
Google Antigravity & Agents CLI
End-to-End Architectural Blueprint: Surfaces, Sandboxes, and Dev-to-Prod Pipelines
Agent-First Development | Infrastructure as Code | v1.0
CI/CD


Layer 1: Developer Harness
Antigravity (App, CLI agy, IDE, SDK).
Note: The coding agent you work with.
Layer 2: Agent-Building Expertise
Agents CLI skills + agents-cli commands.
Note: Teaching the agent ADK, evals, and deployment.
Layer 3: Agent Framework
ADK (open source).
Note: The actual code of the agent you ship.
Layer 4: Production Runtime
Agent Runtime, Cloud Run, GKE ->
Note: Where the shipped agent executes.
Gemini Enterprise
Layer 5: Secure Dev Environment
Cloud Workstations, Agent Sandbox on GKE, Terminal Sandbox.
Note: Where untrusted code is isolated.
Rule 0: Antigravity is the developer's harness. It is never the production runtime for the ADK agent you build.

Antigravity App (GA)

Desktop installer; interactive GUI session.

🖥️ Best for: Supervising parallel subagents and artifact review panes.

💀 Trap: Requires interactive GUI; not suited for unattended CI runners.

Antigravity CLI (agy)

Keyboard-first TUI (agy --help), OS-sandboxed loops.

💀 Trap: Gemini CLI (geminicli.com) was superseded by agy on 2026-06-18. Treat it as legacy. Use agy plugin import gemini to migrate.

Antigravity IDE

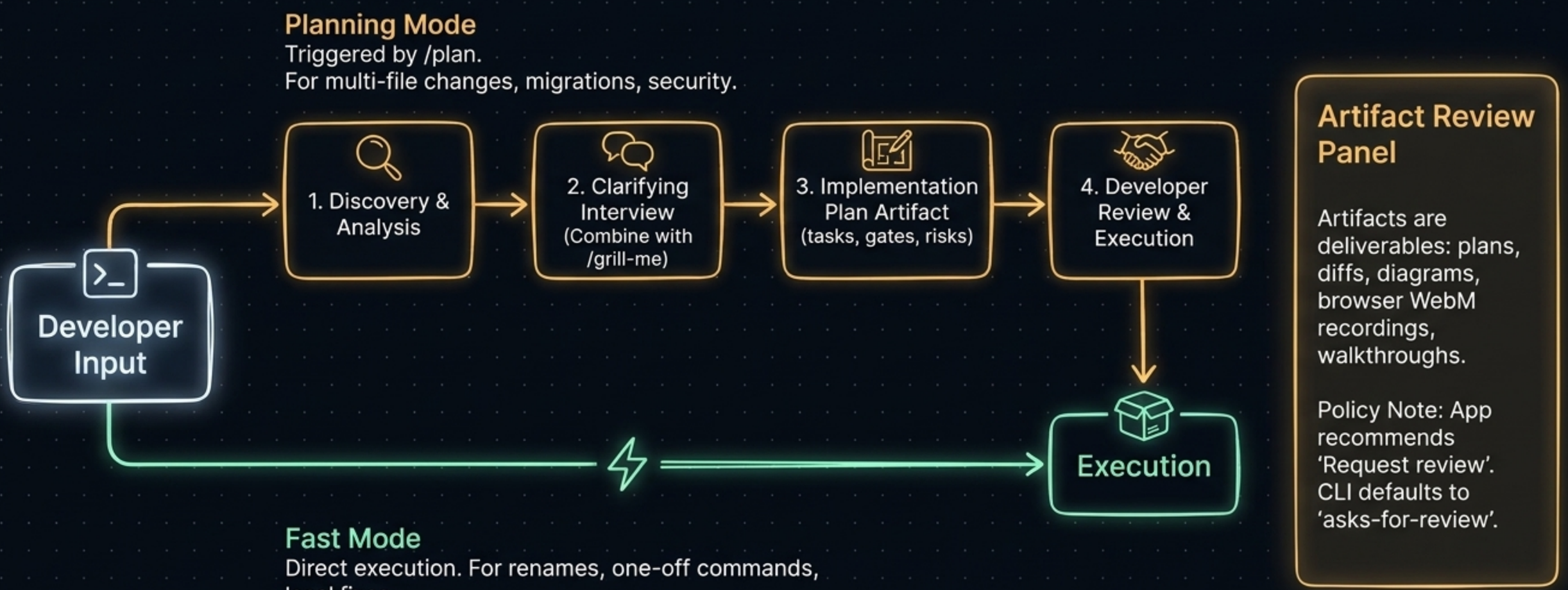
VS Code-based standalone IDE/extensions.

