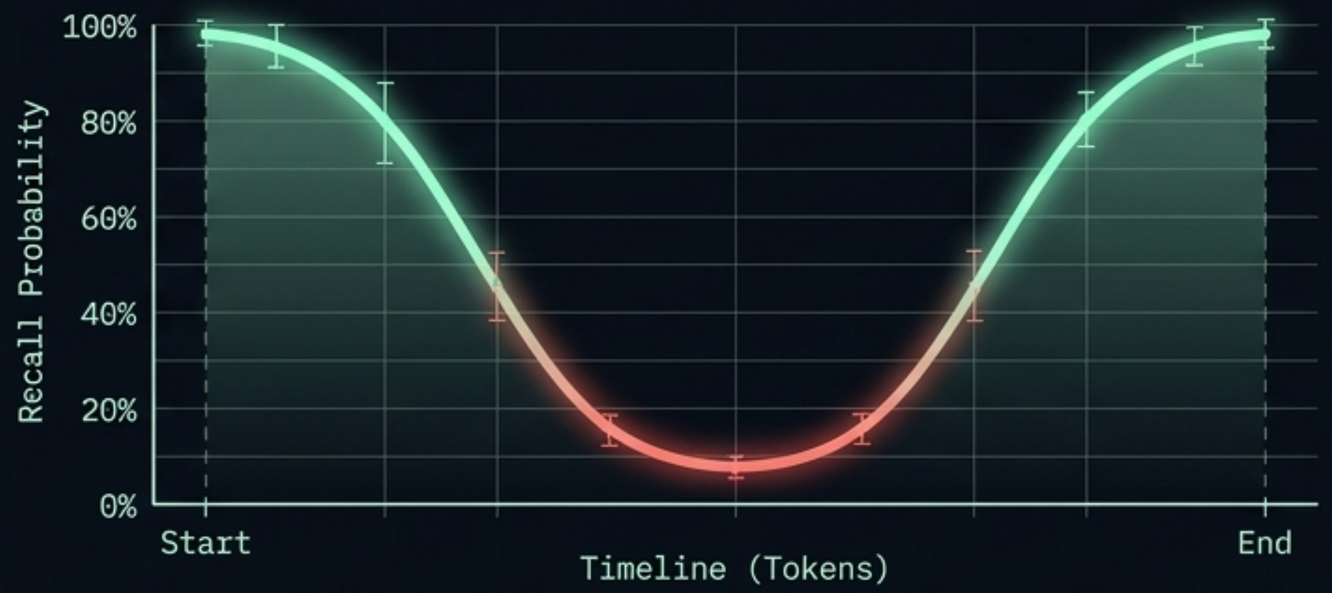
Prompt Engineering (Authoring)	Wording, structure, examples of one instruction. Written once, reused every call.
Context Engineering (Per-turn curation)	Selects, orders, compresses, offloads, and isolates everything in the window on every call.
State (Exact working data)	Deterministic Key/Value data the harness reads/writes (IDs, flags).
Scratchpad (Agent-written notes)	Files the agent writes/re-reads (plans, progress). Outlives a reset.
Memory (Long-term recall)	Facts extracted across sessions. Fetch by search/tool. Probabilistic.
Retrieval/RAG (Shared knowledge)	Global documents (policies, manuals) fetched up front or on demand.

Budget Arithmetic and the Position Effect



Budget Mathematics:

1. Input + Output share the capacity. Always reserve output headroom.
2. Tool definitions are a fixed tax sent on every call.
3. Count tokens via API—tokenizers evolve and generate different volumes.



The Position Effect (Liu et al.):

Models attend heavily to the start and end of the prompt; information in the middle degrades.

Optimization: Place long documents first. Place the active instruction and question last. Wrap documents in XML tags.

The Six Architectural Context Levers

1. Trim

Drop oldest turns once window passes a size threshold (sliding window).

2. Summarize

Replace a span of older turns with a rolling digest (compaction).

3. Clear

Remove stale tool results/thinking blocks while leaving conversation intact.

6. Isolate

Give a noisy subtask its own window, returning only — a short digest.

4. Offload

Write information to state/memory, keeping only a handle in the window.

5. Fetch Late

Load data via tools exactly when needed (JIT Retrieval).

6. Isolate

Give a noisy subtask its own window, returning only a short digest.



Core Rule: Every token draws on the same attention budget. Irrelevant tokens actively lower the recall of relevant ones.

Pathology of an Overloaded Session

Diagnostic Terminal Screen



Symptom 1: Generic Answers.

The agent begins referencing the usual approach instead of citing the specific file, ticket, or decision it successfully retrieved earlier in the session.

Symptom 2: Instruction Drift.

A strict constraint or rule stated early in the run mysteriously stops being applied after a sequence of large tool calls.

Symptom 3: Repeated Work.

The agent actively re-runs identical searches for results it already retrieved, because those results were pushed out of attention or dropped without a trace.

The Diagnostic Reality:

These are not model hallucinations; they are symptoms of context rot requiring immediate context reduction.

