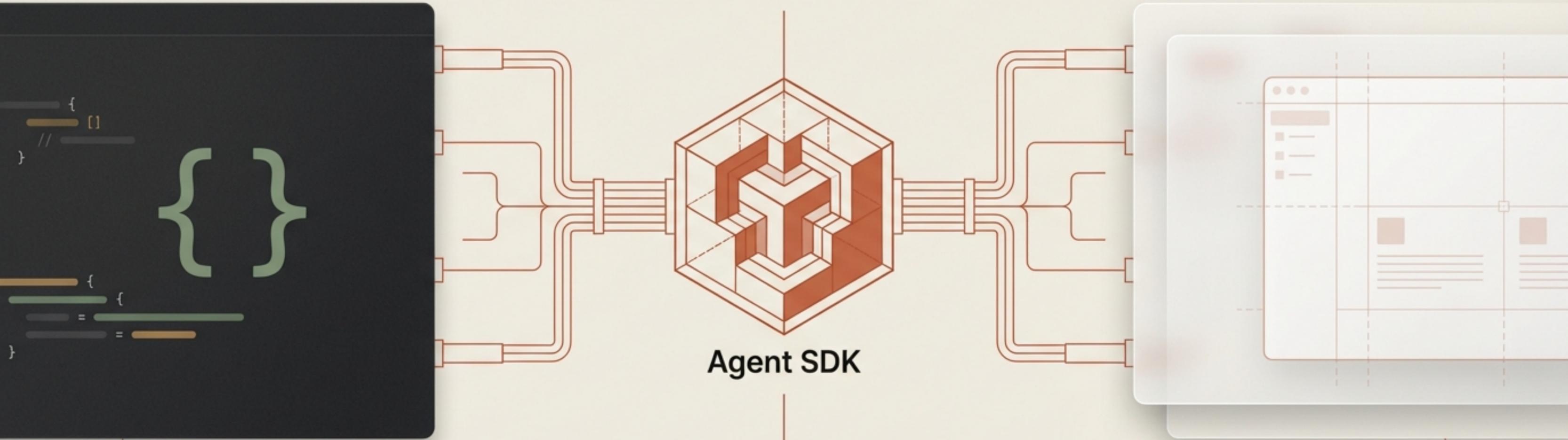


The Dawn of General Agents

How Claude Code and Cowork turned conversational LLMs into autonomous Digital FTEs.

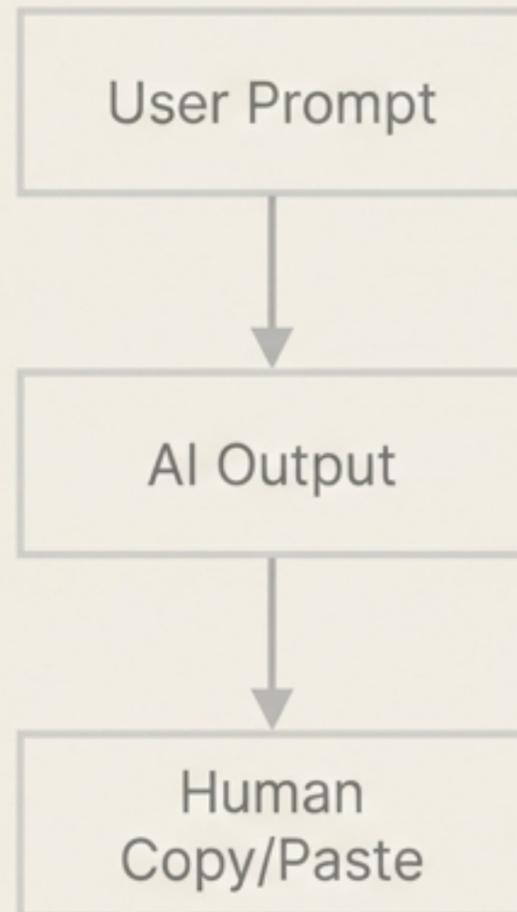


A strategic overview for business leaders, technical managers, and knowledge workers navigating the 2026 poly-agentic landscape.
Stop building agents. Start building skills.

The Paradigm Shift: From Passive Generation to Agentic Execution

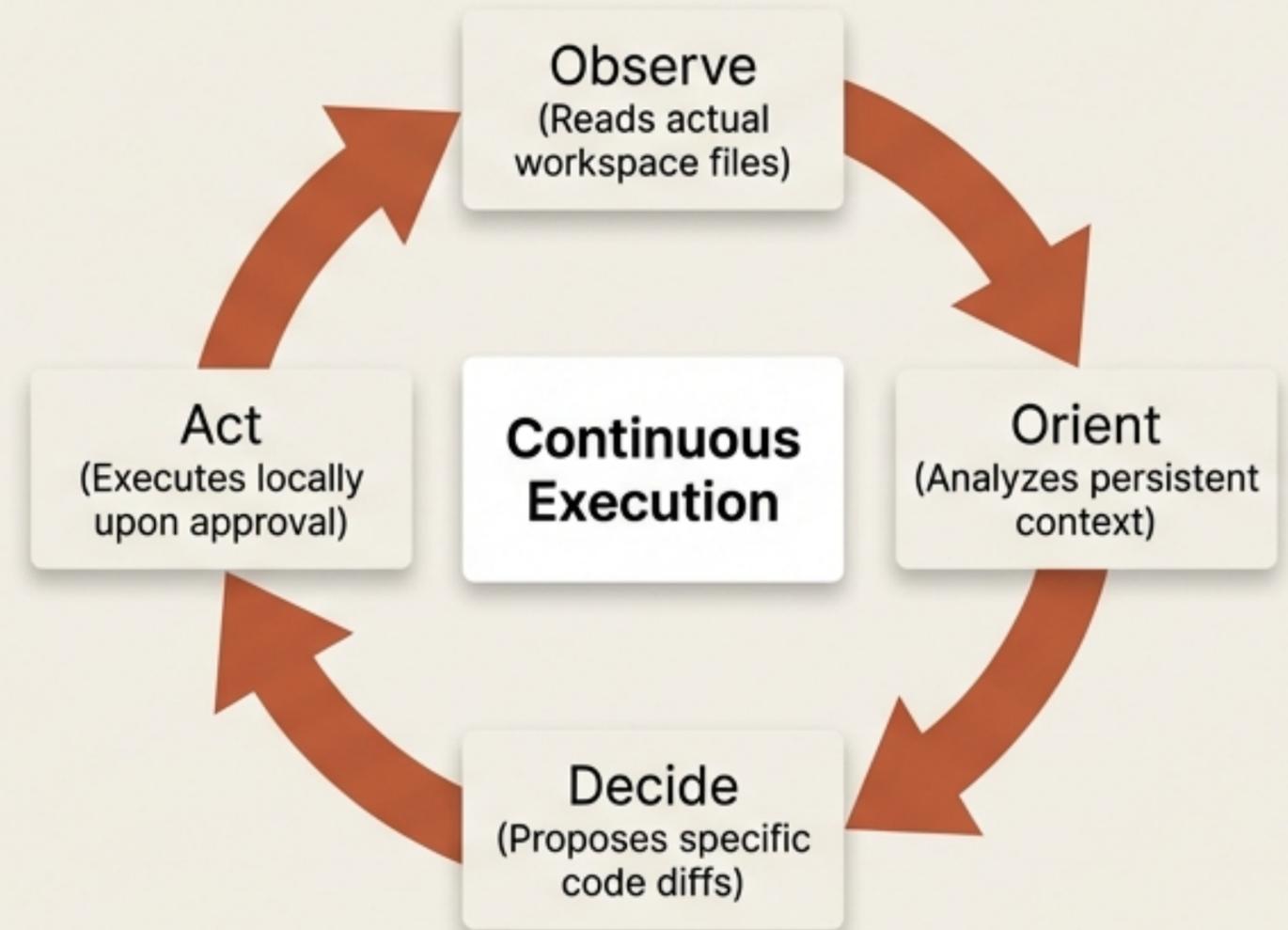
The breakthrough was Product Overhang: the capability to act as a partner already existed, waiting for filesystem access to unlock it.

