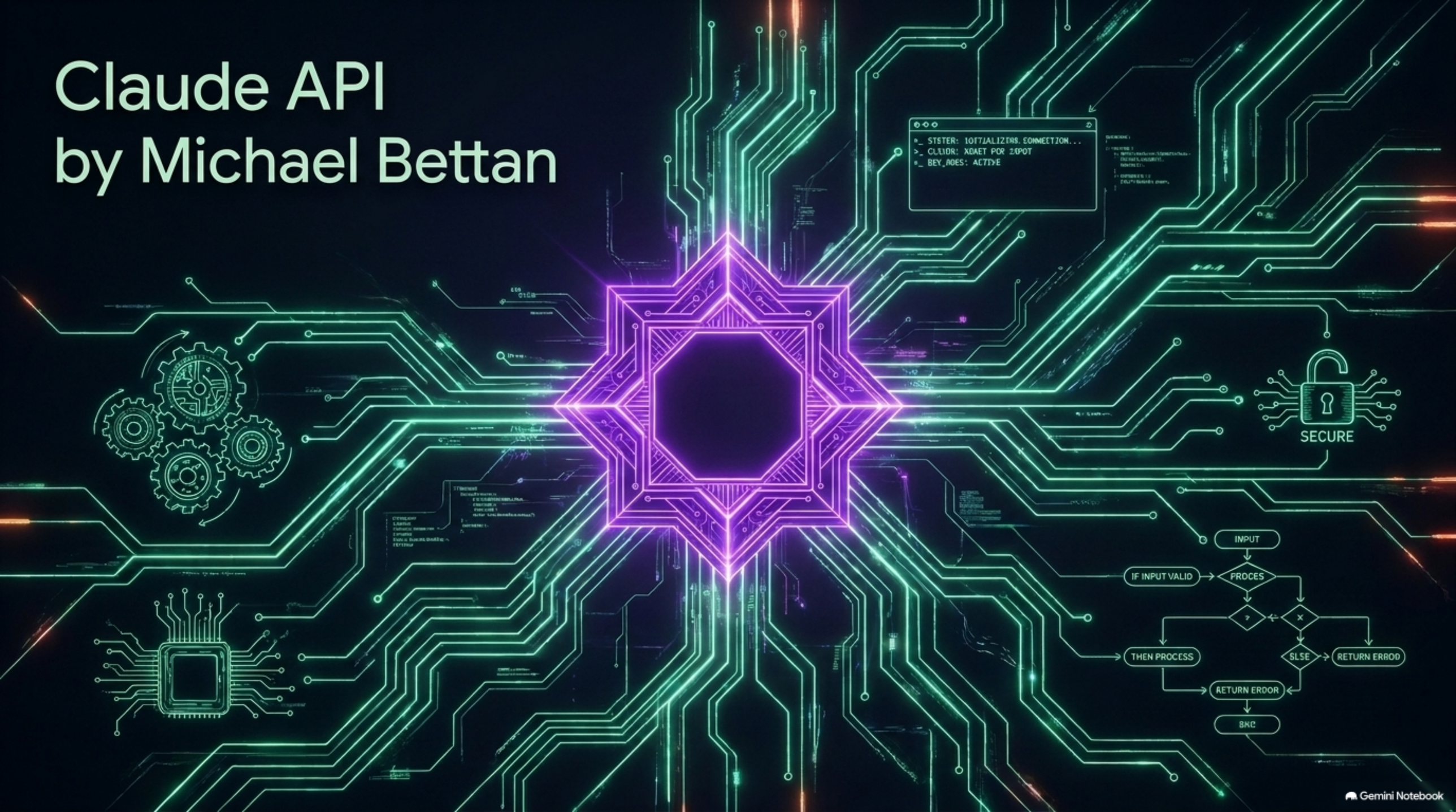
Claude API
by Michael Bettan



Gemini Notebook

The Stateless Transition Engine

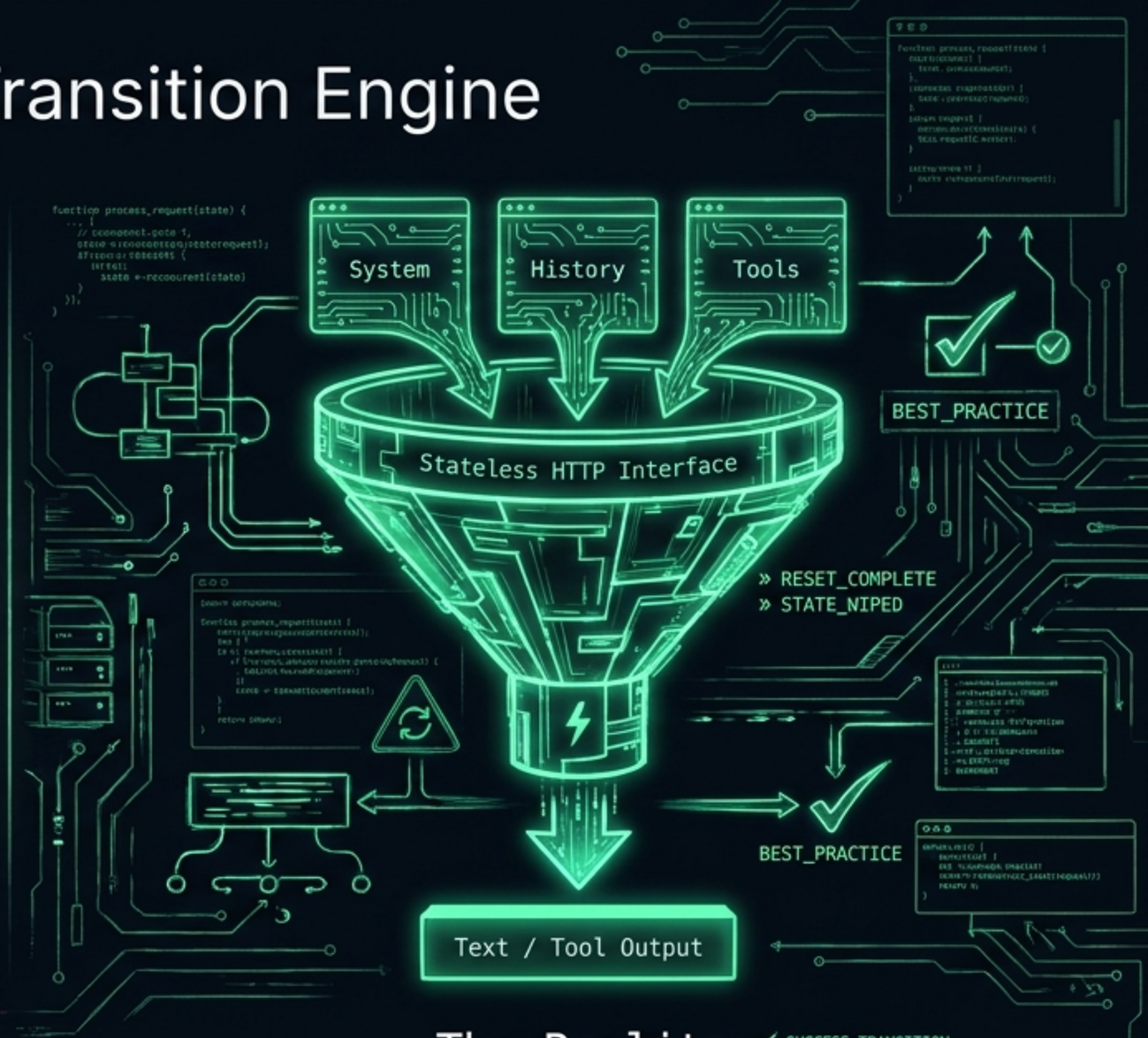


The Illusion

ERR_STATE_PERSISTENCE
404_CONTEXT_NOT_FOUND

The API remembers nothing between calls.
It never runs your tools.

