The Agent Has Left the Chat

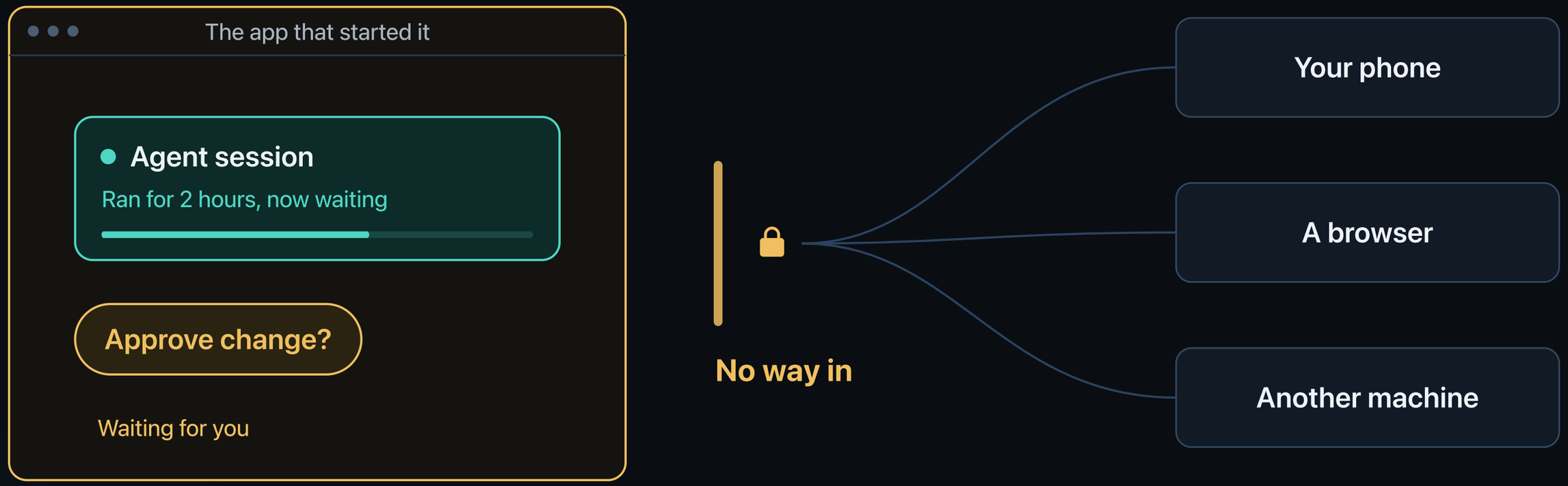
How any app can join a running agent session

Harald Kirschner · VS Code, GitHub Copilot



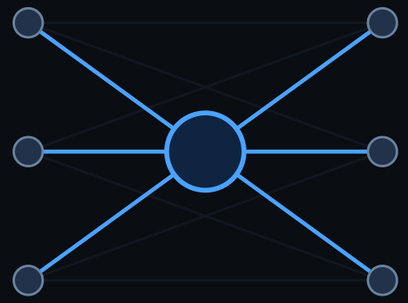
Your agent is waiting on you, and you're on another machine

Your session lives in the app that started it.



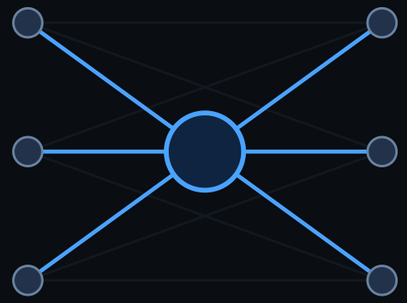
We've fixed this kind of problem four times already

Each time, every app rebuilt the same integration until one standard replaced it.



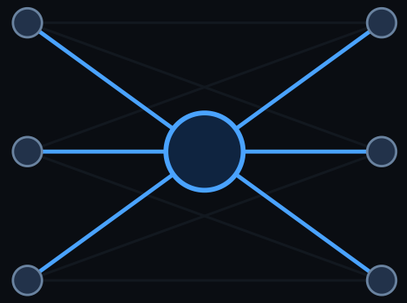
LSP

Language support



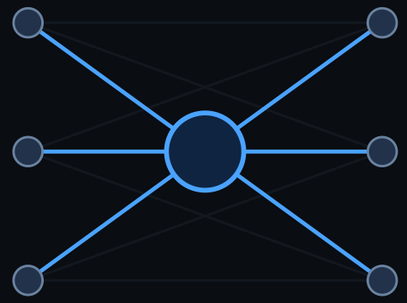
AGENTS.md and Skills

Team know-how



MCP

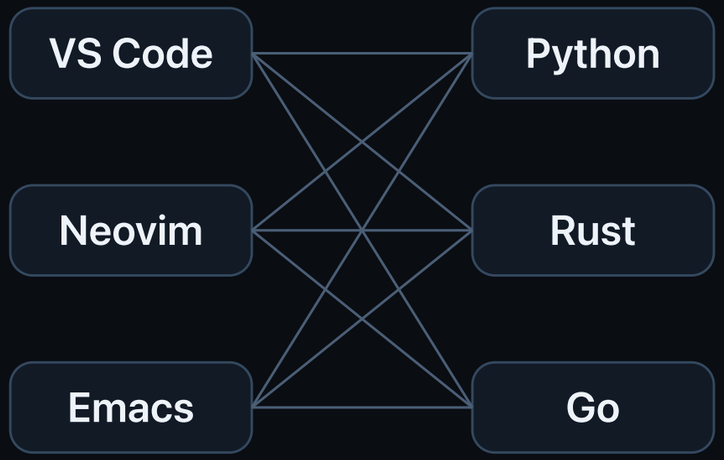
Tools and data



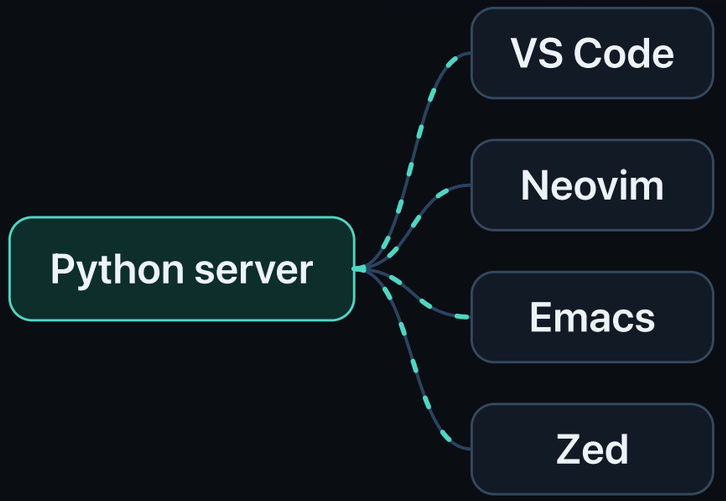
Agent Plugins

Bundled workflows

LSP let one language server work in every editor







THE PROBLEM

Each editor built completion, navigation, and diagnostics for each language.