📝 Features: Tab supercomplete, inline edits, side-panel turns.

Antigravity SDK

Headless Python library for CI/CD, PR triage, and programmatic @hooks.* gates.

💀 Trap: Omitting conversation_id + save_dir makes runs ephemeral. Omitting explicit policies=[deny(…)] leaves write tools unconstrained.

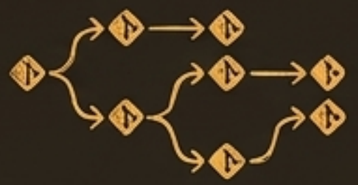

Planning Mode
Triggered by /plan.
For multi-file changes, migrations, security.
1. Discovery & Analysis
2. Clarifying Interview (Combine with /grill-me)
3. Implementation Plan Artifact (tasks, gates, risks)
4. Developer Review & Execution
Developer Input
Execution
Fast Mode
Direct execution. For renames, one-off commands, local fixes.
Artifact Review Panel
Artifacts are deliverables: plans, diffs, diagrams, browser WebM recordings, walkthroughs.
Policy Note: App recommends 'Request review'. CLI defaults to 'asks-for-review'.

Multi-File Refactoring Playbook

Goal: Change structure, identical external behavior.



Baseline via test suite. Map targets with .agents/rules/ and research subagents.



Fan-out execution into isolated Git branch worktrees (one writer per slice).



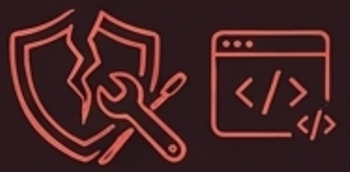
PostToolUse format hooks + separate fresh-context reviewer subagent audits diff.

Automated Vulnerability Patching

Goal: Remove a verified flaw at the root cause.



Reproduce SAST finding via Proof-of-Concept (PoC) exploit in Terminal Sandbox (Seatbelt/seccomp).



Patch root cause (e.g., parameterized queries, path canonicalization).



Regression test passes + Security audit artifact recorded (or leverage CodeMender cm fix Preview).



Rules (Constraints & Invariants)

`AGENTS.md` / `GEMINI.md`: Always-on for the directory.

`.agents/rules/*.md`: Modular rules requiring triggers (`always_on`, `model_decision`, `glob`).

Trap: Missing or camelCase triggers silently discard. Rules over 24KB / 20k tokens reduced to file-path pointer.

Skills (Multi-Step Workflows)