Passive Chat



AI as a chat box.

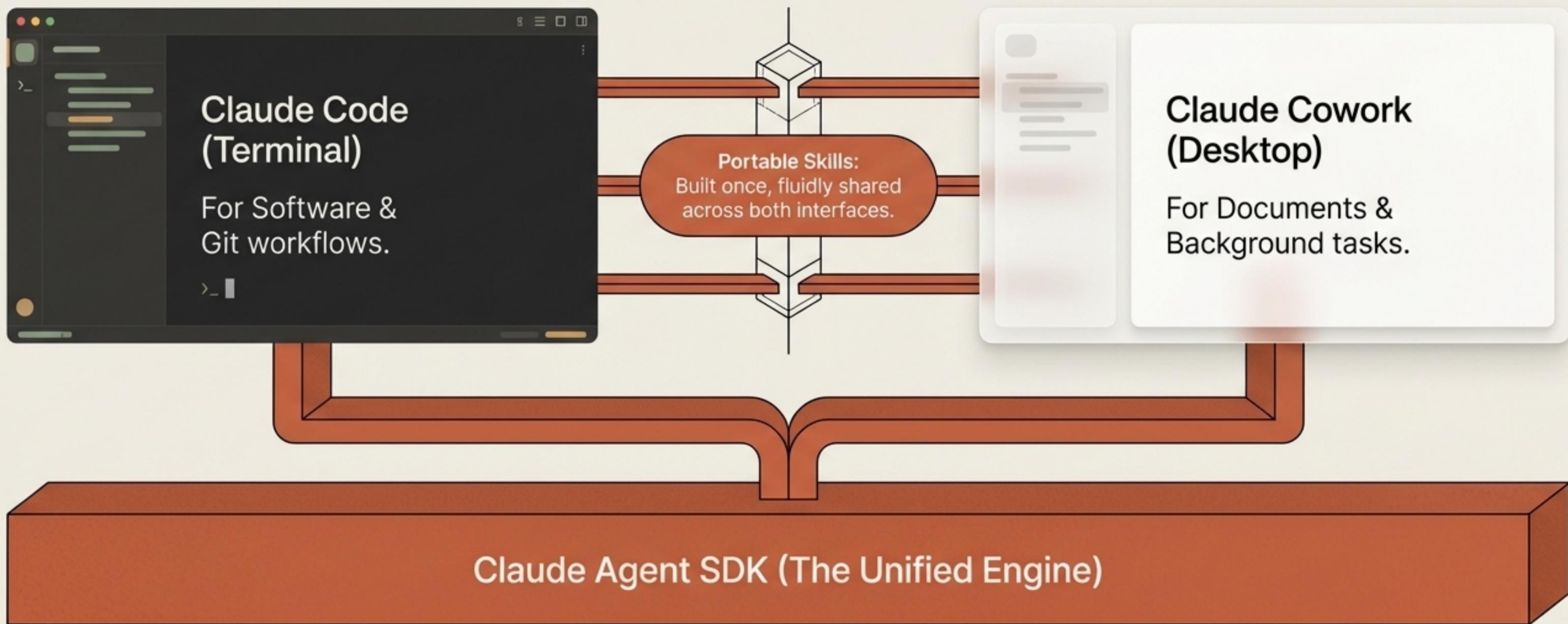
Agentic Execution



AI as an agentic system.

One Shared Engine, Two Distinct Workspaces

The choice of interface dictates the type of work, not the capability of the AI. The shared foundation is the key to enterprise automation.



Claude Code: The Developer's Agentic Copilot

Origin

Prototype: Sept 2024.
General Availability: May 2025.
90% of Claude Code was written
by Claude Code itself.

Scale

20% Day 1 adoption.
>80% daily usage.
Driving \$500M+ annual run-rate.

Context

Solves LLM statelessness via
CLAUDE.md (<200 lines) loaded
at session start.

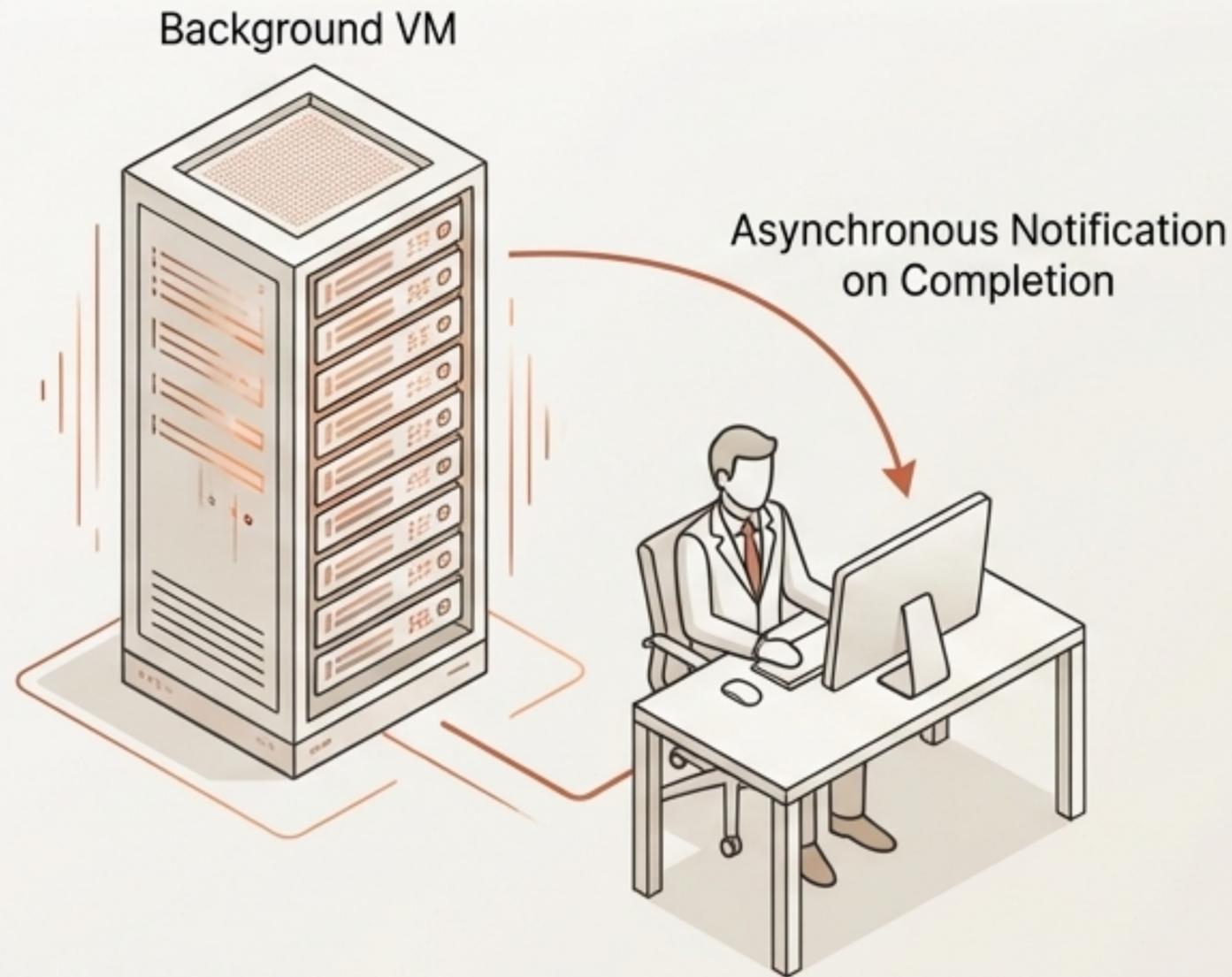
```
function initialize_service(config) {  
  const db = await connect_to_database(config.db_url);  
  const cache = await connect_to_cache(config.cache_url);  
  
- return { db, cache };  
+ const logger = new Logger(config.log_level);  
+ return { db, cache, logger };  
}
```

Describe task -> Claude reads files -> Proposes diffs

[Y/n] Approve execution? ← Trust through transparency.