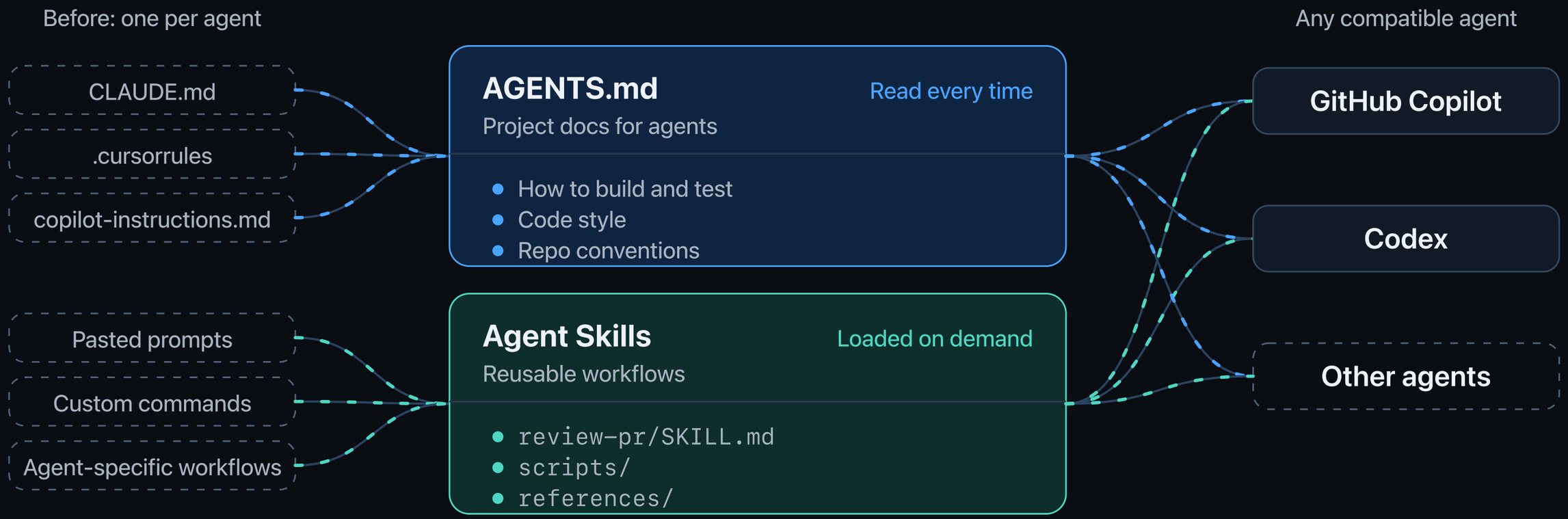
THE SHARED CONTRACT

Editors and language servers speak one protocol.

WHAT YOU GET

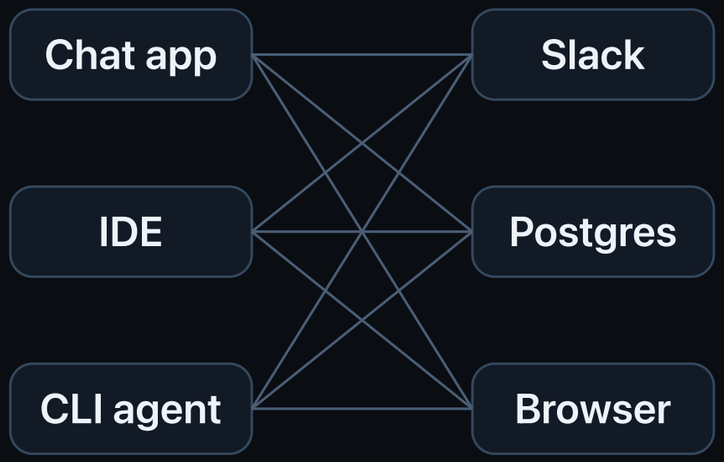
A language team ships one server. Every editor uses it.

AGENTS.md and Skills give every agent the same project docs and workflows

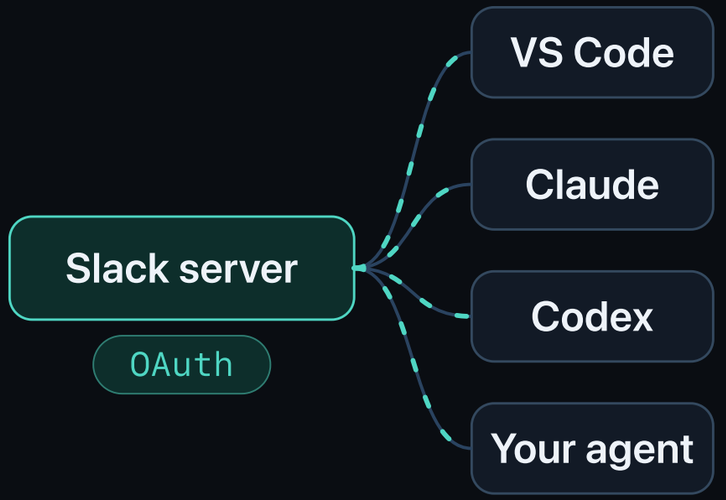


Two open formats. Write them once, and every compatible agent reads them.

MCP lets you build a Slack integration once, sign-in included







THE PROBLEM

Every AI app built its own Slack, database, and browser integration, including sign-in.

THE SHARED CONTRACT

A server exposes tools, resources, and prompts over one protocol, with standard OAuth.

WHAT YOU GET

You build the Slack server once, and every MCP client can use it.

The next MCP spec drops protocol sessions

01 **No session handshake**

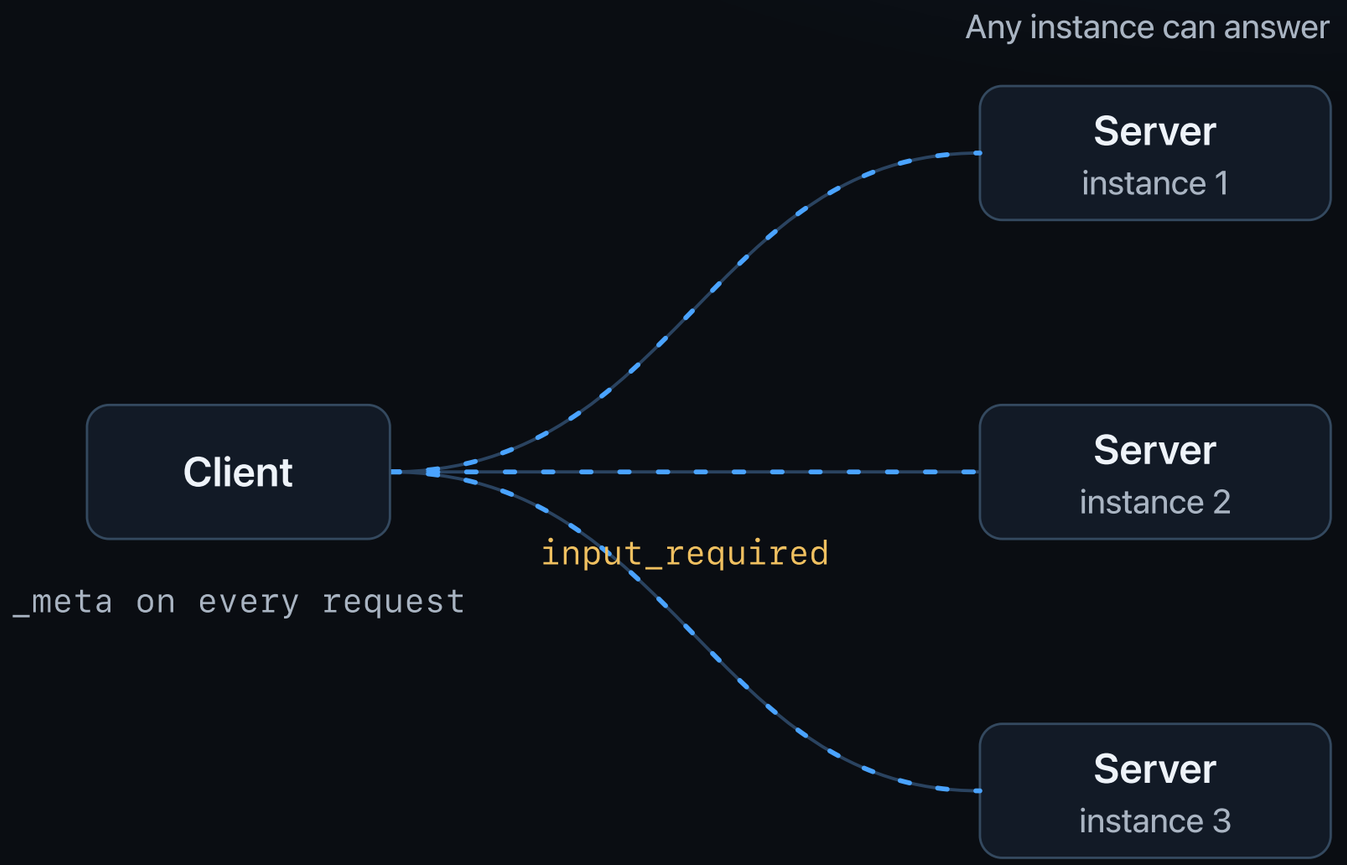
Every request carries its own version and capabilities.

