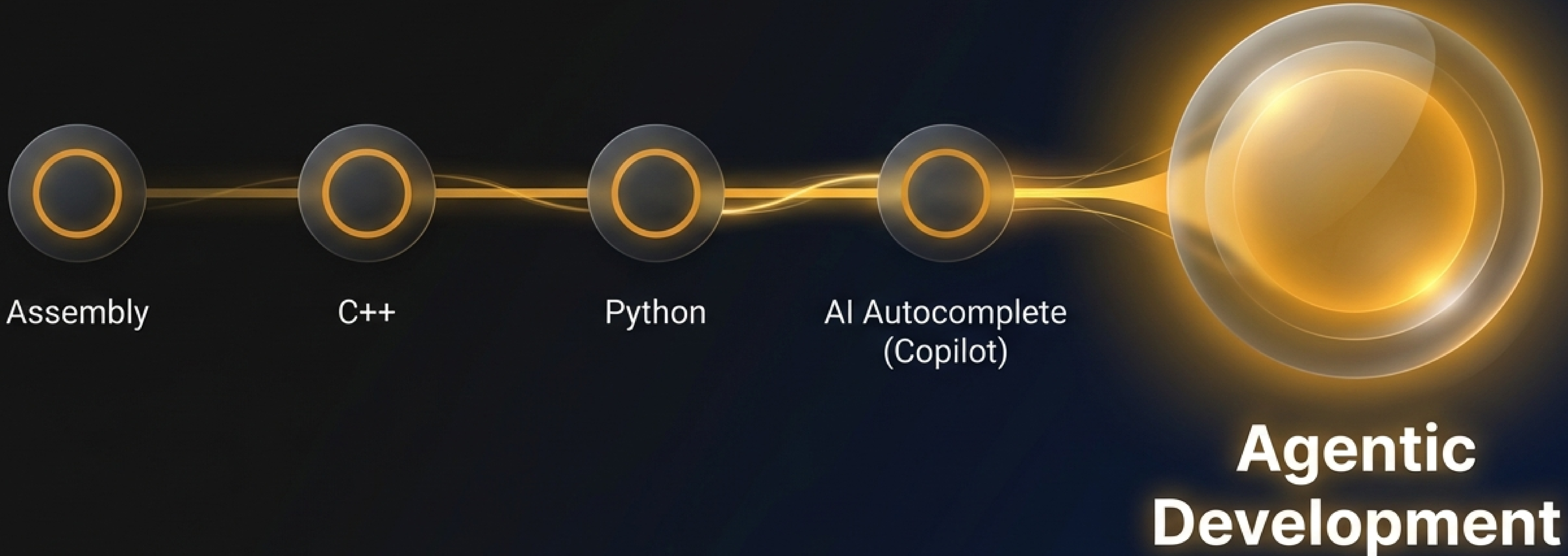
From Coder to Orchestrator

The paradigm shift toward agentic software development and the architecture of Google Antigravity.

The Abstraction of Syntax

The barrier to entry for building complex software has systematically dropped. We are no longer limited by how fast we can type syntax, but by how well we can direct intelligence.

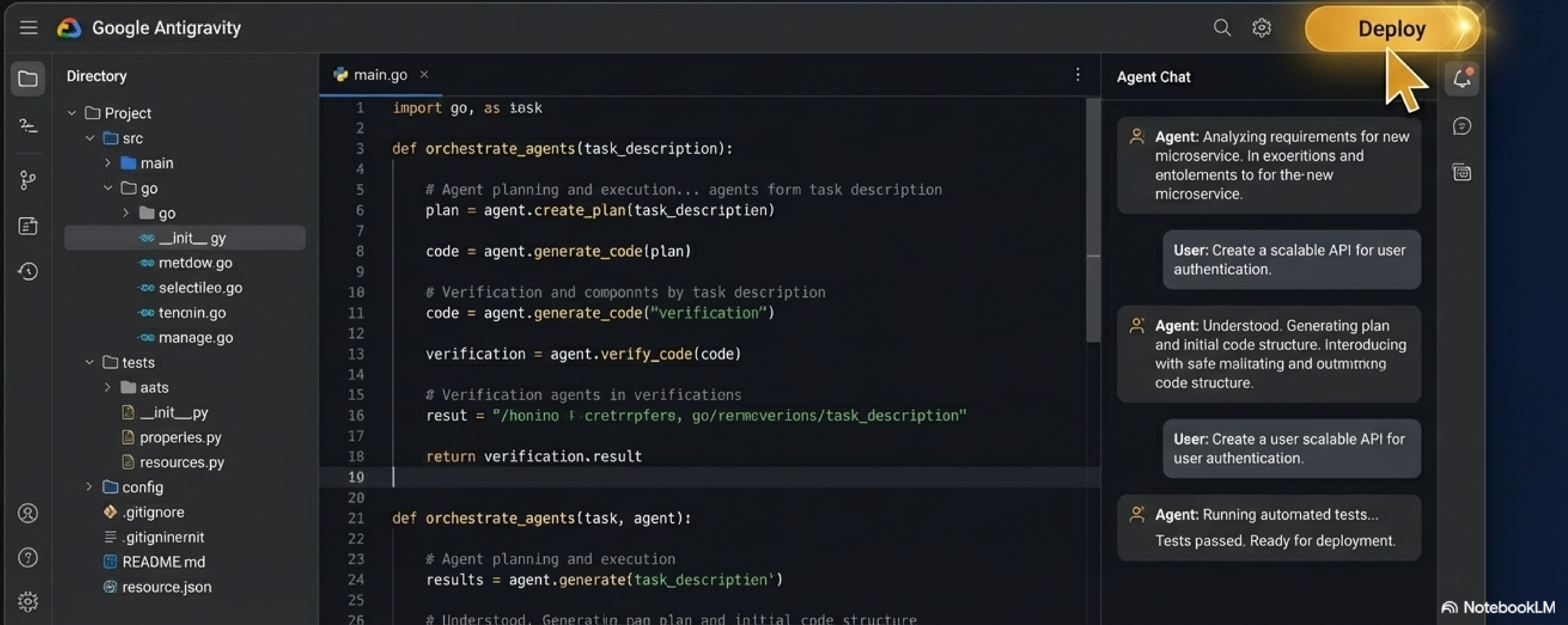


You Have Been Promoted

Developers spend only 20% of their time writing code. The rest is spent planning, reviewing, debugging, and deploying. The new workflow treats the developer as a Manager of Agents, delegating execution to focus purely on architecture and vision.

