

# The model is the engine. The harness is the car.

Every AI coding agent on the market is the same handful of frontier models wrapped in different scaffolding. This deck covers what that scaffolding does, the three interfaces the market has built it into, and why the wrapper now matters more than the engine inside it.

## 3

interface tiers on the market: CLI, IDE, and cloud. One loop underneath all of them.

# Six terms, two scales

| TERM             | WHAT IT IS                                                                                        | LOOPS ON ITS OWN?                      | STOPS WHEN                                    |
|------------------|---------------------------------------------------------------------------------------------------|----------------------------------------|-----------------------------------------------|
| Direct API call  | One request, one response                                                                         | No                                     | The response is generated                     |
| Chatbot          | A back-and-forth conversation you drive turn by turn                                              | No, you send each turn yourself        | You stop sending messages                     |
| Deep research    | The model runs its own search-and-read loop for one request                                       | Yes, but only within that one task     | The report is compiled                        |
| Agentic harness  | The model runs an open-ended loop against a real environment                                      | Yes, until the task is done or stopped | The task passes its checks, or a limit is hit |
| Autonomous agent | A harness whose trigger, a schedule or an event, was authored once and then fires without a human | Yes, each run starts without you       | The task passes its checks, or a limit is hit |
| Workflow         | Fixed code calls the model at specific steps; a human chose the path in advance                   | No, a human wrote the path             | The predefined path finishes                  |

TAKEAWAY

Plot the mode, never the product: the same chat interface moves cells the moment a connector is attached, and the same harness drops a row in plan mode.

# Who loops, who touches

|                  | RUNS ITS OWN LOOP | TOUCHES REAL SYSTEMS | STARTS WITHOUT A HUMAN |
|------------------|-------------------|----------------------|------------------------|
| Direct API call  | –                 | –                    | –                      |
| Chatbot          | –                 | Partial              | –                      |
| Deep research    | Yes               | Partial              | –                      |
| Agentic harness  | Yes               | Yes                  | –                      |
| Autonomous agent | Yes               | Yes                  | Yes                    |
| Workflow         | –                 | Yes                  | Partial                |

Partial cells are the interesting ones: a chatbot writes only once a connector is attached, deep research reads the world but writes nowhere, and a workflow can run on a schedule but a human authored its every step.

## TAKEAWAY

The two scales move independently. Autonomy without tool access is a long essay; tool access without autonomy is a form.

# What a harness actually does

- **Tool execution** Registers the actions the model can request (read a file, run a command, search), parses each request against a schema, runs it for real, and captures the output to feed back.
- **Permissions** Decides what runs automatically and what needs sign-off first: which commands are safe to fire, which directories are writable, which calls need a human in the loop.
- **Context management** Long tasks generate more transcript than fits in one context window, so the harness trims, summarizes, or drops old tool output to keep the task moving.
- **State and environment** Owns the container, working directory, and installed packages. The model only knows what the harness reports back; it never touches the environment directly.

---

## TAKEAWAY

Call the model, execute what it asks for, feed the result back, repeat. Everything else a harness ships is control over that loop.

# You are in the loop three times

- **1 • Kick off** You write the task in plain language. Everything downstream traces back to how much you specified up front; a vague prompt gets more decisions made for you.
- **2 • Approve or deny** Before anything risky (a command, an edit outside the project, an external call) most harnesses pause and ask: allow once, allow for the session, or deny.
- **3 • Review at exit** Once the loop stops you read the diff, the test output, or the opened pull request. The harness surfaces the result; you decide whether it ships.

---

## TAKEAWAY

The permission system is seat 2 in practice, and the autonomous row of Spec 01 is what happens when seat 1 is a schedule instead of a person.

SPEC 03

# Three ways the market has built it

CLI in your shell, agent inside the editor, cloud platform on a remote sandbox. Same loop, three surfaces, and four companies now sell into more than one row.

# The product table

| PRODUCT         | COMPANY            | SURFACE | LICENSE       | NOTE                                                   |
|-----------------|--------------------|---------|---------------|--------------------------------------------------------|
| Claude Code     | Anthropic          | CLI     | Proprietary   | Also ships as IDE extensions and a desktop app         |
| Codex CLI       | OpenAI             | CLI     | Open source   | Sandbox network access off by default                  |
| opencode        | anomalyco          | CLI     | Open source   | Roughly 200k GitHub stars                              |
| Antigravity CLI | Google             | CLI     | Closed source | Replaced Gemini CLI for consumer tiers, June 2026      |
| Cursor          | SpaceX (SpaceXAI)  | IDE     | Proprietary   | Acquired from Anysphere for \$60B, closed Aug 14, 2026 |
| GitHub Copilot  | Microsoft (GitHub) | IDE     | Proprietary   | Agent mode in VS Code and JetBrains                    |
| Devin           | Cognition          | Cloud   | Proprietary   | Sold across four surfaces since June 2026              |
| Jules           | Google             | Cloud   | Proprietary   | Scheduled, self-triggered runs since Dec 2025          |
| OpenHands       | All Hands AI       | Cloud   | Open-core     | Fully self-hostable                                    |

LangGraph, the OpenHands SDK, and CrewAI sit underneath all three tiers: libraries with no default interface, used to build one of the above.

## TAKEAWAY

License and surface are independent questions, and the tier turnover is fast: two of these rows changed hands or names in 2026 alone.

# Where MCP fits

- **One protocol, not N integrations** Before MCP every harness hand-built its own Slack, GitHub, and database connections. MCP standardizes discovery and calling into one protocol inside the tool-call box.
- **The server sets the risk** Whether an MCP tool is read-only or read-write depends entirely on the server. MCP standardizes how a tool is called; the harness still decides what needs your sign-off.
- **Adopted across all three tiers** Claude Code and opencode (CLI), Cline and Cursor (IDE), Copilot cloud agent and Jules (cloud). The tiers differ in trust: local harnesses accept any server, clouds curate.
- **Two siblings** ACP standardizes how editors host coding agents (Devin Desktop, JetBrains, VS Code). A2A lets agents negotiate with each other directly.

---

## TAKEAWAY

MCP is the mechanism behind the read-only and read-write columns of Spec 01: it makes the call possible, and the harness makes it permitted.

# The scoreboard depends on the scaffold

|                                                                                                                                                                                    |                                                                                                                                                                                                  |                                                                                                                                                                                       |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <div><div>~94%</div><div>SWE-BENCH VERIFIED</div><div>Frontier models cluster in the low-to-mid 90s; the set is flagged for contamination and kept mainly for history.</div></div> | <div><div>59.1%</div><div>SWE-BENCH PRO, STANDARDIZED</div><div>Scale AI's contamination-resistant successor: GPT-5.4 leads the public split under one shared scaffold, August 2026.</div></div> | <div><div>~20 pts</div><div>THE HARNESS EFFECT</div><div>Vendor-chosen scaffolds reach 80% on the same benchmark. The spread is produced by harness and data split alone.</div></div> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

One default worth checking: Codex CLI ships with sandbox network access disabled; Claude Code, Devin, and Manus reach the live web out of the box.

## TAKEAWAY

Any quoted score needs its split and scaffold named. Terminal-Bench 2.0, scoring autonomous end-to-end execution, is emerging as the more relevant test.

C L O S E

# The wrapper is the product now

Same engines, different cars. Judge the permission system, the context management, and the scaffold behind the score, not the model badge on the hood.