02 **Long-running work moves to Tasks**

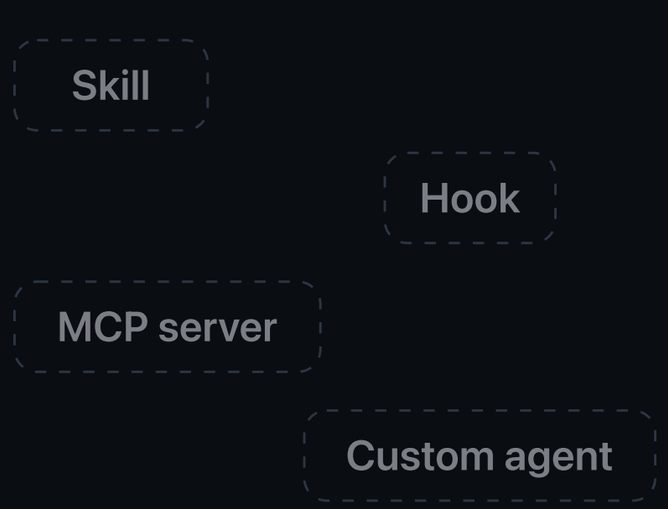
An official extension handles work that outlasts one request.

03 **Servers ask for input in the result**

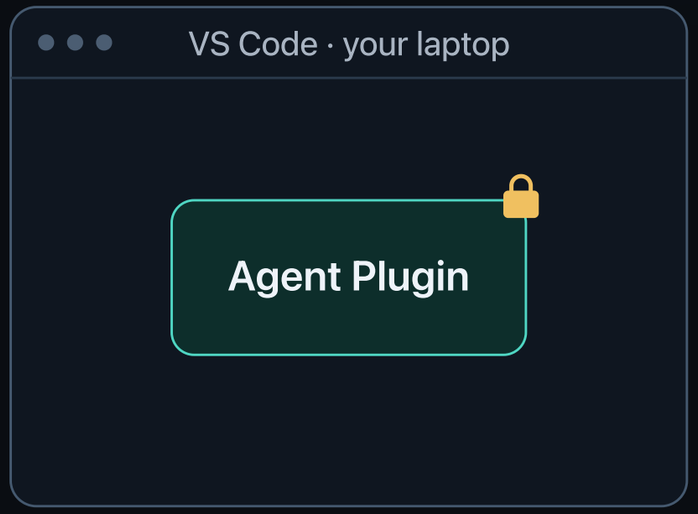
They return input_required instead of calling the client back.



Agent Plugins install a whole workflow in one step







THE PROBLEM

You set up skills, hooks, MCP servers, and custom agents one by one.

THE SHARED CONTRACT

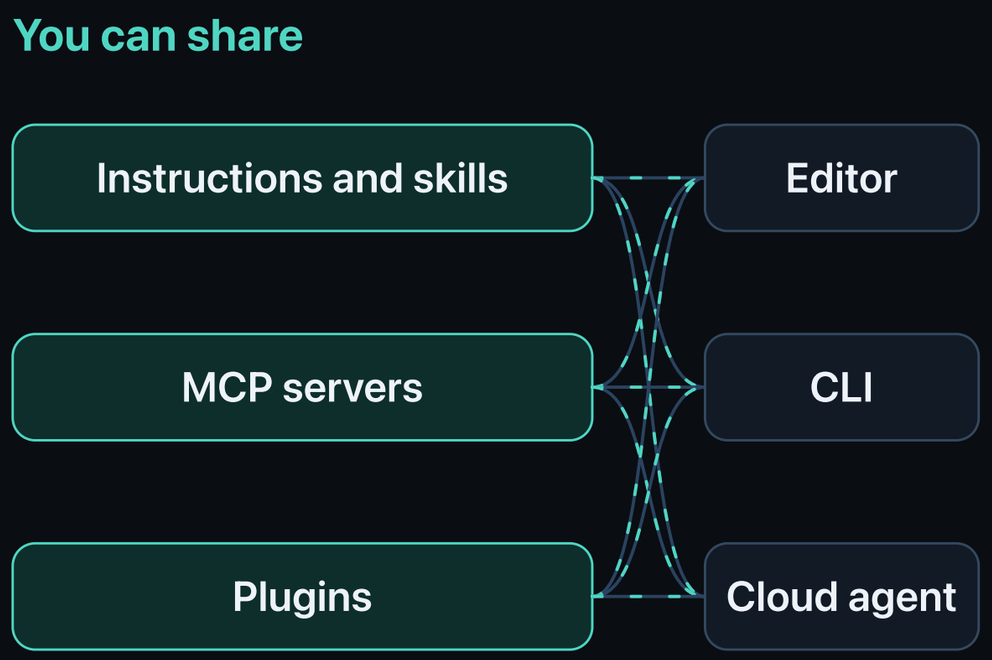
One plugin installs them together.