Claude Cowork: The Autonomous Knowledge Worker

Not an interactive tool, but an autonomous background agent. Describe the outcome, not the steps.



Native Document Parsing:

Handles docx, xlsx, pptx, and pdf effortlessly.



Mobile Dispatch: Assign tasks on-the-go from iOS.



Scheduled Tasks: Set up recurring autonomous automation.



Computer Use: macOS screen control research preview.

Navigating the Dual Interfaces

Claude Code

Target User: Software Developers / DevOps

Primary Interface: CLI / Terminal / VS Code

File Types: Codebases, Git branches, configs

Core Mechanics: Synchronous, fast local execution

Claude Cowork

Target User: Knowledge Workers / Analysts

Primary Interface: Desktop App / Chrome Extension

File Types: Documents, Spreadsheets, Presentations, PDFs

Core Mechanics: Asynchronous background cloud VMs

The Golden Rule: Use Code for software and Git. Use Cowork for documents, reports, and background tasks.

The Computing Stack Analogy

Stop trying to build custom agents from scratch. Build Skills instead.

Skills (The Apps)

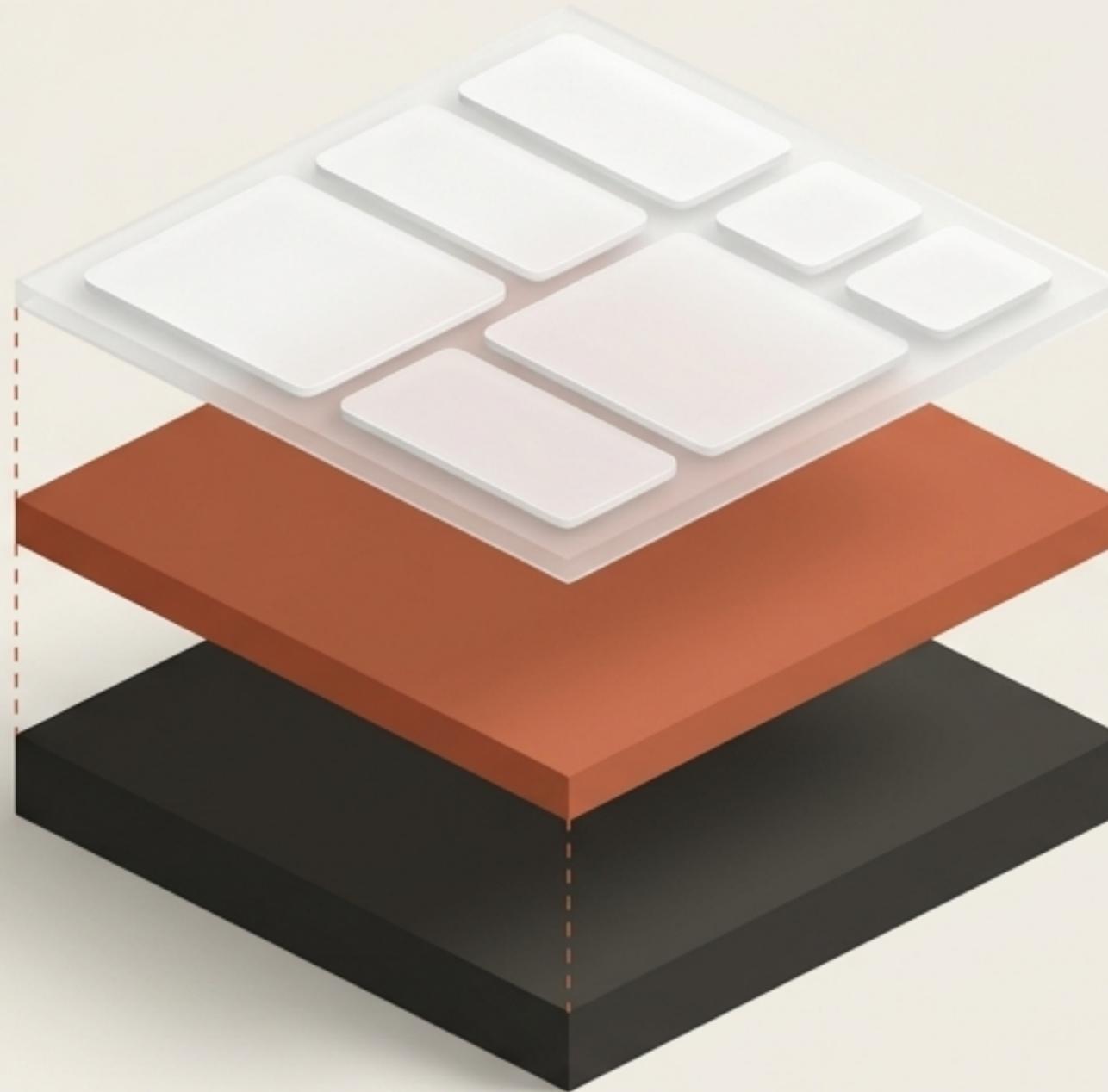
Installable domain expertise that actually does the work.

Agent Runtimes (The OS)

Code/Cowork providing execution and filesystem access.

Models (The CPU)

