THE AI TOOLCHAIN

NO. 059

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 12, 2026 · EVERY DAY

The daily firehose — everything the toolchain shipped today, already filtered.

▸ // HOW THIS ISSUE IS MADE SHOW

VIEW

ISSUE VIEW

scroll



SCROLL | EXPAND IN PLACE | PREVIEW PANE

INFORMATION DENSITY

everything



BRIEFED | EVERYTHING

Every tool in the issue, in full.

all 199 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 38 tools

▾ // THE BRIEF HIDE

What stands out across today's releases, grouped by what it lets you do. Every tool named links to its entry below.

Three unrelated things stand out today. vLLM moves KV cache both ways between prefill and decode nodes, cutting TTFT up to 3.3x on -turn agentic workloads. Modal's Endpoint Candidates mirror live

vim ? help

traffic to a staged inference deployment before promotion. Hugging Face's new `contributor` token scope grants write access only to repos the holder created.

> DEPLOY
>
> **Prove a new inference deployment on real traffic before it serves anyone**
>
> Modal lets you stage an alternative Dedicated Endpoint, mirror production traffic at it, benchmark, then promote from the dashboard — the shadow-deploy rig teams usually build by hand. vLLM's bi-directional KV cache transfer makes disaggregated prefill/decode viable for multi-turn agents, where the cache now moves back as well as forward, and adds day-one Vera Rubin NVL72 support with Rubin-tuned kernels.
>
> Modal · vLLM

> GOVERN
>
> **Hand an agent or a contributor write access without handing over the whole org**
>
> Hugging Face's `contributor` token can write only to repos its holder created, so a fine-tune pipeline or outside collaborator stops needing blanket org write. Anthropic tightened network and URL-fetch restrictions for agents and took org management out of beta across the SDKs and CLI; Copilot CLI adds a session-scoped command to grant sandbox directory access rather than approving a path forever.
>
> Hugging Face · Anthropic · GitHub Copilot CLI

> BUILD
>
> **Resume a long agent run instead of starting it over**
>
> Agno makes sessions and background runs for Claude and Codex external agents durable and resumable, and Composio adds a pause-and-resume flow so a session can stop for human approval and pick up where it left off — the two failure modes that force a full re-run today. Kiro's CLI V3 carries persistent cloud sessions and memory across sessions.
>
> Agno

BUILD

◆ AI Coding Agents

claude-mem

Feedback ⓘ v13.36.0 NOTES ↗ RANK

Gives coding agents persistent memory by capturing session activity, compressing it with a model, and injecting relevant context into later sessions.

> claude-mem v13.36.0 adds four new environment-variable controls for memory injection format, tool-log retention and OpenRouter sampling, alongside new integrations (Screenpipe, Qoder, Atlas Cloud), a duplicate-memory review workflow, auth-method control in the Viewer, and idempotent memory uploads on the server.

└──▷ WHAT SHIPPED · 5 FEATURES most completely described first ? what's the number?

01 New environment-variable controls for memory behaviour NEW

95

Adds `CLAUDE_MEM_OPENROUTER_TEMPERATURE` to control OpenRouter sampling temperature, `CLAUDE_MEM_SEMANTIC_INJECT_FORMAT=index` for compact per-prompt memory injection, `CLAUDE_MEM_WORK_STATE_ENABLED=false` to disable the work-state to-do rule, and `CLAUDE_MEM_TOOL_USES_RETENTION_DAYS` to opt in to pruning of the raw tool log (default keeps everything).

Reduce OpenRouter token variation for more deterministic memory summarisation in a CI environment.

```
$ CLAUDE_MEM_OPENROUTER_TEMPERATURE=0.2 npx claude-mem install --provider openrouter
```

Switch to compact index-style memory injection to shrink per-prompt overhead when context windows are tight.

```
$ CLAUDE_MEM_SEMANTIC_INJECT_FORMAT=index npx claude-mem install
```

Keep the raw tool log from growing unboundedly by pruning entries older than 30 days.

```
$ CLAUDE_MEM_TOOL_USES_RETENTION_DAYS=30 npx claude-mem install
```

— Names exact env vars, defaults, and runnable install commands v13.36.0

02 Screenpipe, Qoder and Atlas Cloud integrations NEW

60

Adds a Screenpipe skill for searching meeting and screen history and saving results to memory, Qoder plugin support, and an Atlas Cloud preset for the OpenAI-compatible provider.

— Names three new integrations but no setup steps given v13.36.0

THINNER COVERAGE BELOW

03 **Duplicate memory candidate review workflow** NEW 35

Adds a workflow to record a distinct or dismissed verdict on duplicate memory candidates.

— States the two verdicts but not where or how to trigger review v13.36.0

04 **Auth method control in Viewer settings** NEW 35

Adds Claude auth method control to the Viewer settings panel.

— Names the UI panel but no detail on options or mechanism v13.36.0

05 **Idempotent memory uploads in server runtime** IMPROVED 30

Adds idempotent memory uploads to the server runtime.

— Bare statement with no mechanism or scope given v13.36.0

WAS THIS USEFUL? 👍 👎

yetone magpie

Feedback ⓘ v0.1.1217 CODE ↗ RANK

Magpie is a macOS menu-bar app, TUI, and CLI that lists every AI coding agent on the machine and switches each one's model, such as Codex on DeepSeek or Claude Code on Kimi, by rewriting only that key in the agent's own config.

> Across 13 releases, Magpie shipped a standalone vision-recognition MCP server, an IP/CIDR gateway allow list, zstd-compressed Codex-backend traffic for ChatGPT accounts, and broad provider/routing expansion — Grok image and video models, CodeBuddy and ZCode session handling, a new Hooks Library category, and Stays: Strict routing — alongside dozens of smaller usage, session, and platform fixes.

└──▷ **WHAT SHIPPED · 37 FEATURES** most completely described first ? what's the number?

01 Vision MCP server at /mcp/magpie/vision NEW 95

Exposes magpie's image-recognition engine as a dedicated MCP server at `/mcp/magpie/vision`, serving Streamable HTTP with one `analyze_image` tool so any agent — including those on vision-incapable models — can call it directly. Accepts images as data URLs in the request body (up to 32 MB) or as file paths (up to 20 MB) via the `analyze_image` tool's `image` argument, with an optional `prompt` argument to ask a specific question about the image. Guards the endpoint with the same gateway-key (`Authorization: Bearer <key>`) and local-network sharing rules used by the existing web-search MCP server.

Mount magpie's vision MCP server from a remote agent that runs on a text-only model, passing a gateway key to authenticate.

```
$ curl -X POST http://127.0.0.1:3425/mcp/magpie/vision \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer <key>' \
  -d '{"jsonrpc":"2.0","id":1,"method":"tools/call","params":
{"name":"analyze_image","arguments":
{"image":"<base64>","prompt":"What CVEs are
visible in this scan output?"}}}'
```

— Endpoint, tool, limits, and runnable example given v0.1.1203

02 Usage page client hide/unhide controls NEW 90

Adds `agentsHidden` field to `POST /api/usage/arrange` and the `UsageAgentsHidden` settings key to hide specific HTTP clients (e.g. `curl`, Go-http-client, `python-httpx`) from the Usage page Agents list while their calls still count in

totals and the chart, with a hide/unhide action menu (three-dot `DOTS` icon) on each client row and a collapsed `hidden` section below to restore them.

Hide noisy HTTP clients like curl and Go-http-client from the Usage page Agents list without losing their call counts in totals.

```
$ curl -X POST http://localhost:<port>/api/usage/arrange -H 'Content-Type: application/json' -d '{"agentsHidden":["curl","Go-http-client"]}'
```

— Named endpoint, settings key, and runnable curl example `v0.1.1205`

03 **IP/CIDR gateway allow list** NEW 88

Adds `MAGPIE_ALLOW_IPS` environment variable and a Settings UI field beside `Share on local network` to define an IP/CIDR allow list for the gateway.

Restrict gateway access to your local subnet and a specific host so only trusted machines can reach magpie.

```
$ MAGPIE_ALLOW_IPS=192.168.1.0/24,10.0.0.5 magpie
```

— Env var, UI field, and runnable command all given `v0.1.1214`

04 **zstd compression for ChatGPT-to-Codex backend traffic**

IMPROVED 80

Gateway compresses ChatGPT account turns with zstd at level 3 and sets `Content-Encoding: zstd` on `POST /responses` requests to the Codex backend, matching the behaviour of Codex's own client (`codex-rs/http-client/src/request.rs`) and avoiding 503 rejections from Envoy on large payloads — only ChatGPT-account `POST /responses` calls are compressed; API-key, other-provider, `/responses/compact` , and `/models` requests are sent plain.

— Endpoint, header, and exclusions fully specified `v0.1.1206`

05 **Hooks as fifth Library category** NEW 73

Adds Hooks as a fifth Library category: a hook is defined once and written into each selected agent's own hooks file (`<claude dir>/settings.json` , `<codex home>/hooks.json` , `~/.gemini/settings.json` , or `~/.cursor/hooks.json`) in that agent's native shape.

— Per-agent file paths are concrete and verifiable `v0.1.1214`

06 **'Get started' onboarding card** NEW 73

Adds a `Get started` onboarding card for new users showing three guided steps — add a model source, connect an agent, and optionally create a routing group — driven by `settings.GuideHidden` (persisted server-side so panel and window agree) and starter templates `steady`, `thrifty`, and `smart` from `provider.Templates`.

— Settings key and template names given, no exact navigation `v0.1.1212`

07 **Pi maxTokens set from model output limit** IMPROVED 71

Sets `maxTokens` in Pi's `~/.pi/agent/models.json` from the model's known output limit (falling back to Pi's own default when unknown), so Pi replicas of magpie models are no longer hard-capped at 16,384 tokens per reply.

— Exact file path and old cap value given `v0.1.1211`

08 **codex=as-sent provider setting** NEW 70

Adds a `codex=as-sent` setting on Responses provider configs, causing the gateway to forward Codex requests exactly as Codex sent them for relay providers that verify request origin.

— Exact config key named, config file location not given `v0.1.1214`

09 **MCP OAuth sign-in improvements** IMPROVED 68

MCP servers whose OAuth sign-in has no client-registration endpoint (such as GitHub's) are now signed in using the user's own OAuth app, and an MCP server's OAuth app callback is now accepted at Claude Code's `http://localhost:<port>/callback`, so apps made for Claude Code can sign magpie in.

— Callback URL named, still an internal sign-in mechanism `v0.1.1217`

10 **Resizable sidebar** NEW 68

Adds resizable sidebar via edge drag, with width clamped between `SIDE_MIN` (168 px) and `SIDE_MAX` (320 px) and persisted across sessions in `localStorage` as `magpie.sideWidth`; dragging past the minimum folds the sidebar away.

— Exact pixel bounds and storage key given `v0.1.1202`

11 **Grok and ChatGPT image/video model support** NEW 65

A Grok subscription now draws any `grok-imagine-image*` `-named` bare model and generates video via any `grok-imagine-video*` model, and a ChatGPT account now lists one GPT Image model, labelled as ChatGPT's pick.

— Model name patterns given, no exact selection steps `v0.1.1217`

12 **Platform-specific tray and window fixes** IMPROVED 65

On Linux, renders the tray panel as a solid opaque window (matching the main window) so scrolling no longer leaves ghost artifacts at the old position; supports the niri Wayland compositor by opening the tray panel under the bar beside the clicked icon (detected via

the `NIRI_SOCKET` environment variable) instead of in the middle of the screen; and on Windows, displays a message box with a link to the WebView2 download page when the WebView2 runtime is missing, instead of silently doing nothing on double-click.

— Env var and conditions named across three platforms `v0.1.1201` `v0.1.1212`

13 **Performance and startup optimizations** IMPROVED 65

Cuts CPU cost of reading a long conversation's request body to roughly a third, via a new lazy JSON decoder (`decodeJSON` in `internal/gateway/fastjson.go`) that skips fields the destination struct has no place for; skips re-parsing `config.toml` on Library and Sessions visits when the file has not changed; and limits warm-up reads at start to a quarter of available cores unless a page is waiting, retaining what was read after a long first read.

— Function and file named, but purely internal behaviour `v0.1.1200`

14 **Web search routing and fallback visibility** IMPROVED 63

Ensures the web searcher selected in Settings is never forwarded to a fallback provider and labels responses that were answered without searching; adds web-search fallback visibility in the gateway Routing row, showing which searcher failed or returned no results and why; and routes the agent web search to the searcher chosen in Settings › Web search, not the model the agent starts on.

— Clear settings path, three increments merged across releases `v0.1.1213` `v0.1.1214` `v0.1.1216`

15 **'Stays: Strict' routing mode** NEW 63

Adds Stays: Strict routing mode in the gateway and GUI, which pins a conversation to the account or key that answered it — even past its cap — and surfaces the error rather than moving to another.

— Mode behaviour described, no exact toggle path `v0.1.1214`

16 **Single-header Anthropic-compatible provider handling** IMPROVED 63

Handles third-party Anthropic-compatible providers that reject requests carrying both `x-api-key` and Authorization headers simultaneously, automatically retrying with a single header learned from the provider's 401 response.

— Headers named precisely, but fix is automatic `v0.1.1211`

17 **WorkBuddy restriction handling with appeal guidance** NEW 60

A restricted WorkBuddy account (403 code `11140`) is rested for 2 hours and Routing surfaces the vendor's appeal action (`SUBMIT_FEEDBACK`) as the path back.

– Exact code and action named, no user steps needed v0.1.1217

18 **Browser extension origin support** NEW 60

Allows browser extensions to be listed as trusted origins by entering a `chrome-extension://<id>` (or equivalent) URL in the web-pages allowlist in Settings.

– Exact URL scheme and settings location given v0.1.1216

19 **Rate-limit window visibility in Routing** NEW 60

Shows each rate-limit window and its reset time for subscription accounts under Routing's Accounts and keys view (matching data already shown on the Usage page), adds a manual refresh button beside the Accounts and keys heading to reload account allowances on demand, and displays each rate-limit window and reset time in the tooltip for an account's seat on the Routing stage.

– Named UI elements, no API surface given v0.1.1216

20 **Sessions full-text search and titling** NEW 60

Adds full-text search to the Sessions page so a session can be found by what was said in it, not only by title, folder, ID, or model; gateway sessions are now titled by their first message, making unnamed `ses_…` sessions discoverable by content search.

– UI path clear, no query syntax detailed v0.1.1213

THINNER COVERAGE BELOW

21 **Usage reporting improvements** IMPROVED 55

Usage stats now report how many of a model's replies were ended for looping and how many cost calls it could not price instead of silently counting them as free, while usage records and request detail views also track cache read and cache creation token counts alongside input/output tokens and add a new per-turn classifier that distinguishes magpie's own requests from the agent's requests in usage records on subscriptions.

– Two thin reporting tweaks with no access detail Metrics named, no endpoint or UI path given

v0.1.1217 v0.1.1216

22 **Per-account proxy routing for sign-in** NEW 55

Routes each subscription's sign-in and sign-out through the proxy configured for that account, rather than always using the global proxy, and shows the proxy a sign-in will use in the sign-in UI, letting a different proxy be picked and saved as that account's proxy.