History Policies: Trimming vs. Compaction

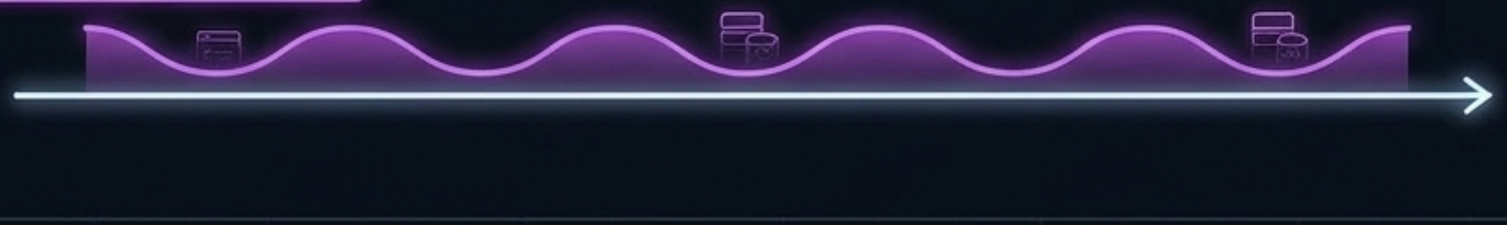
Sliding Window (Trimming)



Sends only the last N turns. Drops older turns abruptly.

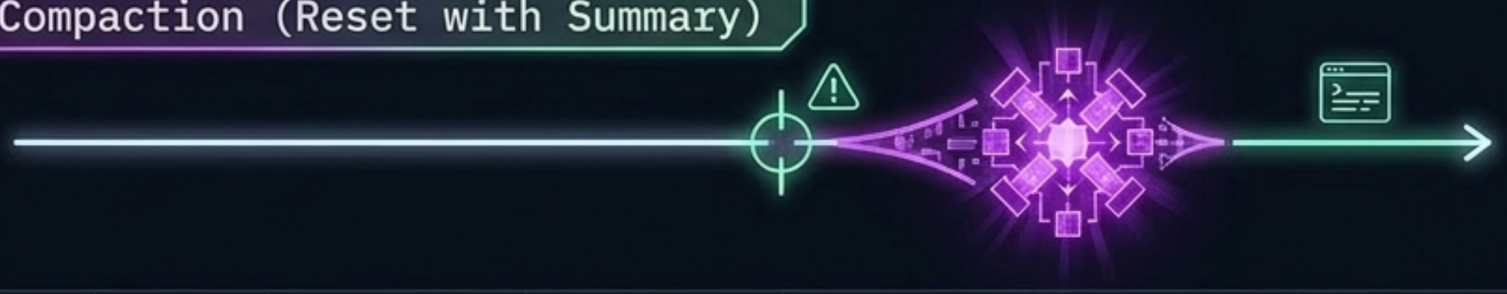
Cheap but blunt; early constraints vanish entirely.

Summary Prefix



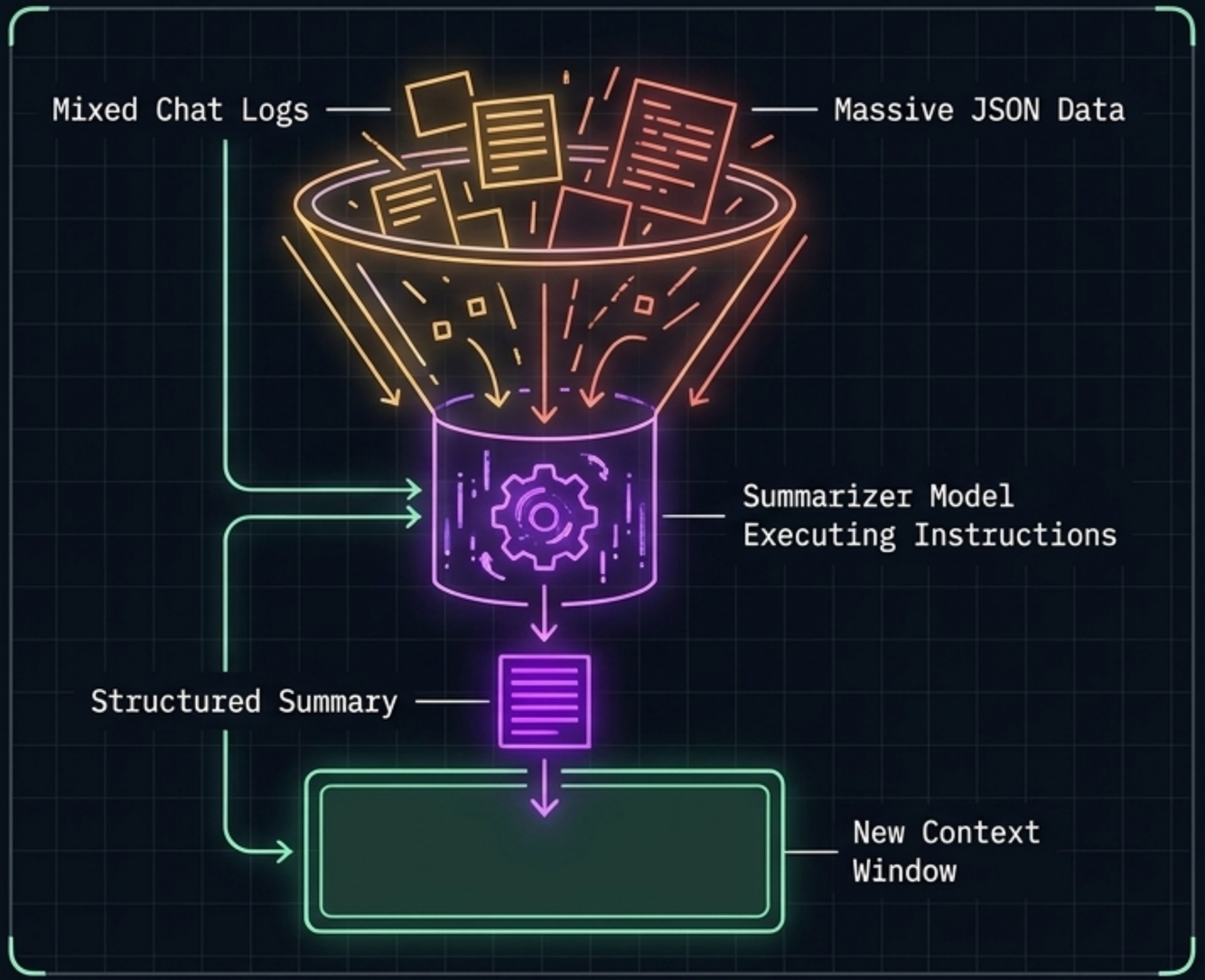
A rolling model-generated digest of older turns. Trades fidelity for continuity, often losing exact values (IDs, amounts).

