

Zero to Production: The Antigravity Vibe Coding Playbook

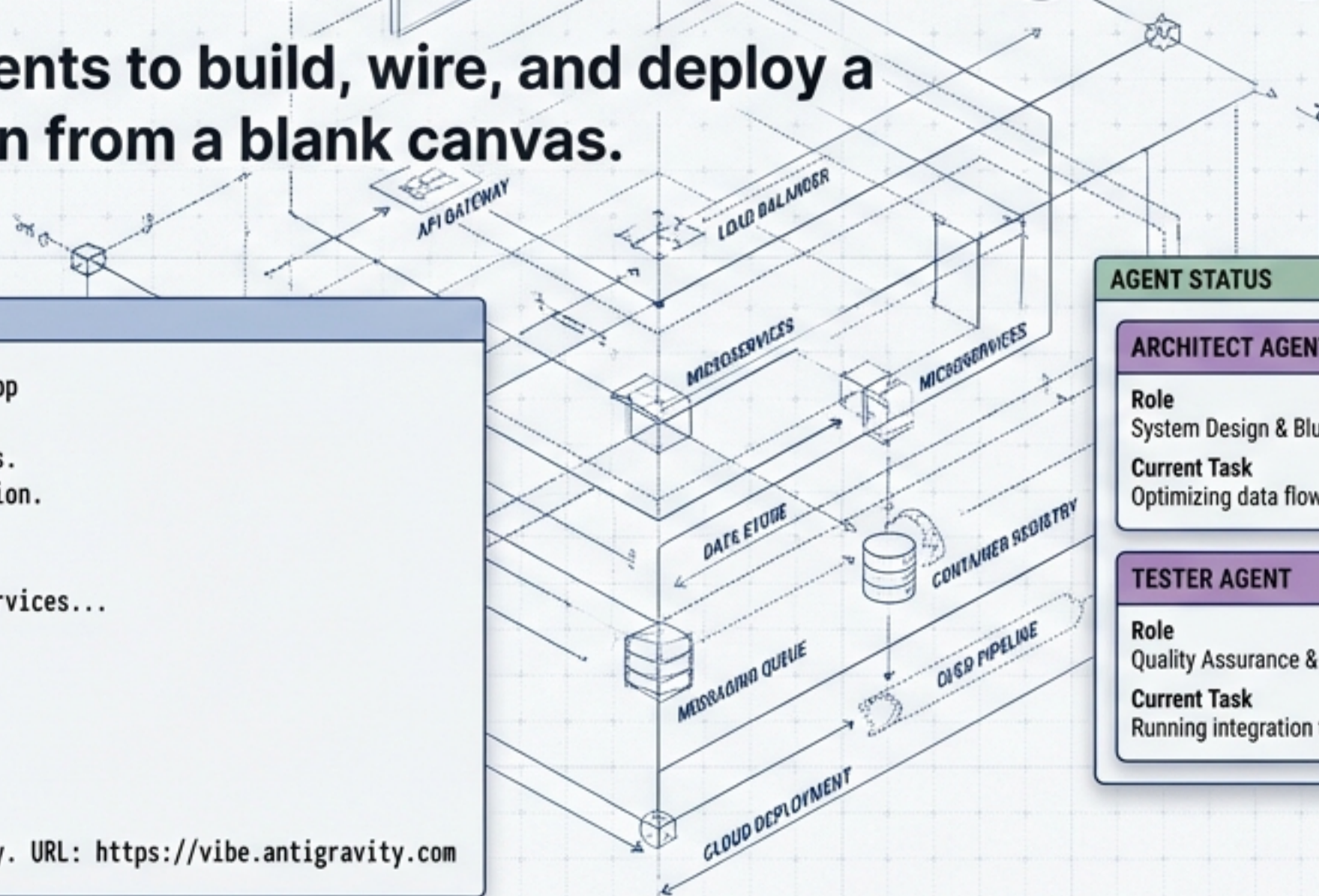
Orchestrating AI Agents to build, wire, and deploy a full-stack application from a blank canvas.