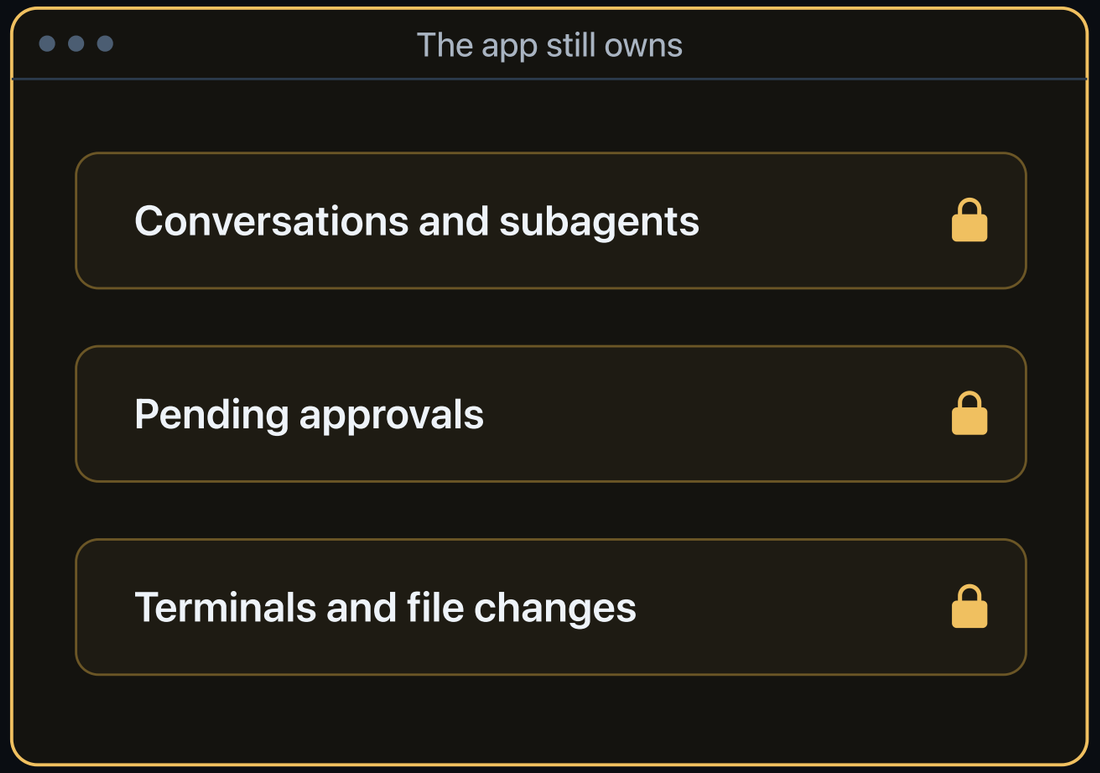
WHAT YOU GET

You get a review agent, the hook that enforces it, and its tools in one install.

But you install it in one window, on one machine.

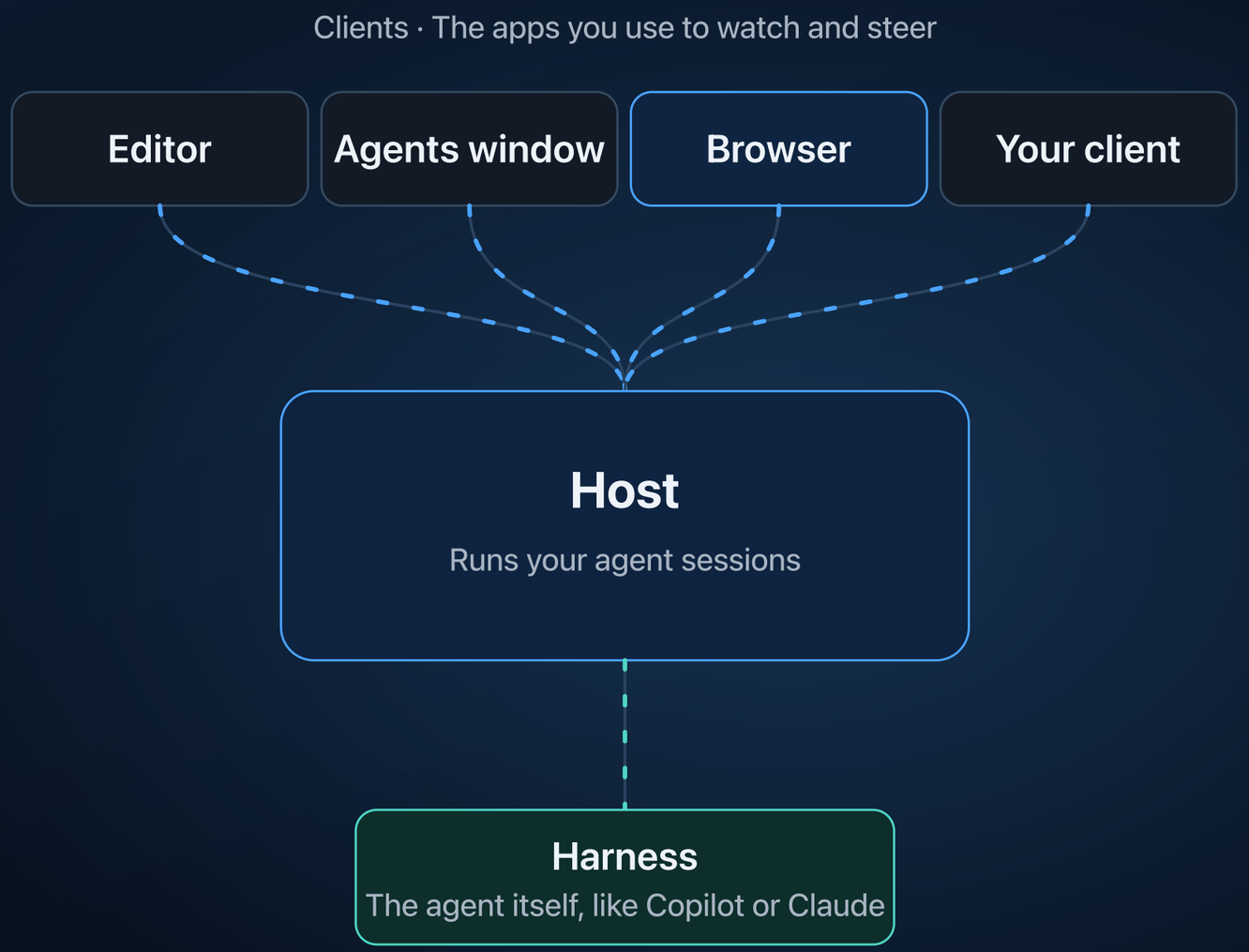
You can share your setup, but not a running session



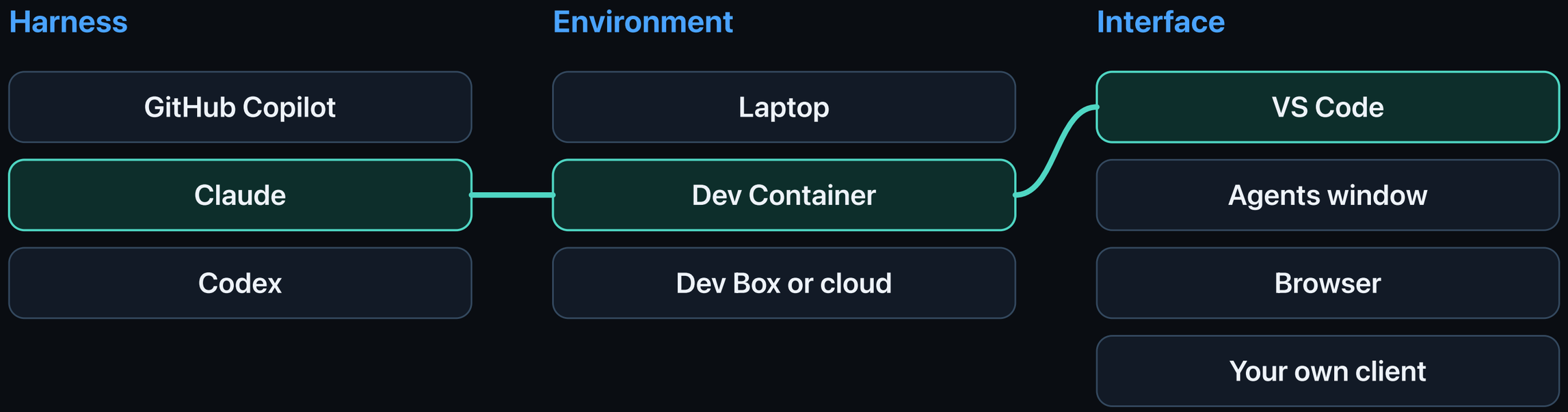


Agent Host Protocol

An open protocol that lets any app connect to agent sessions running on a host.