>_



The Reality

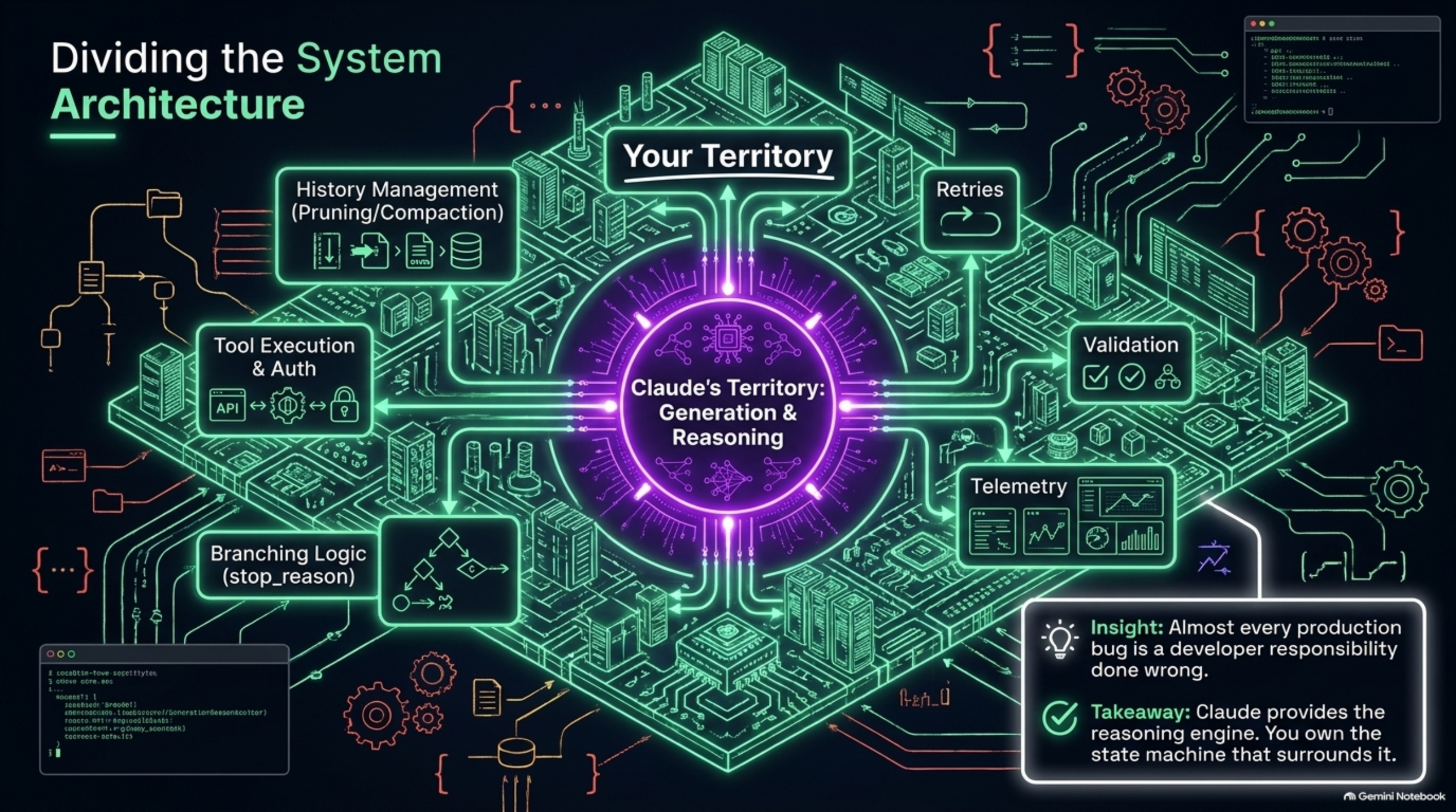
✓ SUCCESS_TRANSITION
✓ 200_OK

Every response is a pure state transition.
Your application must reconstruct and send the entire required state on every single request.

>_ » >_ state = reconstruct_state(request)

Gemini Notebook

Dividing the System Architecture



Insight: Almost every production bug is a developer responsibility done wrong.

Takeaway: Claude provides the reasoning engine. You own the state machine that surrounds it.

Gemini Notebook

The Capability Ceiling

Rule: Do not upgrade the entire pipeline when only one step fails.



Opus

Hardest reasoning, planning, and review of critical output.

Most capable, slowest, highest cost.

Sonnet

The default starting point for most apps, agents, and coding.

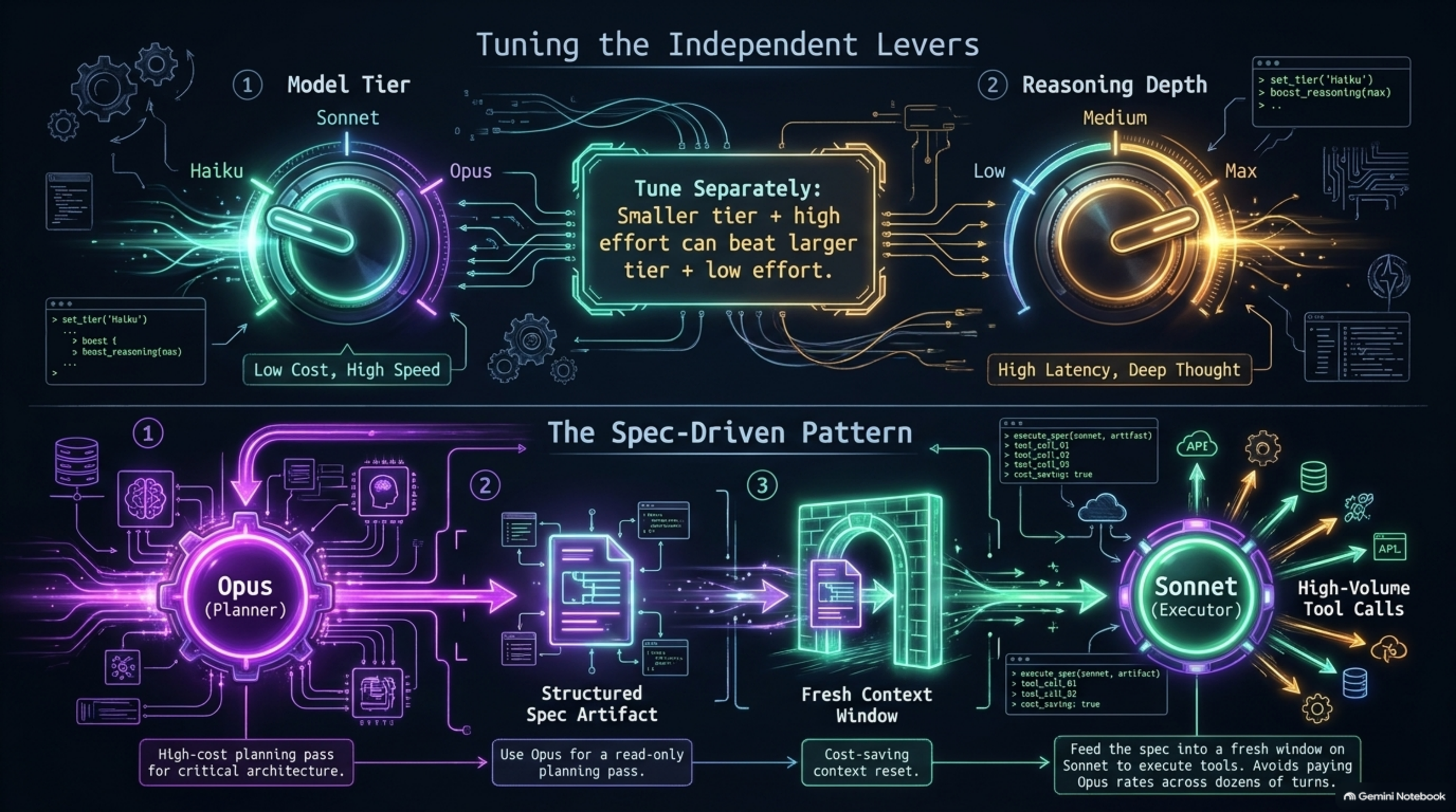
Balanced capability, speed, cost.

Haiku

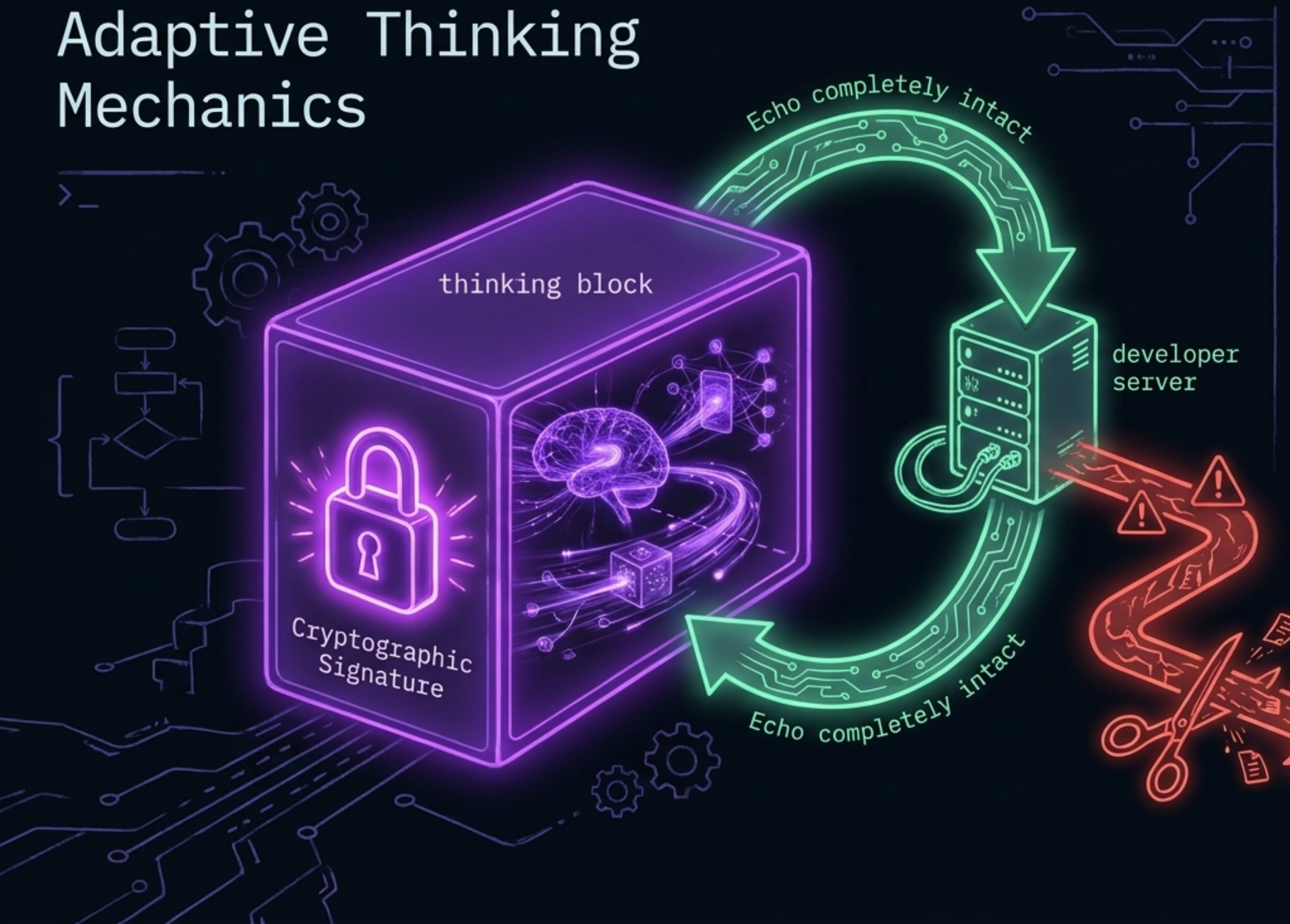
High-volume routing, real-time triage, sub-agent workers.