– UI behaviour described, no exact navigation steps v0.1.1200

23 **ZCode plugin migration** IMPROVED 50

Migrates ZCode accounts to plugin `@magpie-community/opencode-zcode-auth` v0.1.16, restricting app sign-in to ZCode only when signed in to the same account.

— Plugin name and version given, automatic migration v0.1.1205

24 **User-Agent passthrough for key+URL providers** NEW 47

Enables a key+URL provider to send the agent's own User-Agent header in place of magpie's, letting upstream services see the agent's identity.

— Mechanism named but no config example given v0.1.1217

25 **Configurable context window on routing groups** NEW 45

Adds a user-configurable context window on routing groups and providers, which Codex and Claude Code respect instead of magpie's default 272K.

— Default value given, no config key named v0.1.1214

26 **Compaction window selection in Tuned for you panel** NEW 40

The Tuned for you panel now lets a compaction window be selected, showing estimated extra tokens and compaction count per option.

— UI behaviour described, no exact navigation path v0.1.1217

27 **Library Select bar group removal** IMPROVED 40

The Library's Select bar can now remove an entire group — skills, servers, or hooks — at once by checking the group's header.

— Clear UI action named, minimal mechanism detail v0.1.1217

28 **CodeBuddy Code session tracking** NEW 40

Adds CodeBuddy Code session tracking to Usage, reading its session files the same way WorkBuddy's are handled.

— Minimal mechanism, compared to existing WorkBuddy handling v0.1.1214

29 **spawn_agent full model naming** IMPROVED 40

Expands `spawn_agent` in the gateway to name every model a subagent may be placed on, not only the first five.

— Named tool, no usage example given v0.1.1214

30 **Claude Code gateway fixes** IMPROVED 40

Turns off server review for Claude Code on models Anthropic does not review, eliminating the false magpie isn't eligible error in auto mode, and improves gateway turn handling so a turn after a prompt suggestion continues in its run, reclaiming context from the suggestion, while a turn that finds no run reports the reason.

— Specific error quoted, automatic fix needs no action Behavioural fix only, no user action

v0.1.1213 v0.1.1210

31 **Startup model list auto-discovery** IMPROVED 40

Fetches a signed-in account's full model list at startup so the Routing picker exposes every available model without manual configuration.

— General mechanism, no endpoint or config named v0.1.1211

32 **logins.json write locking** IMPROVED 40

Locks all writes to `logins.json` under a shared gateway lock, and skips writing it entirely when a terminal `magpie` run only lists accounts.

— Named file, internal fix with no user action v0.1.1201

33 **Tool image passthrough for Kiro and Cursor** NEW 35

Kiro and Cursor's built-in providers now pass a tool's images to the model, including Codex's `view_image` output.

— Brief, no instructions for enabling or verifying it v0.1.1217

34 **Auto-skip revoked provider accounts** IMPROVED 35

Automatically skips accounts whose sign-in a vendor has revoked or refused, treating them as signed out rather than blocking every request.

— Behaviour change with no user-facing action v0.1.1213

35 **Hidden agents collapsed on Agents page** IMPROVED 35

Moves hidden agents to the end of the Agents page behind a closed disclosure, reducing visual noise.

— Minor UI reorganisation, thin description v0.1.1213

36 **Group-restricted model visibility** NEW 35

Adds the ability for a group to restrict its models from agent-visible lists, exposing them exclusively through the group itself.

— Thin description, no UI path given v0.1.1210

37 **Hermes bot profile provider support** NEW 30

Propagates magpie's provider to Hermes's other bot profiles, and removes it from each profile on Disconnect.

— Thin mechanism, no configuration detail v0.1.1200

NEW INTEGRATION Grok App integration

v0.1.1199 CODE ↗ STABLE

Adds Grok App as a supported agent integration, registering magpie as a provider channel in `agent-home/config.toml`.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.1.1199 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.1199
```

WAS THIS USEFUL? 👍 👎

All Hands AI OpenHands

Feedback ⓘ v1.27.0 NOTES ↗

RANK

OpenHands is an open-source software-development agent that writes code, runs commands, and browses the web.

> OpenHands v1.27.0 adds new browser-automation test commands, new ACP agent integrations (OpenCode, Pi, DigitalOcean Managed Agents), and several UI additions for automation setup and navigation.

└──▷ **WHAT SHIPPED · 6 FEATURES** most completely described first ? what's the number?

01 **Browser control commands in verify-openhands** NEW 83

Adds `browser key down`, `browser key up`, and `browser key release-all` commands to the verify-openhands browser controller for holding and releasing keys during automated browser sessions, plus a `browser record` command to capture videos of browser sessions in end-to-end tests.

Hold a modifier key during a browser automation sequence in verify-openhands, then release all held keys at the end.

```
$ browser key down Shift
browser key down ArrowRight
browser key release-all
```

Capture a video of a browser session while running a verify-openhands end-to-end test for later review.

```
$ browser record
```

— Names exact runnable commands with usage examples. v1.27.0

THINNER COVERAGE BELOW

02 **OpenCode and Pi ACP agents with dynamic model lists** NEW 50

Adds OpenCode and Pi as built-in ACP agents available on local backends; ACP agents now supply their own model lists dynamically instead of using hardcoded values.

— Names the agents but no usage path given. v1.27.0

03 **DigitalOcean Managed Agents in Agent Canvas** NEW 45

Connects Agent Canvas to DigitalOcean Managed Agents, supporting both desktop and web.

— Names the integration but lacks setup detail. v1.27.0

04 **Automate setup split panel with draft management** NEW

45

Adds an automate setup split panel with agent-assisted automation form updates and persisted automation draft management.

— Describes the feature but no UI path or config given. v1.27.0

05 **Compact navigation for tablet viewports** NEW

20

Adds compact section navigation for tablet viewports.

— Thin one-line UI addition. v1.27.0

06 **Splash page mode toggle** NEW

15

Adds a splash page mode toggle to the UI.

— Bare mention with no mechanism or location. v1.27.0

WAS THIS USEFUL? 👍 👎

Graphify Labs Graphify

Feedback ⓘ v0.9.87 NOTES ↗

LATEST STABLE

Graphify converts code, docs, SQL schemas, configs, and PDFs into a queryable knowledge graph with explainable relationships.

Adds per-extension language overrides in `.graphifyrc` and a light/dark/system theme toggle in `graph.html`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.9.87 https://github.com/Graphify-Labs/graphify.git
# already have the repo? check out this version:
$ git checkout v0.9.87
```

NEW SUPPORTED PLATFORM Adds `language.<ext>=<language|.ext>` lines in `.graphifyrc` to declare what an ambiguous file extension means, flowing through classification, extractor selection, imports, watch, and the AST cache.

Adds a light/dark/system theme toggle to `graph.html`; the dark rendering remains byte-identical to the previous default.

UPDATED SUPPORTED PLATFORM PHP static call and self/static/parent resolver improvements

UPDATED SUPPORTED PLATFORM C# nested/generic partial type merging and var-typed member resolution

UPDATED SUPPORTED PLATFORM Python @property getter linking

UPDATED SUPPORTED PLATFORM TypeScript private-field (#name) call edge correctness

UPDATED SUPPORTED PLATFORM Julia parametric and module-qualified functor type naming

UPDATED SUPPORTED PLATFORM Julia language extraction

UPDATED SUPPORTED PLATFORM Rust grouped use-declaration expansion

UPDATED SUPPORTED PLATFORM Scala 3 derives clause extraction

UPDATED INTEGRATION Codex semantic chunk persistence

└──▷ BREAKING ON UPGRADE

! The AST cache schema is bumped by the C# partial-type fix and the Python property-getter fix; the first build after upgrading will re-extract the full corpus automatically.

▾ **1 more release in this issue** · 2026-10-11 → 2026-10-12

v0.9.86 NOTES ↗ STABLE

Adds `explain --all`, distinct exit codes (0/3/4) for lookups, Ollama sampling seeds, and a safety guard on `uninstall --purge` when `GRAPHIFY_OUT` is unset.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.9.86 https://github.com/Graphify-Labs/graphify.git
# already have the repo? check out this version:
$ git checkout v0.9.86
```

Adds `explain --all` flag to retrieve the full answer when output is truncated to the token budget.

Adds distinct exit codes for `explain` / `query` lookups: 0 for an answer, 3 for an ambiguous name, 4 for a missing name — enabling scripts to distinguish a miss from a hit.

Built-in Ollama requests now accept a sampling seed for reproducible extraction.

`graphify uninstall --purge` now refuses to delete the project, home, or a filesystem root when `GRAPHIFY_OUT` is empty or `.`.

Committed `graph.json` is now written as compact single-line JSON, producing a `+1/-1` diff on re-build instead of large whitespace churn.

└──▷ **BREAKING ON UPGRADE**

! The AST cache schema is bumped for Rust grouped `use` declarations; the first build after upgrading will re-extract all Rust files.

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ v0.34.12 NOTES ↗ RANK

Letta Code is a terminal coding agent with persistent memory and identity across development sessions.

> Letta Code added the ability to orchestrate Claude Code and Codex as subagents with live progress, new workflow primitives, an organization ID environment variable, and a new `letta mcp connect` command for OAuth-based MCP server connections.

└──▷ **WHAT SHIPPED · 4 FEATURES** most completely described first ? what's the number?

01 **MCP OAuth connect command** NEW 80

Adds the `letta mcp connect` subcommand, enabling localhost-callback MCP OAuth flows for connecting MCP servers — useful when an MCP provider requires browser-based sign-in.

Authenticate and connect an MCP server that uses a localhost OAuth callback — useful when your MCP provider requires browser-based sign-in.

```
$ letta mcp connect
```

— Exact runnable command paired with a usage example v0.34.11

02 **Organization ID in agent shell env** NEW 65

Exposes `LETTA_ORGANIZATION_ID` as an environment variable in the agent shell env, making the organization ID available to scripts and tools running inside agents.

— Names the exact env var but no usage example given v0.34.12

03 **Claude Code and Codex subagent execution** NEW 60

Letta agents can run Claude Code and Codex as subagents on a connected computer, enabling orchestration of external AI coding agents from within a Letta agent. Their tool calls are shown as live progress in the UI while the subagents execute.

— Explains mechanism and UI behavior but gives no command to try it v0.34.12

THINNER COVERAGE BELOW

04 **Workflow system gains foreground wait and tool introspection** NEW 50

Adds foreground wait, built-in tool calls, and `listTools` / `describeTool` to the workflow system.

— Names new workflow primitives but no usage shown v0.34.12

WAS THIS USEFUL? 👍 👎

Red Hat ripwire

Feedback ⓘ v0.6.7 NOTES ↗ RANK

Gives coding agents a ranked, deterministic map of a repository — call graph, blast radius, and tests to run — from a CLI or over MCP.

> ripwire 0.6.7 adds targeted single-symbol retrieval via `--expand`, a `--quality-delta` regression report with 12 delta kinds, Ruby/Rails call-graph support, and resumable paging across `--for` / `--pack-task` /MCP `explore` — alongside a breaking default change to session routing (`arm=auto` in `~/.ripwire/meter.conf`) and a parser/cache version bump that forces an automatic rebuild.

└──▷ **WHAT SHIPPED · 15 FEATURES** most completely described first ? what's the number?

01 Lean per-question query flags with call-site lines IMPROVED

86

`--callers` and `--path` lean-answer flags join `--for` and `--expand` for targeted per-question queries, exposing `p=` (caller definition location) and `sites_at=` (call sites as pasteable `file:line` tokens) in caller answers so lines to edit arrive with the first response; `sites_at=` was also added to `--safe-delete` rows.

Find every call site of a function and get the exact file:line coordinates to edit in one pass, without a follow-up lookup.

```
$ ripwire --callers=escapeXml
```

— Exact field names and a runnable --callers command. v0.6.7

02 Session routing control via arm= in meter.conf BREAKING

86

Adds an `arm=` key to `~/.ripwire/meter.conf` to control session routing; `arm=treatment` restores the prior full-routing behaviour. A fresh install with no `meter.conf` now defaults to `arm=auto`, splitting sessions so roughly half get no routing hints and no SessionStart primer — add `arm=treatment` to the config to restore the previous behaviour.

Keep the prior full-routing session behaviour after upgrading, since the new default splits sessions for A/B control.

```ini
arm=treatment
```

— Exact config key, file path, and migration value given. v0.6.7

03 **Resumable paging for --for and --pack-task** NEW 81

Adds a `--limit` / `--offset` resumable paging window to `--for` and `--pack-task`, and `limit` / `offset` parameters to MCP `explore`.

Page through a large `--for` answer in chunks to avoid token-budget overruns in an MCP or agent context.

```
$ ripwire --for='what calls grid_set_cell?' --limit=20 --offset=0
```

— Names exact flags, targets, and a runnable paged command. v0.6.7

04 **Single-symbol retrieval via --expand** NEW 80

`--expand=SYM` retrieves a single symbol's body together with its ranked-neighborhood bundle, costing roughly 260–16.5K tokens versus roughly 43K–174K tokens for reading the whole file.

Orient once on an unfamiliar repo, then drill into a specific symbol — getting its body plus ranked neighbors in a fraction of the tokens of reading the whole file.

```
$ ripwire . && ripwire . --expand=SYM
```

— Quantifies token savings with exact numbers and a runnable command. product docs

05 **--quality-delta regression reporting** NEW 80

`--quality-delta` reports only regressions across 12 delta kinds, including `placeholder` (stubs and TODOs a change adds, non-gating) and `defect-shape` (four defect shapes caught by review, gating on format/argument mismatch).

Run quality delta on changed code to surface only regressions, including new stubs and format-argument mismatches, without blocking on non-defect debt.

```
$ ripwire . --quality-delta
```

— Names flag, delta kinds, and gating behaviour with runnable example. product docs

06 **Budget-cut disclosure in orient and expand** NEW 75

`--top-k=0` is now the default for `--expand` and `--outline` so no ride-along map is sent when every requested definition is served, disclosed via `topk_default="0"` with `map_next=` naming the call to fetch the map if needed (an explicit `--top-k=N` restores the map). `unserved_total=` and `unserved_next=` name definitions cut by

the byte budget so they can be recovered without a follow-up grep, and `--max-tokens=N` caps the orient map while disclosing what was cut via `shown=`.

Orient in a context-constrained environment by capping the map; the cut is disclosed so you know what was dropped.