Compaction (Reset with Summary)



The standard for long agent runs. Summarizes the history (decisions, artifacts), discards redundant tool outputs, and restarts in a fresh window seeded with the summary.

Deep Dive: Threshold Compaction Architecture



How it Triggers

The server summarizes the conversation once input crosses a threshold (e.g., 150,000 tokens).

Execution Flow

A summarizer compresses the history based on a specific focus string, distilling actions and parameters.

Tuning the Prompt

1. Maximize recall first on real traces.
2. Raise precision by discarding superseded drafts/raw payloads.

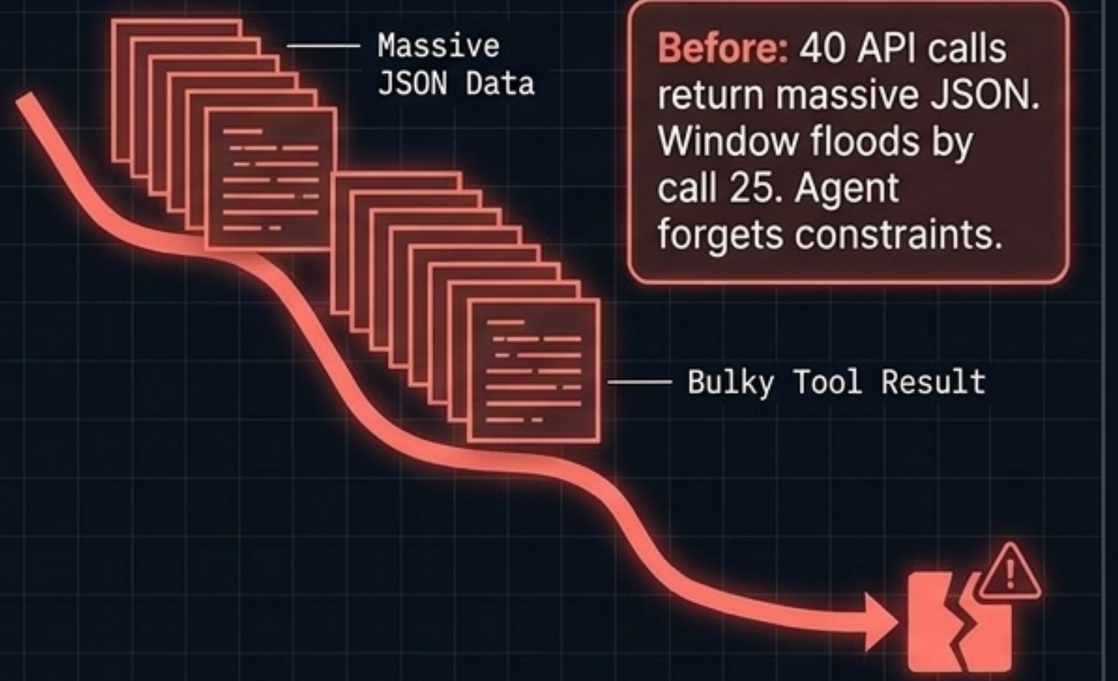
The Trap

Relying on early turns to carry a rule during compaction. Restate durable rules in the system prompt or disk files so they survive resets.

Context Editing and Tool-Result Clearing

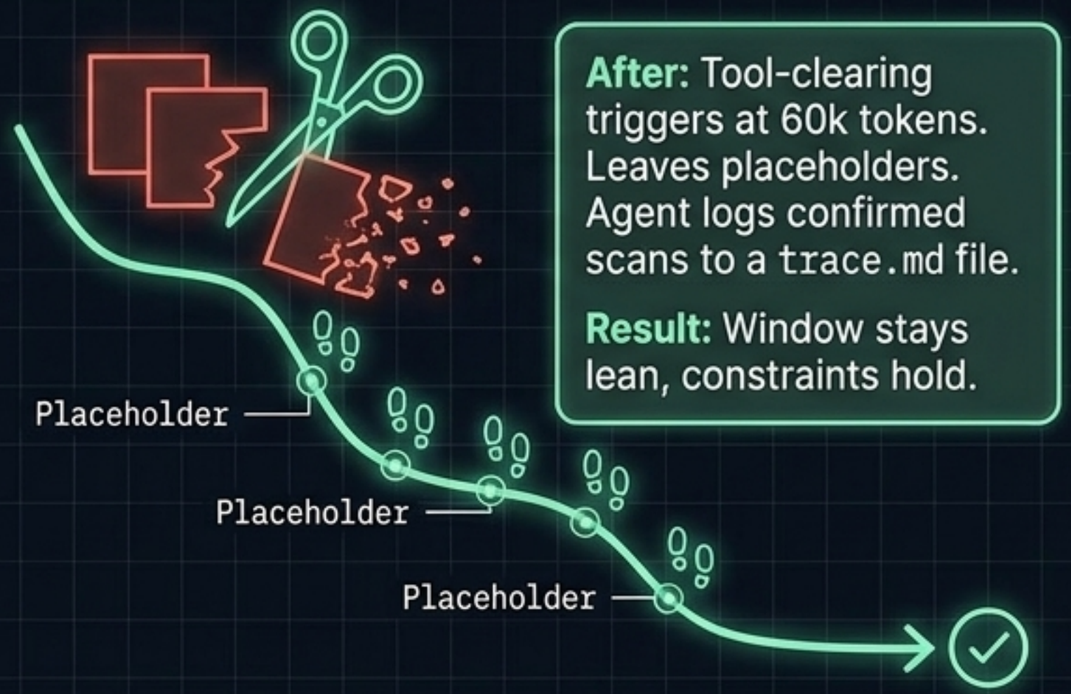
The Mechanism: Replaces older, bulky tool results or thinking blocks with placeholders, freeing space without model-rewriting. The original tool calls remain visible so the agent knows the action occurred.