Fastest, lowest cost.

Gemini Notebook


Tuning the Independent Levers
1 Model Tier
Haiku
Sonnet
Opus
Low Cost, High Speed
2 Reasoning Depth
Low
Medium
Max
High Latency, Deep Thought
Tune Separately:
Smaller tier + high effort can beat larger tier + low effort.
The Spec-Driven Pattern
Opus
(Planner)
Structured
Spec Artifact
Fresh Context
Window
Sonnet
(Executor)
High-Volume
Tool Calls
High-cost planning pass for critical architecture.
Use Opus for a read-only planning pass.
Cost-saving context reset.
Feed the spec into a fresh window on Sonnet to execute tools. Avoids paying Opus rates across dozens of turns.
Gemini Notebook

Adaptive Thinking Mechanics



Mechanic: Claude sizes its own reasoning via `thinking={'type':'adaptive'}`; you steer depth with effort. (Manual `budget_tokens` returns HTTP 400 on models 4.7+).

Billing: All internal thinking is billed as output tokens, even if `display: omitted`.

The Immutable Law: Thinking blocks carry a cryptographic signature. Echo them back exactly as received when returning tool results. Editing or filtering them guarantees a 400 error.

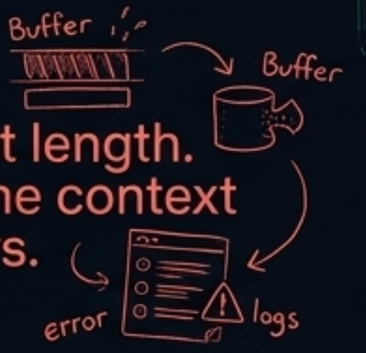
Gemini Notebook

The Messages API Shape

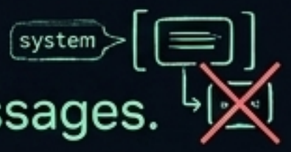
```
{
  "model"
  "max_tokens"
  "system"
  "messages"
    [
      {"role": "user", "content": ...},
      {"role": "assistant", "content": ...},
      {"role": "user", "content": ...}
    ]
```

Pin a fixed ID.

Trap: This strictly caps output length.
It does NOT create room in the context window or fix validation errors.



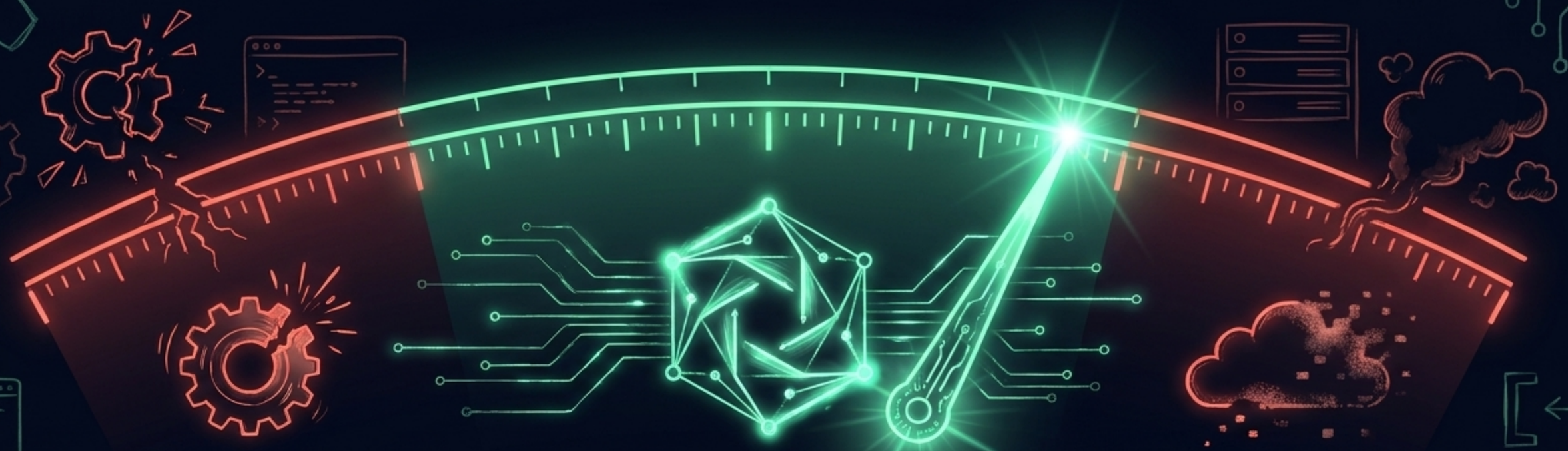
Top-level string/blocks.
No ‘system’ role inside messages.



Structure: Alternating user/assistant turns.
Content can be a string or an array of strictly typed blocks (`text`, `tool_use`, `tool_result`, `thinking`).

Gemini Notebook

The Prompt Altitude Goldilocks Zone



Too Specific / Brittle

Treating the LLM as a deterministic state machine with 100% predetermined branching.

Breaks on unscripted edge cases.

The Goldilocks Zone

1. Clear identity & scope.
2. Goal-oriented tool guidance (not sequences).
3. Repeatable reasoning framework (Understand → Gather → Act → Verify).
4. Explicit escalation triggers.

Too Vague / Underspecified

High-level platitudes and false assumptions of shared context.

Causes high variance and hallucinations.

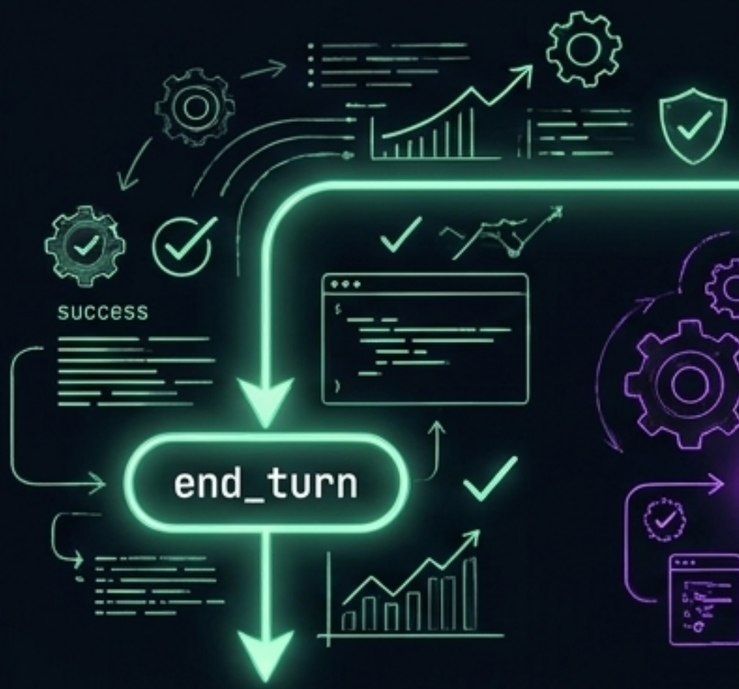
Gemini Notebook

The Critical Node: Branching on Stop Reason

Check stop_reason first.
Every time.



The Root of Most Bugs:
Code that reads content[0].text blindly without checking stop_reason will inevitably crash in production.



Read text.