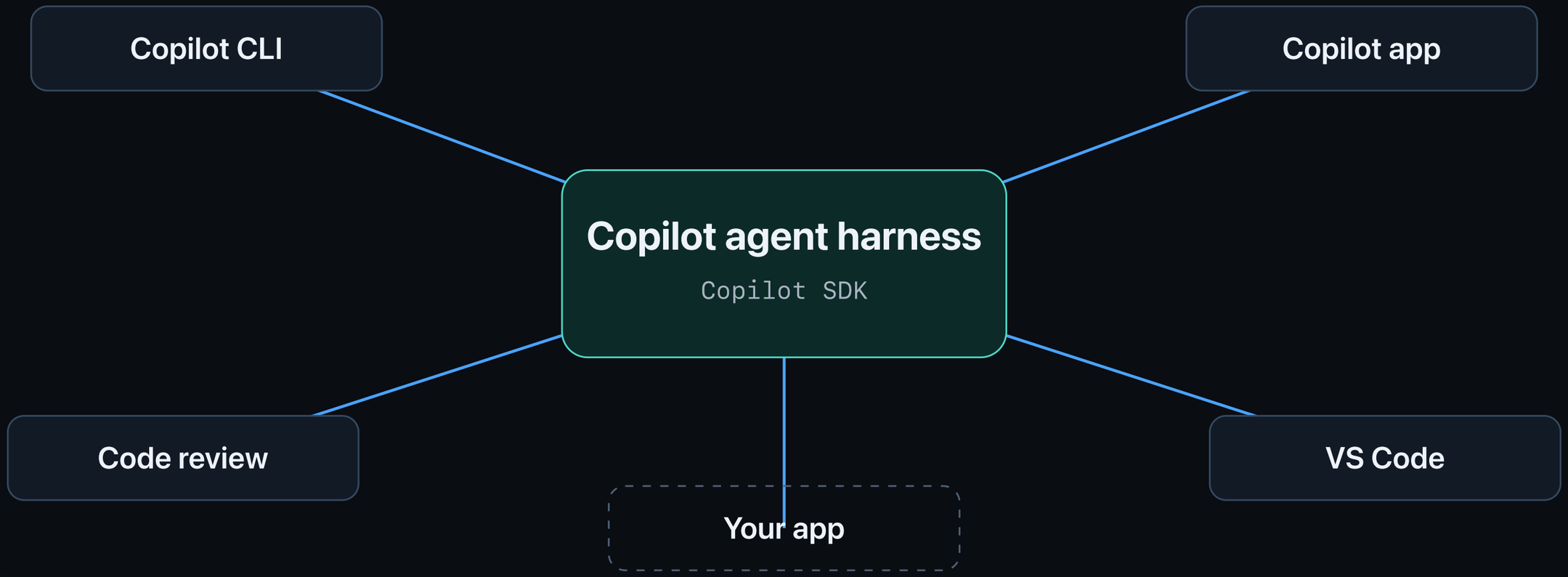


Choose the agent, where it runs, and how you work with it



Claude in a Dev Container on my desktop, used from VS Code on my laptop

Every Copilot product runs on the same agent harness



When we improve the harness, the CLI, the app, code review, and VS Code all get it.

Copilot CLI keeps up with Claude Code and Codex CLI on their own models



GPT-5.5

	Copilot CLI	Codex CLI
Completed	74%	73%
Cost / task	$0.59	$0.63

Copilot CLI: +1 pt, $0.04 less per task

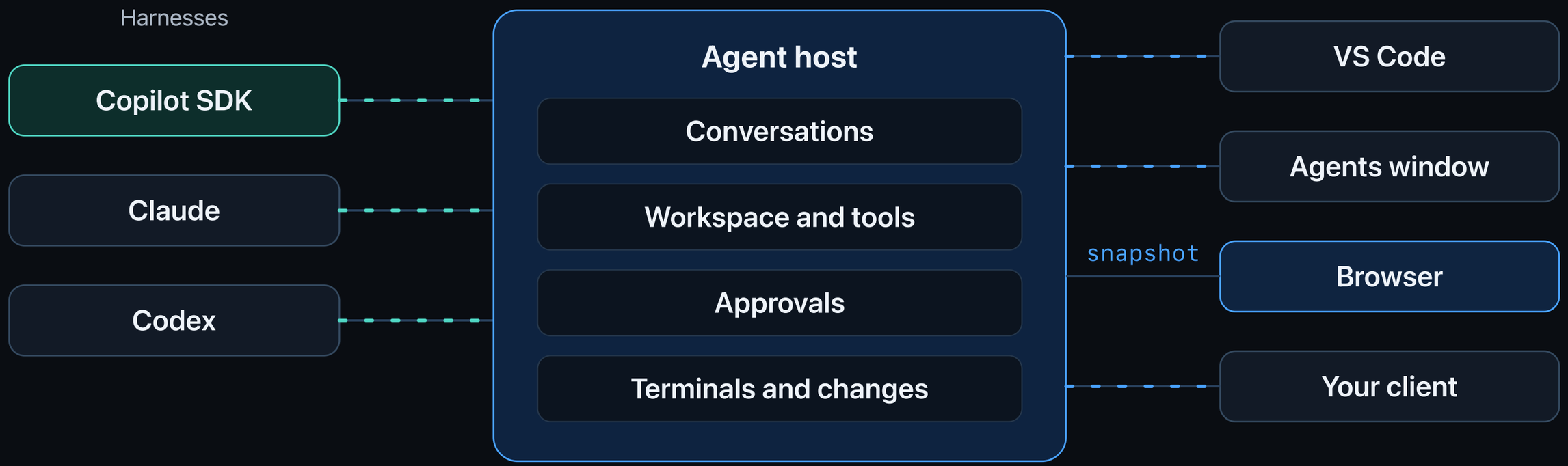
Opus 4.7

	Copilot CLI	Claude Code
Completed	75%	73%
Cost / task	$1.02	$0.99

Copilot CLI: +2 pts, $0.03 more per task

TerminalBench 2 · rounded from GitHub's chart · 5+ runs each · ellipses show run-to-run spread

Any app can join a session running on the host



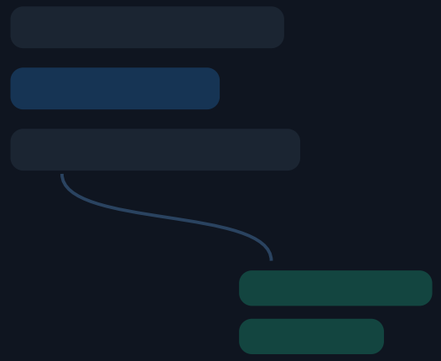
Every app sees the same state, because the host holds it.