`skills/<name>/SKILL.md`

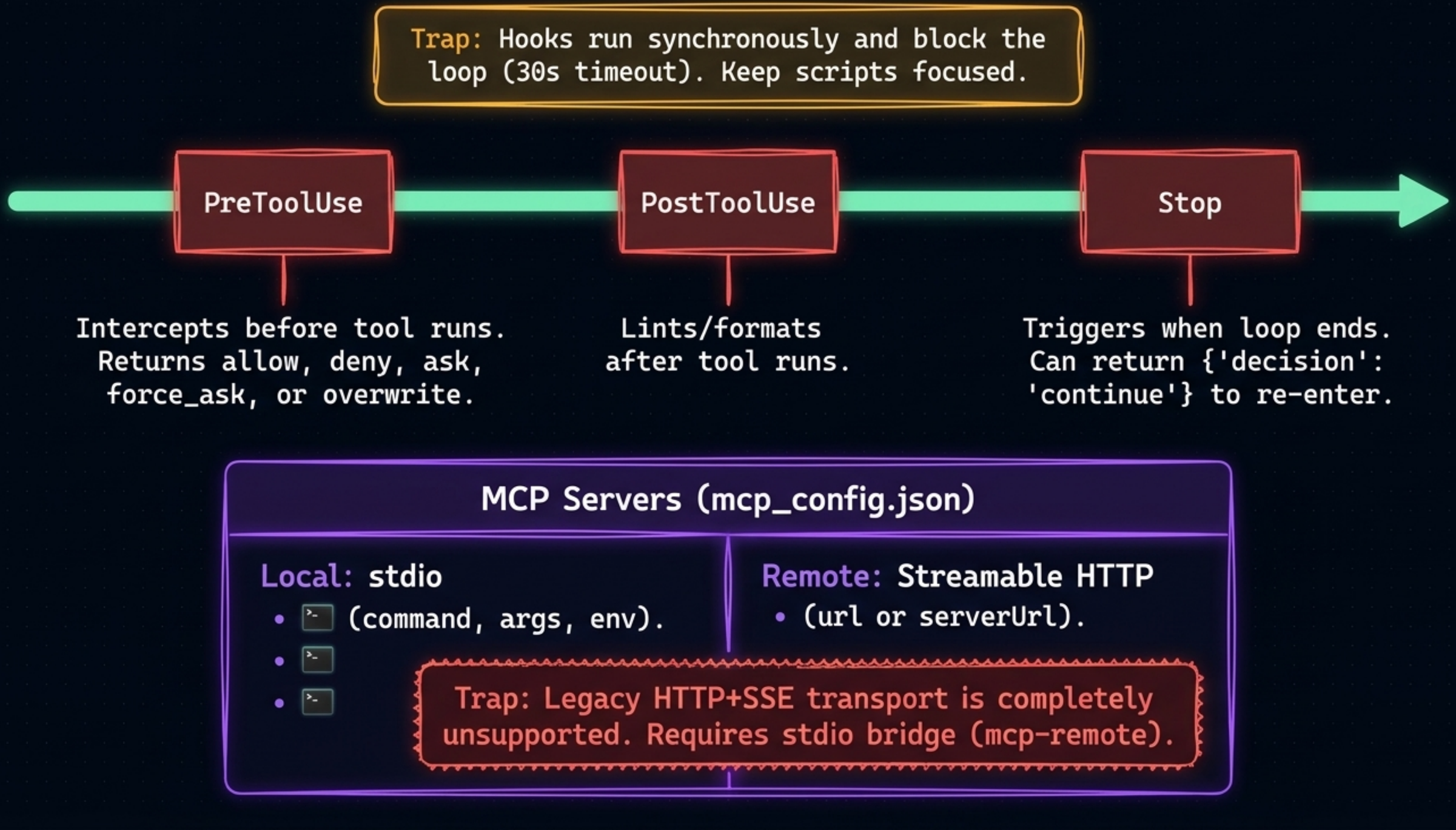
Progressive Disclosure: Only name and description load into initial context. Description decides activation.

Plugins

`plugins/<name>/plugin.json`

Namespaced bundles packaging rules, skills, agents, and hooks.

Legacy IDE workflows stop 2026-10-19 -> migrate via `/migrate-workflows`.


Trap: Hooks run synchronously and block the loop (30s timeout). Keep scripts focused.
PreToolUse
PostToolUse
Stop
Intercepts before tool runs. Returns allow, deny, ask, force_ask, or overwrite.
Lints/formats after tool runs.
Triggers when loop ends. Can return {'decision': 'continue'} to re-enter.
MCP Servers (mcp_config.json)
Local: stdio
(command, args, env).
Remote: Streamable HTTP
(url or serverUrl).
Trap: Legacy HTTP+SSE transport is completely unsupported. Requires stdio bridge (mcp-remote).