Run tools, return paired results, loop.



Truncated.
Prune or raise cap.



Read stop_details.
Do not blind retry.

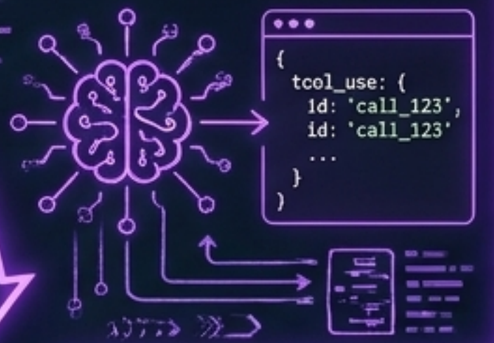
Gemini Notebook

The Immutable Tool-Use Loop

1. Define: Define tools, descriptions, schema.



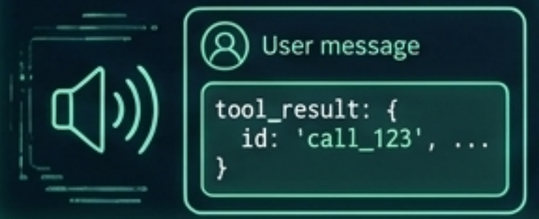
2. Request: Claude emits **`tool_use`** with an ID.



3. Execute: Developer runs authorization and code.



4. Return: Echo assistant turn, append user turn with tool_result matching the ID.



Pairing Rule: Every tool_use needs a tool_result with an exact matching ID.

Chronology Rule: Results go in the very next user message. Nothing in between.

Placement Rule: tool_result blocks must come first in that user message.

Failure Rule: Return failures as is_error: true with a short message. An empty string reads as 'no data', causing confident hallucinations.

Gemini Notebook

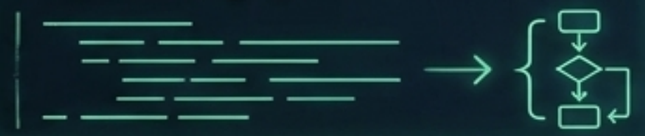
Enforcing Structure

System Prompt

Best-effort prompt guidance (YAML/Bullets).

Use for developer tools, architectural plans, streaming Markdown.

No parsing guarantees.



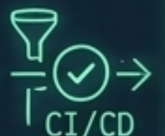
JSON Outputs

Uses `output_config.format`.

Guarantees response text parses and matches schema.

Use when answer must be machine-readable (extraction, CI gates).

```
{
  "key": "value"
}
```

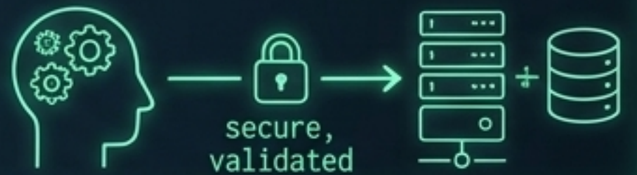


Strict Tools

Uses `strict: true`.

Guarantees tool inputs match schema.

Use for agents feeding real backend systems.



The Reality Check: Shape ≠ Correctness.

A schema-valid object can still hold a hallucinated value. Always maintain business-rule validation in your code. Move format enforcement into the API, but keep business validation local. (Prefilling the assistant turn is rejected on 4.6+).

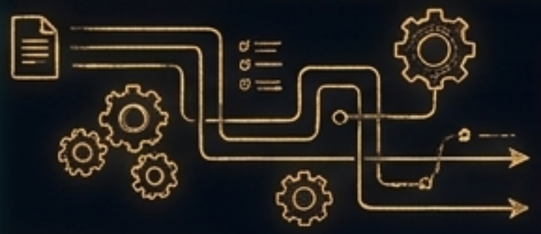
Gemini Notebook

Agent Taxonomy & Scaling Tool Execution

Autonomy

Complexity

2. Shallow ReAct / Standard MCP





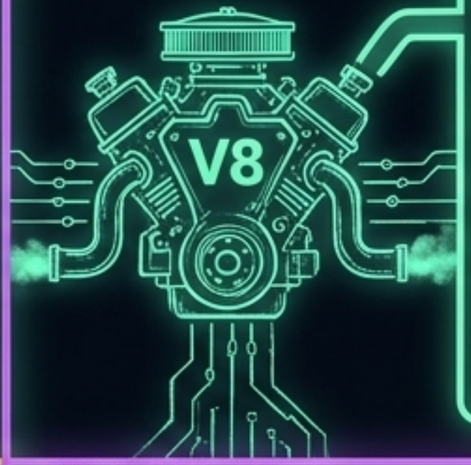
Scaling limits:
Standard MCP is best for 1-5 iterations. Scaling to 50+ tools burns ~50k tokens on Turn 1 (Schema Bloat) and creates massive sequential round-trip latency.



Massive sequential round-trip latency.

4. CodeMode (Programmatic Calling)

```
{ "type": "CodeMode" }
```

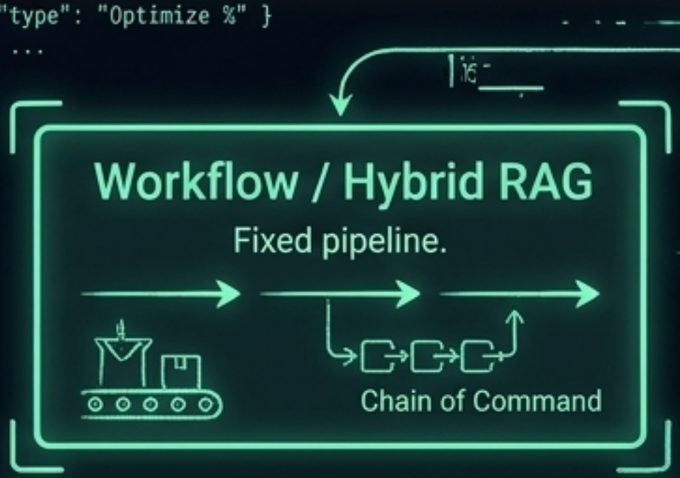


Security Sandbox

Converts MCP schemas to a TypeScript API. Model writes code executed in a network-isolated V8 sandbox, collapsing dozens of LLM round-trips into a single inference pass.




```
{ "type": "Optimize %" }
  ...
```

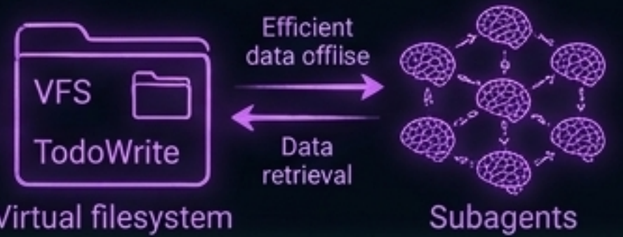



```
if (complexity > 50) {
  ...
}
```

3. Deep Agent Harness

Best for long-horizon tasks (30+ turns). Offloads bulky intermediate state to virtual filesystems to prevent context pollution.

(TodoWrite, VFS, Subagents)



Gemini Notebook

Diagnosing Context Rot

Context Poisoning



Turn 1 failure = Schema Bloat / Tool-schema upfront pollution. Use progressive disclosure.

Antidote:



Hard-reset messages: [ ] or prune the turn (summarization preserves the poison).

Context Confusion



Turn 5 failure = Accumulated Tool Result rot crowding out instructions.

Antidote:



Paginate reads (offset/limit), prune stale tool results, or offload to Virtual FS.

Context Clash



Contradictory rules and stale instructions.

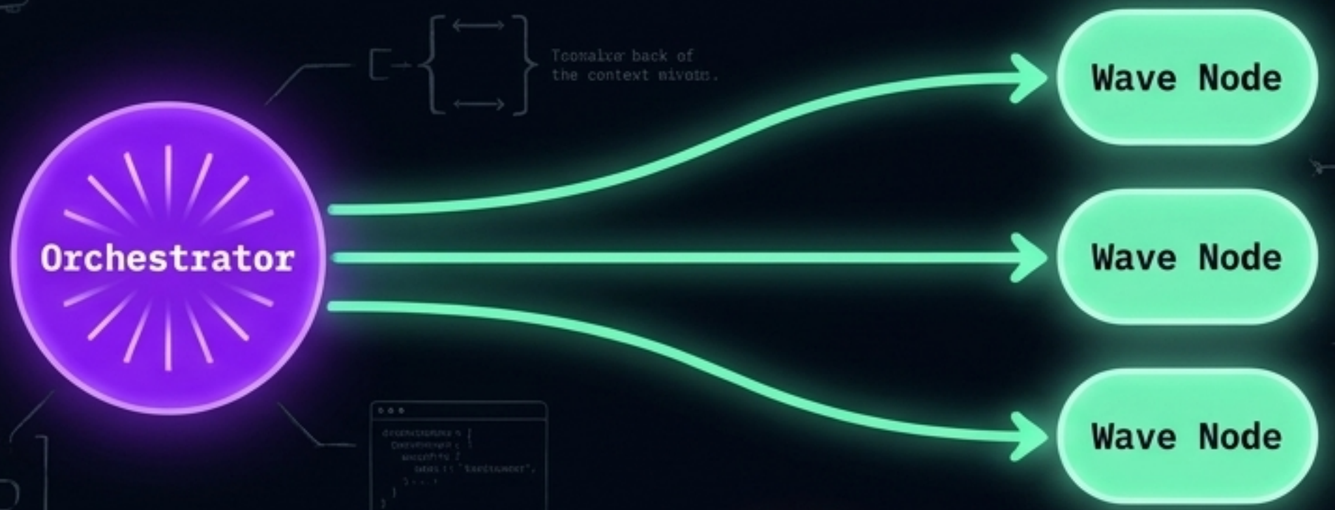
Antidote:



Deduplicate instructions, drop superseded turns.