Every client gets the whole session

See the work. Answer approvals. Review changes.

session

Conversations



Turns, side chats, subagents

Workspace

src/

host.ts

chat.ts

state.ts

Folders, setup, file changes

Tools

run_tests

Approve

supabase · ready

Calls, approvals, MCP status

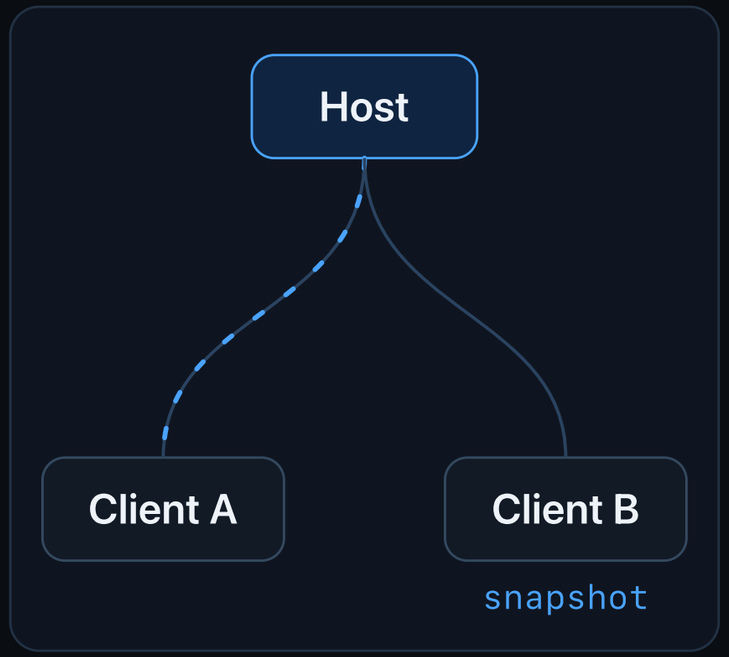
Live work

$ npm test

running

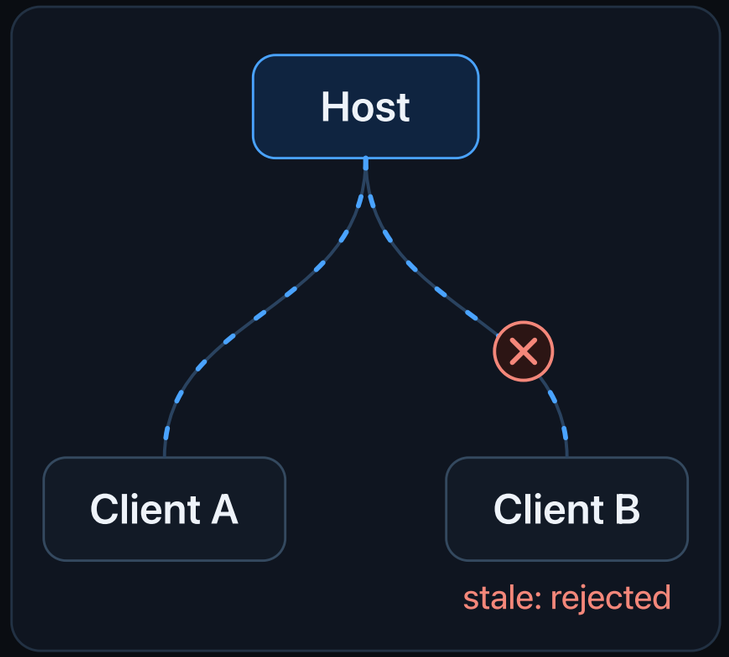
Terminals and progress

Pick up a running session from any device, even after you drop off



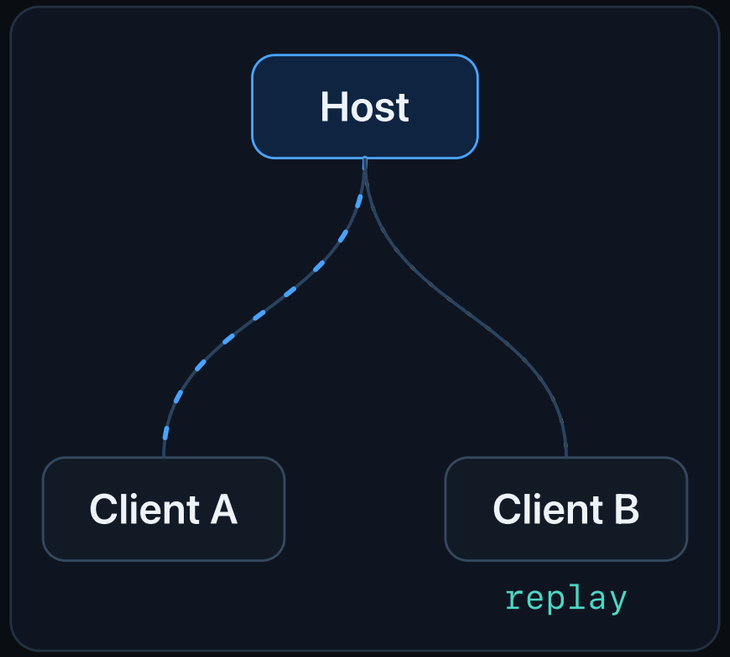
Join a session already running

You get a snapshot, then live updates.



Answer from whatever device is handy

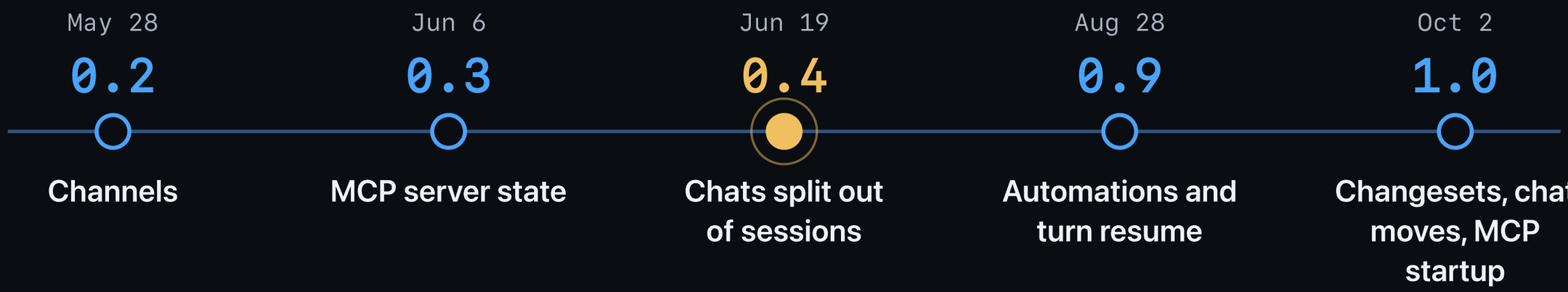
The host puts every action in order, so nobody answers an approval twice.