TERMINAL OUTPUT

```
> antigravity init --project vibe_coding_app
[INFO] Initializing project structure...
[INFO] Generating agent configuration files.
[SUCCESS] Environment ready for orchestration.

> antigravity wire --stack full-stack
[INFO] Connecting front-end to back-end services...
[INFO] Wiring database connections.

> antigravity deploy --target production
[INFO] Packaging application containers.
[INFO] Pushing to registry.
[INFO] Deploying to cluster...
[SUCCESS] Application deployed successfully. URL: https://vibe.antigravity.com
```

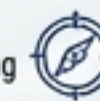


AGENT STATUS

ARCHITECT AGENT

ACTIVE

Role
System Design & Blueprinting

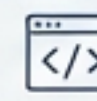


Current Task
Optimizing data flow paths.

DEVELOPER AGENT

ACTIVE

Role
Code Generation & Wiring



Current Task
Implementing API endpoints.

TESTER AGENT

ACTIVE

Role
Quality Assurance & Validation



Current Task
Running integration tests.

OPS AGENT

STANDBY

Role
Deployment & Monitoring



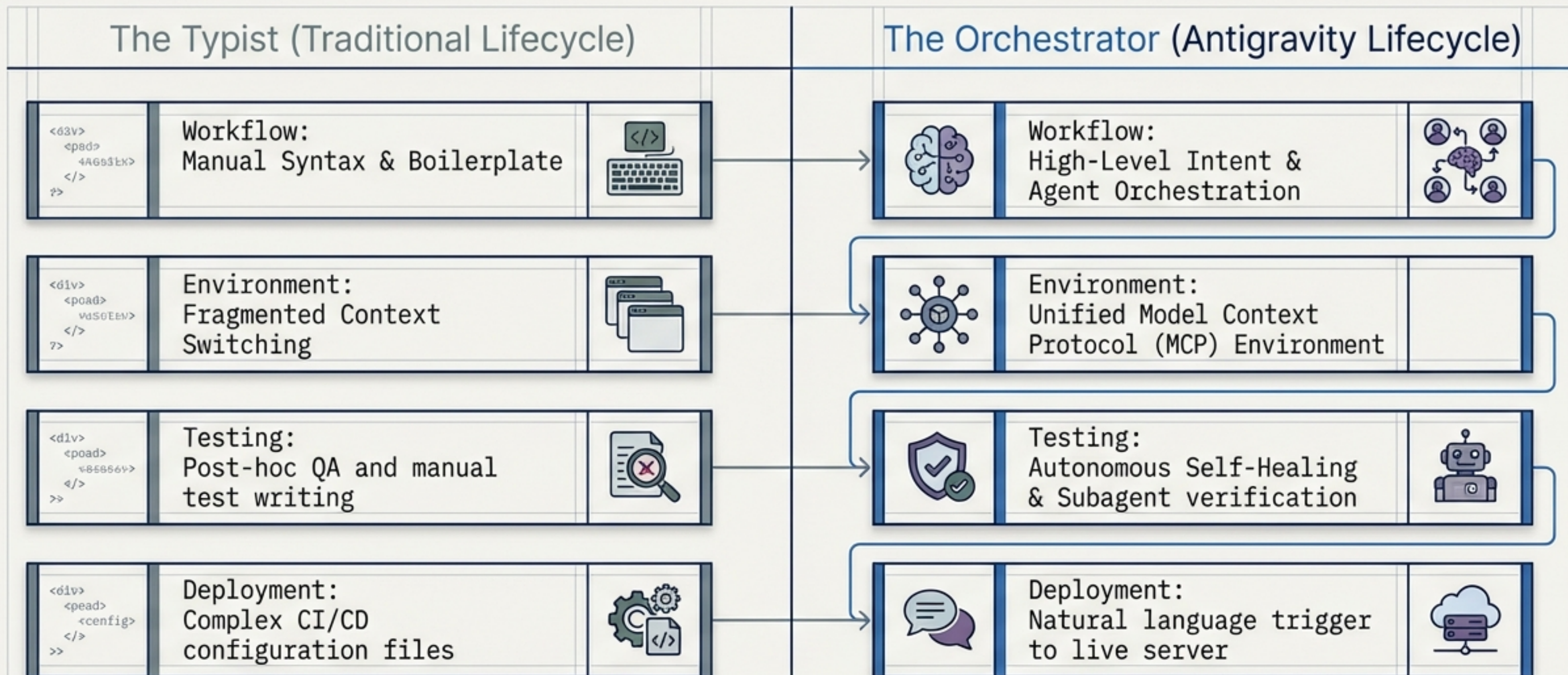
Current Task
Awaiting deployment trigger.