Meet Google Antigravity

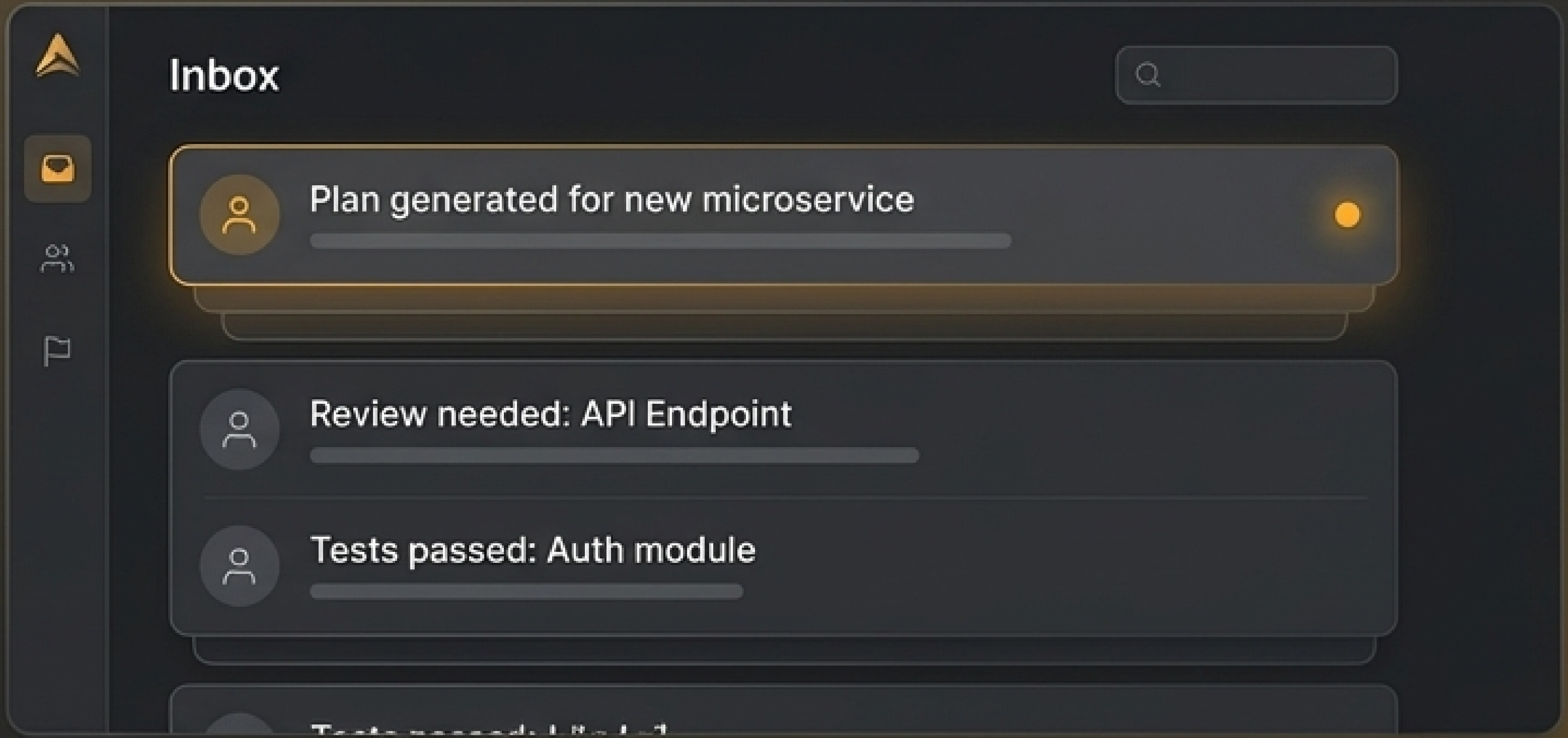
The world's first truly agent-first development environment. Built to help you orchestrate autonomous agents that plan, code, test, and verify complex applications from scratch.



The Holy Trinity of Agentic IDEs

Antigravity replaces the traditional code window with a three-pronged orchestration layer: Mission Control for your agents, a live workspace for the underlying code, and a fully actuated browser for visual verification.