Pick up after a dropped connection

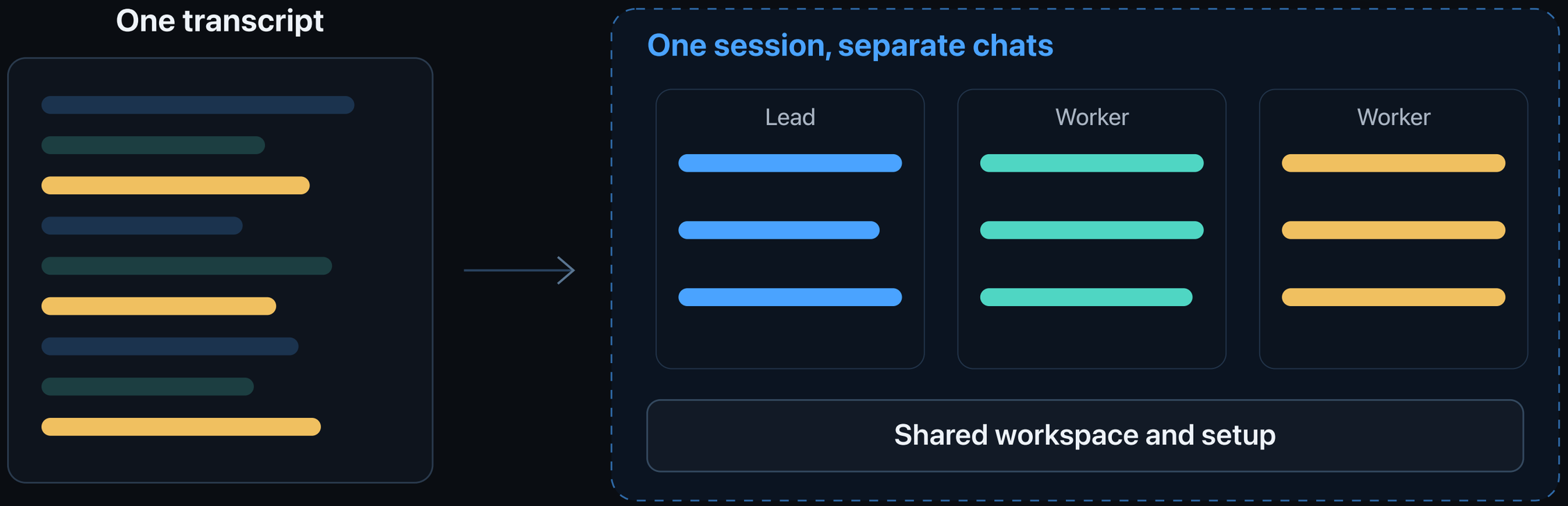
You get what you missed, or a fresh snapshot.

AHP went from 0.2 to 1.0 in four months



Every change came from a problem VS Code hit.

One session needed more than one conversation



Lead agent and workers in one stream

Separate sessions lost shared context

One chat per agent

One shared workspace and setup

Fork or move a single chat

Every client needs to see MCP server status

01 **Starting or ready**

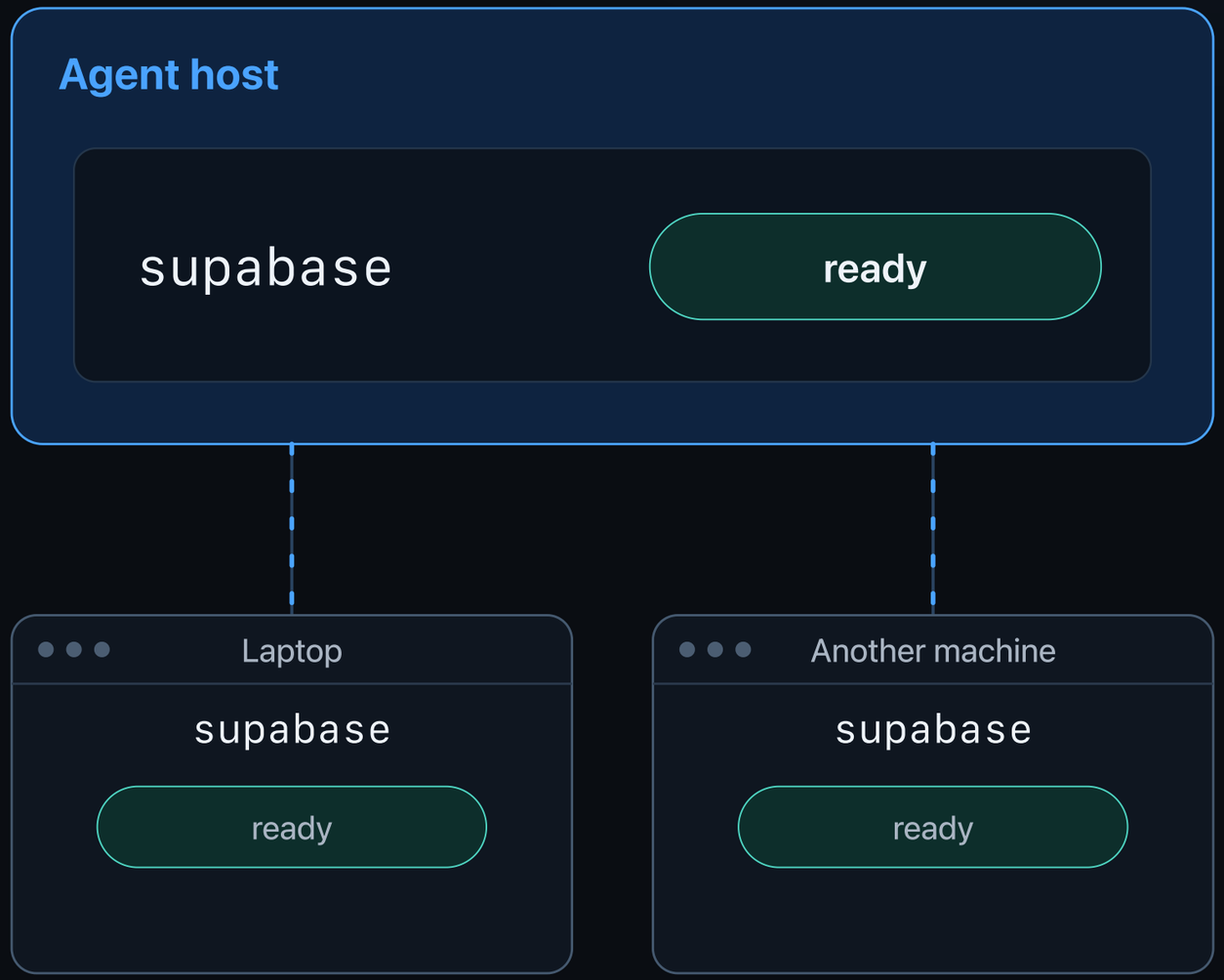
The host reports live status, not only the config.

02 **Needs sign-in or failed**

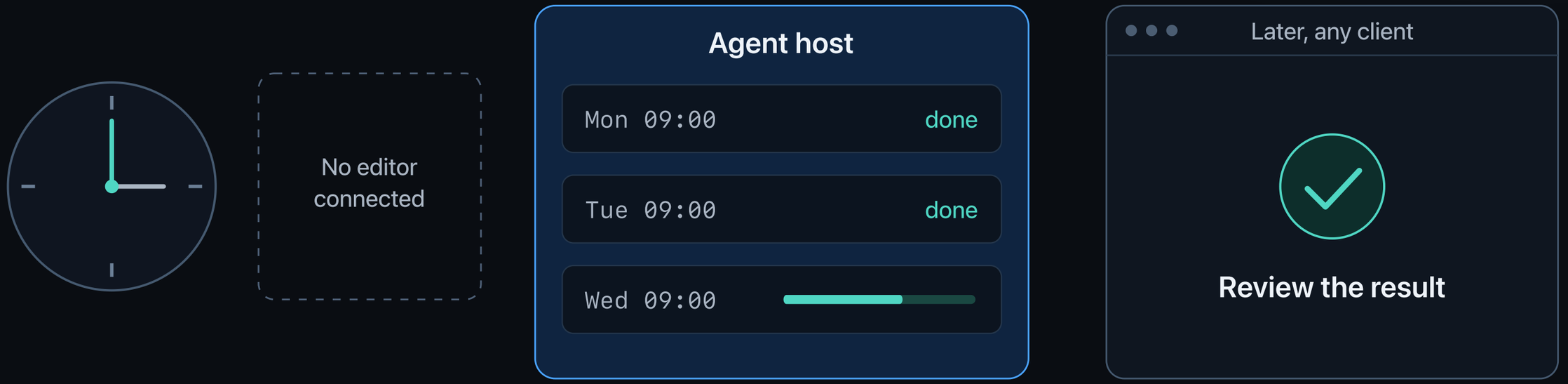
Any client can show what blocks the session.