Antigravity

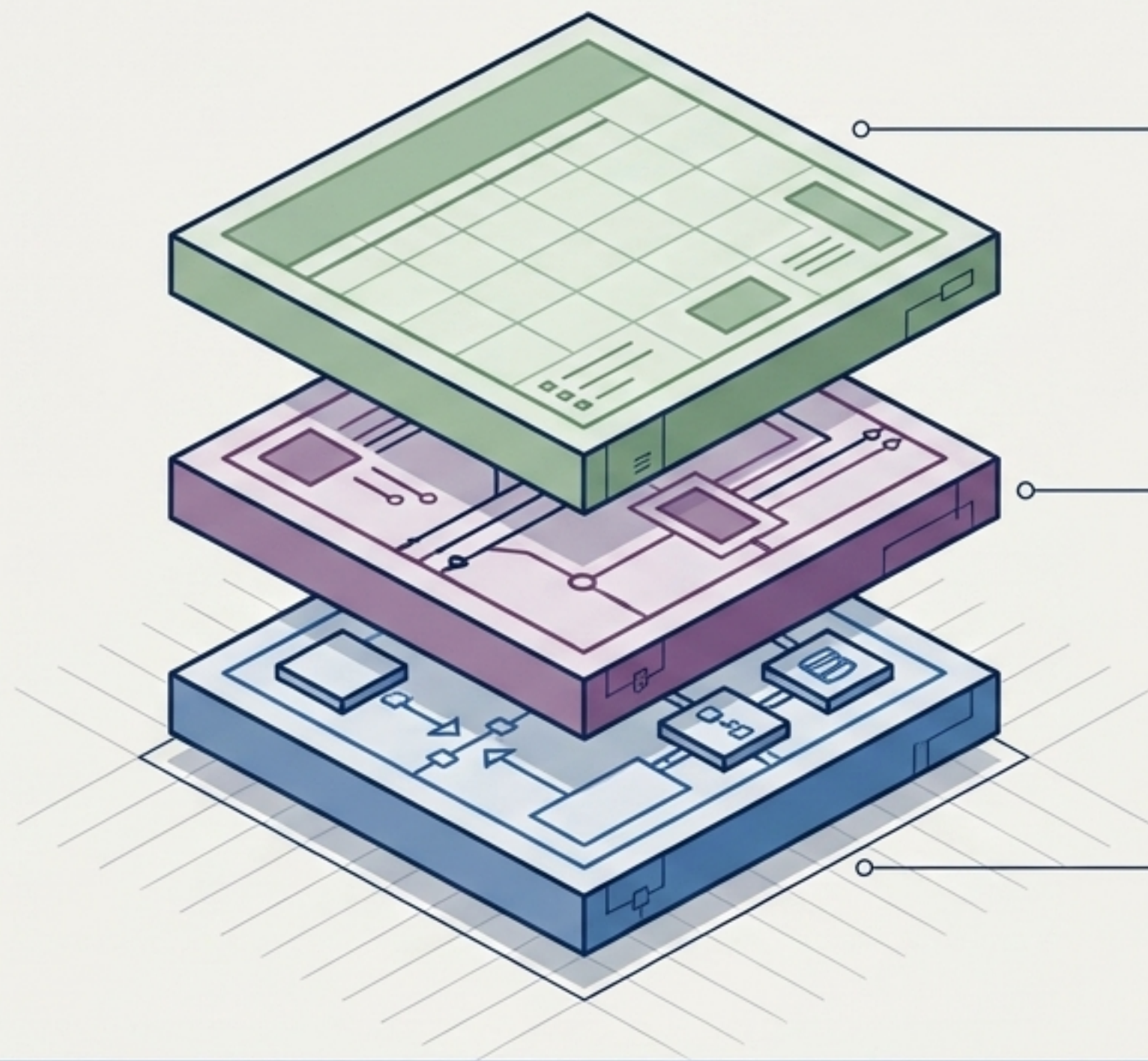
Build. Wire. Deploy.