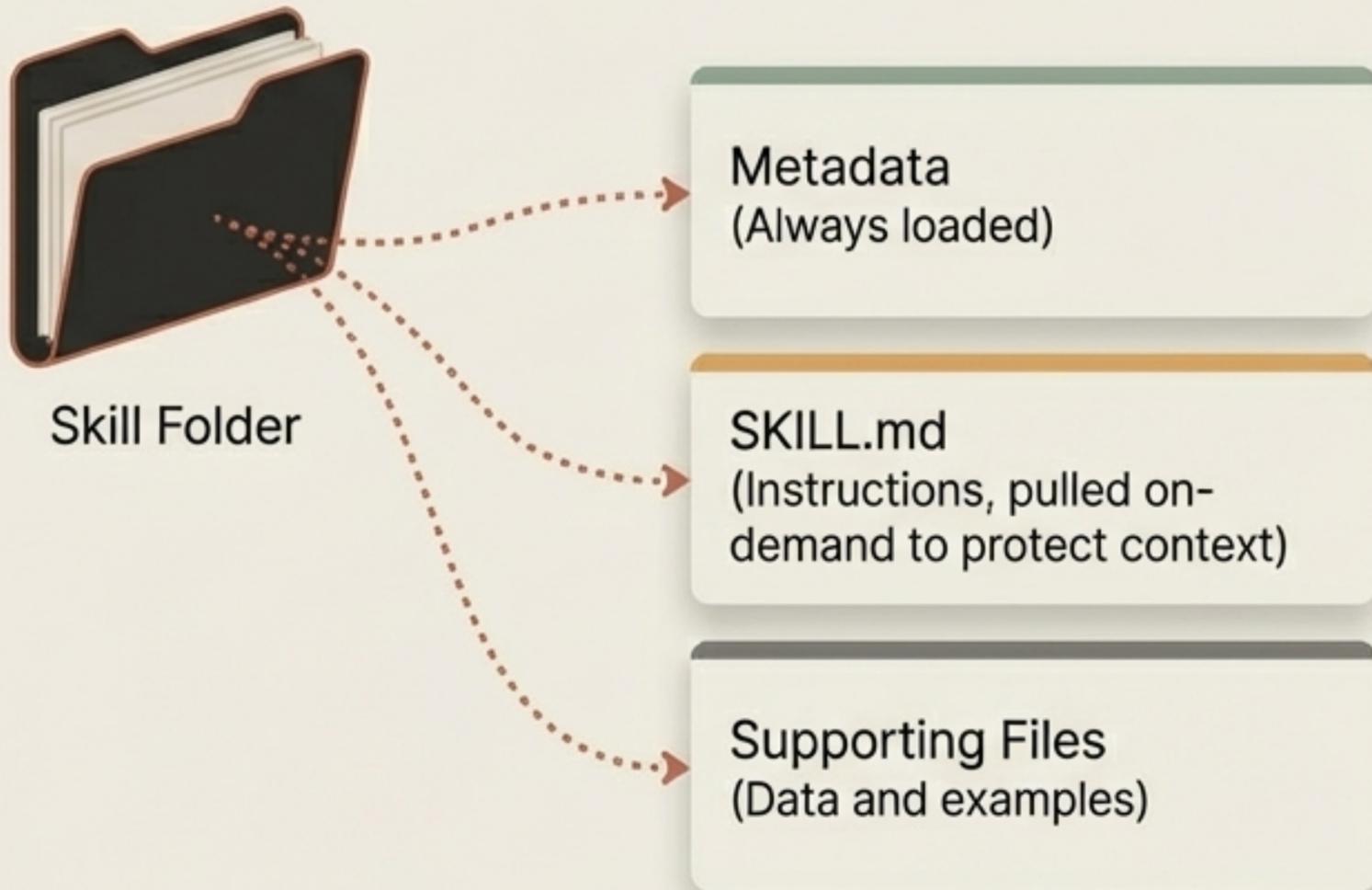
Raw processing power and foundation intelligence.



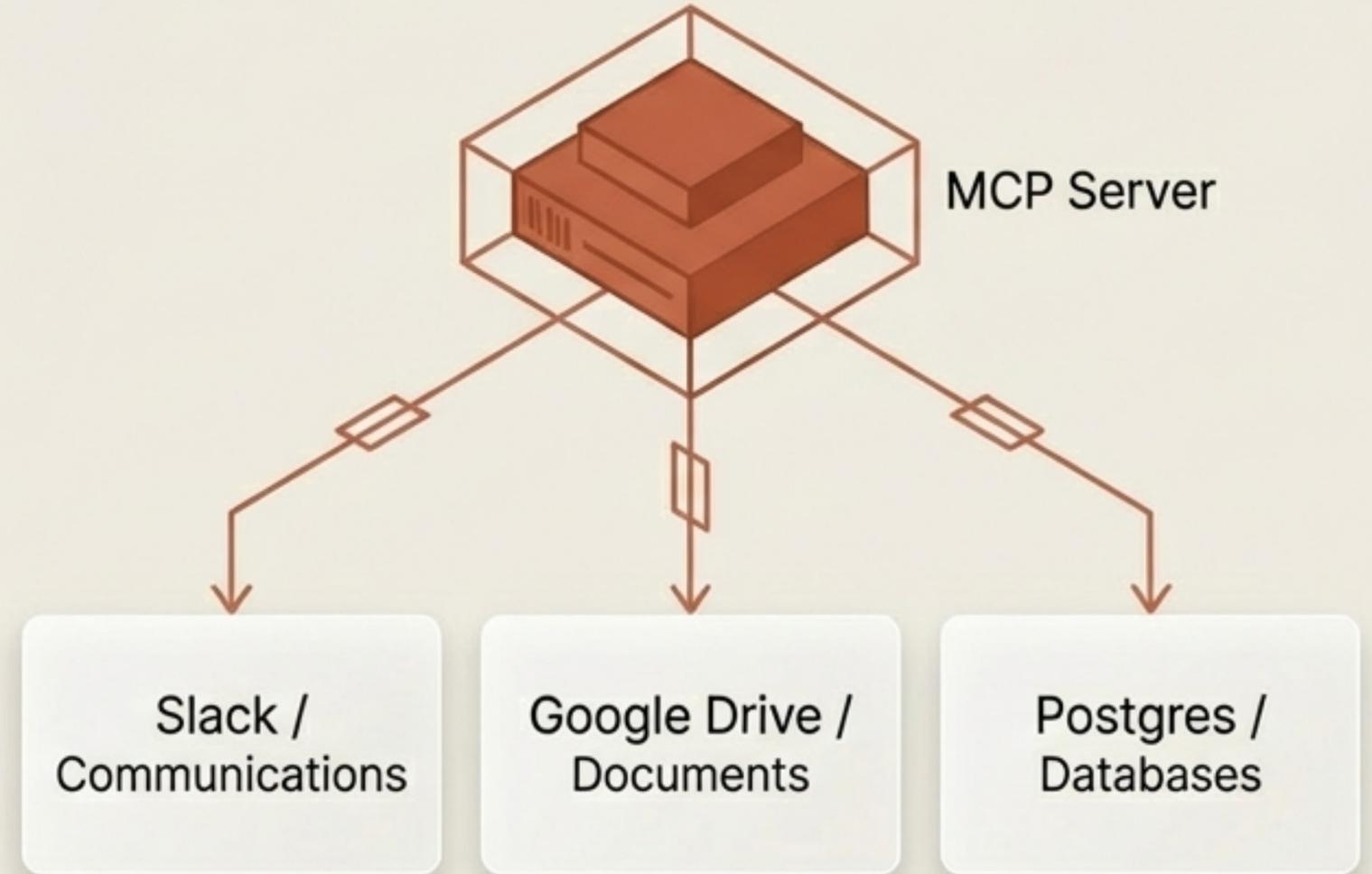
The Logic: Intelligence (models) + Code (execution) = an agent. But the missing piece is domain expertise. Skills act as the installable applications that encode your specific business workflows.