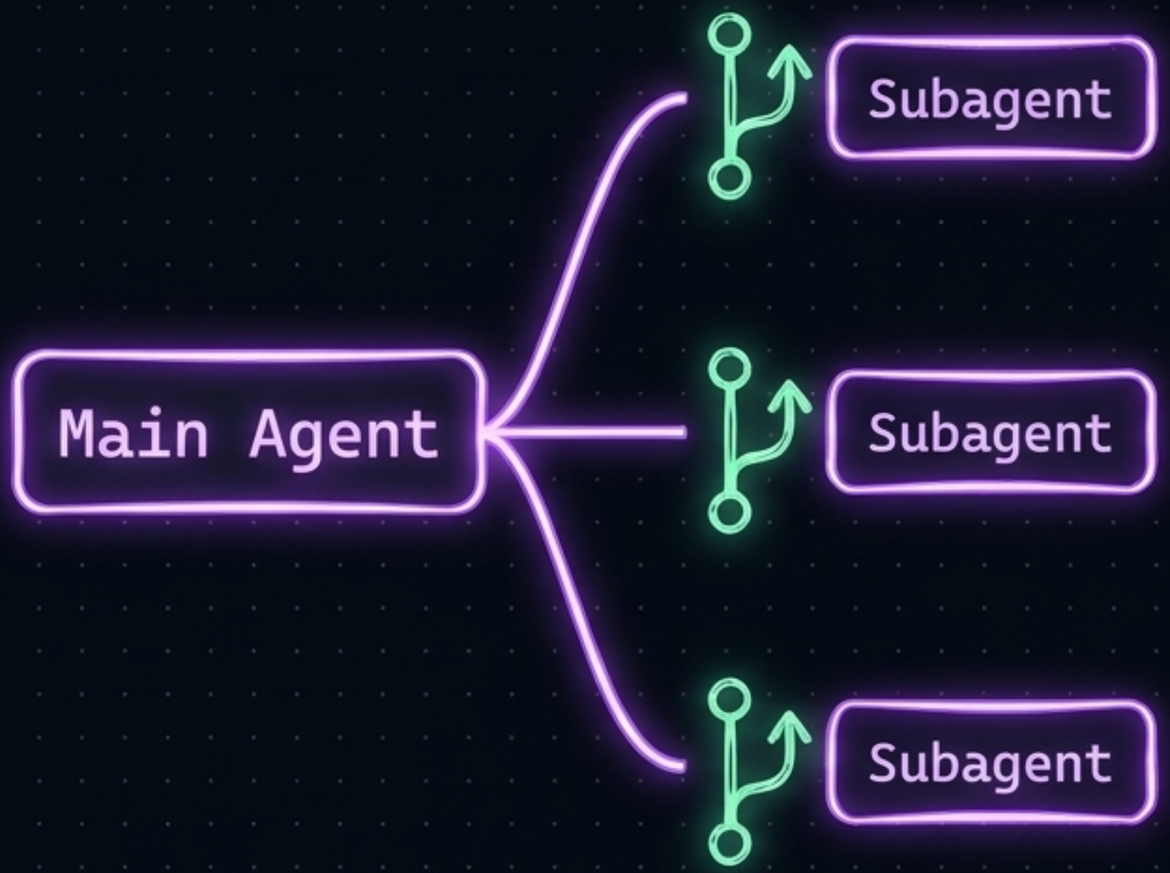
Collision vs. Permission Rules

Name Collisions:
Artifacts resolve strictly
Workspace > Global



Permissions:
Tool/MCP execution evaluates strictly
Deny > Ask > Allow

An explicit Deny always wins, regardless of scope or origin.



Subagent Orchestration

Triggered via invoke_subagent. Every subagent starts with a clean context window (ignoring parent history).

Defined in .agents/agents/<name>.md with YAML frontmatter.

Workspace Isolation Modes

- **branch**: Isolated Git worktree (Recommended).
- **inherit**: Same folder.
- **share**: Shared repo storage.