The Paradigm Shift



Vibe coding isn't about AI writing your code. It's about directing a specialized digital workforce.

The Antigravity Stack



Rules: The Baseline Constraints

Passive, always-on guardrails that dictate coding standards and project style.

Skills (SKILL.md): The Specialized Knowledge

Ephemeral task definitions that act as the agent's brains, loaded on-demand for specific workflows.

MCPs: The External Hands

The universal adapters that directly plug the agent into external stateful infrastructure like databases and cloud providers.

The Vibe Coding Journey

Setup & Scaffolding

Instantiating the workspace using `agents.md` and the 58-Skill Aggregator.

The Data Layer

Plugging in the **Supabase MCP** for autonomous **schema generation** and **CRUD routing**.

AI Intelligence

Injecting the Gemini API and AI SDK to build a smart feature.

Deployment

Using the Cloud Run MCP to push the codebase to a live **production server**.

Step 1: Instantiating the Environment

```
Terminal Output
```

Folder | >_ | ⚡ | 📄 | ⚙️

Prompt: @agents instantiate

System: Setting up environment...
Creating Temp folder...
Generating Environment file...

1. The Directive

Instructing the system on goals, inputs, and tool definitions.

2. Orchestration

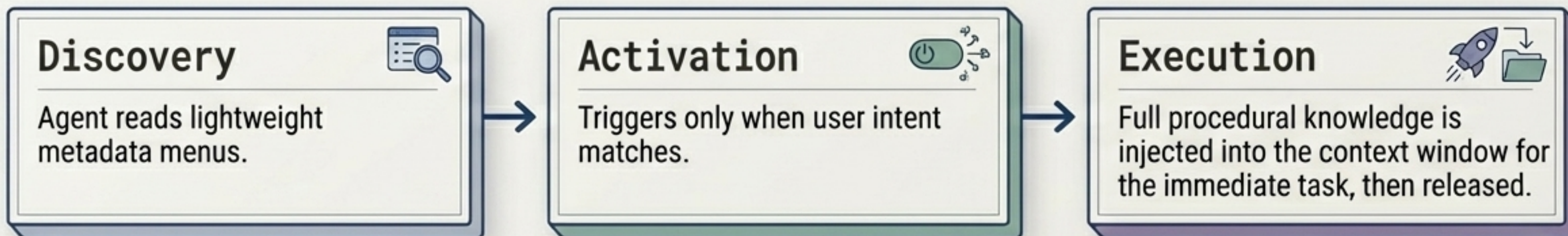
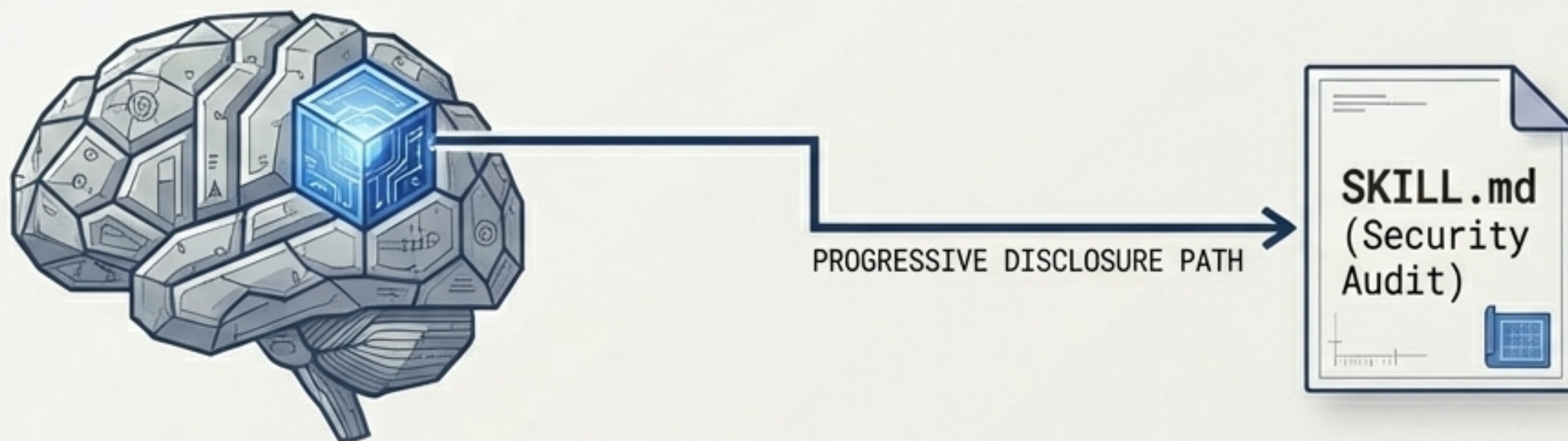
The agent making decisions and interpreting directions into actions.