```
$ ripwire . --max-tokens=8000
```

— Names every disclosure field but no single command showcases them all. product docs

07 **Branch-aware declaration labeling in --whereis** IMPROVED

70

`--whereis` now parses blobs on other branches to label rows as declarations vs. call sites rather than returning raw hits, and caps long-line `t=` fields at a 200-byte preview with full content replayed via `next=`.

— Explains mechanism and named fields but no runnable example. v0.6.7

08 **Ruby and Rails support in --callers and schema rendering** NEW

65

Adds Ruby method-lookup resolution to `--callers` and related verbs, covering mixins, class objects, `self`, built receivers, RSpec targets, and Rails callbacks, plus rendering of Rails `db/schema.rb` columns as first-class definitions.

— Names covered constructs but no example command. v0.6.7

THINNER COVERAGE BELOW

09 **Deep subclass walk in --lego** IMPROVED

55

`--lego` now walks the full subclass tree, returning implementors at every depth via `via=` and `depth=` fields.

— Names fields but gives no example command. v0.6.7

10 **191 flags organized into seven help families** IMPROVED

50

Groups 191 long flags into seven families accessible via `--help` and `--help=<family>`.

— Exact command shown but no family names or further detail. product docs

11 **Parser and cache version bump forces rebuild** BREAKING

50

Parser version is now 167 and cache version is now 31; caches and quality snapshots from earlier builds are rebuilt automatically on first run.

— Gives exact version numbers but automatic migration needs no action. v0.6.7

12 **Per-file reach rollup in --impact** IMPROVED 45

`--impact` now rolls reach up per file, naming each file that transitively depends on the queried symbol.

— Explains behaviour change but no example or field name. v0.6.7

13 **ripwire wrap command for budget-free setup** NEW 40

Ships `ripwire wrap` to write budget-free default setups — budgets remain a caller choice, never imposed by the tool.

— Names the subcommand but no usage example or flags shown. product docs

14 **Doc-link counts in --seams** NEW 40

Adds `--seams` reporting of `doc_links=` count, separating doc-to-doc links from untested seams.

— Names the field but no example or further mechanism. v0.6.7

15 **Memory guard on every root** IMPROVED 30

Adds a memory guard on every root that returns a disclosed partial answer instead of unbounded growth.

— States behaviour change but no named field or example. v0.6.7

└──▷ **BREAKING ON UPGRADE**

! An install with no `~/.ripwire/meter.conf` now defaults to `arm=auto`, splitting sessions so roughly half get no routing hints and no SessionStart primer. Add `arm=treatment` to `~/.ripwire/meter.conf` to restore the previous full-routing behaviour.

! Parser version is now 167 and cache version is now 31; caches and quality snapshots from earlier builds are rebuilt automatically on first run.

WAS THIS USEFUL?  

Amazon Kiro

Feedback ⓘ

Models: Claude Opus 5.5 now available in AWS GovCloud (US): Claude Opus 5.5 is available

BLOG ↗ RANK

Kiro is an agentic development environment that uses spec-driven workflows to plan, build, and maintain software.

> Kiro's biggest release this window is CLI V3, adding persistent cloud sessions, spec-driven workflows, cross-session memory, tangent threads, multi-agent coordination, and agent migration tooling, alongside several breaking changes to the legacy CLI experience. Claude Opus 5.5 and Sonnet 5.5 also became available in AWS GovCloud.

└──▷ **WHAT SHIPPED · 9 FEATURES** most completely described first ? what's the number?

01 Bulk agent upgrade from V2 to V3 NEW

86

`kiro-cli agent upgrade <dir>` (v2.29+) bulk-upgrades a directory of custom agents from V2 to V3 format, with a `--dry-run` flag to preview changes before committing. `/upgrade-agent` runs or repeats the agent configuration migration and shows diagnostics.

Bulk-upgrade a shared agent directory to the V3 format, previewing changes before committing.

```
$ kiro-cli agent upgrade ./agents --dry-run
```

— Exact subcommand, flag, and version gate given. launch-20261009-25f11f16

02 V3 agent harness opt-in and ACP migration path BREAKING

85

`kiro-cli --v3` opts into the V3 agent harness ahead of the CLI 3.0 default rollout, `kiro-cli --v2` switches back to the V2 harness on a per-launch basis, and `kiro-cli acp --agent-engine=v3` starts the V3 ACP server for ACP client integrations. Sessions created in V3 cannot be resumed in V2, and agent configuration migration does not automatically update ACP integrations — ACP clients must follow the ACP migration guide and use `kiro-cli acp --agent-engine=v3`.

— Exact flags and the compatibility break are named. launch-20261009-25f11f16

03 Spec mode, memory, tangents, and session browsing in CLI NEW

83

`/spec new` enters spec mode to review requirements and design before Kiro writes any code, supporting inline comments and selective task execution. `/memories` manages cross-session memory so Kiro learns repository commands and conventions and recalls them later. `/tangent` branches into a parallel side thread without affecting the main session context. `/sessions` lets you browse, bookmark, and rename sessions.

Review requirements and design with Kiro before it writes any code, leaving inline comments on the spec.

```
$ /spec new
```

— Four named commands with behavior and one runnable example. launch-20261009-25f11f16

---

04 **Persistent cloud CLI sessions** NEW 80

`kiro-cli --cloud` starts a persistent cloud session that keeps running after you close your laptop and can be resumed from another machine; currently available only in the US East (`us-east-1`) region.

Hand off a long-running task to a cloud session so work continues after you close your laptop.

```
$ kiro-cli --cloud
```

— Runnable command with clear scope, no deeper mechanism given. launch-20261009-25f11f16

---

05 **`use_aws` tool removed in V3** BREAKING 73

The `use_aws` tool is removed in V3; AWS CLI commands must now run through the shell tool, and existing `use_aws` allow/deny rules must be recreated as shell permission rules.

— Names the removed tool and the required migration step. launch-20261009-25f11f16

---

06 **Claude Opus 5.5 and Sonnet 5.5 in AWS GovCloud** NEW 68

Claude Opus 5.5 and Claude Sonnet 5.5 are now available in the Kiro IDE and CLI across both AWS GovCloud (US) regions (US-East and US-West). Opus 5.5 ships with a 1M-token context window and a 2.0x credit multiplier, down from Opus 5's 2.2x.





Access Claude Opus 5.5 in a GovCloud environment after updating to the latest IDE version.

📍 1. Update to the latest Kiro IDE or CLI version. 2. Restart the IDE or CLI. 3. Open the model selector and choose Claude Opus 5.5.

Access Claude Sonnet 5.5 in a GovCloud environment after updating to the latest IDE or CLI version.

📍 Update to the latest IDE or CLI version, then restart it, and select Claude Sonnet 5.5 from the model selector.

— Names regions, context window, and multiplier; navigation only.

Models: Claude Opus 5.5 now available i… | Models: Claude Sonnet 5.5 now available…

07 **Classic mode and inline autocomplete retired** BREAKING

67

Inline shell autocomplete and the CLI desktop app are retired and will no longer be available with the CLI 3.0 rollout. Classic mode is also retired; the `--classic` flag no longer starts the Classic experience — use `kiro-cli` (terminal UI) instead.

— Names retired flag and its replacement command. launch-20261009-25f11f16

THINNER COVERAGE BELOW

08 **Global hooks and subdirectory AGENTS.md scoping** NEW

51

Global hooks let the same automation be reused across workspaces. `AGENTS.md` files can now be placed in subdirectories, not just the repository root, to scope Kiro instructions to different parts of a project.

— Named config file and placement rule but little mechanism. launch-20261009-25f11f16

09 **Opt-in multi-agent workflows** NEW

40

Opt-in multi-agent workflows coordinate planning, implementation, and review agents in separate sessions, runnable both locally and in the cloud.

— Describes coordination but no command or config to invoke it. launch-20261009-25f11f16

└──▷ ALSO FROM THESE RELEASES

└──▷ BREAKING ON UPGRADE

! Inline shell autocomplete and the CLI desktop app are retired and will no longer be available with the CLI 3.0 rollout.

! Classic mode is retired; `--classic` flag no longer starts the Classic experience — use `kiro-cli` (terminal UI) instead.

! The `use_aws` tool is removed in V3; AWS CLI commands must now run through the shell tool, and existing `use_aws` allow/deny rules must be recreated as shell permission rules.

! Sessions created in V3 cannot be resumed in V2.

! Agent configuration migration does not automatically update ACP integrations — ACP clients must follow the ACP migration guide and use `kiro-cli acp --agent-engine=v3`.

WAS THIS USEFUL? 👍 👎

Diagram Design

Feedback ⓘ changes since 2026-09-29 CODE ↗

Diagram Design is a Claude Code skill that generates editorial HTML and SVG diagrams and imports draw.io or Mermaid sources.

Diagram Design: Adds `scripts/verify-service-blueprint.py`, a new verifier that checks layer order, visibility-line placement, stage-grid alignment, and cell census

└──▷ **TRY IT**

Rebuild all exploded axonometric examples from the shared projection model, or pass `--check` in CI to fail if any generated file is stale.

```
$ python3 scripts/build-exploded-examples.py --check
```

Adds the architecture delta diagram type.

WAS THIS USEFUL? 👍 👎

agents-cli

Feedback ⓘ v1.10.0 NOTES ↗ RANK

The CLI and skills that turn any coding assistant into an expert at creating, evaluating, and deploying AI agents on Google Cloud.

> agents-cli v1.10.0 makes Agent Identity the default runtime for new projects, lets extensions ship their own skills, adds gcloud-style label/secret/env flags to deploy, and tightens error handling with two breaking changes to deploy and run.

└──▷ **WHAT SHIPPED · 6 FEATURES** most completely described first ? what's the number?

01 **Extension-shipped skills** NEW 86

Extensions can now declare skills via a `skills:` field in `agents-cli-extension.yaml`; `agents-cli extension add` copies them to `<project>/.agents/skills/` or, with `--global`, to `~/.agents/skills/`. `agents-cli extension remove` gains a matching `--global` flag to uninstall user-scope extension skills, and `agents-cli extension update` now replaces skills with the new version's and lists any skills it removed.

Install an extension and have its bundled skills land in the user-wide skills directory for use across all projects.

```
$ agents-cli extension add my-extension --global
```

— Names config field, directories, flags across whole extension lifecycle v1.10.0

02 **Label, secret and env-var flags for deploy** NEW 77

`agents-cli deploy` gains `--remove-labels KEY,...`, `--clear-labels`, `--update-labels`, `--update-secrets`, and `--env-vars` flags for Agent Runtime and Cloud Run targets, matching `gcloud run deploy` conventions.

Remove stale labels from a Cloud Run deployment while adding a new one in a single deploy invocation.

```
$ agents-cli deploy --remove-labels env,version --update-labels
env=prod
```

— Lists exact new flags with a runnable deploy example v1.10.0

03 **Agent Identity as default runtime identity** NEW 72

`agents-cli-scaffold create` now emits `agent_runtime_identity_default: agent_identity` so new projects use Agent Identity by default; pass `--agent-runtime-identity-default service_account` to `scaffold create` or `scaffold enhance` to keep the previous service-account behaviour.

Scaffold a new project that uses a service account instead of the new Agent Identity default.

```
$ agents-cli scaffold create --agent-runtime-identity-default service_account
```

— Names config key and exact opt-out flag with runnable example v1.10.0

04 **Deploy rejects unsupported flags on some targets** BREAKING 60

`agents-cli deploy` now rejects `--iap`, `--port`, `--service-account`, and `--cluster-name` on targets that previously silently ignored them, so scripts passing these flags to unsupported targets will now fail.

— Names exact rejected flags but no migration command given v1.10.0

THINNER COVERAGE BELOW

05 **Non-zero exit code on agent run failure** BREAKING 53

`agents-cli run` now exits with a non-zero exit code and prints `Agent run failed: <reason>` when the ADK stream sends an error; scripts that previously treated all `run` exits as successful must be updated.

— Describes behaviour change but no concrete fix example v1.10.0

06 **Scaffold dependency and config file handling** IMPROVED 40

`agents-cli scaffold create` now shows dependency changes, consistent with `enhance` and `upgrade`, and `agents-cli scaffold enhance` now creates missing config and dependency files instead of skipping them.

— Describes behaviour change without naming specific files or keys v1.10.0

└──▷ BREAKING ON UPGRADE

! `agents-cli deploy` now rejects `--iap`, `--port`, `--service-account`, and `--cluster-name` on targets that previously silently ignored them — scripts passing these flags to unsupported targets will now fail.

! `agents-cli run` now exits with a non-zero exit code when the ADK stream sends an error; scripts that previously treated all `run` exits as successful must be updated.

WAS THIS USEFUL?  

GitHub Copilot CLI

Feedback ⓘ v1.0.96 NOTES ↗ RANK

GitHub Copilot CLI is a terminal-based coding agent that can plan, write, and test code in local repositories.

> GitHub Copilot CLI v1.0.96 adds a session-scoped directory access command for the sandbox, surfaces authentication sources and richer permission attribution in the UI, and smooths out model-ID and allow-all policy handling.

└──▷ **WHAT SHIPPED · 6 FEATURES** most completely described first ? what's the number?

01 **`/add-dir` grants sandbox directory access** NEW 80

Adds the `/add-dir` slash command, letting users grant sandbox access to a specified directory for the current session without restarting, e.g. `/add-dir ~/.config/myapp`.

Grant the agent sandbox access to a secrets or config directory mid-session without restarting.

```
$ /add-dir ~/.config/myapp
```

— Exact command with runnable example and session-scoped mechanism v1.0.96

02 **`/user list` shows authentication source** NEW 75

`/user list` now displays the authentication source for each signed-in account, e.g. running `/user list` shows which credential type backs each account.

Check which authentication source (PAT, OAuth, etc.) is backing each signed-in account.

```
$ /user list
```

— Named command with runnable example, limited further detail v1.0.96

03 **Case-insensitive model IDs in `/model` and `/config`** IMPROVED 60

`/model` and `/config` now accept model IDs case-insensitively and save the canonical ID internally.

— Named commands and behavior but no example v1.0.96

04 **`/allow-all` available during policy resolution** IMPROVED 60

`/allow-all` remains available during CLI startup even while enterprise policy is still resolving, rather than being unavailable until resolution completes.

— Describes before/after behavior, no example given v1.0.96

THINNER COVERAGE BELOW

05 **Timeline permission decision attribution** IMPROVED 55

Timeline now shows whether you, Assisted Permissions, policy, or unattended fallback made each permission decision, adding richer attribution for permission grants.

— Explains new categories shown but no navigation path v1.0.96

06 **Sandbox settings suggest secrets and masking hosts** IMPROVED 50

Interactive sandbox settings now suggest possible environment secrets and let users add masking hosts before saving the configuration.

— Describes behavior but no config key or command named v1.0.96

WAS THIS USEFUL? 👍 👎

Sourcegraph Amp

Feedback ⓘ You've Got Comments (2026-10-12)

BLOG ↗

Amp is Sourcegraph's agentic coding tool for the terminal and editor, running multi-step edits with subagents and shared team threads.

Amp agents can now subscribe to PR comments, reviews, and CI results, and spawn a new orb per GitHub issue or PR, managed via Settings → MCP & Integrations → GitHub.

Enables agents to subscribe to PR comments, reviews, and CI results for the latest commit, surfacing failing checks and reviewer questions in the same thread where Amp opened the PR.

Supports spawning a new orb for every new GitHub issue or PR in a repository, with optional filtering by label, so each issue or PR gets its own persistent thread.

Manages GitHub Webhook subscriptions — including editing, pausing, and viewing spawned orbs — via `Settings → MCP & Integrations → GitHub`.

NEW INTEGRATION GitHub Webhooks first-class integration

WAS THIS USEFUL? 👍 👎

Cotool

Feedback ⓘ v0.74.0 NOTES ↗ RANK

Cotool is an AI security operations platform for investigating alerts, enriching threat intelligence, and automating response workflows.

Redesigned UI with consistent light/dark mode and grouped chat summaries; new `all except` role permissions; improved agent chat queuing and Git-managed agent links.

Introduces a refreshed Cotool interface with a redesigned sidebar, unified page headers, consistent light and dark mode, and grouped chat work summaries.

Adds `all except` role permissions so administrators can grant access to every current and future resource except selected exclusions.

Queues follow-up messages in active chats instead of interrupting the agent's current work.

Adds direct links to source YAML and read-only synced triggers in alert routing for Git-managed agents.

NEW INTEGRATION Sourcegraph integration

NEW INTEGRATION Microsoft Sentinel data lake queries and table mapping

UPDATED INTEGRATION Splunk, Freshservice, and Vectra reliability improvements

WAS THIS USEFUL? 👍 👎

◆ Model Training & Fine-Tuning

Unsloth AI Unsloth

Feedback ⓘ v0.1.906-beta NOTES ↗

RANK

Unsloth runs and fine-tunes language and diffusion models locally through a Python library and interface.

> Unsloth v0.1.906-beta adds a built-in browser that local models can control, a new `DiffusionTrainer` for text-diffusion LLMs, LoRA+ training via `loraplus_lr_ratio`, and speculative decoding for MLX on Mac, alongside broader AMD GPU support, new image-generation options, and training-page controls.

└──▷ WHAT SHIPPED · 11 FEATURES most completely described first ? what's the number?

01 **DiffusionTrainer for text-diffusion LLMs** NEW 85

Adds `DiffusionTrainer` to train text-diffusion LLMs including `DiffusionGemma`, `LLaDA`, `LLaDA2`, Dream, Nemotron and `SDAR`.

Train a text-diffusion LLM such as LLaDA or DiffusionGemma using the new DiffusionTrainer instead of the standard causal trainer.

python