Gemini Notebook

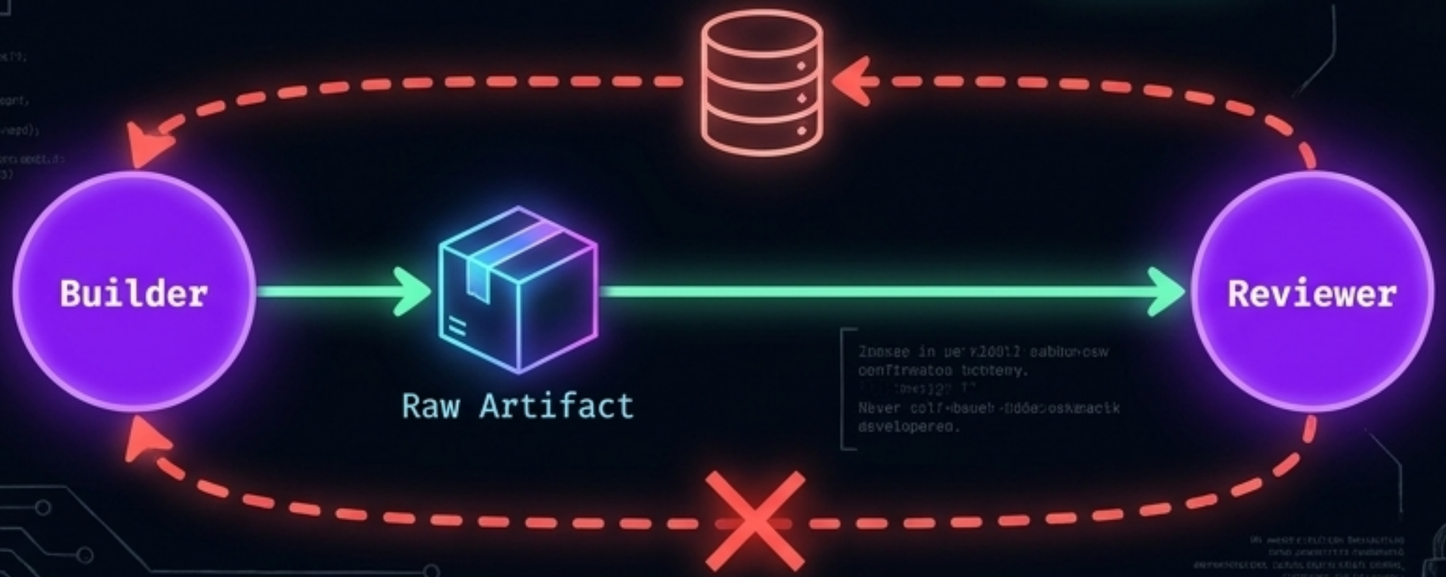
Multi-Agent Context Contracts



Wave-Based Fan-Out

Dispatch large parallel tasks in waves of 3-5 to prevent 429 rate-limit storms.

Never parallelize sequentially coupled tasks.



Builder-Reviewer Isolation

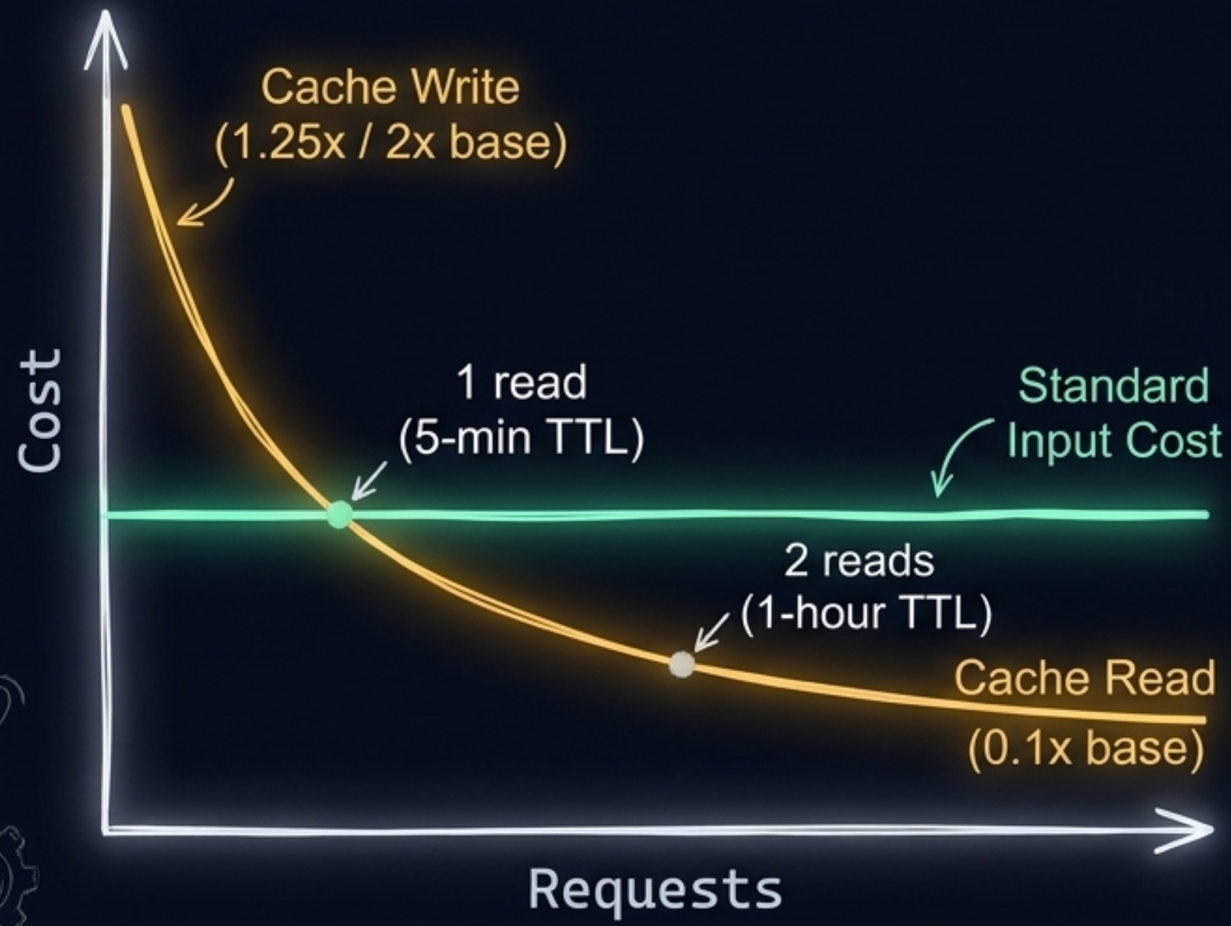
Builder-Reviewer isolation prevents confirmation bias.

Self-Contained Handoff:
A subagent call has zero parent history. Pass a self-contained prompt and return raw artifacts, not prose summaries.

Always run the Reviewer in a completely separate, fresh context window.