Browser Subagent (/browser)

Drives Chrome via DevTools MCP. Isolated context, hidden from lists, records WebM.

Anti-Pattern: Running parallel writer subagents in 'inherit' mode causes them to overwrite each other. Always use 'branch' for parallel slice fan-outs.

The Terminal Sandbox

macOS Seatbelt (sbpl) / Linux seccomp + kernel namespaces.

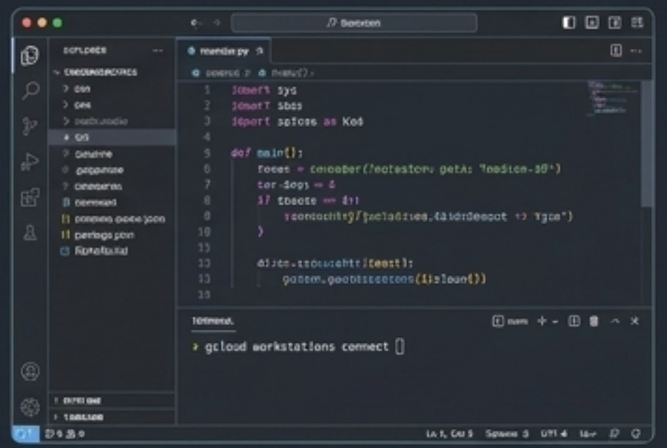
Network: Default zero outbound network.
File System: Host paths like ~/.ssh blocked.
Project folders mounted RW.

CLI Agent Modes (Shift+Tab)	
`default`	Asks review before every write.
`accept-edits`	Auto-approves writes; commands follow toolPermission.
`auto (Preview)`	Auto-approves edits and runs commands in the sandbox. Risky actions outside sandbox pause.

⚠ Trap: --dangerously-skip-permissions (or always-proceed) outside the sandbox gives the agent full host privileges to wipe files.

Cloud Infrastructure (VPC-SC Perimeter)

Cloud Workstations (GA)



Environment Features

Managed Compute Engine dev environments.

CMEK-encrypted persistent /home disks.

Agent-Optimized Setup

Use `idleAction=SUSPEND` with an `idleTimeout` (e.g., 1800s). Deploy keep-alive lifecycle hooks (`/google/scripts/keep_alive.sh`) to prevent dropping live agent context/RAM state while waiting for input.

Security Posture

Private clusters via PSC, zero public IPs (compute.vmExternalIpAccess).

Trap: Enforcing restrictPrivateServiceConnectProducer without allowlisting Google APIs breaks PSC setup.



Agent Sandbox on GKE (GA)

Pod-level **user-space kernel (gVisor)** or **micro-VM (Kata)** isolation. Used for untrusted generated code & sidecars.

SandboxTemplate and SandboxWarmPool. Policy blocks hostNetwork and hostPath.

Agent Runtime Managed Sandboxes

Zero-ops serverless execution invoked as a tool by the runtime agent.

---

1. **Code Execution (GA):** Stateful Python/pandas sandbox retaining state up to 14 days. Supports VPC-SC and CMEK.
2. **Computer Use (Preview):** Headless browser/UI automation.

4-Way Sandbox Decision Matrix

Sandbox Option	Best Fit (Mint Green)	Secrets Identity (Bright Purple)	Operational Trap (Coral Red)
GKE Agent Sandbox (gVisor/Kata)	High-scale tool execution	GKE Workload Identity	Blocking metadata server breaks ADC without workload federation.
Cloud Workstations	Human-supervised agents, zero laptop code	Metadata server + Secret Manager	Stopping instead of SUSPEND drops live RAM state.
Local Terminal Sandbox	Fast, $0 local OS isolation	Blocks ~/.ssh	Shares host OS kernel; host-daemon commands fail.
Runtime Code Execution	Zero-ops 14-day stateful Python execution	Agent Identity	Running ADK orchestrator inside the sandbox instead of invoking it as a tool.