```python
from unsloth import FastLanguageModel
from unsloth.trainer import DiffusionTrainer,
DiffusionTrainingArguments

model, tokenizer =
FastLanguageModel.from_pretrained('google/diffusion-gemma-2b',
load_in_4bit=True)
model = FastLanguageModel.get_peft_model(model, r=16)

trainer = DiffusionTrainer(
    model=model,
    tokenizer=tokenizer,
    args=DiffusionTrainingArguments(
        output_dir='diffusion_output',
        per_device_train_batch_size=2,
        max_steps=100,
    ),
    train_dataset=dataset,
)
trainer.train()
```

— Names trainer class, six supported models, and ships a runnable example. v0.1.906-beta

02 **LoRA+ training via loraplus_lr_ratio** NEW 75

Adds `loraplus_lr_ratio` parameter to `UnslothTrainingArguments` to enable LoRA+ training, giving adapter layers a different learning-rate ratio from base layers to improve convergence on limited data.

Fine-tune a model with LoRA+ to give the adapter layers a higher learning rate than the base layers, improving convergence on limited data.

python

```python
from unsloth import FastLanguageModel, UnslothTrainer,
UnslothTrainingArguments

model, tokenizer =
FastLanguageModel.from_pretrained('unsloth/Llama-3.1-8B',
load_in_4bit=True)
model = FastLanguageModel.get_peft_model(model, r=16)

trainer = UnslothTrainer(
    model=model,
    tokenizer=tokenizer,
    args=UnslothTrainingArguments(
        loraplus_lr_ratio=16.0,
        output_dir='outputs',
        per_device_train_batch_size=2,
    ),
    train_dataset=dataset,
)
trainer.train()
```

— Named config param with runnable training example. v0.1.906-beta

03 **Built-in Unsloth Browser with permissions and live previews** NEW 75

Adds a built-in Unsloth Browser that local models can control — opening pages, clicking, typing, picking options and scrolling — accessible from the `+` menu in chat. Adds permission controls in `Settings > Browser` covering which sites models can open and their access to browsing history and downloads. Renders React code blocks (`JSX` and `TSX`) in replies as live previews in the Browser.

— Names exact UI paths and formats but no API or command. v0.1.906-beta

04 **Speculative decoding for MLX on Mac** NEW 75

Adds speculative decoding for MLX on Mac with automatic draft-depth tuning, supporting `MTP`, `DFlash`, `DSpark` and `EAGLE-3` drafters, for up to 2.8x faster replies.

— Names drafters and a concrete speedup but no exact invocation. v0.1.906-beta

05 **Qwen-Image-2.1-Turbo and image LoRA import/export** NEW

70

Adds `Qwen-Image-2.1-Turbo` (official 8-step version) runnable from the Images page with a 4.2 GB `Q4` GGUF, and enables export and import of image LoRAs as reusable files from user-defined folders.

— Names model, GGUF size/format and UI page, but no command. v0.1.906-beta

THINNER COVERAGE BELOW

06 **Train page VRAM offload control and adapter resume** NEW

55

Adds VRAM budget-based offload-layer control on the Train page (NVIDIA and AMD), and supports resuming LoRA training from a saved adapter.

— Names the UI page and capability but no exact workflow. v0.1.906-beta

07 **Faster oversized MoE GGUF inference** IMPROVED

50

Allows MoE GGUFs larger than available VRAM to run up to 3x faster on NVIDIA and AMD GPUs.

— Gives a concrete speedup number but no mechanism detail. v0.1.906-beta

08 **vLLM engine support on AMD GPUs** NEW

45

Adds vLLM engine support on AMD GPUs (Linux and WSL on Windows).

— Names platforms but no setup steps. v0.1.906-beta

09 **AsyncGRPOTrainer for simultaneous train and generate** NEW

40

Adds `AsyncGRPOTrainer` to train and generate simultaneously on a single GPU.

— Names the class but gives no usage example or further mechanism. v0.1.906-beta

10 **GRPO training on video prompts and tool calls** NEW

40

Enables GRPO training on video prompts and tool calls with TRL 1.15.

— Names TRL version but no further mechanism or example. v0.1.906-beta

11 **GPTQ fine-tuning with LoRA** NEW

30

Supports fine-tuning GPTQ models with LoRA.

— Bare statement with no mechanism or example. v0.1.906-beta

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

Vercel AI SDK

Feedback ⓘ ai@7.0.138 NOTES ↗ RANK

The Vercel AI SDK provides TypeScript APIs for model generation, structured output, tool use, and streaming application interfaces.

> Vercel AI SDK 7.0.138 adds a shared citation representation to generated text and streaming events, distinct from retrieved sources.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **Citation representation in text generation** NEW 55

Adds an optional shared citation representation to `generated-text` and `text-end` stream events, separate from retrieved sources. Citations are preserved in core result content and in UI message text parts, including validation and persistence.

— Names exact event types but no code example or API signature. ai@7.0.138

WAS THIS USEFUL? 👍 👎

CopilotKit

Feedback ⓘ v1.79.0 NOTES ↗ RANK

CopilotKit connects AI agents to application interfaces with shared state, tools, and generative UI components.

> CopilotKit v1.79.0 changes when the `headers` builder is evaluated and adds privacy-safe labeling for clicked controls in its learning system.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 Headers builder now evaluated per request send

BREAKING

84

The `headers` builder is now evaluated on every request send (for both sync and async builders) across React, Vue, and Angular, instead of on each provider render. This enables dynamic tokens without triggering thread-list reloads or realtime rejoins. This is a breaking change: apps that switch users must remount the provider (e.g. `key={userId}`) or call `setHeaders`, or the new user will continue seeing the previous user's threads.

— Names exact mechanism, APIs, and required migration step. v1.79.0

THINNER COVERAGE BELOW

02 Privacy-safe control labels in learning system

IMPROVED

28

Clicked controls are now labeled with a privacy-safe name in the learning system, improving interaction capture.

— Thin prose description, no named surface or mechanism. v1.79.0

└──▷ **BREAKING ON UPGRADE**

! The `headers` builder now runs on every request send instead of on each provider render; apps that switch users must remount the provider (e.g. `key={userId}`) or call `setHeaders`, or the new user will continue seeing the previous user's threads.

WAS THIS USEFUL? 👍 👎

Feynman

Feedback ⓘ v0.5.39 NOTES ↗ RANK

Feynman is a personal AI research assistant that searches papers and the web, builds literature reviews, and produces cited research reports.

> Feynman v0.5.38–v0.5.39 removed the confirmation step from `/deepresearch`, gave failed model calls actionable diagnostics, and hardened web search with self-hosted PDF parsing, provider fallback/timeouts, and improved multi-page fetch answers.

└──▷ **WHAT SHIPPED · 5 FEATURES** most completely described first ? what's the number?

01 Self-hosted PDF parser endpoint NEW

81

Adds `pdf.provider: "endpoint"` in `~/.feynman/agent/web-search.json` to route PDF fetches to a document parser the user runs themselves instead of the built-in handler.

Route all PDF fetches to your own on-premise document parser instead of the built-in handler.

```yaml
pdf.provider: "endpoint"
```

— Exact config key and file path, runnable example v0.5.39

02 Named causes and fixes for failed model calls IMPROVED

78

Failed model calls now surface a named cause and remediation step — wrong model (switch with `/model`), no credits, rate-limit/quota exhaustion, bad credentials (`feynman model login <provider>`), or temporary outage — instead of a bare error.

— Enumerates causes and exact remediation commands v0.5.38

03 Prompt-free /deepresearch with optional plan preview IMPROVED

75

`/deepresearch <topic>` now runs immediately by default without a yes confirmation prompt; adding `show me the plan first` as an opt-in displays the research plan before execution, useful for vetting scope on long parallel runs.

Check the research plan before Feynman proceeds — useful when you want to vet scope or adjust the topic before committing to a long parallel run.

```
$ feynman deepresearch "adversarial prompt injection, show me the plan first"
```

— Runnable command with named opt-in phrase v0.5.38

04 **Hardened auto web search provider fallback** IMPROVED

62

`auto` search mode now skips empty providers, times out stalled Perplexity requests after 30 seconds, and reports DuckDuckGo bot checks as a block instead of returning an empty result.

— Names mechanism and providers but no direct user action v0.5.39

THINNER COVERAGE BELOW

05 **Multi-URL fetch_content answers** IMPROVED

50

`fetch_content` answers now close with the `get_search_content` call that reads the stored page, and multi-URL answer mode surfaces every page's answer.

— Names tool calls but no concrete usage example v0.5.39

UPDATED INTEGRATION PubMed group/consortium author affiliation support

v0.5.35 NOTES ↗ STABLE

Adds `feynman -v`, canonical flag aliases (`--mode json`, `--cwd`), leaner subagent token usage, file-path previews for long reports, and slash-command completion in `/btw`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.5.35 https://github.com/Companion-Inc/feynman.git
# already have the repo? check out this version:
$ git checkout v0.5.35
```

Adds `feynman -v` to print the installed version, and aliases `--json`, `--work-dir`, and `--path` to `--mode json` and `--cwd <dir>` respectively.

Subagent tool definitions shrink from ~18,700 to ~3,300 characters per request, and long subagent reports now arrive as a preview plus file path instead of full inline text.

A wrong agent or model name now returns suggestions instead of a bare error.

> The `/btw` side-chat composer gains slash-command completion.

WAS THIS USEFUL? 👍 👎

CopilotKit OpenBot

Feedback ⓘ v0.2.0 NOTES ↗ RANK

OpenBot runs self-hosted AI coworkers that each get their own browser, files, and gated tools, with every action decided and recorded.

> OpenBot v0.2.0 adds Slack and GitHub account linking, shared Composio app accounts with per-Bot approval rules, and consolidates Agents, Team Bots and Bots into a unified `/bots` UI, alongside several breaking changes affecting migrations, handoffs, group conversations and the default example-coworker catalogue.

└──▷ **WHAT SHIPPED · 10 FEATURES** most completely described first ? what's the number?

01 **Linked Slack and GitHub accounts** NEW 85

A new Linked accounts section under Settings → Connected accounts lets each person connect their personal Slack account (via a one-time code sent to the OpenTag app in DM) and their GitHub account through a GitHub App sign-in flow. GitHub linking is enabled by setting `GITHUB_APP_CLIENT_ID` and `GITHUB_APP_CLIENT_SECRET`; both or neither must be set, and the server refuses to start if only one is set without a public URL configured.

Enable GitHub account linking so team members can connect their personal GitHub identity through the Settings → Connected accounts page.

env

```
GITHUB_APP_CLIENT_ID=Iv1.abc123def456
GITHUB_APP_CLIENT_SECRET=ghsec_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

— Names exact env vars, UI path and refusal condition v0.2.0

02 **Example coworkers no longer auto-loaded** BREAKING 85

Fresh deployments no longer auto-load the ten example coworkers; one can be restored by copying a profile such as `examples/fintech/catalog/risk_analyst.yaml` into `examples/fintech/agents/`. A previously registered protected coworker can be removed from an upgraded deployment via the `agent_profiles` table, e.g. `UPDATE agent_profiles SET deleted_at = NOW() WHERE name = 'risk_analyst';`.

Restore one of the ten example coworkers (e.g. Risk Analyst) that are no longer auto-loaded in a fresh deployment.

```
$ cp examples/fintech/catalog/risk_analyst.yaml
examples/fintech/agents/
```

Remove a previously registered protected coworker from an existing deployment after upgrading, using the `agent_profiles` table field documented in configuration.md.