Before State



Before: 40 API calls return massive JSON. Window floods by call 25. Agent forgets constraints.

After State

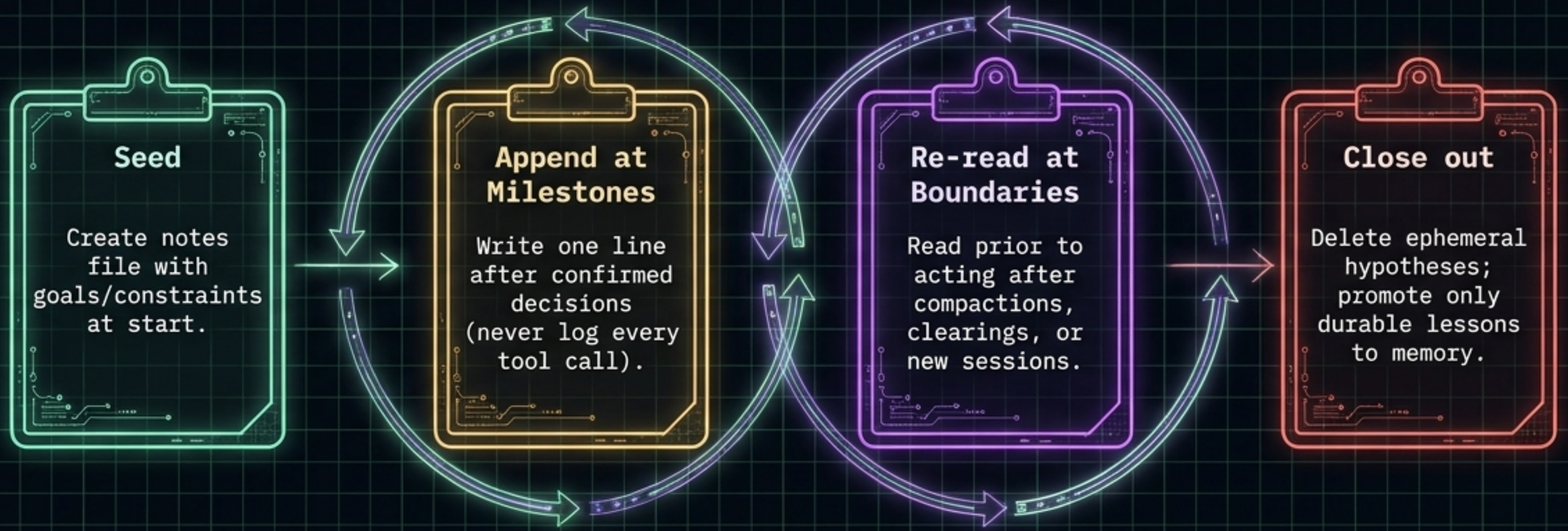


After: Tool-clearing triggers at 60k tokens. Leaves placeholders. Agent logs confirmed scans to a `trace.md` file.

Result: Window stays lean, constraints hold.

Structured Note-Taking (Scratchpads)

The Concept:
Agents periodically write notes to outside storage (e.g.,progress.md) and re-read them after resets, providing persistence with near-zero overhead. Agent-curated, not harness-owned.



Implementation: Executed via memory tool client-side commands (view, insert, str_replace) or artifact services.