03 **A turn waits for startup**

The session shows the wait instead of looking stuck.



Automations run without an editor open



THE PROBLEM

A scheduled task can't wait for someone to open an editor.

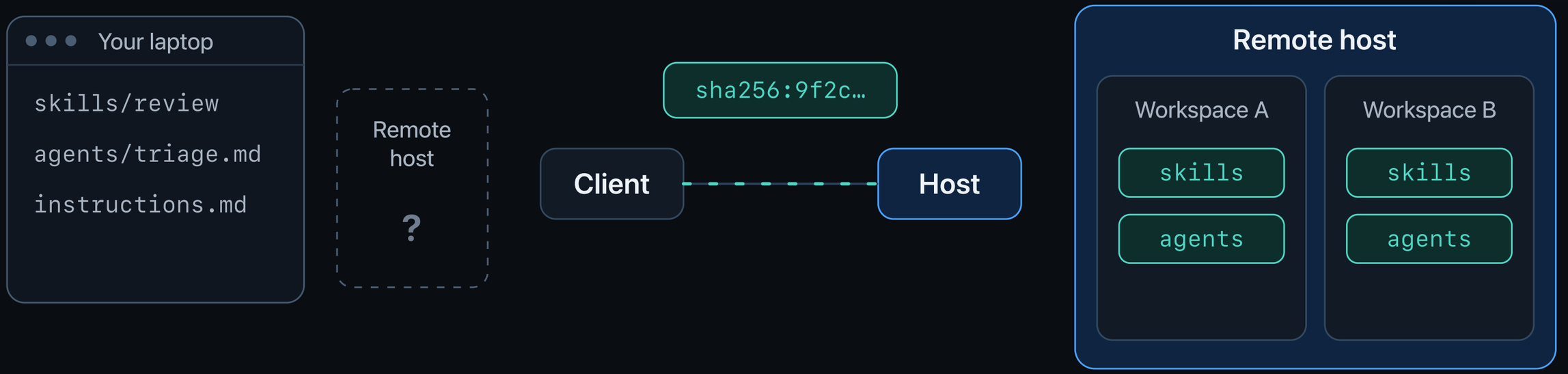
WHAT WE CHANGED

The host owns schedules, runs, history, and recovery.

WHAT YOU GET

Work starts on schedule. You review the result later, from any client.

Your setup has to reach the remote host



THE PROBLEM

Your skills and agents live on your laptop. The session runs somewhere else.

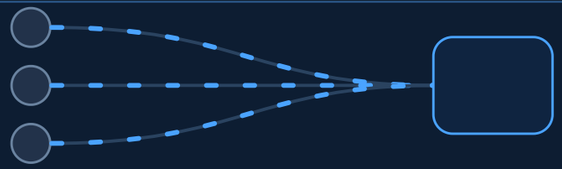
WHAT WE CHANGED

VS Code sends your setup to the host as content-hashed bundles.

WHAT YOU GET

Your setup works remotely, and workspaces don't overwrite each other.

Where AHP fits next to ACP and A2A

ACP	One client		**One agent harness** Embeds one agent in one editor
AHP	Many clients		**A host running sessions** Adds joining live work, shared approvals, and reconnecting
A2A	An agent		**Another agent system** Hands off a task and gets back results

ACP vs AHP

	 ACP	 AHP
Connection	One client ↔ one harness	Many clients ↔ one host
A second client joins	Your app has to build it	It gets a snapshot of the live session
Two clients act	The app coordinates them	The host orders and reconciles actions
Connection drops	Load or resume the session	Replay missed actions, or start from a snapshot

ACP agents can join an AHP workbench

- New in October: a community gateway, now listed in the AHP docs.



AHP

Shares sessions with every client

ACP

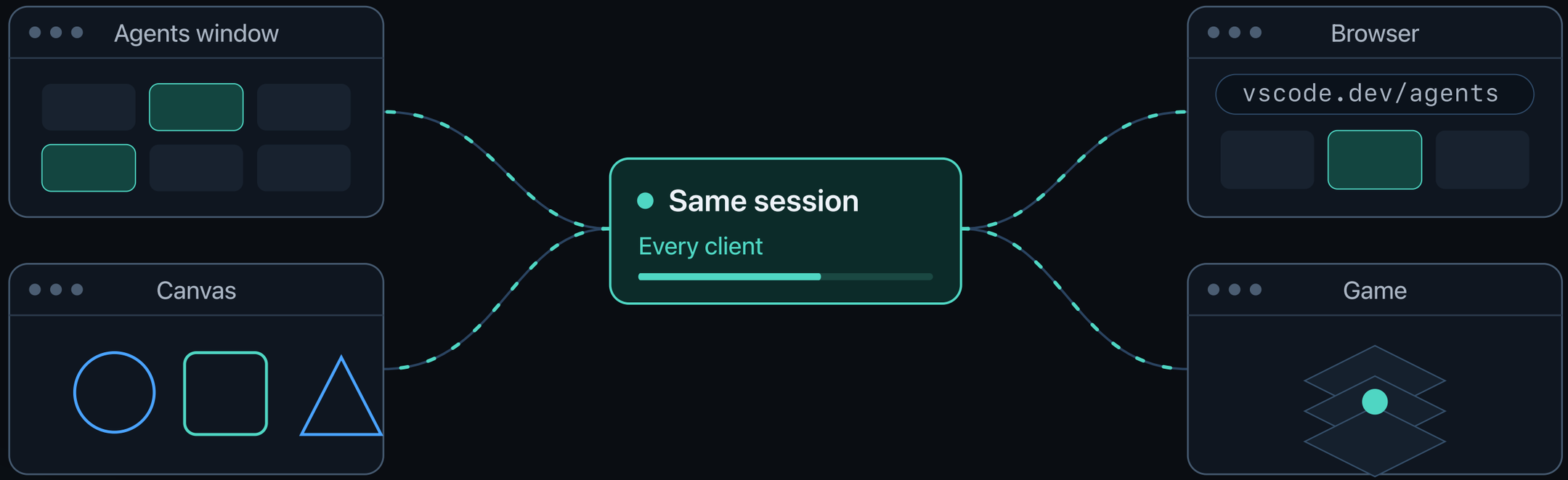
Runs the existing agent

A2A vs AHP

	 A2A	 AHP
Connects	An agent to another agent system	Clients to sessions on a host
Work	Tasks, messages, artifacts	Sessions, chats, approvals, changes
You see	The Agent Card and task results	The session state a client needs
Use it to	Hand work to another system	Watch and steer running work

The next agent UI doesn't have to look like an editor

Up next: the Agents window, a browser, a canvas, and a game.



Your agent session doesn't belong to one app anymore

`vscode.dev/agents`

`microsoft/agent-host-protocol`

`npx ahp-inspector`