```
$ UPDATE agent_profiles SET deleted_at = NOW() WHERE name =
'risk_analyst';
```

— Gives exact restore and removal commands v0.2.0

---

03 **Migration for shared account connections** BREAKING 75

Migration `0052_shared_brokered_accounts` copies every existing Composio connection into a new `brokered_connections` table backing the Shared apps feature. Rolling back to v0.1.2 is possible, but accounts connected after the upgrade become invisible to the older version.

— Names exact migration and table, rollback caveat v0.2.0

---

04 **Shared Composio app accounts** NEW 70

Administrators can connect a Composio app account once and share it across multiple Bots, each of which then acts as that shared identity. Per-Bot approval rules — owner only, named people or groups, or everyone — are checked on every call the Bot makes using the shared account.

— Describes mechanism and approval scopes, no exact UI path v0.2.0

---

05 **Unified /bots list and per-Bot pages** IMPROVED 65

Agents, Team Bots and Bots are consolidated into a single `/bots` list, with a dedicated page per Bot covering routines, skills, reachability, sharing, memory sources, background research and approval rules.

— Names consolidated endpoint and per-Bot page sections v0.2.0

---

06 **Handoffs record run origin** BREAKING 65

Handoffs now record what started their run. Older remote Bots' signed runs created before upgrading carry no such record for up to ten minutes after upgrade, and any calls they make to a Shared app during that window are refused.

— Explains transition window behaviour, no config surface v0.2.0

---

07 **Group conversations started via To: field**

BREAKING

65

The dedicated `/group/new` screen and separate group button are removed; it now redirects to the new-conversation screen, and group conversations are started by adding two or more Bots to the To: field of a new conversation.

— Names removed route and replacement navigation v0.2.0

08 **Folder write/delete permission prompt** NEW

60

A new desktop prompt appears every time a command can change or delete files, with no Always allow option for write/delete commands; read-only commands continue to honour existing Always allow folder grants.

— Describes exact prompt behaviour, no command given v0.2.0

THINNER COVERAGE BELOW

09 **Reorganized Settings and Admin approval pages** IMPROVED

55

GitHub event sources, per-Bot ask-before-acting rules, memories, devices, and recent deliveries move into Settings, while team approval settings and shared account requests move to Admin → Approvals.

— Lists moved items but no deeper mechanism v0.2.0

10 **Needs you section and Bots badge** NEW

55

A new Needs you section on the Bots list, plus a badge on the Bots sidebar entry, aggregates pending questions, approvals and stalled hand-offs across all Bots.

— Clear UI location but no deeper config detail v0.2.0

└──▷ **BREAKING ON UPGRADE**

! Migration `0052_shared_brokered_accounts` copies every Composio connection into a new `brokered_connections` table; rolling back to v0.1.2 is possible, but accounts connected after the upgrade are invisible to the older version.

! Handoffs now record what started their run; older remote Bots' signed runs carry no such record for up to ten minutes after upgrade, and calls they make to a Shared app in that window are refused.

! The `/group/new` screen is removed; it now redirects to the new-conversation screen (put multiple Bots in the To: field instead).

! The separate group button and `/group/new` dedicated flow are gone; group conversations are started by adding two or more Bots to the To: field of a new conversation.

WAS THIS USEFUL? 👍 👎

LangChain

Feedback ⓘ langchain==1.4.5 NOTES ↗ RANK

LangChain is an open-source framework that orchestrates applications powered by language models.

> LangChain's latest point releases add a pluggable summarizer to its summarization middleware and expose mid-conversation support as a discoverable model-profile property for Anthropic and core providers.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 Custom summarizer in SummarizationMiddleware NEW

75

`SummarizationMiddleware` now accepts a `summarizer` parameter, letting callers supply their own summarizer instead of the default to control how conversation history is compressed before being sent to the LLM.

Inject a custom summarizer into `SummarizationMiddleware` to control how conversation history is compressed before hitting the LLM.

python

```python
from langchain.middleware import SummarizationMiddleware
from langchain.chat_models import init_chat_model

summarizer = init_chat_model("openai:gpt-4o-mini")
middleware = SummarizationMiddleware(summarizer=summarizer)
```

— Names exact parameter and shows runnable code example. langchain==1.4.5

THINNER COVERAGE BELOW

02 Mid-conversation support flag in model profiles NEW

35

langchain-anthropic 1.7.6 declares mid-conversation support in model profiles for Anthropic (and OpenAI/core), allowing callers to programmatically discover whether a given model accepts mid-conversation turns.

— Describes capability but no concrete API or code shown. langchain-anthropic==1.7.6

WAS THIS USEFUL? 👍 👎

Agno (formerly Phidata) Feedback ⓘ v3.1.3 NOTES ↗

RANK

Agno is an agent framework and runtime that orchestrates and runs multi-agent systems.

> Agno v3.1.3 adds durable, resumable sessions and background runs for Claude and Codex external agents, alongside retries, structured output, file attachments, per-run metrics, and a new AnakinTools toolkit.

└──▷ **WHAT SHIPPED · 8 FEATURES** most completely described first ? what's the number?

01 **Durable, resumable sessions for external agents** NEW

97

`ClaudeAgent.continue_run` / `acontinue_run` replay steps via `continue_from`, `fork`, `input`, `stream=True`, and `background=True`, with each tool result checkpointed and listable via `GET /agents/{id}/runs/{run_id}/checkpoints`, forking the stored SDK transcript into a new sibling run. `BaseExternalAgent` gains `store_events` and `events_to_skip` so `/agents/{id}/runs/{run_id}/resume` can replay finished runs from the database after the event stream expires. The Claude SDK session id is stored on the Agno session and transcripts are mirrored into a new `agno_transcripts` table in `PostgresDb`, `AsyncPostgresDb`, `SqliteDb`, or `AsyncSqliteDb` so a later turn on another replica resumes from the database. `CodexAgent.compact` / `acompact` and Claude Code `/compact` resume a compacted session from its summary through the same store, and `GET /sessions` / `GET /sessions/{session_id}` responses now include an `sdk` identity field (`'codex'`, `'claude-agent-sdk'`) overlaid at read time with no migration required.

Resume a ClaudeAgent run from the last checkpoint to replay a specific tool-call step without rewriting the original run.

python

```python
from agno.agent.claude import ClaudeAgent

agent = ClaudeAgent()
# Fork from the last user turn of a finished run:
new_run = agent.continue_run(
    run_id='<run_id>',
    continue_from='last_user',
    fork=True,
    input='Re-run with a stricter policy',
    stream=True,
)
```

— Full mechanism, endpoints, tables and fields all named with a runnable example v3.1.3

02 **Structured output schema for ClaudeAgent and CodexAgent** NEW

88

Adds `output_schema` to `ClaudeAgent` and `CodexAgent`, accepting a Pydantic model or a JSON Schema dict per agent or per run, returning the parsed model or dict with `content_type` set; scalar or array roots are rejected before the run.

Require structured JSON output from a ClaudeAgent using a Pydantic model, with results in `content_type` `-tagged` form.

```python
from pydantic import BaseModel
from agno.agent.claude import ClaudeAgent

class ThreatReport(BaseModel):
    severity: str
    summary: str
    cves: list[str]

agent = ClaudeAgent(output_schema=ThreatReport)
result = agent.run('Summarise the top threats in this advisory')
print(result.content)  # parsed ThreatReport instance
```

— Named flag plus a runnable Pydantic example `v3.1.3`

---

03 **Retry support with exponential backoff for external agents** NEW 80

Adds `retries`, `delay_between_retries`, and `exponential_backoff` to `ClaudeAgent` and `CodexAgent`, retrying failed attempts under the same run id with a `type: 'retry'` streaming warning event between attempts, and adds a `metadata` field to `RunSchema` surfacing this as `metadata['attempts']` alongside `warnings`.

— Exact flags and schema field named, but no runnable example given `v3.1.3`

---

04 **Image and file attachments for external-agent runs** NEW 80

Adds image and file attachment support to `ClaudeAgent` and `CodexAgent` runs; attachments are staged under `<cwd>/.agno/uploads/<run_id>/`, recorded on the run's input, restored for later turns on any replica, and removed after the run unless `keep_uploads=True`.

— Exact path and flag named but no example of use `v3.1.3`

---

05 **Per-run metrics for external agents** NEW 74

Adds per-run metrics (`tokens`, `cache usage`, `cost`, `duration`, `time_to_first_token`) on `RunOutput.metrics` for `ClaudeAgent` and `CodexAgent`, with `metrics.details['model']` per model and accumulation into `session_metrics` visible in AgentOS at `/metrics`.

— Named fields and a UI path but no runnable example `v3.1.3`

THINNER COVERAGE BELOW

---

06 **AnakinTools toolkit** NEW 54

Adds `AnakinTools` toolkit exposing Anakin.io scrape, crawl, map, and search APIs as four flag-gated methods that poll asynchronous jobs until completion.

— Named toolkit and methods but no usage example v3.1.3

07 **Nested tool-call tracing via parent_tool_call_id** NEW

46

Adds `parent_tool_call_id` to `ToolExecution` and tool events for Claude subagent tool calls, enabling nested tool spans in tracing.

— Names the field but gives no usage detail v3.1.3

08 **Typed native options for ClaudeAgent and CodexAgent** IMPROVED

44

`ClaudeAgent` and `CodexAgent` can now accept typed native options (tools, permissions, MCP servers, thread and turn options) instead of untyped `_kwargs` dictionaries.

— States the change but no concrete type names or example v3.1.3

WAS THIS USEFUL?  

Nous Research Hermes

Feedback ⓘ v0.21.7 NOTES ↗

CODE ↗ RANK

A terminal-native, self-improving AI agent for autonomous coding and tasks, with persistent memory, agent-created skills, and a multi-platform messaging gateway.

> Hermes v0.21.7 changes how source installs receive updates—defaulting to tagged releases with a new channel switcher—while also improving update-check authentication and fixing reasoning replay on resumed chat sessions.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 Update channel switching for source installs

BREAKING

85

Source installs now track tagged releases by default instead of following every commit. A new `hermes update --set-channel main` command lets users opt back into following every commit on `main`, and a new *Update channel* switch in Settings ▸ About ▸ Updates toggles between Stable releases and Every commit, with the update dialog now showing the active channel.

Keep a source install following every commit on `main` rather than switching to the new stable-release default.

```
$ hermes update --set-channel main
```

Switch a source install's update channel to stable releases via the desktop UI.

📍 1. Open the Hermes desktop app. 2. Go to Settings › About › Updates. 3. Set the Update channel switch to Stable releases.

— Names exact flag, command and UI navigation path v0.21.7

THINNER COVERAGE BELOW

02 GitHub-authenticated stable update checks

IMPROVED

55

Stable update checks now honour `GITHUB_TOKEN` (or `gh auth login`) for authentication, and surface GitHub rate-limit errors as explicit rate-limit errors rather than opaque failures.

— Names env var and auth method but no full workflow v0.21.7

03 Signed reasoning replay on resumed chat sessions

IMPROVED

Resumed chat sessions via `hermes chat --resume` now replay signed reasoning/thinking content, and Bedrock Converse reasoning models are no longer rejected when resuming a session.

— Names the resume flag and affected provider, no deeper mechanism given v0.21.7

WAS THIS USEFUL? 👍 👎

AWS Strands Agents

Feedback ⓘ python/v1.60.0 NOTES ↗

RANK

Strands Agents is AWS's open-source SDK for production agents in Python and TypeScript, with a model-driven loop, tool integration, and deployment targets.

> Strands Agents v1.60.0 adds a built-in Python code-execution tool, auxiliary-model routing, strict tool-use mode for Bedrock TypeScript, and cross-agent stash sharing (a breaking change for isolated multi-agent state), alongside OTel trace propagation and TypeScript MCP/subagent parity work.

└──▷ **WHAT SHIPPED · 7 FEATURES** most completely described first ? what's the number?

01 **auxModel for auxiliary model calls** NEW 80

Adds `auxModel` / `aux_model` configuration allowing an agent to route sub-tasks to a separate, cheaper or specialised model while keeping the primary model for main reasoning, e.g. passing aux_model=BedrockModel(...) alongside the main `model=`.

Route lightweight sub-tasks (e.g. summarisation) to a cheaper auxiliary model while keeping the primary model for reasoning.

python

```python
from strands import Agent
from strands.models import BedrockModel

agent = Agent(
    model=BedrockModel(model_id='anthropic.claude-3-5-sonnet-20241022-v2:0'),
    aux_model=BedrockModel(model_id='anthropic.claude-3-haiku-20240307-v1:0')
)
agent('Summarise this 50-page report and then analyse the key risks')
```

— Exact config key with runnable example of dual-model setup python/v1.60.0

02 **python_repl vended tool** NEW 75

Adds `python_repl` as a vended tool in the Python SDK, giving agents a built-in Python code execution surface they can call directly, e.g. `from strands_tools import python_repl` then Agent(tools=[python_repl]).

Give a Python agent a live code-execution tool so it can run and iterate on Python snippets during a task.

```python
from strands import Agent
from strands_tools import python_repl

agent = Agent(tools=[python_repl])
agent('Write a function that finds all prime numbers up to 100,
then run it and show the result')
```

— Named tool with runnable import and usage example `python/v1.60.0`

THINNER COVERAGE BELOW

03 **strictTools mode on Bedrock TypeScript provider** NEW

50

Adds a `strictTools` config key on the Bedrock TypeScript provider to enable strict tool use mode.

— Named config key but no behavioural detail given `python/v1.60.0`

04 **SEP-2663 task execution in TypeScript MCP** NEW

45

Adds SEP-2663 task execution support via `mcp/ext-tasks` in the TypeScript MCP integration.

— Named extension surface but no usage shown `python/v1.60.0`

05 **Cross-agent stash sharing** BREAKING

40

Shares the stash across agents through a caller-scoped storage view, enabling cross-agent context sharing. This is a breaking change: multi-agent code that relied on isolated per-agent stash state will observe changed behavior after upgrade.

— Explains mechanism and breaking impact, no migration steps given `python/v1.60.0`

06 **OTel baggage propagation across agent calls** IMPROVED

30

Propagates OpenTelemetry baggage context across Python agent calls, enabling distributed trace correlation.

— Describes behaviour but no config or API named `python/v1.60.0`

07 **Subagent tool ported to TypeScript** NEW

25

Ports the vended subagent tool to TypeScript, bringing parity with the existing Python vended subagent tool.

— Bare statement of porting with no detail `python/v1.60.0`

└──▷ **BREAKING ON UPGRADE**

! The stash is now shared across agents through a caller-scoped storage view — multi-agent code that relied on isolated per-agent stash state will observe changed behavior after upgrade (feat(context)! commit b340edbbf).

WAS THIS USEFUL?  

◆ MCP Servers & Tooling

Composio

Feedback ⓘ @composio/slim@0.23.0 NOTES ↗ RANK

Composio connects AI agents to external tools and services with managed authentication and integrations.

> Composio's latest SDK and CLI releases add a human-in-the-loop pause-and-resume flow for session execution, overhaul the tool router session config types as a breaking change, forward user IDs through proxy execution for 2FA, and bring strict-mode and prompt-caching support to CLI providers.

└──▷ WHAT SHIPPED · 5 FEATURES most completely described first ? what's the number?

01 **Human-in-the-loop input-required flow in session execution**

NEW 90

Session execution now throws `ComposioToolInputRequiredError` from session.execute(), session.proxyExecute(), provider-wrapped session tools, and the `execute` / `proxyExecute` helpers when the API returns `result_type: 'input_required'`, exposing pending questions on `error.inputRequests` and continuation state on `error.requestState` (non-enumerable). The return value of session.execute() and the custom-tool `execute` helper now carries a `resultType` field (`'completed'` or `'failed'`) instead of relying on `error` to indicate failure, and `COMPOSIO_MULTI_EXECUTE_TOOL` batch execution throws the same error when the remote half of a mixed local-and-remote batch needs user input.

Catch a human-approval pause in an agentic loop so you can collect answers and resume rather than treating the pause as a tool failure.

```javascript
import { ComposioToolInputRequiredError } from '@composio/slim';

try {
  const result = await session.execute(toolCall);
  // result.resultType is 'completed' or 'failed'
  if (result.resultType === 'failed') {
    console.error('Tool failed:', result.error);
  }
} catch (err) {
  if (err instanceof ComposioToolInputRequiredError) {
    // Present err.inputRequests to the user, collect answers,
    // then resume using err.requestState as continuation state.
    console.log('Approval required:', err.inputRequests);
    const state = err.requestState; // non-enumerable, must read
directly
  }
}
```

Catch an approval-required pause in a session tool call and inspect the pending questions before deciding how to continue.