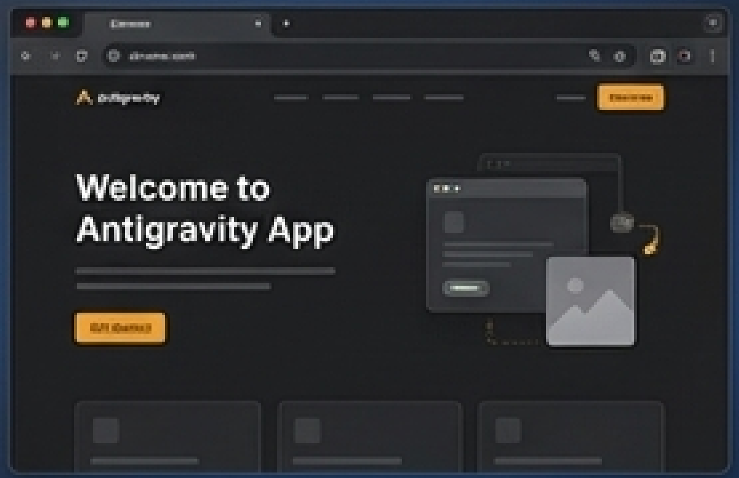
Agent Manager



The Workspace

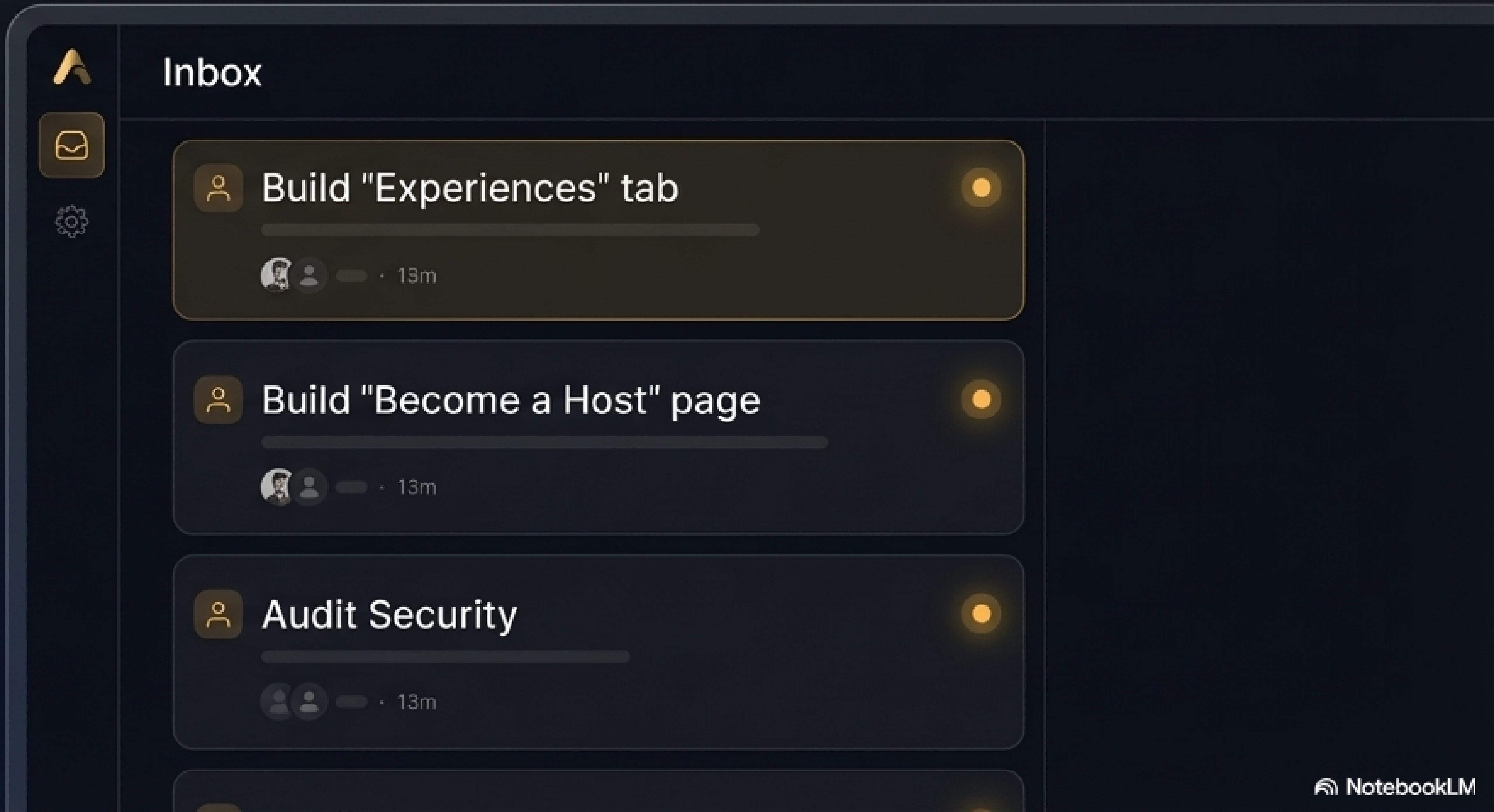
```
const agentPlanning = async (taskDescription) => {
  // Agents form task description
  const plan = await agent.createPlan(taskDescriptien);

  // Verification and components
  const code = await agent.generateCode(plan);

  // Verificatien agents in verificatien
  const verification = await agent.verifyCode(code);

  return verification.result;
};
```

Browser Actuation



Multi-Threading Your Brain

Stop waiting for a single prompt to finish. Antigravity allows you to orchestrate tens or hundreds of agents in parallel across a single workspace. Manage them all from a unified, asynchronous Inbox.