The Architecture of Agent Expertise

Skill Anatomy

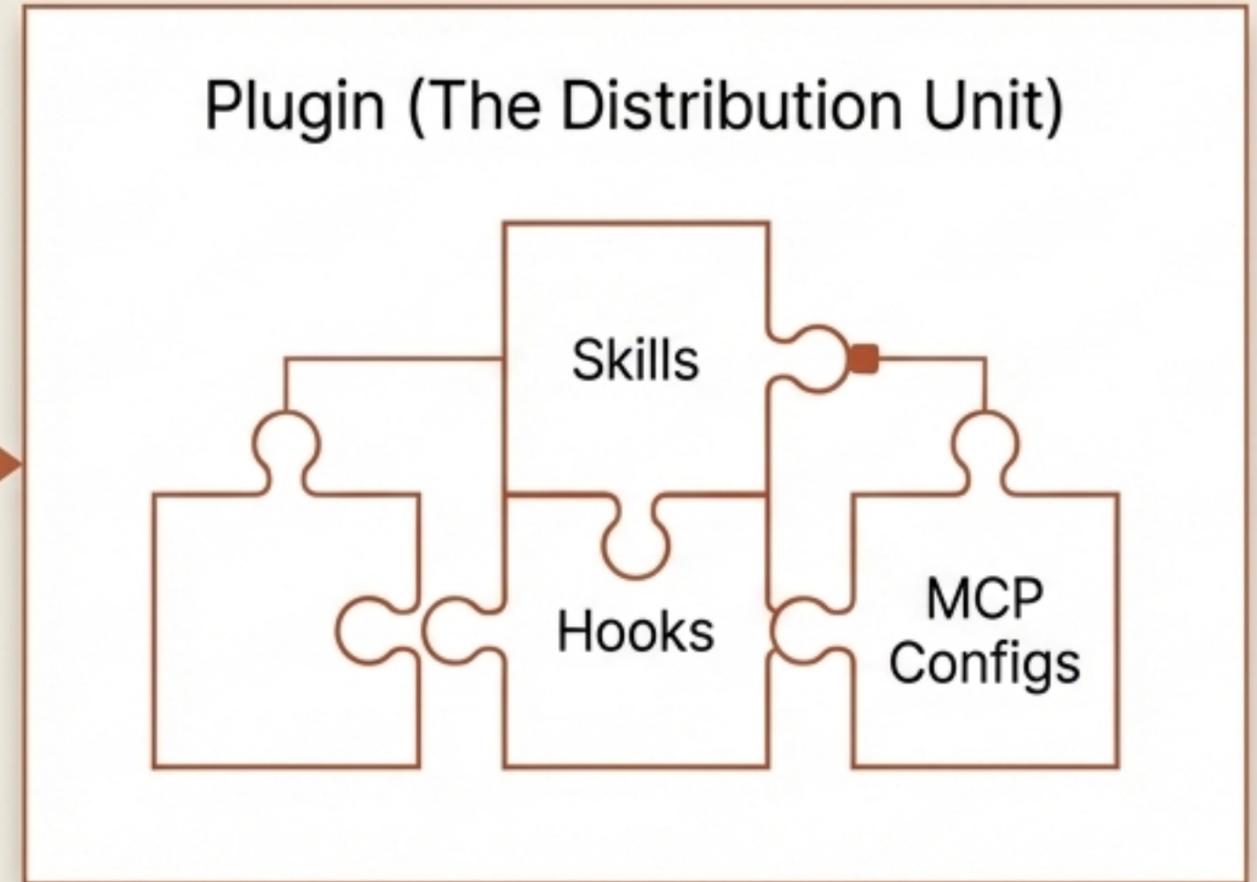
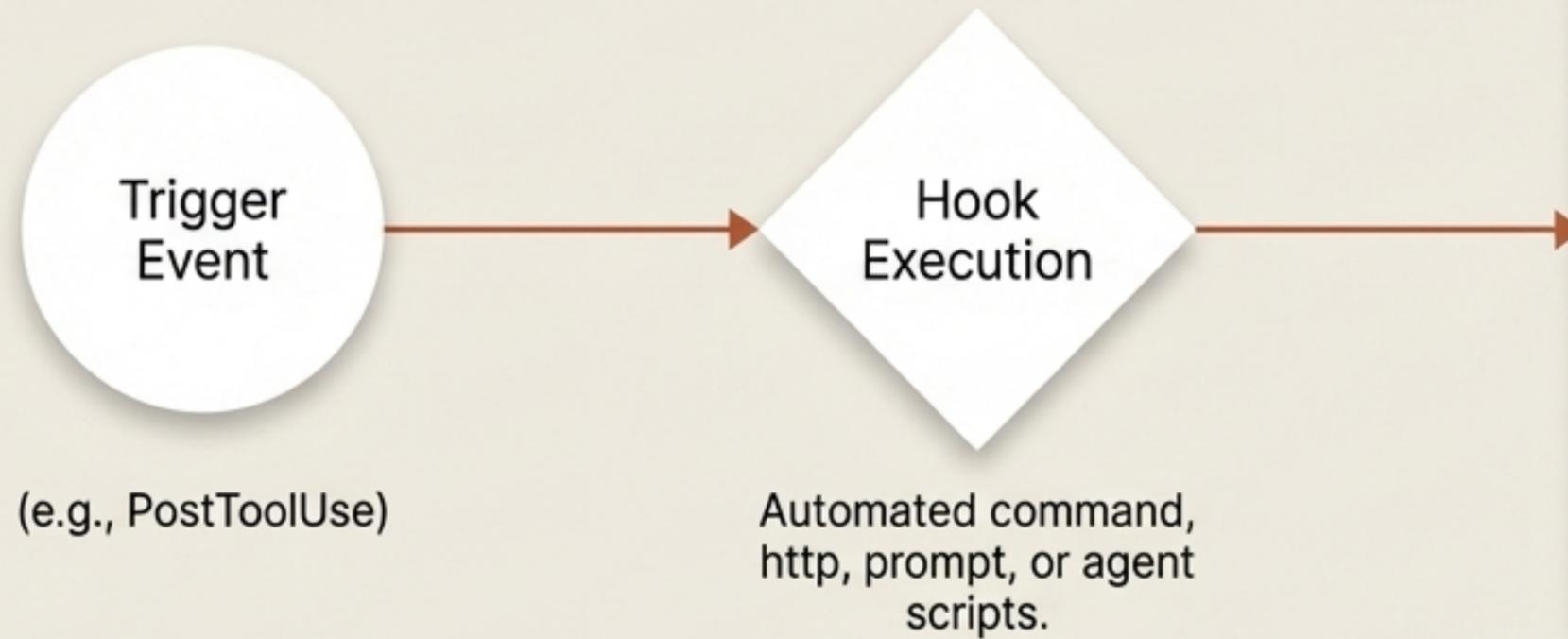


The MCP Connection



Token Economics: Compiling frequent MCP server patterns into dedicated Skills reduces token usage by 80-98%.

Extensibility: Composing the Automation Stack



Hooks

Event-driven automation triggered by system states.

Skills & MCP

Add new capability, external data, and domain expertise.

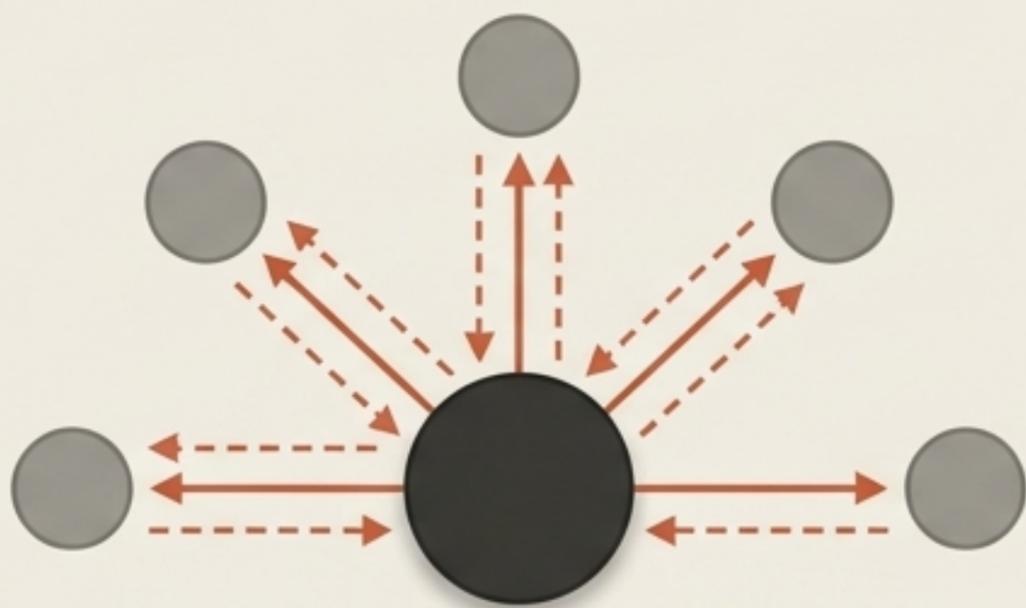
Plugins

Portable enterprise packages bundling Hooks and Skills to share best practices instantly.