```javascript
import { ComposioToolInputRequiredError } from '@composio/core';

try {
  const result = await session.execute(toolCall);
  console.log('Result type:', result.resultType); // 'completed'
or 'failed'
} catch (err) {
  if (err instanceof ComposioToolInputRequiredError) {
    console.log('Approval needed:', err.inputRequests);
    // err.requestState holds continuation state for when
submission is supported
  } else {
    throw err;
  }
}
```

— Named classes, fields and a runnable code example, but no UI path `@composio/slim@0.23.0` `@composio/core@0.23.0`

02 Tool router session config types overhauled BREAKING

85

`ToolRouterSessionConfig` (from `session.config` and session.update()) and `ToolRouterSessionConfigHistoryConfig` (from session.listConfigHistory()) are now the client's own types, with `instant` always present as `false` or the policy object rather than optional; `toolkits` and each `tools` entry can now be a `{ require_approval } -only` object, `tags` can carry `require_approval`, and the config can carry `proxy_execute`. The SDK no longer validates `instant` at runtime or strips a `premium_usage` key, returning the config exactly as the API sends it; `ToolRouterSessionExecuteResponse` now requires `resultType` (a custom `ToolCallSession` must return it since `successful` is no longer derived from `error`), and the `ToolRouterInstantResponseSchema` / `ToolRouterInstantResponse` exports are removed in favor of `ToolRouterSessionConfig['instant']`.

— Exact type and field changes with migration path, no runnable command

`@composio/slim@0.23.0` `@composio/core@0.23.0`

---

03 **User ID forwarding on proxy execution for 2FA** NEW

75

tools.proxyExecute() now accepts a `userId` field, sent as `user_id` to enforce 2FA checks on the connected account; Python's tools.proxy() takes `user_id` too, and the CLI forwards `user_id` on tool proxy execute calls.

Pass a user ID to proxyExecute so 2FA-enabled projects validate the action against the correct connected account.

javascript

```
await tools.proxyExecute({
  toolName: 'GMAIL_SEND_EMAIL',
  input: { to: 'alice@example.com', subject: 'Hello', body: 'Hi!'
},
  userId: 'usr_abc123'
});
```

— Named field with runnable example across SDKs and CLI `@composio/cli@0.4.3-beta.419` `@composio/core@0.23.0`

THINNER COVERAGE BELOW

---

04 **Expanded request telemetry headers** IMPROVED

40

Every API request now sends product identity, product version, language, and runtime version; installed core and slim packages are identified separately while preserving existing telemetry headers.

— Describes what changed but no header names or config `@composio/slim@0.23.0`

---

05 **Strict-mode and prompt-caching for CLI providers** NEW

The CLI adds strict-mode support to the OpenAI Agents Python provider and prompt-caching support to the Anthropic Python provider.

— Named providers but no flags or config shown `@composio/cli@0.4.3-beta.419`

└──▷ **BREAKING ON UPGRADE**

! `ToolRouterSessionConfig` (returned by `session.config` and session.update()) and `ToolRouterSessionConfigHistoryConfig` (returned by session.listConfigHistory()) are now the client's own config types; `instant` is always present as `false` or the policy object instead of being optional.

! In `ToolRouterSessionConfig` and `ToolRouterSessionConfigHistoryConfig`, `toolkits` and each `tools` entry can be a `{ require_approval }-only` object, so checking only for `enabled` or `disabled` no longer covers every case; `tags` can carry `require_approval` and the config can carry `proxy_execute`.

! The session config is returned exactly as the API sends it — the SDK no longer validates `instant` at runtime or removes a `premium_usage` key.

! `ToolRouterSessionExecuteResponse` now requires `resultType`; a custom `ToolCallSession` passed to a provider must return it from execute() because `successful` is no longer derived from `error`.

! `ToolRouterInstantResponseSchema` and `ToolRouterInstantResponse` exports are removed; use `ToolRouterSessionConfig['instant']` for the Instant policy the API returns.

! `ToolRouterSessionConfig` (`session.config`, result of session.update()) and `ToolRouterSessionConfigHistoryConfig` (session.listConfigHistory()) are now the client's own config types; `instant` is always present (`false` or the policy object) instead of optional.

! In those config types, `toolkits` and each `tools` entry can be a `{ require_approval }-only` object, so a check for `enabled` or `disabled` no longer covers every case; `tags` can carry `require_approval` and the config can carry `proxy_execute`.

! The session config is returned as the API sent it; the SDK no longer validates `instant` at runtime or removes a `premium_usage` key.

! `ToolRouterSessionExecuteResponse` now requires `resultType`; a custom `ToolCallSession` passed to a provider must return it from execute(), because `successful` is no longer derived from `error`.

! The `ToolRouterInstantResponseSchema` and `ToolRouterInstantResponse` exports are removed; use `ToolRouterSessionConfig['instant']` for the Instant policy instead.

WAS THIS USEFUL? 👍 👎

gram

Feedback ⓘ server@2.22.0 NOTES ↗ RANK

Gram provides MCP servers and AI-agent tools from APIs with access controls and observability.

> Gram's biggest changes this window are breaking: federated sign-in now requires a per-client redirect URI and a new identity-chaining audit trail, and hosted MCP traffic analytics split into two non-summable counters. Alongside these, Gram shipped Platform MCP tools for managing Explore widgets and dashboards, an Okta sign-in setup flow backed by new readiness reporting, a built-in MCP & Tools dashboard, and stacked bar/area chart types.

└──▷ **WHAT SHIPPED · 7 FEATURES** most completely described first ? what's the number?

01 Per-client federated redirect URIs and identity-chaining audit

90

BREAKING

Adds a per-client redirect URI `/mcp/idp_callback/{clientID}` for federated sign-ins, binding each authorization response to its issuer (RFC 9700 §4.4.2) and supporting identity providers that omit the RFC 9207 `iss` parameter (e.g. Okta org authorization servers). Existing federated IdP apps must register this new URI, now reported as `federated_callback_url` on organization-owned remote session clients; the old shared `/mcp/idp_callback` now serves only WorkOS and misrouted federated callbacks are rejected. The identity-chaining binding lifecycle on remote session clients is now audited via `remote-session-client:enable-identity-chaining`, `remote-session-client:update-identity-chaining-scopes`, and `remote-session-client:disable-identity-chaining` entries (keyed by `client_id`), each carrying binding, issuers, resource, generation, state, and scopes. Identity chaining also no longer sends OpenID Connect scopes (`openid`, `profile`, `email`, `offline_access`, `phone`, `address`) in the ID-JAG token exchange, so a client whose scope came from `scopes_supported` discovery now chains with only its resource scopes or no `scope` parameter when none remain.

— Exact endpoint, migration requirement, audit event names given server@2.22.0

02 Okta sign-in setup and identity-chaining readiness NEW

80

`get_xaa_readiness` adds an `okta_sign_in` field reporting the AI agent's linked-app status, its active public key (JWK, public members only) to paste into Okta, the sign-in redirect URI, which issuers trust the agent, the next sign-in step, and the first unconfirmed Okta checklist step. `get_mcp_readiness` and `get_my_mcp_connection_status` now report `ready` with

`identity_chaining_active` evidence when a usable chained credential exists, or `identity_chaining_configured` otherwise, replacing `upstream_authorization_required` / `no_valid_authorization`. Okta identity provider connections now also expose `remote_session_issuer_id`. The dashboard adds a Trust Okta sign-in action per sign-in issuer row, an Add sign-in issuer action that creates a dedicated sign-in issuer per vendor (checked against every organization sign-in issuer slug), and a Set up Okta sign-in action that automatically creates a new `okta-sign-in` sign-in issuer (or the next free numbered slug) when no issuer yet trusts Okta sign-in, signing with private_key_jwt.

— Backend fields and UI actions both named across two releases `server@2.22.0` `dashboard@0.136.0`

---

03 **Platform MCP tools for Explore widgets and dashboards** NEW

75

Adds Platform MCP tools for agent-driven management of Explore widgets: `list_widgets`, `get_widget`, `save_widget`, `duplicate_widget`, and `delete_widget`, covering query, chart, validity, and dashboard membership with preview-then-confirm versioning so concurrent edits are never overwritten. Also adds `list_dashboards`, `get_dashboard`, `save_dashboard`, `duplicate_dashboard`, `delete_dashboard`, `edit_dashboard_layout`, and `save_dashboard_filters` for full dashboard management with the same permissions, grid rules, and audit trail as the Dashboards UI.

— Every tool named but no usage example given `server@2.22.0`

---

04 **MCP server identity on traffic logs and metrics** BREAKING

60

Hosted MCP server traffic now records MCP server and endpoint identity on tool call and resource read logs, usage events, analytics events, request logs, and tool call and request duration metrics; it emits `mcp_server_tools_list` alongside the legacy `mcp_server_count` for compatibility. Analytics consumers must pick one of `mcp_server_tools_list` or `mcp_server_count` since summing both double-counts traffic.

— Named metric fields with explicit migration caveat `server@2.22.0`

THINNER COVERAGE BELOW

---

05 **Built-in MCP & Tools dashboard** NEW

50

Adds a built-in MCP & Tools dashboard at `/dashboards/builtin/<slug>`, showing agent call frequency, failure rates, and callers; it can be duplicated into a fully editable project dashboard.

— Named path but no further mechanism detail `server@2.22.0`

---

06 **Stacked bar and area chart types** NEW 45

Adds Stacked bar and Stacked area chart types to Explore and widgets, showing composition of totals with a Group-by dimension; excess series are collapsed into an Other band that preserves the real total height.

— Describes behavior but no usage walkthrough `server@2.22.0`

---

07 **Removal of risk.getPolicyStatus API** BREAKING 35

The `risk.getPolicyStatus` API and the View Progress panel are removed.

— Named endpoint removed but no replacement or reason given `dashboard@0.136.0`

└──▷ **BREAKING ON UPGRADE**

! Existing federated identity provider apps must register the new per-client redirect URI `/mcp/idp_callback/{clientID}` ; the old shared `/mcp/idp_callback` now serves only WorkOS, and misrouted federated callbacks are rejected.

! Analytics consumers of hosted tools-list events must select either `mcp_server_tools_list` or `mcp_server_count` — summing both double-counts traffic.

! Identity chaining no longer sends OpenID Connect scopes (`openid` , `profile` , `email` , `offline_access` , `phone` , `address`) in the ID-JAG token exchange; a client whose scope came from `scopes_supported` discovery now chains with only its resource scopes or no `scope` parameter when none remain.

! The `risk.getPolicyStatus` API and the View Progress panel are removed.

WAS THIS USEFUL?  

DEPLOY

◆ Agentic Infrastructure

Hugging Face

Feedback ⓘ Contributor permission for access tokens

NOTES ↗ RANK

Hugging Face is an open platform for building, sharing, and deploying machine-learning models, datasets, and applications.

> Hugging Face introduced a new contributor permission for access tokens, enabling least-privilege write access scoped only to self-created repositories.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **Contributor permission for access tokens** NEW 85

A new `contributor` permission is available for fine-grained access tokens and service account tokens, letting a token create repositories and get full write access only to those repositories it created — without write access to any other repository in the namespace. This supports safer CI/CD pipelines, training jobs, and agents that publish checkpoints or datasets without needing broad write access across an entire namespace or organization.

Provision a least-privilege token for a training job that only needs to publish new model checkpoints — not touch any existing repos.

> 📍 In the Hugging Face settings, go to Access Tokens › Create new token, select Fine-grained, and under Permissions choose contributor. Assign it to the target organization. The token can then create and write to its own repos but cannot modify any pre-existing repositories.

— Names exact permission and gives full setup steps.

Contributor permission for access tokens

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.150 NOTES ↗ RANK

HyperFrames is an open-source HTML-to-video renderer that runs in AI-agent workflows.

> HyperFrames added a programmatic API entry point for transcription and background removal, plus composition-wide audio effects chaining, with two breaking changes to existing audio and frame-injection behavior.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 **hyperframes/api entry point for transcribe and removeBackground**

NEW 70

A new `hyperframes/api` entry point exports `transcribe` and `removeBackground` as importable functions with typed results and typed errors, removing the need to spawn the CLI as a child process.

— Names exact module and functions, no usage example given v0.8.149

02 **Composition-wide audio effects chain** BREAKING 65

Setting `data-fx-chain` on the top-level composition root now supports a composition-wide audio effects chain, applied after all clip and group effects. This is a breaking change: a top-level root's `data-fx-chain` now processes the whole audio mix, so a previously unused root chain should be removed if unchanged audio is desired.

— Names exact attribute and behavior change, no example shown v0.8.150

03 **Expiring temporary URLs in injectVideoFramesBatch** BREAKING 60

For engine API users, `injectVideoFramesBatch` now assigns temporary image URLs that expire after decoding; callers should use the original frame source when serializing or reloading the DOM.

— Names exact function and required workaround v0.8.150

└──▷ **BREAKING ON UPGRADE**

! A top-level composition root's `data-fx-chain` now processes the whole audio mix — remove a previously unused root chain if you want unchanged audio.

! For engine API users, `injectVideoFramesBatch` now assigns temporary image URLs that expire after decoding — use the original frame source when serializing or reloading the DOM.

WAS THIS USEFUL?  

Supermemory

Feedback ⓘ server-v0.0.9 NOTES ↗ RANK

Supermemory is a memory and context engine and app for AI applications, runnable fully locally or hosted, exposed as a memory API.

> Supermemory's server v0.0.9 release adds API v5 support and prebuilt Docker images, alongside several breaking changes to authentication and workflow configuration.

└──▷ **WHAT SHIPPED · 5 FEATURES** most completely described first ? what's the number?

01 **WORKFLOW_ENGINE=direct configuration removed** BREAKING

60

The `WORKFLOW_ENGINE=direct` configuration value is no longer supported and will break any setup using it.

– Names exact config key but no replacement guidance server-v0.0.9

THINNER COVERAGE BELOW

02 **API v5 support added** NEW

55

Supermemory now supports API v5, with a dedicated upgrade guide published at supermemory.ai/changelog/supermemory-api-v5/.

– Names version and guide link but no mechanism detail server-v0.0.9

03 **Prebuilt Docker images for amd64/arm64** NEW

50

Supermemory server v0.0.9 ships prebuilt Docker images for amd64 and arm64 architectures.

– Names architectures but no pull command or registry path server-v0.0.9

04 **API key now required for localhost requests** BREAKING

50

Localhost requests to the Supermemory server now require an API key; previously keyless localhost access is broken by this change.

– Clear before/after but no migration steps given server-v0.0.9

05 Improved uploads, namespace, search and CLI IMPROVED

35

This release improves local uploads, namespace operations, search scoping, and CLI commands.

— Lists areas touched without specifics on what changed `server-v0.0.9`

└──▷ **BREAKING ON UPGRADE**

! Localhost requests now require an API key; previously keyless localhost access is broken.

! The `WORKFLOW_ENGINE=direct` configuration value is no longer supported and will break any setup using it.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

OpenWhispr

Feedback ⓘ v1.10.3 NOTES ↗

OpenWhispr is a desktop dictation and meeting-transcription app that supports local speech models and user-configured cloud providers.

Adds note templates, docked action chat, real table editing, echo cancellation, local model persistence, and a folder picker in meeting notifications.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.10.3
https://github.com/OpenWhispr/openwhispr.git
# already have the repo? check out this version:
$ git checkout v1.10.3
```

Adds note templates that shape the AI Summary, and note actions that answer in a chat docked beside the note instead of overwriting your summary.

Adds real table support in notes — tables remain editable as tables, and a formatting toolbar appears on text selection.

Adds echo cancellation for meetings.

Adds an option to keep the local model loaded in memory.

Adds a folder picker in the meeting notification.

+4 more

NEW INTEGRATION Gmail connector

NEW SUPPORTED PLATFORM Apple sign-in on Windows and Linux

NEW SUPPORTED PLATFORM Live speaker names on Linux

└──▷ BREAKING ON UPGRADE

! The Windows portable `.exe` is discontinued — use the installer instead.