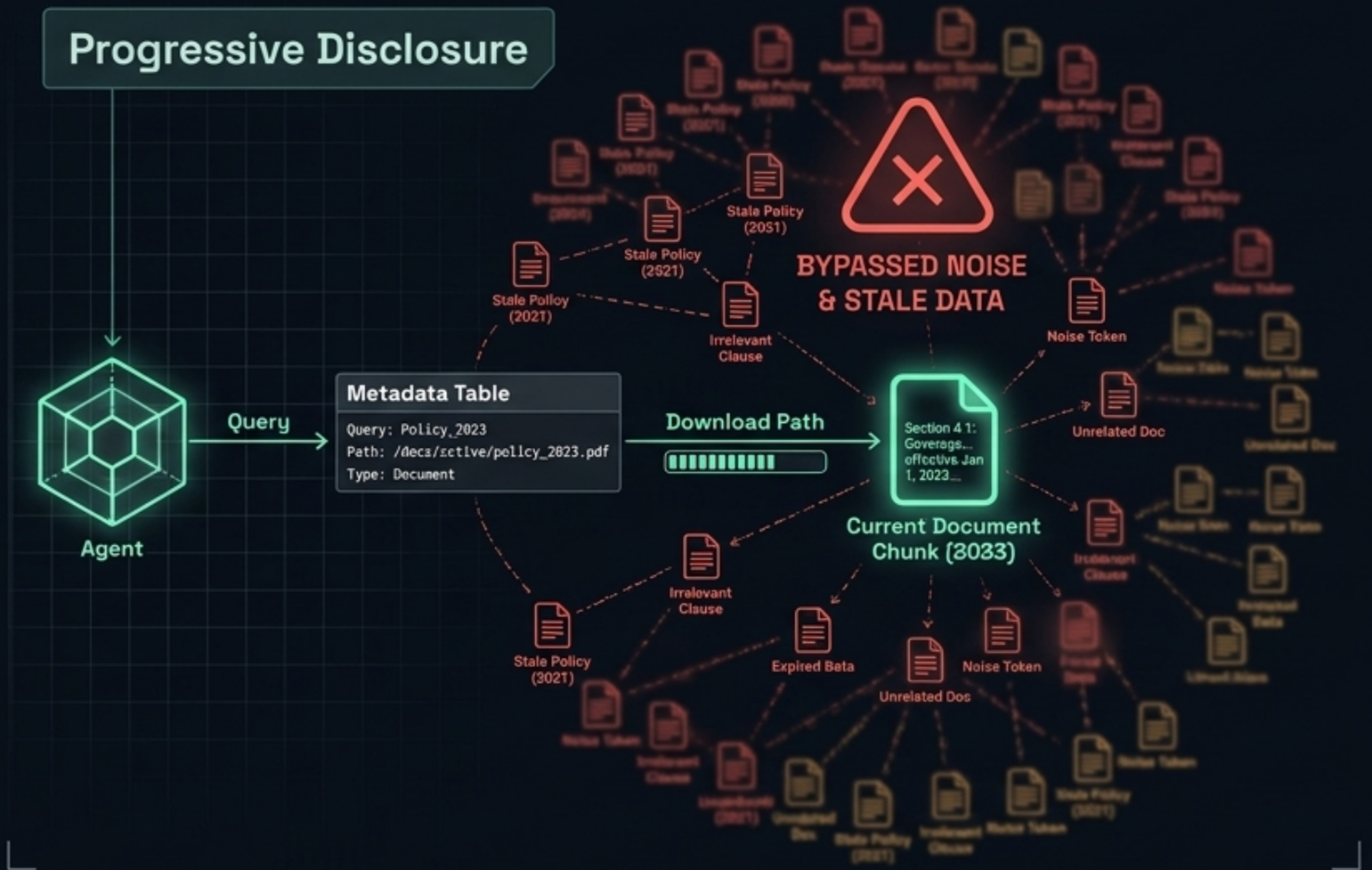
Just-In-Time (JIT) Retrieval vs. Up-Front RAG

Up-Front Retrieval (RAG)

Embeds/searches corpus before inference. Fast, but floods window with loosely relevant chunks from potentially stale indexes.

Just-In-Time (JIT) Retrieval

Keeps lightweight handles (paths, queries) in window. Loads data via tools exactly when needed. Metadata acts as signal.



Scenario: Fernhill Insurance (Policy Lookups)

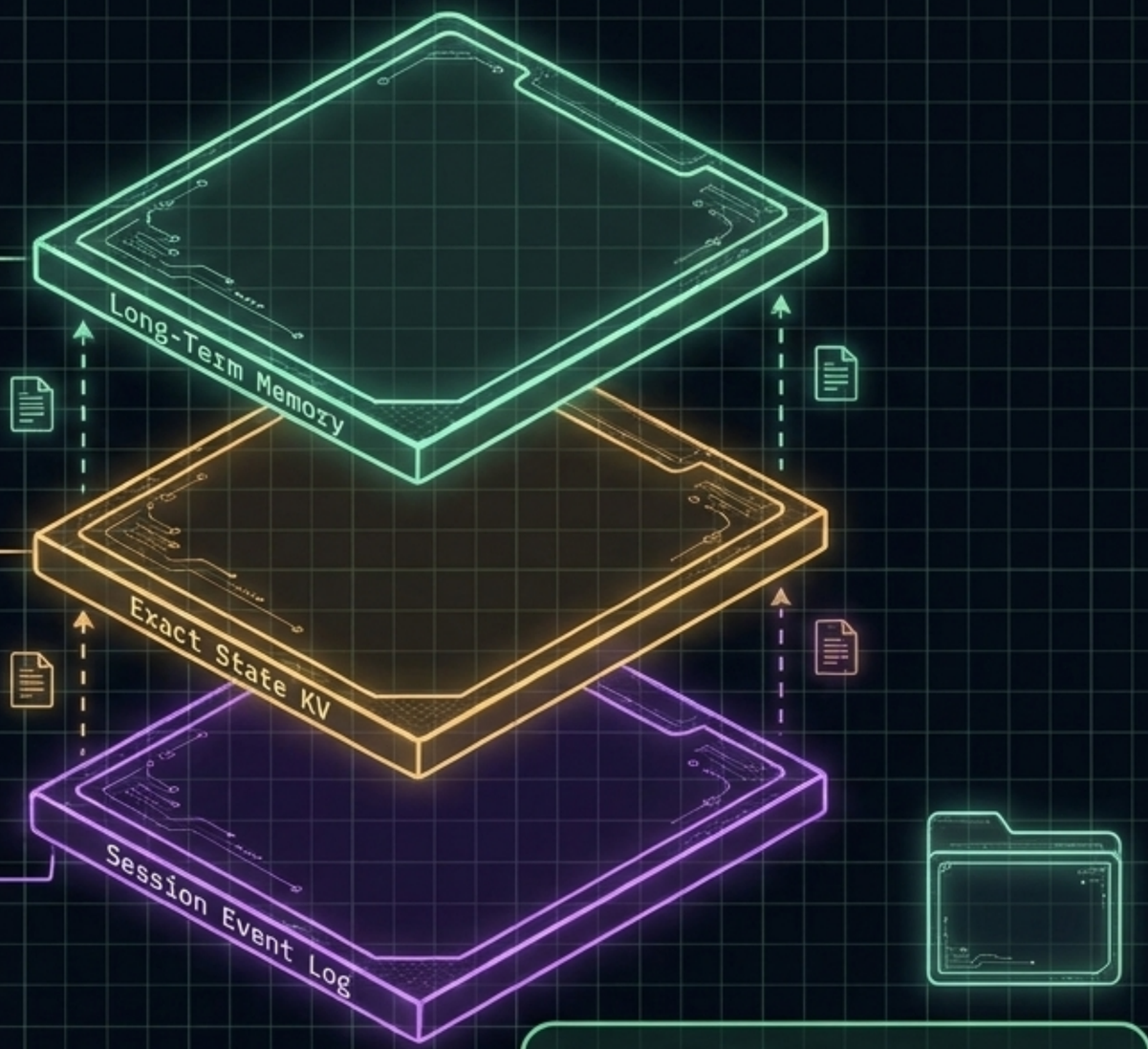
Before: Preloaded top 20 vector chunks per question. 30k tokens of noise, quoted stale policies.
After: Agent queries a short metadata catalog, uses tools to open exactly 2-3 current documents.
Result: Zero hallucinated dates, massive token reduction.