The Fighter Jet and the Pilots

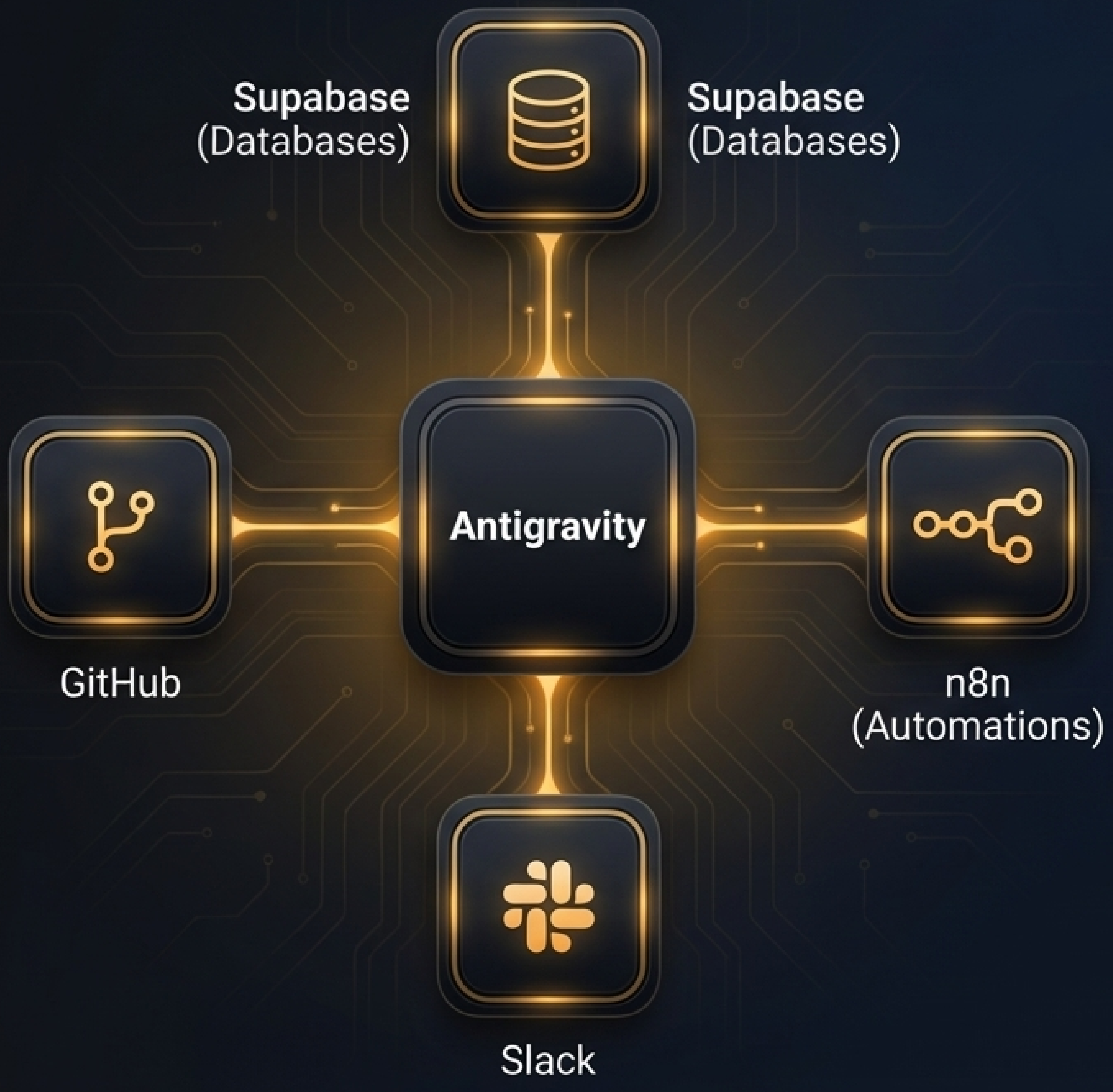
Antigravity seamlessly routes tasks to the optimal frontier model. Gemini 3 Pro handles front-end aesthetics and visual components, while Claude 4.5 Opus powers complex backend architecture and data routing.

Trust Built Through Verification

Agents do not just write code; they prove it works. Antigravity delivers verifiable artifacts—including implementation plans, self-healing execution logs, and actual video screen recordings of the agent navigating the local host browser to test features.