WAS THIS USEFUL? 👍 👎

◆ Inference Providers & GPU Clouds

Modal

Feedback ⓘ announced · Oct 12, 2026 BLOG ↗ RANK

Modal is a cloud infrastructure for teams that develop, train, and serve AI applications at scale.

> Modal launched Endpoint Candidates, letting teams stage, traffic-mirror, benchmark, and promote alternative inference deployments for Dedicated Endpoints directly from the dashboard.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **Endpoint Candidates for Dedicated Endpoints** NEW 80

Modal added Endpoint Candidates, which let you stage alternative model deployments that mirror a configurable percentage of live production traffic without building your own mirroring infrastructure. Each Candidate gets its own URL for running external evaluations independently of production, and dashboard controls let you set the traffic-sample percentage, inspect latency and throughput metrics, run performance benchmarks, and promote a Candidate to production at the existing Endpoint URL while other Candidates remain active for parallel testing. Candidate model weights can be sourced from Modal's open-model catalog, a Hugging Face repo, or a Modal Volume.

When you want to test a new model checkpoint against live traffic before committing to a full rollout, create a Candidate from an existing Dedicated Endpoint.

📍 In the Modal dashboard, open your Dedicated Endpoint, select `Create candidate`, choose a model from the catalog or supply weights via a Hugging Face repo or Modal Volume, then set a traffic sample percentage to begin mirroring production requests.

When benchmark results and evaluation scores look good, promote the Candidate to production without changing your Endpoint URL.

📍 In the Modal dashboard, open the Candidate's metrics view, review latency and throughput under load, then select `Promote` to start a production rollout at the existing Endpoint URL — remaining Candidates stay active for continued parallel testing.

— Full mechanism and named sources described, but only UI steps, no API/flag.

`launch-20261012-f521e06d`

WAS THIS USEFUL? 👍 👎

Baseten

Feedback ⓘ Rolling deployments enabled by default NOTES ↗

RANK

Baseten deploys and serves machine learning models as autoscaling inference endpoints, with a model API catalog and dedicated GPU deployments.

> Baseten now enables rolling deployments by default for newly created model environments, shifting traffic incrementally during promotions.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **Rolling deployments on by default** IMPROVED 40

Newly created model environments now default to rolling deployments, which replace replicas incrementally and shift traffic in steps during promotions, rather than switching all at once.

– Explains behavior change but no config key or opt-out flag named

Rolling deployments enabled by default

WAS THIS USEFUL? 👍 👎

◆ Inference Serving & Optimization

vLLM

Feedback ⓘ announced · Oct 12, 2026 BLOG ↗ RANK

The vLLM inference engine runs large language models with high-throughput, memory-efficient GPU execution.

> vLLM shipped bi-directional KV cache transfer for disaggregated prefill/decode serving that cuts TTFT by up to 3.3x on multi-turn agentic workloads, and added day-one support for NVIDIA's Vera Rubin NVL72 platform with Rubin-tuned kernels and locality-aware MoE.

└──▷ WHAT SHIPPED · 2 FEATURES most completely described first ? what's the number?

01 Bi-directional KV cache transfer for multi-turn agents NEW

96

Adds bi-directional KV cache transfer between prefill and decode nodes in disaggregated serving, cutting time-to-first-token by up to 3.3x on multi-turn agentic workloads. Adds `decoder_kv_blocks_ttl` config (default 480 seconds) controlling how long a decode node retains KV blocks for reuse by the prefill node on the next turn, and `kv_recompute_threshold` config (default 64 tokens) controlling when the prefill node pulls KV from the decode node versus recomputing locally, tunable by model, KV size, and interconnect. Extends the NIXL KV connector to support D→P (decode-to-prefill) NIXL READ transfers, adds a KV-cache-aware router (example proxy on vLLM GitHub) that tracks `conversation_id` to `kv_transfer_params` mappings, and validates the capability as accelerator-agnostic on AWS Trainium and Amazon EC2 P5en GPU clusters with AWS EFA.

Tune KV block retention and the recompute threshold on a 1P1D deployment to maximize TTFT savings for long multi-turn agent conversations where turns arrive within ~8 minutes of each other.

yaml

```yaml
decoder_kv_blocks_ttl: 480
kv_recompute_threshold: 128
```

Disable prefix caching on the prefill node to force bi-directional KV transfer to carry the full conversational context — useful when the router may select a different prefill node across turns.

```
$ vllm serve <model> --no-enable-prefix-caching
```

— Names config keys, defaults, and gives tuning/command examples.

launch-20261012-4969f202

02 **NVIDIA Vera Rubin NVL72 platform support** NEW 85

Adds NVIDIA Vera Rubin NVL72 platform support via daily nightly Docker images tagged `vllm/vllm-openai:cu134-nightly`, built with CUDA 13.4 and PyTorch 2.15. Integrates Rubin-tuned attention, GEMM, and MoE kernels through FlashInfer 0.7.0, including dense NVFP4/MXFP4 GEMM, NVFP4 MoE, FP8 attention, and FP8 MSA prefill kernels, and introduces locality-aware MoE using CUDA 13.4 locality domains (with `cudaDevSmResourceGroupBackfill` backfill mode) to split MoE weights across HBM domains, delivering ~1.2x speedups in small-token decode on Rubin. Enables serving of DeepSeek, Kimi (Moonshot AI), GLM (Z.ai), and MiniMax models on this hardware out of the box.

— Names docker tag and kernel stack but no run command. launch-20261009-aa3e58e1

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

ralforion OrionBelt Ontology Builder

Feedback ⓘ

v1.29.0 NOTES ↗ RANK

Browser-based ontology workbench for OWL ontologies and SKOS vocabularies. Streamlit + rdflib, no Java, no Protégé.

> OrionBelt Ontology Builder v1.29.0 adds a 22-rule Model Quality analyser and full SHACL validation via pySHACL, reorganizing the Validation page into a four-section Validation & Reasoning workspace.

└──▷ WHAT SHIPPED · 4 FEATURES most completely described first ? what's the number?

01 SHACL validation section powered by pySHACL NEW 96

A new SHACL section (powered by `pySHACL` 0.40+) checks data against shapes found in the ontology, pasted/uploaded shapes, or built-in sets (`OrionBelt quality policy` , SKOS integrity , and SKOS best-practice shapes from the `skohub` project under `Apache-2.0`). SHACL shapes can also be generated directly from OWL axioms (cardinalities, `someValuesFrom` , `allValuesFrom` , `hasValue` , functional properties and ranges) to check individuals, derived fresh per check (never stored, but viewable and savable), with an optional `OWL RL` inference pre-pass (on by default, toggleable) run before generated-shape checks so entailed values aren't falsely flagged as missing. Results can be exported as a SHACL report or as findings, the ontology export can write the ontology, its shapes, or both, and all checks enforce `SERVICE` query refusal plus per-check process isolation with a time limit; pySHACL 0.40+ is now a required dependency, and installations without it lack SHACL functionality.

Run built-in SKOS integrity checks against your ontology data without permanently adding shapes to the file.

📍 In Validation & Reasoning › SHACL, select the SKOS integrity built-in set, then click Check (not Copy into ontology) to run a single check without saving.

Generate OWL-axiom-derived SHACL shapes and run them with OWL RL inference to catch constraint violations implied by the model, not just asserted data.

📍 In Validation & Reasoning › SHACL, choose Shapes generated from the model , confirm OWL RL inference first is enabled, click Check , then use Export as SHACL report to save the results.

— Names dependency, built-in sets, axiom types and export formats v1.29.0

02 Model Quality analyser with 22 advisory rules NEW

88

A new Quality section runs 22 advisory rules grouped into metadata, naming, structure and SKOS categories; findings explain themselves, suggest a fix, and link to the affected resource. Profile and rule settings let you start from `General ontology`, `SKOS vocabulary` or `Strict documentation` presets, toggle individual rules on/off, adjust severity, and store settings per ontology in the browser (web) or config file (desktop), saving to and loading from a `JSON` file. The Findings table supports filter-by-severity, rule, kind, or text, with per-row detail, evidence, Open buttons for related resources, suppression with a reason, and export as `JSON` (with run details) or `CSV`.

Tighten quality rules for a SKOS vocabulary and export the tuned settings to share across projects.

> 📍 In Validation & Reasoning › Quality, select the `SKOS vocabulary` profile, toggle individual rules on/off and adjust severities, then use `Save settings to JSON file` to export the configuration.

— Names presets, storage format and export options; UI-only steps `v1.29.0`

THINNER COVERAGE BELOW

03 **Validation & Reasoning page redesign** IMPROVED 37

The Validation page is renamed to `Validation & Reasoning` and reorganised into four sections: Validation, Quality, SHACL and Reasoning.

— Names the page and its four sections only `v1.29.0`

04 **Graph label word-wrapping** IMPROVED 30

Annotation and literal node labels in the graph now wrap onto two lines at word boundaries instead of cutting at 30 characters.

— Describes a small visual change with no action steps `v1.29.0`

└──▷ **BREAKING ON UPGRADE**

! pySHACL 0.40 or later is now a required dependency; installations without it will lack SHACL check functionality.

WAS THIS USEFUL? 👍 👎

Docling

Feedback ⓘ v2.138.0 NOTES ↗ RANK

Docling parses PDFs, Office files, and HTML into structured documents with layout, tables, and reading order preserved for downstream RAG pipelines.

> Docling v2.138.0 adds support for extracting pictures embedded in Apple Numbers sheets via the iWork parser.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **Picture extraction from Numbers sheets** NEW 35

The iWork parser now supports reading pictures placed on a Numbers sheet, extending Docling's structured parsing to images embedded in Apple Numbers files.

— Names the parser and file type but no API or usage detail. v2.138.0

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

Braintrust

Feedback ⓘ changelog · Oct 12, 2026 CHANGELOG ↗

RANK

Braintrust provides evaluation, tracing, and improvement workflows for AI applications.

> Braintrust's latest release adds cooperative cancellation for evaluation tasks, a span-customizer hook for redacting or tagging span data, broader tracing coverage for the Agents/Responses APIs, MCP connectors and Roast workflows, alongside several breaking changes to the .NET SDK tracing builders and the `bt` tracing plugin's span fields.

└──▷ WHAT SHIPPED · 9 FEATURES most completely described first ? what's the number?

01 Cooperative cancellation for task hooks NEW

80

Adds `hooks.cancel_event` (a `threading.Event`) in the Python SDK so task functions can cooperatively check for cancellation and exit early; `timeout` also sets this event automatically when it fires.

Use `hooks.cancel_event` to stop a long-running evaluation task early when a timeout fires or the evaluation is cancelled.

```python
def my_task(input, hooks):
    for chunk in process_chunks(input):
        if hooks.cancel_event.is_set():
            return {'status': 'cancelled'}
        process(chunk)
    return {'status': 'done'}
```

— Mechanism and runnable code example both given changelog-20261012-f4825a57

02 .NET SDK tracing builder API overhaul BREAKING

80

Adds UseBraintrustLLMTracing() on `ChatClientBuilder` and moves UseBraintrustFunctionTracing() from `ChatClientBuilder` to `AIAgentBuilder`. The previous ChatClientBuilder.UseBraintrustTracing() method is removed; existing code must switch to UseBraintrustLLMTracing() on `ChatClientBuilder` or UseBraintrustFunctionTracing() on `AIAgentBuilder`.

— Exact method names and clear migration path, no code sample changelog-20261012-f4825a57

03 Span customizers for Braintrust.init NEW

75

Adds a `span_customizers` argument to `Braintrust.init` that lets callers redact sensitive values or add tags and metadata to each span before the SDK exports it.

Instrument `Braintrust.init` with a span customizer to redact API keys from span metadata before export.

python

```
Braintrust.init(
    project='my-agent',
    span_customizers=[
        lambda span: span.set_attribute('metadata.api_key',
'[REDACTED]')
        if 'api_key' in span.attributes.get('metadata', {})
        else span
    ]
)
```

— Named flag with a concrete redaction code example changelog-20261012-f4825a57

04 **bt tracing plugin pi 1.0 span field changes** BREAKING 75

In the `bt` tracing plugin (pi 1.0), `cost` and `metadata.provider_request` on pi model call spans are replaced by top-level fields, and `skill` loads now use `tool_name: 'read'` and `tool_kind: 'skill'`; existing queries referencing the earlier fields must be updated.

— Exact field names and migration note given changelog-20261012-f4825a57

05 **Tracing for Roast workflows** NEW 70

Adds tracing support for Roast workflows (`roast-ai` v1.0.0 or later): a span is created for each workflow run and each cog, with LLM calls from `chat` cogs nested under their cog span.

— Version requirement and span structure named, no example changelog-20261012-f4825a57

06 **Agent run tracing improvements** IMPROVED 65

Agent run spans now capture more detail: Responses API spans record `access_programs` in metadata, Agents API sessions trace computer use calls (capturing screenshots as attachments), browser authentication requests, and computer use approval items; MCP connector tool calls now appear as named child tool spans (named after the MCP tool, including tool input) instead of the previous generic `mcp` spans; and agent run spans now include sub-agent token usage aggregated across all models, plus `total_cost_usd` metadata.

— Named fields and trace types but no usage example Before/after described, no config or code shown Names the new metadata field but gives no usage detail

changelog-20261012-f4825a57

07 **External-ID-only submission result matching** IMPROVED

60

process_submission_result() in the Python SDK and processSubmissionResult() in the JS SDK now support matching via `external_id` / `externalId` without requiring `run_id` / `runId`.

— Exact function names in both SDKs, no usage example changelog-20261012-f4825a57

THINNER COVERAGE BELOW

08 **Batched evaluation submission via WorkflowBatchTask** NEW

50

A new `WorkflowBatchTask` lets evaluation cases for workflows be submitted in batches to reduce per-case provider round-trips, as shown by the usage example wrapping a workflow task in `Eval`.

Submit workflow evaluation cases in batches using the new `WorkflowBatchTask` to reduce per-case provider round-trips.

python

```python
from braintrust import WorkflowBatchTask, Eval

Eval(
    'my-workflow-project',
    data=cases,
    task=WorkflowBatchTask(my_workflow),
    scores=[...],
)
```

— Known only from an example, not described in changelog text changelog-20261012-f4825a57

09 **Removal of bt_daemon_version from span metadata** DEPRECATED

30

Spans no longer include `metadata.bt_daemon_version`.

— Bare removal notice with no replacement or rationale changelog-20261012-f4825a57

└──▷ **BREAKING ON UPGRADE**

! In the .NET SDK, ChatClientBuilder.UseBraintrustTracing() has been removed; replace it with UseBraintrustLLMTracing() on `ChatClientBuilder`. UseBraintrustFunctionTracing() has moved from `ChatClientBuilder` to `AIAgentBuilder`.

! In the `bt` tracing plugin (pi 1.0), `cost` and `metadata.provider_request` on pi model call spans are replaced by top-level fields; `skill` loads now use `tool_name: 'read'` and `tool_kind: 'skill'`. Update queries that reference the earlier fields.

! Spans no longer include `metadata.bt_daemon_version`.

WAS THIS USEFUL? 👍 👎

Langfuse

Feedback ⓘ v4.57.0 NOTES ↗ RANK

Langfuse provides tracing, evaluation, and monitoring for LLM applications.