The Layered Memory Stack

Long-Term Memory:
Extracted facts across sessions. Probabilistic recall via search or memory tools. Consolidates over time.

State: Exact key/value working data (e.g, Ticket ID, approved budget). Harness reads/writes deterministically. Scoped via prefixes.

Session (History): The ordered turns/tool calls of one conversation. Replayed into the window, aggressively trimmed or compacted.



Files/Artifacts: Large outputs and scratchpads, read by path.

Memory Write Policies & Validation

Extract at Boundaries

Generate memories at session end or task milestones, not on every turn.

Raw Chat Logs

Boundary Extraction

Validation-Retry Loop

Overrides Old Data

Central Truth Node

Consolidrate, Don't Append

A new fact overrides contradictions (e.g., *Moved to Lyon* actively replaces *Lives in Paris*). Append-only stores accumulate contradictions.

Paris (Old) → Lyon (New)

Direct Writes

Allow explicit user facts to be written directly to the store without waiting for background extraction.

Scope Every Record

Strict keying by user and tenant. Retrieval must match exact scope before similarity search to prevent cross-tenant data leakage.

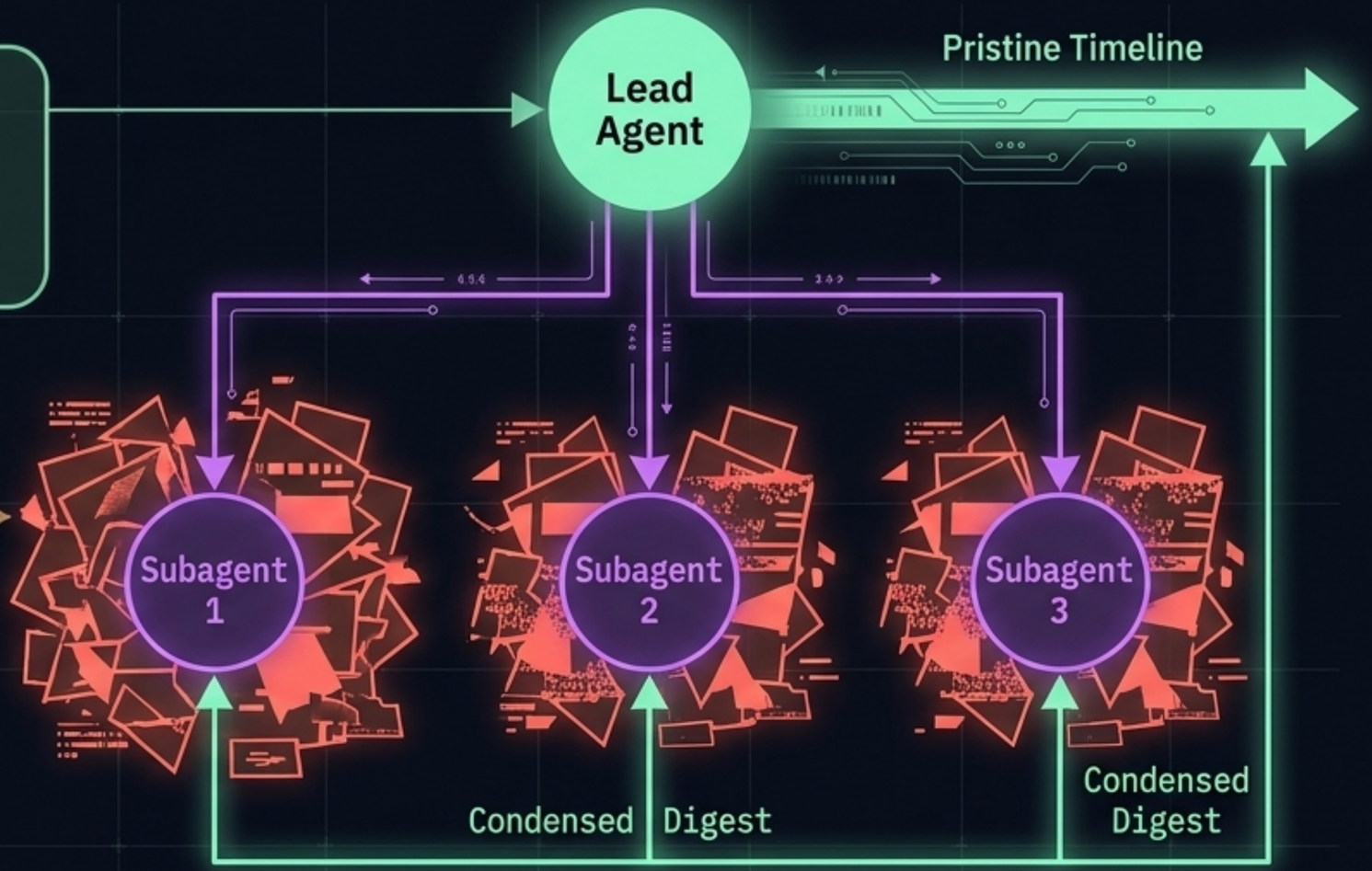
User_A

Tenant_B

Subagent Context Isolation

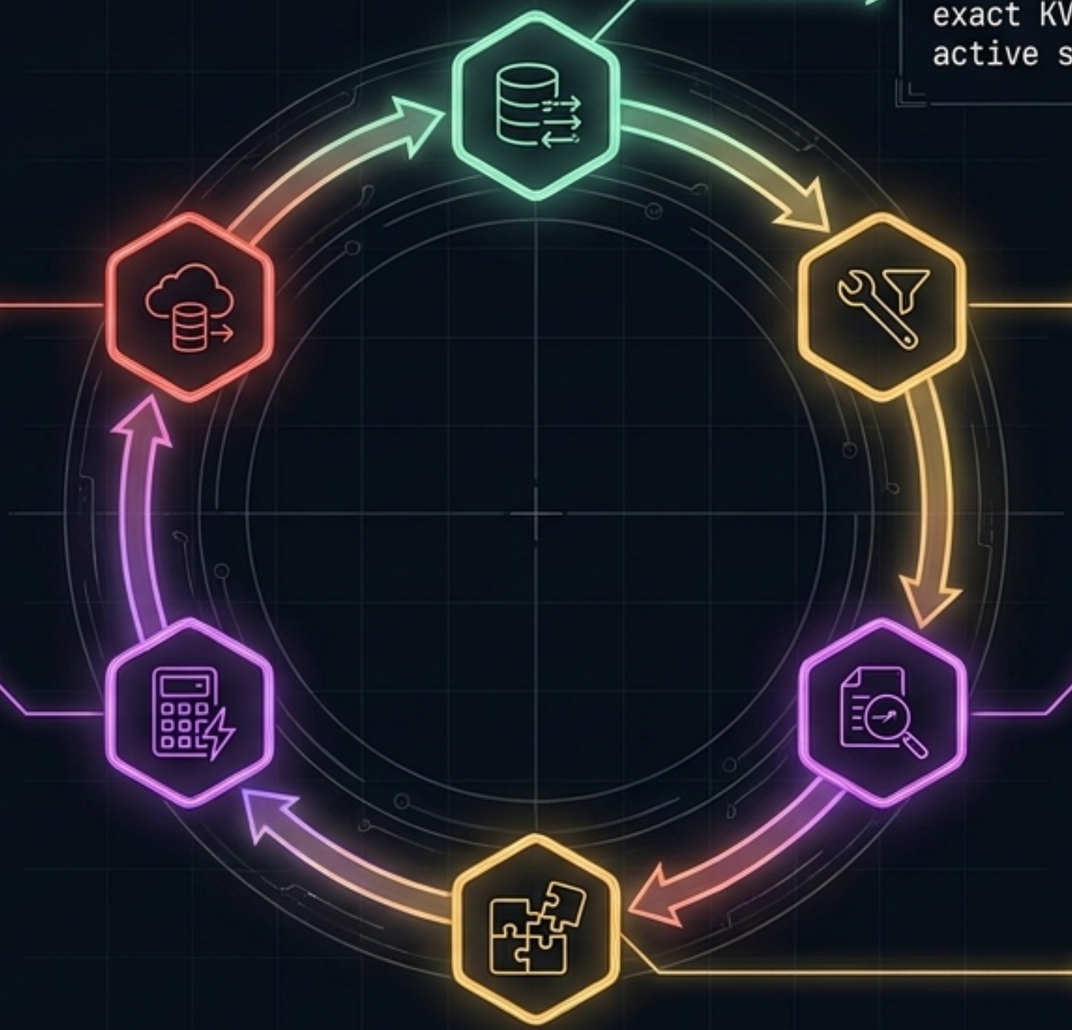
The Pattern
Rather than one agent holding an entire project, a Lead Agent delegates noisy search/exploration tasks to isolated subagents that start with empty windows.

The Output
Subagents spend thousands of tokens exploring wide datasets, but return a strictly condensed digest (1k-2k tokens) containing paths, IDs, and final answers.



Scenario: Pinecrest Labs (Dependency Upgrade).
Lead fans out subagents across 30 services. Subagents search call sites and return tables of file paths. Lead edits file-by-file just-in-time from the digests. The raw search context never touches the Lead's window.