The Formula: Hooks add automation. **Skills** add capability. **Plugins** package them together.

Multi-Agent Orchestration Topologies

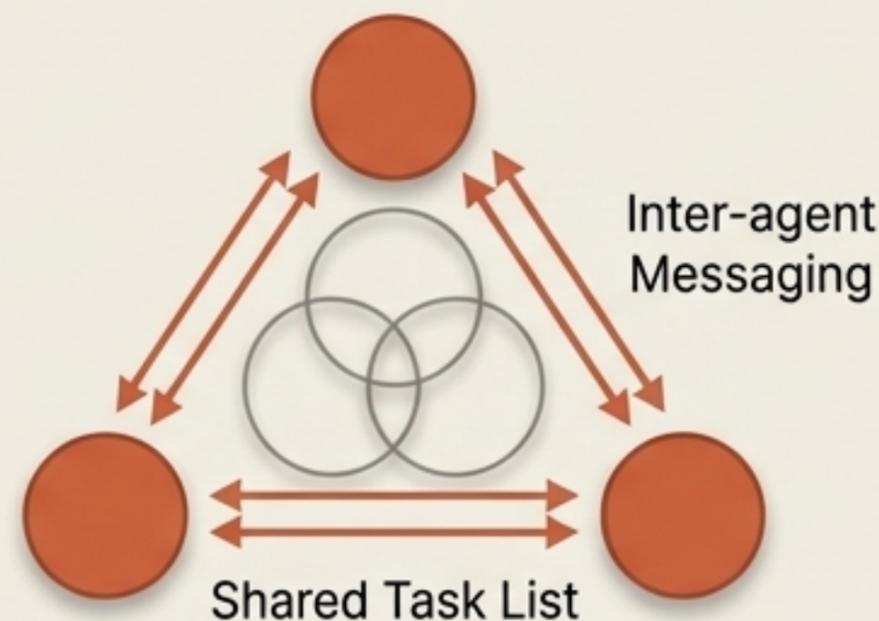
Hub-and-Spoke Subagents



Fire-and-Forget Delegation

Claude automatically delegates specialized tasks. Subagents operate with clean context and report solely back to the caller.

Independent Agent Teams



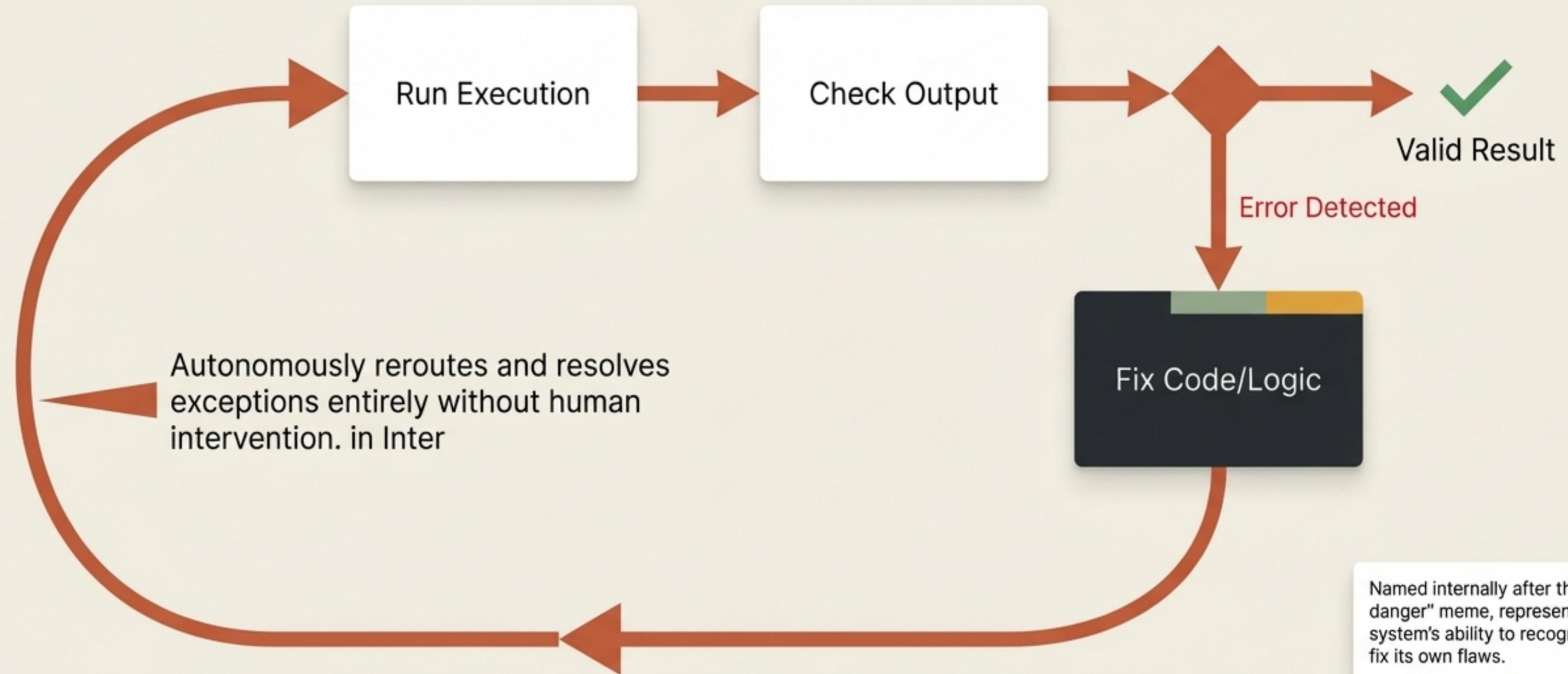
Parallel Investigation

Fully independent sessions coordinate via shared memory workspaces. Supported by Git worktrees for safe isolation.