3. Execution

The generated scripts and files executing the tasks.

Beating Context Bloat via Progressive Disclosure

Instead of dumping 50k tokens of unused tools into active memory, Antigravity uses the 58-Skill Aggregator.



Scaffolding with the Heavyweight Skills



AGENT STATUS: LOKI MODE



Skill:
Loki Mode
Author:
travisvn

Function:
Subagent Driven Development.
Dispatches sub-tasks to itself or other model instances in parallel.

Output:
Transforms the agent from a reactive assistant into an autonomous project manager.



AGENT STATUS: TDD ARCHITECT

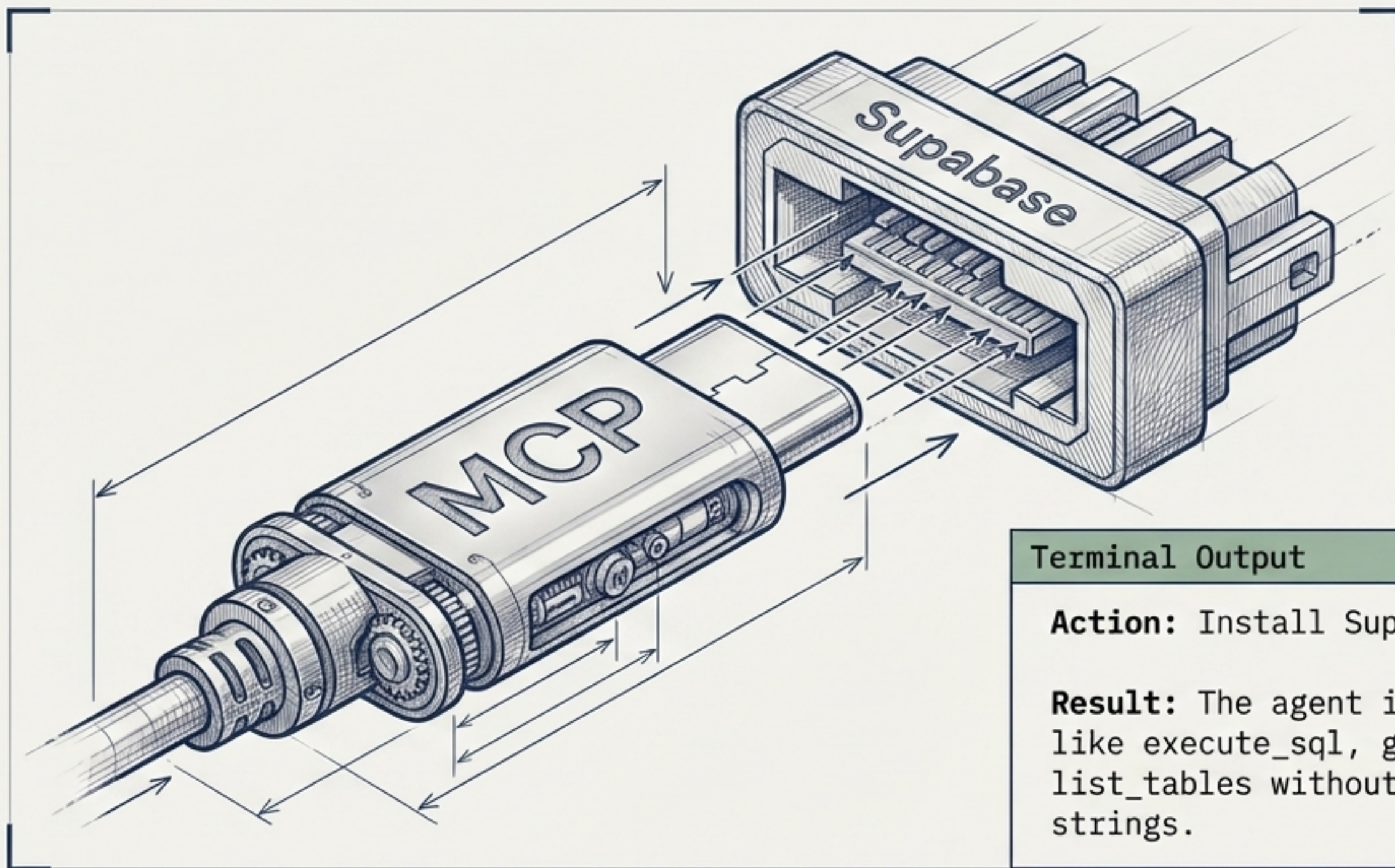


Skill:
TDD Architect
Author:
diet103

Function:
Enforces Verification Gatekeeping.

Output:
Forces a strict loop: Write failing test -> Show user output -> Write implementation code -> Show passing test.

Step 2: The Data Layer & The Universal Adapter



Model Context Protocol (MCP) transforms passive chat agents' agents into active operators with direct access to enterprise infrastructure.