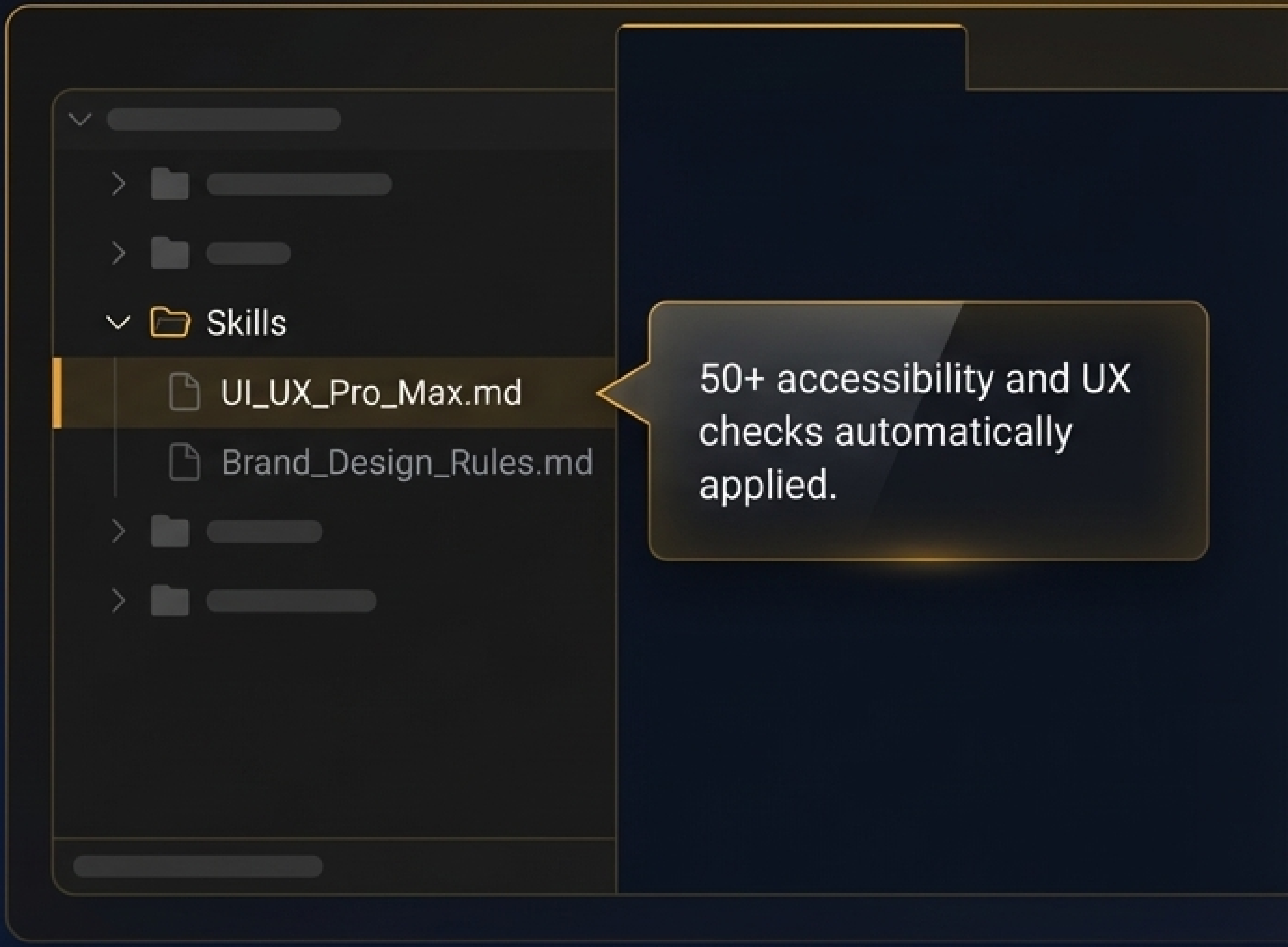
The Universal Remote Control

The Model Context Protocol (MCP) allows Antigravity to physically interact with the outside world. Without manual API wrangling, your agents can spin up Supabase databases, trigger n8n workflows, and deploy to external servers autonomously.



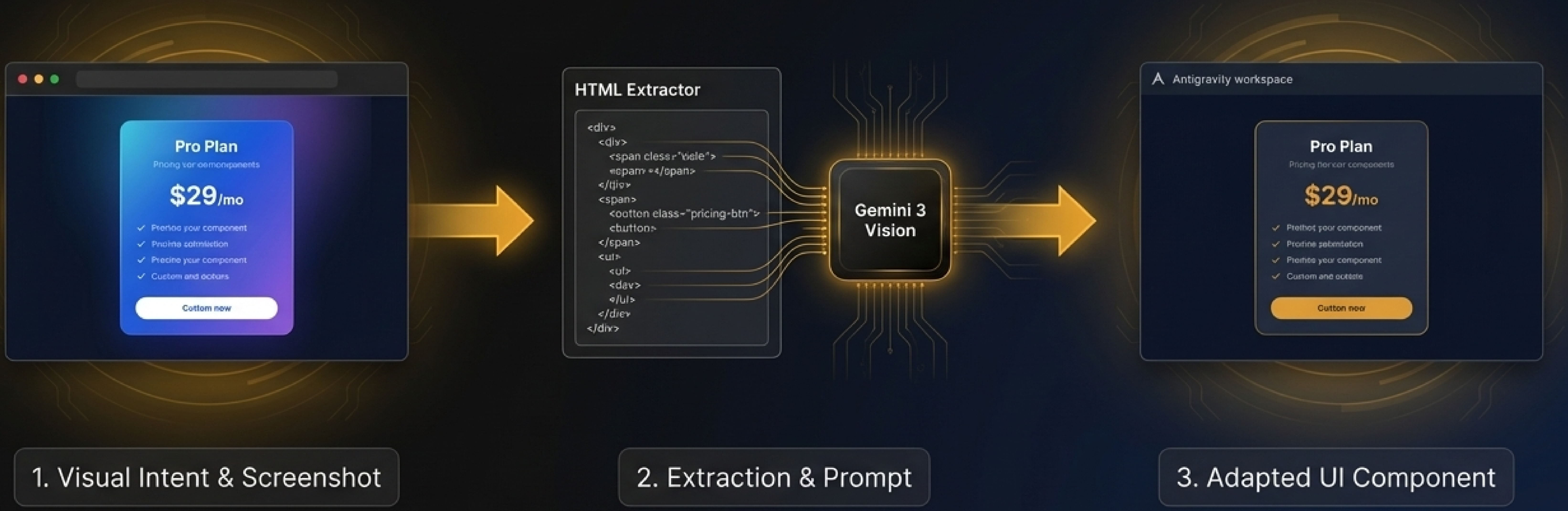
Pre-Packaged Brilliance

Elevate baseline output by injecting Skills into your workspace. By uploading brand guidelines and utilizing community skills like UI/UX Pro Max, the agent inherently understands your typography, accessibility standards, and spacing rules before it writes a single line of code.



Surgical UI Sniping

See a design you love? Extract the visual intent and core HTML, feed it to the agent via screenshot, and let **Gemini 3** perfectly recreate and adapt the component to fit your specific brand guidelines.