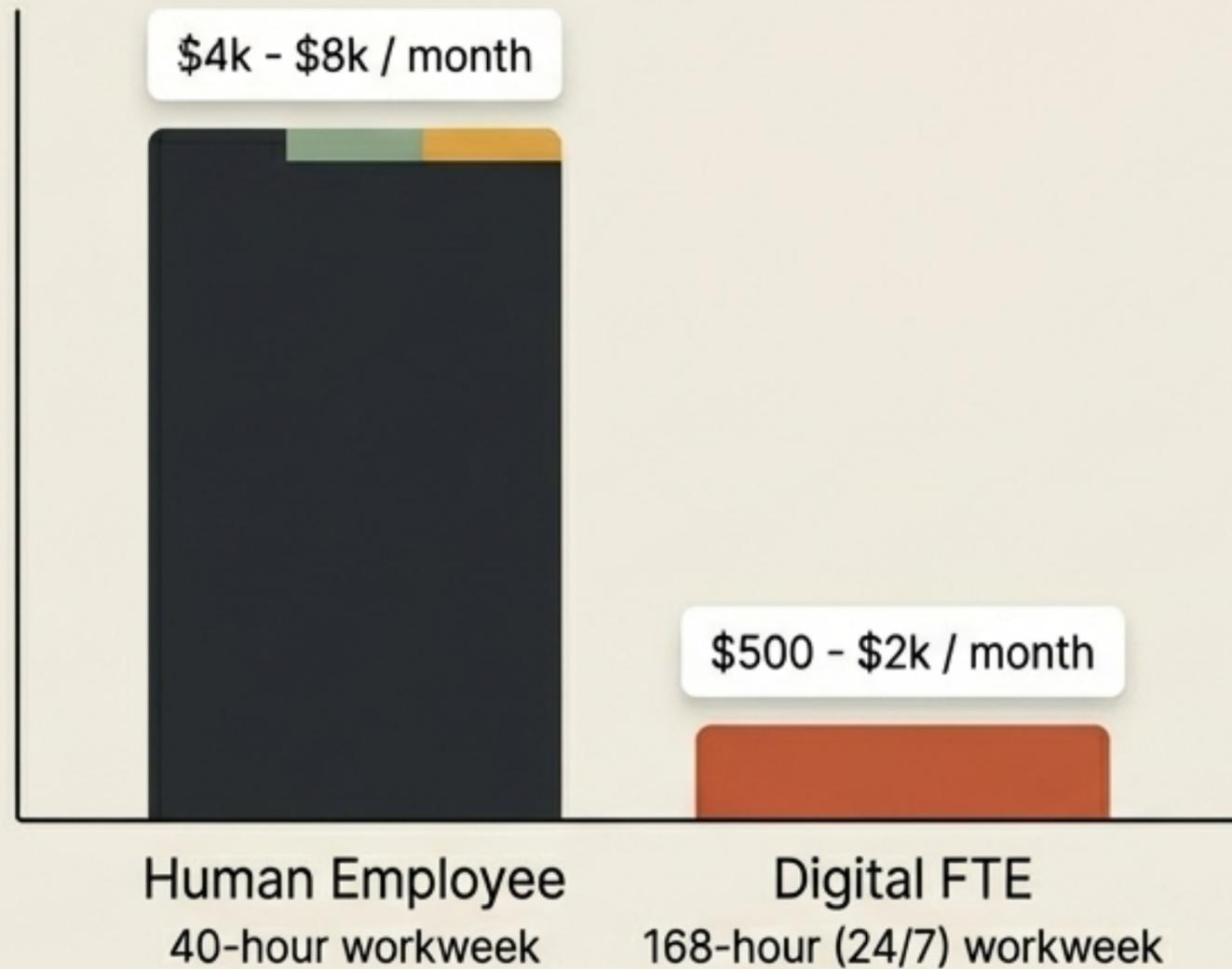
The Tradeoff: Agent Teams use 3-5x more tokens, but parallel investigation drastically improves output quality.

The Ralph Wiggum Loop: Autonomous Self-Correction

The difference between a tool that halts on an error and an employee that solves it.



The Economics of the Digital FTE



$$\text{Pricing} = \frac{\text{(Monthly savings from automation)}}{3}$$

Four Revenue Models for Skills as IP

Subscription.

\$500-\$2k/mo for continuous access to the specialized FTE.

Success Fee.

Pay per verifiable result and completed workflow.

License.

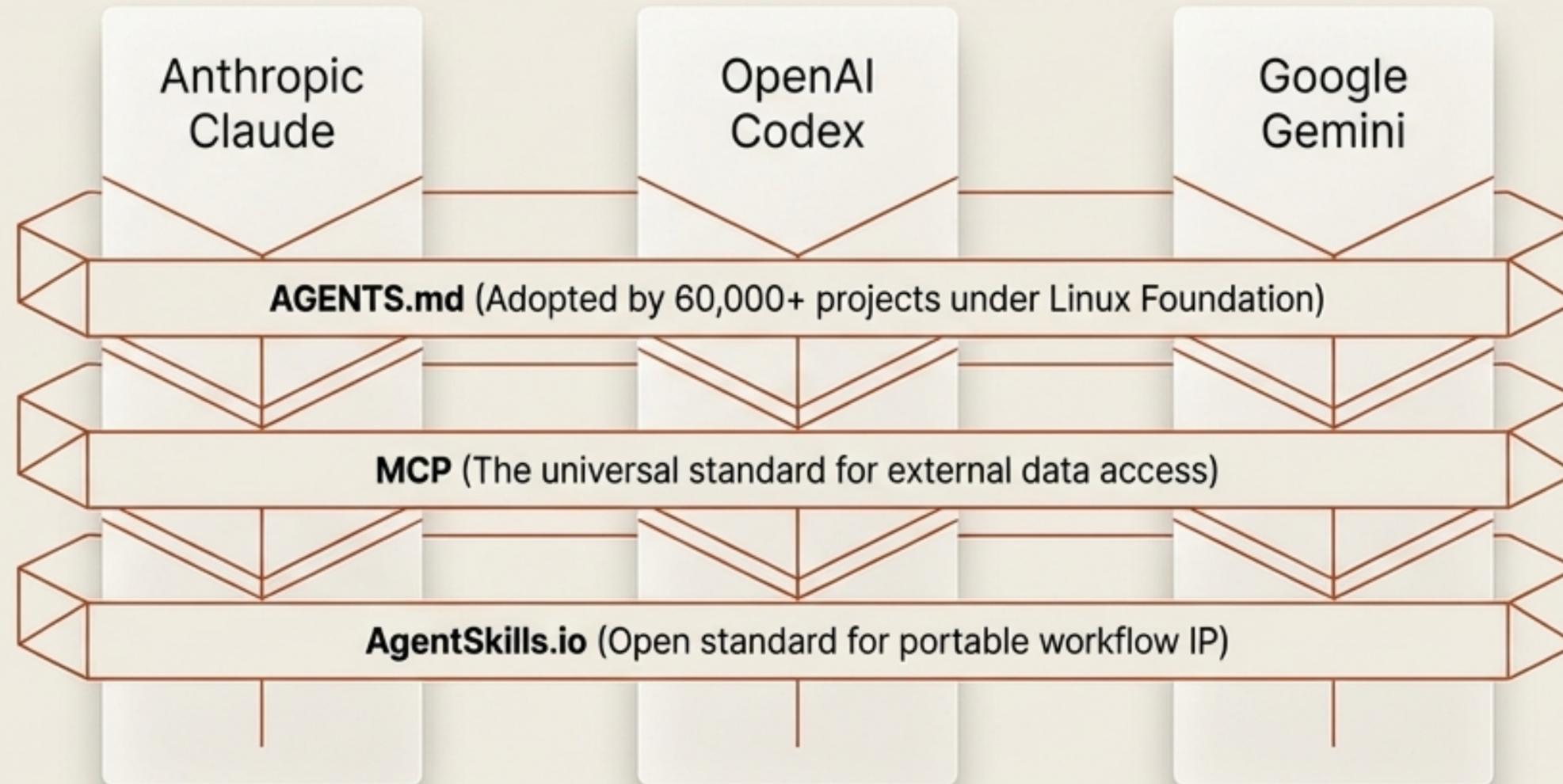
Sell the underlying IP recipe and architecture (\$50k+/year).

Marketplace.

High-volume, low-friction distribution of workflow packages.