Terminal Output

Action: Install Supabase MCP

Result: The agent instantly gains executable tools like `execute_sql`, `get_table_schema`, and `list_tables` without managing manual DB connection strings.

Autonomous Schema Generation

HUMAN PROMPT

Build a full CRUD backend for a task management app using the Supabase MCP. 

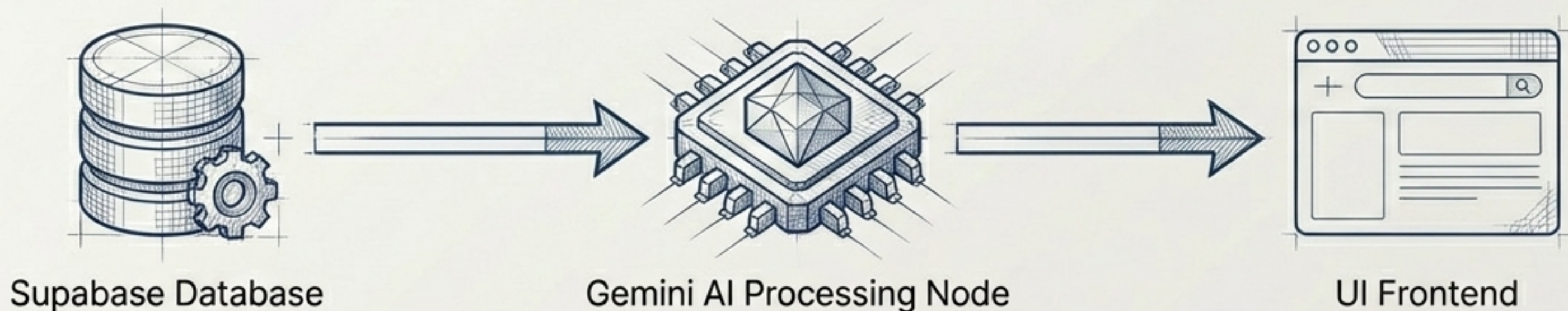
AGENT EXECUTION PATH

1. Uses `execute_sql` to build a robust Tasks table (ID, Title, Status, User_ID).

2. Uses `get_query_plan` to optimize the schema before committing.

3. Writes the frontend React hooks to connect to the new Supabase routes.

Step 3: Injecting AI Intelligence



Action: Wiring the Gemini API and AI SDK into the codebase via Workflows.

Trigger: /new-feature

Prompt: Create a feature that reads a user's task list from Supabase and uses the Gemini API to automatically categorize them and suggest a priority order.

Result: Agent writes server-side logic, injects API keys from .env, and builds the frontend component.