> Langfuse v4.57.0 strengthens enterprise security posture with API key expiration, FIPS-mode compliant password hashing, and native inline media extraction from OTEL traces, alongside smaller reliability and performance fixes.

└──▷ **WHAT SHIPPED · 7 FEATURES** most completely described first ? what's the number?

01 FIPS compliance hardening for credential hashing NEW

55

In FIPS mode, password hashing for credential auth now uses PBKDF2 instead of bcrypt, and the system refuses to start in FIPS mode when storage options that compute MD5 are configured, enforcing FIPS compliance boundaries.

– Explains mechanism and enforcement but no exact config key v4.57.0

02 Inline media extraction from raw OTEL documents NEW

45

Langfuse now extracts inline media natively from raw OTEL documents, enabling media handling without a pre-processing step.

– Describes behaviour but no API or flag given v4.57.0

03 Bcrypt removed from new API key hashing IMPROVED

40

Langfuse stops computing bcrypt for new API keys, switching to a non-bcrypt hashing path for improved performance.

– Names the change but no detail on new hash algorithm v4.57.0

04 Worker startup jitter before queue registration IMPROVED

35

The worker now adds startup jitter before queue registration, reducing thundering-herd effects on restart.

– Explains purpose but no timing or config values v4.57.0

05 API key expiration via gateway NEW

33

API keys can now be set with a lifetime and expire via the gateway, rather than remaining valid indefinitely.

– No config key or flag named for setting expiry v4.57.0

06 **Decision model evaluators now generally available** IMPROVED

30

Decision model evaluators graduate from experimental to generally available.

— Status change stated with no further mechanism detail v4.57.0

07 **Best-effort cleanup of validated S3 files** IMPROVED

30

Adds best-effort cleanup of validated files in S3 after processing.

— Brief description with no scope or trigger detail v4.57.0

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

ToolHive

Feedback ⓘ v0.52.0 NOTES ↗ RANK

ToolHive runs MCP servers in isolated containers and enforces identity and access policies for local and Kubernetes deployments.

> ToolHive v0.52.0 strengthens session storage security with TLS support for Redis and hardens credential handling with compare-and-set updates for DCR credentials, alongside smaller token and caching improvements.

└──▷ **WHAT SHIPPED · 4 FEATURES** most completely described first ? what's the number?

01 **TLS support for Redis session storage** NEW 55

ToolHive now supports TLS-encrypted connections for Redis-backed session storage, with corresponding Kubernetes operator Helm values added to configure encrypted session backends.

— Names Helm values support but not the exact keys v0.52.0

02 **Compare-and-set updates for DCR credentials** IMPROVED 55

Dynamic Client Registration (DCR) credential updates now use compare-and-set semantics to prevent lost-update races when multiple processes update credentials concurrently.

— Explains mechanism and problem solved, no direct action v0.52.0

03 **Preserve custom claims from token introspection** IMPROVED 35

Custom claims from RFC 7662 token introspection responses are now preserved for downstream use rather than being discarded.

— States behavior change but not how claims are used v0.52.0

04 **Default one-hour prompt cache TTL for Claude Code** IMPROVED 30

Claude Code prompt cache now defaults to a one-hour TTL.

— Bare config default, no mechanism detail v0.52.0

WAS THIS USEFUL? 👍 👎

agent-safe-pipeline

Feedback ⓘ v0.3.6 NOTES ↗ RANK

Agent Safe Pipeline provides a security-focused framework for building AI agent workflows with policy checks and safety controls.

> Agent Safe Pipeline v0.3.6 introduces a hosted gateway shadow mode with response sandboxing and a TCP relay, alongside an open-host edge evaluator, AWS-identity bootstrapping for AgentCore, and the first phase of a Commerce Gate for card-purchase authorization holds.

└──▷ **WHAT SHIPPED · 4 FEATURES** most completely described first ? what's the number?

01 **Open host for Decionis edge evaluator** NEW 52

Introduces the open host for the Decionis edge evaluator, with per-replica usage-key registration (`kid` per replica), signed usage reports, and licence warnings.

– Names concrete mechanisms like `kid` per replica but no commands v0.3.6

02 **Hosted gateway shadow mode with sandboxing and relay** NEW 40

Adds a hosted gateway shadow-only mode where status visibility is operator-only, sandboxes every relayed response while in hosted mode, and adds an authenticated hosted TCP connector relay.

– Describes behaviour but no config keys or commands given v0.3.6

03 **Commerce Gate purchase authorization holds** NEW 36

Holds a card purchase's grant for the issuer's authorization step via Commerce Gate , and adds a phase 1 Commerce Gate preflight client contract.

– Names the Commerce Gate surface but lacks API or config specifics v0.3.6

04 **AgentCore bootstrap from AWS identity** NEW 23

Bootstraps `AgentCore` access from AWS identity.

– Bare one-line description with no mechanism detail v0.3.6

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

Anthropic

Feedback ⓘ changelog · Oct 12, 2026 CHANGELOG ↗ RANK

Anthropic provides Claude AI models and APIs for building applications that generate, analyze, and automate work.

> Anthropic shipped dynamic multi-agent workflows for Claude Managed Agents, new Models API capability and grouping fields, and tighter network/URL-fetch restrictions for agents, while promoting organization management out of beta across every SDK and the CLI — alongside a prompt-cache price cut for Claude Sonnet 5.5 and several breaking behavior changes in Claude Haiku 5.5.

└──▷ WHAT SHIPPED · 8 FEATURES most completely described first ? what's the number?

01 Dynamic multi-agent workflows in Claude Managed Agents NEW

90

Claude Managed Agents (beta) can now run dynamic multi-agent workflows: set `multiagent` to `{"type": "multiagent_20261001", "workflows": {"type": "enabled"}}` in the agent config to let an agent automatically spin up parallel sub-agents, with progress surfaced via `workflow_run.*` events on the session event stream.

Enable dynamic multi-agent workflows on an agent so it can automatically spin up parallel sub-agents to process hundreds of documents in phases.

```json
{
  "multiagent": {
    "type": "multiagent_20261001",
    "workflows": {"type": "enabled"}
  }
}
```

— Exact config shape and event names with a runnable example. changelog-20261012-2b4dc80b

02 Models API capability and grouping fields NEW

85

The Models API adds `capabilities.server_tools.code_execution` and `capabilities.server_tools` so callers can check per-model support for web search and code execution tools, a `line` field (e.g. `opus`) to group models without parsing model IDs, and `capabilities.thinking.types.disabled` to report whether a model accepts `thinking: {type: "disabled"}`.

Check whether a specific model supports the code execution and web search tools before routing a request to it.

```
$ curl https://api.anthropic.com/v1/models/claude-sonnet-5-5 \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  | jq '.capabilities.server_tools'
```

Group all available models by their model line without fragile string-parsing of model IDs.

```
$ curl https://api.anthropic.com/v1/models \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  | jq '[.data[] | {id, line: .line}] | group_by(.line)'
```

— Named fields with runnable curl+jq examples. changelog-20261012-2b4dc80b

---

03 **Network and URL-fetch restrictions in Claude Managed Agents**

85

BREAKING

Claude Managed Agents `limited` networking now enforces `allowed_hosts` on web search and URL-fetch tools, returning `url_not_allowed` for a host outside the list, and matches hosts exactly unless the entry starts with `*.` (so `docs.example.com` is not covered by `example.com`). The URL-fetch tool is further restricted to URLs that already appeared in session context, returning `url_not_in_prior_context` for URLs seen only in Claude output or tool results; session creation or updates with disallowed entries now fail with a 400 error.

— Exact error codes and matching rule given, no runnable example.

changelog-20261012-2b4dc80b

---

04 **Claude Haiku 5.5 breaking behavior changes** BREAKING

85

Claude Haiku 5.5 introduces several breaking changes versus Haiku 4.5: manual extended thinking, certain sampling parameters, and assistant message prefill now return a 400 error; the prior computer-use tool returns a 400 error (use the replacement toolset on the Claude API and Google Cloud, or the Bedrock variant on Amazon Bedrock); sending thinking blocks back after changing extended thinking settings or from earlier turns can return a 400 error; structured outputs are unavailable on Amazon Bedrock; and adaptive thinking is on by default, with responses able to begin with thinking blocks whose text is omitted unless `"summarized"` is set, counting as more tokens.

— Lists exact error conditions and mitigations, no runnable example.

changelog-20261012-2b4dc80b

05 **Organization management API promoted out of beta** BREAKING

80

Organization management methods (`client.organization` / `ant organization`) — covering org info, members, invites, workspaces, API keys, rate limits, service accounts, workload identity federation, customer-managed encryption keys, and compliance settings — are now out of beta across Python SDK 1.10.0, TypeScript SDK 0.130.0, C# SDK 12.52.0, Go SDK 1.77.0, Java SDK 2.67.0, PHP SDK 0.53.0, Ruby SDK 1.75.0, and CLI 1.37.0; the CLI's `ant beta:organization` namespace is renamed to `ant organization` , requiring beta callers to update.

— Exact command and SDK versions named, no runnable example. changelog-20261012-2b4dc80b

THINNER COVERAGE BELOW

06 **Prompt cache price cut for Claude Sonnet 5.5** IMPROVED

50

Prompt cache read price on Claude Sonnet 5.5 drops from $0.20 to $0.10 USD per million tokens (0.05x base input price).

— Exact prices given but no action beyond awareness. changelog-20261012-2b4dc80b

07 **Browser and desktop automation toolset SDK classes** NEW

45

Python and TypeScript SDKs add beta classes for browser and desktop automation toolsets, supporting per-tool URL and file policies and an approval callback.

— Mechanism named but no class names or example given. changelog-20261012-2b4dc80b

08 **Compliance API chat and document export additions** NEW

40

Compliance API `chat` endpoints now also return chats from the unified Claude experience (beta, for Claude Enterprise organizations) using an existing Compliance Access Key, and can download Claude Docs documents as Word files (beta).

— Named capabilities but no endpoint path or example. changelog-20261012-2b4dc80b

└──▷ **BREAKING ON UPGRADE**

! In Claude Managed Agents `limited` networking, creating a session now fails with a 400 error when an enabled web tool's URL list contains an entry not within `allowed_hosts` ; a session update adding such an entry also fails with 400.

! An `allowed_hosts` entry matches one exact host unless it starts with `*.` ; `docs.example.com` is not within `["example.com"]` , so existing configs relying on bare-domain matching will break.

- ! Code written for Claude Haiku 4.5 can break on Claude Haiku 5.5: manual extended thinking, certain sampling parameters, and assistant message prefill each return a 400 error.
- ! On Claude Haiku 5.5, the prior computer-use tool returns a 400 error — use the replacement toolset on the Claude API and Google Cloud, or the Bedrock variant on Amazon Bedrock.
- ! On Claude Haiku 5.5, a request that sends thinking blocks back after a change to extended thinking settings or from earlier turns can return a 400 error.
- ! On Amazon Bedrock, structured outputs are not available for Claude Haiku 5.5.
- ! On Claude Haiku 5.5, adaptive thinking is on by default; responses can begin with thinking blocks and their text is omitted unless you set `"summarized"`, and the same text counts as more tokens.
- ! `ant beta:organization` CLI entries are now promoted to `ant organization`; callers using the beta namespace must update to `ant organization`.

WAS THIS USEFUL? 👍 👎

Mistral AI

Feedback ⓘ snapshot-20261012 NOTES ↗ RANK

Learn how to deploy and use Mistral AI's Large Language Models with our comprehensive documentation, guides, and tutorials.

Mistral adds OCR API block-level confidence scores, inline batching, custom guardrails on agents, a Document Library API, and text-to-speech with streaming.

Adds `confidence_scores_granularity` parameter to the OCR API supporting `'block'` granularity — `'page'` returns page-level scores only, `'block'` returns page-level and block-level scores, and `'word'` returns page-level and word-level scores.

Adds `include_blocks` parameter to the OCR API: when set to `true`, each page returns a `blocks` array with paragraph-level bounding boxes and a structural label in reading order.

Adds `pages` parameter in the OCR API now also accepts a string of comma-separated digits and ranges (e.g. `'0,1,2'`, `'0-5'`, or `'0,2-4'`) in addition to a list of integers.

Adds `table_format` parameter to the OCR API, allowing choice between `markdown` and `html` for table formatting.

Adds `extract_footer`, `extract_header`, and hyperlinks output to the OCR API.

+16 more

NEW INTEGRATION Z.ai GLM 5.3 model availability

RETIRED SUPPORTED PLATFORM OCR 4.0 deprecation

RETIRED SUPPORTED PLATFORM Leanstral 1.5 deprecation

RETIRED SUPPORTED PLATFORM Z.ai GLM 5.2 deprecation

UPDATED SUPPORTED PLATFORM OCR 4.1 General Availability

NEW SUPPORTED PLATFORM Voxtral TTS model

NEW SUPPORTED PLATFORM Leanstral formal proof engineering model

NEW SUPPORTED PLATFORM Voxtral Mini Transcribe 2 and Realtime

NEW SUPPORTED PLATFORM Magistral Medium 1.2 and Small 1.2

NEW SUPPORTED PLATFORM Codestral 2508

NEW SUPPORTED PLATFORM Magistral Medium 1.1 and Small 1.1

NEW SUPPORTED PLATFORM Voxtral Small and Mini audio models

NEW SUPPORTED PLATFORM Devstral Small 1.1 and Devstral Medium

NEW SUPPORTED PLATFORM Codestral Embed

NEW SUPPORTED PLATFORM Devstral Small

NEW SUPPORTED PLATFORM Mistral OCR with document understanding

NEW SUPPORTED PLATFORM Mistral Saba

NEW SUPPORTED PLATFORM Codestral 2501

NEW SUPPORTED PLATFORM Ministral 3B and 8B

NEW SUPPORTED PLATFORM Devstral 2 and Devstral Small 2

NEW SUPPORTED PLATFORM Magistral Medium and Small (June 2025)

NEW SUPPORTED PLATFORM Mistral Medium 3 and Mistral OCR 2

NEW SUPPORTED PLATFORM Mistral Medium 3.5

└──▷ **BREAKING ON UPGRADE**

- ! The `p` parameter added to Completion API streaming chunks may break applications that rely on strict parsing of those chunks; applications using the official SDK are unaffected, but users of the `mistral-common` package must update to version 1.8.4 or higher.
- ! The `tool_call_id` field is now mandatory in chat messages with the `tool` role for function calling.
- ! The API now strictly enforces the validity of all parameters and returns `Extra inputs are not permitted` for unsupported parameters, where previously they were silently ignored.
- ! The endpoint `mistral-tiny` is renamed to `mistral-tiny-2312` and will be deprecated in three months.
- ! `mistral-ocr-4-0` is deprecated and retires on September 30, 2026.
- ! `labs-leanstral-1-5` is deprecated and retires on September 30, 2026.
- ! `zai-glm-5-2` is deprecated and retires on October 31, 2026.
- ! The temperature parameter of `pixtral-12b` was downscaled by a multiplier of 0.43, changing output behavior.
- ! Devstral 2.0 moves to paid API access.
- ! Previous `/le-chat/*` and `/mistral-vibe/*` URL paths now redirect to the new `/vibe/*` tree.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

▷ Subscribe Feedback on this issue