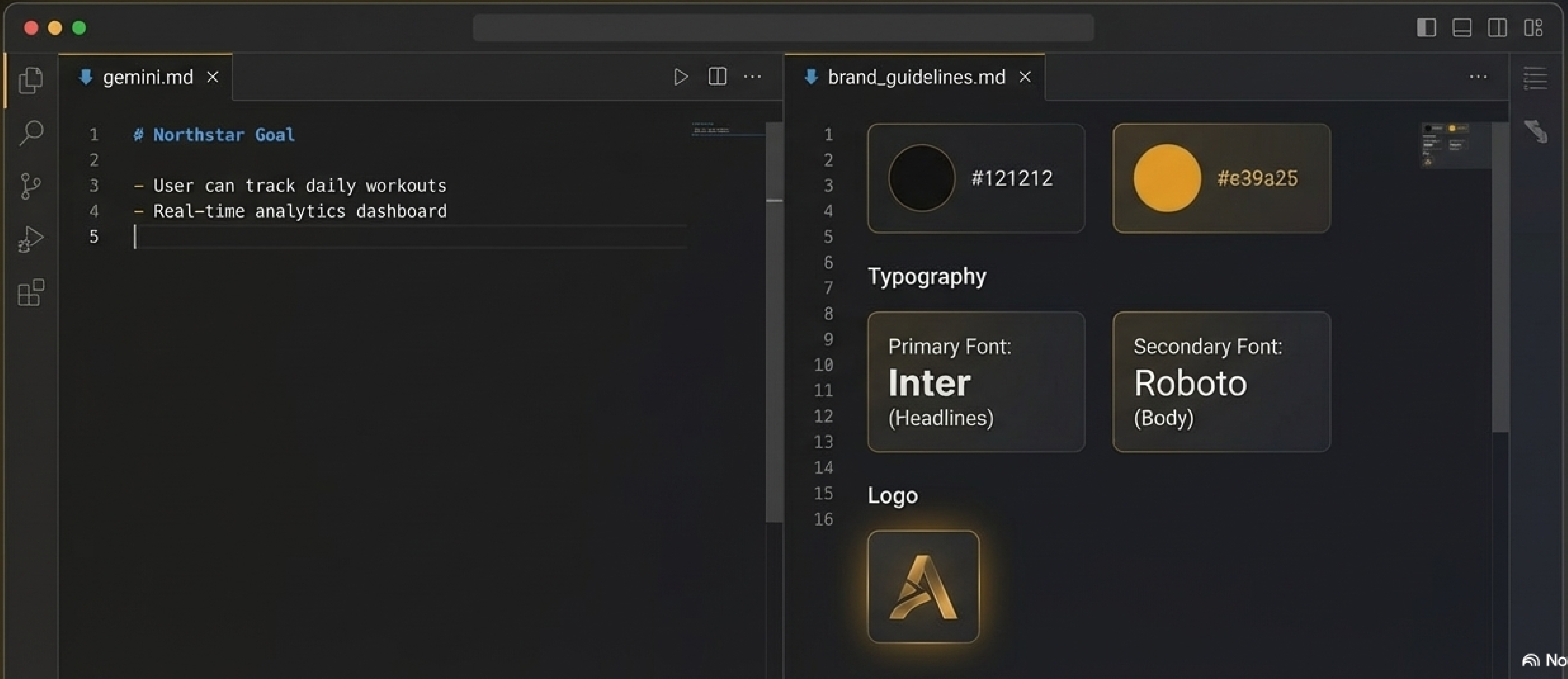
The Execution Playbook

The systematic framework for consistently building production-ready, full-stack applications without writing a single line of manual syntax.

Blueprint
Link
Architect
Style
Trrigger

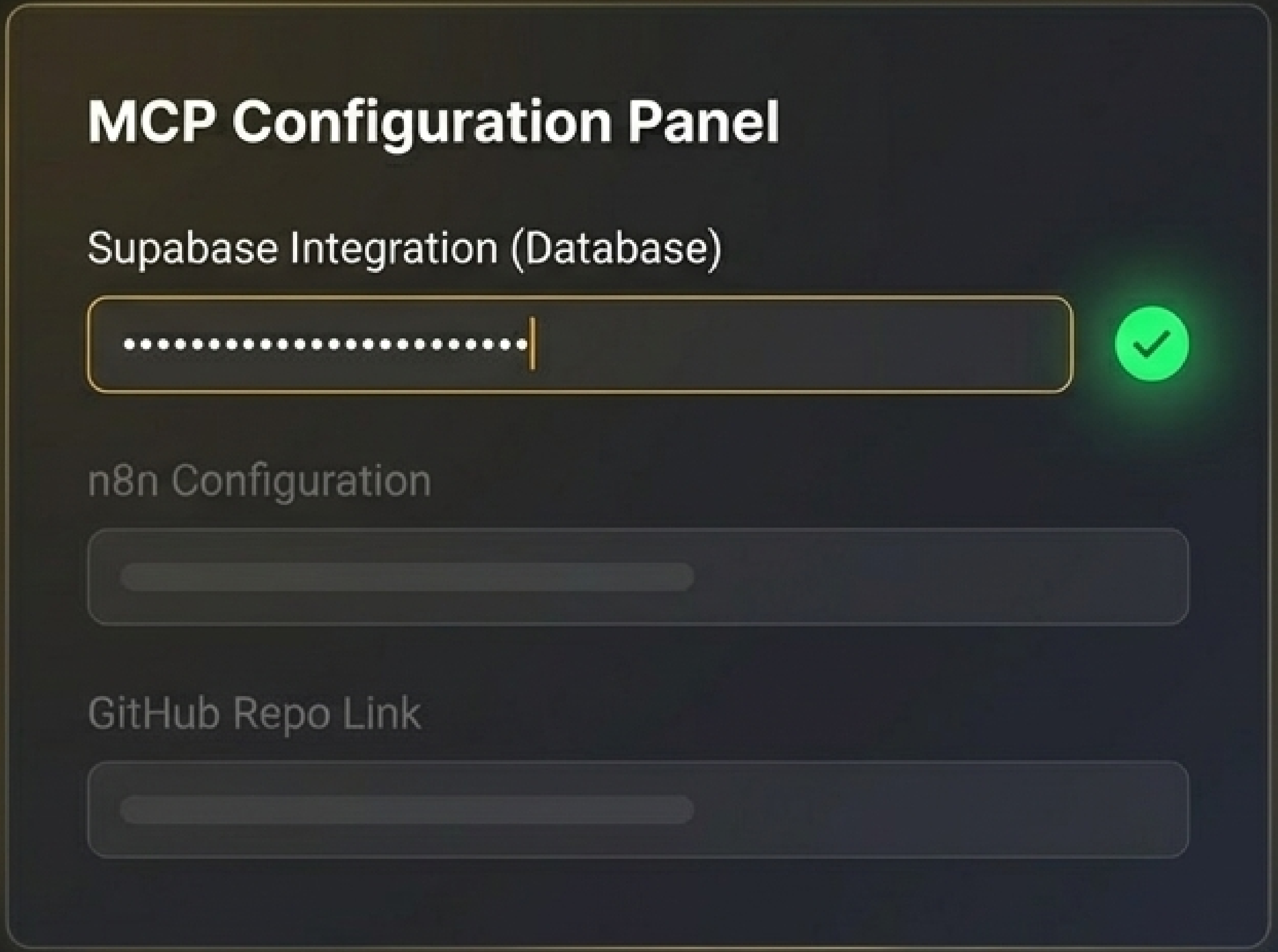
Step 1: Blueprint

Everything starts with context. Define the Northstar of the application in plain English, establish the data schema, and upload your brand guidelines to ground the agent in reality.