Gemini Notebook

Prompt Caching Economics



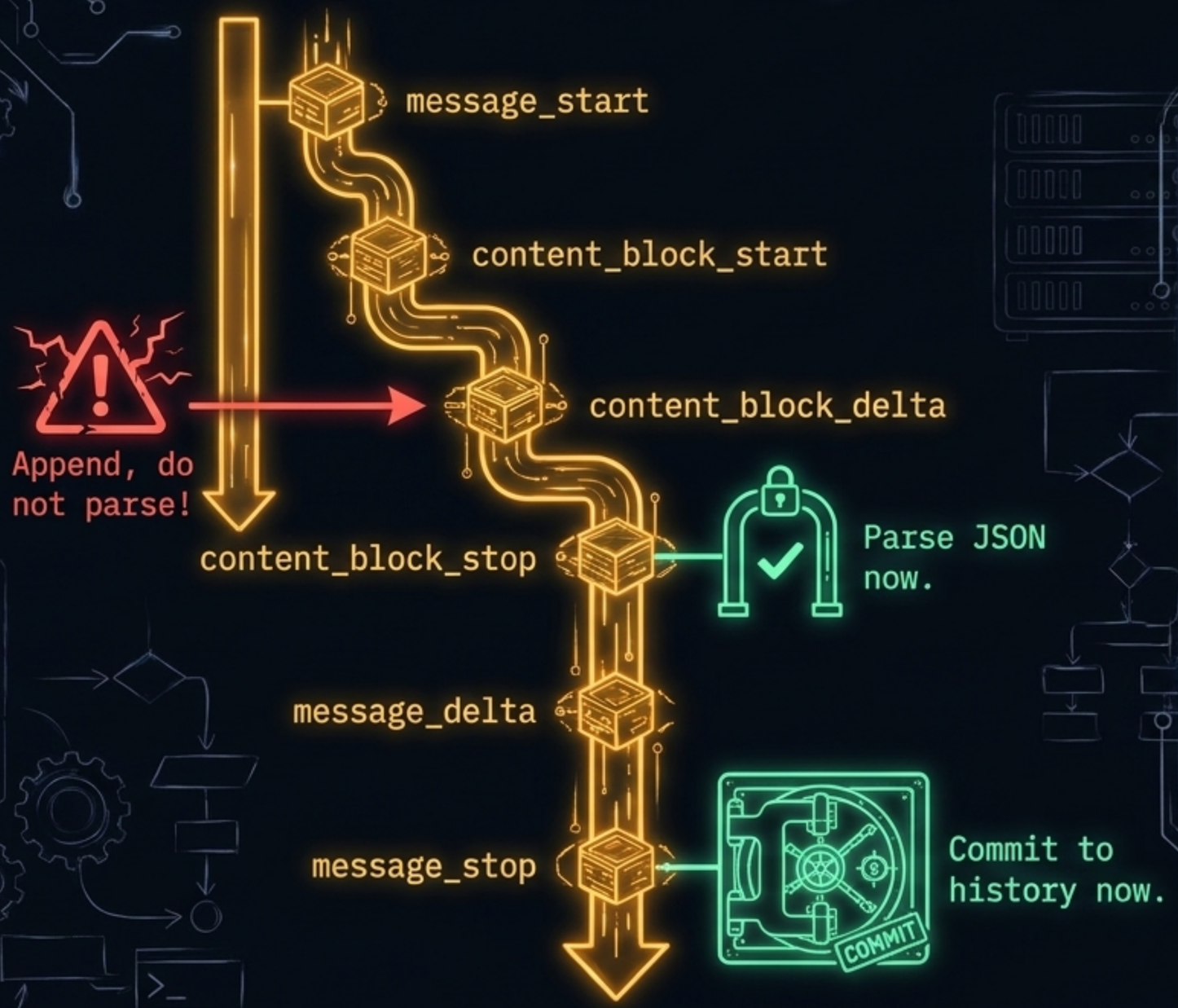
The **Prefix Rule**: Cache hits require an identical prefix order.

[tools → system → messages]

The **Silent Invalidator Trap**: Placing a timestamp, user ID, or toggling tools before a cache breakpoint results in a 0% hit rate. Dynamic text must go after the static prefix.

Gemini Notebook

The Streaming Event Waterfall



Streaming improves perceived latency. It does not reduce tokens or cost.

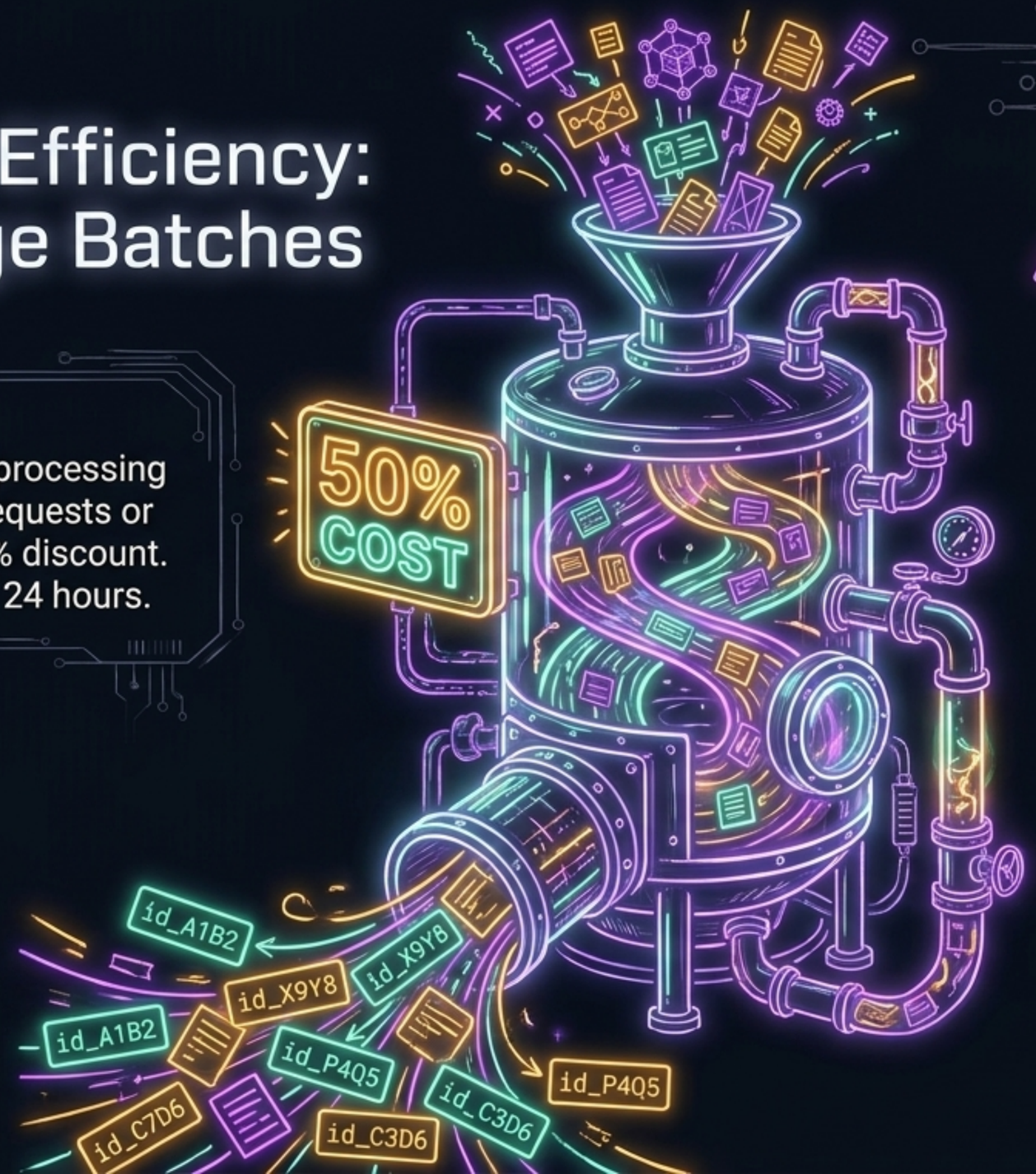
Interrupted Streams:
"The read loop ended" ≠ "the message is complete." On a drop, discard the partial turn entirely.

The Trap:
Saving a half-streamed assistant turn leaves a malformed tool_use in history, silently breaking all future validation on retries.

Gemini Notebook

Offline Efficiency: Message Batches

Mechanism
Asynchronous processing of up to 100k requests or 256MB at a 50% discount. Finished within 24 hours.