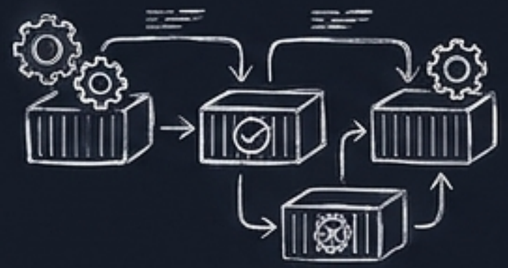
Inner-Loop (Interactivity)

Antigravity App/CLI (agy) & IDE.
Build logic, /plan, parallel subagents, UI.



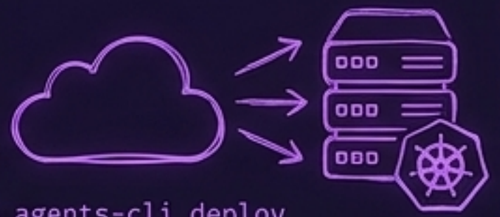
Headless CI/CD Automation

Antigravity SDK.
Persistent pipeline hooks.



Outer-Loop (Scaffold to Prod)

Agents CLI (agents-cli) + ADK CLI (adk).



```
$ agents-cli deploy
$ agents-cli deploy ...
$ adK -n ...
```

Agents CLI Value Proposition

A machine-readable interface packaged as skills (e.g., `google-agents-cli-scaffold`, `-eval`, `-deploy`).
Translates ADK framework concepts into Terraform infra, rubrics, and Agent Runtime/Cloud Run deployments.



agents-cli setup

Injects skills into Antigravity. `--workspace` for Git sharing, `--agent` to target.

agents-cli create

Defaults to adk_python. `-p` flag skips Terraform setup.

agents-cli lint & playground

Ruff checks and ADK web UI on port 8080.

Agent vs. Human Mode Trap

There is NO `--mode agent` flag.

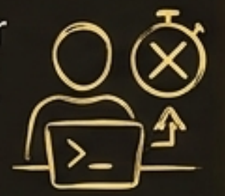
Agent/CI Mode

Uses explicit flags and `-y` / `--yes` to suppress prompts.



Human Mode

Uses `-i` / `--interactive` for menus. Passing `-i` from a CI job leaves the command permanently hung.



The Verification Loop (Agents CLI Eval)

1. Synthesize



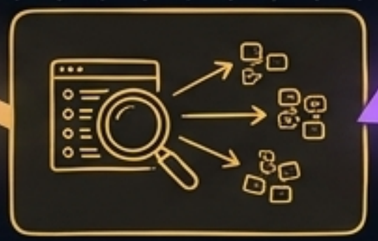
`eval dataset synthesize` generates multi-turn scenarios.

2. Infer & Grade



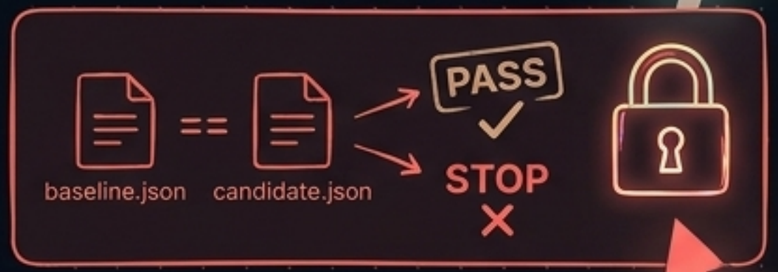
`eval run` executes inference and grades traces via LLM-as-judge.

3. Analyze



`eval analyze` clusters failure modes to identify prompt flaws.

4. Prove (The Gate)



`eval compare baseline.json candidate.json`
Proves no regressions occurred.
Must pass before any cloud step.



Deploying the ADK Agent

`agents-cli deploy` pushes the framework.

Canary Trap: Default is always-latest routing. Use `--update-only` or pin traffic manually.

Publishing

`agents-cli publish gemini` registers the deployed agent to Gemini Enterprise.

(Note: publish gemini-enterprise a hidden legacy alias).