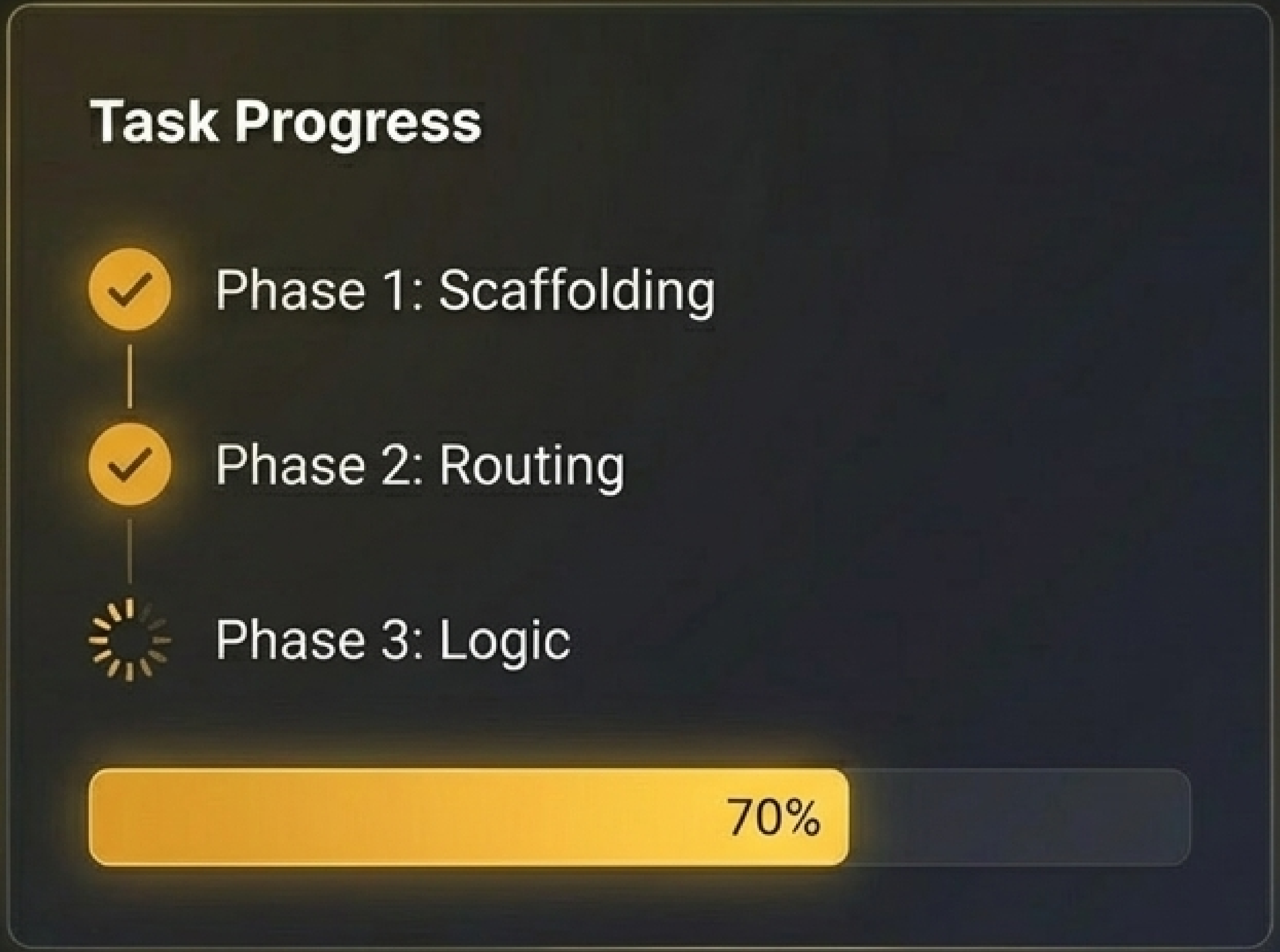
Step 2: Link

Connect your environment to the outside world. By linking essential MCPs like Supabase or GitHub up front, you give the agent the necessary permissions to structure backends and handle authentications autonomously.



Step 3: Architect

The agent goes to work. It generates a step-by-step implementation plan, scaffolds the Minimal Viable Product, and executes the logic layer while you monitor the high-level progress from the Agent Manager.