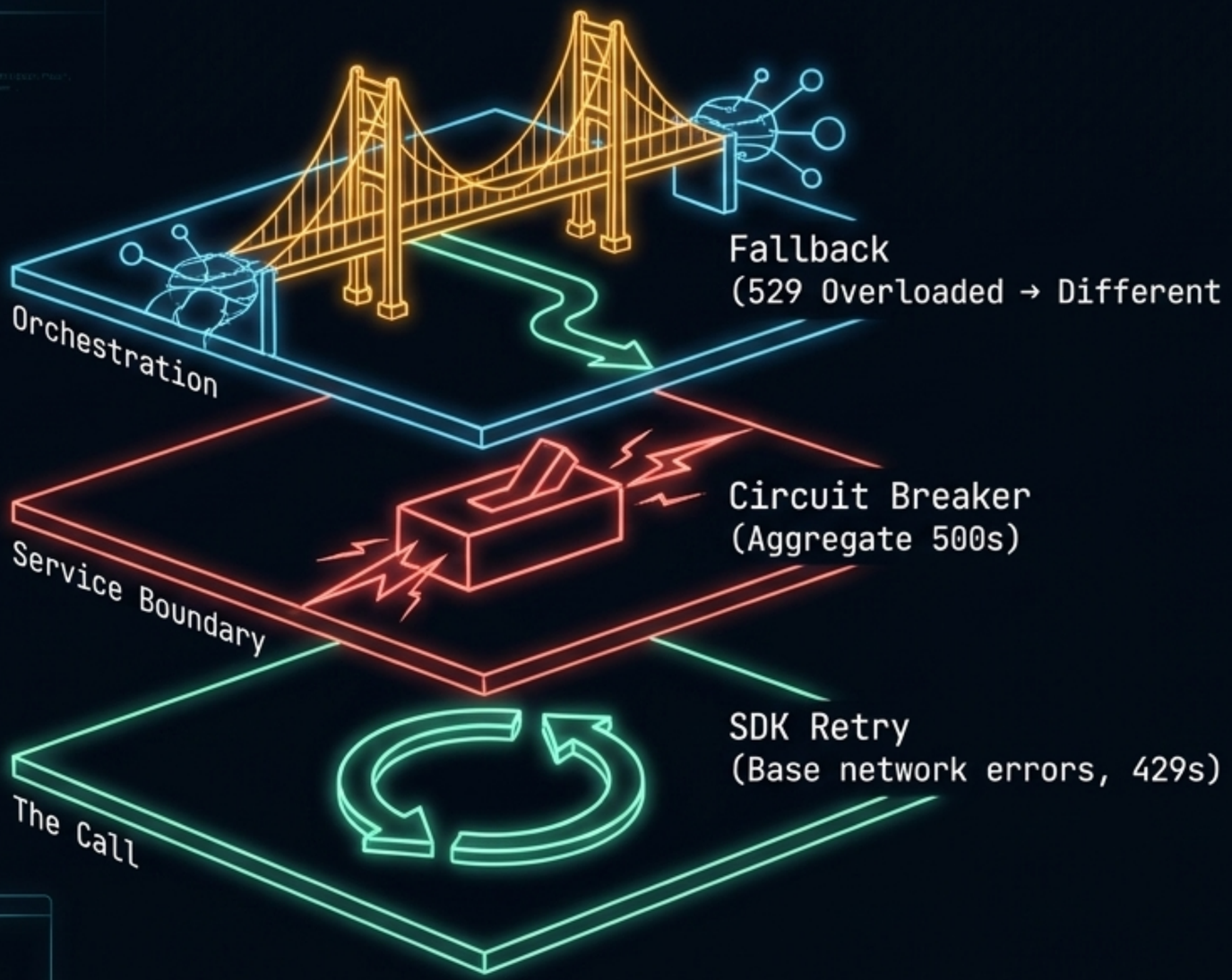


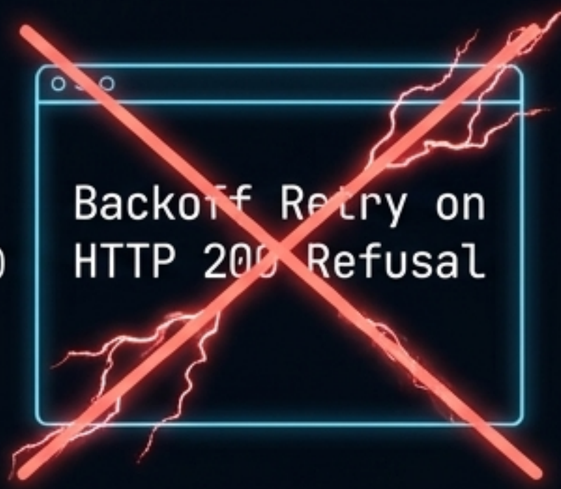
The `custom_id` Mandate
Results return entirely out of order. You must match results using the required `custom_id` (1-64 chars), never by array position.

The Trap
Splitting a list into chunks and calling `messages.create()` in a sync loop is not batching—it's just a rate-limit fight. Use the **Batches API** for bulk offline offline work.

Gemini Notebook

Error Handling & The Retry Circuit





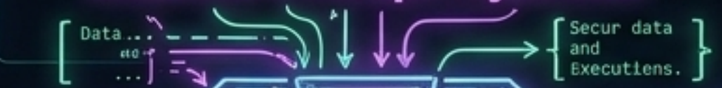
Ownership: SDK handles base retries. Never wrap SDK calls in a custom 5-attempt loop. One layer owns retries.

The Refusal Trap: A safety refusal returns as HTTP 200, not an error. It is a content decision. Exponential backoff on a refusal just repeats the same rejection. Read stop_details and rephrase or clear history.

Gemini Notebook

Security Posture & Realities

Indirect Prompt Injection

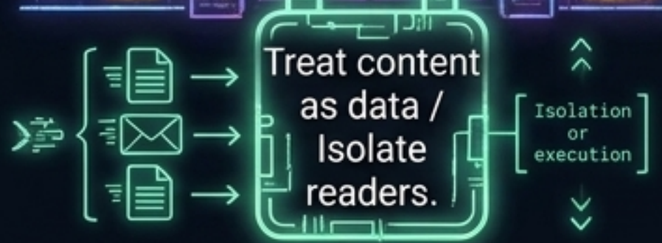


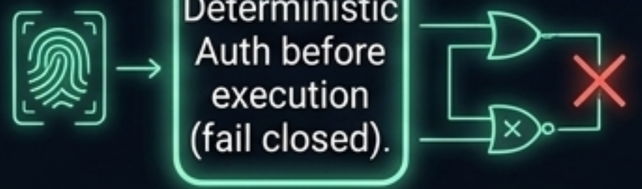
Side-effecting Tool Misuse

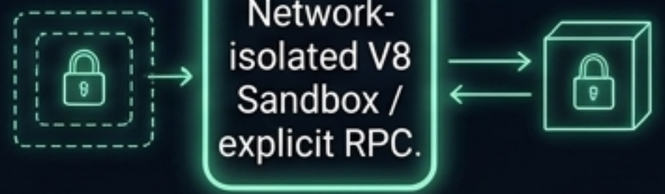
Untrusted CodeMode (MCP)



System Prompt 'Don't do X' is NOT a security control.







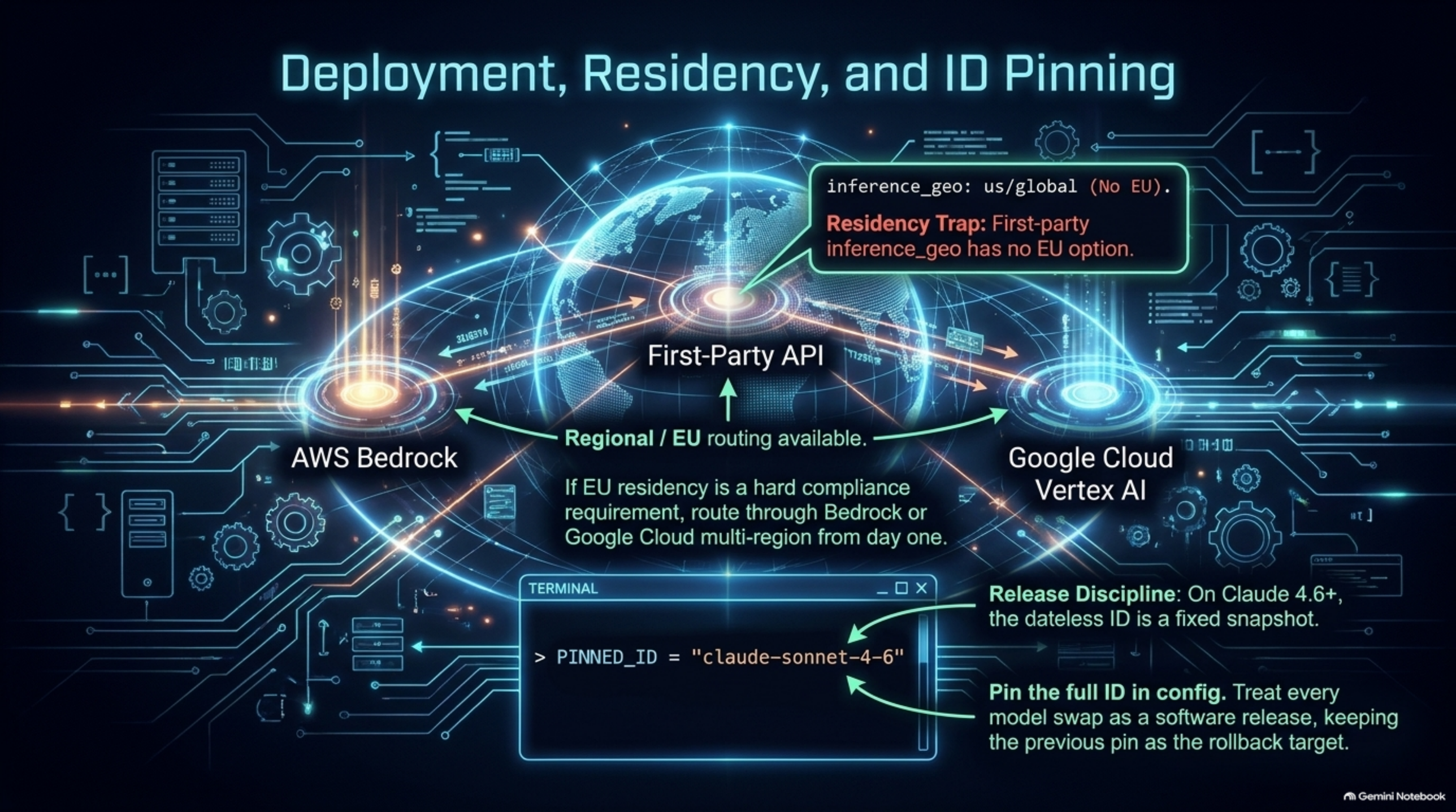
The Attacker:
In API apps, the attacker is whoever wrote the web page, email, or document the agent reads, not just the user.