Observation

`Observability Skill` sets up Cloud Trace, prompt-response logging, and BigQuery Agent Analytics.

Vendor-Neutral Harness Comparison

Dimension	Antigravity	Claude Code
Project Memory	AGENTS.md	CLAUDE.md
Extensibility	MCP servers & SKILL.md (Agents CLI works with both)	MCP servers & SKILL.md (Agents CLI works with both)
Isolation	Terminal Sandbox / Git worktrees	Permission rules / Own sandbox
Model Engine	Gemini-first, some OSS	Anthropic models (via Agent Platform)

Best Fit (Antigravity):
Teams needing parallel subagents and artifact reviews.

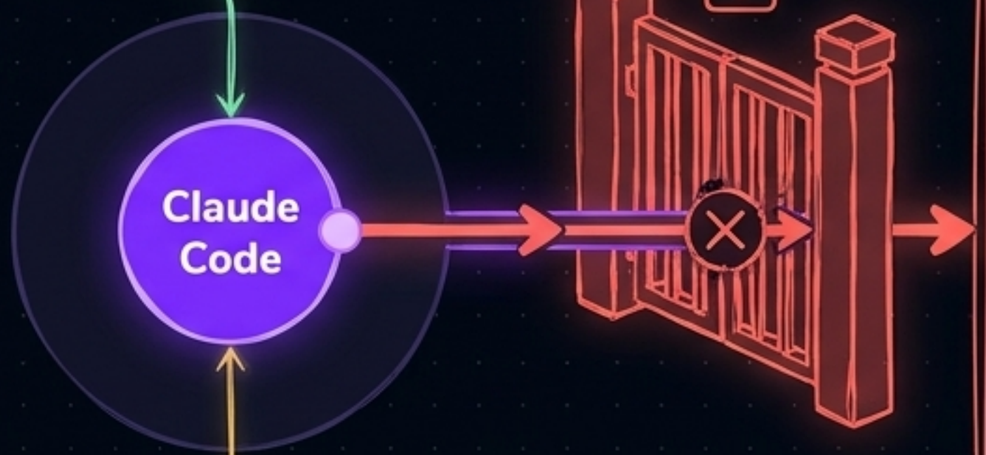
Best Fit (Claude Code):
Teams standardized on Anthropic who require Google Cloud IAM, billing, and data residency.

Claude Code on Agent Platform

- Uses Application Default Credentials (ADC), avoiding long-lived API keys.
- Billed to Google Cloud project.

Cost & Capacity

- **Prompt Caching**: Default 5m, extended 1h. Cache reads cost far less than inputs.
- **Provisioned Throughput**: Reserved capacity for fixed fee (requires regional endpoint).



Google Cloud Org Policies & IAM

Org Policy Guardrails

- `vertexai.allowedModels`: List models by action (e.g., `publishers/anthropic/models/claude-...:predict`).
  - ⚠ Trap: Must also allow `count-tokens:predict`.
- `vertexai.allowedPartnerModelFeatures`: Explicitly allow `:web_search` and `:structured_outputs` (blocked by default with `FAILED_PRECONDITION`).

Critical Exam Traps & Anti-Patterns

⚠ The Shared Working Tree



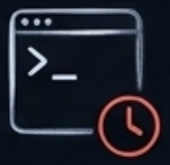
Running parallel writer subagents in inherit mode overwrites files.

Fix: Use branch mode.



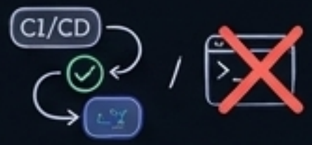
→ `git worktree add -b`

⚠ The Headless Hang



Passing `-i` in CI jobs leaves the agent permanently waiting for prompts.

Fix: Use `-y` and env vars.