Planning Mode: Directing the Digital Workforce

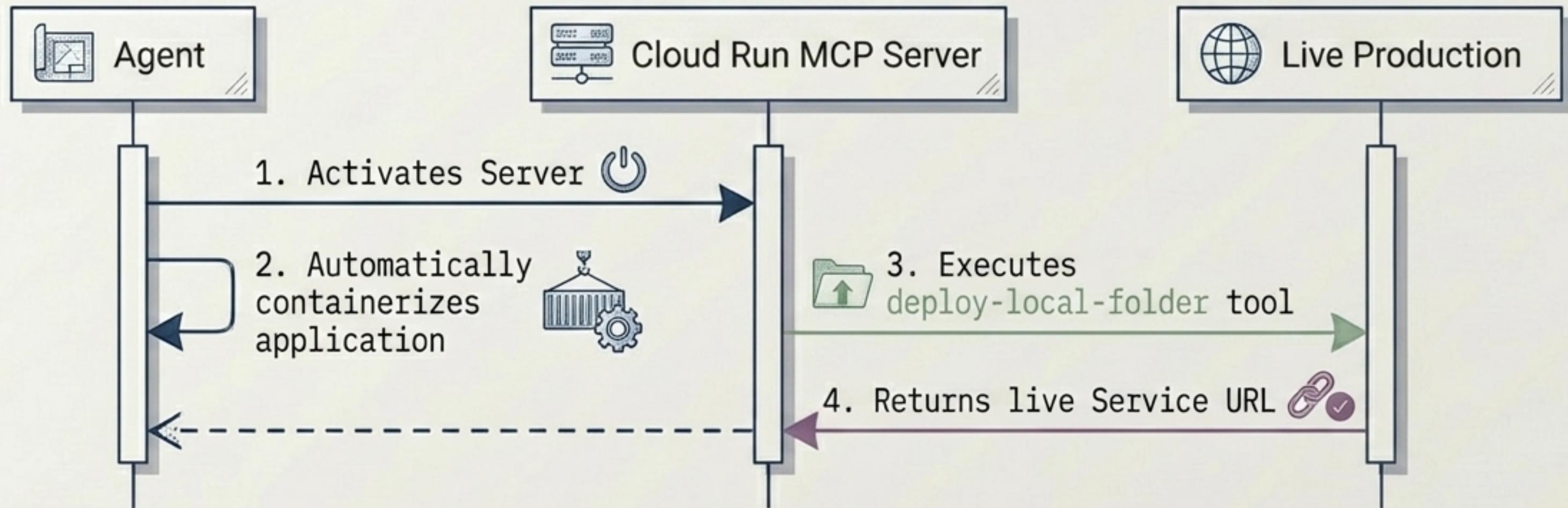
Antigravity generates an Implementation Plan first. Vibe coders steer the ship.

```
12
13 Plan Step 4: Implement primary UI styling.
14
15 Plan Step 5: Use a dark theme with purple gradients.
16
17 // Comments
18 function show
19   const auI =
20     return "purple gradient";
21 });
22 }
```

The **Agent** updates the plan, absorbs the brand rules, and proceeds to build the customized UI flawlessly.