The Universal Interoperability Web

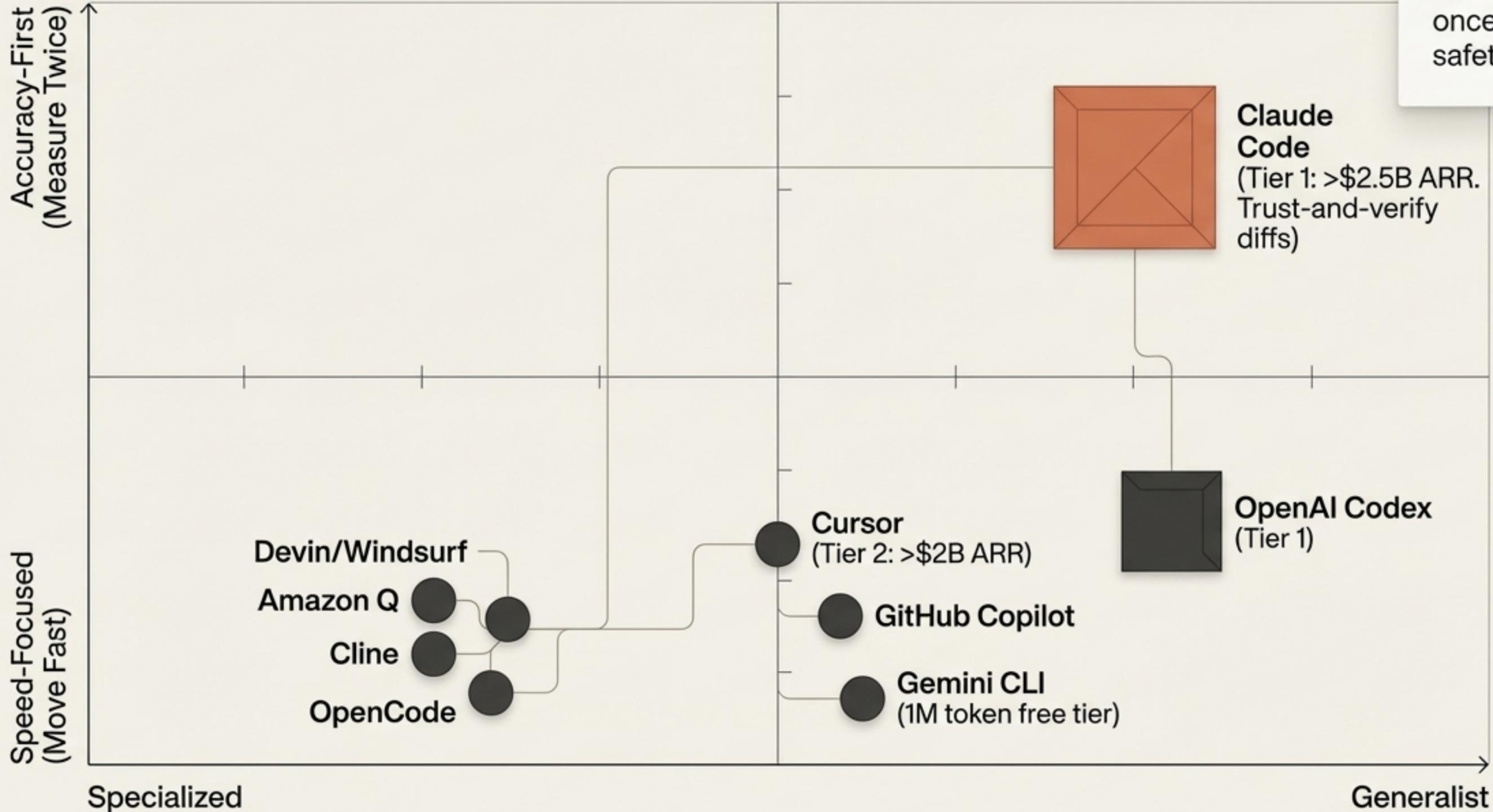
Skills built are not vendor-specific; they are industry-standard patterns.



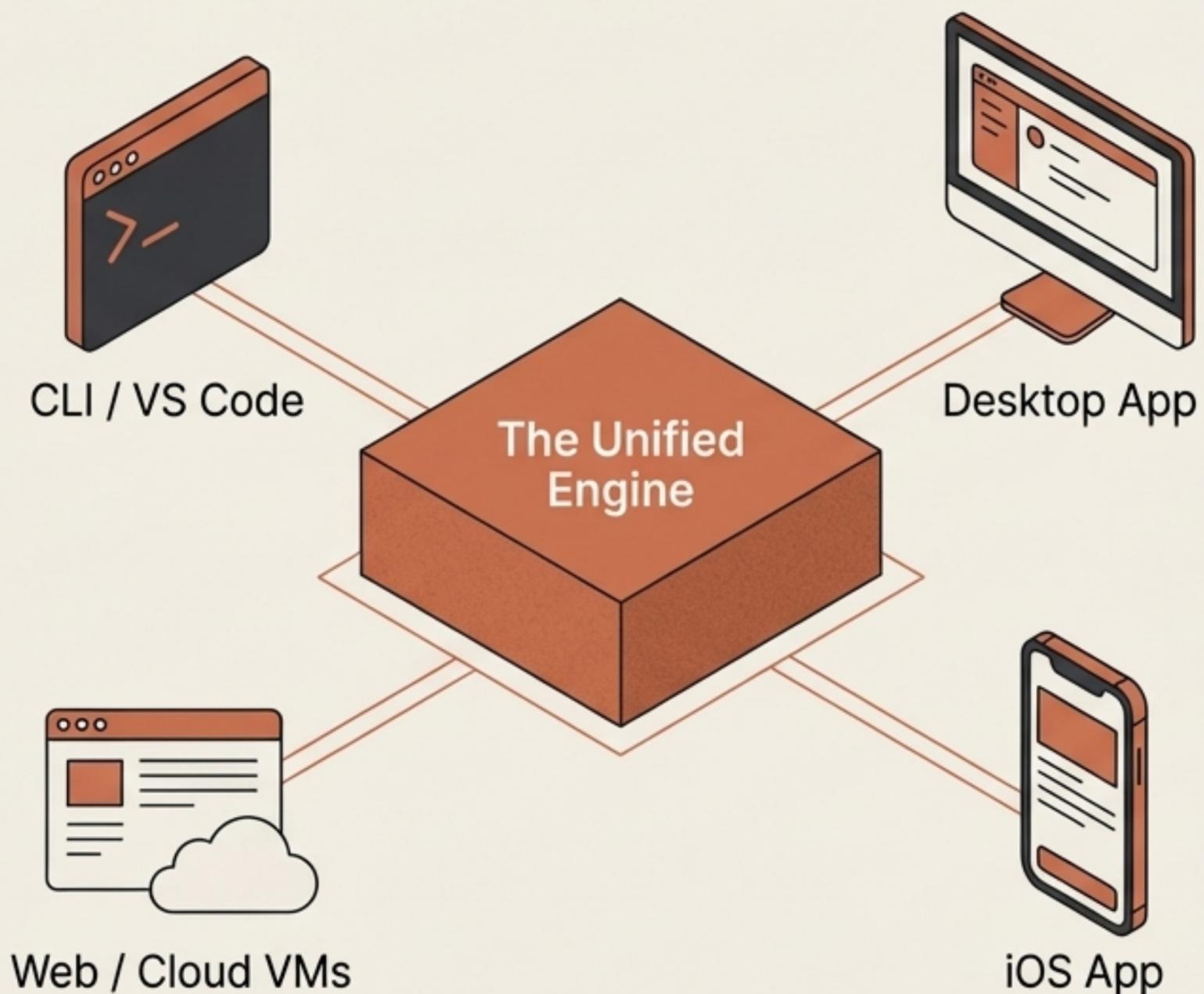
Portability: An investment in these standard protocols makes a company's intellectual property entirely portable across the entire poly-agentic landscape.

The March 2026 Vendor Landscape

The Philosophy:
Claude focuses on
"Measure twice, cut
once," prioritizing
safety and precision.



The Strategic Imperative: Build for the Poly-Agentive Future



The Blurring Boundary:

The interface matters less than the underlying architecture.

The Mandate:

The defining competitive advantage is not the LLM you use, but the proprietary Skills you build on top of universal standards.

Final Takeaway:

Stop managing AI as software. Start managing it as an autonomous workforce.