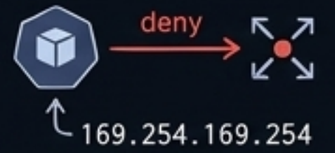


→ `exec = -y`

⚠ GKE ADC Token Failure



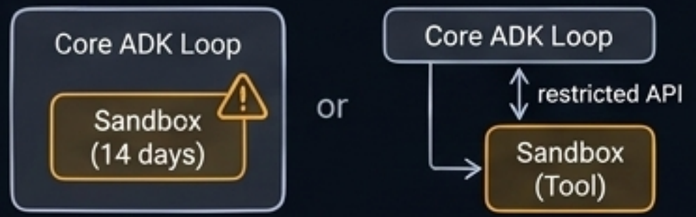
Blocking the Compute Engine metadata server in pod NetworkPolicy without configuring GKE Workload Identity Federation.



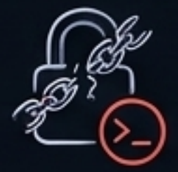
→ Workload Identity Federation

⚠ Sandbox Misplacement

Putting the core ADK orchestration loop inside the 14-day Code Execution sandbox instead of invoking the sandbox strictly as a tool.



⚠ Unconstrained SDK Writes



Omitting `policies=[deny(..)]` in headless runs gives unconstrained shell access once capabilities are unlocked.

`sdk.run(headless=True)` →





1. Plan & Scaffold (Local Workspace)

Antigravity /plan drafts artifact -> Agents CLI scaffolds repo inside Cloud Workstation.

2. Build & Isolate (Inner Loop)

Parallel subagents execute in branch Git worktrees, constrained by .agents/rules/ and Terminal Sandbox.

3. Verify (The Gate)

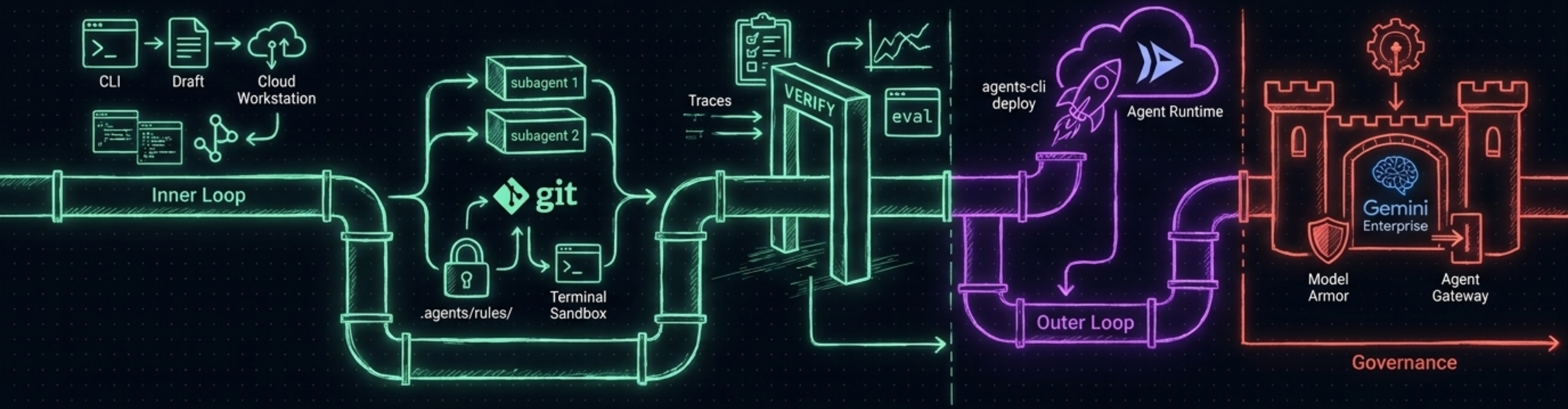
agents-cli eval run grades traces against baseline.

4. Deploy (Outer Loop)

agents-cli deploy ships framework code to Agent Runtime/Cloud Run.

5. Publish & Govern (Production)

Registered to Gemini Enterprise; protected by Model Armor and Agent Gateway.



From Developer Harness to Production Runtime. One Unified Lifecycle.