Step 4: One-Click Production Deployment

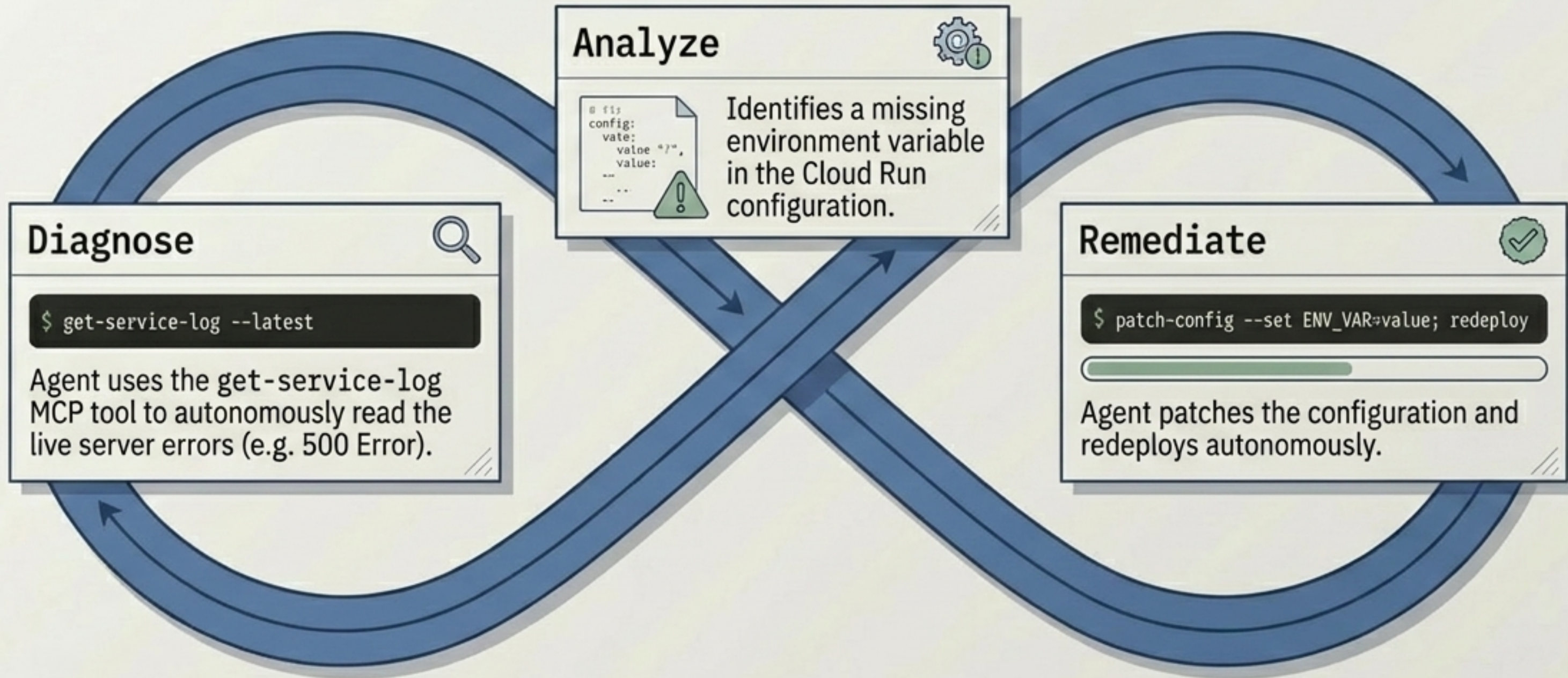
Final Prompt: Deploy this workspace to Cloud Run.



Returns the live production **Service URL** directly in the IDE chat within minutes. No terminal context switching required.

The Agent's Self-Healing Loop

The vibe coder supervises the correction rather than hunting for the bug.

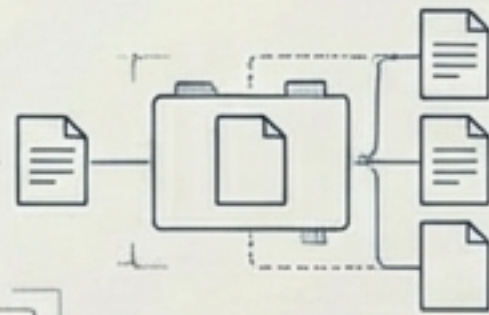


The Orchestrator's Mindset

System over Syntax

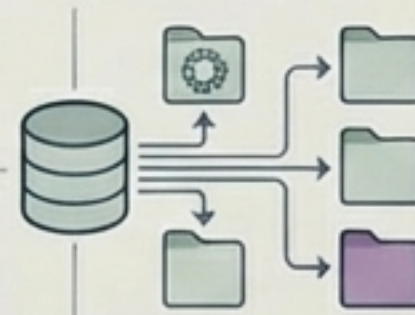
Stop obsessing over manual prompts.

Start building reusable, predictable systems with `agents.md` and `SKILL.md`.



Curation over Creation

The best developers won't write the most code; they will curate the best libraries from the **58-Skill Aggregator** and community MCPs.



Intent to Production

Tools like the Supabase and Cloud Run MCPs have removed the final infrastructure barriers. The distance between a blank canvas and a live application has never been shorter.