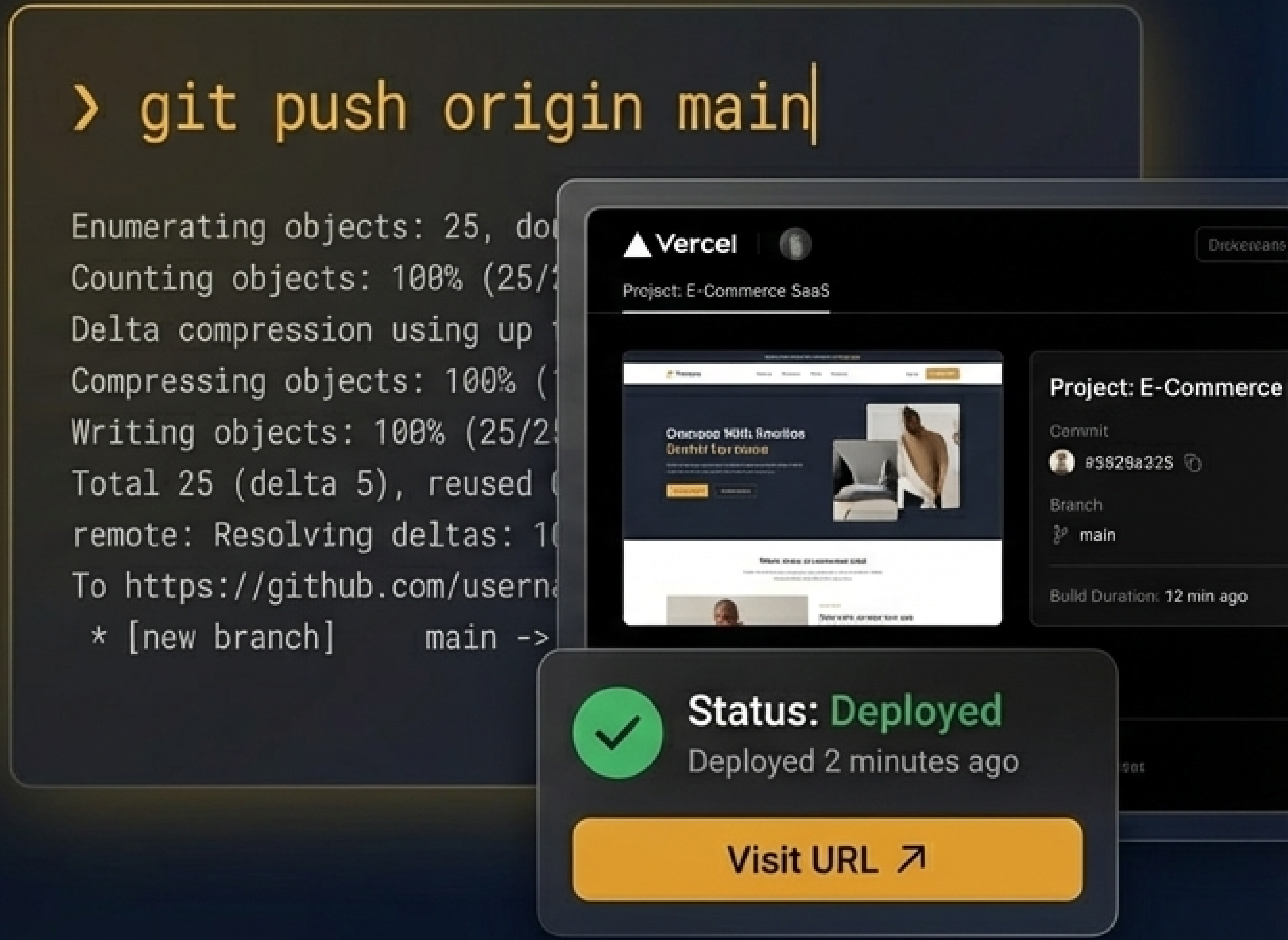


Step 4: Style

Refine the visual output. Inject UI Skills and use visual commenting directly on the interface diffs. Review the browser artifacts to ensure pixel-perfect rendering and responsive behavior.

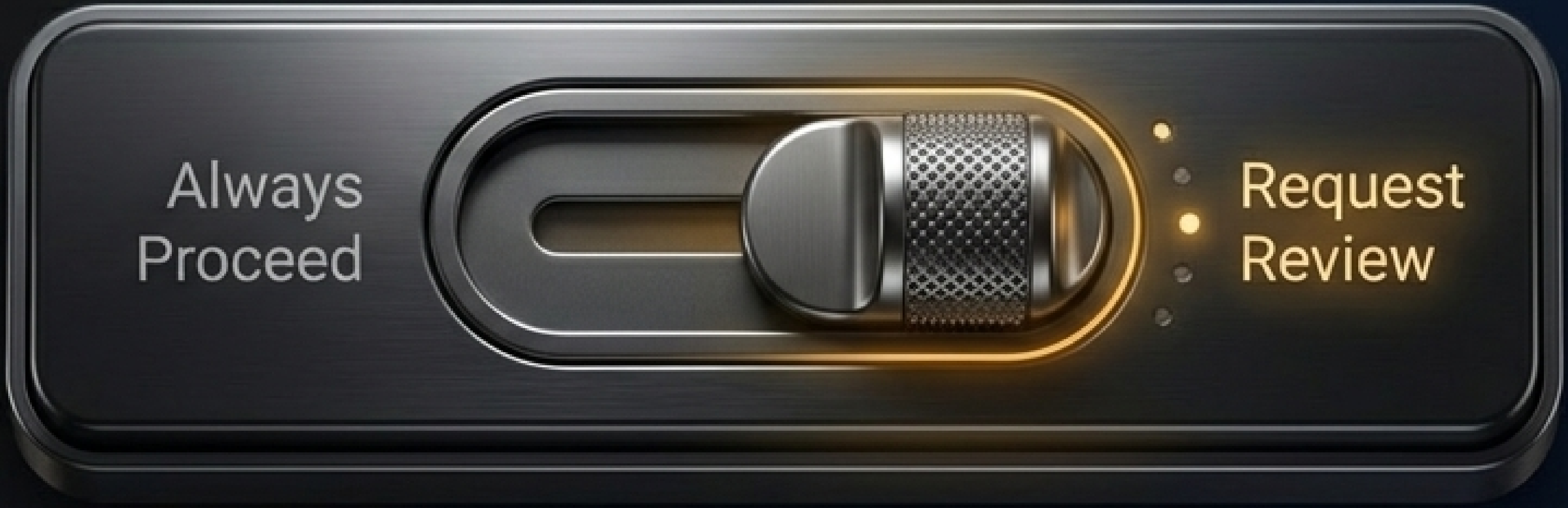
Step 5: Trigger

Turn local code into a live product. Instruct the agent to execute a security audit, push the entire repository to GitHub, and trigger a live production deployment on Vercel.



Dialing in Autonomy

You control the leash. For rapid prototyping, allow the agent to auto-execute terminal commands. For sensitive production environments, require human review for every architectural decision.



The Infinite Ceiling

The barrier to entry for software development has vanished. We are entering an era where output is no longer constrained by the speed of our typing, but purely by the scale of our imagination.