The Trap:
Trusting model-generated arguments or assuming an output filter that runs after execution is sufficient.

The Fix:
Deterministic authorization must happen in your code before the tool executes. Use least-privilege registries.

Gemini Notebook

Deployment, Residency, and ID Pinning



Gemini Notebook

Evals and Cost Levers



Assert properties, never exact strings.

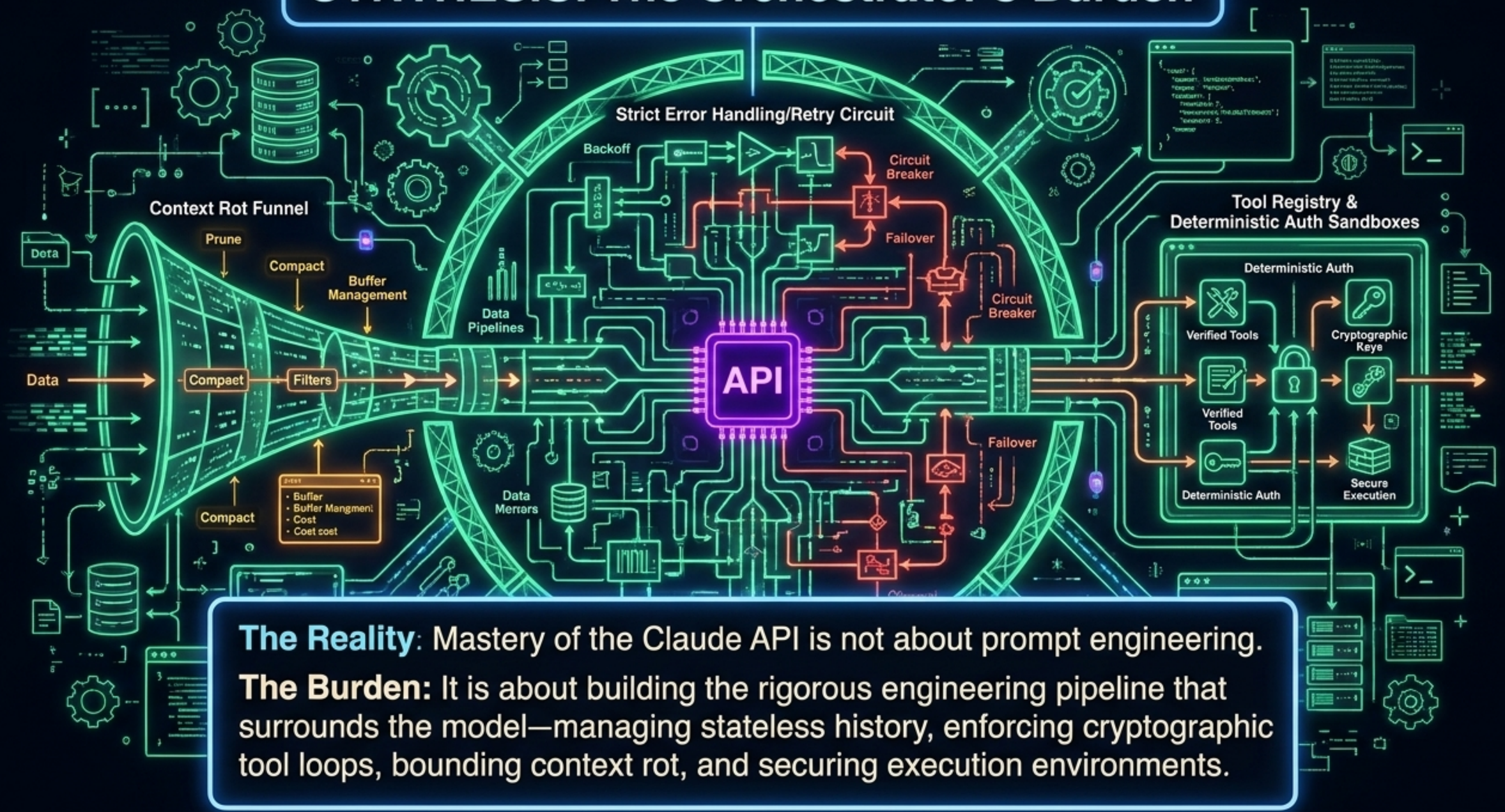
Eval Discipline: Assert properties and ranges in your eval code; do not grade against exact strings, as output naturally varies even when correct.

Golden Sets: Representative inputs + edge cases + adversarial cases. Refresh whenever prompts or retrievals change.

⚠ **The Cost Trap:** Quoting the per-call cost of a pilot. Multi-agent fan-outs multiply tokens exponentially. **Instrument latency and errors** per call from day one, and set reliability floors before pulling cost levers.

Gemini Notebook

SYNTHESIS: The Orchestrator's Burden

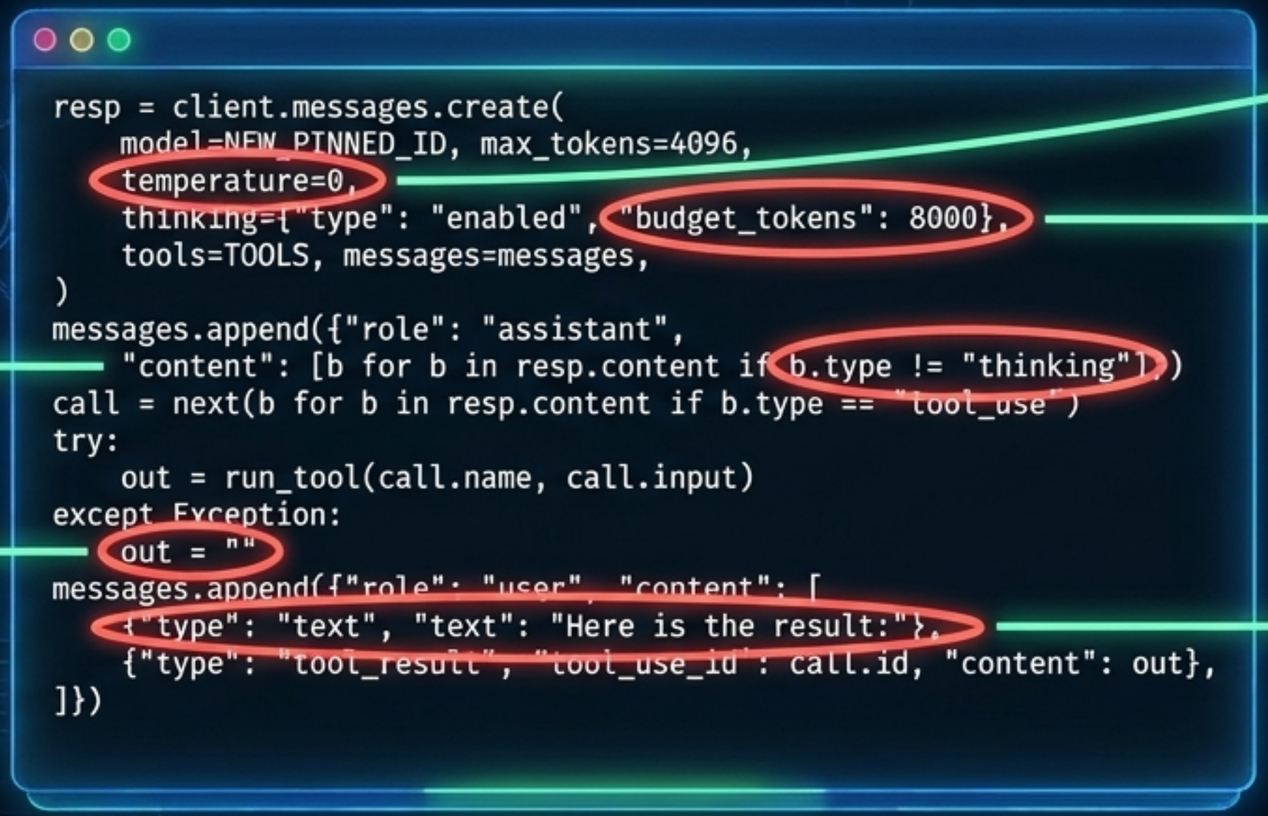


The Reality: Mastery of the Claude API is not about prompt engineering.

The Burden: It is about building the rigorous engineering pipeline that surrounds the model—managing stateless history, enforcing cryptographic tool loops, bounding context rot, and securing execution environments.

Gemini Notebook

Find The Defect: Code Migration Breakdown



Crossed out. 400 error on 4.7+. Omit it.

Crossed out. Deprecated. Use adaptive + effort.

Crossed out. Breaks crypto signature. Echo completely intact.

Crossed out. Causes hallucinations. Return is_error: true.

Crossed out. tool_result block must be strictly first.

Conclusion: Production stability requires strict adherence to the stateless rules, exact payload schemas, and chronological tool pairing. The model provides the intelligence; you provide the discipline.

Gemini Notebook