The Per-Turn Assembly Workflow



1. Load:
Read session events, exact exact KV state, and the active scratchpad.

2. Reduce: Apply the history policy (compaction/trim) and execute tool-result clearing.

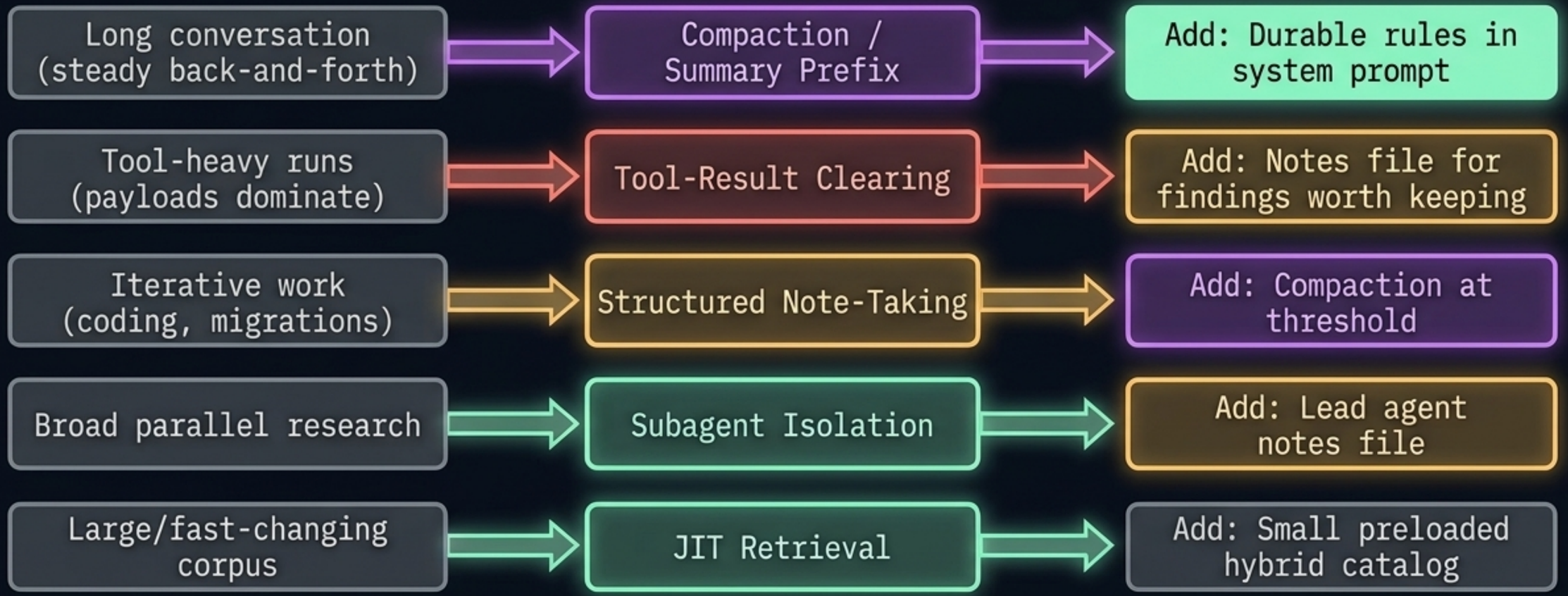
3. Fetch: Execute JIT retrieval tools; pull in metadata and targeted document sections.

4. Place: Assemble strict order: Stable prefix (Tools/System) -> State/Summary -> Recent Turns -> Current Input.

5. Call: Calculate tokens against limits. Reserve output headroom. Execute LLM call.

6. Offload: Write new decisions to state/scratchpad. Replace bulky results with handles.

Strategy Selection Blueprint



Rule of Thumb: Start light. Add clearing before summarization.
Add notes, then subagents, as complexity demands.

Economics and Prefix Caching



The Mechanism

Both Claude and Gemini reuse computation for unchanged prompt prefixes.

Prompts must be strictly ordered to maximize this zone.

Optimal Order

Tools → System Instructions
→ User Data
→ Fixed Examples
→ [Dynamic Turn Content / Retrieved Items]

Cache Breakers

Any mutation before the breakpoint breaks the cache.

Anti-pattern: Putting a timestamp in the system prompt.

Interaction: Clearing tool results alters the prefix from that point onward.

Security and Threat Vectors



Threat Vectors (Coral)	Architectural Controls (Mint)
Memory Poisoning: An attacker plants malicious instructions in a document. The agent extracts it as a fact and saves it to permanent memory.	**Fact Extraction Only:** Ensure memory extraction writes exact facts, not instructions.
Indirect Prompt Injection: Web pages or emails opened via JIT retrieval hijack the active instructions.	**Least Context:** Give agents only the tools they need. Unseen context cannot inject.
Path Traversal: Agents editing local memory files escaping the /memories directory via '..' commands.	**Human-in-the-loop:** Irreversible actions require external validation, bypassing the active context entirely.

Diagnostic Anti-Pattern Matrix

Symptom	Root Cause	Architectural Fix
Agent ignores a rule stated in Turn 2.	Sliding window dropped it.	Move durable rules to System Prompt or CLAUDE.md.
Subtle early decision vanishes after reset.	Compaction prompt tuned for brevity.	Tune compaction for recall on real traces first, then trim noise.
Agent repeats the same lookup endlessly.	Results were cleared with no breadcrumb.	Keep tool calls visible; log findings to a scratchpad.
Lead agent window floods during research.	Subagents returning raw reads.	Enforce short digests with handles.
Old hypotheses resurface in new tasks.	Ephemeral notes promoted to memory.	Prune file at task close; promote durable lessons only.
Cache hit rate remains near zero.	Dynamic variable early in prompt.	Move dynamic values after the stable prefix zone.